

L^AT_EX2_ε: Un manuel de référence officiel

Août 2022

<https://latexref.xyz/fr>

Ce document est un manuel de référence officieux pour \LaTeX , un système de préparation de documents, version « Août 2022 ».

Ce manuel a été traduit du fichier `LATEX.HLP v1.0a` de la bibliothèque d'aide VMS. La version prétraduction a été rédigée par George D. Greenwade de l'université d'état Sam Houston, au Texas (USA). La version \LaTeX 2.09 a été rédigée par Stephen Gilmore. La version $\text{\LaTeX}2e$ a été adaptée à partir de celle-ci par Torsten Martinsen. Karl Berry a fait d'autres mises à jour et ajouts, et reconnaît avec gratitude avoir utilisé comme matière de référence *Hypertext Help with \LaTeX* , de Sheldon Green, et *\LaTeX Command Summary* (pour $\text{\LaTeX}2.09$) de L. Botway et C. Biemesderfer (publié par le \TeX Users Group en tant que *\TeX niques* numéro 10). Nous savons également gré à Martin Herbert Dietze d'avoir de plus utilisé de la matière apparaissant dans son `latex2e-reference`. (À partir de ces références aucun texte n'a été copié directement).

Tous droits réservés © 2015-2022 Vincent Belaïche — traduction.

Tous droits réservés © 2007-2022 Karl Berry.

Tous droits réservés © 1988, 1994, 2007 Stephen Gilmore.

Tous droits réservés © 1994, 1995, 1996 Torsten Martinsen.

Ci-après se trouve la licence sous laquelle est distribuée ce manuel ; une traduction en est donnée en annexe, voir Annexe B [License translation], page 278.

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions.

Sommaire

1	À propos de ce document	1
2	Vue d'ensemble de L ^A T _E X	2
3	Classes de documents	7
4	Polices de caractères	17
5	Mise en page	30
6	Rubricage	42
7	Des renvois	55
8	Environnements	59
9	Saut à la ligne	111
10	Saut de page	117
11	Notes en bas de page	119
12	Définitions	124
13	Compteurs	138
14	Longueurs	141
15	Faire des paragraphes	148
16	Formules mathématiques	153
17	Modes	185
18	Styles de page	187
19	Espaces	191
20	Boîtes	204
21	Couleur	211
22	Graphiques	215
23	Insertions spéciales	228
24	Partitionner le fichier d'entrée	238
25	Parties pré/postliminaires	242
26	Des lettres	255
27	Entrée/sortie console	260
28	Interface ligne de commande	267
A	Patrons de documents	275
B	Traduction de la licence	278
	Index des concepts	279
	Index des commandes	289

Table des matières

1	À propos de ce document	1
2	Vue d'ensemble de L^AT_EX	2
2.1	Début et fin	2
2.2	Fichiers de sortie	3
2.3	Les moteurs T _E X	3
2.4	Syntaxe des commandes L ^A T _E X.....	5
2.5	Les environnements.....	5
2.6	CTAN : The Comprehensive T _E X Archive Network.....	6
3	Classes de documents	7
3.1	Options de classe de document.....	7
3.2	Ajout de paquetages.....	9
3.3	Construction des extensions (classes et paquetages)	9
3.3.1	Structure d'une extension (classe ou paquetage).....	9
3.3.2	Commande pour extension (classe ou paquetage).....	10
4	Polices de caractères	17
4.1	Le paquetage fontenc.....	17
4.1.1	<code>\DeclareFontEncoding</code>	19
4.1.2	<code>\DeclareTextAccent</code>	19
4.1.3	<code>\DeclareTextAccentDefault</code>	19
4.1.4	<code>\DeclareTextCommand</code> & <code>\ProvideTextCommand</code>	20
4.1.5	<code>\DeclareTextCommandDefault</code> & <code>\ProvideTextCommandDefault</code>	21
4.1.6	<code>\DeclareTextComposite</code>	21
4.1.7	<code>\DeclareTextCompositeCommand</code>	22
4.1.8	<code>\DeclareTextSymbol</code>	22
4.1.9	<code>\DeclareTextSymbolDefault</code>	22
4.1.10	<code>\LastDeclaredEncoding</code>	23
4.1.11	<code>\UseTextSymbol</code> & <code>\UseTextAccent</code>	23
4.2	styles des polices	23
4.3	tailles des polices.....	26
4.4	commandes de fontes de bas niveau.....	27
5	Mise en page	30
5.1	<code>\onecolumn</code>	30
5.2	<code>\twocolumn</code>	30
5.3	<code>\flushbottom</code>	32
5.4	<code>\raggedbottom</code>	32
5.5	Paramètres de mise en page	32

5.6	<code>\baselineskip</code> & <code>\baselinestretch</code>	36
5.7	Flottants	37
5.7.1	<code>\caption</code>	40
6	Rubricage	42
6.1	<code>\part</code>	43
6.2	<code>\chapter</code>	44
6.3	<code>\section</code>	46
6.4	<code>\subsection</code>	47
6.5	<code>\subsubsection</code> , <code>\paragraph</code> , <code>\subparagraph</code>	48
6.6	<code>\appendix</code>	49
6.7	<code>\frontmatter</code> , <code>\mainmatter</code> , <code>\backmatter</code>	50
6.8	<code>\@startsection</code> , composer les rubriques	51
7	Des renvois	55
7.1	<code>\label</code>	55
7.2	<code>\pageref{clef}</code>	56
7.3	<code>\ref{clef}</code>	57
7.4	Le paquetage <code>xr</code>	57
8	Environnements	59
8.1	<code>abstract</code>	59
8.2	<code>array</code>	60
8.3	<code>center</code>	61
8.3.1	<code>\centering</code>	62
8.4	<code>description</code>	63
8.5	<code>displaymath</code>	63
8.6	<code>document</code>	64
8.7	<code>enumerate</code>	65
8.8	<code>eqnarray</code>	66
8.9	<code>equation</code>	66
8.10	<code>figure</code>	67
8.11	<code>filecontents</code> : Écrire un fichier externe	68
8.12	<code>flushleft</code>	69
8.12.1	<code>\raggedright</code>	69
8.13	<code>flushright</code>	70
8.13.1	<code>\raggedleft</code>	70
8.14	<code>itemize</code>	71
8.15	environnement <code>letter</code> : écrire des lettres postales	72
8.16	<code>list</code>	72
8.16.1	<code>\item</code> : Une entrée dans une liste	77
8.16.2	<code>trivlist</code> : Une forme restreinte de <code>list</code>	78
8.17	<code>math</code>	78
8.18	<code>minipage</code>	79
8.19	<code>picture</code>	81
8.19.1	<code>\put</code>	83
8.19.2	<code>\multiput</code>	84

8.19.3	<code>\qbezier</code>	84
8.19.4	<code>\graphpaper</code>	85
8.19.5	<code>\line</code>	85
8.19.6	<code>\linethickness</code>	86
8.19.7	<code>\thinlines</code>	86
8.19.8	<code>\thicklines</code>	86
8.19.9	<code>\circle</code>	86
8.19.10	<code>\oval</code>	87
8.19.11	<code>\shortstack</code>	87
8.19.12	<code>\vector</code>	88
8.19.13	<code>\makebox</code>	88
8.19.14	<code>\framebox</code>	88
8.19.15	<code>\frame</code>	89
8.19.16	<code>\dashbox</code>	89
8.20	quotation et quote.....	89
8.21	tabbing.....	90
8.22	table.....	96
8.23	tabular.....	97
8.23.1	<code>\multicolumn</code>	100
8.23.2	<code>\vline</code>	102
8.23.3	<code>\cline</code>	102
8.23.4	<code>\hline</code>	102
8.24	thebibliography.....	102
8.24.1	<code>\bibitem</code>	103
8.24.2	<code>\cite</code>	104
8.24.3	<code>\nocite</code>	105
8.24.4	Utilisation de Bib \TeX	105
8.24.4.1	Messages d'erreur de Bib \TeX	106
8.25	theorem.....	107
8.26	titlepage.....	107
8.27	verbatim.....	108
8.27.1	<code>\verb</code>	109
8.28	verse.....	109

9 Saut à la ligne..... 111

9.1	<code>\</code>	111
9.2	<code>\obeycr</code> & <code>\restorecr</code>	112
9.3	<code>\newline</code>	113
9.4	<code>\-</code> (césure à gré).....	113
9.5	<code>\discretionary</code> (point de césure généralisé).....	114
9.6	<code>\fussy</code> & <code>\sloppy</code>	115
9.6.1	<code>sloppypar</code>	115
9.7	<code>\hyphenation</code>	115
9.8	<code>\linebreak</code> & <code>\nolinebreak</code>	116

10	Saut de page	117
10.1	<code>\clearpage</code> & <code>\cleardoublepage</code>	117
10.2	<code>\newpage</code>	117
10.3	<code>\enlargethispage</code>	118
10.4	<code>\pagebreak</code> & <code>\nopagebreak</code>	118
11	Notes en bas de page	119
11.1	<code>\footnote</code>	119
11.2	<code>\footnotemark</code>	120
11.3	<code>\footnotetext</code>	120
11.4	Note en bas de page dont le renvoi est au sein d'un titre de rubrique	120
11.5	Notes en bas de page dans un tableau	121
11.6	Notes en bas de page sur d'autres notes	122
11.7	Paramètres des notes en bas de page	122
12	Définitions	124
12.1	<code>\newcommand</code> & <code>\renewcommand</code>	124
12.1.1	Séquence de contrôle, mot de contrôle et symbole de contrôle	126
12.2	<code>\providecommand</code>	127
12.3	<code>\makeatletter</code> et <code>\makeatother</code>	127
12.4	<code>\@ifstar</code>	128
12.5	<code>\newcounter</code> : allouer un compteur	129
12.6	<code>\newlength</code>	129
12.7	<code>\newsavebox</code> : allouer une boîte	130
12.8	<code>\newenvironment</code> & <code>\renewenvironment</code>	130
12.9	<code>\newtheorem</code>	132
12.10	<code>\newfont</code> : définit une nouvelle police (obsolète)	134
12.11	<code>\protect</code>	134
12.12	<code>\ignorespaces</code> & <code>\ignorespacesafterend</code>	135
12.13	<code>xspace</code> package	136
13	Compteurs	138
13.1	<code>\alph</code> <code>\Alph</code> <code>\arabic</code> <code>\roman</code> <code>\Roman</code> <code>\fnsymbol</code> : Taper des compteurs	138
13.2	<code>\usecounter{compteur}</code>	138
13.3	<code>\value{compteur}</code>	139
13.4	<code>\setcounter{compteur}{value}</code>	139
13.5	<code>\addtocounter{compteur}{valeur}</code>	139
13.6	<code>\refstepcounter{compteur}</code>	139
13.7	<code>\stepcounter{compteur}</code>	139
13.8	<code>\day</code> & <code>\month</code> & <code>\year</code>	140

14	Longueurs	141
14.1	Unités de longueur	142
14.2	<code>\setlength</code>	143
14.3	<code>\addtolength</code>	144
14.4	<code>\settodepth</code>	144
14.5	<code>\settoheight</code>	145
14.6	<code>\settowidth</code>	145
14.7	Expressions	145
15	Faire des paragraphes	148
15.1	<code>\par</code>	149
15.2	<code>\indent</code> & <code>\noindent</code>	150
15.3	<code>\parindent</code> & <code>\parskip</code>	150
15.4	Notes en marge	151
16	Formules mathématiques	153
16.1	Indices inférieurs & supérieurs	154
16.2	Symboles mathématiques	155
16.2.1	Flèches	166
16.2.2	<code>\boldmath</code> & <code>\unboldmath</code>	167
16.2.3	Pseudo-gras avec des lettres évidées	168
16.2.4	Math Calligraphic	168
16.2.5	Délimiteurs	168
16.2.5.1	<code>\left</code> & <code>\right</code>	169
16.2.5.2	<code>\bigl</code> , <code>\bigr</code> , etc.	171
16.2.6	Points de suspensions, horizontaux ou verticaux	173
16.2.7	Lettres grecques	174
16.3	Fonctions mathématiques	174
16.4	Accents mathématiques	176
16.4.1	<code>\accent</code>	176
16.5	Sur- et sous-ligner	177
16.6	Espacement en mode mathématique	178
16.6.1	<code>\smash</code>	179
16.6.2	<code>\phantom</code> & <code>\vphantom</code> & <code>\hphantom</code>	180
16.6.3	<code>\mathstrut</code>	181
16.7	Styles mathématiques	181
16.8	Recueil de diverses commandes utiles en math	183
16.8.1	Caractère deux-points : & <code>\colon</code>	183
16.8.2	<code>*</code>	183
16.8.3	<code>\frac</code>	183
16.8.4	<code>\sqrt</code>	184
16.8.5	<code>\stackrel</code>	184
17	Modes	185
17.1	<code>\ensuremath</code>	186

18	Styles de page	187
18.1	<code>\maketitle</code>	187
18.2	<code>\pagenumbering</code>	188
18.3	<code>\pagestyle</code>	189
18.4	<code>\thispagestyle{style}</code>	190
19	Espaces	191
19.1	<code>\enspace</code> & <code>\quad</code> & <code>\qquad</code>	191
19.2	<code>\hspace</code>	191
19.3	<code>\hfill</code>	192
19.4	<code>\hss</code>	192
19.5	<code>\spacefactor</code>	192
19.5.1	<code>\@</code>	193
19.5.2	<code>\frenchspacing</code> & <code>\nonfrenchspacing</code>	194
19.5.3	<code>\normalsfcodes</code>	195
19.6	Controblique-espace <code>\</code>	195
19.7	<code>~</code>	196
19.8	<code>\thinspace</code> & <code>\negthinspace</code>	197
19.9	<code>\/</code>	197
19.10	<code>\hrulefill</code> & <code>\dotfill</code>	198
19.11	<code>\bigskip</code> & <code>\medskip</code> & <code>\smallskip</code>	199
19.12	<code>\bigbreak</code> & <code>\medbreak</code> & <code>\smallbreak</code>	199
19.13	<code>\strut</code>	200
19.14	<code>\vspace</code>	201
19.15	<code>\vfill</code>	202
19.16	<code>\addvspace</code>	203
20	Boîtes	204
20.1	<code>\mbox</code> & <code>\makebox</code>	204
20.2	<code>\fbox</code> & <code>\framebox</code>	205
20.3	<code>\parbox</code>	206
20.4	<code>\raisebox</code>	207
20.5	<code>\sbox</code> & <code>\savebox</code>	208
20.6	<code>lrbox</code>	210
20.7	<code>\usebox</code>	210
21	Couleur	211
21.1	Options du paquetage <code>color</code>	211
21.2	Modèles de couleur	211
21.3	Commandes pour <code>color</code>	212
21.3.1	Définir des couleurs	212
21.3.2	Mettre du texte en couleur	213
21.3.3	Faire des boîtes en couleur	214
21.3.4	Faire des pages en couleur	214

22	Graphiques	215
22.1	Options du paquetage <code>graphics</code>	215
22.2	Configuration du paquetage <code>graphics</code>	216
22.2.1	<code>\graphicspath</code>	217
22.2.2	<code>\DeclareGraphicsExtensions</code>	218
22.2.3	<code>\DeclareGraphicsRule</code>	218
22.3	Commandes du paquetage <code>graphics</code>	219
22.3.1	<code>\includegraphics</code>	220
22.3.2	<code>\rotatebox</code>	225
22.3.3	<code>\scalebox</code>	226
22.3.4	<code>\resizebox</code>	226
23	Insertions spéciales	228
23.1	Caractères réservés.....	228
23.2	Capitales et bas de casse.....	228
23.3	Symboles appelés par leur position dans une police.....	229
23.4	Symboles en texte.....	229
23.5	Accents.....	233
23.6	Lettres latines supplémentaires.....	234
23.7	paquetage <code>inputenc</code>	235
23.8	<code>\rule</code>	236
23.9	<code>\today</code>	236
24	Partitionner le fichier d'entrée	238
24.1	<code>\endinput</code>	238
24.2	<code>\include</code> & <code>\includeonly</code>	239
24.3	<code>\input</code>	241
25	Parties pré/postliminaires	242
25.1	Table des matières etc.....	242
25.1.1	<code>\@dottedtocline</code>	243
25.1.2	<code>\addcontentsline</code>	244
25.1.3	<code>\addtocontents</code>	245
25.1.4	<code>\contentsline</code>	246
25.1.5	<code>\nofiles</code>	247
25.1.6	<code>\numberline</code>	247
25.2	Glossaires.....	247
25.3	Index.....	248
25.3.1	Produire l'index manuellement.....	249
25.3.2	<code>\index</code>	249
25.3.3	<code>makeindex</code>	251
25.3.4	<code>\printindex</code>	254

26	Des lettres	255
26.1	<code>\address</code>	256
26.2	<code>\cc</code>	256
26.3	<code>\closing</code>	257
26.4	<code>\encl</code>	257
26.5	<code>\location</code>	257
26.6	<code>\makelabels</code>	257
26.7	<code>\name</code>	258
26.8	<code>\opening{texte}</code>	258
26.9	<code>\ps</code>	258
26.10	<code>\signature{texte}</code>	258
26.11	<code>\telephone</code>	259
27	Entrée/sortie console	260
27.1	<code>\openin</code> & <code>\openout</code>	260
27.2	<code>\read</code>	261
27.3	<code>\typein[cmd]{msg}</code>	261
27.4	<code>\typeout{msg}</code>	261
27.5	<code>\write</code>	262
27.5.1	<code>\write</code> et la sécurité	263
27.5.2	<code>\message</code>	264
27.5.3	<code>\wlog</code>	265
27.5.4	<code>\write18</code>	265
28	Interface ligne de commande	267
28.1	Options passées à la ligne de commande	267
28.2	Spécifier du code LaTeX sur la ligne de commande	269
28.3	Nom d'ouvrage de T _E X	270
28.4	Quand quelque chose se passe mal	273
Annexe A	Patrons de documents	275
A.1	Un patron <code>beamer</code>	275
A.2	Patron <code>article</code>	275
A.3	Un patron <code>book</code>	276
A.4	Un patron <code>book</code> plus élaboré	276
Annexe B	Traduction de la licence	278
	Index des concepts	279
	Index des commandes	289

1 À propos de ce document

Ceci est un manuel de référence officieux pour le système $\text{\LaTeX}2\text{e}$ de préparation de document. Ce système est mis en œuvre sous forme d'un paquet de macros pour le programme de composition \TeX (voir Chapitre 2 [Overview], page 2). La page d'accueil de ce document est <https://latexref.xyz>. Cette page contient des liens vers différentes versions de ce document, dans différents formats et langues, mais aussi vers ses sources, vers les archives des listes de diffusion qui lui sont consacrées ainsi que vers d'autres infrastructures.

Dans ce document, on utilise principalement juste le terme « \LaTeX » plutôt que « $\text{\LaTeX}2\text{e}$ », étant donné que la version précédente de \LaTeX (2.09) est gelée depuis des dizaines d'années.

\LaTeX est désormais maintenu par un groupe de bénévoles (<http://latex-project.org>). La documentation officielle écrite par le projet \LaTeX est disponible à partir de leur site web. Le présent document est complètement officieux et n'a pas été examiné par les mainteneurs de \LaTeX . Ne leur envoyez donc pas de rapports d'anomalie ou quoi que ce soit d'autre.

En revanche, n'hésitez pas à adresser vos commentaires à latexrefman@tug.org.

Ce document est un manuel de référence. Il y a d'autres sources d'information sur \LaTeX : elles sont très nombreuses, et s'adressent à des utilisateurs de tous niveaux. Voici quelques introductions :

<http://ctan.org/pkg/latex-doc-ptr>

Deux pages de références recommandées à \LaTeX documentation.

<http://ctan.org/pkg/first-latex-doc>

Rédaction de votre premier document, avec un peu de texte et de mathématiques.

<http://ctan.org/pkg/usrguide>

Le guide pour les auteurs de documents maintenu par le projet \LaTeX . Bien d'autres guides écrits par bien d'autres gens sont aussi disponibles, indépendamment du projet \LaTeX ; l'article suivant en est un.

<http://ctan.org/pkg/lshort>

Une courte (?) introduction à \LaTeX , traduite en de nombreuses langues (traduction française : <http://ctan.org/pkg/lshort-french>).

<http://tug.org/begin.html>

Introduction au système \TeX , y compris \LaTeX , et plus amples références.

2 Vue d'ensemble de L^AT_EX

Qu'est-ce que L^AT_EX?

L^AT_EX est un système de composition de documents. Il fut à l'origine créé par Leslie Lamport ; il est désormais maintenu par un groupe de volontaires (<http://latex-project.org>). Il est largement utilisé, en particulier pour les documents complexes et techniques, tels que ceux impliquant des mathématiques.

À l'aide d'un simple éditeur de texte, un utilisateur L^AT_EX écrit un fichier, dit *fichier d'entrée*, qui contient son texte ; des *commandes* y sont intercalées : ce sont des instructions servant à décrire la manière dont le texte doit être formaté. En effet, L^AT_EX est mis en œuvre comme un ensemble de commandes liées s'interfaçant avec le programme de composition T_EX de Donald E. Knuth (le terme technique est que L^AT_EX est un *paquet de macros* pour le moteur T_EX). En donnant le fichier d'entrée au moteur T_EX, l'utilisateur produit le *document de sortie* qui contient son texte mis en page.

Le terme L^AT_EX est aussi parfois utilisé pour signifier le langage à balises dans lequel le code source du document est écrit, c'est-à-dire pour signifier l'ensemble des commandes à la disposition d'un utilisateur de L^AT_EX.

Le nom L^AT_EX est l'abréviation de "Lamport T_EX". On le prononce LA-TÈQUE. Au sein du document, on produit le logo avec `\LaTeX`. Là où l'utilisation du logo n'est pas raisonnable, comme dans du texte brut, écrivez le 'LaTeX'.

2.1 Début et fin

Les fichiers L^AT_EX ont une structure globale simple, avec un début et une fin standards. Voici un petit exemple :

```
\documentclass{article}
\begin{document}
Bonjour le monde \LaTeX.
\end{document}
```

Tout document L^AT_EX a une ligne `\begin{document}` et une ligne `\end{document}`.

Ici 'article' est ce que l'on appelle la *classe de document*. Elle est implémentée dans un fichier `article.cls`. N'importe quelle classe de document de votre système peut être utilisée. Quelques classes de document sont définies par L^AT_EX lui-même, et un grand nombre d'autres sont largement disponibles. Voir Chapitre 3 [Document classes], page 7.

Vous pouvez inclure d'autres commandes L^AT_EX entre les commandes `\documentclass` et `\begin{document}`. Cette zone est appelée la *préambule*.

La paire `\begin{document}`, `\end{document}` définit un *environnement* ; l'environnement 'document' (et aucun autre) est obligatoire dans tous les documents L^AT_EX (voir Section 8.6 [document], page 64). L^AT_EX vous fournit beaucoup d'environnements documentés ici (voir Chapitre 8 [Environments], page 59). Vous pouvez en trouver bien plus encore à partir de paquetages externes, notamment ceux disponibles sur le CTAN (voir Section 2.6 [CTAN], page 6).

Les rubriques suivantes sont consacrées aux façons de produire, à partir d'un fichier d'entrée L^AT_EX, des documents PDF ou d'autres formats de sortie.

2.2 Fichiers de sortie

L^AT_EX produit un fichier de sortie principal et au moins deux fichiers auxiliaires. Le nom du fichier de sortie principal se termine soit en `.dvi` soit en `.pdf`.

- `.dvi` S'il est invoqué avec la commande système `latex`, alors il produit un fichier « DeVice Independent¹ » (`.dvi`). Vous pouvez visualiser ce fichier avec une commande comme `xdvi`, ou le convertir en fichier PostScript `.ps` avec `dvips` ou en fichier « Portable Document Format² » `.pdf` avec `dvipdfmx`. Un grand nombre d'autres programmes utilitaires pour fichiers DVI sont disponibles (<http://mirror.ctan.org/dviware>).
- `.pdf` Si L^AT_EX est invoqué avec la commande système `pdflatex`, alors la sortie principale est un fichier « Portable Document Format » (`.pdf`). Typiquement, il s'agit d'un fichier autonome, avec toutes les polices et images incorporées. D'autres commandes (voir Section 2.3 [T_EX engines], page 3) produisent des fichiers PDF.

L^AT_EX produit toujours au moins deux fichiers supplémentaires.

- `.log` Ce fichier de transcription, ou *fichier journal*, contient des informations sommaires telles que la liste des paquetages chargés. Il contient aussi des messages de diagnostic et, possiblement, des informations supplémentaires concernant toutes erreurs.
- `.aux` De l'information auxiliaire est utilisée par L^AT_EX pour des choses telles que les doubles renvois. Par exemple, la première fois que L^AT_EX trouve un renvoi en aval — une double référence à quelque chose qui n'est pas encore apparu dans le code source — il apparaîtra dans la sortie comme un double point d'interrogation `??`. Quand l'endroit auquel le renvoi fait référence finit par apparaître dans le code source, alors L^AT_EX écrit son information de localisation dans ce fichier `.aux`. À l'invocation suivante, L^AT_EX lit l'information de localisation à partir de ce fichier et l'utilise pour résoudre le renvoi, en remplaçant le double point d'interrogation avec la localisation mémorisée.

L^AT_EX peut produire encore d'autres fichiers, caractérisés par la terminaison du nom de fichier. Ceux-ci incluent un fichier `.lof` qui est utilisé pour fabriquer la liste des figures, un fichier `.lot` utilisé pour fabriquer une liste des tableaux, et un fichier `.toc` utilisé pour fabriquer une table des matières (voir Section 25.1 [Table of contents etc.], page 242). Une classe de document particulière peut en créer d'autres ; cette liste n'a pas de fin définie.

2.3 Les moteurs T_EX

L^AT_EX est un vaste ensemble de commandes qui sont exécuté par un programme T_EX (voir Chapitre 2 [Overview], page 2). Un tel ensemble de commande est appelé un *format*, et peut être matérialisé dans un fichier binaire `.fmt`, ce qui permet qu'il soit lu bien plus rapidement que le source T_EX correspondant.

Cette rubrique donne une brève vue d'ensemble des programmes T_EX les plus répandus (voir aussi Chapitre 28 [Command line interface], page 267).

¹ Indépendant du périphérique de sortie, ce n'est toutefois pas un format portable de document

² Format portable de document

`latex`

`pdflatex` Dans T_EX Live (<http://tug.org/texlive>, si L^AT_EX est invoqué avec les commandes système `latex` ou `pdflatex`, alors le moteur pdfT_EX est exécuté (<http://ctan.org/pkg/pdftex>). Selon qu'on invoque `latex` ou `pdflatex`, la sortie principale est respectivement un fichier `.dvi` ou un fichier `.pdf`.

pdfT_EX incorpore les extensions qu'e-T_EX apporte au programme original de Knuth (<http://ctan.org/pkg/etex>), ce qui inclut des caractéristiques supplémentaires de programmation et la composition bidirectionnelle, et a lui-même de nombreuses extensions. e-T_EX est lui-même disponible par la commande système `etex`, mais le langage du fichier d'entrée est du T_EX de base (et le fichier produit est un `.dvi`).

Dans d'autres distributions T_EX, `latex` peut invoquer e-T_EX plutôt que pdfT_EX. Dans tous les cas, on peut faire l'hypothèse que les extensions e-T_EX sont disponibles en L^AT_EX.

`lualatex` Si L^AT_EX est invoqué avec la commandes systèmes `lualatex`, alors le moteur LuaT_EX est exécuté (<http://ctan.org/pkg/lualatex>). Ce programme permet que du code écrit dans le langage script Lua (<http://luatex.org>) interagisse avec la composition faite par T_EX. LuaT_EX traite nativement l'entrée en codage Unicode UTF-8, peut traiter les polices OpenType et TrueType, et produit un fichier `.pdf` par défaut. Il y a aussi `dvilualatex` pour produire un fichier `.dvi`.

`xelatex` Si L^AT_EX est invoqué avec la commandes système `xelatex`, le moteur XeT_EX est exécuté (<http://tug.org/xetex>). Comme LuaT_EX, XeT_EX prend en charge nativement le codage Unicode UTF-8 et les polices TrueType et OpenType, bien que leur implémentation soit complètement différente, en utilisant principalement des bibliothèques externes plutôt que du code interne. XeT_EX produit un fichier `.pdf` en sortie ; il ne prend pas en charge la sortie DVI.

En interne, XeT_EX crée un fichier `.xdv` file, une variante de DVI, et traduit cela en PDF en utilisant le programme `(x)dvipdfmx`, mais ce processus est automatique. Le fichier `.xdv` n'est utile que pour le débogage.

`platex`

`uplatex` Ces commandes fournissent une prise en charge significativement plus riche du japonais et d'autres langues ; la variante en `u` prend en charge l'Unicode. Voir <https://ctan.org/pkg/ptex> et <https://ctan.org/pkg/uptex>.

Depuis 2019, il y a une commande compagne en `-dev` et un format pour tous les moteurs cités plus hauts :

```
dvilualatex-dev
latex-dev
lualatex-dev
pdflatex-dev
platex-dev
uplatex-dev
xelatex-dev
```

Ce sont les candidats pour une parution de L^AT_EX à venir. L'usage principal est de découvrir et traiter les problèmes de compatibilité avant la parution officielle.

Ces formats en `-dev` rendent facile pour quiconque d'aider à l'essai des documents et du code : vous pouvez exécuter, disons, `pdflatex-dev` au lieu de `pdflatex`, sans changer quoi que ce soit d'autre dans votre environnement. En réalité, le plus facile et utile est de toujours d'exécuter les versions en `-dev` au lieu de s'embêter à basculer entre les deux. Pendant les périodes calmes après une parution, ces commandes seront équivalentes.

Ce ne sont pas des échantillons quotidiens de code de développement non testé. L'équipe L^AT_EX fait subir à ces commandes les mêmes essais étendus de non régression avant leur parution.

Pour plus ample information, voir *The L^AT_EX release workflow and the L^AT_EX dev formats* par Frank Mittelbach, TUGboat 40:2, <https://tug.org/TUGboat/tb40-2/tb125mitt-dev.pdf>.

2.4 Syntaxe des commandes L^AT_EX

Dans le fichier d'entrée L^AT_EX, un nom de commande commence avec une controbrique, `\`. Le nom lui-même consiste soit en (a) une chaîne de lettres ou (b) un unique caractère, qui dans ce cas n'est pas une lettre.

Les noms de commandes L^AT_EX sont sensibles à la casse de sorte que `\pagebreak` diffère de `\Pagebreak` (ce dernier n'est pas une commande standard). La plupart des noms de commandes sont en bas de casse, mais quoi qu'il en soit vous devez saisir toute commande en respectant la casse dans laquelle elle est définie.

Une commande peut être suivie de zéro, un ou plus d'arguments. Ces arguments peuvent être soit obligatoires, soit optionnels. Les arguments obligatoires sont entre accolades, `{...}`. Les arguments optionnels sont entre crochets, `[...]`. En général, mais ce n'est pas universel, si la commande prend un argument optionnel, il vient en premier, avant tout argument obligatoire.

Au sein d'un argument optionnel, pour utiliser le crochet fermant `)` cachez le au sein d'accolades, comme dans `\item[crochet fermant {}]`. De même, si un argument optionnel vient en dernier, sans argument obligatoire à sa suite, alors pour que le premier caractère dans le texte suivant soit un crochet ouvrant, cachez le entre accolades.

L^AT_EX obéit à la convention selon laquelle certaines commandes ont une forme étoilée `*` qui est en relation avec la forme sans le `*`, telles que `\chapter` et `\chapter*`. La différence exacte de comportement dépend de la commande.

Ce manuel décrit toutes les options acceptées et les formes en `*` pour les commandes dont il traite (à l'exception des omissions involontaires ou bogues de ce manuel).

2.5 Les environnements

Synopsis :

```
\begin{nom-environnement}
...
\end{nom-environnement}
```

Un *environnement* est une zone du code source L^AT_EX, au sein de laquelle il y a un comportement différent. Par exemple, pour la poésie en L^AT_EX mettez les lignes entre `\begin{verse}` et `\end{verse}`.

```

\begin{verse}
  Plus ne suis ce que j'ai été \\
  ...
\end{verse}

```

Voir Chapitre 8 [Environments], page 59, pour une liste d'environnements. En particulier, il est à noter que tout document L^AT_EX doit avoir un environnement `document` : une paire `\begin{document} ... \end{document}`.

Le *nom-environnement* du début doit s'accorder exactement avec celui de la fin. Cela vaut aussi dans le cas où *nom-environnement* se termine par une étoile (*) ; l'argument à la fois de `\begin` et `\end` doit comprendre l'étoile.

Les environnements peuvent avoir des arguments, y compris des arguments optionnels. L'exemple ci-dessous produit un tableau. Le premier argument est optionnel (il aligne verticalement le tableau sur sa première ligne) alors que le second argument est obligatoire (il spécifie le format des colonnes du tableau).

```

\begin{tabular}[t]{r|l}
  ... lignes du tableau ...
\end{tabular}

```

2.6 CTAN : The Comprehensive T_EX Archive Network

Le «Comprehensive T_EX Archive Network», ou CTAN, est le dépôt de matière libre de la communauté T_EX et L^AT_EX. C'est un ensemble de sites Internet répartis dans le monde qui offre de télécharger de la matière en relation avec L^AT_EX. Visitez le CTAN sur la Toile à l'adresse <https://ctan.org>.

Cette matière est organisée en paquetages, des paquets discrets qui offrent typiquement une fonctionnalité cohérente et sont maintenus par une personne ou un petit nombre de personnes. Par exemple, beaucoup d'organismes d'édition ont un paquetage qui permet aux auteurs de formater leurs articles conformément aux spécifications de cet organisme.

En plus des ressources massives, le site ctan.org offre des fonctions telles que la recherche par nom ou par fonctionnalité.

CTAN n'est pas un site unique, mais au lieu de cela un ensemble de sites. L'un des ces sites est le cœur. Ce site gère activement la matière, par exemple en acceptant les téléversements de matière nouvelle ou mise à jour. Il est hébergé par le groupe T_EX allemand DANTE e.V.

D'autres sites autour du monde apportent leur aide en miroitant, c'est à dire en synchronisant automatiquement leurs collections sur le site maître et en rendant à leur tour leurs copies publiquement disponibles. Ceci donne aux utilisateurs géographiquement proches un meilleur accès et allège la charge sur le site maître. La liste des miroirs se trouve à <https://ctan.org/mirrors>.

3 Classes de documents

La classe globale d'un document est définie avec cette commande, c'est normalement la première commande dans un fichier source \LaTeX .

```
\documentclass[options]{classe}
```

La commande `\documentclass` doit être la première commande dans un fichier source \LaTeX .

Les noms de *classe* de document incorporés à \LaTeX . (Beaucoup de classes de document sont disponibles dans des paquetages séparés ; voir Chapitre 2 [Overview], page 2.)

article	Pour un article de journal, une présentation, et diverses utilisations générales.
book	Des livres complets, y compris des chapitres et possiblement incluant des pièces préliminaires telles qu'une préface, et postliminaires telles qu'un appendice (voir Chapitre 25 [Front/back matter], page 242).
letter	Courrier, incluant optionnellement des étiquettes pour enveloppes (voir Chapitre 26 [Letters], page 255).
report	Pour des documents de longueur entre celle d'un article et celle d'un book , tels que des rapports techniques ou des thèses, et qui peuvent contenir plusieurs chapitres.
slides	Pour des présentations de diapositive — rarement utilisé de nos jours. À sa place la classe beamer est sans doute celle dont l'usage est le plus répandu (https://ctan.org/pkg/beamer). Voir Section A.1 [beamer template], page 275, pour un petit patron de document beamer .

Les *options* standardes sont décrites dans les rubriques suivantes.

3.1 Options de classe de document

Vous pouvez spécifier ce qu'on appelle des *options globales* ou des *options de classe* en les passant entre crochet à la commande `\documentclass`, comme d'habitude. Pour spécifier plus d'une *option*, séparez les par une virgule :

```
\documentclass [option1, option2, ...]{classe}
```

Voici la liste des options standardes de classe.

Toutes les classes standardes, sauf **slides** acceptent les options suivantes pour sélectionner la taille de police de caractères (l'option par défaut est 10pt):

```
12pt 11pt 10pt
```

Toutes les classes standardes acceptent ces options de sélection de la taille du papier (ci-après hauteur sur largeur) :

a4paper	210 sur 297 mm (environ 8,25 sur 11,75 pouces)
a5paper	148 sur 210 mm (environ 5,8 sur 8,3 pouces)
b5paper	176 sur 250 mm (environ 6,9 sur 9,8 pouces)

`executivepaper`

7,25 sur 10,5 pouces

`legalpaper`

8,5 sur 14 pouces

`letterpaper`

8,5 sur 11 pouces (l'option par défaut)

Lorsque on utilise l'un des moteurs pdfL^AT_EX, LuaL^AT_EX, ou XeL^AT_EX (voir Section 2.3 [T_EX engines], page 3), les options autres que `letterpaper` règlent la zone d'impression mais vous devez également régler la taille du papier physique. Une méthode pour faire cela est de placer `\pdfpagewidth=\paperwidth` et `\pdfpageheight=\paperheight` dans le préambule de votre document. Le paquetage `geometry` fournit des méthodes flexibles pour régler la zone d'impression et la taille du papier physique.

Diverses autres options:

`draft`, `final`

Pour marquer/ne marquer pas les boîtes trop pleines avec une grande boîte noire ; l'option par défaut est `final`.

`fleqn`

Pour aligner à gauches les formules hors texte ; par défaut elles sont centrées.

`landscape`

Pour sélectionner le format de page à l'italienne ; l'option par défaut est à la française.

`leqno`

Pour Mettre les numéros d'équation sur le côté gauche des équations ; par défaut ils sont sur le côté droit.

`openbib`

Pour utiliser le format bibliographie `openbib`.

`titlepage`, `notitlepage`

Indique si la page de titre est séparée ; l'option par défaut dépend de la classe.

Ces options ne sont pas disponibles avec la classe `slides` :

`onecolumn`

`twocolumn`

Composer en une ou deux colonnes ; le défaut est `onecolumn`.

`oneside`

`twoside`

Sélectionne la disposition en recto simple ou recto-verso ; le défaut est `oneside` pour recto simple, sauf pour la classe `book`.

Le paramètre `\evensidemargin` (`\oddsidemargin`) détermine la distance sur les pages de numéro pair (impair) entre le côté gauche de la page et la marge gauche du texte. Voir [page layout parameters `oddsidemargin`], page 33.

`openright`

`openany`

Détermine si un chapitre doit commencer sur une page de droite ; défaut est `openright` pour la classe `book`.

La classe `slide` offre l'option `clock` pour l'impression du temps au bas de chaque note.

3.2 Ajout de paquetages

Les paquetages ajoutés *paquo* sont chargés comme ceci :

```
\usepackage[options]{paquo}
```

Pour spécifier plus d'un paquetage, vous pouvez les séparer par une virgule comme dans `\usepackage{paquo1,paquo2,...}`, ou utiliser plusieurs commandes `\usepackage`.

Toutes les options indiquées dans la commande `\documentclass` qui sont inconnues de la classe de document sélectionnée sont transmises aux paquetages chargés par `\usepackage`.

3.3 Construction des extensions (classes et paquetages)

Vous pouvez créer de nouvelles classes de document, et de nouveaux paquetages. Par exemple, si vos notes doivent répondre à des exigences locales, telles qu'un en-tête standard pour chaque page, alors vous pourriez créer une nouvelle classe `cmsnote.cls` et commencer vos documents par `\documentclass{cmsnote}`.

Ce qui distingue un paquetage d'une classe de document c'est que les commandes d'un paquetage sont utilisables pour différentes classes alors que celles dans une classes de document sont spécifiques à cette classes. Ainsi, une commande qui permet de régler les en-têtes de pages irait dans un paquetage alors qu'une commande intitulant en-têtes de pages par *Note du service de mathématique de la CMS* irait dans une classe.

Au sein d'un fichier classe ou paquetage on peut utiliser l'arobe @ comme un caractère dans les noms de commande sans avoir à entourer le code contenant la commande en question par `\makeatletter` et `\makeatother`. Voir Section 12.3 [`\makeatletter` & `\makeatother`], page 127. Ceci permet de créer des commandes que les utilisateurs ne risquent pas de redéfinir accidentellement. Une autre technique est de préfixer les commandes spécifiques à une classe ou paquetage avec une chaîne particulière, de sorte à empêcher votre classe ou paquetage d'interférer avec d'autres. Par exemple, la classe `notecms` pourrait avoir des commandes `\cms@tolist`, `\cms@fromlist`, etc.

3.3.1 Structure d'une extension (classe ou paquetage)

Un fichier de classe pour paquetage comprend typiquement quatre parties.

1. Dans la *partie d'identification* le fichier dit s'il s'agit d'un paquetage ou d'une classe L^AT_EX et s'auto-décrit, en utilisant les commandes `\NeedsTeXFormat` et `\ProvidesClass` ou `\ProvidesPackage`.
2. La partie des *déclarations préliminaires* déclare des commandes et peut aussi charger d'autres fichiers. D'ordinaire ces commandes sont celles nécessaires au code utilisé dans la partie suivante. Par exemple, une classe `notecms` pourrait être appelée avec une option pour lire un fichier où est défini une liste de personnes désignées comme destinataires de la note, comme `\documentclass[destinataires-math]{notecms}`, et donc on a besoin de définir une commande `\newcommand{\defdestinataires}[1]{\def\@liste@destinataires{#1}}` à utiliser dans ce fichier.
3. Dans la partie de *gestion des options* la classes ou le paquetage déclare et traite ses options. Les options de classes permette à l'utilisateur de commencer leur document comme dans `\documentclass[liste d'options]{nom de la classe}`, pour modifier le comportement de la classe. Un exemple est lorsque on déclare

`\documentclass[11pt]{article}` pour régler la taille par défaut de la police du document.

4. Finalement, dans la partie des *déclarations supplémentaires* la classe ou le paquetage effectue la plus grosse partie de son travail : déclarant de nouvelles variables, commandes ou polices, et chargeant d'autres fichiers.

Voici le commencement d'un fichier de classe, ce qui doit être sauvegardé comme `souche.cls` à un emplacement où L^AT_EX peut le trouver, par exemple dans le même répertoire que le fichier `.tex`.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{souche}[2017/07/06 souche à partir de laquelle contruire des classes]
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
\ProcessOptions\relax
\LoadClass{article}
```

Elle s'auto-identifie, traite les options de classe par défaut en les passant toutes à la classe `article`, et puis charge la classe `article` de sorte à fournir la base du code de cette classe.

Pour plus d'information, voir le guide officiel pour les auteurs de classes et de paquetage, le « Class Guide », <http://www.latex-project.org/help/documentation/clsguide.pdf> (la plupart des descriptions faites ici s'inspirent de ce document), ou l'article suivant <https://www.tug.org/TUGboat/tb26-3/tb84heff.pdf> illustrant la construction d'une nouvelle classe.

3.3.2 Commande pour extension (classe ou paquetage)

Voici les commandes conçues pour aider les auteurs d'extension (classes ou paquetages).

```
\AtBeginDvi{specials}
```

Sauvegarde dans une registre de boîte des choses qui sont à écrire dans le fichier `.dvi` au début de l'achèvement de la première page du document.

```
\AtEndOfClass{code}
```

```
\AtEndOfPackage{code}
```

Crochet pour insérer le `code` à exécuter lorsque L^AT_EX termine le traitement de la classe ou du paquetage courants. On peut utiliser ces crochet plusieurs fois ; le `code` sera exécuté dans l'ordre d'appel. Voir aussi `\AtBeginDocument`, page `\AtBeginDocument`, page `\AtBeginDocument`.

```
\CheckCommand{cmd}[num][défaut]{définition}
```

```
\CheckCommand*{cmd}[num][défaut]{définition}
```

Similaire à `\newcommand` (voir Section 12.1 [`\newcommand` & `\renewcommand`], page 124) mais ne définit pas `cmd` ; à la place vérifie que la définition actuelle de `cmd` est exactement celle donnée par `définition` et est ou n'est pas *longue* selon ce que l'on attend. Une commande est dite longue lorsque elle accepte `\par` au sein d'un argument. On attend que la commande `cmd` soit longue avec la version non-étoilée de `\CheckCommand`. Lève une erreur en cas d'échec de la vérification. Ceci vous permet de vérifier avant de redéfinir vous-même `cmd` qu'aucun paquetage ne l'a pas déjà fait.

```

\ClassError{nom de la classe}{texte de l'erreur}{texte d'aide}
\PackageError{nom du paquetage}{texte de l'erreur}{texte d'aide}
\ClassWarning{nom de la classe}{texte avertissement}
\PackageWarning{nom du paquetage}{texte avertissement}
\ClassWarningNoLine{nom de la classe}{texte avertissement}
\PackageWarningNoLine{nom du paquetage}{texte avertissement}
\ClassInfo{nom de la classe}{texte info}
\PackageInfo{nom du paquetage}{texte info}
\ClassInfoNoLine{nom de la classe}{texte info}
\PackageInfoNoLine{nom du paquetage}{texte info}

```

Produit un message d'erreur, ou des messages d'avertissement ou d'information.

Pour `\ClassError` et `\PackageError` le message est *texte de l'erreur*, suivi de l'invite d'erreur ? de \TeX . Si l'utilisateur demande de l'aide en tapant `h`, il voit le *texte d'aide*.

Les quatre commandes d'avertissement (warning) sont similaires à ceci près qu'elles écrivent *texte avertissement* à l'écran sans invite d'erreur. Les quatre commandes d'information écrivent *texte info* seulement dans le fichier journal. Les versions en `NoLine` n'affichent pas le numéro de la ligne générant le message, alors que les autres versions le font.

Pour formater les messages, y compris le *texte d'aide* : utilisez `\protect` pour empêcher une commande de se développer, obtenez un saut de ligne avec `\MessageBreak`, et obtenez une espace avec `\space` lorsque l'utilisation d'un caractère espace ne le permet pas, comme après une commande. Notez que \LaTeX ajoute un point final à chaque message.

`\CurrentOption`

Se développe au contenu de l'option en cours de traitement. Peut uniquement être utilisé au sein de l'argument *code* soit de `\DeclareOption`, soit de `\DeclareOption*`.

```

\DeclareOption{option}{code}
\DeclareOption*{option}{code}

```

Rend un option *option* disponible pour l'utilisateur, de sorte à ce qu'il puisse la passer à leur commande `\documentclass`. Par exemple, la classe `notecms` pourrait avoir une option `logo` pour mettre le logo de leur organisation sur la première page avec `\documentclass[logo]{notcms}`. Le fichier de classe doit contenir `\DeclareOption{logo}{code}` (et plus loin, `\ProcessOptions`).

Si vous invoquez une option qui n'a pas été déclarée, par défaut cela produit un avertissement semblable à `Unused global option(s): [badoption]`. Vous pouvez changer ce comportement avec la version étoilée `\DeclareOption*{code}`. Par exemple, beaucoup de classes étendent une classe existante en utilisant une commande du genre `\LoadClass{article}`, et pour passer les options supplémentaires à la classe sous-jacente utilisent un code tel que celui-ci :

```

\DeclareOption*{%
  \PassOptionsToClass{\CurrentOption}{article}%
}

```

Un autre exemple est que la classes `notecms` permette aux utilisateur de tenir des listes de destinataire des notes dans des fichiers externes. Par exemple l'utilisateur invoque `\documentclass[math]{notecms}` et la classe lit le fichier `math.memo`. Ce code gère le fichier s'il existe et sinon passe l'option à la classe `article`.

```
\DeclareOption*{\InputIfFileExists{\CurrentOption.memo}{}{%
\PassOptionsToClass{\CurrentOption}{article}}}
```

```
\DeclareRobustCommand{cmd}[num][défaut]{définition}
```

```
\DeclareRobustCommand*{cmd}[num][défaut]{définition}
```

Similaire à `\newcommand` et `\newcommand*` (voir Section 12.1 [`\newcommand` & `\renewcommand`], page 124) mais déclare une commande robuste, même si *définition* comprend du code fragile. (Pour une discussion sur les commandes robustes et fragiles voir Section 12.11 [`\protect`], page 134). Utilisez ces commande pour définir de nouvelles commandes robustes, ou redéfinir des commandes existantes en les rendant robustes. Contrairement à `\newcommand` elle ne produit pas d'erreur si la macro *cmd* existe déjà ; au lieu de cela un message est écrit dans le fichier journal pour indiquer la redéfinition de la commande.

Les commandes définies de cette manière sont légèrement moins efficace que celle définies avec `\newcommand`, c'est pourquoi, à moins que le contenu de la commande soit fragile et que la commande soit utilisée au sein d'un argument mouvant, utilisez `\newcommand`.

Le paquetage `etoolbox` offre les commandes `\newrobustcmd`, `\newrobustcmd*`, `\renewrobustcmd`, `\renewrobustcmd*`, `\providrobustcmd`, et `\providrobustcmd*` qui sont similaire aux commandes standards `\newcommand`, `\newcommand*`, `\renewcommand`, `\renewcommand*`, `\providecommand`, et `\providecommand*`, mais définissent une commande *cmd* robuste avec deux avantages par rapport à `\DeclareRobustCommand` :

1. Elle utilisent un mécanisme de protection de bas niveau d'e-TeX au lieu du mécanisme de plus au niveau de L^AT_EX utilisant `\protect`, ainsi elles ne subissent pas la légère perte de performance mentionnée plus haut, et
2. Elles font la même distinction entre `\new...`, `\renew...`, et `\provide...` que les commandes standards, ainsi elle ne font pas qu'envoyer un message dans le journal lorsque vous redéfinissez *cmd* déjà existantes, dans ce cas vous devez utiliser soit `\renew...` soit `\provide...` ou il y a une erreur.

```
\IfFileExists{nom fichier}{si vrai}{si faux}
```

```
\InputIfFileExists{nom fichier}{si vrai}{si faux}
```

Exécute *si vrai* si L^AT_EX peut trouver le fichier *nom fichier* et *si faux* sinon. Dans le second cas, le fichier est lu immédiatement après exécuter *si vrai*. Ainsi `\IfFileExists{img.pdf}{\includegraphics{img.pdf}}{\typeout{AVERTISSEMENT : img.pdf introuvable}}` n'inclut le graphique `img.pdf` que s'il est trouvé, mais autrement produit seulement un avertissement.

Cette commande cherche le fichier dans tous les chemins de recherche que that L^AT_EX utilise, et non pas seulement dans le répertoire courant. Pour chercher uniquement dans le répertoire courant faites quelque chose du genre de

`\IfFileExists{./nom fichier}{si vrai}{si faux}`. Si vous demandez un fichier dont le nom n'a pas d'extension `.tex` alors L^AT_EX commencera par chercher le fichier en apposant `.tex` à son nom ; pour plus ample information sur la façon dont L^AT_EX gère les extensions de nom de fichier voir Section 24.3 [`\input`], page 241.

`\LoadClass[liste d'options]{nom de la classe}[date de parution]`

`\LoadClassWithOptions{nom de la classe}[date de parution]`

Charge une classe, comme avec `\documentclass[options list]{nom de la classe}[release info]`. Voici un exemple : `\LoadClass[twoside]{article}`.

La *liste d'options*, si présente, est une liste ponctuée par des virgules. La *date de parution* est optionnelle. Si elle est présente, elle doit avoir le format `AAAA/MM/JJ`. Si vous demandez une *date de parution* et que la date du paquetage installé sur votre système est antérieure, alors vous obtiendrez un avertissement à l'écran et dans le journal de compilation du genre de `You have requested, on input line 4, version '2038/01/19' of document class article, but only version '2014/09/29 v1.4h Standard LaTeX document class' is available.`

La variante de la commande `\LoadClassWithOptions` utilise la liste des options de la classe courante. Cela veut dire qu'elle ignore toute options passée via `\PassOptionsToClass`. Ceci est une commande de commodité qui vous permet de construire une nouvelle classe en l'héritant d'une classe existante, telle que la classe standard `article`, sans avoir à gérer les options qui furent passée.

`\ExecuteOptions{liste d'options}`

Pour chaque option *option* de la *liste d'options*, dans l'ordre d'apparition, cette commande exécute la commande `\ds@option`. Si cette commande n'est pas définie, alors l'option *option* est ignorée.

Ceci peut être utilisé pour fournir d'un liste d'option par défaut avant le `\ProcessOptions`. Par exemple, si dans un fichier de classe vous désirez utiliser par défaut la taille de police 11pt alors vous devriez spécifier `\ExecuteOptions{11pt}\ProcessOptions\relax`.

`\NeedsTeXFormat{format}[date du format]`

Spécifie le format sous lequel cette classe doit être utilisée. Cette directive est souvent donnée à la première ligne du fichier de classe, et le plus souvent elle est utilisée de cette façon : `\NeedsTeXFormat{LaTeX2e}`. Lorsque un document utilisant cette classe est traité, le nom du format donné ici doit s'accorder avec le format qui est en cours d'exécution (y compris le fait que la chaîne *format* est sensible à la casse). Si il ne s'y accorde pas alors l'exécution est interrompue par une erreur du genre de `'This file needs format 'LaTeX2e' but this is 'xxx'.'`

Pour spécifier une version du format dont vous savez qu'elle prend en charge certaines fonctions, incluez l'argument optionnel *date du format* correspondant au format où ces fonction furent implémentés. Si cette argument est présent il doit être de la forme `AAAA/MM/JJ`. Si la version de format installée sur votre système est antérieure à la *date du format* alors vous obtiendrez un avertisse-

ment du genre de ‘You have requested release ‘2038/01/20’ of LaTeX, but only release ‘2016/02/01’ is available.’

`\OptionNotUsed`

Ajoute l’option courante à la liste des options non utilisées. Ne peut être utilisé qu’au sein de l’argument *code* de `\DeclareOption` ou `\DeclareOption*`.

`\PassOptionsToClass{liste d’options}{nom de la classe}`

`\PassOptionsToPackage{liste d’options}{nom du paquetage}`

Ajoute les options de la liste ponctuée par des virgules *option list* aux options utilisée par toute commande ultérieure `\RequirePackage` ou `\usepackage` pour le paquetage *nom du paquetage* ou la classe *nom de la classe*.

La raison d’être de ces commande est que vous pouvez charger un paquetage autant de fois que vous le voulez sans options, mais que si voulez passer des options alors vous ne pouvez les fournir qu’au premier chargement. Charger un paquetage avec des options plus d’une fois produit une erreur du genre de `Option clash for package toto`. (L^AT_EX lance l’erreur même s’il n’y a pas de conflit entre les options.)

Si votre propre code introduit un paquetage deux fois alors vous pouvez réduire cela en une fois, par exemple en remplaçant les deux `\RequirePackage[landscape]{geometry}\RequirePackage[margins=1in]{geometry}` par un seul `\RequirePackage[landscape,margins=1in]{geometry}`. Mais si vous chargez un paquetage qui à son tour en charge un autre alors vous devez mettre en queue les options que vous désirez pour cet autre paquetage. Par exemple, supposons que le paquetage `toto` charge le paquetage `geometry`. Au lieu de `\RequirePackage{toto}\RequirePackage[draft]{graphics}` vous devez écrire `\PassOptionsToPackage{draft}{graphics}\RequirePackage{toto}`. (Si `toto.sty` charge une option en conflit avec ce que vous désirez alors vous devrez considérer une modification de son code source.)

Ces commandes sont également utiles aux utilisateurs de base et pas seulement aux auteurs de classes et paquetage. Par exemple, supposons qu’un utilisateur veuille charger le paquetage `graphicx` avec l’option `draft` et veuille également utiliser une classe `toto` qui charge le paquetage `graphicx`, mais sans cette option. L’utilisateur peut commencer son fichier L^AT_EX avec `\PassOptionsToPackage{draft}{graphicx}\documentclass{toto}`.

`\ProcessOptions`

`\ProcessOptions* \@options`

Exécute le code pour chaque option que l’utilisateur a invoquée. À inclure dans le fichier classe sous la forme `\ProcessOptions\relax` (à cause de l’existence de la variante étoilée de la commande).

Les options tombent dans deux catégories. Les *options locales* sont spécifiées pour un paquetage particulier au sein de l’argument *options* dans `\PassOptionsToPackage{options}`, `\usepackage[options]`, ou `\RequirePackage[options]`. Les *options globales* sont celles données par l’utilisateur de la classe dans `\documentclass[options]`. (Si une option est spécifiée à la fois localement et globalement, alors elle est locale).

Lorsque `\ProcessOptions` est appelé pour un paquetage `pkg.sty`, il se produit ce qui suit :

1. Pour chaque option *option* déclarée jusqu'à ce point avec `\DeclareOption`, \LaTeX examine si cette option est soit globale soit locale pour `pkg`. Si c'est le cas, il exécute le code déclaré. Ceci est fait dans l'ordre de passage de ces options à `pkg.sty`.
2. Pour chaque option locale restante, il exécute la commande `\ds@option` si elle a été définie quelque part (autrement que par un `\DeclareOption`) ; sinon, il exécute le code de traitement par défaut des options donné dans `\DeclareOption*`. Si aucun code de traitement par défaut n'a été déclaré, il produit un message d'erreur. Ceci est fait dans l'ordre dans lequel ces options ont été spécifiées.

Lorsque `\ProcessOptions` est appelé pour une classe il fonctionne de la même manière à ceci près que toutes les options sont locales, et que le code par défaut pour `\DeclareOption*` et `\OptionNotUsed` plutôt qu'une erreur.

La version étoilée `\ProcessOptions*` exécute le traitement des options dans l'ordre spécifié par les commandes appelante, plutôt que dans l'ordre de déclaration de la classe ou du paquetage. Pour un paquetage, ceci signifie que les options globales sont traitées en premier.

```
\ProvidesClass{nom de la classe}[date de parution brève information
supplémentaire]
\ProvidesClass{nom de la classe}[date de parution]
\ProvidesPackage{nom du paquetage}[date de parution brève information
supplémentaire]
\ProvidesPackage{nom du paquetage}[date de parution]
```

Identifie la classe ou le paquetage, en tapant un message sur la console et dans le fichier journal.

Lorsqu'un utilisateur écrit `\documentclass{notecms}` alors \LaTeX charge le fichier `notecms.cls`. De même, un utilisateur écrivant `\usepackage{essai}` invite \LaTeX à charger le fichier `essai.sty`. Si le nom du fichier ne s'accorde pas à l'argument *nom de la classe* ou *nom du paquetage* alors un avertissement est produit. Ainsi, si vous invoquez `\documentclass{notecms}`, et que le fichier the file `notecms.cls` comprend la déclaration statement `\ProvidesClass{xxx}` alors vous obtiendrez un avertissement du genre de like `You have requested document class 'notecms', but the document class provides 'xxx'`. Cet avertissement n'empêche pas \LaTeX de traiter le reste du fichier de la classe normalement.

Si vous incluez l'argument optionnel, alors vous devez inclure la date, avant le premier espace s'il y en a, et elle doit avoir le format `AAAA/MM/JJ`. Le reste de l'argument est en format libre, toutefois il identifie traditionnellement la classe, et est écrit pendant la compilation à l'écran et dans le journal. Ainsi, si votre fichier `notecms.cls` contient la ligne `\ProvidesClass{smcmem}[2008/06/01 v1.0 Classe note CMS]` la première ligne de votre document est `\documentclass{notecms}` alors vous pourrez voir `Document Class: notecms 2008/06/01 v1.0 Classe note CMS`.

La date dans l'argument optionnel permet aux utilisateurs de classe et de paquetage de demander à être avertis si la version de la classe ou du paquetage installé sur leur système est antérieure à *date de parution*, en utilisant les arguments optionnels comme dans `\documentclass{smcmem}[2018/10/12]` ou `\usepackage{toto}[[2017/07/07]]`. (Notez que les utilisateurs de paquetages incluent seulement rarement une date, et les utilisateurs de classe presque jamais).

`\ProvidesFile{nom fichier}[information supplémentaire]`

Déclare un fichier autre que les fichiers principaux de classe ou de paquetage, tel qu'un fichier de configuration ou un fichier de définition de police. Mettez la commande dans ce fichier et vous obtiendrez dans le journal une information du genre de `File: essai.config 2017/10/12 fichier de configuration pour essai.cls` lorsque *nom fichier* vaut 'essai.config' et que *information supplémentaire* vaut '2017/10/12 fichier de configuration pour essai.cls'.

`\RequirePackage[liste d'options]{nom du paquetage}[date de parution]`

`\RequirePackageWithOptions{nom du paquetage}[date de parution]`

Charge un paquetage, comme la commande `\usepackage` pour les auteurs de documents. Voir Section 3.2 [Additional packages], page 9. Voici un exemple : `\RequirePackage[landscape,margin=1in]{geometry}`. Notez que l'équipe de développement de L^AT_EX recommande fortement l'utilisation de ces commandes de préférence à l'`\input` de T_EX de base ; voir le « Class Guide ».

La *liste d'options*, si présente, est une liste ponctuée de virgules. La *date de parution*, si présente, doit avoir le format *AAAA/MM/JJ*. Si la date de parution du paquetage tel qu'il est installé sur votre système est antérieure à *date de parution* alors vous obtiendrez un avertissement du genre de `You have requested, on input line 9, version '2017/07/03' of package jhtest, but only version '2000/01/01' is available.`

La variante `\RequirePackageWithOptions` utilise la liste d'options de la classe courante. Ceci implique qu'elle ignore toute option passée à la classe via `\PassOptionsToClass`. C'est une commande de commodité pour permettre facilement de construire des classes sur des classes existantes sans avoir à gérer les options qui sont passées.

La différence entre `\usepackage` et `\RequirePackage` est mince. La commande `\usepackage` est à l'intention du fichier document alors que `\RequirePackage` l'est à celle des fichiers paquetage ou classe. Ainsi, utiliser `\usepackage` avant la commande `\documentclass` amène L^AT_EX à produire une erreur du genre de `\usepackage before \documentclass`, là où vous pouvez utiliser `\RequirePackage`.

4 Polices de caractères

\LaTeX gère les polices de caractères en vous offrant de puissantes capacités. Notamment, son schéma *NFSS* vous permet de travailler aisément avec les polices, ou *familles de fontes* dans votre document (par exemple, voir Section 4.2 [Font styles], page 23). De plus, les documents \LaTeX peuvent utiliser la plupart des polices disponibles de nos jours, y compris des versions de Times Roman, Helvetica, Courier, etc. (Notez cependant que beaucoup de polices ne prennent pas en charge les mathématiques).

La première police dans le monde \TeX était Computer Modern, développée par Donald Knuth. Elle est la police par défaut pour les documents \LaTeX et est encore la plus largement utilisée. Mais passer à une autre police n'implique la plupart du temps que peu de commandes. Par exemple, mettre ce qui suit dans votre préambule vous donne une police simili-Palatino, ce qui est élégant et plus lisible sur écran que bien d'autres polices, tout en continuant à vous permettre de composer des mathématiques. (Exemple fourni par Michael Sharpe, <https://math.ucsd.edu/~msharp/RcntFnts.pdf>).

```
\usepackage[osf]{newpctext} % osf pour le texte, non les math
\usepackage{cabin} % sans serif
\usepackage[varqu,varl]{inconsolata} % sans serif tapuscrit
\usepackage[bigdelims,vvarbb]{newpymath} % bb de STIX
\usepackage[cal=boondoxo]{mathalfa} % mathcal
```

De plus, les moteurs `xelatex` ou `lualatex` vous permettent d'utiliser toute police sur votre système en format OpenType ou TrueType (voir Section 2.3 [\TeX engines], page 3).

Le catalogue de polices de \LaTeX (<https://tug.org/FontCatalogue>) donne des échantillons graphiques de polices et du source à copier/coller pour utiliser un grand nombre de polices, dont beaucoup prennent en charges les mathématiques. Son but est de couvrir toutes les polices libres pour l'alphabet latin utilisables facilement avec \LaTeX .

Plus ample information peut également être trouvée sur le \TeX Users Group, à <https://www.tug.org/fonts/>.

4.1 Le paquetage fontenc

Synopsis :

```
\usepackage[codage_fonte]{fontenc}
```

ou

```
\usepackage[codage_fonte1, codage_fonte2, ...]{fontenc}
```

Spécifie les codages de fonte. Un codage de fonte est une application des codes de caractères vers les glyphes de fonte utilisés pour composer votre texte en sortie.

Ce paquetage ne fonctionne que si vous utilisez le moteur `pdflatex` (voir Section 2.3 [\TeX engines], page 3). Si vous utilisez les moteurs `xelatex` ou `lualatex` alors utilisez à la place le paquetage `fontspec`.

La police originale de \TeX , Computer Modern, comprend un nombre limité de caractères. Par exemple, pour faire des caractères accentués ordinaires vous devez utiliser `\accent` (voir Section 16.4.1 [`\accent`], page 176) mais cela désactive la césure. Les utilisateurs de \TeX ont

spécifié plusieurs normes concernant l'accès aux ensembles élargis de caractères fournis par les fontes modernes. Si vous utilisez `pdflatex` alors placez ceci dans le préambule :

```
\usepackage[T1]{fontenc}
```

cela active la prise en charge pour les langages européens les plus répandus, notamment le français, l'italien, le polonais. En particulier, si vous avez des mots avec des lettres accentuées alors \LaTeX gèrera leur césure et le document en sortie peut faire l'objet de copier/coller. (La seconde ligne optionnelle vous permet de taper directement les caractères accentués dans votre fichier source).

Si vous utilisez un codage tel que `T1` et que les caractères ne sont pas nets ou passent mal au zoom alors vos fontes sont probablement définies par des cartes binaires (bitmap), on parle de fontes matricielles ou Type 3. Si vous désirez des fontes vectorielles, utilisez un paquetage tel que `lmodern` ou `cm-super` pour obtenir des fontes vectorielles qui étendent celles utilisées par défaut par \LaTeX .

Pour chaque *codage_fonte* passé en option mais encore non déclaré, ce paquetage charge les fichiers de définitions du codage, nommé *codage_fonteenc.def*. Il règle aussi `\encodingdefault` comme le dernier codage dans la liste d'option.

Voici les valeurs usuelles pour *codage_fonte*.

- `OT1` Le codage original pour \TeX . Limité quasiment aux lettres latines non accentués.
- `OMS, OML` Codage des symboles et des lettres mathématiques.
- `T1` Texte étendu de \TeX . Appelé parfois le codage Cork du nom de la conférence du groupe d'utilisateurs \TeX durant laquelle il fut élaboré. Donne accès à la plupart des caractères accentués européens. C'est l'option la plus fréquente pour ce paquetage.
- `TS1` Codage Texte Companion.

Par défaut \LaTeX charge `OML`, `T1`, `OT1`, et ensuite `OMS`, et règle le codage par défaut à `OT1`.

Même si vous n'utilisez pas de lettres accentuées, vous pourriez avoir besoin d'un codage de font si la fonte que vous utilisez l'exige.

Si vous utilisez des fontes codée en `T1` autre que celles de la police par défaut Computer Modern, alors vous pourriez avoir besoin de charger le paquetage qui sélectionne vos fontes avant de charger `fontenc`, de sorte à empêcher le système de charger l'une des fontes codées en `T1` de la police par défaut.

L'équipe du projet \LaTeX a réservé les noms de codage commençant par : 'T' pour les codages standards de texte à 256 caractères, 'TS' pour les symboles qui étendent les codages T correspondants, 'X' pour les codages de test, 'M' pour les codages standards de math à 256 caractères, 'A' pour les applications spéciales, 'OT' pour les codages standards de texte à 128 caractères, et 'OM' pour les codages standards de math à 128 caractères ('O' est l'initiale d'obsolete').

Ce paquetage fournit un certain nombre de commandes, détaillées ci-dessous. Beaucoup d'entre elles sont spécifiques à un codage, aussi si vous avez défini une commande qui fonctionne pour un codage mais que le codage courant en est différent alors la commande n'est pas active.

4.1.1 `\DeclareFontEncoding`

Synopsis :

```
\DeclareFontEncoding{codage}{réglages-texte}{réglages-math}
```

Déclare le codage de fonte *codage*. Sauvegarde également la valeur de *codage* dans `\LastDeclaredEncoding` (voir Section 4.1.10 [`\LastDeclaredEncoding`], page 23).

Le fichier `t1enc.def` contient cette ligne (suivie de bien d'autres) :

```
\DeclareFontEncoding{T1}{}{}
```

Les *réglages-texte* sont les commandes que \LaTeX exécutera chaque fois qu'il commute d'un codage à une autre avec les commandes `\selectfont` et `\fontencoding`. Les *réglages-math* sont les commandes que \LaTeX utilise lorsque la fonte est accédée en tant qu'alphabet mathématique.

\LaTeX ignore tout espace au sein de *réglages-texte* et *réglages-math*, de sorte à prévenir toute insertion non voulue d'espace en sortie.

Si vous inventez un codage vous devriez choisir un nom de deux ou trois lettres commençant par 'L' pour 'local', ou 'E' pour 'experimental'.

Notez que les fichiers de définition de codage de sortie sont potentiellement lus plusieurs fois par \LaTeX , ainsi utiliser, par ex. `\newcommand` peut causer une erreur. De plus, il est préférable que de tels fichiers contiennent une ligne `\ProvidesFile` (voir Section 3.3.2 [Class and package commands], page 10).

Notez aussi qu'il est préférable de n'utiliser les commandes en `\...Default` que dans un paquetage, et non dans les fichiers de définition de codage, puisque ces fichiers ne devraient contenir que des déclarations spécifiques à ce codage-là.

4.1.2 `\DeclareTextAccent`

Synopsis :

```
\DeclareTextAccent{cmd}{codage}{cassetin}
```

Définit un accent, à mettre au dessus d'autres glyphe, au sein du codage *codage* à la position *cassetin*.

Un *cassetin* est le numéro identifiant un glyphe au sein d'une fonte.

La ligne suivante de `t1enc.def` déclare que pour fabriquer un accent circonflexe comme dans `\^A`, le système va mettre l'accent dans du cassetin 2 sur le caractère 'A', qui est représenté en ASCII comme 65. (Ceci vaut à moins qu'il y ait une déclaration `DeclareTextComposite` ou `\DeclareTextCompositeCommand` pertinente ; voir Section 4.1.6 [`\DeclareTextComposite`], page 21) :

```
\DeclareTextAccent{\^}{T1}{2}
```

Si *cmd* a déjà été défini, alors `\DeclareTextAccent` ne produit pas d'erreur mais il enregistre la redéfinition dans le fichier de transcription.

4.1.3 `\DeclareTextAccentDefault`

Synopsis :

```
\DeclareTextAccentDefault{\cmd}{codage}
```

S'il y a une commande d'accent `\cmd` spécifique à un codage mais qu'il n'y a aucun `\DeclareTextAccent` associé pour ce codage, alors cette commande va prendre le relais en disant de l'utiliser comme décrit pour *codage*.

Par exemple, pour faire que le codage OT1 soit le codage par défaut pour l'accent `\`, déclarez ceci :

```
\DeclareTextAccentDefault{\}{OT1}
```

Si vous tapez un `\` alors que le codage courant n'a pas de définition pour cet accent, alors \LaTeX utilisera la définition issue de OT1.

c.-à-d. que cette commande est équivalente l'appel suivant (voir Section 4.1.11 [`\UseTextSymbol & \UseTextAccent`], page 23) :

```
\DeclareTextCommandDefault[1]{\cmd}
{\UseTextAccent{codage}{\cmd}{#1}}
```

Notez que `\DeclareTextAccentDefault` fonctionne pour l'une quelconque des commandes `fontenc` à un argument, et pas seulement les commandes d'accent.

4.1.4 `\DeclareTextCommand` & `\ProvideTextCommand`

Synopsis, l'un parmi :

```
\DeclareTextCommand{\cmd}{codage}{défn}
\DeclareTextCommand{\cmd}{codage}[nargs]{defn}
\DeclareTextCommand{\cmd}{codage}[nargs][défautargopt]{défn}
```

ou l'un parmi :

```
\ProvideTextCommand{\cmd}{codage}{défn}
\ProvideTextCommand{\cmd}{codage}[nargs]{défn}
\ProvideTextCommand{\cmd}{codage}[nargs][défautargopt]{défn}
```

Définit la commande `\cmd`, laquelle sera spécifique à un codage. Le nom *cmd* de la commande doit être précédé d'une controblique, `\`. Ces commandes ne peuvent apparaître que dans le préambule. Redéfinir `\cmd` ne cause pas d'erreur. La commande définie sera robuste même si le code dans *defn* est fragile (voir Section 12.11 [`\protect`], page 134).

Par exemple, le fichier `tienc.def` contient la ligne suivante :

```
\DeclareTextCommand{\textperthousand}{T1}{\% \char 24 }
```

Avec cela, vous pouvez exprimer des pour-milles comme ceci :

```
\usepackage[T1]{fontenc} % dans le préambule
```

...

La limite légale est de `\(0,8 \)\textperthousand`.

Si vous changez le codage de fonte en OT1 alors vous obtenez une erreur du genre de '`LaTeX Error: Command \textperthousand unavailable in encoding OT1`'.

La variante `\ProvideTextCommand` agit de même, à ceci près qu'elle ne fait rien si `\cmd` est déjà définie. La commande `\DeclareTextSymbol` est plus rapide pour une simple association d'un cassetin à un glyphe (voir Section 4.1.8 [`\DeclareTextSymbol`], page 22)

Les arguments optionnels *nargs* et *défautargopt* jouent le même rôle que dans `\newcommand` (voir Section 12.1 [`\newcommand & \renewcommand`], page 124). En un mot, *nargs* est un entier de 0 à 9 spécifiant le nombre d'arguments que la commande `\cmd` qu'on définit prend. Ce nombre inclut tout argument optionnel. Omettre cet

argument revient au même que spécifier 0 pour dire que `\cmd` n'a aucun argument. Et, si `défaulargopt` est présent, alors le premier argument de `\cmd` est optionnel, et sa valeur par défaut est `défaulargopt` (ce qui peut être la chaîne vide). Si `défaulargopt` n'est pas présent alors `\cmd` ne prend pas d'argument optionnel.

4.1.5 `\DeclareTextCommandDefault` & `\ProvideTextCommandDefault`

Synopsis :

```
\DeclareTextCommandDefault{\cmd}{défn}
```

ou :

```
\ProvideTextCommandDefault{\cmd}{défn}
```

Donne une définition par défaut à `\cmd`, pour la situation où cette commande n'est pas définie dans le codage actuellement en vigueur. La valeur par défaut doit utiliser uniquement les codages connus comme disponibles.

La déclaration ci-dessous rend `\copyright` disponible.

```
\DeclareTextCommandDefault{\copyright}{\textcircled{c}}
```

Elle utilise uniquement un codage (OMS) qui est toujours disponible.

La déclaration `\DeclareTextCommandDefault` n'est pas destinée à survenir dans un fichier de définition de codage puisque ces fichiers ne doivent déclarer que des commandes à utiliser quand vous sélectionnez ce codage. Au lieu de cela, elle devrait être dans un paquetage.

Comme pour les commandes correspondantes qui ne sont pas en `...Default`, la déclaration `\ProvideTextCommandDefault` a exactement le même comportement que `\DeclareTextCommandDefault` à ceci près qu'elle ne fait rien si `\cmd` est déjà définie (voir Section 4.1.4 [`\DeclareTextCommand` & `\ProvideTextCommand`], page 20). De sorte qu'un paquetage peut l'utiliser pour fournir une définition de repli que d'autres paquetages peuvent améliorer.

4.1.6 `\DeclareTextComposite`

Synopsis :

```
\DeclareTextComposite{\cmd}{codage}{objet_simple}{cassetin}
```

Accède à un glyphe accentué directement, c.-à-d. sans avoir à placer une accent au dessus d'un caractère séparé.

La ligne suivante extraite de `t1enc.def` signifie que `\^o` aura pour effet que \LaTeX compose le 'o' bas-de-casse en prenant le caractère directement du cassetin 224 de la fonte.

```
\DeclareTextComposite{\^}{T1}{o}{244}
```

Voir Section 4.1 [fontenc package], page 17, pour une liste des codages usuels. L'`objet_simple` devrait être un seul caractère ou une seule commande. L'argument `cassetin` est d'ordinaire un entier positif représenté en décimal (bien que les notations octale ou hexadécimale soient possibles). Normalement `\cmd` a déjà été déclarée pour ce codage, soit avec `\DeclareTextAccent` soit avec un `\DeclareTextCommand` à un argument. Dans `t1enc.def`, la ligne ci-dessus suit la commande `\DeclareTextAccent{\^}{T1}{2}`.

4.1.7 `\DeclareTextCompositeCommand`

Synopsis :

```
\DeclareTextCompositeCommand{\cmd}{codage}{arg}{code}
```

Une version plus générale de `\DeclareTextComposite` qui exécute un code arbitraire avec `\cmd`.

Ceci permet que des accents sur ‘i’ se comportent de la même façon que des accents sur le i-sans-point, `\i`.

```
\DeclareTextCompositeCommand{\'}{OT1}{i}{\'\i}
```

Voir Section 4.1 [fontenc package], page 17, pour une liste des codages usuels. Normalement `\cmd` doit avoir été déjà déclarée avec `\DeclareTextAccent` ou avec un `\DeclareTextCommand` à un argument.

4.1.8 `\DeclareTextSymbol`

Synopsis :

```
\DeclareTextSymbol{\cmd}{codage}{cassetin}
```

Définit un symbole dans le codage *codage* à la position *cassetin*. Les symboles définis de cette manière sont destinés à être utilisés dans du texte, et non dans des mathématiques.

Par exemple, cette ligne extraite de `t1enc.def` déclare le numéro du glyphe à utiliser pour «, le guillemet gauche :

```
\DeclareTextSymbol{\guillemotleft}{T1}{19}
```

La commande `\DeclareTextCommand{\guillemotleft}{T1}{\char 19}` a le même effet mais est plus lente (voir Section 4.1.4 [`\DeclareTextCommand` & `\ProvideTextCommand`], page 20).

Voir Section 4.1 [fontenc package], page 17, pour une liste des codages usuels. Le *cassetin* peut être spécifié en notation décimale, ou octale (comme dans ‘023), ou hexadécimale (comme dans "13), bien que la notation décimale ait l’avantage d’être insensible à la redéfinition du guillemet anglais simple ou double par un autre paquetage.

Si `\cmd` est déjà définie, alors `\DeclareTextSymbol` ne produit pas une erreur, mais il inscrit que la redéfinition a eu lieu dans le fichier transcript.

4.1.9 `\DeclareTextSymbolDefault`

Synopsis :

```
\DeclareTextSymbolDefault{\cmd}{codage}
```

S’il y a une `\cmd` de symbole spécifique à un codage mais qu’il n’y a pas de `\DeclareTextSymbol` associé pour ce codage, alors cette commande peut prendre le relais, en disant d’obtenir le symbole selon la description faite pour *codage*.

Par exemple, pour déclarer que si le codage courant ne donne pas de signification à `\textdollar` alors on utilise celle issue d’OT1, déclarer ceci :

```
\DeclareTextSymbolDefault{\textdollar}{OT1}
```

c.-à-d. que cette commande est équivalente à cet appel (voir Section 4.1.11 [`\UseTextSymbol` & `\UseTextAccent`], page 23) :

```
\DeclareTextCommandDefault{\cmd}
```

```
{\UseTextSymbol{codage}{\cmd}}
```

Notez que `\DeclareTextSymbolDefault` peut être utilisé pour définir une signification par défaut pour toute commande à zéro argument de `fontenc`.

4.1.10 `\LastDeclaredEncoding`

Synopsis :

```
\LastDeclaredEncoding
```

Obtient le nom du dernier codage déclaré. La commande `\DeclareFontEncoding` stocke le nom de sorte qu'on puisse le retrouver avec cette commande (voir Section 4.1.1 [`\DeclareFontEncoding`], page 19).

Dans l'exemple suivant, on compte sur `\LastDeclaredEncoding` plutôt que de donner le nom du codage explicitement :

```
\DeclareFontEncoding{JH1}{}{}
\DeclareTextAccent{\'}{\LastDeclaredEncoding}{0}
```

4.1.11 `\UseTextSymbol` & `\UseTextAccent`

Synopsis :

```
\UseTextSymbol{codage}{\cmd}
```

ou :

```
\UseTextAccent{codage}{\cmd}{texte}
```

Utilise un symbole ou un accent qui n'est pas défini dans le codage courant.

En général, pour utiliser une commande `fontenc` dans un codage où elle n'est pas définie, et si la commande n'a pas d'arguments, alors vous pouvez utiliser quelque chose de ce genre :

```
\UseTextSymbol{OT1}{\ss}
```

ce qui est équivalent à ceci (notez que les accolades externes forment un groupe, de sorte que \LaTeX revient au codage précédent après le `\ss`) :

```
{\fontencoding{OT1}\selectfont\ss}
```

De même, pour utiliser une commande `fontenc` dans un codage où elle n'est pas définie, et si la commande a un argument, vous pouvez l'utiliser ainsi :

```
\UseTextAccent{OT1}{\'}{a}
```

ce qui est équivalent à cela (encore une fois, notez les accolades externes formant un groupe) :

```
{\fontencoding{OT1}\selectfont
 \'\fontencoding{cdg_en_vigueur}\selectfont a}}
```

Ci-dessus, `cdg_en_vigueur` est le codage en vigueur avant cette séquence de commandes, de sorte que 'a' est composé en utilisant le codage courant, et seulement l'accent est pris de OT1.

4.2 styles des polices

Les commandes de styles suivantes sont prises en charge par \LaTeX .

Dans le tableau ci-dessous listant des commandes, les commandes en `\text...` sont utilisées avec un argument, comme dans `\textit{texte}`. C'est la forme préférée. Mais affichée à sa suite, entre parenthèses, est la *forme déclarative* correspondante, qui est

souvent utile. Cette forme ne prend pas arguments comme dans `{\itshape texte}`. La portée la forme déclarative s'étend jusqu'à la prochaine commande de type style ou jusqu'à la fin du groupe actuel. De plus, chacune a une forme par environnement comme `\begin{itshape}...\end{itshape}`, que l'on décrit plus amplement à la fin de cette rubrique.

Ces commandes, dans n'importe laquelle des trois formes, sont cumulatives ; par exemple, vous pouvez dire aussi bien `\sffamily\bfseries` que `\bfseries\sffamily` pour obtenir du gras sans empattement.

Un avantage de ces commandes est qu'elles fournissent automatiquement une correction d'italique si nécessaire (voir Section 19.9 [V/], page 197). Plus précisément, elles insèrent la correction italique à moins que le caractère suivant dans la liste `\nocorrlist`, cette liste consistant par défaut en le point et la virgule. Pour supprimer l'insertion automatique de la correction italique, utilisez `\nocorr` au début ou à la fin de l'argument de la commande, comme dans `\textit{\nocorr text}` ou `\textsc{text \nocorr}`.

`\textrm (\rmfamily)`
Romain.

`\textit (\itshape)`
Italique.

`\textmd (\mdseries)`
Poids moyen (par défaut).

`\textbf (\bfseries)`
Gras.

`\textup (\upshape)`
Droit (par défaut). Le contraire d'incliné.

`\textsl (\slshape)`
Inclinée.

`\textsf (\sffamily)`
Sans empattement.

`\textsc (\scshape)`
Petites capitales.

`\texttt (\ttfamily)`
Tapuscrit, encore appelé « machine à écrire ».

`\textnormal (\normalfont)`
Police principale du document.

Bien qu'elle change également la fonte, la commande `\emph{texte}` est sémantique, pour que *texte* soit mis en valeur, et ne devrait pas être utilisé en remplacement de `\textit`. Par exemple, le résultat de `\emph{texte début} \emph{texte milieu} texte fin` est que *texte début* et *texte fin* sont en italique, mais *texte milieu* est en romain.

L^AT_EX fournit également les commandes suivantes, qui font passer inconditionnellement à un style donné, c.-à-d. *ne* sont *pas* cumulatives. Elles sont utilisées déclarativement : `{\cmd ... }` au lieu de `\cmd{...}`.

(Les commandes inconditionnelle ci-dessous sont une version plus ancienne de la commutation de fonte. Les commandes qui précèdent sont une amélioration dans les plupart des circonstances ; toutefois il arrive qu'une commutation inconditionnelle soit ce dont on a besoin).

<code>\bf</code>	Passage en gras .
<code>\cal</code>	Passage en lettres calligraphiques pour les mathématiques.
<code>\em</code>	Accent (italique dans romain, romain dans italiques).
<code>\il</code>	Italique.
<code>\rm</code>	Romain.
<code>\sc</code>	Les petites capitales.
<code>\sf</code>	Sans empattement.
<code>\sl</code>	Incliné (oblique).
<code>\tt</code>	Tapuscrit, encore appelé « Machine à écrire » (largeur fixe).

La commande `\em` est la version inconditionnelle de `\emph`.

Les commandes suivantes sont à utiliser en mode math. Elle ne sont pas cumulatives, ainsi `\mathbf{\mathit{symbole}}` ne crée pas un *symbole* en gras italiques ; au lieu de cela, il sera juste en italique. C'est ainsi car typiquement les symboles mathématiques ont besoin d'être toujours traités typographiquement de la même manière, indépendamment de ce qui les entoure.

<code>\mathrm</code>	Romain, pour une utilisation en mode mathématique.
<code>\mathbf</code>	Gras, pour une utilisation en mode mathématique.
<code>\mathsf</code>	Sans empattement, pour une utilisation en mode mathématique.
<code>\mathtt</code>	Tapuscrit, pour une utilisation en mode mathématique.
<code>\mathit</code> (<code>\mit</code>)	Italique, pour une utilisation en mode mathématique.
<code>\mathnormal</code>	Pour une utilisation en mode mathématique, par exemple dans un autre déclaration de type style.
<code>\mathcal</code>	Lettres calligraphiques, pour une utilisation en mode mathématique.

En outre, la commande `\mathversion{bold}` peut être utilisée pour commuter en caractères gras les lettres et les symboles dans les formules. `\mathversion{normal}` restaure la valeur par défaut.

Enfin, la commande `\oldstylenums{chiffres}` sert à composer des chiffres dits *elzévirien*, ou *non alignés*, qui ont des hauteurs et profondeurs (et parfois largeurs) distincts de l'alignement standard des chiffres dits *Didot*, ou *alignés*, qui ont tous la même hauteur que les lettres capitales. Les polices L^AT_EX par défaut prennent en charge cela, et respecteront `\textbf` (mais pas les autres styles, il n'y a pas de elzévirien italique pour les chiffres en Computer Modern). Beaucoup d'autres polices ont des chiffres elzévirien aussi ; parfois des options de paquetage sont mises à disposition pour en faire le style par défaut. Article de FAQ : <https://www.texfaq.org/FAQ-osf>.

4.3 tailles des polices

Les commandes de type de taille standards suivantes sont pris en charge par L^AT_EX. Le tableau indique le nom de la commande et taille utilisée (en points) correspondant réellement pour la police avec les options ‘10pt’, ‘11pt’, et ‘12pt’ de classe de document, respectivement (voir Section 3.1 [Document class options], page 7).

commande	10pt	11pt	12pt
<code>\tiny</code>	5	6	6
<code>\scriptsize</code>	7	8	8
<code>\footnotesize</code>	8	9	10
<code>\small</code>	9	10	10,95
<code>\normalsize</code> (par défaut)	10	10.95	12
<code>\large</code>	12	12	14,4
<code>\Large</code>	14.4	14.4	17,28
<code>\LARGE</code>	17.28	17.28	20,74
<code>\huge</code>	20,74	20,74	24,88
<code>\Huge</code>	24.88	24.88	24.88

Les commandes énumérées ici sont des formes déclaratives (non des formes par environnement), vu que c’est ainsi qu’on les utilise typiquement. Comme dans l’exemple qui suit :

```
\begin{quotation} \small
  Le Tao que l’on peut nommer n’est pas le Tao éternel.
\end{quotation}
```

Dans ce qui précède, la portée de `\small` s’étend jusqu’à la fin de l’environnement `quotation`. Elle s’arrêterait également à la prochaine commande de style ou à la fin du groupe courant, de sorte qu’on aurait pu la mettre entre accolades `{\small Ce texte est en petits caractères.}`.

Essayer d’utiliser ces commandes en math, comme dans `\small mv^2/2$`, donne l’avertissement ‘LaTeX Font Warning: Command `\small` invalid in math mode’, et la taille de fonte n’est pas changée. Pour travailler avec un formule trop grande, le plus souvent la meilleure option est d’utiliser l’environnement `displaymath` (voir Chapitre 16 [Math formulas], page 153), ou l’un des environnements fournis par le paquetage `amsmath`. Pour des mathématiques dans le texte, comme dans un tableau de formules, une alternative est quelque chose du genre de `{\small $mv^2/2$}`. (Il arrive que `\scriptsize` et `\scriptstyle` soient pris l’un pour l’autre. Les deux changent la taille mais le dernier change nombre d’autres aspects conditionnant la composition des mathématiques. Voir Section 16.7 [Math styles], page 181).

Une *forme par environnement* de chacune de ces commandes est également définie ; par exemple, `\begin{tiny}... \end{tiny}`. Toutefois, en pratique cette forme peut aisément mener à des espaces indésirables au début et/ou à la fin de d’environnement si on n’y prête pas attention, ainsi on risque en général moins de faire d’erreur si on s’en tient à la forme déclarative.

(Digression : techniquement, vu la façon dont L^AT_EX définit `\begin` et `\end`, presque toutes les commandes qui ne prennent pas d’argument on une forme par environnement. Mais dans presque tous les cas ce serait juste une source de confusion de l’utiliser. La raison

pour mentionner spécifiquement la forme par environnement des déclarations de taille de police est que cet usage particulier n'est pas rare).

4.4 commandes de fontes de bas niveau

Ces commandes sont principalement destinés aux auteurs de macros et paquetages. Les commandes indiquées ici sont seulement un sous-ensemble de celles disponibles.

`\fontencoding{codage}`

Sélectionne le codage de police, le codage de la police utilisée en sortie. Il y a un grand nombre de codages valides. Les plus usuels sont `OT1`, le codage original de D. Knuth pour la police Computer Modern (la police par défaut), et `T1`, également désigné par codage Cork, qui prend en charge les caractères accentués utilisés par les langues européennes les plus parlées (allemand, français, italien, polonais et d'autres encore), ce qui permet à \TeX de faire la césure des mots contenant des lettres accentuées. Pour plus d'information, voir <https://ctan.org/pkg/encguide>.

`\fontfamily{famille}`

Sélectionne la famille de polices. La page web <https://www.tug.dk/FontCatalogue/> donne un moyen de naviguer facilement à travers un grand nombre des polices utilisées avec \LaTeX . Voici des exemples de quelques-unes des familles les plus usuelles :

<code>pag</code>	Avant Garde
<code>fvs</code>	Bitstream Vera Sans
<code>pbk</code>	Bookman
<code>bch</code>	Charter
<code>ccr</code>	Computer Concrete
<code>cmr</code>	Computer Modern
<code>cmss</code>	Computer Modern Sans Serif
<code>cmtt</code>	Computer Modern Typewriter
<code>pcr</code>	Courier
<code>phv</code>	Helvetica
<code>fi4</code>	Inconsolata
<code>lmr</code>	Latin Modern
<code>lmss</code>	Latin Modern Sans
<code>lmtt</code>	Latin Modern Typewriter
<code>pnc</code>	New Century Schoolbook
<code>ppl</code>	Palatino
<code>ptm</code>	Times
<code>uncl</code>	Uncial
<code>put</code>	Utopia
<code>pzc</code>	Zapf Chancery

`\fontseries{série}`

Sélectionne une série de police. Une *série* combine une *graisse* et une *largeur*. Typiquement, une police prend en charge seulement quelques-unes des combinaisons possibles. Les combinaisons de valeurs de séries comprennent notamment :

m Moyen (normal)
 b Gras
 c Condensé
 bc Condensé Gras
 bx Gras étendu

Les valeurs possibles de graisses, considérées individuellement, sont :

ul Ultra maigre
 el Extra maigre
 l Maigre — ‘l’ pour *light* en anglais
 sl Semi maigre
 m Demi-gras — ‘m’ pour *medium* en anglais
 sb Semi gras
 b Gras — ‘b’ pour *bold* en anglais
 eb Extra gras
 ub Ultra gras

La valeurs possibles de largeur, considérées individuellement sont :

uc Ultra condensé
 ec Extra condensé
 c Condensé
 sc Semi condensé
 m Moyen
 sx Semi étendu
 x étendu
 ex Extra étendu
 ux Ultra étendu

Lorsqu’on forme la chaîne de caractères *série* à partir de la graisse et de la largeur, on laisse tomber le m qui signifie médium concernant la graisse ou la largeur, auquel cas on en utilise juste un (‘m’).

`\fontshape{allure}`

Sélectionne l’allure de police. Les allures valides sont :

n Droit (normal)
 it Italique
 sl Incliné (oblique) — ‘sl’ pour *slanted* en anglais.
 sc Petites capitales
 ui Italique droit — ‘ui’ pour *upright italic* en anglais
 ol Détouré — ‘ol’ pour *outline* en anglais.

Les deux dernières allures ne sont pas disponibles pour la plupart des familles de polices, et les petites capitales sont aussi souvent absentes.

`\fontsize{taille}{interligne}`

Réglage de la taille de police et de l’espacement d’interligne. L’unité des deux paramètres est par défaut le point (pt). L’espacement d’interligne est l’espace vertical nominal entre ligne, de ligne de base à ligne de base. Il est stocké dans le paramètre `\baselineskip`. La valeur par défaut de `\baselineskip` pour la police Computer Modern est 1,2 fois le `\fontsize`. Changer directement `\baselineskip` est déconseillé dans la mesure où sa valeur est réinitialisée

à chaque fois qu'un changement de taille se produit ; au lieu de cela utilisez `\baselinestretch` (voir Section 5.6 [`\baselineskip` & `\baselinestretch`], page 36).

`\linespread{facteur}`

Équivalent à `\renewcommand{\baselinestretch}{facteur}`, et donc doit être suivi de `\selectfont` pour prendre effet. Voir Section 5.6 [`\baselineskip` & `\baselinestretch`], page 36, pour plus ample information.

`\selectfont`

Les modifications apportées en appelant les commandes de fonte décrites ci-dessus n'entrent en vigueur qu'après que `\selectfont` est appelé, comme dans `\fontfamily{nomfamille}\selectfont`. La plupart du temps il est plus pratique de mettre cela dans une macro : `\newcommand*{\mapolice}{\fontfamily{nomfamille}\selectfont}` (voir Section 12.1 [`\newcommand` & `\renewcommand`], page 124).

`\usefont{codage}{famille}{série}{allure}`

Revient à la même chose que d'invoquer `\fontencoding`, `\fontfamily`, `\fontseries` et `\fontshape` avec les paramètres donnés, suivi par `\selectfont`. Par exemple :

```
\usefont{ot1}{cmr}{m}{n}
```

5 Mise en page

Commandes pour contrôler la disposition générale de la page.

5.1 `\onecolumn`

La déclaration `\onecolumn` commence une nouvelle page et produit une sortie à colonne unique. C'est la valeur par défaut.

5.2 `\twocolumn`

Synopsis :

```
\twocolumn [texte1col préliminaire]
```

Commence une nouvelle page et produit une sortie en deux colonnes. Si on a passé l'option de classe `twocolumn` au document alors c'est le comportement par défaut (voir Section 3.1 [Document class options], page 7). Cette commande est fragile (voir Section 12.11 [`\protect`], page 134).

Si l'argument optionnel *texte1col préliminaire* est présent, il est composé dans le mode à une colonne avant que la composition à deux colonnes ne commence.

Les paramètres ci-après contrôlent la composition de la sortie en deux colonnes :

`\columnsep`

La distance entre les colonnes. Vaut 35pt par défaut. Changer la valeur avec une commande telle que `\setlength{\columnsep}{40pt}`. Vous devez faire le changement avant que le mode en deux colonnes ne commence ; le préambule est une bonne place pour le faire.

`\columnseprule`

La largeur du filet entre les colonnes. La valeur par défaut est 0pt, de sorte qu'il n'y a pas de filet. Autrement, le filet apparaît à mi-chemin entre les deux colonnes. Changez la valeur avec une commande telle que `\setlength{\columnseprule}{0.4pt}`, avant que le mode en deux colonnes ne commence.

`\columnwidth`

La largeur d'une colonne. En mode à une colonne ceci vaut `\textwidth`. En mode à deux colonnes par défaut \LaTeX règle la largeur de chacune des deux colonnes, `\columnwidth`, à la moitié de `\textwidth` moins `\columnsep`.

Dans un document en deux colonnes, les environnements étoilés `table*` et `figure*` s'étendent en largeur sur deux colonnes, alors que les environnements non étoilés `table` et `figure` n'occupe qu'une colonne (voir Section 8.10 [figure], page 67, et voir Section 8.22 [table], page 96). \LaTeX place les flottants étoilés au sommet d'une page. Les paramètres ci-après contrôlent le comportement des flottants en cas de sortie à deux colonnes :

`\dbltopfraction`

Fraction maximale au sommet d'une page sur deux colonnes qui peut être occupée par des flottants s'étendant sur deux colonnes en largeur. Par défaut vaut 0,7, ce qui signifie que la hauteur d'un environnement `table*` ou `figure*` ne

doit pas excéder $0.7\text{\texttt{\textbackslash}textheight}$. Si la hauteur de vos environnement flottant excède cela alors vous pouvez prendre l'une des actions suivante pour l'empêcher de flotter jusqu'à la fin du document :

- Utiliser le spécificateur de localisation `[tp]` pour dire à L^AT_EX d'essayer de placer le flottant volumineux sur une page dédiée, ou de même au sommet d'une page.
- Utiliser le spécificateur de localisation `[t!]` pour l'emporter sur l'effet de `\dbltopfraction` pour ce flottant en particulier.
- Augmenter la valeur de `\dbltopfraction` à un nombre de grandeur adéquate, pour éviter de faire flotter les pages si tôt.

Vous pouvez le redéfinir, comme dans `\renewcommand{\dbltopfraction}{0.9}`.

`\dblfloatpagefraction`

Pour une page de flottants s'étendant sur deux colonnes en largeur, c'est la fraction minimum de la page qui doit être occupée par les flottants. La valeur par défaut de L^AT_EX est '0.5'. Se change avec `\renewcommand`.

`\dblfloatsep`

Sur une page de flottants s'étendant sur deux colonnes en largeur, cette longueur est la distance entre les flottants à la fois en haut et en bas de la page. Par défaut vaut '12pt plus2pt minus2pt' pour des documents en '10pt' ou '11pt', et '14pt plus2pt minus4pt' pour '12pt'.

`\dbltextfloatsep`

Cette longueur est la distance entre un flottant multi-colonnes en haut ou en bas d'une page et le texte principal. Par défaut vaut '20pt plus2pt minus4pt'.

`\dbltopnumber`

Sur une page de flottant avec des flottants large de deux colonnes, ce compteur donne le nombre maximal de flottants autorisé au sommet de la page. La valeur par défaut de L^AT_EX est 2.

L'exemple suivant utilise l'argument optionnel de `\twocolumn` pour crée un titre s'étendant sur les deux colonnes du document article :

```
\documentclass[twocolumn]{article}
\newcommand{\authormark}[1]{\textsuperscript{#1}}
\begin{document}
\twocolumn[{\% au sein de cet argument optionnel, du texte en une colonne
\centering
\LARGE Le titre \\[1.5em]
\large Auteur Un\authormark{1},
Auteur Deux\authormark{2},
Auteur Trois\authormark{1} \\[1em]
\normalsize
\begin{tabular}{p{.2\textwidth}@{\hspace{2em}}p{.2\textwidth}}
\authormark{1}Département Un & \authormark{2}Département Deux \\
École Un & \&École Deux
\end{tabular}\\[3em] \% espace sous la partie titre
```

}]

Le texte en deux colonnes est ici.

5.3 `\flushbottom`

Rend toutes les pages du document qui suivent cette déclaration de la même hauteur, en dilatant l'espace vertical là où c'est nécessaire pour remplir le page. On fait cela le plus souvent pour des documents en recto-verso car les différences entre pages vis-à-vis peuvent être flagrantes.

Si \TeX ne parvient pas à dilater de manière satisfaisante l'espace vertical dans une page alors on obtient un message du genre de `'Underfull \vbox (badness 10000) has occurred while \output is active'`. Lorsque on obtient cela, une option est de passer à `\raggedbottom` (voir Section 5.4 [`\raggedbottom`], page 32). Alternativement, vous pouvez ajuster la `textheight` pour rendre compatibles des pages, ou vous pouvez ajouter de l'espace vertical élastique entre les lignes ou entre les paragraphes, comme dans `\setlength{\parskip}{0ex plus0.1ex}`. L'option ultime est, dans une phase finale de rédaction, d'ajuster les hauteurs de page individuellement (voir Section 10.3 [`\enlargethispage`], page 118).

L'état `\flushbottom` est la valeur par défaut seulement lorsque on sélectionne l'option de classe `twocolumn` (voir Section 3.1 [Document class options], page 7), et pour les indexes fabriqués avec `makeidx`.

5.4 `\raggedbottom`

La déclaration `\raggedbottom` rend toutes les pages qui la suivent de la hauteur naturelle de la matière sur cette page ; aucune des longueurs élastiques n'est étirée. Ainsi, sur un document en recto-verso les pages en vis-à-vis sont potentiellement de hauteurs différentes. Cette commande peut être mise n'importe où dans le corps du document. Voir Section 5.3 [`\flushbottom`], page 32.

C'est le comportement par défaut à moins qu'on sélectionne l'option `twocolumn` de classe de document (voir Section 3.1 [Document class options], page 7).

5.5 Paramètres de mise en page

`\columnsep`

`\columnseprule`

`\columnwidth`

La distance entre les deux colonnes, la largeur d'un filet entre les colonnes, et la largeur des colonnes, lorsque l'option de classe de document `twocolumn` est en vigueur (voir Section 3.1 [Document class options], page 7). Voir Section 5.2 [`\twocolumn`], page 30.

`\headheight`

Hauteur de la boîte qui contient l'en-tête de page en cours. La valeur par défaut pour les classes `article`, `report`, et `book` est `'12pt'` quelle que soit la taille de la police.

`\headsep` La distance verticale entre le bas de la ligne d'en-tête et le haut du texte principal. La valeur par défaut est '25pt'. Dans la classe `book` la valeur par défaut est '0.25in' pour un document réglé en 10pt, et '0.275in' pour un document en 11pt ou 12pt.

`\footskip` Distance de la ligne de base de la dernière ligne de texte à la ligne de base du pied de page. La valeur par défaut dans les classes `article` et `report` est '30pt'. Dans la classe `book` la valeur par défaut est '0.35in' pour les documents réglés en 10pt, '0.38in' en 11pt, et '30pt' en 12pt.

`\linewidth` Largeur de la ligne actuelle, diminuée pour chaque `list` imbriquée (voir Section 8.16 [list], page 72). c.-à-d. que la valeur nominale `\linewidth` vaut `\textwidth` mais que pour chaque liste imbriquée la largeur `\linewidth` est diminuée de la somme de `\leftmargin` et `\rightmargin` pour cette liste (voir Section 8.14 [itemize], page 71).

`\marginparpush`

`\marginsep`

`\marginparwidth`

L'espace vertical minimal entre deux notes marginales, l'espace horizontal entre le corps du texte et les notes marginales, et la largeur horizontale des notes.

Normalement les notes en marginales apparaissent à l'extérieur de la page, mais la déclaration `\reversemarginpar` change cela (et `\normalmarginpar` le rétablit).

La valeur par défaut pour `\marginparpush` à la fois dans un document de classe `book` ou `article` est : '7pt' si le document est réglé à 12pt, et '5pt' s'il l'est à 11pt ou 10pt.

Pour `\marginsep`, dans la classe `article` la valeur par défaut est '10pt' sauf si le document est réglé à 10pt et en mode en deux-colonnes où la valeur par défaut est '11pt'.

Pour `\marginsep` dans la classe `book` la valeur par défaut est '10pt' en mode deux-colonnes, et '7pt' en mode mono-colonne.

Pour `\marginparwidth` aussi bien dans les classes `book` que `article`, en mode deux-colonnes la valeur par défaut est 60% ou `\paperwidth - \textwidth`, alors qu'en mode monocolonne c'est 50% de cette distance.

`\oddsidemargin`

`\evensidemargin`

La *marge gauche* est la bande entre le bord gauche de la page et le corps du texte.

La *marge gauche nominale* est la marge gauche lorsque les paramètres `\oddsidemargin` et `\evensidemargin` sont à '0pt'. Sa frontière est une ligne verticale imaginaire située à une distance de `1in + \hoffset` du bord gauche de la page en allant vers le côté droit. La longueur `\hoffset` vaut 0pt, ce paramètre est obsolète il servait dans les années 1980, quand le système T_EX a été conçu, à rattraper les défauts des imprimantes de l'époque dont le

placement horizontal de la page par rapport à la feuille de papier pouvait se dérégler.

La longueur `\oddsidemargin` est la distance entre le bord gauche du texte et la frontière de la marge gauche nominale pour les pages de numéro impair dans les documents en recto-verso, et pour toutes les pages sinon pour les documents en recto simple. La longueur `\evensidemargin` est la distance entre le bord gauche du texte et la frontière de la marge gauche nominale pour les pages de numéro pair dans les documents en recto-verso.

En d'autres termes `\oddsidemargin` ou `\evensidemargin` donnent la différence entre la largeur de la marge gauche réel et la marge gauche nominale. Ces distances `\oddsidemargin` et `\evensidemargin` sont donc signées vers le coté droit, c'est à dire qu'une valeur positive indique que la marge gauche réelle est plus large que la marge gauche nominale, alors qu'une valeur négative indique qu'au contraire elle est plus étroite.

Dans un document en recto-verso le *côté intérieur* de la page est le côté gauche pour les pages de numéro impair, et le côté droit pour les pages de numéro pair. Le *côté extérieur* est le côté opposé au côté intérieur : la longueur `\oddsidemargin` règle donc la position de la marge intérieure des pages de numéro impair, alors que `\evensidemargin` règle la position la marge extérieur des pages de numéro pair.

Pour une marge intérieure de 4cm dans un document en recto-verso, mettre le code suivant dans le préambule :

```
\evensidemargin\dimexpr\paperwidth-\textwidth-1in-4cm
\oddsidemargin\dimexpr4cm-1in
```

Vice-versa, pour une marge extérieure de 5cm dans un document en recto-verso, mettre le code suivant dans le préambule :

```
\evensidemargin\dimexpr5cm-1in
\oddsidemargin\dimexpr\paperwidth-\textwidth-1in-5cm
```

`\paperheight`

La hauteur du papier, à ne pas confondre avec la hauteur de la zone d'impression. Elle est normalement réglée avec une option de classe de document, comme dans `\documentclass[a4paper]{article}` (voir Section 3.1 [Document class options], page 7).

`\paperwidth`

La largeur du papier, à ne pas confondre avec la largeur de la zone d'impression. Elle est normalement réglée avec une option de classe de document, comme dans `\documentclass[a4paper]{article}` (voir Section 3.1 [Document class options], page 7).

`\textheight`

La hauteur verticale normale du corps de la page. Si le document réglé à une taille de police nominale de 10pt, alors pour un document de classe `article` ou `report`, elle vaut `'43\baselineskip'` ; pour une classe `book`, elle vaut `'41\baselineskip'`. Pour 11pt, c'est `'38\baselineskip'` pour toutes les classes de document. Pour `'12pt'` c'est `'36\baselineskip'` pour toutes les classes aussi.

`\textwidth`

La largeur horizontale totale de l'ensemble du corps de la page. Pour un document de classe `article` ou `report`, elle vaut `'345pt'` à 10pt de taille de police, `'360pt'` à 11pt, et `'390pt'` à 12pt. Pour un document `book`, elle vaut `'4.5in'` à 10pt et `'5in'` à 11pt ou 12pt.

En sortie multi-colonne, `\textwidth` reste de la largeur de tout le corps de la page, tandis que `\columnwidth` est la largeur d'une colonne (voir Section 5.2 [`\twocolumn`], page 30).

Dans les listes (voir Section 8.16 [`list`], page 72), `\textwidth` demeure la largeur du corps de la page entière (et `\columnwidth` est la largeur d'une colonne entière), alors que `\linewidth` peut diminuer au sein de listes imbriquées.

À l'intérieur d'une minipage (voir Section 8.18 [`minipage`], page 79,) ou `\parbox` (voir Section 20.3 [`\parbox`], page 206), tous les paramètres liés à la largeur sont mis à la largeur spécifiée, et de reviennent à leurs valeurs normales à la fin du `minipage` ou `\parbox`.

`\hsize`

Par souci d'exhaustivité : `\hsize` est le paramètre \TeX primitif utilisé lorsque le texte est divisé en lignes. Il ne devrait pas être utilisé dans des documents \LaTeX en conditions normales.

`\topmargin`

La *marge supérieure* est la bande entre le bord supérieur de la page et le bord supérieur de l'en-tête de page.

La *marge supérieure nominale* est la marge supérieure lorsque `\topmargin` vaut `'0pt'`. Sa frontière est une ligne horizontale imaginaire située à une distance de `1in + \voffset` du bord supérieur de la page en allant vers le bas. La longueur `\voffset` vaut `0pt`, ce paramètre est obsolète il servait dans les années 1980, quand le système \TeX a été conçu, à rattraper les défauts des imprimantes de l'époque dont le placement vertical de la page par rapport à la feuille de papier pouvait se dérégler.

La distance `\topmargin` sépare le haut de l'en-tête de page de la frontière de la marge supérieure nominale. Elle est signée vers le bas, c'est à dire que quand `\topmargin` est positif la marge supérieure est plus large que la marge supérieure nominale, et que s'il est négatif elle est au contraire plus étroite. En d'autres termes `\topmargin` est la différence entre la largeur de la marge supérieure réelle et la marge supérieure nominale.

L'espace le haut de la page \TeX et le haut du corps de texte est `1in + \voffset + \topmargin + \headheight + \headsep`. La marge inférieure vaut donc `\paperheight - \textheight - \footskip - 1in - \voffset - \topmargin - \headheight - \headsep`.

`\topskip`

La distance minimale entre le sommet du corps de la page et la ligne de base de la première ligne de texte. Pour les classes standard, la valeur par défaut est la même que la taille de la police, par exemple, `'10pt'` à `'10pt'`.

5.6 `\baselineskip` & `\baselinestretch`

La variable `\baselineskip` est une longueur élastique (voir Chapitre 14 [Lengths], page 141). Elle donne l'*interligne*, la distance normale entre les lignes d'un paragraphe, de ligne de base à ligne de base.

D'ordinaire un auteur de documents ne change pas directement `\baselineskip` en écrivant. Au lieu de cela, l'interligne est réglé par la commande de bas niveau de sélection de fonte `\fontsize` (voir [low level font commands fontsize], page 28). La valeur de `\baselineskip` est réinitialisée chaque fois qu'un changement de fonte survient et ainsi tout changement direct de `\baselineskip` serait anéanti dès la prochaine commutation de fonte. Pour la façon d'influencer l'espacement des lignes, voir la discussion de `\baselinestretch` ci-dessous.

D'habitude, une taille de fonte et un interligne sont assignés par l'auteur de la fonte. Ces nombres sont nominaux au sens où si, par exemple, un fichier de style de fonte comprend la commande `\fontsize{10pt}{12pt}` alors cela ne veut pas dire que les caractères dans la fonte font 10 pt de haut ; par exemple, les parenthèses et les capitales accentuées pourraient être plus hautes. Non plus cela ne signifie que si les lignes sont espacées de moins de 12 pt alors elles risquent de se toucher. Ces nombres sont plutôt des jugements typographiques. (Souvent, le `\baselineskip` est environ plus grand de 20 % que la taille de fonte).

Le `\baselineskip` n'est pas une propriété de chaque ligne, mais du paragraphe entier. Il en résulte, que du texte de grande taille au milieu d'un paragraphe, tel qu'un simple `{\Huge Q}`, est serré sur sa ligne. \TeX s'assure qu'il ne racle pas sur la ligne du dessus mais il ne change par le `\baselineskip` pour que cette ligne-là soit surplombée d'un espace supplémentaire. Pour résoudre ce problème, utilisez un `\strut` (voir Section 19.13 [strut], page 200).

La valeur de `\baselineskip` que \TeX utilise pour les paragraphes est celle en vigueur à l'endroit de la commande ou ligne à blanc qui termine l'unité de paragraphe. Ainsi, si un document contient le paragraphe ci-dessous alors ses lignes seront recroquevillées ensemble, comparées au lignes des paragraphes alentour.

Beaucoup de gens considèrent les sauts de page entre du texte est une équation affichée comme du mauvais style, alors qu'en fait l'affiche fait partie du paragraphe. Étant donné que l'affiche ci-dessous est en taille de fonte `footnotesize`, l'entièreté du paragraphe a un espacement d'interligne correspondant à cette taille. `{\footnotesize $$a+b = c$$}`

Le procédé de fabrication des paragraphes est que quand une nouvelle ligne est ajoutée, si la somme de la profondeur de la ligne précédente et de la hauteur de la nouvelle ligne est inférieure à `\baselineskip` alors \TeX insère une glue verticale en quantité suffisante pour faire la différence. Il y a deux points délicats. Le premier est que au cas où les lignes seraient trop proches l'une de l'autre, plus proches que `\lineskiplimit`, alors \TeX au lieu de cela utilise `\lineskip` comme la glue d'interligne. Le second est que \TeX n'utilise pas vraiment la profondeur de la ligne précédente. Au lieu de cela il utilise `\prevdepth`, ce qui d'ordinaire contient cette profondeur. Mais au début d'un paragraphe, (ou de toute liste verticale) ou juste après un filet, `\prevdepth` a la valeur de -1000 pt et cette valeur spéciale dit à \TeX de ne pas insérer de glue d'interligne au démarrage du paragraphe.

Dans les classes standardes `\lineskiplimit` vaut 0 pt et `\lineskip` vaut 1 pt. Selon le procédé exposé dans le paragraphe précédent, la distance entre les lignes peut approcher

zéro, mais si elle devient zéro (ou moins que zéro) alors un écartement de 1 pt est appliqué aux lignes.

Il arrive qu'un auteur doive, à des fins d'édition, mettre le document en double interligne, ou bien en interligne un-et-demi. La façon correcte d'influencer la distance d'interligne est au travers de `\baselinestretch` qui dilate `\baselineskip`, et a une valeur par défaut de '1.0'. C'est une commande, et non une longueur, ainsi on change le facteur d'échelle comme dans `\renewcommand{\baselinestretch}{1.5}\selectfont`.

La façon la plus directe de changer l'interligne pour tout un document est de placer `\linespread{facteur}` dans le préambule. Pour un double espacement, prenez *facteur* à '1.6' et pour un espacement un-et-demi utilisez '1.3'. Ces nombres sont approximatifs : par exemple puisque `\baselineskip` vaut environ 1,2 fois la taille de fonte, le multiplier par 1,6 donne un rapport entre l'interligne et la taille de fonte d'environ 2. (La commande `\linespread` est définie comme `\renewcommand{\baselinestretch}{facteur}` de sorte que son effet n'entre en vigueur que lorsqu'un réglage de fonte survient. Mais cela a toujours lieu au démarrage d'un document, de sorte que là vous n'avez pas besoin de la faire suivre d'un `\selectfont`).

Une approche plus simple est fournie par le paquetage `setspace`. Voici un exemple de base :

```
\usepackage{setspace}
\doublespacing % ou \onehalfspacing pour 1,5
```

Placé dans le préambule ces déclarations démarreront le document dès le début avec ces réglages de taille. Mais vous pouvez aussi les utiliser dans le corps du document pour changer l'espacement à partir de ce point, et par conséquent il y a `\singlespacing` pour revenir à l'espacement normal. Dans le corps du document, une pratique meilleure qu'utiliser ces déclarations est d'utiliser les environnements, tels que `\begin{doublespace} ... \end{doublespace}`. Ce paquetage fournit aussi des commandes pour faire un espacement quelconque : `\setstretch{factor}` et `\begin{spacing}{factor} ... \end{spacing}`. Ce paquetage conserve également un interligne simple là où c'est typiquement désirable, comme dans les notes de bas de page et dans les légendes de figure. Voir la documentation du paquetage.

5.7 Flottants

Certains éléments typographiques, tels que les figures et les tableaux, ne peuvent pas être à cheval sur plusieurs pages. Ils doivent être composés en dehors du flux normal du texte, par exemple flottant au sommet d'une page ultérieure

\LaTeX sait gérer plusieurs classes de matière flottante. Il y a deux classes définies par défaut, `figure` (voir Section 8.10 [figure], page 67) et `table` (voir Section 8.22 [table], page 96), mais vous pouvez créer une nouvelle classes avec le paquetage `float`.

Au sein d'une même classe flottante \LaTeX respecte l'ordre, de sorte que la première figure dans le code source d'un document est toujours composée avant la deuxième figure. Cependant, \LaTeX peut mélanger les classes, ainsi il peut se produire qu'alors que le premier tableau apparaît dans le code source avant la première figure, il apparaisse après elle dans le fichier de sortie.

Le placement des flottants est l'objet de paramètres, donnés ci-dessous, qui limitent le nombre de flottants pouvant apparaître au sommet d'une page, et au bas de page, etc. Si

à cause d'un nombre trop important de flottants mis en queue ces limites les empêchent de tenir tous dans une seule page, alors L^AT_EX place ce qu'il peut et diffère le reste à la page suivante. De la sorte, les flottants peuvent être composés loin de leur place d'origine dans le code source. En particulier, un flottant qui prend beaucoup de place peut migrer jusqu'à la fin du document. Mais alors, parce que tous les flottants dans une classe doivent apparaître dans le même ordre séquentiel, tous les flottants suivant dans cette classe apparaissent aussi à la fin.

En plus de changer les paramètres, pour chaque flottant vous pouvez peaufiner l'endroit où l'algorithme de placement des flottants essaie de le placer en utilisant sont argument *placement*. Les valeurs possibles sont une séquence des lettres ci-dessous. La valeur par défaut pour à la fois `figure` et `table`, dans les deux classes de document `article` et `book`, est `tbp`.

- `t` (pour Top) — au sommet d'une page de texte.
- `b` (pour Bottom) — au bas d'une page de texte. (Cependant, `b` n'est pas autorisé avec des flottants en pleine-largeur (`figure*`) en cas de sortie à double-colonne. Pour améliorer cela, on peut utiliser les paquetages `stfloats` ou `dblfloatfix`, mais voyez la discussion sur les avertissements dans la FAQ : <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=2colfloat>.
- `h` (pour « Here », c.-à-d.« Ici » en anglais) — à la position du texte où l'environnement `figure` apparaît. Cependant, `h` n'est pas autorisé en soi-même ; `t` est ajouté automatiquement.
Pour forcer à tout prix un flottant à apparaître « ici », vous pouvez charger le paquetage `float` et le spécificateur `H` qui y est défini. Pour plus ample discussion, voir l'entrée de FAQ à <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=figurehere>.
- `p` (pour Page de flottants) — sur une *page de flottants* séparée, qui est une page ne contenant pas de texte, seulement des flottants.
- `!` Utilisé en plus de l'un des spécificateurs précédents ; pour ce flottant seulement, L^AT_EX ignore les restrictions à la fois sur le nombre de flottants qui peuvent apparaître et les quantités relatives de texte flottant et non-flottant sur la page. Le spécificateur `!` *ne* signifie *pas* « mets le flottant ici » ; voir plus haut.

Note : l'ordre dans lequel les lettres apparaissent au sein du paramètre *placement* ne change pas l'ordre dans lequel L^AT_EX essaie de placer le flottant ; par exemple `bt

a le même effet que tbp. Tout ce que placement fait c'est que si une lettre n'est pas présente alors l'algorithme n'essaie pas cet endroit. Ainsi, la valeur par défaut de LATEX étant tbp consiste à essayer toutes les localisations sauf celle de placer le flottant là où il apparaît dans le code source.`

Pour empêcher L^AT_EX de rejeter tous les flottants à la fin du document ou d'un chapitre, vous pouvez utiliser la commande `\clearpage` pour commencer une nouvelle page et insérer tous les flottants pendants. Si un saut de page est indésirable alors vous pouvez charger le paquetage `afterpage` et commettre le code `\afterpage{\clearpage}`. Ceci aura l'effet d'attendre la fin de la page courante et ensuite de faire passer tous les flottants encore non placés.

L^AT_EX peut composer un flottant avant l'endroit où il apparaît dans le code source (quoique sur la même page de sortie) s'il y a un spécificateur `t` au sein du paramètre *placement*. Si ceci n'est pas désiré, et que supprimer `t` n'est acceptable car ce spécificateur empêche le flottant d'être placé au sommet de la page suivante, alors vous pouvez empêcher cela soit en utilisant le paquetage `flafter` ou en utilisant ou en utilisant la commande `\suppressfloats[t]`, ce qui entraîne le déplacement vers la page suivante des flottants qui auraient du être placés au sommet de la page courante.

Voici les paramètres en relation aux fractions de pages occupées par du texte flottant et non flottant (on peut les changer avec `\renewcommand{paramètre}{nombre décimal entre 0 et 1}`) :

La fraction maximale de page autorisée à être occupées par des flottants au bas de la page ; la valeur par défaut est `'.3'`.

`\floatpagefraction`

La fraction minimale d'une page de flottants qui doit être occupée par des flottants ; la valeur par défaut est `'.5'`.

`\textfraction`

La fraction minimale d'une page qui doit être du texte ; si des flottants prennent trop d'espace pour préserver une telle quantité de texte, alors les flottants sont déplacés vers une autre page. La valeur par défaut est `'.2'`.

`\topfraction`

Fraction maximale au sommet d'une page que peut être occupée avant des flottants ; la valeur par défaut est `'.7'`.

Les paramètres en relation à l'espace vertical autour des flottants (on peut les changer avec `\setlength{paramètre}{expression longueur}`) :

`\floatsep`

Espace entre les flottants au sommet ou au bas d'une page ; par défaut vaut `'12pt plus2pt minus2pt'`.

`\intextsep`

Espace au dessus et au dessous d'un flottant situé au milieu du texte principal ; vaut par défaut `'12pt plus2pt minus2pt'` pour les styles à `'10pt'` et à `'11pt'`, et `'14pt plus4pt minus4pt'` pour `'12pt'`.

`\textfloatsep`

Espace entre le dernier (premier) flottant au sommet (bas) d'une page ; par défaut vaut `'20pt plus2pt minus4pt'`.

Paramètres en relation avec le nombre de flottant sur une page (on peut les changer avec `\setcounter{ctrname}{natural number}`) :

`\bottomnumber`

Nombre maximal de flottants pouvant apparaître au bas d'une page de texte ; par défaut 1.

`\topnumber`

Nombre maximal de flottants pouvant apparaître au sommet d'une page de texte ; par défaut 2.

\totalnumber

Nombre maximal de flottants pouvant apparaître sur une page de texte ; par défaut 3.

L'article principal de FAQ T_EX en rapport avec les flottants <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=floats> contient des suggestions pour relâcher les paramètres par défaut de L^AT_EX de sorte à réduire les problèmes de flottants rejetés à la fin. Une explication complète de l'algorithme de placement des flottants se trouve dans l'article de Frank Mittelbach « How to influence the position of float environments like figure and table in L^AT_EX? » (<http://latex-project.org/papers/tb111mitt-float.pdf>).

5.7.1 \caption

Synopsis :

```
\caption{texte-légende}
```

ou

```
\caption[texte-légende-bref]{texte-légende}
```

Fabrique une légende pour un environnement flottant, tel que les environnements **figure** ou **table** (voir Section 8.10 [figure], page 67, ou Section 8.22 [table], page 96).

Dans l'exemple suivant, L^AT_EX place une légende sous l'espace blanc vertical laissé par l'auteur pour l'inclusion ultérieure d'une image.

```
\begin{figure}
  \vspace*{1cm}
  \caption{Alonzo Cushing, Battery A, 4th US Artillery.}
  \label{fig:CushingPic}
\end{figure}
```

La commande `\caption` crée une étiquette pour la légende *texte-légende* du genre de 'Figure 1 -- ' pour un document **article** ou 'Figure 1.1 -- ' pour un document **book**. Le texte est centré s'il est plus court que la largeur du texte, ou composé comme un paragraphe sans retrait s'il prend plus d'une ligne.

En plus de placer le *texte-légende* dans la sortie, la commande `\caption` sauvegarde également cette information pour qu'elle soit utilisée dans une liste de figures ou un liste de tableaux (voir Section 25.1 [Table of contents etc.], page 242).

Ci-dessous la commande `\caption` utilise l'argument optionnel *texte-légende-bref*, de sorte que le texte plus bref apparaisse dans la liste des tableaux plutôt que la version longue *texte-légende*.

```
\begin{table}
  \centering
  \begin{tabular}{|*{3}{c}|}
    \hline
    4 & 9 & 2 \\
    3 & 5 & 7 \\
    8 & 1 & 6 \\
    \hline
  \end{tabular}
  \caption[Carré de \textit{Lo Shu}]{%
```

```

    Le carré de \textit{Lo Shu}, il est unique parmi les carrés
    d'ordre trois à la rotation ou réflexion près.}
  \label{tab:LoShu}
\end{table}

```

\LaTeX crée une étiquette pour *texte-légende* du genre de ‘Table 1 -- ’ pour un document *article* ou ‘Table 1.1 -- ’ pour un document *book*.

La légende peut apparaître au sommet d’une **figure** ou **table**. Cela se produirait ainsi dans l’exemple précédent en mettant la `\caption` entre le `\centering` et le `\begin{tabular}`.

Les différents environnements flottants sont numérotés séparément, par défaut. Le compteur de l’environnement **figure** est nommé `figure`, et de même le compteur de l’environnement **table** est `table`.

Le texte qui est mis dans la liste des figures ou des tableaux est un argument mouvant. Si vous obtenez l’erreur \LaTeX ‘! Argument of `\caption` has an extra ’ alors vous devez précéder d’un `\protect` toute commande fragile. Voir Section 12.11 [`\protect`], page 134.

Le paquetage `caption` a beaucoup d’options pour ajuster l’apparence de la légende, par ex. changer la taille de la fonte, faire que la légende soit un texte en retrait plutôt qu’un paragraphe, ou faire que la légende soit toujours un paragraphe, plutôt qu’un texte centré quand elle est courte.

6 Rubricage

Structure votre texte en rubriques : parties, chapitres, sections, etc. Toutes les commandes de rubricage ont la même forme, l’une parmi :

```
cmd-de-rubricage{titre}
cmd-de-rubricage*{titre}
cmd-de-rubricage[titre-tdm]{titre}
```

Par exemple, déclarez le début d’une sous-section comme dans `\subsection{Motivation}`.

Le tableau suivant comprend chacune des commandes *cmd-de-rubricage* de L^AT_EX. Toutes sont disponibles dans toutes les classes de document standards de L^AT_EX `book`, `report`, et `article`, à ceci près que `\chapter` n’est pas disponible dans `article`.

Type de rubrique	Commande	Niveau
Part	<code>\part</code>	-1 (<code>book</code> , <code>report</code>), 0 (<code>article</code>)
Chapter	<code>\chapter</code>	0
Section	<code>\section</code>	1
Subsection	<code>\subsection</code>	2
Subsubsection	<code>\subsubsection</code>	3
Paragraph	<code>\paragraph</code>	4
Subparagraph	<code>\subparagraph</code>	5

Toutes ces commandes ont une forme en `*`, aussi appelée étoilée, qui imprime *titre* comme d’habitude mais sans le numéroté et sans fabriquer une entrée dans la table des matières. Un exemple d’utilisation est pour un appendice dans un `article`. Entrer le code `\appendix\section{Appendice}` donne en sortie ‘A Appendix’ (voir Section 6.6 [Appendix], page 49). Vous pouvez ôter le numéro ‘A’ en entrant à la place `\section*{Appendice}` (les articles omettent couramment d’avoir une table des matières et ont des en-têtes de pages simples alors les autres différences de la commande `\section` peuvent être négligées).

Le titre *titre* fournit la rubrique en tant que titre dans le texte principal, mais il peut également apparaître dans la table des matières et le haut et le bas de page (voir Chapitre 18 [Page styles], page 187). Vous pourriez désirer un texte différent à ces endroits que dans le texte principal. Toute ces commandes ont un argument optionnel *tdm-titre* destiné à ces autres endroits.

Le numéro dans la colonne «Niveau» de la table ci-dessus détermine quelles rubriques sont numérotées, et lesquelles apparaissent dans la table des matières. Si le *niveau* de la commande de rubricage est inférieur ou égal à la valeur du compteur `secnumdepth` alors les rubriques correspondant à ces commandes de rubricage sont numérotées (voir [Sectioning/secnumdepth], page 43). Et, si *niveau* est inférieur ou égal à la valeur du compteur `tocdepth` alors la table des matières comprend une entrée pour cette rubrique (voir [Sectioning/tocdepth], page 43).

L^AT_EX s’attend que avant d’avoir une `\subsection` vous ayez une `\section` et, dans un document de classe `book`, qu’avant une `\section` vous ayez un `\chapter`. Autrement vous pourriez obtenir quelque chose comme une sous-section numérotée ‘3.0.1’.

L^AT_EX vous permet de changer l’apparence des rubriques. Un exemple simple de ceci est que vous pouvez mettre le numéro de rubrique de type section en lettres majuscules avec `\renewcommand\thesection{\Alph{section}}` dans le préambule (voir Section 13.1

[`\alph \Alph \arabic \roman \Roman \fnsymbol`], page 138). Le CTAN a beaucoup de paquetages rendant ce genre d’ajustement plus facile, notamment `titlesec`.

Deux compteurs sont en relation avec l’apparence des rubriques fabriquées par les commandes de rubricage.

`secnumdepth`

Le compteur `secnumdepth` contrôle quels titres de rubriques sont numérotés. Régler le compteur avec `\setcounter{secnumdepth}{niveau}` supprime la numérotation des rubriques à toute profondeur supérieure à `niveau` (voir Section 13.4 [`\setcounter`], page 139). Voir la table plus haut pour la valeur des niveaux. Par exemple, si le `secnumdepth` vaut 1 dans un `article` alors la commande `\section{Introduction}` produit en sortie quelque chose comme ‘1 Introduction’ alors que `\subsection{Discussion}` produit quelque chose comme ‘Discussion’, sans numéro. La valeur par défaut de \LaTeX pour `secnumdepth` vaut 3 dans la classe `article` et 2 dans les classes `book` et `report`.

`tocdepth` Contrôle quelles rubriques sont listées dans la table des matières. Régler `\setcounter{tocdepth}{niveau}` a pour effet que les rubriques au niveau `niveau` sont celles de plus petit niveau à être listées (voir Section 13.4 [`\setcounter`], page 139). Voir la table ci-dessus pour les numéros de niveau. Par exemple, if `tocdepth` vaut 1 alors la table des matières contiendra les rubriques issues de `\section`, mais pas celles de `\subsection`. La valeur par défaut de \LaTeX pour `tocdepth` vaut 3 dans la classe `article` et 2 dans les classes `book` et `report`.

6.1 `\part`

Synopsis, l’un parmi :

```
\part{titre}
\part*{titre}
\part[tdm-titre]{titre}
```

Début une partie de document. Les classes standard de \LaTeX `book`, `report`, et `article` offrent toutes cette commande.

L’exemple suivant produit une partie de document dans un livre :

```
\part{VOLUME I \\\
      PERSONAL MEMOIRS OF U.\ S.\ GRANT}
\chapter{ANCESTRY--BIRTH--BOYHOOD.}
My family is American, and has been for generations,
in all its branches, direct and collateral.
```

Dans chaque classe standard la commande `\part` produit en sortie un numéro de partie tel que ‘Première partie’, seul sur sa ligne, en caractère gras, et en gros caractères. Ensuite \LaTeX produit en sortie *titre*, également seule sur sa ligne, en caractère gras et en caractères encore plus gros. Dans la classe `book`, le comportement par défaut de \LaTeX est de mettre chaque titre de partie seule sur sa propre page. Si le livre est en recto-verso alors \LaTeX saute une page si nécessaire pour que la nouvelle partie commence sur une page à numéro impair. Dans un `report` il est également seul sur une page, mais \LaTeX ne force pas qu’elle

soit de numéro impair. Dans un `article` L^AT_EX ne le place pas sur une nouvelle page, mais au lieu de cela produit en sortie le numéro de partie et le titre de partie sur la même page que le document principal.

La forme en `*` imprime *titre* mais n'imprime pas le numéro de partie, et n'incrémente pas le compteur `part`, et ne produit aucune entrée dans la table des matières.

L'argument optionnel *tdm-titre* apparaît comme le titre de la partie dans la table des matières (voir Section 25.1 [Table of contents etc.], page 242) et dans les hauts de pages (voir Chapitre 18 [Page styles], page 187). S'il n'est pas présent alors *titre* est utilisé à sa place. Dans l'exemple suivante on met un saut de ligne dans *titre* mais on l'enlève dans la table des matières.

```
\part[Up from the bottom; my life]{Up from the bottom\\ my life}
```

Pour déterminer quelles rubrique sont numéroté et lesquelles apparaissent dans la table des matières, le numéro de niveau d'une partie vaut `-1` (voir [Sectioning/secnumdepth], page 43, et voir [Sectioning/tocdepth], page 43).

Dans la classe `article`, si un paragraphe suit immédiatement le titre de partie alors il n'est pas renforcé. Pour obtenir un renforcement vous pouvez utiliser le paquetage `indentfirst`.

Un paquetage pour changer le comportement de `\part` est `titlesec`. Voir sa documentation sur le CTAN.

6.2 \chapter

Synopsis, l'un parmi :

```
\chapter{titre}
\chapter*{titre}
\chapter[tdm-titre]{titre}
```

Commence un chapitre. Les classes standards L^AT_EX `book` et `report` ont cette commande, mais `article` non.

L'exemple suivant produit un chapitre.

```
\chapter{Mirages}
Appelez moi Ismaël.
Voici quelques années --- peu importe combien --- le porte-monnaie vide
ou presque, rien ne me retenant à terre, je songeai à naviguer un peu et
à voir l'étendue liquide du globe.
```

Le comportement par défaut de L^AT_EX est de commence chaque chapitre sur une page neuve, une page à numéro impair si le document est en recto-verso. Il produit un numéro de chapitre tel que 'Chapitre 1' en gros caractère gras (la taille est `\huge`). Ensuite il place le *titre* sur une nouvelle ligne, en caractère gras encore plus gros (taille `\Huge`). Il incrémente également le compteur `chapter`, ajoute une entrée à la table des matières (voir Section 25.1 [Table of contents etc.], page 242), et règle l'information de haut de page (voir Chapitre 18 [Page styles], page 187).

La forme étoilée, ou forme en `*`, affiche *titre* sur une nouvelle ligne, en caractère gras. Mais elle n'affiche pas le numéro de chapitre, ni n'incrémente le compteur `chapter`, et ne produit aucune entrée dans la table des matières, et n'affecte pas le haut de page. (Si vous

utilise le style de page `headings` dans un document recto-verso alors le haut de page sera dérivé du chapitre précédent). Voici un exemple illustrant cela :

```
\chapter*{Préambule}
```

L'argument optionnel *tdm-titre* apparaît comme titre de chapitre dans la table des matières (voir Section 25.1 [Table of contents etc.], page 242) et dans les hauts de page (voir Chapitre 18 [Page styles], page 187). Si il n'est pas présent alors *titre* sera à la place. L'exemple suivant montre le nom complet dans le titre de chapitre,

```
\chapter[Weyl]{Hermann Klaus Hugo (Peter) Weyl (1885--1955)}
```

mais seulement 'Weyl' sur la page de table des matières. L'exemple suivant place un saut de ligne dans le titre, mais ceci ne fonctionne pas bien avec les hauts de page, alors il omet le saut dans la table des matières

```
\chapter[J'ai tout donné ; mon histoire]{J'ai tout donné\\ mon histoire}
```

Pour déterminer quelles rubriques sont numérotées et lesquelles apparaissent dans la table des matières, le numéro de niveau d'un chapitre est 0 (voir [Sectioning/secnumdepth], page 43, et voir [Sectioning/tocdepth], page 43).

Lorsque vous chargez un paquetage d'internationalisation tel que `babel`, `mlp` ou `polyglossia`, et que la langue sélectionnée est le français, alors le premier paragraphe après le titre est normalement renforcé, comme c'est la convention en typographie française. Par contre, si vous restez en langue par défaut, c.-à-d. en anglo-américain, le paragraphe qui suit le titre de chapitre n'est pas renforcé, étant donné que c'est pour l'anglo-américain une pratique typographique standard. Pour obtenir un renforcement dans ce cas utilisez le paquetage `indentfirst`.

Vous pouvez changer ce qui est affiché pour le numéro de chapitre. Pour le changer en quelque chose du genre de 'Cours 1', placez dans le préambule soit `\renewcommand{\chaptername}{Cours}`, soit cela (voir Section 12.3 [`\makeatletter` & `\makeatother`], page 127) :

```
\makeatletter
\renewcommand{\@chapapp}{Cours}
\makeatother
```

Pour que cela soit fonction de la langue principale du document, voir le paquetage `babel`.

Dans un document recto-verso \LaTeX commence les chapitres sur des pages de numéro impair, en laissant si nécessaire une page de numéro pair blanche à l'exception d'un éventuel haut de page. Pour que cette page soit complètement blanche, voir Section 10.1 [`\clearpage` & `\cleardoublepage`], page 117.

Pour changer le comportement de la commande `\chapter`, vous pouvez copier sa définition depuis le fichier de format \LaTeX et faire des ajustements. Mais il y a aussi beaucoup de paquetage sur le CTAN qui traitent de cela. L'un d'eux est `titlesec`. Voir sa documentation, mais l'exemple ci-dessous donne un aperçu de ce qu'il peut faire.

```
\usepackage{titlesec} % dans le préambule
\titleformat{\chapter}
  {\Huge\bfseries} % format du titre
  {} % étiquette, tel que 1.2 pour une sous-section
  {0pt} % longueur de séparation entre l'étiquette et le titre
  {} % code crochet exécuté avant
```

Ceci omet le numéro de chapitre ‘Chapitre 1’ de la page, mais contrairement à `\chapter*` cela conserve le chapitre dans la table des matières et dans les hauts de page.

6.3 `\section`

Synopsis, l’un parmi :

```
\section{titre}
\section*{titre}
\section[tdm-titre]{titre}
```

Commence une section. Les classes L^AT_EX standards `article`, `book`, et `report` ont toutes cette commande.

L’exemple suivant produit une section :

```
Dans cette partie nous nous intéressons plutôt à la fonction, au
comportement d’entrée-sortie, qu’aux détails de la réalisation de ce
comportement.
```

```
\section{Machines de Turing}
```

```
En dépit de ce désir de rester évasif sur l’implémentation, nous suivons
l’approche d’A.~Turing selon laquelle la première étape pour définir
l’ensemble des fonctions calculables est de réfléchir au détails de ce
que des mécanismes peuvent faire.
```

Pour les classes standards L^AT_EX `book` et `report` la sortie par défaut est du genre de ‘1.2 *titre*’ (pour chapitre 1, section 2), seul sur sa ligne et fer à gauche, en caractères gras et plus gros (la taille de police est `\Large`). La même chose vaut pour `article` à ceci près qu’il n’y a pas de chapitre dans cette classe, et donc cela ressemble à ‘2 *titre*’.

La forme en `*` affiche *titre*. Mais elle n’affiche pas le numéro de section, ni n’incrémente le compteur `section`, ne produit aucune entrée dans la table des matières, et n’affecte pas le haut de page. (Si vous utilisez le style de page `headings` dans un document recto-verso, alors le titre de haut de page sera celui de la rubrique précédente).

L’argument optionnel `tdm-titre` apparaît comme titre de section dans la table des matières (voir Section 25.1 [Table of contents etc.], page 242) et dans les titres de haut de page (voir Chapitre 18 [Page styles], page 187). S’il n’est pas présent alors *titre* est à la place. L’exemple suivant montre le nom complet dans le titre de la section,

```
\section[Elisabeth~II]{Elisabeth deux,
Reine par la grâce de Dieu du Royaume Uni,
du Canada et de ses autres Royaumes et Territoires,
Chef du Commonwealth, Défenseur de la Foi.}
```

mais seulement ‘Elisabeth II’ dans la table de matière et sur les hauts de page. Dans l’exemple suivant il y a un saut de ligne dans *titre* mais ça ne fonctionne pas pour la table des matières alors il est omis de la table des matières et des titres de hauts de page.

```
\section[La vérité c’est que j’ai menti ; histoire de ma vie]{La vérité
c’est que j’ai menti\\ histoire de ma vie}
```

Pour déterminer quelles rubriques sont numérotées et lesquelles apparaissent dans la table des matières, le numéro de niveau d’une section est 1 (voir [Sectioning/secnumdepth], page 43, et voir [Sectioning/tocdepth], page 43).

Lorsque vous chargez un paquetage d'internationalisation tel que `babel`, `mlp` ou `polyglossia`, et que la langue sélectionnée est le français, alors le premier paragraphe après le titre est normalement renforcé, comme c'est la convention en typographie française. Par contre, si vous restez en langue par défaut, c.-à-d. en anglo-américain, le paragraphe qui suit le titre de section n'est pas renforcé, étant donné que c'est pour l'anglo-américain une pratique typographique standard. Pour obtenir un renforcement dans ce cas utilisez le paquetage `indentfirst`.

En général, pour changer le comportement de la commande `\section`, il y a diverses options. L'une d'elles est la commande `\@startsection` (voir Section 6.8 [`\@startsection`], page 51). Il y a aussi un grand nombre de paquetages sur le CTAN traitant de cela, dont `titlesec`. Voir sa documentation, mais l'exemple ci-dessous donne un aperçu de ce qu'il peut faire.

```
\usepackage{titlesec} % dans le préambule
\titleformat{\section}
  {\normalfont\Large\bfseries} % format du titre
  {\makebox[1pc][r]{\thesection\hspace{1pc}}} % étiquette
  {0pt} % longueur de séparation entre l'étiquette et le titre
  {} % code crochet exécuté avant
\titlespacing*{\section}
  {-1pc}{18pt}{10pt}{10pt}
```

Cela met le numéro de section dans la marge.

6.4 `\subsection`

Synopsis, l'un parmi :

```
\subsection{titre}
\subsection*{titre}
\subsection[tdm-titre]{titre}
```

Commence une sous-section. Les classes L^AT_EX standards `article`, `book`, et `report` ont toutes cette commande.

L'exemple suivant produit une sous-section :

```
Nous allons montrer qu'il y a plus de fonction que de machines de Turing
et donc que certaines fonctions n'ont pas de machine associée.
```

```
\subsection{Cardinal} Nous allons commencer par deux paradoxes qui
mettent en scène le défi que pose à notre intuition la comparaison des
tailles d'ensembles infinis.
```

Pour les classes L^AT_EX standards `book` et `report` la sortie par défaut est du genre de '1.2.3 *titre*' (pour chapitre 1, section 2, sous-section 3), seul sur sa ligne et fer à gauche, en caractère gras et un peu plus gros (la taille de police est `\large`). La même chose vaut dans `article` à ceci près qu'il n'y a pas de chapitre dans cette classe, alors cela ressemble à '2.3 *titre*'.

La forme en `*` affiche *titre*. Mais elle n'affiche pas le numéro de sous-section, ni n'incrémente le compteur `subsection`, et ne produit aucune entrée dans la table des matières.

L'argument optionnel *tdm-titre* apparaît comme le titre de sous-section dans la table des matières (voir Section 25.1 [Table of contents etc.], page 242). S'il n'est pas présent alors *titre* est à la place. L'exemple suivant montre le texte complet dans le titre de sous-section,

```
\subsection{ $\alpha$ , $\beta$ , $\gamma$  paper}{\textit{The Origin of
  Chemical Elements} by R.A.~Alpher, H.~Bethe, and G.~Gamow}
```

mais seulement ' α, β, γ paper' dans la table des matières.

Pour déterminer quelles rubriques sont numérotées et lesquelles apparaissent dans la table des matières, le numéro de niveau d'une sous-section est 2 (voir [Sectioning/secnumdepth], page 43, et voir [Sectioning/tocdepth], page 43).

Lorsque vous chargez un paquetage d'internationalisation tel que `babel`, `mlp` ou `polyglossia`, et que la langue sélectionnée est le français, alors le premier paragraphe après le titre est normalement renforcé, comme c'est la convention en typographie française. Par contre, si vous restez en langue par défaut, c.-à-d. en anglo-américain, le paragraphe qui suit le titre de sous-section n'est pas renforcé, étant donné que c'est pour l'anglo-américain une pratique typographique standard. Pour obtenir un renforcement dans ce cas utilisez le paquetage `indentfirst`.

Il y a diverses façon de changer le comportement de la commande `\subsection`. L'une d'elles et la commande `\@startsection` (voir Section 6.8 [`\@startsection`], page 51). Il y a aussi divers paquetages sur le CTAN traitant de cela, dont `titlesec`. Voir sa documentation, mais l'exemple ci-dessous donne un aperçu de ce qu'il peut faire.

```
\usepackage{titlesec} % dans le préambule
\titleformat{\subsection}[runin]
  {\normalfont\normalsize\bfseries} % format du titre
  {\thesubsection} % étiquette
  {0.6em} % espacement entre l'étiquette et le titre
  {} % code crochet exécuté avant
```

Cela place le numéro de sous-section et *titre* sur la première ligne de texte.

6.5 `\subsubsection`, `\paragraph`, `\subparagraph`

Synopsis, l'un parmi les suivant :

```
\subsubsection{titre}
\subsubsection*{titre}
\subsubsection[titre-tdm]{titre}
```

ou l'un parmi :

```
\paragraph{titre}
\paragraph*{titre}
\paragraph[titre-tdm]{titre}
```

ou l'un parmi :

```
\subparagraph{titre}
\subparagraph*{titre}
\subparagraph[titre-tdm]{titre}
```

Débuté une rubrique de type sous-sous-section, paragraphe, ou sous-paragraphe. Les classes L^AT_EX standards `article`, `book`, et `report` disposent toutes de ces commandes, bien que leur utilisation n'est pas courante.

L'exemple suivant produit une sous-sous-section :

```
\subsubsection{Piston ring compressors: structural performance}
Provide exterior/interior wall cladding assemblies
capable of withstanding the effects of load and stresses from
consumer-grade gasoline engine piston rings.
```

La sortie produite par défaut de chacune de ces trois commande est la même pour les classes L^AT_EX standard `article`, `book`, et `report`. Pour `\subsubsection` le *titre* est seul sur sa ligne, en caractères gras avec la taille normale de police. Pour `\paragraph` le *titre* est sur la même ligne que le texte qui suit, sans renforcement, en caractères gras avec la taille normale de police. Pour `\subparagraph` le *titre* est sur la même ligne que la texte qui suit, avec un renforcement de paragraphe, en caractère gras et avec la taille normale de police (comme les documents de classe `article` n'ont pas de chapitre, leurs sous-sous-sections sont numérotées et donc cela ressemble à '1.2.3 *titre*', pour section 1, sous-section 2, et sous-sous-section 3. Les deux autres subdivisions ne sont pas numérotées).

La forme en `*` affiche *titre*. Mais elle n'incrmente pas le compteur associé et ne produit pas d'entrée dans la table des matières (et le titre produit pas `\subsubsection` n'a pas de numéro).

L'argument optionnel *titre-tdm* apparaît comme le titre de la rubrique dans la table des matières (voir Section 25.1 [Table of contents etc.], page 242). S'il est omis, alors *titre* est utilisé à la place.

Pour déterminer quelles rubriques sont numérotées et lesquelles apparaissent dans la table des matières, le numéro de niveau d'une sous-sous-section est 3, celui d'un paragraphe est 4, et celui d'un sous-paragraphe est 5 (voir [Sectioning/secnumdepth], page 43, et voir [Sectioning/tocdepth], page 43).

Lorsque vous chargez un paquetage d'internationalisation tel que `babel`, `mlp` ou `polyglossia`, et que la langue sélectionnée est le français, alors le premier paragraphe après le titre est normalement renforcé, comme c'est la convention en typographie française. Par contre, si vous restez en langue par défaut, c.-à-d. en anglo-américain, le paragraphe qui suit le titre de chapitre n'est pas renforcé, étant donné que c'est pour l'anglo-américain une pratique typographique standard. Une façon d'obtenir un renforcement dans ce cas est d'utiliser le paquetage `indentfirst`.

Il y a de nombreuses manières de changer le comportement de ces commandes. L'une est la commande `\@startsection` (voir Section 6.8 [`\@startsection`], page 51). Il y a aussi un grand nombre de paquetages sur le CTAN traitant de cela, dont `titlesec`. Voir sa documentation sur le CTAN.

6.6 `\appendix`

Synopsis :

```
\appendix
```

Ne produit pas directement quelque chose en sortie. Mais dans un document `book` ou `report` cela déclare que toute commande `\chapter` qui suit commence une annexe. Pour les documents `article` cela fait la même chose mais avec les commandes `\section`. Remet également à zéro les compteurs `chapter` et `section` dans un document `book` ou `report`, et dans un `article` les compteurs `section` et `subsection`.

Dans ce document de classe `book` :

```
\chapter{Un} ...
\chapter{Deux} ...
...
\appendix
\chapter{Trois} ...
\chapter{Quatre} ...
```

les deux premières commandes produisent en sortie ‘Chapitre 1’ and ‘Chapitre 2’. Après la commande `\appendix` la numérotation devient ‘Annexe A’ et ‘Annexe B’. Voir Section A.4 [Larger book template], page 276, pour un autre exemple.

Le paquetage `appendix` ajoute la commande `\appendixpage` pour créer une page de titre de partie intitulée ‘Appendices’ dans le corps du document avant la première annexe, ainsi que la commande `\addappheadtotoc` pour créer l’entrée correspondante dans la table des matières. On peut régler le nom ‘Appendices’ avec une commande comme `\renewcommand{\appendixname}{Annexes}`, et il y a plusieurs autres fonctions. Voir la documentation sur le CTAN.

6.7 `\frontmatter`, `\mainmatter`, `\backmatter`

Synopsis, l’un ou plus parmi :

```
\frontmatter
...
\mainmatter
...
\backmatter
...
```

Formate un document de classe `book` différemment selon la partie du document en cours de production. Les trois commandes sont toutes optionnelles.

Traditionnellement, les pièces préliminaires (`\frontmatter`) d’un livre comprennent des choses telles que la page de titre, un abrégé, une table des matières, une préface, une liste des notations, une liste des figures et une liste des tableaux. (Certaines des pages des pièces préliminaires, telles que la page de titres, traditionnellement ne sont pas numérotée). Les pièces postliminaires (`\backmatter`) peuvent contenir des choses telles qu’un glossaire, une bibliographie, et un index.

La commande `\frontmatter` rend les numéros de page en chiffres romains bas de casse, et rend les chapitres non numérotés, bien que les titres de chaque chapitre apparaissent dans la table des matières ; si vous utilisez là aussi d’autres commandes de rubricage, alors utilisez la version en `*` (voir Chapitre 6 [Sectioning], page 42).

La commande `\mainmatter` permet de revenir au comportement attendu, et réinitialise le numéro de page.

La commande `\backmatter` n’affecte pas la numérotation des pages, mais commute de nouveau les chapitres en mode sans numéros.

Voir Section A.4 [Larger book template], page 276, pour un exemple d’usage de ces trois commandes.

6.8 \@startsection, composer les rubriques.

Synopsis :

```
\@startsection{nom}{niveau}{retrait}{avant}{après}{style}
```

Utilisé pour aider à redéfinir le comportement des commandes de rubricage telles que `\section` ou `\subsection`.

Notez que le paquetage `titlesec` rend la manipulation du rubricage plus facile. De plus, bien que la plupart des exigences concernant les commandes de rubricage peuvent être remplies avec `\@startsection`, ce n'est pas le cas de certaines d'entre elles. Par exemple, dans les classes L^AT_EX standardes `book` et `report`, les commandes `\chapter` et `\report` ne sont pas construites de cette manière. Pour fabriquer une telle commande, il est possible d'utiliser la commande `\secdef`.

Techniquement, la commande `\@startsection` a la forme suivante :

```
\@startsection{nom}
  {niveau}
  {retrait}
  {avant}
  {après}
  {style}*[titretdm]{titre}
```

de sorte que faire :

```
\renewcommand{\section}{\@startsection{nom}
  {niveau}
  {retrait}
  {avant}
  {après}
  {style}}
```

redéfinit `\section` en gardant sa forme standard d'appel `\section*[titretdm]{titre}` (dans laquelle on rappelle que l'étoile `*` est optionnelle). Voir Chapitre 6 [Sectioning], page 42. Ceci implique que quand vous écrivez une commande comme `\renewcommand{\section}{...}`, le `\@startsection{...}` doit venir en dernier dans la définition. Voir les exemples ci-dessous.

nom Nom du compteur utilisé pour numéroter les titres de rubrique. Ce compteur doit être défini séparément. Ceux qui sont utilisés le plus communément sont `section`, `subsection`, ou `paragraph`. Bien que dans ces cas-là le nom du compteur soit identique à celui de la commande elle-même, utiliser le même nom n'est pas obligatoire.

Alors `\thename` affiche le numéro de titre, et `\namemark` sert aux en-têtes de page. Voir le troisième exemple plus bas.

niveau Entier donnant la profondeur de la commande de rubricage. Voir Chapitre 6 [Sectioning], page 42, pour une liste des numéros standards de niveaux.

Si *niveau* est inférieur ou égal à la valeur du compteur `secnumdepth`, alors les titres pour cette commande de rubricage sont numérotés (voir [Sectioning/secnumdepth], page 43). Par exemple : dans un `article`, si `secnumdepth` vaut 1, alors une commande `\section{Introduction}`

produira en sortie une chaîne du type « 1 Introduction », alors que `\subsection{Historique}` produira en sortie une chaîne sans numéro de préfixe : « Historique ».

Si *niveau* est inférieur ou égal à la valeur du compteur `tocdepth`, alors la table des matières aura un article pour cette rubrique. Par exemple, dans un `article`, si `tocdepth` vaut 1, la table des matières listera les `sections`, mais pas les `subsections`.

retrait

Une longueur donnant le renfoncement de toutes les lignes du titre par rapport à la marge de gauche. Pour un renfoncement nul, utilisez `0pt`. Une valeur négative telle que `-1em` cause un débord du titre dans la marge de gauche.

avant

Longueur dont la valeur absolue est la longueur de l'espace vertical inséré avant le titre de la rubrique. Cet espacement est ignoré si la rubrique commence au début d'une page. Si ce nombre est négatif, alors le premier paragraphe suivant le titre n'est pas renforcé, s'il est positif ou nul il l'est. (Notez que l'opposé de `1pt plus 2pt minus 3pt` est `-1pt plus -2pt minus -3pt`).

Par exemple si *avant* vaut `-3.5ex plus -1ex minus -0.2ex` alors pour commencer la nouvelle rubrique, `LATEX` ajoute environ 3,5 fois la hauteur d'une lettre x en espace vertical, et le premier paragraphe de la rubrique n'est pas renforcé. Utiliser une longueur élastique, c.-à-d. comprenant `plus` et `minus`, est une bonne pratique ici car cela donne à `LATEX` plus de latitude lors de la fabrication de la page (voir Chapitre 14 [Lengths], page 141).

La quantité totale d'espace vertical entre la ligne de base de la ligne précédant cette rubrique et la ligne de base du titre de la rubrique est la somme du `\parskip` dans la police du corps de texte, du `\baselineskip` de la police du titre, et de la valeur absolue de l'argument *avant*. Cet espace est typiquement élastique de sorte à pouvoir se dilater ou se contracter. (Si la rubrique commence en début d'une page de sorte que cet espace soit ignoré, alors la ligne de base du titre correspond à la ligne de base qu'aurait la première ligne de texte sur cette page si celle-ci commençait par du texte).

après

Longueur. Lorsque *après* est positif ou nul, il s'agit de l'espace vertical à insérer après le titre de la rubrique. Lorsque elle est négative, alors le titre fait corps avec le paragraphe le suivant immédiatement. Dans ce cas la valeur absolue de la longueur donne l'espace horizontal entre la fin du titre et le début du paragraphe suivant. (Notez que l'opposé de `1pt plus 2pt minus 3pt` est `-1pt plus -2pt minus -3pt`).

Comme c'est le cas avec *avant*, utiliser une longueur élastique avec des composantes `plus` et `minus` est une bonne pratique ici puisque elle donne à `LATEX` plus de latitude pour assembler la page.

Si *après* est positif ou nul, la quantité totale d'espace vertical entre la ligne de base du titre de la rubrique et la ligne de base de la première ligne du paragraphe suivant est la somme du `\parskip` de la police du titre, de `\baselineskip` de la police du corps de texte, et de la valeur de *après*. Cet espace est typiquement élastique de sorte qu'il puisse se dilater ou se contracter. (Notez que, du fait

que le signe d'après contrôle que le titre soit indépendant du texte qui le suit ou faisant corps avec lui, vous ne pouvez pas utiliser un *après* négatif pour annuler une partie du `\parskip`).

style Contrôle le style du titre : voir les exemples plus bas. Les commandes typiquement utilisées ici sont `\centering`, `\raggedright`, `\normalfont`, `\hrule`, ou `\newpage`. La dernière commande au sein de *style* peut être une commande prenant un argument, telle que `\MakeUppercase` ou `\fbox`. Le titre de la rubrique est passé en argument à cette commande. Par exemple régler *style* à `\bfseries\MakeUppercase` a pour effet de produire des titres gras et en capitales.

Voici les réglages par défaut de L^AT_EX pour les trois premiers niveaux de rubricage qui sont définis par `\@startsection`, pour les classes `article`, `book`, et `report`.

- Pour *section* : le *niveau* vaut 1, le *retrait* vaut 0pt, le *avant* vaut `-3.5ex plus -1ex minus -0.2ex`, le *après* vaut `2.3ex plus 0.2ex`, et le *style* vaut `\normalfont\Large\bfseries`.
- Pour *subsection* : le *niveau* vaut 2, le *retrait* vaut 0pt, le *avant* vaut `-3.25ex plus -1ex minus -0.2ex`, le *après* vaut `1.5ex plus 0.2ex`, et le *style* vaut `\normalfont\large\bfseries`.
- Pour *subsubsection* : le *niveau* vaut 3, le *retrait* vaut 0pt, le *avant* vaut `-3.25ex plus -1ex minus -0.2ex`, le *après* vaut `1.5ex plus 0.2ex`, et le *style* vaut `\normalfont\normalsize\bfseries`.

Quelques exemples suivent. Ils vont soit au sein d'un fichier de paquetage ou de classe, soit dans le préambule d'un document L^AT_EX. Si vous les mettez dans le préambule, elle doivent être entre une commande `\makeatletter` et une commande `\makeatother`. (Le message d'erreur `You can't use '\spacefactor' in vertical mode.` est le plus probable lorsque on oublie de faire cela). Voir Section 12.3 [`\makeatletter & \makeatother`], page 127.

L'exemple ci-dessous centre les titres de section et les met en gros caractères gras. Il le fait avec `\renewcommand` parce que les classes standards de L^AT_EX ont déjà une commande `\section` de définie. Pour la même raison il ne définit ni un compteur `section`, ni les commandes `\thesection` et `\l@section`.

```
\renewcommand\section{%
  \@startsection{section}% [nom], page 51.
  {1}% [niveau], page 51.
  {0pt}% [retrait], page 52.
  {-3.5ex plus -1ex minus -.2ex}% [avant], page 52.
  {2.3ex plus.2ex}% [après], page 52.
  {\centering\normalfont\Large\bfseries}}% [style], page 53.
```

L'exemple ci-dessous met les titres de *subsection* en petites capitales, et leur fait faire corps avec le paragraphe suivant.

```
\renewcommand\subsection{%
  \@startsection{subsection}% [nom], page 51.
  {2}% [niveau], page 51.
  {0em}% [retrait], page 52.
  {-1ex plus 0.1ex minus -0.05ex}% [avant], page 52.
```

```

    {-1em plus 0.2em}% [après], page 52.
    {\scshape}% [style], page 53.
}

```

Les exemples précédents redéfinissaient les commandes de titre de rubriques existantes. L'exemple suivant définit une nouvelle commande, illustrant la nécessité d'un compteur et de macros pour son affichage.

```

\setcounter{secnumdepth}{6}% affiche les compteurs jusqu'à ce niveau
\newcounter{subsubparagraph}[subparagraph]% compteur pour la
                                           % numérotation
\renewcommand{\thesubsubparagraph}%
    {\thesubparagraph.\@arabic\c@subsubparagraph}% comment afficher
                                           % la numérotation
\newcommand{\subsubparagraph}{\@startsection
    {subsubparagraph}%
    {6}%
    {0em}%
    {\baselineskip}%
    {0.5\baselineskip}%
    {\normalfont\normalsize}}
\newcommand*\l@subsubparagraph%
    {\@dottedtocline{6}{10em}{5em}}% pour la table des matières
\newcommand{\subsubparagraphmark}[1]{}% pour les en-têtes de page

```

7 Des renvois

Une des raisons pour numéroter des choses telles que les figures ou les équations est d'indiquer au lecteur une référence vers elles, comme dans « Voir la figure 3 pour plus de détails. »

Souvent on désire écrire quelque chose du genre de ‘Voir théorème~31’. Mais Inclure manuellement le numéro est une mauvaise pratique. Au lieu de cela, il vaut mieux écrire une *étiquette* du genre `\label{eq:ThmGreens}` puis lui *faire référence* avec `Voir l'équation~\ref{eq:ThmGreens}`. L^AT_EX se charge de déterminer automatiquement le numéro, de le produire en sortie, et de le changer par la suite si besoin est.

```
Cela apparaîtra avec le théorème~\ref{th:ThmGreens}. % référence déclarée en aval
...
\begin{theorem} \label{th:ThmGreens}
...
\end{theorem}
...
Voir le théorème~\ref{th:ThmGreens} page~\pageref{th:ThmGreens}.
```

L^AT_EX garde trace de l'information de renvoi dans un fichier avec le même nom de base que le fichier contenant le `\label{...}` mais avec une extension `.aux`. Ainsi si `\label` est dans `calcul.tex` alors cette information est dans `calcul.aux`. L^AT_EX met cette information dans ce fichier à chaque fois qu'il rencontre un `\label`.

L'effet secondaire le plus courant du paragraphe précédent se produit lorsque votre document a une *référence déclarée en aval*, c.-à-d. un `\ref` qui apparaît avant le `\label` associé. Si c'est la première fois que vous compilez le document alors vous obtiendrez un message `LaTeX Warning: Label(s) may have changed. Rerun to get cross references right.` et dans la sortie la référence apparaîtra comme deux points d'interrogation ‘??’ en caractères gras. Ou, si vous modifiez le document de telle façon que les références changent alors vous obtiendrez le même avertissement et la sortie contiendra l'information de référence de la fois précédente. La solution dans les deux cas est juste de recompiler le document encore une fois.

Le paquetage `cleveref` élargit les possibilités de faire des renvois de L^AT_EX. Vous pouvez faire en sorte que si vous saisissez `\begin{thm}\label{th:Nerode}...\end{thm}` alors `\cref{th:Nerode}` produit en sortie ‘théorème 3.21’, sans que vous ayez à saisir le mot « théorème ».

7.1 \label

Synopsis :

```
\label{clef}
```

Attribut un numéro de référence à *clef*. Au sein de texte ordinaire, `\label{clef}` attribut à *clef* le numéro de la rubrique courante. Au sein d'un environnement numéroté, tel que l'environnement `table` ou `theorem`, `\label{clef}` attribue le numéro de cet environnement à *clef*. On retire le numéro attribué avec la commande `\ref{clef}` (voir Section 7.3 [`\ref`], page 57).

Le nom *clef* peut se composer de n'importe quelle séquence de lettres, chiffres, ou caractères de ponctuation ordinaires. Il est sensible à la casse — lettres capitales ou bas-de-casse.

Pour éviter de créer accidentellement deux étiquettes avec le même nom, l'usage est d'utiliser des étiquettes composées d'un préfixe et d'un suffixe séparés par un caractère : ou .. Certains préfixes classiquement utilisés :

<code>ch</code>	pour les chapitres
<code>sec</code>	les commandes de rubricage de niveau inférieur
<code>fig</code>	pour les figures
<code>tab</code>	pour les tableaux
<code>eq</code>	pour les équations

Ainsi, `\label{fig:Euler}` est une étiquette pour une figure avec un portrait de ce grand homme.

Dans l'exemple ci-dessous la clef `sec:test` se verra attribuée le numéro de la section courante et la clef `fig:test` se verra attribuée le numéro de la figure. Soit dit en passant, mettez les étiquettes (`\label`) après les légendes (`\caption`) au sein des environnements `figure` ou `table`.

```
\section{Titre de la rubrique}
\label{sec:test}
Dans cette rubrique~\ref{sec:test}.
\begin{figure}
...
\caption{Texte de la légende}
\label{fig:test}
\end{figure}
Voir Figure~\ref{fig:test}.
```

7.2 `\pageref{clef}`

Synopsis :

```
\pageref{clef}
```

Produit le numéro de page de l'endroit du texte où la commande correspondante `\label{clef}` apparaît.

Dans cet exemple le `\label{eq:principale}` est utilisé à la fois pour le numéro de la formule et pour le numéro de page. (Notez que les deux références sont des références déclarées en aval, ainsi ce document a besoin d'être compilé deux fois pour les résoudre).

```
Le résultat principal est la formule~\ref{eq:principale} de la
page~\pageref{eq:principale}.
```

```
...
\begin{equation} \label{eq:principale}
\mathbf{P}=\mathbf{NP}
\end{equation}
```

7.3 `\ref{clef}`

Synopsis :

```
\ref{clef}
```

Produit le numéro de la rubrique, équation, note en bas de page, figure, . . . , de la commande correspondante `\label` (voir Section 7.1 [`\label`], page 55). Elle ne produit aucun texte, tel que le mot ‘Section’ ou ‘Figure’, juste le numéro lui-même sans plus.

Dans cet exemple, le `\ref{populaire}` produit ‘2’. Notez que ceci est une référence déclarée en aval puisque elle est faite avant `\label{populaire}`.

```
Le format utilisé le plus largement est à l'article numéro~\ref{populaire}.■
\begin{enumerate}
\item Plain \TeX
\item \label{populaire} \LaTeX
\item Con\TeX t
\end{enumerate}
```

7.4 Le paquetage `xr`

Synopsis :

```
\usepackage{xr}
\externaldocument{nom-de-base-document}
```

ou

```
\usepackage{xr}
\externaldocument[préfixe-renvoi]{nom-de-base-document}
```

Fait des renvois vers le document externe `nom-de-base-document.tex`.

Voici un exemple. Si `cours.tex` comprend ce qui suit dans le préambule :

```
\usepackage{xr}
\externaldocument{exercices}
\externaldocument[IND-]{indications}
\externaldocument{reponses}
```

alors on peut utiliser des étiquettes de renvoi depuis les trois autres documents. Supposons que `exercices.tex` a une liste énumérée qui contient ceci :

```
\item \label{exer:ThmEuler} Que se passe-t-il si chaque nœud a un
degré impair ?
```

et que `indications.tex` comprend une liste énumérée avec ceci :

```
\item \label{exer:ThmEuler} Distinguez le cas à deux nœuds.
```

et que `reponses.tex` a une liste énumérée avec ceci :

```
\item \label{rep:ThmEuler} Il n'y pas pas de chemin d'Euler, sauf
s'il y a exactement deux nœuds.
```

Après avoir compilé les documents `exercices`, `indications`, et `reponses`, saisir ce qui suit dans le corps de `cours.tex` aura pour effet qu'on a accès depuis `cours` aux numéros de renvoi utilisés dans les autres documents.

```
Voir Exercice~\ref{exer:ThmEuler}, avec Indication~\ref{IND-exer:ThmEuler}.■
La solution est Réponse~\ref{rep:ThmEuler}.
```

Le préfixe `IND-` pour les renvois depuis le fichier `indications` est nécessaire parce que l'étiquette dans ce fichier est la même que dans le fichier `exercices`. Sans ce préfixe, les deux renvois auraient le numéro correspondant à ce dernier.

Note : si le document utilise le paquetage `hyperref` alors au lieu de `xr`, placez `\usepackage{xr-hyper}` avant le `\usepackage{hyperref}`. Aussi, si l'un quelconque parmi les documents utilise `hyperref` alors tous doivent l'utiliser.

8 Environnements

L^AT_EX fournit beaucoup d’environnements pour baliser un certain texte. Chaque environnement commence et se termine de la même manière :

```
\begin{nomenv}
...
\end{nomenv}
```

8.1 abstract

Synopsis :

```
\begin{abstract}
...
\end{abstract}
```

Produit un résumé, potentiellement contenant plusieurs paragraphes. Cet environnement n’est défini que dans les classes de document `article` et `report` (voir Chapitre 3 [Document classes], page 7).

Utiliser l’exemple ci-dessous au sein de la classe `article` produit un paragraphe détaché. L’option `titlepage` de la classe de document a pour effet que le résumé soit sur une page séparée (voir Section 3.1 [Document class options], page 7) ; ceci est le comportement par défaut seulement dans la classe `report`.

```
\begin{abstract}
  Nous comparons tous les récits de la proposition faits par Porter
  Alexander à Robert E Lee en lieu de l’Appomattox Court House que
  l’armée continue à combattre dans une guerre de guerilla, ce que Lee
  refusa.
\end{abstract}
```

L’exemple suivant produit un résumé en une-colonne au sein d’un document en deux-colonnes (pour plus solution plus flexible, utilisez le paquetage `abstract`).

```
\documentclass[twocolumn]{article}
...
\begin{document}
\title{Babe Ruth comme ancêtre culturel : une approche atavique}
\author{Smith \ \ Jones \ \ Robinson\thanks{Bourse des chemins de fer.}}
\twocolumn[
  \begin{@twocolumnfalse}
    \maketitle
    \begin{abstract}
      Ruth n’était pas seulement le Sultan du Swat, il était à lui tout
      seul l’équipe du swat.
    \end{abstract}
  \end{@twocolumnfalse}
]
{ % by-hand insert a footnote at page bottom
  \renewcommand{\thefootnote}{\fnsymbol{footnote}}
```

```
\footnotetext[1]{Merci pour tout le poisson.}
}
```

8.2 array

Synopsis :

```
\begin{array}{patron}
entrée-col-1&entrée-col-2 ... &entrée-col-n}\\
...
\end{array}
```

ou

```
\begin{array}[pos]{patron}
entrée-col-1&entrée-col-2 ... &entrée-col-n}\\
...
\end{array}
```

Les tableaux mathématiques sont produits avec l’environnement `array`. Cet environnement ne peut être utilisé qu’en mode `math` (voir Chapitre 17 [Modes], page 185), normalement au sein d’un environnement mathématique hors texte tel qu’`equation` (voir Section 8.9 [equation], page 66). Les entrées dans chaque colonne sont séparées avec une esperluette (&). Les lignes sont terminées par une double controbligue (voir Section 9.1 [\\], page 111).

L’exemple suivant affiche un tableau trois par trois.

```
\begin{equation*}
\chi(x) =
\left|                                     % barre verticale en bordure
\begin{array}{ccc}
x-a & & -b & & -c \\
-d & & x-e & & -f \\
-g & & & & x-i
\end{array}
\right|
\end{equation*}
```

L’argument obligatoire *patron* décrit le nombre de colonnes, l’alignement en leur sein, et le formatage des régions inter-colonne. Par exemple, `\begin{array}{rcl}... \end{array}` produit trois colonnes : la première fer à droite, la deuxième centrée, et la troisième fer à gauche. Voir Section 8.23 [tabular], page 97, pour une description complète de *patron*, et des autres caractéristiques communes aux deux environnements, y compris l’argument optionnel *pos*.

L’environnement `array` diverge de `tabular` par deux aspects. Le premier est que les entrées de `array` sont composées en mode mathématique, en style texte (voir Section 16.7 [Math styles], page 181) (sauf si le *patron* spécifie la colonne avec `p{...}`, ce qui a pour effet que l’entrée est composée en mode texte). Le second est que au lieu du paramètre `\tabcolsep` de `tabular`, l’espace inter-colonne que L^AT_EX met dans un `array` est contrôlé par `\arraycolsep`, qui spécifie la moitié de la largeur entre les colonnes. La valeur par défaut est ‘5pt’ de sorte qu’un espace de 10 pt sépare deux colonnes.

Pour obtenir des tableaux entre accolades la méthode standard est d'utiliser le paquetage `amsmath`. Il comprend les environnements `pmatrix` pour un tableau entre parenthèses (...), `bmatrix` pour un tableau entre crochets [...], `Bmatrix` pour un tableau entre accolades {...}, `vmatrix` pour un tableau entre barres verticales |...|, et `Vmatrix` pour un tableau entre doubles barres verticales ||...||, ainsi que diverses autres constructions de tableaux.

L'exemple suivant utilise le paquetage `amsmath` :

```
\usepackage{amsmath} % dans le préambule

\begin{equation}
  \begin{vmatrix}{cc}
    a & b \\
    c & d
  \end{vmatrix}=ad-bc
\end{equation}
```

Il y a beaucoup de paquetages concernant les tableaux. Le paquetage `array` étend utilement leur possibilités de bien des manières, et notamment en ajoutant des types de colonne. Le paquetage `dcolum` ajoute un type de colonne pour centrer sur le séparateur décimal. Pour les deux voir la documentation sur le CTAN.

8.3 center

Synopsis :

```
\begin{center}
  ligne1 \\
  ligne2 \\
\end{center}
```

L'environnement `center` vous permet de créer un paragraphe consistant de lignes qui sont centrées entre les marges de gauche et de droite de la page courante. On utilise une double controblique, `\\`, pour obtenir un saut de ligne (voir Section 9.1 [`\\`], page 111). Si du texte est trop long pour entrer dans une ligne, alors \LaTeX insère des sauts de ligne en évitant de faire des césures ou de dilater ou contracter tout espace inter-mot.

Cet environnement insère de l'espace au-dessus et au-dessous du corps du texte. Voir Section 8.3.1 [`\centering`], page 62, pour ne pas avoir cet espace, par exemple au sein d'un environnement `figure`.

L'exemple suivant produit trois lignes centrées. Il y a un espace vertical supplémentaire entre les deux dernières lignes.

```
\begin{center}
  Une thèse soumise en remplissant partiellement \\
  les exigences de \[0.5ex]
  l'École pour l'Ingénierie Environnementale
\end{center}
```

Dans l'exemple suivant, selon la largeur de la ligne de la page, \LaTeX pourrait faire un saut de ligne pour la partie avant la double controblique. Si cela se produit, il en centre

chaque ligne, et sinon il en centre l'unique ligne. Ensuite \LaTeX faut un saut de ligne à la double controblique, et centre la partie finale.

```
\begin{center}
  Mon père considérait comme intolérable quiconque fréquentait l'église mais ne buvait
  J'ai grandi dans cette croyance. --- Richard Burton
\end{center}
```

Ajouter une double controblique à la fin de la ligne finale est optionnel. Lorsque elle est présente, cela n'ajoute pas d'espace vertical.

Dans un document en double-colonne le texte est centré dans une colonne, et non sur la page entière.

8.3.1 `\centering`

Synopsis :

```
{\centering ... }
```

ou

```
\begin{group}
  \centering ...
\end{group}
```

Centre la matière dans sa portée. Utilisée le plus souvent l'intérieur d'un environnement tel que `figure` ou dans une `parbox`.

L'exemple suivant de déclaration `\centerin` a pour effet de centrer le graphique horizontalement.

```
\begin{figure}
  \centering
  \includegraphics[width=0.6\textwidth]{ctan_lion.png}
  \caption{CTAN Lion} \label{fig:CTANLion}
\end{figure}
```

La portée de ce `\centering` finit avec le `\end{figure}`.

Contrairement à l'environnement `center`, la commande `\centering` n'ajoute pas d'espacement vertical au-dessus et au-dessous du texte. C'est son avantage dans l'exemple précédent ; il n'y a pas d'espace en trop.

Elle ne commence pas non plus un nouveau paragraphe ; elle change simplement la façon dont \LaTeX formate les unités paragraphe. Si `ww {\centering xx \\\ yy} zz` est entouré de lignes à blanc, alors \LaTeX crée un paragraphe dont la première ligne `ww xx` est centrée, et dont la seconde ligne, non centrée, contient `yy zz`. Généralement, ce qu'on désire c'est que la portée de la déclaration contienne une ligne à blanc ou la commande `\end` d'un environnement tel que `figure` ou `table` qui finisse l'unité paragraphe. Ainsi, si `{\centering xx \\\ yy\par} zz` est entouré de lignes à blanc alors cela fabrique un nouveau paragraphe avec deux lignes centrées 'xx' et 'yy', suivi d'un nouveau paragraphe 'zz' qui est formaté comme d'habitude.

8.4 description

Synopsis :

```
\begin{description}
  \item [étiquette du 1er article] texte du 1er article
  \item [étiquette du 2e article] texte du 2e article
  ...
\end{description}
```

L'environnement `description` est utilisé pour fabriquer des listes d'articles étiquetés. Chaque *étiquette* d'article est composée en gras, alignée à gauche de sorte que les étiquettes longues continuent sur la première ligne du texte de l'article. Il doit y avoir au moins un article ; sans cela on provoque l'erreur L^AT_EX 'Something's wrong--perhaps a missing `\item`'.

Cet exemple montre l'environnement utilisé pour une séquence de définitions.

```
\begin{definition}
  \item[lama] Un prêtre.
  \item[lame] Une pièce coupante.
```

Les étiquettes 'lama' et 'llama' ressortent en gras avec leur bords gauches alignés sur la marge de gauche.

Faites démarrer la liste d'articles avec la commande `\item` (voir Section 8.16.1 [`\item`], page 77). Utilisez l'étiquette optionnelle, comme dans `\item[Point principal]`, en effet il n'y a pas de valeur par défaut sensée. Après le `\item` se trouve du texte optionnel pouvant contenir plusieurs paragraphes.

Comme les étiquettes sont en gras, si le texte de l'étiquette appelle un changement de police effectué dans la forme à argument (voir Section 4.2 [Font styles], page 23) alors il ressortira en gras. Par exemple, si le texte de l'étiquette est en police machine à écrire comme dans `\item[\texttt{texte étiquette}]` alors il apparaîtra en tapuscrit gras, si cela est disponible. La méthode la plus simple pour obtenir la police tapuscrit non grasse est d'utiliser la forme déclarative : `\item[{\tt texte étiquette}]`. De la même façon, obtenez la police romaine standard avec `\item[{\rm texte étiquette}]`.

En ce qui concerne les autres principaux environnements de liste à étiquettes de L^AT_EX, voir Section 8.14 [`itemize`], page 71, et Section 8.7 [`enumerate`], page 65. Contrairement à ces environnements, imbriquer les environnements `description` ne change pas l'étiquette par défaut ; elle est en gras et alignée à gauche à tous les niveaux.

Pour plus d'information sur les paramètres de disposition de liste, y compris les valeurs par défaut, et sur la personnalisation de la disposition de liste, voir Section 8.16 [`list`], page 72. Le paquetage `enumitem` est utile pour personnaliser les listes.

Cet exemple met les étiquettes de description en petites capitales.

```
\renewcommand{\descriptionlabel}[1]{%
  {\hspace{\labelsep}\textsc{#1}}}
```

8.5 displaymath

Synopsis :

```
\begin{displaymath}
```

```

des maths
\end{displaymath}

```

L'environnement `displaymath` compose le texte *des maths* sur sa propre ligne, centré par défaut. L'option globale `fleqn` justifie les équations à gauche ; voir Section 3.1 [Document class options], page 7.

Aucun numéro d'équation n'est ajouté au texte de `displaymath` ; pour obtenir un numéro d'équation, vous pouvez utiliser l'environnement `equation` (voir Section 8.9 [equation], page 66).

L^AT_EX ne fait pas de saut de ligne au sein de *des maths*.

Notez que le paquetage `amsmath` comprend des possibilités beaucoup plus vastes en matière d'affichage d'équations. Par exemple, il offre plusieurs alternatives pour effectuer des sauts de lignes au sein de texte en mode mathématique.

La construction `\[des maths\]` est un synonyme de l'environnement `\begin{displaymath}des maths\end{displaymath}`, mais ce dernier est plus pratique à manipuler dans le fichier source ; par exemple la recherche d'un caractère crochet] peut donner des faux positifs, alors qu'il est plus probable que le mot `displaymath` soit unique.

(Digression : la construction `$$des maths$$` tirée du langage T_EX de base est souvent utilisée à tort comme un synonyme de `displaymath`. Elle n'en est pas un, et n'est pas du tout officiellement prise en charge par L^AT_EX ; `$$` ne prend pas en charge `fleqn` (voir Section 3.1 [Document class options], page 7), gère l'espacement vertical différemment, et n'effectue pas de vérification de cohérence).

Le texte composé par cet exemple est centré et seul sur sa ligne.

```

\begin{displaymath}
\int_1^2 x^2 \, dx = 7/3
\end{displaymath}

```

De plus, le signe intégrale est plus grand que ce que la version en ligne `\(\int_1^2 x^2 \, dx = 7/3 \)` produit.

8.6 document

L'environnement `document` entoure le corps entier d'un document. Il est obligatoire dans tout document L^AT_EX. Voir Section 2.1 [Starting and ending], page 2.

Synopsis :

```
\AtBeginDocument{code}
```

Sauvegarde *code* et exécute le quand `\begin{document}` est exécuté, à la toute fin du préambule. Le code est exécuté après que les tables de sélection de police ont été réglées, ainsi la police normale du document est la police courante. Toutefois, le code est exécuté en tant que faisant partie du préambule, c'est pourquoi on ne peut pas composer du texte avec.

On peut utiliser cette commande plus d'une fois ; les lignes de code successives sont exécutées dans l'ordre de passage à la commande.

Synopsis :

```
\AtEndDocument{code}
```

Sauvegarde *code* et l'exécute vers la fin du document. Plus précisément, il est exécuté lorsque `\end{document}` est exécuté, avant que la dernière page ne soit terminée et avant que tous environnements flottant restants soient traités. Si on désire d'une partie du code soit exécuté après ces deux traitements, alors il suffit d'inclure un `\clearpage` à l'endroit approprié du *code*.

On peut utiliser cette commande plus d'une fois ; les lignes de code successives sont exécutée dans l'ordre de passage à la commande.

8.7 enumerate

Synopsis :

```
\begin{enumerate}
\item article1
\item article2
...
\end{enumerate}
```

L'environnement `enumerate` produit une liste numérotée d'articles. Le format du numéro en étiquette dépend de si cet environnement est imbriqué dans un autre ; voir plus bas.

La liste consiste en au moins un article. L'absence d'article cause l'erreur `LATEX 'Something's wrong--perhaps a missing \item'`. Chaque article est produit avec la commande `\item`.

Cet exemple fait la liste des deux premiers coureurs à l'arrivée du marathon olympique de 1908 :

```
\begin{enumerate}
\item Johnny Hayes (USA)
\item Charles Hefferon (RSA)
\end{enumerate}
```

Les énumérations peuvent être imbriquées les unes dans les autres, jusqu'à une profondeur de quatre niveaux. Elles peuvent aussi être imbriquées au sein d'autres environnements fabriquant des paragraphes, tels que `itemize` (voir Section 8.14 [itemize], page 71) et `description` (voir Section 8.4 [description], page 63). Le format de l'étiquette produite dépend du niveau d'imbrication de la liste. Voici les valeurs par défaut de `LATEX` pour le format à chaque niveau d'imbrication (où 1 est le niveau le plus externe) :

1. numéro arabe suivi d'un point : '1.', '2.', ...
2. lettre en bas de casse et entre parenthèse : '(a)', '(b)' ...
3. numéro romain en bas de casse suivi d'un point : 'i.', 'ii.', ...
4. lettre capitale suivie d'un point : 'A.', 'B.', ...

L'environnement `enumerate` utilise les compteurs `\enumi`, ..., `\enumiv` (voir Chapitre 13 [Counters], page 138). Si vous utilisez l'argument optionnel d'`\item` alors le compteur n'est pas incrémenté pour cet article (voir Section 8.16.1 [`\item`], page 77).

L'environnement `enumerate` utilise les commandes de `\labelenumi` jusqu'à `\labelenumiv` pour produire l'étiquette par défaut. Ainsi, vous pouvez utiliser `\renewcommand` pour changer le format des étiquettes (voir Section 12.1 [`\newcommand` &

`\renewcommand`], page 124). Par exemple, cette liste de premier niveau va être étiquetée avec des lettres capitales, en gras, non suivies point :

```
\renewcommand{\labelenumi}{\textbf{\Alph{enumi}}}
\begin{enumerate}
  \item eI
  \item bi:
  \item si:
\end{enumerate}
```

Pour une liste des commandes comme `\Alph` utilisables pour formater le compteur d'étiquette voir Section 13.1 [`\alph \Alph \arabic \roman \Roman \fnsymbol`], page 138.

Pour plus ample information sur la personnalisation de la forme voir Section 8.16 [`list`], page 72. De même, le paquetage `enumitem` est utile pour cela.

8.8 eqnarray

L'environnement `eqnarray` est obsolète. Il a des maladdresses parmi lesquelles l'espacement qui est incohérent avec d'autres éléments mathématiques. (Voir l'article « Évitez `eqnarray` ! » de Lars Madsen (<http://tug.org/TUGboat/tb33-1/tb103madsen.pdf>). Les nouveaux documents devraient inclure le paquetage `amsmath` et utiliser les environnements d'affichage mathématique que celui-ci fournit, tels que `align`. On inclut une description uniquement pour être complet et pour pouvoir travailler avec d'anciens documents.

Synopsis :

```
\begin{eqnarray} (ou eqnarray*)
  formula1 \\
  formula2 \\
  ...
\end{eqnarray}
```

L'environnement `eqnarray` est utilisé pour afficher une séquence d'équations ou d'inégalités. Il est similaire à un environnement `array` à trois colonnes, avec des lignes consécutives séparées par `\\` et des articles consécutifs au sein d'une ligne séparé par une esperluette `&`.

`*` peut aussi être utilisé pour séparer les équations, avec sa signification normale de ne pas autoriser un saut de page à cette ligne.

Un numéro d'équation est placé sur chaque ligne à moins que cette ligne ait une commande `\nonumber`. Alternativement, la forme étoilé (en `*`) de l'environnement (`\begin{eqnarray*} ... \end{eqnarray*}`) omet la numérotation des équations entièrement, tout en faisant par ailleurs la même chose qu'`eqnarray`.

La commande `\lefteqn` est utilisée pour couper les longues formules sur plusieurs lignes. Elle compose son argument en hors texte et le justifie à gauche dans une boîte de largeur nulle.

8.9 equation

Synopsis :

```
\begin{equation}
```

```

texte mathématique
\end{equation}

```

Même chose que l'environnement `displaymath` (voir Section 8.5 [`displaymath`], page 63) à ceci près que \LaTeX place un numéro d'équation aligné sur la marge de droite. Le numéro d'équation est généré en utilisant le compteur `equation`.

Il ne faut aucune ligne vide entre `\begin{equation}` et `\begin{equation}`, sinon \LaTeX dit qu'il manque une signe dollar.

Le paquetage `amsmath` comprend des moyens étendus pour l'affichage d'équations. Les nouveaux documents devraient inclure ce paquetage.

8.10 figure

Synopsis :

```

\begin{figure}[placement]
  corpsfigure
  \caption[titreldf]{texte} % optionnel
  \label{étiquette}          % optionnel
\end{figure}

```

ou :

```

\begin{figure*}[placement]
  corpsfigure
  \caption[titreldf]{texte} % optionnel
  \label{étiquette}          % optionnel
\end{figure*}

```

Les figures sont de la matière qui ne fait pas partie du texte normal. Un exemple est de la matière qu'on ne peut pas avoir segmenté entre deux pages, comme un graphique. À cause de cela, \LaTeX ne compose pas les figures en séquence avec le texte normal, mais au lieu de cela les fait « flotter » jusqu'à un endroit convenable, tel que le haut de la page suivante (voir Section 5.7 [Floats], page 37).

Le *corpsfigure* peut consister de graphiques importés (voir Chapitre 22 [Graphics], page 215), de texte, de commandes \LaTeX , etc. Il est composé une `\parbox` de largeur `\textwidth`.

Pour les valeurs possibles de *placement* sont **h** pour « ici » (`'here'` en anglais), **t** pour « en haut » (`'top'`), **b** pour « en bas », et **p** pour sur un page séparée de flottants. Pour l'effet de ces options sur l'algorithme de placement des flottants, voir Section 5.7 [Floats], page 37.

La version étoilée `figure*` est utilisée quand un document est en mode double-colonne (voir Section 5.2 [`\twocolumn`], page 30). elle produit une figure qui s'étend sur les deux colonnes, au sommet de la page. Pour ajouter la possibilité de la placer à un bas de page voir la discussion de *placement* **b** dans Section 5.7 [Floats], page 37.

L'étiquette est optionnelle ; elle est utilisée pour les renvois (voir Chapitre 7 [Cross references], page 55). La commande optionnelle `\caption` spécifie la légende *texte* pour la figure (voir Section 5.7.1 [`\caption`], page 40). La légende est numérotée par défaut. Si *titreldf* est présent, il est utilisé dans la liste des figures au lieu de *texte* (voir Section 25.1 [Table of contents etc.], page 242).

Cet exemple fabrique une figure à partir d'un graphique. Il nécessite l'un des paquets `graphics` ou `graphicx`. Le graphique, avec sa légende, est placé au sommet d'une page ou, s'il est rejeté à la fin du document, sur une page de flottants.

```
\begin{figure}[t]
  \centering
  \includegraphics[width=0.5\textwidth]{CTANlion.png}
  \caption{The CTAN lion, by Duane Bibby}
\end{figure}
```

8.11 filecontents: Écrire un fichier externe

Synopsis :

```
\begin{filecontents}[option]{nomfichier}
  texte
\end{filecontents}
```

ou

```
\begin{filecontents*}[option]{nomfichier}
  texte
\end{filecontents*}
```

Crée un fichier nommé *nomfichier* dans le répertoire courant (ou dans le répertoire de sortie s'il a été spécifié ; voir [output directory], page 268) et y écrit *texte*. Par défaut un fichier déjà existant n'est pas écrasé.

La version non étoilée de l'environnement `filecontent` préfixe le contenu du fichier créé d'une en-tête de commentaires `TEX` ; voir l'exemple ci-dessous. La version étoilée `filecontent*` n'inclut par l'en-tête.

Les options possibles sont :

force

overwrite

Écrase le fichier s'il existe.

noheader Omet l'en-tête. Équivalent à utiliser `filecontents*`.

nosearch Vérifie uniquement si un fichier existe dans le répertoire courant (et le répertoire de sortie, s'il a été spécifié), non dans le chemin complet de recherche.

Ces options ont été ajoutées à la parution 2019 de L^AT_EX.

Cet environnement peut être utilisé n'importe où dans le préambule, bien qu'il apparaisse souvent avant la commande `\documentclass`. Elle est classiquement utilisée pour créer un fichier `.bib` ou d'autre fichier de données similaire à partir du document source principale, de sorte à rendre le fichier source autonome. De même, il peut être utilisé pour créer un fichier personnalisé de style ou de classe, rendant ainsi encore le source autonome.

Par exemple, ce document :

```
\documentclass{article}
\begin{filecontents}{JH.sty}
\newcommand{\monnom}{Jim Hef{}feron}
\end{filecontents}
```

```

\usepackage{JH}
\begin{document}
Un article de \monnom.
\end{document}

```

produit ce fichier JH.sty :

```

%% LaTeX2e file 'JH.sty'
%% generated by the 'filecontents' environment
%% from source 'test' on 2015/10/12.
%%
\newcommand{\monnom}{Jim Hef{}feron}

```

8.12 flushleft

Synopsis :

```

\begin{flushleft}
ligne1 \\
ligne2 \\
...
\end{flushleft}

```

L'environnement `flushleft` vous permet de créer un paragraphe ferré à gauche, c'est à dire consistant en lignes qui sont alignées sur la marge de gauche et en dentelures à droite comme un drapeau dont la hampe (fer du composeur) serait à gauche et les franges à droite. Si vous avez des lignes qui sont trop longues alors L^AT_EX insère des sauts de ligne sans faire de césure ni dilater ou contracter les espaces inter-mot. Pour forcer un saut de ligne on utilise une double controblique, `\\`. Pour la forme déclarative, voir Section 8.12.1 [`\raggedright`], page 69.

L'exemple suivant crée une boîte de texte qui fait au plus 7,5cm de large, et dont le texte est fer à gauche et drapeau à droite.

```

\noindent\begin{minipage}{7.5cm}
\begin{flushleft}
  Une longue phrase que \LaTeX{} coupe à l'endroit approprié. \\
  Et, une nouvelle ligne forcée par la double controblique.
\end{flushleft}
\end{minipage}

```

8.12.1 \raggedright

Synopsis :

```

{\raggedright ... }

```

ou :

```

\begin{environnement} \raggedright
...
\end{environnement}

```

Une déclaration qui a pour effet que les lignes sont ferrées sur la marge de gauche et drapeau sur la droite, c.-à-d. que les mots sont alignés sur la gauche comme sur le fer du composeur, et en dentelure comme les franges d'un drapeau à droite. Elle peut être

utilisée au sein d'un *environnement* tel que `quote` ou d'une `parbox`. Pour la forme par environnement voir Section 8.12 [`flushleft`], page 69.

Contrairement à l'environnement `flushleft`, la commande `\raggedright` ne démarre pas un nouveau paragraphe ; elle change seulement la façon dont \LaTeX formate les unités de paragraphe. Pour affecter le format d'une unité de paragraphe, la portée de la déclaration doit contenir la ligne à blanc ou la commande `\end` qui termine l'unité de paragraphe.

Dans l'exemple ci-après `\raggedright` dans la deuxième colonne empêche \LaTeX de faire une composition très maladroite pour faire rentrer le texte dans une colonne étroite. Notez que `\raggedright` est au sein d'accolades `{...}` pour borner son effet.

```
\begin{tabular}{rp{5cm}}
  Équipe alpha & {\raggedright Cette équipe fait le travail réel.} \\
  Équipe bêta  & {\raggedright Cette équipe assure que le système de
                    refroidissement ne manque jamais d'eau.} \\
\end{tabular}
```

8.13 flushright

```
\begin{flushright}
  ligne1 \\
  ligne2 \\
  ...
\end{flushright}
```

L'environnement `flushright` vous permet de créer un paragraphe ferré à droite et drapeau gauche, c.-à-d. consistant de lignes qui sont alignées sur la marge de droite (fer à droite) et en dentelures (drapeau) sur la marge de gauche. Si vous avez des lignes qui sont trop longues pour entrer entre les marges, alors \LaTeX insère des sauts de ligne sans faire de césure ni dilater ou contracter les espaces inter-mot. Pour forcer un saut de ligne on utilise une double controblaque, `\\`. Pour la forme déclarative, voir Section 8.13.1 [`\raggedleft`], page 70.

Pour un exemple en relation avec cet environnement, voir Section 8.12 [`flushleft`], page 69, où il suffit mutatis mutandis de changer `flushright` en `flushleft`.

8.13.1 `\raggedleft`

Synopsis :

```
{\raggedleft ... }
```

ou

```
\begin{environnement} \raggedleft
  ...
\end{environnement}
```

Une déclaration qui a pour effet que les lignes sont ferrées sur la marge de droite et drapeau sur la gauche, c.-à-d. que les mots sont alignés sur la droite comme sur le fer du composeur, et en dentelure comme les franges d'un drapeau à gauche. Elle peut être utilisée au sein d'un *environnement* tel que `quote` ou d'une `parbox`. Pour la forme par environnement voir Section 8.12 [`flushleft`], page 69.

Contrairement à l’environnement `flushright`, la commande `\raggedleft` ne démarre pas un nouveau paragraphe ; elle change seulement la façon dont L^AT_EX formate les unités de paragraphe. Pour affecter le format d’une unité de paragraphe, la portée de la déclaration doit contenir une ligne à blanc ou la commande `\end` qui termine l’unité de paragraphe.

Pour un exemple en relation avec cet environnement, voir Section 8.12.1 [`\raggedright`], page 69, où il suffit mutatis mutandis de changer `\raggedright` en `\raggedleft`.

8.14 itemize

Synopsis :

```
\begin{itemize}
  \item[étiquette optionnelle du 1er article] texte du 1er article
  \item[étiquette optionnelle du 2e article] texte du 2e article
  ...
\end{itemize}
```

L’environnement `itemize` produit une *liste non ordonnée*, qu’on appelle parfois liste à puces. Il doit y avoir au moins une commande `\item` au sein de l’environnement ; sans quoi L^AT_EX produit l’erreur ‘Something’s wrong--perhaps a missing `\item`’.

L’exemple suivant donne une liste à deux articles.

```
\begin{itemize}
  \item Esquisse au crayon et aquarelle par Cassandra
  \item Portrait Rice
\end{itemize}
```

Par défaut — sans charger par ex. le paquetage `babel` avec une autre langue que USenglish — dans une liste au premier niveau d’imbrication l’étiquette est rendue par une puce, **•**. Le format des étiquettes dépend du niveau d’imbrication ; voir plus bas.

On démarre les articles de liste avec la commande `\item` (voir Section 8.16.1 [`\item`], page 77). Si vous passez à `\item` un argument optionnel en le faisant suivre de crochets, comme dans `\item[étiquette optionnelle]`, alors par défaut *étiquette optionnelle* apparaît en gras et aligné à droite, de sorte qu’elle peut s’étendre dans la marge de gauche. Pour des étiquettes alignées à gauche voir l’environnement Section 8.4 [description], page 63. À la suite `\item` se trouve le texte que l’article qui peut être vide ou contenir plusieurs paragraphes.

Les listes non ordonnées peuvent être imbriquées l’une dans l’autre, sur jusqu’à quatre niveaux de profondeur. Elles peuvent aussi être imbriquées avec d’autres environnements fabriquant des paragraphes, comme `enumerate` (voir Section 8.7 [enumerate], page 65).

L’environnement `itemize` utilise les commandes de `\labelitemi` jusqu’à `\labelitemiv` pour produire les étiquettes par défaut (notez la convention que le nombre romain en bas de casse à la fin du nom de la commande indique le niveau d’imbrication). Voici les marques de chaque niveau :

1. **•** (puce, issue de `\textbullet`)
2. **--** (tiret demi-cadratin gras, issu du `\normalfont\bfseries\textendash`)
3. ***** (astérisque, issu de `\textasteriskcentered`)
4. **·** (point centré verticalement, issu de `\textperiodcentered`)

Si vous utilisez le paquetage `babel` avec la langue `french`, alors il y a des tirets pour tous les niveaux comme c'est l'habitude des Français.

Utilisez `\renewcommand` pour changer les étiquettes. Par exemple, pour que le premier niveau utilise des losanges :

```
\renewcommand{\labelitemi}{\diamond}
```

Les paramètres de `\leftmargini` jusqu'à `\leftmarginvi` définissent la distance entre la marge de gauche de l'environnement susjacent et la marge de gauche de la liste. (On utilise aussi la convention de nommage de la commande avec un nombre romain en bas de casse à la fin pour désigner le niveau d'imbrication). Les valeurs par défaut sont : `2.5em` pour le niveau 1 (`2em` en mode deux-colonnes), `2.2em` pour le niveau 2, `1.87em` pour le niveau 3 et `1.7em` pour le niveau 4, avec des valeurs plus petites pour les niveaux plus profondément imbriqués.

Pour les autres environnements majeurs de listes à étiquettes de L^AT_EX, voir Section 8.4 [description], page 63, et Section 8.7 [enumerate], page 65. Les environnements `itemize`, `enumerate` et `description` utilisent les mêmes paramètres de contrôle de la disposition des listes. Pour leur description, y compris les valeurs par défaut, ainsi que la personnalisation de la disposition, voir Section 8.16 [list], page 72. Le paquetage `enumitem` est utile pour personnaliser les listes.

Dans cet exemple on réduit grandement l'espace de marge pour la liste à puces la plus externe :

```
\setlength{\leftmargini}{1.25em} % valeur par défaut 2.5em
```

En particulier pour les listes avec des articles courts, il peut être désirable d'éliminer l'espace entre les articles. Voici un exemple définissant un environnement `itemize*` sans espacement supplémentaire entre les articles, ni entre les paragraphes au sein d'un seul article (`\parskip` n'est pas spécifique aux listes, voir Section 15.3 [`\parindent` & `\parskip`], page 150):

```
\newenvironment{itemize*}%
  {\begin{itemize}%
   \setlength{\itemsep}{0pt}%
   \setlength{\parsep}{0pt}}%
  {\setlength{\parskip}{0pt}}%
  {\end{itemize}}
```

8.15 environnement `letter`: écrire des lettres postales

Cet environnement est utilisé pour créer des lettres. Voir Chapitre 26 [Letters], page 255.

8.16 `list`

Synopsis :

```
\begin{list}{étiquetage}{espacement}
  \item[étiquette optionnelle du premier article] texte du premier
  article
  \item[étiquette optionnelle du deuxième article] texte du deuxième
  article
  ...
\end{list}
```

L'environnement `list` est un environnement générique qui est utilisé pour construire des listes plus spécialisées. On l'utilise la plupart du temps pour créer des listes via les environnements `description`, `enumerate`, et `itemize` (voir Section 8.4 [description], page 63, Section 8.7 [enumerate], page 65, et Section 8.14 [itemize], page 71). Par ailleurs, beaucoup d'environnements standards de L^AT_EX qui visuellement ne sont pas des listes sont construits en utilisant `list`, ceci comprend `quotation`, `quote`, `center`, `verbatim`, et bien d'autres (voir Section 8.20 [quotation & quote], page 89, voir Section 8.3 [center], page 61, voir Section 8.13 [flushright], page 70).

Ici, on décrit l'environnement `list` en définissant un nouvel environnement personnalisé.

```
\newcounter{compteurlistenommage} % numérote les articles
\newenvironment{nommage}
  {\begin{list}
   {Article~\Roman{compteurlistenommage}.} % argument d'étiquetage
   {\usecounter{compteurlistenommage} % argument d'espacement
    \setlength{\leftmargin}{3.5em}} % ...suite argument d'espacement
  }
  {\end{list}}

\begin{nommage}
  \item Apparaît comme « Article~I. »
  \item[Étiquette spéciale.] Apparaît comme « Étiquette spéciale. »
  \item Apparaît comme « Article~II. »
\end{nommage}
```

L'argument obligatoire *étiquetage*, qui est le premier argument de l'environnement `list`, spécifie comment les articles doivent être étiquetés par défaut. Il peut contenir du texte et des commandes L^AT_EX, comme dans l'exemple précédent où il contient à la fois 'Article' et 'Roman{...}'. L^AT_EX forme l'étiquette en mettant l'argument *étiquetage* dans une boîte de largeur `\labelwidth`. Si l'étiquette est plus large que cela, la matière supplémentaire s'étend sur la droite. Lorsque vous fabriquez une instance de `list` vous pouvez l'emporter sur l'étiquetage par défaut en donnant à `\item` un argument optionnel avec des crochets et le texte, comme dans l'exemple précédent `\item[Étiquette spéciale.]` ; voir Section 8.16.1 [item], page 77.

Le second argument obligatoire *espacement* prend une liste de commandes. L'une de celle qu'on y retrouve est `\usecounter{nomcompteur}` (voir Section 13.2 [\usecounter], page 138). Utilisez la pour dire à L^AT_EX de numérotter les articles en utilisant le compteur fourni. Ce compteur est remis à zéro chaque fois que L^AT_EX entre dans l'environnement, et le compteur est incrémenté par un chaque fois que L^AT_EX rencontre une commande `\item`.

Une autre commande qu'on retrouve dans *espacement* est `\makelabel`, pour construire la boîte de l'étiquette. Par défaut elle positionne le contenu fer à droite. Elle prend un argument, l'étiquette. Elle compose son contenu en mode LR. Un exemple de changement de sa définition consiste dans l'exemple précédent `nommage` à ajouter avant la définition de l'environnement `\newcommand{\namedmakelabel}[1]{\textsc{#1}}`, et entre la commande `\setlength` et la parenthèse clôturant l'argument *espacement* à ajouter aussi `\let\makelabel\namedmakelabel`. Ceci composera les étiquettes en petites capitales. De même, en remplaçant cette seconde ligne de code par `\let\makelabel\fbox` on met

les étiquettes dans une boîte encadrée. Ci-après voici un autre exemple de la commande `\makelabel`, dans la définition de l’environnement `etiquetterouge`.

De même on trouve souvent au sein de *espacement* des commandes pour redéfinir l’espacement de la liste. Ci dessous on liste les paramètres d’espacement et leur valeurs par défaut. (Il se peut que les valeurs par défaut pour les environnements dérivés tels que `itemize` soient différentes des valeurs exposées ici). Voir aussi la figure qui suit la liste. Chaque paramètre est une longueur (voir Chapitre 14 [Lengths], page 141). Les espaces verticaux sont d’ordinaire des longueurs élastiques, avec une composante en `plus` ou en `minus`, pour donner à T_EX de la flexibilité dans le réglage de la page. On change chaque paramètre avec une commande du genre de `\setlength{itemsep}{2pt plus1pt minus1pt}`. Pour obtenir certains effets, ces longueurs doivent être mise à zéro, ou à une valeur négative.

`\itemindent`

Espace supplémentaire horizontal de renforcement, ajouté à `leftmargin`, de la première ligne de chaque article. Sa valeur par défaut est `0pt`.

`\itemsep` Espace vertical entre les articles, ajouté à `\parsep`. Les valeurs par défaut pour les trois premiers niveaux dans les classes de L^AT_EX ‘`article`’, ‘`book`’, et ‘`report`’ à la taille de 10 point sont : `4pt plus2pt minus1pt`, `\parsep` (c.-à-d. `2pt plus1pt minus1pt`), et `\topsep` (c.-à-d. `2pt plus1pt minus1pt`). Les valeurs par défaut à 11 points sont : `4.5pt plus2pt minus1pt`, `\parsep` (c.-à-d. `2pt plus1pt minus1pt`), et `\topsep` (c.-à-d. `2pt plus1pt minus1pt`). Les valeurs par défaut à 12 points sont : `5pt plus2.5pt minus1pt`, `\parsep` (c.-à-d. `2.5pt plus1pt minus1pt`), et `\topsep` (c.-à-d. `2.5pt plus1pt minus1pt`).

`\labelsep`

Espace horizontal entre l’étiquette et le texte d’un article. La valeur par défaut pour les classes L^AT_EX ‘`article`’, ‘`book`’, et ‘`report`’ est de `0.5em`.

`\labelwidth`

Largeur horizontale. La boîte contenant l’étiquette est nominalement de cette largeur. Si `\makelabel` renvoie un texte qui est plus large que cela, alors le renforcement de la première ligne de l’article sera augmenté pour créer l’espace nécessaire à cette matière supplémentaire. Si `\makelabel` renvoie un texte de largeur inférieur ou égale à `\labelwidth` alors le comportement par défaut de L^AT_EX est que l’étiquette est composé fer à droite dans une boîte de cette largeur. Le bord de gauche de la boîte de l’étiquette est à `\leftmargin+\itemindent-\labelsep-\labelwidth` de la marge de gauche de l’environnement enveloppant.

La valeur par défaut pour les classes L^AT_EX ‘`article`’, ‘`book`’, et ‘`report`’ pour les listes de premier niveau vaut `\leftmargin-\labelsep`, (c.-à-d. `2em` en mode à une colonne et `1.5em` en mode deux-colonnes). Au deuxième niveau, c’est `\leftmarginii-\labelsep`, et au troisième niveau `\leftmarginiii-\labelsep`. Ces définitions ont pour effet que le bord de gauche de l’étiquette coïncide avec la marge de gauche de l’environnement enveloppant.

`\leftmargin`

Espace horizontal entre la marge de gauche de l’environnement susjacent (ou la marge de gauche de la page pour une liste de premier niveau), et la marge de gauche de cette liste. Doit être positif ou nul.

Dans les classes de document L^AT_EX standardes, ce paramètre est réglé à la valeur `\leftmargini` pour une liste de premier niveau, alors que pour une liste de deuxième niveau, c'est à dire imbriquée dans une liste de premier niveau, il est réglé à `\leftmarginii`. Pour les listes aux niveaux d'imbrication suivants les valeurs de `\leftmarginiii` à `\leftmarginvi` sont utilisées. (Imbriquer des listes au delà de cinq niveaux produit le message d'erreur 'Too deeply nested').

Les valeurs par défaut des trois premiers niveaux dans les classes L^AT_EX 'article', 'book', et 'report' sont les suivantes : `\leftmargini` vaut 2.5em (ou 2em en mode deux-colonnes), `\leftmarginii` vaut 2.2em, et `\leftmarginiii` vaut 1.87em.

`\listparindent`

Espace horizontal de renforcement supplémentaire, au delà de `\leftmargin`, pour le deuxième paragraphe et les suivants au sein d'un article de liste. Une valeur négative produit un *débord*, c.-à-d. un retrait sur la gauche. Sa valeur par défaut est `0pt`.

`\parsep`

Espace vertical entre les paragraphes d'un article. Les valeurs par défaut pour les trois premiers niveaux dans les classes L^AT_EX 'article', 'book' et 'report' à la taille de 10 point sont : `4pt plus2pt minus1pt`, `2pt plus1pt minus1pt`, and `0pt`. La valeurs par défaut à la taille de 11 point sont : `4.5pt plus2pt minus1pt`, `2pt plus1pt minus1pt`, et `0pt`. La valeurs par défaut à la taille de 12 point sont : `5pt plus2.5pt minus1pt`, `2.5pt plus1pt minus1pt`, et `0pt`.

`\partopsep`

Espace vertical ajouté, en plus de `\topsep+\parskip`, en haut et en bas de l'environnement tout entier lorsque la liste considérée est précédée d'une ligne à blanc. (Une ligne à blanc dans le source L^AT_EX avant la liste change l'espacement à la fois en haut et en bas de la liste ; que la ligne suivant la liste soit à blanc est sans importance).

Les valeurs par défaut pour les trois premiers niveaux dans les classes L^AT_EX 'article', 'book', et 'report' à la taille de point sont : `2pt plus1 minus1pt`, `2pt plus1pt minus1pt`, et `1pt plus0pt minus1pt`. Les valeurs par défaut à la taille de 11 point sont : `3pt plus1pt minus1pt`, `3pt plus1pt minus1pt`, et `1pt plus0pt minus1pt`. Les valeurs par défaut à la taille de 12 point sont : `3pt plus2pt minus3pt`, `3pt plus2pt minus2pt`, et `1pt plus0pt minus1pt`.

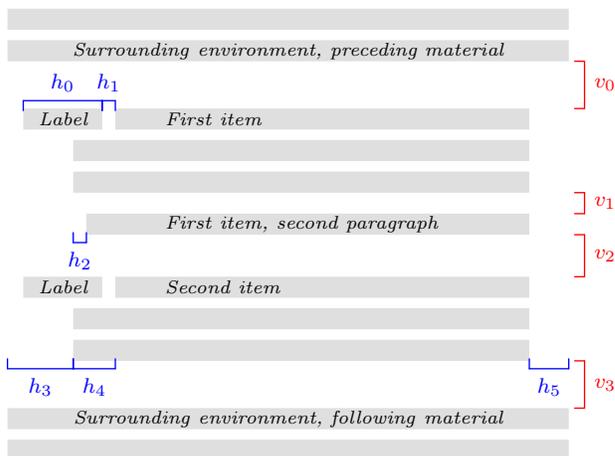
`\rightmargin`

Espace horizontal entre la marge de droite de la liste et la marge de droite de l'environnement l'englobant. Vaut par défaut `0pt`. Doit être positif ou nul.

`\topsep`

Espace vertical ajouté aussi bien au haut qu'au bas de la liste, en plus de `\parskip` (voir Section 15.3 [`\parindent` & `\parskip`], page 150). Les valeurs par défaut pour les trois premiers niveaux dans les classes L^AT_EX 'book', et 'report' à la taille de 10 point sont : `8pt plus2pt minus4pt`, `4pt plus2pt minus1pt`, and `2pt plus1pt minus1pt`. Les valeurs par défaut à la taille de 11 point sont : `9pt plus3pt minus5pt`, `4.5pt plus2pt minus1pt`, and `2pt plus1pt minus1pt`. Les valeurs par défaut à la taille de 12 point sont : `10pt plus4pt minus6pt`, `5pt plus2.5pt minus1pt`, et `2.5pt plus1pt minus1pt`.

La figure suivante illustre les distances horizontales et verticales.



Les longueurs illustrées sont listées ci-dessous. La relation clef est que le bord droit du crochet pour $h1$ est à la même position horizontale que le bord droit du crochet pour $h4$, de sorte que le bord de gauche de la boîte de l'étiquette est à $h3+h4-(h0+h1)$.

$v0$	vaut <code>\topsep+\parskip</code> si l'environnement <code>list</code> ne commence pas un nouveau paragraphe, et <code>\topsep+\parskip+\partopsep</code> sinon
$v1$	<code>\parsep</code>
$v2$	<code>\itemsep+\parsep</code>
$v3$	Même valeur que $v0$. (Cet espace est affecté selon qu'une ligne à blanc apparaît dans le source au dessus de l'environnement ; la présence d'une ligne à blanc au dessous de l'environnement est sans effet).
$h0$	<code>\labelwidth</code>
$h1$	<code>\labelsep</code>
$h2$	<code>\listparindent</code>
$h3$	<code>\leftmargin</code>
$h4$	<code>\itemindent</code>
$h5$	<code>\rightmargin</code>

Les marges de gauche et de droite de la liste, affichées ci-dessus comme $h3$ et $h5$, sont exprimées relativement à celles fournies par l'environnement englobant, ou par rapport aux marges de la page pour une liste au niveau le plus externe. La largeur de ligne utilisée pour la composition des articles de la liste est `\linewidth` (voir Section 5.5 [Page layout parameters], page 32). Par exemple, réglez la marge de gauche de la liste à un quart de la distance entre les marges de gauche et de droite de l'environnement englobant avec `\setlength{\leftmargin}{0.25\linewidth}`.

Les sauts de page au sein d'une structure de liste sont contrôlés par les paramètres ci-dessous. Pour chacun d'eux, la valeur par défaut de L^AT_EX est `-\@lowpenalty`, c.-à-d. -51.

Comme c'est négatif, cela encourage d'une certaine façon un saut de page à tout endroit. On peut changer cela avec un réglage de, par ex., `\@beginparpenalty=9999` ; une valeur de 10000 interdit un saut de page.

`\@beginparpenalty`

La pénalité de saut de page pour un saut avant la liste (par défaut -51).

`\@itempenalty`

La pénalité pour un saut de page avant un article de liste (par défaut -51).

`\@endparpenalty`

La pénalité pour un saut de page après une liste (par défaut -51).

Le paquetage `enumitem` est utile pour personnaliser les listes.

Dans l'exemple suivant les étiquettes sont en rouge. Elles sont numérotées, et le bord gauche des étiquettes est aligné avec le bord gauche du texte des articles. Voir Section 13.2 [`\usecounter`], page 138.

```
\usepackage{color}
\newcounter{cnt}
\newcommand{\makeredlabel}[1]{\textcolor{red}{#1.}}
\newenvironment{redlabel}
{\begin{list}
{\arabic{cnt}}
{\usecounter{cnt}
\setlength{\labelwidth}{0em}
\setlength{\labelsep}{0.5em}
\setlength{\leftmargin}{1.5em}
\setlength{\itemindent}{0.5em} % vaut \labelwidth+\labelsep
\let\makelabel=\makeredlabel
}
}
{\end{list}}
```

8.16.1 `\item`: Une entrée dans une liste

Synopsis :

```
\item texte de l'article
```

ou

```
\item[étiquette optionnelle] texte de l'article
```

Un entrée dans une liste. Les entrées sont préfixées par une étiquette, dont la valeur par défaut dépend du type de liste.

Parce que l'argument optionnel *étiquette optionnelle* est entouré de crochets ([et]), pour utiliser des crochets au sein de l'argument optionnel vous devez les cacher entre accolades, comme dans `\item[Crochet fermant, {}]`. De même, pour utiliser un crochet ouvrant comme premier caractère du texte d'un article, cachez le aussi au sein d'accolade. Voir Section 2.4 [L^AT_EX command syntax], page 5.

Dans cet exemple la liste `enumerate` a deux articles qui utilise l'étiquette par défaut et une qui utilise l'étiquette optionnelle.

```
\begin{enumerate}
```

```

\item Moe
\item[sometimes] Shemp
\item Larry
\end{enumerate}

```

Le premier article est étiqueté ‘1.’, le deuxième article est étiqueté ‘sometimes’, et le troisième article est étiqueté ‘2.’ (notez que, à cause de l’étiquette optionnelle dans le second article, le troisième article ne reçoit pas l’étiquette ‘3.’).

8.16.2 `trivlist`: Une forme restreinte de `list`

Synopsis :

```

\begin{trivlist}
...
\end{trivlist}

```

Une version restreinte de l’environnement `list` dans laquelle les marges sont sans renforcement et `\item` sans l’argument optionnel ne produit pas de texte. Elle est utilisée le plus souvent au sein de macros, de sorte à définir un environnement où la commande `\item` fait partie de la définition de l’environnement. Par exemple, l’environnement `center` est défini pour l’essentiel comme ceci :

```

\newenvironment{center}
{\begin{trivlist}\centering\item\relax}
{\end{trivlist}}

```

Utiliser `trivlist` de la sorte permet à la macro d’hériter du code mutualisé : combiner l’espace vertical provenant de deux environnements adjacents ; détecter si le texte suivant l’environnement doit être considéré comme un nouveau paragraphe ou la suite du paragraphe précédent ; ajuster les marges de gauche et de droit pour d’éventuels environnements de liste imbriqués.

En particulier, `trivlist` utilise les valeurs courantes des paramètres de liste (voir Section 8.16 [list], page 72), à ceci près que `\parsep` est réglé à la valeur de `\parskip`, et que `\leftmargin`, `\labelwidth`, et `\itemindent` sont réglés à zéro.

Cet exemple produit les articles comme deux paragraphes, à ceci près que (par défaut) ils n’ont pas de renforcement de paragraphe et sont séparés verticalement.

```

\begin{trivlist}
\item C’est un fameux trois-mâts, fin comme un oiseau
\item Hisse et ho, Santiano.
\end{trivlist}

```

8.17 `math`

Synopsis :

```

\begin{math}
maths
\end{math}

```

L’environnement `math` insère les *maths* donnés au sein du texte en cours. `\(...\)` et `$. . .$` sont des synonymes. Voir Chapitre 16 [Math formulas], page 153.

8.18 minipage

Synopsis :

```
\begin{minipage}{largeur}
  matière
\end{minipage}
```

ou

```
\begin{minipage}[position][hauteur][pos-interne]{largeur}
  matière
\end{minipage}
```

Place *matière* dans une boîte qui est *largeur* de large. C'est comme une version réduite d'une page ; elle peut contenir ses propres notes en bas de page, listes à puces, etc. (Il y a quelques restrictions, notamment qu'elle ne peut pas avoir de flottants). Cette boîte ne peut pas être coupée sur plusieurs pages. Ainsi `minipage` est similaire à la commande `\parbox` (voir Section 20.3 [`\parbox`], page 206), mais contrairement à `\parbox`, d'autres environnements de production de paragraphe peuvent être utilisés au sein d'une `minipage`.

Cet exemple fait 7,5cm de large, et comprend deux paragraphes.

```
\begin{minipage}{7.5cm}
  Stephen Kleene fut un fondateur de la théorie de la récursivité.

  Il fut l'élève de Church, l'auteur de trois textes qui firent
  autorité, et le président de l'Association for Symbolic Logic, et il
  fut récompensé par la National Medal of Science.
\end{minipage}
```

Voir plus bas pour une discussion sur le renforcement de paragraphe au sein d'une `minipage`.

L'argument obligatoire *largeur* est une longueur rigide (voir Chapitre 14 [Lengths], page 141). Il donne la largeur de la boîte dans laquelle *matière* est composée.

Il y a trois arguments optionnels, *position*, *hauteur*, et *pos-interne*. Il n'est pas nécessaire de fournir les trois à la fois. Par exemple, obtenez les valeurs par défaut pour *position* et réglez *hauteur* avec `\begin{minipage}[c][2.54cm]{\columnwidth} matière \end{minipage}`. (Obtenez la hauteur naturelle avec un argument vide, []).

L'argument optionnel *position* gouverne la façon dont la `minipage` s'aligne verticalement avec la matière l'entourant.

- c (Synonyme m) Valeur par défaut. Positionne la `minipage` de sorte que son centre vertical soit aligné avec le centre des lignes de texte adjacentes.
- t Aligne la ligne du haut de la `minipage` sur la ligne de base du texte l'entourant (comme `\vtop` en T_EX de base).
- b Aligne la ligne du bas de `minipage` avec la ligne de base du texte l'entourant (comme `\vbox` en T_EX de base).

Pour voir l'effet de ces options, comparez le résultat de la compilation de

```
---\begin{minipage}[c]{0.25in}
  first\\ second\\ third
\end{minipage}
```

avec ce qui se passe en remplaçant `c` par `b` ou `t`.

L'argument optionnel *hauteur* est une longueur rigide (voir Chapitre 14 [Lengths], page 141). Il règle la hauteur de la `minipage`. Vous pouvez donner n'importe quelle valeur supérieure, égale ou inférieure à la hauteur naturelle de la `minipage` sans que L^AT_EX ne produise une erreur ou un avertissement. Vous pouvez aussi la régler à une hauteur de valeur nulle ou négative.

L'argument optionnel final *pos-interne* contrôle la disposition de *matière* au sein de la boîte. Voici les valeurs possibles (la valeur par défaut est la valeur de l'argument *position*).

- `t` Place *matière* en haut de la boîte.
- `c` Centre *matière* verticalement.
- `b` Place *matière* en bas de la boîte.
- `s` Dilate *matière* verticalement ; elle doit contenir de l'espace dilatable verticalement.

L'argument *pos-interne* a un sens quand l'option *hauteur* est réglée à une valeur supérieure à la hauteur naturelle de la `minipage`. Pour voir l'effet des options, exécutez l'exemple suivant avec les différents choix possibles au lieu de `b`.

```

Texte avant
\begin{center}
  ---\begin{minipage}[c][8cm][b]{0.25\textwidth}
    premier\ \ deuxi\eme\ \ troisi\eme
  \end{minipage}
\end{center}
Texte après

```

Par défaut, les paragraphes ne sont pas renforcés au sein d'un environnement `minipage`. Vous pouvez restaurer le renforcement avec une commande telle que `\setlength{\parindent}{1pc}` placé au début de *matière*.

Les notes en bas de page au sein d'un environnement `minipage` sont gérées d'une façon qui est particulièrement utile pour mettre des notes en bas de page dans des figures ou des tableaux. Une commande `\footnote` ou `\footnotetext` met la note en bas de page au bas de la `minipage` au lieu de la mettre au bas de la page, et elle utilise le compteur `\mpfootnote` au lieu du compteur ordinaire `footnote` (voir Chapitre 13 [Counters], page 138).

L'exemple ci-dessous place la note argument de `\footnote` en bas de la table, et non en bas de la page.

```

\begin{center}
  \begin{minipage}{6.5cm}
    \begin{center}
      \begin{tabular}{ll}
        \textsc{Monarque} & \textsc{R\`egne}
        \hline
        Elizabeth II & &63 ans\footnote{à ce jour}
        Victoria & &63 ans
        George III & &59 ans
      \end{tabular}
    \end{center}
  \end{minipage}
\end{center}

```

```
\end{minipage}
\end{center}
```

Si vous imbriquez des minipages, alors il y a une bizarrerie dans l'utilisation des notes `\footnote`. Les notes apparaissent au bas du texte terminé par la prochaine `\end{minipage}` ce qui n'est peut-être pas leur place logique.

L'exemple suivant place un tableau de données côte à côte d'un graphique. Ils sont verticalement centrés.

```
% siunitx permet d'avoir le type de colonne S dans les tableaux,
% pour l'alignement sur le séparateur décimal, réglé comme virgule.
\usepackage[output-decimal-marker={,}]{siunitx}
\newcommand*\vcenteredhbox[1]{\begin{tabular}{@{}c@{}}#1\end{tabular}}
...
\begin{center}
\vcenteredhbox{\includegraphics[width=0.3\textwidth]{nyc.png}}
\hspace{0.1\textwidth}
\begin{minipage}{0.5\textwidth}
\begin{tabular}{r|S}
% \multicolumn pour supprimer la barre verticale entre les titres
% de colonnes
\multicolumn{1}{r}{Quartier} &
% les accolades empêchent siunitx de voir le point comme un
% séparateur décimal
{Pop. (million)} \\ \hline
The Bronx & 1.5 \\
Brooklyn & 2.6 \\
Manhattan & 1.6 \\
Queens & 2.3 \\
Staten Island & 0.5
\end{tabular}
\end{minipage}
\end{center}
```

8.19 picture

Synopsis :

```
\begin{picture}(largeur,hauteur)
  commande picture
\end{picture}
```

ou

```
\begin{picture}(largeur,hauteur)(décalagex,décalagey)
  commande picture
\end{picture}
```

où on a autant de *commande picture*(s) qu'on le désire.

L'environnement `picture` vous permet de créer des dessins simples contenant des lignes, des flèches, des boîtes, des cercles, et du texte. Cet environnement n'est pas obsolète, mais

les nouveaux documents utilisent typiquement des systèmes de création de graphiques bien plus puissants tels que TikZ, PStricks, MetaPost, ou Asymptote. Aucun de ceux-ci n'est couvert dans ce document ; voir le CTAN.

Pour commencer, voici un exemple illustrant la méthode du parallélogramme pour additionner deux vecteurs.

```
\setlength{\unitlength}{1cm}
\begin{picture}(6,6) % la boîte de l'image fera 6cm de large sur 6cm
                    % de haut
  \put(0,0){\vector(2,1){4}} % la pente du vecteur est de 1/2
  \put(2,1){\makebox(0,0)[l]{\ premier terme}}
  \put(4,2){\vector(1,2){2}}
  \put(5,4){\makebox(0,0)[l]{\ second terme}}
  \put(0,0){\vector(1,1){6}}
  \put(3,3){\makebox(0,0)[r]{somme\ }}
\end{picture}
```

L'environnement `picture` a un argument obligatoire qui est une paire de nombres réels positifs (*largeur, hauteur*). Le point est utilisé comme séparateur décimal quand ils ne sont pas entiers. En les multipliant par `\unitlength` on obtient la taille nominale de la sortie, c.-à-d. l'espace que L^AT_EX réserve sur la page en sortie. Cette taille nominale n'est pas nécessairement la taille réelle de l'image ; L^AT_EX dessine aussi les parties de l'image en dehors de la boîte de l'image.

L'environnement `picture` a aussi un argument optionnel (*décalagex, décalagey*). Il sert à décaler l'origine. Contrairement aux arguments optionnels ordinaires, cet argument n'est pas contenu entre crochets. Comme pour l'argument obligatoire, c'est une paire de deux nombres. En les multipliant par `\unitlength` on obtient les coordonnées du point situé dans le coin inférieur gauche de l'image.

Par exemple, si `\unitlength` a été réglée à 1mm, la commande

```
\begin{picture}(100,200)(10,20)
```

produit un dessin de largeur 100 millimètres et hauteur 200 millimètres. Son origine est le point (10mm,20mm) et donc le coin inférieur gauche est là, et le coin supérieur droit est à (110mm,220mm). Lorsque vous tracez un dessin, typiquement vous omettez l'argument optionnel, laissant l'origine au coin inférieur gauche. Si ensuite vous voulez modifier le dessin en translatant tout, vous pouvez juste ajouter l'argument optionnel approprié.

Chaque *commande picture* dit à L^AT_EX où placer quelque chose en donnant sa position. Une *position* est une paire telle que (2.4,-5) donnant les coordonnées en x et en y. Une *coordonnée* n'est pas une longueur, c'est un nombre réel (il utilise le point '.' comme séparateur décimal et peut avoir un signe moins). Elle spécifie une longueur en termes de la longueur unité `\unitlength`, de sorte que si `\unitlength` a été réglée à 1cm, alors la coordonnées '2.54' spécifie une longueur de 2,54 centimètres.

La valeur par défaut de L^AT_EX pour `\unitlength` est 1pt. C'est une longueur rigide (voir Chapitre 14 [Lengths], page 141). Utilisez la commande `\setlength` pour la modifier (voir Section 14.2 [\setlength], page 143). Effectuez ce changement uniquement en dehors de l'environnement `picture`.

L'environnement `picture` prend en charge les expressions arithmétiques standards en plus des nombres.

Les coordonnées sont données relativement à une origine, qui est par défaut au coin en bas à gauche de l'image. Notez que lorsque une position apparaît en tant qu'argument, comme dans `\put(1,2){...}`, elle n'est pas entourée d'accolades puisque les parenthèses servent à délimiter l'argument. De même, contrairement à certains systèmes de dessin par ordinateur, l'axe des ordonnées y est orienté vers le haut de la page, par ex. $y = 1$ est *au-dessus* de $y = 0$.

Il y a quatre manières de placer des choses dans une image : `\put`, `\multiput`, `\qbezier`, et `\graphpaper`. La plus utilisée est `\put`. Le code suivant

```
\put (11.3,-0.3){...}
```

met l'objet spécifié par `...` dans le dessin `picture`, avec son point de référence aux coordonnées (11.3, -0.3). Les points de référence des divers objets sont décrits plus bas.

La commande `\put` crée une *LR box* (voir Chapitre 17 [Modes], page 185). Vous pouvez mettre tout ce qui peut aller dans une `\mbox` (voir Section 20.1 [`\mbox` & `\makebox`], page 204) dans l'argument texte de la commande `\put`. Quand vous faites cela, le point de référence devient le coin inférieur gauche de la boîte. Dans cette image :

```
\setlength{\unitlength}{1cm}
... \begin{picture}(1,1)
    \put(0,0){\line(1,0){1}}
    \put(0,0){\line(1,1){1}}
\end{picture}
```

les trois points sont juste légèrement à la gauche du point où les deux lignes forment un angle. (À noter aussi que `\line(1,1){1}` ne demande pas une ligne de longueur un, mais une ligne dont l'étendue selon la coordonnée x est de 1).

Les commandes `\multiput`, `\qbezier`, et `\graphpaper` sont décrites plus bas.

Vous pouvez également utiliser cet environnement pour placer une matière quelconque à un emplacement exact. Par exemple :

```
\usepackage{color,graphicx} % dans le préambule
...
\begin{center}
\setlength{\unitlength}{\textwidth}
\begin{picture}(1,1) % réserve un espace large et haut de \textwidth
  \put(0,0){\includegraphics[width=\textwidth]{desertedisland.jpg}}
  \put(0.25,0.35){\textcolor{red}{X Trésor ici}}
\end{picture}
\end{center}
```

Le X rouge sera précisément à un quart de `\textwidth` à droite de la marge de gauche, et `0.35\textwidth` au dessus du bas de l'image. Un autre exemple de cette utilisation est de placer un code similaire dans l'en-tête de page pour obtenir une matière répétée sur chaque page.

8.19.1 `\put`

```
\put(xcoord,ycoord){contenu}
```

La commande `\put` place *contenu* aux coordonnées (*xcoord*, *ycoord*). Voir la discussion sur les coordonnées et `\unitlength` dans Section 8.19 [picture], page 81. Le *contenu* est

traité en mode LR (voir Chapitre 17 [Modes], page 185) de sorte qu'il ne peut pas contenir de sauts de ligne.

L'exemple ci-dessous inclut le texte dans le dessin `picture`.

```
\put(4.5,2.5){Faire la manœuvre de \textit{retrait}}
```

Le point de référence, la position (4.5,2.5), est le point en bas à gauche de texte, au coin en bas à gauche du 'F'.

8.19.2 `\multiput`

Synopsis :

```
\multiput(x,y)(delta_x,delta_y){nbre-copies}{obj}
```

La commande `\multiput` copie l'objet *obj* en tout *nbre-copies* fois, avec un incrément de *delta_x,delta_y*. Le *obj* est placé en premier à la position (x,y) , puis à la position $(x + \delta x, y + \delta y)$, et ainsi de suite.

L'exemple ci-dessous dessine une grille simple avec une ligne sur cinq en gras (voir aussi Section 8.19.4 [`\graphpaper`], page 85).

```
\begin{picture}(10,10)
  \linethickness{0.05mm}
  \multiput(0,0)(1,0){10}{\line(0,1){10}}
  \multiput(0,0)(0,1){10}{\line(1,0){10}}
  \linethickness{0.5mm}
  \multiput(0,0)(5,0){3}{\line(0,1){10}}
  \multiput(0,0)(0,5){3}{\line(1,0){10}}
\end{picture}
```

8.19.3 `\qBezier`

Synopsis :

```
\qBezier(x1,y1)(x2,y2)(x3,y3)
\qBezier[quant](x1,y1)(x2,y2)(x3,y3)
```

Dessine une courbe de Bézier quadratique dont les points de contrôle sont donnés par les trois arguments obligatoires $(x1,y1)$, $(x2,y2)$, et $(x3,y3)$. c.-à-d. que la courbe va de $(x1,y1)$ à $(x3,y3)$, est quadratique, et est telle que la tangente en $(x1,y1)$ passe par $(x2,y2)$, et de même pour la tangente en $(x3,y3)$.

L'exemple suivant dessine une courbe des coordonnées (1,1) à (1,0).

```
\qBezier(1,1)(1.25,0.75)(1,0)
```

La tangente de la courbe en (1,1) passe par (1.25,0.75), et la tangente en (1,0) fait de même.

L'argument optionnel *quant* donne le nombre de points intermédiaires calculés. Par défaut la commande dessine une courbe lisse dont le nombre maximum de points est `\qBeziermax` (on change cette valeur avec `\renewcommand`).

L'exemple suivant dessine un rectangle avec un haut ondulé, en utilisant `\qBezier` pour cette courbe.

```
\begin{picture}(8,4)
  \put(0,0){\vector(1,0){8}} % axe des x
  \put(0,0){\vector(0,1){4}} % axe des y
```

```

\put(2,0){\line(0,1){3}}      % côté gauche
\put(4,0){\line(0,1){3.5}}    % côté droit
\qbezier(2,3)(2.5,2.9)(3,3.25)
  \qbezier(3,3.25)(3.5,3.6)(4,3.5)
\thicklines % ci-dessous, les lignes sont deux fois plus épaisses
\put(2,3){\line(4,1){2}}
\put(4.5,2.5){\framebox{Méthode des trapèzes}}
\end{picture}

```

8.19.4 `\graphpaper`

Synopsis :

```

\graphpaper(x_init,y_init)(x_dimen,y_dimen)
\graphpaper[espacement](x_init,y_init)(x_dimen,y_dimen)

```

Dessine un quadrillage. Le paquetage `graphpap` est nécessaire. L'origine du quadrillage est (x_init, y_init) . Les lignes de la grille sont espacées de *espacement* unités (par défaut 10). Le quadrillage s'étend sur *x_dimen* unités vers la droite et *y_dimen* vers le haut. Tous les arguments doivent être des entiers positifs.

L'exemple ci-dessous fait une grille avec sept lignes verticales et onze horizontales.

```

\usepackage{graphpap} % dans le préambule
...
\begin{picture}(6,20) % dans le corps du document
  \graphpaper[2](0,0)(12,20)
\end{picture}

```

Les lignes sont numérotées toutes les dix unités.

8.19.5 `\line`

Synopsis :

```

\line(x_avance,y_monte){excursion}

```

Dessine une ligne. Sa pente est telle que la ligne monte de *y_monte* verticalement pour toute avancée horizontale de *x_avance*. L'*excursion* est le déplacement horizontal total — ce n'est pas la longueur du vecteur, mais la variation en *x*. Dans le cas spécial des lignes verticales, où $(x_avance, y_monte) = (0, 1)$, l'*excursion* donne le déplacement en *y*.

L'exemple suivant dessine une ligne commençant aux coordonnées (1,3).

```

\put(1,3){\line(2,5){4}}

```

Pour toute avancée de 2 vers la droite, cette ligne monte de 5. Parce que *excursion* spécifie qu'elle avance de 4, elle monte en tout de 10. De sorte que son extrémité finale est $(1, 3) + (4, 10) = (5, 13)$. En particulier, notez que *excursion* = 4 n'est pas la longueur de la ligne, mais c'est son déplacement en *x*.

Les arguments *x_avance* et *y_monte* sont des entiers qui peuvent être positifs, négatifs, ou nuls (si les deux valent 0, alors L^AT_EX traite le second comme 1). Avec `\put(x_init,y_init){\line(x_avance,y_monte){travel}}`, si *x_avance* est négatif alors l'extrémité finale de la ligne a sa première coordonnée inférieure à *x_init*. Si *y_monte* est négatif alors l'extrémité finale a sa seconde coordonnées inférieure à *y_init*.

Si *excursion* est négatif alors vous obtenez `LaTeX Error: Bad \line or \vector argument`.

De base \LaTeX peut seulement dessiner des lignes avec une plage limitée de pentes parce que ces lignes sont constituées par la juxtaposition de petits segments de droite issus de polices préfabriquées. Les deux nombres *x_avance* et *y_monte* prennent des valeurs entières de -6 jusqu'à 6 . De plus ils doivent être premiers entre eux, de sorte que $(x_avance, y_monte)=(2,1)$ est licite, mais pas $(x_avance, y_monte)=(4,2)$ (is vous choisissez cette dernière forme alors au lieu de lignes vous obtiendrez des séquences de têtes de flèches ; la solution est de revenir à la première forme). Pour obtenir des lignes de pente arbitraire et plein d'autres formes dans un système tel que `picture`, voir le paquetage `pict2e` (<https://ctan.org/pkg/pict2e>). Une autre solution est d'utiliser un système graphique complet tel que TikZ, PSTricks, MetaPost, ou Asymptote.

8.19.6 `\linethickness`

La commande `\linethickness{dim}` déclare que l'épaisseur des lignes horizontales et verticales dans un environnement `picture` vaut *dim*, ce qui doit être une longueur positive (voir Chapitre 14 [Lengths], page 141). Elle diffère de `\thinlines` et `\thicklines` en ce qu'elle n'affecte pas l'épaisseur des lignes obliques, cercles, ou ovaux (voir Section 8.19.10 [Noval], page 87).

8.19.7 `\thinlines`

La commande `\thinlines` déclare que l'épaisseur de trait des lignes (voir Section 8.19.5 [line], page 85), cercles (voir Section 8.19.9 [\circle], page 86) et ovaux (voir Section 8.19.10 [Noval], page 87) qui suivent dans un environnement `picture` est réglée à $0,4\text{pt}$. C'est l'épaisseur par défaut, et donc cette commande n'est nécessaire que si l'épaisseur a été changée par soit Section 8.19.6 [\linethickness], page 86, soit Section 8.19.8 [\thicklines], page 86.

8.19.8 `\thicklines`

La commande `\thicklines` déclare que l'épaisseur de trait des lignes (voir Section 8.19.5 [line], page 85), cercles (voir Section 8.19.9 [\circle], page 86) et ovaux (voir Section 8.19.10 [Noval], page 87) qui suivent dans un environnement `picture` est réglée à $0,8\text{pt}$. Voir aussi soit Section 8.19.6 [\linethickness], page 86, soit Section 8.19.7 [\thinlines], page 86. Cette commande est illustrée dans l'exemple de la méthode des trapèzes de Section 8.19.3 [\qbezier], page 84.

8.19.9 `\circle`

Synopsis :

```
\circle{diamètre}
\circle*{diamètre}
```

La commande `\circle` produit un cercle dont le diamètre est le plus proche possible de celui spécifié. La forme étoilée (en `*`) de la commande dessine un disque, c.-à-d. un cercle plein.

L'exemple suivant trace un cercle de rayon 6 , centré en $(5,7)$.

```
\put(5,7){\circle{6}}
```

Les rayons disponibles pour `circle` sont, en points, les nombres pair de 2 à 20 inclus. Pour `\circle*` ce sont tous les entiers de 1 à 15.

8.19.10 `\oval`

Synopsis :

```
\oval(largeur,hauteur) [portion]
```

La commande `\oval` produit un rectangle aux coins arrondis, ci-après désigné par *ovale*. L'argument optionnel *portion* vous permet de ne produire qu'une moitié de l'ovale selon ce qui suit :

- t sélectionne la moitié supérieure ;
- b sélectionne la moitié inférieure ;
- r sélectionne la moitié de droite ;
- l sélectionne la moitié de gauche.

Il est également possible de ne produire qu'un quart de l'ovale en réglant *portion* à `tr`, `br`, `bl`, ou `tl`.

L'exemple suivant dessine la moitié supérieure d'un ovale qui est large de 3, et haut de 7.

```
\put(5,7){\oval(3,7)[t]}
```

Le (5,7) est le centre de l'ovale entier, et non juste le centre de la moitié supérieure.

Ces formes ne sont pas des ellipses. Ce sont des rectangles dont les « coins » sont fabriqués avec des quarts de cercle. Ces cercles ont un rayon maximal de 20 pt (voir Section 8.19.9 [`\circle`], page 86, pour les tailles). Ainsi les grands ovales sont juste des cadres aux coins quelque peu arrondis.

8.19.11 `\shortstack`

Synopsis :

```
\shortstack[position]{ligne-1\ ...}
```

La commande `\shortstack` produit une pile verticale d'objets.

L'exemple suivant étiquette l'axe des *y* en écrivant le mot 'axe' au dessus du mot '*y*'.

```
\setlength{\unitlength}{1cm}
\begin{picture}(5,2.5)(-0.75,0)
  \put(0,0){\vector(1,0){4}} % axe des x
  \put(0,0){\vector(0,1){2}} % axe des y
  \put(-0.2,2){\makebox(0,0)[r]{\shortstack[r]{axe\ $y$}}}
\end{picture}
```

Pour une pile `\shortstack`, le point de référence est le coin inférieur gauche de la pile. Dans l'exemple plus haut la boîte `\makebox` (voir Section 20.1 [`\mbox` & `\makebox`], page 204) met la pile fer à droite dans une boîte de largeur nulle, de sorte qu'au bout du compte la pile est située légèrement à la gauche de l'axe des *y*.

Les positions valides sont :

- r Aligne les objets fer à droite

- l Aligne les objets fer à gauche
- c Centre les objets (position par défaut)

Les objets sont séparés en lignes avec `\\`. « Short stack » signifie « pile courte » en anglais : ces piles sont courtes au sens où, contrairement à un environnement `tabular` ou `array`, dans leur cas les lignes ne sont pas espacées d'un interligne uniforme. Ainsi, dans `\shortstack{X\o\o\X}` les première et dernière lignes sont plus hautes que les deux du milieu, et donc l'interligne entre les deux du milieu est plus petit que celui entre la troisième et la dernière ligne. Vous pouvez ajuster les hauteurs et profondeurs de ligne en mettant l'interligne habituel avec `\shortstack{X\ \strut o\o\X}` (voir Section 19.13 [`\strut`], page 200), ou explicitement, avec une boîte de largeur nulle `\shortstack{X\ \rule{0pt}{12pt} o\o\X}`, ou encore en utilisant l'argument optionnel de `\\`, comme dans `\shortstack{X\ [2pt] o\o\X}`.

La commande `\shortstack` est également disponible hors de l'environnement `picture`.

8.19.12 `\vector`

Synopsis :

```
\vector(xpente,ypente){longueur}
```

La commande `\vector` dessine une ligne fléchée de la *longueur* et direction (*xpente*, *ypente*) spécifiées. Les valeurs de *xpente* et *ypente* doivent être comprises entre -4 and $+4$ y compris.

8.19.13 `\makebox`

Synopsis :

```
\makebox(largeur,hauteur)[position]{texte}
```

La commande `\makebox` pour l'environnement `picture` est similaire à la commande normale `\makebox` à ceci près que vous devez spécifier une *largeur* et une *hauteur* implicitement multipliés par `\unitlength`.

L'argument optionnel, `[position]`, spécifie le quadrant dans lequel votre *texte* apparaît. Vous pouvez sélectionner jusqu'à deux spécificateurs parmi les suivants :

- t Place l'article au sommet du rectangle.
- b Place l'article en bas du rectangle.
- l Place l'article sur la gauche.
- r Place l'article sur la droite.

Voir Section 20.1 [`\mbox` & `\makebox`], page 204.

8.19.14 `\framebox`

Synopsis :

```
\framebox(largeur,hauteur)[pos]{...}
```

La commande `\framebox` est similaire à `\makebox` (voir section précédent), à ceci près qu'elle met un cadre autour de l'extérieur du contenu de la boîte qu'elle crée.

La commande `\framebox` produit un filet d'épaisseur `\fboxrule`, et laisse un espace `\fboxsep` entre le filet et le contenu de la boîte.

8.19.15 `\frame`

Synopsis :

```
\frame{texte}
```

La commande `\frame` met un cadre rectangulaire autour de *texte*. Le point de référence est le coin en bas à gauche du cadre. Aucun espace supplémentaire n'est mis entre le cadre et l'objet *texte*.

8.19.16 `\dashbox`

Dessine une boîte avec une ligne en tiretés. Synopsis :

```
\dashbox{tlong}(rlargeur,rhauteur)[pos]{texte}
```

`\dashbox` crée un rectangle en tiretés autour de *texte* au sein d'un environnement `picture`. Les tirets sont long de *tlong* unités, et le rectangle a pour largeur totale *rlargeur* et hauteur totale *rhauteur*. Le *texte* est positionné par l'argument optionnel *pos*.

Une boîte en tiretés a meilleure allure quand les paramètres *rlargeur* et *rhauteur* sont des multiples de *tlong*.

8.20 quotation et quote

Synopsis :

```
\begin{quotation}
  texte
\end{quotation}
```

ou

```
\begin{quote}
  texte
\end{quote}
```

Cite du texte.

Dans les deux environnements, les marges sont renforcées à la fois sur la gauche et la droite. Le texte est justifié des deux côtés sur la marge. De même que dans le texte principal, laisser une ligne à blanc au sein du texte produit un nouveau paragraphe.

Pour comparer les deux : dans l'environnement `quotation`, les paragraphes sont renforcés de 1,5em et l'espace entre les paragraphes est petit, 0pt plus 1pt. Dans l'environnement `quote`, les paragraphes ne sont pas renforcés et il y a un espace vertical entre les paragraphes (c'est la longueur élastique `\parsep`). Ainsi, l'environnement `quotation` convient mieux aux documents où les nouveaux paragraphes sont marqués par un renforcement plutôt que par une séparation verticale. De plus, `quote` convient mieux pour des citations courtes ou une suite de citations courtes.

```
\begin{quotation} \small\it
  Il y a quatre-vingt sept ans
  ... ne disparaîtra jamais de la surface de la terre.
  \hspace{1em plus 1fill}--- Abraham Lincoln
\end{quotation}
```

8.21 tabbing

Synopsis :

```
\begin{tabbing}
lgn1col1 \= lgn1col2 ... \\
lgn2col1 \> lgn2col2 ... \\
...
\end{tabbing}
```

L'environnement `tabbing` fournit une technique pour aligner du texte en colonnes. Il fonctionne en réglant des taquets et en tabulant jusqu'à eux à la façon dont on le faisait avec une machine à écrire ordinaire.

L'environnement `tabbing` est peu utilisé, car en général, pour formater du code informatique on utilise un environnement verbatim (voir Section 8.27 [verbatim], page 108). À noter également, pour mettre en vedette la syntaxe, les paquetages `algorithm2e` pour du pseudo-code, et `fancyvrb`, `listings` ou `minted` pour du vrai code. Et pour faire un tableau, on préfère les environnements `tabular` (voir Section 8.23 [tabular], page 97) ou `array` (voir Section 8.2 [array], page 60) pour un petit tableau qu'on met dans un flottant `table` (voir Section 8.22 [table], page 96) et l'environnement `longtable` fourni par le paquetage `longtable` pour un tableau pouvant être coupé sur plusieurs pages.

La description suivante n'est donc fournie que être complet.

Dans l'exemple suivant la première ligne sert à régler les taquets à des largeurs explicites, elle se termine par une commande `\kill` (décrite plus bas) :

```
\begin{tabbing}
\hspace{3cm}\=\hspace{2.5cm}\=\kill
Navire           \>Armement           \>Année   \\
\textit{Sophie}  \>14                 \>1800   \\
\textit{Polychrest} \>24             \>1803   \\
\textit{Lively}   \>38                 \>1804   \\
\textit{Surprise} \>28                 \>1805   \\
\end{tabbing}
```

Le résultat ressemble à cela :

Navire	Armement	Année
<i>Sophie</i>	14	1800
<i>Polychrest</i>	24	1803
<i>Lively</i>	38	1804
<i>Surprise</i>	28	1805

L'environnement `tabbing` se distingue de `tabular` notamment parce qu'avec `tabbing` l'utilisateur est obligé de régler la position des taquets, et le contenu des colonnes est toujours composé en mode LR et s'étend sur la colonne suivante s'il est plus large que la colonne courante, alors qu'avec `tabular`, l'utilisateur peut selon le spécificateur de colonne, soit composer le contenu des cellules en mode LR et laisser \LaTeX déterminer la largeur de colonne automatiquement pour que le contenu ne sorte jamais de la colonne, soit fixer la largeur et composer le contenu des cellules dans une `parbox`. Une autre différence est que `tabular` permet de faire des filets, notamment de border les colonnes par des filets verticaux, et d'avoir un espace `\tabcolsep` entre le contenu et la bordure de colonne. Une autre

distinction est que un `tabbing` peut être coupé sur plusieurs pages, alors que `tabular` génère une boîte, l’environnement `longtable` du paquetage `longtable` étant préféré à `tabbing` pour les longs tableaux.

Finalement, alors que `tabular` peut être utilisé dans n’importe quel mode, `tabbing` ne peut être utilisé qu’en mode paragraphe, et il démarre un nouveau paragraphe.

De plus, comme l’illustre l’exemple donné plus haut, il n’est pas nécessaire d’utiliser la forme étoilée de la commande `\hspace` au début d’une ligne tabulée. La marge de droite de l’environnement `tabbing` est la fin de ligne, de sorte que la largeur de l’environnement est `\linewidth`.

L’environnement `tabbing` contient une suite de *lignes tabulées*, la première ligne tabulée commence immédiatement après `\begin{tabbing}` et chaque ligne se termine par `\\` ou `\kill`. Il est possible d’omettre le `\\` pour la dernière ligne et de terminer juste par `\end{tabbing}`.

L’environnement `tabbing` a un *motif courant de taquets de tabulation* qui est une séquence de n taquets, avec $n > 0$, numérotés 0, 1, etc. Ceci crée n colonnes correspondantes. Le taquet N° 0 est toujours la marge de gauche définie par l’environnement englobant. Le taquet numéro i est réglé quand on lui assigne une position horizontale sur la page avec la commande `\=`. Le taquet numéro i ne peut être réglé que si les autres taquets 0, . . . , $i - 1$ ont déjà été réglés.

L’un des taquets du motif courant ayant été réglé comme le *taquet courant*, par défaut tout texte composé par la suite au sein de l’environnement `tabbing` est composé en drapeau droit avec le fer à gauche sur ce taquet, c’est à dire qu’il est aligné à gauche dans la colonne courante. La composition est faite en mode LR.

Par ailleurs l’environnement `tabbing` a un *taquet de début de ligne courant*. Par défaut, immédiatement après `\begin{tabbing}` c’est le taquet N° 0, et on peut le changer pour les lignes suivantes avec les commandes `\+` et `\-`. Au début de chaque ligne tabulée, le taquet courant est initialement égal au taquet de début de ligne courant. Lorsque le motif de taquets est modifié par une commande `\=`, alors le taquet de début de ligne redevient le taquet N° 0 à partir de la ligne suivante, à moins qu’une commande `\+` le change de nouveau avant le changement de ligne par `\\` ou `\kill`. La commande `\pushtabs` sauvegarde non seulement les positions de taquets, mais aussi le taquet de début de ligne, et inversement `\poptabs` le restaure.

La marge de droite de l’environnement `tabbing` est la fin de ligne, c.-à-d. que la largeur de l’environnement est de longueur `\linewidth`. Il n’y a pas par défaut de taquet correspondant à la marge de droite, même s’il serait possible avec un peu de calcul d’en définir un à cette position (voir exemples ci-après).

En ce qui concerne la marge de gauche, ou autrement dit le taquet N° 0, contrairement à l’environnement `tabular`, il n’est pas nécessaire de mettre un `\noindent` devant `\begin{tabbing}` pour éviter un renforcement : un environnement `tabbing` commence toujours un nouveau paragraphe sans renforcement. De plus il est inutile d’étoiler une commande `\hspace` en début de ligne tabulée pour que l’espace soit pris en compte.

Normalement, au sein du motif de taquets de tabulation, l’ordre des taquets correspond à des positions allant de gauche à droite. Il est cependant possible de positionner n’importe quel taquet, hormis le taquet N° 0 à n’importe quelle position en utilisant des espacements négatifs ou nuls. En réalité, le fait de ne pas pouvoir changer la position du taquet N° 0 n’est

pas restrictif, car on peut utiliser à la place comme taquet de début de ligne le taquet N° 1 grâce à la commande `\+`. Ainsi dans l'exemple ci-dessous, alors qu'on a le même contenu de l'environnement `tabbing` que dans l'exemple précédent, la première et la troisième colonne ont été interverties en changeant uniquement le motif de taquets dans la première ligne se terminant en `\kill`.

```
\begin{tabbing}
\hspace{5.5cm}\=\hspace{-2.5cm}\=\hspace{-3cm}\=\+\kill
Navire           \>Armement           \>Année     \
\textit{Sophie}  \>14                       \>1800     \
\textit{Polychrest} \>24                       \>1803     \
\textit{Lively}   \>38                       \>1804     \
\textit{Surprise} \>28                       \>1805     \
\end{tabbing}
```

Le résultat ressemble donc à cela :

Année	Armement	Navire
1800	14	<i>Sophie</i>
1803	24	<i>Polychrest</i>
1804	38	<i>Lively</i>
1805	28	<i>Surprise</i>

Les commandes suivantes peuvent être utilisées au sein d'un environnement `tabbing` :

`\ (tabbing)`

Finit une ligne tabulée et la compose. Après cette commande une nouvelle ligne tabulée commence, et donc le taquet courant redevient le taquet courant de début de ligne.

`\= (tabbing)`

Règle le prochain taquet à la position courante, et fait de lui le taquet courant. Si le taquet courant est le taquet N° n , alors une commande `\=` règle la position du taquet N° $n + 1$ et fait de lui le taquet courant. Si on avait déjà au moins $n + 2$ taquets, ceci redéfinit la position du taquet N° $n + 1$. Sinon, le nombre de taquets dans le motif courant était à ce moment $(n + 1)$, c'est à dire que dernier des taquets en numéro d'ordre était le N° n , alors la commande `\=` en définissant le taquet N° $n + 1$, porte le nombre de taquets de $n + 1$ à $n + 2$.

La position des taquets peut être définie avec plus de liberté que ce qui se passait avec un machine à écrire ordinaire. En effet il est possible de définir à la même position deux taquets de N° n et N° m distincts, voire de faire en sorte que les taquets N° n et N° m , avec $n < m$, soient à des positions telles que le taquet N° m soit avant le taquet N° n .

`\> (tabbing)`

Change le taquet courant au taquet suivant dans l'ordre logique, c.-à-d. si le taquet courant est le taquet N° n , alors, en supposant qu'il y ait au moins $n + 2$ taquets définis, on passe au taquet N° $n + 1$, sinon cette commande provoque une erreur.

`\<`

Change le taquet courant au taquet précédent dans l'ordre logique. Cette commande ne peut être utilisée qu'au début d'une ligne tabulée, et que lorsque

le taquet de début de ligne courant est le taquet N° n avec $n > 0$. Le taquet courant devient alors le taquet N° $n - 1$.

Il est en fait possible d'avoir m commandes `\<` consécutives à condition que $m \leq n$, le taquet courant devient alors le taquet N° $(n - m)$. Si par contre $m > n$, alors la $n + 1$ commande `\<` provoque une erreur.

Il est possible, même si c'est absurde, d'avoir ensuite des commandes `\>` qui défont ce qu'on fait les commandes `\<`. Ainsi en supposant qu'au début d'une ligne tabulée le taquet de début de ligne courant soit le taquet N° 2, faire commencer la ligne par `\<\<\>\>` n'a aucun effet. Par contre, `\<\<\<\>\>\>` provoquera une erreur, parce que la troisième commande `\<` est illicite. Dans les mêmes conditions faire `\<\>\<\>` provoque également une erreur, car la seconde commande `\<` n'est pas consécutive de la première, elle n'est donc pas, au sens où on l'entend, utilisée en début de ligne.

`\+` Change le taquet de début de ligne courant au taquet suivant dans l'ordre logique pour la ligne tabulée suivante, et toutes celles qui suivent tant qu'une autre commande ne vient pas le modifier. La notion de taquet suivant dans l'ordre logique suit le même principe que pour la commande `\>`, c'est à dire que s'il n'y a que $n + 1$ taquets définis, et que le taquet de début de ligne courant est le taquet N° n , alors cette commande provoque une erreur. La commande `\+` peut être invoquée n'importe où dans la ligne tabulée, et on peut avoir plusieurs commandes `\+` au sein de la même ligne, du moment que pour chacune d'elle un nombre suffisant de taquets est déjà défini. Par exemple le code suivant provoque une erreur parce que bien qu'à la fin de la première ligne on ait trois taquets définis, au moment où la commande `\+` est invoquée le nombre de taquets définis ne vaut que 1, on ne peut donc pas passer du taquet N° 0 au taquet N° 1 :

```
\begin{tabbing}
Vive \+\=1a \=France\\
Vive \>1a\>République\\
\end{tabbing}
```

En revanche le code suivant fonctionne, car lorsque la commande `\+` est passée on a trois taquets définis, les taquets N° 0 à 2, on peut donc passer du taquet N° 0 au taquet N° 1.

```
\begin{tabbing}
Vive \=1a \=France\+\\
Vive \>1a\>République\\
\end{tabbing}
```

C'est pourquoi l'usage est que la ou les commandes `\+` sont toujours invoquées juste avant la fin de ligne `\\` ou `\kill`.

Il est possible, même si cela est absurde, d'avoir sur la même ligne à la fois des commandes `\+` et `\-` s'annulant l'une l'autre, du moment qu'aucune d'elle ne fasse passer sur un taquet non défini.

`\-` Change le taquet de début de ligne courant au taquet précédent dans l'ordre logique pour la ligne tabulée suivante, et toutes celles qui suivent tant qu'une autre commande ne vient pas le modifier. C'est à dire que si le taquet de début de ligne courant est le taquet N° n , et qu'on a m commandes `\-` avec $m \geq n$,

alors le taquet de début de ligne courant de la ligne suivante devient le taquet N° $n - m$. Par contre, si $m > n$, alors la $(n + 1)$ ième commande `\+` provoque une erreur.

`\'` (`tabbing`)

Déplace tout ce que vous avez tapé jusqu'alors dans la colonne courante, c.-à-d. tout ce qui suit la plus récente commande `\>`, `\<`, `\'`, `\\`, ou `\kill`, aligné à droite dans la colonne précédente, le fer à droite étant espacé du taquet courant d'une longueur `\tabbingsep`. Voir aussi `\'`.

`\'` (`tabbing`)

Vous permet de placer du texte justifié à droite en face de n'importe quel taquet, y compris le taquet 0. Les commandes `\'` et `\'` alignent toutes deux le texte à droite, toutefois elles se distinguent fortement par deux aspects :

- Tout d'abord le texte qui est déplacé est, dans le cas de `\'`, le texte qui suit la commande `\'` et qui va jusqu'à la fin de la ligne tabulée, c.-à-d. jusqu'à `\\` ou `\end{tabbing}`. Il ne doit y avoir aucune commande `\>` ou `\'` entre le `\'` et la commande `\\` ou `\end{tabbing}` qui termine la ligne. Dans le cas de la commande `\'` au contraire, c'est le texte qui précède la commande `\'`.
- Par ailleurs, avec `\'` l'alignement se fait sur un taquet et avec un espacement relatif de `\tabbingsep`, alors qu'avec `\'` l'alignement se fait sur la marge de droite et sans espacement relatif.

`\a` (`tabbing`)

Dans un environnement `tabbing`, les commandes `\=`, `\'` et `\'` ne produisent pas d'accents comme d'habitude (voir Section 23.5 [Accents], page 233). À leur place, on utilise les commandes `\a=`, `\a'` et `\a'`.

`\kill`

Règles les taquets sans produire de texte. Fonctionne tout comme `\\` à ceci près que la ligne courante est jetée au lieu de produire une sortie. L'effet de toute commande `\=`, `\+` ou `\-` dans cette ligne demeure en vigueur.

`\poptabs`

Restaure les positions de taquets et le taquet de début de ligne sauvegardés par le dernier `\pushtabs`.

`\pushtabs`

Sauvegarde dans une pile dédiée à cet effet le motif courant de taquets de tabulation ainsi que le taquet de début de ligne. C'est utile pour changer temporairement les positions de taquets au milieu d'un environnement `tabbing`. Après un `\pushtabs`, on a un nouveau motif vierge, c.-à-d. ne contenant qu'un seul taquet, le taquet N° 0 à la position de début de ligne. Pour toute commande `\pushtabs` il doit y avoir une commande `\poptabs` avant la fin de l'environnement, c'est à dire que la pile doit être explicitement vidée avant le `\end{tabbing}`.

`\tabbingsep`

Distance du texte déplacé par `\'` à la gauche du taquet courant.

Il est à noter que les espaces en début de ligne, où après `\=`, `\>` ou `\<` sont gobés, qu'il s'agisse d'espaces produits par la macro `\space` ou par le caractère espace. Par contre

Le résultat ressemble à cela (la police utilisée dans ce document pour les exemples n'est pas ici la mieux adaptée) :

souligné barré

Finalement, voici un exemple où on a placé le taquet N° 5 décalé d'une longueur `\tabbingsep` à droite de la marge de droite.

```
{\ttfamily
\begin{tabbing}
\hspace{1cm}\=\hspace{1cm}\=\hspace{1cm}\=\hspace{1cm}\=
\hspace{\dimexpr\linewidth-4cm+\tabbingsep}\=\kill
1\>2\>X\3\>4\>5\>\a'a gauche de de la marge de droite
avec \textbackslash'\'\
1\>2\>X\3\>4\>5\'\a'a gauche de de la marge de droite
avec \textbackslash'\
\end{tabbing}}
```

Le résultat ressemble à cela :

1	2	X	3	4	5	à gauche de la marge de droite avec \'
1	2	X	3	4	5	à gauche de la marge de droite avec \'

8.22 table

Synopsis :

```
\begin{table}[placement]
  corps-du-tableau
  \caption[titreldt]{titre} % optionnel
  \label{étiquette} % également optionnel
\end{table}
```

Une classe de flottants (voir Section 5.7 [Floats], page 37). Parce qu'ils ne peuvent pas être coupés à cheval sur plusieurs pages, il ne sont pas composés en séquence avec le texte normale, mais à la place sont « flottés » vers un endroit convenable, tel que le sommet d'une page suivante.

Dans l'exemple suivant, l'environnement `table` contient un `tabular` :

```
\begin{table}
\centering\small
\begin{tabular}{ll}
\multicolumn{1}{c}{\textit{Auteur}}
&\multicolumn{1}{c}{\textit{Pièce}} \\ \hline
Bach & Suite Numéro 1 pour violoncelle \\
Beethoven & Sonate Numéro 3 pour violoncelle \\
Brahms & Sonate Numéro 1 pour violoncelle
\end{tabular}
\caption{Meilleures pièces pour violoncelle}
\label{tab:violoncelle}
\end{table}
```

mais vous pouvez placer beaucoup de types différents de contenu au sein d'une `table` : la `table body` peut contenir du texte, des commandes L^AT_EX, des graphiques, etc. Il est composé dans une `parbox` de largeur `\textwidth`.

Pour les valeurs possibles de *placement* et leur effet sur l'algorithme de placement des flottants, voir Section 5.7 [Floats], page 37.

L'étiquette est optionnelle ; elle est utilisée pour les doubles renvois (voir Chapitre 7 [Cross references], page 55). La commande `\caption` est également optionnelle. Elle spécifie la légende *titre* pour le tableau (voir Section 5.7.1 [`\caption`], page 40). La légende est numérotée par défaut. Si *titrelat* est présent, il est utilisé dans la liste des tableaux au lieu de *titre* (voir Section 25.1 [Table of contents etc.], page 242).

Dans l'exemple suivant le tableau et sa légende flottent jusqu'au bas d'une page, à moins qu'il soit repoussé jusqu'à une page de flottants à la fin.

```
\begin{table}[b]
  \centering
  \begin{tabular}{r|p{2in}} \hline
    Un &Le plus solitaire des nombres \\
    Deux &Peut être aussi triste que un.
      C'est le nombre le plus solitaire après le nombre un.
  \end{tabular}
  \caption{Vertus cardinales}
  \label{tab:VertusCardinales}
\end{table}
```

8.23 tabular

Synopsis :

```
\begin{tabular}[pos]{cols}
  article colonne 1 & article colonne 2 ... & article colonne n \\
  ...
\end{tabular}
```

ou

```
\begin{tabular*}{largeur}[pos]{cols}
  article colonne 1 & article colonne 2 ... & article colonne n \\
  ...
\end{tabular*}
```

Ces environnements produisent une boîte consistant d'une séquence de lignes horizontales. Chaque ligne consiste en des articles qui sont alignés verticalement au sein de colonnes. Ci-après une illustration de beaucoup des caractéristiques.

```
\begin{tabular}{l|l}
  \textit{Nom du joueur} & \textit{Coups de circuit de toute sa carrière} \\ \hline
  Hank Aaron & 755 \\
  Babe Ruth & 714
\end{tabular}
```

La sortie a deux colonnes alignées à gauche, avec une barre verticale entre elles. Ceci est spécifié par l'argument `{1|1}` de `tabular`. On place les articles dans des colonnes différentes en les séparant avec une esperluette `&`. La fin de chaque ligne est marquée avec une double controbligue, `\\`. On met un filet horizontal sous une ligne en mettant `\hline` après sa double controbligue. Cette `\\` est optionnelle après la dernière ligne, à moins qu'une commande `\hline` suive, pour mettre un filet sous le tableau.

Les arguments obligatoires et optionnels de `tabular` consistent en :

pos Optionnel. Spécifie la position verticale du tableau. La valeur par défaut est d'aligner le tableau de sorte à ce que son centre vertical s'accorde à la ligne de base du texte autour. Il y a deux autres alignements possibles : `t` aligne le tableau de sorte que la première ligne s'accorde à la ligne de base du texte autour, et `b` fait la même chose pour la dernière ligne du tableau. Ceci n'a un effet que si il y a du texte autour de la table. Dans le cas usuel d'un `tabular` seul au sein d'un environnement `center` cette option ne fait aucune différence.

cols Obligatoire. Spécifie le formatage des colonnes. Il consiste en une séquence des spécificateurs suivants, correspondant à la séquence des colonnes et de la matière inter-colonne.

- `l` Une colonne d'articles alignés à gauche.
- `r` Une colonne d'article alignés à droite.
- `c` Une colonne d'article centrés.
- `|` Une ligne verticale s'étendant complètement sur la hauteur et profondeur de l'environnement.

`@{texte ou espace}`

Ceci insère *texte ou espace* à cette position dans chaque ligne. La matière *texte ou espace* est composée en mode LR. Ce texte est fragile (voir Section 12.11 [`\protect`], page 134).

Si entre deux spécificateurs de colonne il n'y pas d'expression en `'@{...}'` alors les classes `book`, `article` et `report` de L^AT_EX mettent de chaque côté de chaque colonne une gouttière (espace blanc vertical) de largeur `\tabcolsep`, qui par défaut vaut 6 pt. C'est à dire que par défaut des colonnes adjacentes sont séparées de 12 pt (ainsi `\tabcolsep` est un nom trompeur puisque ce n'est pas la séparation entre des colonnes de tableau). De plus, un espace de `\tabcolsep` vient aussi avant la première colonne ainsi qu'après la dernière colonne, à moins que vous placiez un `'@{...}'` à cet endroit.

Si vous l'emportez sur la valeur par défaut et utilisez une expression en `'@{...}'` alors L^AT_EX n'insère pas `\tabcolsep` et vous devez insérer tout espace désiré vous-même, comme dans `@{\hspace{1em}}`.

Une expression vide `@{}` élimine l'espace, y compris l'espace au début ou à la fin, comme dans l'exemple ci-dessous où on veut que les lignes du tableau soient alignées sur la marge de gauche.

```

\begin{flushleft}
  \begin{tabular}{@{}l}
    ..
  \end{tabular}
\end{flushleft}

```

L'exemple suivant montre du texte, une virgule de séparation décimale entre les colonnes, arrangé de sorte que les nombres dans la table soient alignés sur cette virgule.

```

\begin{tabular}{r@{,$,$}l}
  $3$ & $14$ \\
  $9$ & $80665$
\end{tabular}

```

Une commande `\extracolsep{lrg}` au sein d'une expression en '@{...}' a pour effet qu'une espace supplémentaire de largeur *lrg* apparaît sur la gauche de toutes les colonnes suivantes, jusqu'à contre-ordre d'une autre commande `\extracolsep`. Contrairement aux à espace inter-colonnes ordinaires, cette espace supplémentaire n'est pas supprimé par une expression en '@{...}'. Une commande `\extracolsep` peut uniquement être utilisée au sein d'une expression en '@{...}' elle-même au sein de l'argument `cols`. Ci-dessous, L^AT_EX insère la quantité convenable d'espace intercolonne our que le tableau fasse 10 cm de large.

```

\begin{center}
  \begin{tabular*}{10cm}{l@{\extracolsep{\fill}}l}
    Tomber sept fois, se relever huit \ldots
    & c'est comme ça la vie !
  \end{tabular*}
\end{center}

```

Pour insérer des commandes qui sont automatiquement exécutées avant une colonne donnée, charger le paquetage `array` et utilisez le spécificateur `>{...}`.

`p{lrg}` Chaque cellule de la colonne est composée au sein d'une parbox de largeur *lrg*, comme si c'était l'argument d'une commande `\parbox[t]{lrg}{...}`.

Les sauts de ligne de type double-controblique `\` ne sont pas permis au sein d'une telle cellule, sauf à l'intérieur d'un environnement comme `minipage`, `array`, ou `tabular`, ou à l'intérieur d'une `\parbox` explicite, ou dans la portée d'une déclaration `\centering`, `\raggedright`, ou `\raggedleft` (quand elles sont utilisées au sein d'une cellule de colonne en `p` ces déclarations doivent apparaître entre accolades, comme dans `{\centering .. \}`). Sans quoi L^AT_EX prendra la double controblique comme une fin de ligne de tableau. Au lieu de cela, pour obtenir un saut de ligne dans une cellule de ce type utilisez `\newline` (voir Section 9.3 [`\newline`], page 113).

`*{num}{cols}`

Équivalent à *num* copies de *cols*, où *num* est un entier positif et *cols* est toute liste de spécificateurs. Ainsi `\begin{tabular}{|*{3}{l|r}|}` équivaut à `\begin{tabular}{|l|r|l|r|}`. Notez que *cols* peut contenir une autre expression en `*{...}{...}`.

largeur Obligatoire pour `tabular*`, non autorisé pour `tabular`. Spécifie la largeur de l'environnement `tabular*`. Il doit y avoir de l'espace élastique entre les colonnes, comme avec `with @{\extracolsep{\fill}}`, de sorte à permettre au tableau de se dilater ou contracter pour faire la largeur spécifiée, sans quoi vous aurez probablement l'avertissement `Underfull \hbox (badness 10000) in alignment ...`

Paramètres qui contrôlent le formatage :

`\arrayrulewidth`

Une longueur qui est l'épaisseur du filet créé par `|`, `\hline`, et `\vline` au sein des environnements `tabular` et `array`. La valeur par défaut est `'0.4pt'`. On peut la changer comme dans `\setlength{\arrayrulewidth}{0.8pt}`.

`\arraystretch`

Un facteur par lequel l'espacement entre les lignes au sein des environnements `tabular` et `array` est multiplié. La valeur par défaut est `'1'` pour aucune dilatation/contraction. On peut la changer comme dans `\renewcommand{\arraystretch}{1.2}`.

`\doublerulesep`

Une longueur qui est la distance entre les filets verticaux produit par le spécificateur `||`. La valeur par défaut est `'2pt'`.

`\tabcolsep`

Une longueur qui est la moitié de l'espace entre les colonnes. La valeur par défaut est `'6pt'`. On peut la changer avec `\setlength`

Les commandes suivantes peuvent être utilisées à l'intérieur du corps d'un environnement `tabular`, les deux premières au sein d'un article, et les deux suivantes entre les lignes :

8.23.1 `\multicolumn`

Synopsis :

```
\multicolumn{nbrecols}{patron}{texte}
```

Fabrique une entrée de `array` ou `tabular` fusionnée sur plusieurs colonnes. Le premier argument, *nbrecols* spécifie le nombre de colonnes sur lesquelles s'étend la fusion. Le second argument, *patron*, est obligatoire et spécifie le format de l'entrée ; notamment `c` pour centré, `l` pour aligné à gauche, `r` pour aligné à droite. Le troisième argument, *texte* spécifie le texte à mettre dans l'entrée.

Dans l'exemple suivant les trois première colonnes sont fusionnées en le titre unique 'Nom'.

```
\begin{tabular}{lcc|l}
  \textit{ID} & & & \multicolumn{2}{c}{\textit{Nom}} & \textit{Âge} \\ \hline % ligne 1
```

```
978-0-393-03701-2 &O'Brian &Patrick
```

```
&55
```

```
\\ % lign
```

```
...
```

```
\end{tabular}
```

On compte comme colonne chaque partie du patron *patron* de l'environnement `array` ou `tabular` qui, à l'exception de la première, commence par `l`, `c`, `r`, ou `p`. Ainsi à partir de `\begin{tabular}{|r|ccp{4cm}|}` les parties sont `|r|`, `c`, `c`, et `p{1.5in}|`.

L'argument *patron* l'emporte sur la spécification par défaut de zone inter-colonne de l'environnement `array` ou `tabular` contiguë à cette entrée multi-colonne. Pour affecter cette zone, cet argument peut contenir des barres verticales `|` indiquant le placement de filets verticaux, et d'expression `@{...}`. Ainsi si *patron* est `'|c|'` alors l'entrée multi-colonne est centrée et un filet vertical la précède et la suit dans les espaces inter-colonne de part et d'autre. Ce tableau nous permet de décrire en détail le comportement exact.

```
\begin{tabular}{|cc|c|c|}
  \multicolumn{1}{r}{w}      % entrée un
  &\multicolumn{1}{|r|}{x}    % entrée deux
  &\multicolumn{1}{|r}{y}    % entrée trois
  &z                          % entrée quatre
\end{tabular}
```

Avant la première entrée il n'y a pas de filet verticale en sortie parce que le `\multicolumn` a le spécificateur `'r'` dans *patron* sans barre verticale initiale. Entre les entrées un et deux il y a un filet vertical ; en effet bien que le premier *patron* ne se termine pas par une barre verticale, le second *patron* en a une à son commencement. Entre la deuxième et la troisième entrée il y a un seul filet vertical ; bien que le *patron* dans les deux `\multicolumn` correspondant demandent un filet vertical, vous n'en obtenez qu'un seul. Entre les entrées trois et quatre il n'y a aucun filet vertical ; la spécification par défaut en demande un mais le *patron* dans le `\multicolumn` de la troisième entrée n'en demande pas, et cela l'emporte. Finalement, à la suite de la quatrième entrée il y a un filet vertical à cause de la spécification par défaut.

Le nombre de colonnes fusionnées *nbrecols* peut être 1. En plus de donner la possibilité de changer l'alignement horizontal, ceci est aussi utile pour l'emporter pour l'une des lignes sur la définition `tabular` par défaut de la zone inter-colonne, y compris concernant le placement des filets verticaux.

Dans l'exemple ci-dessous, dans la définition `tabular` de la première colonne on spécifie par défaut de justifier, mais au sein de la première ligne l'entrée est centrée avec `\multicolumn{1}{c}{\textsc{Période}}`. De même au sein de la première ligne, les deuxième et troisième colonnes sont fusionnées entre elle avec `\multicolumn{2}{c}{\textsc{Intervalle}}`, l'emportant sur la spécification pour centrer ces deux colonnes sur le trait d'union au sein de la plage de dates.

```
\begin{tabular}{l|r@{-}l}
  \multicolumn{1}{c}{\textsc{Période}}
  &\multicolumn{2}{c}{\textsc{Dates}} \\ \hline
  Baroque          &1600          &1760          \\
  Classique        &1730          &1820          \\
  Romantique       &1780          &1910          \\
  Impressioniste   &1875          &1925          \end{tabular}
```

```
\end{tabular}
```

Notez que bien que la spécification `tabular` par défaut place un filet vertical entre la première et la deuxième colonne, du fait qu'il n'y pas pas de barre verticale dans aucun des patrons *patron* des commandes `\multicolumn` des deux premières colonne, alors aucun filet n'apparaît sur la première ligne.

8.23.2 `\vline`

La commande `\vline` dessine une ligne verticale s'étendant sur la totalité de la hauteur et de la profondeur de sa ligne. Une commande `\hfill` peut être utilisée pour déplacer cette ligne verticale vers le bord de la colonne. La commande `\vline` peut aussi être utilisée dans une expression `@{...}`.

8.23.3 `\cline`

Synopsis :

```
\cline{i-j}
```

La commande `\cline` dessine des lignes horizontales à travers les colonnes spécifiées en arguments obligatoires, en commençant dans la *i* et finissant dans la colonne *j*.

8.23.4 `\hline`

La commande `\hline` dessine une ligne horizontale de la largeur de l'environnement `tabular` ou `array` susjacent. Elle est la plupart du temps utilisée pour dessiner une ligne au sommet, bas, et entre les lignes d'un tableau.

8.24 `thebibliography`

Synopsis :

```
\begin{thebibliography}{étiquette-la-plus-large}
  \bibitem[étiquette]{clef_de_cite}
  ...
\end{thebibliography}
```

L'environnement `thebibliography` produit une bibliographie ou une liste de références. Il y a deux façons de produire des listes bibliographiques. Cet environnement convient quand vous avez un petit nombre de références et pouvez maintenir la liste manuellement. Voir Section 8.24.4 [Using BibTeX], page 105, pour une approche plus sophistiquée.

L'exemple ci-dessous illustre l'environnement avec deux articles.

```
Ce travail s'appuie sur \cite{latexdps}.
Et ensemble à \cite{latexdps, texbook}.
...
\begin{thebibliography}{9}
  \bibitem{latexdps}
    Leslie Lamport.
    \textit{\LaTeX{}}: a document preparation system.
    Addison-Wesley, Reading, Massachusetts, 1993.
  \bibitem{texbook}
    Donald Ervin Knuth.
```

```
\textit{Le Texbook : Composition informatique}.
Vuibert, 1er août 2017.\end{thebibliography}
```

Cela formate la première référence comme ‘[1] Leslie ...’, de sorte que ... s’appuie sur `\cite{latexdps}` produit ‘... s’appuie sur [1]’. Le second `\cite` produit ‘[1, 2]’. On doit compiler le document deux fois pour que ces numéros de référence soient résolus.

L’argument obligatoire *étiquette-la-plus-large* est un texte qui, lorsqu’il est composé, est aussi large que la plus large des étiquettes d’article produit par les commandes `\bibitem`. Traditionnellement on utilise 9 pour des bibliographies de moins de 10 références, 99 pour celles qui en ont moins de 100, etc.

La liste bibliographique a un titre de bibliographie tel que ‘Bibliographie’. Pour le changer il y a deux cas. Dans les classes `book` et `report`, lorsque le niveau de rubricage le plus élevé est `\chapter` et le titre par défaut ‘Bibliographie’, ce titre est dans la macro `\bibname`. Pour `article`, lorsque le niveau de rubricage le plus élevé est `\section` et que le titre par défaut est ‘Références’, le titre est stocké dans la macro `\refname`. Pour le changer redéfinissez la commande, comme dans `\renewcommand{\refname}{Références citées}`, après `\begin{document}`.

Les paquetages de prise en charge linguistique comme `babel` redéfinissent automatiquement `\refname` ou `\bibname` pour s’adapter à la langue sélectionnée.

Voir Section 8.16 [list], page 72, pour les paramètres de contrôle de disposition de la liste.

8.24.1 `\bibitem`

Synopsis :

```
\bibitem{clef_de_cite}
```

ou :

```
\bibitem[étiquette]{clef_de_cite}
```

La commande `\bibitem` génère un article étiqueté par défaut par un numéro généré en utilisant le compteur `enumi`. L’argument *clef_de_cite* est une *clef de référence bibliographique* consistant en une chaîne quelconque de lettres, chiffres, et signes de ponctuation (hormis la virgule).

Voir Section 8.24 [thebibliography], page 102, pour un exemple.

L’argument optionnel *étiquette*, lorsqu’il est présent, sert d’étiquette et le compteur `enumi` n’est pas incrémenté. Dans l’exemple suivant :

```
\begin{thebibliography}
\bibitem[Lampport 1993]{latexdps}
  Leslie Lampport.
  \textit{\LaTeX{}: a document preparation system}.
  Addison-Wesley, Reading, Massachusetts, 1993.
\bibitem{texbook}
  Donald Ervin Knuth.
  \textit{Le Texbook : Composition informatique}.
  Vuibert, 1er août 2017.
\end{thebibliography}
```

le premier article est formaté comme ‘[Lampport 1993] Leslie ...’ (la quantité d’espace horizontal que L^AT_EX laisse pour l’étiquette dépend de l’argument *étiquette-la-plus-large*

de l’environnement `thebibliography` ; voir Section 8.24 [thebibliography], page 102). De même, ... s’appuyant sur `\cite{latexdps}` produit ‘... s’appuyant sur [Lampport 1994]’.

Si vous mélangez des articles `\bibitem` avec et sans *étiquette*, alors L^AT_EX numérote ces derniers séquentiellement. Dans l’exemple ci-dessus l’article `texbook` apparaît comme ‘[1 Donald ...]’, bien que ce soit le deuxième article.

Si vous utilisez la même `clef_de_cite` deux fois, alors vous obtenez ‘LaTeX Warning: There were multiply-defined labels’.

En coulisse, L^AT_EX se souvient de l’information `clef_de_cite` et *étiquette* parce que `\bibitem` écrit dans le fichier auxiliaire `jobname.aux` (voir Section 28.3 [Jobname], page 270). Par exemple dans l’exemple ci-dessus ce fichier contient `\bibcite{latexdps}{Lampport, 1993}` et `\bibcite{texbook}{1}`. Le fichier `.aux` est lu par la commande `\begin{document}` et alors l’information devient disponible pour les commandes `\cite`. Ceci explique qu’il est nécessaire d’exécuter L^AT_EX deux fois pour que les références soient résolues : une pour l’écrire, et une pour le relire.

À cause de cet algorithme en deux passes, quand vous ajoutez un `\bibitem` ou changez sa `clef_de_cite` vous pouvez obtenir ‘LaTeX Warning: Label(s) may have changed. Rerun to get cross-references right’. La solution est de recompiler.

8.24.2 `\cite`

Synopsis :

```
\cite{clefs}
```

ou

```
\cite[subcite]{clefs}
```

Cette commande génère en sortie une citation des références associées à `clefs`. L’argument `clefs` est une liste d’une ou plus de clefs de références bibliographiques (voir Section 8.24.1 [bibitem], page 103), séparées par des virgules.

L’exemple suivant :

```
La source ultime est \cite{texbook}.
...
\begin{thebibliography}
\bibitem{texbook}
  Donald Ervin Knuth.
  \textit{Le Texbook : Composition informatique}.
  Vuibert, 1er août 2017.
\end{thebibliography}
```

produit une sortie du genre ‘... ultime est [1]’. (Vous pouvez changer l’apparence de la citation et de la référence en utilisant un style de bibliographie si vous générez automatiquement l’environnement `thebibliography`. Voir Section 8.24.4 [Using BibTeX], page 105, pour plus ample information).

L’argument optionnel `subcite` est joint à la citation. Par exemple, Voir 14.3 dans `\cite[p.~314]{texbook}` pourrait produire ‘Voir 14.3 dans [1, p. 314]’.

En plus de ce qui apparaît en sortie, `\cite` écrit de l’information dans le fichier auxiliaire `jobname.aux` (voir Section 28.3 [Jobname], page 270). Par exemple, `\cite{latexdps}` écrit

‘`\citation{latexdps}`’ dans ce fichier. Cette information est utilisée par BibTeX pour inclure dans vos listes de référence seulement les ouvrages que vous avez effectivement cités ; voir aussi Section 8.24.3 [`\nocite`], page 105.

Si *clefs* n’est pas dans votre information de bibliographie, alors vous obtenez ‘**LaTeX Warning: There were undefined references**’, et en sortie la citation s’affiche comme un point d’interrogation en gras entre crochets. Il y a deux causes possibles. Si vous avez fait une faute de frappe, comme dans `\cite{texbok}` alors il n’y a qu’à la corriger. Si par contre vous venez juste d’ajouter ou de modifier l’information bibliographique de sorte que le fichier `.aux` est modifié (voir Section 8.24.1 [`\bibitem`], page 103) alors la solution est d’exécuter de nouveau L^AT_EX.

8.24.3 `\nocite`

Synopsis :

```
\nocite{clefs}
```

La commande `\nocite` ne produit aucun texte en sortie, mais écrit *clefs* dans le fichier auxiliaire *jobname.aux* (voir Section 28.3 [`Jobname`], page 270).

L’argument obligatoire *clefs* est une liste d’une ou plusieurs clefs de référence bibliographique (voir Section 8.24.1 [`\bibitem`], page 103). Cette information est utilisée par BibTeX pour inclure ces ouvrages dans votre liste de références même si vous ne les avez pas cités explicitement (voir Section 8.24.2 [`\cite`], page 104).

8.24.4 Utilisation de BibTeX

Comme décrit dans `thebibliography` (voir Section 8.24 [`thebibliography`], page 102), une approche sophistiqué pour gérer les bibliographies est fournie par le programme BibTeX. Ceci est uniquement une introduction ; voir la documentation complète sur le CTAN (voir Section 2.6 [`CTAN`], page 6).

Avec BibTeX vous n’utilisez pas l’environnement `thebibliography` directement (voir Section 8.24 [`thebibliography`], page 102). Au lieu de cela, vous incluez les lignes :

```
\bibliographystyle{stylebib}
\bibliography{ficbib1,ficbib2, ...}
```

L’argument *stylebib* fait référence à un fichier *stylebib.bst* définissant l’aspect que doivent avoir vos référence bibliographique. Les noms *stylebib* standards distribués avec BibTeX sont :

alpha	Les étiquettes sont formées à partir du nom de l’auteur et de l’année de publication. Les articles bibliographiques sont classés alphabétiquement.
plain	Les étiquettes sont des numéros. Les articles sont classés alphabétiquement.
unsrt	Comme plain , mais les entrées sont dans l’ordre où les références y sont faites dans le texte.
abbrv	Comme plain , mais les étiquettes sont plus compactes.

De très nombreux autres styles BibTeX existent, taillés sur mesure pour satisfaire les exigences de diverses publications. Voir le sujet <https://ctan.org/topic/bibtex-sty> sur le CTAN.

La commande `\bibliography` est ce qui produit pour de bon la bibliographie. L'argument de `\bibliography` fait référence aux fichiers nommés `ficbib1.bib`, `ficbib2.bib`, ..., qui sont censés contenir votre base de données au format BibTeX. L'exemple suivant illustre une paire typique d'entrées dans ce format.

```
@book{texbook,
  title      = {Le {\TeX}book~: composition informatique},
  author     = {D.E. Knuth},
  isbn      = {2-7117-4819-7},
  series     = {Vuibert informatique},
  year      = {2003},
  publisher  = {Vuibert}
}
@book{sexbook,
  author     = {W.H. Masters and V.E. Johnson},
  title      = {Les r{\e}action sexuelles},
  year      = {1976},
  publisher  = {Robert Laffont}
}
```

Seules les entrées désignées par les commandes `\cite` ou `\nocite` sont listées dans la bibliographie du document. De la sorte, vous pouvez conserver toutes vos sources ensemble dans un seul fichier, ou un petit nombre de fichiers, et compter sur BibTeX pour inclure dans le document seulement celles que vous avez utilisées.

Avec BibTeX, l'argument *clefs* de `\nocite` peut aussi être le caractère '*' tout seul. Ceci veut dire de citer implicitement tous les articles de toutes les bibliographies données.

8.24.4.1 Messages d'erreur de BibTeX

Si vous oubliez d'utiliser `\bibliography` ou `\bibliographystyle` dans votre document (ou, moins vraisemblablement, faites un document sans aucune commande `\cite` ni `\nocite`), alors BibTeX produit un message d'erreur. Parce que BibTeX est utilisable avec n'importe quel programme, et pas seulement L^AT_EX, les messages d'erreur font référence aux commandes internes lues par BibTeX (du fichier `.aux`), plutôt qu'aux commandes de niveau utilisateur décrites plus haut.

Le tableau ci-après montre les commandes internes mentionnées dans les erreurs BibTeX, et les commandes de niveau utilisateur correspondantes.

```
\bibdata  \bibliography
\bibstyle  \bibliographystyle
\citation  \cite, \nocite
```

Par exemple, si votre document n'a aucune commande `\bibliographystyle`, BibTeX s'en plaint ainsi :

```
I found no \bibstyle command---while reading file document.aux
```

8.25 theorem

Synopsis :

```
\begin{theorem}
  corps du théorème
\end{theorem}
```

L'environnement `theorem` produit « Théorème n » en gras suivi de *corps du théorème*, où les possibilités de numérotation pour n sont décrites par `\newtheorem` (voir Section 12.9 [`\newtheorem`], page 132).

```
\newtheorem{lem}{Lemme}      % dans le préambule
\newtheorem{thm}{Théorème}
...
\begin{lem}                  % dans le corps du document
  texte du lemme
\end{lem}
```

Le résultat suivant en découle immédiatement.

```
\begin{thm}[Gauss] % place « Gauss » entre parenthèses après le titre
  texte du théorème
\end{thm}
```

La plupart des nouveaux documents utilisent les paquetages `amsthm` et `amsmath` de l'American Mathematical Society. Entre autres ces paquetages incluent un grand nombre d'options pour les environnements de théorèmes, telles que des options de formatage.

8.26 titlepage

Synopsis :

```
\begin{titlepage}
  ... texte et espacement ...
\end{titlepage}
```

L'environnement `titlepage` crée une page de titre, c'est à dire une page sur laquelle ne sont inscrit aucun numéro de page ou libellé de rubrique. Il entraîne aussi que la page suivante a le numéro un.

Dans l'exemple suivant tout le formatage, y compris l'espacement vertical, est laissé à l'auteur.

```
\begin{titlepage}
\vspace*{\stretch{1}}
\begin{center}
  {\huge\bfseries Thèse \\\[1ex]
    titre} \\\[6.5ex]
  {\large\bfseries Nom de l'auteur} \\\
  \vspace{4ex}
  Thèse soumise à \\\[5pt]
  \textit{Nom de l'université} \\\[2cm]
  pour obtenir le grade de \\\[2cm]
  \textsc{\Large Docteur en} \\\[2ex]
```

```

\textsc{\large Mathématiques}          \\[12ex]
\vfll
Departement de Mathématiques          \[
Adresse                                \[
\vfll
\today
\end{center}
\vspace{\stretch{2}}
\end{titlepage}

```

Pour produire par contre une page de titre standard sans utiliser l'environnement `titlepage`, utilisez la commande `\maketitle` (voir Section 18.1 [`\maketitle`], page 187).

8.27 verbatim

Synopsis :

```

\begin{verbatim}
texte-litéral
\end{verbatim}

```

L'environnement `verbatim` est un environnement qui fabrique des paragraphes dans lequel \LaTeX produit exactement ce qui vous y tapez. Par exemple au sein de *texte-litéral* le caractère `\` ne commence pas de commande, il produit '`\`' à l'impression, et les retours chariot et les blancs sont pris littéralement. La sortie apparaît dans une police genre tapuscrit (encore appelé « machine à écrire ») à chasse fixe (`\tt`).

```

\begin{verbatim}
Juron symbolique : %&$#!.
\end{verbatim}

```

La seule restriction sur `text-litérale` est qu'il ne peut pas inclure la chaîne `\end{verbatim}`.

Il est impossible d'utiliser l'environnement `verbatim` au sein d'un argument de macro, par exemple dans l'argument passé à `\section`. Ce n'est pas le même problème que celui des commandes fragiles (voir Section 12.11 [`\protect`], page 134), mais c'est juste impossible parce que l'environnement `verbatim` change le régime de catcode avant de traiter son contenu, et le rétablit juste après, or au sein d'un argument de macro le contenu de l'argument est déjà converti en une liste d'unités lexicales selon le régime de catcode courant à l'appel de la macro. Cependant, le paquetage `cprotect` peut être utile dans cette situation.

Une utilisation courante d'entrée `verbatim` est de composer du code informatique. Il y a des paquetages qui améliorent l'environnement `verbatim`. Par exemple, une amélioration est de permettre l'inclusion `verbatim` de fichiers externes, totalement ou en partie. Parmi ces paquetages : `listings` et `minted`.

Un paquetage qui offre beaucoup plus d'options pour les environnements `verbatim` est `fancyvrb`. Un autre `verbatimbox`.

Pour une liste complète de tous les paquetages pertinents voir le CTAN (voir Section 2.6 [CTAN], page 6).

8.27.1 `\verb`

Synopsis :

```
\verbcartexte-littéralcar
\verb*cartexte-littéralcar
```

La commande `\verb` compose *texte-littéral* comme il est entré, en incluant les caractères spéciaux et les espaces, en utilisant la police tapuscrit (`\tt`). Aucun espaces n'est autorisé entre `\verb` ou `\verb*` et le délimiteur *car* qui marque le début et la fin du texte verbatim. Le délimiteur ne doit pas apparaître dans le *texte-littéral*.

La forme étoilée (en `*`) diffère seulement en ce que les espaces sont tapés avec un caractère « espace visible ». (Nommément, `_`).

La sortie correspondant à ce qui suit utilise un espace visible de part et d'autre du mot 'with' :

```
Le premier argument de la commande est \verb*!filename with extension! et ...■
```

Pour composer des adresses réticulaires, appelée aussi URL, le paquetage `url` est une option préférable à la commande `\verb`, puisque il permet les sauts de ligne.

Pour du code informatique il y a beaucoup de paquetages plus avantageux que `\verb`. L'un est `listings`, un autre `minted`.

Vous ne pouvez pas utiliser `\verb` au sein d'un argument de macro, par exemple dans l'argument passé à `\section`. Ce n'est pas une question que `\verb` serait fragile (voir Section 12.11 [`\protect`], page 134), mais c'est juste impossible parce que la commande `\verb` change le régime de catcode avant de lire son argument et le rétablit juste après, or au sein d'un argument de macro le contenu de l'argument a déjà été converti en une liste d'unités lexicales selon le régime de catcode courant l'appel de la macro. Cependant, le paquetage `cprotect` peut être utile dans cette situation.

8.28 `verse`

Synopsis :

```
\begin{verse}
  line1 \\
  line2 \\
  ...
\end{verse}
```

L'environnement `verse` est conçu pour la poésie.

Voici un vers tiré du *Cid* de Pierre Corneille :

```
% dans le préambule :
\newcommand*{\Role[1]{\makebox[\linewidth][c]{\textbf{\textsc{#1}}}}\*}
...
\begin{verse}
  ...

\Role{Chimène}
Va, je ne te hais point.\
\Role{Don Rodrigue}
```

```

\phantom{Va, je ne te hais point. }Tu le dois.\\
\Role{Chimène}
\phantom{Va, je ne te hais point. Tu le dois. }Je ne puis.\\
...
\end{verse}

```

Séparez les lignes de chaque strophe avec \\, et utilisez une ou plusieurs lignes vides pour séparer les strophes.

```

\begin{verse}
\makebox[\linewidth][c]{\textit{De soy-même} --- Clément Marot}
  \\\[1\baselineskip]
Plus ne suis ce que j'ai été      \\
Et plus ne saurai jamais l'être  \\
Mon beau printemps et mon été    \\
Ont fait le saut par la fenêtre

Amour tu as été mon maître      \\
Je t'ai servi sur tous les dieux \\
Ah si je pouvais deux fois naître\\
Comme je te servirais mieux
\end{verse}

```

En sortie, les marges sont renforcées sur la gauche et la droite, les paragraphes ne le sont pas, et le texte n'est pas justifié à droite.

9 Saut à la ligne

La première chose que L^AT_EX fait lorsqu'il traite du texte ordinaire est de traduire votre fichier d'entrée en une séquence de glyphes et d'espaces. Pour produire un document imprimé, cette séquence doit être rompue en lignes (et ces lignes doivent être rompues en pages).

D'ordinaire L^AT_EX effectue pour vous les sauts de ligne (et de page) dans le corps du texte, mais dans certains environnements vous forcez manuellement les sauts.

Un flux de production communément utilisé est d'obtenir une version finale du contenu du document avant de faire une passe finale dessus et de considérer le sauts de ligne (et de page). La plupart des gens ne considèrent pas que L^AT_EX est un traitement de texte parce qu'il n'affiche pas instantanément la sortie. Toutefois, différer la sortie encourage l'utilisateur à remettre jusqu'à la fin des ajustements de format, et ainsi évite bien des va-et-vient sur des sauts susceptibles de changer.

Différer la sortie a d'autres avantages : cela permet de ne faire aucun compromis sur la composition, ce qui assure que ce qu'on voit est exactement ce qu'on obtient, et cela aide également les auteurs à se concentrer soit sur écrire, soit sur se relire, plutôt que les distraire en faisant les deux à la fois.

9.1 `\`

Synopsis, l'un parmi :

```
\
\ [espaceenplus]
```

ou l'un parmi :

```
\*
\* [espaceenplus]
```

Finir la ligne courante. L'argument optionnel *espaceenplus* spécifie l'espace supplémentaire vertical à insérer avant la ligne suivante. C'est une longueur élastique (voir Chapitre 14 [Lengths], page 141) et elle peut être négative. Le texte avant le saut est composé à sa longueur normale, c.-à-d. qu'il n'est pas dilaté pour remplir la largeur de la ligne. Cette commande est fragile (voir Section 12.11 [`\protect`], page 134).

```
\title{Mon histoire~: \ [7mm]
      un conte pathétique}
```

La forme étoilée, `*`, dit à L^AT_EX de ne pas commencer une nouvelle page entre les deux lignes, en émettant un `\nobreak`.

Les sauts de ligne explicites au sein du corps de texte sont inhabituels en L^AT_EX. En particulier, ne commencez pas un nouveau paragraphe avec `\`, mais laissez une ligne à blanc pour le faire. De même, on mettez pas une séquence de `\` pour créer un espace vertical, mais utilisez à la place `\vspace{longueur}`, `\leavevmode\vspace{longueur}`, ou `\vspace*{longueur}` si on veut que l'espace ne soit pas supprimé au sommet d'une nouvelle page (voir Section 19.14 [`\vspace`], page 201).

Cette commande est utilisée principalement en dehors du flux principal de texte comme dans un environnement `tabular` ou `array` ou dans un environnement d'équation.

La commande `\newline` est un synonyme de `\\` (voir Section 9.3 [`\newline`], page 113) dans des circonstances ordinaires (un exemple d’exception est la colonne de type `p{...}` dans un environnement `tabular` ; voir Section 8.23 [`tabular`], page 97) .

La commande `\\` est une macro, et sa définition change selon le contexte, de sorte que sa définition diffère pour chaque cas selon qu’on est dans du texte normal, ou dans un environnement `center`, ou `flushleft`, ou `tabular`. Dans du texte normal quand elle force un saut de ligne, elle est essentiellement un raccourci pour `\newline`. Elle ne termine ni le mode horizontal ni le paragraphe, mais insère juste un ressort et une pénalité de sorte que lorsque le paragraphe se termine un saut de ligne se produit à cet endroit, avec la ligne plus courte bourrée avec de l’espace blanc.

Vous obtenez l’erreur ‘`LaTeX Error: There’s no line here to end`’ si vous utilisez `\\` pour demander une nouvelle ligne, plutôt que pour finir la ligne courante. Par exemple si vous avez `\begin{document} \\` ou, plus vraisemblablement, quelque chose du genre de :

```
\begin{center}
  \begin{minipage}{0.5\textwidth}
  \\
  Dans cet espace vertical apposez votre marque.
  \end{minipage}
\end{center}
```

La solution est de remplacer la double controbligue par quelque chose du genre de `\vspace{\baselineskip}`.

9.2 `\obeycr` & `\restorecr`

La commande `\obeycr` a pour effet qu’un retour chariot dans le fichier d’entrée (‘`^M`’, internement) soit traité de la même façon que `\\` suivi d’un `\relax`. Ainsi chaque passage à la ligne dans l’entrée est aussi un passage à la ligne dans la sortie. La commande `\restorecr` restaure le comportement normal vis à vis des sauts de ligne.

Ceci n’est pas la façon d’afficher textuellement du texte ou du code informatique. Utilisez plutôt `verbatim` (voir Section 8.27 [`verbatim`], page 108).

Avec les réglages par défaut habituels de `LATEX`, l’exemple suivant :

```
aaa
bbb

\obeycr
ccc
ddd
  eee

\restorecr
fff
ggg

hhh
iii
```

produit en sortie quelque chose du genre de :

```

aaa bbb
ccc
ddd
eee

fff ggg
hhh iii

```

où les renforcements sont des renforcements de paragraphes.

9.3 `\newline`

Dans du texte ordinaire, cette commande termine une ligne d'une façon qui ne la justifie pas à droite, de sorte que le texte précédent la fin de ligne n'est pas étiré. c.-à-d. que, en mode paragraphe (voir Chapitre 17 [Modes], page 185), la commande `\newline` est équivalente à une double-contrôle (voir Section 9.1 [`\`], page 111). Cette commande est fragile (voir Section 12.11 [`\protect`], page 134).

Toutefois, les deux commandes diffèrent au sein d'un environnement `tabular` ou `array`. Dans une colonne avec un spécificateur produisant une boîte paragraphe, comme typiquement `p{.}`, `\newline` insère un saut de ligne au sein de la colonne, c.-à-d. insère une fin de ligne à l'intérieur de la colonne, c.-à-d. qu'elle ne rompt pas la ligne entière de tableau. Pour rompre la ligne entière utilisez `\\` ou son équivalent `\tabularnewline`.

L'exemple suivant écrit 'Nom~:' et 'Adresse~:' comme deux lignes au sein d'une seule cellule du tableau.

```

\begin{tabular}{p{2.5cm}@{\hspace{5cm}}p{2.5cm}}
  Nom~: \newline Adresse~: & Date~: \\ \hline
\end{tabular}

```

La 'Date~:' sera alignée sur la ligne de base de 'Nom~:'.

9.4 `\-` (césure à gré)

La commande `\-` dit à \LaTeX qu'il peut faire une césure du mot à cet endroit. Quand vous insérez des commandes `\-` dans un mot, le mot ne pourra subir de césure qu'à ces endroits et non dans aucun des autres points de césure que \LaTeX aurait pu choisir sinon. Cette commande est robuste (voir Section 12.11 [`\protect`], page 134).

\LaTeX est bon en ce qui concerne les césures, et il trouve la plupart des points corrects de césure, tout en n'en utilisant presque jamais un incorrect. La commande `\-` est utilisée pour les cas exceptionnels.

Par ex., \LaTeX d'ordinaire ne fait pas de césure dans les mots contenant un trait d'union. Ci-dessous le mot long et portant une césure indique que \LaTeX doit insérer des espaces inacceptablement long pour composer la colonne étroite :

```

\begin{tabular}{r{4.5cm}}
  Isaac Asimov & La tradition de
  l'anti-intellectualisme
  % l'an\ -ti-in\ -tel\ -lec\ -tu\ -al\ -isme
  a été une tendance constante, qui a fait son chemin dans notre vie

```

```

    politique et culturelle, nourrie par la fausse idée que la démocratie
    signifie que mon ignorance vaut autant que votre savoir.
\end{tabular}

```

Si on met en commentaire la troisième ligne et sort de commentaire la quatrième, alors les choses s’ajusteront bien mieux.

La commande `\-` ne fait qu’autoriser \LaTeX à faire une césure à cet endroit, elle ne force pas la césure. Vous pouvez forcer la césure avec quelque chose du genre de `ef-\linebreak farons`. Bien sûr, si vous changez plus tard le texte, alors cette césure forcée pourrait avoir l’air très étrange, c’est pourquoi cette approche demande de l’attention.

9.5 `\discretionary` (point de césure généralisé)

Synopsis :

```
\discretionary{avant-saut}{après-saut}{sans-saut}
```

Gère la modification d’un mot autour d’une césure. Cette commande est d’un usage rare dans les documents \LaTeX .

Si un saut de ligne survient au point où `\discretionary` apparaît, alors \TeX met *avant-saut* à la fin de la ligne courante et met *après-saut* au début de la ligne suivante. S’il n’y a pas de saut à cet endroit alors \TeX met *sans-saut*.

Dans le mot ‘difficile’ les trois lettres *ffi* forment une ligature. \TeX peut néanmoins faire un saut entre les deux ‘f’ avec ceci :

```
di\discretionary{f-}{fi}{ffi}cile
```

Notez que l’utilisateur n’a pas à faire ceci. C’est typiquement géré automatiquement par l’algorithme de césure de \TeX .

Les arguments de `\discretionary` ne peuvent contenir que des caractères, des boîtes ou des crénaux.

La commande `\discretionary` permet de contrôler finement la césure dans les cas où ne suffisent ni le contrôle standard de la césure fait l’algorithme de césure de \TeX et les règles de césures données par les paquetages de gestion linguistiques, ni les moyens de contrôle explicites offerts par les commandes `\hyphenation` (voir Section 9.7 [`\hyphenation`], page 115) et `\-` (voir Section 9.4 [`\-` (`hyphenation`)], page 113).

L’usage typique de `\discretionary` est par exemple de contrôler la césure au sein d’une formule mathématique en mode ligne (voir aussi Section 16.8 [Math miscellany], page 183). Ci-dessous un exemple de contrôle de la césure au sein d’une adresse réticulaire, où l’on autorise la césure sur les obliques mais en utilisant une controblaque violette en lieu de trait d’union :

```

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage{xcolor}
\usepackage{hyperref}
\usepackage{french}
\newcommand*{\DiscrSlash}{\discretionary{\mbox{\textcolor
{purple}{\textbackslash}}}{/}{/}}

```

```

\begin{document}
Allez donc à \href{http://une/tr\%c3%A8s/tr\%c3%A8s/longue%
/mais/vraiment/tr\%c3%A8s/longue/adresse/r\%C3%A9ticulaire%
/index.html}{http://une\DiscrSlash très\DiscrSlash très\DiscrSlash
longue\DiscrSlash mais\DiscrSlash vraiment\DiscrSlash
très\DiscrSlash longue\DiscrSlash adresse\DiscrSlash
réticulaire\DiscrSlash index.html}
\end{document}

```

9.6 \fussy & \sloppy

Ce sont des déclarations pour que T_EX fasse plus ou moins le difficile à propos des sauts de ligne. Déclarer `\fussy` évite en général trop d'espace entre les mots, au prix d'occasionnellement une boîte trop pleine (`'overfull \hbox'`). Inversement `\sloppy` évite les boîtes trop pleines mais en pâtissant d'un espacement inter-mot relâché.

Le réglage par défaut est `\fussy`. Les sauts de ligne au sein d'un paragraphe sont contrôlés par celle-là des deux déclarations qui est en cours à la fin du paragraphe, c.-à-d. à la ligne à blanc, la commande `\par` ou l'équation hors texte terminant ce paragraphe. De sorte que pour affecter les sauts de ligne vous devez inclure la matière qui termine le paragraphe dans la portée de la commande.

9.6.1 sloppypar

Synopsis :

```

\begin{sloppypar}
... paragraphes ...
\end{sloppypar}

```

Compose les paragraphes avec `\sloppy` en vigueur (voir Section 9.6 [`\fussy & \sloppy`], page 115). Utilisez cela pour ajuster localement les sauts de ligne, pour éviter les erreurs `'Overfull box'` ou `'Underfull box'`.

Voici un exemple simple :

```

\begin{sloppypar}
Son plan arrêté de la sorte pour la matinée, dès qu'elle eut déjeuné,
elle prit \emph{Udolphe} et s'assit, décidée à rester toute à sa
lecture jusqu'à ce que la pendule marquât une heure. Cependant, et
sans que Catherine en fût importunée (l'habitude\ldots), des phrases
sans suite fluaient de Mme Allen~: elle ne parlait jamais beaucoup,
faute de penser, et, pour la même raison, n'était jamais complètement
silencieuse. Qu'elle perdit son aiguille, cassât son fil, entendît le
roulement d'une voiture, aperçût une petite tache sur sa robe, elle
le disait, qu'il y eût là ou non quelqu'un pour la réplique.
\end{sloppypar}

```

9.7 \hyphenation

Synopsis :

```

\hyphenation{mot1 ...}

```

La commande `\hyphenation` déclare les points de césure autorisés au sein des mots donnés dans la liste. Les mots de cette liste sont séparés par des espaces. Spécifiez les points de césure autorisés avec le caractère tiret, `-`.

Utilisez des lettres en bas de casse. `TEX` ne fera de césure que si le mot correspond exactement, aucune désinence n'est essayée. Des commandes `\hyphenation` multiples s'accumulent.

Les paquetages d'internationalisation du genre de `babel` ou `polyglossia` chargent déjà les motifs de césure de la langue courante, aussi on n'utilise la commande `\hyphenation` que pour les mots qui n'y existent pas.

Voici un exemple :

```
\hyphenation{rhi-no-fé-roce sex-cuse doc-tam-bule ædi-peux}
```

Dans une langue s'écrivant avec des diacritiques, comme le français ou le polonais, la césure ne fonctionne correctement avec le moteur `pdflatex` qu'avec des codages modernes (voir Section 2.3 [`TEX engines`], page 3). Il faut donc charger `fontenc` avec un autre codage qu'`OT1`, typiquement pour le français on utilise `T1` (voir Section 4.1 [`fontenc package`], page 17).

9.8 `\linebreak` & `\nolinebreak`

Synopsis, l'un parmi :

```
\linebreak
\linebreak[de-zéro-à-quatre]
```

ou l'un parmi :

```
\nolinebreak
\nolinebreak[de-zéro-à-quatre]
```

Encourage (`\linebreak`) ou décourage (`\nolinebreak`) un saut de ligne. L'argument optionnel *de-zéro-à-quatre* est un entier compris entre 0 et 4 qui vous permet d'adoucir l'instruction. Le réglage par défaut est 4, de sorte que sans l'argument optionnel respectivement la commande force ou interdit le saut. Mais par exemple, `\nolinebreak[1]` est une suggestion qu'il serait préférable de faire le saut à un autre endroit. Plus le nombre est grand, et plus la requête est insistante. Les deux commandes sont fragiles (voir Section 12.11 [`\protect`], page 134).

Dans l'exemple suivant on dit à `LATEX` qu'un bon endroit où faire un saut de ligne est après le `\blabla` juridique :

```
\blablajuridique{\linebreak[2] Nous rappelons que le port de signes
ostentatoires d'appartenance à une communauté religieuse est interdit
parmi le personnel au contact du public.
```

Quand vous émettez un `\linebreak`, les espaces de la ligne sont dilatés de sorte que le point du saut atteigne la marge de droite. Voir Section 9.1 [`\`], page 111, et Section 9.3 [`\newline`], page 113, pour ne pas dilater les espaces.

10 Saut de page

L^AT_EX commence de nouvelles pages de façon asynchrone, lorsque suffisamment de matière a été accumulée pour remplir une page. D'ordinaire ceci se produit automatiquement, mais parfois on peut désirer influencer les sauts.

10.1 `\clearpage` & `\cleardoublepage`

Synopsis :

```
\clearpage
```

ou

```
\cleardoublepage
```

Finit la page courante et envoie en sortie tous les flottants, tableaux et figures (voir Section 5.7 [Floats], page 37). S'il y a trop de flottants pour tenir dans la page, alors L^AT_EX insère des pages supplémentaires contenant seulement les flottants. Dans le cas d'une impression en style `twoside`, `\cleardoublepage` fait aussi que la nouvelle page de contenu soit une page de droite (numérotée impaire), en produisant si nécessaire une page à blanc. La commande `\clearpage` est robuste alors que `\cleardoublepage` est fragile (voir Section 12.11 [`\protect`], page 134).

Les sauts de page de L^AT_EX sont optimisés, aussi, d'ordinaire, vous n'utilisez ces commandes que la touche finale à la dernière version d'un document, ou au sein de commande.

La commande `\cleardoublepage` insère une page à blanc, mais elle contient l'en-tête et le bas de page. Pour obtenir une page vraiment à blanc, utilisez cette commande :

```
\let\origdoublepage\cleardoublepage
\newcommand{\clearempydoublepage}{%
  \clearpage
  {\pagestyle{empty}\origdoublepage}%
}
```

Si vous désirez que la commande standard `\chapter` de L^AT_EX agisse ainsi, alors ajoutez la ligne `\let\cleardoublepage\clearempydoublepage`.

La commande `\newpage` (voir Section 10.2 [`\newpage`], page 117) finit également la page en cours, mais sans débourrer les flottants pendants. Et, si L^AT_EX est en mode deux-colonnes alors `\newpage` finit la colonne courante alors que `\clearpage` et `\cleardoublepage` finissent la page courante.

10.2 `\newpage`

Synopsis :

```
\newpage
```

La commande `\newpage` finit la page courante. Cette commande est robuste (voir Section 12.11 [`\protect`], page 134).

Les sauts de page de L^AT_EX sont optimisés de sorte que d'ordinaire on n'utilise cette commande dans le corps d'un document que pour les finitions sur la version finale, ou à l'intérieur de commandes.

Alors que les commandes `\clearpage` et `\cleardoublepage` finissent également la page courante, en plus de cela elles débloquent les flottants pendants (voir Section 10.1 [`\clearpage` & `\cleardoublepage`], page 117). Et, si L^AT_EX est en mode deux-colonnes alors `\clearpage` et `\cleardoublepage` finissent la page courante en laissant possiblement une colonne vide, alors que `\newpage` finit seulement la colonne courante.

Contrairement à `\pagebreak` (voir Section 10.4 [`\pagebreak` & `\nopagebreak`], page 118), la commande `\newpage` n'a pas pour effet que la nouvelle page commence exactement là où la requête est faite. Dans l'exemple suivant

```
Four score and seven years ago our fathers brought forth on this
continent,
\newpage
\noindent a new nation, conceived in Liberty, and dedicated to the
proposition that all men are created equal.
```

la nouvelle page commence après 'continent', et la ligne où se produit le saut n'est pas justifiée à droite. De plus, `\newpage` n'étire pas verticalement la page, contrairement à `\pagebreak`.

10.3 `\enlargethispage`

```
\enlargethispage{size}
\enlargethispage*{size}
```

Agrandit la `\textheight` de la page courante de la quantité spécifiée ; par exemple `\enlargethispage{\baselineskip}` permet d'avoir une ligne de plus.

La forme étoilée essaie de comprimer la matière sur la page autant que possible. Ceci est normalement utilisé en même temps qu'un `\pagebreak` explicite.

10.4 `\pagebreak` & `\nopagebreak`

Synopsis :

```
\pagebreak[priorité]
\nopagebreak[priorité]
```

Par défaut, la commande `\pagebreak` (`\nopagebreak`) force (empêche) un saut de page à la position courante. Avec `\pagebreak`, l'espace verticale sur la page est dilatée où c'est possible de sorte à s'étendre jusqu'à la marge inférieure normale.

Avec l'argument optionnel *priorité*, vous pouvez convertir la commande `\pagebreak` d'une demande à une exigence. Le nombre doit être compris entre 0 et 4. Plus grand est le nombre, et plus insistant est l'exigence.

11 Notes en bas de page

Mettez une note en bas de la page courante avec la commande `\footnote`, comme ci-dessous.

```
Noël Coward a dit plaisamment que lire une note en bas de page, c'est
comme devoir descendre répondre à la porte d'entrée alors qu'on est en
train de faire l'amour\footnote{Je n'en sais rien, je ne lis pas les
notes en bas de page.}
```

Vous pouvez placer beaucoup de notes en bas de page dans une page. Si le texte devient trop long alors il est continué sur la page suivante.

Vous pouvez aussi produire des notes en bas de page en combinant les commandes `\footnotemark` et `\footnotetext`, ce qui est utiles dans des circonstances spéciales.

Pour faire que les références bibliographiques sortent comme des notes en bas de page vous devez inclure un style bibliographique avec ce comportement (voir Section 8.24.4 [Using BibTeX], page 105).

11.1 `\footnote`

Synopsis :

```
\footnote[numéro]{texte}
```

Place en bas de la page courante une note *texte* numérotée.

```
Il y a plus d'un millier de notes en bas de page dans \textit{Histoire
Du Déclin Et De La Chute De L'empire Romain}\footnote{Après la lecture
d'une version préliminaire, David Hume s'est plaint, « On souffre d'un
fléau avec ces notes, selon la méthode actuelle d'imprimer le livre » et
il suggera qu'elles « ne soient seulement imprimée dans la marge ou en
bas de page »} d'Édouard Gibbon.
```

L'argument optionnel *numéro* vous permet de spécifier le numéro de la note. Le compteur `footnote` donnant le numéro de la note n'est pas incrémenté si vous utilisez cette option, mais l'est dans le cas contraire.

Changez l'aspect que L^AT_EX donne au compteur de note en bas de page avec quelque chose du genre de `\renewcommand{\thefootnote}{\fnsymbol{footnote}}`, ce qui utilise une séquence de symboles (voir Section 13.1 [`\alph \Alph \arabic \roman \Roman \fnsymbol`], page 138). Pour rendre ce changement global mettez cela dans le préambule. Si vous faites ce changement local alors vous désirez sans doute réinitialiser le compteur avec `\setcounter{footnote}{0}`. Par défaut L^AT_EX utilise des nombres arabes.

Le comportement par défaut de L^AT_EX met beaucoup de restrictions sur l'endroit où l'on peut utiliser une `\footnote` ; par exemple, vous ne pouvez pas l'utiliser dans l'argument d'une commande de rubricage telle que `\chapter` (elle ne peut être utilisée qu'en mode paragraphe externe ; voir Chapitre 17 [Modes], page 185). Il y a des contournements ; voir les rubriques suivantes.

Au sein d'un environnement `minipage` la commande `\footnote` utilise le compteur `mpfootnote` au lieu de `footnote`, de sorte que les notes sont numérotées indépendamment. Elles apparaissent au bas de l'environnement, et non au bas de la page. Et par défaut elles apparaissent alphabétiquement. Voir Section 8.18 [minipage], page 79.

11.2 `\footnotemark`

Synopsis, l'un parmi :

```
\footnotemark
\footnotemark[numéro]
```

Place dans le texte la marque de la note courante. Pour spécifier le texte associé de la note voir Section 11.3 [`\footnotetext`], page 120. L'argument optionnel *numéro* a pour effet que la commande utilise ce numéro pour déterminer la marque de note en bas de page. Cette commande peut être utilisée en mode paragraphe interne (voir Chapitre 17 [Modes], page 185).

Si vous utilisez `\footnotemark` sans l'argument optionnel alors elle incrémente le compteur `footnote`, mais si vous utilisez l'argument optionnel *numéro*, alors elle ne l'incrémente pas. L'exemple suivant produit plusieurs marques de note en bas de page faisant référence à la même note :

```
Le premier théorème \footnote{Dû à Gauss.}
et le deuxième théorème\footnotemark[\value{footnote}]
ainis que le troisième théorème\footnotemark[\value{footnote}].
```

S'il y a d'autres notes entre celles-ci, alors vous devez mémoriser la valeur du numéro de la marque commune. L'exemple suivant l'illustre :

```
\title{Un traité sur le théorème du binôme}
\newcounter{sauvenumnote}
...
\setcounter{sauvenumnote}{\value{footnote}}
\author{J Moriarty\thanks{University of Leeds}
\and A C Doyle\thanks{Durham University}
\and S Holmes\footnotemark[\value{sauvenumnote}]}
\begin{document}
\maketitle
```

11.3 `\footnotetext`

Synopsis, l'un des deux :

```
\footnotetext{texte}
\footnotetext[numéro]{texte}
```

Place *texte* au bas de la page comme une note en bas de page. Cette commande fait équipe avec `\footnotemark` (voir Section 11.2 [`\footnotemark`], page 120) et peut apparaître n'importe où après cette commande-là, mais doit apparaître en mode paragraphe externe (voir Chapitre 17 [Modes], page 185). L'argument optionnel *numéro* change le numéro de note affiché.

Voir Section 11.2 [`\footnotemark`], page 120, et Section 11.5 [Footnotes in a table], page 121, pour des exemples d'utilisation.

11.4 Note en bas de page dont le renvoi est au sein d'un titre de rubrique

Mettre une note en bas de page depuis un titre de rubrique, comme dans :

```
\section{Les ensembles complets\protect\footnote{Ce texte est dû à ...}}
```

a pour effet que la note en bas de page apparaît à la fois en bas de la page où la rubrique commence, et de celle où son titre apparaît dans la table des matières, ce qui vraisemblablement n'est pas voulu. Pour qu'elle n'apparaisse pas sur la table des matières utilisez le paquetage `footmisc` avec l'option `stable`.

```
\usepackage[stable]{footmisc}
..
\begin{document}
..
\section{Les ensembles complets}\footnote{Ce texte est dû à R~Jones.}}
```

Notez que le `\protect` a disparu ; l'inclure aurait pour effet que la note en bas de page apparaîtrait de nouveau dans la table des matières.

11.5 Notes en bas de page dans un tableau

Au sein d'un environnement `tabular` ou `array` la commande `\footnote` ne fonctionne pas ; on obtient une marque de note dans la cellule du tableau mais le texte de la note n'apparaît pas. La solution est d'utiliser un environnement `minipage`, comme ci-après (voir Section 8.18 [`minipage`], page 79).

```
\begin{center}
\begin{minipage}{\textwidth} \centering
\begin{tabular}{l|l}
\textsc{Navire} & \textsc{Roman} \\ \hline
\textit{HMS Sophie} & \textit{Maître à bord} \\
\textit{HMS Polychrest} & \textit{Capitaine de vaisseau} \\
\textit{HMS Lively} & \textit{Capitaine de vaisseau} \\
\textit{HMS Surprise} & \textit{Plusieurs romans}\footnote{Le premier étant \\ \textit{La Surprise}.}
\end{tabular}
\end{minipage}
\end{center}
```

Au sein d'une `minipage`, les marques de note sont des lettres en bas de casse. On peut changer cela avec quelque chose du genre de `\renewcommand{\thempfootnote}{\arabic{mpfootnote}}` (voir Section 13.1 [`\alph \Alph \arabic \roman \Roman \fnsymbol`], page 138).

Les notes du premier exemple apparaissent en bas de la `minipage`. Pour qu'elles apparaissent en bas de la page principale et fassent partie de la séquence normales des notes en bas de page, utilisez la pair `\footnotemark` et `\footnotetext` ainsi qu'une instantiation de compteur.

```
\newcounter{sauvegardeLaValeurDeMpFootnote}
\begin{center}
\begin{minipage}{\textwidth}
\setcounter{sauvegardeLaValeurDeMpFootnote}{\value{footnote}} \centering
\begin{tabular}{l|l}
\textsc{Femme} & \textsc{Relation} \\ \hline
Mona & \textit{Liaison}\footnotemark \\
Diana Villiers & \textit{Épouse, à la fin} \\
Christine Hatherleigh Wood & \textit{Fiancée}\footnotemark
\end{tabular}
\end{minipage}
\end{center}
```

```

\end{tabular}
\end{minipage}% le signe pourcent permet de garder roches le texte de la note
% et la minipage
\stepcounter{sauvegardeLaValeurDeMpFootnote}%
\footnotetext[\value{sauvegardeLaValeurDeMpFootnote}]{%
  Les circonstances de sa mort sont peu connues.}%
\stepcounter{sauvegardeLaValeurDeMpFootnote}%
\footnotetext[\value{sauvegardeLaValeurDeMpFootnote}]{%
  Leur relation n'aboutit pas dans le tome XXI.}
\end{center}

```

Pour un environnement flottant `table` (voir Section 8.22 [table], page 96) utilisez le paquetage `tablefootnote`.

```

\usepackage{tablefootnote} % dans le préambule
...
\begin{table}
\centering
\begin{tabular}{l|l}
\textsc{Date} & \textsc{Campagne} \\ \hline
1862 & & Fort Donelson \\
1863 & & Vicksburg \\
1865 & & Armée de Virginie du Nord\tablefootnote{Fin de la
  guerre.}
\end{tabular}
\caption{Forces capturées par le général Grant}
\end{table}

```

La note apparaîtra en bas de la page et sera numérotée dans la même séquence que les autres notes en bas de page.

11.6 Notes en bas de page sur d'autres notes

En particulier dans les sciences humaines, les auteurs peuvent avoir plusieurs classes de notes en bas de pages, y compris une note en bas de page qui porte sur une autre note en bas de page. Le paquetage `bigfoot` étend le mécanisme de note en bas de page par défaut de L^AT_EX de bien des manières, y compris pour permettre de faire les deux notes de l'exemple ci-dessous :

```

\usepackage{bigfoot} % dans le préambule
\DeclareNewFootnote{Default}
\DeclareNewFootnote{from}[alph] % crée la classe \footnotefrom{}
...

```

```

Le troisième théorème est une réciproque partielle du second.\footnotefrom{
  Remarqué par Wilson.\footnote{Deuxième édition seulement.}}

```

11.7 Paramètres des notes en bas de page

`\footnoterule`

Produit le filet de séparation entre texte principal sur une page et les notes en bas de cette page. Les dimensions par défaut : épaisseur (ou largeur)

0.4pt, et longueur `0.4\columnwidth` dans les classes standard de document (à l'exception `slide`, où elle n'apparaît pas).

`\footnotesep`

La hauteur de l'étai placé au début de la note en bas de page. Par défaut, c'est réglé à l'étai normal pour des polices de taille `\footnotesize` (voir Section 4.3 [Font sizes], page 26), donc il n'y a pas d'espace supplémentaire entre les notes. Cela vaut '6.65pt' pour des polices à '10pt', '7.7pt' pour '11pt', et '8.4pt' pour '12pt'.

12 Définitions

L^AT_EX prend en charges la fabrication de nouvelles commandes de bien des genres.

12.1 `\newcommand` & `\renewcommand`

Synopsis, l'un parmi (trois formes ordinaires, trois formes étoilées) :

```
\newcommand{\cmd}{défn}
\newcommand{\cmd}[nargs]{défn}
\newcommand{\cmd}[nargs][optargdéfaut]{défn}
\newcommand*{\cmd}{défn}
\newcommand*{\cmd}[nargs]{défn}
\newcommand*{\cmd}[nargs][optargdéfaut]{défn}
```

ou toutes les mêmes possibilités avec `\renewcommand` au lieu de `\newcommand` :

```
\renewcommand{\cmd}{défn}
\renewcommand{\cmd}[nargs]{défn}
\renewcommand{\cmd}[nargs][optargdéfaut]{défn}
\renewcommand*{\cmd}{défn}
\renewcommand*{\cmd}[nargs]{défn}
\renewcommand*{\cmd}[nargs][optargdéfaut]{défn}
```

Définit ou redéfinit une commande (voir aussi `\DeclareRobustCommand` dans Section 3.3.2 [Class and package commands], page 10).

La forme étoilée (en `*`) de ces commandes interdit que les arguments contiennent une multiplicité de paragraphes de texte (la commande n'est pas `\long`, dans la terminologie de T_EX de base). Avec la forme par défaut, il est possible qu'un argument soit une multiplicité de paragraphes.

Description des paramètres (des exemples suivront) :

cmd Obligatoire ; `\cmd` est le nom de la commande. Il doit commencer par une controbrique, `\`, et ne doit pas commencer avec la chaîne de 4 caractères `\end`. Pour `\newcommand`, il ne doit pas être déjà défini et ne doit pas commencer avec `\end` ; pour `\renewcommand`, il doit être déjà défini.

nargs Optionnel ; un entier compris entre 1 et 9 spécifiant le nombre d'arguments que la commande prend, y compris un éventuel argument optionnel. Si cet argument n'est pas présent, alors la spécification par défaut est que la commande ne prend aucun argument. Lorsque on redéfinit une commande, la nouvelle version peut prendre un nombre différent d'arguments par rapport à l'ancienne.

optargdéfaut

Optionnel ; si cet argument est présent, alors le premier argument de la commande `\cmd` en définition est optionnel, et sa valeur par défaut est *optargdéfaut* (qui peut être une chaîne vide). Si cet argument n'est pas présent, alors `\cmd` ne prend pas d'argument optionnel.

C'est à dire que si `\cmd` est appelée avec à la suite un argument entre crochets, comme dans `\cmd[valopt]`, alors au sein de *défn* le paramètre `#1` est réglé à *valopt*. Par contre si `\cmd` est appelé sans crochet à la suite, alors au sein de

défn le paramètre #1 est réglé à *optargdéfaut*. Dans les deux cas, les arguments obligatoires commencent par #2.

Omettre [*optargdéfaut*] d'une définition est très différent de mettre des crochets vides, comme dans []. Dans le premier cas on dit que la commande ainsi définie ne prend pas d'argument optionnel, de sorte que #1 est le premier argument obligatoire (à supposer que $nargs \geq 1$) ; dans le second cas on règle l'argument optionnel #1 à être une chaîne vide par défaut, c.-à-d. si aucun argument optionnel n'est donné à l'appel.

De même, omettre [*valopt*] d'un appel est également très différent que de donner des crochets vides, comme dans []. Dans le premier cas on règle #1 à la valeur de *valopt* (à supposer que la commande a été définie pour prendre un argument optionnel) ; alors que dans le second on règle #1 à la chaîne vide, comme on le ferait pour n'importe quelle autre valeur.

Si une commande n'est pas définie pour prendre un argument optionnel, mais qu'elle est appelée avec un argument optionnel, les résultats sont imprévisibles : il peut y avoir une erreur L^AT_EX, une sortie incorrectement composée, ou les deux.

défn Obligatoire ; le texte par lequel substituer chaque occurrence de `\cmd`. Les paramètres #1, #2, . . . , #nargs sont remplacés par les valeurs fournies en appelant la commande (ou par *optargdéfaut* dans le cas d'un argument optionnel non spécifié dans l'appel, comme cela vient d'être expliqué).

T_EX ignore les blancs dans le code source à la suite d'un mot de contrôle (voir Section 12.1.1 [Control sequences], page 126), comme dans '`\cmd`'. Si vous désirez une espace réellement à cet endroit, une solution est de taper {} après la commande ('`\cmd{}`'), et une autre est d'utiliser un espace de contrôle explicite ('`\cmd\`').

Un exemple simple de définition d'une nouvelle commande : `\newcommand{\JM}{Jean Martin}` a pour effet le remplacement de `\JM` par le texte plus long de la définition. La redéfinition d'une commande existante est similaire : `\renewcommand{\symboleqcfd}{\small CQFD}`.

Si vous utilisez `\newcommand` et que le nom de commande a déjà été utilisé alors vous obtenez quelque chose du genre de 'LaTeX Error: Command `\fred` already defined. Or name `\end...` illegal, see p.192 of the manual'. Si vous utilisez `\renewcommand` et que le nom de commande n'a pas encore été utilisé alors vous obtenez quelque chose du genre de 'LaTeX Error: `\hank` undefined'.

Dans l'exemple suivant la première définition crée une commande sans argument, et la seconde, une commande avec un argument obligatoire :

```
\newcommand{\etudiant}{Melle~O'Leary}
\newcommand{\defref}[1]{Définition~\ref{#1}}
```

Utilisez la première commande comme dans `Je confie \etudiant{} à vos bons soins`. La seconde commande a un argument variable, de sorte que `\defref{def:base}` se développe en `Définition~\ref{def:base}`, ce qui en fin de compte se développera en quelque chose du genre de 'Définition~3.14'.

Un exemple avec deux arguments obligatoires : `\newcommand{\nbym}[2]{\$#1 \times #2\$}` est invoqué comme `\nbym{2}{k}`.

Un exemple avec un argument optionnel :

```
\newcommand{\salutation}[1][Madame, Monsieur]{#1,}
```

Alors, `\salutation` donne ‘Madame, Monsieur,’ alors `\salutation[Cher Jean]` donne ‘Cher Jean,’. Et `\salutation[]` donne ‘,’.

Cet exemple a un argument optionnel et deux arguments obligatoires.

```
\newcommand{\avocats}[3][cie]{#2, #3, et~#1}
J'emploie \avocats[Odette]{Devoie}{Trichou}.
```

En sortie on a ‘J’emploie Devoie, Trichou, et Odette’. L’argument optionnel, c.-à-d. `Odette`, est associé à `#1`, alors que `Devoie` et `Trichou` le sont à `#2` et `#3`. À cause de l’argument optionnel, `\avocats{Devoie}{Trichou}` donne en sortie ‘J’emploie Devoie, Trichou, et cie’.

Les accolades autour de *défn* ne définissent pas un groupe, c.-à-d. qu’elle ne délimitent pas la portée du résultat du développement de *défn*. Ainsi la définition `\newcommand{\nomnavire}[1]{\it #1}` est problématique dans cette phrase :

```
Le \nomnavire{Monitor} rencontra le \nomnavire{Merrimac}.
```

les mots ‘rencontra le’, et le point, sont mis incorrectement en italique. La solution est de mettre une paire supplémentaire d’accolades au sein de la définition : `\newcommand{\nomnavire}[1]{\it #1}`.

12.1.1 Séquence de contrôle, mot de contrôle et symbole de contrôle

En lisant l’entrée `TEX` convertit le flux des caractères lus en une séquence d’*unités lexicales*, encore appelées *lexèmes*¹. Lorsque `TEX` voit une controblique `\`, il gère les caractères suivants d’une manière spéciale de sorte à former une unité lexicale *séquence de contrôle*.

Les séquences de contrôle peuvent être classées en deux catégories :

- Les *mots de contrôle*, quand la séquence de contrôle est formée à partir d’une `\` suivie par au moins une lettre ASCII (`A-Z` et `a-z`), suivie par au moins une non-lettre.
- Les *symboles de contrôle*, quand la séquence de contrôle est formée à partir d’une `\` suivi d’un unique caractère non-lettre.

La séquence des caractères ainsi trouvés après le `\` est aussi appelé le *nom de la séquence de contrôle*.

Les blancs suivant un mot de contrôle sont ignorés et ne produisent aucun espace blanc en sortie (voir Section 12.1 [`\newcommand` & `\renewcommand`], page 124, et Section 19.6 [`\(SPACE)`], page 195).

Étant donné que la commande `\relax` ne fait rien, l’exemple suivant imprime simplement ‘Bonjour !’ on a utilisé des espaces visible ‘`␣`’ au lieu de blancs :

```
Bon\relax␣␣␣
␣␣␣jour !
```

Ceci parce que les blancs suivant `\relax`, y compris le caractère saut-de-ligne, sont ignorés, et que les blancs en début de ligne sont aussi ignorés (voir [Leading blanks], page 195).

¹ En anglais c’est le terme *jeton* — *token* — qui est utilisé.

12.2 `\providecommand`

Synopsis, l'un parmi :

```
\providecommand{\cmd}{défn}
\providecommand{\cmd}[nargs]{défn}
\providecommand{\cmd}[nargs][optargdéfaut]{défn}
\providecommand*{\cmd}{défn}
\providecommand*{\cmd}[nargs]{défn}
\providecommand*{\cmd}[nargs][optargdéfaut]{défn}
```

Définit une commande, du moment qu'aucune commande de même nom n'existe déjà. Si aucune commande de ce nom n'existe alors ceci a le même effet que `\newcommand` (voir Section 12.1 [`\newcommand` & `\renewcommand`], page 124). Si une commande de ce nom existe déjà alors cette définition est sans effet. Ceci est particulièrement utile dans tout fichier susceptible d'être chargé plus d'une fois, tel qu'un fichier de style. Voir Section 12.1 [`\newcommand` & `\renewcommand`], page 124, pour la description des arguments.

Dans l'exemple suivant :

```
\providecommand{\monaffiliation}{Saint Michael's College}
\providecommand{\monaffiliation}{Lycée Henri IV}
Depuis \monaffiliation.
```

on a en sortie 'Depuis Saint Michael's College'. Contrairement à `\newcommand`, l'utilisation répétée de `\providecommand` pour (essayer de) définir `\monaffiliation` ne produit pas d'erreur.

12.3 `\makeatletter` et `\makeatother`

Synopsis :

```
\makeatletter
... définition de commande comprenant @ dans leur nom ..
\makeatother
```

Utilisez cette paire de commandes quand vous redéfinissez les commandes L^AT_EX dont le nom comprend un caractère arobre '@'. La déclaration `\makeatletter` a pour effet que le caractère arobre ait le code de catégorie des lettres, c.-à-d. le code 11. La déclaration `\makeatother` règle de code de catégorie de l'arobre au code 12, sa valeur d'origine.

À mesure que chaque caractère est lu par T_EX, un code de catégorie lui est assigné. On appelle aussi ce code *catcode* pour faire court. Par exemple, la controblaque \ reçoit le catcode 0. Les noms de commande consistent en un caractère de catégorie 0, d'ordinaire une controblaque, suivi par des lettres, c.-à-d. des caractères de catégorie 11 (à ceci près qu'une commande peut également consister en un caractère de catégorie 0 suivi d'un unique caractère qui n'est pas une lettre).

Le code source de L^AT_EX suit la convention que certaines commandes utilisent @ dans leur nom. Ces commandes sont principalement destinées aux auteurs de paquetages ou de classes. Cette convention empêche les auteurs qui sont juste utilisateurs d'un paquetage ou d'une classe de remplacer accidentellement une telle commande par une commande définie par eux, puisque par défaut l'arobre a le catcode 12 (other).

Utilisez la paire `\makeatletter` et `\makeatother` au sein d'un fichier `.tex`, typiquement dans le préambule, quand vous définissez ou redéfinissez des commandes dont le nom com-

prend @, en entourant votre définition par elles. Ne les utilisez pas au sein de fichiers `.sty` ou `.cls` puisque les commandes `\usepackage` et `\documentclass` font déjà le nécessaire pour que l'arobe ait le catcode d'une lettre, à savoir 11.

Pour une liste complète des macros contenant une arobe dans leur nom, voir le document <http://ctan.org/pkg/macros2e>.

Dans l'exemple suivant une commande `\these@nomuniversite` se trouve dans le fichier de classe, et l'utilisateur veut changer sa définition. Pour cela, il suffit d'insérer les trois lignes suivantes dans le préambule, avant le `\begin{document}` :

```
\makeatletter
\renewcommand{\these@nomuniversite}{Université Lyon III Jean Moulin}
\makeatother
```

12.4 \@ifstar

Synopsis :

```
\newcommand{\macmd}{\@ifstar{\macmd@star}{\macmd@nostar}}
\newcommand{\macmd@nostar}[nostar-nbre-args]{nostar-corps}
\newcommand{\macmd@star}[star-nbre-args]{star-corps}
```

Vous l'avez sans doute remarqué, beaucoup d'environnements ou commandes standards de \LaTeX existent sous une variante avec le même nom mais finissant avec le caractère étoile *, un astérisque. Par exemple c'est le cas des environnements `table` et `table*`, et des commandes `\section` et `\section*`.

Lorsque on définit un environnement, cela est facile puisque `\newenvironment` et `\renewenvironment` autorisent que le nom de l'environnement contienne un astérisque. Il vous suffit donc d'écrire `\newenvironment{monenv}` ou `\newenvironment{monenv*}` et de continuer la définition comme d'habitude. Pour les commandes, c'est plus compliqué car l'étoile n'étant pas une lettre ne peut pas faire partie du nom de la commande. Comme dans le synopsis ci-dessus, on a donc une commande utilisateur, donnée ci-dessus comme `\macmd`, qui doit être capable de regarder si elle est ou non suivie d'une étoile. Par exemple, \LaTeX n'a pas réellement une commande `\section*` ; au lieu de cela la commande `\section` regarde ce qui la suit. Cette première commande n'accepte d'argument, mais au lieu de cela se développe en l'une de deux commandes qui elle accepte des arguments. Dans le synopsis ces commandes sont `\macmd@nostar` et `\macmd@star`. Elles peuvent prendre le même nombre d'arguments ou un nombre différent, ou pas d'argument du tout. Comme d'habitude, dans un document \LaTeX une commande utilisant l'arobe @ dans son nom doit être comprise au sein d'un bloc `\makeatletter ... \makeatother` (voir Section 12.3 [`\makeatletter & \makeatother`], page 127).

Dans cet exemple, on définit `\ciel` comme une commande avec un argument obligatoire et admettant une variante étoilée `\ciel*` également avec un argument obligatoire. Ainsi, `\ciel{bleu}` composera « ciel bleu non étoilé » tandis que `\ciel*{nocturne}` composera « ciel nocturne étoilé ».

```
\makeatletter
\newcommand*\ciel@starred[1]{ciel #1 étoilé}
\newcommand*\ciel@unstarred[1]{ciel #1 non étoilé}
\newcommand\ciel{\@ifstar{\ciel@starred}{\ciel@unstarred}}
```

```
\makeatother
```

Voici un autre exemple, où la variante étoilée prend un nombre d'arguments différent de la non étoilée. Avec cette définition, la célèbre réplique de l'agent 007 « `\agentsecret*{Bond}`, `\agentsecret{James}{Bond}`. » est équivalente à saisir les commandes « Je m'appelle `\textsc{Bond}`, `\textit{James} \textsc{Bond}`. »

```
\makeatletter
\newcommand*\agentsecret@starred[1]{\textsc{#1}}
\newcommand*\agentsecret@unstarred[2]{\textit{#1} \textsc{#2}}
\newcommand\agentsecret{\@ifstar{\agentsecret@starred}{\agentsecret@unstarred}}
\makeatother
```

Après le nom d'une commande, l'étoile est traitée d'une manière similaire à un argument optionnel. (Ceci est différent des noms d'environnement, où l'étoile fait partie du nom lui-même, et peut donc être à n'importe quelle position). D'un point de vue purement technique il est donc possible de mettre un nombre indéfini d'espaces entre la commande et l'étoile. Ainsi `\agentsecret*{Bond}` et `\agentsecret *{Bond}` sont équivalents. Toutefois, la pratique normale est de ne pas insérer de tels espaces.

Il y a deux manières pouvant être plus commodes d'accomplir la même tâche que `\@ifstar`. Le paquetage `suffix` permet la construction `\newcommand\macommande{variante-
non-étoilée}` suivie de `\WithSuffix\newcommand\macommande*{variante-étoilée}`. Et L^AT_EX3 a le paquetage `xparse` qui permet ce code.

```
\NewDocumentCommand\toto{s}{\IfBooleanTF#1
  {variante-étoilée}%
  {variante-non-étoilée}%
}
```

12.5 `\newcounter` : allouer un compteur

Synopsis :

```
\newcounter{nomcompteur}
\newcounter{nomcompteur}[super]
```

La commande `\newcounter` définit un nouveau compteur nommé *nomcompteur*. Le nouveau compteur est initialisé à zéro.

Quand l'argument optionnel `[super]` est fourni, le compteur *nomcompteur* est réinitialisé à chaque incrémentation du compteur nommé *super*.

Voir Chapitre 13 [Counters], page 138, pour plus d'information à propos des compteurs.

12.6 `\newlength`

Synopsis :

```
\newlength{\longueur}
```

Alloue un nouveau registre de longueur (voir Chapitre 14 [Lengths], page 141). L'argument obligatoire `\longueur` doit être une séquence de contrôle (voir Section 12.1.1 [Control sequences], page 126), et donc commencer par une contrôblique `\` dans les circonstances normales. Le nouveau registre détient des longueurs (élastiques) telles que 72.27pt ou 1in plus.2in minus.1in (un registre de longueur L^AT_EX est ce que T_EX de

base appelle un registre `skip`, ou registre de pas). Le registre est créé avec une valeur initiale de zéro. La séquence de contrôle `\longueur` doit ne pas être déjà définie.

Voici un exemple :

```
\newlength{\graphicht}
```

Si vous oubliez la contrôblique alors vous obtenez l'erreur 'Missing control sequence inserted'. Si la séquence de contrôle existe déjà alors vous obtenez quelque chose du genre de 'LaTeX Error: Command \graphicht already defined. Or name \end... illegal, see p.192 of the manual'.

12.7 \newsavebox : allouer une boîte

Alloue un « baquet » pour détenir une boîte. Synopsis :

```
\newsavebox{\cmd}
```

Définit `\cmd` pour se référer à un nouveau baquet pour stocker des boîtes. Une telle boîte sert à détenir de la matière composée, pour l'utiliser plusieurs fois (voir Chapitre 20 [Boxes], page 204) ou pour le mesurer ou le manipuler. Le nom `\cmd` doit commencer par une contrôblique, et ne doit pas être déjà défini.

L'allocation d'une boîte est globale. Cette commande est fragile (voir Section 12.11 [\protect], page 134).

12.8 \newenvironment & \renewenvironment

Ces commandes définissent ou redéfinissent un environnement `env`, c.-à-d., `\begin{env} corps \end{env}`.

Synopsis :

```
\newenvironment{env}[nargs][argoptdéfaut]{défdébut}{déffin}
\newenvironment*{env}[nargs][argoptdéfaut]{défdébut}{déffin}
\renewenvironment{env}[nargs]{défdébut}{déffin}
\renewenvironment*{env}[nargs]{défdébut}{déffin}
```

La forme étoilée de ces commandes exige que les arguments (à ne pas confondre avec le corps de l'environnement) ne contiennent pas de paragraphes de texte.

env Obligatoire ; le nom de l'environnement. Il est constitué seulement de lettres ou du caractère astérisque `*`, et donc ne commence pas par une contrôblique `\`. Il ne doit pas commencer par la chaîne `end`. Pour `\newenvironment`, `env` ne doit pas être le nom d'un environnement déjà existant, et la commande `\env` ne doit pas être définie. Pour `\renewenvironment`, `env` doit être le nom d'un environnement déjà existant.

nargs Optionnel ; un entier de 0 à 9 indiquant le nombre d'arguments que l'environnement attend. Quand l'environnement est utilisé, ces arguments apparaissent après le `\begin`, comme dans `\begin{env}{arg1}...{argn}`. Si cet argument n'est pas présent, alors par défaut l'environnement n'attend pas d'argument. Lorsqu'on redéfinit un environnement, la nouvelle version peut prendre un nombre différent d'arguments que la précédente.

argoptdéfaut

Optionnel ; si cet argument est présent alors le premier argument de l'environnement en définition est optionnel, et sa valeur par défaut est *argoptdéfaut* (éventuellement une chaîne vide). Si cet argument n'est pas présent alors l'environnement ne prend pas d'argument optionnel.

c.-à-d. que si [*argoptdéfaut*] est présent dans la définition de l'environnement alors vous pouvez commencer l'environnement avec des crochets, comme dans `\begin{env}[valopt]{...} ... \end{env}`. Dans ce cas, au sein de *défdébut*, le paramètre #1 est réglé à la valeur de *valopt*. Si par contre `\begin{env}` est appelé sans être suivi de crochet, alors, au sein de *défdébut*, le paramètre #1 est réglé à la valeur par défaut *optargdéfaut*. Dans les deux cas, le premier paramètre obligatoire, s'il en est, est #2.

Omettre [*valopt*] dans l'appel est différent d'avoir des crochets sans contenu, comme dans []. Dans le premier cas #1 se développe en *argoptdéfaut*, et dans le second en une chaîne vide.

défdébut Obligatoire ; le texte qui est développé à toute occurrence de `\begin{env}` ; au sein de *défdébut*, le *n*ème paramètre positionnel, (c.-à-d. #*n*), est remplacé au sein de *défdébut* par le texte du *n*ème argument.

déffin Obligatoire ; le texte développé à toute occurrence de `\end{env}`. Il ne doit contenir aucun paramètre positionnel, ainsi #*n* ne peut pas être utilisé ici (mais voyez l'exemple final ci-après).

Tous les environnements, c'est à dire le code de *défdébut*, le corps de l'environnement, et le code *déffin*, sont traités au sein d'un groupe. Ainsi, dans le premier exemple ci-dessous, l'effet de `\small` est limité à la citation et ne s'étend pas à la matière qui suit l'environnement.

Cet exemple dont un environnement semblable à `quotation` de L^AT_EX à ceci près qu'il sera composé dans une taille de police plus petite :

```
\newenvironment{smallquote}{%
  \small\begin{quotation}
}{%
  \end{quotation}
}
```

Celui-ci montre l'utilisation des arguments ; cela donne un environnement de citation qui affiche l'auteur :

```
\newenvironment{citequote}[1][Corneille]{%
  \begin{quotation}
  \noindent\textit{#1}:
}{%
  \end{quotation}
}
```

Le nom de l'auteur est optionnel, et vaut par défaut 'Corneille'. Dans le document, utilisez l'environnement comme ceci :

```
\begin{citequote}[Clovis, roi des Francs]
...
```

```
\end{citequote}
```

Ce dernier exemple montre comment sauvegarder la valeur d'un argument pour l'utiliser dans *défin*, dans ce cas dans une boîte (voir Section 20.5 [`\sbox & \savebox`], page 208).

```
\newsavebox{\quoteauthor}
\newenvironment{citequote}[1][Corneille]{%
  \sbox\quoteauthor{#1}%
  \begin{quotation}
}%
  \hspace{1em plus 1fill}---\usebox{\quoteauthor}
\end{quotation}
}
```

12.9 `\newtheorem`

Synopsis :

```
\newtheorem{nom}{titre}
\newtheorem{nom}{titre}[numéroté_au_sein_de]
\newtheorem{nom}[numéroté_comme]{titre}
```

Définit un nouvel *environnement simili-théorème*. Vous pouvez spécifier l'un seul de *numéroté_au_sein_de* ou *numéroté_comme*, ou ni l'un ni l'autre, mais non les deux à la fois.

La première forme, `\newtheorem{nom}{titre}`, crée un environnement qui sera étiqueté avec *titre* ; voir le premier exemple ci-dessous.

La seconde forme, `\newtheorem{nom}{titre}[numéroté_au_sein_de]`, crée un environnement dont le compteur est subordonné au compteur déjà existant *numéroté_au_sein_de*, c.-à-d. qui est réinitialisé à chaque réinitialisation de *numéroté_au_sein_de*. Voir le second exemple ci-dessous.

La troisième forme `\newtheorem{nom}[numéroté_comme]{titre}`, avec l'argument optionnel entre les deux arguments obligatoires, crée un environnement dont le compteur partage le compteur déjà défini *numéroté_comme*. Voir le troisième exemple.

Cette déclaration est globale. Elle est fragile (voir Section 12.11 [`\protect`], page 134).

Arguments :

nom Le nom de l'environnement. C'est une chaîne de lettres. Il ne doit pas commencer avec une controblaque (`'\'`). Il ne doit pas être le nom d'un environnement déjà existant ; en fait le nom de commande `\nom` ne doit pas être déjà défini de quelque façon que ce soit.

titre Le texte imprimé au début de l'environnement, avant le numéro. Par exemple, 'Théorème'.

numéroté_au_sein_de

Optionnel ; le nom d'un compteur déjà défini, d'ordinaire une unité sectionnelle telle que `chapter` ou `section`. Lorsque le compteur *numéroté_au_sein_de* est réinitialisé, alors le compteur de l'environnement *nom* l'est aussi.

Si cet argument optionnel est omis alors la commande `\thenom` est définie comme `\arabic{nom}`.

numéroté_comme

Optionnel ; le nom d'un environnement simili-théorème déjà défini. Le nouvel environnement se numérote en séquence avec *numéroté_comme*.

En l'absence des arguments optionnels les environnements sont numérotés en séquence. L'exemple suivant a une déclaration dans le préambule qui résulte en 'Définition 1' et 'Définition 2' en sortie.

```
\newtheorem{defn}{Définition}
\begin{document}
\section{...}
\begin{defn}
  Première déf.
\end{defn}

\section{...}
\begin{defn}
  Seconde déf.
\end{defn}
```

L'exemple suivant reprend le même corps de document que l'exemple précédent. Mais ici l'argument optionnel *numéroté_au_sein_de* de `\newtheorem` est spécifié comme `section`, aussi la sortie est du genre de 'Définition 1.1' et 'Définition 2.1'.

```
\newtheorem{defn}{Definition}[section]
\begin{document}
\section{...}
\begin{defn}
  Première déf.
\end{defn}

\section{...}
\begin{defn}
  Seconde déf.
\end{defn}
```

Dans l'exemple suivant il y a deux déclarations dans le préambule, la seconde desquelles appelle le nouvel environnement `thm` pour utiliser le même compteur que `defn`. Cela donne 'Définition 1.1', suivi de 'Théorème 2.1' et 'Définition 2.2'.

```
\newtheorem{defn}{Définition}[section]
\newtheorem{thm}[defn]{Théorème}
\begin{document}
\section{...}
\begin{defn}
  Première déf.
\end{defn}

\section{...}
\begin{thm}
  Premier théorème
```

```

\end{thm}

\begin{defn}
  Seconde déf.
\end{defn}
\section{...}

```

12.10 `\newfont`: définit une nouvelle police (obsolète)

`\newfont`, désormais obsolète, définit une commande qui commute la police de caractère.

Synopsis :

```
\newfont{\cmd}{description_police}
```

Ceci définit une séquence de contrôle `\cmd` qui change la police courante. \LaTeX cherche sur votre système un fichier nommé `nompolice.tfm`. La séquence de contrôle ne doit pas être déjà définie. Elle doit commencer par une controbligue (`'\'`).

Cette commande est obsolète. c'est une commande de bas niveau pour mettre en place une police individuelle. De nos jours, les polices sont pratiquement toujours définies en familles (ce qui vous permet, par exemple, d'associer un gras et un romain) au travers de ce qu'il est convenu de nommer le « Nouveau Plan de Sélection de Polices de caractère », soit en utilisant des fichiers `.fd` ou à travers l'utilisation d'un moteur qui sait accéder au système de polices de caractère, tel que \XeLaTeX (voir Section 2.3 [\TeX engines], page 3).

Mais puisque cela fait partie de \LaTeX , voici l'explication : le paramètre `description_police` consiste en un `nompolice` et une `clause at` optionnelle ; celle-ci peut avoir soit la forme `at dimen` ou `scaled facteur`, où un `facteur` de '1000' signifie aucune dilatation/contraction. Pour l'usage de \LaTeX , tout ce que ceci fait est de dilater ou contracter le caractère et les autres dimensions relatives à la taille de conception de la police, ce qui est une valeur définie dans le fichier en `.tfm`.

Cet exemple définit deux polices équivalentes et compose quelques caractères dans chacune d'elles :

```

\newfont{\testpoliceat}{cmb10 at 11pt}
\newfont{\testpolicedilatee}{cmb10 scaled 1100}
\testpoliceat abc
\testpolicedilatee abc

```

12.11 `\protect`

Toutes les commandes de \LaTeX sont soit *fragiles* soit *robustes*. Les notes en bas de page, les sauts de ligne, toute commande prenant un argument optionnel, et bien d'autres, sont fragiles. Une commande fragile peut se disloquer et causer une erreur lorsque elle est utilisée au sein de l'argument de certaines commandes. Pour empêcher la dislocation de ces commandes l'une des solutions est de les précéder de la commande `\protect`.

Par exemple, lorsque \LaTeX exécute la commande `\section{nom-rubrique}` il écrit le texte `nom-rubrique` dans le fichier auxiliaire `.aux`, de sorte à pouvoir le tirer de là pour l'utiliser dans d'autres parties du document comme la table des matières. On appelle *argument mouvant* tout argument qui est développé en interne par \LaTeX sans être directement composé en sortie. Une commande est dite fragile si elle se développe pendant ce processus

en un code \TeX non valide. Certains exemples d'arguments mouvants sont ceux qui apparaissent au sein des commandes `\caption{...}` (voir Section 8.10 [figure], page 67), dans la commande `\thanks{...}` (voir Section 18.1 [\maketitle], page 187), et dans les expressions en @ des environnements `tabular` et `array` (voir Section 8.23 [tabular], page 97).

Si vous obtenez des erreurs étranges de commandes utilisées au sein d'arguments mouvants, essayez des les précéder d'un `\protect`. Il faut un `\protect` pour chacune des commandes fragiles.

Bien qu'en général la commande `\protect` ne fait pas de mal, les commandes de gestion de longueurs sont robustes et ne devraient pas être précédées d'une commande `\protect`. Une commande `\protect` ne peut pas non plus être utilisée au sein de l'argument d'une commande Section 13.5 [\addtocounter], page 139, ou Section 13.4 [\setcounter], page 139.

Dans l'exemple qui suit la commande `\caption` produit une erreur mystérieuse

```
\begin{figure}
...
\caption{Company headquarters of A\raisebox{1pt}{B}\raisebox{-1pt}{C}}
\end{figure}
```

Dans l'exemple suivant la commande `\tableofcontents` produit une erreur à cause du `\(..\)` dans le titre de section qui se développe en code incorrect \TeX dans le fichier `.toc`. Vous pouvez résoudre ceci en remplaçant `\(..\)` par `\protect\(..\protect\)`.

```
\begin{document}
\tableofcontents
...
\section{Einstein's \(\ e=mc^2 \)}
...
```

12.12 \ignorespaces & \ignorespacesafterend

Synopsis :

```
\ignorespaces
```

ou

```
\ignorespacesafterend
```

Les deux commandes ont pour effet que \LaTeX ignore l'espace blanc après la commande et jusqu'à rencontrer une boîte ou un caractère non blanc. La première commande est une commande primitive de \TeX , et la seconde est spécifique à \LaTeX .

La commande `\ignorespaces` est souvent utilisée quand on définit des commandes au moyen de `\newcommand`, ou `\newenvironment`, ou `\def`. Ceci est illustré par l'exemple plus bas. Il permet à un utilisateur d'afficher dans la marge combien de points rapporte chaque questions d'un questionnaire, mais il est malcommode parce que, comme illustré dans la liste `enumerate`, l'utilisateur ne doit pas mettre d'espace entre la commande et le texte de la question.

```
\newcommand{\points}[1]{\makebox[0pt]{\makebox[10em][l]{#1~pts}}
\begin{enumerate}
\item\points{10}aucun espace superflu inséré ici
\item\points{15} ici un espace superflu entre le nombre et le mot « ici »■
```

```
\end{enumerate}
```

La solution est de modifier comme ci-dessous :

```
\newcommand{\points}[1]{%
  \makebox[0pt]{\makebox[10em][l]{#1~pts}}\ignorespaces}
```

Dans le second exemple on illustre comment l'espace blanc est enlevé de devant le texte. Les commandes ci-dessous permettent à l'utilisateur d'attacher uniformément une civilité à un nom. Mais si, quand il est donné, une civilité commence accidentellement par un espace alors `\nomcomplet` le reproduira.

```
\newcommand{\honorific}[1]{\def\honorific{#1}} % mémorise le titre
\newcommand{\fullname}[1]{\honorific~#1} % met le titre devant le nom
```

```
\begin{tabular}{|l|}
\honorific{M./Mme} \fullname{Jean} \\ % sans espace superflu
\honorific{ M./Mme} \fullname{Jean} % espace superflu devant la civilité
\end{tabular}
```

Pour réparer cela, modifier le en `\newcommand{\fullname}[1]{\ignorespaces\honorific~#1}`.

Le `\ignorespaces` est aussi souvent utilisé dans un `\newenvironment` à la fin de la clause `begin`, comme dans `\begin{newenvironment}{env name}{... \ignorespaces}{...}`.

Pour enlever l'espace blanc venant immédiatement après un environnement utilisez `\ignorespacesafterend`. Dans l'exemple ci-dessous on affiche un espace vertical bien plus grand entre le premier et le deuxième environnements qu'entre le deuxième et le troisième.

```
\newenvironment{eq}{\begin{equation}}{\end{equation}}
\begin{eq}
e=mc^2
\end{eq}
\begin{equation}
F=ma
\end{equation}
\begin{equation}
E=IR
\end{equation}
```

On peut faire disparaître l'espace vertical en mettant un caractère % de commentaire immédiatement après le `\end{eq}`, mais cela est malcommode. La solution est de modifier en `\newenvironment{eq}{\begin{equation}}{\end{equation}\ignorespacesafterend}`.

12.13 xspace package

Synopsis :

```
\usepackage{xspace}
...
\newcommand{...}{... \xspace}
```

La macro `\xspace`, quand utilisée à la fin de la définition d'une commande, ajoute une espace à moins que la commande soit suivie de certains caractères de ponctuation.

Après la séquence de contrôle d'une commande qui est un mot de contrôle (voir Section 12.1.1 [Control sequences], page 126, contrairement aux symboles de contrôle tels

que $\backslash\$$), $\text{T}_{\text{E}}\text{X}$ avale les caractères blancs. Ainsi, dans la première phrase ci-dessous, en sortie on a ‘Provence’ placé parfaitement contre le point, sans espace entre les deux.

```
\newcommand{\PR}{Provence}
Notre école est en \PR .
Les étés en \PR{} sont agréables.
```

Mais à cause de l’avalement, la seconde phrase a besoin de l’adjonction d’accolades vides {}, sans quoi il n’y aurait pas d’espace entre ‘Provence’ et ‘sont’. (Beaucoup d’auteurs utilisent plutôt un \backslash pour cela. Voir Section 19.6 [$\backslash(\text{SPACE})$], page 195.)

Le paquetage `xspace` fournit la commande $\backslash\text{xspace}$. Elle sert à écrire des commandes conçues pour être utilisées essentiellement dans du texte. Elle doit être placée à la toute fin de la définition de ces commandes. Elle insère un espace après la commande à moins que ce qui la suit immédiatement soit dans une liste d’exceptions. Dans l’exemple suivant, on peut se passer de l’adjonction d’accolades vides.

```
\newcommand{\PR}{Provence\space}
Notre école est en \PR .
Les étés en \PR{} sont agréables.
```

La liste d’exceptions par défaut contient les caractères `,.//?;:~!~-)`, ainsi que l’accolade ouvrante, la commande `controblque-espace` discutée plus haut, et les commandes `\footnote` ou `\footnotemark`. On peut ajouter des éléments à cette liste comme avec `\xspaceaddexceptions{\manotebdpi \manotebdpii}` qui rajoute les séquences de contrôle `\manotebdpi` et `\manotebdpii` à la liste, et retrancher un élément de la liste comme avec `\xspaceremoveexception{!}`.

Comment : nombre d’experts préfèrent ne pas utiliser $\backslash\text{xspace}$. La mettre dans une définition signifie que la commande sera en général suivie du bon espacement. Mais il n’est pas aisé de prédire quand ajouter les accolades vides {} parce que $\backslash\text{xspace}$ se sera trompé, comme lorsque elle est suivie d’une autre commande, et donc $\backslash\text{xspace}$ peut rendre l’édition d’une matière plus difficile et plus faillible que de mettre systématiquement les accolades vides.

13 Compteurs

Tout ce que L^AT_EX numérote pour vous a un compteur associé avec soi. Le nom du compteur est le même que le nom de l'environnement ou de la commande qui produit le numéro, sauf qu'il ne pas de \. (`enumi`–`enumiv` sont utilisés pour les environnements `enumerate` imbriqués). Ci-dessous se trouve une liste des compteurs utilisée dans les classes standards de documents L^AT_EX pour contrôler la numérotation.

<code>part</code>	<code>paragraph</code>	<code>figure</code>	<code>enumi</code>
<code>chapter</code>	<code>subparagraph</code>	<code>table</code>	<code>enumii</code>
<code>section</code>	<code>page</code>	<code>footnote</code>	<code>enumiii</code>
<code>subsection</code>	<code>equation</code>	<code>mpfootnote</code>	<code>enumiv</code>
<code>subsubsection</code>			

13.1 `\alph` `\Alph` `\arabic` `\roman` `\Roman` `\fnsymbol`: Taper des compteurs

Toutes ces commandes prennent un unique compteur en argument, par exemple, `\alph{enumi}`.

`\alph` tape *compteur* en utilisant des lettres en bas de casse : ‘a’, ‘b’, ...

`\Alph` Utilise des lettres capitales : ‘A’, ‘B’, ...

`\arabic` Utilise des numéros en chiffres arabes : ‘1’, ‘2’, ...

`\roman` Utilise des nombres romains en bas de casse : ‘i’, ‘ii’, ...

`\Roman` Utilise des nombres romains en capitales : ‘I’, ‘II’, ...

`\fnsymbol`

Tape la valeur de *compteur* dans une séquence spécifique de neuf symboles (utilisés par convention pour la numérotation des notes en bas de page). La valeur de *compteur* doit être comprise entre 1 et 9 inclus.

Voici ces symboles :

Nom	Commande	Symbole
astérisque	<code>\ast</code>	*
obèle	<code>\dagger</code>	†
double-obèle	<code>\ddagger</code>	‡
marque-de-section	<code>\S</code>	§
marque-de-paragraphe	<code>\P</code>	¶
parallèle	<code>\parallel</code>	
astérisque-double	<code>\ast\ast</code>	**
obèle-double	<code>\dagger\dagger</code>	††
double-obèle-double	<code>\ddagger\ddagger</code>	‡‡

13.2 `\usecounter{compteur}`

Synopsis :

```
\usecounter{compteur}
```

La commande `\usecounter` est utilisée dans le second argument de l'environnement `list` pour spécifier le *compteur* à utiliser pour numéroter les articles de la liste.

13.3 `\value{compteur}`

Synopsis :

```
\value{compteur}
```

Cette commande se développe en la valeur de *compteur*. Elle est souvent utilisé dans `\setcounter` ou `\addtocounter`, mais `\value` peut être utilisé partout là où L^AT_EX attend un nombre. Elle ne doit pas être précédée par `\protect` (voir Section 12.11 [`\protect`], page 134).

La commande `\value` n'est pas utiliser pour composer la valeur du compteur. Voir Section 13.1 [`\alph \Alph \arabic \roman \Roman \fnsymbol`], page 138.

Cet exemple produit en sortie 'Le compteur *essai* vaut 6. Le compteur *autre* vaut 5'.

```
\newcounter{essai} \setcounter{essai}{5}
\newcounter{autre} \setcounter{autre}{\value{essai}}
\addtocounter{essai}{1}
```

```
Le compteur essai vaut \arabic{essai}.
```

```
Le compteur autre vaut \arabic{autre}.
```

Cet exemple insère `\hspace{4\parindent}`.

```
\setcounter{moncptr}{3} \addtocounter{moncptr}{1}
\hspace{\value{moncptr}\parindent}
```

13.4 `\setcounter{compteur}{value}`

Synopsis :

```
\setcounter{compteur}{value}
```

La commande `\setcounter` règle la valeur de *compteur* à l'argument *value*.

13.5 `\addtocounter{compteur}{valeur}`

La commande `\addtocounter` incrémente *compteur* de la quantité spécifiée par l'argument *valeur*, qui peut être négatif.

13.6 `\refstepcounter{compteur}`

La commande `\refstepcounter` fonctionne de la même façon que `\stepcounter` Voir Section 13.7 [`\stepcounter`], page 139, à ceci près qu'elle définit également la valeur courante de `\ref` comme le résultat de `\thecounter`.

13.7 `\stepcounter{compteur}`

La commande `\stepcounter` ajoute un à *compteur* et réinitialise tous les compteurs subsidiaires.

13.8 `\day` & `\month` & `\year`

L^AT_EX définit des compteurs pour `\day` le quantième du mois (nominalement avec une valeur entre 1 et 31), `\month` pour le mois de l'année (nominalement avec une valeur entre 1 et 12), et `\year` pour l'année. Quand T_EX démarre, ils sont mis à la valeur courante du système sur lequel T_EX s'exécute. En relation avec ces compteurs, la commande `\today` produit une chaîne représentant le jour courant (voir Section 23.9 [`\today`], page 236).

Ils ne sont pas remis à jour pendant que le traitement par T_EX progresse, ainsi en principe ils pourraient être incorrect à la fin. De plus, T_EX n'effectue aucun contrôle sanitaire :

```
\day=-2 \month=13 \year=-4 \today
```

ne produit ni erreur ni avertissement, et le résultat est en sortie `'-2, -4'` (la valeur de mois erronée ne produit rien en sortie).

Voir Section 28.2 [Command line input], page 269, pour forcer la date à une valeur donnée sur la ligne de commande.

14 Longueurs

Une *longueur* est une mesure de distance. Beaucoup de commandes L^AT_EX prennent une longueur en argument.

Il y a deux types de longueur. Une *longueur rigide* telle que 10pt ne contient pas de composante en plus ou minus. (En T_EX de base on appelle cela une *dimen*). Une *longueur élastique* (ce qu'en T_EX de base on appelle un *skip* ou une *glue*) telle que dans 1cm plus0.05cm minus0.01cm peut contenir l'un ou l'autre de ces composantes ou les deux. Dans cette longueur élastique, le 1cm est la *longueur naturelle* alors que les deux autres, les composantes en plus et minus, permette à T_EX à dilater ou contracter la longueur pour optimiser la disposition.

Les exemples qui suivent utilisent ces deux commandes.

```
% Fait une barre noire de 10pt de haut et #1 de large
\newcommand{\blackbar}[1]{\rule{#1}{10pt}}
```

```
% Fait une boîte autour de #2 qui est #1 de large (sans la bordure)
\newcommand{\showhbox}[2]{%
  \fboxsep=0pt\fbox{\hbox to #1{#2}}}
```

Cet exemple utilise ces commandes pour afficher une barre noire de 100 points de long entre 'XXX' et 'YYY'. Cette longueur est rigide.

```
XXX\showhbox{100pt}{\blackbar{100pt}}YYY
```

Quant au longueur élastiques, la contraction est le plus simple des deux : avec 1cm minus 0.05cm, la longueur naturelle est 1cm mais T_EX peut la contracter jusqu'à 0,95 cm. T_EX refuse de contracter plus que cela. Ainsi, ci-dessous le premier \showhbox fonctionne bien, un espace de 98 points étant produit entre les deux barres.

```
XXX\showhbox{300pt}{%
  \blackbar{101pt}\hspace{100pt minus 2pt}\blackbar{101pt}}YYY
```

```
XXX\showhbox{300pt}{%
  \blackbar{105pt}\hspace{100pt minus 1pt}\blackbar{105pt}}YYY
```

Mais le second produit un avertissement du genre de 'Overfull \hbox (1.0pt too wide) detected at line 17'. En sortie le premier 'Y' est écrasé par la fin de la barre noire, parce que la matière contenue par la boîte est plus large que les 300 pt alloués, et que T_EX a refusé de contracter le total en deçà de 309 points.

Dilater est similaire à contracter, à ceci près que si T_EX doit dilater plus que la quantité disponible, alors il le fait. Ci-dessous la première ligne fonctionne bien, et produit un espace de 110 points entre les barres.

```
XXX\showhbox{300pt}{%
  \blackbar{95pt}\hspace{100pt plus 10pt}\blackbar{95pt}}YYY
```

```
XXX\showhbox{300pt}{%
  \blackbar{95pt}\hspace{100pt plus 1pt}\blackbar{95pt}}YYY
```

Dans la seconde ligne T_EX a besoin de dilater de 10 points alors que seulement 1 point a été spécifié. T_EX dilate l'espace jusqu'à la longueur requise mais il produit un avertissement du

genre de ‘Underfull \hbox (badness 10000) detected at line 22’. (On ne discute pas ici de ce que signifie « badness »).

Il est possible de mettre à la fois de la dilatation et de la contraction dans la même longueur, comme dans `1ex plus 0.05ex minus 0.02ex`.

Si $\text{T}_{\text{E}}\text{X}$ ajuste plus d’une longueur élastique alors il alloue la dilatation ou la contraction proportionnellement.

```
XXX\showhbox{300pt}{%
  \blackbar{100pt}% gauche
  \hspace{0pt plus 50pt}\blackbar{80pt}\hspace{0pt plus 10pt}% milieu
  \blackbar{100pt}}YYY % droite
```

Les barres de gauche et de droite prennent 100 points, de sorte que celle du milieu a besoin également de 100. La barre du milieu fait 80 points et donc les deux `\hspace` doivent se dilater de 20 points. Comme les deux sont `plus 50pt` et `plus 10pt`, $\text{T}_{\text{E}}\text{X}$ obtient 5/6 de la dilatation du premier espace et 1/6 du second.

La composante `plus` ou `minus` d’une longueur élastique peut contenir un composante en *fill*, comme dans `1in plus2fill`. Cela donne à la longueur une dilatabilité ou contractibilité infinie de sorte que $\text{T}_{\text{E}}\text{X}$ puisse l’ajuster à n’importe quelle distance. Dans l’exemple suivant les deux figures sont également espacées horizontalement sur la page.

```
\begin{minipage}{\linewidth}
  \hspace{0pt plus 1fill}\includegraphics{godel.png}%
  \hspace{0pt plus 1fill}\includegraphics{einstein.png}%
  \hspace{0pt plus 1fill}
\end{minipage}
```

$\text{T}_{\text{E}}\text{X}$ a trois niveaux d’infinité pour les composantes élastiques : `fil`, `fill`, et `filll` dans l’ordre d’infinité croissante. D’ordinaire les auteurs de documents n’utilisent que celle du milieu (voir Section 19.3 [`\hfill`], page 192, et voir Section 19.15 [`\vfill`], page 202).

Multiplier une longueur élastique par un nombre la transforme en une longueur rigide, de sorte qu’après `\setlength{\ylength}{2.5cm plus 0.5cm}` et `\setlength{\zlength}{3\ylength}` alors la valeur de `\zlength` est 2.5cm.

14.1 Unités de longueur

$\text{T}_{\text{E}}\text{X}$ et $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ connaissent ces unités aussi bien en mode mathématique qu’en dehors de celui-ci.

<code>pt</code>	Point (1/72,27) pouce. La conversion (approximative) en unités métriques est 1 point = 0,35146 mm = 0,035146 cm.
<code>pc</code>	Pica, 12 pt
<code>in</code>	Pouce, 72,27 pt
<code>bp</code>	Big point, 1/72 pouce. Cette longueur est la définition d’un point en PostScript et dans beaucoup d’autres systèmes bureautiques d’édition (PAO).
<code>mm</code>	Millimètre, 2,845 pt
<code>cm</code>	Centimètre, 10 mm

dd	Point Didot, 1,07 pt
cc	Cicéro, 12 dd, encore appelé <i>douze</i> .
sp	Point proportionné, (1/65536) pt

Trois autres unités sont définies en fonction de la fonte courante, plutôt que comme des dimensions absolues.

ex La hauteur *ex* d'un x de la fonte courante, traditionnellement la hauteur de la lettre x en bas de casse, est souvent utilisée pour les longueurs verticales.

em De même le *em*, traditionnellement la largeur de la lettre M capitale, est souvent utilisée pour les longueurs horizontales. C'est également souvent la taille de la fonte en cours, par exemple une fonte nominalement en 10 pt a 1 em = 10 pt. L^AT_EX a plusieurs commandes pour produire de l'espace horizontal basé sur le em (voir Section 19.1 [`\enspace` & `\quad` & `\qquad`], page 191).

mu Finalement, en mode mathématique, beaucoup de définitions sont exprimées dans l'unité mathématique *mu* donnée par 1 em = 18 mu, où le em est pris de la famille courante des symboles mathématiques. Voir Section 16.6 [Spacing in math mode], page 178.

L'utilisation de ces unités peut aider améliorer le fonctionnement d'une définition lors des changements de fonte. Par exemple, il est plus probable qu'une définition de l'espace verticale entre les articles d'une liste donnée comme `\setlength{\itemsep}{1ex plus 0.05ex minus 0.01ex}` reste raisonnable si la fonte est changée que si elle était exprimée en points.

14.2 `\setlength`

Synopsis :

```
\setlength{\longueur}{quantité}
```

La commande `\setlength` règle la longueur `\longueur` à *quantité*. Le nom `\longueur` doit être une séquence de contrôle (voir Section 12.1.1 [Control sequences], page 126), et donc commencer par une contrôblique `\` dans les circonstances normales. La *quantité* peut être une longueur élastique (voir Chapitre 14 [Lengths], page 141). Elle peut être positive, négative ou nulle, et peut être exprimée dans n'importe quelle unité comprise par L^AT_EX (voir Section 14.1 [Units of length], page 142).

Ci-dessous, avec les réglages par défaut de L^AT_EX le premier paragraphe est renforcé alors que le second ne l'est pas.

```
J'ai dit au docteur que je m'étais cassé la jambe à deux endroits.
```

```
\setlength{\parindent}{0em}
```

```
Il me répondit d'arrêter d'aller à ces endroits.
```

Si vous n'avez pas déclaré `\longueur` avec `\newlength`, par exemple si vous faites une faute de frappe dessus comme dans `\newlength{\specparindent}\setlength{\sepccparindent}{...}`, alors vous obtiendrez une erreur du genre de 'Undefined control sequence. <argument> \sepcindent'. Si vous oubliez la contrôblique au début du nom de la longueur alors vous obtiendrez une erreur du genre de 'Missing number, treated as zero.'

14.3 `\addtolength`

Synopsis :

```
\addtolength{\longueur}{quantité}
```

Incrémente la longueur `\longueur` par *quantité*. Le nom `\longueur` doit être une séquence de contrôle (voir Section 12.1.1 [Control sequences], page 126), et donc commencer par une controblque `\` dans les circonstances normales. La *quantité* peut être une longueur élastique (voir Chapitre 14 [Lengths], page 141). Elle peut être positive, négative ou nulle, et peut être exprimée dans n'importe quelle unité comprise par L^AT_EX (voir Section 14.1 [Units of length], page 142).

Dans l'exemple ci-dessous, si `\parskip` commence avec la valeur `0pt plus 1pt`

```
Docteur : comment va le garçon qui a avalé une monnaie en argent ?
\addtolength{\parskip}{1pt}
```

```
Infirmière : aucun changement.
```

alors il a la valeur `1pt plus 1pt` pour le second paragraphe.

Si vous n'avez pas déclaré la longueur `\longueur` avec `\newlength`, par exemple si vous faites une faute de frappe dessus comme dans `\newlength{\specparindent}\addtolength{\sepcparindent}<argument>\sepcindent`. Si c'est la *quantité* qui utilise une longueur qui n'a pas été déclarée, par exemple si vous faites la faute de frappe comme cela `\addtolength{\specparindent}{0.6\praindent}`, alors vous obtenez une erreur du genre de 'Undefined control sequence. <argument> \praindent'. Si vous oubliez la controblque au début du nom de la longueur, comme dans `\addtolength{parindent}{1pt}`, alors vous obtiendrez quelque chose du genre de 'You can't use 'the letter p' after \advance'.

14.4 `\settodepth`

Synopsis :

```
\settodepth{\longueur}{texte}
```

La commande `\settodepth` règle la longueur `\longueur` à la profondeur de la boîte que L^AT_EX obtient en composant l'argument `texte`. Le nom `\longueur` doit être une séquence de contrôle (voir Section 12.1.1 [Control sequences], page 126), et donc commencer par une controblque `\` dans les circonstances normales.

L'exemple ci-dessous imprime la profondeur maximale des descendantes de l'alphabet :

```
\newlength{\alphabetdepth}
\settodepth{\alphabetdepth}{abcdefghijklmnopqrstuvwxyz}
\the\alphabetdepth
```

Si vous n'avez pas déclaré la longueur `\longueur` avec `\newlength`, par exemple si vous faites une faute de frappe dans l'exemple ci-dessus comme dans `\settodepth{\aplphabetdepth}{abc...}`, alors vous obtiendrez quelque chose du genre de 'Undefined control sequence. <argument> \aplphabetdepth'. Si vous oubliez la controblque au début du nom de la longueur, comme dans `\settodepth{alphabetdepth}{...}` alors vous obtiendrez quelque chose du genre de 'Missing number, treated as zero. <to be read again> \setbox'.

14.5 `\settoheight`

Synopsis :

```
\settoheight{\longueur}{texte}
```

La commande `\settoheight` règle la longueur `\longueur` à la hauteur de la boîte que \LaTeX obtient en composant l'argument `texte`. Le nom `\longueur` doit être une séquence de contrôle (voir Section 12.1.1 [Control sequences], page 126), et donc commencer par une contrôblique `\` dans les circonstances normales.

L'exemple suivant imprime la hauteur maximale des ascendantes de l'alphabet ASCII bas de casse :

```
\newlength{\alphabetheight}
\settoheight{\alphabetheight}{abcdefghijklmnopqrstuvwxyz}
\the\alphabetheight
```

Si vous n'avez pas déclaré la longueur `\longueur` avec `\newlength`, par exemple si vous faites une faute de frappe dans l'exemple ci-dessus comme dans `\settoheight{\aplphabetheight}{abc...}`, alors vous obtiendrez quelque chose du genre de 'Undefined control sequence. <argument> \aplphabetdepth'. Si vous oubliez la contrôblique au début de `\longueur`, comme dans `\settoheight{alphabetheight}{...}` alors vous obtiendrez quelque chose du genre de 'Missing number, treated as zero. <to be read again> \setbox'.

14.6 `\settowidth`

Synopsis :

```
\settowidth{\longueur}{texte}
```

La commande `\settowidth` règle la longueur `\longueur` à la largeur de la boîte que \LaTeX obtient en composant l'argument `texte`. Le nom `\longueur` doit être une séquence de contrôle (voir Section 12.1.1 [Control sequences], page 126), et donc commencer par une contrôblique `\` dans les circonstances normales.

L'exemple suivant imprime la largeur de l'alphabet ASCII bas de casse :

```
\newlength{\alphabetwidth}
\settowidth{\alphabetwidth}{abcdefghijklmnopqrstuvwxyz}
\the\alphabetwidth
```

Si vous n'avez pas déclaré la longueur `\longueur` avec `\newlength`, par exemple si vous faites une faute de frappe dans l'exemple ci-dessus comme dans `\settowidth{\aplphabetwidth}{abc...}`, alors vous obtiendrez quelque chose du genre de 'Undefined control sequence. <argument> \aplphabetwidth'. Si vous oubliez la contrôblique au début de `\longueur`, comme dans `\settowidth{alphabetwidth}{...}` alors vous obtiendrez quelque chose du genre de 'Missing number, treated as zero. <to be read again> \setbox'.

14.7 Expressions

Synopsis, l'un parmi :

```
\numexpr expression
\dimexpr expression
```

```
\glueexpr expression
\muglue expression
```

En tout lieu où vous pourriez écrire un entier, une dimen, une muglue ou une glue de \TeX , vous pouvez à la place écrire une expression pour écrire ce genre de quantité.

Un exemple est que `\the\dimexpr\linewidth-4pt\relax` produit en sortie la longueur égale à quatre points de moins que la largeur de la ligne (le seul usage de `\the` est d'afficher le résultat dans le document). De même, `\romannumeral\numexpr6+3\relax` produit 'ix', et `\the\glueexpr 5pt plus 1pt * 2 \relax` produit '10.0pt plus 2.0pt'.

Une commodité ici par rapport à effectuer les calculs en allouant des registres et en utilisant `\advance`, etc., est que l'évaluation d'expression n'implique par d'affectations et peut donc être effectuée à des endroits où les affectations ne sont pas autorisées. L'exemple suivant calcule la largeur de la `\parbox`.

```
\newlength{\offset}\setlength{\offset}{2em}
\begin{center}
\parbox{\dimexpr\linewidth-\offset*3}{Sans animosité envers
quiconque, charitables envers tous, et sûrs de notre droit en tant que
Dieu nous en accorde conscience, mettons-nous à l'œuvre afin d'achever
la tâche qui nous occupe, de panser les blessures de notre nation, de
porter soin à l'homme qui a affronté le combat et soulagement à sa veuve
et à son orphelin, enfin de faire tout pour réaliser et honorer une paix
juste et durable entre nous et avec toutes les nations. --- Abraham
Lincoln, second discours d'investiture, inscrit dans le mémorial
Lincoln}
\end{center}
```

L'expression consiste en un ou plusieurs termes du même type (entier, dimension, etc.) qui sont ajoutés ou soustraits. Un terme est un type de nombre, dimension, etc., et consiste en un facteur de ce type, optionnellement multiplié ou divisé par des facteurs. Un facteur d'un type est soit une quantité de ce type ou une sous-expression parenthésés. L'expression produit un résultat du type donné, de sorte que `\numexpr` produit un entier, `\dimexpr` produit une dimension dimension, etc.

Dans l'exemple de citation donné plus haut, changer l'expression en `\dimexpr\linewidth-3*\offset` produit l'erreur `Illegal unit of measure (pt inserted)`. La raison en est que pour `\dimexpr` et `\glueexpr`, l'entrée consiste en une valeur de dimension ou de glue suivie par un facteur multiplicatif optionnel, et non l'inverse. Ainsi `\the\dimexpr 1pt*10\relax` est valide et produit '10.0pt', mais `\the\dimexpr 10*1pt\relax` produit l'erreur `Illegal unit`.

Les expressions absorbent les unités lexicales et effectuent les opérations mathématiques appropriées jusqu'à ce qu'un `\relax` (qui est absorbé), ou jusqu'à ce que la première unité lexicale non valide soit rencontrée. Ainsi, `\the\numexpr2+3px` imprime '5px', parce que \LaTeX lit le `\numexpr2+3`, ce qui est composé de nombres, et ensuite trouve la lettre `p`, qui ne peut pas faire partie d'un nombre. Il termine alors l'expression et produit le '5', suivi par le texte ordinaire 'px'.

Ce comportement de terminaison est utile dans les comparaisons. Dans `\ifnum\numexpr\parindent*2 < 10pt Oui\else Non\fi`, le signe inférieur à termine l'expression et le résultat est 'Non' (dans un document de classe \LaTeX standard article).

Les expressions peuvent utiliser les opérateurs `+`, `-`, `*` et `/` ainsi que les parenthèses pour les sous-expressions, `(...)`. Dans les expressions glue les parties en `plus` et `moins` ne nécessitent pas de parenthèses pour être affectés par un facteur. Ainsi le résultat de `\the\glueexpr 5pt plus 1pt * 2 \relax` est `'10pt plus 2pt'`.

`TEX` convertit les autres types numériques de la même façon que lorsqu'il fait une affectation à un registre. Ainsi le résultat de `\the\numexpr\dimexpr 1pt\relax\relax` est `'65536'`, ce qui est `1pt` exprimé en points proportionnés (voir [units of length sp], page 143, l'unité interne de `TEX`) et ensuite converti en entier. Si ça avait été une `\glueexpr`, on aurait laissé tomber la dilatation et la contraction. Dans l'autre sens, une `\numexpr` au sein d'une `\dimexpr` ou d'une `\glueexpr` nécessite l'ajout d'unité appropriées, comme dans `\the\dimexpr\numexpr 1 + 2\relax pt\relax`, ce qui produit `'3.0pt'`.

Voici les détails de l'arithmétique : chaque facteur est vérifié comme étant compris dans l'intervalle autorisé, les nombres doivent être inférieurs à 2^{31} en valeur absolue, et les composantes de dimensions ou glues doivent être inférieures à 2^{14} points, ou `mu`, ou `fil`, etc. Les opérations arithmétiques sont effectuées individuellement, sauf pour les opérations de dilatation (une multiplication immédiatement suivie d'une division) qui sont faites comme une opération combinée avec un produit sur 64-bit comme valeur intermédiaire. Le résultat de chaque opération est de nouveau vérifié comme appartenant à l'intervalle autorisé.

Finalement, on notera que les divisions et dilatations sont faites avec un arrondi au plus proche (contrairement à l'opération `\divide` de `TEX` qui prend un arrondi vers zéro). Ainsi `\the\dimexpr 5pt*(3/2)\relax` met `'10.0pt'` dans le document, parce qu'il arrondit $3/2$ en 2, tandis que `\the\dimexpr 5pt*(4/3)\relax` produit `'5.0pt'`.

15 Faire des paragraphes

Pour démarrer un paragraphe, tapez juste du texte. Pour finir le paragraphe courant, mettez une ligne vide. Dans l'exemple ci-dessous on a trois paragraphes, dont la séparation est assurée par deux lignes vides :

C'est une vérité universellement reconnue qu'un célibataire pourvu d'une belle fortune doit avoir envie de se marier, et si peu que l'on sache de son sentiment à cet égard, lorsqu'il arrive dans une nouvelle résidence, cette idée est si bien fixée dans l'esprit de ses voisins qu'ils le considère sur-le-champ comme la propriété légitime de l'une ou l'autre de leurs filles.

« Savez-vous mon cher ami, dit un jour Mrs Bennet à son mari, que Netherfield Park est enfin loué ? »

Mr Bennet répondit qu'il l'ignorait.

Le séparateur de paragraphe peut consister en une séquence quelconque d'au moins une ligne à blanc, dans laquelle au moins une ligne n'est pas terminée par un commentaire. Une ligne à blanc est une ligne qui est vide ou ne contient que des caractères blancs tel que l'espace ou la tabulation. Les commentaires dans le code source sont démarrés par un % et s'étendent jusqu'à la fin de la ligne. Dans l'exemple suivant les deux colonnes sont identiques :

```
\documentclass[twocolumn]{article}
\begin{document}
First paragraph.

Second paragraph.
\newpage
First paragraph.

% les lignes de séparation peuvent contenir des blancs.

Second paragraph.
\end{document}
```

Une fois que L^AT_EX a rassemblé tout le contenu d'un paragraphe il divise le paragraphe en lignes d'une manière qui est optimisée sur l'entièreté du paragraphe (voir Chapitre 9 [Line breaking], page 111).

Il y a des endroits où un nouveau paragraphe n'a pas le droit de commencer. Ne mettez pas de lignes à blanc en mode math (voir Chapitre 17 [Modes], page 185) ; dans l'exemple suivant la ligne à blanc précédent `\end{equation}`

```
\begin{equation}
2^{|S|} > |S|

\end{equation}
```

produit l'erreur 'Missing \$ inserted'. De même, la ligne à blanc au sein de l'argument de ce `\section` :

```
\section{aaa
  

  bbb}
```

produit l'erreur 'Runaway argument? {aaa ! Paragraph ended before \@sect was complete}'.

15.1 `\par`

Synopsis (notez que, alors qu'il lit l'entrée, \TeX convertit toute séquence d'au moins une ligne à blanc en un `\par`, voir Chapitre 15 [Making paragraphs], page 148) :

```
\par
```

Termine le paragraphe en cours. La manière habituelle de séparer les paragraphes est avec une ligne à blanc, mais la commande `\par` est entièrement équivalente. Cette commande est robuste (voir Section 12.11 [`\protect`], page 134).

L'exemple suivant utilise `\par` plutôt qu'une ligne à blanc simplement pour la lisibilité.

```
\newcommand{\coursEnJargonLegal}{%
  \CeQuEstLaTricherie\par\CeQuiTArriveQuandOnTAttrape}
```

En mode LR la commande `\par` ne fait rien et est ignorée. En mode paragraphe la commande `\par` termine le mode paragraphe, et commute \LaTeX vers le mode vertical (voir Chapitre 17 [Modes], page 185).

Vous ne pouvez pas utiliser la commande `\par` en mode mathématique. Vous ne pouvez pas non plus l'utiliser au sein de l'argument de beaucoup de commandes, telles que les commandes de rubricage, par ex. `\section` (voir Chapitre 15 [Making paragraphs], page 148, et Section 12.1 [`\newcommand` & `\renewcommand`], page 124).

La commande `\par` ne fait pas la même chose que la commande `\paragraph`. Cette dernière, comme `\section` ou `\subsection`, est une commande de rubricage utilisée par les classes standards de document \LaTeX (voir Section 6.5 [`\subsubsection` & `\paragraph` & `\subparagraph`], page 48).

La commande `\par` ne fait pas la même chose que `\newline` ou que le saut de ligne en double controblique, `\\`. La différence c'est que `\par` termine le paragraphe, et non pas seulement la ligne, et ajoute également de l'espace vertical inter-paragraphe `\parskip` (voir Section 15.3 [`\parindent` & `\parskip`], page 150).

En sortie de l'exemple suivant

```
xyz
  

  \setlength{\parindent}{8cm}
  \setlength{\parskip}{13cm}
  \noindent test\indent test1\par test2
```

on a : après 'xyz' il y a un saut vertical de 13 cm et ensuite 'test' apparaît, aligné sur la marge de gauche. Sur la même ligne on a un espace horizontal vide de 8 cm et ensuite 'test1' apparaît. Finalement, il y a un espace vertical de 13 cm, suivi par un nouveau paragraphe avec un renforcement de 8 cm, et ensuite \LaTeX met le texte 'test2'.

15.2 `\indent` & `\noindent`

Synopsis:

```
\indent
```

ou

```
\noindent
```

Passes en mode horizontal (voir Chapitre 17 [Modes], page 185). La commande `\indent` commence par produire une boîte vide dont la largeur est `\parindent`. Ces commandes sont robustes (voir Section 12.11 [`\protect`], page 134).

D'ordinaire on crée un nouveau paragraphe en insérant une ligne à blanc. Voir Section 15.1 [`\par`], page 149, pour la différence entre cette commande et `\par`. Pour commencer un paragraphe sans renforcement, ou pour continuer un paragraphe interrompu, utiliser `\noindent`.

Au milieu d'un paragraphe la commande `\noindent` est sans effet, parce que \LaTeX y est déjà en mode horizontal. La commande `\indent` n'a pour seul effet que de produire une espace en sortie.

L'exemple suivant démarre un nouveau paragraphe.

```
... fin du paragraphe précédent.
```

```
\noindent Ce paragraphe n'est pas renforcé.
```

et cet exemple-là continue un paragraphe interrompu :

```
Les données
```

```
\begin{center}
  \begin{tabular}{rl} ... \end{tabular}
\end{center}
```

```
\noindent montrent clairement ceci.
```

Pour éliminer le renforcement de paragraphe dans tout un document, mettez `\setlength{\parindent}{0pt}` dans le préambule. Si vous faites cela, vous désirez peut-être régler également la longueur des espaces inter-paragraphes, `\parskip` (voir Section 15.3 [`\parindent` & `\parskip`], page 150).

Les styles \LaTeX par défaut ont le premier paragraphe suivant une rubrique qui n'est pas renforcé, comme cela est traditionnel en typographie anglo-saxonne. Pour changer cela, chercher sur le CTAN le paquetage `indentfirst`.

15.3 `\parindent` & `\parskip`

Synopsis :

```
\setlength{\parindent}{longueur-horizontale}
\setlength{\parskip}{longueur-verticale}
```

Toutes deux sont des longueurs élastiques (voir Chapitre 14 [Lengths], page 141). Elles affectent respectivement le renforcement des paragraphes ordinaires, non des paragraphes au sein de minipages (voir Section 8.18 [minipage], page 79), et l'espace vertical entre les paragraphes, respectivement.

Par exemple, si ce qui suit est mis dans le préambule :

```
\setlength{\parindent}{0em}
\setlength{\parskip}{1ex}
```

Alors le document aura des paragraphes qui ne sont pas renfoncés, mais par contre qui sont verticalement séparés d'environ la hauteur d'un 'x' bas-de-casse.

Dans les documents de classes L^AT_EX standards, la valeur par défaut de `\parindent` pour les documents à une colonne est `15pt` quand la taille par défaut du texte est `10pt`, `17pt` pour `11pt`, et `1.5em` pour `12pt`. Dans les documents en deux-colonnes c'est `1em`. (Ces valeurs sont réglées avant que L^AT_EX appelle `\normalfont` de sorte que `em` est dérivé de la fonte par défaut, Computer Modern. Si vous utilisez une fonte différente alors pour régler `\parindent` à `1em` en accord avec cette fonte, mettez dans votre préambule `\AtBeginDocument{\setlength{\parindent}{1em}}`).

La valeur par défaut de `\parskip` dans les documents de classes standards L^AT_EX est `0pt plus1pt`.

15.4 Notes en marge

Synopsis, l'un parmi :

```
\marginpar{droite}
\marginpar[gauche]{droite}
```

La commande `\marginpar` crée une note dans la marge. La première ligne de la note a la même ligne de base que la ligne dans le texte où le `\marginpar` se trouve.

Lorsque vous spécifiez seulement l'argument obligatoire *droite*, le texte est placé

- dans la marge de droite pour une disposition en recto simple (option `oneside`, voir Section 3.1 [Document class options], page 7) ;
- dans la marge extérieure pour une disposition en recto-verso (option `twoside`, voir Section 3.1 [Document class options], page 7) ;
- dans la plus proche pour une disposition à deux-colonnes (option `twocolumn`, voir Section 3.1 [Document class options], page 7).

La commande `\reversemarginpar` place les notes en marge suivante sur la marge opposée (interne). `\normalmarginpar` les replace dans la position par défaut.

Lorsque vous spécifier les deux arguments, *gauche* est utilisé pour is used la marge de gauche, et *droite* est utilisé pour la marge de is used droite.

Le premier mot doit normalement ne pas être sujet à une césure ; Vous pouvez activer la césure à cet endroit en commençant le nœud avec `\hspace{0pt}`.

Ces paramètres affectent le formatage de la note :

```
\marginparpush
```

Espace verticale minimale entre les notes ; par défaut '`7pt`' pour des documents à '`12pt`', '`5pt`' sinon.

```
\marginparsep
```

Espace horizontale entre le texte principal et la note ; par défaut '`11pt`' pour des documents à '`10pt`', '`10pt`' sinon.

`\marginparwidth`

Largeur de la note même ; la valeur par défaut est ‘90pt’ pour un document en recto simple et à ‘10pt’, ‘83pt’ pour ‘11pt’, et ‘68pt’ pour ‘12pt’ ; ‘17pt’ de plus dans chacun de ces cas pour un document recto-verso. En mode deux-colonnes, la valeur par défaut est ‘48pt’.

La routine standard de \LaTeX pour les notes en marge n’empêche pas les notes de déborder hors du bas de la page.

16 Formules mathématiques

On produit du texte mathématique en mettant L^AT_EX en mode math ou mode math d’affichage (voir Chapitre 17 [Modes], page 185). L’exemple qui suit illustre les deux :

```
L'équation d'onde de \ ( u \ ) est :
\begin{displaymath}
\frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u
\end{displaymath}
où \ ( \nabla^2 \ ) est le Laplacien spatial et \ ( c \ ) une constante.
```

Le mode math est pour des mathématiques en ligne dans le texte. Dans l’exemple ci-dessus il est invoqué pour démarrer avec le `\` (et pour finir avec le `\`) correspondant. Le mode math d’affichage est pour des équations hors texte et il est ici invoqué par l’environnement `displaymath`. Notez que tout texte mathématique, y compris consistant d’un seul caractère, est géré en mode math.

Quand il est en math mode ou mode math d’affichage, L^AT_EX gère beaucoup d’aspects du texte que vous entrez différemment que dans les autres mode texte. Par exemple,

```
comparez x+y et \ ( x+y \ )
```

en mode math les lettres sont en italique et l’espace autour du signe plus est différent.

Il y a trois manières de mettre L^AT_EX en mode math pour fabriquer des formules en ligne dans le texte :

```
\ ( matière mathématique \ )
$ matière mathématique $
\begin{math} matière mathématique \end{math}
```

La première forme est préférée et la deuxième est assez fréquente, mais la troisième est rarement utilisée. Vous pouvez utiliser n’importe laquelle des trois, comme dans `\(x\)` and `y`. Vous pouvez les utiliser en mode paragraphe ou en mode LR (voir Chapitre 17 [Modes], page 185).

Pour fabriquer des formules hors texte, faites passer L^AT_EX en mode math d’affichage avec l’un des deux environnements suivants :

```
\begin{displaymath}
  matière mathématique
\end{displaymath}
```

ou

```
\begin{equation}
  matière mathématique
\end{equation}
```

(voir Section 8.5 [displaymath], page 63, voir Section 8.9 [equation], page 66). La seule différence est qu’avec l’environnement `equation`, L^AT_EX place un numéro de formule sur le côté de la formule. La construction `\[math \]` est équivalente à `\begin{displaymath} math \end{displaymath}`. Ces environnements ne peuvent être utilisés qu’en mode paragraphe (voir Chapitre 17 [Modes], page 185).

La American Mathematical Society a publié un ensemble de paquetages libres d’accès qui étendent largement vos possibilités pour écrire des mathématiques, `amsmath` et `amssymb` (et notez également le paquetage `mathtools` qui est une extension d’`amsmath` et le charge). Les

nouveaux documents contenant du texte mathématique devraient utiliser ces paquetages. Une description des ces paquetages est hors du champ d'application de ce document ; se référer à leur documentation sur le CTAN.

16.1 Indices inférieurs & supérieurs

Synopsis (en mode math en ligne ou d'affichage), l'un parmi :

```
baseexp
base{exp}
```

ou, l'un parmi :

```
baseexp
base{exp}
```

Met *exp* en indice supérieur de *base* (avec le caractère caret `^`) ou inférieur (avec le tiret bas `_`).

Dans cet exemple le 0 et le 1 apparaissent comme indices inférieurs alors que le 2 est un indice supérieur.

```
\( (x_0+x_1)^2 \leq (x_0)^2+(x_1)^2 \)
```

Pour avoir plus d'un caractère au sein de l'indice inférieur ou supérieur *exp*, entourez l'expression *exp* par des des accolades comme dans e^{-2x} . La quatrième ligne de l'exemple suivant illustre l'utilisation d'accolades pour grouper une expression pour l'indice supérieur.

```
\begin{displaymath}
(3^3)^3=27^3=19\,683
\quad
3^{\{(3^3)\}}=3^{\{27\}}=7\,625\,597\,484\,987
\end{displaymath}
```

L^AT_EX sait gérer un indice supérieur sur un indice inférieur, ou inférieur sur inférieur, ou supérieur sur inférieur, ou inférieur sur supérieur. Ainsi des expressions telles que $e^{\{x^2\}}$ et $x_{\{a_0\}}$ seront composées comme il faut. Notez l'utilisation d'accolade pour donner à *base* un indice *exp* déterminé. Si vous saisissez $\{(3^3^3)\}$, c'est interprété comme $\{(3^{\{3\}}^{\{3\}})\}$ et vous obtiendrez l'erreur T_EX 'Double superscript'.

L^AT_EX fait ce qu'il faut quand quelque chose a à la fois un indice inférieur et un indice supérieur. Dans cet exemple l'intégrale a les deux, ils sont rendus à la bonne place sans intervention de l'auteur.

```
\begin{displaymath}
\int_{\{x=a\}}^{\{b\}} f'(x)\,dx = f(b)-f(a)
\end{displaymath}
```

Notez les accolades autour de $x=a$ pour faire de toute l'expression un indice inférieur.

Vous pouvez mettre un indice supérieur ou inférieur avant un symbole avec une construction telle que $\}_{t} K^2$. Les accolades vides $\{\}$ donnent à l'indice inférieur quelque chose sur quoi s'attacher et l'empêchent d'être attaché par accident à quelque symbole qui le précède dans l'expression.

En dehors du mode math, ou math d'affichage, l'utilisation d'un indice inférieur ou supérieur, comme dans l'expression x^2 , vous vaudra l'erreur T_EX 'Missing \$ inserted'.

Une raison usuelle de désirer des indices inférieurs hors d'un mode math est de composer des formules chimiques. Il y a des paquetages spécialisés pour cela, tels que `mhchem` ; voir le CTAN.

16.2 Symboles mathématiques

\LaTeX fournit presque tout symbole mathématique dont vous êtes susceptible d'avoir besoin. Par exemple, si vous incluez `\pi` dans votre code source, alors vous obtenez le symbole π . Voir le paquetage « Comprehensive \LaTeX Symbol List », <https://ctan.org/pkg/comprehensive>.

Ci-dessous se trouve une liste de symboles usuels. Ce n'est en aucun cas une liste exhaustive. Chaque symbole y est décrit brièvement et avec sa classe de symbole (qui détermine l'espacement autour de lui) qui est donnée entre parenthèses. Les commandes pour ces symboles peuvent, sauf mention contraire, être utilisées seulement en mode mathématique. Pour redéfinir une commande de sorte à pouvoir l'utiliser quel que soit le mode, voir Section 17.1 [`\ensuremath`], page 186.

<code>\ </code>	\parallel Parallèle (relation). Synonyme : <code>\parallel</code> .
<code>\aleph</code>	\aleph Aleph, cardinal transfini (ordinaire).
<code>\alpha</code>	α Lettre grecque en bas de casse alpha (ordinaire).
<code>\amalg</code>	\amalg Union disjointe (binaire)
<code>\angle</code>	\angle Angle géométrique (ordinaire). Similaire : signe inférieur-à $<$ et chevron <code>\langle</code> .
<code>\approx</code>	\approx Presque égal à (relation).
<code>\ast</code>	$*$ Opérateur astérisque, convolution, à six branches (binaire). Synonyme : <code>*</code> , ce qui est souvent un indice supérieur ou inférieur, comme dans l'étoile de Kleene. Similaire : <code>\star</code> , qui est à cinq branches, et est quelquefois utilisée comme une opération binaire générale, et quelquefois réservée pour la corrélation croisée.
<code>\asymp</code>	\asymp Équivalent asymptotiquement (relation).
<code>\backslash</code>	\backslash controblique (ordinaire). Similaire : soustraction d'ensemble <code>\setminus</code> , et <code>\textbackslash</code> pour une controblique en dehors du mode mathématique.
<code>\beta</code>	β Lettre grecque en bas de casse bêta (ordinaire).
<code>\bigcap</code>	\bigcap De taille Variable, ou n-aire, intersection (opérateur). Similaire : intersection binaire <code>\cap</code> .
<code>\bigcirc</code>	\bigcirc Cercle, plus grand (binaire). Similaire : composition de fonction <code>\circ</code> .
<code>\bigcup</code>	\bigcup De taille variable, ou n-aire, union (opérateur). Similaire : union binaire <code>\cup</code> .
<code>\bigodot</code>	\bigodot De taille variable, ou n-aire, opérateur point encerclé (opérateur).
<code>\bigoplus</code>	\bigoplus De taille variable, ou n-aire, opérateur plus encerclé (opérateur).
<code>\bigotimes</code>	\bigotimes De taille variable, ou n-aire, opérateur multiplié encerclé (opérateur).

<code>\bigtriangledown</code>	∇ De taille variable, ou n-aire, triangle ouvert pointant vers le bas (opérateur).
<code>\bigtriangleup</code>	\triangle De taille variable, ou n-aire, triangle ouvert pointant vers le haut (opérateur).
<code>\bigsqcup</code>	\sqcup De taille variable, ou n-aire, union carrée (opérateur).
<code>\biguplus</code>	\uplus De taille variable, ou n-aire, opérateur union avec un plus (opérateur). (Notez que le nom a juste un p.)
<code>\bigvee</code>	\vee De taille variable, ou n-aire, ou-logique (opérateur).
<code>\bigwedge</code>	\wedge De taille variable, ou n-aire, et-logique (opérateur).
<code>\bot</code>	\bot Taquet vers le haut, minimum, plus petit élément d'un ensemble partiellement ordonné, ou une contradiction (ordinaire). Voir aussi <code>\top</code> .
<code>\bowtie</code>	\bowtie Jonction naturelle de deux relations (relation).
<code>\Box</code>	\Box Opérateur modal pour la nécessité ; boîte ouverte carrée (ordinaire). Ceci n'est pas disponible en T _E X de base. En L ^A T _E X vous devez charger le paquetage <code>amssymb</code> .
<code>\bullet</code>	• Puce (binaire). Similaire : point de multiplication <code>\cdot</code> .
<code>\cap</code>	\cap Intersection de deux ensembles (binaire). Similaire : opérateur de taille variable <code>\bigcap</code> .
<code>\cdot</code>	· Multiplication (binaire). Similaire : Point puce <code>\bullet</code> .
<code>\chi</code>	χ Lettre grecque en bas de casse chi (ordinaire).
<code>\circ</code>	○ Composition de fonctions, opérateur rond (binaire). Similaire : opérateur de taille variable <code>\bigcirc</code> .
<code>\clubsuit</code>	♣ Trèfle de carte à jouer (ordinaire).
<code>\complement</code>	\complement Complément d'ensemble, utilisée en indice supérieur comme dans S^{\complement} (ordinaire). Ceci n'est pas disponible en T _E X de base. En L ^A T _E X vous devez charger le paquetage <code>amssymb</code> . On utilise aussi : $S^{\mathbf{\complement}}$ ou \bar{S} .
<code>\cong</code>	\cong Congruent (relation).
<code>\coprod</code>	\coprod Coproduit (opérateur).
<code>\cup</code>	\cup Réunion de deux ensembles (binaire). opérateur de taille variable <code>\bigcup</code> .
<code>\dagger</code>	† Relation obèle (binaire).
<code>\dashv</code>	⊥ Taquet gauche, tiret avec barre verticale à droite, tourniquet pour gauchers (relation). Similaire : taquet droit, tourniquet <code>\vdash</code> .

<code>\ddagger</code>	‡ Relation double obèle (binaire).
<code>\Delta</code>	Δ Delta grec capital, utilisé pour indiquer une incrémentation (ordinaire).
<code>\delta</code>	δ Delta grec bas-de-casse (ordinaire).
<code>\Diamond</code>	<code>\Diamond</code> Opérateur grand carreau (ordinaire). Ceci n'est pas disponible en \TeX de base. En \LaTeX vous devez charger le paquetage <code>amssymb</code> .
<code>\diamond</code>	\diamond Opérateur Carreau, ou puce carreau (binaire). Similaire : grand carreau <code>\Diamond</code> , puce disque <code>\bullet</code> .
<code>\diamondsuit</code>	\diamond Carreau de carte à jouer (ordinaire).
<code>\div</code>	\div Signe de division, obélus (binaire).
<code>\doteq</code>	\doteq Approche la limite (relation). Similaire : géométriquement égal à <code>\Doteq</code> .
<code>\downarrow</code>	\downarrow Flèche vers le bas, converge (relation). Similaire : flèche à ligne double vers le bas <code>\Downarrow</code> .
<code>\Downarrow</code>	\Downarrow Flèche à ligne double vers le bas (relation). Similaire : flèche à ligne simple vers le bas <code>\downarrow</code> .
<code>\ell</code>	ℓ Lettre l bas de casse cursive (ordinaire).
<code>\emptyset</code>	\emptyset Symbole ensemble vide (ordinaire). La forme en variante est <code>\varnothing</code> .
<code>\epsilon</code>	ϵ Lettre grecque bas de casse epsilon lunaire (ordinaire). Semblable à la lettre en texte grec. En mathématiques on utilise plus fréquemment l'épsilon minuscule de ronde <code>\varepsilon</code> . En relation avec : la relation ensembliste « appartenir à » <code>\in</code> .
<code>\equiv</code>	\equiv Équivalence (relation).
<code>\eta</code>	η Lettre grecque en bas de casse (ordinaire).
<code>\exists</code>	\exists Quanteur « Il existe » (ordinaire).
<code>\flat</code>	\flat Bémol (ordinaire).
<code>\forall</code>	\forall Quanteur « Pour tout » (ordinaire).
<code>\frown</code>	\frown Moue (ordinaire).
<code>\Gamma</code>	Γ Lettre grecque en capitale (ordinaire).
<code>\gamma</code>	γ Lettre grecque en bas de casse (ordinaire).
<code>\ge</code>	\geq Supérieur ou égal à (relation). Synonyme : <code>\geq</code> .
<code>\geq</code>	\geq Supérieur ou égal à (relation). Synonyme : <code>\ge</code> .
<code>\gets</code>	\leftarrow Auquel est assignée la valeur (relation). Synonyme : <code>\leftarrow</code> .
<code>\gg</code>	\gg Beaucoup plus grand que (relation). Similaire : Beaucoup plus petit que <code>\ll</code> .

<code>\hbar</code>	\hbar Constante de Planck sur deux pi (ordinaire).
<code>\heartsuit</code>	♥ Cœur de carte jouer (ordinaire).
<code>\hookleftarrow</code>	↶ Flèche vers la gauche avec crochet (relation).
<code>\hookrightarrow</code>	↷ Flèche vers la gauche avec crochet (relation).
<code>\iff</code>	↔ Si et seulement si (relation). C'est un <code>\Longleftarrow</code> avec un <code>\thickmuskip</code> de chaque côté.
<code>\Im</code>	ℑ Partie imaginaire (ordinaire). Voir aussi : partie réelle <code>\Re</code> .
<code>\imath</code>	i sans point ; utilisé quand on place un accent sur un i (voir Section 16.4 [Math accents], page 176).
<code>\in</code>	∈ Appartient à (relation). Voir aussi : lettre grecque bas de casse epsilon lunaire <code>\epsilon</code> et l'epsilon minuscule de ronde <code>\varepsilon</code> .
<code>\infty</code>	∞ Infini (ordinaire).
<code>\int</code>	∫ Intégrale (opérateur).
<code>\iota</code>	ι Lettre grecque en bas de casse (ordinaire).
<code>\Join</code>	<code>\Join</code> Symbole jointure condensé (relation). Ceci n'est pas disponible en <code>T_EX</code> de base.
<code>\jmath</code>	j sans point ; utilisé quand on place un accent sur un j (voir Section 16.4 [Math accents], page 176).
<code>\kappa</code>	κ Lettre grecque en bas de casse (ordinaire).
<code>\Lambda</code>	Λ Lettre grecque en capitale (ordinaire).
<code>\lambda</code>	λ Lettre grecque en bas de casse (ordinaire).
<code>\land</code>	∧ Et logique (binaire). Synonyme : <code>\wedge</code> . Voir aussi ou-logique <code>\lor</code> .
<code>\langle</code>	⟨ Chevron mathématique gauche, ou séquence, crochet (ouvrant). Similaire : inférieur à <code><</code> . Correspond à <code>\rangle</code> .
<code>\lbrace</code>	{ Accolade gauche (ouvrant). Synonyme : <code>\{</code> . Correspond à <code>\rbrace</code> .
<code>\lbrack</code>	[Crochet gauche (ouvrant). Synonyme : <code>[</code> . Correspond à <code>\rbrack</code> .
<code>\lceil</code>	⌈ Plafond à gauche, ressemble à un crochet mais avec le pied coupé (ouvrant). Correspond à <code>\rceil</code> .
<code>\le</code>	≤ Inférieur ou égal à (relation). C'est un synonyme de <code>\leq</code> .
<code>\leadsto</code>	<code>\leadsto</code> flèche vers la droite en tire-bouchon (relation). Ceci n'est pas disponible en <code>T_EX</code> de base. En <code>L^AT_EX</code> vous devez charger le paquetage <code>amssymb</code> .
<code>\Leftarrow</code>	⇐ Est impliqué par, flèche vers la gauche à double-ligne (relation). Similaire : flèche vers la gauche à simple ligne <code>\leftarrow</code> .

- `\leftarrow`
 \leftarrow Flèche vers la gauche à simple ligne (relation). Synonyme : `\gets`. Similaire :
 Flèche vers la gauche à double-ligne `\Leftarrow`.
- `\leftharpoondown`
 \leftharpoondown Harpon vers la gauche à simple ligne avec crochet vers le bas (relation).
- `\leftharpoonup`
 \leftharpoonup Harpon vers la gauche à simple ligne avec crochet vers le haut (relation).
- `\Leftrightarrow`
 \Leftrightarrow Bi-implication ; double flèche bilatérale (relation). Similaire : flèche bi-
 latérale `\leftrightharpoonup`.
- `\leftrightharpoonup`
 \leftrightharpoonup Flèche bilatérale (relation). Similaire : double flèche bi-
 latérale `\Leftrightarrow`.
- `\leq` \leq Inférieur ou égal à (relation). Synonyme : `\le`.
- `\lfloor` \lfloor Plancher à gauche (ouvrant). Correspond à : `\rfloor`.
- `\lhd` \lhd Pointe de flèche, c.-à-d. triangle pointant vers la gauche (binaire).
 Pour le symbole normal de sous-groupe on doit charger `amssymb` et
 utiliser `\vartriangleleft` (qui est une relation et ainsi donne un meilleur
 espacement).
- `\ll` \ll Beaucoup plus petit que (relation). Similaire : Beaucoup plus grand que `\gg`.
- `\lnot` \lnot Négation logique (ordinaire). Synonyme : `\neg`.
- `\longleftarrow`
 \longleftarrow Longue flèche vers la gauche à ligne simple (relation). Similaire : longue
 flèche gauche à ligne double `\Longleftarrow`.
- `\longleftrightharpoonup`
 \longleftrightharpoonup Longue flèche bilatérale à ligne simple (relation). Similaire : longue flèche
 bilatérale à ligne double `\Longleftrightharpoonup`.
- `\longmapsto`
 \longmapsto Longue flèche d'un taquet vers la droite à ligne simple (relation). Similaire :
 version courte `\mapsto`.
- `\longrightarrow`
 \longrightarrow Longue flèche vers la droite à ligne simple (relation). Similaire : longue
 flèche vers la droite à ligne double `\Longrightarrow`.
- `\lor` \vee Ou-logique (binaire). Synonyme : `\vee`. Voir aussi et-logique `\land`.
- `\mapsto` \mapsto Flèche d'un taquet vers la droite (relation). Similaire : version
 longue `\longmapsto`.
- `\mho` \mho Conductance, symbole Ohm culbuté (ordinaire). Ceci n'est pas disponible
 en \TeX de base. En \LaTeX vous devez charger le paquetage `amssymb`.

<code>\mid</code>	Barre verticale à ligne simple (relation). une utilisation typique de <code>\mid</code> est pour un ensemble $\{\,x \mid x \geq 5 \, \}$. Similaire : <code>\vert</code> et <code> </code> produisent le même symbole de barre verticale à ligne simple mais sans aucun espacement (on les classe comme « ordinaires ») et vous ne devriez pas les utiliser comme symbole de relation mais seulement comme des ordinaux, c.-à-d. comme des symboles de note en bas de page. Pour dénoter une valeur absolue, voir l'entrée de <code>\vert</code> et pour une norme voir l'entrée de <code>\Vert</code> .
<code>\models</code>	\models inclut ; portillon/tourniquet à double barre, barre verticale et double tiret court (relation). Similaire : barre verticale et double tiret long <code>\vDash</code> .
<code>\mp</code>	\mp Moins ou plus (relation).
<code>\mu</code>	μ Lettre grecque en bas de casse (ordinaire).
<code>\nabla</code>	∇ Del de Hamilton, ou nabla, ou différentiel, opérateur (ordinaire).
<code>\natural</code>	\natural Bécarre (ordinaire).
<code>\ne</code>	\neq Non égal (relation). Synonyme : <code>\neq</code> .
<code>\nearrow</code>	\nearrow Flèche nord-est (relation).
<code>\neg</code>	\neg Négation logique (ordinaire). Synonyme : <code>\lnot</code> . Utilisé parfois au lieu du symbole de négation: <code>\sim</code> .
<code>\neq</code>	\neq Non égal (relation). Synonyme : <code>\ne</code> .
<code>\ni</code>	\ni Signe d'appartenance réfléchi ; « contient comme élément » (relation). Synonyme : <code>\owns</code> . Similaire : « appartient à » <code>\in</code> .
<code>\not</code>	\not Barre oblique longue utilisée pour rayer un opérateur la suivant (relation). Beaucoup opérateurs en négation qui n'ont pas besoin de <code>\not</code> sont disponibles, en particulier avec le paquetage <code>amssymb</code> . Par exemple, <code>\notin</code> est sans doute préférable à <code>\not\in</code> d'un point de vue typographique.
<code>\notin</code>	\notin N'appartient pas à (relation). Similaire : non contenu dans <code>\nsubseteq</code> .
<code>\nu</code>	ν Lettre grecque en bas de casse (ordinaire).
<code>\nwarrow</code>	\nwarrow Flèche nord-ouest (relation).
<code>\odot</code>	\odot Point cerclé (binaire). Similaire : opérateur de taille variable <code>\bigodot</code> .
<code>\oint</code>	\oint Intégrale de contour, intégrale avec un cercle au milieu (opérateur).
<code>\Omega</code>	Ω Lettre grecque en capitale (ordinaire).
<code>\omega</code>	ω Lettre grecque en bas de casse (ordinaire).
<code>\ominus</code>	\ominus Signe moins, ou tiret, cerclé (binaire).
<code>\oplus</code>	\oplus Signe plus cerclé (binaire). Similaire : opérateur de taille variable <code>\bigoplus</code> .
<code>\oslash</code>	\oslash Barre de fraction, ou barre oblique, cerclée (binaire).
<code>\otimes</code>	\otimes Signe de multiplication, ou croix, cerclé (binaire). Similaire : opérateur de taille variable <code>\bigotimes</code> .

<code>\owns</code>	\ni Epsilon d'appartenance réfléchi ; « contient comme élément » (relation). Synonyme : <code>\ni</code> . Similaire : appartient à <code>\in</code> .
<code>\parallel</code>	\parallel Parallèle (relation). Synonyme : <code>\ </code> .
<code>\partial</code>	∂ Différentielle partielle (ordinaire).
<code>\perp</code>	\perp Perpendiculaire (relation). Similaire : <code>\bot</code> utilise le même glyphe mais l'espacement est différent parce qu'il est dans la classe ordinaire.
<code>\Phi</code>	Φ Lettre grecque capitale (ordinaire).
<code>\phi</code>	ϕ Lettre grecque en bas de casse (ordinaire). La forme en variante est <code>\varphi</code> φ .
<code>\Pi</code>	Π Lettre grecque en capitale (ordinaire).
<code>\pi</code>	π Lettre grecque en bas de casse (ordinaire). La forme en variante est <code>\varpi</code> ϖ .
<code>\pm</code>	\pm Plus or minus (binaire).
<code>\prec</code>	\prec Précède (relation). Similaire : inférieur à <code><</code> .
<code>\preceq</code>	\preceq Précède ou égal, précède par-dessus signe égal à une ligne (relation). Similaire : inférieur ou égal <code>\leq</code> .
<code>\prime</code>	' Prime, ou minute au sein d'une expression temporelle (ordinaire). Typiquement utilisé en indice supérieur <code>\$A^{\prime}\$</code> . Il est à noter que <code>\$f^{\prime}\$</code> et <code>\$f'\$</code> produisent le même résultat. Un avantage de la seconde forme est que <code>\$f'''</code> produit le symbole désiré, c.-à-d. le même résultat que <code>\$f^{\prime\prime\prime}\$</code> , mais en nécessitant moins de saisie. Il est à noter que vous ne pouvez utiliser <code>\prime</code> qu'en mode mathématique. Utiliser le quote ' en mode texte produit un caractère différent (apostrophe ou guillemet anglais simple de droite).
<code>\prod</code>	\prod Produit (opérateur).
<code>\propto</code>	\propto Proportionnel à (relation)
<code>\Psi</code>	Ψ Lettre grecque en capitale (ordinaire).
<code>\psi</code>	ψ Lettre grecque en bas de casse (ordinaire).
<code>\rangle</code>	\rangle Chevron mathématique droit, ou séquence, crochet (fermant). Similaire : supérieur à <code>></code> . Correspond à <code>\rangle</code> .
<code>\rbrace</code>	$\}$ Accolade de droite (fermante). Synonyme : <code>\}</code> . Correspond à <code>\lbrace</code> .
<code>\rbrack</code>	$\}$ Crochet droit (fermant). Synonyme : <code>]</code> . Correspond à <code>\lbrack</code> .
<code>\rceil</code>	\lceil Plafond droit (fermant). Correspond à <code>\lceil</code> .
<code>\Re</code>	\Re Partie réelle, nombres réels, R capital gothique (ordinaire). En relation avec : R majuscule ajouré, ou gras de tableau d'école <code>\mathbb{R}</code> ; pour accéder à cela charger le paquetage <code>amsfonts</code> .
<code>\restriction</code>	<code>\restriction</code> Restriction d'une fonction (relation). Synonyme : <code>\upharpoonright</code> . Ceci n'est pas disponible en T _E X de base. En L ^A T _E X vous devez charger le paquetage <code>amssymb</code> .

<code>\reemptyset</code>	<code>\reemptyset</code> Symbole ensemble vide inversé (ordinaire). En relation avec : <code>\varnothing</code> . Ceci n'est pas disponible en $\text{T}_{\text{E}}\text{X}$ de base. En $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ vous devez charger le paquetage <code>stix</code> .
<code>\rfloor</code>	<code>\rfloor</code> Plancher droit, un crochet droit avec la dent du haut coupée (fermant). Correspond à <code>\lfloor</code> .
<code>\rhd</code>	<code>\rhd</code> Pointe de flèche, c.-à-d. u triangle pointant vers la droite (binaire). Ceci n'est pas disponible en $\text{T}_{\text{E}}\text{X}$ de base. En $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ vous devez charger le paquetage <code>amssymb</code> . Pour le symbole normal de sous-groupe vous devriez plutôt charger <code>amssymb</code> et utiliser <code>\vartriangleright</code> (qui est une relation et ainsi donne un meilleur espacement).
<code>\rho</code>	<code>\rho</code> Lettre grecque en bas de casse (ordinaire). La forme en variante est <code>\varrho</code> .
<code>\Rightarrow</code>	<code>\Rightarrow</code> Implique, flèche vers la droite à double-ligne (relation). Similaire : flèche vers la droite <code>\rightarrow</code> .
<code>\rightarrow</code>	<code>\rightarrow</code> Flèche vers la droite à ligne simple (relation). Synonyme : <code>\to</code> . Similaire : flèche vers droite à ligne double <code>\Rightarrow</code> .
<code>\rightharpoondown</code>	<code>\rightharpoondown</code> Harpon vers la droite avec crochet vers le bas (relation).
<code>\rightharpoonup</code>	<code>\rightharpoonup</code> Harpon vers la droite avec crochet vers la haut (relation).
<code>\rightleftharpoons</code>	<code>\rightleftharpoons</code> Harpon vers la droite avec crochet vers le haut au dessus d'un harpon vers la gauche avec crochet vers le bas (relation).
<code>\searrow</code>	<code>\searrow</code> Flèche pointant sud-est (relation).
<code>\setminus</code>	<code>\setminus</code> Différence ensembliste, barre de fraction inversée ou controblique, comme <code>\</code> (binaire). Similaire : controblique <code>\backslash</code> et aussi <code>\textbackslash</code> hors du mode mathématique.
<code>\sharp</code>	<code>\sharp</code> Dièse (ordinaire).
<code>\Sigma</code>	<code>\Sigma</code> Lettre grecque en capitale (ordinaire).
<code>\sigma</code>	<code>\sigma</code> Lettre grecque en bas de casse (ordinaire). La forme en variante est <code>\varsigma</code> .
<code>\sim</code>	<code>\sim</code> Similaire à, dans une relation (relation).
<code>\simeq</code>	<code>\simeq</code> \simeq Similaire ou égal à, dans une relation (relation).
<code>\smallint</code>	<code>\smallint</code> Signe intégrale qui ne change pas en taille plus grande au sein d'une formule hors texte (opérateur).
<code>\smile</code>	<code>\smile</code> Arc concave en haut, sourire (ordinaire).

- `\spadesuit` ♠ Pique de carte à jouer (ordinaire).
- `\sqcap` \sqcap Symbole d'intersection carré (binaire). Similaire : intersection `\cap`.
- `\sqcup` \sqcup Symbole d'union carré (binaire). Similaire : union `\cup`. En relation avec : opérateur de taille variable `\bigsqcup`.
- `\sqsubset`
`\sqsubset` Symbole carré de sous-ensemble (relation). Similaire : sous-ensemble `\subset`. Ceci n'est pas disponible en $\text{T}_{\text{E}}\text{X}$ de base. En $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ vous devez charger le paquetage `amssymb`.
- `\sqsubseteq`
 \sqsubseteq Symbole carré de sous-ensemble ou égal (binaire). Similaire : sous-ensemble ou égal `\subseteq`.
- `\sqsupset`
`\sqsupset` Symbole carré de sur-ensemble (relation). Similaire : superset `\supset`. Ceci n'est pas disponible en $\text{T}_{\text{E}}\text{X}$ de base. En $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ vous devez charger le paquetage `amssymb`.
- `\sqsupseteq`
 \sqsupseteq Symbole carré de sur-ensemble ou égal (binaire). Similaire : sur-ensemble ou égal `\supseteq`.
- `\star` \star Étoile à cinq branches, quelquefois utilisée comme une opération binaire générale mais quelquefois réservée pour la corrélation croisée (binaire). Similaire : les synonymes astérisque `*` et `\ast`, qui sont à six branches, et apparaissent plus souvent comme indice supérieur et inférieur, comme c'est le cas avec l'étoile de Kleene.
- `\subset` \subset Est contenu (occasionnellement, est impliqué par) (relation).
- `\subseteq`
 \subseteq Est contenu ou égal à (relation).
- `\succ` \succ Vient après, suit (relation). Similaire : inférieur à \succ .
- `\succeq` \succeq Suit ou est égal à (relation). Similaire : inférieur ou égal à `\leq`.
- `\sum` \sum Summation (opérateur). Similaire : Lettre grecque capitale `\Sigma`.
- `\supset` \supset Contient (relation).
- `\supseteq`
 \supseteq Contient ou est égal à (relation).
- `\surd` \surd Symbole racine (ordinaire). La commande $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ `\sqrt{..}` compose la racine de l'argument, avec une barre qui s'étend pour couvrir l'argument.
- `\swarrow` \swarrow Flèche pointant au sud-ouest (relation).
- `\tau` τ Lettre grecque en bas de casse (ordinaire).
- `\theta` θ Lettre grecque en bas de casse (ordinaire). La forme en variante est `\vartheta`.

- `\times` × Signe de multiplication d'école primaire (binaire). Voir aussi `\cdot`.
- `\to` → Flèche en ligne simple vers la droite (relation). Synonyme : `\rightarrow`.
- `\top` `\top` Taquet vers le haut ; dessus, plus grand élément d'un ensemble partiellement ordonné (ordinaire). Voir aussi `\bot`.
- `\triangle`
△ Triangle (ordinaire).
- `\triangleleft`
◁ Triangle non-rempli pointant à gauche (binaire). Similaire : `\lhd`. Pour le symbole normal de sous-groupe vous devez charger `amssymb` et utiliser `\vartriangleleft` (qui est une relation et ainsi donne un meilleur espacement).
- `\triangleright`
▷ Triangle non-rempli pointant à droite (binaire). Pour le symbole normal de sous-groupe vous devez à la place charger `amssymb` et utiliser `\vartriangleright` (qui est une relation et ainsi donne un meilleur espacement).
- `\unlhd` `\unlhd` Pointe de flèche non-pleine soulignée pointant vers la gauche, c.-à-d. triangle avec une ligne au-dessous (binaire). Ceci n'est pas disponible en `TEX` de base. En `LATEX` vous devez charger le paquetage `amssymb`. Pour le symbole normal de sous-groupe chargez `amssymb` et utilisez `\vartrianglelefteq` (qui est une relation et ainsi donne un meilleur espacement).
- `\unrhd` `\unrhd` Pointe de flèche non-pleine soulignée pointant vers la droite, c.-à-d. triangle avec une ligne au-dessous (binaire). Ceci n'est pas disponible en `TEX` de base. En `LATEX` vous devez charger le paquetage `amssymb`. Pour le symbole normal de sous-groupe chargez `amssymb` et utilisez `\vartrianglerighteq` (qui est une relation et ainsi donne un meilleur espacement).
- `\Uparrow` ↑↑ Flèche vers le haut à ligne double (relation). Similaire : flèche vers le haut à ligne simple `\uparrow`.
- `\uparrow` ↑ Single-line upward-pointing flèche, diverges (relation). Similaire : double-line up-pointing flèche `\Uparrow`.
- `\Updownarrow`
⇕ Double-line upward-and-downward-pointing flèche (relation). Similaire : single-line upward-and-downward-pointing flèche `\updownarrow`.
- `\updownarrow`
⇕ flèche haut et bas à ligne simple (relation). Similaire : flèche haut et bas à ligne double `\Updownarrow`.
- `\upharpoonright`
`\upharpoonright` Harpon vers le haut avec crochet à droite (relation). Synonyme : `\restriction`. Ceci n'est pas disponible en `TEX` de base. En `LATEX` vous devez charger le paquetage `amssymb`.

`\uplus` \uplus Union de multiensemble, un symbole union avec un symbole plus en son sein (binaire). Similaire : union `\cup`. En relation avec : opérateur de taille variable `\biguplus`.

`\Upsilon` Lettre grecque en capitale (ordinaire).

`\upsilon` υ Lettre grecque en bas de casse (ordinaire).

`\varepsilon` ε Epsilon de ronde en bas de casse (ordinaire). Plus largement utilise en mathématique que la forme non variante epsilon lunaire `\epsilon`. En relation avec : « appartient à » `\in`.

`\vanothing` `\varnothing` Symbole ensemble vide. Similaire : `\emptyset`. En relation avec : `\reemptyset`. Ceci n'est pas disponible en T_EX de base. En L^AT_EX vous devez charger le paquetage `amssymb`.

`\varphi` φ Variante de la lettre grecque bas de casse (ordinaire). La forme non en variante est `\phi`.

`\varpi` ϖ Variante de la lettre grecque bas de casse (ordinaire). La forme non en variante est `\pi`.

`\varrho` ϱ Variante de la lettre grecque bas de casse (ordinaire). La forme non en variante est `\rho`.

`\varsigma` ς Variante de la lettre grecque bas de casse (ordinaire). La forme non en variante est `\sigma`.

`\vartheta` ϑ Variante de la lettre grecque bas de casse (ordinaire). La forme non en variante est `\theta`.

`\vdash` \vdash Taquet droit ; prouve, implique ; portillon/tourniquet ; barre verticale et un tiret (relation). Similaire : portillon inversé `\dashv`.

`\vee` \vee Ou logique ; une forme de v pointant vers le bas (binaire). En relation avec : Et logique `\wedge`. Similaire : Opérateur de taille variable `\bigvee`.

`\Vert` $\|$ Double barre verticale (ordinaire). Similaire : barre verticale simple `\vert`.
Le paquetage `mathtools` vous permet de créer des commandes pour des délimiteurs appariés. Par ex., si vous mettez `\DeclarePairedDelimiter\norm{\lVert}{\rVert}` dans votre préambule, ceci vous donne trois variantes de commande pour faire des barres verticales correctement espacées horizontalement : si dans le corps du document vous écrivez la version étoilée `\norm*{M\perp}` alors la hauteur des barres verticales correspond à celle de l'argument, alors qu'avec `\norm{M\perp}` les barres ne grandissent pas avec la hauteur de l'argument mais à la place reste de la hauteur par défaut, et `\norm[commande-taille]{M\perp}` donne aussi des barres qui ne grandissent pas mais sont de la taille donnée dans la *commande-taille*, par ex. `\Bigg`.

<code>\vert</code>	Barre verticale simple (ordinaire). Similaire : barre à double-ligne verticale <code>\Vert</code> . Pour usage tel qu'au sein de la définition d'un ensemble, utilisez <code>\mid</code> parce que c'est une relation. Pour une valeur absolue vous pouvez utiliser le paquetage <code>mathtools</code> et ajouter <code>\DeclarePairedDelimiter\abs{\lvert}{\rvert}</code> à votre préambule. Ceci vous donne trois variantes de commande pour les barres simples verticales qui sont correctement espacées verticalement : si dans le corps du document vous écrivez la version étoilée <code>\abs*{\frac{22}{7}}</code> alors la hauteur des barres verticales correspond à la hauteur de l'argument, alors qu'avec <code>\abs{\frac{22}{7}}</code> les barres ne grandissent pas avec l'argument mais à la place sont de la hauteur par défaut, et <code>\abs[commande-taille]{\frac{22}{7}}</code> donne aussi des barres qui ne grandissent pas mais qui sont réglées à la taille donnée par la <i>commande-taille</i> , par ex. <code>\Bigg</code> .
<code>\wedge</code>	\wedge Et logique (binaire). Synonyme : <code>\land</code> . Voir aussi ou logique <code>\vee</code> . Similaire : opérateur de taille variable <code>\bigwedge</code> .
<code>\wp</code>	\wp « p » de Weierstrass, fonction elliptique de Weierstrass (ordinaire).
<code>\wr</code>	\wr Produit couronne (binaire).
<code>\Xi</code>	Ξ Lettre grecque en capitale (ordinaire).
<code>\xi</code>	ξ Lettre grecque en bas de casse (ordinaire).
<code>\zeta</code>	ζ Lettre grecque en bas de casse (ordinaire).

Les symboles suivants sont utilisés le plus souvent en texte simple mais \LaTeX fourni des versions pour être utilisés dans du texte mathématique.

<code>\mathdollar</code>	Signe dollar en mode mathématique : $\$$.
<code>\mathparagraph</code>	Signe paragraphe (pied-de-mouche) en mode mathématique : \P .
<code>\mathsection</code>	Signe section en mode mathématique : \S .
<code>\mathsterling</code>	Signe livre sterling mode mathématique : \pounds .
<code>\mathunderscore</code>	Signe « souligné » en mode mathématique : $\underline{\quad}$.

16.2.1 Flèches

Voici les flèches disponibles en standard avec \LaTeX . Les paquetages `latexsym` et `amsfonts` en contiennent beaucoup plus.

Symbole	Commande
\Downarrow	<code>\Downarrow</code>
\downarrow	<code>\downarrow</code>
\hookleftarrow	<code>\hookleftarrow</code>

\hookrightarrow	<code>\hookrightarrow</code>
\leftarrow	<code>\leftarrow</code>
\Leftarrow	<code>\Leftarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>
\longleftarrow	<code>\longleftarrow</code>
\Longleftarrow	<code>\Longleftarrow</code>
\longleftrightarrow	<code>\longleftrightarrow</code>
\Longleftrightarrow	<code>\Longleftrightarrow</code>
\mapsto	<code>\mapsto</code>
\Rightarrow	<code>\Rightarrow</code>
\rightarrow	<code>\rightarrow</code>
\mapsto	<code>\mapsto</code>
\nearrow	<code>\nearrow</code>
\nwarrow	<code>\nwarrow</code>
\Rightarrow	<code>\Rightarrow</code>
\rightarrow	<code>\rightarrow</code> , or <code>\to</code>
\searrow	<code>\searrow</code>
\swarrow	<code>\swarrow</code>
\uparrow	<code>\uparrow</code>
\Uparrow	<code>\Uparrow</code>
\updownarrow	<code>\updownarrow</code>
\Updownarrow	<code>\Updownarrow</code>

Un exemple de la différence entre `\to` et `\mapsto` est : `\(f\colon D\to C \)` given by `\(n\mapsto n^2 \)`.

Pour faire des diagrammes cumulatifs il y a de nombreux paquets, parmi lesquels `tikz-cd` et `amscd`.

16.2.2 `\boldmath` & `\unboldmath`

Synopsis (utilisés en mode paragraphe ou mode LR) :

```
\boldmath \langle math \rangle
```

ou

```
\unboldmath \langle math \rangle
```

Ce sont des déclarations pour mettre en caractères gras les lettres et symboles dans *math*, ou pour contredire cela et revenir à la graisse normale (non-gras) utilisée par défaut. Elles ne doivent être utilisées ni en mode math, ni en mode math hors texte (voir Chapitre 17 [Modes], page 185). Les deux commandes sont fragiles (voir Section 12.11 [`\protect`], page 134).

Voici un exemple dans lequel chaque commande `\boldmath` a lieu au sein d'une `\mbox` :

```
on a  $\mbox{\boldmath \langle v \rangle} = 5\cdot\mbox{\boldmath \langle u \rangle}$ 
```

ce qui signifie que `\boldmath` est seulement utilisé dans un mode texte, en l'occurrence le mode LR, et explique pourquoi L^AT_EX doit basculer en mode math pour composer *v* et *u*.

Si vous utilisez l'une ou l'autre des ces commande en mode math, comme dans **Échec** : `\(\boldmath x \)`, alors vous obtenez quelque chose du genre de 'LaTeX Font Warning:

Command `\boldmath` invalid in math mode on input line 11' et 'LaTeX Font Warning: Command `\mathversion` invalid in math mode on input line 11'.

La commande `\boldmath` est problématique par bien des aspects. Les nouveaux documents devraient utiliser le paquetage `bm` fourni par l'équipe du projet L^AT_EX. Une description complète est hors du champ d'application de ce manuel (voir la documentation complète sur le CTAN) mais même le petit exemple suivant :

```
\usepackage{bm} % dans le préambule
...
on a  $\mathbf{v} = 5\cdot\mathbf{u}$ 
```

illustre déjà que c'est une amélioration par rapport à `\boldmath`.

16.2.3 Pseudo-gras avec des lettres évidées

Synopsis :

```
\usepackage{amssymb} % dans le préambule
...
\mathbb{lettre-capitale}
```

Fournit les lettres mathématiques évidées utilisées traditionnellement pour noter les noms d'ensembles, comme les entiers naturels, relatifs, etc. Le nom `\mathbb` vient de « **blackboard bold math** » qu'on pourrait traduire par *caractères mathématiques gras scolaires*.

Dans l'exemple suivant :

```
\( \forall n \in \mathbb{N}, n^2 \geq 0 \)
```

le `\mathbb{N}` donne le symbole gras scolaire \mathbb{N} , représentant les entiers naturels.

Si vous utilisez autre chose qu'une lettre capitale alors aucune erreur n'est produite, mais vous obtiendrez des résultats étranges, y compris des caractères inattendus.

Il y a des paquetages donnant accès à des symboles autres que les lettres capitales ; voir sur le CTAN.

16.2.4 Math Calligraphic

Synopsis :

```
\mathcal{lettres-capitales}
```

Utilise une fonte scripte, *Math Calligraphic*.

Dans l'exemple suivant l'identificateur du graphe est produit en sortie avec la fonte Math Calligraphic :

```
Soit un graphe  $\mathcal{G}$ .
```

Si vous utilisez autre chose que des lettres capitales alors vous n'aurez pas une erreur mais vous n'aurez pas non plus une sortie en calligraphique math. Par exemple, `\mathcal{g}` produit en sortie une accolade fermante.

16.2.5 Délimiteurs

Les délimiteurs sont les parenthèses, accolades, crochets, ou autres caractères utilisés pour marquer le début et la fin d'une sous-formule. Dans l'exemple suivant la formule a trois ensembles de parenthèses délimitant les trois sous-formules :

```
(z-z_0)^2 = (x-x_0)^2 + (y-y_0)^2
```

Il n'est pas nécessaire que les délimiteurs se correspondent, ainsi vous pouvez saisir $\left([0,1) \right)$.

Voici les délimiteurs usuels :

Délimiteur	Commande	Nom
((Parenthèse gauche
))	Parenthèse droite
\}	{ ou \lbrace	Accolade gauche
\{	} ou \rbrace	Accolade droite
[[ou \lbrack	Crochet gauche
]] ou \rbrack	Crochet droit
⌊	\lfloor	Crochet gauche de partie entière
⌋	\rfloor	Crochet droit de partie entière
⌈	\lceil	Crochet gauche d'arrondi supérieur
⌉	\rceil	Crochet droit d'arrondi supérieur
<	\langle	Chevron gauche
>	\rangle	Chevron droit
/	/	Oblique, ou oblique vers l'avant
\	\backslash	Controblique, ou oblique vers l'arrière
	ou \vert	Barre verticale
	\ ou \Vert	Barre verticale double

Le paquetage `mathtools` vous permet de créer des commandes pour des délimiteurs appariés. Par ex., si vous mettez `\DeclarePairedDelimiter\abs{\lvert}{\rvert}` dans votre préambule alors cela vous donne deux commandes pour les barres simples verticales (qui ne fonctionnent qu'en mode math). La forme étoilée, comme dans `\abs*{\frac{22}{7}}`, a la hauteur des barres verticales accordée à la hauteur de l'argument. La forme non-étoilée, comme dans `\abs{\frac{22}{7}}`, a les barres fixées à la hauteur par défaut. Cette forme accepte un argument optionnel, comme dans `\abs[commande-taille]{\frac{22}{7}}`, où la taille des barres est donnée par la *commande-taille*, par ex. `\Bigg`. En utilisant plutôt `\lVert` et `\rVert` pour les symboles, vous obtenez une notation de norme avec le même comportement.

16.2.5.1 `\left` & `\right`

Synopsis :

```
\left délimiteur1 ... \right délimiteur2
```

Fabrique des parenthèses, crochets, ou autres délimiteurs qui se correspondent. L^AT_EX fabrique les délimiteurs juste assez hauts pour couvrir la taille de la formule qu'ils embrassent.

L'exemple suivant fabrique un vecteur unitaire entouré par des parenthèses assez hautes pour couvrir les composantes :

```
\begin{equation}
\left(\begin{array}{c}
1 \\
0
\end{array}\right)
\end{equation}
```

Voir Section 16.2.5 [Delimiters], page 168, pour une liste des délimiteurs usuels.

Chaque `\left` doit avoir un `\right` qui lui correspond. Dans l'exemple plus haut, si on ne met pas le `\left` (alors on a le message d'erreur 'Extra `\right`'). Si c'est le `\right` qu'on ne met pas, alors on obtient 'You can't use '`\eqno`' in math mode'.

Toutefois, il n'est pas nécessaire que *délimiteur1* et *délimiteur2* se correspondent. Un cas usuel est que vous désiriez une accolade d'un seul côté, comme ci-dessous. Utilisez un point, '.', comme *délimiteur nul* :

```
\begin{equation}
  f(n)=\left\{\begin{array}{ll}
    1 & \&\mbox{si } \backslash(n=0\backslash) \\
    f(n-1)+3n^2 & \&\mbox{sinon}
  \end{array}\right.
\end{equation}
```

Notez que pour obtenir une accolade comme délimiteur vous devez la préfixer d'une contrôliques, `\{` (voir Section 23.1 [Reserved characters], page 228). (Les paquetages `amsmath` et `mathtools` vous permettent d'obtenir la construction ci-dessus avec un environnement `cases`).

La paire `\left ... \right` fabrique un groupe. Une conséquence de cela est que la formule embrassée par la paire `\left ... \right` ne peut pas avoir de saut de ligne dans la sortie. Ceci inclut à la fois les sauts de lignes manuels, et ceux générés automatiquement par \LaTeX . Dans l'exemple suivant, \LaTeX insère un saut de ligne pour que la formule tienne entre les marges :

```
  Lorem ipsum dolor sit amet
  \left( (a+b+c+d+e+f+g+h+i+j+k+l+m+n+o+p+q+r+s+t+u+v+w+x+y+z) \right)
```

Mais avec `\left` et `\right` :

```
  Lorem ipsum dolor sit amet
  \left( \left( a+b+c+d+e+f+g+h+i+j+k+l+m+n+o+p+q+r+s+t+u+v+w+x+y+z \right) \right)
```

\LaTeX n'insère pas de saut, ce qui a pour effet que la formule s'étende jusque dans la marge.

Parce que `\left ... \right` fait un groupe, toutes les règles habituelles concernant les groupes s'appliquent. Ci-dessous, la valeur de `\testlength` réglée au sein de l'équation sera oubliée, en et sortie on a '1.2pt'.

```
\newlength{\testlength} \setlength{\testlength}{1.2pt}
\begin{equation}
  \left( a+b=c \setlength{\testlength}{3.4pt} \right)
  \the\testlength
\end{equation}
```

La paire `\left ... \right` affecte l'espacement horizontal de la formule embrassée de deux manières. La première est que dans `\left(\sin(x) = \sin\left(x\right) \right)` le membre de droite a plus d'espace autour du `x`. C'est parce que `\left(... \right)` insère un nœud interne alors que `(...)` insère un nœud ouvrant. La seconde manière dont l'espacement horizontal est affecté est que parce que la paire forme un groupe, la sous-formule embrassée sera composée à sa longueur naturelle, sans dilatation ni contraction pour que la ligne s'ajuste mieux.

\TeX dilate les délimiteurs selon la hauteur et profondeur de la formule embrassée. Ci-dessous \LaTeX fait grandir les crochets pour qu'ils s'étendent sur toute la hauteur de l'intégrale :

```
\begin{equation}
\left[ \int_{x=r_0}^{\infty} -G\frac{Mm}{r^2}\, dr \right]
\end{equation}
```

Régler la taille manuellement donne souvent un meilleur résultat. Par exemple, bien que le filet ci-dessous ait une profondeur nulle, \TeX crée des délimiteurs qui s'étendent sous le filet :

```
\begin{equation}
\left( \rule{1pt}{1cm} \right)
\end{equation}
```

\TeX peut choisir des délimiteurs qui sont trop petits, comme dans $(\left| |x|+|y| \right|)$. Il peut aussi en choisir de trop grand, comme ci-dessous :

```
\begin{equation}
\left( \sum_{0\leq i<n} i^k \right)
\end{equation}
```

Un troisième cas malheureux c'est quand on a une formule hors texte longue tenant sur plus d'une ligne et que vous devez accorder les tailles du délimiteur ouvrant et fermant ; vous ne pouvez pas utiliser \left sur la première ligne et \right sur la dernière parce qu'ils doivent se correspondre.

Pour régler manuellement la taille des délimiteurs, voir Section 16.2.5.2 [\bigl & \bigr etc.], page 171.

16.2.5.2 \bigl , \bigr , etc.

Synopsis, l'un parmi :

```
\bigldélimiteur1 ... \bigrdélimiteur2
\Bigldélimiteur1 ... \Bigrdélimiteur2
\biggldélimiteur1 ... \biggrdélimiteur2
\Biggldélimiteur1 ... \Biggrdélimiteur2
```

(comme dans $\bigl[...\bigr]$; strictement parlant il n'est pas nécessaire qu'ils soient appariés, voir ci-dessous), ou l'un parmi :

```
\bigmdélimiteur
\Bigmdélimiteur
\biggmdélimiteur
\Biggmdélimiteur
```

(comme avec $\bigl|$), ou l'un parmi :

```
\bigdélimiteur
\Bigdélimiteur
\biggdélimiteur
\Biggdélimiteur
```

(comme avec $\bigl[$).

Produit des délimiteurs de taille réglée manuellement. Pour des délimiteurs de taille réglée automatiquement voir Section 16.2.5.1 [\left & \right], page 169).

L'exemple ci-dessous produit des barres verticales externes légèrement plus grandes :

```
\bigl| |x|+|y| \bigr|
```

Les commandes ci-dessus sont dans l'ordre des tailles croissantes. Vous pouvez utiliser la taille la plus petite comme `\bigl...``\bigr` dans un paragraphe sans que cela ait pour effet que L^AT_EX écarte les lignes. Les tailles plus grandes sont destinées aux équations hors texte.

Voir Section 16.2.5 [Delimiters], page 168, pour une liste des délimiteurs usuels. Dans la famille des commandes en 'l' ou 'r', il n'est pas nécessaire que *délimiteur1* et *délimiteur2* se correspondent.

Les commandes en 'l' ou 'r' produisent délimiteurs ouvrant et fermant qui n'insèrent aucun espace horizontal entre un atome précédent et le délimiteur, alors que les commandes sans 'l' ou 'r' insèrent un peu d'espace (parce que chaque délimiteur est réglé comme une variable ordinaire). Comparez les deux lignes dans l'exemple ci-dessous :

```
\begin{tabular}{l}
  \(\displaystyle \sin\biggl(\frac{1}{2}\biggr) \) \ \ % bon
  \(\displaystyle \sin\bigg(\frac{1}{2}\bigg) \) \ \ % mauvais
\end{tabular}
```

La gestion typographique traditionnelle est sur la première ligne. Sur la seconde ligne la sortie a de l'espace supplémentaire entre le `\sin` et la parenthèse ouvrante.

Les commandes sans 'l' ou 'r' donnent cependant l'espacement correct dans certaines circonstances, comme avec la grande ligne verticale dans l'exemple ci-dessous :

```
\begin{equation}
  \int_{x=a}^b x^2 dx = \frac{1}{3} x^3 \Big|_{x=a}^b
\end{equation}
```

(beaucoup d'auteurs préfèrent remplacer `\frac` par la commande `\tfrac` du paquetage `amsmath`), et comme avec cette grande oblique :

```
\begin{equation}
  \lim_{n\to\infty} \pi(n) \big/ (n/\log n) = 1
\end{equation}
```

Contrairement à la paire `\left...``\right` (voir Section 16.2.5.1 [`\left` & `\right`], page 169), les commandes décrites ici et en 'l' ou 'r' ne forment pas un groupe. Strictement parlant il n'est pas nécessaire qu'elles soient appairées, et donc vous pourriez écrire quelque chose du ce genre :

```
\begin{equation}
  \Biggl[ \pi/6 ]
\end{equation}
```

Les commandes en 'm' sont pour les relations, les choses qui sont au milieu d'une formule, comme ceci :

```
\begin{equation}
  \biggl\{ a \in B \biggm| a = \sum_{0 \leq i < n} 3i^2 + 4 \biggr\}
\end{equation}
```

16.2.6 Points de suspensions, horizontaux ou verticaux

Les points de suspension sont les trois points (habituellement trois) indiquant qu'un motif se prolonge.

```
\begin{array}{cccc}
a_{0,0} & & & \\
a_{1,0} & & & \\
\vdots & & & \\
\end{array}
```

L^AT_EX fournit ces types de points de suspension :

- `\cdots` Des points de suspension horizontaux élevés au centre de la ligne, comme dans \dots . Utilisés comme dans : `\(a_0\cdot a_1\cdots a_{n-1} \)`.
- `\ddots` Des points de suspension en diagonale, \ddots . Voir l'exemple plus haut d'un tableau `array` pour une utilisation.
- `\ldots` Points de suspension sur la ligne de base, \dots . Utilisés comme dans : `\(x_0,\ldots x_{n-1} \)`. Un autre exemple est le tableau `array` donné plus haut. Un synonyme est `\mathellipsis`. Un synonyme du paquetage `amsmath` est `\hdots`.
Vous pouvez également utiliser cette commande au-dehors du mode `math`, comme dans *La boîte de vitesse, les freins, \ldots{} tout est cassé*. (En mode paragraphe ou en mode LR un synonyme de `\ldots` est `\dots`).
- `\vdots` Points de suspension alignés verticalement, \vdots . Voir l'exemple plus haut avec un `array` pour une utilisation possible.

Le paquetage `amsmath` a la commande `\dots` pour baliser sémantiquement les points de suspension. L'exemple suivant produit deux sorties d'aspects distincts pour les deux premiers usages de la commande `\dots`.

```
\usepackage{amsmath} % dans le préambule
```

```
...
```

```
Supposons que \( p_0, p_1, \dots, p_{n-1} \) soit la liste de tous les nombres premier
On remarque que \( p_0\cdot p_1 \dots \cdot p_{n-1} +1 \) n'est pas un
multiple de l'un quelconque des \( p_i \).
```

```
Conclusion : il y a un nombre infini de nombres premiers \( p_0, p_1, \dotsc \).
```

Dans la première ligne L^AT_EX regarde la virgule suivant `\dots` pour déterminer qu'il devrait produire en sortie des points de suspension sur la ligne de base. La seconde ligne a un `\cdot` qui suit `\dots` alors L^AT_EX produit des points de suspension sur l'axe mathématique, verticalement centrés. Toutefois, le troisième usage n'a aucun caractère de suivi, aussi vous devez indiquer à L^AT_EX que faire. Vous pouvez utiliser l'une des commandes : `\dotsc` si vous avez besoin de points de suspension appropriés pour suivre une virgule, `\dotsb` si vous avez des points de suspension qui conviennent quand il y a ensuite un opérateur ou un symbole de relation, `\dotsi` pour les points de suspension utilisés avec des intégrales, ou `\dotso` pour les autres cas.

16.2.7 Lettres grecques

On ne donne les versions en capitale de ces lettres grecques que lorsque elle sont distinctes des lettres romaines capitales.

Symbole	Commande	Nom
α	<code>\alpha</code>	Alpha
β	<code>\beta</code>	Bêta
γ, Γ	<code>\gamma, \Gamma</code>	Gamma
δ, Δ	<code>\delta, \Delta</code>	Delta
ε, ϵ	<code>\varepsilon, \epsilon</code>	Epsilon
ζ	<code>\zeta</code>	Zêta
η	<code>\eta</code>	Êta
θ, ϑ	<code>\theta, \vartheta</code>	Thêta
ι	<code>\iota</code>	Iota
κ	<code>\kappa</code>	Kappa
λ, Λ	<code>\lambda, \Lambda</code>	Lambda
μ	<code>\mu</code>	Mu
ν	<code>\nu</code>	Nu
ξ, Ξ	<code>\xi, \Xi</code>	Xi
π, Π	<code>\pi, \Pi</code>	Pi
ρ, ϱ	<code>\rho, \varrho</code>	Rhô
σ, Σ	<code>\sigma, \Sigma</code>	Sigma
τ	<code>\tau</code>	Tau
ϕ, φ, Φ	<code>\phi, \varphi, \Phi</code>	Phi
χ	<code>\chi</code>	chi
ψ, Ψ	<code>\psi, \Psi</code>	Psi
ω, Ω	<code>\omega, \Omega</code>	Oméga

Pour omicron, si vous utiliser la fonte par défaut de L^AT_EX Computer Modern alors saisissez omicron juste comme ‘o’ ou ‘O’. Si vous désirez avoir le nom ou si votre fonte affiche une différence alors vous pouvez utiliser quelque chose du genre de `\newcommand\omicron{o}`. Le paquetage `unicode-math` comprend `\upomicron` pour un omicron droit et `\mitomicron` pour l’italique mathématique.

Alors que le symbole de relation d’appartenance à un ensemble \in généré par `\in` est similaire à epsilon, il n’est jamais utilisé pour une variable.

16.3 Fonctions mathématiques

En mode mathématique ces commandes produisent des noms de fonction en police romaine et avec l’espace adéquat.

<code>\arccos</code>	Arccosinus, cosinus inverse arccos
<code>\arcsin</code>	Arcsinus, sinus inverse arcsin
<code>\arctan</code>	Arctangente, tangente inverse arctan
<code>\arg</code>	Angle entre l’axe des réels et un point dans le plan complexe arg
<code>\bmod</code>	Opérateur modulo binaire, utilisé comme dans $(5 \bmod 3 = 2)$ $5 \bmod 3 = 2$

<code>\cos</code>	Cosinus cos
<code>\cosh</code>	Cosinus hyperbolique cosh
<code>\cot</code>	Cotangente cot
<code>\coth</code>	Cotangente hyperbolique coth
<code>\csc</code>	Cosécante csc
<code>\deg</code>	Degrés deg
<code>\det</code>	Déterminant det
<code>\dim</code>	Dimension dim
<code>\exp</code>	Exponentielle exp
<code>\gcd</code>	Plus grand commun diviseur gcd
<code>\hom</code>	Homomorphisme hom
<code>\inf</code>	Infimum inf
<code>\ker</code>	Noyau ker
<code>\lg</code>	Logarithme en base 2 lg
<code>\lim</code>	Limite lim
<code>\liminf</code>	Limite inférieure lim inf
<code>\limsup</code>	Limite supérieure lim sup
<code>\ln</code>	Logarithme népérien ln
<code>\log</code>	Logarithme log
<code>\max</code>	Maximum max
<code>\min</code>	Minimum min
<code>\pmod</code>	Modulo parenthésé, comme dans $(5 \equiv 2 \pmod 3)$
<code>\Pr</code>	Probabilité Pr
<code>\sec</code>	Sécante sec
<code>\sin</code>	Sinus sin
<code>\sinh</code>	Sinus hyperbolique sinh
<code>\sup</code>	Supremum sup
<code>\tan</code>	Tangente tan
<code>\tanh</code>	Tangente hyperbolique tanh

Le paquetage `amsmath` ajoute des améliorations à certaines de ces commandes, et vous permet aussi de définir les vôtres propres. La documentation complète est sur le CTAN, mais en un mot, vous pouvez définir un opérateur identité avec `\DeclareMathOperator{\identity}{id}` qui est comme ceux donnés plus haut mais s'imprime en 'id'. La forme étoilée `\DeclareMathOperator*{\op}{op}` règle tout indice supérieur ou inférieur pour être au-dessus et au-dessous, comme c'est la tradition pour `\lim`, `\sup`, ou `\max`.

16.4 Accents mathématiques

L^AT_EX fournit diverses commandes pour produire des lettres accentuées en math. Elles diffèrent de celles concernant les accents en texte normal (voir Section 23.5 [Accents], page 233).

<code>\acute</code>	Accent aigu mathématique \acute{x} .
<code>\bar</code>	Accent barre suscrite mathématique \bar{x}
<code>\breve</code>	Accent brève mathématique \breve{x}
<code>\check</code>	Accent háček (caron) mathématique \check{x}
<code>\ddot</code>	Accent diérèse (tréma) mathématique \ddot{x}
<code>\dot</code>	Accent point suscrit mathématique \dot{x}
<code>\grave</code>	Accent grave mathématique \grave{x}
<code>\hat</code>	Accent chapeau (circonflexe) mathématique \hat{x}
<code>\mathring</code>	Accent rond-en-chef mathématique \mathring{x}
<code>\tilde</code>	Accent tilde mathématique \tilde{x}
<code>\vec</code>	Symbole vecteur mathématique \vec{x}
<code>\widehat</code>	Accent chapeau large mathématique $\widehat{x+y}$
<code>\widetilde</code>	Accent tilde mathématique $\widetilde{x+y}$

Quand on place un accent sur un i ou un j, la tradition est d'en utiliser un sans point, `\imath` ou `\jmath` (voir Section 16.2 [Math symbols], page 155).

16.4.1 `\accent`

Synopsis :

`\accent` *numéro caractère*

Une commande primitive T_EX utilisée pour générer des caractères accentués à partir de diacritiques et de lettres. Le diacritique est sélectionné par *numéro*, un argument numérique, suivi par un espace, et ensuite un argument *caractère* pour construire le caractère accentué dans la fonte courante.

Voici les caractères ‘e’ accentués.

```
\accent18 e
\accent20 e
\accent21 e
\accent22 e
\accent23 e
```

Le premier a un accent grave, le second a un caron, le troisième une brève, le quatrième un macron, et le dernier un rond-en-chef. Ce qui donne è ẽ ē ē ê.

La position de l’accent est déterminée par la fonte au moment de sa conception et ainsi le résultat de `\accent` peut différer d’une fonte à l’autre. Dans L^AT_EX il est préférable d’avoir

des glyphes pour les caractères accentués plutôt que de les construire en utilisant `\accent`. Utiliser des glyphes qui contiennent déjà les caractères accentués (comme dans le codage T1) permet d'obtenir une césure correcte alors que `\accent` désactive la césure (spécifiquement avec le codage de fonte OT1 dans lequel il n'y a pas de glyphe avec accent).

Optionnellement il peut y avoir un changement de fonte entre *numéro* et *caractère*. Notez aussi que cette commande règle le paramètre `\spacefactor` à 1000 (voir Section 19.5 [`\spacefactor`], page 192).

Une caractéristique inévitable de certaines lettres cyrilliques et de la majorité des lettres cyrilliques accentuées est qu'on doit les assembler à partir de multiples éléments (accents, modificateurs, etc.) alors même que `\accent` ne fournit qu'une combinaison entre une seul diacritique et une seule lettre. Il y a aussi des cas où les accents doivent apparaître entre des lettres que `\accent` ne prend pas en charge. Il y a d'autres cas encore comme les lettres I et J dont les points au dessus de leurs homologues en bas-de-casse entrent en conflit avec les diacritiques suscrits. L'utilisation de la commande `\accent` dans ces cas ne fonctionnera pas du fait qu'elle ne peut pas distinguer entre capitales et bas-de-casse.

16.5 Sur- et sous-ligner

L^AT_EX fournit des commandes pour sur-ligner ou souligner, ou pour placer des accolades au-dessous ou au-dessus d'une matière.

`\underline{texte}`

A pour effet que *texte*, ce qui peut être ou non en mode mathématique, soit souligné. Le résultat de `\underline{xyz}` ressemble à cela : \underline{xyz} La ligne est toujours sous le texte, en prenant en compte les jambages, ainsi dans `\(\underline{y}\)` la ligne est plus bas que dans `\(\underline{x}\)`. Cette commande est fragile (voir Section 12.11 [`\protect`], page 134).

Notez que le paquetage `ulem` fait du soulignement en mode texte et permet les sauts de ligne ainsi que plusieurs autres fonctionnalités. Voir sa documentation sur le CTAN. Voir aussi Section 19.10 [`\hrulefill` & `\dotfill`], page 198, pour produire une ligne, pour des choses comme une signature.

`\overline{texte}`

Met une ligne horizontale au dessus de *texte*. Fonctionne en mode mathématiques, et en dehors. Par exemple, `\overline{x+y}`. Le résultat ressemble à cela : $\overline{x+y}$. Notez que cette commande diffère de `\bar` (voir Section 16.4 [Math accents], page 176).

`\underbrace{math}`

Met une accolade au-dessous de *math*. Par exemple, le code `(1-\underbrace{1/2}+(1/2)-1/3)` met en vedette la partie qui s'annule. Le résultat ressemble à cela : $(1 - \underbrace{1/2} + (1/2) - 1/3)$. Attachez du texte à l'accolade en utilisant la commande indice inférieur, `_`, ou indice supérieur, `^`, comme ci-dessous :

```
\begin{displaymath}
  1+1/2+\underbrace{1/3+1/4}_{>1/2}+
    \underbrace{1/5+1/6+1/7+1/8}_{>1/2}+\cdots
\end{displaymath}
```

L'indice supérieur apparaît au-dessus de l'expression, aussi il peut sembler sans relation avec l'accolade du dessous.

`\overbrace{math}`

Met une accolade au-dessus de *math*. Par exemple, `\overbrace{x+x+\cdots+x}^{\text{\mbox{\(k\) fois}}}`. Le résultat ressemble à cela : $\overbrace{x + x + \cdots + x}^{k \text{ fois}}$ Voir aussi `\underbrace`.

Le paquetage `mathtools` ajoute une accolade par dessus ou par dessous, ainsi que des améliorations sur les accolades. Voir la documentation sur le CTAN.

16.6 Espacement en mode mathématique

Quand il compose des mathématiques, \LaTeX insère l'espacement selon les règles normales pour les textes mathématiques. Si vous entrez `y=m x` alors \LaTeX ignore l'espace et dans la sortie le `m` est juste à côté du `x`, comme dans $y = mx$.

Mais les règles de \LaTeX ont occasionnellement besoin de petits ajustements. Par exemple, dans l'intégrale la tradition est de mettre une petite espace supplémentaire entre le $f(x)$ et le dx , obtenue ci-dessous avec la commande `\,` :

`\int_0^1 f(x)\,dx`

\LaTeX fournit les commandes qui suivent utilisables en mode mathématique. Beaucoup de ces définitions d'espacement sont exprimées dans l'unité mathématique *mu* donnée par $1 \text{ em} = 18 \text{ mu}$, où le *em* est pris de la famille courante des symboles mathématiques (voir Section 14.1 [Units of length], page 142). Ainsi, un `\thickspace` est typiquement de $5/18$ fois la largeur d'un 'M'.

`\;` Synonyme : `\thickspace`. Normalement $5.0\text{mu plus } 5.0\text{mu}$. Avec le paquetage `amsmath`, ou depuis la parution 2020-10-01 de \LaTeX , utilisable aussi bien en mode texte qu'en mode mathématique ; sinon, utilisable seulement en mode mathématique.

`\negthickspace`

Normalement $-5.0\text{mu plus } 2.0\text{mu minus } 4.0\text{mu}$. Avec le paquetage `amsmath`, ou depuis la parution 2020-10-01 de \LaTeX , utilisable aussi bien en mode texte qu'en mode mathématique ; sinon, utilisable seulement en mode mathématique.

`\:`

`\>`

Synonyme : `\medspace`. Normalement $4.0\text{mu plus } 2.0\text{mu minus } 4.0\text{mu}$. Avec le paquetage `amsmath`, ou depuis la parution 2020-10-01 de \LaTeX , utilisable aussi bien en mode texte qu'en mode mathématique ; sinon, utilisable seulement en mode mathématique.

`\negmedspace`

Normalement $-4.0\text{mu plus } 2.0\text{mu minus } 4.0\text{mu}$. Avec le paquetage `amsmath`, ou depuis la parution 2020-10-01 de \LaTeX , utilisable aussi bien en mode texte qu'en mode mathématique ; sinon, utilisable seulement en mode mathématique.

- `\,` Synonyme : `\thinspace`. Espace fine, normalement 3μ , c.-à-d. $1/6\mu$. Utilisable à la fois en mode mathématique et en mode texte (voir Section 19.8 [`\thinspace` & `\negthinspace`], page 197).
 Cette espace est très utilisée, par exemple entre la fonction et la variation infinitésimale dans une intégrale `\int f(x)\,dx` et, si un auteur fait cela, avant la ponctuation dans une équation affichée :
- ```

 La primitive est
 \begin{equation}
 3x^{-1/2}+3^{1/2}\,.\,
 \end{equation}

```
- `\!`      Une espace fine négative. Normalement  $-3\mu$ . Avec le paquetage `amsmath`, ou depuis la parution 2020-10-01 de  $\text{\LaTeX}$ , utilisable aussi bien en mode texte qu'en mode mathématique ; sinon, la commande `\!` est utilisable seulement en mode mathématique mais la commande `\negthinspace` a toujours également fonctionné en mode texte (voir Section 19.8 [`\thinspace` & `\negthinspace`], page 197).
- `\quad`    Vaut  $18\mu$ , c.-à-d.  $1\text{em}$ . Souvent utilisé pour l'espace entourant les équations ou expressions, par ex. pour l'espace entre deux équation au sein d'un environnement `displaymath`. Utilisable à la fois en mode mathématique et en mode texte.
- `\qqquad` Une longueur de 2 quads, soit  $36\mu = 2\text{em}$ . Utilisable à la fois en mode mathématique et en mode texte.

### 16.6.1 `\smash`

Synopsis :

```
\smash{sous-formule}
```

Compose *sous-formule* en aplatisant sa boîte comme si sa hauteur et sa profondeur étaient nulles.

Dans l'exemple suivant l'exponentielle est si haut que sans la commande `\smash`  $\text{\LaTeX}$  séparerait sa ligne de la ligne juste au-dessus, et l'irrégularité de l'espacement des lignes pourrait être visuellement malheureuse.

```

 Pour calculer la tétration $\smash{2^{2^{2^2}}}$,
 évaluer de haut en bas, comme $2^{2^4}=2^{16}=65536$.

```

(À cause du `\smash` l'expression imprimée pourrait se superposer à la ligne du dessus, aussi vous voudrez sans doute attendre la version finale du document avant de faire de tels ajustements).

Ce qui suit illustre l'effet de `\smash` en utilisant `\fbox` pour cerner la boîte que  $\text{\LaTeX}$  met sur la ligne. La commande `\barrenoire` fait une barre s'étendant de 10 points sous la ligne de base jusqu'à 20 points au-dessus.

```

 \newcommand{\barrenoire}{\rule[-10pt]{5pt}{30pt}}
 \fbox{\barrenoire}
 \fbox{\smash{\barrenoire}}

```

La première boîte que  $\text{\LaTeX}$  place fait 20 points de haut et 10 points de profondeur. Mais la seconde boîte est traitée par  $\text{\LaTeX}$  comme ayant une hauteur et une profondeur

nulles, même si l'encre imprimée sur la page s'étend toujours bien au-dessus et au-dessous de la ligne.

La commande `\smash` apparaît souvent en mathématiques pour ajuster la taille d'un élément qui entoure une sous-formule. Ci-dessous la première racine s'étend sous la ligne de base alors que la seconde repose juste sur la ligne de base.

```
\begin{equation}
\sqrt{\sum_{0\leq k < n} f(k)}
\sqrt{\vphantom{\sum}\smash{\sum_{0\leq k < n} f(k)}}
\end{equation}
```

Notez l'usage de `\vphantom` pour donner à la commande `\sqrt` un argument avec la même hauteur que la `\sum` (voir Section 16.6.2 [`\phantom` & `\vphantom` & `\hphantom`], page 180).

Alors que le plus souvent on l'utilise en mathématiques, la commande `\smash` peut apparaître dans d'autres contextes. Toutefois, elle n'assure pas le passage en horizontal. Ainsi si elle débute un paragraphe il est préférable de mettre en premier un `\leavevmode`, comme dans la dernière ligne de l'exemple ci-dessous :

```
xxx xxx xxx
```

```
\smash{yyy} % sans renforcement de paragraphe
```

```
\leavevmode\smash{zzz} % renforcement normal de paragraphe
```

Le paquetage `mathtools` a des opérateurs qui fournissent un contrôle encore plus fin de l'aplatissement d'une boîte de sous-formule.

### 16.6.2 `\phantom` & `\vphantom` & `\hphantom`

Synopsis :

```

```

ou

```
\vphantom{sous-formule}
```

ou

```
\hphantom{sous-formule}
```

La commande `\phantom` crée une boîte de la même hauteur, profondeur et largeur que *sous-formule*, mais vide. c.-à-d. qu'avec cette commande  $\LaTeX$  compose l'espace mais ne le remplit pas avec la matière. Dans l'exemple ci-dessous  $\LaTeX$  met un espace blanc de la largeur correcte pour la réponse, mais il n'affiche pas la réponse.

```
\begin{displaymath}
\int x^2 \, dx = \mbox{\underline{\$ \$}}
\end{displaymath}
```

La variante `\vphantom` produit une boîte invisible de la même taille verticale que *sous-formule*, la même hauteur et profondeur, mais ayant une largeur nulle. Et `\hphantom` fabrique une boîte avec la même largeur que *sous-formule* mais de hauteur et profondeur nulles.

Dans l'exemple suivant, la tour des exposants dans l'expression second opérande de la somme est si haute que  $\TeX$  place cette expression plus bas qu'il ne le ferait par défaut. Sans

ajustement, les deux expressions opérantes de la somme seraient à des niveaux différents. Le `\vphantom` dans la première expression dit à  $\TeX$  de laisser autant de place verticalement qu'il fait pour la tour, de sorte que les deux expressions ressortent au même niveau.

```
\begin{displaymath}
 \sum_{j\in\{0,\ldots,10\}}\vphantom{3^{3^j}}
 \sum_{i\in\{0,\ldots,3^{3^j}\}}i\cdot j
\end{displaymath}
```

Ces commandes sont souvent utilisées conjointement avec `\smash`. Voir Section 16.6.1 [`\smash`], page 179, ce qui inclut un autre exemple d'usage de `\vphantom`.

Les trois commandes en `phantom` apparaissent souvent mais notez que  $\LaTeX$  fournit nombre d'autres commandes affectant les tailles de boîtes qui pourraient être plus commodes, notamment `\makebox` (voir Section 20.1 [`\mbox` & `\makebox`], page 204) et également `\settodepth` (voir Section 14.4 [`\settodepth`], page 144), `\settoheight` (voir Section 14.5 [`\settoheight`], page 145), et `\settowidth` (voir Section 14.6 [`\settowidth`], page 145). De plus, le paquetage `mathtools` a beaucoup de commandes qui offrent un contrôle fin de l'espacement.

Les trois commandes produisent une boîte ordinaire, sans aucun statut mathématique spécial. Ainsi pour faire quelque chose comme attacher un indice supérieur vous avez à donner un tel statut, par exemple avec la commande `\operatorname` du paquetage `amsmath`.

Alors que la plupart du temps on les utilise en mathématiques, ces trois commandes peuvent apparaître dans d'autres contextes. Toutefois, elles ne provoquent pas le passage de  $\LaTeX$  en mode horizontal. Ainsi si l'une de ces commandes débute un paragraphe alors vous avez typiquement à la précéder d'un `\leavevmode`.

### 16.6.3 `\mathstrut`

Synopsis :

```
\mathstrut
```

L'homologue de `\strut` pour les mathématiques. Voir Section 19.13 [`\strut`], page 200.

Entrer `$$\sqrt{x} + \sqrt{x^i}$$` donne en sortie une seconde racine plus haute que la première. Pour ajouter de l'espace vertical supplémentaire sans aucun espace horizontal, de sorte que les deux aient la même hauteur, utilisez `$$\sqrt{x\mathstrut} + \sqrt{x^i\mathstrut}$$`.

La commande `\mathstrut` ajoute la hauteur verticale d'une parenthèse ouvrante, (, mais aucun espace horizontal. Elle est définie comme `\vphantom{ ( }` : voir Section 16.6.2 [`\phantom` & `\vphantom` & `\hphantom`], page 180, pour plus ample information. Son avantage par rapport à `\strut` est que `\mathstrut` n'ajoute aucune profondeur, ce qui est le plus souvent ce qui convient pour les formules. Utiliser la hauteur d'une parenthèse ouvrante est juste une convention ; pour un contrôle complet de la quantité d'espace, utilisez `\rule` avec une largeur. Voir Section 23.8 [`\rule`], page 236.

## 16.7 Styles mathématiques

Les règles de  $\TeX$  pour composer une formule dépendent du contexte. Par exemple, au sein d'une équation hors texte, l'entrée `\sum_{0\leq i<n}k^m=\frac{n^{m+1}}{m+1}+\mbox{termes d'ordre inférieur}` donnera une

sortie où l'indice de sommation sera centré sous le symbole de sommation. Mais si cette même entrée est en ligne dans le texte, alors l'indice de sommation est déporté sur la droite plutôt que au-dessous, de sorte qu'il n'écarte pas les lignes alentour. De même, dans un contexte en hors texte, les symboles au numérateur et dénominateur seront plus grands que pour une formule en ligne dans le texte, et en hors-texte mathématique les indices supérieurs et inférieurs sont plus écartés que pour une formule mathématique en ligne dans le texte.

$\TeX$  utilise quatre styles mathématiques.

- Le style affichage est pour une formule en hors texte sur une ligne séparée, comme avec `\begin{equation} ... \end{equation}`.
- Le style texte est pour une formule en ligne dans le texte, comme avec ‘ainsi nous avons  $\dots$ ’.
- Le style script style est pour les parties d'une formule dans un indice inférieur ou supérieur.
- Le style scriptscript est pour les parties d'une formule à un second niveau (ou plus) de placement d'indices inférieurs ou supérieurs.

$\TeX$  détermine un style mathématique par défaut mais vous pouvez l'emporter dessus avec une déclaration de `\displaystyle`, ou `\textstyle`, ou `\scriptstyle`, ou `\scriptscriptstyle`.

Dans l'exemple suivant, la fraction sur la ligne ‘Arithmétique’ aura l'air recroquevillée.

```
\begin{tabular}{r|cc}
\textsc{Nom} & \textsc{Séries} & \textsc{Somme} \\ \hline
Arithmétique & $a+(a+b)+(a+2b)+\cdots+(a+(n-1)b)$ & $\sum_{k=0}^{n-1} (a+kb)$ \\
Géométrique & $a+ab+ab^2+\cdots+ab^{n-1}$ & $a \sum_{k=0}^{n-1} b^k$
\end{tabular}
```

Mais à cause de la déclaration `\displaystyle`, la fraction de la ligne ‘Géométrique’ sera facile à lire, avec des caractères de la même taille que le reste de la ligne.

Un autre exemple est que, comparé à la même entrée sans la déclaration, le résultat de :

```
on obtient
 $\pi=2 \int_0^1 \sqrt{1-x^2} dx$
```

aura un signe intégrale bien plus haut. Notez que dans cet exemple le `\displaystyle` ne s'applique qu'à une partie de la formule, celle qui est délimitée entre accolades, comme ‘`\displaystyle ...`’.

Le dernier exemple est une fraction continue.

```
\begin{equation}
a_0+\frac{1}{
\displaystyle a_1+\frac{\mathstrut 1}{
\displaystyle a_2+\frac{\mathstrut 1}{
\displaystyle a_3}}}
\end{equation}
```

Sans la déclaration `\displaystyle`, les dénominateurs seraient en style script et scriptscript. (Le `\mathstrut` améliore la hauteur des dénominateurs ; voir Section 16.6.3 [`\mathstrut`], page 181).

## 16.8 Recueil de diverses commandes utiles en math

L<sup>A</sup>T<sub>E</sub>X contient moult facilités pour les mathématiques. En voici quelques-unes qu'on n'a pas pu classer ailleurs.

### 16.8.1 Caractère deux-points : & `\colon`

Synopsis, l'un parmi :

```
:
\colon
```

En mathématiques, le caractère deux-points, `:`, est une relation.

Avec des rapport de côtés `\( 3:4 \)` et `\( 4:5 \)`, le triangle est rectangle. ■

De base L<sup>A</sup>T<sub>E</sub>X définit `\colon` pour produire le caractère deux-points avec l'espace approprié pour une ponctuation, comme dans la notation de construction d'ensemble `\{x\colon 0\leq x<1\}`.

Mais le paquetage largement utilisé `amsmath` définit `\colon` pour être utilisé dans la définition de fonctions `f\colon D\to C`. Ainsi si vous désirez utiliser le caractère deux-points comme une ponctuation, utilisez alors `\mathpunct{:}`.

### 16.8.2 `\*`

Synopsis :

```
*
```

Un symbole de multiplication *discrétionnaire*, sur lequel un saut de ligne est autorisé. En l'absence de saut, la multiplication est indiquée implicitement par un espace, alors qu'en cas de saut un symbole `×` est imprimé immédiatement avant le saut. Ainsi

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage{french}
\begin{document}
Mais $(A_3 = 0)$, ainsi le produit de tous les termes de
 (A_1) à (A_4) , c.-à-d. $(A_1 * A_2 * A_3 * A_4)$,
est égal à zéro.
\end{document}
```

produit une sortie de ce type :

Mais  $A_3 = 0$ , ainsi le produit de tous les termes de  $A_1$  à  $A_4$ , c.-à-d.  $A_1 A_2 \times A_3 A_4$ , est égal à zéro.

### 16.8.3 `\frac`

Synopsis :

```
\frac{num}{dén}
```

Produit la fraction *num* divisé par *dén*. Utilisé comme : `\begin{displaymath}\frac{1}{\sqrt{2\pi}\sigma}\end{displaymath}`. En mode mathématique en ligne

cela ressort petit ; voir la discussion de `\displaystyle` (voir Chapitre 16 [Math formulas], page 153).

### 16.8.4 `\sqrt`

Synopsis, l'un parmi :

```
\sqrt{arg}
\sqrt[nième]{arg}
```

Produit la représentation de la racine carrée, ou optionnellement de la racine *nième*, de *arg*. L'argument optionnel *nième* détermine quelle racine produire, c.-à-d. saisissez `\sqrt[3]{x+y}` pour la racine cubique de  $x+y$ . Le résultat ressemble à cela :  $\sqrt[3]{x+y}$ . La taille du radical croît avec celle de *arg* (au fur et à mesure que la hauteur du radical croît, l'angle sur la montant de gauche devient plus abrupt, jusqu'à pour un *arg* suffisamment haut, devenir vertical).

L<sup>A</sup>T<sub>E</sub>X a un symbole `\surd` séparé pour faire une racine carrée sans *are* (voir Section 16.2 [Math symbols], page 155).

### 16.8.5 `\stackrel`

Synopsis :

```
\stackrel{texte}{relation}
```

Place *texte* au dessus de *relation*. Pour mettre un nom de fonction au-dessus d'une flèche entrez `\stackrel{f}{\longrightarrow}`. Le résultat ressemble à cela :  $\xrightarrow{f}$ .

## 17 Modes

Au fur et à mesure que L<sup>A</sup>T<sub>E</sub>X traite votre document, en tout point il est dans l'un parmi six modes. On peut les classer en trois catégories de deux modes chacune, les modes horizontaux, les modes mathématiques, et les modes verticaux. Certaines commandes ne fonctionnent pas dans tous les modes (en particulier, beaucoup de commandes ne fonctionnent que dans les modes mathématiques), et des messages d'erreur y font référence.

- *Mode paragraphe* (appelé *mode horizontal* en T<sub>E</sub>X de base) est le mode dans lequel L<sup>A</sup>T<sub>E</sub>X se trouve lorsqu'il traite du texte ordinaire. Il fragmente votre texte en lignes pour former un paragraphe et détermine les positions des sauts de ligne, de sorte que les sauts de page puissent être effectués en mode vertical. C'est le mode dans lequel L<sup>A</sup>T<sub>E</sub>X est la plupart du temps.

Le *mode LR* (de l'anglais Left-to-Right, c.-à-d. gauche-à-droite ; appelé *mode horizontal restreint* en T<sub>E</sub>X de base), ce mode est en vigueur quand L<sup>A</sup>T<sub>E</sub>X se met à fabriquer une boîte avec une commande `\mbox`. Comme en mode paragraphe, L<sup>A</sup>T<sub>E</sub>X considère la sortie qu'il produit comme une chaîne de mots avec des espaces entre eux. Contrairement au mode paragraphe, en mode LR L<sup>A</sup>T<sub>E</sub>X ne démarre jamais une nouvelle ligne, il continue juste à progresser de la gauche vers la droite. (Bien que L<sup>A</sup>T<sub>E</sub>X ne se plaigne pas qu'une boîte LR puisse être trop longue, quand il a fini et essaie ensuite de mettre cette boîte dans une ligne, il peut se plaindre si la boîte résultante est trop grande pour tenir dans la ligne).

- *Mode math*, L<sup>A</sup>T<sub>E</sub>X est dans ce mode quand il est en train de générer une formule mathématique.

*Mode math d'affichage*, L<sup>A</sup>T<sub>E</sub>X est dans ce mode quand il est en train de générer une formule mathématique hors texte. (Les formules hors texte diffèrent quelque peu des formules en ligne. un exemple est que le placement des indices sur `\int` diffère entre les deux situations).

- *mode vertical Mode vertical*, L<sup>A</sup>T<sub>E</sub>X est dans ce mode quand il est en train de construire la liste des lignes et d'autres matières pour fabriquer la page à sortir, ce qui comprend l'insertion des sauts de page. C'est le mode dans lequel L<sup>A</sup>T<sub>E</sub>X quand il démarre un document.

*Mode vertical interne*, ce mode est en vigueur quand L<sup>A</sup>T<sub>E</sub>X se met à fabriquer une `\vbox`. Il ne comprend pas l'insertion de sauts de page, ce qui fait de lui l'analogue vertical du mode LR.

Par exemple, si vous commencez un article en L<sup>A</sup>T<sub>E</sub>X avec 'Soit  $(x)$  un ...' alors voici les modes : d'abord L<sup>A</sup>T<sub>E</sub>X démarre tout document en mode vertical, ensuite il lit le 'S' et commute en mode paragraphe, et puis la commutation suivante survient sur le ' $($ ' où L<sup>A</sup>T<sub>E</sub>X passe en mode math, et puis quand il quitte la formule il revient en mode paragraphe.

Le mode Paragraphe a deux sous-cas. Si vous utilisez une commande `\parbox` ou une `minipage` alors L<sup>A</sup>T<sub>E</sub>X est mis en mode paragraphe. Mais il n'insérera pas de saut de page à cet endroit. Au sein de l'une de ces boîtes, appelées *parbox*, L<sup>A</sup>T<sub>E</sub>X est en *mode paragraphe interne*. Une situation plus habituelle, quand il peut faire des sauts de page, est le *mode paragraphe externe* (voir Chapitre 10 [Page breaking], page 117).

## 17.1 `\ensuremath`

Synopsis :

```
\ensuremath{formule}
```

La commande `\ensuremath` assure que *formule* est composée en mode mathématique.

Par exemple, vous pouvez redéfinir des commandes qui d'ordinaire ne peuvent être utilisées qu'en mode mathématique, de sorte qu'elles puissent être utilisées à la fois en mode mathématique et en texte simple.

```
\newcommand{\dx}{\ensuremath{dx}}
```

Dans `\int f(x)\, dx`, le `\dx` représente une variation infinitésimale.

Attention : la commande `\ensuremath` est utile mais ce n'est pas la panacée.

```
\newcommand{\alf}{\ensuremath{\alpha}}
```

Vous obtenez un alpha en mode texte : `\alf`.

Mais comparez l'espacement correct de `\alf+\alf` avec celui de `\alf+\alf`.

Le mieux est de composer toute matière mathématique en mode mathématique.

## 18 Styles de page

Le style d'une page détermine où  $\LaTeX$  place les composantes de cette page, telles que l'en-tête et le pied de page, et le corps de texte. Ceci inclut les pages de la partie principale du document mais inclut également les pages spéciales telles que la page de titre d'un livre, une page d'index, ou la première page d'un article.

Le paquetage `fancyhdr` est très utile pour construire des styles de page. Voir sa documentation sur le CTAN.

### 18.1 `\maketitle`

Synopsis :

```
\maketitle
```

La commande `\maketitle` génère un titre. Dans les classes standards le titre apparaît sur une page séparée — hormis dans la classe `article` où il est au sommet de la première page. (Voir Section 3.1 [Document class options], page 7), pour plus ample information sur l'option `titlepage` de classe de document).

L'exemple qui suit illustre la commande `\maketitle` apparaissant à sa place habituelle, immédiatement après `\begin{document}`.

```
\documentclass{article}
\title{Construction d'un réacteur nucléaire en n'utilisant que des noix de coco}
\author{Jonas Grumby\thanks{%
 Avec le soutien de Ginger Grant de la Roy Hinkley Society.} \
Skipper, \textit{Minnow}
\and
Willy Gilligan\thanks{%
 Remerciement à la fondation Mary Ann Summers
 et à Thurston et Lovey Howell.} \
Second, \textit{Minnow}
}
\date{26 septembre 1964}
\begin{document}
\maketitle
Assieds toi un peu pour entendre ce conte, le conte d'un voyage fatidique.
Partis de ce port sous les tropiques, à bord de ce petit navire. Le second était
un vieux loup de mer, le Skipper, courageux et sûr. Cinq passagers firent
voile ce jour-là, pour une balade de trois heures. Une balade de trois heures.
...
```

Vous donnez à  $\LaTeX$  l'information nécessaire à la production du titre en faisant les déclarations suivantes. Elles doivent se faire avant le `\maketitle`, soit dans le préambule, soit dans le corps du document.

```
\author{nom1 \and nom2 \and ...}
```

Obligatoire. Déclare l'auteur, ou les auteurs du document. L'argument est une liste d'auteurs séparés par des commandes `\and`. Pour séparer les lignes au sein

d'une seule entrée d'auteur, par exemple pour donner l'institution ou l'adresse de l'auteur, utiliser une double controbligue, `\\`. Si vous omettez la déclaration `\author` alors vous obtiendrez un avertissement 'LaTeX Warning: No \author given'.

`\date{texte}`

Optionnel. Déclare que *texte* est la date du document. Le *texte* n'est pas nécessairement dans un format de date ; cela peut être un texte quelconque. Si vous omettez `\date` alors L<sup>A</sup>T<sub>E</sub>X utilise la date courante (voir Section 23.9 [`\today`], page 236). Pour omettre la date, utilisez plutôt `\date{}`.

`\thanks{texte}`

Optionnel. Produit une note en bas de page. Vous pouvez l'utiliser dans l'information sur l'auteur pour insérer des remerciement comme illustré dans l'exemple plus haut, mais vous pouvez également l'utiliser dans le titre, ou en tout lieu où une marque de note en bas de page est susceptible de se trouver. Cela peut être un texte quelconque et donc vous pouvez l'utiliser pour n'importe quoi, comme pour imprimer une adresse mél.

`\title{texte}`

Obligatoire. Déclare que *texte* est le titre du document. On obtient des sauts de ligne au sein de *texte* avec une double controbligue, `\\`. Si vous omettez la déclaration `\title` alors la commande `\maketitle` produit l'erreur 'LaTeX Error: No \title given'.

Pour fabriquer votre propre page de titre, voir Section 8.26 [`\titlepage`], page 107. Vous pouvez soit la créer pour une unique fois ou l'inclure comme faisant partie d'une commande `\maketitle` redéfinie. (Beaucoup d'éditeurs fournissent une classe à utiliser à la place de `article` et formatant le titre selon les exigences de leur maison).

## 18.2 `\pagenumbering`

Synopsis :

```
\pagenumbering{style}
```

Spécifie le style des numéros de page, et réinitialise également le numéro de page. Le style de numérotation est reflété sur la page, et aussi dans la table des matières et d'autres références de page. Cette déclaration a une portée globale, ainsi son effet ne s'arrête pas à une fin de groupe comme une accolade fermante ou une fin d'environnement.

Dans l'exemple suivant, avant la rubrique 'Principale' les pages sont numérotées 'a', etc. À partir de la page contenant l'appel à `\pagenumbering` fait dans cette rubrique, les pages sont numérotées '1', etc.

```
\begin{document}\pagenumbering{alph}
...
\section{Principale}\pagenumbering{arabic}
...
```

L'argument *style* peut prendre les valeurs suivantes (voir aussi Section 13.1 [`\alph \Alph \arabic \roman \Roman \fnsymbol`], page 138) :

`arabic`    nombres arabes : 1, 2, ...

|                     |                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>roman</code>  | Nombres romain en bas de casse : i, ii, ...                                                                                                                                                                                                                                                                                                                                                |
| <code>Roman</code>  | Nombres romain en capitales : I, II, ...                                                                                                                                                                                                                                                                                                                                                   |
| <code>alph</code>   | Lettres en bas de casse : a, b, ... Si vous avez plus de 26 pages alors vous obtenez ‘ <code>LaTeX Error: Counter too large</code> ’.                                                                                                                                                                                                                                                      |
| <code>Alph</code>   | Lettre en capitales : A, B, ... Si vous avez plus de 26 pages alors vous obtenez ‘ <code>LaTeX Error: Counter too large</code> ’.                                                                                                                                                                                                                                                          |
| <code>gobble</code> | $\LaTeX$ ne produit pas de numéro de page en sortie, bien qu’il ne soit pas réinitialisé. Les références à cette page sont également à blanc. (Ceci ne fonctionne pas avec le paquetage populaire <code>hyperref</code> , de sorte que pour empêcher le numéro de page d’apparaître une alternative est d’utiliser <code>\pagestyle{empty}</code> ou <code>\thispagestyle{empty}</code> .) |

Traditionnellement, si un document a des matières préliminaires — préface, table des matières, etc. — alors les pages y sont numérotées en chiffres romains bas de casse. Les pages de la matière principale d’un document sont en chiffres arabes. Voir Section 6.7 [`frontmatter & \mainmatter & \backmatter`], page 50.

Si vous désirez changer l’emplacement du numéro de page sur la page, voir Section 18.3 [`\pagestyle`], page 189. Si vous désirez changer la valeur du numéro de page, alors manipulez le compteur `page` (voir Chapitre 13 [`Counters`], page 138).

### 18.3 `\pagestyle`

Synopsis :

```
\pagestyle{style}
```

Déclaration spécifiant comment les en-têtes et pieds de page sont composés à partir de la page courante.

Une discussion avec un exemple suivent. À noter tout d’abord que le paquetage `fancyhdr` est actuellement la manière standard de manipuler les en-têtes et pieds de page. Les nouveaux documents pour lesquels on désire quoi que ce soit d’autre que les options standards listées ci-dessous devraient utiliser ce paquetage. Se référer à sa documentation sur le CTAN.

Valeurs pour *style* :

|                       |                                                                                                                                         |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| <code>plain</code>    | L’en-tête est vide. Le pied de page ne contient qu’un numéro de page, centré.                                                           |
| <code>empty</code>    | L’en-tête et le pied de page sont tous deux vides.                                                                                      |
| <code>headings</code> | Place des en-têtes et pieds de page sur chaque page. Le style de document spécifie ce qu’ils contiennent ; voir la discussion plus bas. |

`myheadings`

En-têtes personnalisés, spécifiés via les commandes `\markboth` ou `\markright`.

Discuter la motivation du mécanisme de  $\LaTeX$  vous aidera à faire fonctionner les options `headings` ou `myheadings`. Le document source ci-dessous produit un article, en recto-verso, avec le style de page `headings`. Sur les pages de gauche de ce document,  $\LaTeX$  désire (en plus du numéro de page) le titre de la rubrique de niveau `\section` en cours. Sur les pages de droite  $\LaTeX$  désire le titre de la rubrique de niveau `\subsection` en cours. Quand il fabrique une page,  $\LaTeX$  obtient ces informations à partir des commandes `\leftmark` et

`\rightmark`. De sorte que c'est aux commandes `\section` et `\subsection` de stocker ces informations là.

```
\documentclass[twoside]{article}
\pagestyle{headings}
\begin{document}
... \section{Section 1} ... \subsection{Subsection 1.1} ...
\section{Section 2}
...
\subsection{Subsection 2.1}
...
\subsection{Subsection 2.2}
...
```

Supposez que la seconde rubrique de niveau `\section` tombe sur une page de gauche. Même si quand la page commence elle est dans la première rubrique de niveau `\section`,  $\LaTeX$  placera ‘Section 2’ dans l’en-tête de la page de gauche. Quant à l’en-tête de droite, si aucune rubrique de niveau `\subsection` ne commence avant la fin de la page de droite alors  $\LaTeX$  laisse en blanc l’en-tête de droite. Si une rubrique de niveau `\subsection` apparaît avant que la page de droite ne finisse alors il y a deux cas. Si au moins une rubrique de niveau `\subsection` commence sur la page de droite alors  $\LaTeX$  place dans l’en-tête de droite le titre de la première rubrique de niveau `\subsection` commençant sur cette page de droite. Si au moins l’un parmi 2.1, 2.2, ..., commence sur la page de gauche mais qu’aucune ne commence sur la droite alors  $\LaTeX$  place dans l’en-tête de droite le titre de la dernière rubrique de niveau `\subsection` à avoir commencé, c.-à-d. celle qui est en cours pendant la page de droite.

Pour accomplir cela, dans un article en recto-verso,  $\LaTeX$  fait que `\section` génère une commande `\markboth` réglant `\leftmark` à ‘Section 2’ et réglant `\rightmark` à un contenu vide. Et,  $\LaTeX$  fait que `\subsection` génère une commande `\markright`, réglant `\rightmark` à ‘Subsection 2.1’, etc.

Ci-après on trouvera les descriptions de `\markboth` et `\markright` :

`\markboth{en-tête-gauche}{en-tête-droite}`

Règle l’information d’en-tête à la fois sur la page de gauche et sur celle de droite dans le cas d’un style de page `headings` ou `myheadings`. Un en-tête *en-tête-gauche* de page de gauche est généré par la dernière commande `\markboth` avant la fin de la page. Un en-tête *en-tête-droite* de page de droite est généré par le premier `\markboth` ou `\markright`, s’il y en a un, de la page, ou sinon par le dernier se trouvant avant cette page.

`\markright{en-tête-droite}`

Règle l’en-tête de la page de droite, en laissant celle de gauche inchangée.

## 18.4 `\thispagestyle{style}`

La commande `\thispagestyle` fonctionne de la même manière que la commande `\pagestyle` (voir section précédente) à ceci près qu’elle change en *style* pour la page courante seulement.

## 19 Espaces

L<sup>A</sup>T<sub>E</sub>X a beaucoup de façons de produire de l'espace blanc (ou rempli). Certaines d'entre elles conviennent mieux à du texte mathématique à ce propos, voir Section 16.6 [Spacing in math mode], page 178.

### 19.1 `\enspace` & `\quad` & `\qquad`

Synopsis, l'un parmi :

```
\enspace
\quad
\qquad
```

Insère un espace horizontal de 1/2 em, 1 em, ou 2 em. Le em est une longueur définie par les auteurs de polices de caractères, généralement considérée comme la largeur d'un M capital. Un avantage de décrire l'espace en termes d'em est que cela peut être plus portable d'un document à l'autre qu'utiliser une mesure absolue telle que des points (voir [Lengths/em], page 143).

L'exemple suivant met l'espace adéquat entre les deux graphiques.

```
\begin{center}
\includegraphics{souriredefemmes.png}%
\qquad\includegraphics{souriredemecs.png}
\end{center}
```

Voir Section 16.6 [Spacing in math mode], page 178, pour `\quad` et `\qquad`. Ce sont des longueurs utilisées depuis des siècles en typographie et donc dans de nombreux cas seraient un meilleur choix que des longueurs arbitraires, comme celles que l'on obtient avec `\hspace`.

### 19.2 `\hspace`

Synopsis, l'un parmi :

```
\hspace{longueur}
\hspace*{longueur}
```

La commande `\hspace` ajoute la quantité d'espace horizontal *longueur*. L'argument *longueur* peut être positif, négatif, ou nul ; ajouter une longueur négative d'espace revient à retourner en arrière. C'est une longueur élastique, c.-à-d. qu'elle peut contenir une composante `plus` ou `minus`, ou les deux (voir Chapitre 14 [Lengths], page 141). Comme l'espace est dilatable ou contractible on l'appelle parfois *glue*.

L'exemple suivant fabrique une ligne dans laquelle 'Nom~:' est à 2,5 cm de la marge de droite.

```
\noindent\makebox[\linewidth][r]{Nom~:\hspace{2.5cm}}
```

La forme en `*` insère de l'espace horizontal non supprimable. Plus précisément, quand T<sub>E</sub>X découpe un paragraphe en lignes, tout espace blanc — glues ou créneaux — qui se trouve à un saut de ligne est supprimé. La forme étoilée évite ceci (techniquement, elle ajoute un objet invisible non-supprimable devant l'espace).

Dans l'exemple suivant :

```
\documentclass[a4paper]{article}
```

```

\begin{document}
\parbox{.9\linewidth}{Remplir chaque blanc~: Le peuple français proclame
 solennellement son attachement aux Droits de \hspace*{2.5cm} et aux
 principes de la \hspace*{2.5cm} nationale tels qu'ils ont été définis
 par la Déclaration de 1789, \ldots}
\end{document}

```

le blanc de 2,5 cm suivant ‘principes de la’ tombe au début de la ligne. Si vous enlevez la \* alors L<sup>A</sup>T<sub>E</sub>X supprime le blanc.

Ci-dessous le `\hspace` sépare les trois graphiques.

```

\begin{center}
\includegraphics{lion.png}% le commentaire empêche l'ajout d'un espace
\hspace{1cm minus 0.25cm}\includegraphics{tiger.png}%
\hspace{1cm minus 0.25cm}\includegraphics{bear.png}
\end{center}

```

Parce que l’argument de chaque `\hspace` comprend `minus 0.25cm`, chacun peut se contracter un peu si les trois figures sont trop larges. Mais chaque espace ne peut contracter plus que 0,25 cm (voir Chapitre 14 [Lengths], page 141).

### 19.3 `\hfill`

La commande `\hfill` de remplissage produit une “longueur élastique” qui n’a aucun espace naturel mais peut s’allonger ou se rétrécir horizontalement autant que nécessaire.

Le paramètre `\fill` est la longueur élastique elle-même (c.-à-d. pour parler technique, la valeur de glue ‘`0pt plus1fill`’) ; ainsi, `\hspace\fill` équivaut à `\hfill`.

### 19.4 `\hss`

Synopsis :

```
\hss
```

Produit un espace horizontal indéfiniment dilatable ou contractible (cette commande est une primitive T<sub>E</sub>X). Les auteurs L<sup>A</sup>T<sub>E</sub>X devraient chercher en premier la commande `\makebox` pour obtenir les effets de `\hss` (voir Section 20.1 [`\mbox` & `\makebox`], page 204).

Dans l’exemple suivant, le `\hss` de la première ligne a pour effet de coller le Z à droite, le superposant au Y. Dans la seconde ligne, le Z dépasse sur la gauche, et se superpose au X.

```

X\hbox to 0pt{Z\hss}Y
X\hbox to 0pt{\hss Z}Y

```

Sans le `\hss` vous auriez quelque chose du genre ‘`Overfull \hbox (6.11111pt too wide) detected at line 20`’.

### 19.5 `\spacefactor`

Synopsis :

```
\spacefactor=entier
```

Influence le comportement de L<sup>A</sup>T<sub>E</sub>X en termes de dilatation / contraction de glue. La plupart des documents de niveau d’utilisateur ne font pas usage de cette commande.

Alors que L<sup>A</sup>T<sub>E</sub>X compose la matière, il peut dilater ou contracter les espaces entre les mots. (Cet espace n'est pas un caractère ; on l'appelle la *glue inter-mot* ; voir Section 19.2 [`\hspace`], page 191). La commande `\spacefactor` (issue de T<sub>E</sub>X de base) vous permet, par exemple, d'avoir l'espace après un point qui se dilate plus que l'espace après une lettre terminant un mot.

Après que L<sup>A</sup>T<sub>E</sub>X a placé chaque caractère, ou filet ou autre boîte, il règle un paramètre qu'on appelle le *facteur d'espace*. Si ce qui suit en entrée est une espace alors ce paramètre affecte la quantité de dilatation ou de contraction pouvant survenir. Un facteur d'espace plus grand que la valeur normale signifie que la glue peut se dilater davantage, et moins se contracter. Normalement, le facteur d'espace vaut 1000. Cette valeur est en vigueur après la plupart des caractères, et toute boîte qui n'est pas un caractère, ou formule mathématique. Mais il vaut 3000 après un point, un point d'exclamation, ou d'interrogation, il vaut 2000 après un caractère deux-points, 1500 après un point-virgule, 1250 après une virgule, et 0 après une parenthèse fermante ou un crochet fermant, ou un guillemet anglais fermant double, ou simple. Finalement, il faut 999 après une lettre capitale.

Si le facteur d'espace  $f$  vaut 1000 alors l'espace de glue sera de la taille normale d'une espace dans la police courante (pour Computer Modern Roman en 10 point, cela vaut 3,3333 points). Sinon, si le facteur d'espace  $f$  est supérieur à 2000 alors T<sub>E</sub>X ajoute la valeur d'espace supplémentaire définie pour la police courante (pour Computer Modern Roman en 10 point cela vaut 1,11111 points), et ensuite la valeur de dilatation normale de la police est multipliée par  $f/1000$  et la valeur normale de contraction par  $1000/f$  (pour Computer Modern Roman en 10 point elles valent 1,66666 et 1,11111 points).

Par exemple, considérez le point à la fin de **Le meilleur ami d'un homme est son chien**. Après celui-ci, T<sub>E</sub>X place un espace supplémentaire fixé, et également permet à la glue de se dilater trois fois plus et de se contracter par 1/3 fois plus que la glue après **ami**, ce mot ne se terminant pas sur un point.

Les règles de fonctionnement des facteurs d'espace sont en fait encore plus complexes car elles jouent d'autres rôles. En pratique, il y a deux conséquences. Tout d'abord, si un point ou une autre ponctuation est suivie d'une parenthèse ou d'un crochet fermant, ou d'un guillemet anglais fermant simple ou double, alors l'effet sur l'espacement de ce point se propage à travers ces caractères (c.-à-d. que la glue qui suit aura plus de dilatation ou de contraction). Deuxièmement, si la ponctuation suit une lettre capitale alors son effet n'est pas là de sorte que vous obtenez un espacement ordinaire. Ce second cas affecte également les abréviations qui ne se terminent pas par une lettre capitale (voir Section 19.5.1 [`\@`], page 193).

Vous pouvez utiliser `\spacefactor` seulement en mode paragraphe ou en mode LR (voir Chapitre 17 [Modes], page 185). Vous pouvez voir la valeur courante avec `\the\spacefactor` ou `\showthe\spacefactor`.

(Commentaire, ne concernant pas vraiment `\spacefactor` : si vous avez des erreurs du genre de 'You can't use '`\spacefactor`' in vertical mode', ou 'You can't use '`\spacefactor`' in math mode.', ou 'Improper `\spacefactor`' alors c'est que vous avez probablement essayer de redéfinir une commande interne. Voir Section 12.3 [`\makeatletter` & `\makeatother`], page 127.)

### 19.5.1 \@

Synopsis :

`lettre-capitale\@.`

Traite un point comme terminant une phrase, là où L<sup>A</sup>T<sub>E</sub>X penserait sinon qu'il fait partie d'une abréviation. L<sup>A</sup>T<sub>E</sub>X pense qu'un point termine une abréviation si le point suit une lettre capitale, et sinon il pense que ce point termine une phrase. Par défaut, en justifiant une ligne L<sup>A</sup>T<sub>E</sub>X ajuste l'espacement après un point terminant une phrase (ou après un point d'interrogation, d'exclamation, une virgule, ou un caractère deux-points) plus qu'il n'ajuste l'espace inter-mot (voir Section 19.5 [`\spacefactor`], page 192).

Cet exemple montre les deux cas dont il faut se souvenir :

Les chansons `\textit{Red Guitar}`, etc.\ sont de Loudon Wainwright~III\@.

Le second point termine la phrase, bien que précédé d'une capitale. On dit à L<sup>A</sup>T<sub>E</sub>X qu'il termine la phrase en mettant `\@` devant. Le premier point termine l'abréviation 'etc.' mais non la phrase. Le controblique-espace, `\`, produit un espace en milieu de phrase.

Ainsi donc : lorsqu'on a une capitale suivie d'un point qui termine une phrase, alors on met `\@` devant le point. Cela vaut même s'il y a une parenthèse droite ou un crochet, ou un guillemet droit simple ou double, parce que la portée de l'effet sur l'espacement du point n'est pas limitée par ces caractères. Dans l'exemple suivant

Utilisez le `\textit{Instructional Practices Guide}`, (un livre de la MAA)\@.

on obtient l'espacement inter-phrase correctement après le point.

La commande `\@` ne convient que pour le mode texte. Si vous l'utilisez hors de ce mode alors vous obtiendrez 'You can't use '`\spacefactor`' in vertical mode' (voir Chapitre 17 [Modes], page 185).

Commentaire : le cas réciproque c'est lorsque un point termine une abréviation dont la dernière lettre n'est pas une capitale, et que cette abréviation n'est pas le dernier mot de la phrase. Pour ce cas faites suivre le point d'une controblique-espace, (`\`), ou un tilde, (`~`), ou `\@`. Par exemple `Nat.\ Acad.\ Science, Mr.~Bean, et (fumier, etc.\@) à vendre` (notez que dans le dernier cas `\@` est placé avant la parenthèse fermante).

### 19.5.2 `\frenchspacing` & `\nonfrenchspacing`

Synopsis, l'un parmi :

```
\frenchspacing
\nonfrenchspacing
```

La première déclaration a pour effet que L<sup>A</sup>T<sub>E</sub>X traite l'espacement entre les phrases de la même manière que l'espacement entre les mots d'une phrase. L'effet de la seconde est de revenir au traitement par défaut où l'espacement entre les phrases se dilate ou contracte davantage (voir Section 19.5 [`\spacefactor`], page 192).

Dans certaines traditions typographiques, y compris l'anglaise, on préfère ajuster l'espace entre les phrases (ou les espaces suivant un point d'interrogation, d'exclamation, une virgule, ou un point virgule) plus que l'espace entre les mots du milieu de la phrase. La déclaration `\frenchspacing` (la commande vient de T<sub>E</sub>X de base) commute vers la tradition où tous les espaces sont traités de la même façon.

### 19.5.3 `\normalsfcodes`

Synopsis :

```
\normalsfcodes
```

Réinitialise les facteurs d'espacement de L<sup>A</sup>T<sub>E</sub>X à leur valeur défaut (voir Section 19.5 [`\spacefactor`], page 192).

## 19.6 Controbligue-espace `\`

Cette rubrique concerne la commande qui consiste en deux caractères, une controbligue suivie d'un espace.

Synopsis :

```
\
```

Produit un espace. Par défaut elle produit de l'espace blanc de longueur 3.33333 pt plus 1.66666 pt minus 1.11111 pt.

Quand vous tapez un ou plusieurs blancs entre des mots, L<sup>A</sup>T<sub>E</sub>X produit de l'espace blanc. Mais ceci diffère d'un espace explicite, comme illustré ci-dessous :

```
\begin{tabular}{r1}
Un blanc~:& fabrique de l'espace\\
Trois blancs~:& à la suite\\
Trois espaces~:& \ \ \ à la suite\\
\end{tabular}
```

Sur la première ligne L<sup>A</sup>T<sub>E</sub>X met de l'espace après le deux-points. Sur la deuxième ligne L<sup>A</sup>T<sub>E</sub>X compte les trois blancs comme un seul pour mettre un seul espace blanc, de sorte que vous obtenez après le deux-points le même espace que dans la première ligne. L<sup>A</sup>T<sub>E</sub>X compterait de la même façon un seul espace blanc si l'un ou plusieurs des blancs étaient remplacés par des caractères tabulation ou saut de ligne. Toutefois, la ligne du bas demandant trois espaces la zone en blanc est plus large. c.-à-d. que la commande controbligue-espace crée une quantité fixe d'espace horizontal. (Notez que vous pouvez définir un espace horizontal d'une largeur quelconque avec `\hspace`; voir Section 19.2 [`\hspace`], page 191).

La commande controbligue-espace a deux usages principaux. Elle est souvent utilisée après une séquence de contrôle pour l'empêcher d'avaler le blanc qui suit, comme après `\TeX` dans `\TeX\` (ou `\LaTeX`). (Mais l'utilisation d'accollades a l'avantage de fonctionner systématiquement, que la séquence de contrôle soit suivie d'un blanc ou de tout autre non-lettre, comme dans `\TeX{}` (ou `\LaTeX{}`) où `{}` peut être rajouté aussi bien après `\LaTeX` qu'après `\TeX`). L'autre utilisation usuelle est de marquer qu'un point termine une abréviation et non une phrase, comme dans `Prof.\ Fabre or Jean et al.\ (1993)` (voir Section 19.5.1 [`\@`], page 193).

Dans les circonstances normales, `\tabulation` et `\sautdeligne` sont équivalents à controbligue-espace, `\`.

Veillez aussi noter que de sorte à autoriser le renforcement du code, dans les circonstances normales, T<sub>E</sub>X ignore les blancs au début de chaque ligne. Ainsi l'exemple suivant imprime 'À moi comte deux mots':

```
À moi, Comte,
deux mots !
```

où l'espace blanc entre ‘Comte,’ et ‘deux’ est produit par le saut de ligne après ‘Comte’, et non par l'espace avant ‘deux’.

## 19.7 ~

Synopsis :

*avant~après*

Le caractère tilde, ~, produit une espace *insécable* entre *avant* et *après*, c'est à dire une espace où un saut de ligne n'est pas autorisé. Par défaut l'espace blanc a une longueur de 3.33333 pt plus 1.66666 pt minus 1.11111 pt (voir Chapitre 14 [Lengths], page 141).

Dans l'exemple suivant L<sup>A</sup>T<sub>E</sub>X ne fait pas de saut de ligne entre les deux derniers mots.

Remerciements au Prof.~Lerman.

De plus, en dépit du point, L<sup>A</sup>T<sub>E</sub>X n'utilise pas l'espacement de fin de phrase (voir Section 19.5.1 [\@], page 193).

Les insécables empêchent qu'une fin de ligne sépare des choses là où cela prêterait à confusion. Ils n'empêchent par contre pas la césure (de l'un ou l'autre des mots de part et d'autre), de sorte qu'ils sont en général préférables à mettre deux mots consécutifs dans une `\mbox` (voir Section 20.1 [\mbox & \makebox], page 204).

Ils sont aussi une question de goût, parfois très dogmatique, parmi les lecteurs. Néanmoins, on présente ci-après quelques usages, dont la plupart sont issus du *T<sub>E</sub>Xbook*.

- Entre l'étiquette et le numéro d'un énumérateur, comme dans les références : `Chapitre~12`, or `Théorème~\ref{th:Wilsons}`, ou `Figure~\ref{fig:KGraph}`.
- Lorsque des alternatives sont énumérés en ligne : (b)~Montrez que  $f(x)$  est (1)~continue, et (2)~bornée.
- Entre un nombre et son unité : `$745.7.8$~watts` (le paquetage `siunitx` a des fonctions spécifiques pour cela) ou `144~œufs`. Ceci comprend le cas du quantième et du nom du mois dans une date : `12~octobre` ou `12~oct`. En général, dans toute expression où un nombre et une abréviation ou un symbole sont séparés par une espace : `565~ap.` ~`\mbox{J.-C.}`, ou `2:50~pm`, ou `Airbus~A320`, ou `26~rue des Oliviers Road`, ou `\$$1.4$~billion`. D'autres choix usuels sont l'espace fine (voir Section 19.8 [\thinspace & \negthinspace], page 197) et aucune espace du tout.
- Quand des expressions mathématiques sont rendues en mots : `égal~$n$`, ou `inférieur à~$\epsilon$`, ou `soit~$X$`, ou `modulo~$p^e$ pour tout~$n$` (mais comparez `vaut~$15$` avec `vaut $15$~fois la hauteur`). Entre symboles mathématiques en apposition avec des noms : `dimension~$d$` ou `fonction~$f(x)$` (mais comparez avec `de longueur $1$~ou plus`). Quand un symbole est un objet étroitement lié à une préposition : `de~$x$`, ou `de $0$ à~$1$`, ou `en commun avec~$m$`.
- Entre les symboles d'une série : `$1$,~$2$, ou~$3$` ou `$1$,~$2$, \ldots,~$n$`.
- Entre les prénoms d'une personne et entre des noms de famille multiples : `Donald~E. Knuth`, ou `Luis~I. Trabb~Pardo`, ou `Charles~XII` — mais vous devez donner à T<sub>E</sub>X des endroits où faire un saut de ligne, par exemple comme dans `Charles Louis Xavier~Joseph de~la Vall\`ee~Poussin`.

## 19.8 `\thinspace` & `\negthinspace`

Synopsis, l'un parmi :

```
\thinspace
\negthinspace
```

Produisent une espace insécable et non-élastique qui vaut  $1/6$  d'em et  $1/6$  d'em respectivement. Ce sont les équivalents mode texte de `\`, and `\!` (voir [Spacing in math mode/`\thinspace`], page 179).

On peut utiliser `\`, comme synonyme de `\thinspace` en mode texte.

Une utilisation usuelle de `\thinspace` est en typographie anglaise comme l'espace entre des guillemets anglais simples emboîtés :

Killick répondit : ‘‘J’ai entendu le capitaine dire ‘Ohé du bateau~!’`\thinspace`’’■

Une autre utilisation est que certains guides de style demandent un `\thinspace` entre les points de suspension et un point terminant une phrase (dans d’autres guides de style on considère que les points de suspension et/ou les points de suspension et le quatrième point suffisent). Une autre utilisation spécifique au style est entre les initiales, comme dans `D.\thinspace E.\ Knuth`.

$\LaTeX$  fournit diverses commandes d’espacement similaires pour le mode mathématique (voir Section 16.6 [Spacing in math mode], page 178). Avec le paquetage `amsmath`, ou depuis la parution 2020-10-01 de  $\LaTeX$ , elle sont utilisables aussi bien en mode texte qu’en mode mathématique, y compris `\!` pour `\negthinspace` ; mais sinon, elles ne sont disponibles qu’en mode mathématique.

## 19.9 `\/`

Synopsis :

```
caractère-précédent\/caractère-suivant
```

La commande `\/` produit une *correction italique*. C’est une petite espace définie par l’auteur de la fonte pour chacun des caractères (elle peut être nulle). Quand vous utilisez `\/`,  $\LaTeX$  prend la correction dans le fichier de métriques de fonte, la dilate/contracte de la même façon que la fonte si tant est que la taille de celle-ci fût dilatée/contractée, et ensuite insère cette quantité d’espace.

Dans l’exemple ci-dessous, sans le `\/`, le *caractère-précédent* f italique taperait dans le *caractère-suivant* H romain :

```
\newcommand{\logoentreprise}{\it f}\/H}
```

parce que la lettre f italique penche loin vers la droite.

Si *caractère-suivant* est un point ou une virgule, il n’est pas nécessaire d’insérer une correction italique, puisque ces symboles de ponctuation sont d’une faible hauteur. Toutefois, avec les points-virgules ou les deux-points, de même qu’avec les lettres normales, elle peut être utile. On l’utilise typiquement à l’endroit où l’on passe d’une fonte italique ou inclinée à une fonte droite.

Quand vous utilisez des commandes telles que `\emph`, `\textit` ou `\textsl` pour changer la fonte,  $\LaTeX$  insère automatiquement la correction italique là où besoin est (voir Section 4.2 [Font styles], page 23). Toutefois, des déclarations telle que `\em`, `\itshape` ou `\slshape` n’insèrent pas automatiquement de correction italique.

Les caractères droits peuvent aussi avoir une correction italique. Un exemple où elle est nécessaire est le nom `pdf\TeX`. Toutefois, la plupart des caractères droits ont une correction italique nulle. Certains auteurs de fonte n'incluent pas de valeur de correction italique même pour les fontes italiques.

Techniquement,  $\text{\LaTeX}$  utilise une autre valeur spécifique à la fonte, appelé *paramètre d'inclinaison* (nommé `\fontdimen1`), pour déterminer s'il est possible d'insérer une correction italique, au lieu d'attacher cette action à des commandes particulières de contrôle de la fonte.

Le concept de correction italique n'existe pas en mode mathématique ; l'espacement y est fait d'une façon différente.

## 19.10 `\hrulefill` & `\dotfill`

Synopsis, l'un parmi :

```
\hrulefill
\dotfill
```

Produit une longueur élastique infinie (voir Chapitre 14 [Lengths], page 141) remplie avec un filet horizontal (c.-à-d. une ligne) ou avec des pointillés, au lieu d'espace blanc.

L'exemple ci-dessous produit une ligne de 5 cm de long :

```
Nom: ~\makebox[5cm]{\hrulefill}
```

Quand on le place entre deux lignes à blanc cet exemple crée un paragraphe qui est justifié à gauche et à droite, et où l'espace au milieu est rempli avec des pointillés uniformément espacés.

```
\noindent Cap\ Robert Surcouf\dotfill{} Saint-Malo
```

Pour que le filet ou les pointillés aillent jusqu'au bout de la ligne utilisez `\null` au début ou à la fin.

Pour changer l'épaisseur du filet, copiez la définition et ajustez là, comme ci-dessous :

```
\renewcommand{\hrulefill}{%
\leavevmode\leaders\hrule height 1pt\hfill\kern\z@}
```

ce qui change l'épaisseur par défaut de 0.4 pt à 1 pt. De même, ajustez l'espacement inter-pointillé comme avec :

```
\renewcommand{\dotfill}{%
\leavevmode\cleaders\hb@xt@1.00em{\hss .\hss }\hfill\kern\z@}
```

ce qui change la longueur par défaut de 0.33 em à 1.00 em.

L'exemple suivant produit une ligne pour signer :

```
\begin{minipage}{4cm}
\centering
Signature:\
\hrulefill
\end{minipage}
```

La ligne fait 4 cm de long.

## 19.11 `\bigskip` & `\medskip` & `\smallskip`

Synopsis, l'un parmi :

```
\bigskip
\medskip
\smallskip
```

Produit une certaine quantité d'espace vertical, grand, de taille moyenne, ou petit. Ces commandes sont fragiles (voir Section 12.11 [`\protect`], page 134).

Dans l'exemple ci-dessous le saut suggère le passage du temps (tiré de *L'océan d'or* de Patrick O'Brian).

```
Mr Saumarez aurait eu à lui dire quelque grossièreté, sans aucun
doute : il était de nouveau comme chez lui, et c'était un plaisir.
```

```
\bigskip
« Cent cinquante-sept miles et un tiers, en vingt-quatre heures », dit Peter.■
```

Chaque commande est associée à une longueur définie dans le source de la classe du document.

`\bigskip` La même chose que `\vspace{\bigskipamount}`, ordinairement environ l'espace d'une ligne, avec dilatation et rétrécissement. La valeur par défaut pour les classes `book` et `article` est 12pt plus 4pt minus 4pt.

`\medskip` La même chose que `\vspace{\medskipamount}`, ordinairement environ la moitié de l'espace d'une ligne, avec dilatation et rétrécissement. La valeur par défaut pour les classes `book` et `article` est 6pt plus 2pt minus 2pt).

`\smallskip` La même chose que `\vspace{\smallskipamount}`, ordinairement environ le quart de l'espace d'une ligne, avec dilatation et rétrécissement. la valeur par défaut pour les classes `book` et `article` est 3pt plus 1pt minus 1pt.

Comme chaque commande est un `\vspace`, si vous l'utilisez au milieu d'un paragraphe, alors elle insère son espace vertical entre la ligne dans laquelle vous l'utilisez et la ligne suivante, et non pas nécessairement à l'endroit où vous l'utilisez. Ainsi il est préférable d'utiliser ces commandes entre les paragraphes.

Les commandes `\bigbreak`, `\medbreak`, et `\smallbreak` sont similaires mais suggèrent également à L<sup>A</sup>T<sub>E</sub>X que l'endroit est adéquat pour un saut de page (voir Section 19.12 [`\bigbreak` & `\medbreak` & `\smallbreak`], page 199).

## 19.12 `\bigbreak` & `\medbreak` & `\smallbreak`

Synopsis, l'un parmi :

```
\bigbreak
\medbreak
\smallbreak
```

Produit un espace vertical grand, de taille moyenne ou petit, et suggère à L<sup>A</sup>T<sub>E</sub>X que l'endroit est adéquat pour un saut de page. (Les pénalités associées sont respectivement -200, -100, et -50).

Voir Section 19.11 [`\bigskip` & `\medskip` & `\smallskip`], page 199, pour plus ample information. Ces commandes produisent le même espace vertical mais différent en ce qu’elles suppriment également un espace vertical précédent s’il est inférieur à ce qu’elles inséreraient (comme avec `\addvspace`). De plus, elles terminent le paragraphe où on les utilise. Dans cet exemple :

```
abc\bigbreak def ghi
```

```
jkl mno pqr
```

on a en sortie trois paragraphes, le premier se termine en ‘abc’ et le deuxième commence, après un espace vertical supplémentaire et un renforcement, par ‘def’.

### 19.13 `\strut`

Synopsis :

```
\strut
```

Étai. Assure que la ligne courante a une hauteur d’au moins `0.7\baselineskip` et une profondeur d’au moins `0.3\baselineskip`. Pour l’essentiel,  $\text{\LaTeX}$  insère dans la ligne un rectangle de largeur nulle, `\rule[-0.3\baselineskip]{0pt}{\baselineskip}` (voir Section 23.8 [`\rule`], page 236). Le `\baselineskip` change avec la fonte courante ou sa taille.

Dans l’exemple suivant le `\strut` empêche la boîte d’avoir une hauteur nulle.

```
\setlength{\fboxsep}{0pt}\framebox[5cm]{\strut}
```

L’exemple suivant comprend quatre listes. Dans la première il y a un espace bien plus large entre les articles 2 et 3 qu’il n’y a entre les articles 1 et 2. La seconde liste répare cela avec un `\strut` à la fin de la deuxième ligne du premier article.

```
\setlength{\fboxsep}{0pt}
\noindent\begin{minipage}[t]{0.2\linewidth}
\begin{enumerate}
\item \parbox[t]{15pt}{test \ \ test}
\item test
\item test
\end{enumerate}
\end{minipage}%
\begin{minipage}[t]{0.2\linewidth}
\begin{enumerate}
\item \parbox[t]{15pt}{test \ \ test\strut}
\item test
\item test
\end{enumerate}
\end{minipage}%
\begin{minipage}[t]{0.2\linewidth}
\begin{enumerate}
\item \fbox{\parbox[t]{15pt}{test \ \ test}}
\item \fbox{test}
\item \fbox{test}
```

```

\end{enumerate}
\end{minipage}%
\begin{minipage}[t]{0.2\linewidth}
\begin{enumerate}
 \item \fbox{\parbox[t]{15pt}{test \ \ test\strut}}
 \item \fbox{test}
 \item \fbox{test}
\end{enumerate}
\end{minipage}%

```

Les deux dernières listes utilisent `\fbox` pour montrer ce qui se passe. La `\parbox` du premier article de la troisième liste ne va que jusqu'au bas du second 'test', or ce mot ne comprend aucun caractère descendant sous la ligne de base. Le `\strut` ajouté dans la quatrième liste donne sous la ligne de base l'espace supplémentaire requis.

La commande `\strut` est souvent utile pour les graphiques, comme en TikZ ou Asymptote. Par exemple, vous pourriez avoir une commande telle que `\graphnode{nom-nœud}` qui ajuste un cercle autour de *nom-nœud*. Toutefois, à moins que vous preniez certaines précautions, prendre 'x' ou 'y' pour *nom-nœud* produit des cercles de diamètres différents parce que les caractères ont des tailles différentes. Une précaution possible contre cela serait que `\graphnode` insère `\strut` juste avant *nom-nœud*, et ne dessine le cercle qu'après cela.

L'approche générale d'utiliser une `\rule` de largeur nulle est utile dans de nombreuses circonstances. Dans le tableau ci-dessous, le filet de largeur nulle empêche le haut de la première intégrale de taper dans la `\hline`. De même, le second filet empêche la seconde intégrale de taper sur la première.

```

\begin{tabular}{rl}
 \textsc{Intégrale} & & \textsc{Valeur} & & \\
 \hline
 $\int_0^x t$, dt$ & & & & \\
 $\int_0^x t^2$, dt$ & & & & \\
\end{tabular}

```

(Bien que la commande double controbligue de terminaison de ligne dispose d'un argument optionnel pour changer la valeur de l'interligne correspondant, cela ne fonctionne pas pour résoudre ce problème. Changer la première double controbligue en quelque chose du genre de `\\[2.5ex]` augmente l'espace entre la ligne d'en-têtes de colonne et le filet `\hline`, et l'intégrale de la deuxième ligne taperait encore dans le filet).

## 19.14 `\vspace`

Synopsis, l'un parmi :

```

\vspace{longueur}
\vspace*{longueur}

```

Ajoute l'espace verticale *longueur*. Celle-ci peut-être positive, négative, ou nulle. C'est une longueur élastique — elle peut contenir une composante `plus` ou `minus` (voir Chapitre 14 [Lengths], page 141).

L'exemple ci-dessous met de l'espace entre les deux paragraphes :

Et je dormis.

```
\vspace{1ex plus 0.5ex}
L'aube fraiche d'un nouveau jour.
```

(Voir Section 19.11 [`\bigskip & \medskip & \smallskip`], page 199, pour les espaces inter-paragraphes usuelles).

La forme en `*` insère de l'espace vertical non suppressible. Plus précisément,  $\text{\LaTeX}$  supprime l'espace vertical au sauts de page et la forme en `*` a pour effet de conserver l'espace. Cet exemple ménage une espace entre les deux questions :

```
Question~: trouver l'intégrale de \(\ 5x^4+5 \).
```

```
\vspace*{2cm plus 0.5cm}
Question~: trouver la dérivée de \(\ x^5+5x+9 \).
```

Cette espace sera présente même si un saut de page survient entre les questions.

Si `\vspace` est utilisée au milieu d'un paragraphe (c.-à-d. en mode horizontal), l'espace est insérée *après* la ligne comprenant la commande `\vspace` sans commencer un nouveau paragraphe.

Dans cet exemple les deux questions sont espacées uniformément verticalement sur la page, avec au moins 2,5 cm d'espace sous chacune d'elle.

```
\begin{document}
1) Z'avez pas vu Mirza ? Oh la la la la la la
\vspace{2.5cm plus 1fill}

2) Où est donc passé ce chien ? Je le cherche partout
\vspace{2.5cm plus 1fill}
\end{document}
```

## 19.15 `\vfill`

Finis le paragraphe courant et insère une longueur élastique verticale (voir Chapitre 14 [`Lengths`], page 141) qui est infinie, de sorte qu'elle puisse se dilater ou se rétrécir autant que nécessaire.

On l'utilise souvent de la même façon que `\vspace{\fill}`, à ceci près que `\vfill` finit le paragraphe courant, tandis que `\vspace{\fill}` ajoute un espace vertical infini sous sa ligne sans tenir compte de la structure du paragraphe. Dans les deux cas cet espace disparaît à la fin de la page ; pour éviter cela voir Section 19.14 [`\vspace`], page 201.

Dans l'exemple qui suit la page est remplie, de sorte que les lignes en haut et en bas contiennent le texte 'Chien perdu !' et que le troisième 'Chien perdu !' est exactement à mi-chemin entre eux.

```
\begin{document}
Chien perdu !
\vfill
Chien perdu !
\vfill
Chien perdu !
\end{document}
```

## 19.16 `\addvspace`

Synopsis :

```
\addvspace{long-vert}
```

Ajoute un espace vertical de longueur *long-vert*. Toutefois, si il y a deux ou plus `\addvspace` à la suite, alors elles ajoutent seulement l'espace nécessaire à fabriquer la longueur naturelle égale au maximum des *long-vert* correspondant. Cette commande est fragile (voir Section 12.11 [`\protect`], page 134). La longueur *long-vert* est une longueur élastique (voir Chapitre 14 [`Lengths`], page 141).

L'exemple suivante illustre l'usage de la commande. L'environnement `picture` dessine une échelle sur laquelle deux filets sont posés. Dans un document  $\text{\LaTeX}$  de classe standard `article` la longueur `\baselineskip` vaut 12 pt. Comme l'indique l'échelle, les deux filets sont écartés de 22 pt : la somme des `\baselineskip` et de 10 pt provenant du premier `\addvspace`.

```
\documentclass{article}
\usepackage{color}
\begin{document}
\setlength{\unitlength}{2pt}%
\noindent\begin{picture}(0,0)%
 \multiput(0,0)(0,-1){25}{\color{blue}\line(1,0){1}}
 \multiput(0,0)(0,-5){6}{\color{red}\line(1,0){2}}
\end{picture}%
\rule{0.25\linewidth}{0.1pt}%
\par\addvspace{10pt}% \addvspace{20pt}%
\par\noindent\rule{0.25\linewidth}{0.1pt}%
\end{document}
```

Si dans l'exemple précédent on décommente le second `\addvspace`, l'écart entre les deux filets ne croit pas de 20 pt ; mais au lieu de cela l'écart devient la somme de `\baselineskip` et de 20 pt. De sorte que `\addvspace` en un sens fait le contraire de ce que son nom laisse entendre — elle vérifie que des espaces verticaux multiples ne s'accumulent pas, mais qu'au lieu de cela seul le plus grand d'eux est utilisé.

$\text{\LaTeX}$  utilise cette commande pour ajuster l'espace vertical au-dessus ou au-dessous d'un environnement qui commence un nouveau paragraphe. Par exemple, un environnement `theorem` commence et finit avec une `\addvspace` de sorte que deux `theorem` consécutifs sont séparés par un espace vertical, et non deux.

Une erreur `'Something's wrong--perhaps a missing \item'` pointant sur un `\addvspace` signifie que vous n'étiez pas en mode vertical quand vous avez invoqué cette commande ; une façon pour sortir de cela et de faire précéder `\addvspace` d'une commande `\par` (voir Section 15.1 [`\par`], page 149), comme dans l'exemple plus haut.

## 20 Boîtes

Dans le fond, L<sup>A</sup>T<sub>E</sub>X met des choses dans des boîtes et ensuite met ces boîtes sur une page. Aussi les commandes qui suivent sont centrales.

De nombreux paquetages sur le CTAN sont utiles pour la manipulation de boîtes. Un ajout utile aux commandes décrites ici est `adjustbox`.

### 20.1 `\mbox` & `\makebox`

Synopsis, l'un parmi les suivants :

```
\mbox{texte}
\makebox{texte}
\makebox[largeur]{texte}
\makebox[largeur][position]{texte}
```

La commande `\mbox` crée une boîte, un conteneur pour de la matière. Le *texte* est composé en mode LR (voir Chapitre 17 [Modes], page 185) ainsi il n'est pas coupé en plusieurs lignes. La commande `\mbox` est robuste, alors que `\makebox` est fragile (voir Section 12.11 [`\protect`], page 134).

Comme *texte* n'est pas coupé en plusieurs lignes, on peut utiliser `\mbox` pour empêcher une césure. Dans l'exemple suivant, L<sup>A</sup>T<sub>E</sub>X ne peut pas faire de césure dans le nom du char, 'T-34'.

```
Le char soviétique \mbox{T-34} est l'un des symboles de la victoire
sur le nazisme.
```

Dans les deux premières versions du synopsis, `\mbox` et `\makebox` sont sensiblement équivalentes. Elles créent une boîte juste assez large pour contenir le *texte*. (Elles sont comme le `\hbox` du T<sub>E</sub>X de base).

Dans la troisième version l'argument optionnel *largeur* spécifie la largeur de la boîte. Notez que l'espace occupé par le *texte* n'est pas nécessairement équivalent à la largeur de la boîte. D'une part, *texte* peut être trop petit ; l'exemple suivant crée une boîte d'une ligne entière de largeur :

```
\makebox[\linewidth]{Examen du chapitre}
```

où 'Examen du chapitre' est centré. Mais *texte* peut aussi être trop large pour *largeur*. Voir l'exemple plus bas pour des boîtes de largeur nulle.

Au sein de l'argument *largeur* vous pouvez utiliser les longueurs suivantes qui font référence aux dimensions de la boîte que L<sup>A</sup>T<sub>E</sub>X obtient en composant *texte* : `\depth`, `\height`, `\width`, `\totalheight` (la somme de la hauteur et de la profondeur de la boîte). Par exemple, pour fabriquer une boîte où le *texte* est dilaté au double de sa largeur naturelle vous pouvez écrire ceci :

```
\makebox[2\width]{Get a stretcher}
```

Pour la quatrième version du synopsis, la position du *texte* au sein de la boîte est déterminée par l'argument optionnel *position* qui peut prendre l'une des valeurs suivantes :

- c           Centré (par défaut).
- l           (Left) fer à gauche.

- r** (Right) fer à droite.
- s** (Stretch) dilate l'espace inter-mot dans *texte* sur la *largeur* entière ; *texte* doit contenir de l'espace dilatable pour que cela marche. Par exemple, ceci pourrait être le libellé d'un communiqué de presse :  
`\noindent\makebox[\textwidth][s]{\large\hfil DÉPÊCHE\hfil IMMÉDIATE\hfil}`

Une utilisation courante de `\makebox` est de fabriquer des boîtes de texte de largeur nulle. Dans l'exemple suivant on place dans un questionnaire la valeur des questions à leur gauche.

```
\newcommand{\pts}[1]{\makebox[0em][r]{#1 points\hspace*{1em}}}
\pts{10}Quelle est la vitesse aérodynamique d'une hirondelle à vide ?

\pts{90}Une hirondelle africaine ou européenne ?
```

Le bord droit de la sortie '10 points' (notez l'espace finale après 'points') est juste avant le 'Quelle'. Vous pouvez utiliser `\makebox` de la même manière quand vous faites des graphiques, comme en `TikZ` ou `Asymptote`, où vous mettez le bord du texte à une position connue, quelle que soit la longueur de ce texte.

Pour des boîtes avec cadre voir Section 20.2 [`\fbox` & `\framebox`], page 205. Pour des boîtes de couleurs voir Section 21.3.3 [Colored boxes], page 214.

Il y a une version apparentée de `\makebox` qui utilisée au sein de l'environnement `picture`, où la longueur est données en termes d'`\unitlength` (voir Section 8.19.13 [`\makebox (picture)`], page 88).

Comme *texte* est composé en mode LR, ni une double controblique `\`, ni `\par` ne produisent un saut de ligne ; par exemple `\makebox{abc def \ gh}` produit en sortie 'abc defghi' alors que `\makebox{abc def \par ghi}` produit 'abc def ghi', tout deux sur une seule ligne. Pour obtenir des lignes multiples voir Section 20.3 [`\parbox`], page 206, et Section 8.18 [`minipage`], page 79.

## 20.2 `\fbox` & `\framebox`

Synopsis, l'un parmi :

```
\fbox{texte}
\framebox{texte}
\framebox[largeur]{texte}
\framebox[largeur][position]{texte}
```

Crée une boîte dans un cadre englobant, c.-à-d. quatre filets entourant le *texte*. Les commandes `\fbox` et `\framebox` sont similaires à `\mbox` et `\makebox`, au cadre près (voir Section 20.1 [`\mbox` & `\makebox`], page 204). La commande `\fbox` est robuste, la commande `\framebox` est fragile (voir Section 12.11 [`\protect`], page 134).

Dans l'exemple suivant

```
\fbox{Avertissement ! Tout devoir non rendu obtiendra une note de zéro.}
```

$\LaTeX$  met le texte dans une boîte, le texte ne peut subir de césure. Autour de la boîte, séparés d'elle par une petit écart, quatre filets forment un cadre.

Les deux premières invocations de la commande, `\fbox{...}` et `\framebox{...}`, sont sensiblement équivalentes. Quant aux troisième et quatrième invocations, les arguments optionnels permettent de spécifier la largeur de la boîte comme *largeur* (une longueur), et la position du texte au sein de la boîte comme *position*. Voir Section 20.1 [`\mbox & \makebox`], page 204, pour la description complète, mais ci-après suit un exemple créant une boîte vide de 0,75 cm de large.

```
\setlength{\fboxsep}{0pt}\framebox[0.75cm]{\strut}
```

Le `\strut` assure que la hauteur totale soit de `\baselineskip` (voir Section 19.13 [`\strut`], page 200).

Les paramètres suivant déterminent la disposition du cadre.

`\fboxrule`

L'épaisseur des filets autour de la boîte englobée. Par défaut elle vaut 0.2 pt. On la change avec une commande telle que `\setlength{\fboxrule}{0.8pt}` (voir Section 14.2 [`\setlength`], page 143).

`\fboxsep` La distance du cadre à la boîte englobée. Vaut par défaut 3 pt. On la change avec une commande telle que `\setlength{\fboxsep}{0pt}` (voir Section 14.2 [`\setlength`], page 143). La régler à 0 pt peut parfois être utile : cela met un cadre autour d'une image sans bordure blanche.

```
{\setlength{\fboxsep}{0pt}%
 \framebox{%
 \includegraphics[width=0.5\textwidth]{prudence.jpg}}
```

Les accolades supplémentaires rendent local l'effet du `\setlength`.

Comme avec `\mbox` et `\makebox`, L<sup>A</sup>T<sub>E</sub>X ne fait pas de sauts de ligne au sein de *texte*. Mais l'exemple ci-dessous a pour effet que L<sup>A</sup>T<sub>E</sub>X effectue des sauts de ligne pour fabriquer un paragraphe, puis encadre le résultat.

```
\framebox{%
 \begin{minipage}{0.6\linewidth}
 « On va bien lentement dans ton pays ! Ici, vois-tu, on est obligé
 de courir tant qu'on peut pour rester au même endroit, dit la
 reine. Si tu veux te déplacer, tu dois courir au moins deux fois
 plus vite ! »
 \end{minipage}}
```

Voir Section 21.3.3 [Colored boxes], page 214, pour des couleurs autres que noir et blanc.

L'environnement `picture` a une version de la commande `\framebox` où les unités dépendent du `\unitlength` de `picture` (voir Section 8.19.14 [`\framebox (picture)`], page 88).

## 20.3 `\parbox`

Synopsis, l'un parmi les suivants :

```
\parbox{largeur}{contenu}
\parbox[position]{largeur}{contenu}
\parbox[position][hauteur]{largeur}{contenu}
\parbox[position][hauteur][pos-interne]{largeur}{contenu}
```

La commande `\parbox` produit une boîte de texte de largeur `largeur`. Cette commande peut être utilisée pour fabriquer une boîte à partir de petits morceaux de texte, et contenant un seul paragraphe. Cette commande est fragile (voir Section 12.11 [`\protect`], page 134).

```
\begin{picture}(0,0)
...
\put(1,2){\parbox{4.5cm}{\raggedright Le graphique étant une
droite sur une échelle logarithmique, la relation est
exponentielle.}}
\end{picture}
```

Le *contenu* est traité en mode texte (voir Chapitre 17 [Modes], page 185) ainsi  $\LaTeX$  fait des sauts de ligne de sorte à former un paragraphe. Mais il ne fabriquera pas plusieurs paragraphes ; pour cela, utilisez un environnement `minipage` (voir Section 8.18 [`minipage`], page 79).

Les options d'une `\parbox` (à l'exception du *contenu*) sont les mêmes que pour une `minipage`. Par commodité on rappelle ces options ici plus bas, mais se référer à Section 8.18 [`minipage`], page 79, pour une description complète.

Il y a deux arguments obligatoires. La *largeur* est une longueur rigide (voir Chapitre 14 [Lengths], page 141). Elle règle la largeur de la boîte dans laquelle  $\LaTeX$  compose le *contenu*. Le *contenu* est le texte qui est placé dans cette boîte. Il ne devrait pas avoir de composants qui fabriquent des paragraphes.

Il y a trois arguments optionnels, *position*, *hauteur*, et *pos-interne*. La *position* donne l'alignement vertical de la *parbox* relativement au texte alentour. La valeurs prises en charge sont `c` ou `m` pour que le centre selon la dimension verticale de la *parbox* soit aligné avec le centre de la ligne de texte adjacente (c'est le réglage par défaut), ou `t` pour accorder la ligne du haut de la *parbox* avec la ligne de base du texte alentour, ou `b` pour accorder la ligne du bas.

L'argument optionnel *hauteur* l'emporte sur la hauteur naturelle de la boîte.

L'argument optionnel *pos-interne* contrôle la position du *contenu* au sein de la *parbox*. Sa valeur par défaut est la valeur de *position*. Les valeurs possibles sont : `t` pour placer le *contenu* en haut de la boîte, `c` pour le centrer verticalement, `b` pour le place en bas de la boîte, et `s` pour le dilater verticalement (pour cela, le texte doit contenir de l'espace vertical élastique).

## 20.4 `\raisebox`

Synopsis, l'un parmi les suivants :

```
\raisebox{distance}{texte}
\raisebox{distance}[hauteur]{texte}
\raisebox{distance}[hauteur][profondeur]{texte}
```

La commande `\raisebox` élève ou abaisse du *texte*. Cette commande est fragile (voir Section 12.11 [`\protect`], page 134).

L'exemple suivant fabrique une commande pour noter la restriction d'une fonction en abaissant le symbole barre verticale.

```
\newcommand*\restreinte[1]{\raisebox{-.5ex}{\mathbb{A}}_{#1}}
$f\restreinte{A}$
```

Le premier argument obligatoire spécifie la longueur dont le *texte* doit être élevé. C'est une longueur rigide (voir Chapitre 14 [Lengths], page 141). Si elle est négative alors *texte* est abaissé. Le *texte* lui-même est traité en mode LR, aussi il ne doit pas contenir de saut de ligne (voir Chapitre 17 [Modes], page 185).

Les arguments optionnels *hauteur* et *profondeur* sont des dimensions. S'ils sont spécifiés, ils l'emportent sur les hauteur et profondeur naturelles de la boîte que L<sup>A</sup>T<sub>E</sub>X obtient en composant *texte*.

Au sein des arguments *distance*, *hauteur*, et *profondeur* vous pouvez utiliser les longueurs suivantes qui font référence à la dimension de la boîte que L<sup>A</sup>T<sub>E</sub>X obtient en composant *texte* : `\depth`, `\height`, `\width`, `\totalheight` (qui est la somme de la hauteur et de la profondeur de la boîte).

L'exemple suivant aligne deux graphiques sur leur bord supérieur (voir Chapitre 22 [Graphics], page 215).

```
\usepackage{graphicx,calc} % dans le préambule
...
\begin{center}
 \raisebox{1ex-\height}{%
 \includegraphics[width=0.4\linewidth]{lion.png}}
 \quad
 \raisebox{1ex-\height}{%
 \includegraphics[width=0.4\linewidth]{meta.png}}
\end{center}
```

Le premier `\height` est la hauteur de `lion.png` alors que le second est celle de `meta.png`.

## 20.5 `\sbox` & `\savebox`

Synopsis, l'un parmi les suivants :

```
\sbox{cmd-boîte}{texte}
\savebox{cmd-boîte}{texte}
\savebox{cmd-boîte}[largeur]{texte}
\savebox{cmd-boîte}[largeur][pos]{texte}
```

Cette commande compose *texte* dans une boîte de la même façon qu'avec `\makebox` (voir Section 20.1 [`\mbox` & `\makebox`], page 204), à ceci près qu'au lieu de sortir la boîte résultante, elle la sauvegarde dans un registre boîte référencé par la variable nommée *cmd-boîte*. Le nom de variable *cmd-boîte* commence avec une controblique, `\`. Vous devez préalablement avoir alloué le registre boîte *cmd-boîte* avec avec `\newsavebox` (voir Section 12.7 [`\newsavebox`], page 130). La commande `\sbox` est robuste, alors que `\savebox` est fragile (voir Section 12.11 [`\protect`], page 134).

L'exemple suivant crée et utilise un registre boîte.

```
\newsavebox{\nomcomplet}
\sbox{\nomcomplet}{Cadet Rousselle}
...
\usebox{\nomcomplet} a trois maisons qui n'ont ni poutres, ni chevrons !
C'est pour loger les hirondelles,
Que direz-vous d'\usebox{\nomcomplet} ?
```

Un avantage de l’usage répété d’un registre boîte par rapport à une variable macro `\newcommand` est l’efficacité, c.-à-d. que  $\text{\LaTeX}$  n’a pas besoin de répéter la composition du contenu. Voir l’exemple plus bas.

Les deux invocations de commande `\sbox{cmd-boîte}{texte}` et `\savebox{cmd-boîte}{texte}` sont sensiblement équivalentes. Quant aux troisième et quatrième, les arguments optionnels vous permettent de spécifier la largeur de la boîte comme *largeur*, et la position du texte au sein de cette boîte comme *position*. Voir Section 20.1 [`\mbox` & `\makebox`], page 204, pour une description complète.

Dans les commandes `\sbox` et `\savebox` le *texte* est composé en mode LR, aussi il n’a pas de sauts de lignes (voir Chapitre 17 [Modes], page 185). Si vous utilisez ce qui suit alors  $\text{\LaTeX}$  ne produit pas d’erreur mais ignore ce que vous voulez faire : si vous entrez `\sbox{\nouvreg}{test \ test}` et `\usebox{\nouvreg}` alors vous obtenez ‘testtest’, et si vous entrez `\sbox{\nouvreg}{test \par test}` et `\usebox{\nouvreg}` alors vous obtenez ‘test test’, mais ni erreur ni avertissement. La solution est d’utiliser une `\parbox` ou une `minipage` comme dans l’exemple qui suit :

```
\newsavebox{\unreg}
\savebox{\unreg}{%
 \begin{minipage}{\linewidth}
 \begin{enumerate}
 \item Premier article
 \item Second article
 \end{enumerate}
 \end{minipage}}
...
\usebox{\unreg}
```

En exemple de l’efficacité de réutiliser le contenu d’un registre boîte, ce qui suit place la même image sur chaque page du document en le plaçant dans l’en-tête.  $\text{\LaTeX}$  ne la compose qu’une fois.

```
\usepackage{graphicx} % tout cela dans le préambule
\newsavebox{\cachetreg}
\savebox{\cachetreg}{%
 \setlength{\unitlength}{2.5cm}%
 \begin{picture}(0,0)%
 \put(1.5,-2.5){%
 \begin{tabular}{c}
 \includegraphics[height=5cm]{companylogo.png} \\
 Bureau du Président
 \end{tabular}}
 \end{picture}%
}
\markright{\usebox{\cachetreg}}
\pagestyle{headings}
```

L’environnement `picture` est bien adapté pour régler finement le placement.

Si le registre `\noreg` n’a pas déjà été défini alors vous obtenez quelque chose du genre de ‘Undefined control sequence. <argument> \noreg’.

## 20.6 lrbox

Synopsis :

```
\begin{lrbox}{cmd-boîte}
 texte
\end{lrbox}
```

Ceci est la forme par environnement des commandes `\sbox` et `\savebox`, et leur est équivalente. Voir Section 20.5 [`\sbox & \savebox`], page 208, pour une description complète.

Le *texte* au sein de l'environnement est sauvegardé dans un registre boîte auquel la variable *cmd-boîte* fait référence. Le nom de variable *cmd-boîte* doit commencer par une controblique, `\`. Vous devez avoir alloué ce registre boîte préalablement avec `\newsavebox` (voir Section 12.7 [`\newsavebox`], page 130). Dans l'exemple suivant l'environnement est commode pour entrer le `tabular` :

```
\newsavebox{\jhreg}
\begin{lrbox}{\jhreg}
 \begin{tabular}{c}
 \includegraphics[height=1in]{jh.png} \\
 Jim Hef{}feron
 \end{tabular}
\end{lrbox}
...
\usebox{\jhreg}
```

## 20.7 \usebox

Synopsis :

```
\usebox{cmd-boîte}
```

La commande `\usebox` produit la boîte la plus récemment sauvegardée dans le registre boîte *cmd-boîte* par une commande `\sbox` ou `\savebox`, ou l'environnement `lrbox`. Voir Section 20.5 [`\sbox & \savebox`], page 208, pour plus ample information et des exemples. (Notez que le nom de la variable *cmd-boîte* commence par une controblique, `\`). Cette commande est robuste (voir Section 12.11 [`\protect`], page 134).

## 21 Couleur

Vous pouvez colorer du texte, des filets, etc. Vous pouvez également colorer l'intérieur d'une boîte ou une page entière et écrire du texte par dessus.

La prise en charge des couleurs se fait via un paquetage supplémentaire. Aussi, toutes les commandes qui suivent ne fonctionnent que si le préambule de votre document contient `\usepackage{color}`, qui fournit le paquetage standard.

Beaucoup d'autres paquetages complètent également les possibilités de L<sup>A</sup>T<sub>E</sub>X en matière de couleurs. Notamment `xcolor` est largement utilisé et étend significativement les possibilités décrites ici, y compris par l'ajout des modèles de couleur 'HTML' et 'Hsb'.

### 21.1 Options du paquetage color

Synopsis (à mettre dans le préambule du document) :

```
\usepackage[liste d'option séparées par des virgules]{color}
```

Quand vous chargez le paquetage `color` il y a deux types d'options disponibles.

Le premier spécifie le *pilote d'impression*. L<sup>A</sup>T<sub>E</sub>X ne contient pas d'information sur les différents systèmes de sortie, mais au lieu de cela dépend de l'information stockée dans un fichier. Normalement vous ne devriez pas spécifier les options de pilote dans le document, mais au lieu de cela vous vous reposez sur les valeurs par défaut de votre système. Un avantage de procéder ainsi est que cela rend votre document portable entre systèmes. Pour être complet nous incluons une liste des pilotes. Ceux présentement pertinents sont : `dvipdfmx`, `dvips`, `dvisvgm`, `luatex`, `pdftex`, `xetex`. Les deux pilotes `xdvi` et `oztex` sont pour l'essentiel des alias de `dvips` (et `xdvi` est monochrome). Ceux qui ne devraient pas être utilisés pour de nouveaux systèmes récent sont : `dvipdf`, `dvipdfm`, `dviwin`, `dvipsone`, `emtex`, `pctexps`, `pctexwin`, `pctexhp`, `pctex32`, `truetex`, `tcidvi`, `vtex` (et `dviwindo` est un alias de `dvipsone`).

Les options du second type, autres que les pilotes, sont listées ci-dessous.

#### monochrome

Désactive les commandes de couleur, de sorte qu'elles ne produisent pas d'erreur, mais ne produisent pas non plus de couleurs.

#### dvipsnames

Rend disponible une liste de 68 noms de couleur qui sont d'usage fréquent, en particulier dans des documents qui ne sont pas modernes. Ces noms de couleurs étaient à l'origine fournis par le pilote `dvips`, d'où le nom de l'option.

#### nodvipsnames

Ne charge pas cette liste de noms de couleur, faisant faire à L<sup>A</sup>T<sub>E</sub>X une toute petite économie d'espace mémoire.

### 21.2 Modèles de couleur

Un *modèle de couleur* est une façon de représenter les couleurs. Les possibilités de L<sup>A</sup>T<sub>E</sub>X dépendent du pilote d'impression. Toutefois, les pilotes `pdftex`, `xetex`, et `luatex` sont de nos jours de loin les plus largement utilisés. Les modèles ci-après fonctionnent pour ces

pilotes. À une exception près ils sont également pris en charge par tous les autres pilotes d'impression utilisés aujourd'hui.

Il est à noter qu'une combinaison de couleur peut être additive, ou soustractive. Les combinaisons additives mélangent les couleurs de la lumière, de sorte que par exemple combiner des intensités pleines de rouge, vert et bleu produit du blanc. Les combinaisons soustractives mélangent les pigments, tels que les encres, de sorte que combiner des intensités pleine de cyan, magenta et jaune produit du noir.

|              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>cmk</b>   | Un liste séparée de virgule avec quatre nombres réel compris entre 0 et 1 inclus. Le premier nombre est l'intensité du cyan, le deuxième celle du magenta, et les autres celle du jaune et du noir. Une valeur du nombre de 0 signifie l'intensité minimale, alors que 1 correspond à l'intensité pleine. Ce modèle est fréquemment utilisé pour l'impression de couleur. C'est un modèle soustractif.                                                                                    |
| <b>gray</b>  | Un unique nombre réel compris entre 0 et 1 inclus. Les couleurs sont des nuances de gris. Le nombre 0 produit du noir, alors que le 1 donne du blanc.                                                                                                                                                                                                                                                                                                                                     |
| <b>rgb</b>   | Une liste séparée de virgules avec trois nombres réels compris entre 0 et 1, inclusive. Le premier nombre est l'intensité de la composante rouge, le deuxième correspond au vert, et le troisième au bleu. Une valeur de 0 donnée au nombre signifie qu'en rien cette composante n'est ajoutée à la couleur, alors que 1 signifie que cela est fait à pleine intensité. C'est un modèle additif.                                                                                          |
| <b>RGB</b>   | (pilotes <code>pdftex</code> , <code>xetex</code> , <code>luatex</code> ) Une liste séparée par des virgules avec trois entiers compris entre 0 et 255 inclus. Ce modèle est une commodité pour utiliser <code>rgb</code> étant donnée qu'en dehors de L <sup>A</sup> T <sub>E</sub> X les couleurs sont souvent décrites dans un modèle rouge-vert-bleu utilisant des nombres dans cette plage. Les valeurs entrées sont converties vers le modèle <code>rgb</code> en divisant par 255. |
| <b>named</b> | On accède au couleur par des noms tels que <code>'PrussianBlue'</code> . La liste des noms dépend du pilote, mais tous prennent en charge les noms <code>'black'</code> , <code>'blue'</code> , <code>'cyan'</code> , <code>'green'</code> , <code>'magenta'</code> , <code>'red'</code> , <code>'white'</code> , et <code>'yellow'</code> (Voir l'option <code>dvipsnames</code> dans Section 21.1 [Color package options], page 211).                                                   |

## 21.3 Commandes pour color

Voici les commandes disponibles avec le paquetage `color`.

### 21.3.1 Définir des couleurs

Synopsis :

```
\definecolor{nom}{model}{spécification}
```

Donne le nom *nom* à la couleur. Par exemple, après `\definecolor{silver}{rgb}{0.75,0.75,0.74}` vous pouvez utiliser cette couleur avec Hé ho, `\textcolor{silver}{Silver}`~!

Cet exemple donne à la couleur un nom plus abstrait, de sorte qu'on puisse la changer sans que cela prête à confusion.

```
\definecolor{logocolor}{RGB}{145,92,131} % RGB nécessite pdflatex
\newcommand{\logo}{\textcolor{logocolor}{Les Brioches de Robert}}
```

Il est fréquent que les couleurs d'un document soient définies dans son préambule, ou dans sa classe ou style, plutôt que dans le corps du document.

### 21.3.2 Mettre du texte en couleur

Synopsis :

```
\textcolor{nom}{...}
\textcolor[modèle de couleur]{spécification de couleur}{...}
```

ou

```
\color{nom}
\color[modèle de couleur]{spécification}
```

Le texte affecté prend la couleur correspondante. La ligne suivante

```
\textcolor{magenta}{Je suis Ozymandias, roi des rois~:} Voyez mon œuvre, ô puissants,
```

met la première moitié en magenta alors que le reste est en noir. Vous pouvez utiliser une couleur déclarée avec `\definecolor` exactement de la même façon dont nous venons d'utiliser la couleur prédéfinie 'magenta'.

```
\definecolor{CriseDeLaQuarantaine}{rgb}{1.0,0.11,0.0}
```

```
Je considère l'idée de m'acheter une \textcolor{CriseDeLaQuarantaine}{voiture de sport}
```

Les deux formes de commandes `\textcolor` et `\color` diffèrent en ce que la première forme prend le texte à colorer en argument. Ceci est souvent plus commode, ou tout du moins plus explicite. La seconde forme est une déclaration, comme dans `La lune est composé de fromage {\color{vert}}`, de sorte qu'elle est en vigueur jusqu'à la fin du groupe ou de l'environnement courant. Ceci est parfois utile quand on écrit des macros, ou comme ci-dessous où on colore tout le contenu de l'environnement `center`, y compris les lignes verticales et horizontales.

```
\begin{center} \color{blue}
\begin{tabular}{l|r}
HG & HD \\ \hline
BG & BD
\end{tabular}
\end{center}
```

Vous pouvez utiliser des couleurs dans les équations. Un document peut avoir `\definecolor{couleurvedette}{RGB}{225,15,0}` dans le préambule, et ensuite contenir cette équation :

```
\begin{equation}
\int_a^b \textcolor{couleurvedette}{f'(x)} dx = f(b) - f(a)
\end{equation}
```

Typiquement les couleurs utilisées dans un document sont déclarées dans une classe ou dans un style mais il arrive qu'on désire fonctionner au coup par coup. C'est le cas d'usage des secondes forme du synopsis.

```
Des couleurs de \textcolor[rgb]{0.33,0.14,0.47}{violet} et {\color[rgb]{0.72,0.60,0.37}}
```

Le format de *spécification de couleur* dépend du modèle de couleur (voir Section 21.2 [Color models], page 211). Par exemple, alors que `rgb` prend trois nombres, `gray` n'en prend qu'un.

```
La sélection a été \textcolor[gray]{0.5}{grisée}.
```

Des couleurs au sein d'autres couleurs ne se combinent pas. Ainsi

```
\textcolor{green}{une sorte de \textcolor{blue}{bleu}}
```

se termine par un mot en bleu, et non pas dans une combinaison de bleu et de vert.

### 21.3.3 Faire des boîtes en couleur

Synopsis :

```
\colorbox{nom}{...}
\colorbox[nom modèle]{spécification couleur arrière-plan boîte}{...}
```

ou

```
\fcolorbox{couleur cadre}{couleur arrière-plan boîte}{...}
\fcolorbox[nom modèle]{spécification couleur cadre}{spécification arrière-
plan boîte}{...}
```

Fabrique une boîte avec la couleur d'arrière-plan indiquée. La commande `\fcolorbox` place un cadre autour de la boîte. Par exemple ceci :

```
Nom~::~\colorbox{cyan}{\makebox[5cm][1]{\strut}}
```

fabrique une boîte de couleur cyan qui fait cinq centimètres de long et dont la profondeur et la hauteur sont déterminées par le `\strut` (de sorte que la profondeur est  $-.3\text{\baselineskip}$  et la hauteur est  $\text{\baselineskip}$ ). Ceci met un texte blanc sur un arrière plan bleu :

```
\colorbox{blue}{\textcolor{white}{Welcome to the machine.}}
```

La commande `\fcolorbox` utilise les mêmes paramètres que `\fbox` (voir Section 20.2 [`\fbox` & `\framebox`], page 205), à savoir `\fboxrule` et `\fboxsep`, pour régler l'épaisseur du filet et de la séparation entre l'intérieur de la boîte et le filet l'entourant. Les valeurs par défaut de L<sup>A</sup>T<sub>E</sub>X sont 0.4pt et 3pt respectivement.

L'exemple suivant change l'épaisseur de la bordure en 0,8 points. Notez qu'il est entre accolades de sorte que le changement cesse d'être en vigueur à la fin de la seconde ligne.

```
{\setlength{\fboxrule}{0.8pt}
\colorbox{black}{red}{En aucun cas ne tournez cette poignée.}}
```

### 21.3.4 Faire des pages en couleur

Synopsis :

```
\pagecolor{nom}
\pagecolor[modèle de couleur]{spécification de couleur}
\nopagecolor
```

Les deux premières commandes règlent l'arrière-plan de la page, et de toutes les pages suivantes, à la couleur indiquée. Pour une explication de la spécification dans la seconde forme voir Section 21.3.2 [Colored text], page 213. La troisième fait revenir l'arrière-plan à la normale, c.-à-d. un arrière-plan transparent. (Si cela n'est pas pris en charge, alors utilisez `\pagecolor{white}`, bien que cela produit un arrière-plan blanc au lieu de l'arrière-plan transparent par défaut.)

```
...
\pagecolor{cyan}
...
\nopagecolor
```

## 22 Graphiques

Vous pouvez utiliser des graphiques tel que des fichiers PNG ou PDF dans vos documents  $\LaTeX$ . Vous aurez besoin d'un paquetage supplémentaire standard de  $\LaTeX$ . Cet exemple montre en bref comment :

```
\include{graphicx} % dans le préambule
...
\includegraphics[width=0.5\linewidth]{graphique.pdf}
```

Pour utiliser les commandes décrites ici, le préambule de votre document doit contenir soit `\usepackage{graphicx}`, soit `\usepackage{graphics}`. (Le plus souvent, c'est `graphicx` qui est préférable.)

Les graphiques peuvent être de deux types, matriciels ou vectoriels.  $\LaTeX$  peut utiliser les deux. Dans les graphiques matriciels le fichier comprend pour chaque position dans un tableau une entrée décrivant la couleur de celle-ci. Un exemple de ceci est une photographie en format JPG. Dans un graphique vectoriel, le fichier contient une liste d'instructions telles que 'dessine un cercle avec tel rayon et tel centre'. Un exemple de ceci est le dessin d'une ligne produit par le programme Asymptote en format PDF. En général les graphiques vectoriels sont plus utiles car on peut les redimensionner sans pixélisation ou autres problèmes, et parce que la plupart du temps ils ont une taille plus petite.

Il y a des systèmes particulièrement bien adaptés pour produire des graphiques pour un document  $\LaTeX$ . Par exemple, qui vous permettent d'utiliser les mêmes polices que dans votre document.  $\LaTeX$  comprend un environnement `picture` (voir Section 8.19 [picture], page 81) qui a des possibilités de base. En plus de cela, il y a d'autres façons d'inclure des commandes de production de graphiques dans le document. Deux systèmes pour cela sont les paquetages `PSTricks` et `TikZ`. Il y a aussi des systèmes externes à  $\LaTeX$  qui génèrent un graphique qu'on peut inclure en utilisant les commandes de ce chapitre. Il y a aussi deux autres systèmes qui utilisent un langage de programmation : `Asymptote` et `MetaPost`. Encore un autre système qui utilise une interface graphique : `Xfig`. Une description complète des ces systèmes est hors du champ d'application de ce document ; reportez vous à leur documentation sur le CTAN.

### 22.1 Options du paquetage `graphics`

Synopsis (à placer dans le préambule du document) :

```
\usepackage[liste d'option séparée par des virgules]{graphics}
```

ou

```
\usepackage[liste d'option séparée par des virgules]{graphicx}
```

Le paquetage `graphicx` a un format pour les arguments optionnels passés à `\includegraphics` qui est commode (c'est le format clef/valeur), ainsi c'est le meilleur des deux choix pour les nouveaux documents. Quand vous chargez les paquetages `graphics` ou `graphicx` avec `\usepackage` il y a deux sortes d'options disponibles.

La première est que  $\LaTeX$  ne contient pas d'information concernant les différents systèmes de sortie, mais au lieu de cela dépend de l'information stockée dans un fichier *pilote d'impression*. Normalement vous ne devriez pas spécifier l'option de pilote dans le

document, mais au lieu de cela vous reposer sur les valeurs par défaut du système. Un avantage de procéder ainsi et que cela rend les documents portables entre systèmes.

Pour être complet voici une liste des pilotes. Ceux qui sont actuellement pertinents sont : `dvipdfmx`, `dvips`, `dvisvgm`, `luatex`, `pdftex`, `xetex`. Les deux pilotes `xdvi` and `oztex` sont essentiellement des alias de `dvips` (et `xdvi` est monochrome). Ceux qu'on ne devrait pas utiliser pour de nouveaux systèmes sont : `dvipdfmx`, `dvips`, `dvisvgm`, `luatex`, `pdftex`, `xetex`. Les deux pilotes `xdvi` et `oztex` sont essentiellement des alias de `dvips` (et `xdvi` est monochrome). Ceux qui ne devraient pas être utilisés pour de nouveaux systèmes sont : `dvipdf`, `dvipdfm`, `dviwin`, `dvipsone`, `emtex`, `pctexps`, `pctexwin`, `pctexhp`, `pctex32`, `truotex`, `tcidvi`, `vtex` (et `dviwindo` est un alias de `dvipsone`). Ces pilotes sont stockés dans des fichiers avec une extension `.def`, tels que `pdftex.def`.

La seconde sorte d'options est décrite ci-après.

- demo**      Au lieu d'un fichier d'image,  $\LaTeX$  met un rectangle de 150 pt sur 100 pt (à moins qu'une autre taille soit spécifiée dans la commande `\includegraphics`).
- draft**      Aucun des fichiers graphiques n'est affiché, mais au lieu de cela le nom du fichier est imprimé au sein d'une boîte de la bonne taille. De sorte à ce que la taille soit déterminée, le fichier doit être présent.
- final**      (Option par défaut) L'emporte sur toute options **draft** précédente, de sorte que le document affiche le contenu des fichiers graphiques.
- hiderotate**      Ne montre pas le texte ayant subi une rotation. (Ceci permet d'utiliser une visionneuse ne prenant pas en charge le texte ayant subi une rotation).
- hidescale**      Ne montre pas le texte rétréci/dilaté. (Ceci permet d'utiliser une visionneuse ne prenant pas en charge le texte ayant subi un rétrécissement/une dilatation.)
- hiresbb**      Dans un fichier PS ou EPS la taille du graphique peut être spécifiée de deux façons. Les lignes `%%BoundingBox` décrivent la taille du graphique en utilisant des multiples entiers d'un point PostScript, c.-à-d. des multiples entiers d' 1/72 de pouce. Une addition plus récente au langage PostScript autorise les multiples décimaux, tel que 1.23, dans les lignes `%%HiResBoundingBox`. Cette option fait que  $\LaTeX$  lit la taille à partir de `%%HiResBoundingBox` plutôt que de `%%BoundingBox`.

## 22.2 Configuration du paquetage graphics

Ces commandes configurent la façon dont  $\LaTeX$  recherche le graphique dans le système le fichier.

Le comportement du code de recherche dans le système de fichier dépend nécessairement de la plateforme. Dans ce document nous couvrons GNU/Linux, Macintosh, et Windows, étant donné que ces systèmes ont des configurations typiques. Pour les autres situations consultez la documentation dans `grfguide.pdf`, ou le source  $\LaTeX$ , ou la documentation de votre distribution  $\TeX$ .

### 22.2.1 `\graphicspath`

Synopsis :

```
\graphicspath{liste des répertoires entre les accolades}
```

Déclare une liste de répertoires dans lesquels rechercher des fichiers graphiques. Ceci permet d'écrire plus tard quelque chose du genre de `\includegraphics{lion.png}` au lieu de donner son chemin.

$\LaTeX$  recherche toujours en premier les fichiers graphiques dans le répertoire courant (et dans le répertoire de sortie, si spécifié ; voir [output directory], page 268). La déclaration ci-dessous dit au système de regarder ensuite dans le sous-répertoire `img`, et ensuite `./img`.

```
\usepackage{graphicx} % ou graphics; à mettre dans le préambule
...
\graphicspath{ {img/} {./img/} }
```

La déclaration `\graphicspath` est optionnelle. Si vous ne l'incluez pas, alors le comportement par défaut de  $\LaTeX$  est de rechercher dans tous les endroits où il recherche d'habitude un fichier (il utilise la commande `\input@path` de  $\LaTeX$ ). En particulier, dans ce cas l'un l'un des endroits où il regarde est le répertoire courant.

Mettez chaque nom de répertoire entre accolades ; en particulier, ci-dessus on a écrit `{img}`. Faites ainsi même si il n'y a qu'un seul répertoire. Chaque nom de répertoire doit se terminer par une oblique vers l'avant `/`. Ceci est vrai même sur Windows, où la bonne pratique est d'utiliser des obliques vers l'avant pour tous les séparateurs de répertoire puisque cela rend le document portable vers d'autres plateformes. Si vous avez des espaces dans votre nom de répertoire alors utilisez des guillemets anglais, comme dans `"mes docs/"`. Enfreindre l'une de ces règles aura pour effet que  $\LaTeX$  rapportera une erreur `Error: File 'filename' not found`.

Pour faire simple, l'algorithme est que dans l'exemple qui suit, après avoir regardé dans le répertoire courant,

```
\graphicspath{ {img/} {./img/} }
...
\usepackage{lion.png}
```

pour chacun des répertoires listés,  $\LaTeX$  le concatène avec le nom du fichier et recherche le fichier ainsi nommé, eh cherchant `img/lion.png` et puis `./img/lion.png`. Cet algorithme signifie que la commande `\graphicspath` ne recherche pas récursivement dans les sous-répertoires : si vous faites `\graphicspath{{a/}}` et que le graphique est dans `a/b/lion.png` alors  $\LaTeX$  ne le trouvera pas. Cela signifie aussi que vous pouvez utiliser des chemins absolus tels que `\graphicspath{{/home/jim/logos/}}` ou `\graphicspath{{C:/Users/Albert/Pictures/}}`. Toutefois, faire cela rend votre document non portable. (Vous pourriez préserver la portabilité en ajustant vos le paramètre `TEXINPUTS` du fichier de configuration des réglages système  $\TeX$  ; voir la documentation de votre système).

Vous pouvez utiliser `\graphicspath` n'importe où dans le document. Vous pouvez l'utiliser plus d'une fois. Sa valeur peut être affichée avec `\makeatletter\typeout{\Ginput@path}\makeatother`.

Les répertoires sont à donner relativement au fichier de base. Pour faire clair, supposez que vous travaillez sur un document basé sur `livre/livre.tex` et

qu'il contienne `\include{chapitres/chap1}`. Si dans `chap1.tex` vous mettez `\graphicspath{{graphiques/}}` alors  $\LaTeX$  ne cherchera pas les graphiques dans `livre/chapitres/graphiques`, mais dans `livre/graphiques`.

### 22.2.2 `\DeclareGraphicsExtensions`

Synopsis :

```
\DeclareGraphicsExtensions{liste séparée par des virgules d'extensions de fichier}
```

Déclare les extensions de nom de fichier à essayer. Ceci vous permet de spécifier l'ordre dans lequel les formats graphiques sont choisis quant les fichiers graphiques sont inclus en donnant le nom de fichier sans l'extension, comme dans `\includegraphics{courbe_de_fonction}`.

Dans l'exemple qui suit,  $\LaTeX$  trouve les fichiers au format PNG avant les fichiers PDF.

```
\DeclareGraphicsExtensions{.png,PNG,.pdf,.PDF}
...
\includegraphics{lion} % trouve lion.png avant lion.pdf
```

Du fait que le nom `lion` ne contient pas de point,  $\LaTeX$  utilise la liste d'extension. Pour chacun des répertoires dans la liste de chemins graphiques (voir Section 22.2.1 [`\graphicspath`], page 217),  $\LaTeX$  essaie les extensions dans l'ordre donné. S'il ne trouve aucun fichier correspondant après avoir essayé tous les répertoires et toutes les extensions, alors il renvoie une erreur '`! LaTeX Error: File 'lion' not found`'. Notez que vous devez inclure le point au début de chaque extension.

Du fait que les noms de fichiers sous GNU/Linux et Macintosh sont sensibles à la casse, la liste des extensions de fichier est sensible à la casse sur ces plateformes. La plateforme Windows n'est pas sensible à la casse.

Vous n'êtes pas tenu d'inclure `\DeclareGraphicsExtensions` dans votre document ; le pilote d'impression a une valeur par défaut raisonnable. Par exemple, la version la plus récente de `pdftex.def` a cette liste d'extensions.

```
.pdf,.png,.jpg,.mps,.jpeg,.jbig2,.jb2,.PDF,.PNG,.JPG,.JPEG,.JBIG2,.JB2
```

Pour modifier cet ordre, utilisez le paquetage `grfext`.

Vous pouvez utiliser cette commande n'importe où dans le document. Vous pouvez l'utiliser plus d'une fois. On peut afficher la valeur courante avec `\makeatletter\typeout{\Gin@extensions}\makeatother`.

### 22.2.3 `\DeclareGraphicsRule`

Synopsis :

```
\DeclareGraphicsRule{extension}{type}{extension taille-fic}{commande}
```

Déclare comment gérer les fichiers graphiques dont le nom se termine en *extension*.

L'exemple suivant déclare que tous les fichiers dont le nom est de la forme `nomfichier-sans-point.mps` doivent être traités comme une sortie de MetaPost, ce qui signifie que le pilote d'impression utilisera son code de gestion MetaPost pour traiter le fichier en entrée.

```
\DeclareGraphicsRule{.mps}{mps}{.mps}{}
```

La déclaration suivante

```
\DeclareGraphicsRule{*}{mps}{*}{}
```

dit à L<sup>A</sup>T<sub>E</sub>X qu'il doit traiter comme une sortie MetaPost tout fichier avec extension non couverte par une autre règle, ainsi il couvre `nomfichier.1`, `nomfichier.2`, etc.

Ici on décrit les quatre arguments.

*extension* L'extension de nom de fichier à laquelle la règle considérée s'applique. L'extension commence au premier point dans le nom de fichier, en incluant le point. Utiliser l'astérisque, `*`, pour désigner la comportement par défaut pour toutes les extensions non déclarées.

*type* Le type du fichier considéré. Ce type est une chaîne de caractères qui doit être définie dans le pilote d'impression. Par exemple, les fichiers avec les extensions `.ps`, `.eps`, ou `.ps.gz` peuvent tous être classifiés sous le type `eps`. Tous les fichiers de même type sont traités en entrée avec la même commande interne par le pilote d'impression. Par exemple, les types de fichiers reconnus par `pdftex` sont : `jpg`, `jbig2`, `mps`, `pdf`, `png`, `tif`.

*extension taille-fic*

L'extension du fichier à lire pour déterminer la taille du graphique, si un tel fichier existe. Elle peut être la même que *extension* mais aussi être différente.

Considérons par exemple un graphique PostScript. Pour le rendre plus petit, il peut être compressé en un fichier `.ps.gz`. Les fichiers compressés ne sont pas lus simplement par L<sup>A</sup>T<sub>E</sub>X aussi vous pouvez mettre l'information de boîte englobante dans un fichier séparé. Si *extension taille-fic* est vide, alors vous devez spécifier l'information de taille dans les arguments de `\includegraphics`.

Si le fichier pilote a une procédure pour lire les tailles de fichier pour *type* alors elle est utilisée, sinon il utilise la procédure pour lire les fichiers `.eps`. (Ainsi vous pourriez spécifier la taille d'un fichier `.bmp` dans un fichier avec une ligne `%%BoundingBox` de style PostScript si aucun autre format n'est disponible).

*commande*

Une commande à appliquer au fichier. La plupart du temps on laisse cet argument vide. Cette commande doit commencer avec un caractère accent grave. Ainsi, `\DeclareGraphicsRule{.eps.gz}{eps}{.eps.bb}{'gunzip -c #1}` spécifie que tout fichier ayant une extension `.eps.gz` doit être traité comme un fichier `eps`, avec l'information de boîte englobante stockée dans le fichier ayant l'extension `.eps.bb`, et que la commande `gunzip -c` sera exécuté sur votre plateforme pour décompresser le fichier.

Une telle commande est spécifique à votre plateforme. De plus, votre système T<sub>E</sub>X doit autoriser l'exécution de commandes externes ; par mesure de sécurité les systèmes modernes restreignent l'exécution de commandes à moins que vous l'autorisiez explicitement. Se référer à la documentation de votre distribution T<sub>E</sub>X.

## 22.3 Commandes du paquetage graphics

Voici les commandes disponibles dans les paquetages `graphics` et `graphicx`.

### 22.3.1 `\includegraphics`

Synopsis pour le paquetage `graphics` :

```
\includegraphics{nomfichier}
\includegraphics[urx,ury]{nomfichier}
\includegraphics[llx,lly][urx,ury]{nomfichier}
\includegraphics*{nomfichier}
\includegraphics*[urx,ury]{nomfichier}
\includegraphics*[llx,lly][urx,ury]{nomfichier}
```

Synopsis pour le paquetage `graphicx` :

```
\includegraphics{nomfichier}
\includegraphics[liste-à-clefs-valeurs]{nomfichier}
\includegraphics*{nomfichier}
\includegraphics*[liste-à-clefs-valeurs]{nomfichier}
```

Inclut un fichier graphique. La forme étoilée `\includegraphics*` rogne le graphique à la taille spécifiée, alors que pour la forme non-étoilée toute partie du graphique en dehors de la boîte de la taille spécifiée se superpose à ce qui l'entoure.

Dans cet exemple

```
\usepackage{graphicx} % dans le préambule
...
\begin{center}
\includegraphics{trace.pdf}
\end{center}
```

on incorpore dans le document le graphique dans `plot.pdf`, centré et à sa taille nominale. Vous pouvez aussi donner un chemin au fichier, comme dans `\includegraphics{graphiques/trace.pdf}`. Pour spécifier une liste de répertoire ou rechercher le fichier, voir Section 22.2.1 [`\graphicspath`], page 217.

Si votre nom de fichier comprend des espaces, alors placer le entre guillemets anglais. Par exemple comme dans `\includegraphics{"image complémentaire.jpg"}`.

La commande `\includegraphics{nomfichier}` décide du type de graphique en segmentant *nomfichier* sur le premier point ('.'). Vous pouvez utiliser *nomfichier* sans extension de nom de fichier, comme dans `\includegraphics{turing}` et  $\LaTeX$  essaie une séquence d'extension telle que `.png` et `.pdf` jusqu'à ce qu'il trouve un fichier avec cette extension (voir Section 22.2.2 [`\DeclareGraphicsExtensions`], page 218).

Si votre fichier comprend des points avant l'extension alors vous pouvez les cacher avec des accolades, comme dans `\includegraphics{{plot.2018.03.12.a}.pdf}`. Ou, si vous utilisez le paquetage `graphicx` alors vous pouvez utiliser les options `type` et `ext` ; voir plus bas. Ces problèmes de noms de fichiers, et d'autres sont aussi traités par le paquetage `grffile`.

L'exemple ci-après place un graphique dans un environnement `figure` de sorte que  $\LaTeX$  puisse le déplacer sur la page suivante si le faire rentrer dans la page courante rend mal (voir Section 8.10 [`figure`], page 67).

```
\begin{figure}
\centering
\includegraphics[width=3cm]{rayonxpoumon.jpg}
```

```
\caption{Ça saute aux yeux : vous ne devriez arrêter de fumer} \label{fig:rayon-x}
\end{figure}
```

L'exemple suivant place un graphique non flottant, de sorte à garantir qu'il apparaisse à ce point-ci du document même si cela oblige L<sup>A</sup>T<sub>E</sub>X à dilater le texte ou à recourir à des zones blanches sur la page. Le graphique est centré et a une légende.

```
\usepackage{caption} % dans le préambule
...
\begin{center}
\includegraphics{sans/images.png}
\captionof{figure}{L'esprit de la nuit} \label{sans:images} % optionnel
\end{center}
```

L'exemple suivant place une boîte avec un graphique le long d'une autre boîte contenant du texte, les deux boîtes étant verticalement centrées :

```
\newcommand*\vcenteredhbox[1]{\begin{tabular}{@{}c@{}}#1\end{tabular}}
...
\begin{center}
\vcenteredhbox{\includegraphics[width=0.4\textwidth]{trace}}
\hspace{1em}
\vcenteredhbox{\begin{minipage}{0.4\textwidth}
\begin{displaymath}
f(x)=x\cdot \sin (1/x)
\end{displaymath}
\end{minipage}}
\end{center}
```

Si vous utilisez le paquetage `graphics` alors les seules options disponibles concernent la taille du graphique (mais voir Section 22.3.2 [`\rotatebox`], page 225, et Section 22.3.3 [`\scalebox`], page 226). Quand un argument optionnel est présent alors c'est [`urx,ury`] et il donne les coordonnées du coin en haut à droite de l'image, comme une paire de dimensions T<sub>E</sub>X (voir Section 14.1 [Units of length], page 142). Si les unités sont omises, alors par défaut l'unité considérée est le bp. Dans ce cas, le coin en bas à gauche de l'image est supposé être à (0,0). Si deux arguments optionnels sont présents alors le premier est [`llx,lly`], et il spécifie les coordonnées du coin en bas à gauche de l'image. Ainsi, `\includegraphics[1in,0.618in]{...}` demande que le graphique soit placé de sorte à avoir 1 pouce de largeur et 0,618 pouces de hauteur et donc son origine est à (0,0).

Le paquetage `graphicx` vous offre beaucoup plus d'options. Spécifiez les sous la forme de paires clef-valeur, comme ci-après :

```
\begin{center}
\includegraphics[width=3cm,angle=90]{lion}
\hspace{2em}
\includegraphics[angle=90,width=3cm]{lion}
\end{center}
```

Les options sont lues de gauche à droite. Ainsi le premier graphique ci-dessus est dilaté à une largeur de 3 cm, et ensuite subit une rotation de 90 degrés, alors que le second subit la rotation en premier et ensuite est dilaté à 3 cm de largeur. Ainsi, à moins que le graphique soit parfaitement carré, à la fin les deux n'ont pas les mêmes largeurs et hauteurs.

Il y a beaucoup d'options. Les principales sont listées en premier.

Notez qu'un graphique est placé par L<sup>A</sup>T<sub>E</sub>X dans une boîte, qu'on désigne traditionnellement comme sa *boîte englobante* (distincte de la BoundingBox PostScript décrite plus bas). La zone sur laquelle le graphique s'imprime peut aller au delà de cette boîte, ou s'inscrire dans cette boîte, mais quand L<sup>A</sup>T<sub>E</sub>X fabrique une page il assemble des boîtes, et celle-ci est la boîte allouée pour le graphique.

**width** Le graphique est affiché de sorte que sa boîte englobante ait cette largeur. Un exemple est `\includegraphics[width=2.5cm]{trace}`. Vous pouvez utiliser les unités de longueur standards de T<sub>E</sub>X (voir Section 14.1 [Units of length], page 142) et également il est commode d'utiliser `\linewidth`, ou, dans un document recto-verso, `\columnwidth` (voir Section 5.5 [Page layout parameters], page 32). Un exemple est qu'en utilisant le paquetage `calc` vous pouvez rendre le graphique 1 cm plus étroit que la largeur du texte avec `\includegraphics[width=\linewidth-1.0cm]{hefferon.jpg}`.

**height**

Le graphique est affiché de sorte que sa boîte englobante est de cette hauteur. Vous pouvez utiliser les unités de longueur standards de T<sub>E</sub>X (voir Section 14.1 [Units of length], page 142), et également il est commode d'utiliser `\pageheight` et `\textheight` (voir Section 5.5 [Page layout parameters], page 32). Par exemple, la commande `\includegraphics[height=0.25\textheight]{godel}` rend le graphique d'une hauteur d'un quart de celle de la zone de texte.

**totalheight**

Le graphique est affiché de sorte que sa boîte englobante ait cette hauteur plus profondeur. Ceci diffère de la hauteur si le graphique a subi une rotation. Par exemple, s'il a subi une rotation de -90 degrés alors il a une hauteur nulle mais une grande profondeur.

**keepaspectratio**

Si réglé à `true`, ou juste spécifié comme ci-dessous

```
\includegraphics[... ,keepaspectratio,...]{...}
```

et que vous donnez en option à la fois `width` et `height` (ou `totalheight`), alors L<sup>A</sup>T<sub>E</sub>X rendra le graphique aussi grand que possible sans le déformer. C'est à dire que L<sup>A</sup>T<sub>E</sub>X assure que le graphique n'est ni plus large que `width` ni plus haut que `height` (ou `totalheight`).

**scale** Facteur par lequel dilater/contracter le graphique. On peut rendre graphique à deux fois sa taille nominale avec `\includegraphics[scale=2.0]{...}`. Ce nombre peut prendre n'importe quelle valeur ; un nombre entre 0 et 1 contracte le graphique et un nombre négatif lui applique une réflexion.

**angle** Applique une rotation au graphique. L'angle est exprimé en degrés et dans le sens antihoraire. La rotation est effectuée autour de l'origine donnée par l'option `origin` ; s'y référer. Pour une description complète de la composition de matière ayant subi une rotation, voir Section 22.3.2 [`\rotatebox`], page 225.

**origin** Le point du graphique autour duquel la rotation s'effectue. Les valeurs possibles sont toute chaîne contenant un ou deux caractères parmi : **l** pour gauche (left), **r** pour droite (right), **b** pour bas, **c** pour centré, **t** pour haut (top), et **B** pour ligne de base. Ainsi, saisir la commande `\includegraphics[angle=180,origin=c]{moon}` a pour effet de mettre la figure tête en bas en tournant autour du centre de l'image, alors que la commande `\includegraphics[angle=180,origin=lB]{LeBateau}` fait la rotation autour du point gauche de la ligne de base. (Le caractère **c** désigne le centre horizontal dans **bc** ou **tc**, mais le centre vertical dans **lc rc**). Le réglage par défaut est **lB**.

Pour faire une rotation autour d'un point arbitraire, voir Section 22.3.2 [`\rotatebox`], page 225.

Voici maintenant des options moins usuelles.

**viewport** Désigne une sous-région du graphique à afficher. Prend quatre arguments, séparés par des espaces et exprimés comme des dimensions  $\TeX$ , comme dans `\includegraphics[... , viewport=0in 0in 1in 0.618in]{...}`. Quand l'unité est omise, les dimensions sont par défaut en big points, **bp**. Elle sont prises relativement à l'origine spécifiée par la boîte englobante. Voir aussi l'option **trim**.

**trim** Désigne les parties du graphique à ne pas afficher. Prend quatre arguments, séparés par des espaces, et exprimés comme des dimensions  $\TeX$  dimensions, comme dans `\includegraphics[... , trim= 0in 0.1in 0.2in 0.3in, ...]{...}`. Ceci donne la quantité du graphique à ne pas afficher, c.-à-d. que  $\LaTeX$  rogne l'image de 0 pouce sur la gauche left, 0.1 pouce sur la bas, 0.2 pouce sur la droite, et 0.3 pouce sur le haut. Voir aussi l'option **viewport**.

**clip** Si réglé à **true**, ou juste spécifié comme ci-dessous

```
\includegraphics[... ,clip,...]{...}
```

alors le graphique est rogné à sa boîte englobante. Ceci revient au même que d'utiliser la forme étoilée de la commande, `\includegraphics* [... ]{...}`.

**page** Donne le numéro de page pour un fichier PDF multi-page. La valeur par défaut est **page=1**.

**pagebox** Spécifie quelle boîte englobante utiliser pour les fichiers PDF parmi les suivantes **mediabox**, **cropbox**, **bleedbox**, **trimbox**, ou **artbox**. Les fichiers PDF n'ont pas la `BoundingBox` que les fichiers PostScript ont, mais peuvent spécifier jusqu'à quatre rectangles prédéfinis. La boîte **mediabox** donne les frontières du médium physique. La boîte **cropbox** est la région à laquelle le contenu de la page devrait être épinglé quand il est affiché. La boîte **bleedbox** la région à laquelle le contenu de la page devrait être épinglé en production. La boîte **trimbox** est correspond aux dimensions désirées de la page finie. La boîte **artbox** est de l'étendue du contenu de la page ayant du sens. Le pilote règle la taille de l'image en se basant sur la **cropbox** si elle est présente, sinon il n'utilise pas les autres, avec un ordre de préférence propre au pilote. La boîte **mediabox** est toujours présente.

**interpolate**

Active ou désactive l'interpolation des images matricielles par la visionneuse. On peut régler l'option avec `interpolate=true` ou juste comme ci-dessous :

```
\includegraphics[... ,interpolate,...]{...}
```

**quiet**

N'écrit pas d'information dans le journal. On peut régler l'option avec `quiet=true` ou juste en la spécifiant avec `\includegraphics[... ,quiet,...]{...}` ;

**draft**

Lorsqu'on règle l'option avec `draft=true` ou juste ainsi

```
\includegraphics[... ,draft,...]{...}
```

alors le graphique n'apparaît pas dans le document, ce qui permet éventuellement d'économiser l'imprimante couleur. À la place  $\LaTeX$  place une boîte vide de la bonne taille avec le nom du fichier imprimé dedans.

Les options suivantes traitent de la boîte englobante pour les fichiers graphique de type PostScript Encapsulé, leur taille est spécifié avec une ligne `%%BoundingBox` qui apparaît dans le fichier. Elle a quatre valeurs donnant la coordonnée  $x$  inférieure, la coordonnée  $y$  inférieure, la coordonnée  $x$  supérieure, et la coordonnée  $y$  supérieure. L'unité est le point PostScript, équivalent au big point de  $\TeX$ , à savoir 1/72 de pouce. Par exemple, si un fichier `.eps` a la ligne `%%BoundingBox 10 20 40 80` alors sa taille naturelle est 30/72 de pouce de large pour 60/72 de pouce de haut.

**bb**

Spécifie la boîte englobante de la région affichée. L'argument est constitué de quatre dimensions séparées par des espaces, comme dans `\includegraphics[... ,bb= 0in 0in 1in 0.618in]{...}`. D'ordinaire `\includegraphics` lit les ces nombre `BoundingBox` dans le fichier EPS automatiquement, de sorte que cette option n'est utile que si la définition de la boîte englobante est absente du fichier ou si vous désirez la changer.

**bbllx, bblly, bburx, bbury**

Règle la boîte englobante. Ces quatre options là sont obsolètes, mais existent encore pour maintenir la rétrocompatibilité avec des paquetages anciens.

**natwidth, natheight**

Une alternative à `bb`. Régler

```
\includegraphics[... ,natwidth=1in,natheight=0.618in,...]{...}
```

revient au même que régler `bb=0 0 1in 0.618in`.

**hiresbb**

Si réglé à `true`, ou juste spécifié comme dans

```
\includegraphics[... ,hiresbb,...]{...}
```

alors  $\LaTeX$  recherche la ligne `%%HiResBoundingBox` plutôt que la ligne `%%BoundingBox`. (La ligne `BoundingBox` n'utilise que des nombres entiers naturels alors que la ligne `HiResBoundingBox` utilise des décimaux ; les deux utilisent une unité équivalente au big point de  $\TeX$ , 1/72 de pouce). Pour l'emporter sur un réglage précédent à `true`, on peut la régler à `false`.

Les options suivantes permettent à l'utilisateur de l'emporter sur la méthode par laquelle  $\LaTeX$  choisit le type d'un graphique en fonction de son extension de nom de fichier. Par exemple avec `\includegraphics[type=png,ext=.xxx,read=.xxx]{lion}` le fi-

chier `lion.xxx` est lu comme s'il s'agissait de `lion.png`. Pour plus ample information, voir Section 22.2.3 [`\DeclareGraphicsRule`], page 218.

|                      |                                                                                                                                                                                                                                                                                            |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>type</code>    | Spécifie le type de graphique.                                                                                                                                                                                                                                                             |
| <code>ext</code>     | Spécifie l'extension du graphique. À n'utiliser qu'en conjonction avec l'option <code>type</code> .                                                                                                                                                                                        |
| <code>read</code>    | Spécifie l'extension du fichier lu. À n'utiliser qu'en conjonction avec l'option <code>type</code> .                                                                                                                                                                                       |
| <code>command</code> | Spécifie une commande à appliquer à ce fichier. À n'utiliser qu'en conjonction avec l'option <code>type</code> . Voir Section 28.1 [Command line options], page 267, pour une discussion sur l'activation de la fonctionnalité <code>\write18</code> pour exécuter des commandes externes. |

### 22.3.2 `\rotatebox`

Synopsis si vous utilisez le paquetage `graphics` :

```
\rotatebox{angle}{matière}
```

Synopsis si vous utilisez le paquetage `graphicx` :

```
\rotatebox{angle}{matière}
```

```
\rotatebox[liste-à-clefs-valeurs]{angle}{matière}
```

Place *matière* dans une boîte et lui applique une rotation de *angle* degrés dans le sens antihoraire.

L'exemple suivant applique une rotation de quarante-cinq degrés aux en-têtes de colonnes.

```
\begin{tabular}{ll}
 \rotatebox{45}{Caractère} & \rotatebox{45}{alphabet marin} \\
 A & & \&ALPHA \\
 B & & \&BRAVO
\end{tabular}
```

Le *matière* peut être quoi que ce soit qui aille dans une boîte, y compris un graphique.

```
\rotatebox[origin=c]{45}{\includegraphics[width=2.5cm]{lion}}
```

Pour placer la matière auquel on désire appliquer une rotation, la première étape est que  $\LaTeX$  met *matière* dans une boîte, avec un point de référence sur la ligne de base à gauche. La deuxième étape est la rotation, par défaut autour du point de référence. La troisième étape est que  $\LaTeX$  calcule une boîte englobante de matière après rotation. Quatrièmement,  $\LaTeX$  déplace cette boîte horizontalement de sorte que son bord de gauche coïncide avec le bord de gauche de la boîte de la première étape (il n'est pas nécessaire qu'elles coïncident verticalement). Cette nouvelle boîte englobante, dans sa nouvelle position, est celle que  $\LaTeX$  utilisera au moment de composer cette matière.

Si vous utilisez le paquetage `graphics` alors la rotation est autour du point de référence de la boîte. Si vous utilisez le paquetage `graphicx` alors voici les options qu'on peut mettre dans *liste-à-clefs-valeurs*, mais notez qu'on peut obtenir le même effet sans ce paquetage, excepté pour les options `x` et `y`.

|                     |                                                                                                                                                                                       |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>origin</code> | Le point de la boîte du <i>matière</i> autour duquel la rotation se produit. Les valeurs possibles sont toute chaîne contenant un ou deux caractères pris parmi : <code>l</code> pour |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

gauche (left), r droite (right), b pour bas, c pour centre, t pour haut (top), et B pour ligne de base (baseline). Ainsi, la première ligne ci-dessous :

```
\rotatebox[origin=c]{180}{LaLune}
\rotatebox[origin=lB]{180}{LeBateau}
```

retourne l'image haut en bas autour de son centre, alors que la seconde la retourne autour de l'extrémité gauche de sa ligne de base. (Le caractère c donne le centre horizontal dans bc ou tc mais donne le centre vertical dans lc ou rc, et donne les deux dans c). Le réglage par défaut est lB.

- x, y** Spécifiez un point de rotation arbitraire avec `\rotatebox[x=dimension TEX,y=dimension TEX]{...}` (voir Section 14.1 [Units of length], page 142). Ces paramètres donnent le décalage relatif au point de référence de la boîte.
- units** Cette clef vous permet d'utiliser une autre unité que celle par défaut, à savoir les degrés antihoraires. Régler `units=-360` change la direction en degrés horaires, et régler `units=6.283185` change l'unité en radians antihoraires.

### 22.3.3 \scalebox

Synopsis :

```
\scalebox{facteur horizontal}{matière}
\scalebox{facteur horizontal}[facteur vertical]{matière}
\reflectbox{matière}
```

Applique une homothétie au *matière*.

Dans l'exemple suivant on divise la taille par deux, à la fois horizontalement et verticalement, du premier texte, et on double la taille du second :

```
\scalebox{0.5}{BOIS MOI} et \scalebox{2.0}{Mange Moi}
```

Si vous ne précisez pas le *facteur vertical* optionnel, alors par défaut il prend la même valeur que le *facteur horizontal*.

Vous pouvez utiliser cette commande pour changer la taille d'un graphique, comme dans :

```
\scalebox{0.5}{\includegraphics{lion}}
```

Si vous utilisez le paquetage `graphicx` alors vous pouvez accomplir la même chose avec les arguments optionnels d'`\includegraphics` (voir Section 22.3.1 [`\includegraphics`], page 220).

La commande `\reflectbox` abrège `\scalebox{-1}[1]{matière}`. Ainsi, `orizabus \reflectbox{orizabus}` affiche le mot 'orizabus' immédiatement suivi de son image par réflexion selon un axe vertical.

### 22.3.4 \resizebox

Synopsis :

```
\resizebox{longueur horizontale}{longueur verticale}{matière}
\resizebox*{longueur horizontale}{longueur verticale}{matière}
```

Étant donnée une taille, telle que `3cm`, transforme le *matière* pour qu'il prenne cette taille. Si soit *longueur horizontale* soit *longueur verticale* est un point d'exclamation ! alors l'autre argument est utilisé pour déterminer un facteur d'échelle appliqué aux deux directions.

Dans l'exemple suivant on rend le graphique 1,3 cm de large et on le dilate/contracte verticalement selon le même facteur pour ne pas changer ses proportions.

```
\resizebox{1.3cm}{!}{\includegraphics{lion}}
```

La forme non étoilée `\resizebox` considère *longueur verticale* comme la hauteur de la boîte alors que la forme étoilée `\resizebox*` la considère comme la somme hauteur + profondeur. Par exemple, rendez le texte d'une hauteur+profondeur de 6 mm avec `\resizebox*{!}{6mm}{\parbox{9cm}{Cette boîte a à la fois de la hauteur et de la profondeur.}}`.

Vous pouvez utiliser `\depth`, `\height`, `\totalheight`, et `\width` pour vous référer aux dimensions originales de la boîte. Ainsi, rendez le texte d'une largeur de 5 cm tout en lui conservant sa hauteur originale avec `\resizebox{5cm}{\height}{Cinq centimètres}`.

## 23 Insertions spéciales

L<sup>A</sup>T<sub>E</sub>X fournit des commandes pour insérer les caractères qui ont une signification spéciale mais ne correspondent à aucun caractère simple que vous pouvez taper.

### 23.1 Caractères réservés

L<sup>A</sup>T<sub>E</sub>X réserve les caractères suivant à un usage spécial (par exemple, le signe pourcent % sert aux commentaires) c'est pourquoi on les appelle *caractères réservés* ou *caractères spéciaux*. Ils sont chacun l'objet d'une discussion ailleurs dans ce manuel.

```
$ % & { } _ ~ ^ \
```

Si vous voulez qu'un caractère réservé soit imprimé comme lui-même, dans la même police que le corps du texte, alors pour tous les caractères hormis les trois derniers de cette liste il suffit de les faire précéder d'une controblaque \. Ainsi saisir `\$1.23` produit \$1.23 en sortie.

Quant aux trois derniers caractères, pour faire un tilde dans la police du corps du texte utilisez `\~{}` (sans les accolades cela mettrait un accent tilde sur le caractère suivant). De même pour faire un accent circonflexe dans la police du corps du texte utilisez `\^{}`. Pour faire une controblaque dans dans la police du corps de texte, saisissez `\textbackslash{}`.

Pour produire les caractères réservés dans la police tapuscrite utilisez `\verb!!` comme ci-dessous (la double controblaque `\\` n'est là que pour aller à la ligne en sortie) :

```
\begin{center}
 \# \$ \% \& \{ \} _ \~{ } \^{ } \textbackslash \\
 \verb!# $ % & { } _ ~ ^ \!
\end{center}
```

### 23.2 Capitales et bas de casse

Synopsis :

```
\uppercase{texte}
\lowercase{texte}
\MakeUppercase{texte}
\MakeLowercase{texte}
```

Change la casse des caractères. Les commandes primitives de T<sub>E</sub>X `\uppercase` et `\lowercase` ne fonctionnent que pour les 26 lettres sans diacritiques a-z et A-Z. Les commandes L<sup>A</sup>T<sub>E</sub>X `\MakeUppercase` et `\MakeLowercase` changent aussi les caractères accessibles par des commandes telles que `\ae` ou `\aa`. Ces commandes `\MakeUppercase` et `\MakeLowercase` sont robustes, mais elles ont des arguments mouvants (voir Section 12.11 [protect], page 134).

Ces commandes ne changent pas la casse des lettres utilisées dans le nom d'une commande au sein de *texte*. Mais par contre elles changent la casse de tout autre lettre latine au sein de l'argument *texte*. Ainsi, `\MakeUppercase{Soit $y=f(x)$}` produit 'SOIT Y=F(X)'. Un autre exemple est que le nom de tout environnement est changé, de sorte que `\MakeUppercase{\begin{tabular} ... \end{tabular}}` produit une erreur puisque la première moitié est changée en `\begin{TABULAR}`.

$\LaTeX$  utilise la même table pour changer la casse tout au long du document. La table utilisée est conçue pour le codage T1 ; ceci fonctionne bien avec les polices standards de  $\TeX$  pour tous les alphabets latins, mais cause des problèmes avec d'autres alphabets.

Pour changer la casse du texte qui résulte d'une macro au sein de *texte* il est nécessaire de faire un développement. Dans l'exemple qui suit la macro `\Nomecole` produit 'UNIVERSITÉ DE MATHÉMATIQUES'.

```
\newcommand{\nomecole}{Universit\'e de math\'ematiques}
\newcommand{\Nomecole}{\expandafter\MakeUppercase\expandafter{\nomecole}}
```

Le paquetage `textcase` comble certaines des lacunes des commandes standards `\MakeUppercase` et `\MakeLowerCase` de  $\LaTeX$ .

Pour mettre en capitale seulement la première lettre d'un mot, on peut utiliser le paquetage `mfirstuc`.

Gérer toutes les règles concernant la casse spécifiée par Unicode, par ex. pour les scripts non latins, est une tâche bien plus conséquente que ce qui avait été considéré dans les  $\TeX$  et  $\LaTeX$  d'origine. Cela a été implémenté dans le paquetage `expl3` à compter de 2020. L'article « Case changing: From  $\TeX$  primitives to the Unicode algorithm », (Joseph Wright, *TUGboat* 41:1, <https://tug.org/TUGboat/tb41-1/tb127wright-case.pdf>), donne un bon aperçu du sujet, passé et présent.

### 23.3 Symboles appelés par leur position dans une police

Vous pouvez accéder à n'importe quel caractère de la police courante en utilisant son numéro avec la commande `\symbol`. Par exemple, le caractère espace visible utilisé dans la commande `\verb*` a le code décimal 32 dans la police standard tapuscrite Computer Modern, ainsi on peut le taper avec `\symbol{32}`.

Vous pouvez également spécifier le numéro en octal (base 8) en utilisant un préfixe `'`, ou en hexadécimal (base 16) avec un préfixe `"`, ainsi l'espace visible de code 32 en décimal pourrait tout aussi bien être écrit comme `\symbol{'40}` ou `\symbol{"20}`.

### 23.4 Symboles en texte

$\LaTeX$  fournit des commandes pour générer divers symboles qui ne sont pas des lettres dans le cours du texte. Certaines d'entre elles, en particulier les plus obscures, ne sont pas disponibles en OT1. Depuis la parution de février 2020 de  $\LaTeX$  tous ces symboles sont disponibles par défaut ; avant cela, il était nécessaire d'utiliser le paquetage `textcomp` pour certains d'entre eux (techniquement, ce qui sont dans le codage de police TS1).

`\copyright`

`\textcopyright`

Le symbole « droit d'auteur », ©.

`\dag`

Le symbole obèle (dans le texte).

`\ddag`

Le symbole double obèle (dans le texte).

`\LaTeX`

Le logo  $\LaTeX$ .

`\LaTeXe`

Le logo  $\LaTeX$ 2e.

`\guillemotleft` («)  
`\guillemotright` (»)  
`\guilsinglleft` (<)  
`\guilsinglright` (>)  
 Guillemets à chevron double et simple, utilisés communément en français : «, » , < , > .

`\ldots`  
`\dots`  
`\textellipsis`  
 Des points de suspension (trois points sur la ligne de base) : ‘...’. `\ldots` et `\dots` peuvent également être utilisés en mode mathématique.

`\lq` guillemet-apostrophe simple de gauche (ouvrant) : ‘.

`\P`  
`\textparagraph`  
 Signe paragraphe : ¶ (pied-de-mouche).

`\pounds`  
`\textsterling`  
 Livre sterling anglais : £.

`\quotedblbase` („)  
`\quotesinglbase` (,)  
 Guillemet-virgule inférieur double et simple : „ et ,.

`\rq` Guillemet-apostrophe simple de droite (fermant) : ’.

`\S` `\itemx` `\textsection` Signe section : §.

`\TeX` Le logo T<sub>E</sub>X.

`\textasciicircum`  
 circonflexe ASCII : ^.

`\textasciitilde`  
 tilde ASCII : ~.

`\textasteriskcentered`  
 Astérisque centré : \*.

`\textbackslash`  
 Controblique : \.

`\textbar` Barre verticale : |.

`\textbardbl`  
 Barre verticale double.

`\textbigcircle`  
 Symbole grand rond.

`\textbraceleft`  
 Accolade gauche : {.

`\textbraceright`  
Accolade droite : }.

`\textbullet`  
Puce : •.

`\textcircled{lettre}`  
*lettre* dans un cercle, comme dans ®.

`\textcompwordmark`  
`\textcapitalcompwordmark`  
`\textascendercompwordmark`  
Marque de mot composé (invisible). La forme `\textcapital...` à la hauteur de capitale de la fonte, alors que la forme `\textascender...` a la hauteur de hampe.

`\textdagger`  
Obèle : †.

`\textdaggerdbl`  
Double-obèle : ‡.

`\textdollar` (ou `\$`)  
Signe Dollar : \$.

`\textemdash` (ou `---`)  
Cadratin : —. Utilisé pour la ponctuation, d'ordinaire similaire à une virgules ou des parenthèses comme dans « Les éliminatoires --- si vous assez de chance pour y arriver --- c'est comme si ça se passait d'un trait. »

`\textendash` (ou `--`)  
Demi cadratin : –. Utilisé pour les plages en anglais comme dans « see pages 12--14 ».

`\texteuro`  
Le symbole monétaire Euro : €. Pour un dessin alternatif du glyphe, essayez le paquetage `eurosym` ; de plus, la plupart des polices de nos jour sont fournies avec leur propre symbole Euro (Unicode U+20AC).

`\textexclamdown` (ou `!'`)  
Point d'exclamation culbuté : ¡.

`\textfiguredash`  
Tiret utilisé entre les nombres, Unicode U+2012. Défini dans la parution juin-2021 de L<sup>A</sup>T<sub>E</sub>X. Quand utilisé avec pdfT<sub>E</sub>X, approximé par un demi-cadratin ; avec un moteur Unicode, soit le glyphe est composé s'il est disponible dans la fonte courante, ou sinon l'avertissement habituel « Missing character » est inscrit au journal de compilation.

`\textgreater`  
Supérieur à : >.

`\texthorizontalbar`  
Caractère barre horizontale, Unicode U+2015. Défini dans la parution juin-2021 de L<sup>A</sup>T<sub>E</sub>X. Comportement similaire à `\textfiguredash` ci-dessus ; l'approximation de pdfT<sub>E</sub>X est un cadratin.

`\textless`  
 Inférieur à : <.

`\textleftarrow`  
 Flèche gauche.

`\textnonbreakinghyphen`  
 Caractère trait d'union insécable, Unicode U+2011. Défini dans la parution juin-2021 de L<sup>A</sup>T<sub>E</sub>X. Comportement similaire à `\textfiguredash` ci-dessus ; l'approximation de pdfT<sub>E</sub>X est un trait d'union ASCII ordinaire (avec saut de ligne non-autorisé juste après).

`\textordfeminine`  
`\textordmasculine`  
 Symboles ordinaux féminin et masculin : <sup>a</sup>, <sup>o</sup>.

`\textperiodcentered`  
 Point centré : .̣.

`\textquestiondown` (ou ?<sup>˘</sup>)  
 Point d'interrogation culbuté : ¿.

`\textquotedblleft` (ou ‘<sup>˘</sup>)  
 Guillemet-apostrophe double culbuté : “.

`\textquotedblright` (ou ’<sup>˘</sup>)  
 Guillemet-apostrophe de droite double : ”.

`\textquoteleft` (ou ‘<sup>˘</sup>)  
 Guillemet-apostrophe simple culbuté : ‘.

`\textquoteright` (ou ’<sup>˘</sup>)  
 Guillemet-apostrophe simple : ’.

`\textquotesingle`  
 Guillemet-apostrophe simple droit. (Du codage TS1.)

`\textquotestraightbase`  
`\textquotestraightdblbase`  
 Guillemets droit simple et double sur la ligne de base.

`\textregistered`  
 Symbole « marque déposée » : ®.

`\textrightarrow`  
 Flèche droite.

`\textthreequartersemdash`  
 Cadratin « trois quarts ».

`\texttrademark`  
 Symbole marque de commerce : ™.

`\texttwelveudash`  
 Deux-tiers cadratin.

`\textunderscore`  
Tiret bas : `_`.

`\textvisiblespace`  
Symbole espace visible.

## 23.5 Accents

L<sup>A</sup>T<sub>E</sub>X a une prise en charge très large de beaucoup des scripts et langages du monde, à travers ce qu'offre le paquetage `babel`. Cette section ne tente pas de couvrir la totalité de cette prise en charge. Elle liste simplement les commandes L<sup>A</sup>T<sub>E</sub>X cœur pour créer des caractères accentués, et plus généralement des caractères portant un signe diacritique.

Les commandes `\capital...` produisent des formes alternatives pouvant être utilisées avec les lettres capitales. Elles ne sont pas disponibles avec l'OT1.

`\"`  
`\capitaldieresis`  
Produit une tréma, comme dans `ö`.

`\'`  
`\capitalacute`  
Produit un accent aigu, comme dans `ó`. Dans l'environnement `tabbing`, pousse la colonne courante à droite de la colonne précédente (voir Section 8.21 [tabbing], page 90).

`\.` Produit un point suscrit à la lettre qui suit, comme dans `ò`.

`\=`  
`\capitalmacron`  
Produit un macron (une barre) suscrit à la lettre qui suit, comme dans `ō`.

`\^`  
`\capitalcircumflex`  
Produit un accent circonflexe (un chapeau) suscrit à la lettre qui suit, comme dans `ô`.

`\``  
`\capitalgrave`  
Produit un accent grave suscrit à la lettre qui suit, comme dans `ò`. Au sein de l'environnement `tabbing`, déplace le texte qui suit à la marge de droite (voir Section 8.21 [tabbing], page 90).

`\~`  
`\capitaltilde`  
Produit un diacritique tilde suscrit à la lettre qui suit, comme dans `ñ`.

`\b` Produit un diacritique barre souscrite à la lettre qui suit, comme dans `ọ`. Voir aussi `\underbar` ci-après.

`\c`  
`\capitalcedilla`  
Produit une cédille souscrite à la lettre qui suit, comme dans `ç`.

|                                   |                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\d</code>                   |                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>\capitaldotaccent</code>    | Produit un point souscrit à la lettre qui suit, comme dans $\text{ø}$ .                                                                                                                                                                                                                                                                                                            |
| <code>\H</code>                   |                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>\capitalhungarumlaut</code> | Produit un long tréma hongrois suscrit à la lettre qui suit, comme dans $\text{ő}$ .                                                                                                                                                                                                                                                                                               |
| <code>\i</code>                   | Produit un i sans point, comme dans ‘ $\text{i}$ ’.                                                                                                                                                                                                                                                                                                                                |
| <code>\j</code>                   | Produit un j sans point, comme dans ‘ $\text{j}$ ’.                                                                                                                                                                                                                                                                                                                                |
| <code>\k</code>                   |                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>\capitalogonek</code>       | Produit un ogonek, comme dans ‘ $\text{Ń}$ ’. Non disponible dans le codage OT1.                                                                                                                                                                                                                                                                                                   |
| <code>\r</code>                   |                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>\capitalring</code>         | Produit un rond en chef, comme dans ‘ $\text{ô}$ ’.                                                                                                                                                                                                                                                                                                                                |
| <code>\t</code>                   |                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>\capitaltie</code>          |                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>\newtie</code>              |                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>\capitalnewtie</code>       | Produit un tirant suscrit, ou double brève renversée, comme dans ‘ $\text{ô̇}$ ’ (utilisé pour la translittération du Cyrillique dans la romanisation ALA-LC). La commande <code>\t</code> attend que l’argument consiste en deux caractères. La forme <code>\newtie</code> est centrée dans sa boîte.                                                                             |
| <code>\u</code>                   |                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>\capitalbreve</code>        | Produit un accent brève, comme dans ‘ $\text{ö}$ ’.                                                                                                                                                                                                                                                                                                                                |
| <code>\underbar</code>            | Ce n’est pas vraiment un diacritique. Produit une barre au-dessous de l’argument texte. L’argument est toujours traité en mode horizontal. La barre est toujours à une position fixée sous la ligne de base, de la sorte elle traverse les descentes. Voir aussi <code>\underline</code> dans Section 16.5 [Over- and Underlining], page 177. Voir aussi <code>\b</code> ci-avant. |
| <code>\v</code>                   |                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>\capitalcaron</code>        | Produit un accent háček (caron), comme dans ‘ $\text{ř}$ ’.                                                                                                                                                                                                                                                                                                                        |

## 23.6 Lettres latines supplémentaires

Voici les commandes de base de  $\text{\LaTeX}$  pour insérer des caractères utilisés généralement utilisés dans des langages autres que l’anglais.

|                  |         |
|------------------|---------|
| <code>\aa</code> |         |
| <code>\AA</code> | å et Å. |
| <code>\ae</code> |         |
| <code>\AE</code> | æ et Æ. |

|                  |                                                                                                                                                                                                                                |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\dh</code> |                                                                                                                                                                                                                                |
| <code>\DH</code> | Lettre islandaise eth : ð et Ð. Non disponible dans le codage OT1, il vous faut le paquetage <code>fontenc</code> pour sélectionner un autre codage de police, tel que T1.                                                     |
| <code>\dj</code> |                                                                                                                                                                                                                                |
| <code>\DJ</code> | d et D barre, lettre d capitale et bas-de-casse avec une barre traversant la hampe. Non disponible dans le codage OT1, il vous faut le paquetage <code>fontenc</code> pour sélectionner un autre codage de police, tel que T1. |
| <code>\ij</code> |                                                                                                                                                                                                                                |
| <code>\IJ</code> | ij et IJ (à ceci près que les deux lettres apparaissent plus liées qu'ici).                                                                                                                                                    |
| <code>\l</code>  |                                                                                                                                                                                                                                |
| <code>\L</code>  | ł et Ł.                                                                                                                                                                                                                        |
| <code>\ng</code> |                                                                                                                                                                                                                                |
| <code>\NG</code> | Lettre lapone eng, utilisée aussi en phonétique. Non disponible dans le codage OT1, il vous faut le paquetage <code>fontenc</code> pour sélectionner un autre codage de police, tel que T1.                                    |
| <code>\o</code>  |                                                                                                                                                                                                                                |
| <code>\O</code>  | ø et Ø.                                                                                                                                                                                                                        |
| <code>\oe</code> |                                                                                                                                                                                                                                |
| <code>\OE</code> | œ et Œ.                                                                                                                                                                                                                        |
| <code>\ss</code> |                                                                                                                                                                                                                                |
| <code>\SS</code> | ß et SS.                                                                                                                                                                                                                       |
| <code>\th</code> |                                                                                                                                                                                                                                |
| <code>\TH</code> | Lettre islandaise thorn : þ et Þ. Non disponible dans le codage OT1, il vous faut le paquetage <code>fontenc</code> pour sélectionner un autre codage de police, tel que T1.                                                   |

## 23.7 paquetage inputenc

Synopsis :

```
\usepackage[nom-codage]{inputenc}
```

Déclare que le codage du texte du fichier d'entrée est *nom-codage*. Par défaut, quand ce paquetage n'est pas chargé, c'est UTF-8. Techniquement, la spécification du nom du codage est optionnelle, mais en pratique il n'est pas utile de l'omettre.

Dans un fichier informatique, les caractères sont stockés selon un schéma appelé le *codage*. Il y a de nombreux différents codages. Le plus simple est l'ASCII, il prend en charge 95 caractères imprimables, soit trop peu pour la plupart des langages du monde. Par exemple, pour composer un e tréma 'ë' dans un fichier source L<sup>A</sup>T<sub>E</sub>X codé en ASCII, on utilise la séquence `\"e`. Ceci rendrait les fichiers source pour quoi que ce soit d'autre que l'anglais difficiles à lire ; et même pour l'anglais il est commode d'avoir un codage plus étendu.

La norme moderne de codage, d'une certaine façon une réunion des autres, est l'UTF-8, l'une des représentations de l'Unicode. C'est le codage par défaut de L<sup>A</sup>T<sub>E</sub>X depuis 2018.

Le paquetage `inputenc` permet à L<sup>A</sup>T<sub>E</sub>X de savoir quel codage est utilisé. Par exemple, la commande suivante dit explicitement que le fichier d'entrée est en UTF-8 (notez l'absence de tiret).

```
\usepackage[utf8]{inputenc}
```

Attention : n'utilisez `inputenc` qu'avec le moteur pdfT<sub>E</sub>X (voir Section 2.3 [T<sub>E</sub>X engines], page 3). (Les moteurs XeT<sub>E</sub>X et LuaT<sub>E</sub>X supposent que le fichier d'entrée est codé en UTF-8). Si vous invoquez L<sup>A</sup>T<sub>E</sub>X avec soit la commande `xelatex` soit la commande `lualatex`, et essayez de déclarer un codage distinct d'UTF-8 avec `inputenc`, comme par ex. `latin1`, alors vous obtiendrez l'erreur `inputenc is not designed for xetex or luatex`.

Une erreur du paquetage `inputenc` telle que `Invalid UTF-8 byte "96` signifie que le fichier d'entrée contient de la matière qui ne suit pas le schéma de codage. Souvent ces erreurs proviennent de la copie de matière issue d'un document utilisant un codage différent de celui du fichier d'entrée ; l'erreur donnée en exemple provient d'un guillemet anglais simple copié d'une page web utilisant `latin1` vers un fichier d'entrée L<sup>A</sup>T<sub>E</sub>X utilisant UTF-8. La solution la plus simple consiste à remplacer tout caractère non-UTF-8 avec leur équivalent UTF-8, ou d'utiliser une commande ou un caractère équivalent en L<sup>A</sup>T<sub>E</sub>X.

Dans certains documents, tel que des collections d'articles de revue écrits pas différents auteurs, changer le codage en milieu de document peut s'avérer nécessaire. Utiliser la commande `\inputencoding{nom-codage}`. Les valeurs les plus usuelles pour *nom-codage* sont : `ascii`, `latin1`, `latin2`, `latin3`, `latin4`, `latin5`, `latin9`, `latin10`, et `utf8`.

## 23.8 \rule

Synopsis :

```
\rule[élévation]{largeur}{épaisseur}
```

la commande `\rule` produit un *filet*, c.-à-d. une ligne ou un rectangle. Les arguments sont :

*élévation* De combien élever le filet (optionnel).

*largeur* La longueur du filet (obligatoire).

*épaisseur* L'épaisseur du filet (obligatoire).

## 23.9 \today

La commande `\today` produit la date d'aujourd'hui, par défaut dans le format '*mois jj, aaaa*' ; par exemple, '`July 4, 1976`'. Elle utilise les compteurs prédéfinis `\day`, `\month`, et `\year` (voir Section 13.8 [`\day & \month & \year`], page 140) pour faire cela. Elle n'est pas mise à jour durant l'exécution de la compilation.

Les extensions multilingues comme entre autres le paquetage `babel` ou la classe `lettre` localisent `\today`. Par exemple le code suivant produit '`4 juillet 1976`' :

```
\year=1976 \month=7 \day=4
\documentclass{minimal}
\usepackage[french]{babel}
\begin{document}
\today
```

```
\end{document}
```

Le paquetage `datetime`, entre autres, produit un large choix d'autres formats de date.

## 24 Partitionner le fichier d'entrée

$\LaTeX$  vous permet de partitionner un document de grande taille en plusieurs plus petits. Ceci peut simplifier son édition et permette à plusieurs auteurs d'y travailler. Cela peut aussi accélérer le traitement.

Indépendamment du nombre de fichiers distincts que vous utilisez, il y a toujours un seul *fichier racine* sur lequel la compilation  $\LaTeX$  démarre. L'exemple ci-dessous illustre un tel fichier avec cinq fichiers inclus.

```

\documentclass{book}
\includeonly{
 % mettre en commentaire les lignes ci-dessous pour ne pas compiler le fichier
 pref,
 chap1,
 chap2,
 append,
 bib
}
\begin{document}
\frontmatter
\include{pref}
\mainmatter
\include{chap1}
\include{chap2}
\appendix
\include{append}
\backmatter
\include{bib}
\end{document}

```

Dans cet exemple on prend la matière de `pref.tex`, `chap1.tex`, `chap2.tex`, `append.tex`, et `bib.tex`. Si vous compilez ce fichier, et ensuite mettez en commentaire toutes les lignes au sein de `\includeonly{...}` hormis `chap1`, et compilez de nouveau, alors  $\LaTeX$  traite seulement la matière du premier chapitre. Ainsi, la sortie apparaît plus rapidement et est plus courte à imprimer. Cependant, l'avantage de la commande `\includeonly` est que  $\LaTeX$  retient les numéros de page et toute l'information de renvoi des autres parties du document de sorte qu'elles apparaissent correctement en sortie.

Voir Section A.4 [Larger book template], page 276, pour un autre exemple de `\includeonly`.

### 24.1 `\endinput`

Synopsis :

```
\endinput
```

Quand vous écrivez `\include{nomfichier}`, alors au sein de `nomfichier.tex` la matière qui vient après `\endinput` n'est pas incluse. Cette commande est optionnelle ; si `nomfichier.tex` ne comprend pas de `\endinput` alors  $\LaTeX$  lit tout le fichier.

Par exemple, supposons que le fichier racine d'un document a un `\input{chap1}` que ce le contenu de `chap1.tex` est ce qui suit :

```
\chapter{One} Cette matière apparaît dans le document. \endinput Ceci
n'apparaît pas.
```

Ceci peut être utile pour placer de la documentation ou des commentaires à la fin d'un fichier, ou pour éviter des caractères indésirables qui pourraient s'ajouter si le fichier est transmis dans le corps d'un courriel. C'est également utile pour déboguer : une stratégie pour localiser des erreurs est de placer un `\endinput` à mi-chemin en cours des fichiers inclus et de voir si l'erreur disparaît. Ensuite, sachant quelle moitié contient l'erreur, en déplaçant le `\endinput` à mi-chemin de cette zone circonscrit de plus près l'endroit où elle se trouve. Ainsi on trouve rapidement la ligne fautive.

Après avoir lu `\endinput`,  $\LaTeX$  continue à lire jusqu'à la fin de la ligne, ainsi il est possible de faire suivre cette ligne par quelque chose qui sera tout de même lu. Ceci vous permet entre autre de fermer un `\if...` par un `\fi`.

## 24.2 `\include` & `\includeonly`

Synopsis :

```
\includeonly{ % dans le préambule du document
...
nomfichier,
...
}
...
\include{nomfichier} % dans le corps du document
```

Apporte de la matière d'un fichier externe `nomfichier.tex` dans un document  $\LaTeX$ .

La commande `\include` fait trois choses : elle exécute `\clearpage` (voir Section 10.1 [`\clearpage` & `\cleardoublepage`], page 117), puis elle insère la matière à partir du fichier `nomfichier.tex` dans le document, ensuite fait encore un `\clearpage`. Cette commande ne peut apparaître que dans le corps du document.

La commande `\includeonly` contrôle quels fichiers seront lus par  $\LaTeX$  lors des commandes `\include` suivantes. Sa liste de noms de fichier est séparée par des virgules, et la commande ne peut apparaître que dans le préambule, ou même plus tôt, par ex. dans la ligne de commande : elle ne peut pas apparaître dans le corps du document.

Dans le document racine en exemple ci-dessous, `constitution.tex`, on inclut trois fichiers, `preambule.tex`, `articles.tex`, et `amendements.tex`.

```
\documentclass{book}
\includeonly{
 preambule,
 articles,
 amendements
}
\begin{document}
\include{preambule}
\include{articles}
```

```
\include{amendements}
\end{document}
```

Le fichier `preambule.tex` ne contient aucun code spécial ; vous avez juste extrait ce chapitre de `constitution.tex` pour le mettre dans un fichier séparé pour rendre son édition plus commode.

```
\chapter{Préambule}
Le peuple français proclame solennellement son attachement aux Droits de
l'Homme et aux principes de la souveraineté nationale tels qu'ils ont
été définis par la Déclaration de 1789, ...
```

Exécuter  $\LaTeX$  sur `constitution.tex` fait apparaître la matière issue de ces trois fichiers dans le document mais génère également les fichiers auxiliaires `preambule.aux`, `articles.aux`, et `amendements.aux`. Ceux-ci contiennent de l'information telle que les numéros de page et les renvois (voir Chapitre 7 [Cross references], page 55). Si maintenant vous mettez en commentaire les lignes de `\includeonly` contenant `preambule` et `amendements` et exécutez  $\LaTeX$  de nouveau alors le document résultant ne contiendra que la matière issue de `articles.tex`, mais pas celui de `preambule.tex` ni de `amendements.tex`. Néanmoins, la totalité de l'information auxiliaire issue des fichiers omis est toujours là, y compris le numéro de la page où commence le chapitre.

S'il n'y a pas de `\includeonly` dans le préambule du document, alors  $\LaTeX$  inclut tous les fichiers demandés par des commandes `\include`.

La commande `\include` fabrique une nouvelle page. Pour éviter cela, voir Section 24.3 [`\input`], page 241, (ce qui, cependant, ne retient pas l'information auxiliaire).

Voir Section A.4 [Larger book template], page 276, pour un autre exemple utilisant `\include` et `\includeonly`. Cet exemple utilise également `\input` pour certaine matière qui ne débute pas nécessairement sur une nouvelle page.

Les noms de fichiers peuvent impliquer un chemin.

```
\documentclass{book}
\includeonly{
 chapitres/chap1,
}
\begin{document}
\include{chapitres/chap1}
\end{document}
```

Pour rendre votre document portable vis à vis des diverses distributions et plateformes vous devriez éviter les espaces dans les noms de fichier. Traditionnellement on utilise des tirets de 6 ou de 8 à la place. Néanmoins, pour le nom `'amo amas amat'`, ce qui suit fonctionne sous  $\TeX$  Live sur GNU/Linux:

```
\documentclass{book}
\includeonly{
 "amo\space amas\space amat"
}
\begin{document}
\include{"amo\space amas\space amat"}
\end{document}
```

et ceci fonctionne sous MiKTeX sur Windows:

```
\documentclass{book}
\includeonly{
 {"amo amas amat"}
}
\begin{document}
\include{"amo amas amat"}
\end{document}
```

Vous ne pouvez pas utiliser `\include` au sein d'un fichier qui lui-même est inclus via `\include`, sans quoi vous obtenez l'erreur 'LaTeX Error: \include cannot be nested'. La commande `\include` ne peut pas apparaître dans le préambule ; sans quoi vous obtenez l'erreur 'LaTeX Error: Missing \begin{document}'.

Si un fichier que vous incluez par `\include` n'existe pas, par exemple si vous tapez la commande `\include{athiesme}` par erreur à la place de `\include{atheisme}`, alors L<sup>A</sup>T<sub>E</sub>X ne produit pas une erreur mais un avertissement 'No file athiesme.tex.' (Il crée également le fichier `athiesme.aux`).

Si vous faites un `\include` du fichier racine dans lui-même alors vous obtenez tout d'abord 'LaTeX Error: Can be used only in preamble'. Les exécutions suivantes donnent 'TeX capacity exceeded, sorry [text input levels=15]'. Pour réparer cela, vous devez enlever l'inclusion `\include{racine}` mais aussi détruire le fichier `racine.aux` et exécuter de nouveau L<sup>A</sup>T<sub>E</sub>X.

### 24.3 \input

Synopsis :

```
\input{nomfichier}
```

La commande `\input` a pour effet que le *nomfichier* spécifié soit lu et traité comme si son contenu était inséré dans le fichier courant à cet endroit.

Si *nomfichier* ne se termine pas en '.tex' (par ex., 'toto' ou 'toto.tata'), on essaie en premier en ajoutant cette extension ('toto.tex' ou 'toto.tata.tex'). Si aucun fichier n'est trouvé avec ce nom, alors on essaie le *nomfichier* original (c.-à-d. 'toto' ou 'toto.tata').

## 25 Parties pré/postliminaires

### 25.1 Table des matières etc.

Synopsis, l'un parmi ceux qui suivent :

```
\tableofcontents
\listoffigures
\listoftables
```

Produit une table des matières, ou une liste des figures, ou une liste des tableaux. Placez la commande à l'endroit même où vous voulez que la table ou liste apparaisse. Vous n'avez pas à la renseigner ; par exemple, typiquement les entrées de la table des matières sont automatiquement générées à partir des commandes de rubricage `\chapter`, etc.

L'exemple ci-dessous illustre la première commande, `\tableofcontents`.  $\text{\LaTeX}$  produit une table des matières sur la première page du livre.

```
\documentclass{book}
% \setcounter{tocdepth}{1}
\begin{document}
\tableofcontents\newpage
...
\chapter{...}
...
\section{...}
...
\subsection{...}
...
\end{document}
```

En décommentant la deuxième ligne on peut limiter la table à ne lister que les rubriques issues de `\chapter` et `\section`, mais pas celle issue de `\subsection`, parce que la commande `\section` a pour niveau 1. Voir Chapitre 6 [Sectioning], page 42, pour les numéros de niveaux des rubriques. Pour plus ample information sur `tocdepth` voir [Sectioning/tocdepth], page 43.

Un autre exemple de l'utilisation de `\tableofcontents` est dans Section A.4 [Larger book template], page 276.

Si vous désirez un saut de page après la table des matières, insérez une commande `\newpage` après la commande `\tableofcontents`, comme ci-dessus.

Pour fabriquer la table des matières,  $\text{\LaTeX}$  stocke l'information dans un fichier auxiliaire nommé *fichier-racine.toc* (voir Chapitre 24 [Splitting the input], page 238). Par exemple, le fichier  $\text{\LaTeX}$  *test.tex* ci-dessous :

```
\documentclass{article}
\begin{document}
\tableofcontents\newpage
\section{Premièrement}
\subsection{Deuxièmement}
...
```

écrit la ligne suivante dans `test.toc` :

```
\contentsline {section}{\numberline {1}Premièrement}{2}
\contentsline {subsection}{\numberline {1.1}Deuxièmement}{2}
```

L'indication `section` ou `subsection` donne le type de rubrique. Le crochet `\numberline` vous permet de changer la façon dont l'information apparaît au sein de la table des matières. Son unique argument, `1` ou `1.1`, est le numéro de la rubrique alors que le reste du deuxième argument de `\contentsline`, `Premièrement` ou `Deuxièmement`, est le titre. Finalement, le troisième argument, `2`, est le numéro de la page à laquelle les rubriques débutent.

Une conséquence de cette stratégie de stockage de fichier auxiliaire est que pour obtenir correctement la page de table des matières vous devez exécuter  $\LaTeX$  deux fois, une fois pour stocker l'information et une fois pour l'obtenir. En particulier, la première fois que vous exécutez  $\LaTeX$  sur un nouveau document, la page de table des matières est vide hormis le titre 'Table des matières'. Il suffit de l'exécuter de nouveau.

Les commandes `\listoffigures` et `\listoftables` produisent une liste des figures et une liste des tables. Elles fonctionnent de la même façon que les commandes pour la table des matières ; notamment elles fonctionnent avec l'information stockée dans des fichiers `.lof` et `.lot`.

Pour modifier le titre de la page de table des matières, inspirez vous la première ligne de l'exemple suivant :

```
\renewcommand{\contentsname}{Table des matières}
\renewcommand{\listfigurename}{Tracé de courbes}
\renewcommand{\listtablename}{Tableaux}
```

De même, les deux autres lignes s'occupent des deux autres. Des paquetages d'internationalisation tels que `babel` ou `polyglossia` modifient les titres en fonction de la langue de base sélectionnée.

Le CTAN a beaucoup de paquetages traitant de la table des matières et des listes de figures et de tableaux. L'un d'eux bien commode pour ajuster certains aspects du traitement par défaut, tels que l'espacement, est `tocloft`. Et `tocbibbind` permet d'ajouter automatiquement la bibliographie, les indexes, etc. à la table des matières.

### 25.1.1 `\@dottedtocline`

Synopsis :

```
\@dottedtocline{numnivrubrique}{renforcement}{largeurnum}{texte}{numpage}
```

Utilisée en interne par  $\LaTeX$  pour formater une ligne d'article dans la table des matières, table des figures ou table des tableaux. Les auteurs ne saisissent pas directement des commandes `\@dottedtocline`.

Cette commande est typiquement utilisée par `\l@section`, `\l@subsection`, etc., pour formater le contenu de ces tables. Par exemple, le fichier `article.cls` contient ces définitions :

```
\newcommand*\l@section{\@dottedtocline{1}{1.5em}{2.3em}}
\newcommand*\l@subsection{\@dottedtocline{2}{3.8em}{3.2em}}
\newcommand*\l@subsubsection{\@dottedtocline{3}{7.0em}{4.1em}}
```

Dans cet exemple, on dirait `\@dottedtocline` ne prend que trois argument. Mais en remontant le code on s'aperçoit qu'elle prend également les deux arguments finaux `texte` et `numéro-page` du synopsis d'appel de `\contentsline` (voir Section 25.1.4 [`\contentsline`], page 246).

Entre la boîte du texte de titre de rubrique et la marge de droite, ces commandes `\@dottedtocline` insèrent des *points de suite*, c.-à-d. une ligne en pointillés uniformément espacés. L'espace inter-point est donné par la commande `\@dotsep`. Par défaut c'est 4.5 (exprimé en unités de longueur mathématique, ou  $\mu$ , soit 1/18 em. On peut le modifier avec `\renewcommand`, comme dans `\renewcommand{\@dotsep}{3.5}`.

Dans la classe standard `book`, L<sup>A</sup>T<sub>E</sub>X n'utilise pas de points de suite pour articles de la table correspondant aux rubriques de niveau `\part` ou `\chapter`, et dans la classe `article`, il n'en utilise pas pour les articles de niveau `\section`.

### 25.1.2 `\addcontentsline`

Synopsis :

```
\addcontentsline{ext}{unit}{texte}
```

Ajoute un article dans le fichier auxiliaire dont l'extension de nom est *ext*.

L'exemple suivant a pour effet d'ajouter une ligne 'Appendices' à la table des matières :

```
\addcontentsline{toc}{section}{\protect\textbf{Appendices}}
```

elle apparaîtra au même niveau de renforcement que les rubriques, sera en caractères gras, et recevra le numéro de page correspondant au point où la commande apparaît dans le fichier d'entrée.

La commande `\addcontentsline` écrit de l'information dans le fichier *nom-racine.ext*, où *nom-racine* est le nom du fichier racine (voir Chapitre 24 [Splitting the input], page 238). Il écrit cette information comme le texte de la commande `\contentsline{unit}{texte}{numéro}`, où *numéro* est la valeur courante du compteur *unit* (voir Section 25.1.4 [`\contentsline`], page 246). Le cas le plus fréquent est la table des matières et dans ce cas *numéro* est la valeur du compteur *unit* dans la première page produite où la commande `\addcontentsline` correspondante fut utilisée.

Cette commande est invoquées par les commandes de rubricage `\chapter`, etc. (voir Chapitre 6 [Sectioning], page 42), et aussi par `\caption` au sein d'un environnement flottant (voir Section 5.7 [Floats], page 37). Mais elle est aussi directement utilisées par des auteurs. Par exemple, un auteur écrivant un livre dont le style est d'avoir une préface non numérotée pourrait utiliser `\chapter*`. Mais cette commande ne produit pas d'article dans la table des matières, on peut alors l'entrer manuellement ainsi :

```
\chapter*{Preface}
\addcontentsline{toc}{chapter}{\protect\numberline{ }Préface}
```

Dans le fichier *nom-racine.toc* L<sup>A</sup>T<sub>E</sub>X insère alors la ligne `\contentsline{chapter}{\numberline{ }Préface}{3}` ; notez que le numéro de page '3' est généré automatiquement par le système, et non saisi manuellement.

Tous les arguments de `\addcontentsline` sont obligatoires.

*ext* Typiquement c'est l'une des trois chaînes suivantes : `toc` pour la table des matières (« table of contents » en langue anglaise), `lof` pour la liste des figures (« list of figures »), ou `lot` pour la liste des tableaux (« list of tables »). L'extension de nom de fichier du fichier dans lequel l'information doit être écrite.

|                  |                                                                                                                                                                                                   |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>unit</i>      | Une chaîne dépendant de la valeur de l'argument <i>ext</i> , typiquement l'une des suivantes :                                                                                                    |
| <code>toc</code> | Pour la table des matières, c'est le nom d'une unité sectionnelle : <code>part</code> , <code>chapter</code> , <code>section</code> , <code>subsection</code> , <code>subsubsection</code> , etc. |
| <code>lof</code> | Pour la liste des figures : <code>figure</code> .                                                                                                                                                 |
| <code>lot</code> | Pour la liste des tableaux : <code>table</code> .                                                                                                                                                 |
| <i>texte</i>     | Le texte de l'entrée. On doit précéder de <code>\protect</code> toute commande fragile (voir Section 12.11 [ <code>\protect</code> ], page 134) y figurant.                                       |

La commande `\addcontentsline` a une interaction avec `\include` (voir Section 24.2 [`\include` & `\includeonly`], page 239). Si vous les utilisez au même niveau comme dans `\addcontentsline{...}{...}{...}\include{...}`, alors les lignes dans la table des matières risquent de sortir dans un ordre erroné. La solution est de déplacer `\addcontentsline` dans le fichier qu'on inclut.

Si vous utiliser une *unit* que L<sup>A</sup>T<sub>E</sub>X ne reconnaît pas, comme la coquille dans l'exemple suivant

```
\addcontentsline{toc}{setcion}{\protect\textbf{Appendices}}
```

alors vous n'obtiendrez pas une erreur mais le formatage de la table des matières sera incohérent.

### 25.1.3 `\addtocontents`

Synopsis :

```
\addtocontents{ext}{texte}
```

Ajoute *texte*, qui peut être du texte ou des commandes de formatage, directement au fichier auxiliaire dont le nom a l'extension `.ext`. Cela est utilisé d'ordinaire pour la table des matières, mais aussi pour la liste des figures ou celle des tableaux.

L'exemple suivant place un peu d'espace vertical dans la table des matières après l'en-tête 'Table des matières' :

```
\tableofcontents\newpage
\addtocontents{toc}{\protect\vspace*{3ex}}
```

L'exemple suivant place le mot 'page', en caractères gras, au dessus de la colonne des numéros de page après l'en-tête.

```
\tableofcontents
\addtocontents{toc}{~\hfill\textbf{page}\par}
\chapter{...}
```

L'exemple ci-dessous ajoute une ligne annonçant l'œuvre d'un nouvel auteur :

```
\addtocontents{toc}{%
\protect\vspace{2ex}
\textbf{Chapitres de N. Autre Auteur}\par}
```

La différence entre `\addtocontents` et `\addcontentsline` est que cette dernière commande est strictement réservée aux lignes, comme le cas d'une ligne donnant le numéro de page pour le début d'un nouveau sous-ensemble des chapitres. Comme les exemples ci-dessus le démontrent, `\addtocontents` permet de la manière telle que de l'espacement.

La commande `\addtocontents` a deux arguments, tous deux obligatoires.

*ext* Typiquement l'un parmi : `toc` pour la table des matières, `lof` pour la liste des figures, ou `lot` pour la liste des tableaux. L'extension de nom de fichier du fichier dans lequel l'information est à écrire.

*texte* Le texte, et possiblement les commandes, à écrire.

Les commandes de rubricage telles que `\chapter` utilisent la commande `\addcontentsline` pour stocker l'information. Cette commande crée des lignes dans le fichier auxiliaire en `.toc` qui contient les commandes `\contentsline` (voir Section 25.1.2 [`\addcontentsline`], page 244). Contrairement à elle, la commande `\addtocontents` place directement la matière dans ce fichier.

La commande `\addtocontents` interagit avec `\include` (voir Section 24.2 [`\include` & `\includeonly`], page 239). Si vous les utilisez au même niveau, comme dans `\addtocontents{...}{...}\include{...}` alors les lignes de la table des matières risquent de sortir dans le mauvais ordre. La solution est de déplacer `\addtocontents` pour la mettre dans le fichier à inclure.

### 25.1.4 `\contentsline`

Synopsis :

```
\contentsline{unit}{texte}{numéro-page}
```

Utilisé en interne par  $\text{\LaTeX}$  pour composer un article dans une table des matières, table des figures, ou table des tableaux (voir Section 25.1 [Table of contents etc.], page 242). Les auteurs ne saisissent pas directement les commandes `\contentsline`.

En général l'ajout de matière à ces tables est fait automatiquement par les commandes `\chapter`, `\section`, etc. pour la table des matières, ou par la commande `\caption` au sein d'un environnement `\figure` ou `\table` (voir Section 8.10 [figure], page 67, et voir Section 8.22 [table], page 96). Ainsi, en supposant que le fichier racine est `these.tex`, et qu'il contient la déclaration `\tableofcontents`, la commande `\chapter{Chapitre premier}` produit quelque chose de ce genre dans le fichier `these.toc` :

```
\contentsline {chapter}{\numberline {1}Chapitre premier}{3}
```

Si le fichier contient la déclaration `\listoffigures` alors un environnement figure comprenant `\caption{Essai}` produira quelque chose de ce genre dans `these.lof`.

```
\contentsline {figure}{\numberline {1.1}{\ignorespaces Essai}}{6}
```

x Pour ajouter manuellement de la matière, utilisez `\addcontentsline{typefic}{unit}{\texte}`, où *typefic* est `toc`, `lof`, ou `lot` (voir Section 25.1.2 [`\addcontentsline`], page 244).

Pour manipuler la façon dont la matière de la commande `\contentline` est composé, voir le paquetage `tocloft`.

Notez que le paquetage `hyperref` change la définition de `\contentsline` (et de `\addcontentsline`) pour ajouter plus d'arguments, fabriquer des hyperliens. C'est la source de l'erreur `Argument of \contentsline has an extra }` quand on ajoute/supprime l'utilisation d'`hyperref` et qu'une compilation a déjà été faite. Pour réparer cette erreur, détruisez les fichiers `.toc`, `.lof` ou `.lot`, et exécutez de nouveau  $\text{\LaTeX}$ .

### 25.1.5 `\nofiles`

Synopsis :

```
\nofiles
```

Empêche L<sup>A</sup>T<sub>E</sub>X d'écrire quelque fichier auxiliaire que ce soit. Les seules sorties seront les fichiers `.log` et `.pdf` (ou `.dvi`). Cette commande ne peut être placée que dans le préambule.

À cause de la commande `\nofiles` dans cet exemple aucun fichier `.toc` n'est produit.

```
\documentclass{book}
\nofiles
\begin{document}
\tableofcontents\newpage
\chapter{...}
...
```

L<sup>A</sup>T<sub>E</sub>X n'efface pas les fichiers auxiliaires existants, de sorte que si vous insérez la commande `\nofiles` après avoir compilé le fichier et obtenu un fichier `.toc` alors la page de table des matières contiendra l'ancienne information.

### 25.1.6 `\numberline`

Synopsis :

```
\numberline{\numéro}
```

Compose son argument fer à gauche dans une boîte. Cela est utilisé au sein d'une commande `\contentsline` pour composer les numéros de rubrique (voir Section 25.1.4 [`\contentsline`], page 246).

Par exemple, cette ligne dans un fichier `.toc` a pour effet que le 1.1 est composé fer à gauche :

```
\contentsline {subsection}{\numberline {1.1}Motivation}{2}
```

Par défaut, L<sup>A</sup>T<sub>E</sub>X compose tout numéro de rubrique au sein d'une boîte de longueur `\@tempdima`. Cette longueur est réglée par les commandes `\l@section`, `\l@subsection`, etc. Pour Mettre les numéros de rubrique dans des boîtes de longueur naturelle utiliser le code `\renewcommand{\numberline}[1]{#1~}` avant `\tableofcontents`.

Cette commande est fragile, aussi vous pourriez avoir besoin de la précéder d'un `\protect` (voir Section 12.11 [`\protect`], page 134). Un exemple est l'utilisation de `\protect` dans la commande suivante :

```
\addcontentsline{toc}{section}{\protect\numberline {}Sommaire}
```

de sorte à obtenir le `\numberline` dans la commande `\contentsline` dans le fichier `.toc` ainsi : `\contentsline {section}{\numberline {}Sommaire}{6}` (le numéro de page '6' est ajouté automatiquement par L<sup>A</sup>T<sub>E</sub>X ; voir Section 25.1.2 [`\addcontentsline`], page 244).

## 25.2 Glossaires

La commande `\makeglossary` active la création des glossaires.

La commande `\glossary{texte}` écrit un article de glossaire pour `texte` dans un fichier auxiliaire nommé avec l'extension `.glo`.

Plus précisément, ce qui est écrit est la commande `\glossaryentry{texte}{numpage}`, où `numpage` est la valeur courante de `\thepage`.

Le paquetage `glossary` disponible sur le CTAN fournit une prise en charge de glossaires plus élaborée.

## 25.3 Index

Si vous dites à  $\text{\LaTeX}$  les termes que vous voulez voir apparaître dans un index, alors il peut produire cet index, classé alphabétiquement et en maintenant automatiquement les numéros de page. Ce qui suit illustre les bases.

```

\documentclass{article}
\usepackage{makeidx} % Fournit les commandes d'indexation
\makeindex
%\usepackage{showidx} % Affiche les notes en marge des articles
 % de l'index
...
\begin{document}
...
Le théorème de Wilson\index{Théorème de Wilson} dit qu'un nombre $n>1$
est premier si et seulement si la factorielle de $n-1$ est congruente à
-1 modulo n.\index{congruence!et théorème de Wilson}
...
\printindex
\end{document}

```

Comme illustré ci-dessus, on déclare des articles d'index avec la commande `\index` (voir Section 25.3.2 [`\index`], page 249). Quand vous exécutez  $\text{\LaTeX}$ , le `\index` inscrit son information, telle que 'Théorème de Wilson', et le numéro de page, dans un fichier auxiliaire dont le nom finit en `.idx`. Ensuite, pour classer par ordre alphabétique, et pour d'autres manipulations, on exécute un programme externe, typiquement `makeindex` (voir Section 25.3.3 [`makeindex`], page 251), ce qui écrit un fichier dont le nom finit en `.ind`. Finalement, `\printindex` ramène cette information manipulée dans la sortie (voir Section 25.3.4 [`\printindex`], page 254).

Ainsi, si le code de l'exemple plus haut est dans le fichier `numth.tex` alors exécuter '`pdflatex numth`' sauvegarde l'information de l'article d'index et son numéro de page dans `numth.idx`. Et ensuite exécuter '`makeindex numth`' classe par ordre alphabétique et sauvegarde les résultats dans `numth.ind`. Finalement, exécuter de nouveau '`pdflatex numth`' affiche l'index désiré, à l'endroit où la commande `\printindex` est dans le code source.

Il, y a beaucoup d'options pour contrôler la sortie. Un exemple est que le point d'exclamation dans `\index{congruence!et théorème de Wilson}` produit un article principal pour 'congruence' avec un sous-article pour 'et théorème de Wilson'. Pour plus d'information, voir Section 25.3.3 [`makeindex`], page 251.

Les commandes `\makeindex` et `\printindex` sont indépendantes. Ne pas mettre le `\makeindex` arrête que  $\text{\LaTeX}$  sauvegarde les articles d'index dans le fichier auxiliaire. Ne pas mettre le `\printindex` a pour effet que  $\text{\LaTeX}$  n'affiche pas l'index dans la sortie.

Nombre de paquetages existent concernant l'indexation. Le paquetage `showidx` a pour effet que chaque article d'index est affiché dans la marge sur la page où le `\index` apparaît. Ceci peut être utile pour préparer l'index. Le paquetage `multind`, avec d'autres, prend en

charge les indexes multiples. Voir aussi l'article de la FAQ T<sub>E</sub>X sur ce sujet, <https://www.texfaq.org/FAQ-multind>, et le sujet CTAN, <https://ctan.org/topic/index-multi>.

### 25.3.1 Produire l'index manuellement

Les documents qui sont brefs et statiques peuvent avoir un index produit manuellement. L'exemple ci-dessous produit une page séparée intitulée 'Index', en format à deux-colonnes.

```
\begin{theindex}
\item acorn squash, 1
\subitem maple baked, 2
\indexspace
\item bacon, 3
\subitem maple baked, 4
\end{theindex}
```

Notez que l'auteur doit saisir les numéros de page, ce qui est rébarbatif et résulte en des numéros inexacts si le document change. C'est pourquoi dans la plupart des cas des méthodes automatisées telles que `makeindex` sont préférables. Voir Section 25.3 [Indexes], page 248.

Toutefois nous couvrons ces commandes pour être exhaustif, et parce que les méthodes automatisées sont basées sur elles. Comme l'exemple l'illustre, un article principal utilise `\item`, un sous-article `\subitem`, et le niveau le plus bas utilise `\subsubitem`. Les lignes à blanc entre les articles sont sans effet. L'exemple ci-dessus inclut `\indexspace` pour produire un espace vertical dans la sortie que certains styles utilisent avant le premier article commençant avec une nouvelle lettre.

### 25.3.2 `\index`

Synopsis :

```
\index{chaîne-article-index}
```

Déclare un article dans l'index. Cette commande est fragile (voir Section 12.11 [`\protect`], page 134).

Par exemple, comme décrit dans Section 25.3 [Indexes], page 248, une façon d'obtenir un index du code ci-dessous est de compiler le document avec `pdflatex test`, puis de traiter les articles de l'index avec `makeindex test`, et ensuite de compiler de nouveau avec `pdflatex test`.

```
% fichier test.tex
...
W~Ackermann (1896--1962).\index{Ackermann}
...
fonction d'Ackermann\index{Ackermann!fonction}
...
taux de croissance\index{Ackermann!fonction!taux de croissance}
```

Les trois articles obtiendront un numéro de page, comme dans 'Ackermann, 22'. L<sup>A</sup>T<sub>E</sub>X formate le deuxième comme un sous-article du premier, sur la ligne du dessous et renforcé, et le troisième comme un sous-article du deuxième. On ne peut pas imbriquer les articles au delà du troisième niveau de sous-articles. (Si vous ajoutez `\index{Ackermann!fonction!taux de croissance!comparaison}` alors `makeindex` dit

‘Scanning input file test.idx...done (4 entries accepted, 1 rejected)’ et le quatrième niveau est silencieusement absent de l’index).

Si vous saisissez un deuxième `\index` avec la même *chaîne-article-index* alors vous obtiendrez un unique article d’index avec deux numéros de page (à moins qu’il se trouve que les deux soient sur la même page). Ainsi, ajouter comme pour `Ackermann.\index{Ackermann}` plus loin dans le même document que ci-dessus donnera un article d’index du genre de ‘Ackermann, 22, 151’. Notez aussi que vous pouvez saisir les articles d’index dans un ordre quelconque, ainsi par exemple `\index{Ackermann!fonction}` peut arriver avant `\index{Ackermann}`.

Obtenez une page de page en sortie, du genre de ‘Hilbert, 23--27’, comme avec le code ci-dessous :

```
W~Ackermann (1896--1962).\index{Ackermann}
...
D~Hilbert (1862--1943)\index{Ackermann!Hilbert|()}
...
désapprouvait son mariage.\index{Ackermann!Hilbert|)}
```

Si le début et la fin de la plage de pages sont égaux alors le système ne donne qu’un seul numéro de page, et non une plage.

Si vous indexez des sous-articles mais sans article principal, comme dans `\index{Jones!programme}` et `\index{Jones!résultats}`, alors la sortie est l’article ‘Jones’ sans virgule ni numéro de page, suivi de deux sous-articles, du genre de ‘programme, 50’ et ‘résultats, 51’.

On génère un article d’index qui dit ‘Voir’ en utilisant un caractère barre verticale : `\index{Ackermann!fonction|see{fonction de P\’eter}}`. On peut également obtenir ‘voir aussi’ avec `seealso`. (Le texte ‘voir’ est défini par `\seename`, et ‘voir aussi’ par `\alsoname` lorsqu’on charge un paquetage d’internationalisation tel que `babel` or `polyglossia`, et que la langue sélectionnée est le français, sinon par défaut c’est ‘see’ et ‘see also’, on peut aussi les redéfinir directement avec quelque chose du genre de `\renewcommand{\seename}{cf.} \renewcommand{\alsoname}{ibid.}`).

La production du mot ‘voir’ fait partie d’une fonctionnalité plus générale. Après la barre verticale vous pouvez mettre le nom *cmd* d’une commande `\cmd` à un argument, comme dans `\index{group|textit}` (notez l’absence de controblisque sur la commande `\textit`) et le système appliquera cette commande au numéro de page dans l’article d’index, donnant ici quelque chose du genre de `\textit{7}`. Vous pouvez aussi définir vos propres commandes comme dans `\newcommand{\pagecours}[1]{\color{blue}#1}` et ensuite `\index{Ackermann!fonction|pagecours}` donnera un numéro de page en bleu (voir Chapitre 21 [Color], page 211). Voici in autre exemple, moins pratique :

```
\newcommand\pageindex[1]{#1, \thepage}
... Epimenides.\index{self-reference|pageindex}
```

qui crée une article citant le numéro de page dans le listing de l’index.

Les deux fonctions décrites plus haut peuvent être combinées, comme dans :

```
\index{Ackermann!fonction|(pagecours)}
...
\index{Ackermann!fonction|)}
```

ce qui en sortie produit un article d'index du genre de 'fonction, 23--27' où la plage de numéros de page est en bleu.

Considérez un article d'index tel que ' $\beta$ -testeur'. Le saisir telle que `\beta-testeur` aura pour effet qu'il sera classé selon le signe dollar. Vous pouvez le saisir en utilisant une arobe, comme dans `\index{bêta-testeur@$\beta-testeur}`. Lorsque vous spécifiez un article avec une arobe qui sépare deux chaînes, `pos@texte`, alors `pos` donne la position alphabétique de l'article alors que `texte` produit le texte de l'article. Un autre exemple est que `\index{Saint Michael's College@SMC}` produit un article d'index 'SMC' classé dans un autre endroit que son épellation donnerait naturellement.

Pour mettre le caractère `!`, `@`, `|` ou `"` dans un article d'index, échappez le en le précédant d'un guillemet anglais double, `"`. (Le guillemet double est ignoré pour le classement alphabétique).

Nombre de paquetages sur le CTAN apportent des fonctionnalités supplémentaires allant au-delà de celles fournies par `makeidx`. L'un deux est `index` qui permet d'avoir des index multiples et contient une commande `\index*{index-entry-string}` qui imprime la *chaîne-article-index* en plus de l'indexer.

La commande `\index` écrit l'information d'indexation dans le fichier `nom-racine.idx`. Plus spécifiquement, elle écrit le texte de la commande `\indexentry{chaîne-article-index}{num-page}`, où `num-page` est la valeur du compteur `\thepage`. Il peut arriver, lorsque la commande `\printindex` perd les pédales, que vous ayez à détruire ce fichier pour repartir de bases saines.

Si vous oubliez l'accolade fermante sur une commande `\index` alors vous obtiendrez un message du genre de :

```
Runaway argument? {Ackermann!function
! Paragraph ended before \@wrindex was complete.
```

### 25.3.3 makeindex

Synopsis, l'un parmi :

```
makeindex nomfic
makeindex -s fichier-style nomfic
makeindex options nomfic0 ...
```

Trie l'information d'index dans le fichier auxiliaire, et fait d'autres traitements dessus. C'est un programme en ligne de commande. Il prend en entrée un ou plusieurs fichiers bruts d'index, les fichiers `nomfic.idx`, et produit en sortie le fichier d'index fini, le fichier `nomfic.ind` qui est l'entrée de `\printindex` (voir Section 25.3.4 [`\printindex`], page 254).

La première forme de la commande suffit pour beaucoup d'usages. La deuxième permet de formater l'index en utilisant un *fichier de style d'index*, un fichier `.isty`. La troisième forme est la plus générale ; voir la documentation complète sur le CTAN.

Ci-dessous on donne un exemple d'un fichier `.isty` simple :

```
% book.isty
% $ makeindex -s book.isty -p odd book.idx
% creates the index as book.ind, starting on an odd page.
preamble
"\pagestyle{empty}
```

```

\\small
\\begin{theindex}
\\thispagestyle{empty}"

postamble
"\n
\\end{theindex}"

```

La présente description ne couvre que certaines des possibilités du *fichier-style*. Pour une liste complète voir la documentation sur le CTAN.

Un fichier de style consiste en une liste de couples : *spécificateur* et *attribut*. Ils peuvent apparaître dans le fichier dans un ordre quelconque. Tous les *attributs* sont des chaînes, sauf mention du contraire. Les chaînes sont entourées de guillemets anglais doubles, " , et la longueur maximale d'une chaîne est de 144 caractères. Le `\n` représente un saut de ligne et le `\t` une tabulation. Les controbliques sont échappées avec une autre controblique, comme cela : `\\`. Si une ligne commence avec un signe pourcent, %, alors c'est un commentaire.

**preamble** Préambule du fichier d'index de sortie. Définit le contexte dans lequel l'index est formaté. Vaut par défaut : `"\\begin{theindex}\n"`.

**postamble** Postambule du fichier d'index de sortie. Vaut par défaut : `"\n\n\\end{theindex}\n"`.

**group\_skip** Traditionnellement les articles d'index sont répartis en groupes, typiquement un groupe pour les articles commençant pas la lettre 'a', etc. Ce spécificateur donne ce qui est inséré quand un nouveau groupe commence. Vaut par défaut : `"\n\n\\indexspace\n"` (`\indexspace` est une commande qui insère une longueur élastique, par défaut 10pt plus5pt minus3pt).

**lethead\_flag** Un entier. Il commande ce qui est inséré pour un nouveau groupe ou une nouvelle lettre. Lorsqu'il vaut 0 (ce qui est le réglage par défaut) alors rien d'autre que `group_skip` n'est inséré avant le groupe. S'il est positif, alors pour une nouvelle lettre les attributs `lethead_prefix` et `lethead_suffix` sont insérés, avec cette lettre en capitale entre eux. S'il est négatif, alors ce qui est inséré est la lettre en bas de casse. Vaut par défaut 0.

**lethead\_prefix** Si un nouveau groupe commence avec une lettre différente alors c'est le préfixe inséré devant l'en-tête présentant la nouvelle lettre. Vaut par défaut : ""

**lethead\_suffix** Si un nouveau groupe commence avec une lettre différente alors c'est le suffixe inséré après l'en-tête présentant la nouvelle lettre. Vaut par défaut : ""

**item\_0** C'est ce qui est mis entre deux articles de niveau 0. Vaut par défaut : `"\n\n\\item "`.

**item\_1** Mis entre deux articles de niveau 1. Vaut par défaut : `"\n\n\\subitem "`.

**item\_2** Mis entre deux articles de niveau 2. Vaut par défaut : `"\n\n\\subsubitem "`.

- `item_01` C'est ce qui est mis entre un article de niveau 0 et un article de niveau 1. Vaut par défaut : `"\n \\\subitem "`.
- `item_x1` C'est ce qui est mis entre un article de niveau 0 et un article de niveau 1 dans le cas où l'article de niveau 0 n'a pas de numéro de page (comme dans `\index{aaa|see{bbb}}`). Vaut par défaut : `"\n \\\subitem "`.
- `item_12` C'est ce qui est mis entre un article de niveau 1 et un article de niveau 2. Vaut par défaut : `"\n \\\subsubitem "`.
- `item_x2` C'est ce qui est mis entre un article de niveau 1 et un article de niveau 2 si l'article de niveau 1 n'a pas de numéro de page. Vaut par défaut : `"\n \\\subsubitem "`.
- `delim_0` Délimiteur mis entre une clef de niveau 0 et son premier numéro de page. Vaut par défaut : une virgule suivie d'un blanc, ", ".
- `delim_1` Délimiteur mis entre une clef de niveau 1 et son premier numéro de page. Vaut par défaut : une virgule suivie d'un blanc, ", ".
- `delim_2` Délimiteur mis entre une clef de niveau 2 et son premier numéro de page. Vaut par défaut : une virgule suivie d'un blanc, ", ".
- `delim_n` Délimiteur mis entre deux numéros de page de la même clef (quel que soit le niveau) . Vaut par défaut : une virgule suivie d'un blanc, ", ".
- `delim_r` Délimiteur mis entre le numéro de début et le numéro de fin d'une plage de numéros de page deux numéros de page. Vaut par défaut : "---".
- `line_max` Un entier. Longueur maximale d'une ligne d'article d'index dans la sortie au delà de quoi la ligne est repliée. Vaut par défaut : 72.
- `indent_space`  
Ce qui est inséré au début d'une ligne repliée. Vaut par défaut : `"\t\t"`.
- `indent_length`  
Un nombre. La longueur d'un renforcement de ligne repliée. La valeur par défaut d'`indent_space` est de deux tabulations, et chaque tabulation compte pour huit espaces de sorte que la valeur par défaut est 16.
- `page_precedence`  
Un document peut avoir différents types de numérotations des pages. Par exemple, un livre peut avoir des matières préliminaires dont les pages sont numérotées en chiffres romains bas de casse alors que les pages de la matière principale sont en chiffres arabes. Cette chaîne spécifie l'ordre dans lequel ils apparaissent dans l'index. La commande `makeindex` prend en charge cinq types différents de numérotation : romain bas de casse `r`, numérique ou arabe `n`, alphabétique bas de casse `a`, romain capital `R`, et alphabétique capital `A`. Vaut par défaut : `"rnaRA"`.

Il y a plusieurs autres programmes faisant le même travail que `makeindex`. L'un d'eux est `xindy` (<https://ctan.org/pkg/xindy>), qui fait de l'internationalisation et peut traiter des documents balisés en utilisant L<sup>A</sup>T<sub>E</sub>X et plusieurs autres langages. Il est écrit en Lisp, hautement configurable, à la fois en termes de balise et en termes d'ordre d'assemblage du texte, comme décrit dans sa documentation.

Un programme d'indexation plus récent prenant en charge l'Unicode est `xindex`, écrit en Lua (<https://ctan.org/pkg/xindex>).

### 25.3.4 `\printindex`

Synopsis :

```
\printindex
```

Place l'index dans la sortie.

Pour obtenir un index vous devez en premier inclure `\usepackage{makeidx}` dans le préambule du document et compiler le document, et ensuite exécuter la commande système `makeindex`, puis de nouveau compiler le document. Voir Section 25.3 [Indexes], page 248, pour plus de détails et un exemple d'usage de `\printindex`.

## 26 Des lettres

Synopsis :

```

\documentclass{letter}
\address{adresse expéditeur}
\signature{nom de l'expéditeur}
\begin{document}
\begin{letter}{adresse destinataire}
\opening{apostrophe}
 corps-de-la-lettre
\closing{texte de salutation}
\end{letter}
... encore des lettres ...
\end{document}

```

Produit une ou plusieurs lettres.

Chaque lettre est dans un environnement séparé `letter`, dont l'argument *adresse destinataire* contient souvent plusieurs lignes séparées par une double controbligue (`\\`). Par exemple, vous pourriez avoir :

```

\begin{letter}{Ninon de l'Enclos \\
 l'hôtel Sagonne}
...
\end{letter}

```

Le début de l'environnement `letter` réinitialise le numéro de page à 1, et le numéro de note en bas de page à 1 aussi.

Les *adresse expéditeur* et *nom de l'expéditeur* sont communs à toutes les lettres, qu'il y en ait une ou plus, ainsi il est préférable de mettre ces déclarations dans le préambule. Comme avec l'adresse du destinataire, souvent *adresse expéditeur* contient plusieurs lignes séparées par une double controbligue `\\`.  $\LaTeX$  place le *nom de l'expéditeur* sous la salutation, après un espace vertical pour la signature manuscrite traditionnelle ; il peut aussi contenir plusieurs lignes.

Le corps de chaque environnement `letter` commence par une commande `\opening` obligatoire telle que `\opening{Madame, Monsieur,}`. Le texte *corps-de-la-lettre* est du code  $\LaTeX$  ordinaire et peut donc tout contenir de la liste énumérée à des formules mathématique en hors texte, à ceci près que la commandes de rubricage telles que `\chapter` qui n'ont pas lieu d'être au sein d'une lettre sont indisponibles. Chaque environnement `letter` se termine typiquement par une commande `\closing` telle que dans `\closing{Je vous prie, Madame, Monsieur, d'agréer l'expression de mes sentiments les meilleurs.}`.

de la matière supplémentaire peut venir après le `\closing`. Vous pouvez préciser qui est en copie de la lettre avec une commande comme `\cc{Le Patron\\Le Patron du Patron}`. Il y a une commande `\encl` similaire pour spécifier une liste de pièces jointes. Et, vous pouvez ajouter un postscriptum avec `\ps`.

Par défaut  $\LaTeX$  renforce la signature et la salutation `\closing` la précédant d'une longueur valant `\longindentation`. Celle-ci vaut par défaut `0.5\textwidth`. Pour les aligner à gauche, il suffit de spécifier `\setlength{\longindentation}{0em}` dans votre préambule.

Pour fixer la date utilisez quelque chose comme `\renewcommand{\today}{2015-10-12}`. En plaçant ce code dans votre préambule, il s'appliquera à toutes les lettres du document.

Cet exemple est avec un seul environnement `letter`. Les trois lignes marquées en commentaire comme optionnelles sont typiquement omises.

```
\documentclass{letter}
\usepackage[T1]{fontenc}
\usepackage{french}
\address{Rue de l'expéditeur \\ Ville de l'expéditeur}
\signature{Nom de l'expéditeur\\ Titre de l'expéditeur}
% optionnel : \location{Boîte Postale 13}
% optionnel : \telephone{(102) 555-0101}
\begin{document}
\begin{letter}{Nom du destinataire \\ Adresse du destinataire}
\opening{Monsieur,}
% optionnel : \thispagestyle{firstpage}
I ne suis pas intéressé à faire affaire avec vous.
\closing{Je vous prie, Monsieur, de recevoir mes salutations distinguées.}
\end{letter}
\end{document}
```

Ces commandes sont utilisées avec la classe `letter`.

## 26.1 `\address`

Synopsis :

```
\address{adresse-expéditeurs}
```

Spécifie l'adresse de retour telle qu'elle apparaît sur la lettre et sur l'enveloppe. Séparez des lignes multiples au sein de `adresse-expéditeurs` avec une double controblque `\\`.

Parce qu'elle peut s'appliquer à des lettres multiples cette déclaration est souvent placée dans le préambule. Toutefois elle peut être faite n'importe où, y compris au sein d'un environnement `letter` individuel.

Cette commande est optionnelle : sans la déclaration `\address` la lettre est formatée avec un espace à blanc en haut de sorte à pouvoir être imprimée sur du papier à en-tête pré-imprimé. (Voir Chapitre 2 [Overview], page 2, pour les détails concernant votre implémentation locale). Avec la déclaration `\address`, elle est formatée comme une lettre personnelle.

En voici un exemple :

```
\address{Stephen Maturin \\
 The Grapes of the Savoy}
```

## 26.2 `\cc`

Synopsis :

```
\cc{premier nom\\...}
```

Produit une liste de noms de personnes mis en copie de la lettre. Cette commande est optionnelle. Si elle est présente, alors elle vient typiquement après le `\closing`. En cas de ligne multiple, séparez les avec une double controblque `\\`, comme dans :

```
\cc{Président \\
Vice-président}
```

### 26.3 \closing

Synopsis :

```
\closing{texte}
```

D'ordinaire à la fin d'une lettre, au dessus de la signature manuscrite, il y a un `\closing` (bien que cette commande soit optionnelle). Par exemple :

```
\closing{Je vous prie, Madame, Monsieur, d'agréer l'expression de ma
confiance.}
```

### 26.4 \encl

Synopsis :

```
\encl{ligne1\\ligne2}
```

Déclare une liste d'une ou plus de pièces jointes.

### 26.5 \location

Synopsis :

```
\location{texte}
```

Le *texte* apparaît centré au bas de chaque page. Il apparaît seulement si le style de la page est `firstpage`.

### 26.6 \makelabels

Synopsis :

```
\makelabels
```

Crée une feuille d'étiquettes d'adresses à partir des adresses des destinataires, une pour chaque lettre. Cette feuille sera sortie avant les lettres, avec dans l'idée de la copier sur une feuille vierge de vignettes autocollantes. Cette commande est à mettre dans le préambule.

Vous pouvez personnaliser les étiquettes en redéfinissant les trois commandes `\startlabels`, `\mlabel` et `\returnaddress` dans le préambule. Ces commandes ne sont pas à appeler directement par l'utilisateur, c'est la classe `letter` qui les appelle. La commande `\startlabels` compose la page sur laquelle les étiquettes sont imprimées en termes de largeur, de hauteur, et de nombre de colonnes, etc. La commande `\mlabel{adresse expéditeur}{adresse destinataire}` est appelée pour chaque lettre du document avec deux arguments :

- Le premier, *adresse expéditeur*, est le contenu courant de la macro `\returnaddress`
- Le second, *adresse destinataire*, est l'adresse du destinataire telle que passée en argument à l'environnement `letter`.

La définition par défaut de `\mlabel` ignore le premier argument, c'est à dire que si vous placez seulement un `\makelabels` dans le préambule, la feuille d'étiquettes ne contiendra des étiquettes que pour les destinataires.

Si vous désirez imprimer une étiquette d'adresse expéditeur et d'adresse destinataire pour chaque lettre, vous devez faire deux choses :

1. La première est de redéfinir `\mlabel` pour fabriquer deux étiquettes au lieu d'une, vous pouvez pour cela mettre ce qui suit dans le préambule :

```
\newcommand*\originalMlabel{}
\let\originalMlabel\mlabel
\def\mlabel#1#2{\originalMlabel{}{#1}\originalMlabel{}{#2}}
```

Ceci ne suffit toutefois pas, parce que la première des deux étiquettes est vide, vu que la définition par défaut de `\returnaddress` l'est.

2. La seconde est de redéfinir `\returnaddress` pour contenir l'adresse de l'expéditeur, vous pouvez faire cela en plaçant dans le préambule :

```
\renewcommand*\returnaddress{\protect\fromname\\protect\fromaddress}
```

## 26.7 `\name`

```
\name{June Davenport}
```

Le nom de l'expéditeur, à imprimer sur l'enveloppe avec l'adresse de retour.

## 26.8 `\opening{texte}`

Synopsis :

```
\opening{texte}
```

Cette commande est obligatoire. Elle commence la lettre, en suivant le `\begin{letter}{...}`. L'argument obligatoire *texte* est le texte d'appel de votre lettre. Par exemple :

```
\opening{Dear John,}
```

## 26.9 `\ps`

Synopsis :

```
\ps{texte}
```

Utilisez la commande `\ps` pour commencer un postscriptum dans une lettre, après `\closing`.

## 26.10 `\signature{texte}`

Synopsis :

```
\signature{première ligne \\
... }
```

Le nom de l'expéditeur. Cette commande est optionnelle, quoique son inclusion soit habituelle.

Le texte de l'argument apparaît à la fin de la lettre, après la clôture et après une espace verticale pour la signature manuscrite traditionnelle. Séparez des lignes multiples avec une double controblique `\\`. Par exemple :

```
\signature{J Fred Muggs \\
}
```

`White House}`

La valeur par défaut de L<sup>A</sup>T<sub>E</sub>X pour l'espace verticale `space` du texte de `\closing` au texte de `\signature` est `6\medskipamount`, ce qui vaut six fois 0,7 em.

Cette commande est d'ordinaire dans le préambule, de sorte à s'appliquer à toutes les lettres du document. Pour qu'elle ne s'applique qu'à une seule lettre, placez la au sein d'un environnement `letter` et avant le `\closing`.

Vous pouvez inclure un graphique dans la signature, par exemple avec `\signature{\vspace{-6\medskipamount}\includegraphics{sig.png}\ My name}` (ceci nécessite d'écrire `\usepackage{graphicx}` dans le préambule).

## 26.11 `\telephone`

Synopsis :

`\telephone{numéro}`

C'est le numéro de téléphone de l'expéditeur. Déclaré typiquement dans le préambule, où il s'applique à toutes les lettres. Ceci apparaît seulement si le style de page (`pagestyle`) `firstpage` est sélectionné. Dans ce cas, il apparaît en bas à droite de la page.

## 27 Entrée/sortie console

### 27.1 `\openin` & `\openout`

Synopsis :

```
\openin numéro=nomfichier
```

ou :

```
\openout numéro=nomfichier
```

Ouvre un fichier pour y lire, ou pour y écrire. Pour la plupart des moteurs, le *numéro* doit être entre 0 et 15, comme dans `\openin3`, dans Lua $\TeX$  *numéro* peut être entre 0 et 127.

Ici  $\TeX$  ouvre le fichier le fichier `presidents.tex` en lecture.

```
\newread\fichierpresidents
\openin\fichierpresidents=presidents
\typeout{fichierpresidents vaut \the\fichierpresidents}
\read\fichierpresidents to\presidentligne
\typeout{\presidentligne}
```

La commande `\newread` alloue des numéros de flux d'entrée de 0 à 15 (il y a aussi une commande `\newwrite`). La variable `\fichierpresidents` est plus facile à mémoriser mais sous le capot ce n'est rien de plus qu'un numéro ; le premier `\typeout` donne quelque chose du genre de '`fichierpresidents vaut 1`'. De plus, `\newread` garde la trace de l'allocation de sorte que si vous en utilisez trop alors vous obtiendrez une erreur du genre de '`! No room for a new \read`'. Le second `\typeout` produit la première ligne du fichier, quelque chose du genre de '`1 Bonaparte, Louis-Napoléon`'.

D'ordinaire  $\TeX$  n'essaie pas d'ouvrir le fichier jusqu'à la prochaine éjection de page. Pour changer cela, utilisez `\immediate\openin numéro=nomfichier` ou `\immediate\openout numéro=nomfichier`.

Fermez les fichiers avec `\closein numéro` et `\closeout numéro`.

La façon dont  $\TeX$  manipule les noms de fichiers varie selon les distributions, et peut même varier d'une version à l'autre d'une même distribution. Si le nom de fichier n'a pas d'extension alors  $\TeX$  y ajoute un `.tex`. L'exemple suivant crée `presidents.tex`, y écrit une ligne, et le ferme.

```
\newwrite\fichierpresidents
\openout\fichierpresidents=presidents
\write\fichierpresidents{1 Bonaparte, Louis-Napoléon}
\closeout\fichierpresidents
```

Mais les noms de fichier avec un point peuvent causer des problèmes : si  $\TeX$  tombe sur un *nomfichier* comme `presidents.dat` il se peut qu'il cherche d'abord `presidents.dat.tex` et ensuite `presidents.dat`, ou bien le contraire. La documentation de votre distribution devrait en dire plus, et si cela fonctionne pour vous tant mieux, mais pour assurer une portabilité complète le mieux est d'utiliser des noms de fichier ne contenant que les vingt-six lettres ASCII (insensible à la casse) et les dix chiffres, ainsi que les tirets de 8 '`_`' et de 6 '`-`', et en particulier sans point ni espace.

Pour `\openin`, si  $\TeX$  ne peut pas trouver le fichier alors il ne produit pas une erreur. Il ne fait que considérer que ce flux n'est pas ouvert (ce qu'on peut vérifier avec `\ifeof` ; on peut également recourir à la commande `\InputIfFileExists`, voir Section 3.3.2 [Class and package commands], page 10). Si vous essayer d'utiliser le même numéro deux fois,  $\LaTeX$  ne produit pas d'erreur. Si vous essayer d'utiliser un mauvais numéro, alors vous obtiendrez un message d'erreur du genre de `! Bad number (16). <to be read again> = 1.30 \openin16=test.jh'`.

## 27.2 `\read`

Synopsis :

```
\read numéro tomacro
```

Fait que la commande *macro* contienne la ligne suivante d'entrée du flux de texte *numéro*, comme dans `\read5 to\data`.

L'exemple suivant ouvre le fichier `mel.tex` en lecture, me le contenu de la première ligne dans la commande `\courriel`, et ensuite ferme le fichier.

```
\newread\fichierreception
\openin\fichierreception=mel
\read\fichierreception to\courriel
\typeout{Adresse mél : \courriel}
\closein\fichierreception
```

Si *numéro* est hors de l'intervalle de 0 à 15 ou si aucun fichier avec ce numéro n'est ouvert, ou si le fichier est à sa fin, alors `\read` prend son entrée à partir du terminal. Notez, cependant, que la méthode naturelle en  $\LaTeX$  de prendre en entrée à partir du terminal est `\typein` (voir Section 27.3 [`\typein`], page 261).

Pour lire un fichier entier en une fois utilisez `\input` (voir Section 24.3 [`\input`], page 241) ou `\include` (voir Section 24.2 [`\include` & `\includeonly`], page 239).

Une raison usuelle de vouloir lire d'un fichier de données est de faire du publipostage. CTAN a plusieurs paquetage pour cela ; parmi lesquels `datatool`.

## 27.3 `\typein[cmd]{msg}`

Synopsis :

```
\typein[\cmd]{msg}
```

`\typein` tape *msg* sur la console et fait que  $\LaTeX$  s'arrête et attend que vous saisissez une ligne d'entrée, en la terminant par un retour chariot. Si l'argument optionnel *\cmd* est omis, l'entrée tapée est traitée comme si elle avec été incluse dans le fichier d'entrée à la place de la commande `\typein`. Si l'argument *\cmd* est présent, cela doit être un nom de commande. Ce nom de commande est alors défini, ou redéfini, pour se développer en le texte en entrée qui a été tapé.

## 27.4 `\typeout{msg}`

Synopsis :

```
\typeout{msg}
```

Tape `msg` sur le terminal et dans le journal (fichier `log`). Les commandes dans `msg` qui sont définies avec `\newcommand` ou `\renewcommand` (entre autres) sont remplacées par leurs définitions avant d'être tapées.

Les règles usuelles de  $\text{\LaTeX}$  pour le traitement de plusieurs espaces comme un seul espace et pour ignorer les espaces après un nom de commande s'appliquent à `msg`. Une commande `\space` dans `msg` provoque un unique espace tapé, indépendant des espaces qui l'entourent. Un `^^J` dans `msg` tape un saut à la ligne.

## 27.5 `\write`

Synopsis :

```
\write numéro{chaîne}
```

Écrit *chaîne* vers le fichier journal (`.log`), vers le terminal, ou vers un fichier ouvert par `\openout`. Par exemple, `\write6` écrit vers le flux textuel numéro 6

Si le code suivant apparaît dans le fichier `fichierbase.tex` alors cela ouvre le fichier `fichierbase.jh`, y écrit 'Bonjour tout le monde !' et un saut le ligne, et ferme ce fichier.

```
\newwrite\monfichier
\immediate\openout\monfichier=\jobname.jh % \jobname est le nom
 % de base du fichier racine
...
\immediate\write\monfichier{Bonjour tout le monde !}
...
\immediate\closeout\monfichier
```

Le `\newwrite` alloue un numéro de flux, en lui donnant un nom symbolique pour se faciliter la vie, de sorte que `flux \newwrite\monfichier\the\monfichier` produit quelque chose du genre de 'flux 3'. Ensuite `\openout` associe le numéro de flux avec le nom de fichier fourni. Et puis à la fin  $\text{\TeX}$  exécute `\write3` ce qui place la chaîne dans le fichier.

Typiquement *numéro* est compris entre 0 et 15, parce que typiquement les auteurs  $\text{\LaTeX}$  suivent l'exemple précédent et le numéro est alloué par le système. Si *numéro* est hors de l'intervalle de 0 à 15 ou si il n'est pas associé à un fichier ouvert alors  $\text{\LaTeX}$  écrit *chaîne* vers le fichier journal. Si *numéro* est positif alors en plus  $\text{\LaTeX}$  écrit *chaîne* vers le terminal. Ainsi, `test \write-1{Bonjour tout le monde !}` place 'Bonjour tout le monde !' suivi d'un saut de ligne dans le fichier journal (c'est ce que la commande `\wlog` fait ; voir Section 27.5.3 [`\wlog`], page 265). Et `\write100{Bonjour tout le monde !}` met la même chose dans le fichier journal mais aussi met 'Bonjour tout le monde !' suivi d'un saut de ligne dans la sortie terminal. (Mais 16, 17, et 18 sont des valeurs spéciales de *numéro* ; voir ci-dessous.)

Dans  $\text{\LuaTeX}$ , au lieu de 16 flux de sortie il y en a 256 (voir Section 2.3 [ $\text{\TeX}$  engines], page 3).

Utilisez `\write\@auxout{chaîne}` pour écrire vers le fichier `.aux` courant qui est associé soit au fichier racine soit au fichier inclus courant ; et utilisez `\write\@mainaux{chaîne}` pour écrire le fichier `.aux` principal. Ces noms symboliques sont définis par  $\text{\LaTeX}$ .

Par défaut  $\text{\LaTeX}$  n'écrit pas *chaîne* vers le fichier immédiatement. C'est ainsi parce que, par exemple, vous pourriez avoir besoin de `\write` pour sauvegarder le numéro de page courant, mais au moment où  $\text{\TeX}$  rencontre une commande `\write` typiquement il ne

connaît pas quel est le numéro de page, puisque il n'a pas encore découpé la sortie en pages. Ainsi, vous utilisez `\write` dans l'un des trois contextes suivants :

```
\immediate\write\@auxout{chaîne} %1
\write\@auxout{chaîne} %2
\protected@write\@auxout{}{chaîne} %3
```

1. Avec le premier,  $\LaTeX$  écrit *chaîne* vers le fichier immédiatement. Toutes les macros dans *chaîne* sont développées à fond (exactement comme dans `\edef` de sorte que pour empêcher le développement vous devez utiliser `\noexpand`, un `\toks`, etc. mais avec la différence que vous devez utiliser `#` au lieu de `##`).
2. Avec le deuxième, *chaîne* est stockée dans la liste courante des choses (comme un élément « extraordinaire » de  $\TeX$ ) et conservé jusqu'à ce que la page soit éjectée et de la même manière les macros ne sont pas développées jusqu'à l'éjection (`\shipout`). Au moment du `\shipout`, *chaîne* est développée à fond.
3. Le troisième, `\protected@write`, est semblable au deuxième sauf que vous pouvez utiliser `\protect` pour éviter le développement. Le premier argument qui est en plus vous permet d'insérer localement des définitions supplémentaires pour que plus de macros soient protégées ou pour avoir d'autres définitions spéciales pendant l'écriture.

Voici un exemple simple de développement avec `\write` dans lequel *chaîne* contient une séquence de contrôle `\triplex` que l'on a définie comme le texte 'XYZ' :

```
\newwrite\fichierjh
\openout\fichierjh=test.jh
\newcommand{\triplex}{XYZ}
\write\fichierjh{test \triplex test}
```

Ceci résulte en ce que le fichier `test.jh` contienne le texte 'test XYZtest' suivi d'un saut de ligne.

Les cas où *numéro* vaut 16, 17, ou 18 sont spéciaux. À cause du comportement de `\write` quand *numéro* est hors de l'intervalle de 0 à 15 qu'on a déjà décrit plus haut, dans Plain  $\TeX$  `\write16` et `\write17` étaient parfois utilisés pour écrire dans le fichier journal et le terminal ; toutefois, dans  $\LaTeX$ , la façon naturelle de faire ceci est d'utiliser `\typeout` (voir Section 27.4 [`\typeout`], page 261). La commande `\write18` est encore plus spéciale ; les systèmes  $\TeX$  modernes l'utilisent pour passer des commandes au système d'exploitation (voir Section 27.5.4 [`\write18`], page 265).

D'ordinaire `\write` produit en sortie une unique ligne. On peut y insérer un saut de ligne avec `^^J`. Ainsi, l'exemple suivant produit deux lignes dans le fichier journal :

```
\wlog{Les lignes parallèles ont beaucoup en commun.^^J
Mais elles ne se rencontrent jamais.}
```

Un cas usuel dans lequel des auteurs tentent d'écrire dans leur propre fichier est pour les réponses à des exercices, ou dans quelque autre situation où vous désirez écrire en sortie verbatim, sans développer les macros. CTAN a plusieurs paquetages pour cela ; parmi lesquels `answers`.

### 27.5.1 `\write` et la sécurité

La possibilité d'écrire dans des fichiers soulève des problèmes de sécurité. Si vous compilez un fichier  $\LaTeX$  téléchargé et qu'il écrase votre fichier de mots de passe alors vous serez à juste titre embêté.

Ainsi par défaut les systèmes T<sub>E</sub>X ne vous autorisent à ouvrir en écriture que les fichiers qui sont dans le répertoire courant ou dans un sous-répertoire. L'exemple suivant

```
\newwrite\fichierjh
\openout\fichierjh=./test.jh
```

produit une erreur du genre de :

```
Not writing to ../test.jh (openout_any = p).
! I can't write on file '../test.jh'
```

Il est possible d'avoir une telle erreur lorsqu'on essaie d'utiliser une commande telle que `\include{./nomfichier}` puisque L<sup>A</sup>T<sub>E</sub>X va essayer d'ouvrir `./nomfichier.aux`. La solution la plus simple est de placer les fichiers inclus dans le même répertoire que le fichier racine, ou dans des sous-répertoires.

### 27.5.2 \message

Synopsis :

```
\message{chaîne}
```

Écrit *chaîne* vers le fichier journal en `.log` et le terminal.

Typiquement, les auteurs L<sup>A</sup>T<sub>E</sub>X utilisent `\typeout` (voir Section 27.4 [`\typeout`], page 261) qui permet d'utiliser `\protect` sur toute commande fragile au sein de *chaîne* (voir Section 12.11 [`\protect`], page 134). Mais la commande `\typeout` insère un saut de ligne à la fin de *chaîne* contrairement à `\message`, aussi cette dernière peut être utile.

Dans l'exemple ci-dessous dans le corps du document

```
avant\message{Un Deux}\message{Trois}\message{Quatre^^JLes}
\message{pouces vont se battre.}Après
```

dans certaines circonstances (voir ci-dessous) L<sup>A</sup>T<sub>E</sub>X écrit ce qui suit à la fois vers le terminal et le fichier journal.

```
Un Deux Trois Quatre
Les pouces vont se battre.
```

Un ^^J produit un saut de ligne. Aussi, dans le document en sortie, on a entre 'avant' et 'Après' un seul espace (issu de la fin de ligne suivant 'Les').

Alors que `\message` vous permet de mieux contrôler le formatage, un truc à piger c'est que L<sup>A</sup>T<sub>E</sub>X peut très bien bousiller ce formatage là en insérant des sauts de ligne selon ce qu'il a déjà sorti. Comparez ce corps de document, où le 'Deux' a été déplacé, à celui donné en exemple plus haut.

```
avant\message{Un}\message{Deux Trois}\message{Quatre^^JLes}
\message{pouces vont se battre.}Après
```

La chose suivante peut se produire : lorsque L<sup>A</sup>T<sub>E</sub>X produit les messages en sortie vers le terminal, le message avec 'Un' désormais plus court tient à la fin de la ligne de terminal, et du coup L<sup>A</sup>T<sub>E</sub>X saute de ligne juste après lui et avant le 'Deux Trois'. Ce saut de ligne apparaît également dans le fichier journal. Cette insertion de saut de ligne peut dépendre de, par exemple, la longueur des noms complet des fichiers inclus. C'est pourquoi il est difficile de produire des lignes formatées précisément et d'une façon qui soit portable, et de même de demander que votre message commence au début d'une ligne.

### 27.5.3 `\wlog`

Synopsis :

```
\wlog{chaîne}
```

Écrit *chaîne* vers le fichier journal `.log`.

```
\wlog{Pour tout n différent de 0, et tout x, (sin x)/n vaut six.}
```

```
\wlog{Ben oui, il suffit de simplifier par n.}
```

D'ordinaire *chaîne* apparaît sur une seule ligne séparée. Utilisez `^^J` pour insérer un saut de ligne.

```
\wlog{C'est Helvetica qui téléphone à Times Roman :}
```

```
\wlog{Change de ligne !^^JJe t'entends très mal.}
```

### 27.5.4 `\write18`

Synopsis :

```
\write18{commande_shell}
```

Émet une commande vers l'interpréteur du système d'exploitation. Le système d'exploitation exécute la commande et l'exécution de  $\text{\LaTeX}$  est bloquée jusqu'à ce que cela soit fini.

La séquence suivante (sur Unix)

```
\usepackage{graphicx} % dans le préambule
```

```
...
```

```
\newcommand{\numerofig}{1}
```

```
\immediate\write18{cd pix && asy figure\numerofig}
```

```
\includegraphics{pix/figure\numerofig.pdf}
```

exécute Asymptote (le programme `asy`) sur le fichier `pix/figure1.asy`, de sorte que le document peut par la suite lire le graphique en résultant (voir Section 22.3.1 [`\includegraphics`], page 220). Comme pour tout `\write`, ici  $\text{\LaTeX}$  développe les macros contenues dans *commande\_shell* de sorte que `\numerofig` est remplacé par '1'.

Un autre exemple est que vous pouvez exécuter automatiquement  $\text{BibTeX}$  au début de chaque exécution de  $\text{\LaTeX}$  (voir Section 8.24.4 [Using BibTeX], page 105) en incluant `\immediate\write18{bibtex8 \jobname}` comme première ligne du fichier. Notez que `\jobname` se développe par défaut en le nom de base du fichier racine, ou bien l'argument de l'option `--jobname` si celle-ci est passée sur la ligne de commande.

Il peut arriver que vous ayez besoin de plusieurs étapes pour obtenir l'information désirée. L'exemple suivant insère en entrée une liste de tous les fichiers PDF du répertoire courant (voir aussi `texosquery` plus bas) :

```
\immediate\write18{ls *.pdf > tmp.dat}
```

```
\input{tmp.dat}
```

Le comportement standard de tout `\write` est d'attendre jusqu'à la prochaine éjection de page avant de développer les macros ou d'écrire dans le flux (voir Section 27.5 [`\write`], page 262). Mais il se peut que vous désiriez que cela soit immédiat. Pour cela, utilisez `\immediate\write18{commande_shell}`.

Autoriser l'usage de commandes système au sein d'un fichier  $\text{\LaTeX}$  pose des problèmes de sécurité évidents. Si vous télécharger un fichier depuis la Toile et qu'il contient des

commandes pour détruire tous vos fichiers, cela vous rendrait bien malheureux. Le réglage standard dans les distributions modernes est d'interdire l'accès complet à l'interpréteur de commande du système d'exploitation. Vous pouvez l'autoriser, si vous êtes sûr qu'il n'y a pas de danger, en compilant avec `latex --enable-write18 nomfichier` (voir Section 28.1 [Command line options], page 267). (L'option `--shell-escape` est un synonyme dans  $\text{\TeX}$  Live.)

Au lieu de l'accès complet, les distributions modernes par défaut utilisent une version restreinte qui permet à certaines commandes de fonctionner, telles que celles qui font tourner Metafont pour générer les polices manquantes, même si vous n'utilisez pas l'option `enable-write18`. Par défaut cette liste de commandes autorisées est courte et ne fait figurer que les commandes qui sont sous contrôle des mainteneurs de la distribution (voir Section 28.1 [Command line options], page 267).

Certains paquets ont besoin d'exécuter des commandes système externes. Par exemple `sagetex` vous permet de faire faire des calculs ou tracer des graphiques au logiciel de mathématiques *Sage*, puis d'incorporer ces sorties dans votre document. Pour cela vous devez exécuter  $\text{\LaTeX}$  avec la fonction `\write18` pleinement débridée.

Le texte `commande-shell` est toujours passé à `/bin/sh` sur les systèmes d'exploitation unixiens, et à l'interpréteur de commande DOS `cmd.exe` sur Windows.  $\text{\LaTeX}$  ignore tout réglage d'un autre interpréteur de commande par l'utilisateur, ainsi que la variable d'environnement `SHELL`.

Si votre besoin c'est d'avoir de l'information sur le système, comme le nom du système d'exploitation, celui de la locale, ou le contenu d'un répertoire, alors jeter un coup d'œil au paquetage `texosquery` qui fournit une interface commode et sûre pour cela, contrairement aux exemples donnés plus haut utilisant `\write18` à l'état brut : <https://ctan.org/pkg/texosquery>.

## 28 Interface ligne de commande

Synopsis (depuis la ligne de commande d'un terminal) :

```
pdflatex options argument
```

Exécute L<sup>A</sup>T<sub>E</sub>X sur *argument*. À la place de `pdflatex` vous pouvez aussi utiliser (pour une sortie PDF) `xelatex`, ou `lualatex`, ou (pour une sortie DVI) `latex` ou `dvilualatex`, entre autres (voir Section 2.3 [T<sub>E</sub>X engines], page 3).

L'exemple suivant exécute L<sup>A</sup>T<sub>E</sub>X sur le fichier `these.tex` produisant en sortie `these.pdf`.

```
pdflatex these
```

Notez que `.tex` est l'extension par défaut des noms de fichier.

pdfT<sub>E</sub>X est une évolution du programme original T<sub>E</sub>X, comme le sont XeT<sub>E</sub>X et LuaT<sub>E</sub>X (voir Section 2.3 [T<sub>E</sub>X engines], page 3). Ils sont complètement rétrocompatibles. Mais le programme original avait son propre format de sortie, DVI, alors que les plus récents peuvent produire directement un PDF en sortie. Ceci leur permet de profiter des fonctions supplémentaires des PDF tels que les hyperliens, la prise en charge de formats modernes d'images tels que JPG et PNG, et les programmes de visionnage que l'on trouve de partout. En bref, si vous exécutez `pdflatex` ou `xelatex` ou `lualatex` alors par défaut vous obtenez un PDF et avez accès à toutes ses fonctions modernes. Si vous exécutez `latex`, ou `dvilualatex`, alors vous obtenez un DVI. La description faite ici suppose `pdflatex`.

Voir Section 28.1 [Command line options], page 267, pour une sélection des options de ligne de commande les plus utiles. Quant à *argument*, le cas habituel est qu'il ne commence pas par une controblaque, dans ce cas le système le prend pour un nom de fichier, et compile ce fichier. Si *argument* commence avec une controblaque, alors le système l'interprète comme une ligne d'entrée à L<sup>A</sup>T<sub>E</sub>X, ce qui peut être utilisé pour obtenir certains effets spéciaux (voir Section 28.2 [Command line input], page 269).

Si vous ne donnez aucun arguments ni options alors `pdflatex` invite à saisir à partir du terminal. Vous pouvez vous échapper de cela en entrant `CTRL-D`.

Si L<sup>A</sup>T<sub>E</sub>X trouve une erreur dans votre document alors par défaut il s'arrête et vous interroge à ce propos. Voir Section 28.4 [Recovering from errors], page 273, pour un aperçu de quoi faire.

### 28.1 Options passées à la ligne de commande

On décrit ici les options de ligne de commande qui sont pertinentes pour rédaction ordinaire de documents. Pour une liste complète, essayez d'exécuter `'latex --help'` depuis la ligne de commande.

Avec nombre d'implémentations on peut spécifier les options de ligne de commande avec un préfixe `'-'` ou `'--'`. C'est le cas à la fois pour T<sub>E</sub>X Live (y compris MacT<sub>E</sub>X) et MiK<sub>T</sub>E<sub>X</sub>. Ici on utilise les deux conventions indistinctement. Si une option prend une valeur, celle-ci peut être spécifiée soit comme un argument séparé (`'--truc bidule'`), ou comme un argument unique avec un signe = (`'--truc=bidule'`), à condition qu'il n'y ait aucun espace autour du =. On utilisera en général la syntaxe en =.

**-version** Affiche la version courante, comme ‘pdfTeX 3.14159265-2.6-1.40.16 (TeX Live 2015/Debian)’ ainsi qu’une petite quantité d’information supplémentaire, et se termine.

**-help** Produit un message bref d’utilisation servant d’aide-mémoire et se termine.

**-interaction=*mode***

zTeX compile un document en l’un parmi quatre modes possibles d’interaction : **batchmode**, **nonstopmode**, **scrollmode**, **errorstopmode**. Dans le mode *errorstopmode* (le mode par défaut), TeX s’arrête à chaque erreur et demande une intervention de l’utilisateur. Dans le mode *batchmode* il n’imprime rien sur le terminal, on défile sur les erreurs comme si l’utilisateur appuyait *RETURN* à chaque erreur, et les fichiers manquants entraînent la terminaison de la tâche. Dans le mode *nonstopmode*, les messages de diagnostic apparaissent sur le terminal mais comme dans le mode *batchmode* il n’y a pas d’interaction avec l’utilisateur. Dans le mode *scrollmode*, TeX s’arrête en cas de fichiers manquants ou d’entrée clavier, mais pour nulle autre raison.

Par exemple, en démarrant L<sup>A</sup>T<sub>E</sub>X avec cette ligne de commande :

```
pdflatex -interaction=batchmode nomfichier
```

on élimine la plupart des sorties terminal.

**-jobname=*chaîne***

Règle la valeur du *nom d’ouvrage* de TeX à la chaîne de caractères *chaîne*. Le fichier journal et le fichier de sortie s’appelleront *chaîne.log* et *chaîne.pdf*. voir Section 28.3 [Jobname], page 270.

**-output-directory=*répertoire***

Écrit les fichiers dans le répertoire *répertoire*. Il doit déjà exister. Ceci s’applique à tous les fichiers externes créés par TeX ou L<sup>A</sup>T<sub>E</sub>X, tels que le fichier *.log* de l’exécution, les fichiers *.aux*, *.toc*, etc., créés par L<sup>A</sup>T<sub>E</sub>X, de même que le fichier de sortie principal *.pdf* ou *.dvi* lui-même.

Lorsqu’il est spécifié, le répertoire de sortie *répertoire* est aussi vérifié en premier automatiquement pour tout fichier lu en entrée, de sorte que les fichiers externes peuvent être relus en entrée, si désiré. Le vrai répertoire courant (dans lequel L<sup>A</sup>T<sub>E</sub>X été lancé) demeure inchangé, et est également vérifié pour trouver les fichiers en entrée.

**--enable-write18**

**--disable-write18**

**--shell-escape**

**--no-shell-escape**

Activation et désactivation `\write18{commande_shell}` (voir Section 27.5.4 [`\write18`], page 265). Les deux premières options sont prises en charge à la fois par TeX Live et MiKTeX, alors que les deux suivantes sont des synonymes pris en charge par TeX Live.

Activer cette fonction a des implications lourdes de sécurité, puisque cela permet à un fichier L<sup>A</sup>T<sub>E</sub>X d’exécuter n’importe quelle commande. Ainsi, par défaut, le `\write18` non restreint n’est pas autorisé. (Le réglage par défaut de TeX

Live, MacTeX, et MiKTeX est d'autoriser l'exécution d'un nombre limité de programmes en relation avec TeX, et qu'elles distribuent).

Par exemple, si vous invoquez L<sup>A</sup>T<sub>E</sub>X avec l'option `no-shell-escape`, et que dans votre document vous appelez `\write18{ls -l}`, alors vous n'obtenez pas d'erreur mais votre fichier journal dit `'runsystem(ls -l)...disabled'`.

`-halt-on-error`

Arrête le traitement sur la première erreur.

`-file-line-error`

`-no-file-line-error`

Active ou désactive le style `nomfichier:numéroligne:erreur` des messages d'erreur. Ces options ne sont disponibles qu'avec TeX Live ou MacTeX.

## 28.2 Spécifier du code L<sup>A</sup>T<sub>E</sub>X sur la ligne de commande

Au sein de l'invocation de ligne de commande

```
pdflatex options argument
```

vous pouvez spécifier du code L<sup>A</sup>T<sub>E</sub>X arbitraire en entrée en commençant *argument* par une contrôblique (ce qui est pris en charge par tous les moteurs). Ceci permet de faire certains effets spéciaux.

Par exemple, ce fichier (qui utilise le paquetage `hyperref` pour les hyperliens) peut produire deux types de sortie, l'un pour une consultation sur document papier et l'autre pour une consultation sur écran.

```
\ifdefined\paperversion % dans le préambule
\newcommand{\urlcolor}{black}
\else
\newcommand{\urlcolor}{blue}
\fi
\usepackage[colorlinks=true,urlcolor=\urlcolor]{hyperref}
...
\href{https://www.ctan.org}{CTAN} % dans le corps
...
```

Lorsqu'on compile ce document `book.tex` avec la ligne de commande `pdflatex book` le lien 'CTAN' sera en bleu. Mais si on le compile avec

```
pdflatex "\def\paperversion{}\input book.tex"
```

alors le lien sera en noir. On utilise des guillemets anglais double pour empêcher l'interprétation des symboles par l'interpréteur de ligne de commande (désigné également par *shell*). (Ceci fonctionne généralement à la fois sur les systèmes Unix et Windows, mais il y a nombre de particularités dans l'usage des guillemets par les shells, lisez donc la documentation de votre système si besoin est).

De même, à partir d'un seul fichier `principal.tex` vous pouvez compiler deux versions distinctes.

```
pdflatex -jobname=etudiants "\def\etudiants{}\input{principal}"
pdflatex -jobname=professeurs "\def\profs{}\input{principal}"
```

L'option `jobname` est là parce que sinon les deux fichiers en sortie s'appelleraient `principal.pdf` et le second écraserait le premier. (voir Section 28.3 [Jobname], page 270).

Dans cet exemple on passe l'option `draft` au paquetage `graphicx` :

```
pdflatex "\PassOptionsToPackage{draft}{graphicx}\input{aa.tex}"
```

de sorte que les fichiers graphiques ne sont lus que pour l'information de dimension de boîte englobante, mais remplacés dans le PDF par une boîte de même taille ne contenant que le nom de fichier, ce qui accélère le temps de compilation, et économise l'encre d'imprimante.

Un autre exemple classique est de cribler certains chapitres à la compilation en les passant à `\includeonly` dans la ligne de commande. Ainsi, supposons qu'on a un fichier racine `livre.tex` ainsi formé :

```
\documentclass{book}
\begin{document}
\frontmatter
\include{pref}
\mainmatter
\include{chap1}
\include{chap2}
\appendix
\include{append}
\backmatter
\include{bib}
\end{document}
```

En compilant `livre.tex` avec la ligne de commande suivante :

```
pdflatex '\includeonly{chap1}\input{livre}
```

seul le chapitre contenu dans `chap1.tex` sera produit en sortie. Voir Chapitre 24 [Splitting the input], page 238.

Dans ce dernier exemple, on force la date au 14 juillet 1789 :

```
pdflatex '\day=14 \month=7 \year=1789 \input revolution'
```

Ainsi, si le fichier `revolution.tex` contient ceci :

```
\documentclass[french]{minimal}
\usepackage{babel}
\begin{document}
\today\ c'est la révolution !
\end{document}
```

alors vous aurez en sortie un truc du genre :

```
14 juillet 1789 c'est la révolution !
```

### 28.3 Nom d'ouvrage de T<sub>E</sub>X

Exécuter L<sup>A</sup>T<sub>E</sub>X crée plusieurs fichiers, dont la sortie principale en PDF (ou DVI) et bien d'autres. Ces fichiers sont nommés selon ce qu'on appelle le *nom d'ouvrage*. Le cas le plus fréquent est aussi le plus simple : la commande `pdflatex these` crée `these.pdf` et aussi `these.log` et `these.aux`. Ici le nom d'ouvrage est `these`.

En général,  $\LaTeX$  est invoqué comme *moteur-latex options arguments*, où *moteur-latex* est `pdflatex`, `lualatex`, etc. (voir Section 2.3 [TeX engines], page 3). Si *arguments* ne commence pas par une controblique, comme c'est le cas ci-dessus avec `these`, alors  $\TeX$  insère un argument `\input` en tête de la liste d'arguments *arguments* et considère le texte formé en joignant les éléments de cette liste séparés par des blancs comme un code source pris en entrée. Ainsi la ligne de commande `pdflatex these` conduit  $\LaTeX$  à compiler le code source suivant :

```
\input these
```

Ceci permet de faire une indirection vers le fichier `these.tex` qui est censé contenir l'entièreté du code source à compiler, et qu'on appelle à ce titre *fichier racine* (voir Chapitre 24 [Splitting the input], page 238).

On appelle *nom racine* le nom du fichier racine, sans l'extension `.tex` s'il en a une. Voir Section 24.3 [`\input`], page 241, pour la gestion de l'extension `.tex`.

Notez bien, que si on a toujours un nom d'ouvrage, on n'a pas systématiquement de nom racine. En effet, dès lors qu'on a du code  $\LaTeX$  passé sur la ligne de commande (voir Section 28.2 [Command line input], page 269), l'entièreté du code compilé n'est pas issu d'un seul fichier. En supposant ici et ci-après que le shell est `bash`, et donc que les guillemets anglais simples ou doubles sont interprétés par le shell (voir (info "(bash) Quoting")), considérez par exemple ligne de commande suivante :

```
pdflatex '\documentclass{article}\input' corps
```

Dans l'exemple précédent, on ne peut pas considérer `corps.tex` comme un fichier racine, puisqu'il ne peut pas être compilé de façon autonome vu qu'il ne contient pas la commande `\documentclass`. Par abus de langage on continue toutefois de parler de fichier racine si le code  $\LaTeX$  passé en ligne de commande ne sert qu'à configurer la façon dont le fichier est compilé, comme dans :

```
pdflatex '\PassOptionsToClass{printout}{beamer}\input' mapresentation
```

où `mapresentation.tex` contient une présentation `beamer`, et on veut juste la compiler de façon adéquate pour une impression plutôt que pour une projection.

Voici par contre un exemple extrême où on n'a pas de fichier racine, considérez la ligne de commande :

```
pdflatex fic "Bonjour tout le monde~!" '\end{document}'
```

ce qui conduit `pdflatex` à compiler le code source suivant :

```
\input fic Bonjour tout le monde~! \end{document}
```

À supposer que `fic.tex` soit un fichier contenant `\documentclass{minimal}\begin{document}` le résultat sera un document PDF nommé `fic.pdf` et contenant le texte 'Bonjour tout le monde !'.

Dans la mesure où on n'a pas systématiquement de fichier racine, la règle que  $\TeX$  utilise pour déterminer implicitement le nom d'ouvrage, c.-à-d. lorsqu'il n'est pas spécifié explicitement par l'option `-jobname` (voir Section 28.1 [Command line options], page 267), est que le nom d'ouvrage soit le nom du fichier correspondant à l'argument passé au premier `\input` rencontré durant la compilation, débarrassé de l'extension `.tex` s'il en a une. Dans l'exemple précédent, le nom d'ouvrage est donc bien `fic`. Voir aussi plus bas la primitive `\jobname` pour une dérogation à cette règle.

Ce qui précède vaut également si  $\TeX$  est en mode interactif, la seule différence est alors que *arguments* n'est pas passé entièrement sur la ligne de commande, mais saisi interactivement par l'utilisateur. Par exemple si dans le shell vous tapez la ligne de commande `pdflatex`, puis qu'à l'invite de  $\TeX$  vous tapez `these`, vous aurez le même résultat que dans le premier exemple donné, à savoir si vous aviez tapé directement `pdflatex these` depuis le shell.

Dans la même veine, la seule différence si *arguments* commence avec une controblaque est que le `\input` n'est pas inséré, et donc `pdflatex "\input" these` est sensiblement équivalent à `pdflatex these`.

Il y a deux autres possibilités pour le nom d'ouvrage. Pour ce qui est de la première, il peut être directement spécifié avec l'option `-jobname`, comme dans `pdflatex -jobname=monnom` (voir Section 28.2 [Command line input], page 269, pour un véritable exemple).

La seconde possibilité est `texput`, c'est la valeur par défaut de repli si aucun autre nom n'est disponible pour  $\TeX$ . Par exemple si aucune option `-jobname` n'a été spécifiée et que la compilation s'arrête avant qu'une commande `\input` ne soit trouvée, alors le fichier journal s'appelle `texput.log`.

Au sein du document, la primitive `\jobname` se comporte comme une macro qui se développe en le nom d'ouvrage. Quand le nom d'ouvrage contient au moins un blanc, un guillemet anglais est ajouté et début et fin de la chaîne renvoyée par `\jobname` et tout blanc est remplacé par un espace, c.-à-d. qu'un caractère tabulation est remplacé par un espace. De plus, au cours du développement, tous les caractères sont de catcode 12 (other), y compris les lettres qui normalement sont de catcode 11, mais à l'exception des espaces auquel un catcode 10 est attribué.

À cause de cette situation des catcodes, utiliser le nom d'ouvrage dans une expression conditionnelle peut devenir compliqué. Une solution est d'utiliser la macro `\IfBeginWith` du paquetage `xstring` dans sa forme étoilée pour être insensible au catcode. Dans le texte suivant par exemple, la note en bas de page « dont la Republica Bananensis Francorum. » n'est présente que si le nom d'ouvrage commence par `moi-meme`.

```
Si une démocratie est juste un régime où les citoyens votent alors
toutes les républiques bananières\IfBeginWith*{\jobname}{moi-meme}%
{\footnote{dont la Republica Bananensis Francorum.}}{} sont des
démocraties.
```

Redéfinir `\jobname` dans le code source avec un `\renewcommand` peut s'avérer vain et risqué. Tout d'abord, si c'était ce qui vous cherchiez, cela ne changera pas le nom d'ouvrage au sens où les fichiers créés directement par le moteur continuent à utiliser le nom d'ouvrage original : il s'agit notamment du fichier de sortie principale `.pdf` ou `.dvi`, du journal de compilation `.log` et du fichier de liste fichiers `.fls` si l'option `-recorder` a été utilisée. Par contre les fichiers auxiliaires `.aux`, `.bib`, etc. pourront être renommés car leur nom est produit par le noyau  $\LaTeX$  en développant `\jobname` et non directement par le moteur, mais à condition bien sûr que la redéfinition soit faite avant l'ouverture des fichiers en question. Le paquetage `multibib` utilise ce truc pour gérer plusieurs fichiers de référence bibliographique.

La séquence de contrôle `\jobname` se comporte comme une macro, au sens où on peut faire dessus toutes les opérations qu'on ferait sur une macro, et notamment la

développer, mais ce n'est pas une macro, c'est une primitive  $\TeX$ . Ainsi, une seconde règle de détermination du nom d'ouvrage est que le premier développement de `\jobname` fige le nom d'ouvrage. Dans l'exemple suivant de ligne de commande :

```
pdflatex '\typeout{\jobname}\input' these
```

le nom d'ouvrage sera la valeur par défaut `texput`, et non `these`. En effet le code  $\LaTeX$  `\typeout{\jobname}` a pour effet que `\jobname` est développé avant que le premier `\input` ne soit rencontré, et donc on déroge à la première règle énoncée plus haut, c.-à-d. que dans ce cas le nom d'ouvrage n'est pas donné par le premier `\input`.

Cette seconde règle a pour autre conséquence que dans l'exemple suivant de ligne de commande :

```
pdflatex '\documentclass{article}\input' corps
```

le nom d'ouvrage sera également `texput`, en effet en sous-main les commandes `\documentclass` et `\RequirePackage` développent `\jobname`, et donc la seconde règle déroge à la première règle, celle du premier `\input` rencontré.

Il est à noter que ce qui précède n'était pas vrai avec de vieilles versions de  $\LaTeX$  (approximativement) antérieures à 2020. Avec ces vieilles versions `\documentclass` et `\RequirePackage` faisaient un `\input` sans développement préalable de `\jobname` et donc dans l'exemple précédent le nom d'ouvrage aurait été `article`, ce qui était déroutant.

Par ailleurs, l'équivalence entre `pdflatex "\input" these` et `pdflatex these` peut être mise en défaut lorsque le nom du fichier comprend un blanc. Comparer par exemple les deux lignes de commande suivantes :

```
pdflatex '\input' 'to\space\space to'
pdflatex 'to\space\space to'
```

En remplaçant les espaces par des espaces visibles `␣` pour la clarté de l'exposé, la première ligne de commande qui a un `\input` explicite cherche un fichier `to.tex`, ou à défaut `to`, et ajoute à la suite de l'`\input` de ce fichier, le code source `\space␣to`, en effet le premier des deux `\space` est consommé par l'`\input`. Elle aura donc pour nom d'ouvrage `to`. La seconde, qui a un `\input` implicite, a par contre un comportement qui dépend de la distribution. Avec  $\MiKTeX$  elle est équivalente à la première. Mais avec  $\TeX$  Live, elle cherche un fichier `to␣␣to.tex`, ou à défaut `to␣␣to`, et n'ajoute rien à la suite de l'`\input` de ce fichier. Elle aura donc pour nom d'ouvrage `"to␣␣to"`.

## 28.4 Quand quelque chose se passe mal

Si  $\LaTeX$  trouve une erreur dans votre document alors il vous produit un message d'erreur et vous invite à la saisie par un point d'interrogation, `?`. Par exemple, exécuter  $\LaTeX$  sur ce fichier

```
\newcommand{\NP}{\ensuremath{\textbf{NP}}}
Le problème \PN{} est mis à prix à un million de dollars.
```

a pour effet qu'il affiche ceci, et attende une saisie :

```
! Undefined control sequence.
1.5 Le problème \PN
 {} est mis à prix à un million de dollars.
?
```

Le plus simple est de saisir `x` et `RETURN` et de corriger la coquille. Alternativement vous pourriez saisir `?` et `RETURN` pour voir les autres options.

Il y a deux autres scénarios. Le premier est que vous ayez omis le `\end{document}` ou fait une faute de frappe dessus. Dans ce cas `LATEX` produit en invite `*`. Vous pouvez revenir à la ligne de commande en tapant `\stop` et `RETURN`.

Le dernier scénario est que vous ayez fait une faute de frappe sur le nom de fichier. Par exemple, au lieu de `pdflatex test` vous auriez tapé `pdflatex tste`.

```
! I can't find file 'tste'.
<*> tste
```

```
(Press Enter to retry, or Control-D to exit)
Please type another input file name:
```

Le plus simple est de saisir `CTRL D` (en appuyant sur les touches Contrôle et d simultanément), et d'ensuite retaper la ligne de commande correcte.

## Annexe A Patrons de documents

Bien qu'ils soient illustratifs, ces patrons de document vous seront peut-être utiles. Davantage de ressources de patrons sont listées à <https://tug.org/interest.html#laxtextemplates>.

### A.1 Un patron beamer

La classe `beamer` crée des présentations en diapositives. Elle a de très nombreuses possibilités, mais voici un patron de base :

```
\documentclass{beamer}
\usepackage[latin1]{inputenc}

\title{Patron pour la classe Beamer}
\author{Alex L'Auteur}
\date{31 juillet 2020}

\begin{document}

\maketitle

% sans [fragile], n'importe quel code {verbatim} se prend des erreurs mystérieuses.■
\begin{frame}[fragile]
 \frametitle{Première diapositive}

\begin{verbatim}
 C'est \verbatim!
\end{verbatim}

\end{frame}

\end{document}
```

Le paquetage Beamer sur le CTAN : <https://ctan.org/pkg/beamer>.

### A.2 Patron article

Un patron simple pour un article.

```
\documentclass{article}
\title{Patron de la classe article}
\author{Alex Author}

\begin{document}
\maketitle

\section{Première section}
Du texte.
```

```

\subsection{Première section, première sous-section}
Encore du texte.

\section{Deuxième section}
Et du texte.

\end{document}

```

### A.3 Un patron book

Voici un patron simple pour un livre. Voir Section A.4 [Larger book template], page 276, pour un patron plus élaboré.

```

\documentclass{book}
\title{Le Patron de la classe book}
\author{Alex L'Auteur}

\begin{document}
\maketitle

\chapter{Premier}
Du texte.

\chapter{Second}
Et du texte.

\section{Un sujet secondaire}
La fin.

\end{document}

```

### A.4 Un patron book plus élaboré

Voici un patron quelque peu élaboré pour un livre. Voir Section A.3 [book template], page 276, pour un patron plus simple.

Ce patron utilise un `\frontmatter`, `\mainmatter`, et `\backmatter` pour contrôler la typographie des trois zones principales d'un livre (voir Section 6.7 [`\frontmatter` & `\mainmatter` & `\backmatter`], page 50). Le livre a une bibliographie et un index.

Il est également à noter qu'il utilise `\include` et `\includeonly` (voir Chapitre 24 [Splitting the input], page 238). Lorsque vous travaillez sur un chapitre vous pouvez ne pas compiler tous les autres chapitres en mettant en commentaire les entrées correspondantes dans les arguments de `\includeonly`. Ceci accélère la compilation sans perdre aucune information telle que les renvois. (La matière qu'il n'est pas nécessaire de mettre sur une nouvelle page est inclus avec `\input` au lieu de `\include` ; mais ceci au dépens du bénéfice de maintenir les valeurs de renvois).

```

\documentclass[titlepage]{book}
\usepackage{makeidx}\makeindex

```

```
\title{Book Class Template}
\author{Alex Author}

\includeonly{%
% frontcover,
 preface,
 chap1,
% appenA,
}

\begin{document}
\frontmatter
\include{frontcover}
 % maybe comment out while drafting:
\maketitle \input{dedication} \input{copyright}
\tableofcontents
\include{preface}

\mainmatter
\include{chap1}
...
\appendix
\include{appenA}
...

\backmatter
\bibliographystyle{apalike}
\addcontentsline{toc}{chapter}{Bibliography}
\bibliography
\addcontentsline{toc}{chapter}{Index}
\printindex
\include{backcover}
\end{document}
```

## Annexe B Traduction de la licence

La traduction qui suit est donnée uniquement à titre informatif, et ne remplace pas le texte en anglais donné au début de ce document.

Permission vous est donnée de distribuer des copies conformes de ce manuel à condition que les mentions du droit d'auteur et de permission soient préservées sur toutes les copies.

Permission vous est donnée de copier et distribuer des versions modifiées de ce manuel dans les conditions d'une copie conforme, à condition que l'ensemble de l'ouvrage dérivé résultant soit distribué sous les termes d'une mention de permission identique à celle-ci.

Permission vous est donnée de copier et distribuer des traductions de ce manuel dans une autre langue, dans les conditions ci-dessus pour les versions modifiées.

## Index des concepts

- \***
- \* , forme en \* des commandes de rubricage . . . . . 42
  - '\*', invite . . . . . 274
  - '\*', pour faire \nocite sur toutes les clefs . . . . . 106
- 
- 1, écrire dans le flux numéro . . . . . 262
- .**
- .glo, fichier . . . . . 247
  - .idx, fichier . . . . . 248, 249, 251
  - .ind, fichier . . . . . 251
  - .isty, fichier . . . . . 251
- ^**
- ^^J, dans \write . . . . . 263
- \**
- \ (controbligue-espace) . . . . . 195
  - \ESPACE . . . . . 195
  - \fboxrule . . . . . 206
  - \fboxsep . . . . . 206
  - \include imbriqués non autorisés . . . . . 241
  - \SAUTDELIGNE . . . . . 195
  - \TABULATION . . . . . 195
  - \write et sécurité . . . . . 263
  - \write immédiat . . . . . 263
  - \write retardé . . . . . 263
  - \write, flux 16, 17, 18 . . . . . 263
  - \write18, activation . . . . . 268
- à gré, césure . . . . . 114
- écrire des lettres . . . . . 255
- écrire un fichier externe . . . . . 68
- élément extraordinaire . . . . . 263
- élastiques, définir de nouvelles longueurs . . . . . 129
- élastiques, longueurs . . . . . 141
- équations, aligner . . . . . 66
- équations, environnement pour . . . . . 66
- étai . . . . . 123, 200
- étai mathématique . . . . . 181
- étiquette . . . . . 55
- étiquette machine à écrire dans les listes . . . . . 63
- étiquette tapuscrite dans les listes . . . . . 63
- étirement, omettre l'étirement vertical . . . . . 32
- étoilée, forme des commandes de rubricage . . . . . 42
- étoilées, variantes de commandes . . . . . 128
- évidées, lettres . . . . . 168
- évidées, lettres grasses . . . . . 168
- A**
- a e liés . . . . . 234
  - a rond-en-chef . . . . . 234
  - a, e dans l'a . . . . . 234
  - accéder à un caractère
    - quelconque d'une police . . . . . 229
  - accent . . . . . 25
  - accent aigu . . . . . 233
  - accent aigu en math . . . . . 176
  - accent barre suscrite en math . . . . . 176
  - accent brève en math . . . . . 176
  - accent caron . . . . . 234
  - accent caron en math . . . . . 176
  - accent chapeau en math . . . . . 176
  - accent circonflexe . . . . . 233
  - accent circonflexe en math . . . . . 176
  - accent diérèse en math . . . . . 176
  - accent double-brève-renversée . . . . . 234
  - accent grave en math . . . . . 176
  - accent háček . . . . . 234
  - accent háček en math . . . . . 176
  - accent macron en math . . . . . 176
  - accent mathématique, chapeau large . . . . . 176
  - accent mathématique, tilde large . . . . . 176
  - accent point . . . . . 233
  - accent point en chef . . . . . 233
  - accent point suscrit . . . . . 233
  - accent point suscrit en math . . . . . 176
  - accent rond en chef . . . . . 234
  - accent rond-en-chef en math . . . . . 176
  - accent tilde en math . . . . . 176
  - accent tréma . . . . . 233
  - accent tréma en math . . . . . 176
  - accent tréma hongrois . . . . . 234
  - accents . . . . . 233
  - Accents mathématiques . . . . . 176
  - accents, définir . . . . . 19, 21, 22
  - accentuées, lettres . . . . . 233
  - accentués, caractères . . . . . 233
  - accolade droite, dans le texte . . . . . 231
  - accolade gauche, dans le texte . . . . . 230
  - accolades . . . . . 168
  - afficher du texte cité avec
    - renforcement de paragraphe . . . . . 89
  - afficher un texte cité sans
    - renforcement de paragraphe . . . . . 89
  - agrandir la page courante . . . . . 118
  - aigu, accent . . . . . 233
  - align environnement,
    - d'@codeamsmath@comment . . . . . 66

|                                                                                        |     |
|----------------------------------------------------------------------------------------|-----|
| alignement au moyen de <code>tabbing</code> .....                                      | 90  |
| aligner des équations.....                                                             | 66  |
| aligner du texte dans des tableaux.....                                                | 97  |
| aligner du texte en utilisant des<br>taquets de tabulation.....                        | 90  |
| allure de police.....                                                                  | 28  |
| <code>@codeamsmath@comment</code> paquetage,<br>remplaçant <code>eqnarray</code> ..... | 66  |
| annexes d'un livre.....                                                                | 50  |
| anomalies, rapports.....                                                               | 1   |
| aperçu de <code>L<sup>A</sup>T<sub>E</sub>X</code> .....                               | 2   |
| appariés, délimiteurs.....                                                             | 169 |
| appel.....                                                                             | 258 |
| appendice.....                                                                         | 49  |
| appendices.....                                                                        | 49  |
| arguments mouvants.....                                                                | 134 |
| arguments optionnels, définir et utiliser.....                                         | 124 |
| arrière-plan coloré.....                                                               | 214 |
| article d'une liste, compteur spécifiant.....                                          | 138 |
| articles numérotés, compteur spécifiant.....                                           | 138 |
| ASCII, circonflexe dans le texte.....                                                  | 230 |
| ASCII, tilde dans le texte.....                                                        | 230 |
| astérisque centré, dans le texte.....                                                  | 230 |
| auteur, pour la page de titre.....                                                     | 187 |
| aval, référence déclarée en.....                                                       | 55  |
| avant parution, essais.....                                                            | 4   |

## B

|                                                         |     |
|---------------------------------------------------------|-----|
| barre au-dessous.....                                   | 234 |
| barre horizontale, caractère.....                       | 231 |
| barre par-dessus, accent.....                           | 233 |
| barre souscrite.....                                    | 234 |
| barre verticale double, dans le texte.....              | 230 |
| barre verticale, dans le texte.....                     | 230 |
| barre, diacritique.....                                 | 233 |
| barre, double verticale, dans le texte.....             | 230 |
| barre-souscrite, diacritique.....                       | 233 |
| Bas de casse.....                                       | 228 |
| bas de page, créer des notes.....                       | 119 |
| bas de page, des paramètres pour.....                   | 32  |
| bases de <code>L<sup>A</sup>T<sub>E</sub>X</code> ..... | 2   |
| <code>batchmode@comment</code> .....                    | 268 |
| <code>@codebeamer@comment</code> patron et classe.....  | 275 |
| bibliographie, créer (automatiquement).....             | 105 |
| bibliographie, créer (manuellement).....                | 102 |
| <code>BibT<sub>E</sub>X</code> , messages d'erreur..... | 106 |
| <code>bibT<sub>E</sub>X</code> , utilisation.....       | 105 |
| big point.....                                          | 142 |
| blancs, après une séquence de contrôle.....             | 125 |
| boîte.....                                              | 204 |
| boîte englobante.....                                   | 222 |
| boîte, allouer une nouvelle.....                        | 130 |
| boîte, sauvegarde.....                                  | 208 |
| boîte, utilise une boîte sauvegardée.....               | 210 |
| boîtes.....                                             | 204 |
| boîtes colorées.....                                    | 214 |
| boîtes noires, en omettant.....                         | 8   |

|                                                |     |
|------------------------------------------------|-----|
| bogues, rapports.....                          | 1   |
| Bonjour le monde.....                          | 2   |
| <code>book</code> , annexes.....               | 50  |
| <code>book</code> , pièces postliminaires..... | 50  |
| <code>book</code> , pièces préliminaires.....  | 50  |
| <code>book</code> , pièces principales.....    | 50  |
| brève, accent.....                             | 234 |
| brève, diacritique.....                        | 234 |

## C

|                                                |          |
|------------------------------------------------|----------|
| cédille, diacritique.....                      | 233      |
| césure à gré.....                              | 114      |
| césure discrétionnaire.....                    | 114      |
| césure, définir.....                           | 115      |
| césure, empêcher.....                          | 204      |
| césure, forcer.....                            | 113      |
| cadratin.....                                  | 143, 231 |
| cadratin, demi.....                            | 231      |
| cadratin, deux tiers.....                      | 232      |
| cadratin, trois-quarts.....                    | 232      |
| calligraphique, fontes.....                    | 168      |
| Capitales.....                                 | 228      |
| captions.....                                  | 40       |
| caractère invisible.....                       | 180, 181 |
| caractère tiret numérique.....                 | 231      |
| caractère, codage de.....                      | 235      |
| caractères latins non diacrités.....           | 234      |
| caractères réservés.....                       | 228      |
| caractères spéciaux.....                       | 228, 234 |
| caron, diacritique.....                        | 234      |
| carrée, racine.....                            | 184      |
| Casse de caractères.....                       | 228      |
| cassetin, fonte.....                           | 19       |
| catégorie, code de caractère.....              | 127      |
| catcode.....                                   | 127      |
| centimètre.....                                | 142      |
| centré, astérisque, dans le texte.....         | 230      |
| centré, point, dans le texte.....              | 232      |
| centrer du texte, déclaration pour.....        | 62       |
| centrer du texte, environnement pour.....      | 61       |
| Changer la casse de caractères.....            | 228      |
| changer la taille.....                         | 226      |
| chapeau large, accent, math.....               | 176      |
| chapeau, accent.....                           | 233      |
| chapitre.....                                  | 42, 44   |
| <code>chapter</code> .....                     | 42       |
| chargement des paquetages supplémentaires..... | 9        |
| chevron double, guillemets.....                | 230      |
| chevron droit, guillemets.....                 | 230      |
| chevron gauche, guillemets.....                | 230      |
| chevron simple, guillemets.....                | 230      |
| chiffres alignés.....                          | 25       |
| chiffres bas-de-casse.....                     | 25       |
| chiffres Didot.....                            | 25       |
| chiffres elzéviens.....                        | 25       |
| chiffres français.....                         | 25       |
| chiffres minuscules.....                       | 25       |

- chiffres non alignés ..... 25
  - chiffres suspendus ..... 25
  - cicéro ..... 143
  - circonflexe ASCII, dans le texte ..... 230
  - circonflexe, accent ..... 233
  - circonflexe, diacritique ..... 233
  - citation ..... 89
  - classe de document, commandes ..... 9
  - classe de document, définition ..... 2
  - classe et paquetage, commandes ..... 10
  - classe et paquetage, différence ..... 9
  - classe et paquetage, structure ..... 9
  - classe, disposition du fichier ..... 9
  - classe, fichier d'exemple ..... 10
  - classe, options ..... 9, 11
  - classes de documents ..... 7
  - clause at, au sein d'une définition de police ..... 134
  - clef de référence bibliographique ..... 103
  - CLI ..... 267
  - codage de fonte ..... 19
  - codage des fichiers d'entrée ..... 235
  - code de catégorie de caractère ..... 127
  - code entré sur la ligne de commande ..... 269
  - code source, composer du ..... 108
  - color options du paquetage ..... 211
  - color, commandes du paquetage ..... 212
  - coloré, texte ..... 213
  - colorée, page ..... 214
  - colorées, boîtes ..... 214
  - commande longue ..... 10
  - commandes de fontes de bas niveau ..... 27
  - commandes de fontes, de bas niveau ..... 27
  - commandes de mise en page ..... 30
  - commandes de rubricage, ..... 42
  - commandes des classes de document ..... 9
  - commandes exécutées depuis L<sup>A</sup>T<sub>E</sub>X ..... 265
  - commandes fragiles ..... 134
  - commandes pour classe ou paquetage ..... 10
  - commandes robustes ..... 134
  - commandes, définir de nouvelles ..... 127
  - commandes, en définir de nouvelles ..... 124
  - commandes, ignorer les espaces ..... 135, 136
  - commandes, paquetage graphics ..... 219
  - commandes, redéfinir ..... 124
  - commandes, syntaxe des ..... 5
  - commandes, variantes étoilées ..... 128
  - commencer une nouvelle page ..... 117
  - commencer une nouvelle page et
    - débourrer les flottants ..... 117
  - commencer une page de droite ..... 117
  - compteurs, définir de nouveaux ..... 129
  - compteurs, obtenir la valeur d'un ..... 139
  - compteurs, réglage ..... 139
  - compteurs, une liste de ..... 138
  - configuration du paquetage graphics ..... 216
  - contracter ..... 226
  - controbligue, dans le texte ..... 230
  - copie, liste des personnes en, dans une lettre ..... 256
  - Correction italique ..... 197
  - correspondantes, parenthèses ..... 169
  - correspondants, crochets ..... 169
  - couleur ..... 211, 212, 213, 214
  - couleur, définir ..... 212
  - coupure de mot, empêcher ..... 204
  - crédits, note en bas de page ..... 188
  - créer des images ..... 81
  - créer des tableaux ..... 96
  - crochets ..... 168
  - CTAN ..... 6
- ## D
- débord ..... 75
  - débourrer les flottants et
    - commencer une page ..... 117
  - début de document, crochet ..... 64
  - début et fin ..... 2
  - déclaration en aval de référence ..... 55
  - déclarative, forme des commandes de contrôle
    - de la taille de police ..... 26
  - déclarative, forme des commandes
    - de style de police ..... 23
  - définir de nouveaux environnements ..... 130
  - définir de nouveaux théorèmes ..... 132
  - définir de nouvelles polices ..... 134
  - définir une couleur ..... 212
  - définir une nouvelle commande ..... 124, 127
  - définitions ..... 124
  - délimiteur nul ..... 169
  - délimiteurs ..... 168
  - délimiteurs appairés ..... 169
  - DANTE e.V.@comment ..... 6
  - date d'aujourd'hui ..... 236
  - date, pour la page de titre ..... 188
  - demi-cadratin ..... 231
  - deux-points, caractère ..... 183
  - deux-tiers cadratin ..... 232
  - diacritique accent aigu ..... 233
  - diacritique barre-souscrite ..... 233
  - diacritique brève ..... 234
  - diacritique cédille ..... 233
  - diacritique caron ..... 234
  - diacritique circonflexe ..... 233
  - diacritique double-brève-renversée ..... 234
  - diacritique háček ..... 234
  - diacritique macron-souscrit ..... 233
  - diacritique ogonek ..... 234
  - diacritique point suscrit ..... 233
  - diacritique point-souscrit ..... 234
  - diacritique rond en chef ..... 234
  - diacritique tilde ..... 233
  - diacritique tirant suscrit ..... 234
  - diacritique tréma ..... 233
  - diacritique tréma hongrois ..... 234
  - diacritiques, signes ..... 233
  - différence entre classe et paquetage ..... 9

dilatation verticale infinie ..... 202  
 dilater ..... 226  
 dimen  $\TeX$  de base ..... 141  
 dimension de fonte, inclinaison ..... 198  
 discrétionnaire, césure ..... 114  
 document, nom racine ..... 270  
 dollar, signe ..... 231  
 double espacement ..... 36  
 double obèle, dans le texte ..... 229  
 double renvoi, résolution ..... 3  
 double, barre verticale, dans le texte ..... 230  
 double, guillemet-apostrophe culbuté ..... 232  
 double, guillemet-apostrophe de droite ..... 232  
 double, obèle, dans le texte ..... 231  
 double-brève-renversée, diacritique ..... 234  
 douze ..... 143  
 droit d'auteur, symbole ..... 229  
 droit, guillemet double sur la ligne de base ..... 232  
 droit, guillemet simple sur la ligne de base ..... 232  
 droit, guillemet-apostrophe simple ..... 232  
 droite, accolade, dans le texte ..... 231  
 droite, flèche, dans le texte ..... 232  
 droite, guillemet anglais simple ..... 230  
 droite, guillemet-apostrophe simple ..... 230

## E

e dans l'a ..... 234  
 e dans l'o ..... 235  
 e- $\TeX$  ..... 4  
 elzéviens, chiffres ..... 25  
 em ..... 143  
 emphasis ..... 24  
 en-tête de page ..... 189  
 en-tête et pied de page en cours ..... 32  
 en-tête, des paramètres pour ..... 32  
 encerclée, lettre, dans le texte ..... 231  
 englobante, boîte ..... 222  
 entré, code sur la ligne de commande ..... 269  
 entrée Unicode, native ..... 3  
 entrée/sortie console ..... 260  
 environnement ..... 2  
 environnement simili-théorème ..... 132  
 Environnements ..... 59  
 environnements, définir ..... 130  
 EPS, fichiers ..... 216, 220  
 errorstopmode@comment ..... 268  
 espace blanc ..... 191  
 espace entre les lignes ..... 36  
 espace fine ..... 179, 197  
 espace fine négative ..... 179, 197  
 espace horizontal ..... 192  
 espace horizontal dilatable ..... 192  
 espace insécable ..... 196  
 espace vertical ..... 203  
 espace vertical avant les paragraphes ..... 150  
 espace vertical, insérer ..... 203  
 espace vertical, mode mathématique ..... 179

espace verticale ..... 201  
 espace visible ..... 109  
 espace visible, symbole, dans le texte ..... 233  
 espace, insérer de l'espace horizontal ..... 192  
 espacement au sein du mode mathématique ... 178  
 espacement inter-phrase ..... 194, 195  
 espacement, mode mathématique ..... 180, 181  
 espaces ..... 191  
 espaces, ignorer au voisinage  
   d'une commande ..... 135  
 espaces, ignorer autour des commandes ..... 136  
 eszett lettre allemande ..... 235  
 eth@comment islandais ..... 235  
 eth@comment , lettre islandaise ..... 235  
 euro, symbole ..... 231  
 ex ..... 143  
 exploitation, information système de ..... 266  
 exposant ..... 154  
 extension, structure ..... 9  
 externe, mode paragraphe ..... 185  
 externes, commandes ..... 265

## F

féminin, symbole ordinal ..... 232  
 fabriquer une boîte ..... 204  
 fabriquer une page de titre ..... 107  
 Faire des paragraphes ..... 148  
 famille de polices ..... 27  
 fermant, guillemet anglais simple ..... 230  
 fermant, guillemet-apostrophe simple ..... 230  
 fichier auxiliaire ..... 3  
 fichier d'entrée ..... 238  
 fichier de transcription ..... 3  
 fichier externe, écrire ..... 68  
 fichier format  $\TeX$  ..... 3  
 fichier journal ..... 3  
 fichier racine ..... 238, 270  
 fichier, fermeture ..... 260  
 fichier, lecture d'un ..... 261  
 fichier, ouverture ..... 260  
 fichiers source, rendre autonomes ..... 68  
 figures, insertion ..... 67  
 figures, notes en bas de page au sein des ..... 80  
 fin d'une lettre ..... 257  
 fin document, crochet ..... 64  
 fin et début ..... 2  
 fine, espace ..... 179, 197  
 fine, espace, négative ..... 197  
 flèche droite, dans le texte ..... 232  
 flèche gauche, left, dans le texte ..... 232  
 flèches ..... 166  
 fonctions mathématiques ..... 174  
 fonte, codage ..... 17, 19  
 fonte, déclarer le codage ..... 19  
 fonte, dimension, inclinaison ..... 198  
 fontes ..... 17  
 fontes scriptes mathématiques ..... 168

**force**, option de `filecontents` ..... 68  
 format L<sup>A</sup>T<sub>E</sub>X, fichier (`.fmt`) ..... 3  
 format T<sub>E</sub>X, fichier (`.fmt`) ..... 3  
 forme étoilée des commandes  
   d'environnement ..... 130  
 forme étoilée des commandes de rubricage ..... 42  
 forme étoilée, définir de  
   nouvelles commandes ..... 124  
 forme déclarative des commandes de contrôle  
   de la taille de police ..... 26  
 forme déclarative des commandes  
   de style de police ..... 23  
 forme en \* des commandes d'environnement ... 130  
 forme en \* des commandes de rubricage ..... 42  
 forme en \*, définir de nouvelles commandes ... 124  
 forme par environnement des commandes de  
   contrôle de la taille de police ..... 26  
 formules en ligne ..... 78  
 formules mathématiques ..... 153  
 formules, environnement pour ..... 66  
 formules, math ..... 153  
 fraction ..... 183  
 fragile, commande ..... 134  
 français, guillemets ..... 230

## G

gauche, accolade, dans le texte ..... 230  
 gauche, flèche, dans le texte ..... 232  
 gauche, guillemet-apostrophe simple de ..... 230  
 globales, options ..... 9  
 glossaires ..... 247  
 glue T<sub>E</sub>X de base ..... 141  
 graisses de police ..... 28  
 Grand rond, symbole, dans le texte ..... 230  
 graphics ..... 216  
 graphics, changer la taille ..... 226  
 graphics, commandes du paquetage ..... 219  
 graphics, options du paquetage ..... 215  
 graphics, paquetage ..... 215, 216, 220  
 graphiques ..... 215, 220  
 graphiques, changer d'échelle ..... 226  
 graphiques, contracter ..... 226  
 graphiques, dilatation & contraction ..... 226  
 graphiques, dilater ..... 226  
 graphiques, homothétie ..... 226  
 graphiques, paquetages ..... 86  
 gras ..... 25  
 gras machine à écrire, éviter ..... 63  
 gras tapuscrit, éviter ..... 63  
 gras, caractères, en mathématiques ..... 167  
 gras, mathématiques ..... 25  
 grave accent ..... 233  
 grecques, lettres ..... 174  
 guillemet droit double sur la ligne de base ..... 232  
 guillemet droit simple sur la ligne de base ..... 232  
 guillemet-apostrophe de droite double ..... 232  
 guillemet-apostrophe double culbuté ..... 232

guillemet-apostrophe simple ..... 232  
 guillemet-apostrophe simple culbuté ..... 232  
 guillemet-apostrophe simple droit ..... 232  
 guillemet-virgule inférieur double ..... 230  
 guillemet-virgule inférieur simple ..... 230  
 guillemet-virgule inférieur, simple et double ... 230  
 guillemets double ..... 230  
 guillemets français ..... 230  
 guillemets simple ..... 230

## H

homothétie ..... 226  
 horizontal, espace ..... 192  
 horizontal, espacement ..... 180  
 horizontal, renforcement d'un paragraphe ..... 150

## I

i sans point ..... 234  
 i sans point en math ..... 158  
 ici, placer les flottants ..... 38  
 idx@comment, fichier ..... 251  
 images, créer ..... 81  
 imbrication de `\include` non autorisée ..... 241  
 implémentations de T<sub>E</sub>X ..... 3  
 importer des graphiques ..... 220  
 inclure des graphiques ..... 220  
 index ..... 248  
 index, articles, « voir » et « voir aussi » ..... 250  
 index, articles, sous-articles ..... 249  
 index, impression ..... 254  
 index, page de pages ..... 250  
 index, production manuelle ..... 249  
 indexes multiples ..... 248  
 indice inférieur ..... 154  
 indice supérieur ..... 154  
 inférieur à, symbole dans le texte ..... 232  
 inférieur, guillemet-virgule double ..... 230  
 inférieur, guillemet-virgule simple ..... 230  
 insécable, caractère trait d'union ..... 232  
 insécable, espace ..... 196  
 insérer des figures ..... 67  
 interface ligne de commande ..... 267  
 interligne, espace ..... 36  
 interne, mode paragraphe ..... 185  
 interpréteur de commandes du système  
   d'exploitation, accès à ..... 265  
 invisible, caractère ..... 180, 181  
 invite '\*' ..... 274  
 islandaise, thorn@comment, lettre ..... 235  
 italique ..... 25

**J**

|                                                    |          |
|----------------------------------------------------|----------|
| j sans point                                       | 234      |
| j sans point en math                               | 158      |
| jobname                                            | 268      |
| jobname, nom d'ouvrage                             | 270      |
| journal, écrire vers le fichier                    | 262      |
| JPEG, fichiers                                     | 216, 220 |
| JPG, fichiers                                      | 216, 220 |
| justifié à gauche du texte                         | 69       |
| justification, drapeau droit                       | 69       |
| justification, drapeau gauche                      | 70       |
| justifier à droite, environnement pour             | 70       |
| justifier à gauche du texte,<br>environnement pour | 69       |
| justifier le texte à droite                        | 70       |

**K**

|                  |   |
|------------------|---|
| Knuth, Donald E. | 2 |
|------------------|---|

**L**

|                                                                                     |     |
|-------------------------------------------------------------------------------------|-----|
| l polonais                                                                          | 235 |
| L'équipe de projet L <sup>A</sup> T <sub>E</sub> X                                  | 1   |
| Lamport T <sub>E</sub> X                                                            | 2   |
| Lamport, Leslie                                                                     | 2   |
| Largeurs de police                                                                  | 28  |
| L <sup>A</sup> T <sub>E</sub> X aperçu                                              | 2   |
| L <sup>A</sup> T <sub>E</sub> X ou L <sup>A</sup> T <sub>E</sub> X2e, que choisir ? | 1   |
| L <sup>A</sup> T <sub>E</sub> X, format, fichier (.fmt)                             | 3   |
| L <sup>A</sup> T <sub>E</sub> X, logo                                               | 229 |
| L <sup>A</sup> T <sub>E</sub> X2e logo                                              | 229 |
| lecture d'un fichier                                                                | 261 |
| Lettre grecques                                                                     | 155 |
| lettre ij@comment, hollandais                                                       | 235 |
| lettres calligraphiques pour les<br>mathématiques                                   | 25  |
| lettres de script pour les mathématiques                                            | 25  |
| lettres grecques                                                                    | 174 |
| lettres latines non diacritées                                                      | 234 |
| lettres, écrire des                                                                 | 255 |
| lettres, commencer                                                                  | 258 |
| lettres, salutation                                                                 | 257 |
| ligne de commande, interface                                                        | 267 |
| ligne tabulée (tabbing)                                                             | 91  |
| lignes dans des tableaux                                                            | 97  |
| lire un fichier                                                                     | 261 |
| liste des figures, fichier                                                          | 3   |
| liste des personnes en copie, dans une lettre                                       | 256 |
| liste des tableaux, fichier                                                         | 3   |
| listes à puces                                                                      | 71  |
| listes étiquetées, créer                                                            | 63  |
| listes d'articles                                                                   | 71  |
| listes d'articles numérotés                                                         | 65  |
| listes d'articles, générique                                                        | 72  |
| listes de description, créer                                                        | 63  |
| listes non ordonnées                                                                | 71  |
| livre, annexes                                                                      | 50  |

|                                              |     |
|----------------------------------------------|-----|
| livre, pièces postliminaires                 | 50  |
| livre, pièces préliminaires                  | 50  |
| livre, pièces principales                    | 50  |
| livre, symbole                               | 230 |
| locale, information, du système              | 266 |
| logo L <sup>A</sup> T <sub>E</sub> X         | 229 |
| logo L <sup>A</sup> T <sub>E</sub> X2e       | 229 |
| logo T <sub>E</sub> X                        | 230 |
| longueur, ajouter à une                      | 144 |
| longueur, régler une                         | 143 |
| longueurs élastiques, définir de nouvelles   | 129 |
| longueurs, allouer de nouvelles              | 129 |
| longueurs, définir et utiliser               | 141 |
| LuaT <sub>E</sub> X                          | 4   |
| LuaT <sub>E</sub> X, 256 flux de sortie dans | 262 |

**M**

|                                          |     |
|------------------------------------------|-----|
| m, largeur d'un                          | 143 |
| macron, diacritique                      | 233 |
| macron-souscrit, diacritique             | 233 |
| Madsen, Lars@comment                     | 66  |
| makeindex, programme                     | 251 |
| marge gauche                             | 33  |
| marge gauche nominale                    | 33  |
| marge supérieure                         | 35  |
| marge supérieure nominale                | 35  |
| marque déposée, symbole                  | 232 |
| marque de fabrique, symbole              | 232 |
| masculin, symbole ordinal                | 232 |
| Math Calligraphic                        | 168 |
| math, accent aigu                        | 176 |
| math, accent barre suscrite              | 176 |
| math, accent brève                       | 176 |
| math, accent grave                       | 176 |
| math, accent macron                      | 176 |
| math, accent point suscrit               | 176 |
| math, accent rond-en-chef                | 176 |
| math, accents                            | 176 |
| math, fonctions                          | 174 |
| math, miscellanées                       | 183 |
| math, styles                             | 181 |
| mathématiques, gras                      | 25  |
| matières, fichier                        | 3   |
| messages d'erreur de BibT <sub>E</sub> X | 106 |
| millimètre                               | 142 |
| minipage, créer une                      | 79  |
| miroirs du CTAN                          | 6   |
| miscellanées en math                     | 183 |
| mise en page, les paramètres de          | 32  |
| modèle de document                       | 275 |
| modèles de couleur                       | 211 |
| mode gauche-à-droite (left-to-right)     | 185 |
| mode LR                                  | 185 |
| mode math                                | 185 |
| mode math d'affichage                    | 185 |
| mode mathématique, entrer                | 153 |
| mode mathématique, espace vertical       | 179 |

mode mathématique, espacement . . . . 178, 180, 181  
mode paragraphe . . . . . 185, 206  
mode vertical . . . . . 185  
mode vertical interne . . . . . 185  
modes . . . . . 185  
monnaie, dollar . . . . . 231  
monnaie, euro . . . . . 231  
mot de contrôle, défini . . . . . 126  
moteur pdf $\TeX$  . . . . . 4  
moteurs,  $\TeX$  . . . . . 3  
motif courant de taquets de  
  tabulation (**tabbing**) . . . . . 91  
mpfootnote, compteur . . . . . 119  
mu, unité mathématique . . . . . 143  
multiples, indexes . . . . . 248  
multiplication discrétionnaire . . . . . 183

## N

NBSP . . . . . 196  
**noheader**, option de **filecontents** . . . . . 68  
nom d'ouvrage (jobname) . . . . . 270  
nom de fichier de la tâche courante . . . . . 268  
nonstopmode@comment . . . . . 268  
**nosearch**, option de **filecontents** . . . . . 68  
note en bas de page portant sur  
  une autre note . . . . . 122  
note en bas de page, créer . . . . . 119  
note en bas de page, récursivement . . . . . 122  
note en bas de page, symboles au  
  lieu de numéros . . . . . 119  
notes dans la marge . . . . . 151  
notes en bas de page au sein des figures . . . . . 80  
notes en bas de page, au sein d'une minipage . . . . . 119  
notes en bas de page, dans les  
  titres de rubrique . . . . . 120  
Notes en bas de page, dans un tableau . . . . . 121  
notes en bas de page, paramètres . . . . . 122  
Notes en marge . . . . . 151  
nouvelle commande, définition . . . . . 12  
nouvelle commande, vérification . . . . . 10  
nouvelle commandes, définir . . . . . 124  
nouvelle ligne, commencer une . . . . . 111  
nouvelle ligne, commencer une  
  (mode paragraphe) . . . . . 113  
nouvelle page, commencer . . . . . 117  
nouvelles classes, commandes . . . . . 9  
nouvelles commandes, définir . . . . . 127  
nul, délimiteur . . . . . 169  
numéro d'équation, omettre . . . . . 66  
numéro d'équation, renvoi . . . . . 57  
numéro de figure, renvoi . . . . . 57  
numéro de note en bas de page, renvoi . . . . . 57  
numéro de page, renvoi . . . . . 56  
numéro de rubrique, renvoi . . . . . 57  
numéros de rubrique, composition . . . . . 43  
numérotés, listes d'articles . . . . . 65  
numérotation de page, style . . . . . 188

## O

o e liés . . . . . 235  
o, e dans l'o . . . . . 235  
obèle double, dans le texte . . . . . 231  
obèle, dans le texte . . . . . 229, 231  
officieuse, nature de ce document . . . . . 1  
ogonek, diacritique . . . . . 234  
option **stable** du paquetage  
  @codefootmisc@comment . . . . . 120  
optionnels, arguments, définir et utiliser . . . . . 124  
options de classe . . . . . 7, 11  
options de classe de document . . . . . 7  
options de paquetage . . . . . 11  
options globales . . . . . 7, 9  
options passée à la ligne de commande . . . . . 267  
options pour classe de document . . . . . 9  
options pour paquetage . . . . . 9  
options, classe de document . . . . . 7  
options, paquetage color . . . . . 211  
options, paquetage graphics . . . . . 215  
Ordinaux espagnols, féminin et masculin . . . . . 232  
Ordinaux, féminin et masculin . . . . . 232  
oslash . . . . . 235  
OT1 . . . . . 17  
ouvrant, , guillemet-apostrophe simple . . . . . 230  
ouvrir un fichier . . . . . 260  
**overwrite**, option de **filecontents** . . . . . 68

## P

page colorée . . . . . 214  
page de flottants . . . . . 38  
page, paramètres de mise en page . . . . . 32  
page, styles . . . . . 187  
pages de titre, création . . . . . 107  
pages, saut . . . . . 117  
paquet de macro,  $\LaTeX$  en tant que . . . . . 2  
paquetage, disposition du fichier . . . . . 9  
paquetage, options . . . . . 9, 11  
paquetages, le chargement . . . . . 9  
**paragraph** . . . . . 42, 48  
paragraphe . . . . . 42  
paragraphe, dans une boîte . . . . . 206  
paragraphe, mode . . . . . 206  
paragraphe, symbole . . . . . 230  
paragraphe, terminaison . . . . . 149  
paragraphe . . . . . 148  
paragraphe, renforcement au sein  
  d'une minipage . . . . . 80  
paramètres, la mise en page . . . . . 32  
paramètres, pour notes en bas de page . . . . . 122  
paramètres, substitution . . . . . 125  
parenthèses . . . . . 168  
**part** . . . . . 42, 43  
partie . . . . . 42  
partitionner le fichier d'entrée . . . . . 238  
parution, candidats . . . . . 4  
pas, registre de,  $\TeX$  de base . . . . . 129

patron (simple), **article**..... 275  
patron **book** ..... 276  
patron de document ..... 275  
patron, @codebeamer@comment ..... 275  
patron, **book**..... 276  
PDF, fichiers graphiques ..... 216, 220  
pdfTeX..... 3  
pdfTeX, moteur ..... 4  
pièces jointes, liste ..... 257  
pièces postliminaires d'un livre ..... 50  
pièces préliminaires d'un livre ..... 50  
pièces principales d'un livre ..... 50  
pica ..... 142  
pied de page ..... 189  
Pied-de-mouche ..... 230  
pile, math ..... 184  
placement des flottants ..... 38  
PNG, fichiers ..... 216, 220  
poésie, un environnement pour la ..... 109  
point ..... 142  
point centré, dans le texte ..... 232  
point d'exclamation culbuté ..... 231  
point d'interrogation culbuté ..... 232  
point Didot ..... 143  
point en chef, diacritique ..... 233  
point PostScript ..... 142  
point proportionné (scaled point) ..... 143  
point suscrit, diacritique ..... 233  
point terminant une abréviation ..... 193  
point terminant une phrase ..... 193  
point, espacement après ..... 193  
points de suite, pointillés dans la  
table des matières ..... 243  
points de suspension ..... 173, 230  
police de largeur fixe ..... 25  
police inclinée ..... 25  
police machine à écrire ..... 25  
police oblique ..... 25  
police petites capitales ..... 25  
police romaine ..... 25  
police tapuscrit ..... 25  
police, symboles par leur numéro ..... 229  
polices de caractères ..... 17  
polices OpenType ..... 3  
polices TrueType ..... 3  
polices, catalogue ..... 27  
polices, commandes pour  
déclarer de nouvelles ..... 134  
position dans picture ..... 82  
postscriptum, dans une lettre ..... 258  
pouce (inch) ..... 142  
préambule, définition ..... 2  
programmes informatiques, composer du ..... 108  
prononciation ..... 2  
puce, dans le texte ..... 231

## Q

quad ..... 179

## R

référence bibliographique, clef ..... 103  
référence déclarée en aval ..... 55  
réglage de compteurs ..... 139  
répertoire de sortie pour tous les  
fichiers externes ..... 268  
répertoire, listage, du système ..... 266  
répertoires parents, impossible d'écrire vers ..... 264  
réservés, caractères ..... 228  
résumé ..... 59  
racine du nom du document ..... 270  
racine, fichier ..... 238, 270  
radical ..... 184  
radical ..... 184  
rapports d'anomalies ..... 1  
rapports de bogues ..... 1  
redéfinir des environnements ..... 130  
redéfinir une commande ..... 124  
relation, texte au-dessus ..... 184  
remarques dans la marge ..... 151  
remerciements, pour la page de titre ..... 188  
renforcement d'un paragraphe ..... 150  
renforcement des paragraphes  
dans du texte cité ..... 89  
renforcement des paragraphes dans un  
texte cité, omettre ..... 89  
renforcement des paragraphes, au  
sein d'une minipage ..... 80  
renforcement, forcer ..... 150  
renvoi avec numéro de page ..... 56  
renvoi en aval, résolution ..... 3  
renvoi, résolution ..... 3  
renvoi, symbolique ..... 57  
renvois ..... 55  
renvois entre documents ..... 57  
repli, nom d'ouvrage ..... 272  
ressort, registre de, TeX de base ..... 129  
restreint, accès interpréteur de commande du  
système d'exploitation ..... 266  
restreint, accès shell ..... 266  
retrait d'un paragraphe ..... 150  
rigides, longueurs ..... 141  
robuste, commande ..... 134  
rond en chef, diacritique ..... 234  
rond, grand symbole, dans le texte ..... 230  
rotation ..... 225  
rotation, appliquer à du texte ..... 225  
rotation, appliquer à un graphique ..... 225  
rubricage, commandes ..... 42  
rubricage, part ..... 43

## S

- s long..... 235
- sécurité et `\write` ..... 263
- séquences de contrôle ..... 126
- séries de polices ..... 27
- sans empattement..... 25
- sans point, i, math ..... 158
- sans point, j, math ..... 158
- saut à la ligne ..... 111
- saut de ligne ..... 113
- saut de ligne, dans `\write` ..... 263
- saut de ligne, forcer ..... 111
- saut de ligne, sortie semblable à
  - l'entrée pour le ..... 112
- saut de page ..... 117
- saut de page, empêcher ..... 118
- saut de page, forcer ..... 118
- sauter à la ligne ..... 111
- sauts de ligne, changer ..... 115
- sauts de ligne, empêcher ..... 116
- sauts de ligne, forcer ..... 116
- scriptes, fontes ..... 168
- scrollmode@comment ..... 268
- section ..... 42, 46
- section, redéfinir ..... 51
- section, symbole ..... 230
- shell escape ..... 268
- signes diacritiques ..... 233
- simple droit, guillemet-apostrophe ..... 232
- simple, guillemet-apostrophe ..... 232
- simple, guillemet-apostrophe culbuté ..... 232
- simuler du texte tapuscrit ..... 108
- skip  $\TeX$  de base ..... 141
- sloppypar@comment, environnement ..... 115
- sortie à deux colonnes ..... 30
- sortie à une colonne ..... 30
- sortie, pour tous les fichiers
  - externes, répertoire de ..... 268
- souligné, dans le texte ..... 233
- souligner ..... 177
- sources autonomes ..... 68
- sous-paragraphe ..... 42
- sous-section ..... 42
- sous-sous-section ..... 42
- spéciaux, caractères ..... 228, 234
- spécificateur, placement de flottants ..... 38
- stable**, option du paquetage
  - @codefootmisc@comment ..... 120
- sterling, symbole ..... 230
- style affichage ..... 182
- style d'en-tête et pied de page ..... 189
- style de numérotation de page ..... 188
- style script ..... 182
- style scriptscript ..... 182
- style texte ..... 182
- styles de page ..... 187
- styles de police ..... 23
- styles de texte ..... 23
- styles typographiques ..... 23
- subparagraph** ..... 42, 48
- subsection** ..... 42, 47
- subsubsection** ..... 42, 48
- supérieur à, symbole dans le texte ..... 231
- sur-ligner ..... 177
- suspension, points de ..... 230
- symbole de multiplication, saut de
  - ligne discrétionnaire ..... 183
- symbole droit d'auteur ..... 229
- symbole marque déposée ..... 232
- symbole marque de fabrique ..... 232
- symbole puce ..... 156
- symbole vecteur en math ..... 176
- symbole, définition ..... 22
- symboles d'une police, par leur numéro ..... 229
- symboles de contrôle, défini ..... 126
- symboles mathématiques ..... 155
- symboles texte ..... 229
- symboles, en caractères gras ..... 167
- symboles, flèches ..... 166
- système, commandes exécutées depuis  $\LaTeX$  .. 265
- système, information ..... 266

## T

- T1 ..... 17
- table des matières ..... 246
- table des matières, éviter les notes
  - en bas de page ..... 120
- table des matières, ajouter
  - manuellement un article ..... 244
- table des matières, création ..... 242
- table des matières, crée une ligne pointillée ..... 243
- table des matières, fichier ..... 3
- table des matières, impression des
  - numéros de rubrique ..... 43
- tableaux mathématiques ..... 60
- tableaux, créer ..... 96
- taille de conception, au sein d'une
  - définition de police ..... 134
- tailles de police ..... 26
- tailles de texte ..... 26
- tailles des polices de caractères ..... 26
- taquet courant (**tabbing**) ..... 91
- taquet de début de ligne courant (**tabbing**) ..... 91
- taquets de tabulation, utiliser ..... 90
- terminal, écrire vers le ..... 262
- $\TeX$ , format ..... 3
- $\TeX$ , logo ..... 230
- texte cité avec renforcement des
  - paragraphe, affichage ..... 89
- texte cité sans renforcement des
  - paragraphe, affichage ..... 89
- texte drapeau droit ..... 69
- texte drapeau droit, environnement pour ..... 69
- texte drapeau gauche ..... 70
- texte drapeau gauche, environnement pour ..... 70

|                                                |     |                                              |          |
|------------------------------------------------|-----|----------------------------------------------|----------|
| texte en couleur .....                         | 213 | TS1, codage .....                            | 229      |
| texte fer à gauche .....                       | 69  | <b>U</b>                                     |          |
| texte fer à gauche, environnement pour .....   | 69  | Unicode .....                                | 235      |
| texte ferré à gauche .....                     | 69  | unités, de longueur .....                    | 142      |
| texte ferré à gauche, environnement pour ..... | 69  | UTF-8 .....                                  | 3        |
| texte sur plusieurs colonnes .....             | 30  | UTF-8, prise en charge des fontes .....      | 17       |
| texte tapuscrit, simulation de .....           | 108 | utilisation de BibTeX .....                  | 105      |
| texte verbatim .....                           | 108 | <b>V</b>                                     |          |
| texte verbatim, en-ligne .....                 | 109 | variables, une liste de .....                | 138      |
| texte, changer la taille .....                 | 226 | vertical, espacement .....                   | 180, 181 |
| texte, contracter .....                        | 226 | verticale infinie, dilatation .....          | 202      |
| texte, dilater .....                           | 226 | verticale, barre dans le texte .....         | 230      |
| texte, homothétie sur du .....                 | 226 | verticale, espace .....                      | 201      |
| texte, symboles .....                          | 229 | visible, symbole espace, dans le texte ..... | 233      |
| théorème, environnement simili- .....          | 132 | voir et voir aussi, articles d'index .....   | 250      |
| théorèmes, composition des .....               | 107 | <b>W</b>                                     |          |
| théorèmes, définition .....                    | 132 | Wright, Joseph .....                         | 229      |
| thickspace .....                               | 178 | <b>X</b>                                     |          |
| thorn@comment , lettre islandaise .....        | 235 | x, hauteur d'un .....                        | 143      |
| tilde ASCII, dans le texte .....               | 230 | XeTeX .....                                  | 4        |
| tilde large, accent, math .....                | 176 | xindy, programme .....                       | 253      |
| tilde, diacritique .....                       | 233 |                                              |          |
| tirant suscrit, diacritique .....              | 234 |                                              |          |
| tiret bas, dans le texte .....                 | 233 |                                              |          |
| titre, pour la page de titre .....             | 188 |                                              |          |
| titres, fabrication .....                      | 187 |                                              |          |
| tréma hongrois, accent .....                   | 234 |                                              |          |
| tréma hongrois, diacritique .....              | 234 |                                              |          |
| trait d'union insécable, caractère .....       | 232 |                                              |          |
| trois-quarts, cadratin .....                   | 232 |                                              |          |

# Index des commandes

## \$

\$ en ligne, math ..... 78  
 \$\$...\$\$ displaymath, forme  
 approchée en T<sub>E</sub>X de base ..... 64

## &

& (pour cellules de tableau) ..... 98

—

--disable-write18 ligne de  
 commande, option ..... 268  
 --enable-write18 ligne de  
 commande, option ..... 268  
 --file-line-error ligne de  
 commande, option ..... 269  
 --halt-on-error ligne de commande, option .. 269  
 --help ligne de commande, option ..... 268  
 --interaction ligne de commande, option .... 268  
 --jobname ligne de commande, option ..... 268  
 --no-file-line-error ligne de  
 commande, option ..... 269  
 --no-shell-escape ligne de  
 commande, option ..... 268  
 --output-directory ligne de  
 commande, option ..... 268  
 --shell-escape ligne de commande, option ... 268  
 --version ligne de commande, option ..... 268

•

.aux fichier ..... 3  
 .aux fichier et commandes BibT<sub>E</sub>X ..... 106  
 .fd fichier ..... 134  
 .fmt, fichier ..... 3  
 .lof fichier ..... 3, 242  
 .lof, fichier ..... 246  
 .lot fichier ..... 3, 242  
 .lot, fichier ..... 246  
 .pdf fichier ..... 3  
 .tex, extension par défaut ..... 267  
 .toc fichier ..... 3  
 .toc, fichier ..... 242, 246  
 .xdv fichier ..... 4

/

/bin/sh, utilisé par \write18 ..... 266

:

: (pour les math) ..... 183

[

[...] (pour les arguments optionnels) ..... 5

^

^ indice supérieur ..... 154

—

\_ indice inférieur ..... 154

\

\ (controbligue-espace) ..... 195

\ caractère de début des commandes ..... 5

\! ..... 179

\" (accent tréma) ..... 233

\# ..... 228

\\$ ..... 228

\% ..... 228

\& ..... 228

\' (accent aigu) ..... 233

\' (tabbing) ..... 94

\(...\) en ligne, math ..... 78

\\* ..... 183

\+ ..... 93

\, ..... 179

\- ..... 93

\- (césure) ..... 113

\. (dot-over accent) ..... 233

\/ ..... 197

\: ..... 178

\; ..... 178

\< ..... 92

\= (macron accent) ..... 233

\= (tabbing) ..... 92

\> ..... 92, 178

\> (tabbing) ..... 92

\[...] displaymath ..... 64

\^ ..... 228

\^ (accent circonflexe) ..... 233

\\_ ..... 228

\' (accent grave) ..... 233

\' (tabbing) ..... 94

\@ ..... 193

\@auxout ..... 262

\@beginparpenalty ..... 77

\@dottedtocline ..... 243

\@endparpenalty ..... 77

\@fnsymbol ..... 119

\@ifstar ..... 128

\@itempenalty ..... 77

\@mainaux ..... 262

\@startsection ..... 51

|                                                       |     |                                                           |        |
|-------------------------------------------------------|-----|-----------------------------------------------------------|--------|
| <code>\l</code> (pour flushleft).....                 | 69  | <code>\AtEndOfClass</code> .....                          | 10     |
| <code>\l</code> (for eqnarray).....                   | 66  | <code>\AtEndOfPackage</code> .....                        | 10     |
| <code>\l</code> (force un saut de ligne).....         | 111 | <code>\author{nom1 \and nom2 \and ...}</code> .....       | 187    |
| <code>\l</code> (pour \author).....                   | 187 | <code>\b</code> (accent barre-souscrite).....             | 233    |
| <code>\l</code> (pour \title).....                    | 188 | <code>\backmatter</code> .....                            | 50     |
| <code>\l</code> (pour center).....                    | 61  | <code>\backslash</code> .....                             | 155    |
| <code>\l</code> (pour flushright).....                | 70  | <code>\bar</code> .....                                   | 176    |
| <code>\l</code> (pour les lettres).....               | 255 | <code>\baselineskip</code> .....                          | 28, 36 |
| <code>\l</code> (pour les objets \shortstack).....    | 88  | <code>\baselinestretch</code> .....                       | 36     |
| <code>\l</code> (pour tabular).....                   | 98  | <code>\begin</code> .....                                 | 59     |
| <code>\l</code> (pour verse).....                     | 110 | <code>\beta</code> .....                                  | 155    |
| <code>\l</code> (tabbing).....                        | 92  | <code>\bf</code> .....                                    | 25     |
| <code>\l*</code> (pour eqnarray).....                 | 66  | <code>\bfseries</code> .....                              | 24     |
| <code>\{</code> .....                                 | 228 | <code>\bibdata</code> .....                               | 106    |
| <code>\}</code> .....                                 | 228 | <code>\bibitem</code> .....                               | 103    |
| <code> </code> .....                                  | 155 | <code>\bibliography</code> .....                          | 105    |
| <code>~</code> .....                                  | 228 | <code>\bibliography et \bibdata interne</code> .....      | 106    |
| <code>˘</code> (accent tilde).....                    | 233 | <code>\bibliographystyle</code> .....                     | 105    |
| <code>\a</code> (tabbing).....                        | 94  | <code>\bibliographystyle et \bibstyle interne</code> .... | 106    |
| <code>\a'</code> (aigu, accent dans tabbing).....     | 94  | <code>\bibname</code> .....                               | 103    |
| <code>\a=</code> (macron, accent dans tabbing).....   | 94  | <code>\bibstyle</code> .....                              | 106    |
| <code>\a'</code> (grave, accent dans tabbing).....    | 94  | <code>\bigbreak</code> .....                              | 199    |
| <code>\aa</code> (å).....                             | 234 | <code>\bigcap</code> .....                                | 155    |
| <code>\AA</code> (Å).....                             | 234 | <code>\bigcirc</code> .....                               | 155    |
| <code>\accent</code> .....                            | 176 | <code>\bigcup</code> .....                                | 155    |
| <code>\acute</code> .....                             | 176 | <code>\bigl</code> .....                                  | 171    |
| <code>\addcontentsline{ext}{unit}{texte}</code> ..... | 244 | <code>\bigodot</code> .....                               | 155    |
| <code>\address</code> .....                           | 256 | <code>\bigoplus</code> .....                              | 155    |
| <code>\addtocontents{ext}{texte}</code> .....         | 245 | <code>\bigotimes</code> .....                             | 155    |
| <code>\addtocounter</code> .....                      | 139 | <code>\bigr</code> .....                                  | 171    |
| <code>\addtolength</code> .....                       | 144 | <code>\bigskip</code> .....                               | 199    |
| <code>\addvspace</code> .....                         | 203 | <code>\bigskipamount</code> .....                         | 199    |
| <code>\ae</code> (æ).....                             | 234 | <code>\bigsqcup</code> .....                              | 156    |
| <code>\AE</code> (Æ).....                             | 234 | <code>\bigtriangledown</code> .....                       | 156    |
| <code>\aleph</code> .....                             | 155 | <code>\bigtriangleup</code> .....                         | 156    |
| <code>\Alph</code> .....                              | 138 | <code>\biguplus</code> .....                              | 156    |
| <code>\alph</code> .....                              | 138 | <code>\bigvee</code> .....                                | 156    |
| <code>\Alph exemple</code> .....                      | 66  | <code>\bigwedge</code> .....                              | 156    |
| <code>\alpha</code> .....                             | 155 | <code>\bmod</code> .....                                  | 174    |
| <code>\alsoname</code> .....                          | 250 | <code>\boldmath</code> .....                              | 167    |
| <code>\amalg</code> .....                             | 155 | <code>\bot</code> .....                                   | 156    |
| <code>\and</code> (pour \author).....                 | 187 | <code>\bottomfraction</code> .....                        | 39     |
| <code>\angle</code> .....                             | 155 | <code>\bottomnumber</code> .....                          | 39     |
| <code>\appendix</code> .....                          | 49  | <code>\bowtie</code> .....                                | 156    |
| <code>\approx</code> .....                            | 155 | <code>\Box</code> .....                                   | 156    |
| <code>\arabic</code> .....                            | 138 | <code>\breve</code> .....                                 | 176    |
| <code>\arccos</code> .....                            | 174 | <code>\bullet</code> .....                                | 156    |
| <code>\arcsin</code> .....                            | 174 | <code>\c</code> (cédille).....                            | 233    |
| <code>\arctan</code> .....                            | 174 | <code>\cal</code> .....                                   | 25     |
| <code>\arg</code> .....                               | 174 | <code>\cap</code> .....                                   | 156    |
| <code>\arraycolsep</code> .....                       | 60  | <code>\capitalacute</code> .....                          | 233    |
| <code>\arrayrulewidth</code> .....                    | 100 | <code>\capitalbreve</code> .....                          | 234    |
| <code>\arraystretch</code> .....                      | 100 | <code>\capitalcaron</code> .....                          | 234    |
| <code>\ast</code> .....                               | 155 | <code>\capitalcedilla</code> .....                        | 233    |
| <code>\asympt</code> .....                            | 155 | <code>\capitalcircumflex</code> .....                     | 233    |
| <code>\AtBeginDocument</code> .....                   | 64  | <code>\capitaldieresis</code> .....                       | 233    |
| <code>\AtBeginDvi</code> .....                        | 10  | <code>\capitaldotaccent</code> .....                      | 234    |
| <code>\AtEndDocument</code> .....                     | 64  | <code>\capitalgrave</code> .....                          | 233    |

|                                               |            |                                                            |        |
|-----------------------------------------------|------------|------------------------------------------------------------|--------|
| <code>\capitalhungarumlaut</code> .....       | 234        | <code>\dblfloatsep</code> .....                            | 31     |
| <code>\capitalmacron</code> .....             | 233        | <code>\dbltextfloatsep</code> .....                        | 31     |
| <code>\capitalnewtie</code> .....             | 234        | <code>\dbltopfraction</code> .....                         | 30     |
| <code>\capitalogonek</code> .....             | 234        | <code>\dbltopnumber</code> .....                           | 31     |
| <code>\capitalring</code> .....               | 234        | <code>\ddag</code> .....                                   | 229    |
| <code>\capitaltie</code> .....                | 234        | <code>\ddagger</code> .....                                | 157    |
| <code>\capitaltilde</code> .....              | 233        | <code>\ddot</code> .....                                   | 176    |
| <code>\caption</code> .....                   | 40, 67, 97 | <code>\ddots</code> .....                                  | 173    |
| <code>\cc</code> .....                        | 256        | <code>\DeclareFontEncoding</code> .....                    | 19     |
| <code>\cdot</code> .....                      | 156        | <code>\DeclareGraphicsExtensions</code> .....              | 218    |
| <code>\cdots</code> .....                     | 173        | <code>\DeclareGraphicsRule</code> .....                    | 218    |
| <code>\centering</code> .....                 | 62         | <code>\DeclareOption</code> .....                          | 11     |
| <code>\chapter</code> .....                   | 42, 44     | <code>\DeclareOption*</code> .....                         | 11     |
| <code>\check</code> .....                     | 176        | <code>\DeclareRobustCommand</code> .....                   | 12     |
| <code>\CheckCommand</code> .....              | 10         | <code>\DeclareRobustCommand*</code> .....                  | 12     |
| <code>\CheckCommand*</code> .....             | 10         | <code>\DeclareTextAccent</code> .....                      | 19     |
| <code>\chi</code> .....                       | 156        | <code>\DeclareTextAccentDefault</code> .....               | 19     |
| <code>\circ</code> .....                      | 156        | <code>\DeclareTextCommand</code> .....                     | 20     |
| <code>\circle</code> .....                    | 86         | <code>\DeclareTextCommandDefault</code> .....              | 21     |
| <code>\citation</code> .....                  | 106        | <code>\DeclareTextComposite</code> .....                   | 21     |
| <code>\cite</code> .....                      | 104        | <code>\DeclareTextCompositeCommand</code> .....            | 22     |
| <code>\cite et \citation interne</code> ..... | 106        | <code>\DeclareTextSymbol</code> .....                      | 20, 22 |
| <code>\ClassError</code> .....                | 11         | <code>\DeclareTextSymbolDefault</code> .....               | 22     |
| <code>\ClassInfo</code> .....                 | 11         | <code>\deg</code> .....                                    | 175    |
| <code>\ClassInfoNoLine</code> .....           | 11         | <code>\delta</code> .....                                  | 157    |
| <code>\ClassWarning</code> .....              | 11         | <code>\Delta</code> .....                                  | 157    |
| <code>\ClassWarningNoLine</code> .....        | 11         | <code>\det</code> .....                                    | 175    |
| <code>\cleardoublepage</code> .....           | 117        | <code>\dh (ð)</code> .....                                 | 235    |
| <code>\clearpage</code> .....                 | 117        | <code>\DH (Ð)</code> .....                                 | 235    |
| <code>\cline</code> .....                     | 102        | <code>\diamond</code> .....                                | 157    |
| <code>\closein</code> .....                   | 260        | <code>\Diamond</code> .....                                | 157    |
| <code>\closeout</code> .....                  | 260        | <code>\diamondsuit</code> .....                            | 157    |
| <code>\closing</code> .....                   | 257        | <code>\dim</code> .....                                    | 175    |
| <code>\clubsuit</code> .....                  | 156        | <code>\displaystyle</code> .....                           | 182    |
| <code>\colon</code> .....                     | 183        | <code>\div</code> .....                                    | 157    |
| <code>\columnsep</code> .....                 | 30, 32     | <code>\dj</code> .....                                     | 235    |
| <code>\columnseprule</code> .....             | 30, 32     | <code>\DJ</code> .....                                     | 235    |
| <code>\columnwidth</code> .....               | 30, 32     | <code>\documentclass</code> .....                          | 7      |
| <code>\complement</code> .....                | 156        | <code>\documentclass, et nom d'ouvrage texput</code> ..... | 273    |
| <code>\cong</code> .....                      | 156        | <code>\dot</code> .....                                    | 176    |
| <code>\contentsline</code> .....              | 244, 246   | <code>\doteq</code> .....                                  | 157    |
| <code>\coprod</code> .....                    | 156        | <code>\dotfill</code> .....                                | 198    |
| <code>\copyright</code> .....                 | 229        | <code>\dots</code> .....                                   | 230    |
| <code>\cos</code> .....                       | 175        | <code>\doublerulesep</code> .....                          | 100    |
| <code>\cosh</code> .....                      | 175        | <code>\downarrow</code> .....                              | 157    |
| <code>\cot</code> .....                       | 175        | <code>\Downarrow</code> .....                              | 157    |
| <code>\coth</code> .....                      | 175        | <code>\ell</code> .....                                    | 157    |
| <code>\csc</code> .....                       | 175        | <code>\em</code> .....                                     | 25     |
| <code>\cup</code> .....                       | 156        | <code>\emph</code> .....                                   | 24     |
| <code>\CurrentOption</code> .....             | 11         | <code>\emptyset</code> .....                               | 157    |
| <code>\d (dot-under accent)</code> .....      | 234        | <code>\encl</code> .....                                   | 257    |
| <code>\dag</code> .....                       | 229        | <code>\end</code> .....                                    | 59     |
| <code>\dagger</code> .....                    | 156        | <code>\endinginput</code> .....                            | 238    |
| <code>\dashbox</code> .....                   | 89         | <code>\enlargethispage</code> .....                        | 118    |
| <code>\dashv</code> .....                     | 156        | <code>\enspace</code> .....                                | 191    |
| <code>\date{texte}</code> .....               | 188        | <code>\enumi</code> .....                                  | 65     |
| <code>\day</code> .....                       | 140        | <code>\enumii</code> .....                                 | 65     |
| <code>\dblfloatpagefraction</code> .....      | 31         | <code>\enumiii</code> .....                                | 65     |

|                                                       |         |                                                                                  |                 |
|-------------------------------------------------------|---------|----------------------------------------------------------------------------------|-----------------|
| <code>\enumiv</code> .....                            | 65      | <code>\H</code> (tréma hongrois) .....                                           | 234             |
| <code>\epsilon</code> .....                           | 157     | <code>\hat</code> .....                                                          | 176             |
| <code>\equiv</code> .....                             | 157     | <code>\hbar</code> .....                                                         | 158             |
| <code>\ESPACE</code> .....                            | 195     | <code>\headheight</code> .....                                                   | 32              |
| <code>\eta</code> .....                               | 157     | <code>\headsep</code> .....                                                      | 33              |
| <code>\evensidemargin</code> .....                    | 8, 33   | <code>\heartsuit</code> .....                                                    | 158             |
| <code>\ExecuteOptions</code> .....                    | 13      | <code>\hfill</code> .....                                                        | 192             |
| <code>\exists</code> .....                            | 157     | <code>\hline</code> .....                                                        | 102             |
| <code>\exp</code> .....                               | 175     | <code>\hom</code> .....                                                          | 175             |
| <code>\externaldocument</code> .....                  | 57      | <code>\hookleftarrow</code> .....                                                | 158             |
| <code>\extracolsep</code> .....                       | 99      | <code>\hookrightarrow</code> .....                                               | 158             |
| <code>\fbox</code> .....                              | 205     | <code>\hphantom</code> .....                                                     | 180             |
| <code>\fboxrule</code> .....                          | 88, 206 | <code>\hrulefill</code> .....                                                    | 198             |
| <code>\fboxsep</code> .....                           | 88, 206 | <code>\hspace</code> .....                                                       | 35              |
| <code>\fill</code> .....                              | 192     | <code>\hspace</code> .....                                                       | 191             |
| <code>\flat</code> .....                              | 157     | <code>\hss</code> .....                                                          | 192             |
| <code>\floatpagefraction</code> .....                 | 39      | <code>\huge</code> .....                                                         | 26              |
| <code>\floatsep</code> .....                          | 39      | <code>\Huge</code> .....                                                         | 26              |
| <code>\flushbottom</code> .....                       | 32      | <code>\hyphenation</code> .....                                                  | 115             |
| <code>\fnsymbol</code> .....                          | 138     | <code>\i</code> (i sans point) .....                                             | 234             |
| <code>\fnsymbol</code> , et note en bas de page ..... | 119     | <code>\IfBeginWith*</code> macro from<br><code>@codexstring@comment</code> ..... | 272             |
| <code>\fontdimen1</code> .....                        | 198     | <code>\iff</code> .....                                                          | 158             |
| <code>\fontencoding</code> .....                      | 27      | <code>\IfFileExists</code> .....                                                 | 12              |
| <code>\fontfamily</code> .....                        | 27      | <code>\ignorespaces</code> .....                                                 | 135             |
| <code>\fontseries</code> .....                        | 27      | <code>\ignorespacesafterend</code> .....                                         | 135             |
| <code>\fontshape</code> .....                         | 28      | <code>\ij</code> (ij) .....                                                      | 235             |
| <code>\fontsize</code> .....                          | 28      | <code>\IJ</code> (IJ) .....                                                      | 235             |
| <code>\footnote</code> .....                          | 119     | <code>\il</code> .....                                                           | 25              |
| <code>\footnotemark</code> .....                      | 120     | <code>\Im</code> .....                                                           | 158             |
| <code>\footnoterule</code> .....                      | 122     | <code>\imath</code> .....                                                        | 158             |
| <code>\footnotesep</code> .....                       | 123     | <code>\immediate\write</code> .....                                              | 263             |
| <code>\footnotesize</code> .....                      | 26      | <code>\in</code> .....                                                           | 158             |
| <code>\footnotetext</code> .....                      | 120     | <code>\include</code> .....                                                      | 239             |
| <code>\footskip</code> .....                          | 33      | <code>\includegraphics</code> .....                                              | 220             |
| <code>\forall</code> .....                            | 157     | <code>\includeonly</code> .....                                                  | 239             |
| <code>\frac</code> .....                              | 183     | <code>\indent</code> .....                                                       | 150             |
| <code>\frame</code> .....                             | 89      | <code>\index</code> .....                                                        | 248, 249        |
| <code>\framebox</code> .....                          | 88, 205 | <code>\indexentry</code> .....                                                   | 251             |
| <code>\frenchspacing</code> .....                     | 194     | <code>\indexspace</code> .....                                                   | 249, 252        |
| <code>\frontmatter</code> .....                       | 50      | <code>\inf</code> .....                                                          | 175             |
| <code>\frown</code> .....                             | 157     | <code>\infty</code> .....                                                        | 158             |
| <code>\fussy</code> .....                             | 115     | <code>\input</code> .....                                                        | 241             |
| <code>\gamma</code> .....                             | 157     | <code>\inputencoding</code> .....                                                | 236             |
| <code>\Gamma</code> .....                             | 157     | <code>\InputIfFileExists</code> .....                                            | 12              |
| <code>\gcd</code> .....                               | 175     | <code>\int</code> .....                                                          | 158             |
| <code>\ge</code> .....                                | 157     | <code>\intertextsep</code> .....                                                 | 39              |
| <code>\geq</code> .....                               | 157     | <code>\iota</code> .....                                                         | 158             |
| <code>\gets</code> .....                              | 157     | <code>\item</code> .....                                                         | 63, 65, 71, 249 |
| <code>\gg</code> .....                                | 157     | <code>\itemindent</code> .....                                                   | 74              |
| <code>\glossary</code> .....                          | 247     | <code>\itemsep</code> .....                                                      | 74              |
| <code>\glossaryentry</code> .....                     | 247     | <code>\itshape</code> .....                                                      | 24              |
| <code>\graphicspath</code> .....                      | 217     | <code>\j</code> (j sans point) .....                                             | 234             |
| <code>\graphpaper</code> .....                        | 85      | <code>\jmath</code> .....                                                        | 158             |
| <code>\grave</code> .....                             | 176     | <code>\jobname</code> .....                                                      | 272             |
| <code>\guillemotleft</code> («) .....                 | 230     | <code>\Join</code> .....                                                         | 158             |
| <code>\guillemotright</code> (») .....                | 230     | <code>\k</code> (ogonek) .....                                                   | 234             |
| <code>\guilsinglleft</code> (<) .....                 | 230     | <code>\kappa</code> .....                                                        | 158             |
| <code>\guilsinglright</code> (>) .....                | 230     |                                                                                  |                 |

|                                          |          |                                                                   |          |
|------------------------------------------|----------|-------------------------------------------------------------------|----------|
| <code>\ker</code> .....                  | 175      | <code>\lineskip</code> .....                                      | 36       |
| <code>\kill</code> .....                 | 94       | <code>\lineskiplimit</code> .....                                 | 36       |
| <code>\l (l)</code> .....                | 235      | <code>\linespread</code> .....                                    | 29, 36   |
| <code>\l@chapter</code> .....            | 246      | <code>\linethickness</code> .....                                 | 86       |
| <code>\l@section</code> .....            | 246      | <code>\linewidth</code> .....                                     | 33       |
| <code>\l@subsection</code> .....         | 246      | <code>\listoffigures</code> .....                                 | 242, 246 |
| <code>\L (L)</code> .....                | 235      | <code>\listoftables</code> .....                                  | 242, 246 |
| <code>\label</code> .....                | 55       | <code>\listparindent</code> .....                                 | 75       |
| <code>\labelenumi</code> .....           | 65       | <code>\ll</code> .....                                            | 159      |
| <code>\labelenumii</code> .....          | 65       | <code>\ln</code> .....                                            | 175      |
| <code>\labelenumiii</code> .....         | 65       | <code>\lnot</code> .....                                          | 159      |
| <code>\labelenumiv</code> .....          | 65       | <code>\LoadClass</code> .....                                     | 13       |
| <code>\labelitemi</code> .....           | 71       | <code>\LoadClassWithOptions</code> .....                          | 13       |
| <code>\labelitemii</code> .....          | 71       | <code>\location</code> .....                                      | 257      |
| <code>\labelitemiii</code> .....         | 71       | <code>\log</code> .....                                           | 175      |
| <code>\labelitemiv</code> .....          | 71       | <code>\long</code> .....                                          | 124      |
| <code>\labelsep</code> .....             | 74       | <code>\longleftarrow</code> .....                                 | 159      |
| <code>\labelwidth</code> .....           | 74       | <code>\longlefttrightarrow</code> .....                           | 159      |
| <code>\Lambda</code> .....               | 158      | <code>\longmapsto</code> .....                                    | 159      |
| <code>\lambda</code> .....               | 158      | <code>\longrightarrow</code> .....                                | 159      |
| <code>\land</code> .....                 | 158      | <code>\lor</code> .....                                           | 159      |
| <code>\langle</code> .....               | 158      | <code>\lq</code> .....                                            | 230      |
| <code>\large</code> .....                | 26       | <code>\mainmatter</code> .....                                    | 50       |
| <code>\Large</code> .....                | 26       | <code>\makeatother</code> .....                                   | 127      |
| <code>\LARGE</code> .....                | 26       | <code>\makebox</code> .....                                       | 204      |
| <code>\LastDeclaredEncoding</code> ..... | 23       | <code>\makebox (picture)</code> .....                             | 88       |
| <code>\LaTeX</code> .....                | 229      | <code>\makeglossary</code> .....                                  | 247      |
| <code>\LaTeXe</code> .....               | 229      | <code>\makeindex</code> .....                                     | 248      |
| <code>\lbrace</code> .....               | 158      | <code>\makelabel</code> .....                                     | 73       |
| <code>\lbrack</code> .....               | 158      | <code>\makelabels</code> .....                                    | 257      |
| <code>\lceil</code> .....                | 158      | <code>\maketitle</code> .....                                     | 187      |
| <code>\ldots</code> .....                | 173, 230 | <code>\mapsto</code> .....                                        | 159      |
| <code>\le</code> .....                   | 158      | <code>\marginpar</code> .....                                     | 151      |
| <code>\leadsto</code> .....              | 158      | <code>\marginparpush</code> .....                                 | 33, 151  |
| <code>\left</code> .....                 | 169      | <code>\marginparsep</code> .....                                  | 151      |
| <code>\leftarrow</code> .....            | 159      | <code>\marginparwidth</code> .....                                | 33, 152  |
| <code>\Leftarrow</code> .....            | 158      | <code>\marginsep</code> .....                                     | 33       |
| <code>\lefteqn</code> .....              | 66       | <code>\markboth{en-tête-<br/>gauche}{en-tête-droite}</code> ..... | 190      |
| <code>\leftharpoondown</code> .....      | 159      | <code>\markright{en-tête-droite}</code> .....                     | 190      |
| <code>\leftharpoonup</code> .....        | 159      | <code>\mathbf</code> .....                                        | 25       |
| <code>\leftmargin</code> .....           | 72, 74   | <code>\mathcal</code> .....                                       | 25       |
| <code>\leftmargini</code> .....          | 72       | <code>\mathdollar</code> .....                                    | 166      |
| <code>\leftmarginii</code> .....         | 72       | <code>\mathnormal</code> .....                                    | 25       |
| <code>\leftmarginiii</code> .....        | 72       | <code>\mathparagraph</code> .....                                 | 166      |
| <code>\leftmarginiv</code> .....         | 72       | <code>\mathring</code> .....                                      | 176      |
| <code>\leftmarginv</code> .....          | 72       | <code>\mathrm</code> .....                                        | 25       |
| <code>\leftmarginvi</code> .....         | 72       | <code>\mathsection</code> .....                                   | 166      |
| <code>\leftrightarrow</code> .....       | 159      | <code>\mathsf</code> .....                                        | 25       |
| <code>\Leftrightarrow</code> .....       | 159      | <code>\mathsterling</code> .....                                  | 166      |
| <code>\leq</code> .....                  | 159      | <code>\mathstrut</code> .....                                     | 181      |
| <code>\lfloor</code> .....               | 159      | <code>\mathtt</code> .....                                        | 25       |
| <code>\lg</code> .....                   | 175      | <code>\mathunderscore</code> .....                                | 166      |
| <code>\lhd</code> .....                  | 159      | <code>\mathversion</code> .....                                   | 25       |
| <code>\lim</code> .....                  | 175      | <code>\max</code> .....                                           | 175      |
| <code>\liminf</code> .....               | 175      | <code>\mbox</code> .....                                          | 204      |
| <code>\limsup</code> .....               | 175      | <code>\mdseries</code> .....                                      | 24       |
| <code>\line</code> .....                 | 85       | <code>\medbreak</code> .....                                      | 199      |
| <code>\linebreak</code> .....            | 116      |                                                                   |          |

|                                                       |          |                                                  |         |
|-------------------------------------------------------|----------|--------------------------------------------------|---------|
| <code>\medskip</code> .....                           | 199      | <code>\o (<math>\emptyset</math>)</code> .....   | 235     |
| <code>\medskipamount</code> .....                     | 199      | <code>\O (<math>\emptyset</math>)</code> .....   | 235     |
| <code>\medspace</code> .....                          | 178      | <code>\obeycr</code> .....                       | 112     |
| <code>\message</code> .....                           | 264      | <code>\oddsidemargin</code> .....                | 8, 33   |
| <code>\mho</code> .....                               | 159      | <code>\odot</code> .....                         | 160     |
| <code>\mid</code> .....                               | 160      | <code>\oe (<math>\text{\oe}</math>)</code> ..... | 235     |
| <code>\min</code> .....                               | 175      | <code>\OE (<math>\text{\OE}</math>)</code> ..... | 235     |
| <code>\models</code> .....                            | 160      | <code>\oint</code> .....                         | 160     |
| <code>\month</code> .....                             | 140      | <code>\oldstylenums</code> .....                 | 25      |
| <code>\mp</code> .....                                | 160      | <code>\omega</code> .....                        | 160     |
| <code>\mu</code> .....                                | 160      | <code>\Omega</code> .....                        | 160     |
| <code>\multicolumn</code> .....                       | 100      | <code>\ominus</code> .....                       | 160     |
| <code>\multirow</code> .....                          | 84       | <code>\onecolumn</code> .....                    | 30      |
| <code>\nabla</code> .....                             | 160      | <code>\openin</code> .....                       | 260     |
| <code>\name</code> .....                              | 258      | <code>\opening</code> .....                      | 258     |
| <code>\natural</code> .....                           | 160      | <code>\openout</code> .....                      | 260     |
| <code>\ne</code> .....                                | 160      | <code>\oplus</code> .....                        | 160     |
| <code>\nearrow</code> .....                           | 160      | <code>\OptionNotUsed</code> .....                | 14      |
| <code>\NeedsTeXFormat</code> .....                    | 13       | <code>\oslash</code> .....                       | 160     |
| <code>\neg</code> .....                               | 160      | <code>\otimes</code> .....                       | 160     |
| <code>\negmedspace</code> .....                       | 178      | <code>\oval</code> .....                         | 87      |
| <code>\negthickspace</code> .....                     | 178      | <code>\overbrace{<i>math</i>}</code> .....       | 178     |
| <code>\negthinspace</code> .....                      | 179, 197 | <code>\overline{<i>texte</i>}</code> .....       | 177     |
| <code>\neq</code> .....                               | 160      | <code>\owns</code> .....                         | 161     |
| <code>\newcommand</code> .....                        | 124      | <code>\PackageError</code> .....                 | 11      |
| <code>\newcounter</code> .....                        | 129      | <code>\PackageInfo</code> .....                  | 11      |
| <code>\newenvironment</code> .....                    | 130      | <code>\PackageInfoNoLine</code> .....            | 11      |
| <code>\newfont</code> .....                           | 134      | <code>\PackageWarning</code> .....               | 11      |
| <code>\newlength</code> .....                         | 129      | <code>\PackageWarningNoLine</code> .....         | 11      |
| <code>\newline</code> .....                           | 113      | <code>\pagebreak</code> .....                    | 118     |
| <code>\newpage</code> .....                           | 117      | <code>\pagenumbering</code> .....                | 188     |
| <code>\newsavebox</code> .....                        | 130      | <code>\pageref</code> .....                      | 56      |
| <code>\newtheorem</code> .....                        | 132      | <code>\pagestyle</code> .....                    | 189     |
| <code>\newtie</code> .....                            | 234      | <code>\paperheight</code> .....                  | 34      |
| <code>\newwrite</code> .....                          | 262      | <code>\paperwidth</code> .....                   | 34      |
| <code>\ng</code> .....                                | 235      | <code>\par</code> .....                          | 149     |
| <code>\NG</code> .....                                | 235      | <code>\paragraph</code> .....                    | 42, 48  |
| <code>\ni</code> .....                                | 160      | <code>\parallel</code> .....                     | 161     |
| <code>\nocite</code> .....                            | 105      | <code>\parbox</code> .....                       | 206     |
| <code>\nocite {*}</code> , pour toutes les clefs..... | 106      | <code>\parindent</code> .....                    | 80, 150 |
| <code>\nocite et \citation interne</code> .....       | 106      | <code>\parsep</code> .....                       | 75      |
| <code>\nocorr</code> .....                            | 24       | <code>\parskip</code> .....                      | 150     |
| <code>\nocorrlist</code> .....                        | 24       | <code>\parskip exemple</code> .....              | 72      |
| <code>\nofiles</code> .....                           | 247      | <code>\part</code> .....                         | 42, 43  |
| <code>\noindent</code> .....                          | 150      | <code>\partial</code> .....                      | 161     |
| <code>\nolinebreak</code> .....                       | 116      | <code>\partopsep</code> .....                    | 75      |
| <code>\nonfrenchspacing</code> .....                  | 194      | <code>\PassOptionsToClass</code> .....           | 14      |
| <code>\nonumber</code> .....                          | 66       | <code>\PassOptionsToPackage</code> .....         | 14      |
| <code>\nopagebreak</code> .....                       | 118      | <code>\pdfpageheight</code> .....                | 8       |
| <code>\normalfont</code> .....                        | 24       | <code>\pdfpagewidth</code> .....                 | 8       |
| <code>\normalmarginpar</code> .....                   | 151      | <code>\perp</code> .....                         | 161     |
| <code>\normalsfcodes</code> .....                     | 195      | <code>\phantom</code> .....                      | 180     |
| <code>\normalsize</code> .....                        | 26       | <code>\phi</code> .....                          | 161     |
| <code>\not</code> .....                               | 160      | <code>\Phi</code> .....                          | 161     |
| <code>\notin</code> .....                             | 160      | <code>\pi</code> .....                           | 161     |
| <code>\nu</code> .....                                | 160      | <code>\Pi</code> .....                           | 161     |
| <code>\numberline</code> .....                        | 247      | <code>\pm</code> .....                           | 161     |
| <code>\narrow</code> .....                            | 160      | <code>\pmod</code> .....                         | 175     |

|                                                           |          |                                              |        |
|-----------------------------------------------------------|----------|----------------------------------------------|--------|
| <code>\poptabs</code> .....                               | 94       | <code>\rho</code> .....                      | 162    |
| <code>\pounds</code> .....                                | 230      | <code>\right</code> .....                    | 169    |
| <code>\Pr</code> .....                                    | 175      | <code>\Rightarrow</code> .....               | 162    |
| <code>\prec</code> .....                                  | 161      | <code>\rightarrow</code> .....               | 162    |
| <code>\preceq</code> .....                                | 161      | <code>\rightharpoondown</code> .....         | 162    |
| <code>\prevdepth</code> .....                             | 36       | <code>\rightharpoonup</code> .....           | 162    |
| <code>\prime</code> .....                                 | 161      | <code>\rightleftharpoons</code> .....        | 162    |
| <code>\printindex</code> .....                            | 254      | <code>\rightmargin</code> .....              | 75     |
| <code>\ProcessOptions</code> .....                        | 14       | <code>\rm</code> .....                       | 25     |
| <code>\ProcessOptions*</code> .....                       | 14       | <code>\rmfamily</code> .....                 | 24     |
| <code>\prod</code> .....                                  | 161      | <code>\Roman</code> .....                    | 138    |
| <code>\propto</code> .....                                | 161      | <code>\roman</code> .....                    | 138    |
| <code>\protect</code> .....                               | 134      | <code>\rotatebox</code> .....                | 225    |
| <code>\protected@write</code> .....                       | 263      | <code>\rq</code> .....                       | 230    |
| <code>\providecommand</code> .....                        | 127      | <code>\rule</code> .....                     | 236    |
| <code>\ProvidesClass</code> .....                         | 15       | <code>\savebox</code> .....                  | 208    |
| <code>\ProvidesFile</code> .....                          | 16       | <code>\sbox</code> .....                     | 208    |
| <code>\ProvidesPackage</code> .....                       | 15       | <code>\sc</code> .....                       | 25     |
| <code>\ProvideTextCommand</code> .....                    | 20       | <code>\scalebox</code> .....                 | 226    |
| <code>\ProvideTextCommandDefault</code> .....             | 21       | <code>\scriptscriptstyle</code> .....        | 182    |
| <code>\ps</code> .....                                    | 258      | <code>\scriptsize</code> .....               | 26     |
| <code>\psi</code> .....                                   | 161      | <code>\scriptstyle</code> .....              | 182    |
| <code>\Psi</code> .....                                   | 161      | <code>\scshape</code> .....                  | 24     |
| <code>\pushtabs</code> .....                              | 94       | <code>\searrow</code> .....                  | 162    |
| <code>\put</code> .....                                   | 83       | <code>\sec</code> .....                      | 175    |
| <code>\P</code> .....                                     | 230      | <code>\section</code> .....                  | 42, 46 |
| <code>\qbezier</code> .....                               | 84       | <code>\seename</code> .....                  | 250    |
| <code>\qquad</code> .....                                 | 179, 191 | <code>\selectfont</code> .....               | 29     |
| <code>\quad</code> .....                                  | 179, 191 | <code>\setcounter</code> .....               | 139    |
| <code>\quotedblbase (,,)</code> .....                     | 230      | <code>\setlength</code> .....                | 143    |
| <code>\quotesinglbase (,)</code> .....                    | 230      | <code>\setminus</code> .....                 | 162    |
| <code>\r (ring accent)</code> .....                       | 234      | <code>\settodepth</code> .....               | 144    |
| <code>\raggedbottom</code> .....                          | 32       | <code>\settoheight</code> .....              | 145    |
| <code>\raggedleft</code> .....                            | 70       | <code>\settowidth</code> .....               | 145    |
| <code>\raggedright</code> .....                           | 69       | <code>\sf</code> .....                       | 25     |
| <code>\raisebox</code> .....                              | 207      | <code>\sffamily</code> .....                 | 24     |
| <code>\rangle</code> .....                                | 161      | <code>\sharp</code> .....                    | 162    |
| <code>\rbrace</code> .....                                | 161      | <code>\shipout et développement</code> ..... | 263    |
| <code>\rbrack</code> .....                                | 161      | <code>\shortstack</code> .....               | 87     |
| <code>\rceil</code> .....                                 | 161      | <code>\sigma</code> .....                    | 162    |
| <code>\Re</code> .....                                    | 161      | <code>\Sigma</code> .....                    | 162    |
| <code>\read</code> .....                                  | 261      | <code>\signature</code> .....                | 258    |
| <code>\ref</code> .....                                   | 57       | <code>\sim</code> .....                      | 162    |
| <code>\reflectbox</code> .....                            | 226      | <code>\simeq</code> .....                    | 162    |
| <code>\refname</code> .....                               | 103      | <code>\sin</code> .....                      | 175    |
| <code>\refstepcounter</code> .....                        | 139      | <code>\sinh</code> .....                     | 175    |
| <code>\renewcommand</code> .....                          | 124      | <code>\sl</code> .....                       | 25     |
| <code>\renewenvironment</code> .....                      | 130      | <code>\sloppy</code> .....                   | 115    |
| <code>\RequirePackage</code> .....                        | 16       | <code>\slshape</code> .....                  | 24     |
| <code>\RequirePackage, et nom d'ouvrage texput</code> ... | 273      | <code>\small</code> .....                    | 26     |
| <code>\RequirePackageWithOptions</code> .....             | 16       | <code>\smallbreak</code> .....               | 199    |
| <code>\resizebox</code> .....                             | 226      | <code>\smallint</code> .....                 | 162    |
| <code>\restorecr</code> .....                             | 112      | <code>\smallskip</code> .....                | 199    |
| <code>\restriction</code> .....                           | 161      | <code>\smallskipamount</code> .....          | 199    |
| <code>\revemptyset</code> .....                           | 162      | <code>\smile</code> .....                    | 162    |
| <code>\reversemarginpar</code> .....                      | 151      | <code>\spacefactor</code> .....              | 192    |
| <code>\rfloor</code> .....                                | 162      | <code>\spadesuit</code> .....                | 163    |
| <code>\rhd</code> .....                                   | 162      | <code>\sqcap</code> .....                    | 163    |

|                                                      |          |                                                            |          |
|------------------------------------------------------|----------|------------------------------------------------------------|----------|
| <code>\sqcup</code> .....                            | 163      | <code>\textellipsis</code> .....                           | 230      |
| <code>\sqrt</code> .....                             | 184      | <code>\textemdash</code> (ou <code>---</code> ).....       | 231      |
| <code>\sqsubset</code> .....                         | 163      | <code>\textendash</code> (ou <code>--</code> ).....        | 231      |
| <code>\sqsubseteq</code> .....                       | 163      | <code>\texteuro</code> .....                               | 231      |
| <code>\sqsupset</code> .....                         | 163      | <code>\textexclamdown</code> (ou <code>!'</code> ).....    | 231      |
| <code>\sqsupseteq</code> .....                       | 163      | <code>\textfiguredash</code> .....                         | 231      |
| <code>\ss</code> ( $\beta$ ).....                    | 235      | <code>\textfloatsep</code> .....                           | 39       |
| <code>\SS</code> ( <code>SS@comment</code> ).....    | 235      | <code>\textfraction</code> .....                           | 39       |
| <code>\stackrel</code> .....                         | 184      | <code>\textgreater</code> .....                            | 231      |
| <code>\star</code> .....                             | 163      | <code>\textheight</code> .....                             | 34       |
| <code>\stepcounter</code> .....                      | 139      | <code>\texthorizontalbar</code> .....                      | 231      |
| <code>\stop</code> .....                             | 274      | <code>\textit</code> .....                                 | 24       |
| <code>\strut</code> .....                            | 200      | <code>\textleftarrow</code> .....                          | 232      |
| <code>\subitem</code> .....                          | 249      | <code>\textless</code> .....                               | 232      |
| <code>\subparagraph</code> .....                     | 42, 48   | <code>\textmd</code> .....                                 | 24       |
| <code>\subsection</code> .....                       | 42, 47   | <code>\textnonbreakinghyphen</code> .....                  | 232      |
| <code>\subset</code> .....                           | 163      | <code>\textnormal</code> .....                             | 24       |
| <code>\subseteq</code> .....                         | 163      | <code>\textordfeminine</code> .....                        | 232      |
| <code>\subsubitem</code> .....                       | 249      | <code>\textordmasculine</code> .....                       | 232      |
| <code>\subsubsection</code> .....                    | 42, 48   | <code>\textparagraph</code> .....                          | 230      |
| <code>\succ</code> .....                             | 163      | <code>\textperiodcentered</code> .....                     | 232      |
| <code>\succeq</code> .....                           | 163      | <code>\textquestiondown</code> (ou <code>?'</code> ).....  | 232      |
| <code>\sum</code> .....                              | 163      | <code>\textquotedblleft</code> (ou <code>'</code> ).....   | 232      |
| <code>\sup</code> .....                              | 175      | <code>\textquotedblright</code> (ou <code>''</code> )..... | 232      |
| <code>\suppressfloats</code> .....                   | 39       | <code>\textquoteleft</code> (ou <code>'</code> ).....      | 232      |
| <code>\supset</code> .....                           | 163      | <code>\textquoteright</code> (ou <code>'</code> ).....     | 232      |
| <code>\supseteq</code> .....                         | 163      | <code>\textquotesingle</code> .....                        | 232      |
| <code>\surd</code> .....                             | 163      | <code>\textquotestraightbase</code> .....                  | 232      |
| <code>\swarrow</code> .....                          | 163      | <code>\textquotestraightdblbase</code> .....               | 232      |
| <code>\symbol</code> .....                           | 229      | <code>\textregistered</code> .....                         | 232      |
| <code>\S</code> .....                                | 230      | <code>\textrightarrow</code> .....                         | 232      |
| <code>\t</code> (tie-after accent).....              | 234      | <code>\textrm</code> .....                                 | 24       |
| <code>\tabbingsep</code> .....                       | 94       | <code>\textsc</code> .....                                 | 24       |
| <code>\tabcolsep</code> .....                        | 100      | <code>\textsf</code> .....                                 | 24       |
| <code>\tableofcontents</code> .....                  | 242, 246 | <code>\textsl</code> .....                                 | 24       |
| <code>\tan</code> .....                              | 175      | <code>\textsterling</code> .....                           | 230      |
| <code>\tanh</code> .....                             | 175      | <code>\textstyle</code> .....                              | 182      |
| <code>\tau</code> .....                              | 163      | <code>\textthreequartersemdash</code> .....                | 232      |
| <code>\telephone</code> .....                        | 259      | <code>\texttrademark</code> .....                          | 232      |
| <code>\textascendercompwordmark</code> .....         | 231      | <code>\texttt</code> .....                                 | 24       |
| <code>\textasciicircum</code> .....                  | 230      | <code>\texttwelveudash</code> .....                        | 232      |
| <code>\textasciitilde</code> .....                   | 230      | <code>\textunderscore</code> .....                         | 233      |
| <code>\textasteriskcentered</code> .....             | 230      | <code>\textup</code> .....                                 | 24       |
| <code>\textbackslash</code> .....                    | 228, 230 | <code>\textvisiblespace</code> .....                       | 233      |
| <code>\textbar</code> .....                          | 230      | <code>\textwidth</code> .....                              | 35       |
| <code>\textbardbl</code> .....                       | 230      | <code>\TeX</code> .....                                    | 230      |
| <code>\textbf</code> .....                           | 24       | <code>\th</code> (p).....                                  | 235      |
| <code>\textbigcircle</code> .....                    | 230      | <code>\TH</code> (P).....                                  | 235      |
| <code>\textbraceleft</code> .....                    | 230      | <code>\thanks{<i>texte</i>}</code> .....                   | 188      |
| <code>\textbraceright</code> .....                   | 231      | <code>\theta</code> .....                                  | 163      |
| <code>\textbullet</code> .....                       | 231      | <code>\thicklines</code> .....                             | 86       |
| <code>\textcapitalcompwordmark</code> .....          | 231      | <code>\thickspace</code> .....                             | 178      |
| <code>\textcircled{<i>lettre</i>}</code> .....       | 231      | <code>\thinlines</code> .....                              | 86       |
| <code>\textcompwordmark</code> .....                 | 231      | <code>\thinspace</code> .....                              | 179, 197 |
| <code>\textcopyright</code> .....                    | 229      | <code>\thispagestyle</code> .....                          | 190      |
| <code>\textdagger</code> .....                       | 231      | <code>\tilde</code> .....                                  | 176      |
| <code>\textdaggerdbl</code> .....                    | 231      | <code>\times</code> .....                                  | 164      |
| <code>\textdollar</code> (ou <code>\\$</code> )..... | 231      | <code>\tiny</code> .....                                   | 26       |

|                                                                                                                                            |     |                                                  |                                                                |
|--------------------------------------------------------------------------------------------------------------------------------------------|-----|--------------------------------------------------|----------------------------------------------------------------|
| <code>\title{texte}</code> .....                                                                                                           | 188 | <code>\verb</code> .....                         | 109                                                            |
| <code>\to</code> .....                                                                                                                     | 164 | <code>\Vert</code> .....                         | 165                                                            |
| <code>\today</code> .....                                                                                                                  | 236 | <code>\vert</code> .....                         | 166                                                            |
| <code>\top</code> .....                                                                                                                    | 164 | <code>\vfill</code> .....                        | 202                                                            |
| <code>\topfraction</code> .....                                                                                                            | 39  | <code>\vline</code> .....                        | 102                                                            |
| <code>\topmargin</code> .....                                                                                                              | 35  | <code>\vphantom</code> .....                     | 180                                                            |
| <code>\topnumber</code> .....                                                                                                              | 39  | <code>\vspace</code> .....                       | 201                                                            |
| <code>\topsep</code> .....                                                                                                                 | 75  | <code>\vtop</code> T <sub>E</sub> X de base..... | 79                                                             |
| <code>\topskip</code> .....                                                                                                                | 35  | <code>\wedge</code> .....                        | 166                                                            |
| <code>\totalnumber</code> .....                                                                                                            | 40  | <code>\widehat</code> .....                      | 176                                                            |
| <code>\triangle</code> .....                                                                                                               | 164 | <code>\widetilde</code> .....                    | 176                                                            |
| <code>\triangleleft</code> .....                                                                                                           | 164 | <code>\wlog</code> .....                         | 265                                                            |
| <code>\triangleright</code> .....                                                                                                          | 164 | <code>\wp</code> .....                           | 166                                                            |
| <code>\tt</code> .....                                                                                                                     | 25  | <code>\wr</code> .....                           | 166                                                            |
| <code>\ttfamily</code> .....                                                                                                               | 24  | <code>\write</code> .....                        | 262                                                            |
| <code>\twocolumn</code> .....                                                                                                              | 30  | <code>\write18</code> .....                      | 265                                                            |
| <code>\typein</code> .....                                                                                                                 | 261 | <code>\Xi</code> .....                           | 166                                                            |
| <code>\typeout</code> .....                                                                                                                | 261 | <code>\xi</code> .....                           | 166                                                            |
| <code>\u</code> (brève, accent).....                                                                                                       | 234 | <code>\xspace</code> .....                       | 136                                                            |
| <code>\unboldmath</code> .....                                                                                                             | 167 | <code>\year</code> .....                         | 140                                                            |
| <code>\underbar</code> .....                                                                                                               | 234 | <code>\zeta</code> .....                         | 166                                                            |
| <code>\underbrace{math}</code> .....                                                                                                       | 177 |                                                  |                                                                |
| <code>\underline{texte}</code> .....                                                                                                       | 177 | {                                                |                                                                |
| <code>\unitlength</code> .....                                                                                                             | 82  | {...} (pour les arguments obligatoires) .....    | 5                                                              |
| <code>\unlhd</code> .....                                                                                                                  | 164 | ~                                                |                                                                |
| <code>\unrhd</code> .....                                                                                                                  | 164 | ~ .....                                          | 196                                                            |
| <code>\uparrow</code> .....                                                                                                                | 164 |                                                  |                                                                |
| <code>\Uparrow</code> .....                                                                                                                | 164 | <b>A</b>                                         |                                                                |
| <code>\Updownarrow</code> .....                                                                                                            | 164 | abstract, environnement.....                     | 59                                                             |
| <code>\updownarrow</code> .....                                                                                                            | 164 | adjustbox, paquetage.....                        | 204                                                            |
| <code>\upharpoonright</code> .....                                                                                                         | 164 | algorithm2e, paquetage.....                      | 90                                                             |
| <code>\uplus</code> .....                                                                                                                  | 165 | amscd, paquetage.....                            | 167                                                            |
| <code>\upshape</code> .....                                                                                                                | 24  | amsfonts, paquetage.....                         | 153, 166                                                       |
| <code>\upsilon</code> .....                                                                                                                | 165 | amsmath, paquetage..                             | 60, 61, 64, 67, 107, 153, 170,<br>172, 173, 175, 181, 183, 197 |
| <code>\Upsilon</code> .....                                                                                                                | 165 | amsthm, paquetage.....                           | 107                                                            |
| <code>\usebox</code> .....                                                                                                                 | 210 | answers, paquetage.....                          | 263                                                            |
| <code>\usecounter</code> .....                                                                                                             | 138 | appendix, paquetage.....                         | 50                                                             |
| <code>\usefont</code> .....                                                                                                                | 29  | arobe.....                                       | 193                                                            |
| <code>\usepackage</code> .....                                                                                                             | 9   | array, environnement.....                        | 60                                                             |
| <code>\UseTextAccent</code> .....                                                                                                          | 23  | array, paquetage.....                            | 61                                                             |
| <code>\UseTextSymbol</code> .....                                                                                                          | 23  | Asymptote, paquetage.....                        | 86, 201, 205, 265                                              |
| <code>\v</code> (brève, accent).....                                                                                                       | 234 |                                                  |                                                                |
| <code>\value</code> .....                                                                                                                  | 139 | <b>B</b>                                         |                                                                |
| <code>\vanotthing</code> .....                                                                                                             | 165 | babel, paquetage.....                            | 45, 103, 116, 243, 250                                         |
| <code>\varepsilon</code> .....                                                                                                             | 165 | beamer, paquetage.....                           | 275                                                            |
| <code>\varphi</code> .....                                                                                                                 | 165 | bigfoot, paquetage.....                          | 122                                                            |
| <code>\varpi</code> .....                                                                                                                  | 165 | bm, paquetage.....                               | 168                                                            |
| <code>\varrho</code> .....                                                                                                                 | 165 | bp.....                                          | 142                                                            |
| <code>\varsigma</code> .....                                                                                                               | 165 |                                                  |                                                                |
| <code>\vartheta</code> .....                                                                                                               | 165 |                                                  |                                                                |
| <code>\vbox</code> T <sub>E</sub> X de base.....                                                                                           | 79  |                                                  |                                                                |
| <code>\vdash @math@xeatspaces @backslashchar</code><br><code>\vdash@comment Taquet droit ; prouve,</code><br><code>implique ;</code> ..... | 165 |                                                  |                                                                |
| <code>\vdots</code> .....                                                                                                                  | 173 |                                                  |                                                                |
| <code>\vec</code> .....                                                                                                                    | 176 |                                                  |                                                                |
| <code>\vector</code> .....                                                                                                                 | 88  |                                                  |                                                                |
| <code>\vee</code> .....                                                                                                                    | 165 |                                                  |                                                                |

**C**

cadre, épaisseur du filet ..... 206  
 cadre, séparation d'avec le contenu ..... 206  
 cc ..... 143  
 center, environnement ..... 61  
 chapter@comment , compteur ..... 138  
 classe article ..... 7  
 classe book ..... 7  
 classe letter ..... 7  
 classe report ..... 7  
 classe slides ..... 7  
 cleveref, paquetage ..... 55  
 cm ..... 142  
 cm-super, paquetage ..... 18  
 cmd.exe, utilisé par \write18 ..... 266  
 commande dvipdfmx ..... 3  
 commande dvips ..... 3  
 commande dvitype ..... 3  
 commande latex ..... 3  
 commande pdflatex ..... 3  
 commande xdvi ..... 3  
 comprehensive, paquetage ..... 155  
 cprotect, paquetage ..... 108, 109

**D**

datatool, paquetage ..... 261  
 datetime, paquetage ..... 237  
 dcolumn, paquetage ..... 61  
 dd ..... 143  
 delim\_0 ..... 253  
 delim\_1 ..... 253  
 delim\_2 ..... 253  
 delim\_n ..... 253  
 delim\_r ..... 253  
 description, environnement ..... 63  
 displaymath, environnement ..... 63, 153  
 document, environnement ..... 64  
 dvilualatex-dev ..... 4

**E**

em ..... 143  
 enumerate, environnement ..... 65  
 enumi@comment , compteur ..... 138  
 enumii@comment , compteur ..... 138  
 enumiii@comment , compteur ..... 138  
 enumitem, paquetage ..... 72, 77  
 enumiv@comment , compteur ..... 138  
 environnement abstract ..... 59  
 environnement array ..... 60  
 environnement center ..... 61  
 environnement description ..... 63  
 environnement displaymath ..... 63, 153  
 environnement document ..... 64  
 environnement enumerate ..... 65  
 environnement eqnarray ..... 66  
 environnement equation ..... 66, 153

environnement figure ..... 67  
 environnement filecontents ..... 68  
 environnement filecontents\* ..... 68  
 environnement flushleft ..... 69  
 environnement flushright ..... 70  
 environnement itemize ..... 71  
 environnement letter ..... 72  
 environnement list ..... 72  
 environnement math ..... 78, 153  
 environnement minipage ..... 79  
 environnement picture ..... 81  
 environnement quotation ..... 89  
 environnement quote ..... 89  
 environnement sloppypar ..... 115  
 environnement tabbing ..... 90  
 environnement table ..... 96  
 environnement tabular ..... 97  
 environnement thebibliography ..... 102  
 environnement theindex ..... 249  
 environnement theorem ..... 107  
 environnement titlepage ..... 107  
 environnement verbatim ..... 108  
 environnement verse ..... 109  
 eqnarray, environnement ..... 66  
 equation@comment , compteur ..... 138  
 equation, environnement ..... 66, 153  
 etex, commande ..... 4  
 etoolbox, paquetage ..... 12  
 eurosym, paquetage ..... 231  
 ex ..... 143  
 expl3, paquetage ..... 229  
 expressions ..... 145

**F**

fancyhdr, paquetage ..... 187, 189  
 fancyvrb, paquetage ..... 90, 108  
 fichier .dvi ..... 3  
 fichier .log ..... 3  
 figure@comment , compteur ..... 138  
 figure, environnement ..... 67  
 filecontents\*, environnement ..... 68  
 filecontents, environnement ..... 68  
 first-latex-doc document ..... 1  
 flafter, paquetage ..... 38  
 float, paquetage ..... 38  
 flushleft, environnement ..... 69  
 flushright, environnement ..... 70  
 fontenc ..... 17  
 fontenc, paquetage ..... 116  
 fontspec, paquetage ..... 17  
 footmisc, paquetage ..... 120  
 footnote@comment , compteur ..... 138

**G**

|                           |     |
|---------------------------|-----|
| geometry, paquetage ..... | 8   |
| graphpap, paquetage ..... | 85  |
| grfext, paquetage.....    | 218 |
| group_skip.....           | 252 |

**H**

|                                                                              |               |
|------------------------------------------------------------------------------|---------------|
| <a href="https://latexref.xyz">https://latexref.xyz</a> page d'accueil ..... | 1             |
| hyperref, paquetage.....                                                     | 189, 246, 269 |

**I**

|                              |                         |
|------------------------------|-------------------------|
| in .....                     | 142                     |
| indent_length .....          | 253                     |
| indent_space .....           | 253                     |
| indentfirst, paquetage....   | 44, 45, 46, 48, 49, 150 |
| index, fichier de style..... | 251                     |
| index, paquetage.....        | 251                     |
| index, traitement.....       | 251                     |
| inputenc.....                | 235                     |
| item_0 .....                 | 252                     |
| item_01 .....                | 253                     |
| item_1 .....                 | 252                     |
| item_12 .....                | 253                     |
| item_2 .....                 | 252                     |
| item_x1 .....                | 253                     |
| item_x2 .....                | 253                     |
| itemize, environnement ..... | 71                      |

**L**

|                                       |              |
|---------------------------------------|--------------|
| latex.....                            | 4            |
| latex-dev .....                       | 4            |
| latex-doc-ptr document .....          | 1            |
| latexrefman@tug.org adresse mél. .... | 1            |
| latexsym, paquetage .....             | 166          |
| lethead_flag .....                    | 252          |
| lethead_prefix.....                   | 252          |
| lethead_suffix.....                   | 252          |
| letter, environnement.....            | 72           |
| line_max .....                        | 253          |
| list, environnement .....             | 72           |
| listings, paquetage.....              | 90, 108, 109 |
| lmodern, paquetage.....               | 18           |
| longtable, paquetage .....            | 90           |
| LR box.....                           | 83           |
| lrbox.....                            | 210          |
| lshort document.....                  | 1            |
| lualatex .....                        | 4            |
| lualatex commande.....                | 4            |
| lualatex-dev.....                     | 4            |

**M**

|                                    |                                      |
|------------------------------------|--------------------------------------|
| macros2e, paquetage.....           | 128                                  |
| makeidx, paquetage .....           | 254                                  |
| makeindex, fichier de style.....   | 251                                  |
| math, environnement.....           | 78, 153                              |
| math, flèches.....                 | 166                                  |
| mathtools, paquetage ....          | 153, 165, 169, 170, 178,<br>180, 181 |
| MetaPost, paquetage.....           | 86                                   |
| mfirstuc, paquetage.....           | 229                                  |
| mhchem, paquetage.....             | 154                                  |
| minipage, environnement.....       | 79                                   |
| minted, paquetage .....            | 90, 108, 109                         |
| mlp, paquetage.....                | 45                                   |
| mm.....                            | 142                                  |
| mpfootnote@comment , compteur..... | 138                                  |
| mu.....                            | 143                                  |
| multind, paquetage .....           | 248                                  |

**O**

|                             |   |
|-----------------------------|---|
| option 10pt.....            | 7 |
| option 11pt.....            | 7 |
| option 12pt.....            | 7 |
| option a4paper .....        | 7 |
| option a5paper .....        | 7 |
| option b5paper .....        | 7 |
| option draft.....           | 8 |
| option executivepaper ..... | 7 |
| option final.....           | 8 |
| option fleqn.....           | 8 |
| option landscape .....      | 8 |
| option legalpaper .....     | 7 |
| option leqno.....           | 8 |
| option letterpaper.....     | 7 |
| option notitlepage.....     | 8 |
| option onecolumn .....      | 8 |
| option onside .....         | 8 |
| option openany .....        | 8 |
| option openbib .....        | 8 |
| option openright.....       | 8 |
| option titlepage .....      | 8 |
| option twocolumn .....      | 8 |
| option twoside .....        | 8 |

## P

package, paquetage..... 41  
 page@comment , compteur..... 138  
 page\_precedence..... 253  
 paquetage adjustbox..... 204  
 paquetage algorithm2e..... 90  
 paquetage amscd..... 167  
 paquetage amsfonts..... 153, 166  
 paquetage amsmath... 60, 61, 64, 67, 107, 153, 170,  
     172, 173, 175, 181, 183, 197  
 paquetage amsthm..... 107  
 paquetage answers..... 263  
 paquetage appendix..... 50  
 paquetage array..... 61  
 paquetage Asymptote..... 86, 201, 205, 265  
 paquetage babel..... 45, 103, 116, 243, 250  
 paquetage beamer..... 275  
 paquetage bigfoot..... 122  
 paquetage bm..... 168  
 paquetage cleveref..... 55  
 paquetage cm-super..... 18  
 paquetage comprehensive..... 155  
 paquetage cprotect..... 108, 109  
 paquetage datatool..... 261  
 paquetage datetime..... 237  
 paquetage dcolumn..... 61  
 paquetage enumitem..... 72, 77  
 paquetage etoolbox..... 12  
 paquetage eurosym..... 231  
 paquetage expl3..... 229  
 paquetage fancyhdr..... 187, 189  
 paquetage fancyvrb..... 90, 108  
 paquetage flafter..... 38  
 paquetage float..... 38  
 paquetage fontenc..... 116  
 paquetage fontspec..... 17  
 paquetage footmisc..... 120  
 paquetage geometry..... 8  
 paquetage graphpap..... 85  
 paquetage grfext..... 218  
 paquetage hyperref..... 189, 246, 269  
 paquetage indentfirst..... 44, 45, 46, 48, 49, 150  
 paquetage index..... 251  
 paquetage latexsym..... 166  
 paquetage listings..... 90, 108, 109  
 paquetage lmodern..... 18  
 paquetage longtable..... 90  
 paquetage macros2e..... 128  
 paquetage makeidx..... 254  
 paquetage mathtools..... 153, 165, 169, 170, 178,  
     180, 181  
 paquetage MetaPost..... 86  
 paquetage mfirstuc..... 229  
 paquetage mhchem..... 154  
 paquetage minted..... 90, 108, 109  
 paquetage mlp..... 45  
 paquetage multind..... 248  
 paquetage package..... 41

paquetage pict2e..... 86  
 paquetage polyglossia..... 45, 116, 243, 250  
 paquetage PSTricks..... 86  
 paquetage setspace..... 37  
 paquetage showidx..... 248  
 paquetage siunitx..... 81, 196  
 paquetage soul..... 95  
 paquetage suffix..... 129  
 paquetage tablefootnote..... 122  
 paquetage texosquery..... 266  
 paquetage textcase..... 229  
 paquetage textcomp..... 25, 229  
 paquetage tikz-cd..... 167  
 paquetage TikZ..... 86, 201, 205  
 paquetage titlesec..... 42, 44, 45, 47, 48, 49  
 paquetage tocbibbind..... 243  
 paquetage tocloft..... 243, 246  
 paquetage ulem..... 177  
 paquetage unicode-math..... 174  
 paquetage url..... 109  
 paquetage verbatimbox..... 108  
 paquetage xparse..... 129  
 paquetage xr..... 57  
 paquetage xr-hyper..... 57  
 paquetage xspace..... 136  
 paquetage xstring..... 272  
 paragraph@comment , compteur..... 138  
 part@comment , compteur..... 138  
 pc..... 142  
 pdflatex..... 4  
 pdflatex-dev..... 4  
 pict2e, paquetage..... 86  
 picture, environnement..... 81  
 platex..... 4  
 platex-dev..... 4  
 polyglossia, paquetage..... 45, 116, 243, 250  
 postamble..... 252  
 preamble..... 252  
 PSTricks, paquetage..... 86  
 pt..... 142

## Q

quotation, environnement..... 89  
 quote, environnement..... 89

## S

secnumdepth..... 43  
 secnumdepth compteur..... 43  
 section@comment , compteur..... 138  
 setspace, paquetage..... 37  
 sh, utilisé par \write18..... 266  
 SHELL, environnement, variables de..... 266  
 SHELL, variables d'environnement..... 266  
 showidx, paquetage..... 248  
 siunitx, paquetage..... 81, 196  
 sloppypar, environnement..... 115

soul, paquetage ..... 95  
 sp ..... 143  
 subparagraph@comment , compteur ..... 138  
 subsection@comment , compteur ..... 138  
 subsubsection@comment , compteur ..... 138  
 suffix, paquetage ..... 129

**T**

tabbing, environnement ..... 90  
 table@comment , compteur ..... 138  
 table, environnement ..... 96  
 tablefootnote, paquetage ..... 122  
 tabular, environnement ..... 97  
 texosquery, paquetage ..... 266  
 texput, nom d'ouvrage par défaut ..... 272  
 textcase, paquetage ..... 229  
 textcomp, paquetage ..... 25, 229  
 thebibliography, environnement ..... 102  
 theindex ..... 249  
 theindex, environnement ..... 249  
 theorem, environnement ..... 107  
 tikz-cd, paquetage ..... 167  
 TikZ, paquetage ..... 86, 201, 205  
 titlepage, environnement ..... 107  
 titlesec, paquetage ..... 42, 44, 45, 47, 48, 49  
 tocbibbind, paquetage ..... 243  
 tocdepth ..... 43  
 tocdepth counter ..... 43  
 tocloft, paquetage ..... 243, 246

topmargin ..... 35

**U**

ulem, paquetage ..... 177  
 unicode-math, paquetage ..... 174  
 uplatex ..... 4  
 uplatex-dev ..... 4  
 url, paquetage ..... 109  
 usrguide documentation officielle ..... 1

**V**

verbatim, environnement ..... 108  
 verbatimbox, paquetage ..... 108  
 verse, environnement ..... 109

**X**

xdvipdfmx ..... 4  
 xelatex ..... 4  
 xelatex commande ..... 4  
 xelatex-dev ..... 4  
 xindex, programme ..... 253  
 xparse, paquetage ..... 129  
 xr, paquetage ..... 57  
 xr-hyper, paquetage ..... 57  
 xspace, paquetage ..... 136  
 xstring, paquetage ..... 272