

L^AT_EX2_ε: Un manuel de référence officiel

Septembre 2020

<https://latexref.xyz>

Ce document est un manuel de référence officieux pour \LaTeX , un système de préparation de documents, version « Septembre 2020 ».

Ce manuel a été traduit du fichier `LATEX.HLP v1.0a` de la bibliothèque d'aide VMS. La version pré-translation 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 $\text{\LaTeX}2\epsilon$ 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 *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) ; il a utilisé ces documents en tant que matériel de référence et n'en a copié aucun texte directement.

Tous droits réservés © 2015-2020 Vincent Belaïche — traduction.

Tous droits réservés © 2007-2020 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 191.

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	23
6	Rubricage	29
7	Des renvois	42
8	Environnements	45
9	Saut à la ligne	85
10	Saut de page	88
11	Note en bas de page	90
12	Définitions	93
13	Compteurs	103
14	Longueurs	106
15	Faire des paragraphes	110
16	Formules mathématiques	112
17	Modes	129
18	Styles de page	131
19	Espaces	132
20	Boîtes	137
21	Couleur	140
22	Graphiques	144
23	Insertions spéciales	156
24	Partitionner le fichier d'entrée	164
25	Parties pré/post-liminaires	168
26	Des lettres	172
27	Entrée/sortie console	177
28	Ligne de commande	184
A	Patrons de documents	188
B	Traduction de la licence	191
	Index des concepts	192
	Index des commandes	200

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	4
2.4.1	Les déclarations de commandes	5
2.5	Les environnements	5
2.6	CTAN : The Comprehensive T _E X Archive Network	5
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	styles des polices	17
4.2	tailles des polices	19
4.3	commandes de fontes de bas niveau	19
5	Mise en page	23
5.1	<code>\onecolumn</code>	23
5.2	<code>\twocolumn</code>	23
5.3	<code>\flushbottom</code>	24
5.4	<code>\raggedbottom</code>	24
5.5	Paramètres de mise en page	24
5.6	Flottants	25
6	Rubricage	29
6.1	<code>\part</code>	30
6.2	<code>\chapter</code>	31
6.3	<code>\section</code>	33
6.4	<code>\subsection</code>	34
6.5	<code>\subsubsection</code> , <code>\paragraph</code> , <code>\subparagraph</code>	35
6.6	<code>\appendix</code>	36
6.7	<code>\frontmatter</code> , <code>\mainmatter</code> , <code>\backmatter</code>	37
6.8	<code>\@startsection</code> , composer les rubriques	37

7	Des renvois	42
7.1	<code>\label</code>	42
7.2	<code>\pageref{clef}</code>	43
7.3	<code>\ref{clef}</code>	44
8	Environments	45
8.1	<code>abstract</code>	45
8.2	<code>array</code>	46
8.3	<code>center</code>	47
8.3.1	<code>\centering</code>	47
8.4	<code>description</code>	47
8.5	<code>displaymath</code>	48
8.6	<code>document</code>	49
8.7	<code>enumerate</code>	49
8.8	<code>eqnarray</code>	51
8.9	<code>equation</code>	51
8.10	<code>figure</code>	51
8.11	<code>filecontents</code> : Écrire un fichier externe	52
8.12	<code>flushleft</code>	53
8.12.1	<code>\raggedright</code>	53
8.13	<code>flushright</code>	54
8.13.1	<code>\raggedleft</code>	54
8.14	<code>itemize</code>	54
8.15	environnement <code>letter</code> : écrire des lettres postales	56
8.16	<code>list</code>	56
8.16.1	<code>\item</code> : Une entrée dans une liste	61
8.16.2	<code>trivlist</code> : Une forme restreinte de <code>list</code>	62
8.17	<code>math</code>	62
8.18	<code>minipage</code>	63
8.19	<code>picture</code>	65
8.19.1	<code>\circle</code>	67
8.19.2	<code>\makebox</code>	67
8.19.3	<code>\framebox</code>	67
8.19.4	<code>\dashbox</code>	67
8.19.5	<code>\frame</code>	68
8.19.6	<code>\line</code>	68
8.19.7	<code>\linethickness</code>	68
8.19.8	<code>\thicklines</code>	69
8.19.9	<code>\thinlines</code>	69
8.19.10	<code>\multiput</code>	69
8.19.11	<code>\oval</code>	69
8.19.12	<code>\put</code>	69
8.19.13	<code>\shortstack</code>	69
8.19.14	<code>\vector</code>	70
8.20	quotation et <code>quote</code>	70
8.21	<code>tabbing</code>	70
8.22	<code>table</code>	75
8.23	<code>tabular</code>	76

8.23.1	<code>\multicolumn</code>	79
8.23.2	<code>\cline</code>	80
8.23.3	<code>\hline</code>	80
8.23.4	<code>\vline</code>	81
8.24	<code>thebibliography</code>	81
8.24.1	<code>\bibitem</code>	81
8.24.2	<code>\cite</code>	82
8.24.3	<code>\nocite</code>	82
8.24.4	Utilisation de Bib _T E _X	82
8.25	<code>theorem</code>	83
8.26	<code>titlepage</code>	83
8.27	<code>verbatim</code>	83
8.27.1	<code>\verb</code>	83
8.28	<code>verse</code>	84
9	Saut à la ligne	85
9.1	<code>\</code>	85
9.2	<code>\obeycr</code> & <code>\restorecr</code>	85
9.3	<code>\newline</code>	85
9.4	<code>\-</code> (césure à gré).....	86
9.5	<code>\discretionary</code> (point de césure généralisé).....	86
9.6	<code>\fussy</code>	87
9.7	<code>\sloppy</code>	87
9.8	<code>\hyphenation</code>	87
9.9	<code>\linebreak</code> & <code>\nolinebreak</code>	87
10	Saut de page	88
10.1	<code>\clearpage</code> & <code>\cleardoublepage</code>	88
10.2	<code>\newpage</code>	88
10.3	<code>\enlargethispage</code>	88
10.4	<code>\pagebreak</code> & <code>\nopagebreak</code>	89
11	Note en bas de page	90
11.1	<code>\footnote</code>	90
11.2	<code>\footnotemark</code>	90
11.3	<code>\footnotetext</code>	91
11.4	Notes en bas de page dans un tableau.....	91
11.5	Note en bas de page dont le renvoi est au sein d'un titre de rubrique.....	92
11.6	Paramètres des notes en bas de page.....	92

12	Définitions	93
12.1	<code>\newcommand</code> & <code>\renewcommand</code>	93
12.2	<code>\providecommand</code>	94
12.3	<code>\makeatletter</code> et <code>\makeatother</code>	94
12.4	<code>\@ifstar</code>	95
12.5	<code>\newcounter</code> : allouer un compteur.....	96
12.6	<code>\newlength</code> : allouer une longueur.....	97
12.7	<code>\newsavebox</code> : allouer une boîte.....	97
12.8	<code>\newenvironment</code> & <code>\renewenvironment</code>	97
12.9	<code>\newtheorem</code>	99
12.10	<code>\newfont</code> : définit une nouvelle police (obsolète).....	101
12.11	<code>\protect</code>	101
13	Compteurs	103
13.1	<code>\alph</code> <code>\Alph</code> <code>\arabic</code> <code>\roman</code> <code>\Roman</code> <code>\fnsymbol</code> : Taper des compteurs.....	103
13.2	<code>\usecounter{compteur}</code>	103
13.3	<code>\value{compteur}</code>	104
13.4	<code>\setcounter{compteur}{valeur}</code>	104
13.5	<code>\addtocounter{compteur}{valeur}</code>	104
13.6	<code>\refstepcounter{compteur}</code>	104
13.7	<code>\stepcounter{compteur}</code>	104
13.8	<code>\day</code> <code>\month</code> <code>\year</code> : Prédefinés compteurs.....	105
14	Longueurs	106
14.1	Unités de longueur.....	107
14.2	<code>\setlength</code>	108
14.3	<code>\addtolength</code>	108
14.4	<code>\settodepth</code>	108
14.5	<code>\settoheight</code>	108
14.6	<code>\settowidth</code>	109
14.7	Longueurs prédéfinies.....	109
15	Faire des paragraphes	110
15.1	<code>\indent</code>	110
15.2	<code>\noindent</code>	110
15.3	<code>\parindent</code> & <code>\parskip</code>	110
15.4	Notes en marge.....	110
16	Formules mathématiques	112
16.1	Indices inférieurs & supérieurs.....	112
16.2	Symboles mathématiques.....	113
16.3	Fonctions mathématiques.....	124
16.4	Accents mathématiques.....	125
16.5	Espacement en mode mathématique.....	126
16.6	Recueil de diverses commandes utiles en math.....	126

17	Modes	129
17.1	<code>\ensuremath</code>	129
18	Styles de page	131
18.1	<code>\maketitle</code>	131
18.2	<code>\pagenumbering</code>	131
18.3	<code>\pagestyle</code>	131
18.4	<code>\thispagestyle{style}</code>	131
19	Espaces	132
19.1	<code>\hspace</code>	132
19.2	<code>\hfill</code>	132
19.3	<code>\<espace></code> et <code>\@</code>	132
19.4	<code>\</code> après une séquence de contrôle.....	133
19.5	<code>\frenchspacing</code> , rendre égaux les espacements inter-phrase et inter-mot.....	133
19.6	<code>\thinspace</code>	133
19.7	<code>\/</code> : insérer une correction italique.....	133
19.8	<code>\hrulefill</code> <code>\dotfill</code>	134
19.9	<code>\addvspace</code>	134
19.10	<code>\bigskip</code> <code>\medskip</code> <code>\smallskip</code>	135
19.11	<code>\vfill</code>	135
19.12	<code>\vspace</code>	135
20	Boîtes	137
20.1	<code>\mbox{texte}</code>	137
20.2	<code>\fbox</code> and <code>\framebox</code>	137
20.3	<code>lrbox</code>	137
20.4	<code>\makebox</code>	137
20.5	<code>\parbox</code>	138
20.6	<code>\raisebox</code>	139
20.7	<code>\savebox</code>	139
20.8	<code>\sbox{\boxcmd}{texte}</code>	139
20.9	<code>\usebox{\boxcmd}</code>	139
21	Couleur	140
21.1	Options du paquetage <code>color</code>	140
21.2	Modèles de couleur.....	140
21.3	Commandes pour <code>color</code>	141
21.3.1	Définir des couleurs	141
21.3.2	Mettre du texte en couleur	142
21.3.3	Faire des boîtes en couleur	143
21.3.4	Faire des pages en couleur	143

22	Graphiques	144
22.1	Options du paquetage <code>graphics</code>	144
22.2	Configuration du paquetage <code>graphics</code>	145
22.2.1	<code>\graphicspath</code>	146
22.2.2	<code>\DeclareGraphicsExtensions</code>	147
22.2.3	<code>\DeclareGraphicsRule</code>	147
22.3	Commandes du paquetage <code>graphics</code>	148
22.3.1	<code>\includegraphics</code>	148
22.3.2	<code>\rotatebox</code>	154
22.3.3	<code>\scalebox</code>	155
22.3.4	<code>\resizebox</code>	155
23	Insertions spéciales	156
23.1	Caractères réservés.....	156
23.2	Capitales et bas de casse.....	156
23.3	Symboles appelés par leur position dans une police.....	157
23.4	Symboles en texte.....	157
23.5	Accents.....	160
23.6	Lettres latines supplémentaires.....	162
23.7	<code>\rule</code>	163
23.8	<code>\today</code>	163
24	Partitionner le fichier d'entrée	164
24.1	<code>\endinput</code>	164
24.2	<code>\include</code> & <code>\includeonly</code>	165
24.3	<code>\input</code>	167
25	Parties pré/post-liminaires	168
25.1	Table des matières etc.....	168
25.1.1	<code>\addcontentsline</code>	169
25.1.2	<code>\addtocontents</code>	170
25.1.3	<code>\nofiles</code>	170
25.2	Glossaires.....	170
25.3	Index.....	171
26	Des lettres	172
26.1	<code>\address</code>	173
26.2	<code>\cc</code>	173
26.3	<code>\closing</code>	174
26.4	<code>\encl</code>	174
26.5	<code>\location</code>	174
26.6	<code>\makelabels</code>	174
26.7	<code>\name</code>	175
26.8	<code>\opening{texte}</code>	175
26.9	<code>\ps</code>	175
26.10	<code>\signature{texte}</code>	175
26.11	<code>\telephone</code>	176

27	Entrée/sortie console	177
27.1	<code>\openin</code> & <code>\openout</code>	177
27.2	<code>\read</code>	178
27.3	<code>\typein[cmd]{msg}</code>	178
27.4	<code>\typeout{msg}</code>	178
27.5	<code>\write</code>	179
27.5.1	<code>\write</code> et la sécurité	180
27.5.2	<code>\message</code>	181
27.5.3	<code>\wlog</code>	181
27.5.4	<code>\write18</code>	182
28	Ligne de commande	184
28.1	Options passées à la ligne de commande	184
28.2	Spécifier du code LaTeX sur la ligne de commande	186
28.3	Quand quelque chose se passe mal	187
Annexe A	Patrons de documents	188
A.1	Un patron <code>beamer</code>	188
A.2	Patron <code>article</code>	188
A.3	Un patron <code>book</code>	189
A.4	Un patron <code>book</code> plus élaboré	189
Annexe B	Traduction de la licence	191
	Index des concepts	192
	Index des commandes	200

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 exemple : il s'agit d'obtenir un document contenant la phrase « Bonjour le monde ».

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

Ici 'article' est ce que l'on appelle la *classe de document* ; elle est implémentée dans une fichier `article.cls`. N'importe quelle classe de document 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 le *préambule*).

Le code `\begin{document} ... \end{document}` est ce qu'on appelle un *environnement* ; l'environnement 'document' (et aucun autre) est obligatoire dans tous les documents L^AT_EX (voir Section 8.6 [document], page 49). L^AT_EX fournit lui-même beaucoup d'environnements, et bien plus encore sont définis séparément. Voir Chapitre 8 [Environments], page 45.

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.

2.2 Fichiers de sortie

L^AT_EX produit un fichier de sortie principal et au moins deux fichiers accessoires. 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 aussi 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. 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 défini comme un ensemble de commandes qui sont exécutées par une implémentation T_EX (voir Chapitre 2 [Overview], page 2). Cette section donne une brève vue d'ensemble des principaux programmes.

`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

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

² Format portable de document

(<http://ctan.org/pkg/pdftex>). Selon qu'on invoque `latex` ou `pdflatex`, la sortie principale est respectivement un fichier `.dvi` ou un fichier `.pdf`.

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

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

lualatex Si L^AT_EX est invoqué avec la commande système `lualatex`, alors le moteur Lua`TEX` est exécuté (<http://ctan.org/pkg/luatex>). Ce programme permet que du code écrit dans le langage script Lua (<http://luatex.org>) interagisse avec la composition faite par `TEX`. Lua`TEX` 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`, mais cela est rarement utilisé.

xelatex Si L^AT_EX est invoqué avec la commande système `xelatex`, le moteur Xe`TEX` est exécuté (<http://tug.org/xetex>). Comme Lua`TEX`, Xe`TEX` 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. Xe`TEX` produit un fichier `.pdf` en sortie ; il ne prend pas en charge la sortie DVI.

En interne, Xe`TEX` 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.

D'autres variantes de L^AT_EX et `TEX` existent, par exemple pour fournir une prise en charge supplémentaire du japonais et d'autres langages ([u]p`TEX`, <http://ctan.org/pkg/ptex>, <http://ctan.org/pkg/uptex>).

2.4 Syntaxe des commandes L^AT_EX

Dans le fichier d'entrée L^AT_EX, un nom de commande commence avec une contre-oblique, `\`. 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 elles sont sensibles à la casse : vous devez toujours saisir les commandes en respectant la casse où elles ont été définies.

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).

2.4.1 Les déclarations de commandes

Certaines commandes changent la valeur ou la signification d'une autre commande ou d'un paramètre. Par exemple, la commande `\mainmatter` change le réglage de la numérotation des pages en passant de chiffres romains à des chiffres arabes.

2.5 Les environnements

Synopsis :

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

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 45, pour une liste des environnements.

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 matériel libre de la communauté T_EX et L^AT_EX. C'est un ensemble de sites Internet réparti dans le monde qui offre de télécharger du matériel en relation avec L^AT_EX. Visitez CTAN sur la Toile à l'adresse <https://ctan.org>.

Ce matériel est organisé 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 sites est le cœur. Ce site gère activement le matériel, par exemple en acceptant les téléversements de matériel nouveau ou mis 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 168).
letter	Courrier, incluant optionnellement des étiquettes pour enveloppes (voir Chapitre 26 [Letters], page 172).
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 188, 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 *d'options globales* ou des *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, 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. Les valeurs par défaut varient en fonction de la taille du papier, de la disposition recto ou recto-verso sélectionnée. Pour une impression en recto le texte est centré, pour recto-verso, `\oddsidemargin` vaut 40% de la différence entre `\paperwidth` et `\textwidth`, `\evensidemargin` valant le reste.

`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'une 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 94. 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 \LaTeX 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 \LaTeX 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 \LaTeX 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 93) 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}{warning text}
\PackageWarning{nom du paquetage}{warning text}
\ClassWarningNoLine{nom de la classe}{warning text}
\PackageWarningNoLine{nom du paquetage}{warning text}
\ClassInfo{nom de la classe}{info text}
\PackageInfo{nom du paquetage}{info text}
\ClassInfoNoLine{nom de la classe}{info text}
\PackageInfoNoLine{nom du paquetage}{info text}

```

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*.

The four warning commands are similar except that they write *warning text* on the screen with no error prompt. The four info commands write *info text* only in the transcript file. The `NoLine` versions do not show the number of the line generating the message, while the other versions do show that number.

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 `notcms` 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 déclaration 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 fichier 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 93) 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 101). Utilisez ces commande pour définir de nouvelles commandes robustes, ou redéfinir des commandes existantes en les rendant robustes. Contrairement à `\newcommand` elle ne produisent 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{AVERSTISSEMENT : 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 chemin 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 167.

`\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 optionnel. Si elle est présente, elle doit avoir le format *AAA/MM/JJ*. Si vous demandez une *date de parution* et que la date du paquetage installée 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 laquelle 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 antérieur à la *date du format* alors vous obtiendrez un avertissement

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 charge 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 défini 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 `\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 antérieur 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 antérieur 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

Deux aspects importants de la sélection d'une *police* sont la spécification d'une taille et celle d'un style. Les commandes \LaTeX pour ce faire sont décrites ci-après.

4.1 styles des polices

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

Ce premier groupe de commandes est généralement utilisé avec un argument, comme dans `\textit{texte en italique}`. Dans le tableau ci-dessous, la commande correspondante entre parenthèses est la « forme déclarative », qui ne prend pas arguments. La portée la forme déclarative s'étend jusqu'à la prochaine commande de type style ou jusqu'à la fin du groupe actuel.

Ces commandes, à la fois dans la forme à argument et dans la forme déclarative, sont cumulatives ; par exemple, vous pouvez dire aussi bien `\sffamily\bfseries` que `\bfseries\sffamily` pour obtenir du gras sans serif.

Vous pouvez également utiliser une forme de déclaration par environnement ; par exemple, `\begin{ttfamily} ... \end{ttfamily}`.

Ces commandes fournissent automatiquement une correction d'italique si nécessaire.

<code>\textrm</code>	<code>(\rmfamily)</code> Romain.
<code>\textit</code>	<code>(\itshape)</code> Italique.
<code>\emph</code>	Accent (commute entre <code>\textit</code> et <code>\textrm</code> selon le contexte).
<code>\textmd</code>	<code>(\mdseries)</code> Poids moyen (par défaut).
<code>\textbf</code>	<code>(\bfseries)</code> Gras.
<code>\textup</code>	<code>(\upshape)</code> Droit (par défaut). Le contraire d'incliné.
<code>\textsl</code>	<code>(\slshape)</code> Inclinée.
<code>\textsf</code>	<code>(\sffamily)</code> Sans serif.
<code>\textsc</code>	<code>(\scshape)</code> Petites capitales.
<code>\texttt</code>	<code>(\ttfamily)</code> Tapuscrit, encore appelé « machine à écrire ».
<code>\textnormal</code>	<code>(\normalfont)</code> Police principale du document.
<code>\mathrm</code>	Roman, pour une utilisation en mode mathématique.

<code>\mathbf</code>	Gras, pour une utilisation en mode mathématique.
<code>\mathsf</code>	Sans serif, 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 de “à l’ancienne”, qui ont des hauteurs et profondeur (et parfois largeurs) distincts de l’alignement standard des chiffres. 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 style à l’ancienne italique pour les chiffres en Computer Modern). Beaucoup d’autres polices ont des chiffre à l’ancienne aussi; parfois le paquetage `textcomp` doit être chargé, et parfois des options de paquet sont mises à disposition pour en faire le style par défaut. Entrée de FAQ : <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=osf>.

L^AT_EX fournit également les commandes suivantes, qui font passer inconditionnellement à un style donné, c.-à-d. *ne* sont *pas* cumulatives. De plus, elles sont utilisées différemment des commandes ci-dessus : `{\cmd ... }` au lieu de `\cmd{...}`. Ce sont deux constructions sans relation l’une avec l’autre.

<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 serif.
<code>\sl</code>	Incliné (oblique).
<code>\tt</code>	Tapuscrit, encore appelé « Machine à écrire » (largeur fixe).

Certaines personnes considèrent que les commandes commutation inconditionnelles de polices, comme `\tt`, sont obsolète et que *seulement* les commandes cumulatives (`\texttt`) doivent être utilisées. Je (Karl) ne suis pas d’accord. il y a situations tout à fait raisonnables quand un commutateur de police inconditionnelle est précisément ce dont vous avez besoin pour obtenir le résultat souhaité ; pour un exemple, voir Section 8.4 [description], page 47. Les deux ensembles de commandes ont leur place.

4.2 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*. On est utilise en les déclarant comme dans l'exemple qui suit :

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

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 en particulier la forme par environnement des déclarations de taille de police est que leur utilisation s'avère assez courante, en dépit des problèmes qu'il pose).

4.3 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 T_EX 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 L^AT_EX. 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 :

<code>m</code>	Moyen (normal)
<code>b</code>	Gras
<code>c</code>	Condensé
<code>bc</code>	Condensé Gras
<code>bx</code>	Gras étendu

Les valeurs possibles de graisses, considérées individuellement, sont :

<code>ul</code>	Ultra maigre
<code>e1</code>	Extra maigre
<code>l</code>	Maigre (L ight)
<code>s1</code>	Semi maigre

m Demi-gras (**M**edium, normal)
sb Semi gras
b Gras (**B**old)
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

Quand Lorsqu'on forme la chaîne de caractères *série* à partir de la graisse et de la largeur, on laisse tomber le **m** qui signifie medium 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)
sc Petites capitales
ui Italique droit
ol Plan

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 ; voir ensuite `\baselinestretch`.

`\baselinestretch`

L'espacement des lignes est également multiplié par la valeur du paramètre `\baselinestretch` ; la valeur par défaut est 1. Tout changement prend effet lorsque `\selectfont` (voir plus bas) est appelé. Vous pouvez modifier l'interligne pour tout le document en faisant, par ex. pour le doubler, `\renewcommand{\baselinestretch}{2.0}` dans le préambule,

Cependant, la meilleure façon de mettre un document en « espacement double » est d'utiliser le paquetage `setspace`. En plus de proposer une pluralité d'options

d'espacement, ce paquetage conserve 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.

`\linespread{facteur}`

Est équivalent à `\renewcommand{\baselinestretch}{facteur}`, et doit donc être suivie par `\selectfont` pour avoir un effet. Il vaut mieux que ce soit dans le préambule, ou utiliser le paquetage `setspace`, comme on vient de le décrire.

`\selectfont`

Les modifications apportées en appelant les commandes de polices 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 93).

`\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]
```

La déclaration `\twocolumn` commence une nouvelle page et produit sortie à deux colonnes. Si l'argument optionnel *texte1col* est présent, il est composée dans le mode à une colonne avant que la composition à deux colonnes ne commence.

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

`\columnsep`

La distance entre les colonnes (35pt par défaut).

`\columnseprule`

La largeur du filet entre les colonnes ; la valeur par défaut est 0pt, de sorte qu'il n'y a pas de filet.

`\columnwidth`

La largeur de la colonne en cours ; il est égal à `\textwidth` dans le cas d'un texte composé en une seule colonne.

Les paramètres ci-après contrôlent le comportement des flottants en cas de production à deux colonnes :

`\dbltopfraction`

Fraction maximale au sommet d'une page sur deux colonnes qui peut être occupée par des flottants. Par défaut vaut '0,7', peut être utilement redéfini en (par exemple) '0,9' pour aller moins tôt sur des pages de flottants.

`\dblfloatpagefraction`

La fraction minimum d'une page de flottants qui doit être occupée par des flottants, pour une page à flottant à deux colonnes. Par défaut vaut '0,5'.

`\dblfloatsep`

Distance entre les flottants en haut ou en bas d'une page de flottants à deux colonnes. Par défaut vaut '12pt plus2pt minus2pt' pour des documents en '10pt' ou '11pt', et '14pt plus2pt minus4pt' pour '12pt'.

`\dbltextfloatsep`

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

5.3 `\flushbottom`

La déclaration `\flushbottom` rend toutes les pages de texte de la même hauteur, en ajoutant de l'espace vertical supplémentaire si nécessaire pour remplir le page.

C'est la valeur par défaut si le mode `twocolumn` est sélectionné (voir Section 3.1 [Document class options], page 7).

5.4 `\raggedbottom`

La déclaration `\raggedbottom` rend toutes les pages de la hauteur naturelle de la matière sur cette page. Aucune des longueurs d'élastique ne seront étirée.

5.5 Paramètres de mise en page

`\headheight`

Hauteur de la boîte qui contient la tête en cours de traitement. La valeur par défaut est `'30pt'`, sauf dans la classe `book`, où elle varie en fonction de la taille de la police.

`\headsep` La distance verticale entre le bas de la ligne d'en-tête et la partie supérieure du texte principal. La valeur par défaut est `'25pt'`, sauf dans la classe `book`, où elle varie avec la taille de la police.

`\footskip`

Distance de la ligne de base de la dernière ligne de texte à la ligne de base du bas de page. La valeur par défaut est `'30pt'`, sauf dans la classe `book` où elle varie avec la taille de la police.

`\linewidth`

Largeur de la ligne actuelle, diminuée pour chaque `list` imbriquée (voir Section 8.16 [list], page 56). Plus précisément, elle est inférieure à `\textwidth` par la somme de `\leftmargin` et `\rightmargin` (voir Section 8.14 [itemize], page 54). La valeur par défaut varie en fonction de la taille de la police, la largeur du papier, le mode à deux colonnes, etc. Pour un document de classe `article` en taille de police `'10pt'`, elle vaut `'345pt'` ; dans le mode à deux colonnes, elle passe à `'229.5pt'`.

`\textheight`

La hauteur verticale normale du corps de la page ; la valeur par défaut varie en fonction de la taille de la police, de la classe du document, etc. Pour un document de classe `article` ou `report` en taille de police `'10pt'`, elle vaut `'43\baselineskip'` ; pour une classe `book`, elle vaut `'41\baselineskip'`. Pour `'11pt'`, c'est `'38\baselineskip'` et pour `'12pt'` c'est `'36\baselineskip'`.

`\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).

\textwidth

La largeur horizontale totale de l'ensemble du corps de la page; la valeur par défaut varie comme d'habitude. 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 23).

Dans les listes (voir Section 8.16 [list], page 56), `\textwidth` est le reste 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 63,) ou `\parbox` (voir Section 20.5 [\parbox], page 138), 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`.

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

\topmargin

L'espace entre le haut de la page T_EX (un pouce à partir du haut de la feuille, par défaut) et le sommet de l'en-tête de page. La valeur par défaut est calculée sur la base de nombreux autres paramètres : `\paperheight - 2in - \headheight - \headsep - \textheight - \footskip`, et ensuite divisé par deux.

\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 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

L^AT_EX sait gérer plusieurs classes de matériel flottant. Il y a deux classes définies par défaut, `figure` (voir Section 8.10 [figure], page 51) et `table` (voir Section 8.22 [table], page 75), mais vous pouvez créer une nouvelle classes avec le paquetage `float`.

Au sein d'une même classe flottante L^AT_EX respecte l'ordre, de sorte que la première figure dans le code source d'un document est toujours composée avant la deuxième figure. Cependant, L^AT_EX 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 apparaît 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 son 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 L^AT_EX 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.

\LaTeX 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 paramètre en relation aux fractions de pages occupées par du texte flottant et non flottant (on peut les changer avec `\renewcommand{parameter}{decimal between 0 and 1}`) :

La fraction maximal 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 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{parameter}{length expression}`) :

`\floatsep`

Espace entre les floattants 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 \TeX 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 \TeX de sorte à réduire les problèmes de flottants rejetés à la fin. Une explication complète de l’algorithme de placement des flottant se trouve dans l’article de Frank Mittelbach « How to influence the position of float environments like figure and table in \TeX ? » (<http://latex-project.org/papers/tb111mitt-float.pdf>).

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 36). 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 131). 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 30). 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 30).

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 103). 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 104). 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 104). 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 168) et dans les hauts de pages (voir Chapitre 18 [Page styles], page 131). 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 30, et voir [Sectioning/tocdepth], page 30).

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 standardes 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 168), et règle l'information de haut de page (voir Chapitre 18 [Page styles], page 131).

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'incrément 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-version 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 168) et dans les hauts de page (voir Chapitre 18 [Page styles], page 131). 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 haut 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 30, et voir [Sectioning/tocdepth], page 30).

Lorsque 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 94) :

```
\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 88.

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 168) et dans les titres de haut de page (voir Chapitre 18 [Page styles], page 131). 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 matières 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 30, et voir [Sectioning/tocdepth], page 30).

Lorsque 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 renfoncé, 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 renfoncé, 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 37). 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 \LaTeX 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 \LaTeX 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 168). 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 30, et voir [Sectioning/tocdepth], page 30).

Lorsque 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 37). 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'incrémente 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 168). 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 30, et voir [Sectioning/tocdepth], page 30).

Lorsque 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 37). 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 189, 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 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 déclaration `\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 29). 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 189, pour un exemple d’usage des 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 standards `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, cette commande 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 29. 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 29, 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 30). 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 renforcement de toutes les lignes du titre par rapport à la marge de gauche. Pour un renforcement 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, L^AT_EX 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 L^AT_EX plus de latitude lors de la fabrication de la page (voir Chapitre 14 [Lengths], page 106).

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 L^AT_EX 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`.

Voici des exemples. Pour les utiliser, mettez-les 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 94.

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 38.
  {1}% [niveau], page 38.
  {0pt}% [retrait], page 39.
  {-3.5ex plus -1ex minus -.2ex}% [avant], page 39.
  {2.3ex plus.2ex}% [après], page 39.
  {\centering\normalfont\Large\bfseries}}% [style], page 39.
```

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 38.
  {2}% [niveau], page 38.
  {0em}% [retrait], page 39.
  {-1ex plus 0.1ex minus -0.05ex}% [avant], page 39.
  {-1em plus 0.2em}% [après], page 39.
  {\scshape}% [style], page 39.
}
```

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 re-compiler 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 44).

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 tables
<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 attribué le numéro de la section courante et la clef `fig:test` se verra attribué 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 that the 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 42). 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'item item numéro~\ref{populaire}.■
\begin{enumerate}
\item Plain \TeX
\item \label{populaire} \LaTeX
\item Con\TeX t
\end{enumerate}
```

8 Environments

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-colonne (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`, normalement au sein d’un environnement `equation` (voir Section 8.9 [equation], page 51). Les entrées dans chaque colonne sont séparées avec une esperluette (&). Les lignes sont terminées par une double contr’oblique (voir Section 9.1 [\\], page 85).

L’argument obligatoire *patron* décrit le nombre de colonnes, l’alignement en leur sein, et le formatage des régions inter-colonne. Voir Section 8.23 [tabular], page 76, 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, dans le texte (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’.

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.

Voici un exemple d’un tableau :

```
\begin{equation}
\begin{array}{cr}
\sqrt{y} & \&12.3 \\
x^2 & \&3.4
\end{array}
\end{equation}
```

L’exemple suivante nécessite `\usepackage{amsmath}` dans le préambule :

```
\begin{equation}
\begin{vmatrix}{cc}
```



```

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

```

8.3 center

Synopsis :

```

\begin{center}
line1 \\
line2 \\
\end{center}

```

L'environnement `center` vous permet de créer un paragraphe consistant de lignes qui sont centrée entre les marges de gauche et de droite de la page courante. Chaque ligne est terminée avec la chaîne `\\`.

8.3.1 \centering

La déclaration `\centering` correspond à l'environnement `center`. Cette déclaration peut être utilisée à l'intérieur d'un environnement tel que `quote` ou d'une `parbox`. Ainsi, le texte d'une figure ou d'une table peut être centré sur la page en mettant une commande `\centering` au début de l'environnement de la figure ou table.

Contrairement à l'environnement `center`, la commande `\centering` ne commence pas un nouveau paragraphe ; elle change simplement la façon dont \LaTeX formate les unités paragraphe. Pour affecter le format d'une unité paragraphe, la portée de la déclaration doit contenir une ligne à blanc ou la commande `\end` (d'un environnement tel que `quote`) qui finit l'unité de paragraphe.

Voici un exemple :

```

\begin{quote}
\centering
first line \\
second line \\
\end{quote}

```

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 provoque l'erreur \LaTeX 'Something's wrong--perhaps a missing \item'.

Cet exemple montre l’environnement utilisé pour une séquence de définitions. Les étiquettes ‘lama’ et ‘llama’ ressortent en gras avec leur bords gauches alignés sur la marge de gauche.

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

Faites démarrer la liste d’articles avec la commande `\item` (voir Section 8.16.1 [`\item`], page 61). 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 4.1 [Font styles], page 17) 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 \LaTeX , voir Section 8.14 [`itemize`], page 54, et Section 8.7 [`enumerate`], page 49. Contrairement à ces environnements, imbriquer les environnement `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 56. 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 51).

\LaTeX 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 environment 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 exécutée 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 54) et `description` (voir Section 8.4 [description], page 47). 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 103). 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 61).

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 93). 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 103.

Pour plus ample information sur la personnalisation de la forme voir Section 8.16 [list], page 56. De même, le paquetage `enumitem` est utile pour cela.

8.8 eqnarray

Tout d'abord une mise en garde : l'environnement `eqnarray` a quelques maladresse que l'on ne peut pas surmonter ; l'article « Évitez eqnarray ! » de Lars Madsen les décrit en détail (<http://tug.org/TUGboat/tb33-1/tb103madsen.pdf>). Ce qu'il faut retenir c'est qu'il vaut mieux utiliser l'environnement `align` (et d'autres) du paquetage `amsmath`.

Néanmoins, voici une description d'`eqnarray` :

```
\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 esperluète `&`.

`*` 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}
  math
\end{equation}
```

L'environnement `equation` commence un environnement en `displaymath` (voir Section 8.5 [`displaymath`], page 48), par ex. en centrant le texte `math` sur la page, et aussi en plaçant un numéro d'équation dans la marge de droite.

8.10 figure

Synopsis :

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

ou

```
\begin{figure*}[placement]
```

```

corpsfigure
\caption[titreldf]{texte}
\label{étiquette}
\end{figure*}

```

Une classe de flottant (voir Section 5.6 [Floats], page 25). Parce qu’elles ne peuvent pas être coupées à cheval sur plusieurs pages, elles ne sont pas composées en séquence avec le texte normal, mais au contraire sont “flottées” vers un endroit convenable, tel que le sommet d’une page suivante.

Pour les valeurs possibles de *placement* et leur effet sur l’algorithme de placement des flottants, voir Section 5.6 [Floats], page 25.

La version étoilée **figure*** est utilisée quand un document est en mode double-colonne (voir Section 5.2 [twocolumn], page 23). 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 5.6 [Floats], page 25.

Le corps de la figure est composé dans une **parbox** de largeur `\textwidth` and ainsi il peut contenir du texte, des commandes, etc.

L’étiquette est optionnelle ; elle est utilisée pour les double renvois (voir Chapitre 7 [Cross references], page 42). La commande `\caption` spécifie la légende *texte* pour la figure. 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 168).

Cet exemple fabrique une figure à partir d’un graphique. Il nécessite l’un des paquetage **graphics** ou **graphicx**. Le graphique, avec sa légende, est placé au sommet d’une page ou, s’il est rejet 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 185) 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

```
\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 composteur) serait à gauche et les franges à droite. Chaque ligne doit être terminée par la séquence de contrôle `\\`.

8.12.1 \raggedright

La déclaration `\raggedright` correspond à l'environnement `flushleft`. Cette déclaration peut être utilisée au sein d'un environnement tel que `quote` ou d'une `parbox`.

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 où la commande `\end` qui termine l'unité de paragraphe.

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. Chaque ligne doit être terminée par la séquence de contrôle `\\`.

8.13.1 \raggedleft

La déclaration `\raggedleft` correspond à l'environnement `flushright`. Cette déclaration peut être utilisée au sein d'un environnement tel que `as quote` ou d'une `parbox`.

Contrairement à l'environnement `flushright`, la commande `\raggedleft` ne commence 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.

8.14 itemize

Synopsis :

```
\begin{itemize}
\item item1
\item item2
...
\end{itemize}
```

L'environnement `itemize` produit une liste « non ordonnée », « à puces ». Les environnements `itemize` peuvent être imbriqués l'un dans l'autre, jusqu'à quatre niveaux de profondeur. Ils peuvent aussi être imbriqués avec d'autres environnements fabriquant des paragraphes, comme `enumerate` (voir Section 8.7 [enumerate], page 49).

Chaque article d'une liste `itemize` commence avec une commande `\item`. Il doit y avoir au moins une commande `\item` au sein de l'environnement.

Par défaut, les marques de chaque niveau ressemblent à cela :

1. • (puce)
2. -- (tiret demi-cadratin gras)
3. * (asterisque)
4. · (point centré verticalement)

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

L'environnement `itemize` utilise les commandes de `\labelitemi` jusqu'à `\labelitemiv` pour produire les étiquettes par défaut. Ainsi, vous pouvez utiliser `\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 surjacent et la marge de gauche de la liste. (Notez 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 47, et Section 8.7 [enumerate], page 49. Pour des informations concernant les paramètres de contrôle de la disposition des listes, y compris les valeurs par défaut, ainsi que la personnalisation de la disposition, voir Section 8.16 [list], page 56. 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
```

Quelques paramètres qui affectent le formatage des listes :

`\itemindent`

Renforcement supplémentaire avant chaque articles dans une liste ; la valeur par défaut est zéro.

`\labelsep`

Espace entre l'étiquette et le texte d'un article ; la valeur par défaut est `' .5em'`.

`\labelwidth`

Largeur de l'étiquette ; la valeur par défaut est `'2em'`, ou `'1.5em'` en mode deux-colonnes.

`\listparindent`

Renforcement supplémentaire ajoutée aux paragraphes suivants à commencer par le second au sein d'un article de liste ; la valeur par défaut est `'0pt'`.

`\rightmargin`

Distance horizontale entre la marge de droite de la liste et l'environnement surjacent ; la valeur par défaut est `'0pt'`, sauf à l'intérieur des environnements `quote`, `quotation`, et `verse`, où elle est réglée pour être égale à `\leftmargin`.

Paramètres affectant l'espacement vertical entre les articles de liste (plutôt relâché, par défaut).

`\itemsep`

Espace vertical entre les articles. La valeur par défaut est `2pt plus1pt minus1pt` pour les documents en `10pt`, `3pt plus2pt minus1pt` pour ceux en `11pt`, et `4.5pt plus2pt minus1pt` pour ceux en `12pt`.

- `\parsep` Espace vertical supplémentaire entre les paragraphes au sein d'un article de liste. Par défaut vaut la même chose que `\itemsep`.
- `\topsep` Espace vertical entre le premier article et le paragraphe précédent. Pour les listes au niveau le plus haut, la valeur par défaut est `8pt plus2pt minus4pt` pour les documents en `10pt`, `9pt plus3pt minus5pt` pour ceux en `11pt`, et `10pt plus4pt minus6pt` pour ceux en `12pt`. Ces valeurs sont réduites pour les listes avec au moins un niveau d'imbrication.
- `\partopsep` Espace supplémentaire ajouté à `\topsep` quand l'environnement de liste démarre un paragraphe. La valeur par défaut est `2pt plus1pt minus1pt` pour les documents en `10pt`, `3pt plus1pt minus1pt` pour ceux en `11pt`, et `3pt plus2pt minus2pt` pour ceux en `12pt`.

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 110):

```
\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 172.

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 47, Section 8.7 [enumerate], page 49, et Section 8.14 [itemize], page 54). 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 70, voir Section 8.3 [center], page 47, voir Section 8.13 [flushright], page 54).

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 \LaTeX , comme dans l'exemple précédent où il contient à la fois 'Article' et ' $\Roman{\dots}$ '. \LaTeX forme l'étiquette en mettant l'argument *étiquetage* dans une boîte de largeur `\labelwidth`. Si l'étiquette est plus large que cela, le matériel 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 61.

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 103). Utilisez la pour dire à \LaTeX de numéroter les article en utilisant le compteur fourni. Ce compteur est remis à zéro chaque fois que \LaTeX entre dans l'environnement, et le compteur est incrémenté par un chaque fois que \LaTeX 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 sont 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 cloturant l'argument *espacement* à ajouter aussi `\let\makelabel\namedmakelabel`. Ceci composera les étiquette 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 listes 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 106). Les espaces verticaux sont d'ordinaire des longueurs élastiques, avec une composante en plus ou en minus, pour donner à \TeX 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`

Largueur 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 ce matériel 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 classe L^AT_EX ‘`article`’, ‘`book`’, et ‘`report`’ pour les liste de premier niveau vaut `\leftmargin-\labelsep`, (c-à-d. `2em` en mode à une colonne et `1.5em` en mode à 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.

r

`\leftmargin`

Espace horizontal entre la marge de gauche de l’environnement enveloppant (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 `\leftmargin` 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 liste 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 : `\leftmargin` vaut

2.5em (ou 2em en mode à deux colonnes), `\leftmarginii` vaut 2.2em, et `\leftmarginiii` vaut 1.87em.

`\listparindent`

Espace horizontal de renforcement supplémentaire, en plus de `\leftmargin`, pour le deuxième paragraphes et les suivants au sein d’un article de liste. A negative value makes this an “outdent”. Its default value is 0pt.

`\parsep`

Espace vertical entre les paragraphe 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`, and `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’, anet ‘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`.

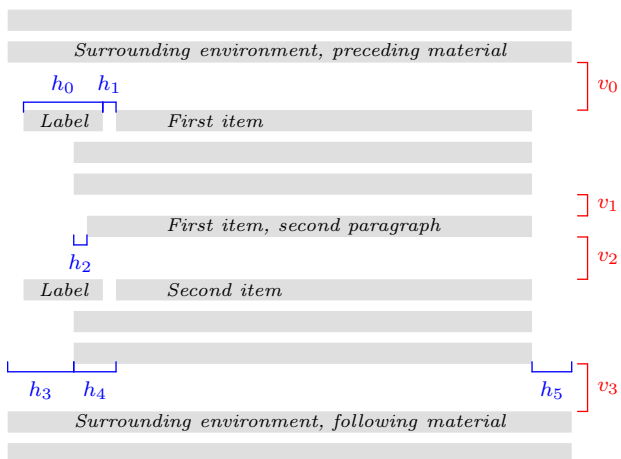
`\rightmargin`

Espace horizontal entre la marge de droite de la liste et la marge de droite de l’environnement l’englobant. Horizontal space between the right margin of the list and the right margin of the enclosing environment. Its default value is 0pt. It must be non-negative.

`\topsep`

Espace vertical ajouté aussi bien au haut qu’au bas de la lise, en plus de `\parskip` (voir Section 15.3 [`\parindent` & `\parskip`], page 110). Les valeurs par défaut pour les trois premiers niveaux dans les classes L^AT_EX ‘book’, anet ‘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`, and `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>

The list's left and right margins, shown above as $h3$ and $h5$, are with respect to the ones provided by the surrounding environment, or with respect to the page margins for a top-level list. The line width used for typesetting the list items is `\linewidth` (voir Section 5.5 [Page layout parameters], page 24). For instance, set the list's left margin to be one quarter of the distance between the left and right margins of the enclosing environment with `\setlength{\leftmargin}{0.25\linewidth}`.

Page breaking in a list structure is controlled by the three parameters below. For each, the \LaTeX default is `-\@lowpenalty`, that is, `-51`. Because it is negative, it somewhat encourages a page break at each spot. Change it with, e.g., `\@beginparpenalty=9999`; a value of 10000 prohibits a page break.

`\@beginparpenalty`

The page breaking penalty for breaking before the list (default -51).

`\@itempenalty`

The page breaking penalty for breaking before a list item (default -51).

`\@endparpenalty`

The page breaking penalty for breaking after a list (default -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 103.

```
\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} % equals \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 [`LATEX` command syntax], page 4.

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 renfoncement 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 droite pour d’éventuels environnements de liste imbriqués.

En particulier, `trivlist` utilise les valeurs courantes des paramètres de liste (voir Section 8.16 [list], page 56), 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 renfoncement 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 112.

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 coupé sur plusieurs pages. Ainsi `minipage` est similaire à la commande `\parbox` (voir Section 20.5 [`\parbox`], page 138), mais contrairement à `\parbox`, d'autres environnements de production de paragraphe peuvent être utilisés au sein d'une `minipage`. This example will be 3 inches wide, and has two paragraphs.

```
\begin{minipage}{3in}
  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 106). 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 valeur par défaut pour *position* et réglez *hauteur* avec `\begin{minipage}[c][2.54cm] 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 \TeX de base).
- b Aligne la ligne du bas de `minipage` avec la ligne de base du texte l'entourant (comme `\vbox` en \TeX 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}
```



```
\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
% et le séparateur décimal
\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{1}{r}{Quartier} & 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)
... commandes picture ...
\end{picture}
```

ou

```
\begin{picture}(largeur,hauteur)(décalagex,décalagey)
... commandes picture ...
\end{picture}
```

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. Ceux-ci ne sont pas couverts dans ce documents ; voir le CTAN.

Vous dites à \LaTeX où mettre les choses dans le dessin en spécifiant leur coordonnées. Une coordonnée est un nombre qui peut avoir un séparateur décimal (point et non virgule) et un signe moins — un nombre comme 5, 0.3 ou -3.1416. Une coordonnée spécifie une

longueur en multiples de l'unité de longueur `\unitlength`, ainsi si `\unitlength` a été réglée à `1cm`, alors la coordonnée `2.54` spécifie une longueur de 2,54 centimètres.

Vous devriez changer la valeur de `\unitlength`, en utilisant la commande `\setlength`, en dehors d'un environnement `picture`. La valeur par défaut est `1pt`.

Le paquetage `picture` redéfinit l'environnement `picture` de sorte qu'où que soit utilisé un nombre au sein de *commandes picture* pour spécifier une coordonnée, on puisse alternativement utiliser une longueur. Notez bien cependant que ceci empêche de dilater/contracter ces longueurs en changeant `\unitlength`.

Une *position* est une paire de coordonnées, telle que `(2.4,-5)`, ce qui spécifie un point d'abscisse `2.4` et d'ordonnée `-5`. Les coordonnées sont spécifiées de la manière habituelle relativement à une origine, qui est normalement dans le coin inférieur gauche du dessin. Notez que lorsque une position apparaît en argument, elle n'est pas entourée d'accolades ; on délimite l'argument avec des parenthèses.

L'environnement `picture` a un argument obligatoire qui est une position (*largeur,hauteur*) qui spécifie la taille du dessin. L'environnement produit une boîte rectangulaire dont les dimensions sont cette *largeur* et cette *hauteur*.

L'environnement `picture` a aussi un argument optionnel de position (*décalagex,décalagey*), qui suit l'argument de taille, et qui peut changer l'origine. (Contrairement aux arguments optionnels ordinaire, cet argument n'est pas contenu entre crochets.) L'argument optionnel donne les coordonnées du point dans le coin inférieur gauche du dessin (par là même déterminant l'origine). 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, donc le coin inférieur gauche est le `(10,20)` et dont le coin supérieur doit être donc le point `(110,220)`. Lorsque vous dessinez la première fois 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.

L'argument obligatoire de l'environnement détermine la taille nominale du dessin. Il n'est pas nécessaire que cela soit en relation à la taille réelle du dessin ; `LATEX` vous laissera sans broncher mettre des choses en dehors de la boîte définie par la taille du dessin, et même en dehors de la page. La taille nominale du dessin est utilisée par `LATEX` pour la détermination de la quantité de place à prévoir pour lui.

Tout ce qui apparaît dans un dessin `picture` est dessiné par la commande `\put`. La commande

```
\put (11.3,-.3){...}
```

met l'objet spécifié par `...` dans le dessin `picture`, avec son point de référence aux coordonnées `(11.3, -.3)`. Les points de référence des divers objets sont décrits plus bas.

La commande `\put` crée une *LR box*. Vous pouvez mettre tout ce qui peut aller dans une `\mbox` (voir Section 20.1 [`\mbox`], page 137) dans l'argument texte de la commande `\put`. Quand vous faites cela, le point de référence devient le coin inférieur gauche de la boîte .

Les commandes `picture` sont décrites dans les rubriques suivantes.

8.19.1 `\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.

Des cercles jusqu'à 40 pt peuvent être dessinés.

8.19.2 `\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 20.4 [`\makebox`], page 137.

8.19.3 `\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.

8.19.4 `\dashbox`

Dessine une boîte avec une ligne en tireté. Synopsis :

```
\dashbox{tlong}(rlargeur,rhauteur)[pos]{texte}
```

`\dashbox` crée un rectangle en tireté 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 tireté a meilleure allure quand les paramètres *rlargeur* et *rhauteur* sont des multiple de *tlong*.

8.19.5 `\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*.

8.19.6 `\line`

Synopsis :

```
\line(xpente,ypente){excursion}
```

Dessine une ligne. Sa pente est telle que la ligne monte de *ypente* verticalement pour toute avancée horizontale de *xpente*. L'*excursion* est le déplacement horizontal total — ce n'est pas la longueur du vecteur, mais la variation en *x*. Dans ce cas spécial des ligne vertical, où $(xpente,ypente)=(0,1)$, 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 *xpente* et *ypente* sont des entiers qui peuvent être positifs, négatifs, ou nuls zero (si les deux valent 0, alors L^AT_EX traite le second comme 1). Avec `\put(x_init,y_init){\line(xpente,ypente){travel}}`, si *xpente* est négatif alors l'extrémité finale de la ligne a sa première coordonnée inférieure à *x_init*. Si *ypente* est négatif alors l'extrémité finale a sa seconde coordonnée inférieure à *y_init*.

Si *excursion* est négatif alors vous obtenez **LaTeX Error: Bad \line or \vector argument**.

De base L^AT_EX 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 *xpente* et *ypente* prennent des valeurs entières de -6 jusqu'à 6. De plus ils doivent être premiers entre eux, de sorte que $(xpente,ypente)=(2,1)$ est licite, mais pas $(xpente,ypente)=(4,2)$ (si 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.

8.19.7 `\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.

Elle diffère de `\thinlines` et `\thicklines` en ce qu'elle n'affecte pas l'épaisseur des lignes obliques, cercles, ou quarts de cercles dessinés par `\oval`.

8.19.8 `\thicklines`

La commande `\thicklines` est une épaisseur de trait alternative pour les lignes verticales et horizontales dans un environnement `picture` ; cf. Section 8.19.7 [`\linethickness`], page 68, et Section 8.19.9 [`\thinlines`], page 69.

8.19.9 `\thinlines`

La commande `\thinlines` est l'épaisseur de trait par défaut pour les lignes horizontales et verticales dans un environnement `picture` ; cf. Section 8.19.7 [`\linethickness`], page 68, et Section 8.19.8 [`\thicklines`], page 69.

8.19.10 `\multiput`

Synopsis :

```
\multiput(x,y)(delta_x,delta_y){n}{obj}
```

La commande `\multiput` copie l'objet *obj* en un motif régulier à travers la figure. *obj* est placé en premier à la position (x, y) , puis à la position $(x + \delta x, y + \delta y)$, et ainsi de suite, *n* fois.

8.19.11 `\oval`

Synopsis :

```
\oval(largeur,hauteur)[portion]
```

La commande `\oval` produit un rectangle aux coins arrondis. L'argument optionnel *portion* vous permet de ne produire qu'une moitié de l'ovale selon ce qui suit :

- t** selectionne la moitié supérieure ;
- b** selectionne la moitié inférieure ;
- r** selectionne la moitié de droite ;
- l** selectionne 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**.

Les « coins » de l'ovale sont fabriqués avec des quarts de cercle d'un rayon maximal de 20 pt, ainsi les « ovales » de grande taille ressembleront plus à des boîtes aux coins arrondis.

8.19.12 `\put`

```
\put(xcoord,ycoord){ ... }
```

La commande `\put` place l'article spécifié par l'argument obligatoire aux coordonnées *xcoord* et *ycoord* fournies.

8.19.13 `\shortstack`

Synopsis :

```
\shortstack[position]{...\...\}
```

La commande `\shortstack` produit une pile d'objets. Les positions valides sont :

- r** Déplace les objets à la droite de la pile.

- l Déplace les objets à la gauche de la pile
- c Déplace les objets au centre de la pile (position par défaut)

Les objets sont séparés avec `\`.

8.19.14 `\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.

8.20 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}
\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}
row1col1 \= row1col2 \= row1col3 \= row1col4 \\
row2col1 \>                \> row2col3 \
```



```
...
\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. Il convient le mieux pour les cas où la largeur de chaque colonne est constante et connue d'avance.

Cet environnement peut être rompu entre plusieurs pages, contrairement à l'environnement `tabular`.

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 de terminer la dernière ligne juste par `\end{tabbing}`, ce qui est équivalent à `\\end{tabbing}`.

L'environnement `tabbing` a un *motif courant de taquets de tabulation* qui est une séquence de n taquets, avec $n > 0$, ce qui crée n colonnes correspondantes. Immédiatement après `\begin{tabbing}` on a $n = 1$, c.-à-d. le motif courant de taquets contient un seul taquet, désigné comme le taquet $N^{\circ} 0$, et dont la position est le début de ligne, c.-à-d. que la première colonne commence au début de la ligne.

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 $N^{\circ} 0$, mais cela peut être changé avec les commandes `\+` et `\-`. Au début de chaque ligne tabulée, le taquet courant est initialement égal au taquet de début de ligne courant.

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 $N^{\circ} 0$, 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.

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 un taquet à la position courante. La n ème commande `\=` de la ligne tabulée courante définit le taquet $N^{\circ} n$, ou le redéfinit s'il avait été déjà défini dans une ligne tabulée précédente. Lorsque le nombre de taquets dans le motif courant est $(n + 1)$, c'est à dire qu'on a des taquets $N^{\circ} 0$ à $N^{\circ} n$, alors la

$(n + 1)$ ième commande `\=` définit le taquet $N^\circ n + 1$, ce qui porte le nombre de taquets 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 numéros $N^\circ n$ et $N^\circ m$ distincts, voire de faire en sorte que les taquets $N^\circ n$ et $N^\circ m$, avec $n < m$, soient à des positions telles que le taquet $N^\circ m$ soit avant le taquet $N^\circ n$.

`\>` (tabbing)

Change le taquet courant au taquet suivant dans l'ordre logique, c.-à-d. si le taquet courant est le taquet $N^\circ n$, alors, en supposant qu'il y ait au moins $n + 2$ taquets définis, on passe au taquets $N^\circ n + 1$, 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 $N^\circ n$ avec $n > 0$. Le taquet courant devient alors le taquet $N^\circ n - 1$.

Il est en fait possible d'avoir m commandes `\<` consécutives à condition que $m \leq n$, le taquet courant devient alors le taquet $N^\circ (n - m)$. 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 $N^\circ 2$, 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 défini, et que le taquet de début de ligne courant est le taquet $N^\circ n$, 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 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 $N^\circ 0$ au taquet $N^\circ 1$:

```
\begin{tabbing}
Vive \+\=la \=France\\
Vive \>la\>République\\
\end{tabbing}
```

En revanche le code suivant fonctionne, car lorsque la commande `\+` est passée on a trois taquets définis, les taquets $N^\circ 0$ à 2, on peut donc passer du taquet $N^\circ 0$ au taquet $N^\circ 1$.

```
\begin{tabbing}
```

```
Vive \=la \=France\+\\
Vive \>la\>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 $N^{\circ} n$, 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 $N^{\circ} n - m$. 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 `\tabbingospace`, 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 160). 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 sauvegardés par le dernier `\pushtabs`.

`\pushtabs`

Sauvegarde dans une pile dédiée à cet effet le motif courant de taquets de tabulation. 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 l'espace produit par la le caractère de contrôle espace (`\`, c.-à-d. une contr'oblique suivie d'un caractère espace) n'est pas gobé.

Cet exemple compose un fonction en Pascal dans un format traditionnel :

```
{\ttfamily
\begin{tabbing}
function \= fact(n : integer) : integer;\+\\
    begin \=+\+\\
        if \= n $>$ 1 then\+\\
            fact := n * fact(n-1)\-\\
        else\+\\
            fact := 1;\-\\
        end;\+\\
\end{tabbing}}
```

Le résultat ressemble à cela :

```
function fact(n : integer) : integer;
begin
    if n > 1 then
        fact := n * fact(n-1);
    else
        fact := 1;
    end;
```

Ceci n'était qu'un exemple illustrant `tabbing`, pour formater du code considérez plutôt les paquetages `algorithm2e` pour du pseudo-code, et `listings` pour du vrai code.

Voici un autre exemple où le taquet N° 2 est entre le taquet N° 0 et le taquet N° 1 :

```
{\ttfamily
\begin{tabbing}
    12\=34\=\kill
    12345678\=9\\
    1\>2\>3\\
\end{tabbing}}
```

Le résultat ressemble à cela :

```
123456789
1 3 2
```

Voici un exemple où les taquets N° 0 et N° 1 sont confondus, ainsi que les taquets N° 2 et N° 3, ce qui permet de superposer le texte aux taquets impairs sur celui au taquets pairs,

L'étiquette est optionnelle ; elle est utilisée pour les double renvois (voir Chapitre 7 [Cross references], page 42). La commande `\caption` spécifie la légende *texte* pour la figure. La légende est numérotée par défaut. Si *titredt* est présent, il est utilisé dans la liste des tableaux au lieu de *titre* (voir Section 25.1 [Table of contents etc.], page 168).

```
\begin{table}[b]
  \centering
  \begin{tabular}{r|p{2in}} \hline
    Un &Le plus solitaire des nombres \\
    Deux &Peut être aussi triste qu'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}
column 1 entry & column 2 entry ... & column n entry \\
...
\end{tabular}
```

ou

```
\begin{tabular*}{largeur}[pos]{cols}
column 1 entry & column 2 entry ... & column n entry \\
...
\end{tabular*}
```

Ces environnements produisent une boîte consistant d'une séquence de ligne 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}
```

Le format vertical de deux colonnes alignées à gauche, avec une barre verticale entre elles, est spécifié par l'argument `{l|l}` de `tabular`. Les colonnes sont séparées avec une esperluette `&`. On crée un filet horizontal entre deux lignes avec `\hline`. La fin de chaque ligne est marquée avec une double-contr'oblique `\\`. 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 :

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 ...`

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 possible : `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 du matériel intercolonne.

- `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. Le matériel *texte ou espace* est composé en mode LR. Ce texte est fragile (voir Section 12.11 [`\protect`], page 101).

Ce spécificateur est optionnel : à moins que vous mettiez dans votre propre @-expression alors les classes `book`, `article` et `report` de \LaTeX mettent de chaque côté de chaque colonne un espace de longueur `\tabcolsep`, qui par défaut vaut `'6pt'`. C'est à dire que par défaut des colonnes adjacentes sont séparées de 12pt (ainsi `\tabcolsep` est un nom trompeur puisque ce n'est pas la séparation entre des colonnes de tableau). Aussi, par défaut un espace de 6pt vient après la première colonne ainsi qu'après la dernière colonne, à moins que vous placiez un `@{...}` ou `|` à cet endroit.

Si vous l'emportez sur la valeur par défaut et utilisez une expression en `@{...}` alors 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}
```

Cet exemple montre du texte, une virgule de séparation décimale, entre les colonnes, arrangé de sorte que les nombres dans la table sont 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`.

```
\begin{center}
  \begin{tabular*}{10cm}{l@{\ \ldots\extracolsep{\fill}}l}
    Tomber sept fois, se relever huit
    & 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*.

Note que les sauts de ligne de type double-contr'oblique `\\` 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 contr'oblique comme une fin de ligne.

`*{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 `*{...}{...}`.

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 `'.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 ente 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` :

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|}
  \textit{ID}      & \multicolumn{2}{c}{\textit{Nom}} & \textit{Âge} \\ \hline % ligne 1
  978-0-393-03701-2 & O'Brian & Patrick & 55 \\
  \dots & & & \\
\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 partie 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` contigüe à cette entrée multi-colonne. Pour affecter cette zone, cet argument peut contenir des barre verticale `|` 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 intercolonne 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 entrée 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 entrée 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 ligne 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}
```

Noter ue 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 `\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*.

8.23.3 `\hline`

La commande `\hline` dessine une ligne horizontale de la largeur de l’environnement `tabular` ou `array` surjacent. Elle est la plupart du temps utilisé pour dessiner une ligne au sommet, bas, et entre les lignes d’un tableau.

8.23.4 `\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 `@{...}`.

8.24 `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.

Dans la classe `article`, cette liste de références est intitulée 'Références' et ce titre est stocké dans la macro `\refname` ; dans la classe `report`, elle est intitulée 'Bibliographie' et ce titre est stocké dans la macro `\bibname`.

Vous pouvez changer le titre en redéfinissant celle des commandes `\refname` ou `\bibname` qui est applicable selon la classe :

- Pour les classes standards dont le niveau le plus élevé de rubricage est `\chapter` le titre est dans la macro `\bibname`, par exemple les classes `book` et `report` tombent dans cette catégorie.
- Pour les classes standards dont le niveau le plus élevé de rubricage est `\section` le titre est dans la macro `\refname`, par exemple la classe `article` tombe dans cette catégorie.

Typiquement il n'est ni nécessaire ni désirable de redéfinir directement `\refname` ou `\bibname` ; les paquetages linguistiques comme `babel` s'en chargent.

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`. C'est typiquement donné comme `9` pour des bibliographies de moins de 10 références, `99` pour celles qui en ont moins de 100, etc.

8.24.1 `\bibitem`

Synopsis :

```
\bibitem[étiquette]{clef_de_cite}
```

La commande `\bibitem` génère une entrée étiquetée par *étiquette*. Si l'argument *étiquette* est omis, un numéro est automatiquement généré en utilisant le compteur `enumi`. L'argument *clef_de_cite* est une *clef de référence bibliographique* consistant en une séquence quelconque de lettres, chiffres, et signes de ponctuation ne contenant aucune virgule.

Cette commande écrit une entrée dans le fichier `.aux`, et cette entrée contient les *étiquette* et *clef_de_cite* de l'article considéré. Lorsque le fichier `.aux` est lu par la commande `\begin{document}`, l'*étiquette* de l'article est associée à sa *clef_de_cite*, ce qui a pour effet que toute référence à *clef_de_cite* avec une commande `\cite` (voir Section 8.24.2 [`\cite`], page 82) produit l'*étiquette* associée.

8.24.2 `\cite`

Synopsis :

```
\cite[subcite]{clefs}
```

L'argument *clefs* est une liste d'une ou plus de clefs de références bibliographiques (voir Section 8.24.1 [`\bibitem`], page 81), séparées par des virgules. Cette commande génère une référence bibliographique au sein du textes contenant les étiquettes associées aux clefs contenues dans *clefs* par les entrées respectives au sein du fichier `.aux`.

Le texte de l'argument optionnel *subcite* apparaît après l'étiquette de référence. Par exemple, `\cite[p.~314]{knuth}` pourrait produire '`[Knuth, p. 314]`'.

8.24.3 `\nocite`

Synopsis :

```
\nocite{clefs}
```

La commande `\nocite` ne produit aucun texte, mais écrit *clefs*, ce qui est une liste d'une ou plusieurs clefs de référence bibliographique dans le fichier `.aux`.

8.24.4 Utilisation de Bib \TeX

Si vous utilisez le programme Bib \TeX d'Oren Patashnik (ce qui est fortement recommandé si vous avez besoin d'une bibliographie de plus d'un ou deux titre) pour maintenir votre bibliographie, vous n'utilisez pas l'environnement `thebibliography` (voir Section 8.24 [`thebibliography`], page 81). Au lieu de cela, vous incluez les lignes :

```
\bibliographystyle{stylebib}
\bibliography{ficbib1,ficbib2}
```

La commande `\bibliographystyle` ne produit aucune sortie en elle-même. En revanche, elle définit le style dans lequel la bibliographie est à produire : *stylebib* fait référence à un fichier *stylebib.bst* définissant l'aspect que doivent avoir vos référence bibliographique. Les nom *stylebib* standards distribués avec Bib \TeX sont :

<code>alpha</code>	Classé alphabétiquement. Les étiquettes sont formées à partir du nom de l'auteur et de l'année de publication.
<code>plain</code>	Classé alphabétiquement. Les étiquettes sont des numéros.
<code>unsrt</code>	Comme <code>plain</code> , mais les entrées sont dans l'ordre où les références y sont faites dans le texte.
<code>abbrv</code>	Comme <code>plain</code> , mais les étiquettes sont plus compactes.

De plus, de nombreux autres styles Bib \TeX existent faits sur mesure pour satisfaire les exigences de diverses publications. Voir <http://mirror.ctan.org/biblio/bibtex/contrib>.

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 donnée au format Bib \TeX . Seules les entrées désignées par les commandes `\cite` ou `\nocite` sont listées dans la bibliographie.

8.25 theorem

Synopsis :

```
\begin{theorem}
  texte-théorème
\end{theorem}
```

L'environnement `theorem` produit « Théorème *n* » en gras suivi de *texte-théorème*, où les possibilité de numérotation pour *n* sont décrites par `\newtheorem` (voir Section 12.9 [`\newtheorem`], page 99).

8.26 titlepage

Synopsis :

```
\begin{titlepage}
  texte
\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. Le formatage de la page de titre vous est laissé. La commande `\today` peut être utile sur les pages de titre (voir Section 23.8 [`\today`], page 163).

Vous pouvez utiliser la commande `\maketitle` (voir Section 18.1 [`\maketitle`], page 131) pour produire une page de titre standard sans utiliser l'environnement `titlepage`.

8.27 verbatim

Synopsis :

```
\begin{verbatim}
  literal-texte
\end{verbatim}
```

L'environnement `verbatim` est un environnement qui fabrique des paragraphes dans lequel \LaTeX produit exactement ce qui vous y taez ; par exemple le caractère `\` produit `'\` à l'impression . Il, fait de \LaTeX une machine à écrire, c.-à-d. que le retour chariot et les blancs y ont un tel effet.

L'environnement `verbatim` utilise une police de type tapuscrit (encore appelé « machine à écrire ») à chasse fixe (`\tt`).

8.27.1 \verb

Synopsis :

```
\verb<cartexte-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, `_`).

8.28 `verse`

Synopsis :

```
\begin{verse}  
line1 \\  
line2 \\  
...  
\end{verse}
```

L'environnement `verse` est conçu pour la poésie, bien que vous pouvez lui trouver d'autres usages.

Les marges sont renforcées sur la gauche et la droite, les paragraphes ne le sont pas, et le texte n'est pas justifié. Séparez le ligne de chaque strophe avec `\\`, et utilisez une ou plusieurs lignes vides pour séparer les strophes.

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 la rupture de ligne (et de page) pour vous, mais dans certains environnements, vous faites la rupture vous-même avec la commande `\`, et vous pouvez toujours forcer manuellement les ruptures.

9.1 `\`

Synopsis :

```
\[espaceenplus]
```

ou

```
\*[espaceenplus]
```

Commencer une nouvelle ligne. L'argument optionnel *espaceenplus* spécifie l'espace supplémentaire vertical à insérer avant la ligne suivante. Ceci peut être une longueur 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.

Les sauts de ligne explicites au sein du corps de texte sont inhabituels en L^AT_EX. en particulier, pour commencer un nouveau paragraphe laissez plutôt une ligne à blanc. Cette commande est utilisée principalement en dehors du flux principal de texte comme dans un environnement `tabular` ou `array`.

Dans des circonstances ordinaires (par ex. en dehors d'une colonne spécifiée par `p{...}` dans un environnement `tabular`) la commande `\newline` est un synonyme de `\` (voir Section 9.3 [`\newline`], page 85).

En plus de commencer une nouvelle ligne, la forme étoilée `*` dit à L^AT_EX de ne pas commencer une nouvelle page entre les deux lignes, en émettant un `\nobreak`.

```
\title{Mon histoire~: \[7mm]
      un conte pathétique}
```

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.

`\restorecr` restaure le comportement normal vis à vis des saut de ligne.

9.3 `\newline`

Dans du texte ordinaire ceci est équivalent à une double-contr'oblique (voir Section 9.1 [`\`], page 85) ; cela rompt la ligne, sans dilatation du texte le précédant.

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 un saut de ligne à l'intérieur de la colonne, c.-à-d. qu'il ne rompt pas la ligne entière. 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 `\-` commande dit à \LaTeX qu'il peut faire une césure du mot à cet endroit. \LaTeX est très 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.

Quand vous insérez des commandes `\-` dans un mot, le mot ne pourra avoir de césure qu'à ces endroits et non dans aucun des points de césure que \LaTeX aurait pu choisir sinon.

9.5 \discretionary (point de césure généralisé)

Synopsis :

```
\discretionary{avant-saut}{après-saut}{sans-saut}
```

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 9.8 [`\hyphenation`], page 87) et `\-` (voir Section 9.4 [`\-` (`hyphenation`)], page 86).

`\discretionary` indique qu'un saut de ligne est possible à cet endroit. Dans le cas d'un saut de ligne, le texte *avant-saut* est placé immédiatement avant le saut, et le texte *après-saut* immédiatement après. Dans le cas sans saut de ligne le texte *sans-saut* est imprimé.

Les arguments de `\discretionary` ne peuvent contenir que des caractères, des boîtes ou des crénages.

L'usage typique de `\discretionary` est par exemple de contrôler la césure au sein d'une formule mathématique en mode ligne (voir aussi Section 16.6 [Math miscellany], page 126). 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 contr'oblique 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

La déclaration `\fussy` (ce qui est le réglage par défaut) a pour effet que $\text{T}_{\text{E}}\text{X}$ fait le difficile à propos des saut de ligne. Ceci évite d'ordinaire trop d'espace entre les mots, au prix d'occasionnellement une boîte trop pleine (`'overfull \hbox'`).

Cette commande annule l'effet d'une commande `\sloppy` précédente (voir Section 9.7 [`\sloppy`], page 87).

9.7 \sloppy

La déclaration `\sloppy` a pour effet que $\text{T}_{\text{E}}\text{X}$ fasse moins le difficile à propos des sauts de ligne. Ceci évite les boîtes trop pleines, au prix d'un espacement lâche entre les mots.

L'effet dure jusqu'à la prochaine commande `\fussy` (voir Section 9.6 [`\fussy`], page 87).

9.8 \hyphenation

Synopsis :

```
\hyphenation{mot-un mot-deux}
```

La commande `\hyphenation` déclare les points de césure autorisé avec un caractère - au sein des mots donnés. Les mots sont séparés par des espaces. $\text{T}_{\text{E}}\text{X}$ ne fera de césure que si le mot correspond exactement, aucune désinence n'est essayée. Des commandes `\hyphenation` multiple s'accumulent. Voici quelques exemples en anglais (les motifs de césure par défaut de $\text{T}_{\text{E}}\text{X}$ manquent les césures de ces mots) :

```
\hyphenation{ap-pen-dix col-umns data-base data-bases}
```

9.9 \linebreak & \nolinebreak

Synopsis :

```
\linebreak[priorité]
\nolinebreak[priorité]
```

Par défaut, la commande `\linebreak` (`\nolinebreak`) force (empêche) un saut de ligne à la position courante. Pour `\linebreak`, les espaces de la ligne sont dilatés jusqu'à ce qu'elle s'étende jusqu'à la marge de droite, comme d'habitude.

Avec l'argument optionnel *priorité*, vous pouvez convertir la commande d'une simple demande à une requête. La *priorité* doit être un nombre compris entre 0 et 4. Plus ce nombre est grand, et plus la requête est insistante.

10 Saut de page

L^AT_EX commence de nouvelles pages de façon asynchrone, lorsque suffisamment de matériel a été accumulé pour remplir une page. D'ordinaire ceci se produit automatiquement, mais parfois on peut désirer influencer les sauts.

10.1 `\clearpage` & `\cleardoublepage`

La commande `\cleardoublepage` finit la page courante et fait que L^AT_EX imprime tous les flottants, tableaux et figures, qui sont apparus en entrée jusqu'à ce point et sont en attente. Dans le cas d'une impression en style `twoside`, elle fait aussi que la nouvelle page et une page de droite (numérotée impaire), en produisant si nécessaire une page à blanc.

La commande `\clearpage` finit la page actuelle et fait que L^AT_EX imprime tous les flottants, tableaux et figures, qui sont apparus en entrée jusqu'à ce point et sont en attente.

10.2 `\newpage`

Synopsis :

```
\newpage
```

La commande `\newpage` finit la page courante. Cette commande est robuste (voir Section 12.11 [`\protect`], page 101).

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 débourent les flottants pendants (voir Section 10.1 [`\clearpage` & `\cleardoublepage`], page 88). Et, si L^AT_EX est en mode à deux colonne 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 89), 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 le matériel 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 Note 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 ne peux pas le savoir ; je ne lis
jamais 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.

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 acutelle 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 103). Pour rendre ce changement global mettez cela dans le préambule. Si vous faites ce changement local alors vous désirerez sans doute réinitialiser le compte 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 seulement en mode paragraphe le plus externe). 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 63.

11.2 `\footnotemark`

Synopsis :

```
\footnotemark
```

Sans l'argument optionnel, la commande `\footnotemark` place dans le texte le numéro de la note courante. Cette commande peut être utilisée en mode paragraphe interne. Vous pouvez donner le texte de la note séparément la commande `\footnotetext`.

Cette commande pour produire plusieurs marque de note en bas de page faisant référence à la même note avec

```
\footnotemark[\value{footnote}]
```

après la première commande `\footnote`.

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 This commande peut apparaître n'importe où après la commande `\footnotemark`. L'argument optionnel *numéro* change le numéro de note affiché. La commande `\footnotetext` ne doit être utilisée qu'en mode paragraphe externe.

11.4 Notes en bas de page dans un tableau

Au sein d'un environnement `table` la commande `\footnote` ne fonctionne pas. Par exemple, si le code ci-dessous apparaît sans plus alors la note disparaît tout simplement ; il y aura une marque de note dans la cellule du tableau mais rien n'est composé en bas de la page.

```
\begin{center}
\begin{tabular}{l|l}
\textsc{Ship} & \textsc{Book} \\ \hline
\textit{HMS Sophie} & Master and Commander \\
\textit{HMS Polychrest} & Post Captain \\
\textit{HMS Lively} & Post Captain \\
\textit{HMS Surprise} & A number of books\footnote{Starting with
HMS Surprise.}
\end{tabular}
\end{center}
```

La solution est d'entourer l'environnement `tabular` d'un environnement `minipage`, comme ci-après (voir Section 8.18 [`minipage`], page 63).

```
\begin{center}
\begin{minipage}{.5\textwidth}
.. matériel tabulaire ..
\end{minipage}
\end{center}
```

La même technique marche au sein d'un environnement `table` flottant (voir Section 8.22 [`table`], page 75). To get the footnote at the bottom of the page use the `tablefootnote` package, comme illustré dans l'exemple ci-apès. Si vous placez `\usepackage{tablefootnote}` dans le préambule et utilisez le code source ci-dessous, alors la note apparaîtra en bas de la page et sera numérotée dans la même séquence que les autres notes en bas de page.

```
\begin{table}
```

```

\centering
\begin{tabular}{l|l}
\textsc{Date} & \textsc{Campagne} \\ \hline
1862 & & \textsc{Fort Donelson} \\
1863 & & \textsc{Vicksburg} \\
1865 & & \textsc{Armée de Virginie du Nord} \footnote{Fin de la
guerre.}
\end{tabular}
\caption{Forces capturées par le général Grant}
\end{table}

```

11.5 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.

11.6 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 dimension 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 nominal pour des police de taille `\footnotesize` (voir Section 4.2 [Font sizes], page 19), 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`

`\newcommand` définit une nouvelle commande, et `\renewcommand` la redéfinit.

Synopsis :

```

\newcommand{\cmd}[nargs] [optargdéfaut]{défn}
\renewcommand{\cmd}[nargs] [optargdéfaut]{défn}
\newcommand*{\cmd}[nargs] [optargdéfaut]{défn}
\renewcommand*{\cmd}[nargs] [optargdéfaut]{défn}

```

Définit ou redéfinit une commande (voir aussi la discussion de `\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.

cmd Obligatoire ; `\cmd` est le nom de la commande. 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 commande peut prendre, 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 present, 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 utilisé avec des crochets à la suite, comme dans `\cmd[maval]`, alors au sein de *défn* le premier paramètre positionnel `#1` se développe en *maval*. Par contre si `\cmd` est appelé sans crochet à la suite, alors au sein de *défn* le paramètre positionnel `#1` se développe en la valeur par défaut *optargdéfaut*. Dans tous les cas, tout argument obligatoire sera désigné en commençant par `#2`.

Omettre `[maval]` dans un appel n'est pas la même chose qu'avoir les crochets ne contenant rien, comme dans `[]`. Dans le premier cas le paramètre `#1` se développe en *optargdéfaut* ; alors que dans le second `#1` se développe en une chaîne vide.

défn Le texte par lequel substituer chaque occurrence de `\cmd` ; le paramètre positionnel `#n` au sein de *défn* est remplacée par le texte passé comme *nième* argument.

TEX ignore les espaces dans le code source à la suite d'une commande (ou de toute séquence de contrôle), comme dans `\cmd`. Si vous désirez une espace réellement à cet endroit, une solution est de taper `{}` après la commande (`\cmd{}`), et un 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 l'abréviation `\JM` par le texte plus long de la définition.

La redéfinition d'une commande existante est similaire : `\renewcommand{\symboleccqfd}{\small CQFD}`.

Voici la définition d'une commande avec un argument obligatoire :

```
\newcommand{\defref}[1]{Définition~\ref{#1}}
```

Alors, `\defref{def:basis}` se développe en `Définition~\ref{def:basis}`, 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 `','`.

Les accolades autour de *defn* ne définissent pas un groupe, c.-à-d. qu'elle ne délimitent pas la portée du résultat du développement de *defn*. Ainsi la définition `\newcommand{\nomnavire}[1]{\it #1}` est problématique ; dans cette phrase

```
Le \nomnavire{Monitor} rencontra le \nomnavire{Merrimac}.
```

les mots `'rencontra le'` sont mis incorrectement en italique. Une paire supplémentaire d'accolades est nécessaire, comme cela : `\newcommand{\nomnavire}[1]{\it #1}`. Ces accolades font partie de la définition et par conséquent définissent bien un groupe.

12.2 \providecommand

Définit une commande, du moment qu'aucune commande de même nom n'existe déjà.

Synopsis :

```
\providecommand{cmd}[nargs][optargdéfaut]{defn}
\providecommand*{cmd}[nargs][optargdéfaut]{defn}
```

Si aucune commande de ce nom n'existe alors ceci a le même effet que `\newcommand` (voir Section 12.1 [`\newcommand` & `\renewcommand`], page 93). Si une commande de ce nom existe déjà alors cette définition est sans effet. Ceci est particulièrement utile dans un fichier de style, ou dans tout fichier susceptible d'être chargé plus d'une fois.

12.3 \makeatletter et \makeatother

Synopsis :

```
\makeatletter
... définition de commande comprenant @ dans leur nom ..
\makeatother
```


Utilisez cette paire de commande quand vous redéfinissez les commandes L^AT_EX dont le nom comprend un caractère arobase @. La déclaration `\makeatletter` a pour effet que le caractère arobase @ ait le code de catégorie des lettres, c.-à-d. 11. La déclaration `\makeatother` règle de code de catégorie de @ à 12, sa valeur d'origine.

À mesure que chaque caractère est lu par T_EX pour L^AT_EX, un code de catégorie lui est assigné. On appelle aussi ce code *catcode* pour faire court. Par exemple, la contre-oblique \ reçoit le catcode 0. Les noms de commande consistent en un caractère de catégorie 0, d'ordinaire une contr'oblique, 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'arobase 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 comprend @, 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'arobase ait le catcode d'une lettre, à savoir 11.

Pour une liste complète des macros contenant une arobase 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

Vous l'avez sans doute remarqué, beaucoup d'environnements ou commandes standards de L^AT_EX existent sous une variante étoilée. Vous pouvez avoir envie de reprendre cette idée quand vous définissez vos commandes et arguments.

Si vous avez lu Section 12.8 [`\newenvironment` & `\renewenvironment`], page 97, vous aurez compris que c'est immédiat pour les environnements car l'étoile est un caractère autorisé dans les noms d'environnements. Il vous suffit donc de faire `\newenvironment{monenv}` et `\newenvironment{monenv*}` avec les définitions souhaitées.

Pour les commandes, c'est plus compliqué car l'étoile ne peut pas faire partie du nom de la commande. Il y aura donc une première commande, qui devra être capable de regarder si elle est ou non suivie d'une étoile et d'adapter son comportement en conséquence. Cette première commande ne pourra pas accepter d'argument, mais pourra faire appel à des commandes qui en acceptent. Par ailleurs, on utilise la commande `\@ifstar` interne du noyau L^AT_EX et vous pouvez vous reporter à « Section 12.3 [`\makeatletter` & `\makeatother`], page 94, » pour comprendre pourquoi cela implique l'usage d'un bloc `\makeatletter... \makeatother`.

Dans cet exemple, on définit `\ciel` comme une commande avec un argument obligatoire et admettant une variante étoilée `\ciel*` également avec un argument obligatoire :

```
\makeatletter
\newcommand*\ciel@starred[1]{ciel #1 étoilé}
\newcommand*\ciel@unstarred[1]{ciel #1 non étoilé}
\newcommand\ciel{\@ifstar{\ciel@starred}{\ciel@unstarred}}
\makeatother
```

ainsi, `\ciel{bleu}` composera « ciel bleu non étoilé » tandis que `\ciel*{nocturne}` composera « ciel nocturne étoilé ».

Notez que les commandes intermédiaires `\ciel@starred` et `\ciel@unstarred` comportent un `@` dans leur nom ce qui est possible puisque elles apparaissent au sein d'un bloc `\makeatletter... \makeatother`. Ceci assure qu'elles ne puissent pas être utilisées directement dans le document. L'utilisation de `@` dans le nom comporte certes un risque de collision avec une commande interne du noyau L^AT_EX, mais une telle collision est facile à détecter et corriger car `\newcommand*{\cmd}` vérifie que si la commande `\cmd` n'est pas déjà définie et génère une erreur si c'est le cas.

Voici un autre exemple, où la commande `\agentsecret` comporte deux arguments, tandis que sa variante étoilée `\agentsecret*` n'en présente qu'un seul :

```
\makeatletter
\newcommand*\agentsecret@starred[1]{\textsc{#1}}
\newcommand*\agentsecret@unstarred[2]{\textit{#1} \textsc{#2}}
\newcommand\agentsecret{\@ifstar{\agentsecret@starred}
{\agentsecret@unstarred}}
\makeatother
```

Avec cette définition, la célèbre réplique de l'agent 007 :

```
Je m'appelle \agentsecret*{Bond}, \agentsecret{James}{Bond}.
```

est équivalente à :

```
Je m'appelle \textsc{Bond}, \textit{James} \textsc{Bond}.
```

Un dernier mot : dans le nom d'une commande, l'étoile est comme une sorte d'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 T_EXnique 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 commune est de ne jamais insérer de tels espaces.

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 103, pour plus d'information à propos des compteurs.

12.6 `\newlength` : allouer une longueur

Alloue un nouveau registre de *longueur*. Synopsis :

```
\newlength{\arg}
```

Cette commande prend un argument obligatoire qui doit commencer par une contr'oblique (`\`). Elle crée un nouveau registre de longueur nommé `\arg`, ce qui constitue un emplacement où détenir des longueurs (élastiques) telles que `1in plus.2in minus.1in` (ce que \TeX de base appelle un registre `skip`, ou registre de pas). Le registre est créé avec une valeur initiale de zéro. La séquence de contrôle `\arg` doit ne pas être déjà définie.

Voir Chapitre 14 [Lengths], page 106, pour plus d'information à propos des longueurs.

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 du matériel composé, pour l'utiliser plusieurs fois (voir Chapitre 20 [Boxes], page 137) ou pour le mesurer ou le manipuler. Le nom `\cmd` doit commencer par une contr'oblique, et ne doit pas être déjà défini.

L'allocation d'une boîte est globale. Cette commande est fragile (voir Section 12.11 [`\protect`], page 101).

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 contr'oblique `\`. 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, et que `\begin{env}` est utilisé avec des crochets à la suite, comme dans `\begin{env}[maval]`, alors, au sein de *défdébut*, le paramètre positionnel #1 se développe en *maval*. Si par contre `\begin{env}` est appelé sans être suivi de crochet, alors, au sein de *défdébut*, le paramètre positionnel #1 se développe en la valeur par défaut, c.-à-d. *optargdéfaut*. Dans les deux cas, le premier paramètre positionnel faisant référence à argument obligatoire est #2. Omettre [*maval*] dans l'appel est différent d'avoir des crochets sans contenu, c.-à-d. []. 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 *nième* paramètre positionnel, (c.-à-d. #*n*), est remplacé au sein de *défdébut* par le texte du *nième* 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 au matériel 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 20.8 [\sbox], page 139).

```
\newsavebox{\quoteauthor}
\newenvironment{citequote}[1][Corneille]{%
  \sbox\quoteauthor{#1}%
  \begin{quotation}
} {%
  \hspace{1em plus 1fill}---\usebox{\quoteauthor}
  \end{quotation}
}
```

12.9 \newtheorem

Définit une nouvel *environnement simili-théorème*. Synopsis :

```
\newtheorem{nom}{titre}[numéroté_au_sein_de]
\newtheorem{nom}[numéroté_comme]{titre}
```

Ces deux commandes créent un environnement simili-théorème *nom*. Utiliser la première des deux formes,

```
\newtheorem{nom}{titre}[numéroté_au_sein_de]
```

avec l'argument optionnel après le second argument obligatoire, crée un environnement dont le compteur est subordonné au compteur déjà existant *numéroté_au_sein_de*, c.-à-d. qui est réinitialisé à it will chaque réinitialisation de *numéroté_au_sein_de*.

Utiliser la seconde forme,

```
\newtheorem{nim}[numéroté_comme]{titre}
```

avec l'argument optionnel entre les deux arguments obligatoire, crée une environnement dont le compteur partage le compteur déjà défini *numéroté_comme*.

Vous pouvez spécifier l'un seul de *numéroté_au_sein_de* et *numéroté_comme*, ou ni l'un ni l'autre, mais non les deux à la fois.

Cette commande crée un compteur nommé *name*. De plus, à moins que l'argument optionnel *numéroté_comme* soit utilisé, la valeur courante de \ref est celle de \thenuméroté_au_sein_de (voir Section 7.3 [\ref], page 44).

Cette déclaration est globale. Elle est fragile (voir Section 12.11 [\protect], page 101).

nom Le nom de l'environnement. Il ne doit pas commencer avec une contr'oblique ('\''). 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 re-initialisé, 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 simil-théorème déjà défini. Le nouvel environnement se numéroté en séquence avec *numéroté_comme*.

En l'absence des arguments optionnels les environnement 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}
```

Parce que l'exemple suivant spécifie `section` comme argument optionnel *numéroté_au_sein_de* à `\newtheorem`, l'exemple, reprenant le même corps de document que l'exemple précédent, donne '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 desquelle appelle le nouvel environnement `thm` pour utilise 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 contr’oblique (`\`).

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 fichier `.fd` ou à travers l’utilisation d’un moteur qui sait accéder au système de polices de caractère, tel que $\Xe\LaTeX$ (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 autre dimension relative à la taille de conception de la police, ce qui est une valeur définie dans le fichier en `.tfm`.

Cet exemple définit deux polices équivalente 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 saut 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 partie 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 51), dans la commande `\thanks{...}` (voir Section 18.1 [`\maketitle`], page 131), et dans les expressions en @ des environnements `tabular` et `array` (voir Section 8.23 [tabular], page 76).

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 104, ou Section 13.4 [`\setcounter`], page 104.

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 \(\ e=mc^2 \)}
```


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.

```
part paragraphe figure enumi
chapter subparagraph table enumii
section page footnote enumiii
subsection équation mpfootnote enumiv
subsubsection
```

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 lettre en bas de casse : ‘a’, ‘b’, ...

`\Alph` Utilise des lettres capitales : ‘A’, ‘B’, ...

`\arabic` Utilise des numéro 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>	
asterisque-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 \LaTeX attend un nombre. Elle ne doit pas être précédée par `\protect` (voir Section 12.11 [`\protect`], page 101).

La commande `\value` n'est pas utiliser pour composer la valeur du compteur. Voir Section 13.1 [`\alph \Alph \arabic \roman \Roman \fnsymbol`], page 103.

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 104, à ceci près qu'elle définit également la valeur courante de `\ref` comme le résultat de `\thecounter`.

13.7 `\stepcounter{compteur}`

La commande `\stepcounter` ajouter un à *compteur* et re-initialise tous les compteurs subsidiaires.

13.8 `\day` `\month` `\year`: Predefined compteurs

L^AT_EX définit des compteurs pour le quantième du mois (`\day`, 1–31), le mois de l'année (`\month`, 1–12), et l'année (`\year`, de l'ère chrétienne). Quand T_EX démarre, ils sont mis à la valeur courante du système sur lequel T_EX s'exécute. Ils ne sont pas remis à jour pendant que le traitement par T_EX progresse.

En relation avec ces compteurs, la commande `\today` produit une chaîne représentant le jour courant (voir Section 23.8 [`\today`], page 163).

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 0,95cm. 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 300pt 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 contractabilité 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}
```

$\text{T}_{\text{E}}\text{X}$ en réalité 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 19.2 [`\hfill`], page 132, et voir Section 19.11 [`\vfill`], page 135).

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 en unités métriques <code>units</code> , avec deux chiffres après la virgule, est 1 point = 2,85 mm = 28,45 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>cm</code>	Centimetre
<code>mm</code>	Millimètre

<code>dd</code>	Point Didot, 1,07 pt
<code>cc</code>	Cicero, 12 dd
<code>sp</code>	Point proportionné, (1/65536) pt

Deux autres longueurs sont souvent utilisées et sont des valeurs réglées par les créateurs de caractères typographiques. 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. De même le `em`, traditionnellement la largeur de la lettre `M` capitale, est souvent utilisée pour les longueurs horizontales (il y a aussi `\enspace` qui vaut `0.5em`). 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.

En mode mathématique, beaucoup de définitions sont exprimées dans l'unité mathématique `mu` donnée par `1em = 18mu`, où le `em` est pris de la famille courante des symboles mathématiques. Voir Section 16.5 [Spacing in math mode], page 126.

14.2 `\setlength`

Synopsis :

```
\setlength{\longueur}{valeur}
```

La commande `\setlength` règle la valeur d'une *commande de longueur* `\longueur` à l'argument `valeur` qui peut être exprimé dans n'importe quelle unité comprise par L^AT_EX, c.-à-d. des pouces (`in`), des millimètres (`mm`), des points (`pt`), des « big points » (`bp`), etc.

14.3 `\addtolength`

Synopsis :

```
\addtolength{\longueur}{valeur}
```

La commande `\addtolength` incrémente une commande de longueur `\longueur` de la quantité spécifiée par l'argument *quantité*, ce qui peut être négatif.

14.4 `\settodepth`

Synopsis :

```
\settodepth{\longueur}{texte}
```

La commande `\settodepth` règle la valeur d'une commande de longueur `\longueur` à la profondeur de l'argument `texte`.

14.5 `\settoheight`

Synopsis :

```
\settoheight{\longueur}{texte}
```

La commande `\settoheight` règle la valeur d'une commande de longueur `\longueur` à la hauteur de l'argument `texte`.

14.6 `\settoheight`

Synopsis :

```
\settoheight{\longueur}{texte}
```

La commande `\settoheight` règle la valeur d'une commande le longueur `\longueur` à la largeur de l'argument `texte`.

14.7 Longueurs prédéfinies

```
\width
```

```
\height
```

```
\depth
```

```
\totalheight
```

Ces paramètres de longueur peuvent être utilisés au sein des arguments des commandes de fabrication de boîte (voir Chapitre 20 [Boxes], page 137). Ils spécifient la largeur naturelle, etc., du texte dans la boîte. `\totalheight` vaut `\height + \depth`. Pour fabriquer une boîte dont le texte est dilaté au double de sa taille naturelle, écrivez par ex. :

```
\makebox[2\width]{Dilatez moi}
```

15 Faire des paragraphes

Un paragraphe se termine par une ou plusieurs lignes complètement blanches — des lignes ne contenant même pas un %. Une ligne à blanc ne devrait pas apparaître là où un nouveau paragraphe n'a pas le droit de commencer, tel que en mode mathématique ou dans l'argument d'une commande de rubricage.

15.1 `\indent`

`\indent` produit un espace horizontal dont la largeur est égale à la longueur `\parindent`, le *renforcement* normal d'un paragraphe. Elle est utilisée pour ajouter un renforcement de paragraphe là où il serait autrement supprimé.

La valeur par défaut de `\parindent` est `1em` en mode `two-column`, autrement elle vaut `15pt` pour les documents `10pt`, `17pt` pour `11pt`, et `1.5em` pour `12pt`.

15.2 `\noindent`

Utilisée au commencement d'un paragraphe, la commande `\noindent` supprime tout renforcement de paragraphe.

... fin du paragraphe précédent.

`\noindent` Ce paragraphe n'est pas renforcé.

Elle est sans effet quand elle est utilisée au milieu d'un paragraphe.

Pour éliminer le renforcement de paragraphe dans tout un document, mettez `\setlength{\parindent}{0pt}` dans le préambule.

15.3 `\parindent` & `\parskip`

`\parskip` est une longueur élastique définissant l'espace vertical additionnel ajouté avant chaque paragraphe. La valeur par défaut est `0pt plus1pt`.

15.4 Notes en marge

Synopsis :

`\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 nearest 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 formattage 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 à 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

Il y a trois environnements qui mettent L^AT_EX en mode mathématique :

`math` Pour les formules qui apparaissent directement dans le texte.

`displaymath`

Pour les formules qui apparaissent sur leur propre ligne.

`equation` Parait que l'environnement `displaymath` sauf qu'il ajoute un numéro d'équation dans la marge de droite.

L'environnement `math` peut être utilisé aussi bien en mode paragraphe qu'en mode LR, mais les environnements `displaymath` et `equation` ne peuvent être utilisés qu'en mode paragraphe. Les environnements `math` et `displaymath` sont utilisés si souvent qu'ils ont les formes abrégées suivantes :

`\(...\)` au lieu de `\begin{math}...\end{math}`

`[\(...\)]` au lieu de `\begin{displaymath}...\end{displaymath}`

En fait, l'environnement `math` est si fréquent qu'il a une forme encore plus courte :

`$... $` au lieu de `\(...\)`

La commande `\boldmath` change les lettres and symboles mathématiques pour qu'ils soient en gras. Elle est utilisée *en dehors* du mode mathématique. À l'inverse, la commande `\unboldmath` change les glyphes mathématique pour qu'ils soient dans une fonte normale ; elle aussi est à utiliser *en dehors* du mode mathématique.

La déclaration `\displaystyle` force la taille et le style de la formule à être ceux de `displaymath`, par ex., avec les limites au dessus et en dessous le signe de sommation. Par exemple

`\displaystyle \sum_{n=0}^{\infty} x_n`

16.1 Indices inférieurs & supérieurs

En mode mathématique, utilisez le caractère caret `^` pour qu'une expression `exp` apparaisse en indice supérieur, c.-à-d. tapez `^{exp}`. Similairement, en mode mathématique, `_ {exp}` fait un indice inférieur de `exp`.

Dans cet exemple le 0 et le 1 apparaissent comme indices inférieurs alors que le 2 est un indice supérieur.

`\((x_0+x_1)^2 \)`

Pour avoir plus d'un caractère au sein de `exp` utilisez des accolades comme dans `e^{-2x}`.

L^AT_EX manipule la mise en indice supérieur d'un matériel et tout ce qui tient de cela de la manière naturelle, de sorte que des expressions telles que `e^{x^2}` et `x_{a_0}` seront composées comme il faut. Il fait même ce qu'il faut quand quelque-chose a à la fois un indice supérieur et un indice inférieur. Dans cet exemple le 0 apparaît au bas du signe intégrale alors que le 10 apparaît en haut.

`\int_0^{10} x^2 \, dx`

Vous pouvez mettre un indice supérieur ou inférieur avant un symbole avec une construction telle que `{_t K^2` en mode mathématique (le `{}` initial empêche l'indice préfixé d'être attaché à quelque symbole qui le précède dans l'expression).

En dehors du mode mathématique, une construction comme `Un test$_\textnormal{indice inf.}` produit un indice inférieur composé en mode texte, non pas en mode mathématique. Notez qu'il y a des paquetage spécialisée pour écrire des formule chimique, tel que `mhchem`.

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 .

Ci-dessous se trouve une liste de symboles généralement disponibles. 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 129.

<code>\l</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 : sign inférieur-à <code><</code> 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 quelque-fois utilisée comme une opération binaire générale, and quelques-fois réservée pour la corrélation-croisée.
<code>\asymp</code>	\asymp Équivalent asymptotiquement (relation).
<code>\backslash</code>	\backslash contr'oblique (ordinaire). Similaire : soustraction d'ensemble <code>\setminus</code> , et <code>\textbackslash</code> pour une contr'oblique en-dehors du mode mathématique.
<code>\beta</code>	β Lettre grecque en bas de casse beta (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>	\otimes 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, et-logique (opérateur).
<code>\bigwedge</code>	\wedge De taille variable, ou n-aire, ou-logique (opérateur).
<code>\bot</code>	<code>\bot</code> 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>	<code>\Box</code> Opérateur modal pour la nécessité ; boîte ouverte carrée (ordinaire). 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>\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>	<code>\complement</code> Complement d'ensemble, utilisée en indice supérieur comme dans <code>\$S^{\complement}</code> (ordinaire). 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> . On utilise aussi : <code>\$S^{\mathsf{c}}</code> ou <code>\$\bar{S}</code> .
<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>	\dashv Taquet gauche, tiret avec barre verticale à droite, tourniquet pour gauchers (relation). Similaire : taquet droit, tourniquet <code>\vdash</code> .
<code>\ddagger</code>	\ddagger 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>	\Diamond 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>	\diamondsuit 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). Sembable à la lettre en texte grec. En mathématiques on utilise plus fréquemment l'épsilon minuscule de ronde <code>\varepsilon</code> ε . 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>	\ge 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>	\heartsuit Cœur de carte jouer (ordinaire).
<code>\hookleftarrow</code>	\hookleftarrow Flèche vers la gauche avec crochet (relation).
<code>\hookrightarrow</code>	\hookrightarrow Flèche vers la gauche avec crochet (relation).
<code>\iff</code>	\iff Si et seulement si (relation). C'est un <code>\Longleftarrow</code> avec un <code>\thickmuskip</code> de chaque côté.
<code>\Im</code>	\Im Partie imaginaire (ordinaire). Voir aussi : partie réelle <code>\Re</code> .
<code>\in</code>	\in Appartient à (relation). Voir aussi : lettre grecque bas de casse epsilon lunaire <code>\epsilon</code> et l'épsilon minuscule de ronde <code>\varepsilon</code> .
<code>\infty</code>	∞ Infini (ordinaire).
<code>\int</code>	\int 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>\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>	\wedge Et logique (binaire). Synonyme : <code>\wedge</code> . Voir aussi ou-logique <code>\lor</code> .
<code>\langle</code>	\langle 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>	\lceil Plafond à gauche, ressemble à un crochet mais avec le pied coupé (ouvrant). Correspond à <code>\rceil</code> .
<code>\le</code>	\leq 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>	\Leftarrow 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 `\Leftrightarrow`.
- `\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 bilatérale `\leftrightharpoonup`.
- `\leftrightharpoonup`
 \leftrightharpoonup Flèche bilatérale (relation). Similaire : double flèche bilatérale `\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). Ceci n'est pas disponible en `TEX` de base. En `LATEX` vous devez charger le paquetage `amssymb`. For the normal subgroup symbol you should load `amssymb` and use `\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 : `\wedge`.
- `\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 Vrai, ou satisfait ; double turnstile, short double dash (relation). Similaire : long double dash <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 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 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>\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 $>$. 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 sousgroupe 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 contr'oblique, comme <code>\</code> (binaire). Similaire : contr'oblique <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> Similaire ou égal à, dans une relation (relation).
<code>\smallint</code>	<code>\int</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 à (binaire). Similaire : subset or égal à `\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 à (binaire). Similaire : sur-ensemble ou égal `\supseteq`.
- `\star` ★ Étoile à cinq branches, quelque-fois utilisée comme une opération binaire générale mais quelque-fois réservée pour la corrélation croisée (binaire). Similaire : les synonymes astérisque `*` anet `\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 $>$.
- `\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` ↙ 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 poset (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 en 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 en 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` Υ 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 : `\renewcommand`. 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`.
Pour une norme vous pouvez utiliser le paquetage `mathtools` et ajouter `\DeclarePairedDelimiter\norm{\lVert}{\rVert}` à 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 pabarres 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).

16.3 Fonctions mathématiques

En mode mathématique ces commandes produisent des noms de fonction en police romaine et avec l'espacement adéquat.

<code>\arccos</code>	<code>arccos</code>
<code>\arcsin</code>	<code>arcsin</code>
<code>\arctan</code>	<code>arctan</code>
<code>\arg</code>	<code>arg</code>
<code>\bmod</code>	Opérateur modulo binaire ($x \bmod y$)
<code>\cos</code>	<code>cos</code>
<code>\cosh</code>	<code>cosh</code>
<code>\cot</code>	<code>cos</code>
<code>\coth</code>	<code>cosh</code>
<code>\csc</code>	<code>csc</code>
<code>\deg</code>	<code>deg</code>
<code>\det</code>	<code>deg</code>
<code>\dim</code>	<code>dim</code>

<code>\exp</code>	exp
<code>\gcd</code>	gcd
<code>\hom</code>	hom
<code>\inf</code>	inf
<code>\ker</code>	ker
<code>\lg</code>	lg
<code>\lim</code>	lim
<code>\liminf</code>	lim inf
<code>\limsup</code>	lim sup
<code>\ln</code>	ln
<code>\log</code>	log
<code>\max</code>	max
<code>\min</code>	min
<code>\pmod</code>	Modulo parenthésé, comme dans $(\pmod 2)^n - 1$
<code>\Pr</code>	Pr
<code>\sec</code>	sec
<code>\sin</code>	sin
<code>\sinh</code>	sinh
<code>\sup</code>	sup
<code>\tan</code>	tan
<code>\tanh</code>	tanh

16.4 Accents mathématiques

\LaTeX 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 160).

<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>\imath</code>	I sans point mathématique.
<code>\jmath</code>	J sans point mathématique.
<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 : $\widetilde{x+y}$.

16.5 Espacement en mode mathématique

Au sein d'un environnement `math`, L^AT_EX ignore les espaces que vous tapez et met à la place l'espacement correspondant aux règles normales pour les textes mathématiques.

En mode mathématique, beaucoup de définitions 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 107).

<code>\;</code>	Normalement 5.0mu plus 5.0mu . Le nom long est <code>\thickspace</code> . Utilisable seulement en mode mathématique.
<code>\:</code>	
<code>\></code>	Normalement 4.0mu plus 2.0mu minus 4.0mu . The longer name is <code>\medspace</code> . Utilisable seulement en mode mathématique.
<code>\,</code>	Normalement 3mu . Le nom long est <code>\thinspace</code> . Utilisable à la fois en mode mathématique et en mode texte. Voir Section 19.6 [<code>\thinspace</code>], page 133.
<code>\!</code>	Une espace finie négative. Normalement -3mu . Utilisable seulement en mode mathématique.
<code>\quad</code>	Vaut 18mu , c.-à-d. 1 em . Souvent utilisé pour l'espace entourant les équation ou expression, par ex. pour l'espace entre deux équation au sein d'un environnement <code>displaymath</code> . Utilisable à la fois en mode mathématique et en mode texte.
<code>\qqquad</code>	Une longueur de 2 quads, soit $36\text{mu} = 2\text{ em}$. Utilisable à la fois en mode mathématique et en mode texte.

Dans cet exemple une espace fine sépare la fonction de la forme différentielle.

```
\int_0^1 f(x)\,dx
```

16.6 Recueil de diverses commandes utiles en math

<code>*</code>	Un symbole de multiplication <i>discrétionnaire</i> , 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 \times 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.

`\cdots` Des points de suspension élevés au centre de la ligne. Comme dans : ‘...’.

`\ddots` Des points de suspension en diagonale : ‘⋮’.

`\frac{num}{dén}`

Produit la fraction *num* divisé par *dén*. Par ex. $\frac{1}{4}$.

`\left delim1 ... \right delim2`

Les deux délimiteurs ne se correspondent par nécessairement ; ‘.’ tient lieu de *délimiteur nul*, ne produisant rien en sortie. Les délimiteurs sont dimensionnés selon la portion de formule mathématique qu’ils embrassent. Exemple : `\left(\sum_{i=1}^{10} a_i \right)`.

`\mathdollar`

Signe dollar en mode mathématique : \$.

`\mathellipsis`

Points de suspension (épacés pour du texte) en mode mathématique : ...

`\mathparagraph`

Signe paragraphe (pied-de-mouche) en mode mathématique : ¶.

`\mathsection`

Signe section en mode mathématique.

`\mathsterling`

Signe livre sterling mode mathématique : £.

`\mathunderscore`

Signe « souligné » en mode mathématique : ..

`\overbrace{math}`

Génère une accolade au dessus de *math*. Par exemple, `\overbrace{x+\cdots+x}^k`

`;``\textrm{fois}`. Le résultat ressemble à cela : $\overbrace{x + \cdots + x}^{k \text{ fois}}$

`\overline{texte}`

Génère une ligne horizontale au dessus de *texte*. Par exemple, `\overline{x+y}`.

Le résultat ressemble à cela : $\overline{x + y}$.

`\sqrt[nième]{arg}`

Produit la représentation de la racine carrée de *arg*. L’argument optionnel *nième* détermine quelle racine produire. Par exemple, la racine cubique de $x+y$ serait tapée comme `\sqrt[3]{x+y}`. Le résultat ressemble à cela : $\sqrt[3]{x + y}$.

`\stackrel{texte}{relation}`

Place *texte* au dessus de *relation*. Par exemple, `\stackrel{f}{\longrightarrow}`. ■

Le resultat ressemble à cela : \xrightarrow{f} .

`\underbrace{math}`

Génère *math* avec une accolade en-dessous. Par exemple, `\underbrace{x+y+z}_{>0}`.

Le résultat ressemble à cela : $\underbrace{x + y + z}_{>0}$.

`\underline{texte}`

A pour effet que *texte*, ce qui peut être ou non en mode mathématique, soit souligné. La ligne est toujours sous le texte, en prenant en compte les jambages.

Le résultat ressemble à cela : \underline{xyz}

`\vdots`

Produit des points de suspension alignés verticalement. Le résultat ressemble à

cela : \vdots .

17 Modes

Quand \LaTeX traite votre texte en entrée, il est forcément dans l'un de ces trois modes :

- Mode paragraphe
- Mode math
- Mode gauche-à-droite, appelé également mode LR (left-to-right) pour faire bref

\LaTeX change le mode seulement lorsqu'il monte ou descend un escalier le menant à un niveau différent, quoique tous les changements de niveau ne produisent pas de changement de mode. Les changements de mode se produisent seulement lorsqu'on entre dans ou quitte un environnement, ou quand \LaTeX traite l'argument de certaines commandes qui produisent du texte.

Le « mode paragraphe » est le plus usuel ; c'est celui dans lequel \LaTeX se trouve lorsqu'il traite du texte ordinaire. Dans ce mode-là, \LaTeX fragmente votre texte en lignes et fragmente les lignes en pages. \LaTeX est en « mode math » quand il est en train de générer une formule mathématique. En « mode LR », comme en mode paragraphe, \LaTeX considère la sortie qu'il produit comme une chaîne de mots avec des espaces entre eux. Toutefois, contrairement au mode paragraphe, \LaTeX continue à progresser de la gauche vers la droite ; il ne démarre jamais une nouvelle ligne en mode LR. Même si vous mettez une centaine de mots dans une `\mbox`, \LaTeX continue à les composer de gauche à droite au sein d'une seule boîte, et ensuite se plaindra que la boîte résultante est trop large pour tenir dans la ligne.

\LaTeX est en mode LR quand il commence à fabriquer une boîte avec une commande `\mbox`. Vous pouvez le faire entrer dans un mode différent au sein de la boîte — par exemple, vous pouvez le faire entrer en mode math pour mettre une formule dans la boîte. Il y a aussi plusieurs commandes qui produisent du texte et un environnement pour fabriquer une boîte qui met \LaTeX en mode paragraphe. La boîte fabriquée par l'une de ces commandes ou environnements est appelée une `parbox`. Quand \LaTeX est en mode paragraphe alors qu'il fabrique une boîte, on dit qu'il est en « mode paragraphe interne ». Son mode paragraphe normal, celui dans lequel il démarre, est appelé « mode paragraphe externe ».

17.1 `\ensuremath`

Synopsis :

```
\ensuremath{formule}
```

La commande `\ensuremath` assure que *formule* est composée en mode mathématique quel que soit le mode courant où la commande apparaît.

Par exemple :

```
\documentclass{report}
\usepackage{french}
\newcommand{\ab}{\ensuremath{(\delta, \varepsilon)}}
\begin{document}
Le couple \ab vaut alors (\frac{1}{\pi}, 0), ...
\end{document}
```

Il est possible de redéfinir des commandes qui ne peuvent être utilisées qu'en mode mathématique pour qu'elles puissent être utilisées en tout mode comme dans l'exemple ci-dessous avec `\leadsto` :

```
\documentclass{report}
\usepackage{amssymb}
\usepackage{french}
\newcommand{\originalMeaningOfLeadsTo}{}
\let\originalMeaningOfLeadsTo\leadsto
\renewcommand\leadsto{\ensuremath{\originalMeaningOfLeadsTo}}
\begin{document}
Tous les chemins \leadsto\ Rome.
\end{document}
```

18 Styles de page

La commande `\documentclass` détermine la taille et la position de l'en-tête et du bas de page. Le style de page détermine leur contenu.

18.1 `\maketitle`

18.2 `\pagenumbering`

Synopsis :

```
\pagenumbering{style}
```

Spécifie le style des numéros de page numbers selon *style*. `\pagenumbering` remet également le numéro de page à 1. L'argument *style* peut prendre les valeurs suivantes :

<code>arabic</code>	nombres arabes
<code>roman</code>	Nombres romain en bas de casse
<code>Roman</code>	Nombres romain en capitales
<code>alph</code>	Lettres en bas de casse
<code>Alph</code>	Lettre en capitales

18.3 `\pagestyle`

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^AT_EX a beaucoup de façons de produire de l'espace blanc (ou rempli).

Une autre commande produisant de l'espace est `\`, pour produire une espace « fine » (d'habitude 1/6 quad). Elle peut être utilisée en mode texte, mais est le plus souvent utile en mode math (voir Section 16.5 [Spacing in math mode], page 126).

19.1 `\hspace`

Synopsis :

```
\hspace{longueur}
\hspace*{longueur}
```

La commande `\hspace` ajoute de l'espace horizontal. L'argument *longueur* peut être exprimé en toute unité que L^AT_EX comprend : points, pouces, etc. C'est une longueur élastique. Vous pouvez ajouter aussi bien de l'espace négatif que de l'espace positif avec une commande `\hspace` ; ajouter de l'espace négatif revient à retourner en arrière.

L^AT_EX enlève normalement l'espace horizontal qui vient au début ou à la fin d'une ligne. Pour préserver cet espace, utilisez la forme étoilée, c.-à-d. avec le suffixe `*` optionnel.

19.2 `\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 ‘`opt plusifill`’) ; ainsi, `\hspace\fill` équivaut à `\hfill`.

19.3 `\<espace>` et `\@`

Ces commandes permettent de marquer un caractère de ponctuation, typiquement un point, comme soit finissant une phrase, soit finissant une abréviation.

Par défaut, lorsque L^AT_EX justifie une ligne il ajuste l'espace après le point final de chaque phrase (ou le point d'interrogation, la virgule, ou le deux-points) plus que pour l'espace inter-mot (voir Section 19.5 [`\frenchspacing`], page 133). L^AT_EX suppose que le point finit une phrase à moins qu'il soit précédé d'une lettre capitale, dans quel cas il considère que ce point fait partie d'une abréviation. Il est à noter que si un point final de phrase est immédiatement suivi d'une parenthèse ou crochet de droite, ou d'un guillemet-apostrophe simple ou double de droite, alors l'espace inter-phrase suit cette parenthèse ou ce guillemet.

Si vous avez un point finissant une abréviation dont la dernière lettre n'est pas une capitale, et que cette abréviation n'est pas le dernier mot d'une phrase, alors faites suivre ce point d'un contr'oblique-espace (`\`), où d'un tilde (`~`). Voici des exemples : `Nat.\ Acad.\ Science`, et `Mr.~Bean`, et `(du fumier, etc.)\ à vendre`.

Pour une autre utilisation de `\` , voir Section 19.4 [`\(SPACE)` after control sequence], page 133.

Dans la situation opposée, si vous avez une lettre capitale suivi d'un point qui finit la phrase, alors placez `\@` avant ce point. Par ex., `book by the MAA\@.` aura l'espacement inter-phrase après ce point.

Par contre, placer `\@` après un point dit à \TeX que ce point ne finit pas la phrase. Dans l'exemple les mot réservés (`if`, `then`, `etc.\@`) sont différents, \TeX met un espace inter-mot après la parenthèse fermante (notez que `\@` est avant la parenthèse).

19.4 `\` après une séquence de contrôle

La commande `\` est aussi souvent utilisée après les séquences de contrôle pour les empêcher d'avaler les espaces qui suivent, comme dans `'\TeX\ c'est chouette.'` Et, dans des circonstances normales, `\tabulation` et `\sautdeligne` sont équivalents à `\`. Pour une autre utilisation de `\`, voir aussi Section 19.3 [`\(SPACE)` and `\@`], page 132.

Certaines personnes préfèrent utiliser `{}` dans le même but, comme dans `\TeX{} c'est chouette`. Ceci a l'avantage que vous écrivez toujours la même chose, en l'occurrence `\TeX{}`, qu'elle soit suivie d'un espace ou d'un signe de ponctuation. Comparez :

```
\TeX\ est un chouette système. \TeX, un chouette système.
```

```
\TeX{} est un chouette système. \TeX{}, un chouette système.
```

Certaines commandes, notamment celles définies avec le paquetage `xspace`, ne suivent pas ce comportement standard.

19.5 `\frenchspacing`, rendre égaux les espacements inter-phrase et inter-mot

Cette déclaration (provenant de \TeX de base) cause que \LaTeX traite l'espacement inter-phrase de la même façon que l'espacement inter-mot.

En justifiant le texte sur une ligne, certaine traditions typographique, y compris l'anglaise, préfèrent ajuster l'espace entre les phrase (ou après d'autres signes de ponctuation) plus que l'espace entre les mots. À la suite de cette déclaration, tous les espaces sont traités de la même façon.

Revenez au comportement par défaut en déclarant `\nonfrenchspacing`.

19.6 `\thinspace`

`\thinspace` produit une espace insécable et non-élastique qui vaut 1/6 d'em. C'est l'espace adéquate entre des guillemets anglais simples emboîtés, comme dans `'''`.

19.7 `\/` : insérer une correction italique

La commande `\/` produit une *correction italique*. C'est une espace fine défini par le créateur de caractères typographiques pour un caractère donné, pour éviter que le caractère entre en collision avec ce qui suit. Le caractère italique *f* a typiquement une valeur importante de correction italique.

Si le caractère suivant est un point ou une virgule, il n'est pas nécessaire d'insérer une correction italique, puisque ces symboles de ponctuation ont une très faible hauteur. Toutefois, avec les point-virgules ou les deux-points, de même qu'avec les lettres normales, il peut être utile. Comparez : `f: f;` avec `f: f;`.

Lorsque on change de fontes avec des commandes telles que `\textit{italic text}` ou `{\itshape italic text}`, L^AT_EX insère automatiquement une correction italique appropriée si nécessaire (voir Section 4.1 [Font styles], page 17).

Malgré leur nom, les caractères romains peuvent aussi avoir une correction italique. Comparez : pdfT_EX avec pdfT_EX.

Le concept de correction italique n'existe pas en mode mathématique ; l'espacement y est fait d'une façon différente.

19.8 `\hrulefill` `\dotfill`

Produit une longueur élastique infinie (voir Chapitre 14 [Lengths], page 106) remplie avec un filled filet horizontal (c.-à-d. une ligne) ou avec des pointillés, au lieu de just de l'espace blanche.

Quand on le place entre deux lignes à blanc cet exemple crée un paragraphe qui est justifié à gauche et à droite, où l'espace au milieu est rempli avec des pointillés uniformément espacés.

```
\noindent Jack Aubrey\dotfill Melbury Lodge
```

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 avec `\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.

19.9 `\addvspace`

```
\addvspace{length}
```

Ajoute un espace vertical de longueur *length*, ce qui est une longueur élastique (voir Chapitre 14 [Lengths], page 106). Toutefois, si l'espace vertical a déjà été ajouté au même point de la sortie par une commande `\addvspace` précédente, alors cette commande n'ajoutera pas plus d'espace que nécessaire à fabriquer la longueur naturelle de l'espace vertical total égale à *length*.

Utilisez 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 est défini pour commencer et finir en `\addvspace{.}` de sorte que deux Theorem consécutifs sont séparés par un espace vertical, et non deux.)

Cette commande est fragile (voir Section 12.11 [`\protect`], page 101).

L'erreur 'Something's wrong--perhaps a missing `\item`' 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 cette commande d'une commande `\par`.

19.10 `\bigskip` `\medskip` `\smallskip`

Ces commandes produisent une quantité donnée d'espace, spécifiée par la classe de 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` et 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` et 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` et 3pt plus 1pt minus 1pt).

19.11 `\vfill`

Finist le paragraphe courant et insère une longueur élastique verticale (voir Chapitre 14 [Lengths], page 106) 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 19.12 [`\vspace`], page 135.

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}
```

19.12 `\vspace`

Synopsis, l'une des deux formes suivantes :

```
\vspace{longueur}
\vspace*{longueur}
```

Ajout l'espace verticale *longueur*. Celle-ci peut-être négative ou positive, et c'est une longueur élastique (voir Chapitre 14 [Lengths], page 106).

\LaTeX enlève l'espace verticale provenant de `\vspace` aux sauts de page, c.-à-d. au sommet et au bas des pages. La version étoilée `\vspace*{...}` permet de conserver cette espace.

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}
```

20 Boîtes

Tous les paramètres de longueurs prédéfinies (voir Section 14.7 [Predefined lengths], page 109) peuvent être utilisés au sein des arguments des commandes de fabrication de boîtes.

20.1 `\mbox{texte}`

La commande `\mbox` crée un boîte qui a la largeur juste du texte créé par son argument. Le *texte* n'est pas coupé en plusieurs lignes, c'est pourquoi cette commande peut être utilisée pour empêcher la césure.

20.2 `\fbox` and `\framebox`

Synopsis :

```
\fbox{texte}
\framebox[largeur][position]{texte}
```

Les commandes `\fbox` et `\framebox` sont similaires à `\mbox`, à ceci près qu'elle mettent un cadre autour de l'extérieur de la boîte ainsi créée.

De plus, la commande `\framebox` permet de spécifier explicitement la largeur de la boîte avec l'argument optionnel *largeur* (une longueur), et sa position sur la ligne avec l'argument optionnel *position*.

Les deux commandes produisent un filet d'épaisseur `\fboxrule` (valant par défaut '`.4pt`'), et laisse un espace de `\fboxsep` (valant par défaut '`3pt`') entre le filet et le contenu de la boîte.

Voir Section 8.19.3 [`\framebox` (picture)], page 67, pour la commande `\framebox` dans l'environnement `picture`.

20.3 `lrbox`

Synopsis :

```
\begin{lrbox}{\cmd}
  texte
\end{lrbox}
```

Ceci est la forme environnement de la commande Section 20.8 [`\sbox`], page 139.

Le *texte* au sein de l'environnement est sauvegarder dans une boîte `\cmd` qui doit avoir préalablement été déclarée avec `\newsavebox`.

20.4 `\makebox`

Synopsis :

```
\makebox[largeur][position]{texte}
```

La commande `\makebox` crée une boîte juste assez large pour contenir le *texte* spécifié. La largeur de la boîte peut-être emportée par l'argument optionnel *largeur*. 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) justifié à gauche.
- r (Right) justifié à droite.
- s (Stretch) justifié à travers la *largeur* entière ; *texte* doit contenir de l'espace dilatable pour que cela marche.

`\makebox` est aussi utilisé au sein de l'environnement `picture`, voir Section 8.19.2 [`\makebox (picture)`], page 67.

20.5 `\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 101).

```
\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 taité en mode texte (voir Chapitre 17 [Modes], page 129) 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 63).

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 63, pour une description complète.

Il y a deux arguments obligatoires. La *largeur* est une longueur rigide (voir Chapitre 14 [Lengths], page 106). 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 s'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*. Les valeurs possibles sont : `t` 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).

20.6 `\raisebox`

Synopsis :

```
\raisebox{distance}[hauteur][profondeur]{texte}
```

La commande `\raisebox` élève ou abaisse du *texte*. Le premier argument obligatoire spécifie la longueur dont le *texte* doit être élevé (ou abaissé si c'est une longueur négative). Le *texte* lui-même est traité en mode LR.

Les arguments optionnels *hauteur* et *profondeur* sont des longueurs. S'ils sont spécifiés, L^AT_EX traite *texte* comme s'il s'étendait une certaine distance (*hauteur*) au-dessus de la ligne de base, ou au dessous (*profondeur*), en ignorant ses hauteur et profondeur naturelles.

20.7 `\savebox`

Synopsis :

```
\savebox{\boxcmd}[largeur][pos]{texte}
```

Cette commande compose *texte* dans une boîte de la même façon qu'avec `\makebox` (voir Section 20.4 [`\makebox`], page 137), à ceci près qu'au lieu d'imprimer la boîte résultante, elle la sauvegarde dans une boîte étiquetée `\boxcmd`, ce qui doit avoir été déclaré avec `\newsavebox` (voir Section 12.7 [`\newsavebox`], page 97).

20.8 `\sbox{\boxcmd}{texte}`

Synopsis :

```
\sbox{\boxcmd}{texte}
```

`\sbox` tape *texte* au sein d'une boîte tout comme `\mbox` (voir Section 20.1 [`\mbox`], page 137) à ceci près qu'au lieu que la boîte résultante soit incluse dans la sortie normale, elle est sauvegardée dans la boîte étiquetée `\boxcmd`. `\boxcmd` doit avoir été préalablement déclarée avec `\newsavebox` (voir Section 12.7 [`\newsavebox`], page 97).

20.9 `\usebox{\boxcmd}`

Synopsis :

```
\usebox{\boxcmd}
```

`\usebox` produit la boîte la plus récemment sauvegardée dans le bac `\boxcmd` par une commande `\savebox` (voir Section 20.7 [`\savebox`], page 139).

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^AT_EX 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^AT_EX 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 récents 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^AT_EX 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^AT_EX 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 pleines de cyan, magenta et jaune produit du noir.

cmk	Un liste séparée de virgule avec quatre nombres réels 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.
gray	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.
rgb	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, la deuxième correspond au vert, et la 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.
RGB	(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 donné qu'en dehors de L ^A T _E 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.
named	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 140).

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 140). 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 cinquante 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 \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` and `\framebox`], page 137), 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^AT_EX 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 142. 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 fichier 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 décrite 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 photograpgie en format JPG. Dans une graphique vectoriel, le fichier contient une liste d'instructions telles que 'dessin 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 vectoriel 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 petites.

Il y a des systèmes particulièrement bien adapté 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 65) 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 externe 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'informtion stockée dans un fichier *pilote d'impression*. Normalment 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 fichier graphique 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 185). 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 endroit 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 séparateur de répertoire puisque cela rend le document portable vers d'autres plateformes. Si vous avez des espace 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{{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'improte-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 L^AT_EX 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'extentions 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, L^AT_EX 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, L^AT_EX utilise la liste d'extension. Pour chacun des répertoires dans la liste de chemins graphiques (voir Section 22.2.1 [`\graphicspath`], page 146), L^AT_EX 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.

```
.png,.pdf,.jpg,.mps,.jpeg,.jbig2,.jb2,.PNG,.PDF,.JPG,.JPEG,.JBIG2,.JB2
```

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}
```

Declare comment gérer les fichiers graphiques dont le nom se termine en *extension*.

L'exemple suivant déclare que tous le fichiers dont ele est de la forme `nomfichier-sans-point.mps` doivent être traité 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 \LaTeX 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 \LaTeX 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 bitmap dans un fichier avec une ligne `%%BoundingBox` de style PostScript style 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 \TeX 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 \TeX .

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 146.

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 147).

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 51).

```

\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^AT_EX à dilater le texte ou à recourir à des zones blanche 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 154, et Section 22.3.3 [`\scalebox`], page 155). 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_EX (voir Section 14.1 [Units of length], page 107). 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 supposée 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 inch de largeur et 0,618 inches 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^AT_EX 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^AT_EX 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_EX (voir Section 14.1 [Units of length], page 107) et également il est commode d'utiliser `\linewidth`, ou, dans un document recto-verso, `\columnwidth` (voir Section 5.5 [Page layout parameters], page 24). Un exemple est qu'en utilisant le paquetage `calc` vous pouvez rendre le graphique 1cm 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_EX (voir Section 14.1 [Units of length], page 107), et également il est commode d'utiliser `\pageheight` et `\textheight` (voir Section 5.5 [Page layout parameters], page 24). 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^AT_EX rendra le graphique aussi grand que possible sans le déformer. C'est à dire que L^AT_EX 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 anti-horaire. La rotation est effectuée autour de l'origine donnée par l'option `origin` ; s'y référer. Pour une description complète de la composition de matériel ayant subi une rotation, voir Section 22.3.2 [`\rotatebox`], page 154.

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 154.

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 fichier PDF parmi les suivantes **mediabox**, **cropbox**, **bleedbox**, **trimbox**, ou **artbox**. Les fichier PDF n'ont pas la **BoundingBox** que les fichiers PostScript ont, mais peuvent spécifier jusqu'à quatre rectangle prédéfinis. La boîte **MediaBox** donne les frontière 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 au dimensions désirée de la page finie. La boîte **ArtBox** est l'étendu 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 **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 L^AT_EX 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 donant 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 T_EX, 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 box. Ces quatre options là sont obsolète, mais existent encore pour maintenir la rétrocompatibilité avec des paquets 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 L^AT_EX 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 T_EX, 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 L^AT_EX 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 fichier `lion.xxx` est lu comme s'il s'agissait de `lion.png`. Pour plus ample information, voir Section 22.2.3 [`\DeclareGraphicsRule`], page 147.

type Spécifie le type de graphique.

ext Spécifie l'extension du graphique. À n'utiliser qu'en conjonction avec l'option `type`.

units This key allows you to change the default of degrees counterclockwise. Setting `units=-360` changes the direction to degrees clockwise and setting `units=6.283185` changes to radians counterclockwise.

22.3.3 `\scalebox`

Synopses:

```
\scalebox{horizontal factor}{material}
\scalebox{horizontal factor}[vertical factor]{material}
\reflectbox{material}
```

Scale the *material*.

This example halves the size, both horizontally and vertically, of the first text and doubles the size of the second.

```
\scalebox{0.5}{DRINK ME} and \scalebox{2.0}{Eat Me}
```

If you do not specify the optional *vertical factor* then it defaults to the same value as the *horizontal factor*.

You can use this command to resize a graphic, as with `\scalebox{0.5}{\includegraphics{lion}}`. If you use the `graphicx` package then you can accomplish the same thing with optional arguments to `\includegraphics` (voir Section 22.3.1 [`\includegraphics`], page 148).

The `\reflectbox` command abbreviates `\scalebox{-1}[1]{material}`. Thus, `Able was I\reflectbox{Able was I}` will show the phrase ‘Able was I’ immediately followed by its mirror reflection.

22.3.4 `\resizebox`

Synopses:

```
\resizebox{horizontal length}{vertical length}{material}
\resizebox*{horizontal length}{vertical length}{material}
```

Given a size, such as 3cm, transform *material* to make it that size. If either *horizontal length* or *vertical length* is an exclamation point ! then the other argument is used to determine a scale factor for both directions.

This example makes the graphic be a half inch wide and scales it vertically by the same factor to keep it from being distorted.

```
\resizebox{0.5in}{!}{\includegraphics{lion}}
```

The unstarred form `\resizebox` takes *vertical length* to be the box’s height while the starred form `\resizebox*` takes it to be height+depth. For instance, make the text have a height+depth of a quarter inch with `\resizebox*{!}{0.25in}{\parbox{1in}{This box has both height and depth.}}`.

You can use `\depth`, `\height`, `\totalheight`, and `\width` to refer to the original size of the box. Thus, make the text two inches wide but keep the original height with `\resizebox{2in}{\height}{Two inches}`.

23 Insertions spéciales

L^AT_EX 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^AT_EX 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 contr'oblique \. Ainsi `\$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 `\^{}`. Une contr'oblique dans la police du corps de texte se fait avec `\textbackslash{}`.

Pour produire les caractères réservés dans la police tapuscrite utilisez `\verb!!` comme ci-dessous (la double contr'oblique `\\` n'est là que pour aller à la ligne en sortie) :

```
\begin{center}
  \# \$ \% \& \{ \} \_ \~{ } \^{ } \textbackslash \\
  \verb!# $ % & { } _ ~ ^ \!
\end{center}
```

Dans cet exemple la double contr'oblique `\\` n'est là que pour sauter à la ligne.

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_EX `\uppercase` et `\lowercase` ne fonctionnent que pour les 26 lettres sans diacritiques a-z et A-Z. Les commandes L^AT_EX `\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 101).

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}`.

L^AT_EX 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 T_EX 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 L^AT_EX.

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 T_EX et L^AT_EX d’origine. Cela a été implémenté dans le paquetage `expl3` à compter de 2020. L’article « Case changing: From T_EX 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, 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’exemple précédent pourrait tout aussi bien être écrit comme `\symbol{’40}` ou `\symbol{"20}`.

23.4 Symboles en texte

L^AT_EX fournit des commandes pour générer divers symboles qui ne sont pas des lettres dans le cours sein du texte. Certaines d’entre elles, en particulier les plus obscures, ne sont pas disponible en OT1 ; il se peut que vous deviez charger le paquetage `textcomp`.

<code>\copyright</code>	
<code>\textcopyright</code>	Le symbole « droit d’auteur », ©.
<code>\dag</code>	Le symbole obèle (dans le texte).
<code>\ddag</code>	Le symbole double obèle (dans le texte).
<code>\LaTeX</code>	Le logo L ^A T _E X.
<code>\LaTeXe</code>	Le logo L ^A T _E X2e.

`\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` (,)
 Guillemets-virgule inférieur double et simple : „, et ,.

`\rq` Guillemet-apostrophe simple de droite (fermant) : ’.

`\S` `\itemx \textsection` Signe section : §.

`\TeX` Le logo T_EX.

`\textasciicircum`
 circonflexe ASCII : ^.

`\textasciitilde`
 tilde ASCII : ~.

`\textasteriskcentered`
 Astérisque centré : *.

`\textbackslash`
 Contr’oblique : \.

`\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 : — (pour la ponctuation).

`\textendash` (ou `--`)
 Demi cadratin : – (pour les plages en anglais).

`\texteuro`
 Le symbole Euro : €.

`\textexclamdown` (ou `!'`)
 Point d'exclamation culbuté : ¡.

`\textgreater`
 Supérieur à : >.

`\textless`
 Inférieur à : <.

`\textleftarrow`
 Flèche gauche.

`\textordfeminine`
`\textordmasculine`
 Symboles ordinaux féminin et masculin : ^a, ^o.

`\textperiodcentered`
 Point centré : ·.

`\textquestiondown` (ou `?'`)
 Point d'interrogation culbuté : ¿.

`\textquotedblleft` (ou ‘‘)
Guillemet-apostrophe double culbuté : “.

`\textquotedblright` (ou ’’)
Guillemet-apostrophe de droite double : ”.

`\textquoteleft` (ou ‘)
Guillemet-apostrophe simple culbuté : ‘.

`\textquoteright` (ou ’)
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^AT_EX 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^AT_EX 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 disponible 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 70).
- `\.` Produit un point suscrit à la lettre qui suit, comme dans ô.
- `\=`
`\capitalmacron`
Produit un macron (une barre) suscrit à la lettre qui suit, comme dans õ.
- `\^`
`\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 70).
- `\~`
`\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 ȝ. Voir aussi `\underbar` ci-après.
- `\c`
`\capitalcedilla`
Produit une cedille souscrite à la lettre qui suit, comme dans ç.
- `\d`
`\capitaldotaccent`
Produit un point souscrit à la lettre qui suit, comme dans ȝ.
- `\H`
`\capitalhungarumlaut`
Produit un long tréma hongrois suscrit à la lettre qui suit, comme dans ő.
- `\i` Produit un i sans point, comme dans ‘ı’.
- `\j` Produit un j sans point, comme dans ‘j’.
- `\k`
`\capitalogonek`
Produit un ogonek, comme dans ‘ǫ’. Non disponible dans le codage OT1.
- `\r`
`\capitalring`
Produit un rond en chef, comme dans ‘ŀ’.

`\t`
`\capitaltie`
`\newtie`
`\capitalnewtie`
 Produit un tirant suscrit (double brève renversée), comme dans ‘ôô’. La forme `\newtie` est centrée dans sa boîte.

`\u`
`\capitalbreve`
 Produit un accent brève, comme dans ‘ö’.

`\underbar`
 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 `\underline` dans Section 16.6 [Math miscellany], page 126. Voir aussi `\b` ci-avant.

`\v`
`\capitalcaron`
 Produit un accent háček (caron), comme dans ‘ö’.

23.6 Lettres latines supplémentaires

Voici les commandes de base de \LaTeX pour insérer des caractères utilisés généralement utilisés dans des langages autres que l’anglais.

`\aa`
`\AA` å et Å.

`\ae`
`\AE` æ et Æ.

`\dh`
`\DH` Lettre islandaise eth : ð et Ð. Non disponible dans le codage OT1, il vous faut le paquetage `fontenc` pour sélectionner un autre codage de police, tel que T1.

`\dj`
`\DJ` 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 `fontenc` pour sélectionner un autre codage de police, tel que T1.

`\ij`
`\IJ` ij et IJ (à ceci près que les deux lettre apparaissent plus liées qu’ici).

`\l`
`\L` ł et Ł.

`\ng`
`\NG` Lettre eng laponais, utilisé aussi en phonétique. Non disponible dans le codage OT1, il vous faut le paquetage `fontenc` 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 `\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).

23.8 `\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 105) 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

L^AT_EX 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 L^AT_EX 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 le matériel 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 L^AT_EX traite seulement le matériel du premier chapitre. Ainsi, la sortie apparaît plus rapidement et est plus courte à imprimer. Cependant, l'avantage de la commande `\includeonly` est que L^AT_EX 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 189, pour un autre exemple de `\includeonly`.

24.1 `\endinput`

Synopsis :

```
\endinput
```

Quand vous écrivez `\include{nomfichier}`, alors au sein de `nomfichier.tex` le matériel qui vient après `\endinput` n'est pas inclus. Cette commande est optionnelle ; si `nomfichier.tex` ne comprend pas de `\endinput` alors L^AT_EX 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} Ce matériel 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 circonscrite 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 du matériel 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 88), puis elle insère le matériel 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 `preamble.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 le matériel issu de ces trois fichiers dans le document mais génère également les fichiers auxiliaires `preamble.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 42). Si maintenant vous mettez en commentaire les lignes de `\includeonly` contenant `preamble` et `amendements` et exécutez \LaTeX de nouveau alors le document résultant ne contiendra que le matériel issu de `articles.tex`, mais pas celui de `preamble.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 167, (ce qui, cependant, ne retient pas l'information auxiliaire).

Voir Section A.4 [Larger book template], page 189, pour un autre exemple utilisant `\include` et `\includeonly`. Cet exemple utilise également `\input` pour certain matériel 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^AT_EX 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^AT_EX.

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/post-liminaires

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`. \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 29, pour les numéros de niveaux des rubriques. Pour plus ample information sur `tocdepth` voir [Sectioning/tocdepth], page 30.

Un autre exemple de l'utilisation de `\tableofcontents` est dans Section A.4 [Larger book template], page 189.

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, \LaTeX stocke l'information dans un fichier auxiliaire nommé *fichier-racine.toc* (voir Chapitre 24 [Splitting the input], page 164). Par exemple, le fichier \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 `\addcontentsline`

Synopsis :

```
\addcontentsline{ext}{unit}{texte}
```

La commande `\addcontentsline` ajoute une entrée a la liste ou table spécifiée où :

ext L'extension de nom de fichier du fichier dans lequel l'information doit être écrite, typiquement c'est l'une des trois suivante :: `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 »).

unit Le nom de l'unité sectionnelle à ajouter, typiquement l'une des suivantes, selon la valeur de l'argument *ext* :

`toc` Les unités sectionnelles : `part`, `chapter`, `section`, `subsection`, `subsubsection`.

`lof` Pour la liste des figures : `figure`.

`lot` Pour la liste des tableaux : `table`.

`texte` Le texte de l'entrée.

Ce qui est écrit dans le fichier `.ext` est la commande `\contentsline{unit}{texte}{numéro}`, où `numéro` est la valeur courante du compteur `unit`.

25.1.2 `\addtocontents`

La commande `\addtocontents{ext}{texte}` ajoute du texte (ou des commandes de formatage) directement au fichier `.ext` qui génère la table des matières, ou la liste des figures ou des tableaux. generates the table of contents or listes of figures or tables.

`ext` L'extension de nom de fichier du fichier dans lequel l'information est à écrire : `toc` (table des matières), `lof` (liste des figures), ou `lot` (liste des tableaux).

`texte` Le texte à écrire.

25.1.3 `\nofiles`

Synopsis :

```
\nofiles
```

Empêche L^AT_EX 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^AT_EX n'efface pas les fichiers auxiliaires existants, de sorte que si vous insérer la commande `\nofiles` après avoir compilé le fichier et obtenu un fichier `.toc` alors la page de table des matières contiendra l'ancienne information.

25.2 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

La commande `\makeindex` active la création d'index. Placez là dans le préambule.

La commande `\index{texte}` écrit un article d'index pour *texte* dans un fichier auxiliaire nommé avec une extension `.idx`.

Plus précisément, ce qui est écrit est la commande `\indexentry{texte}{numpage}`, où *numpage* est la valeur courante de `\thepage`.

Pour générer un article d'index pour « toto » qui dit « Voir titi », utilisez une barre verticale : `\index{toto|see{titi}}`. Utilisez `seealso` au lieu de `see` pour fabriquer un article en « Voir aussi ».

Le texte « Voir » est défini par la macro `\seename`, et le texte « Voir aussi » par la macro `\alsoname`. Ils peuvent être redéfinis pour d'autres langages.

Le fichier en `.idx` généré est ensuite trié avec une commande externe, d'ordinaire soit `makeindex` (<http://mirror.ctan.org/indexing/makeindex>) ou la commande (multilingue) `xindy` (<http://xindy.sourceforge.net>). Ceci résulte en un fichier en `.ind`, ce qui peut ensuite être lu pour composer l'index.

L'index est d'ordinaire généré avec la commande `\printindex`. Celle-ci est définie dans le paquetage `makeidx`, ainsi vous devez mettre `\usepackage{makeidx}` dans le préambule.

La longueur élastique `\indexspace` est insérée avant chaque nouvelle lettre dans l'index imprimé ; sa valeur par défaut est `'10pt plus5pt minus3pt'`.

Le paquetage `showidx` a pour effet que chaque article de l'index est affiché dans la marge de la page où l'article apparaît. Ceci peut aider à la préparation de l'index.

Le paquetage `multind` prend en charge la pluralité des index. Voir aussi l'article de la FAQ `TEX` FAQ à ce propos, <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=multind>.

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 contr'oblique (`\`). Par exemple, vous pourriez avoir :

```

\begin{letter}{M. Joseph Fabre \
  25 rue de la République\
  35700 Rennes}
...
\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 contr'oblique `\`. \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.}`.

Du matériel 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 contr'oblique `\\`.

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 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.

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 contr'oblique `\\`, 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 contr'oblique `\\`. Par exemple :

```
\signature{J Fred Muggs \
White House}
```

La valeur par défaut de L^AT_EX 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 preambule, 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 telephone 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. Le *numéro* doit être entre 0 et 15, comme dans `\openin3` (dans Lua \TeX *numéro* peut être entre 0 et 255).

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 \LaTeX 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 case-sensitive) et les dix chiffres, ainsi que les tiret 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 178).

Pour lire un fichier entier en une fois utilisez `\input` (voir Section 24.3 [`\input`], page 167) ou `\include` (voir Section 24.2 [`\include` & `\includeonly`], page 165).

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 définis avec `\newcommand` ou `\renewcommand` (entre autres) sont remplacées par leurs définitions avant d'être tapée.

Les règles usuelles de L^AT_EX 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 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 T_EX exécute `\write3` ce qui place la chaîne dans le fichier.

Typiquement *numéro* est compris entre 0 et 15, parce que typiquement les auteurs L^AT_EX 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 L^AT_EX écrit *chaîne* vers le fichier journal. Si *numéro* est positif alors ne plus L^AT_EX 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 181). 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 LuaT_EX, au lieu de 16 flux de sortie il y en a 256 (voir Section 2.3 [T_EX 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 L^AT_EX.

Par défaut L^AT_EX 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 T_EX 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 stocké 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 \tripler que l'on a définie comme le texte 'XYZ' :

```
\newwrite\ichierjh
\openout\ichierjh=test.jh
\newcommand{\tripler}{XYZ}
\write\ichierjh{test \tripler 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 178). 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 182).

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 fichier 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 \TeX 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\ichierjh
```

```
\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 \LaTeX va essayer d'ouvrir `../filename.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 \LaTeX utilisent `\typeout` (voir Section 27.4 [`\typeout`], page 178) qui permet d'utiliser `\protect` sur toute commande fragile au sein de *chaîne* (voir Section 12.11 [`\protect`], page 101). 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) \LaTeX 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 \LaTeX 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 \LaTeX 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 \LaTeX 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 \LaTeX est bloquée jusqu'à ce que cela soit fini.

La séquence suivante (sur Unix)

```
\usepackage{graphicx} % dans le préambule
...
\newcommand{\numérofig}{1}
\immediate\write18{cd pix && asy figure\numérofig}
\includegraphics{pix\figure\numérofig.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 148). Comme pour tout `\write`, ici \LaTeX développe les macros contenues dans *commande_shell* de sorte que `\numérofig` est remplacé par '1'.

Un autre exemple est que vous pouvez exécuter automatiquement \BibTeX au début de chaque exécution de \LaTeX (voir Section 8.24.4 [Using \BibTeX], page 82) 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 179). 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 \LaTeX pose des problèmes de sécurité évidents. Si vous téléchargez 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 184). (L'option `--shell-escape` est un synonyme dans $\text{T}_{\text{E}}\text{X}$ 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 184).

Certains paquetage ont besoin d'exécuter des commandes système externes. Par exemple `sagetex` vous permet de faire faire des calcul ou tracer des graphiques au logiciel de mathématiques *Sage*, puis d'incorporer ces sorties dans votre document. Pour cela vous devez exécuter $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 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. $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 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 Ligne de commande

Synopsis (depuis la ligne de commande d'un terminal) :

```
pdflatex options argument
```

Exécute \LaTeX sur *argument*. À la place de `pdflatex` vous pouvez aussi utiliser `xelatex`, ou `lualatex`, ou `latex`.

L'exemple suivant exécute \LaTeX sur le fichier `these.tex` produisant en sortie `these.pdf`.

```
pdflatex these
```

Notez que `.tex` est l'extension par défaut des noms de fichier.

`pdfTeX` est une évolution du programme original `TeX`, comme le sont `XeTeX` et `LuaTeX` (voir Section 2.3 [TeX 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 184, 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 contr'oblique, dans ce cas le système le prend pour un nom de fichier, et compile ce fichier. Si *argument* commence avec une contr'oblique, alors le système l'interprète comme une ligne d'entrée à \LaTeX , ce qui peut être utilisé pour obtenir certains effets spéciaux (voir Section 28.2 [Command line input], page 186).

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 \LaTeX trouve une erreur dans votre document alors par défaut il s'arrête et vous interroge à ce propos. Voir Section 28.3 [Recovering from errors], page 187, 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.

Les implementations permettent en général de spécifier les options de ligne de commande avec un préfixe `'-'` ou `'--'`. C'est le cas à la fois pour `TeX Live` (et `MacTeX`) et `MiKTeX`. Ici on utilise les deux conventions indistinctement.

- `-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

\TeX 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 batch 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 démarrnant \LaTeX avec cette ligne de commande :

```
pdflatex -interaction=batchmode nomfichier
```

on élimine la plupart des sorties terminal.

-jobname=chaîne

Règle la valeur du *nom de tâche* de \TeX à la chaîne de caractères *chaîne*. Le fichier journal et le fichier de sortie s'appelleront `chaîne.log` et `chaîne.pdf`.

Quand vous exécutez `pdflatex options argument`, si *argument* ne commence pas par une contr'oblique alors \TeX considère que c'est le nom d'un fichier à lire en entrée. Autrement il attend la première instruction `\input` et le nom du fichier en entrée devient le nom de tâche (`\jobname`). Celui-ci est utilisé pour nommer le fichier journal et le fichier de sortie. L'option `jobname` l'emporte et spécifie directement le nom. Voir Section 28.2 [Command line input], page 186, pour un exemples d'usage.

-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 \LaTeX , tels que le fichier `.log` de l'exécution, les fichiers `.aux`, `.toc`, etc., créés par \LaTeX , 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 \LaTeX 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 182). 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 \LaTeX d'exécuter n'importe quelle commande. Ainsi, par défaut,

le `\write18` non restreint n'est pas autorisé. (Le réglage par défaut de \TeX Live, $\text{Mac}\TeX$, et $\text{MiK}\TeX$ est d'autoriser l'exécution d'un nombre limité de programmes en relation avec \TeX , et qu'elles distribuent).

Par exemple, si vous invoquez \LaTeX 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 $\text{Mac}\TeX$.

28.2 Spécifier du code \LaTeX sur la ligne de commande

Au sein de l'invocation de ligne de commande `pdflatex options argument` vous pouvez spécifier du code \LaTeX arbitraire en entrée en commençant `argument` par une contr'oblique. Ceci permet de faire certains effet spéciaux.

Par exemple, ce fichier (qui utilise le paquetage `hyperref` pour les hyperliens) peut produire deux types de sortie, l'un pour l'impression papier et l'autre pour une consultation PDF.

```
\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. (Notez l'utilise des guillemets anglais pour empêcher l'interprétation des symboles par l'interpréteur de ligne de commande ; votre système fait peut-être cela différemment).

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.

Dans ce dernier exemple on charge le paquetage `graphicx` avec l'option `draft`

```
pdflatex -jobname=aa "\RequirePackage[draft]{graphicx}\input{aa.tex}"
```

de sorte que les fichiers graphiques ne sont lus que pour l'information de dimension de boîte englobante, mais non incorporés dans le PDF. (L'option `jobname` est la parce que

sinon autrement le fichier de sortie s'appellerait `graphicx.pdf`, comme `\RequirePackage` fait lui-même un `\input`).

28.3 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 Control 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#latextemplates>.

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 189, 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 189, 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 37). 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 164). 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. (Le matériel 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,
  ...
}

\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 29
- '*' , invite 187

—

- 1, écrire dans le flux numéro 179

•

- .glo, fichier 170
- .idx, fichier 171
- .ind, fichier 171

^

- ^^J, dans `\write` 180

\

- `\include` imbriqués non autorisés 167
- `\write` et sécurité 180
- `\write` immédiat 180
- `\write` retardé 180
- `\write`, flux 16, 17, 18 180
- `\write18`, activation 185

- écrire des lettres 172
- écrire un fichier externe 52
- élément extraordinaire 180
- élastiques, définir de nouvelles longueurs 97
- élastiques, longueurs 106
- équations, aligner 51
- équations, environnement pour 51
- étiquette 42
- étiquette machine à écrire dans les listes 48
- étiquette tapuscrite dans les listes 48
- étirement, omettre l'étirement vertical 24
- étoilée, forme des commandes de rubricage 29
- étoilées, variantes de commandes 95

A

- a e liés 162
- a rond-en-chef 162
- a, e dans l'a 162
- accéder à un caractère
 - quelconque d'une police 157
- accent 17, 18
- accent aigu 161
- accent aigu en math 125
- accent barre suscrite en math 125
- accent brève en math 125
- accent caron 162
- accent caron en math 125
- accent chapeau en math 125
- accent circonflexe 161
- accent circonflexe en math 125
- accent diérèse en math 125
- accent double-brève-renversée 162
- accent grave en math 125
- accent háček 162
- accent háček en math 125
- accent macron en math 125
- accent point 161
- accent point en chef 161
- accent point suscrit 161
- accent point suscrit en math 125
- accent rond en chef 161
- accent rond-en-chef en math 126
- accent tilde en math 126
- accent tréma 160
- accent tréma en math 125
- accent tréma hongrois 161
- accents 160
- Accents mathématiques 125
- accentuées, lettres 160
- accentués, caractères 160
- accolade droite, dans le texte 159
- accolade gauche, dans le texte 158
- afficher du texte cité avec
 - renforcement de paragraphe 70
- afficher un texte cité sans
 - renforcement de paragraphe 70
- agrandir la page courante 88
- align environnement, d'`amsmath` 51
- alignement au moyen de `tabbing` 70
- aligner des équations 51
- aligner du texte dans des tableaux 76
- aligner du texte en utilisant des
 - taquets de tabulation 70
- allure de police 21
- `amsmath` paquetage, remplaçant `eqnarray` 51
- anomalies, rapports 1
- aperçu de `LATEX` 2
- appel 175

- appendices 36
 - appendix 36
 - arguments mouvants 101
 - arrière-plan coloré 143
 - article d'index, « Voir » et « Voir aussi » 171
 - article d'une liste, compteur spécifiant 103
 - articles numérotés, compteur spécifiant 103
 - ASCII, circonflexe dans le texte 158
 - ASCII, tilde dans le texte 158
 - astérisque centré, dans le texte 158
 - aval, référence déclarée en 42
- B**
- barre au-dessous 162
 - barre par-dessus, accent 161
 - barre sousscite 162
 - barre verticale double, dans le texte 158
 - barre verticale, dans le texte 158
 - barre, diacritique 161
 - barre, double verticale, dans le texte 158
 - barre-sousscite, diacritique 161
 - Bas de casse 156
 - bas de page, créer des notes 90
 - bas de page, des paramètres pour 24
 - bases de L^AT_EX 2
 - batchmode 185
 - beamer** patron et classe 188
 - bibliographie, créer (automatiquement) 82
 - bibliography, creating (manually) 81
 - bib_TE_X, utilisation 82
 - Big point 107
 - boîte englobante 150
 - boîte, allouer une nouvelle 97
 - boîtes 137
 - boîtes colorées 143
 - boîtes noires, en omettant 8
 - bogues, rapports 1
 - Bonjour le monde 2
 - book, annexes 37
 - book, pièces postliminaires 37
 - book, pièces préliminaires 37
 - book, pièces principales 37
 - brève, accent 162
 - brève, diacritique 162
- C**
- cédille, diacritique 161
 - césure, définir 87
 - césure, empêcher 137
 - césure, forcer 86
 - cadratin 108, 159
 - cadratin, demi 159
 - cadratin, deux tiers 160
 - cadratin, trois-quarts 160
 - Capitales 156
 - caractères latins non diacrités 162
 - caractères réservés 156
 - caractères spéciaux 156
 - caractères speciaux 162
 - caron, diacritique 162
 - Casse de caractères 156
 - catégorie, code de caractère 95
 - catcode 95
 - Centimètre 107
 - centré, astérisque, dans le texte 158
 - centré, point, dans le texte 159
 - centrer du texte, déclaration pour 47
 - centrer du texte, environnement pour 47
 - Changer la casse de caractères 156
 - chapeau, accent 161
 - chapitre 29, 31
 - chapter** 29
 - chargement des paquetages supplémentaires 9
 - chevron double, guillemets 158
 - chevron droit, guillemets 158
 - chevron gauche, guillemets 158
 - chevron simple, guillemets 158
 - chiffres de revêtement 18
 - chiffres de style ancien 18
 - chiffres de style ancien, 18
 - Cicero 108
 - circonflexe ASCII, dans le texte 158
 - circonflexe, accent 161
 - circonflexe, diacritique 161
 - citation 70
 - 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 101
 - clef de référence bibliographique 81
 - code de catégorie de caractère 95
 - code entré sur la ligne de commande 186
 - code source, composer du 83
 - color options du paquetage 140
 - color, commandes du paquetage 141
 - coloré, texte 142
 - colorée, page 143
 - colorées, boîtes 143
 - commande de longueur 108
 - commande longue 10
 - commandes de fontes de bas niveau 19
 - commandes de fontes, de bas niveau 19
 - commandes de mise en page 23
 - commandes de rubricage, 29
 - commandes des classes de document 9
 - commandes exécutées depuis L^AT_EX 182
 - commandes fragiles 101
 - commandes pour classe ou paquetage 10

commandes robustes	101
commandes, définir de nouvelles	94
commandes, en définir de nouvelles	93
commandes, paquetage graphics	148
commandes, syntaxe des	4
commandes, variantes étoilées	95
commencer une nouvelle page	88
commencer une nouvelles page et débourrer les flottant	88
commencer une page de droite	88
compteurs, a list of	103
compteurs, définir de nouveaux	96
compteurs, obtenir la valeur d'un	104
compteurs, réglage	104
configuration du paquetage graphics	145
contr'oblique, dans le texte	158
copie, liste des personnes en, dans une lettre ..	173
Correction italique	133
couleur	140, 141, 142, 143
couleur, définir	141
coupure de mot, empêcher	137
créer des images	65
créer des tableaux	75
crésure, discrétionnaire	86
CTAN	5
diacritique tirant sussescrit	162
diacritique tréma hongrois	161
diacritique accent aigu	161
diacritique tréma	160
diacritiques, signes	160
différence entre classe et paquetage	9
dilatation verticale infinie	135
dimen \TeX de base	106
discrétionnaire, césure	86
dollar, signe	159
double obèle, dans le texte	157
double renvoi, résolution	3
double, barre verticale, dans le texte	158
double, guillemet-apostrophe culbuté	160
double, guillemet-apostrophe de droite	160
double, obèle, dans le texte	159
double-brève-renversée, diacritique	162
droit d'auteur, symbole	157
droit, guillemet double sur la ligne de base ..	160
droit, guillemet simple sur la ligne de base ..	160
droit, guillemet-apostrophe simple	160
droite, accolade, dans le texte	159
droite, flèche, dans le texte	160
droite, guillemet anglais simple	158
droite, guillemet-apostrophe simple	158

D

débourrer les flottant et commencer une page ..	88
début de document, crochet	49
début et fin	2
déclaration en aval de référence	42
déclarative, forme des commandes de contrôle de la taille de police	19
définir de nouveaux théorèmes	99
définir de nouvelles polices	101
définir une couleur	141
définir une nouvelle commande	93, 94
définitions	93
délimiteur nul	127
DANTE e.V.	6
date d'aujourd'hui	163
defining new environnements	97
demi-cadratin	159
deux-tiers cadratin	160
diacritique barre-souscrite	161
diacritique brève	162
diacritique cédille	161
diacritique caron	162
diacritique circonflexe	161
diacritique double-brève-renversée	162
diacritique háček	162
diacritique macron-souscrit	161
diacritique ogonek	161
diacritique point suscrit	161
diacritique point-souscrit	161
diacritique rond en chef	161
diacritique tilde	161

E

e dans l'a	162
e dans l'o	163
e- \TeX	3
em	108
en-tête et pied de page en cours de traitement ..	24
en-tête, des paramètres pour	24
encerclée, lettre, dans le texte	159
englobante, boîte	150
entré, code sur la ligne de commande	186
entrée Unicode, native	3
entrée/sortie console	177
environnements	45
environnement	2
environnement simili-théorème	99
environnements, defining	97
EPS, fichiers	145, 148
errorstopmode	185
espace vertical	134
espace vertical avant les paragraphes	110
espace vertical, insérer	134
espace verticale	135
espace visible	83
espace visible, symbole, dans le texte	160
espacement au sein du mode mathématique ..	126
espacement inter-phrase	133
espaces	132
eszet lettre allemande	163
eth islandais	162
eth, lettre islandaise	162
euro, symbole	159

exploitation, information système de	183
exposant	112
extension, structure	9
externes, commandes	182

F

féminin, symbole ordinal	159
fabriquer une page de titre	83
Faire des paragraphes	110
famille de polices	20
fermant, guillemet anglais simple	158
fermant, guillemet-apostrophe simple	158
fichier auxiliaire	3
fichier d'entrée	164
fichier de transcription	3
fichier externe, écrire	52
fichier journal	3
fichier racine	164
fichier, lecture d'un	178
fichiers source, rendre autonomes	53
figures, insertion	51
figures, notes en bas de page au sein des	64
file, closing	177
file, opening	177
fin d'une lettre	174
fin document, crochet	49
fin et début	2
flèche droite, dans le texte	160
flèche gauche, left, dans le texte	159
fonctions mathématiques	124
force , option de <code>filecontents</code>	53
forme étoilée de commandes de rubricage	29
forme étoilée des commandes d'environnement	97
forme étoilée, définir de nouvelles commandes ..	93
forme déclarative des commandes de contrôle de la taille de police	19
forme en * de commandes de rubricage	29
forme en * des commandes d'environnement	97
forme en *, définir de nouvelles commandes	93
forme par environnement des commandes	19
formules en ligne	62
formules mathématiques	112
formules, environnement pour	51
formules, math	112
fragile, commande	101
français, guillemets	158

G

gauche, accolade, dans le texte	158
gauche, flèche, dans le texte	159
gauche, guillemet-apostrophe simple de	158
globales, options	9
glossaires	170
glue T _E X de base	106
grasses de police	20
Grand rond, symbole, dans le texte	158
graphics	145
graphics, commandes du paquetage	148
graphics, options du paquetage	144
graphics, paquetage	144, 145, 148
graphics, resizing	155
graphics, scaling	155
graphiques	144, 148
graphiques, paquetages	68
gras	18
gras machine à écrire, éviter	48
gras tapuscrit, éviter	48
grave accent	161
guillemet droit double sur la ligne de base	160
guillemet droit simple sur la ligne de base	160
guillemet-apostrophe de droite double	160
guillemet-apostrophe double culbuté	160
guillemet-apostrophe simple	160
guillemet-apostrophe simple culbuté	160
guillemet-apostrophe simple droit	160
guillemet-virgule inférieur double	158
guillemet-virgule inférieur simple	158
guillemets double	158
guillemets français	158
guillemets simple	158
guillemets-virgule inférieurs, simple et double	158

I

i sans point	161
i sans point en math	126
Icelandic thorn	163
ici, placer les flottants	26
images, créer	65
imbrication de <code>\include</code> non autorisée	167
implémentations de T _E X	3
importer des graphiques	148
inclure des graphiques	148
index	171
indice inférieur	112
indice supérieur	112
inférieur à, symbole dans le texte	159
inférieur, guillemet-virgule double	158
inférieur, guillemet-virgule simple	158
insérer des figures	51
interligne double	21
interpréteur de commandes du système d'exploitation, accès à	182
invite '*'	187

italique 18

J

j sans point 161
 j sans point en math 126
 jobname 185
 journal, écrire vers le fichier 179
 JPEG, fichiers 145, 148
 JPG, fichiers 145, 148
 jsutifier à gauche du texte,
 environnement pour 53
 justifié à gauche du texte 53
 justification, drapeau droit 53
 justification, drapeau gauche 54
 justifier à droite, environnement pour 54
 justifier le texte à droite 54

K

Knuth, Donald E. 2

L

l polonais 162
 L'équipe de projet L^AT_EX 1
 L^Ampo_TE_X 2
 L^Ampo_TE_X, Leslie 2
 Largeurs de police 21
 L^AT_EX aperçu 2
 L^AT_EX ou L^AT_EX₂e, que choisir ? 1
 L^AT_EX, logo 157
 L^AT_EX₂e logo 157
 lecture d'un fichier 178
 Lettre grecques 113
 lettre ij, Holandais 162
 lettres calligraphiques pour les
 mathématiques 18
 lettres de script pour les mathématiques 18
 lettres latines non diacritées 162
 lettres, écrire des 172
 lettres, commencer 175
 lettres, salutation 174
 ligne de commande 184
 ligne tabulée, tabbing 71
 lignes dans des tableaux 76
 lire un fichier 178
 liste des figures, fichier 3
 liste des personnes en copie, dans une lettre ... 173
 liste des tableaux, fichier 3
 listes à puces 54
 listes d'articles 54
 listes d'articles, générique 56
 listes de description, créer 47
 listes non ordonnées 54
 lists étiquetée, créer 47
 lists of items, numbered 49
 livre, annexes 37

livre, pièces postliminaires 37
 livre, pièces préliminaires 37
 livre, pièces principales 37
 livre, symbole 158
 locale, information, du système 183
 logo L^AT_EX 157
 logo L^AT_EX₂e 157
 logo T_EX 158
 longueur, ajouter à une 108
 longueur, commande de 108
 longueur, régler une 108
 longueurs élastiques, définir de nouvelles 97
 longueurs prédéfinies 109
 longueurs, allouer de nouvelles 97
 longueurs, définir et utiliser 106
 LuaT_EX 4
 LuaT_EX, 256 flux de sortie dans 179

M

m, largeur d'un 108
 macron, diacritique 161
 macron-souscrit, diacritique 161
 Madsen, Lars 51
 makeindex, programme 171
 marque déposée, symbole 160
 marque de fabrique, symbole 160
 masculin, symbole ordinal 159
 math, accent aigu 125
 math, accent barre suscrite 125
 math, accent brève 125
 math, accent grave 125
 math, accent macron 125
 math, accent point suscrit 125
 math, accent rond-en-chef 126
 math, accents 125
 math, fonctions 124
 math, miscellanées 126
 mathématiques gras 18
 mathématiques, gras 18
 matières, fichier 3
 Millimètre 107
 minipage, créer une 63
 mirrors of CTAN 6
 miscellanées en math 126
 mise en page, les paramètres de 24
 modèle de document 188
 modèles de couleur 140
 mode gauche-à-droite (left-to-right) 129
 mode LR 129
 mode math 129
 mode mathématique, entrer 112
 mode mathématique, espacement 126
 mode paragraphe 129, 138
 modes 129
 monnaie, dollar 159
 monnaie, euro 159
 moteur pdfT_EX 3

moteurs, \TeX 3
 motif courant de taquets de
 tabulation, `tabbing` 71
 mpfootnote, compteur 90
 mu, unité mathématique 108
 multiplication discrétionnaire 126

N

`noheader`, option de `filecontents` 53
 nom de fichier de la tâche courante 185
 nonstopmode 185
`nosearch`, option de `filecontents` 53
 note en bas de page, créer 90
 note en bas de page, symboles au
 lieu de numéros 90
 notes dans la marge 110
 notes en bas de page au sein des figures 64
 Notes en bas de page, au sein d'une minipage... 90
 notes en bas de page, dans les
 titres de rubrique 92
 Notes en bas de page, dans un tableau 91
 notes en bas de page, paramètres 92
 Notes en marge 110
 nouvelle commande, définition 12
 nouvelle commande, vérification 10
 nouvelles commandes, définir 93
 nouvelle ligne, commencer une 85
 nouvelle ligne, commencer une
 (mode paragraphe) 85
 nouvelle page, commencer 88
 nouvelles classes, commandes 9
 nouvelles commandes, définir 94
 numéro d'équation, omettre 51
 numéro d'équation, renvoi 44
 numéro de figure, renvoi 44
 numéro de note en bas de page, renvoi 44
 numéro de page, renvoi 43
 numéro de rubrique, renvoi 44
 numéros de rubrique, composition 30
 numérotation de page, style 131

O

o e liés 163
 o, e dans l'o 163
 obèle double, dans le texte 159
 obèle, dans le texte 157, 159
 officieuse, nature de ce document 1
 ogonek, diacritique 161
 open a file 177
 option `stable` du paquetage `footmisc` 92
 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 184
 options pour classe de document 9

options pour paquetage 9
 options, classe de document 7
 options, paquetage `color` 140
 options, paquetage `graphics` 144
 Ordinaux espagnols, féminin et masculin 159
 Ordinaux, féminin et masculin 159
 oslash 163
 ouvrant, , guillemet-apostrophe simple 158
`overwrite`, option de `filecontents` 53

P

page colorée 143
 page de flottants 26
 page styles 131
 page, paramètres de mise en page 24
 pages de titre, création 83
 pages, saut 88
 paquet de macro, \LaTeX en tant que 2
 paquetage, disposition du fichier 9
 paquetage, options 9, 11
 paquetages, le chargement 9
`paragraph` 29, 35
 paragraphe 29
 paragraphe, dans une boîte 138
 paragraphe, mode 138
 paragraphe, symbole 158
 paragraphes 110
 paragraphes, renforcement au sein
 d'une minipage 64
 paramètre positionnel 93, 98
 paramètres, la mise en page 24
 paramètres, pour notes en bas de page 92
`part` 29, 30
 partie 29
 partitionner le fichier d'entrée 164
 pas, registre de, \TeX de base 97
 patron (simple), `article` 188
 patron `book` 189
 patron de document 188
 patron, `beamer` 188
 patron, `book` 189
 PDF, fichiers graphiques 145, 148
 pdf \TeX 3
 pdf \TeX , moteur 3
 pièces jointes, liste 174
 pica 107
`pict2e`, paquetage 68
 Pied-de-mouche 158
 placement des flottants 26
 PNG, fichiers 145, 148
 poésie, un environnement pour la 84
 Point 107
 point centré, dans le texte 159
 point d'exclamation culbuté 159
 point d'interrogation culbuté 159
 point Didot 108
 point en chef, diacritique 161

Point proportionné (Scaled point) 108
 point suscrit, diacritique 161
 points de suspension 158
 police de largeur fixe 18
 police inclinée 18
 police machine à écrire 18
 police oblique 18
 police petites capitales 18
 police romaine 18
 police tapuscrit 18
 police, symboles par leur numéro 157
 polices de caractères 17
 polices OpenType 3
 polices TrueType 3
 polices, catalogue 20
 polices, commandes pour
 déclarer de nouvelles 101
 position, dans un dessin `picture` 66
 postscriptum, dans une lettre 175
 préambule, définition 2
 prédéfinies, longueurs 109
 prononciation 2
 puce, dans le texte 159

Q

quad 126

R

référence bibliographique, clef 81
 référence déclarée en aval 42
 réglage de compteurs 104
 répertoire de sortie pour tous les
 fichiers externes 185
 répertoire, listage, du système 183
 répertoires parents, impossible d'écrire vers 181
 réservés, caractères 156
 résumé 45
 racine, fichier 164
 rapports d'anomalies 1
 rapports de bogues 1
 redefining environnements 97
 remarks in the marge 110
 renforcement des paragraphes
 dans du texte cité 70
 renforcement des paragraphes dans un
 texte cité, omettre 70
 renforcement des paragraphes, au
 sein d'une minipage 64
 renforcement, forcer 110
 renforcement, supprimer 110
 renvoi avec numéro de page 43
 renvoi en aval, résolution 3
 renvoi, résolution 3
 renvoi, symbolique 44
 renvois 42
 resizing 155

ressort, registre de, T_EX de base 97
 restreint, accès interpréteur de commande du
 système d'exploitation 183
 restreint, accès shell 183
 rigides, longueurs 106
 robuste, commande 101
 rond en chef, diacritique 161
 rond, grand symbole, dans le texte 158
 rotating graphics 154
 rotating text 154
 rotation 154
 rubricage, commandes 29
 rubricage, part 30

S

s long 163
 séries de polices 20
 sans serif 18
 saut à la ligne 85
 saut de ligne 85
 saut de ligne, dans `\write` 180
 saut de ligne, forcer 85
 saut de ligne, sortie semblable à
 l'entrée pour le 85
 saut de page 88
 saut de page, empêcher 89
 saut de page, forcer 89
 sauter à la ligne 85
 sauts de ligne, empêcher 87
 sauts de ligne, forcer 87
 scaling 155
 scrollmode 185
 section 29, 33
 section, redéfinir 37
 section, symbole 158
 sécurité et `\write` 180
 shell escape 185
 signes diacritiques 160
 simple droit, guillemet-apostrophe 160
 simple, guillemet-apostrophe 160
 simple, guillemet-apostrophe culbuté 160
 simuler du texte tapuscrit 83
 skip T_EX de base 106
 sortie à deux colonnes 23
 sortie à une colonne 23
 sortie, pour tous les fichiers
 externes, répertoire de 185
 souligné, dans le texte 160
 source logiciel, composer du 83
 sources autonomes 53
 sous-paragraphe 29
 sous-section 29
 sous-sous-section 29
 spéciaux, caractères 156
 spécificateur, placement de flottants 26
 speciaux, caractères 162
stable, option du paquetage `footmisc` 92

sterling, symbole.....	158
style de numérotation de page.....	131
styles de police.....	17
styles de texte.....	17
styles typographiques.....	17
styles, page.....	131
subparagraph	29, 35
subsection	29, 34
subsubsection	29, 35
supérieur à, symbole dans le texte.....	159
suspension, points de.....	158
symbole de multiplication, saut de	
ligne discretionnaire.....	126
symbole droit d'auteur.....	157
symbole marque déposée.....	160
symbole marque de fabrique.....	160
symbole puce.....	114
symbole vecteur en math.....	126
symboles d'une police, par leur numéro.....	157
symboles mathématiques.....	113
symboles texte.....	157
symbols, mathématiques.....	113
système, commandes exécutées depuis \LaTeX ..	182
système, information.....	183

T

table des matières, création.....	168
table des matières, éviter les notes	
en bas de page.....	92
table des matières, ajouter	
manuellement un article.....	169
table des matières, fichier.....	3
table des matières, impression des	
numéros de rubrique.....	30
tableaux mathématiques.....	46
tableaux, créer.....	75
taille de conception, au sein d'une	
définition de police.....	101
tailles de police.....	19
tailles de texte.....	19
tailles des polices de caractères.....	19
taquet courant, tabbing.....	71
taquet de début de ligne courant, tabbing.....	71
taquets de tabulation, utiliser.....	70
terminal, écrire vers le.....	179
\TeX , logo.....	158
text, resizing.....	155
text, scaling.....	155
texte cité avec renforcement des	
paragraphes, affichage.....	70
texte cité sans renforcement des	
paragraphes, affichage.....	70

texte drapeau droit.....	53
texte drapeau droit, environnement pour.....	53
texte drapeau gauche.....	54
texte drapeau gauche, environnement pour.....	54
texte en couleur.....	142
texte fer à gauche.....	53
texte fer à gauche, environnement pour.....	53
texte ferré à gauche.....	53
texte ferré à gauche, environnement pour.....	53
texte sur plusieurs colonnes.....	23
texte tapuscrit, simulation de.....	83
texte verbatim.....	83
texte verbatim, en-ligne.....	83
théorème, environnement simili.....	99
théorèmes, composition des.....	83
théorèmes, définition.....	99
thickspace.....	126
thinspace.....	126
thorn, lettre islandaise.....	163
tilde ASCII, dans le texte.....	158
tilde, diacritique.....	161
tirant suscrit, diacritique.....	162
tiret bas, dans le texte.....	160
tréma hongrois, accent.....	161
tréma hongrois, diacritique.....	161
trois-quarts, cadratin.....	160

U

unités, de longueur.....	107
UTF-8.....	3
utilisation de \BibTeX	82

V

variables, a list of.....	103
verticale infinie, dilatation.....	135
verticale, barre dans le texte.....	158
verticale, espace.....	135
visible, symbole espace, dans le texte.....	160
Voir et Voir aussi, article d'index.....	171

W

wide hat accent, math.....	126
wide tilde accent, math.....	126
Wright, Joseph.....	157

X

x, hauteur d'un.....	108
\XeTeX	4
xindy, programle.....	171

Index des commandes

\$

\$	112
\$ en ligne, math	62
\$\$...\$\$ displaymath, forme approchée en T _E X de base	49

&

& pour cellules de tableau	76
----------------------------	----

—

--disable-write18 ligne de commande, option	185
--enable-write18 ligne de commande, option	185
--file-line-error ligne de commande, option	186
--halt-on-error ligne de commande, option	186
--help ligne de commande, option	184
--interaction ligne de commande, option	184
--jobname ligne de commande, option	185
--no-file-line-error ligne de commande, option	186
--no-shell-escape ligne de commande, option	185
--output-directory ligne de commande, option	185
--shell-escape ligne de commande, option	185
--version ligne de commande, option	184

.

.aux fichier	3
.fd fichier	101
.lof fichier	3, 168
.lot fichier	3, 168
.pdf fichier	3
.tex, extension par défaut	184
.toc fichier	3
.toc, fichier	168
.xdv fichier	4

/

/bin/sh, utilisé par \write18	183
-------------------------------	-----

[

[...] pour les arguments optionnels	4
-------------------------------------	---

^

^ indice supérieur	112
--------------------	-----

-

_ indice inférieur	112
--------------------	-----

\

\ caractère de début des commandes	4
\!	126
\" (accent tréma)	160
\#	156
\\$	156
\%	156
\&	156
\' (acute accent)	161
\' (tabbing)	73
\(112
\(...\) en ligne, math	62
\)	112
*	126
\+	72
\,	126
\-	73
\- (césure)	86
\. (dot-over accent)	161
\/	133
\:	126
\;	126
\<	72
\= (macron accent)	161
\= (tabbing)	71
\>	72, 126
\> (tabbing)	72
\[112
\[...] displaymath	48
\]	112
\^	156
\^ (accent circonflexe)	161
_	156
\' (accent grave)	161
\' (tabbing)	73
\@auxout	179
\@beginparpenalty	61
\@endparpenalty	61
\@fnsymbol	90
\@ifstar	95
\@itempenalty	61
\@mainaux	179
\@startsection	37
\\ (for eqnarray)	51
\\ (pour center)	47
\\ (pour flushright)	54

<code>\</code> (pour les objets <code>\shortstack</code>).....	70	<code>\baselinestretch</code>	21
<code>\</code> (tabbing).....	71	<code>\begin</code>	45
<code>\</code> for <code>flushleft</code>	53	<code>\beta</code>	113
<code>\</code> force un saut de ligne.....	85	<code>\bf</code>	18
<code>\</code> pour les lettres.....	172	<code>\bfseries</code>	17
<code>\</code> pour <code>tabular</code>	76	<code>\bibitem</code>	81
<code>\</code> pour <code>verse</code>	84	<code>\bibliography</code>	82
<code>\</code> * (pour <code>eqnarray</code>).....	51	<code>\bibliographystyle</code>	82
<code>\{</code>	156	<code>\bibname</code>	81
<code>\}</code>	156	<code>\bigcap</code>	113
<code>\ </code>	113	<code>\bigcirc</code>	113
<code>\~</code>	156	<code>\bigcup</code>	113
<code>\~</code> (accent tilde).....	161	<code>\bigodot</code>	113
<code>\a</code> (tabbing).....	73	<code>\bigoplus</code>	113
<code>\a'</code> (aigu, accent dans tabbing).....	73	<code>\bigotimes</code>	114
<code>\a=</code> (macron, accent dans tabbing).....	73	<code>\bigskip</code>	135
<code>\a'</code> (grave, accent dans tabbing).....	73	<code>\bigskipamount</code>	135
<code>\aa</code> (å).....	162	<code>\bigskipcup</code>	114
<code>\AA</code> (Å).....	162	<code>\bigtriangledown</code>	114
<code>\acute</code>	125	<code>\bigtriangleup</code>	114
<code>\addcontentsline{ext}{unit}{texte}</code>	169	<code>\biguplus</code>	114
<code>\address</code>	173	<code>\bigvee</code>	114
<code>\addtocontents{ext}{texte}</code>	170	<code>\bigwedge</code>	114
<code>\addtocounter</code>	104	<code>\bmod</code>	124
<code>\addtolength</code>	108	<code>\boldmath</code>	112
<code>\addvspace</code>	134	<code>\bot</code>	114
<code>\ae</code> (æ).....	162	<code>\bottomfraction</code>	27
<code>\AE</code> (Æ).....	162	<code>\bottomnumber</code>	27
<code>\aleph</code>	113	<code>\bowtie</code>	114
<code>\Alph</code>	103	<code>\Box</code>	114
<code>\alph</code>	103	<code>\breve</code>	125
<code>\Alph example</code>	50	<code>\bullet</code>	114
<code>\alpha</code>	113	<code>\c</code> (cédille).....	161
<code>\alsoname</code>	171	<code>\cal</code>	18
<code>\amalg</code>	113	<code>\cap</code>	114
<code>\angle</code>	113	<code>\capitalacute</code>	161
<code>\appendix</code>	36	<code>\capitalbreve</code>	162
<code>\approx</code>	113	<code>\capitalcaron</code>	162
<code>\arabic</code>	103	<code>\capitalcedilla</code>	161
<code>\arccos</code>	124	<code>\capitalcircumflex</code>	161
<code>\arcsin</code>	124	<code>\capitaldieresis</code>	160
<code>\arctan</code>	124	<code>\capitaldotaccent</code>	161
<code>\arg</code>	124	<code>\capitalgrave</code>	161
<code>\arraycolsep</code>	46	<code>\capitalhungarumlaut</code>	161
<code>\arrayrulewidth</code>	78	<code>\capitalmacron</code>	161
<code>\arraystretch</code>	79	<code>\capitalnewtie</code>	162
<code>\ast</code>	113	<code>\capitalogonek</code>	161
<code>\asympt</code>	113	<code>\capitalring</code>	161
<code>\AtBeginDocument</code>	49	<code>\capitaltie</code>	162
<code>\AtBeginDvi</code>	10	<code>\capitaltilde</code>	161
<code>\AtEndDocument</code>	49	<code>\caption</code>	52, 76
<code>\AtEndOfClass</code>	10	<code>\cc</code>	173
<code>\AtEndOfPackage</code>	10	<code>\cdot</code>	114
<code>\b</code> (accent barre-souscrite).....	161	<code>\cdots</code>	127
<code>\backmatter</code>	37	<code>\centering</code>	47
<code>\backslash</code>	113	<code>\chapter</code>	29, 31
<code>\bar</code>	125	<code>\check</code>	125
<code>\baselineskip</code>	21	<code>\CheckCommand</code>	10

<code>\CheckCommand*</code>	10	<code>\DH (Ð)</code>	162
<code>\chi</code>	114	<code>\Diamond</code>	115
<code>\circ</code>	114	<code>\diamond</code>	115
<code>\circle</code>	67	<code>\diamondsuit</code>	115
<code>\cite</code>	82	<code>\dim</code>	124
<code>\ClassError</code>	11	<code>\displaystyle</code>	112
<code>\ClassInfo</code>	11	<code>\div</code>	115
<code>\ClassInfoNoLine</code>	11	<code>\dj</code>	162
<code>\ClassWarning</code>	11	<code>\DJ</code>	162
<code>\ClassWarningNoLine</code>	11	<code>\documentclass</code>	7
<code>\cleardoublepage</code>	88	<code>\dot</code>	125
<code>\clearpage</code>	88	<code>\doteq</code>	115
<code>\cline</code>	80	<code>\dotfill</code>	134
<code>\closein</code>	177	<code>\dots</code>	158
<code>\closeout</code>	177	<code>\doublerulesep</code>	79
<code>\closing</code>	174	<code>\downarrow</code>	115
<code>\clubsuit</code>	114	<code>\Downarrow</code>	115
<code>\columnsep</code>	23	<code>\ell</code>	115
<code>\columnseprule</code>	23	<code>\em</code>	18
<code>\columnwidth</code>	23	<code>\emph</code>	17
<code>\complement</code>	114	<code>\emptyset</code>	115
<code>\cong</code>	114	<code>\encl</code>	174
<code>\contentsline</code>	170	<code>\end</code>	45
<code>\coprod</code>	114	<code>\endinput</code>	164
<code>\copyright</code>	157	<code>\enlargethispage</code>	88
<code>\cos</code>	124	<code>\enumi</code>	50
<code>\cosh</code>	124	<code>\enumii</code>	50
<code>\cot</code>	124	<code>\enumiii</code>	50
<code>\coth</code>	124	<code>\enumiv</code>	50
<code>\csc</code>	124	<code>\epsilon</code>	115
<code>\cup</code>	114	<code>\equiv</code>	115
<code>\CurrentOption</code>	11	<code>\espace</code>	132
<code>\d (dot-under accent)</code>	161	<code>\eta</code>	115
<code>\dag</code>	157	<code>\evensidemargin</code>	8
<code>\dagger</code>	114	<code>\ExecuteOptions</code>	13
<code>\dashbox</code>	67	<code>\exists</code>	115
<code>\dashv</code>	115	<code>\exp</code>	125
<code>\day</code>	105	<code>\extracolsep</code>	78
<code>\dblfloatpagefraction</code>	23	<code>\fbox</code>	137
<code>\dblfloatsep</code>	23	<code>\fboxrule</code>	67, 137
<code>\dbltextfloatsep</code>	23	<code>\fboxsep</code>	67, 137
<code>\dbltopfraction</code>	23	<code>\fill</code>	132
<code>\ddag</code>	157	<code>\flat</code>	115
<code>\ddagger</code>	115	<code>\floatpagefraction</code>	27
<code>\ddot</code>	125	<code>\floatsep</code>	27
<code>\ddots</code>	127	<code>\flushbottom</code>	24
<code>\DeclareGraphicsExtensions</code>	147	<code>\fnsymbol</code>	103
<code>\DeclareGraphicsRule</code>	147	<code>\fnsymbol, et note en bas de page</code>	90
<code>\DeclareOption</code>	11	<code>\fontencoding</code>	19
<code>\DeclareOption*</code>	11	<code>\fontfamily</code>	20
<code>\DeclareRobustCommand</code>	12	<code>\fontseries</code>	20
<code>\DeclareRobustCommand*</code>	12	<code>\fontshape</code>	21
<code>\deg</code>	124	<code>\fontsize</code>	21
<code>\delta</code>	115	<code>\footnote</code>	90
<code>\Delta</code>	115	<code>\footnotemark</code>	90
<code>\depth</code>	109	<code>\footnoterule</code>	92
<code>\det</code>	124	<code>\footnotesep</code>	92
<code>\dh (ð)</code>	162	<code>\footnotesize</code>	19

<code>\footnotetext</code>	91	<code>\index</code>	171
<code>\footskip</code>	24	<code>\indexentry</code>	171
<code>\forall</code>	115	<code>\indexspace</code>	171
<code>\frac</code>	127	<code>\inf</code>	125
<code>\frac{num}{dén}</code>	127	<code>\infty</code>	116
<code>\frame</code>	68	<code>\input</code>	167
<code>\framebox</code>	67, 137	<code>\InputIfFileExists</code>	12
<code>\frenchspacing</code>	133	<code>\int</code>	116
<code>\frontmatter</code>	37	<code>\intertextsep</code>	27
<code>\frown</code>	115	<code>\iota</code>	116
<code>\fussy</code>	87	<code>\item</code>	47, 48, 50, 54
<code>\gamma</code>	115	<code>\itemindent</code>	55, 58
<code>\Gamma</code>	115	<code>\itemsep</code>	55, 58
<code>\gcd</code>	125	<code>\itshape</code>	17
<code>\ge</code>	115	<code>\j</code> (j sans point)	161
<code>\geq</code>	115	<code>\jmath</code>	126
<code>\gets</code>	116	<code>\Join</code>	116
<code>\gg</code>	116	<code>\k</code> (ogonek)	161
<code>\glossary</code>	170	<code>\kappa</code>	116
<code>\glossaryentry</code>	170	<code>\ker</code>	125
<code>\graphicspath</code>	146	<code>\kill</code>	73
<code>\grave</code>	125	<code>\l</code> (\llcorner)	162
<code>\guillemotleft</code> («)	158	<code>\L</code> (\llcorner)	162
<code>\guillemotright</code> (»)	158	<code>\label</code>	42
<code>\guilsinglleft</code> (<)	158	<code>\labelenumi</code>	50
<code>\guilsinglright</code> (>)	158	<code>\labelenumii</code>	50
<code>\H</code> (tréma hongrois)	161	<code>\labelenumiii</code>	50
<code>\hat</code>	125	<code>\labelenumiv</code>	50
<code>\hbar</code>	116	<code>\labelitemi</code>	55
<code>\headheight</code>	24	<code>\labelitemii</code>	55
<code>\headsep</code>	24	<code>\labelitemiii</code>	55
<code>\heartsuit</code>	116	<code>\labelitemiv</code>	55
<code>\height</code>	109	<code>\labelsep</code>	55, 58
<code>\hfill</code>	132	<code>\labelwidth</code>	55, 58
<code>\hline</code>	80	<code>\Lambda</code>	116
<code>\hom</code>	125	<code>\lambda</code>	116
<code>\hookleftarrow</code>	116	<code>\land</code>	116
<code>\hookrightarrow</code>	116	<code>\langle</code>	116
<code>\hrulefill</code>	134	<code>\Large</code>	19
<code>\hspace</code>	25	<code>\large</code>	19
<code>\hsize</code>	132	<code>\LARGE</code>	19
<code>\huge</code>	19	<code>\LaTeX</code>	157
<code>\Huge</code>	19	<code>\LaTeXe</code>	157
<code>\hyphenation</code>	87	<code>\lbrace</code>	116
<code>\i</code> (i sans point)	161	<code>\lbrack</code>	116
<code>\iff</code>	116	<code>\lceil</code>	116
<code>\IfFileExists</code>	12	<code>\ldots</code>	158
<code>\ij</code> (ij)	162	<code>\le</code>	116
<code>\IJ</code> (IJ)	162	<code>\leadsto</code>	116
<code>\il</code>	18	<code>\left delim1 ... \right delim2</code>	127
<code>\Im</code>	116	<code>\leftarrow</code>	117
<code>\imath</code>	126	<code>\Leftarrow</code>	116
<code>\immediate\write</code>	180	<code>\lefteqn</code>	51
<code>\in</code>	116	<code>\leftharpoondown</code>	117
<code>\include</code>	165	<code>\leftharpoonup</code>	117
<code>\includegraphics</code>	148	<code>\leftmargin</code>	55, 58
<code>\includeonly</code>	165	<code>\leftmargini</code>	55
<code>\indent</code>	110	<code>\leftmarginii</code>	55

<code>\leftmarginiii</code>	55	<code>\mathtt</code>	18
<code>\leftmarginiv</code>	55	<code>\mathunderscore</code>	127
<code>\leftmarginv</code>	55	<code>\mathversion</code>	18
<code>\leftmarginvi</code>	55	<code>\max</code>	125
<code>\leftrightharrow</code>	117	<code>\mbox</code>	137
<code>\Leftrightharrow</code>	117	<code>\mdseries</code>	17
<code>\leq</code>	117	<code>\medskip</code>	135
<code>\lfloor</code>	117	<code>\medskipamount</code>	135
<code>\lg</code>	125	<code>\medspace</code>	126
<code>\lhd</code>	117	<code>\message</code>	181
<code>\lim</code>	125	<code>\mho</code>	117
<code>\liminf</code>	125	<code>\mid</code>	118
<code>\limsup</code>	125	<code>\min</code>	125
<code>\line</code>	68	<code>\models</code>	118
<code>\linebreak</code>	87	<code>\month</code>	105
<code>\linespread</code>	22	<code>\mp</code>	118
<code>\linethickness</code>	68	<code>\mu</code>	118
<code>\linewidth</code>	24	<code>\multicolumn</code>	79
<code>\listoffigures</code>	168	<code>\multipt</code>	69
<code>\listoftables</code>	168	<code>\nabla</code>	118
<code>\listparindent</code>	55, 59	<code>\name</code>	175
<code>\ll</code>	117	<code>\natural</code>	118
<code>\ln</code>	125	<code>\ne</code>	118
<code>\lnot</code>	117	<code>\narrow</code>	118
<code>\LoadClass</code>	13	<code>\NeedsTeXFormat</code>	13
<code>\LoadClassWithOptions</code>	13	<code>\neg</code>	118
<code>\location</code>	174	<code>\neq</code>	118
<code>\log</code>	125	<code>\newcommand</code>	93
<code>\long</code>	93	<code>\newcounter</code>	96
<code>\longleftarrow</code>	117	<code>\newenvironment</code>	97
<code>\longleftrightarrow</code>	117	<code>\newfont</code>	101
<code>\longmapsto</code>	117	<code>\newlength</code>	97
<code>\longrightarrow</code>	117	<code>\newline</code>	85
<code>\lor</code>	117	<code>\newpage</code>	88
<code>\lq</code>	158	<code>\newsavebox</code>	97
<code>\mainmatter</code>	37	<code>\newtheorem</code>	99
<code>\makebox</code>	137	<code>\newtie</code>	162
<code>\makebox (picture)</code>	67	<code>\newwrite</code>	179
<code>\makeglossary</code>	170	<code>\ng</code>	162
<code>\makeindex</code>	171	<code>\NG</code>	162
<code>\makelabel</code>	57	<code>\ni</code>	118
<code>\makelabels</code>	174	<code>\nocite</code>	82
<code>\mapsto</code>	117	<code>\nofiles</code>	170
<code>\marginpar</code>	110	<code>\noindent</code>	110
<code>\marginparpush</code>	111	<code>\nolinebreak</code>	87
<code>\marginparsep</code>	111	<code>\nonfrenchspacing</code>	133
<code>\marginparwidth</code>	111	<code>\nonumber</code>	51
<code>\mathbf</code>	18	<code>\nopagebreak</code>	89
<code>\mathcal</code>	18	<code>\normalfont</code>	17
<code>\mathdollar</code>	127	<code>\normalmarginpar</code>	111
<code>\mathellipsis</code>	127	<code>\normalsize</code>	19
<code>\mathnormal</code>	18	<code>\not</code>	118
<code>\mathparagraph</code>	127	<code>\notin</code>	118
<code>\mathring</code>	126	<code>\nu</code>	118
<code>\mathrm</code>	17	<code>\nwarrow</code>	118
<code>\mathsection</code>	127	<code>\o (ø)</code>	163
<code>\mathsf</code>	18	<code>\O (Ø)</code>	163
<code>\mathsterling</code>	127	<code>\obeycr</code>	85

<code>\oddsidemargin</code>	8	<code>\ProcessOptions</code>	14
<code>\odot</code>	118	<code>\ProcessOptions*</code>	14
<code>\oe</code> (œ)	163	<code>\prod</code>	119
<code>\OE</code> (Œ)	163	<code>\propto</code>	119
<code>\oint</code>	118	<code>\protect</code>	101
<code>\oldstylenums</code>	18	<code>\protected@write</code>	180
<code>\Omega</code>	118	<code>\providecommand</code>	94
<code>\omega</code>	118	<code>\ProvidesClass</code>	15
<code>\ominus</code>	118	<code>\ProvidesFile</code>	16
<code>\onecolumn</code>	23	<code>\ProvidesPackage</code>	15
<code>\openin</code>	177	<code>\ps</code>	175
<code>\opening</code>	175	<code>\psi</code>	119
<code>\openout</code>	177	<code>\Psi</code>	119
<code>\oplus</code>	118	<code>\pushtabs</code>	73
<code>\OptionNotUsed</code>	14	<code>\put</code>	69
<code>\oslash</code>	118	<code>\P</code>	158
<code>\otimes</code>	118	<code>\qqquad</code>	126
<code>\oval</code>	69	<code>\quad</code>	126
<code>\overbrace{math}</code>	127	<code>\quotedblbase</code> („)	158
<code>\overline{texte}</code>	127	<code>\quotesinglbase</code> (,)	158
<code>\owns</code>	119	<code>\r</code> (ring accent)	161
<code>\PackageError</code>	11	<code>\raggedbottom</code>	24
<code>\PackageInfo</code>	11	<code>\raggedleft</code>	54
<code>\PackageInfoNoLine</code>	11	<code>\raggedright</code>	53
<code>\PackageWarning</code>	11	<code>\raisebox</code>	139
<code>\PackageWarningNoLine</code>	11	<code>\rangle</code>	119
<code>\pagebreak</code>	89	<code>\rbrace</code>	119
<code>\pagenumbering</code>	131	<code>\rbrack</code>	119
<code>\pageref</code>	43	<code>\rceil</code>	119
<code>\paperheight</code>	24	<code>\Re</code>	119
<code>\paperwidth</code>	25	<code>\read</code>	178
<code>\paragraph</code>	29, 35	<code>\ref</code>	44
<code>\parallel</code>	119	<code>\reflectbox</code>	155
<code>\parbox</code>	138	<code>\refname</code>	81
<code>\parindent</code>	64, 110	<code>\refstepcounter</code>	104
<code>\parsep</code>	56, 59	<code>\renewenvironment</code>	97
<code>\parskip</code>	110	<code>\RequirePackage</code>	16
<code>\parskip exemple</code>	56	<code>\RequirePackageWithOptions</code>	16
<code>\part</code>	29, 30	<code>\resizebox</code>	155
<code>\partial</code>	119	<code>\restorecr</code>	85
<code>\partopsep</code>	56, 59	<code>\restriction</code>	119
<code>\PassOptionsToClass</code>	14	<code>\revertset</code>	120
<code>\PassOptionsToPackage</code>	14	<code>\reversemarginpar</code>	111
<code>\pdfpageheight</code>	8	<code>\rfloor</code>	120
<code>\pdfpagewidth</code>	8	<code>\rhd</code>	120
<code>\perp</code>	119	<code>\rho</code>	120
<code>\phi</code>	119	<code>\right</code>	127
<code>\Pi</code>	119	<code>\Rightarrow</code>	120
<code>\pi</code>	119	<code>\rightarrow</code>	120
<code>\pm</code>	119	<code>\rightharpoondown</code>	120
<code>\pmod</code>	125	<code>\rightharpoonup</code>	120
<code>\poptabs</code>	73	<code>\rightleftharpoons</code>	120
<code>\pounds</code>	158	<code>\rightmargin</code>	55, 59
<code>\Pr</code>	125	<code>\rm</code>	18
<code>\prec</code>	119	<code>\rmfamily</code>	17
<code>\preceq</code>	119	<code>\roman</code>	103
<code>\prime</code>	119	<code>\Roman</code>	103
<code>\printindex</code>	171	<code>\rotatebox</code>	154

<code>\rq</code>	158	<code>\succ</code>	121
<code>\rule</code>	163	<code>\succeq</code>	121
<code>\sautdeligne</code>	132	<code>\sum</code>	121
<code>\savebox</code>	139	<code>\sup</code>	125
<code>\sbox</code>	139	<code>\suppressfloats</code>	27
<code>\sc</code>	18	<code>\supset</code>	121
<code>\scalebox</code>	155	<code>\supseteq</code>	121
<code>\scriptsize</code>	19	<code>\surd</code>	121
<code>\scshape</code>	17	<code>\swarrow</code>	121
<code>\searrow</code>	120	<code>\symbol</code>	157
<code>\sec</code>	125	<code>\S</code>	158
<code>\section</code>	29, 33	<code>\t</code> (tie-after accent).....	162
<code>\seename</code>	171	<code>\tabbingsep</code>	74
<code>\selectfont</code>	22	<code>\tabcolsep</code>	79
<code>\setcounter</code>	104	<code>\tableofcontents</code>	168
<code>\setlength</code>	108	<code>\tabulation</code>	132
<code>\setminus</code>	120	<code>\tan</code>	125
<code>\settodepth</code>	108	<code>\tanh</code>	125
<code>\settoheight</code>	108	<code>\tau</code>	121
<code>\settowidth</code>	109	<code>\telephone</code>	176
<code>\sf</code>	18	<code>\textascendercompwordmark</code>	159
<code>\sffamily</code>	17	<code>\textasciicircum</code>	158
<code>\sharp</code>	120	<code>\textasciitilde</code>	158
<code>\shipout</code> et développement.....	180	<code>\textasteriskcentered</code>	158
<code>\shortstack</code>	69	<code>\textbackslash</code>	156, 158
<code>\Sigma</code>	120	<code>\textbar</code>	158
<code>\sigma</code>	120	<code>\textbardbl</code>	158
<code>\signature</code>	175	<code>\textbf</code>	17
<code>\sim</code>	120	<code>\textbigcircle</code>	158
<code>\simeq</code>	120	<code>\textbraceleft</code>	158
<code>\sin</code>	125	<code>\textbraceright</code>	159
<code>\sinh</code>	125	<code>\textbullet</code>	159
<code>\sl</code>	18	<code>\textcapitalcompwordmark</code>	159
<code>\slshape</code>	17	<code>\textcircled{lettre}</code>	159
<code>\small</code>	19	<code>\textcompwordmark</code>	159
<code>\smallint</code>	120	<code>\textcopyright</code>	157
<code>\smallskip</code>	135	<code>\textdagger</code>	159
<code>\smallskipamount</code>	135	<code>\textdaggerdbl</code>	159
<code>\smile</code>	120	<code>\textdollar</code> (ou <code>\\$</code>).....	159
<code>\spadesuit</code>	121	<code>\textellipsis</code>	158
<code>\sqcap</code>	121	<code>\textemdash</code> (ou <code>---</code>).....	159
<code>\sqcup</code>	121	<code>\textendash</code> (ou <code>--</code>).....	159
<code>\sqrt[nième]{arg}</code>	127	<code>\texteuro</code>	159
<code>\sqsubset</code>	121	<code>\textexclamdown</code> (ou <code>!'</code>).....	159
<code>\sqsubseteq</code>	121	<code>\textfloatsep</code>	27
<code>\sqsupset</code>	121	<code>\textfraction</code>	27
<code>\sqsupseteq</code>	121	<code>\textgreater</code>	159
<code>\ss</code> (β).....	163	<code>\textheight</code>	24
<code>\SS</code> (SS).....	163	<code>\textit</code>	17
<code>\stackrel{texte}{relation}</code>	128	<code>\textleftarrow</code>	159
<code>\star</code>	121	<code>\textless</code>	159
<code>\stepcounter</code>	104	<code>\textmd</code>	17
<code>\stop</code>	187	<code>\textnormal</code>	17
<code>\subparagraph</code>	29, 35	<code>\textordfeminine</code>	159
<code>\subsection</code>	29, 34	<code>\textordmasculine</code>	159
<code>\subset</code>	121	<code>\textparagraph</code>	158
<code>\subseteq</code>	121	<code>\textperiodcentered</code>	159
<code>\subsetneq</code>	121	<code>\textquestiondown</code> (ou <code>?'</code>).....	159
<code>\subsubsection</code>	29, 35		

<code>\textquotedblleft</code> (ou ‘ ‘)	160	<code>\unlhd</code>	122
<code>\textquotedblright</code> (ou ’ ’)	160	<code>\unrhd</code>	122
<code>\textquoteleft</code> (ou ‘)	160	<code>\uparrow</code>	122
<code>\textquoteright</code> (ou ’)	160	<code>\Uparrow</code>	122
<code>\textquotesingle</code>	160	<code>\updownarrow</code>	122
<code>\textquotestraightbase</code>	160	<code>\Updownarrow</code>	122
<code>\textquotestraightdblbase</code>	160	<code>\upharpoonright</code>	122
<code>\textregistered</code>	160	<code>\uplus</code>	123
<code>\textrightarrow</code>	160	<code>\upshape</code>	17
<code>\textrm</code>	17	<code>\Upsilon</code>	123
<code>\textsc</code>	17	<code>\upsilon</code>	123
<code>\textsf</code>	17	<code>\usebox</code>	139
<code>\textsl</code>	17	<code>\usecounter</code>	103
<code>\textsterling</code>	158	<code>\usefont</code>	22
<code>\textthreequartersemdash</code>	160	<code>\usepackage</code>	9
<code>\texttrademark</code>	160	<code>\v</code> (breve accent)	162
<code>\texttt</code>	17	<code>\value</code>	104
<code>\texttwelveudash</code>	160	<code>\vanotthing</code>	123
<code>\textunderscore</code>	160	<code>\varepsilon</code>	123
<code>\textup</code>	17	<code>\varphi</code>	123
<code>\textvisiblespace</code>	160	<code>\varpi</code>	123
<code>\textwidth</code>	25	<code>\varrho</code>	123
<code>\TeX</code>	158	<code>\varsigma</code>	123
<code>\th</code> (p)	163	<code>\vartheta</code>	123
<code>\TH</code> (P)	163	<code>\vbox</code> T _E X de base	63
<code>\theta</code>	121	<code>\vdash @math@xeatspaces @backslashchar</code> <code>\vdash@comment</code> Taquet droit ; prouve, <code>implique ;</code>	123
<code>\thicklines</code>	69	<code>\vdots</code>	128
<code>\thickspace</code>	126	<code>\vec</code>	126
<code>\thinlines</code>	69	<code>\vector</code>	70
<code>\thinspace</code>	126, 133	<code>\vee</code>	123
<code>\thispagestyle</code>	131	<code>\verb</code>	83
<code>\tilde</code>	126	<code>\vert</code>	124
<code>\times</code>	122	<code>\Vert</code>	123
<code>\tiny</code>	19	<code>\vfill</code>	135
<code>\to</code>	122	<code>\vline</code>	81
<code>\today</code>	163	<code>\vspace</code>	135
<code>\top</code>	122	<code>\vtop</code> T _E X de base	63
<code>\topfraction</code>	27	<code>\wedge</code>	124
<code>\topmargin</code>	25	<code>\widehat</code>	126
<code>\topnumber</code>	28	<code>\widetilde</code>	126
<code>\topsep</code>	56, 59	<code>\width</code>	109
<code>\topskip</code>	25	<code>\wlog</code>	181
<code>\totalheight</code>	109	<code>\wp</code>	124
<code>\totalnumber</code>	28	<code>\wr</code>	124
<code>\triangle</code>	122	<code>\write</code>	179
<code>\triangleleft</code>	122	<code>\write18</code>	182
<code>\triangleright</code>	122	<code>\Xi</code>	124
<code>\tt</code>	18	<code>\xi</code>	124
<code>\ttfamily</code>	17	<code>\year</code>	105
<code>\twocolumn</code>	23	<code>\zeta</code>	124
<code>\typein</code>	178		
<code>\typeout</code>	178		
<code>\u</code> (breve accent)	162		
<code>\unboldmath</code>	112		
<code>\underbar</code>	162		
<code>\underbrace{math}</code>	128		
<code>\underline{texte}</code>	128		
<code>\unitlength</code>	65		
		{	
		{...}	pour les arguments obligatoires 4

A

`abstract`, environnement 45
`algorithm2e`, paquetage 74
`amsmath`, paquetage 46, 48
`answers`, paquetage 180
`appendix`, paquetage 37
`array`, environnement 46
`Asymptote`, paquetage 68, 182

B

`babel`, paquetage 32, 81, 169, 171
`beamer`, paquetage 188
`bp` 107

C

`cc` 108
`center`, environnement 47
classe `article` 7
classe `book` 7
classe `letter` 7
classe `report` 7
classe `slides` 7
`cleveref`, paquetage 42
`cm` 107
`cmd.exe`, utilisé par `\write18` 183
commande `dvipdfmx` 3
commande `dvips` 3
commande `dvitype` 3
commande `latex` 3
commande `pdflatex` 3
commande `xdvi` 3

D

`datatool`, paquetage 178
`datetime`, paquetage 163
`dd` 108
`description`, environnement 47
`displaymath`, environnement 48, 112
`document`, environnement 49

E

`em` 108
`enumerate`, environnement 49
`enumitem`, paquetage 55, 61
environnement `abstract` 45
environnement `array` 46
environnement `center` 47
environnement `description` 47
environnement `displaymath` 48, 112
environnement `document` 49
environnement `enumerate` 49
environnement `eqnarray` 51
environnement `equation` 51, 112

environnement `figure` 51
environnement `filecontents` 52
environnement `filecontents*` 52
environnement `flushleft` 53
environnement `flushright` 54
environnement `itemize` 54
environnement `letter` 56
environnement `list` 56
environnement `math` 62, 112
environnement `minipage` 63
environnement `picture` 65
environnement `quotation` 70
environnement `quote` 70
environnement `tabbing` 70
environnement `table` 75
environnement `tabular` 76
environnement `thebibliography` 81
environnement `theorem` 83
environnement `titlepage` 83
environnement `verbatim` 83
environnement `verse` 84
`eqnarray`, environnement 51
`equation`, environnement 51, 112
`etex`, commande 3
`etoolbox`, paquetage 12
`ex` 108
`expl3`, paquetage 157

F

fichier `.dvi` 3
fichier `.log` 3
`figure`, environnement 51
`filecontents*`, environnement 52
`filecontents`, environnement 52
`first-latex-doc` document 1
`flafter`, paquetage 27
`float`, paquetage 26
`flushleft`, environnement 53
`flushright`, environnement 54
`footmisc`, paquetage 92

G

`geometry`, paquetage 8

H

<https://latexref.xyz> page d'accueil 1
`hyperref`, paquetage 186

I

`in` 107
`indentfirst`, paquetage 31, 32, 33, 35, 36
`itemize`, environnement 54

L

latex-doc-ptr document	1
latexrefman@tug.org adresse mél.	1
letter, environnement	56
list, environnement	56
listings, paquetage	74
lR box	66
lrbox	137
lshort document	1
lualatex commande	4

M

macros2e, paquetage	95
makeidx, paquetage	171
math, environnement	62, 112
MetaPost, paquetage	68
mfirstuc, paquetage	157
minipage, environnement	63
mm	107
mu	108
multind, paquetage	171

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 oneside	8
option openany	8
option openbib	8
option openright	8
option titlepage	8
option twocolumn	8
option twoside	8

P

paquetage algorithm2e	74
paquetage amsmath	46, 48
paquetage answers	180
paquetage appendix	37
paquetage Asymptote	68, 182
paquetage babel	32, 81, 169, 171
paquetage beamer	188
paquetage cleveref	42
paquetage datatool	178
paquetage datetime	163
paquetage enumitem	55, 61
paquetage etoolbox	12
paquetage expl3	157
paquetage flafter	27
paquetage float	26
paquetage footmisc	92
paquetage geometry	8
paquetage hyperref	186
paquetage indentfirst	31, 32, 33, 35, 36
paquetage listings	74
paquetage macros2e	95
paquetage makeidx	171
paquetage MetaPost	68
paquetage mfirstuc	157
paquetage multind	171
paquetage pict2e	68
paquetage picture	66
paquetage polyglossia	169, 171
paquetage PSTricks	68
paquetage setspace	21
paquetage showidx	171
paquetage siunitx	65
paquetage soul	75
paquetage texosquery	183
paquetage textcase	157
paquetage textcomp	18
paquetage TikZ	68
paquetage titlesec	29, 31, 32, 34, 35, 36
paquetage tocbibbind	169
paquetage tocloft	169
paquetage xspace	133
pc	107
pict2e, paquetage	68
picture, environnement	65
picture, paquetage	66
polyglossia, paquetage	169, 171
pouce (inch)	107
PSTricks, paquetage	68
pt	107

Q

quotation, environnement	70
quote, environnement	70

S

<code>secnumdepth</code>	30
<code>secnumdepth</code> compteur.....	30
<code>setspace</code> , paquetage.....	21
<code>sh</code> , utilisé par <code>\write18</code>	183
<code>SHELL</code> , environnement, variables de.....	183
<code>SHELL</code> , variables d'environnement.....	183
<code>showidx</code> , paquetage.....	171
<code>siunitx</code> , paquetage.....	65
<code>soul</code> , paquetage.....	75
<code>sp</code>	108

T

<code>tabbing</code> , environnement.....	70
<code>table</code> , environnement.....	75
<code>tabular</code> , environnement.....	76
<code>texosquery</code> , paquetage.....	183
<code>textcase</code> , paquetage.....	157
<code>textcomp</code> , paquetage.....	18, 157
<code>thebibliography</code> , environnement.....	81
<code>theorem</code> , environnement.....	83

<code>TikZ</code> , paquetage.....	68
<code>titlepage</code> , environnement.....	83
<code>titlesec</code> , paquetage.....	29, 31, 32, 34, 35, 36
<code>tocbibbind</code> , paquetage.....	169
<code>tocdepth</code>	30
<code>tocdepth</code> counter.....	30
<code>tocloft</code> , paquetage.....	169

U

<code>usrguide</code> documentation officielle.....	1
---	---

V

<code>verbatim</code> , environnement.....	83
<code>verse</code> , environnement.....	84

X

<code>xdvipdfmx</code>	4
<code>xelatex</code> commande.....	4
<code>xspace</code> , paquetage.....	133