

L^AT_EX2_ε: Un manuel de référence officiel

Août 2017

<http://puszcza.gnu.org.ua/software/latexrefman/>

Ce document est un manuel de référence officiel pour L^AT_EX, un système de préparation de documents, version « Août 2017 ».

Ce manuel a été traduit du fichier L^AT_EX.HLP v1.0a de la bibliothèque d'aide VMS. La version pré-translation a été rédigé par George D. Greenwade de Sam Houston State University. La version L^AT_EX 2.09 a été rédigée par Stephen Gilmore. La version L^AT_EX2e a été adapté 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 L^AT_EX*, de Sheldon Green, et *L^AT_EX Command Summary* (pour L^AT_EX2.09) de L. Botway et C. Biemesderfer (publié par le T_EX Users Group en tant que *T_EXniques* numéro 10), il l'a utilisé en tant que matériel de référence (aucun texte n'était directement copié).

Tous droits réservés © 2015-2017 Vincent Belaïche — traduction.

Tous droits réservés © 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017 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 132.

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

L ^A T _E X2e : un manuel de référence officieux	1
1 À propos de ce document	2
2 Vue d'ensemble de L ^A T _E X	3
3 Classes de documents	12
4 Polices de caractères	22
5 Mise en page	26
6 Des rubriques	32
7 Des références croisées	33
8 Environments	36
9 Saut à la ligne	63
10 Saut de page	66
11 Note en bas de page	67
12 Définitions	70
13 Compteurs	78
14 Longueurs	80
15 Faire des paragraphes	82
16 Formules mathématiques	84
17 Modes	101
18 Styles de page	103
19 Espaces	104
20 Boîtes	109
21 Insertions spéciales	112
22 Partitionner le fichier d'entrée	119
23 Prologue et épilogue	120
24 Des lettres	122
25 Entrée/sortie console	127
26 Ligne de commande	128
A Patrons de documents	129
B Traduction de la licence	132
Index des concepts	133
Index des commandes	140

Table des matières

LaTeX2e : un manuel de référence officieux	1
1 À propos de ce document	2
2 Vue d'ensemble de LaTeX	3
2.1 Début et fin	3
2.2 Fichiers de sortie	4
2.3 Les moteurs TeX	4
2.4 Syntaxe des commandes LaTeX	5
2.4.1 Les environnements	6
2.4.2 Les déclarations de commandes	6
2.4.3 <code>\makeatletter</code> et <code>\makeatother</code>	6
2.4.3.1 <code>\@startsection</code>	7
2.4.3.2 <code>\@ifstar</code>	10
3 Classes de documents	12
3.1 Options de classe de document	12
3.2 Ajout de paquetages	13
3.3 Construction des extensions (classes et paquetages)	13
3.3.1 Structure d'une extension (classe ou paquetage)	14
3.3.2 Commande pour extension (classe ou paquetage)	15
4 Polices de caractères	22
4.1 styles des polices	22
4.2 tailles des polices	24
4.3 commandes de fontes de bas niveau	24
5 Mise en page	26
5.1 <code>\onecolumn</code>	26
5.2 <code>\twocolumn</code>	26
5.3 <code>\flushbottom</code>	27
5.4 <code>\raggedbottom</code>	27
5.5 Paramètres de mise en page	27
5.6 Flottants	28
6 Des rubriques	32
7 Des références croisées	33
7.1 <code>\label</code>	33
7.2 <code>\pageref{clef}</code>	34
7.3 <code>\ref{clef}</code>	34

8	Environments	36
8.1	abstract	36
8.2	array	37
8.3	center	38
8.3.1	\centering	38
8.4	description	38
8.5	displaymath	39
8.6	document	39
8.7	enumerate	40
8.8	eqnarray	41
8.9	equation	42
8.10	figure	42
8.11	filecontents: Écrire un fichier externe	43
8.12	flushleft	44
8.12.1	\raggedright	44
8.13	flushright	44
8.13.1	\raggedleft	44
8.14	itemize	44
8.15	environnement letter: écrire des lettres postales	46
8.16	list	46
8.17	math	47
8.18	minipage	48
8.19	picture	48
8.19.1	\circle	49
8.19.2	\makebox	49
8.19.3	\framebox	50
8.19.4	\dashbox	50
8.19.5	\frame	50
8.19.6	\line	50
8.19.7	\linethickness	51
8.19.8	\thicklines	51
8.19.9	\thinlines	51
8.19.10	\multipt	51
8.19.11	\oval	51
8.19.12	\put	51
8.19.13	\shortstack	52
8.19.14	\vector	52
8.20	quotation et quote	52
8.21	tabbing	53
8.22	table	54
8.23	tabular	55
8.23.1	\multicolumn	58
8.23.2	\cline	59
8.23.3	\hline	59
8.23.4	\vline	59
8.24	thebibliography	60
8.24.1	\bibitem	60
8.24.2	\cite	60

8.24.3	<code>\nocite</code>	60
8.24.4	Utilisation de BibTeX.....	61
8.25	<code>theorem</code>	61
8.26	<code>titlepage</code>	61
8.27	<code>verbatim</code>	62
8.27.1	<code>\verb</code>	62
8.28	<code>verse</code>	62
9	Saut à la ligne	63
9.1	<code>\\</code>	63
9.2	<code>\obeycr</code> & <code>\restorecr</code>	63
9.3	<code>\newline</code>	63
9.4	<code>\-</code> (césure à gré).....	64
9.5	<code>\discretionary</code> (point de césure généralisé).....	64
9.6	<code>\fussy</code>	65
9.7	<code>\sloppy</code>	65
9.8	<code>\hyphenation</code>	65
9.9	<code>\linebreak</code> & <code>\nolinebreak</code>	65
10	Saut de page	66
10.1	<code>\cleardoublepage</code>	66
10.2	<code>\clearpage</code>	66
10.3	<code>\newpage</code>	66
10.4	<code>\enlargethispage</code>	66
10.5	<code>\pagebreak</code> & <code>\nopagebreak</code>	66
11	Note en bas de page	67
11.1	<code>\footnote</code>	67
11.2	<code>\footnotemark</code>	67
11.3	<code>\footnotetext</code>	68
11.4	Notes en bas de page dans un tableau.....	68
11.5	Note en bas de page dont le renvoi est au sein d'un titre de rubrique.....	69
11.6	Paramètres des notes en bas de page.....	69
12	Définitions	70
12.1	<code>\newcommand</code> & <code>\renewcommand</code>	70
12.2	<code>\newcounter</code>	71
12.3	<code>\newlength</code>	71
12.4	<code>\newsavebox</code>	72
12.5	<code>\newenvironment</code> & <code>\renewenvironment</code>	72
12.6	<code>\newtheorem</code>	74
12.7	<code>\newfont</code> : définit une nouvelle police (obsolète).....	76
12.8	<code>\protect</code>	76

13	Compteurs	78
13.1	<code>\alph \Alph \arabic \roman \Roman</code> <code>\fnsymbol</code> : Taper des compteurs	78
13.2	<code>\usecounter{compteur}</code>	78
13.3	<code>\value{compteur}</code>	79
13.4	<code>\setcounter{compteur}{value}</code>	79
13.5	<code>\addtocounter{compteur}{valeur}</code>	79
13.6	<code>\refstepcounter{compteur}</code>	79
13.7	<code>\stepcounter{compteur}</code>	79
13.8	<code>\day \month \year</code> : Predefined compteurs	79
14	Longueurs	80
14.1	Unités de longueur	80
14.2	<code>\setlength</code>	80
14.3	<code>\addtolength</code>	81
14.4	<code>\settodepth</code>	81
14.5	<code>\settoheight</code>	81
14.6	<code>\settowidth</code>	81
14.7	Longueurs prédéfinies	81
15	Faire des paragraphes	82
15.1	<code>\indent</code>	82
15.2	<code>\noindent</code>	82
15.3	<code>\parskip</code>	82
15.4	Notes en marge	82
16	Formules mathématiques	84
16.1	Indices inférieurs & supérieurs	84
16.2	Symboles mathématiques	85
16.3	Fonctions mathématiques	96
16.4	Accents mathématiques	97
16.5	Espacement en mode mathématique	98
16.6	Recueil de diverses commandes utiles en math	98
17	Modes	101
17.1	<code>\ensuremath</code>	101
18	Styles de page	103
18.1	<code>\maketitle</code>	103
18.2	<code>\pagenumbering</code>	103
18.3	<code>\pagestyle</code>	103
18.4	<code>\thispagestyle{style}</code>	103

19	Espaces	104
19.1	<code>\hspace</code>	104
19.2	<code>\hfill</code>	104
19.3	<code>\<espace></code> et <code>\@</code>	104
19.4	<code>\</code> après une séquence de contrôle	105
19.5	<code>\frenchspacing</code> , rendre égaux les espacements inter-phrase et inter-mot	105
19.6	<code>\thinspace</code>	105
19.7	<code>\/</code> : insérer une correction italique	105
19.8	<code>\hrulefill</code> <code>\dotfill</code>	106
19.9	<code>\addvspace</code>	106
19.10	<code>\bigskip</code> <code>\medskip</code> <code>\smallskip</code>	107
19.11	<code>\vfill</code>	107
19.12	<code>\vspace</code>	107
20	Boîtes	109
20.1	<code>\mbox{texte}</code>	109
20.2	<code>\fbox</code> and <code>\framebox</code>	109
20.3	<code>lrbox</code>	109
20.4	<code>\makebox</code>	109
20.5	<code>\parbox</code>	110
20.6	<code>\raisebox</code>	110
20.7	<code>\savebox</code>	111
20.8	<code>\sbox{<boxcmd>}{texte}</code>	111
20.9	<code>\usebox{<boxcmd>}</code>	111
21	Insertions spéciales	112
21.1	Caractères réservés	112
21.2	Symboles appelés par leur position dans une police	112
21.3	Symboles en texte	112
21.4	Accents	115
21.5	Lettres latines supplémentaires	117
21.6	<code>\rule</code>	118
21.7	<code>\today</code>	118
22	Partitionner le fichier d'entrée	119
22.1	<code>\include</code>	119
22.2	<code>\includeonly</code>	119
22.3	<code>\input</code>	119
23	Prologue et épilogue	120
23.1	Tables des matières	120
23.1.1	<code>\addcontentsline</code>	120
23.1.2	<code>\addtocontents</code>	120
23.2	Glossaires	121
23.3	Index	121

24	Des lettres	122
24.1	<code>\address</code>	123
24.2	<code>\cc</code>	123
24.3	<code>\closing</code>	124
24.4	<code>\encl</code>	124
24.5	<code>\location</code>	124
24.6	<code>\makelabels</code>	124
24.7	<code>\name</code>	125
24.8	<code>\opening{texte}</code>	125
24.9	<code>\ps</code>	125
24.10	<code>\signature{texte}</code>	125
24.11	<code>\telephone</code>	126
25	Entrée/sortie console	127
25.1	<code>\typein[cmd]{msg}</code>	127
25.2	<code>\typeout{msg}</code>	127
26	Ligne de commande	128
Annexe A	Patrons de documents	129
A.1	Un patron <code>beamer</code>	129
A.2	Un patron <code>book</code>	129
A.3	Un patron <code>tugboat</code>	130
Annexe B	Traduction de la licence	132
Index des concepts	133
Index des commandes	140

L^AT_EX2e : un manuel de référence officiel

Ce document est un manuel de référence officiel pour L^AT_EX2e, un système de préparation de document, dans sa version « Août 2017 ».

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 oeuvre sous forme d'un paquet de macros pour le programme de composition \TeX (voir Chapitre 2 [Overview], page 3). La page d'accueil de ce document est <http://puszcza.gnu.org.ua/software/latexrefman/>. Cette page contient des liens vers les produits courants de fabrication du document dans différents formats, les sources, les listes de diffusion, et 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. Au lieu de cela, s'il vous plaît envoyez tous commentaires à latexrefman@tug.org.

Ce document est un manuel de référence. Il y a une vaste gamme d'autres sources d'information sur \LaTeX , 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 \LaTeX

Qu'est-ce que \LaTeX ?

\LaTeX est un système de composition de document. Il fut à l'origine créé par Leslie Lamport et 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.

Un utilisateur \LaTeX écrit un fichier d'entrée contenant le texte d'un document avec des commandes qui y sont intercalées pour décrire comment le texte doit être formaté. \LaTeX est mis en œuvre comme un ensemble de commandes liées s'interfaçant avec le programme de composition \TeX de Donald E. Knuth (le terme technique est que \LaTeX est un *paquet de macros* pour le moteur \TeX). L'utilisateur produit le document de sortie en donnant ce fichier d'entrée au moteur \TeX .

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

Le nom \LaTeX est l'abréviation de "Lamport \TeX ". 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 \LaTeX ont une structure globale simple, avec un début et une fin standards. Voici un exemple « Bonjour le monde » :

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

Ici 'article' et ce qu'on appelle la *classe de document*, implémentée dans un fichier `article.cls`. N'importe quelle classe de document peut être utilisée. Quelques classes de document sont définies par \LaTeX lui-même, et un grand nombre d'autres sont largement disponibles. Voir Chapitre 3 [Document classes], page 12.

Vous pouvez inclure d'autres commandes \LaTeX 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 \LaTeX (voir Section 8.6 [document], page 39). \LaTeX fournit lui-même beaucoup d'environnements, et bien plus encore sont définis séparément. Voir Chapitre 8 [Environnements], page 36.

Les rubriques suivantes discutent de la façon de produire des PDF et d'autres formats de sortie à partir d'un fichier d'entrée \LaTeX .

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` Si 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 DVI sont disponibles (<http://mirror.ctan.org/dviware>).
- `.pdf` Si L^AT_EX est invoqué avec la commande système `pdflatex`, parmi d'autres commandes (voir Section 2.3 [T_EX engines], page 4), 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.

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 commande qui sont exécutées par une implémentation T_EX (voir Chapitre 2 [Overview], page 3). Cette section donne une vue d'ensemble laconique 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 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 UTF-8 Unicode et les polices TrueType et OpenType, bien que l'implémentation soit complètement différente, 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 ex. pour fournir une prise en charge supplémentaires du japonais des 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 contr'oblique, `\`. Le nom lui-même consiste soit en (a) une chaîne de lettres ou (b) une unique non-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 en tout cas vous devez saisir toutes les commandes dans la même casse où elles sont 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 a la convention que certaines commandes ont une forme en `a *` 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 accepté 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 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}
  There once was a man from Nantucket \\\
...
\end{verse}
```

Voir Chapitre 8 [Environnements], page 36, pour une liste des environnements.

Le *nom environnement* au début doit s'accorder exactement avec celui à la fin. Ceci comprend 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 (et implique de la table est alignée verticalement sur sa première ligne) alors que le second argument est obligatoire (il spécifie le format des colonnes).

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

2.4.2 Les déclarations de commandes

Une commande qui change la valeur, ou change la signification, d'une autre commande ou paramètre. Par exemple, la commande `\mainmatter` change le réglage de la numérotation des pages en passant de numéros romains à des numéros arabes.

2.4.3 `\makeatletter` et `\makeatother`

Synopsis :

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

Utilisé pour redéfinir des commandes internes de L^AT_EX. `\makeatletter` a pour effet que le caractère arobre `@` ait le code de catégorie des lettres, c.-à-d. 11. `\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 contr'oblique `\` reçoit le catcode 0, qui correspond aux caractères qui commencent une commande. Ces deux commandes altèrent le catcode assigné à `@`.

Cette altération est nécessaire parce que beaucoup des commandes de L^AT_EX utilisent `@` dans leur nom, de sorte à empêcher les utilisateurs de définir accidentellement une commande qui remplacerait l'une des commandes privées de L^AT_EX. Les noms de commandes consistent en un caractère de catégorie 0, d'ordinaire une contr'oblique, suivi de lettres, c.-à-d. des caractères de catégorie 11 (à ceci près qu'un nom de commande peut aussi consister d'un caractère de catégorie 0 suivi d'un seul symbole non-lettre). Ainsi sous le régime par défaut de codes de catégorie, les commandes définies par l'utilisateur ne peuvent pas contenir de `@`. Mais `\makeatletter` et `\makeatother` permettent aux utilisateurs de définir et re-définir des commandes dont le nom comprend un `@`.

À utiliser dans un fichier `.tex`, dans le préambule. Ne pas utiliser dans des fichiers `.sty` ou `.cls` puisque les commandes `\usepackage` et `\documentclass` règlent le code de catégorie de l'arobe à celui d'une lettre.

Pour une liste complète des macros contenant un arobre dans leur nom, voir le document <http://ctan.org/pkg/macros2e>. Ces macros sont surtout à l'intention des auteurs de classes et de paquetages.

L'exemple ci-après est typique. Une commande `\these@nomuniversite` se trouve dans le fichier de classe de l'utilisateur. 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
```

2.4.3.1 `\@startsection`

Synopsis :

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

Cette commande permet de redéfinir les 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 toutes. 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 :

```
\@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}`. Voir Chapitre 6 [Sectioning], page 32, et les exemples plus bas.

nom Nom du compteur (qui doit être défini séparément) utilisé pour numéroter le titre de rubrique. Ceux le plus communément utilisés sont `section`, `subsection`, ou `paragraph`. Bien que dans ces trois cas le nom du compteur soit le même que celui de la commande elle-même, utiliser le même nom n'est pas exigé.

Alors `\thename` affiche le numéro de titre, et `\namemark` sert aux en-têtes de page.

niveau Un entier donnant la profondeur de la commande de rubricage, par ex. 0 pour `chapter` (ceci ne s'applique qu'aux classes standards `book` et `report`), 1 pour `section`, 2 pour `subsection`, 3 pour `subsubsection`, 4 pour `paragraph` et 5 pour `subparagraph`. Dans les classes `book` et `report`, `part` a pour niveau -1, alors que dans la classe `article`, `part` a pour niveau 0.

Si *niveau* est inférieur ou égal à la valeur de `secnumdepth` alors les titres pour cette commande de rubricage sont numérotés. Par ex. dans un `article`, si `secnumdepth` vaut 1 alors une commande `\section{Introduction}` produira en sortie quelque chose du genre de 1 Introduction alors que `\subsection{Historique}` produit en sortie quelque chose du genre de Historique, sans numéro en préfixe. Voir [Sectioning/secnumdepth], page 32.

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

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

avant Longueur dont la valeur absolue est la longueur de l'espace vertical à insérer avant le titre de la rubrique. Cet espacement est ignoré si la rubrique commence au début d'une page. Si en plus *avant* est négatif, alors le premier paragraphe suivant immédiatement 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.

La quantité totale d'espace vertical entre la ligne de base de la ligne précédant le titre de cette rubrique est la ligne de base du titre 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 et peut 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 elle est positive ou nulle, 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 alors 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 le suivant, ou fasse 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 des exemples. Pour les utiliser mettez les soit au sein d'un fichier de paquetage ou de classe, ou placez les dans le préambule d'un document L^AT_EX entre des commandes `\makeatletter` et `\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 2.4.3 [`\makeatletter` and `\makeatother`], page 6.

L'exemple suivant redéfinit la commande de rubricage `\section` pour qu'elle centre le titre et le mette en gras :

```
\renewcommand\section{%
  \@startsection{section}% [nom], page 8.
  {1}% [niveau], page 8.
  {0pt}% [retrait], page 8.
  {-3.5ex plus -1ex minus -.2ex}% [avant], page 8.
  {2.3ex plus.2ex}% [après], page 9.
  {\centering\normalfont\Large\bfseries}}% [style], page 9.
```

Cet exemple met les titres de `subsection` en petites capitales, et leur fait faire corps avec le paragraphe suivant.

```
\renewcommand\subsection{%
  \@startsection{subsection}% [nom], page 8.
  {2}% [niveau], page 8.
  {0em}% [retrait], page 8.
  {-1ex plus 0.1ex minus -0.05ex}% [avant], page 8.
  {-1em plus 0.2em}% [après], page 9.
  {\scshape}% [style], page 9.
}
```

Les exemples précédents redéfinissait les commandes de titre de rubriques existantes. celui-ci en définit une nouvelle, illustrant la nécessité d'un compteur et de macros pour son affichage.

```
\setcounter{secnumdepth}{6}% affiche les compteur 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
```

2.4.3.2 \@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 12.5 [`\newenvironment` & `\renewenvironment`], page 72, 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 2.4.3 [`\makeatletter` and `\makeatother`], page 6, » 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 et sa variante étoilée `\agentsecret*` 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 : contrairement aux noms d'environnement où l'étoile fait partie du nom lui-même, et pourrait donc être à n'importe quelle position, dans le nom d'une commande l'étoile est comme une sorte d'argument optionnel. 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.

3 Classes de documents

La classe d'un document donné est définie avec la commande:

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

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

Les noms de *classe* \LaTeX encadrés sont (beaucoup d'autres classes de documents sont disponibles en modules ; voir Chapitre 2 [Overview], page 3):

```
article report book letter slides
```

Les *options* standardes sont décrites ci-dessous.

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 $\text{pdf}\LaTeX$, $\text{Lua}\LaTeX$, ou $\text{Xe}\LaTeX$ (voir Section 2.3 [TeX engines], page 4), les options autres que `letterpaper` règlent la zone d'impression mais vous devez également régler la taille du papier physique. Un 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 pour paquetage on peu 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 2.4.3 [`\makeatletter` and `\makeatother`], page 6. 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 L^AT_EX termine le traitement de la classe ou du paquetage courants. On peut utiliser ces crochet plusieurs fois ; le *code* sera exécuté dans l’ordre d’appel. Voir aussi `\undefined` [`\AtBeginDocument`], page `\undefined`.

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

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

Similaire à `\newcommand` (voir Section 12.1 [`\newcommand` & `\renewcommand`], page 70) 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}`

Porduit 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 T_EX. 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 L^AT_EX ajoute un point final à chaque message.

`\CurrentOption`

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

`\DeclareOption{option}{code}`

`\DeclareOption*{option}{code}`

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

Si vous invoquez une option qui n'a pas été déclarée, par défaut cela produit un avertissement semblable à `Unused global option(s): [badoption]`. Vous pouvez changer ce comportement avec la version étoilée `\DeclareOption*{code}`. Par exemple, beaucoup de classes étendent une classe existante en utilisant une 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 classe `notecms` permette aux utilisateurs de tenir des listes de destinataires des notes dans des fichiers externes. Par exemple l'utilisateur invoque `\documentclass[math]{notecms}` et la classe lit le fichier `math.memo`. Ce code gère le fichier s'il existe et sinon passe l'option à la classe `article`.

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

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

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

Similaire à `\newcommand` et `\newcommand*` (voir Section 12.1 [`\newcommand` & `\renewcommand`], page 70) mais déclare une commande robuste, même si *définition* comprend du code fragile. (Pour une discussion sur les commandes robustes et fragiles voir Section 12.8 [`\protect`], page 76). Utilisez ces commandes pour définir de nouvelles commandes robustes, ou redéfinir des commandes existantes en les rendant robustes. Contrairement à `\newcommand` elle ne produit

pas d'ereru 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*`, `\providcommand`, et `\providcommand*`, mais définissent une commande *cmd* robuste avec deux avances 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 `\provid...` 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 `\provid...` ou il y a une erreur.

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

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

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

Cette commande cherche le fichier dans tous les 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 22.3 [input], page 119.

```
\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 associé à 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 variant 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`, L^AT_EX 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 antérieure à *date de parution*, en utilisant les arguments optionnels comme dans `\documentclass{smcmem}[2018/10/12]` ou `\usepackage{toto}[[2017/07/07]]`. (Notez que les utilisateurs de paquetages incluent seulement rarement une date, et les utilisateurs de classe presque jamais).

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

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

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

Charge un paquetage, comme la commande `\usepackage` pour les auteurs de documents. Voir Section 3.2 [Additional packages], page 13. Voici un exemple : `\RequirePackage[landscape,margin=1in]{geometry}`. Notez que l'équipe de développement de L^AT_EX recommande fortement l'utilisation de ces commande de préférence à l'`\input` de Plain T_EX ; 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> 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>	Machine à écrire, 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>	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 38. Les deux ensembles de commandes ont leur place.

4.2 tailles des polices

Les commandes de type de taille standard suivants 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 12).

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 “forme déclaratives”. La portée d’une forme déclarative s’étend jusqu’à la prochaine la commande de type style ou la fin du groupe courant. Vous pouvez également utiliser la “forme d’environnement” de ces commandes ; par exemple, `\begin{tiny} ... \end{tiny}`.

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}`

Sélectionnez le codage de police. Les codages valides comprennent OT1 et T1.

`\fontfamily{famille}`

Sélectionnez la famille de polices. Familles valides sont :

- `cmr` pour Computer Modern Roman
- `cmss` pour Computer Modern Sans Serif
- `cmtt` pour Computer Modern Typewriter

et de nombreuses autres.

`\fontseries{série}`

Sélectionnez série de police. Les séries valides sont :

- `m` Moyen (normal)
- `b` Gras
- `c` condensé
- `bc` condensé Gras
- `bx` Gras étendu

et diverses autres combinaisons.

`\fontshape{forme}`

Sélectionnez 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.

`\fontsize{taille}{interligne}`

Réglage de la taille de police. Le premier paramètre est la taille de la police pour basculer et le deuxième est l'espacement d'interligne ; ceci est stocké dans un paramètre nommé `\baselineskip`. L'unité des deux paramètres est par défaut le `pt`. La valeur par défaut de `\baselineskip` pour la police Computer Modern est 1,2 fois le `\fontsize`.

L'espacement des lignes est également multiplié par la valeur du paramètre `\baselinestretch` en cas de changement de taille de type ; la valeur défaut est 1. Cependant, la meilleure façon de mettre un document en "espacement double", si vous avez la malchance d'avoir à produire une telle chose, est d'utiliser le paquetage `setspace` ; voir <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=linespace>.

`\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 décrit juste au-dessus.

Les modifications apportées en appelant les commandes de polices décrites ci-dessus n'entrent en vigueur qu'après que `\selectfont` est appelé.

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

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

5 Mise en page

Diverses 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 12).

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 46). Plus précisément, elle est inférieure à `\textwidth` par la somme de `\leftmargin` et `\rightmargin` (voir Section 8.14 [itemize], page 44). 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 12).

\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 12).

\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 26).

Dans les listes (voir Section 8.16 [list], page 46), `\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 48,) ou `\parbox` (voir Section 20.5 [\parbox], page 110), 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 \TeX primitif utilisé lorsque le texte est divisé en lignes. Il ne devrait pas être utilisé dans des documents \LaTeX en conditions normales.

\topmargin

L'espace entre le haut de la page \TeX (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

\LaTeX sait gérer plusieurs classes de matériel flottant. Il y a deux classes définies par défaut, `figure` (voir Section 8.10 [figure], page 42) et `table` (voir Section 8.22 [table], page 54), mais vous pouvez créer une nouvelle classes avec le paquetage `float`.

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

tableau apparaît dans le code source avant la première figure, il 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 `btp` 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.

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

Voici les 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 Des rubriques

Les commandes de rubricage fournissent les moyens de structurer votre texte en unités sectionnelles, ou rubriques :

```
\part
\chapter (seulement pour classes report et book)
\section
\paragraph
\subsubsection
\paragraph
\alinea
```

Toutes les commandes ont la même forme générale, par exemple,

```
\chapter[titretdm]{titre}
```

En plus de fournir le titre de rubrique *titre* dans le texte principal, le titre peut apparaître dans deux autres endroits :

1. La table des matières.
2. L'en-tête en cours de traitement en haut de la page.

Vous ne voudrez peut-être pas le même texte dans ces endroits que dans le texte principal. Pour gérer cela, les commandes de rubricage ont un argument optionnel *titretdm* qui, lorsqu'il est fourni, précise le texte de ces autres endroits.

En outre, toutes les commandes de rubricage ont des formes en *, aussi appelée étoilée, qui compose *titre* comme d'habitude, mais ne comprennent pas de numéro et ne produisent pas d'entrée dans la table des matières. Par exemple :

```
\section*{Préambule}
```

La commande `\appendix` modifie la façon dont les unités sectionnelles suivantes sont numérotées. La commande `\appendix` elle-même ne génère aucun texte et n'affecte pas la numérotation des parties. L'utilisation normale de cette commande est quelque chose comme cela :

```
\chapter{Un chapitre}
...
\appendix
\chapter{La première annexe }
```

Le compteur `secnumdepth` contrôle la composition des numéros de rubrique. le paramètre

```
\setcounter{secnumdepth}{niveau}
```

supprime le numéros de rubrique à n'importe quelle profondeur $> \textit{niveau}$, où `chapter` est le niveau zéro. (Voir Section 13.4 [`\setcounter`], page 79.)

7 Des références croisées

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. »

Inclure le numéro de figure dans le code source est une mauvaise pratique puisque ce si ce numéro change avec l'évolution du document, alors on doit penser à mettre à jour cet référence manuellement. Au lieu de cela, L^AT_EX vous fait écrire une *étiquette* du genre `\label{eq:ThmGreens}` et lui faire référence avec `Voir l'équation~\ref{eq:ThmGreens}`.

L^AT_EX écrit l'information provenant des étiquettes dans un fichier avec le même nom que le fichier contenant le `\label{...}` mais avec une extension `.aux`. (L'information a le format `\newlabel{étiquette}{\{étiquette-courante\}{numéro-de-page}}` où *étiquette-courante* est la valeur de la macro `\@currentlabel` qui est d'ordinaire mise à jour à chaque fois que that is vous appelez `\refstepcounter{compteur}`.)

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{clef}` qui apparaît avant le `\label{clef}` associé ; voir l'exemple dans la description de `\pageref{...}`. L^AT_EX obtient les référence à partir du fichier `.aux`. 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.

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 34).

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

`eq` pour les équations

Ansi, `\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 33). 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 12).

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 12) ; 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 42). Les entrées dans chaque colonne sont séparées avec une esperluette (&). Les lignes sont terminées par une double contr’oblique (voir Section 9.1 [\\], page 63).

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 55, 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 [étiquette1] article1
\item [étiquette2] article2
...
\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. Bien que les étiquettes sur les articles sont optionnelles il n'y a pas de valeur par défaut sensée, c'est pourquoi tous les articles devraient avoir une étiquette.

La liste consiste en au moins un article ; voir `\item`, page `\item`, (l'absence d'article cause l'erreur L^AT_EX 'Something's wrong--perhaps a missing \item'). Chaque article est produit avec une commande `\item`.

Une autre variation : puisque le style gras est appliqué aux étiquettes, si vous composez une étiquette en style tapuscrit avec `\texttt`, vous obtiendrez du gras tapuscrit : `\item[\texttt{gras et tapuscrit}]`. C'est peut-être trop gras, entre autres problèmes. Pour obtenir juste le style tapuscrit, utilisez `\tt`, ce qui réinitialise toutes les autres variations de style : `\item[{\tt tapuscrit de base}]`.

Pour les détails concernant l'espacement des listes, voir Section 8.14 [itemize], page 44.

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 12.

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 42).

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

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

La construction `\[des maths\]` est essentiellement un synonyme de `\begin{displaymath}des maths\end{displaymath}`, mais ce dernier est plus pratique à manipuler dans le code 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. (La construction `$$des maths$$` tirée du langage T_EX de base est souvent utilisée à tort comme un synonyme de `displaymath`. Ce n'en est pas un, parce que l'environnement `displaymath` vérifie qu'il ne commence pas en mode mathématique, parce qu'il gère l'espacement vertical environment différemment, et parce qu'il honore l'option `fleqn`.)

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 3.

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 44) et `description` (voir Section 8.4 [description], page 38). 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 78). Si vous utilisez l’argument optionnel d’`\item` alors le compteur n’est pas incrémenté pour cet article (voir `\undefined` [`\item`], page `\undefined`).

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 70). 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 78.

Pour plus ample information sur la personnalisation de la forme voir Section 8.16 [list], page 46. 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’artice « 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 consecutives 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}
 $\end{equation}$ 
```

L'environnement `equation` commence un environnement en `displaymath` (voir Section 8.5 [`displaymath`], page 39), 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]
 $\caption[titre]{texte}
\label{étiquette}
\end{figure}$ 
```

ou

```
\begin{figure*}[placement]
 $\caption[titre]{texte}
\label{étiquette}
\end{figure*}$ 
```

Une classe de flottant (voir Section 5.6 [Floats], page 28). 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 28.

La version étoilée `figure*` est utilisée quand un document est en mode double-colonne (voir Section 5.2 [`twocolumn`], page 26). 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 28.

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 33). La commande `\caption` spécifie la légende `texte` pour la figure. La légende est numérotée par défaut. Si `titre` est présent, il est utilisé dans la liste des figures au lieu de `texte` (voir Section 23.1 [Tables of contents], page 120).

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}{nomfichier}
  texte
\end{filecontents}
```

ou

```
\begin{filecontents*}{nomfichier}
  texte
\end{filecontents*}
```

Crée un fichier nommé *nomfichier* et le remplit de *texte*. La version non étoilée de l'environnement `filecontent` préfixe le contenu du fichier créé d'une en-tête ; voir l'exemple ci-dessous. La version étoilée `filecontent*` n'inclut pas l'en-tête.

Cet environnement peut être utilisé n'importe-où dans le préambule, bien qu'elle apparaisse en général avant la commande `\documentclass`. Elle est typiquement utilisée quand un fichier source a besoin d'un fichier non standard de style ou de classe. L'environnement permet d'écrire ce fichier dans le répertoire contenant ce fichier et ainsi de rendre le code source autonome. Une autre utilisation est d'inclure des références au format `bib` dans le fichier source, de même pour le rendre autonome.

L'environnement vérifie si un fichier de même nom existe déjà, et si oui, ne fait rien. Il y a un paquetage `filecontents` qui redéfinit l'environnement `filecontents` de sorte qu'au lieu de ne rien faire dans ce cas, il écrase le fichier existant.

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 `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 40).

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. Par convention, `\leftmargin` est réglée à la `\leftmarginN` appropriée quand on entre dans un nouveau niveau d'imbrication.

Les valeurs par défaut varient de `' .5em'` (pour les niveaux d'imbrication les plus élevés) jusqu'à `'2.5em'` (premier niveau), et sont légèrement réduits en mode deux-colonnes. Cet exemple réduit grandement l'espace de marge pour la liste la plus extérieure :

```
\setlength{\leftmargini}{1.25em} % 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 [`\parskip`], page 82):

```
\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 24 [Letters], page 122.

8.16 `list`

L'environnement `list` est un environnement générique qui est utilisé pour définir beaucoup des environnements plus spécifiques. Il est rarement utilisé dans des documents, mais souvent dans des macros.

```
\begin{list}{étiquetage}{espacement}
\item item1
\item item2
```

```
...
\end{list}
```

L'argument obligatoire *étiquettage* spécifie comment les articles doivent être étiquetés (à moins que son argument optionnel soit fourni à `\item`). Cet argument est un morceau de texte qui est inséré dans une boîte pour former l'étiquette. Il peut contenir, et d'habitude c'est le cas, d'autres commandes L^AT_EX.

L'argument obligatoire *espacement* contient des commandes pour changer les paramètres d'espacement pour la liste. Cet argument est le plus souvent vide, c.-à-d. {}, ce qui laisse l'espacement par défaut.

L'argument utilisé pour composer les articles de la liste est spécifiée par `\linewidth` (voir Section 5.5 [Page layout parameters], page 27).

Synopsis :

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

ou

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

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

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

Dans cet exemple la liste `enumerate` a deux articles qui utilisent 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.').

8.17 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 84.

8.18 minipage

```
\begin{minipage}[position][hauteur][pos-interne]{largeur}
texte
\end{minipage}
```

L'environnement `minipage` compose son corps *texte* dans un bloc qui ne peut pas être coupé sur plusieurs pages. C'est similaire à la commande `\parbox` (voir Section 20.5 [`\parbox`], page 110), mais contrairement à `\parbox`, d'autres environnements de production de paragraphe peuvent être utilisés au sein d'une `minipage`.

Les arguments sont les mêmes que pour `\parbox` (voir Section 20.5 [`\parbox`], page 110).

Par défaut, les paragraphes ne sont pas renforcés au sein d'un environnement `minipage`. Vous pouvez restaurer le renforcement avec une commande telle que `\setlength{\parindent}{1pc}`.

Les notes en bas de page au sein d'un environnement `minipage` sont gérées d'une façon qui est particulièrement utiles pour mettre des notes en bas de page dans des figures ou des tableaux. Une commande `\footnote \footnotetext` met la note en bas de page au bas de la `minipage` au lieu de la mettre au bas de la page, et elle utilise le compteur `\mpfootnote` au lieu du compteur ordinaire `footnote` (voir Chapitre 13 [Counters], page 78).

Cependant, ne mettez pas une `minipage` à l'intérieur d'une autre si vous utilisez les notes en bas de page ; elles pourraient terminer en bas de la mauvaise `minipage`.

8.19 picture

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

L'environnement `picture` vous permet de créer toute sorte de dessin dans lequel vous voulez du texte, des lignes, des flèches et des cercles. 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 droit est 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, -0.3). 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 109) 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 109.

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){longueur}
```

La commande `\line` dessine un ligne avec la *longueur* donnée et la pente *xpente/ypente*.

De base \LaTeX peut seulement dessiner des ligne dont la *pente* = x/y , où *x* et *y* prennent des valeurs entières de -6 jusqu'à 6 . Pour des ligne de pente quelconque, sans parler d'autres formes, voir le paquetage `curve2e` et bien d'autres sur le CTAN.

8.19.7 `\linethickness`

La commande `\linethickness{dim}` déclare que l'épaisseur des lignes horizontales et verticales dans un environnement `picture` vaut *dim*, qui doit être une longueur positive.

`\linethickness` 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 51, et Section 8.19.9 [`\thinlines`], page 51.

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 51, et Section 8.19.8 [`\thicklines`], page 51.

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 une espace verticale entre les paragraphes (c'est la longueur élastique `\parsep`). Ainsi, l'environnement `quotation` convient mieux environm 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 taquet et en tabulant jusqu'à eux bien comme on le ferait sur une machine à écrire. 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`.

Les commandes suivante peuvent être utilisée au sein d'un environnement `tabbing` :

- `\ (tabbing)`
Finit une ligne.
- `\= (tabbing)`
Règle un taquet à la position courante.
- `\> (tabbing)`
Avance au taquet suivant.
- `\<`
Place le texte qui suit à la gauche de la marge locale (sans changer cette marge). Ne peut être utilisé qu'au début d'une ligne.
- `\+`
Déplace la marge de gauche d'un taquet vers la droite pour la ligne suivante et toutes celles qui suivent.
- `\-`
Déplace la marge de gauche d'un taquet vers la gauche pour la ligne suivante et toutes celles qui suivent.
- `\' (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`, à la droite de la colonne précédente, aligné à droite sur le taquet de la colonne courante.
- `\' (tabbing)`
Vous permet de placer du texte justifié à droite sur n'importe quel taquet, y compris le taquet 0. Toutefois, ne peut pas placer du texte à la droite de la dernière colonne parce qu'il n'y a pas de taquet à cet endroit. La commande `\'` déplace tout le texte qui la suit, jusqu'à la commande `\` ou `\end{tabbing}` qui termine la ligne.
- `\a (tabbing)`
Dans un environnement `tabbing`, les commandes `\=`, `\'` et `\'` ne produisent pas d'accents comme d'habitude (voir Section 21.4 [Accents], page 115). 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 positions courantes de tous les taquets. C'est utile pour changer temporairement les positions de taquets au milieu d'un environnement `tabbing`.
- `\tabbingsep` Distance à la gauche d'un taquet déplacé par `\'`.

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

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

8.22 table

Synopsis :

```
\begin{table}[placement]
  corps-du-talbeau
  \caption[titreltd]{titre}
\end{table}
```

Une classe de flottants (voir Section 5.6 [Floats], page 28). Parce qu'ils ne peuvent pas être coupés à cheval sur plusieurs pages, il ne sont pas composés en séquence avec le texte normale, mais à la place sont « flottés » vers un endroit convenable, tel que le sommet d'une page suivante.

Pour les valeurs possibles de *placement* et leur effet sur l'algorithme de placement des flottants, voir Section 5.6 [Floats], page 28.

Le corps du tableau 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 33). La commande `\caption` spécifie la légende *texte* pour la figure. La légende est numérotée par défaut. Si *titreltd* est présent, il est utilisé dans la liste des talbeaux au lieu de *titre* (voir Section 23.1 [Tables of contents], page 120).

```
\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 ligne 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 `\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 12.8 [`\protect`], page 76).

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 utilisée au sein d'une cellule de colonne en `p` ces déclaration doivent apparaître entre accolade, comme dans `{\centering .. \\ ..}`. Sans quoi `LaTeX` 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}{lcccl}
  \textit{ID}      & \multicolumn{2}{c}{\textit{Nom}} & \textit{Âge} \\
  978-0-393-03701-2 & 0'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` contiguë 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 entré 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 » ; dans la classe `report`, elle est intitulée « Bibliographie ». Vous pouvez changer le titre (dans les classes standards) en redéfinissant la commande `\refname`. Par exemple, cela l'élimine complètement :

```
\renewcommand{\refname}{} 
```

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 pas 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 60) 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 60), 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 BibTeX

Si vous utilisez le programme BibTeX 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 60). 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 références bibliographiques. Les noms *stylebib* standards distribués avec BibTeX sont :

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

De plus, de nombreux autres styles BibTeX 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 données au format BibTeX. 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 possibilités de numérotation pour *n* sont décrites par `\newtheorem` (voir Section 12.6 [`\newtheorem`], page 74).

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 21.7 [`\today`], page 118).

Vous pouvez utiliser la commande `\maketitle` (voir Section 18.1 [`\maketitle`], page 103) 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 machine à écrire à chasse fixe (`\tt`).

8.27.1 \verb

Synopsis :

```
\verbcartexte-littéralcar
\verb*cartexte-littéralcar
```

La commande `\verb` compose *texte-littéral* comme il est entré, en incluant les caractères spéciaux et les espaces, en utilisant la police tapuscrit (`\tt`). Aucun espaces n'est autorisé entre `\verb` ou `\verb*` et le délimiteur *car* qui marque le début et la fin du texte verbatim. Le délimiteur ne doit pas apparaître dans le *texte-littéral*.

La forme étoilée (en `*`) diffère seulement en ce que les espaces sont tapés avec un caractère « espace visible ». (Nommément, `_`).

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

9.1 \backslash

Synopsis :

```
 $\backslash$ [espaceenplus]
```

ou

```
 $\backslash$ *[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 \LaTeX . 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 \backslash (voir Section 9.3 [`\newline`], page 63).

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

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

9.2 `\obeycr` & `\restorecr`

La commande `\obeycr` a pour effet qu'un retour chariot dans le fichier d'entrée (' \sim M', internement) soit traité de la même façon que \backslash (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 [\backslash], page 63) ; cela rompt la ligne, sans dilatation du texte le précédent.

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 L^AT_EX qu'il peut faire une césure du mot à cet endroit. L^AT_EX 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 L^AT_EX 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 T_EX 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 65) et `\-` (voir Section 9.4 [`\-` (`hyphenation`)], page 64).

`\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 98). 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 65).

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 65).

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 `\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.

10.2 `\clearpage`

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.

10.3 `\newpage`

La commande `\newpage` finit la page courante, mais ne débouffe pas tous les flottants (voir Section 10.2 [`\clearpage`], page 66).

10.4 `\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.

10.5 `\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 78). Pour rendre ce changement global mettez cela dans le préambule. Si vous faites ce changement local alors vous désirez 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 48.

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 48).

```
\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 54). 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 normal pour des police de taille `\footnotesize` (voir Section 4.2 [Font sizes], page 24), 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}

```

La forme étoilée (en `*`) de ces commandes exige que les arguments ne contiennent pas de paragraphes de texte (la commande n'est pas `\long`, dans la terminologie de T_EX de base).

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 secon *#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.

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

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

les mots 'rencontra le' 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 \newcounter

Synopsis :

```
\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 78, pour plus d'information à propos des compteurs.

12.3 \newlength

Synopsis :

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

La commande `\newlength` définit l'argument obligatoire comme une commande de *longueur* avec une valeur de 0mm. L'argument doit être une séquence de contrôle, comme dans `\newlength{foo}`. Une erreur se produit si `\foo` est déjà défini.

Voir Chapitre 14 [Lengths], page 80, pour comment régler la nouvelle longueur à une valeur non nulle, et pour plus d'information à propos des longueurs en général.

12.4 `\newsavebox`

Synopsis :

```
\newsavebox{\cmd}
```

Définit `\cmd` pour se référer à un nouveau bac pour stocker des boîtes. Une telle boîte sert à détenir du matériel composé, pour l'utiliser plusieurs fois (voir Chapitre 20 [Boxes], page 109) 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 12.8 [\protect], page 76).

12.5 `\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. *optargdefault*. Dans les deux cas, le premier paramètre positionnel faisant référence à argument obligatoire est #2.

Omettre `[mava1]` 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 *déffin*, dans ce cas dans une boîte (voir Section 20.8 [`\sbox`], page 111).

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

```

\end{quotation}
}

```

12.6 \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 34).

Cette déclaration est globale. Elle est fragile (voir Section 12.8 [`\protect`], page 76).

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 simili-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}{Définition}[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{...}

```

12.7 `\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 \XeLaTeX (voir Section 2.3 [\TeX engines], page 4).

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
```

12.8 `\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 ceut qui apparaissent au sein des commande `\caption{...}` (voir Section 8.10 [figure], page 42), dans la commande `\thanks{...}` (voir Section 18.1 [`\maketitle`], page 103), et dans les expression en @ des environnements `tabular` et `array` (voir Section 8.23 [tabular], page 55).

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 79, ou Section 13.4 [`\setcounter`], page 79.

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}
```

La commande `\value` produit la valeur de *compteur*. Ça peut être utilisé partout là où \LaTeX attend un nombre, par exemple :

```
\setcounter{moncompteur}{3}
\addtocounter{moncompteur}{1}
\hspace{\value{moncompteur}\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 79, à ceci près qu'elle définit également la valeur courante de `\ref` comme le résultat de `\thecounter`.

13.7 `\stepcounter{compteur}`

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

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

\LaTeX 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 \TeX démarre, ils sont mis à la valeur courante du système sur lequel \TeX s'exécute. Ils ne sont pas remis à jour pendant que le traitement par \TeX progresse.

En relation avec ces compteurs, la commande `\today` produit une chaîne représentant le jour courant (voir Section 21.7 [`\today`], page 118).

14 Longueurs

Une *longueur* est une mesure de distance. Beaucoup de commandes L^AT_EX prennent une longueur en argument.

14.1 Unités de longueur

T_EX et L^AT_EX 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>	Centimètre
<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 à 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 98.

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 le 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 le longueur `\longueur` à la hauteur de l'argument *texte*.

14.6 `\settowidth`

Synopsis :

```
\settowidth{\longueur}{texte}
```

La commande `\settowidth` 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 109). 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 *renfoncement* normal d'un paragraphe. Elle est utilisée pour ajouter un renfoncement 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 renfoncement 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 renfoncement de paragraphe dans tout un document, mettez `\setlength{\parindent}{0pt}` dans le préambule.

15.3 `\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 12) ;
- dans la marge extérieure pour une disposition en recto-verso (option `twoside`, voir Section 3.1 [Document class options], page 12) ;
- dans la nearest la plus proche pour une disposition à deux colonnes (option `twocolumn`, voir Section 3.1 [Document class options], page 12).

La commande `\reversemarginpar` place les notes en marge suivante sur la marge opposée (interne). `\normalmarginpar` les replace dans la position par défaut.

Lorsque vous spécifier les deux arguments, *gauche* est utilisé pour is used la marge de gauche, et *droite* est utilisé pour la marge de is used droite.

Le premier mot doit normalement ne pas être sujet à une césure ; Vous pouvez activer la césure à cet endroit en commençant le nœud avec `\hspace{0pt}`.

Ces paramètres affectent le formatage de la note :

`\marginparpush`

Espace verticale minimale entre les notes ; par défaut ‘7pt’ pour des documents à ‘12pt’, ‘5pt’ sinon.

`\marginparsep`

Espace horizontale entre le texte principal et la note ; par défaut ‘11pt’ pour des documents à ‘10pt’, ‘10pt’ sinon.

`\marginparwidth`

Largeur de la note même ; la valeur par défaut est ‘90pt’ pour un document en recto simple et à ‘10pt’, ‘83pt’ pour ‘11pt’, et ‘68pt’ pour ‘12pt’ ; ‘17pt’ de plus dans chacun de ces cas pour un document recto-verso. En mode 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 101.

<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>	⊗ 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>	△ De taille variable, ou n-aire, triangle ouvert pointant vers le haut (opérateur).
<code>\bigsqcup</code>	⊔ De taille variable, ou n-aire, union carrée (opérateur).
<code>\biguplus</code>	⊕ De taille variable, ou n-aire, opérateur union avec un plus (opérateur). (Notez que le nom a juste un p.)
<code>\bigvee</code>	∨ De taille variable, ou n-aire, et-logique (opérateur).
<code>\bigwedge</code>	∧ 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>	⋈ 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>	∩ 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>	≅ Congruent (relation).
<code>\coprod</code>	∐ Coproduit (opérateur).
<code>\cup</code>	∪ 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>	<code>\Diamond</code> Opérateur grand carreau (ordinaire). Ceci n'est pas disponible en \TeX de base. En \LaTeX vous devez charger le paquetage <code>amssymb</code> .
<code>\diamond</code>	\diamond Opérateur Carreau, ou puce carreau (binaire). Similaire : grand carreau <code>\Diamond</code> , puce disque <code>\bullet</code> .
<code>\diamondsuit</code>	\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>	\geq Supérieur ou égal à (relation). Synonyme : <code>\geq</code> .
<code>\geq</code>	\geq Supérieur ou égal à (relation). Synonyme : <code>\ge</code> .

<code>\gets</code>	\leftarrow Auquel est assignée la valeur (relation). Synonyme : <code>\leftarrow</code> .
<code>\gg</code>	\gg Beaucoup plus grand que (relation). Similaire : Beaucoup plus petit que <code>\ll</code> .
<code>\hbar</code>	\hbar Constante de Planck sur deux pi (ordinaire).
<code>\heartsuit</code>	\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'epsilon 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>\rightarrow</code> Harpon vers la droite avec crochet vers le bas (relation).
<code>\rightharpoonup</code>	<code>\rightarrow</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` \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` \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 : `\reemptyset`. Ceci n'est pas disponible en T_EX de base. En L^AT_EX vous devez charger le paquetage `amssymb`.
- `\varphi` φ Variante de la lettre grecque bas de casse (ordinaire). La forme non en variante est `\phi`.
- `\varpi` ϖ Variante de la lettre grecque bas de casse (ordinaire). La forme non en variante est `\pi`.
- `\varrho` ϱ Variante de la lettre grecque bas de casse (ordinaire). La forme non en variante est `\rho`.
- `\varsigma` ς Variante de la lettre grecque bas de casse (ordinaire). La forme non en variante est `\sigma`.
- `\vartheta` ϑ Variante de la lettre grecque bas de casse (ordinaire). La forme non en variante est `\theta`.
- `\vdash` \vdash Taquet droit ; prouve, implique ; portillon/tourniquet ; barre verticale et un tiret (relation). Similaire : portillon inversé `\dashv`.
- `\vee` \vee Ou logique ; une forme de v pointant vers le bas (binaire). En relation avec : Et logique `\wedge`. Similaire : Opérateur de taille variable `\bigvee`.
- `\Vert` $\|$ Double barre verticale (ordinaire). Similaire : barre verticale simple `\vert`.
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 21.4 [Accents], page 115).

<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 80).

<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 moins 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 105.
<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 pas 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 (espacés 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`
; `\text{rm}{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 98).

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 plus1fill`’) ; 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 105). 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 105.

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 104.

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 22).

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 80) 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 80). 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 12.8 [`\protect`], page 76).

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 courante et insère une longueur élastique verticale (voir Chapitre 14 [Lengths], page 80) qui est infinie, de sorte qu'elle puisse se dilater ou that is se rétrécire 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 une espace verticale infinie sous sa ligne sans tenir compte de la structure du paragraphe. Dans les deux cas cette espace disparaît à la fin de la page ; pour éviter cela voir Section 19.12 [`\vspace`], page 107.

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 80).

\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 81) 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 50, 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 111.

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 49.

20.5 `\parbox`

Synopsis :

```
\parbox[position][hauteur][pos-interne]{largeur}{texte}
```

La commande `\parbox` produit une boîte dont le contenu est créé en *mode paragraphe*. On devrait l'utiliser pour fabriquer une boîte à partir de petits morceaux de texte, avec rien de trop élaboré à l'intérieur. En particulier, vous ne devriez utiliser aucun environnement de fabrication de paragraphe au sein de l'argument *texte* d'une `\parbox`. Pour des morceaux de texte de plus grande taille, y compris ceux comprenant un environnement de fabrication de paragraphe, vous devriez utiliser un environnement `minipage` (voir Section 8.18 [`minipage`], page 48).

`\parbox` a deux arguments obligatoires :

- largeur* la largeur de la parbox;
- texte* le texte qui à mettre dans la parbox.

Par défaut \LaTeX positionne verticalement un parbox de sorte que son centre soit aligné avec le centre de la ligne de texte alentour. Lorsque l'argument optionnel *position* est présent et vaut soit 't' soit 'b', cela vous permet respectivement d'aligner soit la ligne du haut, soit la ligne du bas de la parbox avec la ligne de base du texte alentour. Il est possible de spécifier *position* à 'm' pour avoir le comportement par défaut.

L'argument optionnel *hauteur* l'emporte sur la hauteur naturelle de la boîte.

L'argument *pos-interne* contrôle la position du texte au sein de la boîte, comme suit ; s'il n'est pas spécifié, *position* est utilisé.

- t** le texte est placé en haut de la boîte.
- c** le texte est centré dans la boîte.
- b** le texte est placé en bas de la boîte.
- s** dilatation (stretch) verticale ; le texte doit contenir de l'espaces dilatable verticalement pour que cela fonctionne.

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 109), 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 12.4 [`\newsavebox`], page 72).

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 109) 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 12.4 [`\newsavebox`], page 72).

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 111).

21 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.

21.1 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*.

```
# $ % & { } _ ~ ^ \
```

Si vous voulez qu'un caractère réservés 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 d'une machine à écrire utilisez `\verb!!` comme ci-dessous :

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

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

21.2 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}`.

21.3 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`.

```
\copyright
```

```
\textcopyright
```

Le symbole « droit d'auteur », ©.

```
\dag
```

Le symbole obèle (dans le texte).

```
\ddag
```

Le symbole double obèle (dans le texte).

`\LaTeX` Le logo L^AT_EX.
`\LaTeXe` Le logo L^AT_EX2e.
`\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 gauche (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`
Symbol espace visible.

21.4 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 53).
- `\.` Produit un point suscrit à la lettre qui suit, comme dans ò.
- `\=`
`\capitalmacron`
Produit un macron (une barre) suscrit à la lettre qui suit, comme dans 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 53).
- `\~`
`\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 98. Voir aussi `\b` ci-avant.

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

21.5 Lettres latines supplémentaires

Voici les commandes de base de L^AT_EX 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.

21.6 `\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).

21.7 `\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 79) 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.

22 Partitionner le fichier d'entrée

Un document de grande taille nécessite beaucoup de données en entrée. Plutôt que de mettre toutes les entrées dans un seul grand fichier, il est plus efficace de les partitionner en plusieurs fichiers plus petits. Indépendamment du nombre de fichiers distincts que vous utilisez, il y en a un seul qui est le *fichier racine* ; c'est le fichier dont vous passez le nom à la ligne de commande quand vous exécutez \LaTeX .

Voir Section 8.11 [filecontents], page 43, pour un environnement qui permet de grouper un fichier externe à créer avec le document principal.

22.1 `\include`

Synopsis :

```
\include{fichier}
```

Si aucune commande `\includeonly` n'est présente, la commande `\include` exécute un `\clearpage` pour commencer une nouvelle page (voir Section 10.2 [clearpage], page 66), puis elle lit *fichier*, et ensuite fait encore un `\clearpage`.

Précédé d'une commande `\includeonly`, les actions de `\include` ne sont exécutées que si *fichier* est un élément des fichiers listés en argument de l'`\includeonly`. Voir Section 22.2 [includeonly], page 119.

La commande `\include` ne doit apparaître ni dans le préambule, ni au sein d'un fichier lu par une autre commande `\include`.

22.2 `\includeonly`

Synopsis :

```
\includeonly{fichier1,fichier2,...}
```

La commande `\includeonly` contrôle quels fichiers sont lisibles par toute commande `\include` venant ensuite. La liste des noms de fichier est à éléments séparés par des virgules. Chaque élément *fichier1*, *fichier2*, ... doit exactement correspondre à un nom de fichier spécifié en argument d'une commande `\include` pour que la sélection soit effective.

Cette commande ne peut apparaître que dans le préambule.

22.3 `\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`).

23 Prologue et épilogue

23.1 Tables des matières

On produit une table des matières avec la commande `\tableofcontents`. Placez la commande à l'endroit même où vous voulez que la table des matières apparaisse ; \LaTeX fera le reste pour vous. Une seconde compilation est nécessaire pour qu'un fichier `.toc` ait été produit à la précédente.

La commande `\tableofcontents` produit un titre, mais il ne produit pas automatiquement un début de nouvelle page. Si vous voulez une nouvelle page après la table des matières, mettez une commande `\newpage` après la commande `\tableofcontents`.

Les commandes analogues `\listoffigures` et `\listoftables` produisent respectivement une liste des figures et une listes des tableaux. Le fonctionnement est exactement le même que pour la table des matières.

La commande `\nofiles` l'emporte sur ces commandes, et *empêche* la génération de ces listes.

23.1.1 `\addcontentsline`

Synopsis :

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

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

<i>ext</i>	L'extension de nom de fichier du fichier dans lequel l'information doit être écrite, typiquement c'est l'une des trois suivante :: <code>toc</code> pour la table des matières (« table of contents » en langue anglaise), <code>lof</code> pour la liste des figures (« list of figures »), ou <code>lot</code> pour la liste des tableaux (« list of tables »).
<i>unit</i>	Le nom de l'unité sectionnelle à ajouter, typiquement l'une des suivantes, selon la valeur de l'argument <i>ext</i> :
<code>toc</code>	Les unités sectionnelles : <code>part</code> , <code>chapter</code> , <code>section</code> , <code>subsection</code> , <code>subsubsection</code> .
<code>lof</code>	Pour la liste des figures : <code>figure</code> .
<code>lot</code>	Pour la liste des tableaux : <code>table</code> .
<i>texte</i>	Le texte de l'entrée.

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

23.1.2 `\addtocontents`

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

<i>ext</i>	L'extension de nom de fichier du fichier dans lequel l'information est à écrire : <code>toc</code> (table des matières), <code>lof</code> (liste des figures), ou <code>lot</code> (liste des tableaux).
<i>texte</i>	Le texte à écrire.

23.2 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.

23.3 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 `\alsiname`. 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 T_EX FAQ à ce propos, <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=multind>.

24 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`.

24.1 `\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 entête pré-imprimé. (Voir Chapitre 2 [Overview], page 3, pour les détails concernant votre implémentation locale). Avec la déclaration `\address`, elle est formatée comme une lettre personnelle.

24.2 `\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}
```

24.3 `\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.}
```

24.4 `\encl`

Synopsis :

```
\encl{ligne1\\ligne2}
```

Déclare une liste d'une ou plus de pièces jointes.

24.5 `\location`

Synopsis :

```
\location{texte}
```

Le *texte* apparaît centré au bas de chaque page. Il apparaît seulement si le style de la page est `firstpage`.

24.6 `\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}
```

24.7 `\name`

```
\name{June Davenport}
```

Le nom de l'expéditeur, à imprimer sur l'enveloppe avec l'adresse de retour.

24.8 `\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,}
```

24.9 `\ps`

Synopsis :

```
\ps{texte}
```

Utilisez la commande `\ps` pour commencer un postscriptum dans une lettre, après `\closing`.

24.10 `\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).

24.11 `\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.

25 Entrée/sortie console

25.1 `\typein[cmd]{msg}`

Synopsis :

```
\typein[cmd]{msg}
```

`\typein` tape *msg* sur la console et fait que L^AT_EX 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.

25.2 `\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.

26 Ligne de commande

La spécification du fichier d'entrée indique le fichier à être formaté ; \TeX utilise `.tex` comme extension de fichier par défaut. Si vous omettez complètement le fichier d'entrée, \TeX accepte en entrée les caractères saisis à partir de la console. Vous spécifiez les options de commande en fournissant une chaîne en paramètre de la commande ; par ex.

```
latex '\nonstopmode\input foo.tex'
```

Traitera `foo.tex` sans faire de pause après chaque erreur.

Si \LaTeX s'arrête au milieu du document et vous fait une invite en `*`, c'est qu'il attend que vous saisissez quelque chose. Vous pouvez taper `\stop` (et retour-chariot) et il terminera prématurément le document.

Voir Section 2.3 [\TeX engines], page 4, pour les autres commandes système invoquant \LaTeX .

Annexe A Patrons de documents

Bien que ce ne soit pas un matériel de référence, peut-être que ces patrons de document vous seront utiles. Davantage de ressources de patrons sont listé à <http://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 2007}

\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}
```

Une ressource web pour cela : <http://robjhyndman.com/hyndsight/beamer/>.

A.2 Un patron book

```
\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.3 Un patron tugboat

TUGboat est la revue du groupe des utilisateur de T_EX (T_EX Users Group), <http://tug.org/TUGboat>.

```
\documentclass{ltugboat}
\usepackage{graphicx}
\usepackage{ifpdf}
\ifpdf
\usepackage[breaklinks,hidelinks]{hyperref}
\else
\usepackage{url}
\fi

\title{Exemple d'article \TUB}

% répétez l'information pour chaque auteur.
\author{Prénom Nom}
\address{Voie\\ Ville, Code postal \\ Pays}
\netaddress{user (at) exemple dot org}
\personalURL{http://exemple.org/~user/}

%% Fin des métadonnées %%

\begin{document}

\maketitle

\begin{abstract}
Ceci est un exemple d'article pour le \TUB{}.
Prière d'écrire un abrégé.
\end{abstract}

\section{Introduction}

Ceci est un exemple d'article pour le \TUB, à partir du lien
\url{http://tug.org/TUGboat/location.html}.
```

Nous recommandons le paquetage `graphicx` pour les inclusionj d'iimage, et le paquetage `hyperref` pour des adresses réticulaires (`url`) actives (dans la sortie `\acro{PDF}`). De nos jour le `\TUB\` est produit en utilisant exclusivement des fichiers `\acro{PDF}`.

La classe `\texttt{ltugboat}` fournit ces abréviations (et bien d'autres) :

```
% verbatim blocks are often better in \small
\begin{verbatim}[\small]
\AllTeX \AMS \AmS \AmSLaTeX \AmSTeX \aw \AW
\BibTeX \CTAN \DTD \HTML
\ISBN \ISSN \LaTeXe
\mf \MFB
\plain \POBox \PS
\TUB \TUG \tug
\UNIX \XeT \WEB \WEAVE

\, \bull \Dash \dash \hyph

\acro{FRED} -> {\small[er] fred} % prière d'utiliser !
\cs{fred}   -> \fred
\meta{fred} -> <fred>
\nth{n}     -> 1st, 2nd, ...
\sfrac{3/4} -> 3/4
\booktitle{Book of Fred}
\end{verbatim}
```

Pour faire référence à d'autre numéros du `\TUB\`, veuillez utiliser le format `\textsl{volno:issno}`, par ex., "`\TUB\ 32:1`" pour notre numéro `\nth{100}`.

Ce fichier n'est qu'un patron. La documentation du style `\TUB\` est dans le document `\texttt{ltubguid}` se trouvant à `\url{http://ctan.org/pkg/tugboat}`. (Pour les références au `\CTAN\` nous recommandons cette forme d'URL utilisant `\texttt{/pkg/}` quand cela fait sens ; ou, si vous avez besoin de faire référence spécifiquement à une localisation : `\texttt{http://mirror.ctan.org/\textsl{chemin}}`.)

crivez par mél à `\verb|tugboat@tug.org|` en cas de problèmes ou questions.

```
\bibliographystyle{plain} % nous recommandons le style de bibliographie plain
\nocite{book-minimal}    % pour forcer la bibliographie à ne pas être vide
\bibliography{xampl}     % xampl.bib est compris dans BibTeX

\makesignature
\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

*

'*', invite 128

.

.glo, fichier 121

.idx, fichier 121

.ind, fichier 121

\

\include emboîtés interdits 119

«

« Voir » et « Voir aussi », article d'index 121

É

écrire des lettres 122

écrire un fichier externe 43

équations, aligner 41

équations, environnement pour 42

étiquette 33

étiquette tapuscrite dans les listes 39

étirement, omettre l'étirement vertical 27

étoilées, variantes de commandes 10

A

a e liés 117

a rond-en-chef 117

a, e dans l'a 117

accéder à un caractère
quelconque d'une police 112

accent 22, 23

accent aigu 116

accent aigu en math 97

accent barre suscrite en math 97

accent brève en math 97

accent caron 117

accent caron en math 97

accent chapeau en math 97

accent circonflexe 116

accent circonflexe en math 97

accent diérèse en math 97

accent double-brève-renversée 117

accent grave en math 97

accent háček 117

accent háček en math 97

accent macron en math 97

accent point 116

accent point en chef 116

accent point suscrit 116

accent point suscrit en math 97

accent rond en chef 116

accent rond-en-chef en math 98

accent tilde en math 98

accent tréma 115

accent tréma en math 97

accent tréma hongrois 116

accents 115

Accents mathématiques 97

accentuées, lettres 115

accentués, caractères 115

accolade droite, dans le texte 114

accolade gauche, dans le texte 114

afficher du texte cité avec

renforcement de paragraphe 52

afficher un texte cité sans

renforcement de paragraphe 52

agrandir la page courante 66

align environnement, d'amsmath 41

alignement au moyen de tabbing 53

aligner des équations 41

aligner du texte dans des tableaux 55

aligner du texte en utilisant des

taquets de tabulation 53

amsmath paquetage, remplaçant eqnarray 41

annexe, création 32

anomalies, rapports 2

aperçu de L^AT_EX 3

appel 125

arguments mouvants 76

article d'une liste, compteur spécifiant 78

articles numérotés, compteur spécifiant 78

ASCII, circonflexe dans le texte 113

ASCII, tilde dans le texte 113

astérisque centré, dans le texte 113

aval, référence déclarée en 33

B

barre au-dessous 117

barre par-dessus, accent 116

barre sousscite 117

barre verticale double, dans le texte 113

barre verticale, dans le texte 113

barre, diacritique 116

barre, double verticale, dans le texte 113

barre-souscrite, diacritique 116

bas de page, créer des notes 67

bas de page, des paramètres pour 27

bases de L^AT_EX 3

beamer patron et classe 129

bibliographie, créer (automatiquement) 61

bibliography, creating (manually) 60

bib _T E _X , utilisation	61
Big point	80
boîtes	109
boîtes noires, en omettant	12
bogues, rapports	2
Bonjour le monde	3
brève, accent	117
brève, diacritique	117

C

cédille, diacritique	116
césure, définir	65
césure, empêcher	109
césure, forcer	64
cadratin	80, 114
cadratin, demi	114
cadratin, deux tiers	115
cadratin, trois-quarts	115
caractères latins non diacrités	117
caractères réservés	112
caractères spéciaux	112
caractères speciaux	117
caron, diacritique	117
catégorie, code de ... de caractère	7
catcode	7
Centimètre	80
centré, astérisque, dans le texte	113
centré, point, dans le texte	114
centrer du texte, déclaration pour	38
centrer du texte, environnement pour	38
chapeau, accent	116
chargement des paquetages supplémentaires	13
chevron double, guillemets	113
chevron droit, guillemets	113
chevron gauche, guillemets	113
chevron simple, guillemets	113
chiffres de revêtement	23
chiffres de style ancien	23
chiffres de style ancien,	23
Cicero	80
circonflexe ASCII, dans le texte	113
circonflexe, accent	116
circonflexe, diacritique	116
citation	52
class options	16
classe de document, commandes	13
classe de document, définition	3
classe et paquetage, commandes	15
classe et paquetage, différence	14
classe et paquetage, structure	14
classe <code>ltugboat</code>	130
classe, disposition du fichier	14
classe, fichier d'exemple	14
classe, options	14
classes de documents	12
clause at, au sein d'une définition de police	76
clef de référence bibliographique	60

code de catégorie de caractère	7
code source, composer du	62
commande de longueur	80
commande longue	15
commandes de fontes de bas niveau	24
commandes de fontes, de bas niveau	24
commandes de mise en page	26
commandes des classes de document	13
commandes fragiles	76
commandes pour classe ou paquetage	15
commandes robustes	76
commandes, en définir de nouvelles	70
commandes, syntaxe des	5
commandes, variantes étoilées	10
commencer une nouvelle page	66
commencer une nouvelles page et débouurrer les flottant	66
commencer une page de droite	66
compteurs, a list of	78
compteurs, définir de nouveaux	71
compteurs, obtenir la valeur d'un	79
compteurs, réglage	79
contr'oblique, dans le texte	113
copie, liste des personnes en, dans une lettre	123
Correction italique	105
coupure de mot, empêcher	109
créer des images	48
créer des tableaux	54
crésure, discrétionnaire	64

D

débouurrer les flottant et commencer une page	66
début de document, crochet	39
début et fin	3
déclaration en aval de référence	33
définir de nouveaux théorèmes	74
définir de nouvelles polices	76
définir une nouvelle commande	70
définitions	70
délimiteur nul	99
date d'aujourd'hui	118
defining new environnements	72
demi-cadraitin	114
deux-tiers cadraitin	115
diacritique barre-souscrite	116
diacritique brève	117
diacritique cédille	116
diacritique caron	117
diacritique circonflexe	116
diacritique double-brève-renversée	117
diacritique háček	117
diacritique macron-souscrit	116
diacritique ogonek	116
diacritique point suscrit	116
diacritique point-souscrit	116
diacritique rond en chef	116
diacritique tilde	116

diacritique tirant sусcrit	117
diacritique tréma hongrois	116
diacritique accent aigu	116
diacritique tréma	115
diacritiques, signes	115
différence entre classe et paquetage	14
dilatation verticale infinie	107
discrétionnaire, césure	64
dollar, signe	114
double obèle, dans le texte	112
double renvoi, résolution	4
double, barre verticale, dans le texte	113
double, guillemet-apostrophe culbuté	115
double, guillemet-apostrophe de droite	115
double, obèle, dans le texte	114
double-brève-renversée, diacritique	117
droit d'auteur, symbole	112
droit, guillemet double sur la ligne de base	115
droit, guillemet simple sur la ligne de base	115
droit, guillemet-apostrophe simple	115
droite, accolade, dans le texte	114
droite, flèche, dans le texte	115
droite, guillemet anglais simple	113
droite, guillemet-apostrophe simple	113

E

e dans l'a	117
e dans l'o	118
e- $\text{T}_{\text{E}}\text{X}$	4
em	80
en-tête et pied de page en cours de traitement ..	27
en-tête, des paramètres pour	27
encerclée, lettre, dans le texte	114
entrée Unicode, native	4
entrée/sortie console	127
environnements	36
environnement	3
environnement simili-théorème	74
environnements, défining	72
espace vertical	106
espace vertical avant les paragraphes	82
espace vertical, insérer	106
espace verticale	107
espace visible	62
espace visible, symbole, dans le texte	115
espacement au sein du mode mathématique	98
espacement inter-phrase	105
espaces	104
eszet lettre allemande	118
eth islandais	117
eth, lettre islandaise	117
euro, symbole	114
exposant	84
extension, structure	14

F

féminin, symbole ordinal	114
fabriquer une page de titre	61
Faire des paragraphes	82
fermant, guillemet anglais simple	113
fermant, guillemet-apostrophe simple	113
fichier auxiliaire	4
fichier d'entrée	119
fichier de transcription	4
fichier externe, écrire	43
fichier journal	4
fichier racine	119
figures, insertion	42
figures, notes en bas de page au sein des	48
fin d'une lettre	124
fin document, crochet	40
fin et début	3
flèche droite, dans le texte	115
flèche gauche, left, dans le texte	114
fonctions mathématiques	96
forme étoilée de commandes de rubricage	32
forme étoilée des commandes d'environnement	72
forme étoilée, définir de nouvelles commandes ..	70
forme en * de commandes de rubricage	32
forme en * des commandes d'environnement	72
forme en *, définir de nouvelles commandes	70
formules en ligne	47
formules mathématiques	84
formules, environnement pour	42
formules, math	84
fragile, commande	76
français, guillemets	113

G

gauche, accolade, dans le texte	114
gauche, flèche, dans le texte	114
gauche, guillemet-apostrophe simple de	113
globales, options	13
glossaires	121
Grand rond, symbole, dans le texte	113
gras	23
gras tapuscrit, éviter	39
grave accent	116
guillemet droit double sur la ligne de base	115
guillemet droit simple sur la ligne de base	115
guillemet-apostrophe de droite double	115
guillemet-apostrophe double culbuté	115
guillemet-apostrophe simple	115
guillemet-apostrophe simple culbuté	115
guillemet-apostrophe simple droit	115
guillemet-virgule inférieur double	113
guillemet-virgule inférieur simple	113
guillemets double	113
guillemets français	113

guillemets simple 113
 guillemets-virgule inférieurs,
 simple et double 113

I

i sans point 116
 i sans point en math 98
 Icelandic thorn 118
 ici, placer les flottants 29
 images, créer 48
 implémentations de \TeX 4
 index 121
 indice inférieur 84
 indice supérieur 84
 inférieur à, symbole dans le texte 114
 inférieur, guillemet-virgule double 113
 inférieur, guillemet-virgule simple 113
 insérer des figures 42
 interligne double 25
 invite ‘*’ 128
 italique 23

J

j sans point 116
 j sans point en math 98
 jsutifier à gauche du texte,
 environnement pour 44
 justifié à gauche du texte 44
 justification, drapeau droit 44
 justification, drapeau gauche 44
 justifier à droite, environnement pour 44
 justifier le texte à droite 44

K

Knuth, Donald E. 3

L

l polonais 117
 L'équipe de projet \LaTeX 2
 Lamport \TeX 3
 Lamport, Leslie 3
 \LaTeX aperçu 3
 \LaTeX ou \LaTeX 2e, que choisir ? 2
 \LaTeX , logo 113
 \LaTeX 2e logo 113
 Lettre grecques 85
 lettre ij, Holandais 117
 lettres calligraphiques pour les
 mathématiques 23
 lettres de script pour les mathématiques 23
 lettres latines non diacritées 117
 lettres, écrire des 122
 lettres, commencer 125
 lettres, salutation 124

ligne de commande 128
 lignes dans des tableaux 55
 liste des figures, fichier 4
 liste des personnes en copie, dans une lettre ... 123
 liste des tableaux, fichier 4
 listes à puces 44
 listes d'articles 44
 listes d'articles, générique 46
 listes de description, créer 38
 listes non ordonnées 44
 lists étiquetée, créer 38
 lists of items, numbered 40
 livre, symbole 113
 logo \LaTeX 113
 logo \LaTeX 2e 113
 logo \TeX 113
 longueur 80
 longueur, ajouter à une 81
 longueur, commande de 80
 longueur, régler une 80
 longueurs prédéfinies 81
 longueurs, définir de nouvelles 71
 longueurs, définir et utiliser 80
 Lua \TeX 5

M

m, largeur d'un 80
 macron, diacritique 116
 macron-souscrit, diacritique 116
 Madsen, Lars 41
`makeindex`, programme 121
 marque déposée, symbole 115
 marque de fabrique, symbole 115
 masculin, symbole ordinal 114
 math, accent aigu 97
 math, accent barre suscrite 97
 math, accent brève 97
 math, accent grave 97
 math, accent macron 97
 math, accent point suscrit 97
 math, accent rond-en-chef 98
 math, accents 97
 math, fonctions 96
 math, miscellanées 98
 mathématiques gras 23
 mathématiques, gras 23
 matières, fichier 4
 Millimètre 80
 minipage, créer une 48
 miscellanées en math 98
 mise en page, les paramètres de 27
 modèle de document 129
 mode gauche-à-droite (left-to-right) 101
 mode LR 101
 mode math 101
 mode mathématique, entrer 84
 mode mathématique, espacement 98

mode paragraphe	101, 110
modes	101
monnaie, dollar	114
monnaie, euro	114
moteur pdf \TeX	4
moteurs, \TeX	4
mpfootnote, compteur	67
mu, unité mathématique	80
multiplication discrétionnaire	98

N

note en bas de page, créer	67
note en bas de page, symboles au lieu de numéros	67
notes dans la marge	82
notes en bas de page au sein des figures	48
Notes en bas de page, au sein d'une minipage	67
notes en bas de page, dans les titres de rubrique	69
Notes en bas de page, dans un tableau	68
notes en bas de page, paramètres	69
Notes en marge	82
nouvelle commande, définition	16
nouvelle commande, vérification	15
nouvelle commandes, définir	70
nouvelle ligne, commencer une	63
nouvelle ligne, commencer une (mode paragraphe)	63
nouvelle page, commencer	66
nouvelles classes, commandes	13
numéro d'équation, omettre	41
numéro d'équation, référence croisée	34
numéro de figure, référence croisée	34
numéro de note en bas de page, référence croisée	34
numéro de page, référence croisée	34
numéro de rubrique, référence croisée	34
numéros de rubrique, composition	32
numérotation de page, style	103

O

o e liés	118
o, e dans l'o	118
obèle double, dans le texte	114
obèle, dans le texte	112, 114
officieuse, nature de ce document	2
ogonek, diacritique	116
option <code>stable</code> du paquetage <code>footmisc</code>	69
options de classe	12
options de classe de document	12
options globales	12, 13
options pour classe de document	14
options pour paquetage	14
options, class	16
options, classe de document	12
options, package	16

Ordinaux espagnols, féminin et masculin	114
Ordinaux, féminin et masculin	114
oslash	118
ouvrant, , guillemet-apostrophe simple	113

P

package options	16
page de flottants	29
page styles	103
page, paramètres de mise en page	27
pages de titre, création	61
pages, saut	66
paquet de macro, \LaTeX en tant que	3
paquetage, disposition du fichier	14
paquetage, options	14
paquetages, le chargement	13
paragraphe, mode	110
paragraphe, symbole	113
paragraphe, symbole	82
paramètre positionnel	70, 72
paramètres, la mise en page	27
paramètres, pour notes en bas de page	69
partitionner le fichier d'entrée	119
patron de document	129
patron, <code>beamer</code>	129
patron, <code>book</code>	129
patron, TUGboat	130
pdf \TeX	4
pdf \TeX , moteur	4
pièces jointes, liste	124
pica	80
Pied-de-mouche	113
placement des flottants	29
poésie, un environnement pour la	62
Point	80
point centré, dans le texte	114
point d'exclamation culbuté	114
point d'interrogation culbuté	115
point Didot	80
point en chef, diacritique	116
Point proportionné (Scaled point)	80
point suscrit, diacritique	116
points de suspension	113
police de largeur fixe	23
police inclinée	23
police machine à écrire	23
police oblique	23
police petites capitales	23
police romaine	23
police, symboles par leur numéro	112
polices de caractères	22
polices OpenType	4
polices TrueType	4
polices, commandes pour déclarer de nouvelles	76
position, dans un dessin <code>picture</code>	48
postscriptum, dans une lettre	125

préambule, définition	3
prédéfinies, longueurs	81
prononciation	3
puce, dans le texte	114

Q

quad	98
------------	----

R

référence bibliographique, clef	60
référence croisée avec numéro de page	34
référence croisée, symbolique	34
référence déclarée en aval	33
références croisées	33
réglage de compteurs	79
réservés, caractères	112
résumé	36
racine, fichier	119
rapports d'anomalies	2
rapports de bogues	2
redefining environnements	72
remarks in the marge	82
renforcement des paragraphes	
dans du texte cité	52
renforcement des paragraphes dans un	
texte cité, omettre	52
Renforcement des paragraphes, au	
sein d'une minipage	48
renforcement, forcer	82
renforcement, supprimer	82
renvoi en aval, résolution	4
renvoi, résolution	4
robuste, commande	76
rond en chef, diacritique	116
rond, grand symbole, dans le texte	113
rubricage	32

S

s long	118
sans serif	23
saut à la ligne	63
saut de ligne	63
saut de ligne, forcer	63
saut de ligne, sortie semblable à	
l'entrée pour le	63
saut de page	66
saut de page, empêcher	66
saut de page, forcer	66
sauter à la ligne	63
sauts de ligne, empêcher	65
sauts de ligne, forcer	65
section, symbole	113
signes diacritiques	115
simple droit, guillemet-apostrophe	115
simple, guillemet-apostrophe	115

simple, guillemet-apostrophe culbuté	115
simuler du texte tapuscrit	62
sortie à deux colonnes	26
sortie à une colonne	26
souligné, dans le texte	115
source logiciel, composer du	62
spéciaux, caractères	112
spécificateur, placement de flottants	29
speciaux, caractères	117
stable , option du paquetage footmisc	69
sterling, symbole	113
style de numérotation de page	103
styles de police	22
styles de texte	22
styles typographiques	22
styles, page	103
supérieur à, symbole dans le texte	114
suspension, points de	113
symbole de multiplication, saut de	
ligne discrétionnaire	98
symbole droit d'auteur	112
symbole marque déposée	115
symbole marque de fabrique	115
symbole puce	86
symbole vecteur en math	98
symboles d'une police, par leur numéro	112
symboles mathématiques	85
symboles texte	112
symbols, mathématiques	85

T

table des matière, création	120
table des matières, éviter les notes	
en bas de page	69
table des matières, fichier	4
table of contents entry, manually adding	120
tableaux mathématiques	37
tableaux, créer	54
taille de conception, au sein d'une	
définition de police	76
tailles de police	24
tailles de texte	24
tailles des polices de caractères	24
taquets des tabulation, utiliser	53
T _E X, logo	113
texte cité avec renforcement des	
paragraphes, affichage	52
texte cité sans renforcement des	
paragraphes, affichage	52
texte drapeau droit	44
texte drapeau droit, environnement pour	44
texte drapeau gauche	44
texte drapeau gauche, environnement pour	44
texte fer à gauche	44
texte fer à gauche, environnement pour	44
texte ferré à gauche	44
texte ferré à gauche, environnement pour	44

texte sur plusieurs colonnes	26
texte tapuscrit, simulation de	62
texte verbatim	62
texte verbatim, en-ligne	62
théorème, environnement simili-	74
théorèmes, composition des	61
théorèmes, définition	74
thickspace	98
thinspace	98
thorn, lettre islandaise	118
tilde ASCII, dans le texte	113
tilde, diacritique	116
tirant suscrit, diacritique	117
tiret bas, dans le texte	115
tréma hongrois, accent	116
tréma hongrois, diacritique	116
trois-quarts, cadratin	115

U

unités, de longueur	80
UTF-8	4
utilisation de Bib _T E _X	61

V

variables, a list of	78
verticale infinie, dilatation	107
verticale, barre dans le texte	113
verticale, espace	107
visible, symbole espace, dans le texte	115

W

wide hat accent, math	98
wide tilde accent, math	98

X

x, hauteur d'un	80
Xe _T E _X	5
xindy, programle	121

Index des commandes

\$	
\$	84
&	
&	55
.	
.aux fichier	4
.fd fichier	76
.lof fichier	4
.lot fichier	4
.pdf fichier	4
.toc fichier	4
.toc, fichier	120
.xdv fichier	5
[
[...] pour les arguments optionnels	5
^	
^	84
-	
-	84
\	
\ caractère de début des commandes	5
\!	98
\" (accent tréma)	115
\#	112
\\$	112
\%	112
\&	112
\' (acute accent)	116
\' (tabbing)	53
\(.....	84
\)	84
*	98
\+	53
\,	98
\-	53
\- (césure)	64
\. (dot-over accent)	116
\/	105
\:	98
\;	98
\<	53
\= (macron accent)	116
\= (tabbing)	53
\>	53, 98
\> (tabbing)	53
\@fnsymbol	67
\@ifstar	10
\[.....	84
\]	84
\^	112
\^ (accent circonflexe)	116
_	112
\` (accent grave)	116
\` (tabbing)	53
\ (for eqnarray)	41
\ (pour center)	38
\ (pour flushright)	44
\ (pour les objets \shortstack)	52
\ (tabbing)	53
\ for flushleft	44
\ force un saut de ligne	63
\ pour les lettres	122
\ pour tabular	55
\ pour verse	62
\ * (pour eqnarray)	41
\{	112
\}	112
\ 	85
\~	112
\~ (accent tilde)	116
\a (tabbing)	53
\a' (acute accent in tabbing)	53
\a= (macron accent in tabbing)	53
\a' (grave accent in tabbing)	53
\aa (å)	117
\AA (Å)	117
\acute	97
\addcontentsline{ext}{unit}{texte}	120
\address	123
\addtocontents{ext}{texte}	120
\addtocounter	79
\addtolength	81
\addvspace	106
\ae (æ)	117
\AE (Æ)	117
\aleph	85
\alinea	32
\alph	78
\Alph	78
\Alph example	41
\alpha	85
\alsiname	121
\amalg	85
\angle	85
\appendix	32
\approx	85
\arabic	78

<code>\arccos</code>	96	<code>\capitaldotaccent</code>	116
<code>\arcsin</code>	96	<code>\capitalgrave</code>	116
<code>\arctan</code>	96	<code>\capitalhungarumlaut</code>	116
<code>\arg</code>	96	<code>\capitalmacron</code>	116
<code>\arraycolsep</code>	37	<code>\capitalnewtie</code>	117
<code>\arrayrulewidth</code>	57	<code>\capitalogonek</code>	116
<code>\arraystretch</code>	57	<code>\capitalring</code>	116
<code>\ast</code>	85	<code>\capitaltie</code>	117
<code>\asymp</code>	85	<code>\capitaltilde</code>	116
<code>\AtBeginDocument</code>	39	<code>\caption</code>	42, 54
<code>\AtBeginDvi</code>	15	<code>\cc</code>	123
<code>\AtEndDocument</code>	40	<code>\cdot</code>	86
<code>\AtEndOfClass</code>	15	<code>\cdots</code>	99
<code>\AtEndOfPackage</code>	15	<code>\centering</code>	38
<code>\b</code> (accent barre-souscrite)	116	<code>\chapter</code> (seulement pour classes report et book)	32
<code>\backslash</code>	85	<code>\check</code>	97
<code>\bar</code>	97	<code>\CheckCommand</code>	15
<code>\baselineskip</code>	25	<code>\CheckCommand*</code>	15
<code>\baselinestretch</code>	25	<code>\chi</code>	86
<code>\begin</code>	36	<code>\circ</code>	86
<code>\beta</code>	85	<code>\circle</code>	49
<code>\bf</code>	23	<code>\cite</code>	60
<code>\bfseries</code>	22	<code>\ClassError</code>	15
<code>\bibitem</code>	60	<code>\ClassInfo</code>	15
<code>\bibliography</code>	61	<code>\ClassInfoNoLine</code>	15
<code>\bibliographystyle</code>	61	<code>\ClassWarning</code>	15
<code>\bigcap</code>	85	<code>\ClassWarningNoLine</code>	15
<code>\bigcirc</code>	85	<code>\cleardoublepage</code>	66
<code>\bigcup</code>	85	<code>\clearpage</code>	66
<code>\bigodot</code>	85	<code>\cline</code>	59
<code>\bigoplus</code>	85	<code>\closing</code>	124
<code>\bigotimes</code>	86	<code>\clubsuit</code>	86
<code>\bigskip</code>	107	<code>\columnsep</code>	26
<code>\bigskipamount</code>	107	<code>\columnseprule</code>	26
<code>\bigsqcup</code>	86	<code>\columnwidth</code>	26
<code>\bigtriangledown</code>	86	<code>\complement</code>	86
<code>\bigtriangleup</code>	86	<code>\cong</code>	86
<code>\biguplus</code>	86	<code>\contentsline</code>	120
<code>\bigvee</code>	86	<code>\coprod</code>	86
<code>\bigwedge</code>	86	<code>\copyright</code>	112
<code>\bmod</code>	96	<code>\cos</code>	96
<code>\boldmath</code>	84	<code>\cosh</code>	96
<code>\bot</code>	86	<code>\cot</code>	96
<code>\bottomfraction</code>	30	<code>\coth</code>	96
<code>\bottomnumber</code>	30	<code>\csc</code>	96
<code>\bowtie</code>	86	<code>\cup</code>	86
<code>\Box</code>	86	<code>\CurrentOption</code>	16
<code>\breve</code>	97	<code>\d</code> (dot-under accent)	116
<code>\bullet</code>	86	<code>\dag</code>	112
<code>\c</code> (cédille)	116	<code>\dagger</code>	86
<code>\cal</code>	23	<code>\dashbox</code>	50
<code>\cap</code>	86	<code>\dashv</code>	87
<code>\capitalacute</code>	116	<code>\day</code>	79
<code>\capitalbreve</code>	117	<code>\dblfloatpagefraction</code>	26
<code>\capitalcaron</code>	117	<code>\dblfloatsep</code>	26
<code>\capitalcedilla</code>	116	<code>\dbltextfloatsep</code>	26
<code>\capitalcircumflex</code>	116	<code>\dbltopfraction</code>	26
<code>\capitaldieresis</code>	115		

<code>\ddag</code>	112	<code>\flushbottom</code>	27
<code>\ddagger</code>	87	<code>\fnsymbol</code>	78
<code>\ddot</code>	97	<code>\fnsymbol</code> , et note en bas de page.....	67
<code>\ddots</code>	99	<code>\fontencoding</code>	24
<code>\DeclareOption</code>	16	<code>\fontfamily</code>	24
<code>\DeclareOption*</code>	16	<code>\fontseries</code>	24
<code>\DeclareRobustCommand</code>	16	<code>\fontshape</code>	25
<code>\DeclareRobustCommand*</code>	16	<code>\fontsize</code>	25
<code>\deg</code>	96	<code>\footnote</code>	67
<code>\delta</code>	87	<code>\footnotemark</code>	67
<code>\Delta</code>	87	<code>\footnoterule</code>	69
<code>\depth</code>	81	<code>\footnotesep</code>	69
<code>\det</code>	96	<code>\footnotesize</code>	24
<code>\dh (ð)</code>	117	<code>\footnotetext</code>	68
<code>\DH (Ð)</code>	117	<code>\footskip</code>	27
<code>\Diamond</code>	87	<code>\forall</code>	87
<code>\diamond</code>	87	<code>\frac{num}{den}</code>	99
<code>\diamondsuit</code>	87	<code>\frame</code>	50
<code>\dim</code>	96	<code>\framebox</code>	50, 109
<code>\displaystyle</code>	84	<code>\frenchspacing</code>	105
<code>\div</code>	87	<code>\frown</code>	87
<code>\dj</code>	117	<code>\fussy</code>	65
<code>\DJ</code>	117	<code>\gamma</code>	87
<code>\documentclass</code>	12	<code>\Gamma</code>	87
<code>\dot</code>	97	<code>\gcd</code>	97
<code>\doteq</code>	87	<code>\ge</code>	87
<code>\dotfill</code>	106	<code>\geq</code>	87
<code>\dots</code>	113	<code>\gets</code>	88
<code>\doublerulesep</code>	58	<code>\gg</code>	88
<code>\downarrow</code>	87	<code>\glossary</code>	121
<code>\Downarrow</code>	87	<code>\glossaryentry</code>	121
<code>\ell</code>	87	<code>\grave</code>	97
<code>\em</code>	23	<code>\guillemotleft («)</code>	113
<code>\emph</code>	22	<code>\guillemotright (»)</code>	113
<code>\emptyset</code>	87	<code>\guilsinglleft (‹)</code>	113
<code>\encl</code>	124	<code>\guilsinglright (›)</code>	113
<code>\end</code>	36	<code>\H (tréma hongrois)</code>	116
<code>\enlargethispage</code>	66	<code>\hat</code>	97
<code>\enumi</code>	41	<code>\hbar</code>	88
<code>\enumii</code>	41	<code>\headheight</code>	27
<code>\enumiii</code>	41	<code>\headsep</code>	27
<code>\enumiv</code>	41	<code>\heartsuit</code>	88
<code>\epsilon</code>	87	<code>\height</code>	81
<code>\equiv</code>	87	<code>\hfill</code>	104
<code>\espace</code>	104	<code>\hline</code>	59
<code>\eta</code>	87	<code>\hom</code>	97
<code>\evensidemargin</code>	13	<code>\hookleftarrow</code>	88
<code>\ExecuteOptions</code>	18	<code>\hookrightarrow</code>	88
<code>\exists</code>	87	<code>\hrulefill</code>	106
<code>\exp</code>	97	<code>\hspace</code>	28
<code>\extracolsep</code>	56	<code>\hspace</code>	104
<code>\fbox</code>	109	<code>\Huge</code>	24
<code>\fboxrule</code>	50, 109	<code>\huge</code>	24
<code>\fboxsep</code>	50, 109	<code>\hyphenation</code>	65
<code>\fill</code>	104	<code>\i (i sans point)</code>	116
<code>\flat</code>	87	<code>\iff</code>	88
<code>\floatpagefraction</code>	30	<code>\IfFileExists</code>	17
<code>\floatsep</code>	30		

<code>\ij</code> (ij)	117	<code>\left delim1 ... \right delim2</code>	99
<code>\IJ</code> (IJ)	117	<code>\Leftarrow</code>	88
<code>\il</code>	23	<code>\leftarrow</code>	89
<code>\Im</code>	88	<code>\lefteqn</code>	41
<code>\imath</code>	98	<code>\leftharpoondown</code>	89
<code>\in</code>	88	<code>\leftharpoonup</code>	89
<code>\include</code>	119	<code>\leftmargin</code>	45
<code>\includeonly</code>	119	<code>\leftmargini</code>	45
<code>\indent</code>	82	<code>\leftmarginii</code>	45
<code>\index</code>	121	<code>\leftmarginiii</code>	45
<code>\indexentry</code>	121	<code>\leftmarginiv</code>	45
<code>\indexspace</code>	121	<code>\leftmarginv</code>	45
<code>\inf</code>	97	<code>\leftmarginvi</code>	45
<code>\infty</code>	88	<code>\leftrightarrow</code>	89
<code>\input</code>	119	<code>\Leftrightarrow</code>	89
<code>\InputIfFileExists</code>	17	<code>\leq</code>	89
<code>\int</code>	88	<code>\lfloor</code>	89
<code>\intertextsep</code>	30	<code>\lg</code>	97
<code>\iota</code>	88	<code>\lhd</code>	89
<code>\item</code>	38, 40, 44	<code>\lim</code>	97
<code>\itemindent</code>	45	<code>\liminf</code>	97
<code>\itemsep</code>	46	<code>\limsup</code>	97
<code>\itshape</code>	22	<code>\line</code>	50
<code>\j</code> (j sans point).....	116	<code>\linebreak</code>	65
<code>\jmath</code>	98	<code>\linespread</code>	25
<code>\Join</code>	88	<code>\linethickness</code>	51
<code>\k</code> (ogonek).....	116	<code>\linewidth</code>	27
<code>\kappa</code>	88	<code>\listoffigures</code>	120
<code>\ker</code>	97	<code>\listoftables</code>	120
<code>\kill</code>	54	<code>\listparindent</code>	45
<code>\l</code> (\l).....	117	<code>\ll</code>	89
<code>\L</code> (L).....	117	<code>\ln</code>	97
<code>\label</code>	33	<code>\lnot</code>	89
<code>\labelenumi</code>	41	<code>\LoadClass</code>	17
<code>\labelenumii</code>	41	<code>\LoadClassWithOptions</code>	17
<code>\labelenumiii</code>	41	<code>\location</code>	124
<code>\labelenumiv</code>	41	<code>\log</code>	97
<code>\labelitemi</code>	45	<code>\longleftarrow</code>	89
<code>\labelitemii</code>	45	<code>\longlefttrightarrow</code>	89
<code>\labelitemiii</code>	45	<code>\longmapsto</code>	89
<code>\labelitemiv</code>	45	<code>\longrightarrow</code>	89
<code>\labelsep</code>	45	<code>\lor</code>	89
<code>\labelwidth</code>	45	<code>\lq</code>	113
<code>\Lambda</code>	88	<code>\makebox</code>	109
<code>\lambda</code>	88	<code>\makebox (picture)</code>	49
<code>\land</code>	88	<code>\makeglossary</code>	121
<code>\langle</code>	88	<code>\makeindex</code>	121
<code>\Large</code>	24	<code>\makelabels</code>	124
<code>\large</code>	24	<code>\mapsto</code>	89
<code>\LARGE</code>	24	<code>\marginpar</code>	82
<code>\LaTeX</code>	113	<code>\marginparpush</code>	83
<code>\LaTeXe</code>	113	<code>\marginparsep</code>	83
<code>\lbrace</code>	88	<code>\marginparwidth</code>	83
<code>\lbrack</code>	88	<code>\mathbf</code>	23
<code>\lceil</code>	88	<code>\mathcal</code>	23
<code>\ldots</code>	113	<code>\mathdollar</code>	99
<code>\le</code>	88	<code>\mathellipsis</code>	99
<code>\leadsto</code>	88	<code>\mathnormal</code>	23

<code>\mathparagraph</code>	99	<code>\narrow</code>	90
<code>\mathring</code>	98	<code>\o (\emptyset)</code>	118
<code>\mathrm</code>	22	<code>\O (\emptyset)</code>	118
<code>\mathsection</code>	99	<code>\obeycr</code>	63
<code>\mathsf</code>	23	<code>\oddsidemargin</code>	13
<code>\mathsterling</code>	99	<code>\odot</code>	90
<code>\mathtt</code>	23	<code>\oe (\oe)</code>	118
<code>\mathunderscore</code>	99	<code>\OE (\OE)</code>	118
<code>\mathversion</code>	23	<code>\oint</code>	90
<code>\max</code>	97	<code>\oldstylenums</code>	23
<code>\mbox</code>	109	<code>\Omega</code>	90
<code>\mdseries</code>	22	<code>\omega</code>	90
<code>\medskip</code>	107	<code>\ominus</code>	90
<code>\medskipamount</code>	107	<code>\onecolumn</code>	26
<code>\medspace</code>	98	<code>\opening</code>	125
<code>\mho</code>	89	<code>\oplus</code>	90
<code>\mid</code>	90	<code>\OptionNotUsed</code>	18
<code>\min</code>	97	<code>\oslash</code>	90
<code>\models</code>	90	<code>\otimes</code>	90
<code>\month</code>	79	<code>\oval</code>	51
<code>\mp</code>	90	<code>\overbrace{math}</code>	99
<code>\mu</code>	90	<code>\overline{texte}</code>	99
<code>\multicolumn</code>	58	<code>\owns</code>	91
<code>\multipt</code>	51	<code>\PackageError</code>	15
<code>\nabla</code>	90	<code>\PackageInfo</code>	15
<code>\name</code>	125	<code>\PackageInfoNoLine</code>	15
<code>\natural</code>	90	<code>\PackageWarning</code>	15
<code>\ne</code>	90	<code>\PackageWarningNoLine</code>	15
<code>\nearrow</code>	90	<code>\pagebreak</code>	66
<code>\NeedsTeXFormat</code>	18	<code>\pagenumbering</code>	103
<code>\neg</code>	90	<code>\pageref</code>	34
<code>\neq</code>	90	<code>\paperheight</code>	27
<code>\newcommand</code>	70	<code>\paperwidth</code>	28
<code>\newcounter</code>	71	<code>\paragraph</code>	32
<code>\newenvironment</code>	72	<code>\parallel</code>	91
<code>\newfont</code>	76	<code>\parbox</code>	110
<code>\newlength</code>	71	<code>\parindent</code>	48, 82
<code>\newline</code>	63	<code>\parsep</code>	46
<code>\newpage</code>	66	<code>\parskip</code>	82
<code>\newsavebox</code>	72	<code>\parskip exemple</code>	46
<code>\newtheorem</code>	74	<code>\part</code>	32
<code>\newtie</code>	117	<code>\partial</code>	91
<code>\ng</code>	117	<code>\partopsep</code>	46
<code>\NG</code>	117	<code>\PassOptionsToClass</code>	18
<code>\ni</code>	90	<code>\PassOptionsToPackage</code>	18
<code>\nocite</code>	60	<code>\pdfpageheight</code>	12
<code>\nofiles</code>	120	<code>\pdfpagewidth</code>	12
<code>\noindent</code>	82	<code>\perp</code>	91
<code>\nolinebreak</code>	65	<code>\phi</code>	91
<code>\nonfrenchspacing</code>	105	<code>\pi</code>	91
<code>\nonumber</code>	41	<code>\Pi</code>	91
<code>\nopagebreak</code>	66	<code>\pm</code>	91
<code>\normalfont</code>	22	<code>\pmod</code>	97
<code>\normalmarginpar</code>	83	<code>\poptabs</code>	54
<code>\normalsize</code>	24	<code>\pounds</code>	113
<code>\not</code>	90	<code>\Pr</code>	97
<code>\notin</code>	90	<code>\prec</code>	91
<code>\nu</code>	90	<code>\preceq</code>	91

<code>\prime</code>	91	<code>\sc</code>	23
<code>\printindex</code>	121	<code>\scriptsize</code>	24
<code>\ProcessOptions</code>	19	<code>\scshape</code>	22
<code>\ProcessOptions*</code>	19	<code>\searrow</code>	92
<code>\prod</code>	91	<code>\sec</code>	97
<code>\propto</code>	91	<code>\section</code>	32
<code>\protect</code>	76	<code>\seename</code>	121
<code>\ProvidesClass</code>	20	<code>\selectfont</code>	25
<code>\ProvidesFile</code>	20	<code>\setcounter</code>	79
<code>\ProvidesPackage</code>	20	<code>\setlength</code>	80
<code>\ps</code>	125	<code>\setminus</code>	92
<code>\Psi</code>	91	<code>\settodepth</code>	81
<code>\psi</code>	91	<code>\settoheight</code>	81
<code>\pushtabs</code>	54	<code>\settowidth</code>	81
<code>\put</code>	51	<code>\sf</code>	23
<code>\P</code>	113	<code>\sffamily</code>	22
<code>\qquad</code>	98	<code>\sharp</code>	92
<code>\quad</code>	98	<code>\shortstack</code>	52
<code>\quotedblbase (,,)</code>	113	<code>\Sigma</code>	92
<code>\quotesinglbase (,)</code>	113	<code>\sigma</code>	92
<code>\r (ring accent)</code>	116	<code>\signature</code>	125
<code>\raggedbottom</code>	27	<code>\sim</code>	92
<code>\raggedleft</code>	44	<code>\simeq</code>	92
<code>\raggedright</code>	44	<code>\sin</code>	97
<code>\raisebox</code>	110	<code>\sinh</code>	97
<code>\rangle</code>	91	<code>\sl</code>	23
<code>\rbrace</code>	91	<code>\slshape</code>	22
<code>\rbrack</code>	91	<code>\small</code>	24
<code>\rceil</code>	91	<code>\smallint</code>	92
<code>\Re</code>	91	<code>\smallskip</code>	107
<code>\ref</code>	34	<code>\smallskipamount</code>	107
<code>\refstepcounter</code>	79	<code>\smile</code>	92
<code>\renewenvironment</code>	72	<code>\spadesuit</code>	93
<code>\RequirePackage</code>	21	<code>\sqcap</code>	93
<code>\RequirePackageWithOptions</code>	21	<code>\sqcup</code>	93
<code>\restorecr</code>	63	<code>\sqrt[nième]{arg}</code>	99
<code>\restriction</code>	91	<code>\sqsubset</code>	93
<code>\revemptyset</code>	92	<code>\sqsubseteq</code>	93
<code>\reversemarginpar</code>	83	<code>\sqsupset</code>	93
<code>\rfloor</code>	92	<code>\sqsupseteq</code>	93
<code>\rhd</code>	92	<code>\ss (ß)</code>	118
<code>\rho</code>	92	<code>\SS (SS)</code>	118
<code>\right</code>	99	<code>\stackrel{texte}{relation}</code>	100
<code>\Rightarrow</code>	92	<code>\star</code>	93
<code>\rightarrow</code>	92	<code>\stepcounter</code>	79
<code>\rightharpoondown</code>	92	<code>\stop</code>	128
<code>\rightharpoonup</code>	92	<code>\subset</code>	93
<code>\rightleftharpoons</code>	92	<code>\subseteq</code>	93
<code>\rightmargin</code>	46	<code>\subsubsection</code>	32
<code>\rm</code>	23	<code>\succ</code>	93
<code>\rmfamily</code>	22	<code>\succeq</code>	93
<code>\roman</code>	78	<code>\sum</code>	93
<code>\Roman</code>	78	<code>\sup</code>	97
<code>\rq</code>	113	<code>\suppressfloats</code>	30
<code>\rule</code>	118	<code>\supset</code>	93
<code>\sautdeligne</code>	104	<code>\supseteq</code>	93
<code>\savebox</code>	111	<code>\surd</code>	93
<code>\sbox</code>	111	<code>\swarrow</code>	93

<code>\symbol</code>	112	<code>\textrm</code>	22
<code>\S</code>	113	<code>\textsc</code>	22
<code>\t</code> (tie-after accent)	117	<code>\textsf</code>	22
<code>\tabbingsep</code>	54	<code>\textsl</code>	22
<code>\tabcolsep</code>	58	<code>\textsterling</code>	113
<code>\tableofcontents</code>	120	<code>\textthreequartersemdash</code>	115
<code>\tabulation</code>	104	<code>\texttrademark</code>	115
<code>\tan</code>	97	<code>\texttt</code>	22
<code>\tanh</code>	97	<code>\texttwelveudash</code>	115
<code>\tau</code>	93	<code>\textunderscore</code>	115
<code>\telephone</code>	126	<code>\textup</code>	22
<code>\textascendercompwordmark</code>	114	<code>\textvisiblespace</code>	115
<code>\textasciicircum</code>	113	<code>\textwidth</code>	28
<code>\textasciitilde</code>	113	<code>\TeX</code>	113
<code>\textasteriskcentered</code>	113	<code>\th</code> (þ)	118
<code>\textbackslash</code>	112, 113	<code>\TH</code> (Þ)	118
<code>\textbar</code>	113	<code>\theta</code>	93
<code>\textbardbl</code>	113	<code>\thicklines</code>	51
<code>\textbf</code>	22	<code>\thickspace</code>	98
<code>\textbigcircle</code>	113	<code>\thinlines</code>	51
<code>\textbraceleft</code>	114	<code>\thinspace</code>	98, 105
<code>\textbraceright</code>	114	<code>\thispagestyle</code>	103
<code>\textbullet</code>	114	<code>\tilde</code>	98
<code>\textcapitalcompwordmark</code>	114	<code>\times</code>	94
<code>\textcircled{lettre}</code>	114	<code>\tiny</code>	24
<code>\textcompwordmark</code>	114	<code>\to</code>	94
<code>\textcopyright</code>	112	<code>\today</code>	118
<code>\textdagger</code>	114	<code>\top</code>	94
<code>\textdaggerdbl</code>	114	<code>\topfraction</code>	30
<code>\textdollar</code> (ou <code>\\$</code>)	114	<code>\topmargin</code>	28
<code>\textellipsis</code>	113	<code>\topnumber</code>	31
<code>\textendash</code> (ou <code>--</code>)	114	<code>\topsep</code>	46
<code>\textendash</code> (ou <code>--</code>)	114	<code>\topskip</code>	28
<code>\texteuro</code>	114	<code>\totalheight</code>	81
<code>\textexclamdown</code> (ou <code>!</code>)	114	<code>\totalnumber</code>	31
<code>\textfloatsep</code>	30	<code>\triangle</code>	94
<code>\textfraction</code>	30	<code>\triangleleft</code>	94
<code>\textgreater</code>	114	<code>\triangleright</code>	94
<code>\textheight</code>	27	<code>\tt</code>	23
<code>\textit</code>	22	<code>\ttfamily</code>	22
<code>\textleftarrow</code>	114	<code>\twocolumn</code>	26
<code>\textless</code>	114	<code>\typein</code>	127
<code>\textmd</code>	22	<code>\typeout</code>	127
<code>\textnormal</code>	22	<code>\u</code> (breve accent)	117
<code>\textordfeminine</code>	114	<code>\unboldmath</code>	84
<code>\textordmasculine</code>	114	<code>\underbar</code>	117
<code>\textparagraph</code>	113	<code>\underbrace{math}</code>	100
<code>\textperiodcentered</code>	114	<code>\underline{texte}</code>	100
<code>\textquestiondown</code> (ou <code>?'</code>)	115	<code>\unitlength</code>	48
<code>\textquotedblleft</code> (ou <code>'</code>)	115	<code>\unlhd</code>	94
<code>\textquotedblright</code> (ou <code>'</code>)	115	<code>\unrhd</code>	94
<code>\textquoteleft</code> (ou <code>'</code>)	115	<code>\Uparrow</code>	94
<code>\textquoteright</code> (ou <code>'</code>)	115	<code>\uparrow</code>	94
<code>\textquotesingle</code>	115	<code>\updownarrow</code>	94
<code>\textquoteststraightbase</code>	115	<code>\Updownarrow</code>	94
<code>\textquoteststraightdblbase</code>	115	<code>\upharpoonright</code>	94
<code>\textregistered</code>	115	<code>\uplus</code>	95
<code>\textrightarrow</code>	115	<code>\upshape</code>	22

<code>\Upsilon</code>	95
<code>\upsilon</code>	95
<code>\usebox</code>	111
<code>\usecounter</code>	78
<code>\usefont</code>	25
<code>\usepackage</code>	13
<code>\v</code> (breve accent).....	117
<code>\value</code>	79
<code>\vanothing</code>	95
<code>\varepsilon</code>	95
<code>\varphi</code>	95
<code>\varpi</code>	95
<code>\varrho</code>	95
<code>\varsigma</code>	95
<code>\vartheta</code>	95
<code>\vdash</code> <code>\vdash</code> Taquet droit ; prouve, implique ;.....	95
<code>\vdots</code>	100
<code>\vec</code>	98
<code>\vector</code>	52
<code>\vee</code>	95
<code>\verb</code>	62
<code>\Vert</code>	95
<code>\vert</code>	96
<code>\vfill</code>	107
<code>\vline</code>	59
<code>\vspace</code>	107
<code>\wedge</code>	96
<code>\widehat</code>	98
<code>\widetilde</code>	98
<code>\width</code>	81
<code>\wp</code>	96
<code>\wr</code>	96
<code>\xi</code>	96
<code>\Xi</code>	96
<code>\year</code>	79
<code>\zeta</code>	96

{

{...} pour les arguments obligatoires 5

A

<code>abstract</code> , environnement.....	36
<code>amsmath</code> , paquetage.....	37, 39
<code>array</code> , environnement.....	37

B

<code>bp</code>	80
-----------------------	----

C

<code>cc</code>	80
<code>center</code> , environnement.....	38
classe <code>article</code>	12
classe <code>book</code>	12
classe <code>letter</code>	12
classe <code>report</code>	12
classe <code>slides</code>	12
<code>cm</code>	80
commande <code>dvipdfmx</code>	4
commande <code>dvips</code>	4
commande <code>dvitype</code>	4
commande <code>latex</code>	4
commande <code>pdflatex</code>	4
commande <code>xdvi</code>	4

D

<code>datetime</code> , paquetage.....	118
<code>dd</code>	80
<code>description</code> , environnement.....	38
<code>displaymath</code> , environnement.....	39, 84
<code>document</code> , environnement.....	39

E

<code>em</code>	80
<code>enumerate</code> , environnement.....	40
environnement <code>abstract</code>	36
environnement <code>array</code>	37
environnement <code>center</code>	38
environnement <code>description</code>	38
environnement <code>displaymath</code>	39, 84
environnement <code>document</code>	39
environnement <code>enumerate</code>	40
environnement <code>eqnarray</code>	41
environnement <code>equation</code>	42, 84
environnement <code>figure</code>	42
environnement <code>filecontents</code>	43
environnement <code>filecontents*</code>	43
environnement <code>flushleft</code>	44
environnement <code>flushright</code>	44
environnement <code>itemize</code>	44
environnement <code>letter</code>	46
environnement <code>list</code>	46
environnement <code>math</code>	47, 84
environnement <code>minipage</code>	48
environnement <code>picture</code>	48
environnement <code>quotation</code>	52
environnement <code>quote</code>	52
environnement <code>tabbing</code>	53
environnement <code>table</code>	54
environnement <code>tabular</code>	55
environnement <code>thebibliography</code>	60
environnement <code>theorem</code>	61
environnement <code>titlepage</code>	61
environnement <code>verbatim</code>	62
environnement <code>verse</code>	62

eqnarray, environnement 41
 equation, environnement 42, 84
 etex, commande 4
 etoolbox, paquetage 17
 ex 80

F

fichier .dvi 4
 fichier .log 4
 figure, environnement 42
 filecontents*, environnement 43
 filecontents, environnement 43
 first-latex-doc document 2
 flafter, paquetage 30
 float, paquetage 29
 flushleft, environnement 44
 flushright, environnement 44
 footmisc, paquetage 69

G

geometry, paquetage 12

H

<http://puszcza.gnu.org.ua/software/latexrefman/>
 page d'accueil 2

I

in 80
 itemize, environnement 44

L

latex-doc-ptr document 2
 latexrefman@tug.org adresse mél. 2
 letter, environnement 46
 list, environnement 46
 lR box 49
 lrbox 109
 lshort document 2
 lualatex commande 5

M

macros2e, paquetage 7
 makeidx, paquetage 121
 math, environnement 47, 84
 minipage, environnement 48
 mm 80
 mu 80
 multind, paquetage 121

O

option 10pt 12
 option 11pt 12
 option 12pt 12
 option a4paper 12
 option a5paper 12
 option b5paper 12
 option draft 12
 option executivepaper 12
 option final 12
 option fleqn 12
 option landscape 12
 option legalpaper 12
 option leqno 12
 option letterpaper 12
 option notitlepage 12
 option onecolumn 13
 option oneside 13
 option openany 13
 option openbib 12
 option openright 13
 option titlepage 12
 option twocolumn 13
 option twoside 13

P

paquetage amsmath 37, 39
 paquetage datetime 118
 paquetage etoolbox 17
 paquetage flafter 30
 paquetage float 29
 paquetage footmisc 69
 paquetage geometry 12
 paquetage macros2e 7
 paquetage makeidx 121
 paquetage multind 121
 paquetage picture 48
 paquetage setspace 25
 paquetage showidx 121
 paquetage textcomp 23
 paquetage xspace 105
 pc 80
 picture, environnement 48
 picture, paquetage 48
 pouce (inch) 80
 pt 80

Q

quotation, environnement 52
 quote, environnement 52

S

<code>secnumdepth</code> counter.....	32
<code>setspace</code> , paquetage	25
<code>showidx</code> , paquetage	121
<code>sp</code>	80

T

<code>tabbing</code> , environnement	53
<code>table</code> , environnement	54
<code>tabular</code> , environnement	55
<code>textcomp</code> , paquetage.....	23, 112
<code>thebibliography</code> , environnement.....	60
<code>theorem</code> , environnement	61
<code>titlepage</code> , environnement.....	61

U

<code>usrguide</code> documentation officielle	2
--	---

V

<code>verbatim</code> , environnement.....	62
<code>verse</code> , environnement	62

X

<code>xdvipdfmx</code>	5
<code>xelatex</code> commande	5
<code>xspace</code> , paquetage.....	105