

LaTeX: Un manuel de référence non officiel

Avril 2016

<http://home.gna.org/latexrefman>

Ce document est un manuel de référence officieux pour \LaTeX , un système de préparation de documents, version de Avril 2016.

Ce manuel a été traduit du fichier `LATEX.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 \LaTeX 2.09 a été rédigée par Stephen Gilmore. La version $\text{\LaTeX}2\epsilon$ 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 \LaTeX* , de Sheldon Green, et *\LaTeX Command Summary* (pour $\text{\LaTeX}2.09$) de L. Botway et C. Biemesderfer (publié par le \TeX Users Group en tant que *\TeX niques* numéro 10), il l'a utilisé en tant que matériel de référence (aucun texte n'était directement copié).

Copyright 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014 Karl Berry.

Droits d'auteur 1988, 1994, 2007 Stephen Gilmore.

Droits d'auteur 1994, 1995, 1996 Torsten Martinsen.

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.

Sommaire

L ^A T _E X2e	1
1 À propos de ce document	2
2 Vue d'ensemble de L ^A T _E X	3
3 Classes de documents	9
4 Polices de caractères	11
5 Mise en page	15
6 Des rubriques	21
7 des références croisées	22
8 Environments	23
9 Saut à la ligne	49
10 Saut de page	52
11 Note en bas de page	53
12 Définitions	56
13 Compteurs	64
14 Longueurs	66
15 Faire des paragraphes	68
16 Formules mathématiques	70
17 Modes	87
18 Styles de page	89
19 Espaces	90
20 Boîtes	94
21 Insertions spéciales	97
22 Partitionner le fichier d'entrée	104
23 Prologue et épilogue	105
24 Des lettres	107
25 Entrée/sortie console	112
26 Ligne de commande	113
A Patrons de documents	114
Index des concepts	117
Index des commandes	124

Table des matières

L^AT_EX2_ε	1
1 À propos de ce document	2
2 Vue d'ensemble de L^AT_EX	3
2.1 Début et fin	3
2.2 Fichiers de sortie	4
2.3 Les moteurs T _E X	4
2.4 L ^A T _E X command syntax	5
3 Classes de documents	9
3.1 Options de classe de document	9
4 Polices de caractères	11
4.1 styles des polices	11
4.2 tailles des polices	13
4.3 commandes de fontes de bas niveau	13
5 Mise en page	15
5.1 <code>\onecolumn</code>	15
5.2 <code>\twocolumn</code>	15
5.3 <code>\flushbottom</code>	16
5.4 <code>\raggedbottom</code>	16
5.5 Paramètres de mise en page	16
5.6 Floats	17
6 Des rubriques	21
7 des références croisées	22
7.1 <code>\label</code>	22
7.2 <code>\pageref{key}</code>	22
7.3 <code>\ref{clef}</code>	22
8 Environments	23
8.1 <code>abstract</code>	23
8.2 <code>array</code>	24
8.3 <code>center</code>	25
8.3.1 <code>\centering</code>	25
8.4 <code>description</code>	25

8.5	<code>displaymath</code>	26
8.6	<code>document</code>	26
8.7	<code>enumerate</code>	26
8.8	<code>eqnarray</code>	27
8.9	<code>equation</code>	28
8.10	<code>figure</code>	28
8.11	<code>filecontents</code> : Écrire un fichier externe	29
8.12	<code>flushleft</code>	30
8.12.1	<code>\raggedright</code>	30
8.13	<code>flushright</code>	30
8.13.1	<code>\raggedleft</code>	30
8.14	<code>itemize</code>	31
8.15	environnement <code>letter</code> : écrire des lettres postales	32
8.16	<code>list</code>	32
8.17	<code>math</code>	33
8.18	<code>minipage</code>	34
8.19	<code>picture</code>	34
8.19.1	<code>\circle</code>	35
8.19.2	<code>\makebox</code>	35
8.19.3	<code>\framebox</code>	36
8.19.4	<code>\dashbox</code>	36
8.19.5	<code>\frame</code>	36
8.19.6	<code>\line</code>	36
8.19.7	<code>\linethickness</code>	36
8.19.8	<code>\thicklines</code>	37
8.19.9	<code>\thinlines</code>	37
8.19.10	<code>\multipt</code>	37
8.19.11	<code>\oval</code>	37
8.19.12	<code>\put</code>	37
8.19.13	<code>\shortstack</code>	37
8.19.14	<code>\vector</code>	38
8.20	quotation et <code>quote</code>	38
8.21	<code>tabbing</code>	38
8.22	<code>table</code>	40
8.23	<code>tabular</code>	41
8.23.1	<code>\multicolumn</code>	44
8.23.2	<code>\cline</code>	45
8.23.3	<code>\hline</code>	45
8.23.4	<code>\vline</code>	45
8.24	<code>thebibliography</code>	45
8.24.1	<code>\bibitem</code>	46
8.24.2	<code>\cite</code>	46
8.24.3	<code>\nocite</code>	46
8.24.4	Utilisation de Bib $\text{T}_{\text{E}}\text{X}$	46
8.25	<code>theorem</code>	47
8.26	<code>titlepage</code>	47
8.27	<code>verbatim</code>	48
8.27.1	<code>\verb</code>	48

8.28	<code>verse</code>	48
9	Saut à la ligne	49
9.1	<code>\</code>	49
9.2	<code>\obeycr</code> & <code>\restorecr</code>	49
9.3	<code>\newline</code>	49
9.4	<code>\-</code> (césure à gré).....	50
9.5	<code>\discretionary</code> (césure à gré avec contrôle de l’union).....	50
9.6	<code>\fussy</code>	51
9.7	<code>\sloppy</code>	51
9.8	<code>\hyphenation</code>	51
9.9	<code>\linebreak</code> & <code>\nolinebreak</code>	51
10	Saut de page	52
10.1	<code>\cleardoublepage</code>	52
10.2	<code>\clearpage</code>	52
10.3	<code>\newpage</code>	52
10.4	<code>\enlargethispage</code>	52
10.5	<code>\pagebreak</code> & <code>\nopagebreak</code>	52
11	Note en bas de page	53
11.1	<code>\footnote</code>	53
11.2	<code>\footnotemark</code>	53
11.3	<code>\footnotetext</code>	54
11.4	Notes en bas de page dans un tableau.....	54
11.5	Note en bas de page dont le renvoi est au sein d’un titre de rubrique.....	55
11.6	Paramètres des notes en bas de page.....	55
12	Définitions	56
12.1	<code>\newcommand</code> & <code>\renewcommand</code>	56
12.2	<code>\newcounter</code>	57
12.3	<code>\newlength</code>	57
12.4	<code>\newsavebox</code>	58
12.5	<code>\newenvironment</code> & <code>\renewenvironment</code>	58
12.6	<code>\newtheorem</code>	60
12.7	<code>\newfont</code> : définit une nouvelle police (obsolète).....	61
12.8	<code>\protect</code>	62

13	Compteurs	64
13.1	<code>\alph \Alph \arabic \roman \Roman</code> <code>\fnsymbol</code> : Taper des compteurs	64
13.2	<code>\usecounter{compteur}</code>	64
13.3	<code>\value{compteur}</code>	64
13.4	<code>\setcounter{compteur}{value}</code>	65
13.5	<code>\addtocounter{compteur}{valeur}</code>	65
13.6	<code>\refstepcounter{compteur}</code>	65
13.7	<code>\stepcounter{compteur}</code>	65
13.8	<code>\day \month \year</code> : Prédefinies	65
14	Longueurs	66
14.1	Unités de longueur	66
14.2	<code>\setlength</code>	66
14.3	<code>\addtolength</code>	67
14.4	<code>\settodepth</code>	67
14.5	<code>\settoheight</code>	67
14.6	<code>\settowidth</code>	67
14.7	Longueurs prédéfinies	67
15	Faire des paragraphes	68
15.1	<code>\indent</code>	68
15.2	<code>\noindent</code>	68
15.3	<code>\parskip</code>	68
15.4	Notes en marge	68
16	Formules mathématiques	70
16.1	Indices inférieurs & supérieurs	70
16.2	Symboles mathématiques	71
16.3	Fonctions mathématiques	82
16.4	Accents mathématiques	83
16.5	Espacement en mode mathématique	84
16.6	Recueil de diverses commandes utiles en math	84
17	Modes	87
17.1	<code>\ensuremath</code>	87
18	Styles de page	89
18.1	<code>\maketitle</code>	89
18.2	<code>\pagenumbering</code>	89
18.3	<code>\pagestyle</code>	89
18.4	<code>\thispagestyle{style}</code>	89

19	Espaces	90
19.1	<code>\hspace</code>	90
19.2	<code>\hfill</code>	90
19.3	<code>\espace</code> et <code>\@</code>	90
19.4	<code>\</code> après une séquence de contrôle	91
19.5	<code>\frenchspacing</code> , rendre égaux les espacements inter-phrase et inter-mot	91
19.6	<code>\thinspace</code>	91
19.7	<code>\/</code> : insérer une correction italique	91
19.8	<code>\hrulefill</code> <code>\dotfill</code>	92
19.9	<code>\addvspace</code>	92
19.10	<code>\bigskip</code> <code>\medskip</code> <code>\smallskip</code>	92
19.11	<code>\vfill</code>	93
19.12	<code>\vspace</code>	93
20	Boîtes	94
20.1	<code>\mbox{texte}</code>	94
20.2	<code>\fbox</code> and <code>\framebox</code>	94
20.3	<code>lrbox</code>	94
20.4	<code>\makebox</code>	94
20.5	<code>\parbox</code>	95
20.6	<code>\raisebox</code>	95
20.7	<code>\savebox</code>	96
20.8	<code>\sbox{<i>\boxcmd</i>}{texte}</code>	96
20.9	<code>\usebox{<i>\boxcmd</i>}</code>	96
21	Insertions spéciales	97
21.1	Caractères réservés	97
21.2	Symboles appelés par leur position dans une police	97
21.3	Symboles en texte	97
21.4	Accents	100
21.5	Lettres latines supplémentaires	102
21.6	<code>\rule</code>	103
21.7	<code>\today</code>	103
22	Partitionner le fichier d'entrée	104
22.1	<code>\include</code>	104
22.2	<code>\includeonly</code>	104
22.3	<code>\input</code>	104
23	Prologue et épilogue	105
23.1	Tables des matières	105
23.1.1	<code>\addcontentsline</code>	105
23.1.2	<code>\addtocontents</code>	105
23.2	Glossaires	106
23.3	Index	106

24	Des lettres	107
24.1	<code>\address</code>	108
24.2	<code>\cc</code>	108
24.3	<code>\closing</code>	109
24.4	<code>\encl</code>	109
24.5	<code>\location</code>	109
24.6	<code>\makelabels</code>	109
24.7	<code>\name</code>	110
24.8	<code>\opening{texte}</code>	110
24.9	<code>\ps</code>	110
24.10	<code>\signature{texte}</code>	110
24.11	<code>\telephone</code>	111
25	Entrée/sortie console	112
25.1	<code>\typein[cmd]{msg}</code>	112
25.2	<code>\typeout{msg}</code>	112
26	Ligne de commande	113
Annexe A	Patrons de documents	114
A.1	Un patron beamer	114
A.2	Un patron book	114
A.3	Un patron tugboat	115
	Index des concepts	117
	Index des commandes	124

L^AT_EX2e

Ce document est un manuel de référence officieux pour L^AT_EX, un système de préparation de document, la version du Avril 2016. Il est destiné à couvrir L^AT_EX2e, qui a été la version standard de L^AT_EX pendant de nombreuses années.

1 À propos de ce document

Le système \LaTeX de préparation de document est mis en oeuvre sous forme de macro forfait pour le programme de composition \TeX de Donald E. Knuth. \LaTeX a été créé à l'origine par Leslie Lamport ; c'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 non officiel et n'a pas été examiné par les responsables \LaTeX . Ne pas envoyer des rapports de bugs ou rien d'autre sur ce document pour eux. Au lieu de cela, s'il vous plaît envoyer tous commentaires à latexrefman-discuss@gn.org.

La page d'accueil de ce document est <http://home.gna.org/latexrefman>. Cette page contient des liens vers la courant de sortie dans différents formats, sources, listes de diffusion, et d'autres infrastructure.

Bien sûr, il ya beaucoup, beaucoup d'autres sources d'information sur \LaTeX . Voici quelques-unes:

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

Deux pages de références recommandées à \LaTeX documentation.

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

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

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

Le guide pour les auteurs de documents conservés dans le cadre de \LaTeX ; là plusieurs autres.

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

Introduction à l' \TeX système, y compris \LaTeX .

2 Vue d'ensemble de L^AT_EX

Qu'est-ce que L^AT_EX?

L^AT_EX 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 L^AT_EX 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. should be formatted. L^AT_EX est mis en œuvre comme un ensemble de commandes liés s'interfaçant avec le programme de composition T_EX de Donald E. Knuth (le terme technique est que L^AT_EX est un *paquetage de macros* pour le moteur T_EX). L'utilisateur produit le document de sortie en donnant ce fichier d'entrée au moteur T_EX.

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

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

2.1 Début et fin

Les fichiers L^AT_EX ont une structure globale simple, avec un début et une fin standards. Voici un exemple « 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 L^AT_EX lui-même, et un grand nombre d'autres sont largement disponibles. Voir Chapitre 3 [Document classes], page 9.

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

Le code `\begin{document} ... \end{document}` est ce qu'on appelle un *environnement* ; l'environnement 'document' (et aucun autre) est obligatoire dans tous les documents L^AT_EX (voir Section 8.6 [document], page 26). L^AT_EX fournit lui-même beaucoup d'environnements, et bien plus encore sont définis séparément. Voir Chapitre 8 [Environments], page 23.

Les sections suivantes discutent de la façon de produire des PDF et d'autres formats de sortie à partir d'un fichier d'entrée L^AT_EX.

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` ou en `.pdf`.

- `.dvi` Si il est invoqué avec la commande système `latex`, alors il produit un fichier “Device Independent”¹ (`.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 Format”² `.pdf` avec `dvipdfmx`. Un grand nombre d'autres programmes utilitaires DVI sont disponibles (<http://mirror.ctan.org/tex-archive/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` La “fichier de transcription” ou fichier `.log` qui 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-T_EX 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-T_EX est lui-même disponible par la commande système `etex`, mais le langage du fichier d'entrée est T_EX (et le fichier produit est un `.dvi`).

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

lualatex Si L^AT_EX est invoqué avec la commande système `lualatex`, alors le moteur LuaT_EX est exécuté (<http://ctan.org/pkg/lualatex>). Ce programme permet que du code écrit dans le langage script Lua (<http://luatex.org>) interagisse avec la composition faite par T_EX. LuaT_EX traite nativement l'entrée en 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 XeT_EX est exécuté (<http://tug.org/xetex>). Comme LuaT_EX, XeT_EX 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. XeT_EX produit un fichier `.pdf` en sortie ; il ne prend pas en charge la sortie DVI.

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

D'autres variantes de L^AT_EX et T_EX existent, par ex. pour fournir une prise en charge supplémentaires du japonais des d'autres langages ([u]pT_EX, <http://ctan.org/pkg/ptex>, <http://ctan.org/pkg/uptex>).

2.4 L^AT_EX command syntax

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

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 [Environments], page 23, 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}
```

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.

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 utilisateur 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 q'un nom de commande peut aussi consister d'un catactère 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 une @.

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.

Des exemple d'utilisation sont donnés ci-après :

Synopsis :

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

Cette commande permet de redéfinir les commandes de rubricage. Par ex. le code source ci-dessous, si placé dans le préambule, redéfinit la commande de rubricage `\section` pour qu'elle centre le titre, le mette en gras, et ajuste l'espacement :

```
\makeatletter
\renewcommand\section{%
  \@startsection{section}% [\@startsection/name], page 7
  {1}% [\@startsection/level], page 7, : 0=part, 1=chapter, etc.
  {0pt}% [\@startsection/retrait], page 7
  {-3.5ex plus -1ex minus -.2ex}% [\@startsection/beforeskip], page 7
  {2.3ex plus .2ex}% [\@startsection/afterskip], page 7
  {\centering\normalfont\Large
   \bfseries}}% [\@startsection/style], page 8
\makeatother
```

La commande `\@startsection` prend six arguments.

<i>nom</i>	Nom de la commande de rubricage, par ex. <code>paragraph</code>
<i>niveau</i>	Profondeur de la commande de rubricage, par ex. 0 pour <code>part</code> , 1 pour <code>chapter</code> , 2 pour <code>section</code> . Cette valeur est comparée aux compteurs <code>secnumdepth</code> et <code>tocdepth</code> pour contrôler la numérotation des rubriques et la profondeur de la table des matières.
<i>retrait</i>	Renforcement du titre. Pour un renforcement nul, vous pouvez utiliser la macro <code>\z@</code> définie à <code>0pt</code> pour que le code soit plus efficace.
<i>avant</i>	Longueur dont la valeur absolue est la longueur de l'espace verticale à insérer avant le titre. Pour une bonne composition, utilisez une longueur élastique. Si en plus <i>avant</i> est négatif, alors le premier paragraphe suivant immédiatement le titre n'est pas renforcé.
<i>après</i>	Longueur. Lorsque elle est positive, il s'agit de l'espace verticale à insérer après le titre. Pour une bonne composition, utilisez dans ce cas une longueur élastique.

Lorsque elle est négative, alors le titre fait corps avec le paragraphe le suivant immédiatement, comme c'est le cas pour les rubriques `\paragraph` dans la classe de document `article`, et la valeur absolue d'`après` donne l'espace horizontale entre la fin du dernier mot du titre et le début du premier mot du paragraphe avec lequel il fait corps.

style Les commandes qui règlent le style du titre, par ex. `\bfseries` pour des caractères gras.

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 nom de *classe* \LaTeX encastés 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 (l'option par défaut est `letterpaper`) :

```
a4paper a5paper b5paper executivepaper legalpaper letterpaper
```

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-version 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.

Les paquetages ajoutés sont chargés comme ceci :

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

Pour spécifier plus d'un *paquo*, vous pouvez les séparer par une virgule, 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`.

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 L^AT_EX pour ce faire sont décrites ci-après.

4.1 styles des polices

Les commandes de styles suivantes sont prises en charge par L^AT_EX.

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 25. 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 9).

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 nombreux 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 forme de police. Les formes valides sont :

- `n` Droit (normal)
- `it` Italique
- `sl` Incliné (oblique)
- `sc` Petites capitales
- `ui` Italique droit
- `ol` Plan

Les deux dernières formes 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'espace 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'espace 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 "espace 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 9).

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 étirées.

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 32). Plus précisément, elle est inférieure à `\textwidth` par la somme de `\leftmargin` et `\rightmargin` (voir Section 8.14 [itemize], page 31). 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'`.

`\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 15).

Dans les listes (voir Section 8.16 [list], page 32), `\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 34,) ou `\parbox` (voir Section 20.5 [`\parbox`], page 95), 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 Floats

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 28) et `table` (voir Section 8.22 [table], page 40), mais vous pouvez créer une nouvelle classes avec le paquetage `float`.

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

Le placement des flottant 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 \LaTeX 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 flottant dans une classe doivent apparaître dans le même ordre séquentiel, tous les flottants suivant dans cette classe apparaissent aussi à la fin.

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

- `t` (pour Top) — au sommet d'une page de texte.
- `b` (pour Bottom) — au bas d'une page de texte. (Cependant, `b` n'est pas autorisé avec des flottants en pleine-largeur (`figure*`) en cas de sortie à double-colonne. Pour améliorer cela, on peut utiliser les paquetages `stfloats` ou `dblfloatfix`, mais voyez la discussion sur les avertissements dans la FAQ : <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=2colfloat>.
- `h` (pour Here) — à la position du texte où l'environnement `figure` apparaît. Cependant, `h` n'est pas autorisé en soi-même ; `t` est ajouté automatiquement. 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 spécificateur précédents ; pour ce flottant seulement, `LATEX` ignore les restrictions à la fois sur le nombre de flottants qui peuvent apparaître et les quantité 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 `LATEX` 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 `LATEX` étant `tbp` consiste à essayer toutes les localisations sauf celle de placer le flottant là où il apparaît dans le code source.

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

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

Voici les paramètres 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 maximale de page autorisée à être occupées par des flottants au bas de la page ; la valeur par défaut est `’.3’`.

`\floatpagefraction`

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

`\textfraction`

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

`\topfraction`

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

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

`\floatsep`

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

`\intextsep`

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

`\textfloatsep`

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

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

`\bottomnumber`

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

`\topnumber`

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

`\totalnumber`

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

L’article principal de FAQ T_EX en rapport avec les flottants <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=floats> contient des suggestions pour relâcher les paramètres par défaut de L^AT_EX de sorte à réduire les problèmes de flottants rejetés à la fin. Une

explication complète de l'algorithme de placement des floatant se trouve dans l'article de Frank Mittelbach « How to influence the position of float environments like figure and table in L^AT_EX? » <http://latex-project.org/papers/tb111mitt-float.pdf>.

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 65.)

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. »

7.1 `\label`

Synopsis :

```
\label{clef}
```

Une commande `\label` apparaissant dans le texte ordinaire crée une étiquette en attribuant à `clef` le numéro de l'unité sectionnel courante ; lorsque `\label` apparaît dans un environnement numéroté, elle attribue ce numéro à `clef`.

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

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

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

Ansï, une étiquette pour une figure devrait ressembler à `fig:narquois` ou à `fig.narquois`.

7.2 `\pageref{key}`

Synopsis :

```
\pageref{key}
```

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

7.3 `\ref{clef}`

Synopsis :

```
\ref{clef}
```

La commande `\ref` produit le numéro de l'unité sectionnelle, équation, note en bas de page, figure, . . . , de la commande correspondante `\label` (voir Section 7.1 [`\label`], page 22). Elle ne produit aucun texte, tel que le mot 'Section' ou 'Figure', juste le numéro lui-même sans plus.

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 9).

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 9) ; 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 guerrilla, 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
    \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 28). Les entrées dans chaque colonne sont séparées avec une esperluette (&). Les lignes sont terminées par une double contr’oblique (voir Section 9.1 [\\], page 49).

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 41, 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 utilisé 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 affecte 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 `<undefined>`, (l'absence d'article cause l'erreur \LaTeX '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'espace des listes, voir Section 8.14 [itemize], page 31.

8.5 displaymath

Synopsis :

```
\begin{displaymath}
math
\end{displaymath}
```

or

```
\[math\]
```

L'environnement `displaymath` (`\[...\]` est un synonyme) compose le texte *math* sur sa propre ligne, centré par défaut. L'option globale `fleqn` justifie les équations à gauche ; voir Section 3.1 [Document class options], page 9.

Aucun numéro d'équation n'est ajouté au texte de `displaymath` ; pour obtenir un numéro d'équation, utilisez l'environnement `equation` (voir Section 8.9 [equation], page 28).

8.6 document

L'environnement `document` entoure le corps d'un document. Il est obligatoire dans tout document \LaTeX . Voir Section 2.1 [Starting and ending], page 3.

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 31) et `description` (voir Section 8.4 [description], page 25). Le format de l'étiquette produite dépend du niveau d'imbrication de la liste. Voici les valeurs par défaut de L^AT_EX 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 64). Si vous utilisez l'argument optionnel d'`\item` alors le compteur n'est pas incrementé 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 56). 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 64.

Pour plus ample information sur la personnalisation de la forme voir Section 8.16 [list], page 32. De même, le paquetage `enumitem` est utile pour cela.

8.8 eqnarray

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

Néanmoins, voici une description d'`eqnarray` :

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

L'environnement `eqnarray` est utilisé pour afficher une séquence d'équations ou d'inégalités. Il est similaire à un environnement `array` à trois colonnes, avec des lignes 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{equation*} ... \end{equation*}`) omet la numérotation des équations entièrement, tout en faisant par ailleurs la même chose qu'`equation`.

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

8.9 equation

Synopsis :

```
\begin{equation}
  math
\end{equation}
```

L'environnement `equation` commence un environnement en `displaymath` (voir Section 8.5 [`displaymath`], page 26), par ex. en centrant le texte `math` sur la page, et aussi en plaçant un numéro d'équation dans la marge de droite.

8.10 figure

Synopsis :

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

ou

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

Une classe de flottant (voir Section 5.6 [`Floats`], page 17). 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 17.

La version étoilée `figure*` est utilisée quand un document est en mode double-colonne (voir Section 5.2 [`twocolumn`], page 15). 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 17.

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 22). La commande `\caption` spécifie la légende *texte* pour la figure. La légende est numérotée par défaut. Si *titreldf* est présent, il est utilisé dans la liste des figures au lieu de *texte* (voir Section 23.1 [Tables of contents], page 105).

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 crée 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 référence 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 `as quote` ou d'une `parbox`.

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

8.14 itemize

Synopsis :

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

L'environnement `itemize` produit une liste « non ordonnée », « à puces ». Les environnements `itemize` peuvent être imbriqués l'un dans l'autre, jusqu'à quatre niveaux de profondeur. Ils peuvent aussi être imbriqués avec d'autres environnements fabriquant des paragraphes, comme `enumerate` (voir Section 8.7 [enumerate], page 26).

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 `\leftmarginiv` 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 68):

```
\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 107.

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}{étiquettage}{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 16).

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 70.

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 95), mais contrairement à `\parbox`, d'autres environnements de production de paragraphe peuvent être utilisés au sein d'une `minipage`.

Les arguments sont les même que pour `\parbox` (voir Section 20.5 [`\parbox`], page 95).

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 64).

Cependant, ne mettez pas une `minipage` à l'intérieur d'une autre si vous utilisez les note 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 chose dans le dessin en spécifiant leur coordonnées. Une coordonnée est un nombre qui peut avoir une séparateur décimal (point et non virgule) et un signe moins — un nombre comme 5, 0.3 ou -3.1416. Une coordonnées 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.

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 spécifiée 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 Note that when a position appears as an argument, elles 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 94) 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 sections suivantes.

8.19.1 `\circle`

Synopsis :

```
\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 94.

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` environnement 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 36, et Section 8.19.9 [`\thinlines`], page 37.

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 36, et Section 8.19.8 [`\thicklines`], page 37.

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 environnement, 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 paragraphe 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 elastique `\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 meieux 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 suivantes peuvent être utilisées 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 100). 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 function 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 17). Parce qu'ils ne peuvent pas être coupés à cheval sur plusieurs pages, ils 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 17.

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 22). 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 105).

```
\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}
  \label{tab:VertusCardinales}
  \caption{Vertus cardinales}
\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 colonne. 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 `with @{\extracolsep{\fill}}`, de sorte à permettre au tableau de se dilater ou contracter pour faire la largeur spécifiée, sans quoi vous aurez probablement l'avertissement `Underfull \hbox (badness 10000) in alignment ...`
- pos** Optionnel. Spécifie la position verticale du tableau. La valeur par défaut est d'aligner le tableau de sorte à ce que son centre vertical s'accorde à la ligne de base du texte autour. Il y a deux autres alignements possible : `t` aligne le tableau de sorte que la première ligne s'accorde à la ligne de base du texte autour, et `b` fait la même chose pour la dernière ligne du tableau. Ceci n'a un effet que si il y a du texte autour de la table. Dans le cas usuel d'un `tabular` seul au sein d'un environnement `center` cette option ne fait aucune différence.
- cols** Obligatoire. Spécifie le formatage des colonnes. Il consiste en une séquence des spécificateurs suivants, correspondant à la séquence des colonnes et du matériel intercolonne.
- 1 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 62).

Ce spécificateur est optionnel : à moins que vous mettiez dans votre propre @-expression alors les classe `book`, `article` et `report` de L^AT_EX mette 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 adjacente sont séparée 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, commande `@{\hspace{1em}}`.

Une expression vide `@{}` élimine l'espace, y compris l'express au début ou à la fin, commande dans l'exemple ci-dessous où on veut que les ligne du tableau soient alignée 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, arangé 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}{1@{\ \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 L^AT_EX prendra la double contr'oblique comme une fin de ligne.

`*{num}{cols}`

Équivalent à *num* copies de *cols*, où *num* est un entier positif et *cols* est toute liste de spécificateurs. Ainsi `\begin{tabular}{|*{3}{l|r}|}` équivaut à `\begin{tabular}{|l|r|l|r|}`. Notez que *cols* peut contenir une autre expression en `*{...}{...}`.

Paramètres qui contrôlent le formatage :

`\arrayrulewidth`

Une longueur qui est l'épaisseur du filet créé par `|`, `\hline`, et `\vline` au sein des environnements `tabular` et `array`. La valeur par défaut est `'0.4pt'`. On peut la changer comme dans `\setlength{\arrayrulewidth}{0.8pt}`.

`\arraystretch`

Un facteur par lequel l'espacement entre les lignes au sein des environnements `tabular` et `array` est multiplié. La valeur par défaut est `'1'` pour aucune dilatation/contraction. On peut la changer comme dans `\renewcommand{\arraystretch}{1.2}`.

`\doublerulesep`

Une longueur qui est la distance ente les filets verticaux produit par le spécificateur `||`. La valeur par défaut est `'2pt'`.

`\tabcolsep`

Une longueur qui est la moitié de l'espace entre les colonnes. La valeur par défaut est `'6pt'`. On peut la changer avec `\setlength`

Les commandes suivantes peuvent être utilisées à l'intérieur du corps d'un environnement `tabular` :

8.23.1 `\multicolumn`

Synopsis :

```
\multicolumn{nbrecols}{patron}{texte}
```

Fabrique une entrée de `array` ou `tabular` fusionnée sur plusieurs colonnes. Le premier argument, *nbrecols* spécifie le nombre de colonnes sur lesquelles s'étend la fusion. Le second argument, *patron*, est obligatoire et spécifie le format de l'entrée ; notamment `c` pour centré, `l` pour aligné à gauche, `r` pour aligné à droite. Le troisième argument, *texte* spécifie le texte à mettre dans l'entrée.

Dans l'exemple suivant les trois première colonnes sont fusionnées en le titre unique 'Nom'.

```
\begin{tabular}{lcc|}
  \textit{ID}      & \multicolumn{2}{c}{\textit{Nom}} & \textit{Âge} \\ \hline % ligne 1
  978-0-393-03701-2 & 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 parties sont `|r|`, `c`, `c`, et `p{1.5in}|`.

L'argument *patron* l'emporte sur la spécification par défaut de zone inter-colonne de l'environnement `array` ou `tabular` contiguë à cette entrée multi-colonne. Pour affecter cette zone, cet argument peut contenir des barres verticales `|` indiquant le placement de filets verticaux, et d'expression `@{...}`. Ainsi si *patron* est `|c|` alors l'entrée multi-colonne est centrée et un filet vertical la précède et la suit dans les espaces 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 vertical en sortie parce que le `\multicolumn` a le spécificateur `r` dans *patron* sans barres verticales initiales. Entre les entrées un et deux il y a un filet vertical ; en effet bien que le premier *patron* ne se termine pas par une barre verticale, le second *patron* en a une à son commencement. Entre la deuxième et la troisième entrée il y a un seul filet vertical ; bien que le *patron* dans les deux `\multicolumn` correspondant demandent un filet vertical, vous n'en obtenez qu'un seul. Entre les entrées trois et quatre il n'y a aucun filet vertical ; la spécification par défaut en demande un mais le *patron* dans le `\multicolumn` de la troisième entrée n'en demande pas, et cela l'emporte. Finalement, à la suite de la quatrième entrée il y a un filet vertical à cause de la spécification par défaut.

Le nombre de colonnes fusionnées *nbrecols* peut être 1. En plus de donner la possibilité de changer l'alignement horizontal, ceci est aussi utile pour l'emporter pour l'une des lignes sur la définition `tabular` par défaut de la zone inter-colonne, y compris concernant le placement des filets verticaux.

Dans l'exemple ci-dessous, dans la définition `tabular` de la première colonne on spécifie par défaut de justifier, mais au sein de la première ligne l'entrée est centrée avec `\multicolumn{1}{c}{\textsc{Période}}`. De même au sein de la première ligne, les deuxième et troisième colonnes sont fusionnées entre elle avec `\multicolumn{2}{c}{\textsc{Intervalle}}`, l'emportant sur la spécification pour centrer ces deux colonnes sur le trait d'union au sein de la plage de dates.

```
\begin{tabular}{l|r@{-}l}
  \multicolumn{1}{c}{\textsc{Période}}
    &\multicolumn{2}{c}{\textsc{Dates}} \\ \hline
  Baroque          &1600          &1760          \\
  Classique        &1730          &1820          \\
  Romantique       &1780          &1910          \\
  Impressioniste   &1875          &1925          \\
\end{tabular}
```

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 46) 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 46), 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 45). 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://www.ctan.org/tex-archive/biblio/bibtex/contrib>.

La commande `\bibliography` est ce qui produit pour de bon la bibliographie. L'argument de `\bibliography` fait référence à des 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 la possibilité de numérotation pour *n* sont décrites par `\newtheorem` (voir Section 12.6 [`\newtheorem`], page 60).

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 inscrits 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 103).

Vous pouvez utiliser la commande `\maketitle` (voir Section 18.1 [`\maketitle`], page 89) 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 tuez ; 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 renforcées sur la gauche et la droite, les paragraphes ne le sont pas, et le texte n’est pas justifié. Séparez le ligne de chaque strophe avec `\\`, et utilisez une ou plusieurs lignes vides pour séparer les strophes.

9 Saut à la ligne

La première chose que L^AT_EX fait lorsqu'il traite du texte ordinaire est de traduire votre fichier d'entrée en une séquence de glyphes et d'espaces. Pour produire un document imprimé, cette séquence doit être rompue en lignes (et ces lignes doivent être rompues en pages).

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

9.1 `\`

Synopsis :

```
\[espaceenplus]
```

ou

```
\*[espaceenplus]
```

Commencer une nouvelle ligne. L'argument optionnel *espaceenplus* spécifie l'espace supplémentaire vertical à insérer avant la ligne suivante. Ceci peut être une longueur négative. Le texte avant le saut est composé à sa longueur normale, c.-à-d. qu'il n'est pas dilaté pour remplir la largeur de la ligne.

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

Dans des circonstances ordinaires (par ex. en dehors d'une colonne spécifiée par `p{...}` dans un environnement `tabular`) la commande `\newline` est un synonyme de `\` (voir Section 9.3 [`\newline`], page 49).

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

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

9.2 `\obeycr` & `\restorecr`

La commande `\obeycr` a pour effet qu'un retour chariot dans le fichier d'entrée ('`^M`', internement) soit traité de la même façon que `\` (suivi d'un `\relax`). Ainsi chaque passage à la ligne dans l'entrée est aussi un passage à la ligne dans la sortie.

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

9.3 `\newline`

Dans du texte ordinaire ceci est équivalent à une double-contr'oblique (voir Section 9.1 [`\`], page 49) ; 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 \LaTeX qu'il peut faire une césure du mot à cet endroit. \LaTeX est très bon en ce qui concerne les césures, et il trouve la plupart des points corrects de césure, tout en n'en utilisant presque jamais un incorrect. La commande `\-` est utilisée pour les cas exceptionnels.

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

9.5 \discretionary (césure à gré avec contrôle de l'union)

Synopsis :

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

La commande `\discretionary` permet de contrôler finement la césure dans les cas où ne suffisent ni le contrôle standard de la césure fait l'algorithme de césure de \TeX et les règles de césures données par les paquetages de gestion linguistiques, ni les moyens de contrôle explicites offerts par les commandes `\hyphenation` (voir Section 9.8 [`\hyphenation`], page 51) et `\-` (voir Section 9.4 [`\-` (`hyphenation`)], page 50).

`\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 84). 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 51).

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 51).

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 débouresse pas tous les flottants (voir Section 10.2 [`\clearpage`], page 52).

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 64). 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 34.

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 `table` d'un environnement `minipage`, comme ci-après (voir Section 8.18 [minipage], page 34).

```
\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 40). 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.

Lorsque on écrit :

```
\section{Les ensembles complets\protect\footnote{Ce texte est dû à R~Jones.}}
```

alors 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. 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 ; le mettre aurait pour effet que la note en bas de page apparaîtrait de nouveau dans la table des matières.)

11.6 Paramètres des notes en bas de page

`\footnoterule`

Produit le filet de séparation entre texte principal sur une page et les notes en bas de cette page. Les dimension par défaut : épaisseur (ou largeur) `0.4pt`, et longueur `0.4\columnwidth` dans les classes standard de document (à l'exception `slide`, où elle n'apparaît pas).

`\footnotesep`

La hauteur de l'étai placé au début de la note en bas de page. Par défaut, c'est réglé à l'étai nominal pour des police de taille `\footnotesize` (voir Section 4.2 [Font sizes], page 13), 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 prend. 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[mava1]`, alors au sein de *défn* le premier *paramètre positionnel #1* se développe en *mava1*. Alors que 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 pas *#2*.

Omettre `[mava1]` dans l'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'un séquence de contrôle alphabétique, comme dans `\cmd '`. Si vous désirez réellement à cet endroit, une solution est de taper `{}` après la commande (`\cmd{}`) ; une autre solution est d'utiliser un espace de control explicite (`\cmd\ '`).

Un exemple simple de définition d'une nouvelle commande : `\newcommand{\JH}{Jim Hef{}feron}` a pour effet que l'abréviation `\JH` est remplacé par le texte plus long de la définition.

Redéfinir une commande se fait sensiblement de la même manière : `\renewcommand{\symboleqcfd}{\small CQFD}`.

Voici la définition d'une commande qui utilise des arguments :

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

Alors, `\defreference{def:basis}` se développe en quelque-chose du genre de 'Définition~3.14'.

Un exemple avec deux arguments : `\newcommand{\nbym}[2]{#1\!\times\!#2}` est invoqué comme `\nbym{2}{k}`.

Un exemple avec des arguments optionnels :

```
\newcommand{\salutation}[1][Madame, Monsieur]{#1,}
```

Alors, `\salutation` donne 'Madame, Monsieur,' alors `\salutation[Cher Jean]` donne 'Cher Jean,'. Et `\salutation[]` donne ','.

Les accolades autour de *defn* ne délimitent pas la portée du résultat du développement de *defn*. Ainsi la définition `\newcommand{\nomnavire}[1]{\it #1}` est incorrect puisque dans la phrase

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

les mots 'rencontra le' sont mis incorrectement en italique. Une paire supplémentaire d'accolades `\newcommand{\nomnavire}[1]{\it #1}` corrige ce problème.

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 64, 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 66, 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 94) 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 62).

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 1 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 optionel, 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 [*mava*1] 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}` ; le *nième* argument positionnel, c.-à-d. une construction de la forme #*n*, est remplacé au sein de *défdébut* par le texte du *nième* argument.

déffin Obligatoire ; le texte développé à toute occurrence de `\end{env}`. Notez qu'il ne doit contenir aucun paramètre positionnel, ainsi #*n* ne peut pas être utilisé ici (mais voyez l'exemple final ci-après).

L'environnement *env* délimite la portée du résultat du développement de *débutdéf* et de *findef*. 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}
}
```

Ceci 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 ici :

```
\begin{citequote}[Clovis, roi des Francs]
...
\end{citequote}
```

Ce dernier exemple montre comment sauvegarder la valeur d'un argument pour l'utiliser dans *findef*.

```
\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 22).

Cette déclaration est globale. Elle est fragile (voir Section 12.8 [`\protect`], page 62).

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 $Xe\LaTeX$ (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 mobile* 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 mobiles sont ceut qui apparaissent au sein des commande `\caption{...}` (voir Section 8.10 [figure], page 28), dans la commande `\thanks{...}` (voir Section 18.1 [`\maketitle`], page 89), et dans les expression en `@` des environnements `tabular` et `array` (voir Section 8.23 [tabular], page 41).

Si vous obtenez des erreurs étranges de commandes utilisées au sein d’arguments mobiles, 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 65, ou Section 13.4 [`\setcounter`], page 65.

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 les symboles : * † ‡ § ¶ || ** †† ‡‡

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 numérotter 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 L^AT_EX 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 65, 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**

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

En relation avec ces compteurs, la commande `\today` produit une chaîne représentant le jour courant (voir Section 21.7 [`\today`], page 103).

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 84.

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 94). Il spécifient la largeur naturelle, etc., du texte dans la boîte. `\totalheight` vaut `\height + \depth`. Pour fabriquer une boîte dont le texte est dilaté au double de sa taille naturelle, écrivez par ex. :

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

15 Faire des paragraphes

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

15.1 `\indent`

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

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

15.2 `\noindent`

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

... fin du paragraphe précédent.

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

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

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

15.3 `\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 9) ;
- dans la marge extérieure pour une disposition en recto-verso (option `twoside`, voir Section 3.1 [Document class options], page 9) ;
- dans la nearest la plus proche pour une disposition à deux colonnes (option `twocolumn`, voir Section 3.1 [Document class options], page 9).

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

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

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

Ces paramètres affectent le formattage de la note :

`\marginparpush`

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

`\marginparsep`

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

`\marginparwidth`

Largeur de la note même ; la valeur par défaut est ‘90pt’ pour un document en recto simple et à ‘10pt’, ‘83pt’ pour ‘11pt’, et ‘68pt’ pour ‘12pt’ ; ‘17pt’ de plus dans chacun de ces cas pour un document recto-verso. En mode à deux colonnes, la valeur par défaut est ‘48pt’.

La routine standard de \LaTeX pour les notes en marge n’empêche pas les notes de déborder hors du bas de la page.

16 Formules mathématiques

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

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

`displaymath`

Pour les formules qui apparaissent sur leur propre ligne.

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

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

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

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

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

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

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

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

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

16.1 Indices inférieurs & supérieurs

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

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

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

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

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

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

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

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

16.2 Symboles mathématiques

\LaTeX fournit presque tout symbole mathématique dont vous êtes susceptible d'avoir besoin. Par exemple, si vous incluez `\pi` dans votre code source, alors vous obtenez le symbole π .

Ci-dessous se trouve une liste de symboles généralement disponibles. Ce n'est en aucun cas une liste exhaustive. Chaque symbole y est décrit brièvement et avec sa classe de symbole (qui détermine l'espacement autour de lui) qui est donnée entre parenthèses. Les commandes pour ces symboles peuvent, sauf mention contraire, être utilisées seulement en mode mathématique.

Pour redéfinir une commande de sorte à pouvoir l'utiliser quel que soit le mode, voir Section 17.1 [`\ensuremath`], page 87.

<code>\parallel</code>	Parallèle (relation). Synonyme : <code>\parallel</code> .
<code>\aleph</code>	\aleph Aleph, cardinal transfini (ordinaire).
<code>\alpha</code>	α Lettre grecque en bas de casse alpha (ordinaire).
<code>\amalg</code>	\amalg Union disjointe (binaire)
<code>\angle</code>	\angle Angle géométrique (ordinaire). Similaire : sign inférieur-à <code><</code> et chevron <code>\langle</code> .
<code>\approx</code>	\approx Presque égal à (relation).
<code>\ast</code>	$*$ Opérateur astérisque, convolution, à six branches (binaire). Synonyme : <code>*</code> , ce qui est souvent un indice supérieur ou inférieur, comme dans l'étoile de Kleene. Similaire : <code>\star</code> , qui est à cinq branches, et est quelque-fois utilisée comme une opération binaire générale, and quelques-fois réservée pour la corrélation-croisée.
<code>\asymp</code>	\asymp Équivalent asymptotiquement (relation).
<code>\backslash</code>	\backslash contr'oblique (ordinaire). Similaire : soustraction d'ensemble <code>\setminus</code> , et <code>\textbackslash</code> pour une contr'oblique en-dehors du mode mathématique.
<code>\beta</code>	β Lettre grecque en bas de casse beta (ordinaire).
<code>\bigcap</code>	\bigcap De taille Variable, ou n-aire, intersection (opérateur). Similaire : intersection binaire <code>\cap</code> .
<code>\bigcirc</code>	\bigcirc Cercle, plus grand (binaire). Similaire : composition de fonction <code>\circ</code> .
<code>\bigcup</code>	\bigcup De taille variable, ou n-aire, union (opérateur). Similaire : union binaire <code>\cup</code> .
<code>\bigodot</code>	\bigodot De taille variable, ou n-aire, opérateur point encerclé (opérateur).
<code>\bigoplus</code>	\bigoplus De taille variable, ou n-aire, opérateur plus encerclé (opérateur).

<code>\bigotimes</code>	\otimes De taille variable, ou n-aire, opérateur multiplié encerclé (opérateur).
<code>\bigtriangledown</code>	∇ De taille variable, ou n-aire, triangle ouvert pointant vers le bas (opérateur).
<code>\bigtriangleup</code>	\triangle De taille variable, ou n-aire, triangle ouvert pointant vers le haut (opérateur).
<code>\bigsqcup</code>	\sqcup De taille variable, ou n-aire, union carrée (opérateur).
<code>\biguplus</code>	\uplus De taille variable, ou n-aire, opérateur union avec un plus (opérateur). (Notez que le nom a juste un p.)
<code>\bigvee</code>	\vee De taille variable, ou n-aire, et-logique (opérateur).
<code>\bigwedge</code>	\wedge De taille variable, ou n-aire, ou-logique (opérateur).
<code>\bot</code>	\bot Taquet vers le haut, minimum, plus petit élément d'un ensemble partiellement ordonné, ou une contradiction (ordinaire). Voir aussi <code>\top</code> .
<code>\bowtie</code>	\bowtie Jonction naturelle de deux relations (relation).
<code>\Box</code>	\Box Opérateur modal pour la nécessité ; boîte ouverte carrée (ordinaire). Ceci n'est pas disponible en $\text{T}_{\text{E}}\text{X}$ de base. En $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ vous devez charger le paquetage <code>amssymb</code> .
<code>\bullet</code>	• Puce (binaire). Similaire : point de multiplication <code>\cdot</code> .
<code>\cap</code>	\cap Intersection de deux ensembles (binaire). Similaire : opérateur de taille variable <code>\bigcap</code> .
<code>\cdot</code>	· Multiplication (binaire). Similaire : Point puce <code>\bullet</code> .
<code>\chi</code>	χ Lettre grecque en bas de casse chi (ordinaire).
<code>\circ</code>	◦ Composition de fonctions, opérateur rond (binaire). Similaire : opérateur de taille variable <code>\bigcirc</code> .
<code>\clubsuit</code>	♣ Trèfle de carte à jouer (ordinaire).
<code>\complement</code>	<code>\complement</code> Complement d'ensemble, utilisée en indice supérieur comme dans S^{\complement} (ordinaire). 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> . On utilise aussi : S^{c} ou \bar{S} .
<code>\cong</code>	\cong Congruent (relation).
<code>\coprod</code>	\coprod Coproduit (opérateur).
<code>\cup</code>	\cup Réunion de deux ensembles (binaire). opérateur de taille variable <code>\bigcup</code> .
<code>\dagger</code>	† Relation obèle (binaire).

<code>\dashv</code>	\dashv Taquet gauche, tiret avec barre verticale à droite, tourniquet pour gauchers (relation). Similaire : taquet droit, tourniquet <code>\vdash</code> .
<code>\ddagger</code>	\ddagger Relation double obèle (binaire).
<code>\Delta</code>	Δ Delta grec capital, utilisé pour indiquer une incrémentation (ordinaire).
<code>\delta</code>	δ Delta grec bas-de-casse (ordinaire).
<code>\Diamond</code>	\Diamond Opérateur grand carreau (ordinaire). Ceci n'est pas disponible en \TeX de base. En \LaTeX vous devez charger le paquetage <code>amssymb</code> .
<code>\diamond</code>	\diamond Opérateur Carreau, ou puce carreau (binaire). Similaire : grand carreau <code>\Diamond</code> , puce disque <code>\bullet</code> .
<code>\diamondsuit</code>	\diamondsuit Carreau de carte à jouer (ordinaire).
<code>\div</code>	\div Signe de division, obélus (binaire).
<code>\doteq</code>	\doteq Approche la limite (relation). Similaire : géométriquement égal à <code>\Doteq</code> .
<code>\downarrow</code>	\downarrow Flèche vers le bas, converge (relation). Similaire : flèche à ligne double vers le bas <code>\Downarrow</code> .
<code>\Downarrow</code>	\Downarrow Flèche à ligne double vers le bas (relation). Similaire : flèche à ligne simple vers le bas <code>\downarrow</code> .
<code>\ell</code>	ℓ Lettre l bas de casse cursive (ordinaire).
<code>\emptyset</code>	\emptyset Symbole ensemble vide (ordinaire). La forme en variante est <code>\varnothing</code> .
<code>\epsilon</code>	ϵ Lettre grecque bas de casse epsilon lunaire (ordinaire). Sembable à la lettre en texte grec. En mathématiques on utilise plus fréquemment l'épsilon minuscule de ronde <code>\varepsilon</code> ε . En relation avec : la relation ensembliste « appartenir à » <code>\in</code> .
<code>\equiv</code>	\equiv Équivalence (relation).
<code>\eta</code>	η Lettre grecque en bas de casse (ordinaire).
<code>\exists</code>	\exists Quanteur « Il existe » (ordinaire).
<code>\flat</code>	\flat Bémol (ordinaire).
<code>\forall</code>	\forall Quanteur « Pour tout » (ordinaire).
<code>\frown</code>	\frown Moue (ordinaire).
<code>\Gamma</code>	Γ Lettre grecque en capitale (ordinaire).
<code>\gamma</code>	γ Lettre grecque en bas de casse (ordinaire).
<code>\ge</code>	\ge Supérieur ou égal à (relation). Synonyme : <code>\geq</code> .
<code>\geq</code>	\geq Supérieur ou égal à (relation). Synonyme : <code>\ge</code> .

<code>\gets</code>	\leftarrow Auquel est assignée la valeur (relation). Synonyme : <code>\leftarrow</code> .
<code>\gg</code>	\gg Beaucoup plus grand que (relation). Similaire : Beaucoup plus petit que <code>\ll</code> .
<code>\hbar</code>	\hbar Constante de Planck sur deux pi (ordinaire).
<code>\heartsuit</code>	\heartsuit Cœur de carte jouer (ordinaire).
<code>\hookleftarrow</code>	\hookleftarrow Flèche vers la gauche avec crochet (relation).
<code>\hookrightarrow</code>	\hookrightarrow Flèche vers la gauche avec crochet (relation).
<code>\iff</code>	\iff Si et seulement si (relation). C'est un <code>\Longleftarrow</code> avec un <code>\thickmuskip</code> de chaque côté.
<code>\Im</code>	\Im Partie imaginaire (ordinaire). Voir aussi : partie réelle <code>\Re</code> .
<code>\in</code>	\in Appartient à (relation). Voir aussi : lettre grecque bas de casse epsilon lunaire <code>\epsilon</code> et l'épsilon minuscule de ronde <code>\varepsilon</code> .
<code>\infty</code>	∞ Infini (ordinaire).
<code>\int</code>	\int Intégrale (opérateur).
<code>\iota</code>	ι Lettre grecque en bas de casse (ordinaire).
<code>\Join</code>	<code>\Join</code> Symbole jointure condensé (relation). Ceci n'est pas disponible en <code>T_EX</code> de base.
<code>\kappa</code>	κ Lettre grecque en bas de casse (ordinaire).
<code>\Lambda</code>	Λ Lettre grecque en capitale (ordinaire).
<code>\lambda</code>	λ Lettre grecque en bas de casse (ordinaire).
<code>\land</code>	\wedge Et logique (binaire). Synonyme : <code>\wedge</code> . Voir aussi ou-logique <code>\lor</code> .
<code>\langle</code>	\langle Chevron mathématique gauche, ou séquence, crochet (ouvrant). Similaire : inférieur à <code><</code> . Correspond à <code>\rangle</code> .
<code>\lbrace</code>	$\{$ Accolade gauche (ouvrant). Synonyme : <code>\{</code> . Correspond à <code>\rbrace</code> .
<code>\lbrack</code>	$[$ Crochet gauche (ouvrant). Synonyme : <code>[</code> . Correspond à <code>\rbrack</code> .
<code>\lceil</code>	\lceil Plafond à gauche, ressemble à un crochet mais avec le pied coupé (ouvrant). Correspond à <code>\rceil</code> .
<code>\le</code>	\leq Inférieur ou égal à (relation). C'est un synonyme de <code>\leq</code> .
<code>\leadsto</code>	<code>\leadsto</code> flèche vers la droite en tire-bouchon (relation). Ceci n'est pas disponible en <code>T_EX</code> de base. En <code>L^AT_EX</code> vous devez charger le paquetage <code>amssymb</code> .
<code>\Leftarrow</code>	\Leftarrow Est impliqué par, flèche vers la gauche à double-ligne (relation). Similaire : flèche vers la gauche à simple ligne <code>\leftarrow</code> .

- `\leftarrow`
 \leftarrow Flèche vers la gauche à simple ligne (relation). Synonyme : `\gets`. Similaire : Flèche vers la gauche à double-ligne `\Leftrightarrow`.
- `\leftharpoondown`
 \leftharpoondown Harpon vers la gauche à simple ligne avec crochet vers le bas (relation).
- `\leftharpoonup`
 \leftharpoonup Harpon vers la gauche à simple ligne avec crochet vers le haut (relation).
- `\Leftrightarrow`
 \Leftrightarrow Bi-implication ; double flèche bilatérale (relation). Similaire : flèche bilatérale `\leftleftrightarrow`.
- `\leftleftrightarrow`
 \leftleftrightarrow Flèche bilatérale (relation). Similaire : double flèche bilatérale `\Leftrightarrow`.
- `\leq` \leq Inférieur ou égal à (relation). Synonyme : `\leq`.
- `\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`.
- `\longleftleftrightarrow`
 \longleftleftrightarrow Longue flèche bilatérale à ligne simple (relation). Similaire : longue flèche bilatérale à ligne double `\Longleftleftrightarrow`.
- `\longmapsto`
 \longmapsto Longue flèche d'un taquet vers la droite à ligne simple (relation). Similaire : version courte `\mapsto`.
- `\longrightarrow`
 \longrightarrow Longue flèche vers la droite à ligne simple (relation). Similaire : longue flèche vers la droite à ligne double `\Longrightarrow`.
- `\lor` \vee Ou-logique (binaire). Synonyme : `\wedge`.
- `\mapsto` \mapsto Flèche d'un taquet vers la droite (relation). Similaire : version longue `\longmapsto`.
- `\mho` `\mho` Conductance, symbole Ohm culbuté (ordinaire). Ceci n'est pas disponible en `TEX` de base. En `LATEX` vous devez charger le paquetage `amssymb`.

<code>\mid</code>	Barre verticale à ligne simple (relation). une utilisation typique de <code>\mid</code> est pour un ensemble $\{\,x \mid x \geq 5\, \}$. Similaire : <code>\vert</code> et <code> </code> produisent le même symbole de barre verticale à ligne simple mais sans aucun espacement (on les classe comme « ordinaires ») et vous ne devriez pas les utiliser comme symbole de relation mais seulement comme des ordinaux, c.-à-d. comme des symboles de note en bas de page. Pour dénoter une valeur absolue, voir l'entrée de <code>\vert</code> et pour une norme voir l'entrée de <code>\Vert</code> .
<code>\models</code>	\models Vrai, ou satisfait ; double turnstile, short double dash (relation). Similaire : long double dash <code>\vDash</code> .
<code>\mp</code>	\mp Moins ou plus (relation).
<code>\mu</code>	μ Lettre grecque en bas de casse (ordinaire).
<code>\nabla</code>	∇ Del de Hamilton, ou différentiel, opérateur (ordinaire).
<code>\natural</code>	\natural Bécarre (ordinaire).
<code>\ne</code>	\neq Non égal (relation). Synonyme : <code>\neq</code> .
<code>\nearrow</code>	\nearrow Flèche nord-est (relation).
<code>\neg</code>	\neg Négation logique (ordinaire). Synonyme : <code>\lnot</code> . Utilisé parfois au lieu du symbole de négation: <code>\sim</code> .
<code>\neq</code>	\neq Non égal (relation). Synonyme : <code>\ne</code> .
<code>\ni</code>	\ni Signe d'appartenance réfléchi ; « contient comme élément » (relation). Synonyme : <code>\owns</code> . Similaire : « appartient à » <code>\in</code> .
<code>\not</code>	\not Barre oblique longue utilisée pour rayer un opérateur la suivant (relation). Beaucoup opérateurs en négation qui n'ont pas besoin de <code>\not</code> sont disponibles, en particulier avec le paquetage <code>amssymb</code> . Par exemple, <code>\notin</code> est sans doute préférable à <code>\not\in</code> d'un point de vue typographique.
<code>\notin</code>	\notin N'appartient pas à (relation). Similaire : non contenu dans <code>\nsubseteq</code> .
<code>\nu</code>	ν Lettre grecque en bas de casse (ordinaire).
<code>\nwarrow</code>	\nwarrow Flèche nord-ouest (relation).
<code>\odot</code>	\odot Point cerclé (binaire). Similaire : opérateur de taille variable <code>\bigodot</code> .
<code>\oint</code>	\oint Intégrale de contour, intégrale avec un cercle au milieu (opérateur).
<code>\Omega</code>	Ω Lettre grecque en capitale (ordinaire).
<code>\omega</code>	ω Lettre grecque en bas de casse (ordinaire).
<code>\ominus</code>	\ominus Signe moins, ou tiret, cerclé (binaire).
<code>\oplus</code>	\oplus Signe plus cerclé (binaire). Similaire : opérateur de taille variable <code>\bigoplus</code> .
<code>\oslash</code>	\oslash Barre de fraction, ou barre oblique, cerclée (binaire).
<code>\otimes</code>	\otimes Signe de multiplication, ou croix, cerclé (binaire). Similaire : opérateur de taille variable <code>\bigotimes</code> .

<code>\owns</code>	\ni Epsilon d'appartenance réfléchi ; « contient comme élément » (relation). Synonyme : <code>\ni</code> . Similaire : appartient à <code>\in</code> .
<code>\parallel</code>	\parallel Parallèle (relation). Synonyme : <code>\ </code> .
<code>\partial</code>	∂ Différentielle partielle (ordinaire).
<code>\perp</code>	\perp Perpendiculaire (relation). Similaire : <code>\bot</code> utilise le même glyphe mais l'espacement est différent parce qu'il est dans la classe ordinaire.
<code>\phi</code>	ϕ Lettre grecque en bas de casse (ordinaire). La forme en variante est <code>\varphi</code> φ .
<code>\Pi</code>	Π Lettre grecque en capitale (ordinaire).
<code>\pi</code>	π Lettre grecque en bas de casse (ordinaire). La forme en variante est <code>\varpi</code> ϖ .
<code>\pm</code>	\pm Plus or minus (binaire).
<code>\prec</code>	\prec Précède (relation). Similaire : inférieur à $<$.
<code>\preceq</code>	\preceq Précède ou égal, précède par-dessus signe égal à une ligne (relation). Similaire : inférieur ou égal <code>\leq</code> .
<code>\prime</code>	' Prime, ou minute au sein d'une expression temporelle (ordinaire). Typiquement utilisé en indice supérieur <code>A^\prime</code> . Il est à noter que <code>f^\prime</code> et <code>f''</code> produisent le même résultat. Un avantage de la seconde forme est que <code>f''''</code> produit le symbole désiré, c.-à-d. le même résultat que <code>f^{\prime\prime\prime}</code> , mais en nécessitant moins de saisie. Il est à noter que vous ne pouvez utiliser <code>\prime</code> qu'en mode mathématique. Utiliser le quote ' en mode texte produit un caractère différent (apostrophe ou guillemet anglais simple de droite).
<code>\prod</code>	\prod Produit (opérateur).
<code>\propto</code>	\propto Proportionnel à (relation)
<code>\Psi</code>	Ψ Lettre grecque en capitale (ordinaire).
<code>\psi</code>	ψ Lettre grecque en bas de casse (ordinaire).
<code>\rangle</code>	\rangle Chevron mathématique droit, ou séquence, crochet (fermant). Similaire : supérieur à $>$. Correspond à <code>\rangle</code> .
<code>\rbrace</code>	$\}$ Accolade de droite (fermante). Synonyme : <code>\}</code> . Correspond à <code>\lbrace</code> .
<code>\rbrack</code>	$\}$ Crochet droit (fermant). Synonyme : <code>\}</code> . Correspond à <code>\lbrack</code> .
<code>\rceil</code>	\lceil Plafond droit (fermant). Correspond à <code>\lceil</code> .
<code>\Re</code>	\Re Partie réelle, nombres réels, R capital gothique (ordinaire). En relation avec : R majuscule ajouré, ou gras de tableau d'école <code>\mathbb{R}</code> ; pour accéder à cela charger le paquetage <code>amsfonts</code> .
<code>\restriction</code>	<code>\restriction</code> Restriction d'une fonction (relation). Synonyme : <code>\upharpoonright</code> . Ceci n'est pas disponible en T _E X de base. En L ^A T _E X vous devez charger le paquetage <code>amssymb</code> .

<code>\reemptyset</code>	<code>\reemptyset</code> Symbole ensemble vide inversé (ordinaire). En relation avec : <code>\varnothing</code> . Ceci n'est pas disponible en $\text{T}_{\text{E}}\text{X}$ de base. En $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ vous devez charger le paquetage <code>stix</code> .
<code>\rfloor</code>	<code>\rfloor</code> Plancher droit, un crochet droit avec la dent du haut coupée (fermant). Correspond à <code>\lfloor</code> .
<code>\rhd</code>	<code>\rhd</code> Pointe de flèche, c.-à-d. u triangle pointant vers la droite (binaire). Ceci n'est pas disponible en $\text{T}_{\text{E}}\text{X}$ de base. En $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ vous devez charger le paquetage <code>amssymb</code> . Pour le symbole normal de sousgroupe vous devriez plutôt charger <code>amssymb</code> et utiliser <code>\vartriangleright</code> (qui est une relation et ainsi donne un meilleur espacement).
<code>\rho</code>	<code>\rho</code> Lettre grecque en bas de casse (ordinaire). La forme en variante est <code>\varrho</code> .
<code>\Rightarrow</code>	<code>\Rightarrow</code> Implique, flèche vers la droite à double-ligne (relation). Similaire : flèche vers la droite <code>\rightarrow</code> .
<code>\rightarrow</code>	<code>\rightarrow</code> Flèche vers la droite à ligne simple (relation). Synonyme : <code>\to</code> . Similaire : flèche vers droite à ligne double <code>\Rightarrow</code> .
<code>\rightharpoondown</code>	<code>\rightharpoondown</code> Harpon vers la droite avec crochet vers le bas (relation).
<code>\rightharpoonup</code>	<code>\rightharpoonup</code> Harpon vers la droite avec crochet vers la haut (relation).
<code>\rightleftharpoons</code>	<code>\rightleftharpoons</code> Harpon vers la droite avec crochet vers le haut au dessus d'un harpon vers la gauche avec crochet vers le bas (relation).
<code>\searrow</code>	<code>\searrow</code> Flèche pointant sud-est (relation).
<code>\setminus</code>	<code>\setminus</code> Difference ensembliste, barre de faction 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> \simeq 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 Unnion de multienemble, 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 mathematique que la forme non variante epsilon lunaire `\epsilon`. En relation avec : « appartient à » `\in`.
- `\vanothing` `\varnothing` Symbole ensemble vide. Similaire : `\emptyset`. En relation avec : `\revertemptyset`. 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` \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 100).

<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 66).

<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 91.
<code>\!</code>	Une espace finie négative. Normalement -3mu . Utilisable seulement en mode mathématique.
<code>\quad</code>	Vaut 18mu , c.-à-d. 1 em . Souvent utilisé pour l'espace entourant les équation ou expression, par ex. pour l'espace entre deux équation au sein d'un environnement <code>displaymath</code> . Utilisable à la fois en mode mathématique et en mode texte.
<code>\qqquad</code>	Une longueur de 2 quads, soit $36\text{mu} = 2\text{ em}$. Utilisable à la fois en mode mathématique et en mode texte.

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

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

16.6 Recueil de diverses commandes utiles en math

<code>*</code>	Un symbole de multiplication <i>discrétionnaire</i> , sur lequel un saut de ligne est autorisé. En l'absence de saut, la multiplication est indiquée implicitement par un espace, alors qu'en cas de saut un symbole \times est imprimé immédiatement avant le saut. Ainsi
-----------------	--

```
\documentclass{article}
```

```

\usepackage[utf8]{inputenc}
\usepackage{french}
\begin{document}
Mais  $(A_3 = 0)$ , ainsi le produit de tous les termes de
 $(A_1)$  à  $(A_4)$ , c.-à-d.  $(A_1 * A_2 * A_3 * A_4)$ ,
est égal à zéro.
\end{document}

```

produit une sortie de ce type :

Mais $A_3 = 0$, ainsi le produit de tous les termes de A_1 à A_4 , c.-à-d. $A_1 A_2 \times A_3 A_4$, est égal à zéro.

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

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

`\frac{num}{dén}`

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

`\left delim1 ... \right delim2`

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

`\overbrace{texte}`

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

;\textrm{fois}}. Le résultat ressemble à cela : $x + \cdots + x$

`\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 résultat ressemble à cela : \xrightarrow{f} .

`\underbrace{math}`

Génère *math* avec une accolade en-dessous. 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) .

Un 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 84).

19.1 `\hspace`

Synopsis :

```
\hspace[*]{longueur}
```

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

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

19.2 `\hfill`

La commande `\hfill` de remplissage produit une "longueur élastique" qui n'a aucun espace naturel mais peut s'allonger ou se rétrécir horizontalement autant que nécessaire.

Le paramètre `\fill` est la longueur élastique elle-même (c.-à-d. pour parler technique, la valeur de glue 'opt plusifill') ; ainsi, `\hspace\fill` équivaut à `\hfill`.

19.3 `\espace et \@`

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

Par défaut, lorsque L^AT_EX justifie une ligne il ajuste l'espace après le point final de chaque phrase (ou le point d'interrogation, la virgule, ou le deux-points) plus que pour l'espace inter-mot (voir Section 19.5 [frenchspacing], page 91). 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`.

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

En contraste, placer `\@` à la droite d'un point dit à T_EX que ce point ne finit pas la phrase. Dans l'exemple `les mot réservés (if, then, etc.\@) sont différents`, T_EX 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\ est un chouette système`. Et, dans des circonstances normales, `\tabulation` et `\sautdeligne` sont équivalents à `\`.

Certaines personnes préfèrent utiliser `{}` dans le même but, comme dans `\TeX{} est un chouette système`. Ceci a l’avantage que vous n’avez pas à écrire toujours la même chose, comme `\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 interphrase et inter-mot

Cette déclaration (provenant de `TeX` de base) cause que `LATEX` traite l’espacement interphrase de la même façon que l’espacement inter-mot.

En justifiant le texte sur une ligne, certaines traditions typographiques, y compris l’anglaise, préfèrent ajuster l’espace entre les phrases (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 définie 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`, `LATEX` insère automatiquement une correction italique appropriée si nécessaire (voir Section 4.1 [Font styles], page 11).

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

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 66) 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 66). Toutefois, si l'espace vertical a déjà été ajoutée 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 verticale 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 verticale, et non deux.)

Cette commande est fragile (voir Section 12.8 [`\protect`], page 62).

L'erreur `'Something's wrong--perhaps a missing \item'` signifie que vous n'étiez pas en mode verticale 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` est `3pt plus 1pt minus 1pt`).

19.11 \vfill

Finir le paragraphe courante et insère une longueur élastique verticale (voir Chapitre 14 [Lengths], page 66) 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 93.

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 66).

L^AT_EX 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 67) 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 36, 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 96.

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 35.

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 fantaisiste à l'intérieur. En particulier, vous ne devriez utiliser aucun environnements 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 34).

`\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 94), 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 58).

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 94) 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 58).

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 96).

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` Symbole 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 : ’.

`\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 38).
- `\.`
Produit un point suscrit à la lettre qui suit, comme dans ô.
- `\=`
`\capitalmacron`
Produit un macron (une barre) suscrit à la lettre qui suit, comme dans õ.
- `\^`
`\capitalcircumflex`
Produit un accent circonflexe (un chapeau) suscrit à la lettre qui suit, comme dans ô.
- `\‘`
`\capitalgrave`
Produit un accent grave suscrit à la lettre qui suit, comme dans ò. Au sein de l’environnement `tabbing`, déplace le texte qui suit à la marge de droite (voir Section 8.21 [tabbing], page 38).
- `\~`
`\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 ‘ŏ’.

`\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 84. 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 \LaTeX pour insérer des caractères utilisés généralement utilisés dans des langages autres que l’anglais.

`\aa`
`\AA` å et Å.

`\ae`
`\AE` æ et Æ.

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

`\dj`
`\DJ` d et D barre, lettre d capitale et bas-de-casse avec une barre traversant la hampe. Non disponible dans le codage OT1, il vous faut le paquetage `fontenc` pour sélectionner un autre codage de police, tel que T1.

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

`\l`
`\L` ł et Ł.

`\ng`
`\NG` Lettre eng laponais, utilisé aussi en phonétique. Non disponible dans le codage OT1, il vous faut le paquetage `fontenc` pour sélectionner un autre codage de police, tel que T1.

<code>\o</code>	
<code>\O</code>	ø et Ø.
<code>\oe</code>	
<code>\OE</code>	œ et Œ.
<code>\ss</code>	
<code>\SS</code>	ß et SS.
<code>\th</code>	
<code>\TH</code>	Lettre islandaise thorn : þ et Þ. Non disponible dans le codage OT1, il vous faut le paquetage <code>fontenc</code> pour sélectionner un autre codage de police, tel que T1.

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 65) 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}
```

La paquetage `datetime` package, entre autres, can produce a wide variety of other date formats.

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 29, 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 52), 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 104.

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`

The `\addtocontents{ext}{texte}` commande adds *texte* (or formatting commandes) directly to the `.ext` file that generates the table of contents or listes of figures or tables.

<i>ext</i>	The extension of the file on which information is to be written: <code>toc</code> (table of contents), <code>lof</code> (list of figures), or <code>lot</code> (list of tables).
<i>texte</i>	The <i>texte</i> to be written.

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}`, where *numpage* est la valeur courante de `\thepage`.

Le paquetage `glossary` disponible sur le CTAN fournit une prise en charge de glossaires plus glossaries.

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 `lettre` 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 nom de personne 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 `\\`.

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{first line \
... }
```

Le nom de l'expéditeur. Cette commande est optionnelle, quoique son inclusion est 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 `supplying` 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 saisissiez 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 \TUB\ article}

% repeat info for each author.
\author{First Last}
\address{Street Address \\ Town, Postal \\ Country}
\netaddress{user (at) exemple dot org}
\personalURL{http://exemple.org/~user/}

\begin{document}

\maketitle

\begin{abstract}
This is an exemple article for \TUB{}.
\end{abstract}

\section{Introduction}

This is an exemple article for \TUB, from
\url{http://tug.org/TUGboat/location.html}.

We recommend the graphicx package for image inclusions, and the
hyperref package for active url's (in the \acro{PDF} output).
Nowadays \TUB\ is produced using \acro{PDF} files exclusively.

The \texttt{ltugboat} class provides these abbreviations and many more:

% verbatim blocks are often better in \small
```

```

\begin{verbatim}[\small]
\AllTeX \AMS \AmS \AmSLaTeX \AmSTeX \aw \AW
\BibTeX \CTAN \DTD \HTML
\ISBN \ISSN \LaTeXe
\Mc \mf \MFB \mtex \PCTeX \pcTeX
\PiC \PiCTeX \plain \POBox \PS
\SC \SGML \SliTeX \TANGLE \TB \TP
\TUB \TUG \tug
\UG \UNIX \VAX \XeT \WEB \WEAVE

\Dash \dash \vellipsis \bull \cents \Dag
\careof \thinspace

\acro{FRED} -> {\small[er] fred} % please use!
\cs{fred}   -> \fred
\env{fred}  -> \begin{fred}
\meta{fred} -> <fred>
\nth{n}     -> 1st, 2nd, ...
\sfrac{3/4} -> 3/4
\booktitle{Book of Fred}
\end{verbatim}

```

For more information, see the ltubguid document at:
[\url{http://mirror.ctan.org/macros/latex/contrib/tugboat}](http://mirror.ctan.org/macros/latex/contrib/tugboat)
 (we recommend using `\verb|mirror.ctan.org|` for `\CTAN\` references).

Email `\verb|tugboat@tug.org|` if problems or questions.

```

\bibliographystyle{plain} % we recommend the plain bibliography style
\nocite{book-minimal}    % just making the bibliography non-empty
\bibliography{xampl}      % xampl.bib comes avec BibTeX

\makesignature
\end{document}

```

Index des concepts

- « Voir » et « Voir aussi », article d'index 106
- écrire des lettres 107
 écrire un fichier externe 29
 équations, aligner 27
 équations, environnement pour 28
 étiquette tapuscrite dans les listes 25
 étirement, omettre l'étirement vertical 16
- ***
- '*', invite 113
- .
- .glo, fichier 106
 .idx, fichier 106
 .ind, fichier 106
- \
- \include emboîtés interdits 104
- A**
- a e liés 102
 a rond-en-chef 102
 a, e dans l'a 102
 accéder à un caractère
 quelconque d'une police 97
 accent 11, 12
 accent aigu 101
 accent aigu en math 83
 accent barre suscrite en math 83
 accent brève en math 83
 accent caron 102
 accent caron en math 83
 accent chapeau en math 83
 accent circonflexe 101
 accent circonflexe en math 83
 accent diérèse en math 83
 accent double-brève-renversée 102
 accent grave en math 83
 accent háček 102
 accent háček en math 83
 accent macron en math 83
 accent point 101
 accent point en chef 101
 accent point suscrit 101
 accent point suscrit en math 83
 accent rond en chef 101
- accent rond-en-chef en math 84
 accent tilde en math 84
 accent tréma 100
 accent tréma en math 83
 accent tréma hongrois 101
 accents 100
 Accents mathématiques 83
 accentuées, lettres 100
 accentués, caractères 100
 accolade droite, dans le texte 99
 accolade gauche, dans le texte 99
 afficher du texte cité avec
 renforcement de paragraphe 38
 afficher un texte cité sans
 renforcement de paragraphe 38
 agrandir la page courante 52
 align environnement, d'amsmath 27
 alignement au moyen de tabbing 38
 aligner des équations 27
 aligner du texte dans des tableaux 41
 aligner du texte en utilisant des
 taquets de tabulation 38
 amsmath paquetage, remplaçant eqnarray 27
 annexe, création 21
 aperçu de L^AT_EX 3
 appel 110
 arguments mobiles 62
 article d'une liste, compteur spécifiant 64
 articles numérotés, compteur spécifiant 64
 ASCII, circonflexe dans le texte 98
 ASCII, tilde dans le texte 98
 astérisque centré, dans le texte 98
- B**
- barre au-dessous 102
 barre par-dessus, accent 101
 barre sousscite 102
 barre verticale double, dans le texte 98
 barre verticale, dans le texte 98
 barre, diacritique 101
 barre, double verticale, dans le texte 98
 barre-souscrite, diacritique 101
 bas de page, créer des notes 53
 bas de page, des paramètres pour 16
 bases de L^AT_EX 3
 beamer patron et classe 114
 bibliographie, créer (automatiquement) 46
 bibliography, creating (manually) 45
 bibT_EX, utilisation 46
 Big point 66
 boîtes 94
 boîtes noires, en omettant 9
 Bonjour le monde 3

brève, accent 102
brève, diacritique 102

C

cédille, diacritique 101
césure, définir 51
césure, empêcher 94
césure, forcer 50
cadratin 66, 99
cadratin, demi 99
cadratin, deux tiers 100
cadratin, trois-quarts 100
caractères latins non diacrités 102
caractères réservés 97
caractères spéciaux 97
caractères speciaux 102
caron, diacritique 102
catégorie, code de ... de caractère 6
catcode 6
Centimètre 66
centré, astérisque, dans le texte 98
centré, point, dans le texte 99
centrer du texte, déclaration pour 25
centrer du texte, environnement pour 25
chapeau, accent 101
chargement des paquetages supplémentaires ... 10
chevron double, guillemets 98
chevron droit, guillemets 98
chevron gauche, guillemets 98
chevron simple, guillemets 98
chiffres de revêtement 12
chiffres de style ancien 12
chiffres de style ancien, 12
Cicero 66
circonflexe ASCII, dans le texte 98
circonflexe, accent 101
circonflexe, diacritique 101
citation 38
classe de document, définition 3
classe `ltugboat` 115
classes de documents 9
clause at, au sein d'une définition de police 62
clef de référence bibliographique 46
code de catégorie de caractère 6
code source, composer du 48
commande de longueur 66
commandes de fontes de bas niveau 13
commandes de fontes, de bas niveau 13
commandes de mise en page 15
commandes fragiles 62
commandes robustes 62
commandes, en définir de nouvelles 56
commandes, syntaxe des 5
commencer une nouvelle page 52
commencer une nouvelles page et
 débourrer les flottant 52
commencer une page de droite 52

compteurs, a list of 64
compteurs, définir de nouveaux 57
compteurs, obtenir la valeur d'un 64
compteurs, réglage 65
contr'oblique, dans le texte 98
copie, liste des personnes en, dans une lettre... 108
Correction italique 91
coupure de mot, empêcher 94
créer des images 34
créer des tableaux 40
crésure, discrétionnaire 50

D

débourrer les flottant et commencer une page... 52
définir de nouveaux théorèmes 60
définir de nouvelles polices 61
définir une nouvelle commande 56
définitions 56
délimitateur nul 85
date d'aujourd'hui 103
`datetime`, paquetage 103
de démarrage et de fin 3
defining new environnements 58
demi-cadratin 99
deux-tiers cadratin 100
diacritique barre-souscrite 101
diacritique brève 102
diacritique cédille 101
diacritique caron 102
diacritique circonflexe 101
diacritique double-brève-renversée 102
diacritique háček 102
diacritique macron-souscrit 101
diacritique ogonek 101
diacritique point suscrit 101
diacritique point-souscrit 101
diacritique rond en chef 101
diacritique tilde 101
diacritique tirant suscrit 102
diacritique tréma hongrois 101
diacritique accent aigu 101
diacritique tréma 100
diacritiques, signes 100
dilatation verticale infinie 93
discrétionnaire, césure 50
dollar, signe 99
double obèle, dans le texte 97
double renvoi, résolution 4
double, barre verticale, dans le texte 98
double, guillemet-apostrophe culbuté 100
double, guillemet-apostrophe de droite 100
double, obèle, dans le texte 99
double-brève-renversée, diacritique 102
droit d'auteur, symbole 97
droit, guillemet double sur la ligne de base ... 100
droit, guillemet simple sur la ligne de base ... 100
droite, accolade, dans le texte 99

droite, flèche, dans le texte 100
 droite, guillemet anglais simple 98
 droite, guillemet-apostrophe simple 98

E

e dans l'a 102
 e dans l'o 103
 e-TeX 4
 em 66
 en-tête et pied de page en cours de traitement . . 16
 en-tête, des paramètres pour 16
 encerclée, lettre, dans le texte 99
 entrée Unicode, native 4
 entrée/sortie console 112
 environnements 23
 environnement 3
 environnement simili-théorème 60
 environnements, defining 58
 espace vertical 92
 espace vertical avant les paragraphes 68
 espace vertical, insérer 92
 espace verticale 93
 espace visible 48
 espace visible, symbole, dans le texte 100
 espacement au sein du mode mathématique 84
 espacement inter-phrased 91
 espaces 90
 eszet lettre allemande 103
 eth islandais 102
 eth, lettre islandaise 102
 euro, symbole 99
 exposant 70

F

féminin, symbole ordinal 99
 fabriquer une page de titre 47
 Faire des paragraphes 68
 fermant, guillemet anglais simple 98
 fermant, guillemet-apostrophe simple 98
 fichier auxiliaire 4
 fichier d'entrée 104
 fichier de transcription 4
 fichier externe, écrire 29
 fichier journal 4
 fichier racine 104
 fichier table des matières 4
 figures, insertion 28
 figures, notes en bas de page au sein des 34
 fin d'une lettre 109
 fin et à partir 3
 flèche droite, dans le texte 100
 flèche gauche, left, dans le texte 99
 fonctions mathématiques 82
 forme étoilée de commandes de rubricage 21
 forme étoilée des commandes
 d'environnement 58

forme étoilée, définir de nouvelles commandes . . 56
 forme en * de commandes de rubricage 21
 forme en * des commandes d'environnement 58
 forme en *, définir de nouvelles commandes 56
 formules en ligne 33
 formules mathématiques 70
 formules, environnement pour 28
 formules, math 70
 fragile, commande 62
 français, guillemets 98

G

gauche, accolade, dans le texte 99
 gauche, flèche, dans le texte 99
 gauche, guillemet-apostrophe simple de 98
 globales, options 10
 glossaires 106
 Grand rond, symbole, dans le texte 98
 gras 12
 gras tapuscrit, éviter 25
 grave accent 101
 guillemet droit double sur la ligne de base 100
 guillemet droit simple sur la ligne de base 100
 guillemet-apostrophe de droite double 100
 guillemet-apostrophe double culbuté 100
 guillemet-apostrophe simple 100
 guillemet-apostrophe simple culbuté 100
 guillemet-virgule inférieur double 98
 guillemet-virgule inférieur simple 98
 guillemets double 98
 guillemets français 98
 guillemets simple 98
 guillemets-virgule inférieurs, simple et double . . 98

I

i sans point 101
 i sans point en math 84
 Icelandic thorn 103
 ici, placer les flottants 18
 images, créer 34
 implémentations de TeX 4
 index 106
 indice inférieur 70
 indice supérieur 70
 inférieur à, symbole dans le texte 99
 inférieur, guillemet-virgule double 98
 inférieur, guillemet-virgule simple 98
 insérer des figures 28
 interligne double 14
 invite '*' 113
 italique 12

J

j sans point	101
j sans point en math	84
jsutifier à gauche du texte, environnement pour	30
justifié à gauche du texte	30
justification, drapeau droit	30
justification, drapeau gauche	30
justifier à droite, environnement pour	30
justifier le texte à droite	30

K

Knuth, Donald E.	2, 3
-----------------------	------

L

l polonais	102
L'équipe de projet L ^A T _E X	2
Lamport T _E X	3
Lamport, Leslie	2, 3
L ^A T _E X aperçu	3
L ^A T _E X, logo	98
L ^A T _E X2e logo	98
Lettre grecques	71
lettre ij, Holandais	102
lettres calligraphiques pour les mathématiques	12
lettres de script pour les mathématiques	12
lettres latines non diacritées	102
lettres, écrire des	107
lettres, commencer	110
lettres, salutation	109
ligne de commande	113
lignes dans des tableaux	41
list of figures file	4
list of tables file	4
liste des personnes en copie, dans une lettre ...	108
listes à puces	31
listes d'articles	31
listes d'articles, générique	32
listes de description, créer	25
listes non ordonnées	31
lists étiquetée, créer	25
lists of items, numbered	26
livre, symbole	98
logo L ^A T _E X	98
logo L ^A T _E X2e	98
logo T _E X	98
longueur	66
longueur, ajouter à une	67
longueur, commande de	66
longueur, régler une	66
longueurs prédéfinies	67
longueurs, définir de nouvelles	57
longueurs, définir et utiliser	66
LuaT _E X	5

M

m, largeur d'un	66
macron, diacritique	101
macron-souscrit, diacritique	101
Madsen, Lars	27
makeidx , paquetage	106
makeindex , programme	106
marque déposée, symbole	100
marque de fabrique, symbole	100
masculin, symbole ordinal	99
mathématiques gras	12
mathématiques, gras	12
math, accent aigu	83
math, accent barre suscrite	83
math, accent brève	83
math, accent grave	83
math, accent macron	83
math, accent point suscrit	83
math, accent rond-en-chef	84
math, accents	83
math, fonctions	82
math, miscellanées	84
matières, fichier	4
Millimètre	66
minipage, créer une	34
miscellanées en math	84
mise en page, les paramètres de	16
modèle de document	114
mode gauche-à-droite (left-to-right)	87
mode LR	87
mode math	87
mode mathématique, entrer	70
mode mathématique, espacement	84
mode paragraphe	87
modes	87
monnaie, dollar	99
monnaie, euro	99
moteur pdfT _E X	4
moteurs, T _E X	4
mpfootnote, compteur	53
mu, math unit	66
multind , paquetage	106
multiplication discrétionnaire	84

N

note en bas de page, créer	53
note en bas de page, symboles au lieu de numéros	53
notes dans la marge	68
notes en bas de page au sein des figures	34
Notes en bas de page, au sein d'une minipage ..	53
Notes en bas de page, dans un tableau	54
notes en bas de page, paramètres	55
Notes en marge	68
nouvelle commandes, définir	56
nouvelle ligne, commencer une	49
nouvelle ligne, commencer une (mode paragraphe)	49
nouvelle page, commencer	52
numéro d'équation, omettre	28
numéro d'équation, référence croisée	22
numéro de figure, référence croisée	22
numéro de note en bas de page, référence croisée	22
numéro de page, référence croisée	22
numéro de rubrique, référence croisée	22
numéros de rubrique, composition	21
numérotation de page, style	89

O

o e liés	103
o, e dans l'o	103
obèle double, dans le texte	99
obèle, dans le texte	97, 99
ogonek, diacritique	101
options de classe	9
options de classe de document	9
options globales	9, 10
options, classe de document	9
Ordinaux espagnols, féminin et masculin	99
Ordinaux, féminin et masculin	99
oslash	103
ouvrant, , guillemet-apostrophe simple	98

P

page de flottants	18
page styles	89
page, paramètres de mise en page	16
pages de titre, création	47
pages, saut	52
paquetage de macro, L ^A T _E X en tant que	3
paquetage float	18
paquetage setspace	14
paquetage textcomp	12
paquetages, le chargement	10
paragraphe, symbole	98
paragraphe	68
paramètre positionnel	56, 58
paramètres, la mise en page	16

paramètres, pour notes en bas de page	55
partitionner le fichier d'entrée	104
patron de document	114
patron, beamer	114
patron, book	114
patron, TUGboat	115
pdfT _E X	4
pdfT _E X, moteur	4
pièces jointes, liste	109
pica	66
Pied-de-mouche	98
pistion, dans un dessin picture	34
placement des flottants	17
poésie, un environnement pour la	48
Point	66
point centré, dans le texte	99
point d'exclamation culbuté	99
point d'interrogation culbuté	100
point Didot	66
point en chef, diacritique	101
Point proportionné (Scaled point)	66
point suscrit, diacritique	101
points de suspension	98
police de largeur fixe	12
police inclinée	12
police machine à écrire	12
police oblique	12
police petites capitales	12
police romaine	12
police, symboles par leur numéro	97
polices de caractères	11
polices OpenType	4
polices TrueType	4
polices, commandes pour déclarer de nouvelles	61
postscriptum, dans une lettre	110
préambule, définition	3
prédéfinies, longueurs	67
prononciation	3
puce, dans le texte	99

Q

quad	84
------------	----

R

référence bibliographique, clef	46
référence croisée avec numéro de page	22
référence croisée, symbolique	22
références croisées	22
réglage de compteurs	65
réservés, caractères	97
résumé	23
racine, fichier	104
rapports de bogues	2
redefining environnements	58
remarks in the marge	68
renforcement des paragraphes	
dans du texte cité	38
renforcement des paragraphes dans un	
texte cité, omettre	38
Renforcement des paragraphes, au	
sein d'une minipage	34
renforcement, forcer	68
renforcement, supprimer	68
renvoi en aval, résolution	4
renvoi, résolution	4
robuste, commande	62
rond en chef, diacritique	101
rond, grand symbole, dans le texte	98
rubricage	21

S

s long	103
sans serif	12
saut à la ligne	49
saut de ligne	49
saut de ligne, forcer	49
saut de ligne, sortie semblable à	
l'entrée pour le	49
saut de page	52
saut de page, empêcher	52
saut de page, forcer	52
sauter à la ligne	49
sauts de ligne, empêcher	51
sauts de ligne, forcer	51
section, symbole	98
<code>showidx</code> , paquetage	106
signes diacritiques	100
simple, guillemet-apostrophe	100
simple, guillemet-apostrophe culbuté	100
simuler du texte tapuscrit	48
sortie à deux colonnes	15
sortie à une colonne	15
souligné, dans le texte	100
source logiciel, composer du	48
spéciaux, caractères	97
spécificateur, placement de flottants	17
speciaux, caractères	102
sterling, symbole	98
style de numérotation de page	89
styles de police	11

styles de texte	11
styles typographiques	11
styles, page	89
supérieur à, symbole dans le texte	99
suspension, points de	98
symbole de multiplication, saut de	
ligne discretionnaire	84
symbole droit d'auteur	97
symbole marque déposée	100
symbole marque de fabrique	100
symbole puce	72
symbole vecteur en math	84
symboles d'une police, par leur numéro	97
symboles mathématiques	71
symboles texte	97
symbols, mathématiques	71

T

table des matière, création	105
table of contents entry, manually adding	105
tableaux mathématiques	24
tableaux, créer	40
taille de conception, au sein d'une	
définition de police	62
tailles de police	13
tailles de texte	13
tailles des polices de caractères	13
taquets des tabulation, utiliser	38
\TeX , logo	98
texte cité avec renforcement des	
paragraphes, affichage	38
texte cité sans renforcement des	
paragraphes, affichage	38
texte drapeau droit	30
texte drapeau droit, environnement pour	30
texte drapeau gauche	30
texte drapeau gauche, environnement pour	30
texte fer à gauche	30
texte fer à gauche, environnement pour	30
texte ferré à gauche	30
texte ferré à gauche, environnement pour	30
texte sur plusieurs colonnes	15
texte tapuscrit, simulation de	48
texte verbatim	48
texte verbatim, en-ligne	48
théorème, environnement simili-	60
théorèmes, composition des	47
théorèmes, définition	60
thickspace	84
thinspace	84
thorn, lettre islandaise	103
tilde ASCII, dans le texte	98
tilde, diacritique	101
tirant suscrit, diacritique	102
tiret bas, dans le texte	100
tréma hongrois, accent	101

tréma hongrois, diacritique..... 101
trois-quarts, cadratin 100

U

unités, de longueur..... 66
UTF-8..... 4
utilisation de Bib \TeX 46

V

variables, a list of..... 64
verticale infinie, dilatation..... 93
verticale, barre dans le texte 98
verticale, espace..... 93
visible, symbole espace, dans le texte..... 100

W

wide hat accent, math 84
wide tilde accent, math 84

X

x, hauteur d'un..... 66
Xe \TeX 5
xindy, programle 106
xspace package..... 91

Index des commandes

\$	
\$	70
&	
&	41
.	
.aux fichier	4
.fd fichier	62
.lof fichier	4
.lot fichier	4
.pdf fichier	4
.toc fichier	4
.toc, fichier	105
.xdv fichier	5
[
[...] pour les arguments optionnels	5
^	
^	70
-	
-	70
\	
\ caractère de début des commandes	5
\!	84
\" (accent tréma)	100
\#	97
\\$	97
\%	97
\&	97
\' (acute accent)	101
\' (tabbing)	39
\(.....	70
\)	70
*	84
\+	39
\,	84
\-	39
\- (césure)	50
\. (dot-over accent)	101
\/	91
\:	84
\;	84
\<	39
\= (macron accent)	101
\= (tabbing)	39
\>	39, 84
\> (tabbing)	39
\@fnsymbol	53
\[.....	70
\]	70
\^	97
\^ (accent circonflexe)	101
_	97
\` (accent grave)	101
\` (tabbing)	39
\ (for eqnarray)	27
\ (pour center)	25
\ (pour flushright)	30
\ (pour les objets \shortstack)	38
\ (tabbing)	39
\ for flushleft	30
\ force un saut de ligne	49
\ pour les lettres	107
\ pour tabular	41
\ pour verse	48
\ * (pour eqnarray)	27
\{	97
\}	97
\ 	71
\~	97
\~ (accent tilde)	101
\a (tabbing)	39
\a' (acute accent in tabbing)	39
\a= (macron accent in tabbing)	39
\a' (grave accent in tabbing)	39
\aa (á)	102
\AA (Å)	102
\acute	83
\addcontentsline{ext}{unit}{texte}	105
\address	108
\addtocontents{ext}{texte}	105
\addtocounter	65
\addtolength	67
\addvspace	92
\ae (æ)	102
\AE (Æ)	102
\aleph	71
\aline	21
\alph	64
\Alph	64
\Alph example	27
\alpha	71
\alsiname	106
\amalg	71
\angle	71
\appendix	21
\approx	71
\arabic	64

<code>\arccos</code>	82	<code>\capitalogonek</code>	101
<code>\arcsin</code>	82	<code>\capitalring</code>	101
<code>\arctan</code>	82	<code>\capitaltie</code>	102
<code>\arg</code>	82	<code>\capitaltilde</code>	101
<code>\arraycolsep</code>	24	<code>\caption</code>	29, 40
<code>\arrayrulewidth</code>	43	<code>\cc</code>	108
<code>\arraystretch</code>	43	<code>\cdot</code>	72
<code>\ast</code>	71	<code>\cdots</code>	85
<code>\asymp</code>	71	<code>\centering</code>	25
<code>\b</code> (accent barre-souscrite)	101	<code>\chapter</code> (seulement pour classes report et book)	21
<code>\backslash</code>	71, 97	<code>\check</code>	83
<code>\bar</code>	83	<code>\chi</code>	72
<code>\baselineskip</code>	14	<code>\circ</code>	72
<code>\baselinestretch</code>	14	<code>\circle</code>	35
<code>\begin</code>	23	<code>\cite</code>	46
<code>\beta</code>	71	<code>\cleardoublepage</code>	52
<code>\bf</code>	12	<code>\clearpage</code>	52
<code>\bfseries</code>	11	<code>\cline</code>	45
<code>\bibitem</code>	46	<code>\closing</code>	109
<code>\bibliography</code>	46	<code>\clubsuit</code>	72
<code>\bibliographystyle</code>	46	<code>\columnsep</code>	15
<code>\bigcap</code>	71	<code>\columnseprule</code>	15
<code>\bigcirc</code>	71	<code>\columnwidth</code>	15
<code>\bigcup</code>	71	<code>\complement</code>	72
<code>\bigodot</code>	71	<code>\cong</code>	72
<code>\bigoplus</code>	71	<code>\contentsline</code>	105
<code>\bigotimes</code>	72	<code>\coprod</code>	72
<code>\bigskip</code>	92	<code>\copyright</code>	97
<code>\bigskipamount</code>	92	<code>\cos</code>	82
<code>\bigsqcup</code>	72	<code>\cosh</code>	82
<code>\bigtriangledown</code>	72	<code>\cot</code>	82
<code>\bigtriangleup</code>	72	<code>\coth</code>	82
<code>\biguplus</code>	72	<code>\csc</code>	82
<code>\bigvee</code>	72	<code>\cup</code>	72
<code>\bigwedge</code>	72	<code>\d</code> (dot-under accent)	101
<code>\bmod</code>	82	<code>\dag</code>	97
<code>\boldmath</code>	70	<code>\dagger</code>	72
<code>\bot</code>	72	<code>\dashbox</code>	36
<code>\bottomfraction</code>	19	<code>\dashv</code>	73
<code>\bottomnumber</code>	19	<code>\day</code>	65
<code>\bowtie</code>	72	<code>\dblfloatpagefraction</code>	15
<code>\Box</code>	72	<code>\dblfloatsep</code>	15
<code>\breve</code>	83	<code>\dbltextfloatsep</code>	15
<code>\bullet</code>	72	<code>\dbltopfraction</code>	15
<code>\c</code> (cédille)	101	<code>\ddag</code>	97
<code>\cal</code>	12	<code>\ddagger</code>	73
<code>\cap</code>	72	<code>\ddot</code>	83
<code>\capitalacute</code>	101	<code>\ddots</code>	85
<code>\capitalbreve</code>	102	<code>\deg</code>	82
<code>\capitalcaron</code>	102	<code>\delta</code>	73
<code>\capitalcedilla</code>	101	<code>\Delta</code>	73
<code>\capitalcircumflex</code>	101	<code>\depth</code>	67
<code>\capitaldieresis</code>	100	<code>\det</code>	82
<code>\capitaldotaccent</code>	101	<code>\dh</code> (ð)	102
<code>\capitalgrave</code>	101	<code>\DH</code> (Ð)	102
<code>\capitalhungarumlaut</code>	101	<code>\diamond</code>	73
<code>\capitalmacron</code>	101	<code>\Diamond</code>	73
<code>\capitalnewtie</code>	102		

<code>\diamondsuit</code>	73	<code>\frame</code>	36
<code>\dim</code>	82	<code>\framebox</code>	36, 94
<code>\displaystyle</code>	70	<code>\frenchspacing</code>	91
<code>\div</code>	73	<code>\frown</code>	73
<code>\dj</code>	102	<code>\fussy</code>	51
<code>\DJ</code>	102	<code>\gamma</code>	73
<code>\documentclass</code>	9	<code>\Gamma</code>	73
<code>\dot</code>	83	<code>\gcd</code>	83
<code>\doteq</code>	73	<code>\ge</code>	73
<code>\dotfill</code>	92	<code>\geq</code>	73
<code>\dots</code>	98	<code>\gets</code>	74
<code>\doublerulesep</code>	43	<code>\gg</code>	74
<code>\downarrow</code>	73	<code>\glossary</code>	106
<code>\Downarrow</code>	73	<code>\glossaryentry</code>	106
<code>\ell</code>	73	<code>\grave</code>	83
<code>\em</code>	12	<code>\guillemotleft</code> («).....	98
<code>\emph</code>	11	<code>\guillemotright</code> (»).....	98
<code>\emptyset</code>	73	<code>\guilsinglleft</code> (‹).....	98
<code>\encl</code>	109	<code>\guilsinglright</code> (›).....	98
<code>\end</code>	23	<code>\H</code> (tréma hongrois).....	101
<code>\enlargethispage</code>	52	<code>\hat</code>	83
<code>\enumi</code>	27	<code>\hbar</code>	74
<code>\enumii</code>	27	<code>\headheight</code>	16
<code>\enumiii</code>	27	<code>\headsep</code>	16
<code>\enumiv</code>	27	<code>\heartsuit</code>	74
<code>\epsilon</code>	73	<code>\height</code>	67
<code>\equiv</code>	73	<code>\hfill</code>	90
<code>\espace</code>	90	<code>\hline</code>	45
<code>\eta</code>	73	<code>\hom</code>	83
<code>\evensidemargin</code>	10	<code>\hookleftarrow</code>	74
<code>\exists</code>	73	<code>\hookrightarrow</code>	74
<code>\exp</code>	83	<code>\hrulefill</code>	92
<code>\extracolsep</code>	42	<code>\hspace</code>	17
<code>\fbox</code>	94	<code>\hspace</code>	90
<code>\fboxrule</code>	36, 94	<code>\huge</code>	13
<code>\fboxsep</code>	36, 94	<code>\Huge</code>	13
<code>\fill</code>	90	<code>\hyphenation</code>	51
<code>\flat</code>	73	<code>\i</code> (i sans point).....	101
<code>\floatpagefraction</code>	19	<code>\iff</code>	74
<code>\floatsep</code>	19	<code>\ij</code> (ij).....	102
<code>\flushbottom</code>	16	<code>\IJ</code> (IJ).....	102
<code>\fnsymbol</code>	64	<code>\il</code>	12
<code>\fnsymbol</code> , et note en bas de page.....	53	<code>\Im</code>	74
<code>\fontencoding</code>	13	<code>\imath</code>	84
<code>\fontfamily</code>	13	<code>\in</code>	74
<code>\fontseries</code>	13	<code>\include</code>	104
<code>\fontshape</code>	14	<code>\includeonly</code>	104
<code>\fontsize</code>	14	<code>\indent</code>	68
<code>\footnote</code>	53	<code>\index</code>	106
<code>\footnotemark</code>	53	<code>\indexentry</code>	106
<code>\footnoterule</code>	55	<code>\indexspace</code>	106
<code>\footnotesep</code>	55	<code>\inf</code>	83
<code>\footnotesize</code>	13	<code>\infty</code>	74
<code>\footnotetext</code>	54	<code>\input</code>	104
<code>\footskip</code>	16	<code>\int</code>	74
<code>\forall</code>	73	<code>\intertextsep</code>	19
<code>\frac</code>	85	<code>\iota</code>	74
<code>\frac{num}{dén}</code>	85	<code>\item</code>	25, 26, 31

<code>\itemindent</code>	31	<code>\liminf</code>	83
<code>\itemsep</code>	32	<code>\limsup</code>	83
<code>\itshape</code>	11	<code>\line</code>	36
<code>\j</code> (j sans point).....	101	<code>\linebreak</code>	51
<code>\jmath</code>	84	<code>\linespread</code>	14
<code>\Join</code>	74	<code>\linethickness</code>	36
<code>\k</code> (ogonek).....	101	<code>\linewidth</code>	16
<code>\kappa</code>	74	<code>\listoffigures</code>	105
<code>\ker</code>	83	<code>\listoftables</code>	105
<code>\kill</code>	39	<code>\listparindent</code>	32
<code>\l</code> (\lrcorner).....	102	<code>\ll</code>	75
<code>\L</code> (\Lrcorner).....	102	<code>\ln</code>	83
<code>\label</code>	22	<code>\lnot</code>	75
<code>\labelenumi</code>	27	<code>\location</code>	109
<code>\labelenumii</code>	27	<code>\log</code>	83
<code>\labelenumiii</code>	27	<code>\longleftarrow</code>	75
<code>\labelenumiv</code>	27	<code>\longlefttrightarrow</code>	75
<code>\labelitemi</code>	31	<code>\longmapsto</code>	75
<code>\labelitemii</code>	31	<code>\longrightarrow</code>	75
<code>\labelitemiii</code>	31	<code>\lor</code>	75
<code>\labelitemiv</code>	31	<code>\lq</code>	98
<code>\labelsep</code>	31	<code>\makebox</code>	94
<code>\labelwidth</code>	31	<code>\makebox (picture)</code>	35
<code>\lambda</code>	74	<code>\makeglossary</code>	106
<code>\Lambda</code>	74	<code>\makeindex</code>	106
<code>\land</code>	74	<code>\makelabels</code>	109
<code>\langle</code>	74	<code>\mapsto</code>	75
<code>\large</code>	13	<code>\marginpar</code>	68
<code>\Large</code>	13	<code>\marginparpush</code>	69
<code>\LARGE</code>	13	<code>\marginparsep</code>	69
<code>\LaTeX</code>	98	<code>\marginparwidth</code>	69
<code>\LaTeXe</code>	98	<code>\mathbf</code>	12
<code>\lbrace</code>	74	<code>\mathcal</code>	12
<code>\lbrack</code>	74	<code>\mathnormal</code>	12
<code>\lceil</code>	74	<code>\mathring</code>	84
<code>\ldots</code>	98	<code>\mathrm</code>	11
<code>\le</code>	74	<code>\mathsf</code>	12
<code>\leadsto</code>	74	<code>\mathtt</code>	12
<code>\left delim1 ... \right delim2</code>	85	<code>\mathversion</code>	12
<code>\leftarrow</code>	75	<code>\max</code>	83
<code>\Leftarrow</code>	74	<code>\mbox</code>	94
<code>\lefteqn</code>	28	<code>\mdseries</code>	11
<code>\leftharpoondown</code>	75	<code>\medskip</code>	92
<code>\leftharpoonup</code>	75	<code>\medskipamount</code>	92
<code>\leftmargin</code>	31	<code>\medspace</code>	84
<code>\leftmargini</code>	31	<code>\mho</code>	75
<code>\leftmarginii</code>	31	<code>\mid</code>	76
<code>\leftmarginiii</code>	31	<code>\min</code>	83
<code>\leftmarginiv</code>	31	<code>\models</code>	76
<code>\leftmarginv</code>	31	<code>\month</code>	65
<code>\leftmarginvi</code>	31	<code>\mp</code>	76
<code>\leftrightarrow</code>	75	<code>\mu</code>	76
<code>\Leftrightarrow</code>	75	<code>\multicolumn</code>	44
<code>\leq</code>	75	<code>\multiput</code>	37
<code>\lfloor</code>	75	<code>\nabla</code>	76
<code>\lg</code>	83	<code>\name</code>	110
<code>\lhd</code>	75	<code>\natural</code>	76
<code>\lim</code>	83	<code>\ne</code>	76

<code>\narrow</code>	76	<code>\parindent</code>	34, 68
<code>\neg</code>	76	<code>\parsep</code>	32
<code>\neq</code>	76	<code>\parskip</code>	68
<code>\newcommand</code>	56	<code>\parskip exemple</code>	32
<code>\newcounter</code>	57	<code>\part</code>	21
<code>\newenvironment</code>	58	<code>\partial</code>	77
<code>\newfont</code>	61	<code>\partopsep</code>	32
<code>\newlength</code>	57	<code>\perp</code>	77
<code>\newline</code>	49	<code>\phi</code>	77
<code>\newpage</code>	52	<code>\pi</code>	77
<code>\newsavebox</code>	58	<code>\Pi</code>	77
<code>\newtheorem</code>	60	<code>\pm</code>	77
<code>\newtie</code>	102	<code>\pmod</code>	83
<code>\ng</code>	102	<code>\poptabs</code>	39
<code>\NG</code>	102	<code>\pounds</code>	98
<code>\ni</code>	76	<code>\Pr</code>	83
<code>\nocite</code>	46	<code>\prec</code>	77
<code>\nofiles</code>	105	<code>\preceq</code>	77
<code>\noindent</code>	68	<code>\prime</code>	77
<code>\nolinebreak</code>	51	<code>\printindex</code>	106
<code>\nonfrenchspacing</code>	91	<code>\prod</code>	77
<code>\nonumber</code>	28	<code>\propto</code>	77
<code>\nopagebreak</code>	52	<code>\protect</code>	62
<code>\normalfont</code>	11	<code>\ps</code>	110
<code>\normalmarginpar</code>	69	<code>\psi</code>	77
<code>\normalsize</code>	13	<code>\Psi</code>	77
<code>\not</code>	76	<code>\pushtabs</code>	39
<code>\notin</code>	76	<code>\put</code>	37
<code>\nu</code>	76	<code>\quad</code>	84
<code>\narrow</code>	76	<code>\quotedblbase (")</code>	98
<code>\o (\varnothing)</code>	103	<code>\quotesinglbase (,)</code>	98
<code>\O (\emptyset)</code>	103	<code>\r (ring accent)</code>	101
<code>\obeycr</code>	49	<code>\raggedbottom</code>	16
<code>\oddsidemargin</code>	10	<code>\raggedleft</code>	30
<code>\odot</code>	76	<code>\raggedright</code>	30
<code>\oe (\o)</code>	103	<code>\raisebox</code>	95
<code>\OE (\O)</code>	103	<code>\rangle</code>	77
<code>\oint</code>	76	<code>\rbrace</code>	77
<code>\oldstylenums</code>	12	<code>\rbrack</code>	77
<code>\omega</code>	76	<code>\rceil</code>	77
<code>\Omega</code>	76	<code>\Re</code>	77
<code>\ominus</code>	76	<code>\ref</code>	22
<code>\onecolumn</code>	15	<code>\refstepcounter</code>	65
<code>\opening</code>	110	<code>\renewenvironment</code>	58
<code>\oplus</code>	76	<code>\restorecr</code>	49
<code>\oslash</code>	76	<code>\restriction</code>	77
<code>\otimes</code>	76	<code>\revertset</code>	78
<code>\oval</code>	37	<code>\reversemarginpar</code>	69
<code>\overbrace{texte}</code>	85	<code>\rfloor</code>	78
<code>\overline{texte}</code>	85	<code>\rhd</code>	78
<code>\owns</code>	77	<code>\rho</code>	78
<code>\P</code>	98	<code>\right</code>	85
<code>\pagebreak</code>	52	<code>\rightarrow</code>	78
<code>\pagenumbering</code>	89	<code>\Rightarrow</code>	78
<code>\pageref</code>	22	<code>\rightharpoondown</code>	78
<code>\paragraph</code>	21	<code>\rightharpoonup</code>	78
<code>\parallel</code>	77	<code>\rightleftharpoons</code>	78
<code>\parbox</code>	95	<code>\rightmargin</code>	32

<code>\rm</code>	12	<code>\succ</code>	79
<code>\rmfamily</code>	11	<code>\succeq</code>	79
<code>\roman</code>	64	<code>\sum</code>	79
<code>\Roman</code>	64	<code>\sup</code>	83
<code>\rq</code>	98	<code>\suppressfloats</code>	18
<code>\rule</code>	103	<code>\supset</code>	79
<code>\S</code>	98	<code>\supseteq</code>	79
<code>\sautdeligne</code>	90	<code>\surd</code>	79
<code>\savebox</code>	96	<code>\swarrow</code>	79
<code>\sbox</code>	96	<code>\symbol</code>	97
<code>\sc</code>	12	<code>\t</code> (tie-after accent).....	102
<code>\scriptsize</code>	13	<code>\tabbingsep</code>	40
<code>\scshape</code>	11	<code>\tabcolsep</code>	43
<code>\searrow</code>	78	<code>\tableofcontents</code>	105
<code>\sec</code>	83	<code>\tabulation</code>	90
<code>\section</code>	21	<code>\tan</code>	83
<code>\seenname</code>	106	<code>\tanh</code>	83
<code>\selectfont</code>	14	<code>\tau</code>	79
<code>\setcounter</code>	65	<code>\telephone</code>	111
<code>\setlength</code>	66	<code>\TeX</code>	98
<code>\setminus</code>	78	<code>\textascendercompwordmark</code>	99
<code>\settodepth</code>	67	<code>\textasciicircum</code>	98
<code>\settoheight</code>	67	<code>\textasciitilde</code>	98
<code>\settowidth</code>	67	<code>\textasteriskcentered</code>	98
<code>\sf</code>	12	<code>\textbackslash</code>	98
<code>\sffamily</code>	11	<code>\textbar</code>	98
<code>\sharp</code>	78	<code>\textbardbl</code>	98
<code>\shortstack</code>	37	<code>\textbf</code>	11
<code>\sigma</code>	78	<code>\textbigcircle</code>	98
<code>\Sigma</code>	78	<code>\textbraceleft</code>	99
<code>\signature</code>	110	<code>\textbraceright</code>	99
<code>\sim</code>	78	<code>\textbullet</code>	99
<code>\simeq</code>	78	<code>\textcapitalcompwordmark</code>	99
<code>\sin</code>	83	<code>\textcircled{lettre}</code>	99
<code>\sinh</code>	83	<code>\textcompwordmark</code>	99
<code>\sl</code>	12	<code>\textcopyright</code>	97
<code>\slshape</code>	11	<code>\textdagger</code>	99
<code>\small</code>	13	<code>\textdaggerdbl</code>	99
<code>\smallint</code>	78	<code>\textdollar</code> (ou <code>\\$</code>).....	99
<code>\smallskip</code>	93	<code>\textellipsis</code>	98
<code>\smallskipamount</code>	93	<code>\textemdash</code> (ou <code>---</code>).....	99
<code>\smile</code>	78	<code>\textendash</code> (ou <code>--</code>).....	99
<code>\spadesuit</code>	79	<code>\texteuro</code>	99
<code>\sqcap</code>	79	<code>\textexclamdown</code> (ou <code>!'</code>).....	99
<code>\sqcup</code>	79	<code>\textfloatsep</code>	19
<code>\sqrt[nième]{arg}</code>	85	<code>\textfraction</code>	19
<code>\sqsubset</code>	79	<code>\textgreater</code>	99
<code>\sqsubseteq</code>	79	<code>\textheight</code>	16
<code>\sqsupset</code>	79	<code>\textit</code>	11
<code>\sqsupseteq</code>	79	<code>\textleftarrow</code>	99
<code>\SS</code> (SS).....	103	<code>\textless</code>	99
<code>\ss</code> (ß).....	103	<code>\textmd</code>	11
<code>\stackrel{texte}{relation}</code>	85	<code>\textnormal</code>	11
<code>\star</code>	79	<code>\textordfeminine</code>	99
<code>\stepcounter</code>	65	<code>\textordmasculine</code>	99
<code>\stop</code>	113	<code>\textparagraph</code>	98
<code>\subset</code>	79	<code>\textperiodcentered</code>	99
<code>\subseteq</code>	79	<code>\textquestiondown</code> (ou <code>?'</code>).....	100
<code>\subsetneq</code>	79		
<code>\subsubsection</code>	21		

C

cc	66
center, environnement	25
classe article	9
classe book	9
classe letter	9
classe report	9
classe slides	9
cm	66
commande dvipdfmx	4
commande dvips	4
commande dvitype	4
commande latex	4
commande pdflatex	4
commande xdvi	4

D

dd	66
description, environnement	25
displaymath, environnement	26, 70
document, environnement	26

E

em	66
enumerate, environnement	26
environnement abstract	23
environnement array	24
environnement center	25
environnement description	25
environnement displaymath	26, 70
environnement document	26
environnement enumerate	26
environnement eqnarray	27
environnement equation	28, 70
environnement figure	28
environnement filecontents	29
environnement filecontents*	29
environnement flushleft	30
environnement flushright	30
environnement itemize	31
environnement letter	32
environnement list	32
environnement math	33, 70
environnement minipage	34
environnement picture	34
environnement quotation	38
environnement quote	38
environnement tabbing	38
environnement table	40
environnement tabular	41
environnement thebibliography	45
environnement theorem	47
environnement titlepage	47
environnement verbatim	48
environnement verse	48
eqnarray, environnement	27

equation, environnement	28, 70
etex, commande	4
ex	66

F

fichier .dvi	4
fichier .log	4
figure, environnement	28
filecontents*, environnement	29
filecontents, environnement	29
flushleft, environnement	30
flushright, environnement	30

I

in	66
itemize, environnement	31

L

latexrefman-discuss@gna.org adresse mél	2
letter, environnement	32
list, environnement	32
lR box	35
lrbox	94
lualatex commande	5

M

math, environnement	33, 70
minipage, environnement	34
mm	66
mu	66

O

option 10pt	9
option 11pt	9
option 12pt	9
option a4paper	9
option a5paper	9
option b5paper	9
option draft	9
option executivepaper	9
option final	9
option fleqn	9
option landscape	9
option legalpaper	9
option leqno	9
option letterpaper	9
option notitlepage	9
option onecolumn	9
option oneside	9
option openany	9
option openbib	9
option openright	9
option titlepage	9

option `twocolumn` 9
option `twoside` 9

P

`pc` 66
`picture`, environnement 34
`pouce (inch)` 66
`pt` 66

Q

`quotation`, environnement 38
`quote`, environnement 38

S

`secnumdepth counter` 21
`sp` 66

T

`tabbing`, environnement 38
`table`, environnement 40
`tabular`, environnement 41
`textcomp`, paquetage 97
`thebibliography`, environnement 45
`theorem`, environnement 47
`titlepage`, environnement 47

V

`verbatim`, environnement 48
`verse`, environnement 48

X

`xdvipdfmx` 5
`xelatex` commande 5