

AIDE-MÉMOIRE L^AT_EX

Vincent SEGUIN <seguin@via.ecp.fr> (8 novembre 1998)

1 Structures du document

1.1 Format d'un document type L^AT_EX 2_ε

<pre>\documentclass[a4paper,french,10pt]{article} \usepackage[T1]{fontenc} \usepackage{babel} ... autres packages... \title{Aide-mémoire L^AT_EX} \author{Filou \and Oli \and Sandrine \and Vador} \date{35 mai 1998} ... commandes et paramètres propres au document... \begin{document} ... texte... \end{document}</pre>	déclaration de la <i>classe de document</i> packages titre auteur(s) date (automatique si non spécifiée) texte du document
---	---

Classes de document : `article`, `report`, `letter`, `book`, `slides`.

Options standard : `10pt`, `11pt`, `12pt`, `a4paper`, `french`, `twocolumn`, `twoside`. Les options de `\documentclass` sont transmises à toutes les commandes `\usepackage`.

1.2 Packages usuels

`amsmath` : extensions de l'*American Mathematical Society*. `amsmath` inclut entre autre les extensions `amsbsy`, `amscd`, `amfonts`, `amssymb` et `amstext`.

`array` : étend les options des environnements `array` et `tabular` ;

`babel` : renomme les noms de chapitres, dates et autres textes insérés par L^AT_EX dans la langue choisie. `babel` accorde aussi la typographie aux règles en vigueur selon les pays. Les commandes `\selectlanguage{langue}` et `\iflanguage{langue}{expression-vrai}{expression-faux}` permettent d'écrire des documents en plusieurs langues ou pouvant être compilés dans des langues différentes. Les langues supportés à l'heure actuelle incluent : `catalan`, `croatian`, `czech`, `danish`, `dutch`, `english`, `esperanto`, `finnish`, `french`, `galician`, `german`, `italian`, `magyar`, `norsk`, `polish`, `portuges`, `romanian`, `russian`, `slovak`, `slovene`, `spanish`, `turkish`, et les variantes `american`, `austrian`, `brazil`, `nynorsk`, `germanb` et `frenchb`.

`color` : permet l'utilisation des couleurs ;

`draftcopy` : imprime "draft" (ou "brouillon") en fond de page ;

`endnotes` : renvoie les notes en fin de document ;

`graphics` : définit plusieurs commandes de manipulation de boîtes et d'importation de graphismes ;

`fancybox` : définit plusieurs commandes d'encadrement supplémentaires ;

`fontenc` : avec l'option `T1`, autorise le compilateur à utiliser le nouveau format d'encodage de fontes. Cette option n'est pas activée par défaut pour des raisons de compatibilité avec les versions de L^AT_EX précédentes, mais est à utiliser systématiquement.

`ifthen` : permet d'écrire des structures de décision et des boucles en L^AT_EX ;

`latexsym` : définit de nombreux symboles mathématiques ;

`marvosym` : ensemble de commandes facilitant l'usage de la fonte "Martin Vogel", composée de symboles divers, dont le symbole de l'euro € (`\EUR`).

`shadow` : définit la commande `\shabox`, qui permet de faire des boîtes ombrées.

1.3 Structure hiérarchique

<code>\part{titre}</code>	partie
<code>\chapter{titre}</code>	chapitre (<code>report</code> et <code>book</code> seulement)
<code>\section{titre}</code>	section
<code>\subsection{titre}</code>	sous-section
<code>\subsubsection{titre}</code>	sous-section (niveau 2)
<code>\paragraph{titre}</code>	sous-section (niveau 3)
<code>\subparagraph{titre}</code>	sous-section (niveau 4)

La commande `\appendix` transforme les chapitres suivants en appendices, au sein d'une partie.

`\paragraph` et `\subparagraph` sont nommés ainsi pour des raisons historiques, mais n’ont rien à voir avec les paragraphes proprement dits.

Les formes étoilées de ces commandes ne sont pas numérotées et ne créent pas d’entrées dans la table des matières.

1.4 Insertion de fichiers

`\input{fichier}` : est remplacé par le contenu de *fichier.tex*. *fichier* peut lui-même inclure une commande `\input`.

`\include{fichier}` : insère *fichier.tex* ou sa version pré-compilée. Le fichier ne sera pas recompilé s’il n’a pas été modifié depuis la dernière compilation. `\include` ne peut apparaître dans l’en-tête, et le texte inséré doit former un ensemble de pages indépendantes.

`\includeonly{fichier1, fichier2, ...}` : placé dans l’en-tête, restreint la liste des fichiers insérés à l’aide de la commande `\include`. Les fichiers absents de la liste ne seront pas remis à jour même s’ils ont été modifiés depuis la dernière compilation.

2 Commandes de base

2.1 Caractères de commandes

#	paramètre de macro	~	espace insécable	^ et _	exposant et indice
\$	mode mathématique	&	alignement de tableau	{...}	groupe
%	commentaire	\	commande		

Les caractères de ponctuation (séparateurs) sont : : ; , ? ! ‘ ’ () [] - / * @ et .

2.2 Caractères spéciaux

Les caractères codés en ISO-8859-1 sont compris par le compilateur : les lettres accentuées usuelles peuvent être saisies directement. *i* et *j* doivent perdre leurs points s’ils sont accentués. Les commandes `\i` et `\j` produisent “i” et “j” à cet effet.

ò \‘{o}	ô \^ {o}	ó \. {o}	õ \~ {o}	â \aa	æ \ae	ß \ss
ó \’ {o}	ö \v {o}	ô \d {o}	ø \c {o}	Å \AA	Æ \AE	¿ ?‘
ö \" {o}	ø \u {o}	ō \= {o}	ø \o	ł \l	œ \oe	¡ !‘
ó \H {o}	ôo \t {oo}	o \b {o}	Ø \O	Ł \L	Œ \OE	
‡ \dag	¶ \P	# \#	\ \backslash	- (cerf-volant)	-	{ \{
‡ \ddag	& \&	\$ \\$	© \copyright	- (14-27)	--	} \}
§ \S	_ _	% \%	£ \pounds	— (ponctuation)	---	

2.3 Commandes usuelles

<code>\maketitle</code>	produit un titre
<code>\TeX, \LaTeX, \LaTeXe, \AllTeX</code>	TeX, L ^A TeX, L ^A TeX 2 _ε , (L ^A)TeX
<code>\verb!...!</code>	mode <i>verbatim</i> — “!” est un caractère quelconque
<code>\protect commande</code>	protège une commande fragile
<code>\begin{env}... \end{env}</code>	bloc inclus dans un environnement
<code>\ensuremath{...}</code>	force le mode mathématique

2.4 Fontes et styles

<code>\textrm{...}</code>	<code>{\rmfamily ...}</code>	fonte type roman
<code>\textsf{...}</code>	<code>{\sffamily ...}</code>	fonte type sans serif
<code>\texttt{...}</code>	<code>{\ttfamily ...}</code>	fonte type typewriter
<code>\textmd{...}</code>	<code>{\mdseries ...}</code>	corps moyen (medium)
<code>\textbf{...}</code>	<code>{\bfseries ...}</code>	corps gras (boldface)
<code>\textup{...}</code>	<code>{\upshape ...}</code>	forme droite (upright)
<code>\textit{...}</code>	<code>{\itshape ...}</code>	forme <i>italique</i> (italic)
<code>\textsl{...}</code>	<code>{\slshape ...}</code>	forme <i>penché</i> (slanted)
<code>\textsc{...}</code>	<code>{\scshape ...}</code>	forme PETITES MAJUSCULES (small caps)
<code>\emph{...}</code>	<code>{\em ...}</code>	mis en <i>évidence</i> (emphasized)
<code>\textnormal{...}</code>	<code>{\normalfont ...}</code>	forme normale

Tailles: `tiny` `scriptsize` `footnotesize` `small` `normalsize` `large` `Large` `LARGE` `huge` `Huge` (exemple d'utilisation: `{\large large}`).

2.5 Environnements

2.5.1 Alignements

<code>{\centering ...}</code>	<code>\begin{center} ... \end{center}</code>	centré
<code>{\raggedright ...}</code>	<code>\begin{flushleft} ... \end{flushleft}</code>	aligné à gauche
<code>{\raggedleft ...}</code>	<code>\begin{flushright} ... \end{flushright}</code>	aligné à droite

2.5.2 Listes

<code>\begin{itemize} ... \item ... \end{itemize}</code>	liste avec puces
<code>\begin{description} ... \item[item] ... \end{description}</code>	liste de descriptions
<code>\begin{enumerate} ... \item ... \end{enumerate}</code>	énumération

2.5.3 L'environnement tabular et le format des colonnes

```
\begin{tabular}{format colonnes}
  case(1,1) & case(2,1) \\
  case(2,1) & case(2,2) \\
\end{tabular}
```

`\hline` dessine une ligne horizontale.

format colonnes est une série d'indicateurs de format, par exemple `{l|p{3cm}lr|l}`. Les indicateurs "m", "b", "!", ">" et "<" nécessitent le package `array`.

<i>indicateur de format</i>	<i>signification</i>
<code>l</code>	colonne alignée à gauche
<code>r</code>	colonne alignée à droite
<code>c</code>	colonne centrée
<code>p{largeur}</code>	équivalent à <code>\parbox[t]{largeur}</code>
<code>@{decl}</code>	remplace l'espace inter-colonnes par <i>decl</i>
<code> </code>	ligne verticale
<code>*{num}{cols}</code>	est remplacé par <i>num</i> fois le contenu de <i>cols</i>
<code>m{largeur}</code>	colonne en mode paragraphe centrée verticalement
<code>b{largeur}</code>	équivalent à <code>\parbox[b]{largeur}</code>
<code>!{decl}</code>	équivalent à <code>@</code> mais ne supprime pas l'espace inter-colonnes
<code>>{decl}</code>	suit de <code>l</code> , <code>r</code> , <code>c</code> ou <code>p</code> : insère <i>decl</i> à gauche de la colonne
<code><{decl}</code>	après <code>l</code> , <code>r</code> , <code>c</code> ou <code>p</code> : insère <i>decl</i> à droite de la colonne

2.5.4 Autres environnements

<code>abstract</code>	résumé
<code>letter{destinataire}</code>	en classe de document <code>letter</code> , lettre indépendante
<code>minipage[<i>position</i>]{<i>largeur</i>}</code>	ensemble de texte complexe à l'intérieur d'une page
<code>picture</code>	environnement de dessin
<code>quotation</code>	citation, avec indentation des paragraphes
<code>quote</code>	citation, sans indentation des paragraphes
<code>theorem</code>	théorème numéroté
<code>titlepage</code>	page de titre
<code>verbatim</code>	imprime en <code>typewriter</code> le bloc non interprété
<code>verbatim*</code>	<i>verbatim</i> , les espaces sont marquées <code>□</code>
<code>verse</code>	poésie

3 Mise en page

3.1 Espacements et sauts

<code>□</code>	espace	<code>\-</code>	emplacement de césure	<code>\\[<i>h</i>]</code>	retour à la ligne, <i>h</i> est l'interligne
<code>\,</code>	petit espace	<code>\/</code>	espace après italique	<code>*[<i>h</i>]</code>	retour à la ligne sans saut de page
<code>~</code>	espace insécable			<code>\newline</code>	saut de ligne

<code>\hspace{l}</code>	espace horizontal, ignoré en cas de saut de ligne
<code>\vspace{h}</code>	espace vertical, ignoré en cas de saut de page
<code>\hspace*{l}</code>	espace horizontal
<code>\vspace*{h}</code>	espace vertical
<code>\hfill</code>	espace élastique horizontal
<code>\vfill</code>	espace élastique vertical
<code>\hrulefill</code>	ligne élastique horizontale
<code>\dotfill</code>	points élastiques horizontaux
<code>\smallskip</code>	petit espace vertical
<code>\medskip</code>	moyen espace vertical
<code>\bigskip</code>	grand espace vertical

3.2 Aspect

<code>\sloppy</code>	justification stricte
<code>\fussy</code>	justification lâche
<code>\indent</code>	force l'indentation en début d'un paragraphe
<code>\noindent</code>	supprime l'indentation en début d'un paragraphe
<code>\pagebreak[<i>fact</i>]</code>	encourage le saut de page d'un facteur <i>fact</i>
<code>\nopagebreak[<i>fact</i>]</code>	décourage le saut de page d'un facteur <i>fact</i>
<code>\enlargethispage{h}</code>	augmente la taille de la page de <i>h</i>
<code>\enlargethispage*{h}</code>	augmente la taille de la page et compresse la page
<code>\newpage</code>	saut de page
<code>\clearpage</code>	force l'impression des tables et figures
<code>\cleardoublepage</code>	force l'impression des figures et commence une page impaire

3.3 En-têtes et pieds de page

<code>\pagestyle{style}</code>	dans le préambule, définit le style de page du document
<code>\thispagestyle{style}</code>	définit le style de la page courante
<code>\markright{droite}</code>	définit l'en-tête droit
<code>\markboth{droite}{gauche}</code>	définit les en-têtes droits et gauches
<i>style</i>	<i>signification</i>
plain	numéro de page dans le pied de page, en-tête vide
empty	en-tête et pied de page vides
headings	numéro de page et autres informations dans l'en-tête, pied de page vide
myheadings	en-tête défini par par <code>\markboth</code> ou <code>\markright</code>

4 Notes et références

4.1 Notes

<code>\footnote{note}</code>	note de bas de page
<code>\marginpar[gauche]{note}</code>	note dans la marge. <i>gauche</i> est utilisé si la note se retrouve à gauche
<code>\endnote{note}</code>	note de fin de document — nécessite le package <code>endnotes</code>

4.2 Références

<code>\label{label}</code>	définit un label
<code>\ref{label}</code>	référence un label (suivant le contexte)
<code>\pageref{label}</code>	référence la page d'un label
<code>\theobj</code>	référence un objet

Les `\the`-commandes sont définies pour de nombreux objets (`\thesection`, `\thechapter`, `\theequation`, `\thefootnote`, `\theCodelineNo`, etc. — ici `\thesection` donnera 4).

Plusieurs compilations peuvent être nécessaires pour construire les références.

4.3 Tables

<code>\tableofcontents</code>	table des matières
<code>\listoftables</code>	liste des tables
<code>\listoffigures</code>	liste des figures

5 Graphisme, tables et figures

5.1 Tables et figures








Les tables et figures sont des environnements flottants n’incluant pas forcément ce que leur nom laisse supposer. Ils sont numérotés et placés par L^AT_EX de manière différente du reste du texte.

<code>\begin{figure}[position] ... \end{figure}</code>	figure (généralement un graphe ou une image)
<code>\begin{table}[position] ... \end{table}</code>	table (généralement un tableau)
<code>\caption{...}</code>	produit une légende (dans une figure ou une table)

Les formes étoilées `figure*` et `table*` produisent des flottants en deux colonnes si l’environnement le permet. *position* est une suite de lettres décrivant le placement souhaité pour l’objet flottant, par ordre de préférence. La valeur par défaut de *position* est `tbp`.

<i>position</i>	<i>signification</i>
<code>h</code>	<i>Here</i> : la figure est placée là où elle apparaît dans le texte
<code>t</code>	<i>Top</i> : la figure est placée en haut d’une page de texte
<code>b</code>	<i>Bottom</i> : la figure est placée en bas d’une page de texte
<code>p</code>	<i>Page of floats</i> : la figure est placée sur une page ne contenant que des flottants
<code>!</code>	encourage l’option qui suit le “!”

5.2 Cadres et boîtes

<code>\mbox{...}</code>	isole une portion de texte dans une boîte (version courte)
<code>\makebox[larg][pos]{...}</code>	isole une portion de texte dans une boîte
<code>\parbox[largeur]{...}</code>	isole une portion de texte en mode paragraphe dans une boîte
<code>\newsavebox{nom}</code>	déclare une boîte
<code>\sbox{nom}{...}</code>	remplit une boîte (version courte)
<code>\savebox{nom}[larg][pos]{...}</code>	remplit une boîte
<code>\usebox{nom}</code>	utilise le contenu d’une boîte
<code>\scalebox{fact}[fact-y]{...}</code>	change l’échelle d’une boîte
<code>\resizebox{larg}{haut}{...}</code>	change la taille d’une boîte
<code>\rotatebox{angle}{...}</code>	<i>tourne</i> la boîte de <i>angle</i> degrés
<code>\reflectbox{...}</code>	produit une image <i>mirror</i> de la boîte
<code>\fbox{...}</code>	 boîte (version courte)
<code>\framebox[larg][pos]{...}</code>	 boîte
<code>\shadowbox{...}</code>	boîte  ombrée
<code>\doublebox{...}</code>	 boîte double
<code>\ovalbox{...}</code>	boîte  ovale
<code>\Ovalbox{...}</code>	 boîte ovale épaisse
<code>\shabox{...}</code>	boîte  ombrée centrée

`shadowbox`, `doublebox`, `ovalbox` et `Ovalbox` nécessitent le package `fancybox`, `shabox` le package `shadow`. `scalebox`, `resizebox`, `rotatebox` et `reflectbox` sont définies par le package `graphics`.

Les noms de boîtes doivent commencer par “\”. “!” utilisé comme argument de `\resizebox` permet de conserver les proportions de la boîte. Pour les commandes `\makebox`, `\savebox` et `\framebox`, *pos* peut être “c” (*center* centré), “l” (*left*, aligné à gauche), “r” (*right*, aligné à droite) ou “s” (*stretched*, étiré).

`\raisebox{décalage}[profondeur][hauteur]{...}` décale une boîte, et change ses profondeur et hauteur apparentes. `\rule[décalage]{largeur}{hauteur}` crée une règle (boîte pleine). En utilisant une largeur nulle, `\rule` permet de fixer arbitrairement les hauteur et profondeur apparentes d’une boîte. Une telle règle est appelée *strut*.

5.3 Couleurs

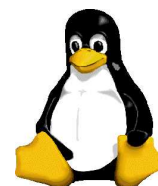
<code>\textcolor{couleur}{...}</code>	change la couleur du texte
<code>\color{couleur}</code>	change la couleur du bloc courant
<code>\colorbox{couleur}{...}</code>	crée une boîte de couleur
<code>\fcolorbox{bordure}{couleur}{...}</code>	crée une boîte de couleur encadrée
<code>\pagecolor{couleur}</code>	sélectionne la couleur de fond de la page

Ces commandes sont définies dans le package `color`.

`\definecolor{nom}{modèle}{valeur}` permet de définir une nouvelle couleur. *modèle* peut être `rgb`, `gray`, ou `cmymk`. Les couleurs sont données sous forme de coordonnées de 0 à 1, séparées par des virgules (exemple: `\definecolor{lightgreen}{rgb}{.5,1,.5}`). `black`, `white`, `red`, `green`, `blue`, `yellow`, `cyan` et `magenta` sont définies par défaut.

5.4 Inclusion d'images

`\includegraphics[x,y][x+largeur,y+hauteur]{fichier}` insère une image. Les paramètres optionnels spécifient la position des angles supérieur gauche et inférieur droit de l'image. Si le format de celle-ci n'inclut pas de données sur sa taille, les paramètres sont indispensables. La forme étoilée `\includegraphics*` coupe l'image aux dimensions données. La forme standard laissera dépasser l'image du cadre si elle est trop grande. `\includegraphics` est défini dans le package `graphics`.



6 Mathématiques

6.1 Environnements mathématiques

<code>\$...\$</code>	Expression mathématique intégrée à un paragraphe
<code>\(...\)</code>	
<code>\begin{math}... \end{math}</code>	
<code>\[...]</code>	Expression mathématique isolée
<code>\begin{displaymath}... \end{displaymath}</code>	
<code>\begin{equation}... \end{equation}</code>	Équation numérotée en mode <code>\displaymath</code>
<code>\begin{equation*}... \end{equation*}</code>	Équation non numérotée

`$$...$$` est à éviter: c'est une commande T_EX qui ne correspond pas à un environnement L^AT_EX_{2 ϵ} .

6.2 Familles de caractères

<code>\mathbb{...}</code>	alphabet <i>Blackboard</i> : ABC	<code>\mathrm{...}</code>	roman: ABCabc123
<code>\mathcal{...}</code>	alphabet calligraphié: ABC	<code>\mathbf{...}</code>	gras: ABCabc123
<code>\mathfrak{...}</code>	alphabet <i>Euler Fraktur</i> : $\mathfrak{A}\mathfrak{B}\mathfrak{C}\mathfrak{a}\mathfrak{b}\mathfrak{c}\mathfrak{1}\mathfrak{2}\mathfrak{3}$	<code>\mathsf{...}</code>	sans serif: ABCabc123
<code>\mathtt{...}</code>	typewriter: ABCabc123	<code>\mathnormal{...}</code>	normal: ABCabc123
<code>\mathit{...}</code>	italique: ABCabc123		

`\mathbb` et `\mathfrak` nécessitent le packages `amssymb` ou `amsfonts`.

6.3 Symboles et structures

6.3.1 Caractères spéciaux

<code>\hat{a}</code>	<code>\acute{a}</code>	<code>\bar{a}</code>	<code>\dot{a}</code>	<code>\breve{a}</code>
<code>\check{a}</code>	<code>\grave{a}</code>	<code>\vec{a}</code>	<code>\ddot{a}</code>	<code>\tilde{a}</code>

`\imath` et `\jmath` permettent d'afficher *i* et *j* pour une accentuation éventuelle.

Les lettres grecques sont obtenues à l'aide des commandes `\alpha` (α), `\beta` (β) ... `\Omega` (Ω), etc. Le tableau ci-dessous présente les lettres pour lesquelles des variantes existent.

<code>\epsilon</code>	<code>\varepsilon</code>	<code>\theta</code>	<code>\vartheta</code>
<code>\pi</code>	<code>\varpi</code>	<code>\rho</code>	<code>\varrho</code>
<code>\sigma</code>	<code>\varsigma</code>	<code>\phi</code>	<code>\varphi</code>

6.3.2 Symboles mathématiques

<code>=</code>	<code><</code>	<code>\circ</code>	<code>\cup</code>	<code>\times</code>
<code>\neq</code>	<code>></code>	<code>\bullet</code>	<code>\cap</code>	<code>\div</code>
<code>\equiv</code>	<code>\leq</code>	<code>*</code>	<code>\subset</code>	<code>\cdot</code>
<code>\approx</code>	<code>\geq</code>	<code>\star</code>	<code>\supset</code>	<code>\oplus</code>
<code>\sim</code>	<code>\ll</code>	<code>\parallel</code>	<code>\in</code>	<code>\otimes</code>
<code>\mid</code>	<code>\gg</code>	<code>\perp</code>	<code>\ni</code>	<code>\pm</code>

\leftarrow <code>\leftarrow</code>	\longleftarrow <code>\longleftarrow</code>	\nearrow <code>\nearrow</code>
\rightarrow <code>\rightarrow</code>	\longrightarrow <code>\longrightarrow</code>	\searrow <code>\searrow</code>
\uparrow <code>\uparrow</code>	\dashleftarrow <code>\dashleftarrow</code>	\swarrow <code>\swarrow</code>
\downarrow <code>\downarrow</code>	\dashrightarrow <code>\dashrightarrow</code>	\nwarrow <code>\nwarrow</code>
\leftrightarrow <code>\leftrightarrow</code>	\longleftrightarrow <code>\longleftrightarrow</code>	\leftharpoonup <code>\leftharpoonup</code>
\updownarrow <code>\updownarrow</code>		\leftharpoondown <code>\leftharpoondown</code>
\Leftarrow <code>\Leftarrow</code>	\Lleftarrow <code>\Lleftarrow</code>	\rightharpoonup <code>\rightharpoonup</code>
\Rightarrow <code>\Rightarrow</code>	\Rrightarrow <code>\Rrightarrow</code>	\rightharpoondown <code>\rightharpoondown</code>
\Uparrow <code>\Uparrow</code>	\nLeftarrow <code>\nLeftarrow</code>	\leftrightarrows <code>\leftrightarrows</code>
\Downarrow <code>\Downarrow</code>	\nRrightarrow <code>\nRrightarrow</code>	\rightleftarrows <code>\rightleftarrows</code>
\Leftrightarrow <code>\Leftrightarrow</code>	\Leftrightarrow <code>\Leftrightarrow</code>	\leftrightharpoons <code>\leftrightharpoons</code>
\Updownarrow <code>\Updownarrow</code>		\rightleftharpoons <code>\rightleftharpoons</code>
\mapsto <code>\mapsto</code>	\hookrightarrow <code>\hookrightarrow</code>	\curvearrowleft <code>\curvearrowleft</code>
\mapsto <code>\longmapsto</code>	\hookrightarrow <code>\hookrightarrow</code>	\curvearrowright <code>\curvearrowright</code>

\dots <code>\ldots</code>	\forall <code>\forall</code>	\triangle <code>\triangle</code>	\natural <code>\natural</code>	i <code>\imath</code>
\cdots <code>\cdots</code>	\exists <code>\exists</code>	\angle <code>\angle</code>	\sharp <code>\sharp</code>	j <code>\jmath</code>
\vdots <code>\vdots</code>	\aleph <code>\aleph</code>	∇ <code>\nabla</code>	\flat <code>\flat</code>	\Re <code>\Re</code>
\ddots <code>\ddots</code>	∞ <code>\infty</code>	\emptyset <code>\emptyset</code>	\hbar <code>\hbar</code>	\Im <code>\Im</code>
\prime <code>\prime</code>	\neg <code>\neg</code>		∂ <code>\partial</code>	\wp <code>\wp</code>

`\not` devant un symbole barre celui ci (exemple: `\not\subset` $\not\subset$). Nombre de ces symboles nécessitent les packages de l'*American Mathematical Society*.

6.3.3 Symboles de taille variable et délimiteurs

\sum <code>\sum</code>	\prod <code>\prod</code>	\int <code>\int</code>	\oint <code>\oint</code>	\bigcap <code>\bigcap</code>	\bigcup <code>\bigcup</code>
$\{$ <code>\{</code>	$($ <code>(</code>	\langle <code>\langle</code>		$ $ <code>\left </code>	$\left $ <code>\left </code>
$\}$ <code>\}</code>	$)$ <code>)</code>	\rangle <code>\rangle</code>		$\left $ <code>\left </code>	$\left $ <code>\left </code>
$[$ <code>[</code>	$ $ <code> </code>	$\left\{$ <code>\left\{</code>		$\left $ <code>\left </code>	$\left $ <code>\left </code>
$]$ <code>]</code>	$\ $ <code>\ </code>	$\left\}$ <code>\left\}</code>		$\left $ <code>\left </code>	$\left $ <code>\left </code>

Les délimiteurs peuvent encadrer une expression à l'aide de `\left` *délimiteur* et `\right` *délimiteur*. Le délimiteur “ ” sert alors de délimiteur vide. Ils peuvent aussi être précédés de `\big`, `\Big`, `\bigg`, et `\Bigg` qui modifient leur taille.

6.4 Commandes

6.4.1 Constructions

\overleftarrow{abc} <code>\overleftarrow{abc}</code>	\overline{abc} <code>\overline{abc}</code>	\widehat{abc} <code>\widehat{abc}</code>
\overrightarrow{abc} <code>\overrightarrow{abc}</code>	\underline{abc} <code>\underline{abc}</code>	\widetilde{abc} <code>\widetilde{abc}</code>
\overbrace{abc} <code>\overbrace{abc}</code>	$\frac{abc}{xyz}$ <code>\frac{abc}{xyz}</code>	$\sqrt[n]{abc}$ <code>\sqrt[n]{abc}</code>
\underbrace{abc} <code>\underbrace{abc}</code>	$\stackrel{abc}{xyz}$ <code>\stackrel{abc}{xyz}</code>	$(\text{mod } b)$ <code>\pmod{b}</code>

6.4.2 Autres commandes

<code>\smash{...}</code>		retient le contenu d'une boîte mais considère qu'elle a une taille nulle
<code>\mbox{...}</code>		insère du texte dans une expression mathématique
<code>\text{...}</code>		équivalent à <code>\mbox</code> , mais gère plus intelligemment les tailles

Les commandes suivantes affichent les fonctions correspondantes en lettres droites :

<code>\arccos</code>	<code>\arg</code>	<code>\arcsin</code>	<code>\arctan</code>	<code>\cos</code>	<code>\cosh</code>	<code>\cot</code>	<code>\coth</code>	<code>\csc</code>	<code>\deg</code>	<code>\dim</code>
<code>\det</code>	<code>\exp</code>	<code>\gcd</code>	<code>\hom</code>	<code>\inf</code>	<code>\ker</code>	<code>\lg</code>	<code>\lim</code>	<code>\liminf</code>	<code>\limsup</code>	<code>\ln</code>
<code>\log</code>	<code>\max</code>	<code>\min</code>	<code>\Pr</code>	<code>\sec</code>	<code>\sin</code>	<code>\sinh</code>	<code>\sup</code>	<code>\tan</code>	<code>\tanh</code>	

6.4.3 Espacements

<i>commande</i>	<i>abréviation</i>	<i>exemple</i>	<i>commande</i>	<i>abréviation</i>	<i>exemple</i>
<code>\negthinspace</code>	<code>\!</code>	<i>ab</i>	<code>\thickspace</code>	<code>\;</code>	<i>a b</i>
		<i>ab</i>	<code>_</code>	<code>_</code>	<i>a b</i>
<code>\thinspace</code>	<code>\,</code>	<i>a b</i>	<code>\quad</code>		<i>a b</i>
<code>\medspace</code>	<code>\:</code>	<i>a b</i>	<code>\qquad</code>		<i>a b</i>

6.4.4 Environnements en mode mathématique

<code>align</code>	tableau de deux colonnes justifiées à gauche et à droite, numérotées
<code>array</code>	semblable à <code>tabular</code> , mais en mode mathématique
<code>eqnarray</code>	<code>array</code> de 3 colonnes numérotées, sauf si la commande <code>\nonumber</code> est présente
<code>multiline</code>	environnement autorisant les retours à la ligne, numéroté

Les environnements numérotés ont souvent une version étoilée sans numérotation.

7 Définition de commandes et longueurs

7.1 Commandes et environnements

`\newcommand{nom}[nb. param][défaut]{définition}`: définit une nouvelle commande. Les noms de commandes doivent commencer par “\”. *nb. param* est le nombre de paramètres attendus. Si *défaut* est spécifié, le premier paramètre (#1) est optionnel et a pour valeur *défaut*.

`\renewcommand`: redéfinit ou définit une commande;

`\providecommand`: définit une commande sauf si elle existe déjà;

`\newenvironment{nom}[nb. param][défaut]{début}{fin}`: définit un nouvel environnement. Les blocs *début* et *fin* remplaceront `\begin{nom}` et `\end{nom}`;

Dans une définition, #n est remplacé par le n-ème paramètre.

7.2 Longueurs

7.2.1 Unités et longueurs élastiques

Les longueurs numériques sont toujours suivies d’une unité (exemple: `\hspace{3pt}`).

<code>sp</code>	<i>scaled point</i> (65536 sp = 1 pt) — la plus petite unité de T _E X	— (×1000000)
<code>pt</code>	point (1 pt = $\frac{1}{72.27}$ in = 0.351 mm)	————— (×100)
<code>bp</code>	<i>big point</i> (1 pt = $\frac{1}{72}$ in) — point PostScript	————— (×100)
<code>dd</code>	point Didôt ($\frac{1}{72}$ de pouce français, soit 0.376 mm)	— (×10)
<code>mm</code>	millimètre (1 mm = 2.845 pt)	————— (×10)
<code>pc</code>	pica (1 pc = 12 pt = 4.218 mm)	————— (×10)
<code>cc</code>	cicéro (1 cc = 12 dd = 4.531 mm)	————— (×10)
<code>cm</code>	centimètre (1 cm = 10 mm = 2.371 pc)	—————
<code>in</code>	<i>inch</i> (1 in = 25.4 mm = 72.27 pt = 6.022 pc)	—————
<code>ex</code>	hauteur d’un “x” minuscule dans la fonte courante	—
<code>em</code>	largeur d’un “M” majuscule dans la fonte courante	—
<code>mu</code>	<i>math unit</i> (18 mu = 1 em)	— (×10)

`\fill`: longueur élastique, ayant la faculté de s’étendre de 0 à n’importe quelle longueur positive;

`\stretch{num}`: longueur élastique ayant *num* fois l’élasticité de `\fill`. *num* peut être négatif.

7.2.2 Définition

<code>\newlength{nom}</code>	déclare <i>nom</i> comme une longueur
<code>\setlength{nom}{longueur}</code>	fixe la valeur de <i>nom</i>
<code>\addtolength{nom}{longueur}</code>	ajoute <i>longueur</i> à <i>nom</i>
<code>\settowidth{nom}{texte}</code>	fixe la valeur de <i>nom</i> à la largeur de <i>texte</i>
<code>\settoheight{nom}{texte}</code>	fixe la valeur de <i>nom</i> à la hauteur de <i>texte</i>
<code>\settodepth{nom}{texte}</code>	fixe la valeur de <i>nom</i> à la profondeur de <i>texte</i>

Comme les noms de commandes, les noms de longueurs doivent commencer par “\”.