

Démarrer en L^AT_EX avec MikTeX et WinEdt

Marie-Claude Mondini

26 juin 2006

Sommaire

1	Introduction	1
1.1	Un brin d'histoire	1
1.2	Pourquoi L ^A T _E X ?	1
1.3	Comment ça "marche" ?	1
2	Les outils	2
2.1	Les logiciels	2
2.2	Installation	2
2.2.1	Télécharger	2
2.2.2	Installer	2
2.3	Quelques réglages	3
3	On se lance !	3
3.1	Tester l'installation	3
3.2	Plus fort !	4
4	Structure de base d'un document	4
4.1	Préambule	4
4.2	Corps du document	5
5	Commandes de base	6
5.1	Caractères particuliers	6
5.2	Commandes de fonte	6

5.2.1	Taille des caractères	6
5.2.2	Types de fonte	7
5.3	Les espaces	8
5.3.1	Les blancs	8
5.3.2	Sauts de ligne, de page	9
5.3.3	Les tirets, points de suspension	9
5.3.4	Indentation	9
5.4	Commandes d'ornements	10
5.4.1	Souligner	10
5.4.2	Encadrer	10
	(a) boîtes de ligne	10
	(b) boîtes de paragraphe	10
5.4.3	Colorer	11
5.5	Les titres	11
5.5.1	Niveaux	11
5.5.2	Forme étoilée	12
5.5.3	Page de garde	12
5.6	Les environnements	12
5.6.1	Dispositions du texte	12
5.6.2	L'environnement <i>verbatim</i>	13
5.7	Les listes	14
6	Commandes plus élaborées	15
6.1	Les tableaux	15
6.1.1	Une aide pour commencer	15
6.1.2	L'environnement <i>tabular</i>	15
	(a) Syntaxe "basique"	16
	(b) Lignes horizontales et verticales	17
	(c) Regroupement de cellules	17

6.2	Les références	18
6.3	En-tête et pied de page	18
6.3.1	Structure "de base"	18
6.3.2	Avec fancyhdr	18
6.4	Notes de bas de page	19
6.5	Table des matières	19
6.6	Index	20
7	Ecrire des mathématiques	20
7.1	Principes	20
7.2	Les modes de présentation	20
7.3	L'aspect du texte	21
7.4	Structures de base	22
7.4.1	Opérations élémentaires	22
7.4.2	Exposants et indices	22
7.4.3	Fractions	22
7.4.4	Délimiteurs	22
7.5	Vecteurs, matrices et déterminants	23
7.5.1	Array	23
7.5.2	Matrix	24
7.6	Equations	25
7.6.1	Numérotées	25
7.6.2	Alignées	25
7.7	Symboles en vrac	25
7.7.1	Cases	25
7.7.2	Symboles empilés	25
8	Les insertions	26
8.1	Images	26

8.2	Dessin	26
8.3	Fichiers	27
9	Les erreurs	27
10	Personnaliser L^AT_EX	27
10.1	Compteurs	27
10.2	Nouvelle commande	27
10.2.1	la macro "newtheorem"	27
10.2.2	macro très simple	28
10.2.3	macros .. moins simples	29
(a)	newcommand	29
(b)	newenvironment	29
11	Où trouver de l'aide ?	30
11.1	Sur internet	30
11.2	Dans les livres	30

Liste des tableaux

1	Obtention des caractères réservés	6
2	Caractères spéciaux	6
3	Macros de taille de caractère	7
4	Macros de type de caractère	7
5	Délimiteurs des modes mathématiques	21
6	Styles en mode mathématique	21
7	Délimiteurs	23
8	Environnement <i>matrix</i>	24
9	Empilement de symboles	26

L'index en dernière page est à l'état embryonnaire et n'est fait qu'à titre d'illustration du paragraphe index de ce mini-manuel

1 Introduction

1.1 Un brin d'histoire

T_EX est un langage de programmation, logiciel libre écrit par l'Américain Donald E.Knuth dans les années 80.

L^AT_EX en est un sur-ensemble, écrit par Leslie Lamport dans les années 90, il simplifie l'usage de T_EX en apportant, par exemple, la notion de module ("package") qui rend le système extensible.

1.2 Pourquoi L^AT_EX ?

Ce n'est pas un logiciel "Wysiwyg" (What You see Is What You Get) ce qui peut sembler déroutant et difficile à utiliser au début, et nécessite un temps d'apprentissage.

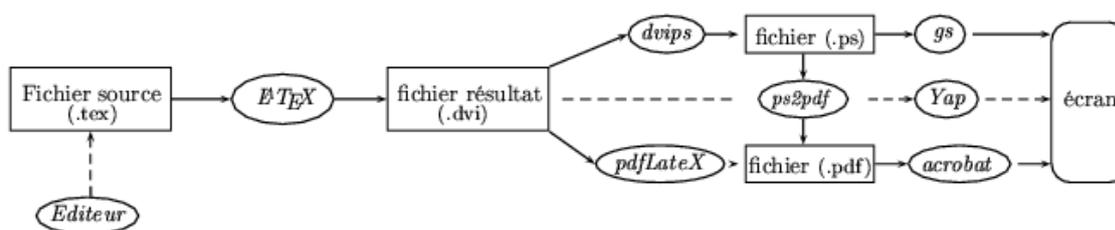
Mais on gagne

- ✓ en qualité typographique (type "imprimerie"),
- ✓ en portabilité (le logiciel s'implante sur toute plateforme, le fichier source est utilisable sous Mac / Linux / Unix / Windows)
- ✓ en "évolutivité" : le point d'échelle est de $5,36 \times 10^{-6}$ mm, les imprimantes ont encore des progrès à faire !!
- ✓ en taille (le fichier source peut être 1 000 fois plus léger que le fichier Word correspondant)
- ✓ en pérennité les documents écrits il y a 20 ans sont toujours utilisables, pas de problème de compatibilité de version
- ✓ en personnalisation L^AT_EX est programmable donc hautement personnalisable !

Particulièrement adapté aux domaines scientifiques, et plus encore aux mathématiques, il permet (avec des fichiers d'extension) de créer des partitions musicales, des graphiques complexes (Chimie, électronique, jeux,...)

A noter, une bibliographie existante de qualité et la multitude de Faq¹ et aides sur Internet² : on trouve toujours une réponse aux questions qu'on se pose !

1.3 Comment ça "marche" ?



Quelques explications :

- on écrit le texte du fichier appelé « source »(exemple : **toto.tex**) avec un **éditeur de texte** [cf 2.1.2]) (on peut même le faire avec un bloc-note "basique" mais cela nécessite alors une bonne connaissance de la syntaxe et des commandes de base)
- on le compile avec **L^AT_EX**[cf 2.1.3]) qui produit plusieurs fichiers :
 - (i) un fichier **toto.dvi** (*device independant* = indépendant de tout matériel)
 - (ii) un fichier **toto.log** : c'est le journal de la compilation, avec la liste des erreurs éventuelles
 - (iii) un fichier **toto.aux** : informations relatives aux compteurs ..., réutilisées lors d'une prochaine compilation
 - (iv) éventuellement d'autres fichiers (par exemple **toto.toc** : *table of contents*, **toto.lof** : *list of figures*, **toto.lot** : *list of tables*,...)

¹ Foires Aux Questions

² Tester "tutorial Latex" sous Google par exemple

- on visualise à l'écran avec un "**viewer**"[cf 2.1.4)] et on peut imprimer
- éventuellement on transforme en fichier **toto.ps**[cf 2.1.5)] ou **toto.pdf**[cf 2.1.5)] avant visualisation et/ou impression

2 Les outils

2.1 Les logiciels

- * **la base L^AT_EX** : nous utiliserons la distribution MiKTeX pour Windows
- * **l'éditeur** : nous utiliserons WinEdt(shareware à 30\$US la licence); il y en a d'autres : TeXnicCenter (gratuit semble voisin de WinEdt), TeXmaker (en Français ! mais semble moins complet que les 2 précédents)
- * **les visualiseurs** : nous utiliserons Yap pour lire les fichiers **.dvi**, Ghostview pour lire les fichiers **.ps** et Acrobat Reader pour les fichiers **.pdf**

2.2 Installation

On pourra se laisser guider par la page Internet : <http://mcmperso.ifrance.com/Latex/Installer.htm>

2.2.1 Télécharger

- * **MikTeX** : <http://www.miktex.org/setup.html>
cliquer sur "**download the setup wizard**" puis choisir le site de téléchargement (ovh en France par exemple); télécharger; lancer le programme obtenu ("**setup-2.4.1705.exe**") (choisir "download only"), choisir de télécharger la version "large", puis le site de téléchargement (par exemple "inria") .. et partir boire un bon café, ça prend du temps !!
- * **ghostscript puis ghostview** : <http://www.cs.wisc.edu/ghost/>
- * **Adobe reader** : déjà installé, probablement, sinon <http://www.adobe.fr/products/acrobat/readstep2.html>
- * **WinEdt** : <http://www.winedt.com/> cliquer sur "**Download WinEdt 5.4 : [Build : 20050701] from CTAN**", puis télécharger la dernière version. Télécharger ensuite le dictionnaire français sur la page de téléchargements, cliquer sur "**Dictionaries (word lists) for WinEdt**" puis à la ligne "French", sur "**DIC**" (fichier fr.zip).

2.2.2 Installer

Installer les logiciels **dans l'ordre** de téléchargement ci-dessus

- * **MikTeX** : lancer à nouveau "**setup-2.4.1705.exe**" et cette fois, choisir "Install", suivre les indications, installer le programme comme suggéré à la racine du disque dur
- * **ghostscript puis ghostview et enfin Adobe reader** :
 - a) lancer l'application "**gs853w32.exe**" et installer GhostScript ;
(par exemple sur **C:\Program Files\GhostScript**)
 - b) lancer l'application **gsv48w32.exe** et installer GSview ; (par exemple sur **C:\Program Files\Ghostgum**)
 - c) Installer si nécessaire (cad si ce n'est pas déjà fait) Adobe Reader
- * **WinEdt** :
 - a) Le programme : lancer "**winedt54.exe**" et l'installer (par exemple sur **C:\Program Files\WinEdt Team**) en suivant les instructions
 - b) Le dictionnaire : créer le dossier *fr* dans **WinEdt\Dict** (par exemple sur **C:\Program Files\WinEdt Team\WinEdt\Dict\fr**) et y mettre les fichiers *fr.dic* et *fr.txt* (qui sont dans le fichier compressé *fr.zip*); lancer WinEdt; dans l'onglet Options, cliquer sur **Dictionary Manager**, faire un clic droit puis cliquer sur **Insert**; sous **Dictionaries**, taper : **fr**; sous **Definition** tapez : **% B \ Dict \ fr \ fr.dic** cocher **Enabled**, puis cliquer sur **Load**, puis sur **OK**; (la case **Status** se remplit)

2.3 Quelques réglages

Quelques conseils personnels : dans WinEdt

* cliquer sur le symbole " Σ " pour faire apparaître un grand nombre de raccourcis mathématiques (voir aussi [7 page 20](#))

* cliquer sur : "Options" → "Settings" ; Choisir l'onglet "Active Strings" puis colonne de gauche " $\backslash begin{\?}$ " et cocher en colonne de droite "on type"

3 On se lance !

3.1 Tester l'installation

Pour tester les liens : créer un petit fichier sous WinEdt, puis le compiler :

Enregistrer, compiler, transformer, visualiser les fichiers (.dvi, .ps puis .pdf), en utilisant les icônes ci-contre, de gauche à droite :



ligne 1 : Textify (tex vers dvi+visualisation) ; Compilation LateX ; Compilation TeX (inutile ici) ; pdfTextify (tex vers pdf+visualisation) ; pdfflatex (créé .pdf à partir de .tex sous latex) ; idem sous tex (inutile ici) ; Bibtex (bibliographies) ; make Index (pour faire un index)

ligne 2 : compile le bloc sélectionné + visualise en dvi (utile dans un fichier long) ; dvi search (affiche le fichier .dvi, **retour vers le fichier .tex par click droit sur "Edit Text source"**) ; visualiser .dvi (idem sans retour sur .tex) ; visualiser .ps ; visualiser .pdf ; créer .pdf à partir de .dvi ; créer .ps à partir de .dvi ; créer .pdf à partir de .ps + visualisation

mais d'abord, le fichier :

```
\documentclass[a4paper]{article}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[français]{babel}
\usepackage{amsmath,amssymb}
\usepackage{amsfonts}

\begin{document}
Premier essai \newline $\dfrac{\pi}{2}$ $
\end{document}
```

voir [4 page 4](#) pour explications

1) **obligatoirement en premier**

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

où *options*, comme son nom l'indique, est facultatif et peut contenir : la taille des caractères (10 pt par défaut, on peut choisir 11 ou 12 pt), la taille du papier (*a4paper* pour nous, ici), le nombre de colonnes (*onecolumn* par défaut)... *classe* est obligatoire et peut prendre pour valeur : article, report, book, letter, slides (*article* le plus souvent)

2) **la déclaration des modules** utilisés sous la forme

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

où *module* est le nom du module utilisé ; je vous conseille les suivants (systématiquement)

```
\usepackage[latin1]{inputenc}% Document avec accents format windows,
\usepackage[T1]{fontenc} % pour taper directement les lettres accentuées
\usepackage[francais]{babel} % en français : date, table des matières..
\usepackage{amsmath} % les fontes mathématiques : la + importante
\usepackage{amsfonts} % + de symboles et \mathbb
\usepackage{amssymb} % + de symboles
\usepackage{amsthm} % + de possibilités pour mise en forme des thms
\usepackage{array} % pour les tableaux (mode texte ou maths)
\usepackage{fancyhdr} % gère en tête et pied de page (3 parties)
```

et éventuellement ceux-ci, si besoin :

```
\usepackage[francais]{layout} % affiche la maquette
\usepackage{pstricks} % pour les graphiques
\usepackage{graphicx} % pour les graphiques
\usepackage{eurosym} % pour le symbole euro, donné par \euro
\usepackage{aequill} % beau rendu écran en pdf, permet
% d'utiliser les guillemets français
\usepackage{hhline} % gère finement les lignes dans un tableau
\usepackage{multirow} % pour les tableaux (multicolonnage)
\usepackage{color} % gère les couleurs (!)
```

3) **mise en page** : voici ce que j'ai déclaré pour ce document :

```
\voffset 0mm \hoffset 0mm \topmargin -20mm \headsep 5mm \headheight
24pt \oddsidemargin -10mm \textwidth 180mm \textheight 260mm
\footskip 8mm \parindent 3mm
```

ces valeurs sont à régler avec la visualisation de la page, obtenue par le module `layout` (cf ci-dessus) et l'appel `\layout`

au début du corps de document

4) **en tête et pied de page** : voir l'utilisation du module `fancyhdr` et [6.3 page 18](#)

5) les éventuelles macros utilisées dans le document

4.2 Corps du document

Le texte et les déclarations "locales",

```
\begin{document}
ici le texte à imprimer
\end{document}
```

sera développé plus loin

5 Commandes de base

5.1 Caractères particuliers

Les 10 caractères `\ % ~ { } $ _ ^ #` et `&` ont des rôles précis et ne peuvent pas être employés pour obtenir les caractères correspondants au niveau de la sortie. Le tableau 1 montre les dix macros permettant de retrouver ces caractères. On voit que

caract.	macro	caract.	macro	caract.	macro	caract.	macro	caract.	macro
<code>\</code>	<code>\(backslash)</code>	<code>%</code>	<code>\%</code>	<code>~</code>	<code>\~{}</code>	<code>{</code>	<code>\{</code>	<code>}</code>	<code>\}</code>
<code>\$</code>	<code>\\$</code>	<code>_</code>	<code>_</code>	<code>^</code>	<code>\^{}</code>	<code>#</code>	<code>\#</code>	<code>&</code>	<code>\&</code>

TAB. 1 – Obtention des caractères réservés

pour sept d'entre eux, la macro a le nom du caractère lui-même. Les exceptions concernent la barre oblique inverse (backslash en anglais), l'accent circonflexe et le niña.

Remarquons que la macro "barre oblique inverse" doit être entourée des signes `\(` et `\)` qui indiquent que l'intérieur est composé en mode mathématique (Cf. section 7).

Ainsi, on a :

caractère	effet
<code>%</code>	introduit un commentaire qui ne sera pas imprimé
<code>\</code>	introduit une commande qui sera interprétée par L ^A T _E X
<code>\$</code>	encadre du texte mathématique (à mettre au début et à la fin)

Voici quelques caractères non disponibles sur le clavier et les macros pour les obtenir :

caract.	macro	caract.	macro	caract.	macro	caract.	macro	caract.	macro
œ	<code>\oe</code>	Œ	<code>\OE</code>	æ	<code>\ae</code>	Æ	<code>\AE</code>	å	<code>\aa</code>
Å	<code>\AA</code>	ø	<code>\o</code>	Ø	<code>\O</code>	†	<code>\ </code>	Ł	<code>\L</code>
§	<code>\S</code>	¶	<code>\P</code>	©	<code>\copyright</code>	£	<code>\pounds</code>		

TAB. 2 – Caractères spéciaux

Une dernière remarque pour les guillemets « à la française » : si on a chargé le module *aeguill*, on les obtient par `\og texte\fg` (`og=` ouvre guillemets, `fg=` ..devinez !)

5.2 Commandes de fonte

Une fonte est l'ensemble des caractères présentant une forme commune (même taille, même type de dessin) ; le mot de "police" est un néologisme informatique qu'il vaut mieux éviter. On peut classer les opérations sur les fontes en deux grandes catégories : les macros qui vont agir sur la taille du texte et les macros qui vont agir sur le type des caractères (italique, romain, sans empattement, etc.).

5.2.1 Taille des caractères

Si on ne dit rien du tout, les caractères sont pris dans la fonte Computer Modern (inventée par Knuth et ressemblant un peu à la fonte Times) en taille 10. C'est ce que vous avez sous les yeux en ce moment.

Pour changer de taille de caractère, L^AT_EX propose 10 macros présentées dans le tableau 3 page 7 :

Macro	Résultat
<code>\tiny</code>	minuscule
<code>\scriptsize</code>	taille scripte
<code>\footnotesize</code>	note de bas de page
<code>\small</code>	petit
<code>\normalsize</code>	taille normale
<code>\large</code>	grand
<code>\Large</code>	Grand
<code>\LARGE</code>	GRAND
<code>\huge</code>	énorme
<code>\Huge</code>	Énorme

TAB. 3 – Macros de taille de caractère

Attention à leur utilisation. Si on met une de ces macros dans le fichier source, tout le texte qui va suivre sera de la taille spécifiée. Une façon naturelle de revenir à la taille précédente serait d'écrire `\normalsize` mais ce n'est pas une si bonne idée car la taille précédente n'était pas forcément la taille normale et, en plus, c'est long à écrire ! Une façon beaucoup plus propre et rapide est d'utiliser un groupe.

source	résultat
<code>{\Large Jamais !} Jamais</code>	Jamais ! Jamais
<code>{\LARGE Jamais}</code>	Jamais
<code>{\tiny D'accord.} D'accord</code>	D'accord. D'accord

5.2.2 Types de fonte

Il existe 9 paires de macros permettant de spécifier le type de caractère voulu. Ces macros se répartissent en trois groupes : les macros indiquant la forme (droit (up), italique (italic), penché (slanted) et petite capitale (small cap)), les macros indiquant la graisse (moyen (medium) ou gras (boldface)) et les macros indiquant la famille (romain (roman), sans empattement (sans serif) et non proportionnel dite type machine à écrire (typewriter)). Le tableau 4 présente quelques unes de ces 18 macros. Les

Macro I	Macro II	Résultat
Macros pour la forme		
<code>\textup</code>	<code>\upshape</code>	Un exemple pour voir
<code>\textit</code>	<code>\itshape</code>	<i>Un exemple pour voir</i>
<code>\textsc</code>	<code>\scshape</code>	UN EXEMPLE POUR VOIR
Macro pour la graisse		
<code>\textbf</code>	<code>\bfseries</code>	Un exemple pour voir
Macros pour la famille		
<code>\textrm</code>	<code>\rmfamily</code>	Un exemple pour voir
<code>\textsf</code>	<code>\sffamily</code>	Un exemple pour voir

TAB. 4 – Macros de type de caractère

premières lignes de chaque série de macros (`\textup`, `\textmd`, `\textrm` et les macros de type II correspondantes) sont les types

par défaut des caractères.

La différence entre les macros de type I et celles de type II porte sur la façon d'indiquer ce qui va être affecté par le changement. Les macros de type II se comportent comme les macros de changement de taille vues précédemment. Les macros de type I n'affectent que ce qui suit immédiatement.

5.3 Les espaces

5.3.1 Les blancs

Avant de voir les principaux espaces, il faut noter que L^AT_EX a un comportement qui peut sembler déroutant au début, en ce qui concerne les espaces :

- s'il y a plusieurs blancs successifs, un seul est pris en compte
- les blancs en début de ligne sont ignorés
- les retours à la ligne sont traités comme des blancs simples, voici un exemple :

<u>source</u>	<u>résultat</u>
<i>voici un petit espace, un plus grand, un passage à la ligne, un retrait, tout cela inutile !!</i>	voici un petit espace, un plus grand, un passage à la ligne, un retrait, tout cela inutile !!

Ceci présente des avantages : on peut taper son source sans se soucier du résultat à l'écran, en le rendant le plus lisible possible

Comment, alors, ajouter des espaces à notre grè ? On peut "protéger" les blancs : pour cela, comparer les expressions suivantes

<u>source</u>	<u>résultat</u>
<code>\LaTeX supprime les blancs</code>	L ^A T _E Xsupprime les blancs
<code>\LaTeX\ supprime les blancs</code>	L ^A T _E X supprime les blancs
<code>\LaTeX\ supprime \ \ \ les blancs</code>	L ^A T _E X supprime les blancs

Mais si l'on veut un espace horizontal plus important, on peut traiter tous les cas avec les macros :

`\hspace{largeur}` à l'intérieur d'une ligne ou `\hspace*{largeur}` cette dernière reste efficace même en début ou fin de ligne

De même, pour un espace vertical, on utilisera :

`\vspace{largeur}` ou `\vspace*{largeur}` cette dernière reste efficace même en début de page

L'inconvénient est qu'il faut savoir quelle dimension indiquer. On peut aussi laisser L^AT_EX répartir le texte. Voici quelques exemples avec `\hfill`.

<u>source</u>	<u>résultat</u>
<code>Un\hfill texte.</code>	Un texte.
<code>Un\hfill autre\hfill texte.</code>	Un autre texte.
<code>Un\hfill autre\hfill\hfill texte.</code>	Un autre texte.

On peut voir que le texte s'est étiré sur toute la largeur de la ligne. En prenant une règle, on pourra vérifier que sur la deuxième ligne, les mots "Un", "autre" et "texte" sont séparés par exactement le même espacement. Plus fort, sur la troisième ligne, il y a deux fois plus d'espace entre "autre" et "texte" qu'entre "Un" et "autre". Autre exemple :

```
{}\hfill \textbf{Super}\hfill\hfill{}
```

Super

Ici la présence des groupes vides est nécessaire car ces commandes en début de ligne ou en fin de ligne n'ont pas d'action (comme `\hspace` et `\vspace`). Il faut donc placer quelque chose qui n'a aucune répercussion au niveau du document final : le groupe vide est un candidat idéal !

Bien sûr, cette façon de répartir régulièrement du texte au niveau d'une ligne a son pendant vertical avec l'emploi de `\vfill` qui permet de répartir régulièrement des paragraphes dans une page.

5.3.2 Sauts de ligne, de page

Saut de ligne :

- laisser une ligne vide pour forcer le passage à la ligne : ceci indique à L^AT_EX la fin du paragraphe, crée un retrait par indentation (cf 5.3.4) et nouveau paragraphe (rappel : 1 ou plusieurs lignes vides créent le même effet)
- on peut utiliser `\newline` qui force la passage à la ligne à l'intérieur du paragraphe. L'espace vertical est plus petit (si l'espace entre paragraphes est supérieur à l'interligne) et il n'y a pas de retrait d'indentation)

Sauts de page : j'utilise essentiellement `\newpage` qui crée un saut de page inconditionnel, là où on le place ; on peut aussi utiliser `\pagebreak[deg]` qui suggère un saut de page avec un degré d'exigence (*deg*) de 0 à 4 `\nopagebreak[deg]` est l'instruction contraire

5.3.3 Les tirets, points de suspension

	source	résultat		source	résultat
Les tirets :	-	-	Les points de suspension :	<code>\dots</code>	...
	--	—		<code>\ldots</code>	...
	---	—			

5.3.4 Indentation

Les paragraphes commencent par défaut avec un retrait d'alinéa au niveau de leur première ligne. On peut inhiber cette fonction en faisant précéder le paragraphe de la macro `\noindent` comme dans le paragraphe qui suit immédiatement. On remarquera aussi qu'à partir de ce paragraphe, l'indentation n'est plus la même que d'habitude ainsi que le saut vertical entre les paragraphes. Le comportement normal sera retrouvé à la prochaine section.

Ce retrait peut être défini par l'utilisateur. Normalement, ce type de déclaration se fait au niveau du préambule puisqu'il est rare de changer d'indentation au milieu d'un document, avec, par exemple dans ce petit manuel : `\parskip 5mm` et `\parindent 3mm` qui désignent respectivement l'espace (vertical) entre 2 paragraphes et le retrait (horizontal) de paragraphe. Pour les modifier en cours de texte, il suffit d'employer la syntaxe :

```
\setlength{dim}{longueur}
```

où *dim* est l'une des dimensions de paragraphe (parmi lesquelles nous avons vu `\parskip` et `\parindent` Pour la longueur, T_EX étant très scrupuleux, il permet d'utiliser toutes les unités typographiques usuelles anglo-saxonnes et françaises ainsi que les unités courantes du système international plus une unité à lui qui est le point d'échelle.

5.4 Commandes d'ornements

5.4.1 Souligner

On souligne avec la commande `\underline` On trace une ligne avec `\rule[décalage]{largeur}{hauteur}` Ce qui donne :

source	résultat
<code>\underline{Un texte en évidence}</code>	<u>Un texte en évidence</u>
<code>\rule{4cm}{.3pt}</code>	_____
<code>\rule{.3pt}{5mm}</code>	
<code>\rule{1ex}{1ex}</code>	■

5.4.2 Encadrer

(a) boîtes de ligne

Elles sont destinées à recevoir des textes tenant sur une ligne ; le texte est alors un bloc qui ne sera pas coupé et débordera éventuellement hors des marges, voir de la page ! La commande la plus simple est : `\fbox{Une boîte}` soit ; Une boîte

(b) boîtes de paragraphe

On utilise `\begin{minipage}...\end{minipage}` (je préfère à `\parbox` car permet l'utilisation d'environnements cf 5.6 page 12) pour une boîte non encadrée

Syntaxe :

`\begin{minipage}[position][hauteur][position interne]{largeur} texte \end{minipage}` Comme d'habitude, les arguments entre crochets sont optionnels, ceux entre accolades obligatoires :

position interne :

- b** texte en bas de la boîte ("bottom")
- t** texte en haut de la boîte ("top")
- c** texte centré dans la boîte (en hauteur)
- s** texte étiré dans la boîte (remplit tout l'espace)

Pour encadrer cette boîte, on met le tout dans une `\fbox{}`

Exemple :

source	résultat
<pre>\fbox{\begin{minipage}[b]{5cm} Voici un essai de boîte\ avec des \emph{changements de paramètres} \end{minipage}}</pre>	<div style="border: 1px solid black; padding: 5px;"> Voici un essai de boîte avec des <i>changements de paramètres</i> </div>

Remarque : il y a beaucoup d'autres possibilités, en jouant sur la forme, le type d'encadrement, la disposition du texte ... Ce sera pour une autre fois !

5.4.3 Colorer

On peut changer la couleur du texte, d'une boîte, du fond de page.

Pour cela, on doit déclarer en préambule `\usepackage{color}` On peut soit utiliser une couleur prédéfinie (black, blue, cyan, green, magenta, red, white, yellow), soit définir sa propre couleur dans le système "rgb" (pour red-green-blue où une couleur est repérée par un triplet de 3 nombres entre 0 et 1) ou "cmyk" (pour cyan-magenta-yellow-black où une couleur est repérée par un quadruplet de 4 nombres entre 0 et 1)

Pour le texte :

<u>syntaxe</u>	<u>effet</u>	<u>source</u>	<u>résultat</u>
<code>\textcolor{couleur}{Texte}</code>	change la couleur de <i>Texte</i>	<code>du \textcolor{red}{rouge} ici</code>	du rouge ici
<code>\color{couleur}</code>	change la couleur à partir de la commande	<code>\color{blue} tout bleu !</code>	tout bleu !

Pour les boîtes :

<u>syntaxe</u>	<u>effet</u>	<u>source</u>	<u>résultat</u>
<code>\colorbox{couleur}{Texte}</code>	imprime Texte sur fond de couleur indiquée	<code>\colorbox{yellow}{jaune}</code>	jaune
<code>\fcolorbox{Cadre}{Fond}{Texte}</code>	imprime <i>Texte</i> sur fond de couleur <i>Fond</i> , encadré en couleur <i>Cadre</i>	<code>\fcolorbox{red}{green}{vert}</code>	vert

5.5 Les titres

5.5.1 Niveaux

L^AT_EX connaît 7 niveaux de titres. Dans l'ordre d'importance on trouve `\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph` et `\subparagraph`. La syntaxe est très simple. Ainsi les trois titres précédents ont été obtenus avec les lignes :

```
\section{ Commandes de base}
\subsection{Les titres}
\subsubsection{Niveaux}
```

Une restriction : (c'est la seule) dans la classe *article* la macro `\chapter` n'existe pas. D'autre part, la mise en page n'est pas la même en fonction de la classe de document. Essentiellement, les espaces verticaux seront plus ou moins importants. Par exemple, un `\chapter` prendra une page complète et sera composé sur une page impaire dans le style *book* alors qu'avec la classe *report* ce ne sera pas le cas. On peut modifier ces réglages (voir 10 page 27)

Si le document a été correctement structuré avec ces 7 macros, il devient simplissime de construire la table des matières (cf. 6.5 page 19)

5.5.2 Forme étoilée

Chaque commande de hiérarchisation a une forme étoilée qui ne numérote pas le titre, ne modifie pas la numérotation des autres sections et ne produit pas d'entrée dans la table des matières sauf indication contraire (cf 6.5 page 19). On utilise par exemple : `\section*{titre}`

5.5.3 Page de garde

L^AT_EX permet d'imprimer une page de garde à partir d'informations contenues dans le préambule : `\title{titre}` indique le titre du document. On peut utiliser `\` pour passer à la ligne. `\author{nom}` indique le nom de l'auteur. S'il y en a plusieurs, leurs noms doivent être séparés par `\and`. On peut utiliser `\` pour passer à la ligne. `\date{date}` indique la date du document. Les commandes `\title` et `\author` sont obligatoires. Si la commande `\date` n'est pas utilisée, la date du jour de la compilation est mise automatiquement. Si l'on ne veut pas de la date, il suffit d'écrire : `\date{}` Pour imprimer la page de garde, il faut appeler la commande `\maketitle` juste après le `\begin{document}`. Par exemple, pour ce document :

```
\documentclass[a4paper]{article}
\usepackage[latin1]{inputenc}
...
\author{Marie-Claude Mondini}
\title{\textbf{Démarrer en \LaTeX} avec MikTeX et WinEdt}
\begin{document}
\maketitle
```

5.6 Les environnements

De manière générale, on rentre dans un environnement *env* par `\begin{env}` et on en sort par `\end{env}`; ce qui est entre `{ }` est obligatoire et entre `[]` est optionnel.

Pour les tableaux, voir 6.1 page 15 (mode texte), ou 7.5.1 page 23 (mode mathématiques)

5.6.1 Dispositions du texte

Un paragraphe est, par défaut, indenté (cad avec retrait à la première ligne, cf 5.3.4) et justifié (les bords gauche et droit étant alignés avec les marges gauche et droite). Il est quelquefois souhaitable de ne pas obtenir cette présentation et L^AT_EX offre un grand choix de commandes permettant d'obtenir des effets très spéciaux. Il est hors de question de faire le tour de la question et je citerai essentiellement les macros que j'utilise le plus.

Il existe quatre présentations classiques de paragraphe : justifié (ce qui est la présentation par défaut), au fer à gauche³ (seul le bord gauche du paragraphe est aligné avec la marge), au fer à droite (le bord droit du paragraphe aligné avec la marge de droite) et centré. Pour obtenir les trois dernières présentations, il existe à chaque fois une macro et un environnement. Voyons les environnements en premier :

³La dénomination exacte serait plutôt « en drapeau au fer à gauche » ce qui est plus explicite.

Voici un paragraphe normal (avec une indentation de première ligne nulle pour gagner de la place). Cet exemple suffit à voir l'alignement des bords gauche et droit.

Un paragraphe au fer à gauche. La syntaxe anglaise est évidemment cohérente en remplaçant droite par gauche.

Voici un paragraphe au fer à droite où l'alignement ne se fait plus qu'au niveau du bord droit. La syntaxe anglaise peut être traduite par *aligné à droite*.

Paragraphe centré.
Seules les lignes relativement courtes ont un intérêt sinon, on ne voit pas très bien le caractère centré de la chose !

Les macros permettant d'obtenir ces présentations sont respectivement `\raggedleft`, `\raggedright` et `\centering`. On notera l'inversion de *left* et *right*. Cela s'explique par la traduction de *flush* qui veut dire ici *aligné* et de *ragged* qui signifie *décheté* ; si un paragraphe est aligné d'un côté, il est décheté de l'autre !

Attention, ces macros portent sur tout ce qui suit, il est donc nécessaire de les inclure dans des groupes si on ne veut pas que leurs actions se poursuivent jusqu'à la fin du document.

On notera que le paragraphe doit être déclaré à l'intérieur des accolades externes. En effet, ces macros ne portent que sur des paragraphes (c'est logique) donc n'auront aucune action sur une portion de texte qui n'est pas un paragraphe.

Exemple qui tombe à l'eau.

C'est mieux ainsi !

source :

```
{\centering Exemple qui tombe à l'eau.}\par
{\centering C'est mieux ainsi !\par}
```

5.6.2 L'environnement *verbatim*

C'est le seul environnement où le texte n'est pas du tout modifié. C'est ainsi que j'ai pu noter les commandes L^AT_EX que vous lisez. On utilisera : `verbatim`, `verb` (pour quelques caractères, encadrés par + ou |).

Voici quelques exemples qui suffisent à la compréhension :

```
\begin{verbatim}
  du texte non interprété
  et des maths
  $\dfrac{1}{n^2}$
```

```
\end{verbatim}
```

Pour une seule ligne :

```
\verb+{$\dfrac{1}{n^2}$}+
```

5.7 Les listes

L^AT_EX définit cinq environnements gérant les listes. Trois types de listes sont prédéfinies ; leur syntaxe générale est :

```
\begin{type_de_liste}
  \item premier élément de la liste
  \item deuxième élément de la liste
  :
  \item dernier élément de la liste
\end{type_de_liste}
```

type_de_liste pouvant être *itemize*, *enumerate* ou *description*. Pour la liste *itemize*, chaque élément de la liste sera précédé d'un tiret. Pour la liste *enumerate*, chaque élément de la liste est précédé d'un nombre qui augmente de 1 à chaque fois qu'on passe à l'élément suivant. Enfin, pour la liste *description*, c'est l'utilisateur qui indique à chaque fois ce qui doit se trouver au début des éléments de la liste.

Voyons quelques exemples simples pour commencer :

<u>itemize</u>	<u>enumerate</u>	<u>description</u>
<ul style="list-style-type: none"> – en français, les premiers éléments d'une liste se terminent par un point virgule ; – chaque élément commence par une minuscule ; – le dernier élément a un point. 	<ol style="list-style-type: none"> 1) C'est le premier point. 2) Voici le deuxième. 3) Et enfin le dernier. 	<p>Un Article indéfini singulier. Deux Nombre premier. Trois et etc. Pour faire un groupe</p>

Si on veut modifier ponctuellement le symbole de la liste *itemize*, il suffit d'ajouter le nouveau symbole entre crochets à la suite de la macro *\item* (on peut faire de même avec *enumerate*, mais on perd l'avantage de la numérotation automatique)

<u>source</u>	<u>résultat</u>
<pre>\begin{itemize} \item[\P] jolis symboles ; \item[\&] très esthétique ; \item[?] et illisible ! \end{itemize}</pre>	<p>¶ jolis symboles ; & très esthétique ; ? et illisible !</p>

Les listes peuvent être imbriquées les unes dans les autres. Pour la liste *enumerate*, chaque niveau dispose de son propre symbole de numérotation. L^AT_EX autorise quatre niveaux d'imbrication. Un exemple trouvé sur le net :

<u>source</u>	<u>résultat</u>
<pre> Les matières enseignées sont : \begin{enumerate} \item les mathématiques en raison : \begin{enumerate} \item de leur rigueur ; \item de leur beauté. \end{enumerate} \item la physique parce que : \begin{enumerate} \item il faut que tout le monde vive : \begin{enumerate} \item les profs ; \item les autres \begin{itemize} \item les grands \item les petits \dots \end{itemize} \end{enumerate} \end{enumerate} \item il y en a quand même un peu besoin. \end{enumerate} \end{pre> </pre>	<p>Les matières enseignées sont :</p> <ol style="list-style-type: none"> 1) les mathématiques en raison : <ol style="list-style-type: none"> a) de leur rigueur ; b) de leur beauté. 2) la physique parce que : <ol style="list-style-type: none"> a) il faut que tout le monde vive : <ol style="list-style-type: none"> (i) les profs ; (ii) les autres <ul style="list-style-type: none"> – les grands – les petits ... b) il y en a quand même un peu besoin.

6 Commandes plus élaborées

6.1 Les tableaux

6.1.1 Une aide pour commencer

Vous trouverez à l'adresse :

<http://stephfevre.free.fr/logiciels/LaTable.php> une aide concernant le logiciel gratuit "LaTable",

téléchargeable à <http://www.g32.org/latable/index.html#Downloads>

Ce petit logiciel permet de générer facilement le code L^AT_EX de tableaux, s'ils ne sont pas trop compliqués. Sa prise en main est quasi-instantanée et peut vous aider à démarrer

6.1.2 L'environnement *tabular*

Il permet de faire tous les tableaux (disons que je ne lui ai pas trouvé encore de limitation)

(a) *Syntaxe "basique"*

```
\begin{tabular}[position]{description}
cellule (1,1) & cellule (1,2) \\ % ligne 1
cellule (2,1) & cellule (2,2) \\ % ligne 2
...&... \\
champ1 & champ2 \\ % ligne dernière
\end{tabular}
```

★ On sépare 2 colonnes par & et deux lignes par \\

★ **position** : pour positionner le tableau verticalement, on utilisera :

[t] le haut du tableau est sur la ligne courante

[b] le bas du tableau est sur la ligne courante

aucun paramètre de position (option par défaut) : le milieu du tableau est sur la ligne courante

★ **description** : indique le nombre de colonnes, leur alignement, les filets de séparation verticaux éventuels

l indique une colonne alignée à gauche (**left**) ;

r indique une colonne alignée à droite (**right**) ;

c indique une colonne centrée (**center**) ;

p{*d*} indique une colonne qui peut accueillir des paragraphes dont la largeur est donnée par la dimension *d*, et alignés sur la première ligne

m{*d*} de même mais paragraphe centré verticalement sur la ligne

b{*d*} de même mais paragraphe aligné sur la dernière ligne

*{*num*}{*col*} répète *num* fois le contenu de *col*

Par exemple, si {*description*} est :

{|l|*{3}{c}}, cela revient à {|l|c|c|c|}

Autre exemple :

{|l|p{2cm}|cc} : il y a 4 colonnes, qui sont : alignée à gauche (col 1), à paragraphe de 2 cm (col 2), centrées (col 3 et 4), il y a un trait vertical devant le tableau et entre les colonnes 2 et 3, un double trait vertical entre les colonnes 1 et 2, rien entre les colonnes 3 et 4, ce qui donne :

col1	paragraphe de 2 cm, on peut mettre texte et maths $\sum_{n \geq 1} \frac{x^n}{n!} = e^x$	col3	col4
encore 1	2	3	4

Voici pour comparer p,m,b :

col1	on peut mettre texte et maths $\sum_{n \geq 1} \frac{x^n}{n!} = e^x$	col3
col1	on peut mettre texte et maths $\sum_{n \geq 1} \frac{x^n}{n!} = e^x$	col3
col1	on peut mettre texte et maths $\sum_{n \geq 1} \frac{x^n}{n!} = e^x$	col3

Quelques remarques : Pour chaque colonne de type l, c et r la largeur sera la largeur de la cellule la plus importante (plus les espaces intercolonnes). En revanche, le rôle même des types p, m, b est d'avoir une taille fixe.

Les traits horizontaux sont demandés (après \\) par *hline* ou avant la première ligne

6.2 Les références

Faire référence à un emplacement d'un document peut être une tâche ingrate si elle n'est pas automatisée. En effet, il faudrait être sûr de la page qu'on veut référencer ainsi que du numéro de titre, de tableau, de figure.

Pour cela, il suffit de placer la macro `\label` suivie d'un groupe donnant le mot-clé permettant de retrouver cette référence. Ensuite, la macro `\ref` suivie d'un groupe ayant le même mot-clé donnera le numéro de la structure référencée (titre, tableau, figure, équation) et la macro `\pageref` donnera son numéro de page. Pour arriver à ceci, L^AT_EX écrit ces informations dans un fichier auxiliaire lors de la compilation et il est nécessaire de compiler la source une seconde fois pour que ces informations puissent être lues.

Par exemple, un des sous-titres de ce document : `\subsection{Les environnements\label{envir}}` a changé plusieurs fois de place, aucune importance, grâce à :

source	résultat
<i>Les commandes relatives aux environnements peuvent être vues à la section~\ref{envir} page~\pageref{envir}.</i>	Les commandes relatives aux environnements peuvent être vues à la section 5.6 page 12.

(Vous pouvez vérifier, je suis certaine du résultat puisque ce n'est pas moi qui ai écrit les numéros !)

6.3 En-tête et pied de page

6.3.1 Structure "de base"

Par défaut, dans les styles *article* et *report*, les pages sont numérotées en milieu de bas de page. On peut modifier la présentation d'un document (ou d'une seule page) en déclarant dans le préambule :

commande	effet
<code>\pagestyle{empty}</code>	les entêtes et pieds de page sont vides, le numéro des pages n'apparaît plus
<code>\pagestyle{headings}</code>	les pieds de page sont vides, en en-tête : le titre des sections ou chapitres selon le type de document

(pour une seule page : `\thispagestyle{style}` là où on souhaite modifier, et avec *style* prenant les valeurs *empty*, *plain*, *headings*)⁴

6.3.2 Avec fancyhdr

Vous devez déclarer en préambule : `\usepackage{fancyhdr}`

puis `\pagestyle{fancy}`

ce qui vous donne accès à 6 commandes de placement (3 pour l'en-tête [**head**], 3 pour le bas de page [**foot**]), soit :

⁴on peut aussi personnaliser : non traité ici

<code>\lhead{haut gauche}</code>	<code>\chead{haut centre}</code>	<code>\rhead{haut droit}</code>
Le Texte		
<code>\tfoot{bas gauche}</code>	<code>\tfoot{bas centre}</code>	<code>\tfoot{bas droit}</code>

Par exemple, j'ai déclaré pour ce document :

```
\pagestyle{fancy}
\rhead{p.\thepage}\chead{ Démarrer en \LaTeX}\lhead{\small{M.C.Mondini}}
\tfoot{}
\renewcommand{\headrulewidth}{0.1pt}
```

Remarques : j'ai déclaré : `\tfoot{}` qui est un groupe vide : en effet, par défaut L^AT_EX met le numéro de la page au centre du bas de page

`\renewcommand{\headrulewidth}{0.1pt}` sert à tracer un filet d'épaisseur 0.1pt de séparation entre en-tête et corps de texte `\renewcommand{\footrulewidth}{0.1pt}` ferait de même entre corps de texte et bas de page ; (par défaut, la valeur est 0 cad pas de filet)

6.4 Notes de bas de page

Hypersimple ! Vous avez pu en lire une au début de ce document, elle a été écrite ainsi sur le fichier source :

```
la multitude de Faq\footnote{ Foires Aux Questions}
```

6.5 Table des matières

Avec les titres et sous-titres (cf 5.5 page 11), tout est prêt ! Il suffit d'écrire la commande

`\tableofcontents` là où on désire qu'apparaisse la table des matières. Ici encore, on peut personnaliser en indiquant avant l'appel à "tableofcontents" :

```
\renewcommand{\contentsname}{Sommaire} par exemple et "Table des matières" devient "Sommaire"
```

`\setcounter{tocdepth}{n}` où n est un entier entre -1 et 5 indique la profondeur voulue dans la table des matières, par exemple avec `\setcounter{tocdepth}{4}` en classe article, la table des matières indiquera des parties (niveau 0), aux paragraphes (niveau 4)

Remarque : il faut compiler 2 fois (sauf si on utilise "Texify" ou "PDF Texify" cf 3.1 page 3)

Remarque bis : les titres sous forme étoilée (cf 5.5.2 page 12) n'apparaissant pas, par défaut, dans la table des matières, on doit écrire `\addcontentsline{toc}{niveau}{Titre}` pour les y inclure, où *niveau* indique s'il s'agit d'une section, sous-section etc. et *titre* est le nom sous lequel figurera cette entrée dans la table des matières.

6.6 Index

La construction d'un index est plus délicate que celle de la table des matières et je ne décrirai pas toutes les possibilités. En premier lieu, il faut dire que le document comportera un index en appelant l'extension *makeidx* dans le préambule, avec la syntaxe :

```
\usepackage{makeidx}
```

puis spécifier, également dans le préambule la macro *makeindex*. L'index proprement dit est construit grâce à la macro *printindex* placée là où on veut que l'index soit écrit (le plus souvent en fin de document).

Pour mettre un mot dans un index, il suffit d'employer la macro *index* suivie par un groupe donnant le mot (ou la suite de mots) à indexer. Cette macro présente des syntaxes différentes permettant des comportements variés. Je n'en exposerai que quelques unes.

Attention, supposons que l'on veuille faire apparaître le symbole ζ au niveau de l'index tout en voulant le classer comme s'il s'agissait du mot « zeta », écrire, *index{\zeta\$}* ne produira pas le résultat escompté car ζ sera classé au niveau des mots commençant par le caractère contre-oblique (très au début de l'index, avant toutes les lettres romaines). Pour pouvoir « tromper » le programme chargé du tri, il suffit de taper *index{zeta@\$\zeta\$}* où le caractère @ sert à séparer la partie gauche qui servira de référence pour l'ordre alphabétique de la partie droite qui sera affichée réellement au niveau de l'index. Cette façon de faire est très utile lorsqu'on met des mots accentués dans un index car le programme de tri ne range pas les lettres accentuées au même niveau que les autres lettres.

pour finir, on peut hiérarchiser les entrées : ainsi *index{module!aiguill}* indiqué en page 6 par exemple produira, à la lettre **m** l'entrée :

```
module
  aiguill, 6
```

7 Ecrire des mathématiques

7.1 Principes

L^AT_EX gère seul les caractéristiques particulières de l'écriture mathématique, il suffit pour cela de lui indiquer le passage en mode mathématique (cf Tab. 5 page 21), une fois entré dans ce mode, il faut savoir que :

- toutes les lettres sont en italique (sauf instruction contraire)
- L^AT_EX gère les blancs et ne reconnaît plus les accents, ce qui est gênant pour du texte éventuel, ainsi *\$ fonction définie\$* devient *fonction définie*, mais bien pour ...les maths (heureusement, c'est fait pour !), ainsi *\$2+3=5\$* et *\$2 + 3 = 5\$* donnent le même résultat : $2 + 3 = 5$
- et toujours les caractères réservés (cf Tab. 1 page 6) à respecter
- T_EX gère automatiquement la taille des caractères dans les formules en fonction de leur emplacement (exposant, exposant d'exposant, composantes de fraction, etc.) et du mode (cf encore Tab. 5 page 21)

En conséquence, si l'on veut insérer du texte et des maths dans la même phrase (par exemple : $y = x^2$ où $x > 0$), on pourra

- soit sortir du mode maths puis y entrer à nouveau *\$ y=x^2 \$ où \$ x>0 \$*
- soit passer localement en texte par *\$ y=x^2 \text{ où } x>0 \$* attention dans ce cas à ne pas oublier les blancs de séparation, ici avant et après "où"

7.2 Les modes de présentation

Il y a deux modes de présentation : le mode ligne et le mode visualisation ("display"). Pour chacun, on doit indiquer à L^AT_EX l'entrée et la sortie du mode mathématique choisi.

délimiteurs	mode	exemple
$\$ \dots \$$	ligne	$\sum_{n=1}^{\infty} \frac{1}{n^2}$
$\langle \dots \rangle$	ligne	$\sum_{n=1}^{\infty} \frac{1}{n^2}$
$\backslash begin\{math\} \dots \backslash end\{math\}$	ligne	$\sum_{n=1}^{\infty} \frac{1}{n^2}$
$\$\$ \dots \$\$$	"display"	$\sum_{n=1}^{\infty} \frac{1}{n^2}$
$\langle \dots \rangle$	"display"	$\sum_{n=1}^{\infty} \frac{1}{n^2}$
$\backslash begin\{displaymath\} \dots \backslash end\{displaymath\}$	"display"	$\sum_{n=1}^{\infty} \frac{1}{n^2}$

TAB. 5 – Délimiteurs des modes mathématiques

Personnellement, j'utilise \$ et \$\$ (cad les premiers délimiteurs) car je les "vois" mieux dans le fichier source (et il n'y a qu'un seul caractère à taper dans le premier cas !!) ; remarque : l'entrée en mode "display" s'effectue avec passage à la ligne et le texte est centré ; si on veut éviter de centrer mais utiliser la disposition "display", on écrira par exemple :

```
 $\$ \displaystyle \{ \sum_{n=1}^{\infty} \frac{1}{n^2} \} \$$ 
```

qui donne :

$$\sum_{n=1}^{\infty} \frac{1}{n^2}$$

En tout, il y a quatre tailles prédéfinies : la taille hors texte (display), texte (text), scripte (script) et sous-scripte (scriptscript). La taille peut alors être forcée grâce aux macros `\displaystyle`, `\textstyle`, `\scriptstyle` et `\scriptscriptstyle`. Tout ce qui se trouve après ces macros va prendre la taille spécifiée, à moins que des automatismes de L^AT_EX n'entrent en jeu pour réduire cette taille (fraction, exposant, ...).

7.3 L'aspect du texte

Pour mettre en forme du texte *mathématique* dans une équation, vous devez utiliser des commandes spéciales. Il y en a une pour chaque style de caractère (les styles "slanted" et "petites capitales" ne sont pas disponibles en mode mathématique) :

fonte	source	résultat
Italique	$\$ \Delta f(x)=ab \mathit{ab} \Delta x \$$	$\Delta f(x) = abab\Delta x$
Gras	$\$ f(x)=ab \mathbf{ab} x \$$	$f(x) = ababx$
Machine à écrire	$\$ f(x)=ab \mathtt{ab} x \$$	$f(x) = ababx$
Sans serif	$\$ f(x)=ab \mathsf{ab} x \$$	$f(x) = ababx$
Roman	$\$ f(x)=ab \mathrm{ab} x \$$	$f(x) = ababx$
Calligraphie	$\$ \mathcal{M}_n(K) \$$	$\mathcal{M}_n(K)$
Gothique	$\$ S = \mathfrak{S} \$$	$S = \mathfrak{S}$

TAB. 6 – Styles en mode mathématique

Remarques :

- `\mathit` appelle les italiques du mode texte (seuls les chiffres et les majuscules grecques sont affectés par cette macro car, par défaut, les chiffres et les majuscules grecques ne sont pas en italique).
- Le passage en "gras" fait perdre le caractère "mathématique" (retour aux lettres normales voir le "a"), pour éviter cela, on peut utiliser `\boldmath $f(x)=ababx$ \unboldmath` qui donne : $f(x) = ababx$
- Petit complément : si vous préférez $\mathbb{N}, \mathbb{Z}, \mathbb{R}, \mathbb{C}, \mathbb{Q}$ à $\mathbb{N}, \mathbb{Z}, \mathbb{R}, \mathbb{C}, \mathbb{Q}$, vous devrez utiliser la fonte `mathds`, pour cela il faut déclarer `\usepackage{dsfont}` en préambule

7.4 Structures de base

7.4.1 Opérations élémentaires

Les signes plus, moins et égal sont présents sur le clavier mais pas ceux de la multiplication et de la division (celle de l'école primaire). On peut bien évidemment taper un « / » pour la division mais la macro `\div` donne le symbole \div . Le signe multiplié est obtenu avec la macro `\times` qui donne le symbole \times .

7.4.2 Exposants et indices

Les exposants et les indices utilisent deux caractères réservés de T_EX, à savoir respectivement `^` et `_`. Leur syntaxe est on ne peut plus simple : ce qui suit **immédiatement** `^` est placé en exposant (même chose avec `_` pour les indices). Il faut donc faire attention aux groupes éventuels, ainsi : u_{n+1} s'obtient par `u_{n+1}` et non par `u_n+1` qui donne : $u_n + 1$

Il est également possible de mélanger exposants et indices sans se soucier de l'ordre de spécification.

7.4.3 Fractions

Voici quelques exemples assez parlants pour se passer de commentaires :

source	résultat
<code>\$\$\frac{1}{x^2+1}\$\$</code>	$\frac{1}{x^2+1}$
<code>\$\$\dfrac{1}{x^2+1}\$\$</code>	$\frac{1}{x^2+1}$
<code>\$\$\dfrac{e^{\frac{i\pi}{n}}}{1+n^2}\$\$</code>	$\frac{e^{\frac{i\pi}{n}}}{1+n^2}$

Remarque : le "d" de `dfrac` est en fait une abréviation de `displaystyle`

7.4.4 Délimiteurs

Les délimiteurs sont des symboles destinés à encadrer des sous-formules. On peut modifier leur taille soit manuellement, soit automatiquement. Nous ne verrons ici que la façon automatique et nous renverrons le lecteur avide à des ouvrages spécialisés !

Dans l'expression $f(x)$, les deux parenthèses sont des délimiteurs. La parenthèse gauche est appelée « délimiteur gauche » et je vous laisse deviner comment on appelle la parenthèse droite. Un exemple plus compliqué :

$$f\left(\frac{1+x}{x}\right) = x^2 + \left(\frac{1}{x}\right)^2$$

Le résultat n'est pas du tout esthétique ! Les deux macros qui permettent de régler automatiquement la taille des délimiteurs à la hauteur de la sous-formule sont `\left` à mettre immédiatement avant le délimiteur gauche et `\right` à mettre immédiatement avant le délimiteur droit. Le même exemple donnera alors :

<u>source</u>	<u>résultat</u>
$\$ \left(\frac{1+x}{x} \right) = x^2 + \left(\frac{1}{x} \right)^2 \$$	$f\left(\frac{1+x}{x}\right) = x^2 + \left(\frac{1}{x}\right)^2$

La liste de ces délimiteurs est donnée en table 7

<u>source</u>	<u>résultat</u>	<u>source</u>	<u>résultat</u>	<u>source</u>	<u>résultat</u>
(())	\uparrow	↑
[[]]	\downarrow	↓
\{	{	\}	}	\updownarrow	↕
\lfloor	⌊	\rfloor	⌋	\Uparrow	⇑
\lceil	⌈	\rceil	⌉	\Downarrow	⇓
\langle	⟨	\rangle	⟩	\Updownarrow	↕
/	/	\backslash	\		
		\			

TAB. 7 – Délimiteurs

On peut imbriquer des couples *\left*...*\right* mais il faut toujours qu'un *\left* s'équilibre avec un *\right*, quitte à utiliser un délimiteur vide (cf fin du 7.5.1) sous peine de voir L^AT_EX protester avec véhémence.

7.5 Vecteurs, matrices et déterminants

Il s'agit ici de tableaux mathématiques, en effet le vecteur "simple" est noté : \vec{u} , soit $\$ \vec{u} \$$ pour 1 lettre et \overrightarrow{AB} soit $\$ \overrightarrow{AB} \$$ s'il y en a plusieurs.

2 méthodes sont possibles :

- 1) utiliser un tableau mathématique (7.5.1 page 23)
- 2) utiliser matrix (pas le film, of course !)

7.5.1 Array

De façon générale, les matrices sont considérées comme des tableaux, la seule différence étant que les tableaux étaient construits en utilisant l'environnement *tabular* alors que les matrices sont construites avec l'environnement *array*. Par exemple :

1 2 3	$\$ \begin{array}{*3}{c}$
2 3 1	1 & 2 & 3 \\\
3 4 1	2 & 3 & 1 \\\
	3 & 4 & 1 \\\
	\end{array} \\$

Pour les « vraies » matrices, il faudra ajouter des parenthèses sous la forme de délimiteurs extensibles (donc avec les préfixes *\left* et *\right*). La forme pour les déterminants suivra le même principe en prenant des barres verticales comme délimiteurs. L'exemple précédent devient alors :

```


$$\begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 4 & 1 \end{pmatrix} \quad \begin{array}{l} \left( \begin{array}{*3}{c} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 4 & 1 \end{array} \right) \\ 1 \& 2 \& 3 \\ 2 \& 3 \& 1 \\ 3 \& 4 \& 1 \end{array} \\ \end{array} \right) \$$$

```

```


$$\begin{vmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 4 & 1 \end{vmatrix} \quad \begin{array}{l} \left| \begin{array}{*3}{c} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 4 & 1 \end{array} \right| \\ 1 \& 2 \& 3 \\ 2 \& 3 \& 1 \\ 3 \& 4 \& 1 \end{array} \\ \end{array} \$$$


```

On peut bien sûr remplacer les parenthèses par des crochets, des accolades. A noter qu'ici encore, tout délimiteur ouvert doit avoir un délimiteur fermé associé, c'est pour cela que L^AT_EX définit le délimiteur « vide » qui ne produit aucun symbole : `\left.$` pour ouvrir et `\right.$` pour fermer. Par exemple :

source	résultat
<pre> $\left. \begin{array}{l} x \in E \\ \text{ou} \\ x \in F \end{array} \right\} \iff x + x^2 = 0$ </pre>	$x \in E \left. \begin{array}{l} \text{ou} \\ x \in F \end{array} \right\} \iff x + x^2 = 0$
<pre> $P_{i,j} = \begin{cases} 0 & \text{si } i - j \text{ est impair,} \\ i!(-1)^{(i-j)/2} & \text{si } i - j \text{ est pair.} \end{cases}$ </pre>	$P_{i,j} = \begin{cases} 0 & \text{si } i - j \text{ est impair,} \\ i!(-1)^{(i-j)/2} & \text{si } i - j \text{ est pair.} \end{cases}$

Dans ce deuxième cas, je préfère l'utilisation de "cases" (cf 7.7.1 page 25)

7.5.2 Matrix

Le module `amsmath` permet d'aller plus vite grâce aux environnements `ymatrix` où y peut prendre différentes valeurs, analogues par ailleurs aux déclarations de tableaux à l'exception suivante : il n'est pas, ici, nécessaire de faire la déclaration du nombre de colonnes. Ainsi

```


$$\begin{pmatrix} 1 & 2 & \cos \theta \\ 0 & 1 & -2 \end{pmatrix} \text{ est obtenue par } \begin{array}{l} \begin{pmatrix} 1 & 2 & \cos \theta \\ 0 & 1 & -2 \end{pmatrix} \\ \end{array} \$$$


```

Les diverses possibilités sont données par le tableau

matrix	pmatrix	bmatrix	Bmatrix	vmatrix	Vmatrix	smallmatrix
$\begin{matrix} 1 & 0 \\ 1 & 0 \end{matrix}$	$\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$	$\begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$	$\begin{Bmatrix} 1 & 0 \\ 1 & 0 \end{Bmatrix}$	$\begin{vmatrix} 1 & 0 \\ 1 & 0 \end{vmatrix}$	$\begin{Vmatrix} 1 & 0 \\ 1 & 0 \end{Vmatrix}$	$\begin{smallmatrix} 1 & 0 \\ 1 & 0 \end{smallmatrix}$

TAB. 8 – Environnement `matrix`

7.6 Equations

7.6.1 Numérotées

L^AT_EX numérote automatiquement les équations qui sont dans l'environnement "equation" (cf 5 p.21) Exemple :

source	résultat
<pre>\begin{equation}\label{ED} y''-2y'+y=0 \end{equation}</pre>	$y'' - 2y' + y = 0 \quad (1)$

A noter : par "equation" on entre directement en mode mathématique "display"; on peut appeler les équations par leur référence (appel par label et ref, cf 6.2 p.18), ici, l'équation `\ref{ED}` donnera : l'équation 1

7.6.2 Alignées

C'est ici l'environnement "eqnarray" que nous utilisons qui se traite comme un tableau (cf 7.5.1 p.23) et dans lequel les lignes sont numérotées et où, pour supprimer le numéro de certaines lignes, on ajoute en fin de ligne `\nonumber` Exemple :

source	résultat
<pre>\begin{eqnarray} \int_1^x \frac{1}{t} \text{d}t & = \left[\ln(t) \right]_1^x & \nonumber \\ & = \ln(x) & \end{eqnarray}</pre>	$\int_1^x \frac{1}{t} dt = [\ln(t)]_1^x = \ln(x) \quad (2)$

7.7 Symboles en vrac

7.7.1 Cases

Très utile pour les fonctions définies par morceaux, les disjonctions de cas et autres :

source	résultat
<pre>\$f(x)= \begin{cases} 1 \text{ si } x=0 \\ \frac{\sin x}{x} \text{ si } x \neq 0 \end{cases}\$</pre>	$f(x) = \begin{cases} 1 & \text{si } x = 0 \\ \frac{\sin x}{x} & \text{si } x \neq 0 \end{cases}$

7.7.2 Symboles empilés

Empiler à droite, à gauche, en haut, en bas ne pose aucun problème, voici quelques exemples (souvent fantaisistes !) :

instruction	exemple	
	source	résultat
<code>\overset{dessus}{texte}</code>	<code>\overset{\diamond}{\alpha}</code>	$\overset{\diamond}{\alpha}$
<code>\underset{dessous}{texte}</code>	<code>\underset{\text{définition}}{=}2</code>	$a \underset{\text{définition}}{=} 2$
<code>\sideset{gauche}{droite}{texte}</code>	<code>\sideset_{a^t}_{\{ij\}^{kl}}\bigcap A_{pq}</code>	$\bigcap_{i,j}^{k,l} A_{pq}$
<code>texte_{\substack{ligne1 \\ ligne2 ..}}</code>	<code>\displaystyle{\sum_{\substack{1 < p, q < n \\ p < q \\ p \wedge q = 1}} a_{pq}}</code>	$\sum_{\substack{1 < p, q < n \\ p < q \\ p \wedge q = 1}} a_{pq}$

TAB. 9 – Empilement de symboles

8 Les insertions

8.1 Images

Vaste sujet, je ne fais que l'aborder pour vous donner quelques rudiments pour démarrer. Deux possibilités essentiellement :

- sous pdf, avec en préambule `\usepackage{graphicx}` : on peut insérer des images au format pdf, png, jpeg. Pour cela on écrit : `\includegraphics{image}` où image est le nom du fichier (avec ou sans son extension), si on veut imposer la taille de l'image, on utilisera `\resizebox{largeur}{hauteur}{\includegraphics{image}}` où largeur et hauteur sont exprimées avec leurs unités, si on remplace hauteur par !, l'image obtenue a la largeur demandée, et pour hauteur la hauteur obtenue par application d'un facteur de proportionnalité (et c'est analogue si on remplace largeur par !)
- sous dvi (ou ps) seul le format .eps est reconnu, on a les mêmes possibilités que ci-dessus (includegraphics et resizebox)

Exemple : avec `\resizebox{!}{6cm}{\includegraphics{Lagaffe.jpg}}`, déclaré dans un environnement centré



D'autres possibilités : légendier une image, l'insérer dans un tableau, faire "couler" du texte autour d'une image. Ce sera pour .. une autre fois

8.2 Dessin

on peut dessiner directement sous L^AT_EX grâce à l'environnement "picture". Là encore, ce sera pour une autre fois !

8.3 Fichiers

On peut insérer un fichier (son contenu) par l'instruction `\input{fichier}` à condition qu'il soit situé au même endroit que votre fichier de travail.

Sinon par exemple `\input{F:/Travail/Macros/ toto}` insère le fichier toto.tex situé dans le dossier F :/Travail/Macros

9 Les erreurs

Lors de la compilation, L^AT_EX génère un fichier .log qui donne des renseignements sur cette compilation. Il y en a 3 types essentiellement :

- **les messages de traçage** de compilation : ce sont de simples renseignements, je n'en parlerai plus
- **les avertissements** : souvent peu importants, inévitables dans un gros document ; ils n'empêchent pas la compilation (L^AT_EX répare tout seul !), je ne m'en occupe pas non plus
- **les erreurs** : plus graves car la compilation s'arrête, le compilateur passe en mode interactif et propose plusieurs actions qui doivent être suivies de la touche **Entr** dont :
 - **H** : demande d'aide supplémentaire ;
 - **Entr** ignorer : dans certains cas, la compilation peut continuer
 - **X** : arrêter la compilation, retour au fichier où, en fonction de l'erreur signalée, on fera les corrections voulues

10 Personnaliser L^AT_EX

On peut paramétrer presque tout : créer de nouveaux environnements, des macros, des compteurs, utiliser des instructions conditionnelles...

Voici quelques notions

10.1 Compteurs

Si on veut numéroter ses exercices, par exemple, on peut

- **créer** un compteur : `\newcounter{nom}`, où nom est .. le nom donné à notre compteur
- en **modifier** la valeur par `\addtocounter{nom}{n}` où n est la valeur entière à ajouter,
- lui **donner une valeur** choisie par `setcounter{nom}{p}` où p est la valeur entière à attribuer au compteur
- **utiliser la valeur** du compteur avec `\thenom`

A noter : la valeur du compteur appelé (par exemple) *compte* est obtenu par la variable `\thecompte`, ainsi le numéro de page est donné par `\thepage` : *ici nous sommes en page* `\thepage` donne : *ici nous sommes en page 27* (cf autre exemple p.29)

10.2 Nouvelle commande

10.2.1 la macro "newtheorem"

En première approche, pour créer un type d'énoncés (par exemple une liste d'exercices, si on ne veut pas utiliser une liste d'énumération) on peut utiliser la macro (déclarée de préférence dans le préambule) :

```
\newtheorem{nom abrégé}{titre de l'énoncé}[compteur de référence]
```

où

- *nom abrégé* est le nom par lequel on désigne ce type d'énoncé (n'apparaît pas dans le document final),
- *titre de l'énoncé* est le nom de ce type d'énoncé, tel qu'il apparaît dans le document final
- *compteur de référence* (option facultative) : la numérotation dépend de ce compteur (par exemple une section) et reprend donc à 1 quand on en change (dans cet exemple, si on change de section)

Un exemple sera plus clair : si on déclare :

```
\newtheorem{exo}{Exercice}
```

source	résultat
<pre>\begin{exo} Montrer que f est constante \end{exo}</pre>	<p>Exercice 1. <i>Montrer que f est constante</i></p>
<pre>\begin{exo}[facultatif] Vérifier $\int_0^1 2t dt = 1$ \end{exo}</pre>	<p>Exercice 2 (facultatif). <i>Vérifier</i></p> $\int_0^1 2t dt = 1$

Une remarque : on peut ajouter (en option) un sous-titre, qui sera écrit entre parenthèses (ici, c'est l'indication *[facultatif]* ajoutée après le "begin{exo}")

Autre :

```
\newtheorem{ques}{Question}[section]
```

source	résultat
<pre>\begin{ques} Montrer que f est constante \end{ques}</pre>	<p>Question 10.1. <i>Montrer que f est constante</i></p>
<pre>\begin{ques}[facultatif] Vérifier $\int_0^1 2t dt = 1$ \end{ques}</pre>	<p>Question 10.2 (facultatif). <i>Vérifier</i></p> $\int_0^1 2t dt = 1$

On peut aussi créer ses propres environnements avec ses compteurs (cf exemple p.29)

10.2.2 macro très simple

Essentiellement pour raccourcir une saisie longue et assez fréquente, on utilise `\def`.

Par exemple `\def\ssi{si et seulement si}` (que l'on peut déclarer dans le préambule), ou de manière équivalente

`\newcommand{\ssi}{si et seulement si}` (cf (a))

En écrivant `\ssi`, on aura : si et seulement si

Une remarque à ce sujet : pour bien gérer les espaces, je vous conseille (voir l'exemple) de ne pas laisser d'espace dans la définition, ni en début ni en fin d'expression, ainsi vous aurez le choix :

`\ssi. On a ..` qui donne : "si et seulement si. On a .." **sans** espace

`\ssi\ on a ..` qui donne : "si et seulement si on a .." **avec** espace

10.2.3 macros .. moins simples

(a) *newcommand*

Syntaxe : `\newcommand{nomCommande}[NbParam]{LaMacro}` où

- `\nomCommande` est bien sûr le nom qui désigne cette macro
- `NbParam` est le nombre de paramètres (cad d'arguments) utilisés dans la macro (entier inférieur à 9, 0 par défaut)
- `LaMacro` est la succession des commandes à effectuer. Dans cette liste, chacun des arguments sera désigné par #n où n est son numéro d'ordre.

Par exemple,

```
\newcommand{\dte}[3]{\$#1 x+#2 y+#3 =0\$}
```

permet d'obtenir l'équation d'une droite du plan : *la droite d'équation `\dte{-}{2}{5}` donne :*
la droite d'équation $-x + 2y + 5 = 0$

Pour modifier une macro existante, on remplacera `newcommand` par `renewcommand`

(b) *newenvironment*

Un nouvel environnement sera déclaré par : `\newenvironment{nomEnvir}[NbParam]{DéfinitionsDeb}{DéfinitionsFin}`
où

- `nomEnvir` est .. le nom de l'environnement (ça c'est original !)
- `NbParam` est le nombre de paramètres utilisés dans la macro (entier entre 1 et 9, 0 par défaut)
- `DéfinitionsDeb` indique la configuration et les commandes d'entrée d'environnement
- `DéfinitionsFin` indique la configuration et les commandes de sortie d'environnement

Exemple 1 :

Vous voulez écrire un exercice, avec **exercice** en gras, énoncé en italique puis changer de paragraphe en fin d'exercice

```
\newenvironment{ex}[1][\textbf{Exercice : } \itshape #1 ]{\par}
```

```
\begin{ex}
```

Montrer que $f(x)=2x+5$

```
\end{ex} L'exo suivant est en nouvelle ligne
```

Exercice : *Montrer que $f(x) = 2x + 5$*

L'exo suivant est en nouvelle ligne

Encore mieux : on veut numéroter automatiquement les exercices :

```
\newcounter{nbexer}
```

```
\newenvironment{exer}[1][\addtocounter{nbexer}{1}\textbf{Exercice \thenbexer : } \itshape #1 ]{\par}
```

```
\begin{exer}
Montrer que  $f(x)=x^2-1$ 
\end{exer}
\begin{exer}[(\textnormal{bonus})]
Résoudre  $f(x)=0$ 
\end{exer}
```

Exercice 1 : *Montrer que $f(x) = x^2 - 1$*

Exercice 2 : (bonus) *Résoudre $f(x) = 0$*

11 Où trouver de l'aide ?

11.1 Sur internet

demander à Google ! Voici quelques références (volontairement en nombre restreint, car il y en a .. beaucoup !) que j'ai trouvées intéressantes :

- 1) <http://www.grappa.univ-lille3.fr/FAQ-LaTeX/>
- 2) <http://latex.perseguers.ch/>
- 3) <http://www.tuteurs.ens.fr/logiciels/latex/manuel.html>

et des documents pdf :

- 1) <http://www.ctan.org/tex-archive/info/lshort/french/flshort-3.20.pdf>
Remarque : la version dvi est plus légère, vous pouvez la consulter à :
<http://www.ctan.org/tex-archive/info/lshort/french/>
chercher le document *flshort-3.20.dvi*
- 2) <http://tex.loria.fr/general/aide-memoire-latex-seguin1998.pdf>

11.2 Dans les livres

Personnellement, j'utilise, de façon complémentaire les 2 excellents ouvrages suivants :

- 1) **ROLLAND, Christian** *L^AT_EX par la pratique* Paris : O'Reilly, 1999. ISBN 2-84177-073-7
- 2) **DESGRAUPES, Bernard** *L^AT_EX Apprentissage, guide et référence 2^e édition*. Paris : Vuibert, 2003. ISBN 2-7117-4809-X

Un petit ouvrage "portable" à garder avec soi

Collectif d'auteurs *L^AT_EX pour l' impatient* H-K 2005. ISBN 2-914010-67-2

Index

array, [23](#)

cases

`\cases`, [25](#)

eqnarray

`\eqnarray`, [25](#)

équation

`\eqnarray`, [25](#)

`\equation`, [25](#)

fonte, [7](#)

mathématique, [21](#)

guillemets

`\fg`, [6](#)

`\og`, [6](#)

matrix, [23](#)

module

`aeguill`, [6](#)

`dsfont`, [21](#)

newcommand

`\newcommand`, [29](#)

`\renewcommand`, [29](#)

newenvironment

`\newenvironment`, [29](#)

page

`\thepage`, [27](#)

vecteur, [23](#)

`\vec`, [23](#)