

Table des matières

Avant-propos	5
1 Introduction	7
1.1 L ^A T _E X, késako?	7
1.2 Les traitements de texte disponibles à l'ENSTA	7
1.3 Conventions typographiques utilisées dans ce manuel	8
2 Pourquoi apprendre L^AT_EX?	9
2.1 La philosophie de L ^A T _E X	9
3 Votre premier document	10
3.1 Avant de commencer	10
3.2 Le programme <i>x_latex</i>	10
3.3 Allons-y!	12
3.3.1 L'édition	12
3.3.2 La compilation	12
3.3.3 La détection et la correction des erreurs	12
3.3.4 La visualisation	13
3.3.5 L'impression	14
3.3.6 La fin	14
3.4 Résumé	14
4 Les concepts de base de L^AT_EX	15
4.1 Le document L ^A T _E X	15
4.1.1 Syntaxe d'une commande L ^A T _E X	15
4.1.2 La commande <code>\documentstyle</code>	16
4.1.3 Le préambule	17
4.1.4 Le document	17
4.2 Structure du texte	17
4.2.1 Les paragraphes	17
4.2.2 Les mots	18
4.2.3 L'espace insécable:	18
4.2.4 Les commentaires	19
4.3 Caractères spéciaux utilisés par L ^A T _E X	19

4.4	Les problèmes de césure : l'avertissement <code>Overfull \hbox</code>	19
5	Les commandes de L^AT_EX	21
5.1	Syntaxe d'une commande	21
5.2	Caractères spéciaux	21
5.2.1	Symboles	22
5.2.2	Les lettres accentuées	23
5.3	Commandes de changement de style et de taille	24
5.4	Commandes de saut de ligne, de saut de page et d'espacement	26
5.5	Commandes de sectionnement du document	27
5.6	La page de garde	28
5.7	Commandes d'annotation du document	29
5.7.1	Les notes de bas de page	29
5.7.2	Les notes dans la marge	30
5.8	Les références croisées	30
6	Les environnements de L^AT_EX	31
6.1	L'environnement <code>em</code>	32
6.2	Les environnements <code>quote</code> et <code>quotation</code>	32
6.3	L'environnement <code>verse</code>	33
6.4	Les environnements <code>center</code> , <code>flushleft</code> et <code>flushright</code>	34
6.5	L'environnement <code>verbatim</code>	35
6.6	Les environnements de liste : <code>itemize</code> , <code>enumerate</code> et <code>description</code>	35
6.7	L'environnement <code>tabbing</code>	37
6.8	Les tableaux : l'environnement <code>tabular</code>	39
6.9	Les environnements <code>table</code> et <code>figure</code>	41
7	L^AT_EX mathématique	43
7.1	Les modes mathématiques	43
7.1.1	L'environnement <code>math</code>	43
7.1.2	L'environnement <code>displaymath</code>	44
7.2	Structures couramment utilisées en mathématiques	44
7.2.1	Indices et exposants	44
7.2.2	Texte à l'intérieur d'une formule	45
7.3	Lettres, symboles et accents	46
7.3.1	Lettres grecques	46
7.3.2	Majuscules calligraphiques (ou anglaises)	47
7.3.3	Symboles d'opérateurs binaires	47
7.3.4	Symboles de relations binaires	47
7.3.5	Flèches	47
7.3.6	Symboles divers	49
7.3.7	Accents	49
7.3.8	Superposer deux symboles	50
7.4	Modifier l'espacement	50

7.5	Fractions	51
7.6	Racines	51
7.7	Délimiteurs	51
7.8	Symboles de taille variable	52
7.9	Logarithmes et autres fonctions	54
7.10	Les tableaux : l'environnement <code>array</code>	55
7.11	Les équations numérotées : l'environnement <code>equation</code>	56
7.12	Les équations alignées : l'environnement <code>eqnarray</code>	57
7.13	Théorèmes, définitions...	57
8	L_AT_EX pro	59
8.1	Définir de nouvelles commandes et de nouveaux environnements	59
8.1.1	Définir de nouveaux environnements	60
8.2	Inclure des dessins avec <code>xfig</code>	61
8.2.1	Sauvegarde d'un dessin	61
8.2.2	Inclure un dessin dans un document L _A T _E X	61
8.3	Faire soi-même sa page de garde : l'environnement <code>titlepage</code>	62
8.4	Faire une bibliographie	63
8.4.1	L'environnement <code>thebibliography</code>	63
8.4.2	BIB _T E _X	64
8.5	Faire un index	64
8.5.1	La commande <code>\index</code>	65
8.5.2	Formatage de l'index : le programme <code>makeindex</code>	66
8.5.3	Automatiser la compilation du document en utilisant un <i>Makefile</i>	66
A	Règles typographiques françaises et anglaises	67
A.1	Règles typographiques françaises	67
A.2	Règles typographiques anglaises	68
B	S_LT_EX	69
B.1	Le fichier de base	69
B.2	Les fichiers de transparents	70
C	Reprogrammation des touches de fonction	71
C.1	Comment X gère le clavier	71
C.2	Modifier les associations <code>keycode/keysym</code>	71
	Bibliographie	72
	Index	73

Liste des tableaux

5.1	Symboles divers	22
5.2	Symboles pour le français et les langues étrangères	22
5.3	Accents	23
5.4	Commandes simplifiées d'accentuation	23
5.5	Accents et touches de fonction	24
5.6	Commandes de changement de style	24
5.7	Commandes de changement de taille	25
5.8	Commandes de sectionnement	28
6.1	Exemple d'environnement <code>table</code>	42
7.1	Lettres grecques	46
7.2	Symboles d'opérateurs binaires	47
7.3	Symboles de relations binaires	48
7.4	Flèches	48
7.5	Symboles divers	49
7.6	Accents	49
7.7	Commandes d'espacement	51
7.8	Délimiteurs	52
7.9	Symboles de taille variable	53
7.10	Logarithmes et autres	54

Liste des figures

3.1	Xlatex	11
3.2	Xdvi	13
3.3	Fenêtre d'impression	14
3.4	Session L ^A T _E X	14
5.1	Figure vide	27

Avant-propos

Au sujet de ce manuel

Ce manuel a été écrit à l'intention des élèves de l'École Nationale Supérieure de Techniques Avancées (ENSTA). Il décrit donc L^AT_EX tel qu'il y est installé. Il se peut que l'installation de L^AT_EX sur votre site soit quelque peu différente. En particulier, peuvent être absents :

- le fichier */etc/proto/tex_env*;
- le programme *xlatex*;
- les options de style `A4`, `french`, `epsf`, `farticle` et `freport`;
- la reprogrammation des touches de fonction en caractères accentués.

Consultez votre administrateur L^AT_EX pour savoir ce qui est disponible sur votre site ou pour lui demander d'installer ce qui manque.

Où trouver ce manuel?

L'arborescence source complète de ce manuel (avec les fichiers DVI et PostScript) ainsi que le fichier PostScript seul sont disponibles par FTP anonyme sur les sites suivants :

- *grasp1.univ-lyon1.fr:/*
- *excalibur.ens.fr:/pub/babafou*

Je tiens d'ailleurs à remercier Christophe WOLFHUGEL et Thierry BESANÇON de m'avoir permis d'utiliser leurs archives FTP.

Remarques, commentaires et félicitations

Je serais heureux de savoir ce que vous pensez de mon manuel (en bien ou en mal). Pour cela, vous pouvez m'envoyer un message à l'adresse suivante :

`babafou@ensta.fr`

Chapitre 1

Introduction

CE MANUEL est destiné à vous apprendre $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ¹ dans la joie et la bonne humeur. Après l'avoir lu, vous devrez être en mesure d'utiliser $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ pour taper tous vos documents (rapports, articles pour la PP...).

1.1 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, késako?

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ est un traitement de texte d'une très grande puissance et produisant des documents d'une excellente qualité. Il est utilisé par beaucoup d'étudiants, de chercheurs et d'éditeurs à travers le monde.

Pour vous donner une idée de ce que $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ est capable de faire, feuillotez ce manuel, il a été entièrement réalisé avec $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

L'intérêt de $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ par rapport à d'autres traitements de texte est qu'il dispose d'un langage permettant de lui ajouter de nouvelles fonctions (voir [6]).

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ est une extension d'un traitement de texte appelé $\text{T}_{\text{E}}\text{X}$ (le nom $\text{T}_{\text{E}}\text{X}$ vient du grec $\tau\epsilon\chi\eta$ ², qui veut dire « art »).

1.2 Les traitements de texte disponibles à l'ENSTA

Deux traitements de texte sont disponibles à l'ENSTA :

- $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, installé sur toutes les stations UNIX de l'école, sauf les Sun 3 (soit environ soixante-dix machines);
- Word, installé sur les PCs sous MS-DOS des salles saumon et beige (soit vingt machines).

1. Prononcez $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ comme « varech ».

2. D'où vient le mot « technique », ce qui explique la prononciation de $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

1.3 Conventions typographiques utilisées dans ce manuel

Afin de faciliter la lecture de ce document, un certain nombre de conventions typographiques ont été utilisées :

- le style `machine à écrire` est utilisé pour les exemples de code `LATEX`, pour les références aux commandes `LATEX` et pour les commandes UNIX tapées au clavier ;
- l'*italique* est utilisé lors de la définition de nouveaux termes, pour les arguments des commandes `LATEX`, pour les noms de fichiers et les commandes UNIX.

Voici donc un exemple de code `LATEX` :

```
Ceci est un exemple de code \LaTeX. Comme vous le voyez, cela n'a
rien de très compliqué.
```

Ici, je fais référence à la commande `\bigskip`, je définis ce qu'est un *environnement*, j'explique la signification de l'argument de la commande `\chapter{nom}`, je parle du fichier *toto.tex* et de la commande UNIX *ls*, alors que là, je tape la commande `ls`.

Chapitre 2

Pourquoi apprendre \LaTeX ?

Vous avez décidé d'apprendre à utiliser \LaTeX : bravo ! Mais vous vous demandez certainement quel est l'intérêt de \LaTeX par rapport à Word. En effet, \LaTeX est réputé être d'un abord difficile alors que Word est beaucoup plus simple et s'utilise de manière intuitive (qui donc a lu la documentation de Word ?). Voici, à mon avis, les principaux arguments en faveur de \LaTeX :

- il n'y a que vingt PCs sur lesquels on peut utiliser Word alors que \LaTeX est disponible sur toutes les stations UNIX (il y en a environ soixante-dix) ;
- \LaTeX est particulièrement bien adapté aux textes contenant des formules mathématiques ;
- les fonctions complexes, telles la confection d'une table des matières ou d'un index, sont très difficiles à réaliser avec Word, alors qu'avec \LaTeX , une seule commande suffit ;
- \LaTeX est extensible (voir section 1.1).

Bien sûr, \LaTeX a aussi ses défauts (enfin, surtout un), mais nous les verrons après, lorsque vous en saurez un peu plus à son sujet.

2.1 La philosophie de \LaTeX

\LaTeX n'est pas un traitement de texte à proprement parler, comme Word, c'est un *formateur de texte*. Un document \LaTeX est un fichier ASCII tapé avec n'importe quel éditeur (*aXe*, *vi*...) et contenant le texte du document mélangé avec des commandes de formatage. Ceci a une conséquence très importante : lors de la frappe, on ne voit pas le document tel qu'il sera imprimé. On dit que \LaTeX n'est pas *WYSIWYG* (*What You See Is What You Get*), contrairement à Word. Pour visualiser le document, il faut le compiler avec \LaTeX , puis utiliser un programme de visualisation. C'est certainement le gros inconvénient de \LaTeX .

Chapitre 3

Votre premier document

VOUS voilà au pied du mur. Dans ce chapitre, vous allez réaliser votre premier document avec L^AT_EX. Vous allez apprendre :

- à compiler un document ;
- à détecter et à corriger les erreurs qu'il peut contenir ;
- à le visualiser ;
- à l'imprimer.

La babasse ne s'apprenant bien que sur une babasse, allez donc vous logger.

3.1 Avant de commencer

Avant de pouvoir utiliser L^AT_EX, il y a certaines choses à faire. Éditez votre fichier `.cshrc` et ajoutez la ligne

```
source /etc/proto/tex_env
```

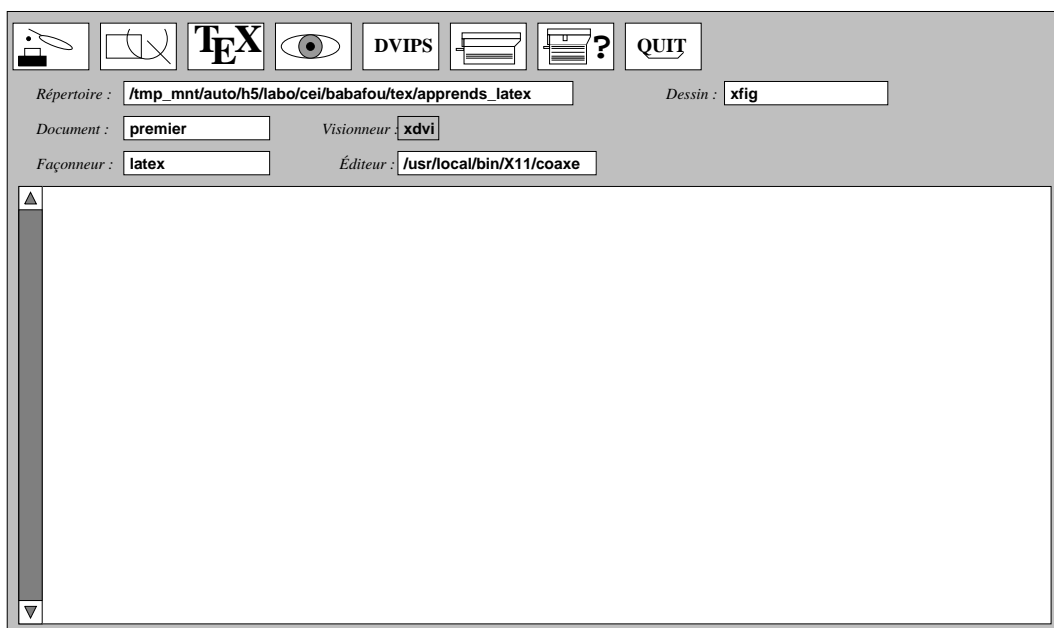
ou enlevez le dièse qui se trouve devant. Cette commande permet de définir des variables d'environnement dont L^AT_EX a besoin. Puis tapez

```
source ~/.cshrc
```

afin de prendre cette modification en compte. Enfin, créez un répertoire `tex` dans lequel vous rangerez vos textes. Tout ceci étant fait une fois pour toutes, vous allez pouvoir taper votre premier texte avec L^AT_EX.

3.2 Le programme *xlatex*

Placez-vous dans votre répertoire `tex`, puis lancez le programme `xlatex` par la commande `xlatex premier.tex`. *Xlatex* est une interface graphique permettant d'utiliser

FIG. 3.1 - *Xlax*

de manière simple L^AT_EX et quelques autres programmes. Une fenêtre s'affiche (voir figure 3.1).

Il est tout à fait possible de se passer de *xlatex* et de taper des commandes directement dans un shell, mais *xlatex* est tellement pratique...

Les boutons regroupés dans la partie supérieure de la fenêtre ont les rôles suivants (les commandes UNIX équivalentes sont précisées lorsque cela est nécessaire) :

- le premier bouton permet d'éditer le document avec l'éditeur de texte dont le nom figure dans le champ *Editeur*, un peu en dessous (vous pouvez y taper le nom de votre éditeur favori si l'éditeur par défaut ne vous convient pas) ;
- le deuxième bouton appelle le programme *xfig* (c'est un programme de dessin, voir section 8.2) ;
- le troisième bouton compile le document avec L^AT_EX (ce qui revient à taper `latex premier.tex`) ;
- le quatrième bouton permet de visualiser le document avec le programme *xdvi* (ce qui revient à taper `xdvi premier.dvi`) ;
- le cinquième bouton convertit le fichier *.dvi* en PostScript (ce qui revient à taper `dvips premier.dvi`) ;
- le sixième bouton permet d'imprimer le document ;

- le septième bouton affiche la file d’attente de l’imprimante;
- enfin, le dernier bouton permet de quitter *xlatex*.

3.3 Allons-y !

C’est parti, on commence.

3.3.1 L’édition

Éditez votre texte en cliquant sur le premier bouton de *xlatex* et tapez ceci :

```
\documentstyle[11pt,A4,french]{article}

\begin{document}

Voici mon premier document avec \LaTeX.

\end{document}
```

3.3.2 La compilation

Puis sauvez votre texte et compilez-le (troisième bouton de *xlatex*). Les commentaires de $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ s’affichent dans la partie centrale de la fenêtre. Si vous n’avez pas fait d’erreur, `--- Exit latex ---` apparaît après ces commentaires.

3.3.3 La détection et la correction des erreurs

Si vous vous êtes trompé quelque part, c’est un message de ce genre qui s’affiche :

```
! Undefined control sequence.
1.5 Voici mon premier document avec \Latex
?
```

Lorsqu’il détecte une erreur, $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ vous indique :

- la nature de cette erreur (! Undefined control sequence.);
- la ligne où se situe cette erreur (1.5);
- ainsi que l’endroit précis de cette erreur (pour vous l’indiquer, $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ passe à la ligne) : ici, la commande inconnue est `\Latex`.

Dans ce cas, tapez `x` ou `<Control>D` pour arrêter la compilation, corrigez l’erreur dans le texte, sauvez-le, puis recompilez.

3.3.4 La visualisation

Pour visualiser votre document, cliquez sur le quatrième bouton de *latex*. Une nouvelle fenêtre s'ouvre (voir figure 3.2).

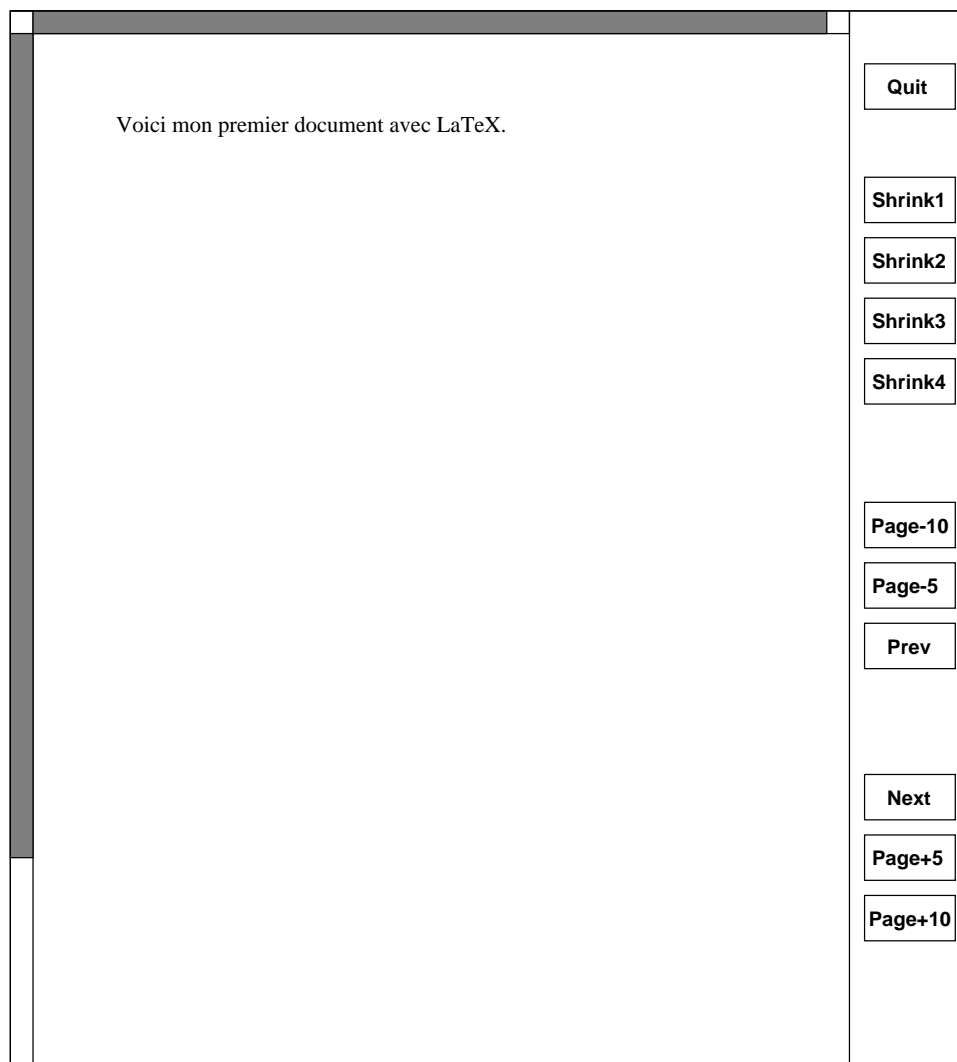


FIG. 3.2 - *Xdvi*

Vous pouvez vous déplacer dans le document et modifier le grossissement grâce aux boutons placés dans la partie droite de la fenêtre. Vous pouvez aussi zoomer en cliquant sur le texte avec les boutons de la souris (bouton de gauche = petit zoom, bouton du milieu = moyen zoom, bouton de droite = grand zoom).

3.3.5 L'impression

Pour imprimer votre document, cliquez sur le sixième bouton de *xlatex*. La fenêtre de la figure 3.3 apparaît.

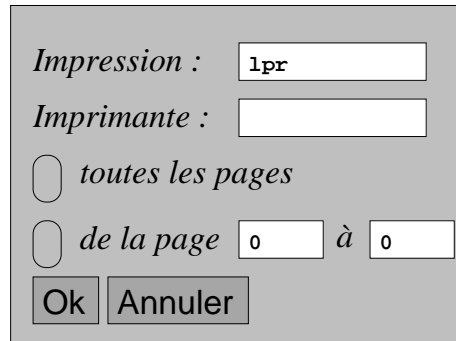


FIG. 3.3 - Fenêtre d'impression

Tapez le nom de l'imprimante dans le champ *Imprimante*, puis cliquez sur le bouton **Ok** pour lancer l'impression.

Il est possible de n'imprimer que certaines pages en cliquant sur le deuxième bouton ovale et en indiquant les pages à imprimer.

3.3.6 La fin

Enfin, pour quitter *xlatex*, cliquez sur le dernier bouton.

3.4 Résumé

La session L^AT_EX type est résumée dans la figure 3.4 :

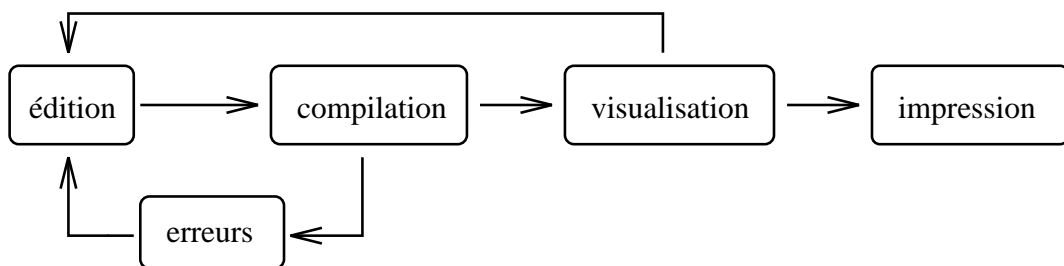


FIG. 3.4 - Session L^AT_EX

Chapitre 4

Les concepts de base de L^AT_EX

AVANT de pouvoir utiliser pleinement L^AT_EX et sa puissance, il faut comprendre certains concepts. Ce chapitre vous initiera plus profondément à la philosophie et à l'esprit L^AT_EX.

4.1 Le document L^AT_EX

Un document L^AT_EX est un mélange de texte et de commandes de formatage. Reprenons l'exemple précédent :

```
\documentstyle[11pt,A4,french]{article}

\begin{document}

Voici mon premier document avec \LaTeX.

\end{document}
```

Comme vous l'avez vu dans le chapitre précédent, tout ceci imprime finalement le texte: « Voici mon premier document avec L^AT_EX. ». Mais, attention, le reste n'est pas inutile, loin de là! Ce sont les fameuses commandes de formatage. Examinons la première:

```
\documentstyle[11pt,A4,french]{article}
```

Cette commande est indispensable et doit figurer en première ligne de tout document L^AT_EX. Elle définit le *style* du document ainsi que les *options* à lui appliquer.

4.1.1 Syntaxe d'une commande L^AT_EX

On reconnaît ici la syntaxe générale d'une commande L^AT_EX :

```
\commande[option]{argument}
```

- une commande commence par un backslash \ ;

- celui-ci est suivi par le nom de la commande ;
- viennent ensuite les arguments optionnels, entre crochets ;
- puis les arguments obligatoires, entre accolades.

Voici quelques exemples de commandes :

- commande sans argument :
`\bigskip`
- commande avec argument :
`\chapter{nom}`
- commande avec argument optionnel :
`\item[nom]`
- commande avec argument obligatoire et argument optionnel :
`\cite[page 150]{lport:latex}`

4.1.2 La commande `\documentstyle`

Revenons à la commande `\documentstyle` et à ses arguments.

Le style d'un document

Le style définit le type du document. Il existe quatre styles :

`article` est utilisé pour les documents courts (rapports de TP, de mini-projet...);

`report` est utilisé pour les documents plus longs qui doivent être découpés en chapitres (le style `article` ne gère pas les chapitres);

`book` est utilisé pour les livres (il y a peu de différences avec le style `report`);

`letter` permet d'écrire des lettres.

Si vous n'utilisez pas l'option de style `french` (voir juste après), chargez les styles `farticle` et `freport` à la place de `article` et `report`. Ces deux styles redéfinissent quelques noms en bon français, comme « Table des matières » ou « Chapitre » (c'est mieux que "Table of contents" et "Chapter"). Néanmoins, je vous conseille quand même d'utiliser l'option de style `french` avec les styles standards.

Les options de style

Les options de style permettent de modifier la mise en page ou de définir de nouvelles commandes. Ce sont elles qui permettent d'ajouter de nouvelles fonctions à $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

Expliquons quelque peu les options utilisées ici :

`11pt` porte la taille des caractères à onze points¹ (la taille normale, sans option, est de dix points ; il existe aussi une option `12pt`) ;

`A4` permet de régler les marges du document pour une impression sur du papier au format A4 (cette option est indispensable, sinon, vous vous retrouverez avec des marges pour papier américain) ;

`french` adapte les styles $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ au français et définit de nouvelles commandes, je vous conseille de toujours employer cette option.

Il existe beaucoup d'autres options de style (voir [6]). Nous en verrons quelques unes par la suite.

4.1.3 Le préambule

Le *préambule* est la partie du texte comprise entre les commandes `\documentstyle` et `\begin{document}`. Elle est vide dans notre exemple, mais elle peut contenir certaines commandes.

4.1.4 Le document

Le document est contenu entre `\begin{document}` et `\end{document}`. Tout texte placé après `\end{document}` est ignoré.

On remarque ici une structure très utilisée en $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$: la structure d'*environnement* (voir le chapitre 6 pour une description complète des environnements de $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$).

4.2 Structure du texte

4.2.1 Les paragraphes

Un texte $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ est composé de *paragraphes* séparés les uns des autres par une ou plusieurs lignes blanches :

Voici un paragraphe assez court.

Ici commence un autre paragraphe. Comme vous le voyez, les fins de ligne n'ont aucune importance sur le formatage du paragraphe. Seules les lignes blanches comptent.

Enfin, le dernier paragraphe conclut cet exemple.

1. Le point est l'unité de mesure utilisée en typographie. Un point vaut $1/72^{\text{e}}$ de pouce (un pouce = 2,54 cm).

Ce qui produit le résultat suivant :

Voici un paragraphe assez court.

Ici commence un autre paragraphe. Comme vous le voyez, les fins de ligne n'ont aucune importance sur le formatage du paragraphe. Seules les lignes blanches comptent.

Enfin, le dernier paragraphe conclut cet exemple.

4.2.2 Les mots

Un paragraphe est composé de *mots* séparés par des signes de ponctuation ou des espaces (un retour à la ligne est considéré comme un espace). Vous pouvez insérer entre deux mots autant d'espaces que vous voulez (mais un seul fera parfaitement l'affaire). Ainsi, les deux textes suivants produiront le même résultat :

Voici un exemple de texte, avec des espaces.

```
Voici      un      exemple
de      texte,    avec
      des  espaces.
```

Enfin, un petit conseil. Comme un retour à la ligne est équivalent à un espace, profitez-en pour couper vos paragraphes en fin de ligne. En effet, il est plus facile de se déplacer avec les touches fléchées dans un paragraphe composé de plusieurs lignes que dans un paragraphe qui tient en une seule ligne. De plus, certains éditeurs de texte limitent la longueur des lignes (souvent à 127 caractères).

4.2.3 L'espace insécable : ~

Dans certains cas, il faut empêcher L^AT_EX de séparer deux mots en fin de ligne, comme ici :

```
Connaissez-vous le nom de la série dans laquelle le héros est le numéro
6?
```

Le « 6 » en début de ligne est du plus mauvais effet.

On aimerait pouvoir coller le « 6 » au mot qui le précède pour que L^AT_EX ne les sépare jamais (c'est beau l'amour!). Ceci est possible grâce au caractère ~. Ce caractère est un *espace insécable*. Il est considéré comme un espace (et donc apparaîtra comme tel à l'impression), mais L^AT_EX ne séparera jamais en fin de ligne les deux mots qu'il lie pour le meilleur et pour le pire :

```
Connaissez-vous le nom de la série dans laquelle le héros est le
numéro~6 ?
```

Connaissez-vous le nom de la série dans laquelle le héros est le numéro 6 ?

N'utilisez pas systématiquement le ~, cela nuit à la lisibilité du texte. Ne vous en servez que lorsque cela est nécessaire.

4.2.4 Les commentaires

L^AT_EX permet d'inclure des commentaires dans le texte. Ceux-ci ne seront pas imprimés. Ils peuvent servir de points de repère à l'auteur, lui rappeler qu'il n'a pas fini d'écrire un paragraphe, etc.

Un commentaire commence par un % et se termine à la fin de la ligne :

```
Ceci est du texte normal.      % ceci est un commentaire
Ceci est la suite du texte.
```

4.3 Caractères spéciaux utilisés par L^AT_EX

Vous avez remarqué que certains caractères ont une signification spéciale pour L^AT_EX. Ainsi, le \ indique une commande, le ~ un espace insécable et le % un commentaire.

Il existe dix caractères réservés :

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

Ceux-ci ne peuvent donc pas être imprimés tels quels.

Pour les sept premiers, il existe une commande spéciale consistant en un \ (puisque c'est une commande) suivi du caractère en question. Ainsi,

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

impriment

```
# $ % & _ { }
```

Pour les trois autres, il n'existe aucune commande permettant de les imprimer. On peut cependant utiliser la commande \verb (voir ???) ou les commandes \tilde, \chap et \backslash, définies par l'option de style french.

4.4 Les problèmes de césure : l'avertissement `Overfull \hbox`

Il arrive parfois que L^AT_EX ne sache pas à quel endroit couper un mot en fin de ligne. Et, plutôt que de le couper n'importe où, il préfère ne rien faire. Alors, le mot en question dépasse de la marge droite, c'est pourquoi L^AT_EX génère l'avertissement suivant :

```
Overfull \hbox (0.64244pt too wide) in paragraph at lines 916--918
```

L'avertissement `Overfull \hbox` signifie que quelque chose dépasse de la marge droite. L^AT_EX signale de combien le texte dépasse (0.64244pt) et à quelles lignes se situe le problème (lines 916--918).

Rassurez-vous, cela n'arrive que très rarement (dans ce manuel, je n'ai eu que 8 avertissements), et quasiment toujours avec des mots contenant une lettre accentuée.

Lorsque ça vous arrive, il est possible de préciser à L^AT_EX où il doit couper le mot. Pour cela, repérez l'endroit approprié grâce à *x_dvi* et placez dans le texte la commande \- où vous voulez que L^AT_EX coupe le mot.

La commande \- agit sur le formatage du reste du paragraphe, puisqu'une partie du mot va se retrouver sur la ligne suivante et décaler la suite du texte. Il se peut donc que de d'autres avertissements de césure apparaissent ou disparaissent. C'est pourquoi je vous conseille d'effectuer vos césures manuelles une par une, en vérifiant à chaque fois le résultat (on peut le faire visuellement avec *x_dvi*, mais il suffit que le message `Overfull \hbox` ait disparu pour être sûr d'avoir bien effectué la césure) et seulement lorsque vous avez fini d'écrire le paragraphe.

Chapitre 5

Les commandes de \LaTeX

LES commandes sont fréquemment utilisées avec \LaTeX . Que ce soit pour produire certains caractères ne figurant pas sur le clavier, pour modifier la typographie, pour sectionner le document, etc.

5.1 Syntaxe d'une commande

Voici la syntaxe d'une commande \LaTeX :

```
 $\backslash$ commande[option]{argument}
```

- une commande commence par un backslash \backslash ;
- celui-ci est suivi par le nom de la commande ;
- viennent ensuite les arguments optionnels, entre crochets ;
- puis les arguments obligatoires, entre accolades.

Certaines commandes ont ce qu'on appelle une *forme étoilée* qui est une variante de la commande. Pour obtenir la forme étoilée d'une commande, il suffit de rajouter une étoile entre le nom de la commande et ses arguments :

```
 $\backslash$ commande*[option]{argument}
```

5.2 Caractères spéciaux

Certains caractères, certains symboles, ne figurent pas sur nos claviers. C'est en particulier le cas pour les lettres accentuées, indispensables en français. Heureusement, \LaTeX met à votre disposition une série de commandes palliant ce manque.

†	<code>\dag</code>	§	<code>\S</code>	©	<code>\copyright</code>
‡	<code>\ddag</code>	¶	<code>\P</code>	£	<code>\pounds</code>

TAB. 5.1 - *Symboles divers*

5.2.1 Symboles

L^AT_EX définit des commandes permettant d'imprimer divers symboles (voir le tableau 5.1) ainsi que des symboles utilisés en français ou dans des langues étrangères (voir le tableau 5.2).

Signalons aussi l'existence de la commande `--` qui imprime un tiret utilisé pour les intervalles de nombres (comme dans 93–94) et de la commande `---` qui imprime un tiret de ponctuation — comme celui-ci.

œ	<code>\oe</code>	à	<code>\aa</code>	ł	<code>\l</code>	¿	<code>?'</code>
Œ	<code>\OE</code>	Å	<code>\AA</code>	Ł	<code>\L</code>	¡	<code>!'</code>
æ	<code>\ae</code>	ø	<code>\o</code>	ß	<code>\ss</code>		
Æ	<code>\AE</code>	Ø	<code>\O</code>				

TAB. 5.2 - *Symboles pour le français et les langues étrangères*

L^AT_EX définit également deux autres commandes permettant d'imprimer les logos T_EX (`\TeX`) et L^AT_EX (`\LaTeX`).

Voyons donc ce que donne le texte suivant :

Essai des commandes `\TeX` et `\LaTeX` dans un texte.

Voici le résultat :

Essai des commandes T_EX et L^AT_EX dans un texte.

Ce n'est pas exactement ce à quoi nous nous attendions. En effet, L^AT_EX ignore tout espace (un ou plusieurs) suivant une commande sans argument (ce qui est le cas des commandes `\TeX` et `\LaTeX`). Il y a trois façons de s'en tirer :

- utiliser la commande `_` (backslash-espace) qui imprime un espace

Essai des commandes `\TeX_` et `\LaTeX_` dans un texte.

- mettre un argument vide

Essai des commandes `\TeX{}` et `\LaTeX{}` dans un texte.

- mettre la commande entre accolades (c'est ce qu'on appelle des *accolades de groupement*, voir page 25)

Essai des commandes `{\TeX}` et `{\LaTeX}` dans un texte.

Ces méthodes sont équivalentes (elles donnent exactement le même résultat) mais la première est plus élégante (mettre un argument vide à une commande qui n'en a pas n'est pas très logique) et donne un texte source plus facile à lire. Quant à la dernière, elle est surtout employée pour les symboles du tableau 5.2. Ainsi, cœur s'écrit `c{\oe}ur`.

5.2.2 Les lettres accentuées

L^AT_EX définit aussi des commandes permettant d'imprimer toutes sortes de lettres accentuées (voir le tableau 5.3).

è	<code>\'e</code>	ñ	<code>\~n</code>	ë	<code>\v{e}</code>	ç	<code>\c{c}</code>
é	<code>\'e</code>	ē	<code>\=e</code>	ë	<code>\H{e}</code>	ç	<code>\d{e}</code>
ê	<code>\^e</code>	è	<code>\.e</code>	êê	<code>\t{ee}</code>	ç	<code>\b{e}</code>
ë	<code>\"e</code>	ë	<code>\u{e}</code>				

TAB. 5.3 - *Accents*

Mais tout ceci est loin d'être pratique. L^AT_EX permet donc d'utiliser, pour les commandes de la forme `\symbole`, une forme simplifiée présentée dans le tableau 5.4.

è	<code>\'e</code>	ñ	<code>\~n</code>
é	<code>\'e</code>	ē	<code>\=e</code>
ê	<code>\^e</code>	è	<code>\.e</code>
ë	<code>\"e</code>		

TAB. 5.4 - *Commandes simplifiées d'accentuation*

L'accentuation de la lettre « i » demande un traitement particulier. En effet, les commandes `\^i` et `\"i` donnent « î » et « ï ». Il faut en fait utiliser les commandes `\^{i}` et `\"i` qui donnent bien « î » et « ï ». La commande `\i` produit un « i » sans point qui peut alors accueillir l'accent en question. Il existe de même une commande `\j`.

Voici un exemple de texte en français utilisant des caractères accentués :

```
Les caract\`eres accentu\`es sont tr\`es utilis\`es en
fran\c{c}ais.
```

Ce n'est tout de même pas très pratique. L'idéal serait de pouvoir taper directement les lettres accentuées au clavier. Mais comme nos claviers sont des claviers QWERTY américains, il n'y a pas de touches prévues pour les accents. Or il y a douze touches de fonction (touches F1 à F12 en haut du clavier) qui ne servent à rien. Elles ont donc été reprogrammées pour permettre la frappe directe des caractères accentués les plus utilisés en français (voir le tableau 5.5).

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12
é	è	ê	ë	à	â	î	ï	ù	û	ô	ç

TAB. 5.5 - *Accents et touches de fonction*

ATTENTION : sur les claviers des stations HP, les touches F9 à F12 sont les quatre touches blanches situées au dessus du pavé numérique.

L'intérêt de ce système est que vous pouvez reprogrammer ces touches de fonction en leur affectant d'autres caractères accentués, ce qui peut être utile pour les texte en espagnol ou en italien (voir l'annexe C).

5.3 Commandes de changement de style et de taille

L^AT_EX dispose de commandes permettant de changer le style des caractères (voir le tableau 5.6) ainsi que leur taille (voir le tableau 5.7).

<code>\rm</code>	roman	<code>\it</code>	<i>italic</i>	<code>\sc</code>	SMALL CAPITALS
<code>\em</code>	<i>emphasized</i>	<code>\sl</code>	<i>slanted</i>	<code>\tt</code>	typewriter
<code>\bf</code>	boldface	<code>\sf</code>	sans serif		

TAB. 5.6 - *Commandes de changement de style*

Quelques remarques concernant ces styles :

- Le style roman est le style normal.
- Le style slanted n'est pas un style italique mais un simple roman penché.
- Le style sans serif est, comme son nom l'indique, sans sérif¹.
- Quant à la différence entre les styles `emphasized` et `italic`, elle est plus que subtile. Vous avez sans doute remarqué que tous deux produisent un texte en italique. La seule différence entre ces deux styles est d'ordre logique. La commande `\em` est à utiliser pour mettre en valeur une partie du texte (dans un texte manuscrit ou tapé à la machine, ceci est fait par un soulignement, mais, lorsqu'on dispose de plusieurs styles, l'usage est de lui préférer la mise en italique) alors que la commande `\it` est une simple mise en italique.

Ces commandes agissent sur tout le reste du texte. Il n'est donc pas question de les utiliser ainsi :

1. Les sérifs sont les petits traits qui se trouvent aux extrémités des lettres.

<code>tiny</code>	<code>\tiny</code>	<code>large</code>	<code>\large</code>
<code>scriptsize</code>	<code>\scriptsize</code>	<code>Large</code>	<code>\Large</code>
<code>footnotesize</code>	<code>\footnotesize</code>	<code>LARGE</code>	<code>\LARGE</code>
<code>small</code>	<code>\small</code>	<code>huge</code>	<code>\huge</code>
<code>normalsize</code>	<code>\normalsize</code>	<code>Huge</code>	<code>\Huge</code>

TAB. 5.7 - *Commandes de changement de taille*

Je mets la suite du texte en `\it` italique. Le reste du document est en italique.

Je mets la suite du texte en *italique*. Le reste du document est en italique.

On pourrait insérer une commande `\rm` comme ceci :

Je mets le mot suivant en `\it` italique`\rm`. Le reste du document est normal.

Je mets le mot suivant en *italique*. Le reste du document est normal.

Mais on peut faire plus élégant en utilisant des *accolades de groupement* pour limiter la portée des commandes :

Je mets le mot suivant en `{\it` italique`}`. Le reste du document est normal.

Je mets le mot suivant en *italique*. Le reste du document est normal.

On peut imbriquer plusieurs niveaux d'accolades :

Voici un texte `{\Large` grand `{\it` et en italique`}`.

Voici un texte **grand** *et en italique*.

Attention, ces accolades n'ont rien à voir avec les accolades qui entourent les arguments d'une commande. En particulier, il n'est pas question d'écrire ceci :

Je mets le mot suivant en `\it{italique}`. Et je me suis bien trompé.

Je mets le mot suivant en *italique*. Et je me suis bien trompé.

Dans certains cas, avec les styles inclinés (`\em`, `\it` et `\sl`), lors d'un retour au style roman, il peut être utile d'utiliser la commande `\/`, qui ajoute un petit espace afin d'aérer le texte :

`{\it` c'est`}` pour aérer.

`{\it` c'est`\/}` pour aérer.

c'est pour aérer.

c'est pour aérer.

5.4 Commandes de saut de ligne, de saut de page et d'espacement

Quelques commandes plus ou moins utiles permettent de forcer la main à L^AT_EX.

La commande `\newline` permet, comme ici, de forcer un saut de ligne. La commande `\\` est une abréviation pour `\newline` que vous rencontrerez souvent.

La commande `\newpage` permet de débiter une nouvelle page (suite page suivante).

Les commandes `\hspace{taille}` et `\vspace{taille}` insèrent un espace (horizontal pour `\hspace` et vertical pour `\vspace`) de taille *taille*.

Les tailles peuvent être exprimées dans l'une des unités suivantes :

cm centimètre ;

mm millimètre ;

in pouce (inch), un pouce valant 2,54 cm ;

pt point (1/72^e de pouce), c'est une unité typographique ;

pc pica (1 pica = 12 points), une autre unité typographique ;

em 1 em est égal à la largeur de la lettre « M » dans la fonte courante, unité propre à `LATEX` ;

ex 1 ex est égal à la hauteur de la lettre « x » dans la fonte courante, unité propre à `LATEX` ;

La commande `\hspace` n'est quasiment jamais utilisée, contrairement à la commande `\vspace` qui sert le plus souvent à laisser un espace dans un texte ou dans une figure, pour pouvoir y coller un dessin ou un schéma. Ainsi, l'espace de 3 centimètres de la figure 5.1 a été obtenu grâce à la commande :

```
\vspace{3cm}
```

FIG. 5.1 - *Figure vide*

Enfin, les commandes `\bigskip`, `\medskip` et `\smallskip` insèrent un espace vertical plus ou moins grand. Elles sont assez pratique car la taille de l'espace qu'elles génèrent est fonction de la fonte utilisée.

5.5 Commandes de sectionnement du document

Sectionner un document (l'organiser en parties, chapitres, sections, sous-sections...) est l'une des choses les plus faciles à faire avec `LATEX`. Pour cela, on utilise les commandes du tableau 5.8.

<code>\part{nom}</code>	<code>\subsection{nom}</code>
<code>\chapter{nom}</code>	<code>\subsubsection{nom}</code>
<code>\section{nom}</code>	

TAB. 5.8 - *Commandes de sectionnement*

La commande `\chapter` n'est pas disponible dans le style `article`. C'est pourquoi on utilise le style `report` lorsqu'on a besoin de faire des chapitres et le style `article` sinon.

Ces commandes sont simples et puissantes. Ainsi, le titre de cette section a été obtenu ainsi :

```
\section{Commandes de sectionnement du document}
```

Ce qui est très intéressant, c'est que c'est L^AT_EX qui gère tout seul la numérotation. On peut insérer des chapitres, des sections ou modifier leur ordre sans se soucier du décalage des numéros.

Avec la même simplicité, la commande `\tableofcontents` imprime la table des matières. Il faut deux compilations successives pour avoir une table des matières à jour. La première compilation enregistre la table des matières dans un fichier `.toc`² et la seconde l'inclut dans le document.

Il existe également une commande `\listoffigures`, qui imprime la liste des figures, et une commande `\listoftables`, qui imprime la liste des tableaux (voir section 6.9). Pour que ces listes soient à jour, il faut aussi deux compilations successives (mettant en jeu les fichiers `.lof` et `.lot`).

Les commandes de sectionnement ont une forme étoilée qui ne numérote pas les titres, ne modifie pas la numérotation des autres sections et ne produit pas d'entrée dans la table des matières. Elles sont principalement utilisées à la place des commandes normales dans les documents très courts où la numérotation n'est pas nécessaire.

Enfin, la commande `\appendix` permet de signaler à L^AT_EX que les chapitres (pour les styles `report` et `book`) ou les sections (pour le style `article`) qui suivront doivent être considérées comme des annexes. Les annexes sont numérotées A, B...

5.6 La 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 a plusieurs auteurs, leurs noms doivent être séparés par `\and`. On peut utiliser `\\` pour passer à la ligne.

². toc pour Table Of Contents.

`\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 est mise automatiquement.

Si l'on ne veut pas de la date, il suffit d'écrire :

```
\date{\ }
```

Pour imprimer la page de garde, il suffit d'appeler la commande `\maketitle` juste après le `\begin{document}`.

Voici un exemple de document avec une page de garde :

```
\documentstyle[11pt,A4,french]{report}

\title{Ceci est le titre \ et la suite}
\author{Babafou \and Babasse}
\date{26 février 1994}

\begin{document}

\maketitle

...
```

On peut annoter le titre, l'auteur ou la date grâce à la commande `\thanks`, qui insère une note de bas de page :

```
\title{Titre avec note\thanks{voici la note}}
```

Pour les styles `report` et `book`, la page de garde est imprimée seule au milieu d'une page alors que pour le style `article`, elle est imprimée en haut de la première page. Pour avoir la page de garde toute seule, il faut inclure l'option de style `titlepage`.

Cette page de garde est, il est vrai, assez spartiate. Mais il est possible de la mettre en page soi-même grâce à l'environnement `titlepage` (voir section 8.3).

5.7 Commandes d'annotation du document

On peut très facilement annoter un document en utilisant des notes de bas de page ou des notes dans la marge.

5.7.1 Les notes de bas de page

Une note de bas de page³ s'obtient très facilement grâce à la commande

```
\footnote{note}
```

\LaTeX s'occupe de la numérotation :

```
Une note de bas de page\footnote{Comme celle-ci.} s'obtient ...
```

3. Comme celle-ci.

5.7.2 Les notes dans la marge

note dans
la marge

Une note dans la marge s'obtient aussi facilement grâce à la commande

```
marginpar[texte de gauche]{texte de droite}
```

Une note dans la marge `\marginpar{note dans la marge}` s'obtient ...

Si la note est située dans la marge gauche et que l'argument optionnel *texte de gauche* est précisé, celui-ci est utilisé. Dans les autres cas, *texte de droite* est utilisé.

5.8 Les références croisées

Avec L^AT_EX, il est très aisé de faire référence à une partie du document (page, chapitre, tableau, équation...).

La commande `\label{nom}` permet de placer un repère, invisible à l'impression, sur la partie du document à laquelle on veut se référer.

La commande `\ref{nom}` permet de faire référence au numéro de la partie de texte repérée par *nom*. Bien entendu, il doit s'agir de quelque chose de numéroté comme un chapitre, une section, un tableau, une équation...

La commande `\pageref{nom}` permet de faire référence à la page où se trouve le repère *nom*.

Ainsi, j'ai placé un repère sur cette section de la façon suivante :

```
\section{Les références croisées} \label{sec:references}
```

Je peux alors faire référence au numéro de cette section (5.8, n'est-ce pas ?) par la commande :

```
\ref{sec:references}
```

et à la page où elle débute (30, je crois...) par :

```
\pageref{sec:references}
```

La séquence précédente a été réalisée sans trucages !

L'argument de la commande `\label` est une combinaison de lettres, de chiffres et de caractères de ponctuation. Personnellement, j'utilise une chaîne de la forme *type:nom* où *type* représente le type de la chose référencée (chp pour un chapitre, sec pour une section, tab pour un tableau, fig pour une figure...). Ainsi, le texte avec lequel j'ai repéré cette section est `sec:references` parce que c'est une section parlant des références croisées. C'est, à mon avis, clair et pratique.

Chapitre 6

Les environnements de L^AT_EX

LES ENVIRONNEMENTS sont des structures très puissantes et très utilisées en L^AT_EX. Un environnement est délimité par les commandes

```
\begin{environnement}
```

et

```
\end{environnement}
```

Tout comme les commandes, les environnements peuvent avoir des arguments, auquel cas la commande de début d'environnement à la forme suivante :

```
\begin{environnement}[option]{argument}
```

Comme les commandes, certains environnements ont une *forme étoilée*, qui est une variante de l'environnement :

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

Tout changement de style ou de taille à l'intérieur d'un environnement ne se propage pas à la sortie de cet environnement. Ainsi :

```
\begin{quote}
Je passe la suite du texte en \it italique. Blabla...
\end{quote}
Mais le texte est normal hors de l'environnement.
```

donne :

```
Je passe la suite du texte en italique. Blabla...
Mais le texte est normal hors de l'environnement.
```

Vous utilisez déjà depuis un moment l'environnement `document`, mais il en existe beaucoup d'autres...

6.1 L'environnement `em`

L'environnement `em` met son contenu en style `\em`. Cela permet de mettre en relief un ou plusieurs paragraphes sans avoir à utiliser d'accolades de groupement :

```
\begin{em}
Le texte contenu dans cet environnement est mis en relief.
\end{em}
```

Ce qui donne :

Le texte contenu dans cet environnement est mis en relief.

6.2 Les environnements `quote` et `quotation`

Ces deux environnements ont un comportement voisin. Ils augmentent tous deux les marges gauche et droite du texte qu'ils contiennent. Leurs différences se situent au niveau de l'alinéa et de l'espacement des paragraphes :

- l'environnement `quotation` met un alinéa en début de paragraphe et espace les paragraphes normalement;
- l'environnement `quote` ne met pas d'alinéa en début de paragraphe et espace les paragraphes davantage.

```
\begin{quotation}
Voici un exemple de texte sur deux paragraphes, afin de montrer le
comportement des environnements de citation.
```

```
Voici un exemple de texte sur deux paragraphes, afin de montrer le
comportement des environnements de citation.
```

```
\end{quotation}
```

donne :

Voici un exemple de texte sur deux paragraphes, afin de montrer le comportement des environnements de citation.

Voici un exemple de texte sur deux paragraphes, afin de montrer le comportement des environnements de citation.

Alors que :

```
\begin{quote}
Voici un exemple de texte sur deux paragraphes, afin de montrer
le comportement des environnements de citation.
```

```
Voici un exemple de texte sur deux paragraphes, afin de montrer
le comportement des environnements de citation.
```

```
\end{quote}
```

donne :

Voici un exemple de texte sur deux paragraphes, afin de montrer le comportement des environnements de citation.

Voici un exemple de texte sur deux paragraphes, afin de montrer le comportement des environnements de citation.

6.3 L'environnement verse

L^AT_EX met l'environnement `verse` à la disposition des poètes :

- les marges sont augmentées comme avec l'environnement `quote` ;
- les lignes doivent être terminées par `\\`, sauf la dernière de chaque strophe ;
- les strophes sont séparées par des lignes blanches.

Si L^AT_EX avait existé à l'époque de RONSARD :

```
\begin{verse}
Comme un chevreuil, quand le printemps détruit \\
L'oiseau cristal de la morne gelée, \\
Pour mieux brouter l'herbette emmiellée \\
Hors de son bois avec l'aube s'enfuit

Et seul, et sûr, loin de chiens et de bruit, \\
Or sur un mont, or dans une vallée, \\
Or près d'une onde à l'écart recélée, \\
Libre, folâtre où son pied le conduit

De rets ni d'arc sa liberté n'a crainte, \\
Sinon alors que sa vie est atteinte, \\
D'un trait meurtrier empourpré de son sang

Ainsi j'allais sans espoir de dommage, \\
Le jour qu'un {\oe}il sur l'avril de mon âge \\
Tira d'un coup mille traits dans mon flanc
\end{verse}
```

```
Comme un chevreuil, quand le printemps détruit
L'oiseau cristal de la morne gelée,
Pour mieux brouter l'herbette emmiellée
Hors de son bois avec l'aube s'enfuit

Et seul, et sûr, loin de chiens et de bruit,
Or sur un mont, or dans une vallée,
```

Or près d'une onde à l'écart recélée,
 Libre, folâtre où son pied le conduit
 De rets ni d'arc sa liberté n'a crainte,
 Sinon alors que sa vie est atteinte,
 D'un trait meurtrier empourpré de son sang
 Ainsi j'allais sans espoir de dommage,
 Le jour qu'un œil sur l'avril de mon âge
 Tira d'un coup mille traits dans mon flanc

6.4 Les environnements center, flushleft et flushright

Ces environnements permettent de centrer du texte ou de ne le justifier que d'un côté. On peut forcer un retour à la ligne grâce à la commande `\\`.

```
\begin{center}
Voici un paragraphe centré. \LaTeX se débrouille tout seul
pour faire la mise en page et couper les lignes où il faut.

Voici deux lignes \\
centrées.
\end{center}
```

Voici un paragraphe centré. L^AT_EX se débrouille tout seul pour faire la mise en page et couper les lignes où il faut.

Voici deux lignes
centrées.

```
\begin{flushleft}
Voici un paragraphe justifié à gauche. \LaTeX se débrouille
tout seul pour faire la mise en page et coupe les lignes où
il faut.

Voici deux lignes \\
justifiées à gauche.
\end{flushleft}
```

Voici un paragraphe justifié à gauche. L^AT_EX se débrouille tout seul pour faire la mise en page et coupe les lignes où il faut.

Voici deux lignes
justifiées à gauche.

```
\begin{flushright}
Voici un paragraphe justifié à droite. \LaTeX se débrouille
tout seul pour faire la mise en page et coupe les lignes où
```

il faut.

```
Voici deux lignes \\
justifiées à droite.
\end{flushright}
```

Voici un paragraphe justifié à droite. $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ se débrouille tout seul pour faire la mise en page et coupe les lignes où il faut.

Voici deux lignes
justifiées à droite.

6.5 L'environnement verbatim

L'environnement `verbatim` est très simple : il ne fait rien ! Il se contente de mettre son contenu en style `\tt`. Aucune interprétation des commandes et des caractères spéciaux n'est faite à l'intérieur de cet environnement.

L'environnement `verbatim` est principalement utilisé pour imprimer des programmes ou du code $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ (tous les exemples de ce manuel ont été réalisés avec l'environnement `verbatim`).

```
\begin{verbatim}
```

Il ne se passe rien dans l'environnement `{\tt verbatim}`. Les commandes `\LaTeX` et les caractères spéciaux (`#$\%&~^{\{}`) ne sont pas interprétés.

```
\end{verbatim}
```

donne :

```
Il ne se passe rien dans l'environnement {\tt verbatim}. Les
commandes \LaTeX et les caractères spéciaux ($\%&~^{\{}) ne
sont pas interprétés.
```

6.6 Les environnements de liste : `itemize`, `enumerate` et `description`

Les environnements de liste sont très souvent employés par les utilisateurs de $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. Ils permettent :

- d'aérer le texte ;
- d'améliorer sa lisibilité.

La liste ci-dessus a été obtenue grâce à l’environnement `itemize` ainsi :

```
\begin{itemize}
  \item d'aérer le texte ;
  \item d'améliorer sa lisibilité.
\end{itemize}
```

Chaque élément de la liste doit commencer par la commande `\item`. Remarquez comme j’ai indenté les éléments à l’intérieur de l’environnement `itemize`. Cela rend le texte plus lisible.

Ne soyez pas surpris si vous utilisez l’environnement `itemize` sans l’option de style `french`. Dans ce cas, chaque élément de la liste débutera par une puce (•) au lieu d’un tiret (-), ce qui est l’usage en Anglais.

L’environnement `enumerate` s’utilise comme l’environnement `itemize`, mais il numérote ses éléments :

```
\begin{enumerate}
  \item premier élément ;
  \item second élément ;
  \item troisième élément.
\end{enumerate}
```

donne :

1. premier élément ;
2. second élément ;
3. troisième élément.

L’environnement `description` permet de choisir soi-même le texte figurant au début de chaque élément de la liste au moyen de l’argument optionnel de la commande `\item` :

```
\begin{description}
  \item[itemize]    pour faire des listes simples ;
  \item[enumerate] pour faire des listes numérotées ;
  \item[description] pour faire des listes dont chaque
                    élément débute par le texte de son
                    choix.
\end{description}
```

donne :

itemize pour faire des listes simples ;

enumerate pour faire des listes numérotées ;

description pour faire des listes dont chaque élément débute par le texte de son choix.

Remarquez que le texte de l'argument optionnel de `\item` a été mis en gras.

On peut imbriquer les listes :

```

Je vous conseille d'utiliser le plus possible les
environnements de liste car :
\begin{enumerate}
  \item ils sont simples à utiliser ;
  \item ils rendent le document plus clair :
    \begin{itemize}
      \item en l'aérant ;
      \item en facilitant sa lecture.
    \end{itemize}
  \item ils structurent les idées.
\end{enumerate}

```

donne :

Je vous conseille d'utiliser le plus possible les environnements de liste car :

1. ils sont simples à utiliser ;
2. ils rendent le document plus clair :
 - en l'aérant ;
 - en facilitant sa lecture.
3. ils structurent les idées.

Un dernier mot sur les usages français en matière de listes :

- pour les listes simples (dont les éléments ne contiennent qu'une phrase) chaque élément doit se terminer par un point-virgule, sauf le dernier, qui doit se terminer comme un point ;
- pour les listes dont les éléments contiennent au moins deux phrases, chaque élément doit se terminer par un point.

6.7 L'environnement tabbing

L'environnement `tabbing` permet d'utiliser des *tabulations* comme on le ferait avec une machine à écrire. Son utilisation est assez bizarre et peu pratique, aussi je vous recommande, dans la mesure du possible, de lui préférer l'environnement `tabular` (voir section 6.8).

Voici un exemple simple d'utilisation de l'environnement `tabbing` :

```

\begin{tabbing}
Troisième ligne \= super bof \= pas génial \kill
Première ligne  \> bof          \> pas génial \\

```

```

Deuxième ligne \> super bof \> nul      \\
Troisième ligne \> ouais      \> minable
\end{tabbing}

```

ce qui donne :

```

Première ligne bof      pas génial
Deuxième ligne super bof nul
Troisième ligne ouais   minable

```

Tout ceci n'est pas très clair et mérite donc quelques explications :

- les tabulations sont mises en place grâce à la commande `\=` ;
- on se place sur la prochaine tabulation grâce à la commande `\>` ;
- la commande `\kill` permet de ne pas imprimer la ligne sur laquelle elle se trouve ;
- chaque ligne (sauf la dernière et celles se terminant par `\kill`) doit se terminer par `\\`.

Dans notre exemple, on a donc placé les tabulations par rapport aux éléments les plus longs de chaque ligne (dans la première ligne qui n'a pas été imprimée à cause de la commande `\kill`).

On peut redéfinir la position des tabulations au milieu de l'environnement :

```

\begin{tabbing}
bof 1 \= ouais 1 \= ok 1 \\
bof 2 \> ouais 2 \> ok 2 \\
super bof 1 \= reouais 1 \\
super bof 2 \> reouais 2
\end{tabbing}

```

ce qui donne :

```

bof 1 ouais 1 ok 1
bof 2 ouais 2 ok 2
super bof 1 reouais 1
super bof 2 reouais 2

```

Il existe d'autres commandes permettant de jouer avec les tabulations, mais je n'en parlerai pas ici. Si vous voulez vraiment utiliser l'environnement `tabbing`, reportez vous à [1, pages 62 et 179].

6.8 Les tableaux : l'environnement tabular

C'est de loin l'environnement le plus puissant de $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. Avec lui, vous pourrez faire tous les tableaux possibles et imaginables (ou presque).

Commençons par un exemple simple :

```
\begin{tabular}{lcr}
Sparc & SunOS & 4.1.3C \\
HP & HP-UX & 9.01 \\
PC & NetBSD & 0.9
\end{tabular}
```

ce qui donne :

```
Sparc SunOS 4.1.3C
HP HP-UX 9.01
PC NetBSD 0.9
```

On a donné trois arguments à l'environnement `tabular`. Cela signifie que le tableau comporte trois colonnes. Détaillons ces arguments :

`l` (*left*) indique que la première colonne sera alignée à gauche ;

`c` (*center*) indique que la deuxième colonne sera centrée ;

`r` (*right*) indique que la troisième colonne sera alignée à droite.

À l'intérieur de l'environnement `tabular`, les lignes sont terminées par `\\` (sauf la dernière) et les colonnes sont séparées par `&`.

Vous avez remarqué que j'ai placé les `&` les uns en dessous des autres, pour améliorer la lisibilité du tableau ($\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ s'en moque, mais pas vous) et que j'ai aligné les colonnes comme elles doivent être imprimées. Je trouve cela plus propre et plus lisible et je vous conseille d'en faire autant.

Voyons maintenant un tableau un peu plus compliqué :

```
\begin{tabular}{|l|c|r|}
\hline
Sparc & SunOS & 4.1.3C \\
\hline
HP & HP-UX & 9.01 \\
\hline
PC & NetBSD & 0.9 \\
\hline
\end{tabular}
```

Sparc	SunOS	4.1.3C
HP	HP-UX	9.01
PC	NetBSD	0.9

C'est le même tableau que le premier, sauf qu'on a rajouté des lignes. Ceci est très simple à faire :

- les lignes verticales sont obtenues en plaçant des | entre les colonnes dans l'argument de l'environnement `tabular` (ces lignes s'étendent sur toute la hauteur du tableau);
- les lignes horizontales sont obtenues par la commande `\hline`, appelée entre chaque ligne du tableau (remarquez qu'il n'y a pas `\\` après).

Pour bien comprendre comment ça marche, vous pouvez essayer de supprimer ou de rajouter des lignes.

Encore un tableau :

```
\begin{tabular}{|p{3cm}|*{2}{c|}|}
\hline
& Contenance & Quantité \\
\hline
\bf Blanche & 33 cl & 10 \\
\hline
\bf Guinness & 1 pint & 5 \\
\hline
\bf Kro & 33 cl & 0 \\
\hline
\end{tabular}
```

	Contenance	Quantité
Blanche	33 cl	10
Guinness	1 pint	5
Kro	33 cl	0

On découvre deux nouveaux arguments de l'environnement `tabular` :

`p{taille}` qui déclare une colonne de taille *taille* (le texte est aligné à gauche par défaut, mais on peut le centrer en utilisant la commande `\centering` ou l'aligner à droite en utilisant la commande `\raggedright`);

`*{nombre}{format}` qui est équivalent à *nombre* fois la déclaration *format* (ici, `c|c|`), ce qui peut être utile pour améliorer la lisibilité de la définition du format des grands tableaux.

Vous remarquerez qu'on peut laisser une case vide et qu'un changement de style ou de taille dans une case n'affecte pas les autres.

Allez, un dernier petit tableau :

```

\begin{tabular}{|c|c|c|}
\cline{2-2}
\multicolumn{3}{|c|}{tableau} \\
\multicolumn{3}{|c|}{pipo} \\
\hline
centré & centré & à droite \\
\hline
bof & bof & \multicolumn{1}{r|}{bof} \\
\hline
\end{tabular}

```

tableau pipo		
centré	centré	à droite
bof	bof	bof

La commande

```
\cline{i-j}
```

trace une ligne horizontale entre les colonnes i et j du tableau.

Il est possible de modifier localement le format du tableau grâce à la commande

```
\multicolumn{nombre}{format}{texte}
```

qui remplace *nombre* colonnes du tableau par *texte* conformément à *format* (regardez le tableau précédent et vous comprendrez). Cette commande est généralement utilisée pour étendre un élément du tableau sur plusieurs colonnes (c'est le cas pour les deux premières lignes du tableau ci-dessus) ou pour modifier l'alignement d'une case (comme dans la dernière ligne du tableau).

Pour quelques autres petites subtilités concernant l'environnement `tabular`, reportez-vous à [1, pages 182–185].

6.9 Les environnements table et figure

Ces environnements ne sont que des cadres, permettant de faire *flotter*¹ tableaux et figures, de leur donner un titre et de les numéroté.

Il est important de pouvoir faire flotter les tableaux et les figures car L^AT_EX ne les coupe pas en bas de page.

Voici un exemple type d'utilisation de l'environnement `table` (l'environnement `figure` s'utilise exactement de la même façon) :

```

\begin{table}[htbp]
\begin{center}

```

1. faire flotter signifie déplacer s'il le faut

```

\begin{tabular}{|l|l|}
\hline
\tt table & tableaux \\
\hline
\tt figure & dessins \\
\hline
\end{tabular}
\end{center}
\caption{Exemple d'environnement {\tt table}}
\label{tab:table}
\end{table}

```

table	tableaux
figure	dessins

TAB. 6.1 - *Exemple d'environnement table*

Les arguments optionnels de l'environnement `table` sont importants et déterminent l'ordre de préférence du placement :

`h` (*here*) ici, si possible ;

`t` (*top*) en haut d'une page (celle-ci ou la suivante) ;

`b` (*bottom*) en bas d'une page (celle-ci ou la suivante) ;

`p` (*page of floats*) à la fin fin du document.

L'ordre de préférence par défaut est `[tbp]`, c'est pourquoi il est préférable de préciser `[htbp]` si on veut que le tableau soit placé, dans la mesure du possible, conformément à sa position dans le texte.

Vous remarquerez que l'on a utilisé l'environnement `center` pour centrer le tableau ainsi que deux commandes :

- la commande `\caption` permet de donner un titre au tableau ou à la figure ;
- la commande `\label`, placée obligatoirement après la commande `\caption`, permet de lui faire référence grâce aux commandes `\ref` ou `\pageref`.

Chapitre 7

L^AT_EX mathématique

L'UN des atouts de L^AT_EX est certainement la facilité avec laquelle il permet de réaliser les équations les plus complexes.

7.1 Les modes mathématiques

L^AT_EX connaît un très grand nombre d'opérateurs et de relations et sait comment les disposer. Mais il faut pouvoir lui indiquer que l'on va taper un texte mathématique. Pour cela, L^AT_EX dispose de deux modes mathématiques :

- l'un est utilisé pour les formules devant apparaître dans le corps du texte, on y rentre grâce à l'environnement `math` ;
- l'autre est utilisé pour les formules que l'on désire voir apparaître seules sur une ligne, on y rentre grâce à l'environnement `displaymath`.

7.1.1 L'environnement `math`

C'est le mode mathématique permettant de taper des formules dans le corps du texte :

Considérons l'équation `\begin{math} x + y + z = n \end{math}`.

ce qui donne :

Considérons l'équation $x + y + z = n$.

Si vous faites grand usage de formules mathématiques, la structure d'environnement s'avère lourde et diminue la lisibilité du texte source. C'est pourquoi il existe deux formes plus simples, `\(... \)` et `$... $`. L'exemple précédent peut donc s'écrire plus simplement :

Considérons l'équation `\(x + y + z = n \)`.

ou :

Considérons l'équation `$ x + y + z = n $`.

On utilise généralement la forme $\$ \dots \$$.

Remarquez que l'espace n'a aucune importance en mode mathématique. Ainsi, $\$x+y+z=n\$$ produira le même résultat que $\$ x + y + z = n \$$.

7.1.2 L'environnement `displaymath`

C'est le mode mathématique permettant de taper des formules seules sur une ligne :

Ce qui conduit à

```
\begin{displaymath}
x + y + z = n
\end{displaymath}
```

Ce qui conduit à

$$x + y + z = n$$

Comme pour l'environnement `math`, il existe des formes plus simples, `\[\dots \]` et `$$ \dots $$`. On aurait donc pu écrire :

Ce qui conduit à

```
\[
x + y + z = n
\]
```

ou :

Ce qui conduit à

```
$$
x + y + z = n
$$
```

ATTENTION : ne laissez surtout pas de ligne blanche à l'intérieur de l'environnement `displaymath`, cela provoquerait une erreur.

L'environnement `displaymath` centre les formules. Si vous préférez qu'elles soient alignées à gauche, vous pouvez utiliser l'option de style `fleqn`.

7.2 Structures couramment utilisées en mathématiques

7.2.1 Indices et exposants

Avec L^AT_EX, on rencontre les indices et les exposants un peu partout. Par exemple, les bornes d'une somme ou d'une intégrale sont tapées comme des indices et des exposants.

Un indice s'obtient grâce au caractère `_` et un exposant s'obtient grâce à `^` :

$$\$ x^2 + y^2 = 1 \$, \quad \$ x_1 = x_2 \$, \quad \$ x_1^2 = x^2_1 \$$$

$$x^2 + y^2 = 1, x_1 = x_2, x_1^2 = x_1^2$$

Si l'indice ou l'exposant fait plus d'un caractère, il faut utiliser des accolades de groupement :

$$\text{\$ } x^{\{2y\}} = z \text{\$, } \text{\$ } a_{\{ij\}} = 0 \text{\$, } \text{\$ } x^{\{y^{\{z\}}\}} = e \text{\$}$$

$$x^{2y} = z, a_{ij} = 0, x^{y^z} = e$$

De même, on peut utiliser des accolades pour indiquer ce à quoi s'applique l'indice ou l'exposant :

$$\text{Voyez-vous une différence entre } (x^2+y^2)^n \text{ et } \{(x^2+y^2)\}^n \text{ ?}$$

$$\text{Voyez-vous une différence entre } (x^2 + y^2)^n \text{ et } (x^2 + y^2)^n \text{ ?}$$

Dans le premier cas, l'exposant s'applique au caractère qui le précède (la parenthèse), il est donc à la même hauteur que les carrés. Mais dans le second cas, à cause des accolades, l'exposant n s'applique à toute l'expression $(x^2 + y^2)$ et est donc placé un peu plus haut, puisque cette expression comporte déjà un niveau d'exposant. C'est à vous de voir quelle forme vous préférez, l'une étant, du point de vue mathématique, aussi valable que l'autre.

7.2.2 Texte à l'intérieur d'une formule

Il peut être utile de mettre quelques mots dans une formule. Mais pour cela il faut ramener L^AT_EX du mode mathématique au mode texte. Ceci est possible grâce à la commande `\mbox` :

$$\text{\$}\text{\$}$$

$$f(x) > 1 \text{\mbox{ si }} x < 3$$

$$\text{\$}\text{\$}$$

$$f(x) > 1 \text{ si } x < 3$$

Remarquez les espaces autour du « si » dans la commande `\mbox`. Ils sont nécessaires parce que l'espacement mathématique de L^AT_EX n'a rien à voir avec son espacement en mode texte. Voyez vous même :

$$\text{\$}\text{\$}$$

$$f(x) > 1 \text{\mbox{si}} x < 3$$

$$\text{\$}\text{\$}$$

$$f(x) > 1 \text{si} x < 3$$

Bien entendu, on n'a pas besoin de la commande `\mbox` dans l'environnement `math` :

$$\text{Et donc } f(x) > 1 \text{\$ si } x < 3 \text{\$}.$$

$$\text{Et donc } f(x) > 1 \text{ si } x < 3.$$

7.3 Lettres, symboles et accents

Il est souvent fait usage, dans les formules mathématiques, de lettres grecques, de divers symboles ou d'accents. L^AT_EX dispose de nombreuses commandes permettant de les imprimer.

Il existe une différence par rapport au mode texte de L^AT_EX. En mode mathématique, il n'est nul besoin d'utiliser la commande `_` pour avoir un espace après un symbole. L^AT_EX gère l'espacement en mode mathématique tout seul.

7.3.1 Lettres grecques

Les commandes permettant d'obtenir les lettres grecques sont présentées dans le tableau 7.1.

Minuscules							
α	<code>\alpha</code>	θ	<code>\theta</code>	π	<code>\pi</code>	ϕ	<code>\phi</code>
β	<code>\beta</code>	ϑ	<code>\vartheta</code>	ϖ	<code>\varpi</code>	φ	<code>\varphi</code>
γ	<code>\gamma</code>	ι	<code>\iota</code>	ρ	<code>\rho</code>	χ	<code>\chi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	ϱ	<code>\varrho</code>	ψ	<code>\psi</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	σ	<code>\sigma</code>	ω	<code>\omega</code>
ε	<code>\varepsilon</code>	μ	<code>\mu</code>	ς	<code>\varsigma</code>		
ζ	<code>\zeta</code>	ν	<code>\nu</code>	τ	<code>\tau</code>		
η	<code>\eta</code>	ξ	<code>\xi</code>	υ	<code>\upsilon</code>		
Majuscules							
Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>	Ψ	<code>\Psi</code>
Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Φ	<code>\Phi</code>		

TAB. 7.1 - *Lettres grecques*

Les lettres grecques minuscules sont obtenues par un `\` suivi du nom de la lettre. Pour les majuscules, il suffit de mettre la première lettre du nom en majuscule.

Pour certaines lettres minuscules, il existe une deuxième forme dont la commande commence par `var`.

Notez que ces commandes (ainsi que toutes celles qui sont présentées dans ce chapitre) ne sont disponibles qu'en mode mathématique :

`\pi` vaut approximativement 3,1416.

π vaut approximativement 3,1416.

7.3.2 Majuscules calligraphiques (ou anglaises)

La commande `\cal` permet d'obtenir des majuscules calligraphiques et s'utilise comme les commandes de changement de style ou de taille :

Soit `\cal C` le cercle de centre 0 et de rayon 1,
`{\cal D}_1` et `{\cal D}_2` deux droites.

Soit \mathcal{C} le cercle de centre O et de rayon 1, \mathcal{D}_1 et \mathcal{D}_2 deux droites.

Voici ce à quoi ressemblent ces majuscules :

ABCDEFGHIJKLMN OPQRSTUVWXYZ

7.3.3 Symboles d'opérateurs binaires

Les commandes présentées dans le tableau 7.2 permettent d'obtenir des symboles d'opérateurs binaires.

\pm	<code>\pm</code>	\cap	<code>\cap</code>	\diamond	<code>\diamond</code>	\oplus	<code>\oplus</code>
\mp	<code>\mp</code>	\cup	<code>\cup</code>	\triangleup	<code>\bigtriangleup</code>	\ominus	<code>\ominus</code>
\times	<code>\times</code>	\uplus	<code>\uplus</code>	∇	<code>\bigtriangledown</code>	\otimes	<code>\otimes</code>
\div	<code>\div</code>	\sqcap	<code>\sqcap</code>	\triangleleft	<code>\triangleleft</code>	\oslash	<code>\oslash</code>
$*$	<code>\ast</code>	\sqcup	<code>\sqcup</code>	\triangleright	<code>\triangleright</code>	\odot	<code>\odot</code>
\star	<code>\star</code>	\vee	<code>\vee</code>	\triangleleft	<code>\lhd</code>	\bigcirc	<code>\bigcirc</code>
\circ	<code>\circ</code>	\wedge	<code>\wedge</code>	\triangleright	<code>\rhd</code>	\dagger	<code>\dagger</code>
\bullet	<code>\bullet</code>	\setminus	<code>\setminus</code>	\triangleleft	<code>\unlhd</code>	\ddagger	<code>\ddagger</code>
\cdot	<code>\cdot</code>	\wr	<code>\wr</code>	\triangleleft	<code>\unrhd</code>	\amalg	<code>\amalg</code>

TAB. 7.2 - *Symboles d'opérateurs binaires*

7.3.4 Symboles de relations binaires

Les commandes présentées dans le tableau 7.3 permettent d'obtenir des symboles de relations binaires.

On peut obtenir la négation de ces opérateurs grâce à la commande `\not` :

`x \not\in {\cal F}`

$x \notin \mathcal{F}$

7.3.5 Flèches

Les commandes présentées dans le tableau 7.4 permettent d'obtenir des flèches.

\leq	<code>\leq</code>	\geq	<code>\geq</code>	\equiv	<code>\equiv</code>	\models	<code>\models</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\sim	<code>\sim</code>	\perp	<code>\perp</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\simeq	<code>\simeq</code>	$ $	<code>\mid</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\asymp	<code>\asymp</code>	\parallel	<code>\parallel</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>	\bowtie	<code>\bowtie</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>	\Join	<code>\Join</code>
\sqsubset	<code>\sqsubset</code>	\sqsupset	<code>\sqsupset</code>	\neq	<code>\neq</code>	\smile	<code>\smile</code>
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\doteq	<code>\doteq</code>	\frown	<code>\frown</code>
\in	<code>\in</code>	\ni	<code>\ni</code>	\propto	<code>\propto</code>		
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>				

TAB. 7.3 - *Symboles de relations binaires*

\leftarrow	<code>\leftarrow</code>	\longleftarrow	<code>\longleftarrow</code>	\uparrow	<code>\uparrow</code>
\Leftarrow	<code>\Leftarrow</code>	\Lleftarrow	<code>\Lleftarrow</code>	\Uparrow	<code>\Uparrow</code>
\rightarrow	<code>\rightarrow</code>	\longrightarrow	<code>\longrightarrow</code>	\downarrow	<code>\downarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>	\Downarrow	<code>\Downarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>	\updownarrow	<code>\updownarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Llongleftrightarrow	<code>\Llongleftrightarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>	\nearrow	<code>\nearrow</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookleftarrow	<code>\hookleftarrow</code>	\searrow	<code>\searrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>	\swarrow	<code>\swarrow</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoondown	<code>\rightharpoondown</code>	\nwarrow	<code>\nwarrow</code>
\rightrightarrows	<code>\rightrightarrows</code>	\leadsto	<code>\leadsto</code>		

TAB. 7.4 - *Flèches*

7.3.6 Symboles divers

Les commandes présentées dans le tableau 7.5 permettent d'obtenir des symboles divers.

\aleph	<code>\aleph</code>	$'$	<code>\prime</code>	\forall	<code>\forall</code>	∞	<code>\infty</code>
\hbar	<code>\hbar</code>	\emptyset	<code>\emptyset</code>	\exists	<code>\exists</code>	\square	<code>\Box</code>
i	<code>\imath</code>	∇	<code>\nabla</code>	\neg	<code>\neg</code>	\diamond	<code>\Diamond</code>
j	<code>\jmath</code>	\surd	<code>\surd</code>	\flat	<code>\flat</code>	\triangle	<code>\triangle</code>
ℓ	<code>\ell</code>	\top	<code>\top</code>	\natural	<code>\natural</code>	\clubsuit	<code>\clubsuit</code>
\wp	<code>\wp</code>	\perp	<code>\perp</code>	\sharp	<code>\sharp</code>	\diamondsuit	<code>\diamondsuit</code>
\Re	<code>\Re</code>	$\ $	<code>\ </code>	\backslash	<code>\backslash</code>	\heartsuit	<code>\heartsuit</code>
\Im	<code>\Im</code>	\angle	<code>\angle</code>	∂	<code>\partial</code>	\spadesuit	<code>\spadesuit</code>
\mho	<code>\mho</code>						

TAB. 7.5 - *Symboles divers*

De plus, les commandes `\ldots` et `\cdots` permettent d'obtenir des points de suspension (alignés à gauche ou centrés).

7.3.7 Accents

Le tableau 7.6 présente les accents disponibles en mode mathématique.

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

TAB. 7.6 - *Accents*

Pour placer un accent sur un i ou un j , utilisez les commandes `\imath` et `\jmath` qui produisent des lettres sans point.

Il existe aussi des commandes permettant d'avoir des versions longues de certains accents :

`\widehat{xyz}` \$, `\widetilde{abc}` \$

\widehat{xyz} , \widetilde{abc}

La commande `\overrightarrow` permet d'avoir une grande flèche :

`\overrightarrow{AB}` \$

\overrightarrow{AB}

Les commandes `\overline` et `\underline` permettent de surligner et de souligner :

`$ \overline{z+1} = \bar{z} + 1 $`, `$ \underline{\alpha + \beta} $`

$$\overline{z+1} = \bar{z} + 1, \underline{\alpha + \beta}$$

Enfin, les commandes `\overbrace` et `\underbrace` placent des accolades sur ou sous le texte. On peut ajouter un commentaire en mettant un exposant après la commande `\overbrace` ou un indice après la commande `\underbrace` :

`$$`
`a^n = \overbrace{a \times a \times \cdots`
`\times a}^n \mbox{ fois}`
`$$`

$$a^n = \overbrace{a \times a \times \cdots \times a}^{n \text{ fois}}$$

7.3.8 Superposer deux symboles

Il est possible de superposer deux symboles grâce à la commande :

`\stackrel{dessus}{dessous}`

`$ x \stackrel{f}{\longmapsto} f(x) $`

$$x \xrightarrow{f} f(x)$$

On peut utiliser la commande `\stackrel` pour obtenir des tenseurs :

`$ \stackrel{\rightarrow}{\Lambda} (\stackrel{\rightarrow}{\varepsilon}) $`

$$\vec{\sigma} = \Lambda(\vec{\varepsilon})$$

7.4 Modifier l'espacement

Dans certains cas, il peut être utile de modifier l'espacement, afin d'éloigner ou de rapprocher certains symboles.

Nous verrons, lorsque cela est utile, où ces commandes peuvent servir.

<code>\,</code>	petit espace	<code>\!</code>	petit espace négatif
<code>\:</code>	moyen espace	<code>\quad</code>	petite espace interformules
<code>\;</code>	grand espace	<code>\quad\quad</code>	moyen espace interformules

TAB. 7.7 - *Commandes d'espacement*

7.5 Fractions

Les fractions les plus complexes s'obtiennent simplement grâce à la commande :

```
\frac{numérateur}{dénominateur}
```

```
$$
\frac{x+1}{x+2}
$$
```

donne :

$$\frac{x + 1}{x + 2}$$

7.6 Racines

Les racines s'obtiennent grâce à la commande :

```
\sqrt[nième]{argument}
$ \sqrt{x^2+y^2} $, $ \sqrt[n]{\alpha_i} $
```

donne :

$$\sqrt{x^2 + y^2}, \sqrt[n]{\alpha_i}$$

7.7 Délimiteurs

Les *délimiteurs*, présentés dans le tableau 7.8, sont des symboles permettant d'encadrer des formules. Leur hauteur s'adapte à celle de la formule.

Les délimiteurs doivent toujours aller par deux : un délimiteur ouvrant et un délimiteur fermant. Un délimiteur ouvrant s'obtient par la commande `\left` suivie du délimiteur, et un délimiteur fermant par la commande `\right` suivie du délimiteur. Bien entendu, on peut imbriquer les paires de délimiteurs.

Si l'on ne désire pas voir apparaître de délimiteur fermant, il faut alors mettre `\right.` (tout `\left` doit avoir un `\right.`, le `\right.` n'imprime rien, mais sert à fermer le `\left` qui lui correspond).

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

TAB. 7.8 - *Délimiteurs*

```

$$
\left[
\begin{array}{cc}
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{array}
\right]

```

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

```

$$
|x| =
\left\{
\begin{array}{rl}
x & \text{si } x \geq 0 \\
-x & \text{si } x < 0
\end{array}
\right.

```

$$|x| = \begin{cases} x & \text{si } x \geq 0 \\ -x & \text{si } x < 0 \end{cases}$$

7.8 Symboles de taille variable

Les symboles présentés dans le tableau 7.9 sont de *taille variable*: il n'ont pas la même taille dans l'environnement `math` et dans l'environnement `displaymath`.

Ces symboles sont plus petits en environnement `math` afin d'être de même taille que le reste du texte. De plus, leurs indices et exposants ne sont pas disposés de la même façon dans les deux environnements :

$\Sigma \Sigma$	<code>\sum</code>	$\cap \cap$	<code>\bigcap</code>	$\odot \odot$	<code>\bigodot</code>
$\prod \prod$	<code>\prod</code>	$\cup \cup$	<code>\bigcup</code>	$\otimes \otimes$	<code>\bigotimes</code>
$\coprod \coprod$	<code>\coprod</code>	$\sqcup \sqcup$	<code>\bigsqcup</code>	$\oplus \oplus$	<code>\bigoplus</code>
$\int \int$	<code>\int</code>	$\vee \vee$	<code>\bigvee</code>	$\uplus \uplus$	<code>\biguplus</code>
$\oint \oint$	<code>\oint</code>	$\wedge \wedge$	<code>\bigwedge</code>		

TAB. 7.9 - *Symboles de taille variable*

```
$ \sum_{i=0}^{+\infty} x_i \quad \int_a^b f(x) dx $
$$ \sum_{i=0}^{+\infty} x_i \quad \int_a^b f(x) dx $$
```

$$\sum_{i=0}^{+\infty} x_i \quad \int_a^b f(x) dx$$

$$\sum_{i=0}^{+\infty} x_i \quad \int_a^b f(x) dx$$

On peut, en environnement `math`, utiliser la commande `\displaystyle` qui donne aux symboles de taille variable la taille qu'ils ont en environnement `displaymath` (mais ceci modifie l'interligne du texte):

```
On en déduit donc
$ \displaystyle \sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6} $
\frac{\pi^2}{6} $.
```

On en déduit donc $\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$.

Signalons aussi l'existence des commandes `\scriptstyle` et `\scriptscriptstyle` qui mettent respectivement le texte à la taille d'un exposant et d'un exposant d'exposant (désolé, il n'y a pas plus petit).

Dans les intégrales, on utilise souvent les commandes d'espacement pour éloigner le dx ou pour rapprocher les symboles d'intégration des intégrales multiples :

```
$$
\int_a^b f(x) \, dx \quad \int \int \int_{\Omega} d\tau
V = \int \int \int_{\Omega} d\tau
$$
```

$$\int_a^b f(x) dx \quad V = \iiint_{\Omega} d\tau$$

7.9 Logarithmes et autres fonctions

Il est d'usage d'écrire certains noms de fonctions ou d'opérateurs en style roman :

$$\lim_{x \rightarrow +\infty} \ln x = +\infty$$

On pourrait écrire cette égalité ainsi :

```
$$
\lim_{x \rightarrow +\infty} \ln x = +\infty
$$
```

$$\lim_{x \rightarrow +\infty} \ln x = +\infty$$

Mais alors les opérateurs `lim` et `ln` ne sont pas en roman. Essayons donc ainsi :

```
$$
\mbox{\lim}_{x \rightarrow +\infty} \mbox{\ln} x = +\infty
$$
```

$$\lim_{x \rightarrow +\infty} \ln x = +\infty$$

C'est mieux, mais on n'a pas le bon espacement, et $x \rightarrow +\infty$ n'est pas placé sous le symbole `lim` comme il devrait l'être.

Pour résoudre ces problèmes, L^AT_EX dispose des commandes présentées dans le tableau 7.10.

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

TAB. 7.10 - *Logarithmes et autres*

La bonne façon d'écrire

$$\lim_{x \rightarrow +\infty} \ln x = +\infty$$

est donc :

```
$$
\lim_{x \rightarrow +\infty} \ln x = +\infty
$$
```

Certaines des commandes du tableau 7.10 se comportent comme des symboles de taille variable. C'est en particulier le cas des limites :

```
$ \lim_{x \rightarrow 0} \ln x = -\infty $
```

$$\lim_{x \rightarrow 0} \ln x = -\infty$$

Enfin, les commandes `\bmod` et `\pmod` permettent d'imprimer les deux formes de l'opérateur *modulo* :

```
$ x = a \bmod b $
$ a \equiv b \pmod{n} $
```

```
x = a mod b
a ≡ b (mod n)
```

7.10 Les tableaux : l'environnement array

Les tableaux (matrices...) s'obtiennent grâce à l'environnement `array`, qui s'utilise comme l'environnement `tabular` :

```
$$
{\cal A} = \left(
\begin{array}{ccc}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{array}
\right)
\right)
$$
```

$$\mathcal{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

L'argument de l'environnement `array` indique le nombre de colonnes du tableau ainsi que leur alignement. Chaque caractère représente une colonne et peut être :

`l` (*left*) la colonne est alignée à gauche ;

`c` (*center*) la colonne est centrée ;

`r` (*right*) la colonne est alignée à droite.

Ainsi, la matrice de notre exemple a trois colonnes (`ccc`), chacune d'elle étant centrée.

À l'intérieur de l'environnement `array`, les lignes sont terminées par `\\` (sauf la dernière) et les colonnes sont séparées par `&`.

Les commandes `\cdots` (`\dots`), `\vdots` (`\dots`) et `\ddots` (`\dots`) sont souvent utilisées dans les matrices :

```
$$
```



```

{\cal A} = \left(
\begin{array}{cccc}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{m1} & a_{m2} & \cdots & a_{mn}
\end{array}
\right)

```

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

L'environnement `array` permet de faire toutes sortes de tableaux en mode mathématique, pas seulement des matrices :

```

$$
\sum_{\begin{array}{cc}
\scriptstyle i>0 \\
\scriptstyle j>0
\end{array}} a_{ij} = \alpha
$$

```

$$\sum_{\substack{i>0 \\ j>0}} a_{ij} = \alpha$$

7.11 Les équations numérotées : l'environnement `equation`

L'environnement `equation` s'utilise comme l'environnement `displaymath` mais il génère en plus un numéro d'équation :

```

\begin{equation} \label{eqn:euler}
e^{i\pi} + 1 = 0
\end{equation}

```

$$e^{i\pi} + 1 = 0 \tag{7.1}$$

Comme tout ce qui est numéroté, on peut référencer une équation par la commande `\label`, puis rappeler son numéro par la commande `\ref`.

Le numéro d'équation apparaît à droite par défaut. On peut le faire apparaître à gauche en précisant l'option de style `leqno`

7.12 Les équations alignées : l'environnement eqnarray

Il peut parfois être nécessaire de superposer des formules ou de dérouler un calcul en gardant un symbole aligné (le signe =, par exemple). Pour cela, on utilise l'environnement `eqnarray` :

```
\begin{eqnarray}
\ln xy & = & \ln x + \ln y \\
\exp(x+y) & = & \exp x \cdot \exp y
\end{eqnarray}
```

$$\ln xy = \ln x + \ln y \quad (7.2)$$

$$\exp(x + y) = \exp x \exp y \quad (7.3)$$

L'environnement `eqnarray` s'utilise comme un tableau dont le format serait `rcl`.

On peut supprimer la numérotation d'une ligne en mettant la commande `\nonumber` en fin de ligne :

```
\begin{eqnarray}
\int_1^2 x^2 dx & = & \left[ \frac{x^3}{3} \right]_1^2 \nonumber \\
& = & \frac{2^3}{3} - \frac{1^3}{3} \nonumber \\
& = & \frac{8}{3} - \frac{1}{3} \nonumber \\
& = & \frac{7}{3}
\end{eqnarray}
```

$$\begin{aligned} \int_1^2 x^2 dx &= \left[\frac{x^3}{3} \right]_1^2 \\ &= \frac{2^3}{3} - \frac{1^3}{3} \\ &= \frac{8}{3} - \frac{1}{3} \\ &= \frac{7}{3} \end{aligned} \quad (7.4)$$

Si l'on ne désire aucune numérotation, il vaut mieux utiliser la forme étoilée de l'environnement `eqnarray`. L'environnement `eqnarray*` s'utilise comme l'environnement `eqnarray`, mais ne numérote pas les équations.

7.13 Théorèmes, définitions...

Théorème 1 (Babafou) L^AT_EX ne se limite pas aux formules et aux signes cabalistiques, il permet aussi de faire des théorèmes de toute beauté (comme celui-ci).

La commande

```
\newtheorem{nom}{texte}
```

définit un nouvel environnement qui a pour nom *nom*.

Par exemple, le théorème de Babafou ci-dessus a été réalisé ainsi :

```
\newtheorem{theoreme}{Théorème}

\begin{theoreme}[Babafou]
\LaTeX\ ne se limite pas aux formules et aux signes cabalistiques,
il permet aussi de faire des théorèmes de toute beauté (comme
celui-ci).
\end{theoreme}
```

L'environnement créé par la commande `\newtheorem` réalise les actions suivantes :

1. il imprime *texte* (deuxième argument de la commande `\newtheorem`);
2. puis un numéro (à chaque environnement créé est associé un compteur);
3. puis, éventuellement, son argument optionnel (c'est généralement l'auteur du théorème);
4. enfin, il imprime son contenu, préalablement mis en italique.

On peut créer plusieurs environnements différents, un pour les théorèmes, un pour les définitions, un pour les lemmes, un pour les hypothèses...

Chapitre 8

LaTeX pro

C^E chapitre explique comment réaliser avec LaTeX des fonctions dites « complexes ». Complexes pour les autres traitements de texte, mais pas pour LaTeX !

8.1 Définir de nouvelles commandes et de nouveaux environnements

subsection Définir de nouvelles commandes

Il est possible de définir ses propres commandes grâce à la commande

```
\newcommand{\nom}[nombre d'arguments]{définition}
```

En gros, toute occurrence de `\nom` sera remplacée par *définition*.

Les définitions de commandes peuvent se trouver dans le préambule ou dans le corps du document.

Prenons un exemple simple :

```
\newcommand{\ba}{babasse}
```

Je peux maintenant utiliser la commande `\ba` qui imprimera « babasse » :

```
La \ba\ c'est chouette !
```

```
La babasse c'est chouette!
```

On peut créer des commandes qui prennent des arguments. Il suffit d'en préciser le nombre dans le paramètre *nombre d'arguments* de `\newcommand` :

```
\newcommand{\itfootnote}[1]{\footnote{\it #1}}
```

La commande `\itfootnote` met une note de bas de page en italique¹. Elle prend un argument (ici, *nombre d'arguments* vaut 1). Dans la définition de la commande, on fait référence à cet argument avec `#1` (s'il y avait un deuxième argument, ce serait `#2`, etc.).

1. Vous voyez, ça marche.

8.1.1 Définir de nouveaux environnements

Il est possible de définir ses propres environnements grâce à la commande

```
\newenvironment{\nom}[nombre d'arguments]{définition de début}{définition de fin}
```

Les définitions d'environnements peuvent se trouver dans le préambule ou dans le corps du document.

Lorsque L^AT_EX rencontrera `\begin{nom}`, il le remplacera par *définition de début* et lorsqu'il rencontrera `\end{nom}`, il le remplacera par *définition de fin* :

```
\newenvironment{italitemize}{\begin{itemize} \it}{\end{itemize}}

\begin{italitemize}
  \item mais oui,
  \item ça marche !
\end{italitemize}

- mais oui,
- ça marche !
```

Généralement, les nouveaux environnements sont créés à partir d'environnements existants (comme ici).

Il est possible de créer des environnements avec argument. On en précise le nombre par *nombre d'arguments* dans `\newenvironment` et on y fait référence dans *définition de début* par #1, #2...

```
\newenvironment{tab}[1]%
{\begin{tabular}{|#1|}\hline}%
{\hline\end{tabular}}

\begin{tab}{cc}
  essai      & de  \\
l'environnement & tab \\
\end{tab}
```

essai	de
l'environnement	tab

Regardez comment j'ai utilisé la commande `\newenvironment`. J'ai placé les définitions de début et de fin du nouvel environnement chacune sur une ligne (c'est plus clair). Mais pour cela, j'ai mis un % à la fin des deux premières lignes. Ce % indique un commentaire. Pourtant, je n'ai mis aucun texte après. À quoi sert-il ? Il sert à empêcher L^AT_EX de considérer le retour à la ligne comme un espace. De cette façon, je peux mettre la définition de mon nouvel environnement sur plusieurs lignes. Si je n'avais pas mis de %, L^AT_EX aurait vu des espaces entre les trois derniers arguments de `\newenvironment`, ce qui, pour lui, est une erreur de syntaxe.

8.2 Inclure des dessins avec *xfig*

Le programme de dessin vectoriel *xfig* permet de réaliser toute sorte de dessins, figures et autres schémas. Il s'appelle en cliquant sur le deuxième bouton de *xlatex*.

Je ne détaillerai pas ici l'utilisation de *xfig* (ce serait bien trop long et puis, *xfig* est si simple d'emploi...), je me contenterai de vous indiquer comment faire pour inclure un dessin dans un document L^AT_EX.

8.2.1 Sauvegarde d'un dessin

Lorsque vous avez terminé votre dessin, sauvez-le au format *.fig*, pour pouvoir le modifier si besoin est :

- cliquez sur le bouton File..., pour faire apparaître le menu de sauvegarde ;
- donnez un nom à votre fichier (par exemple *dessin.fig*) en le tapant dans la case Filename: ;
- puis cliquez sur le bouton Save.

Mais, pour pouvoir l'inclure dans votre document, il faut que le dessin soit au format *PostScript encapsulé* (c'est une variante du PostScript des imprimantes). Pour l'exporter à ce format :

- cliquez sur le bouton Export..., pour faire apparaître le menu d'exportation ;
- cliquez sur Landscape, gardez le bouton de la souris enfoncé et descendez sur Portrait, puis relâchez-le (ceci oriente le dessin dans le bon sens) ;
- enfin, cliquez sur Export.

Le dessin est alors enregistré en PostScript encapsulé avec l'extension *.eps*.

8.2.2 Inclure un dessin dans un document L^AT_EX

Afin d'inclure un fichier au format PostScript encapsulé dans un document L^AT_EX, il faut tout d'abord charger l'option de style `epsf`.

Pour afficher votre dessin, utilisez la commande

```
\epsfbox{dessin.eps}
```

Il est préférable de mettre le dessin dans une figure, cela permet de lui donner une légende et un numéro et surtout de pouvoir le déplacer s'il risque d'être coupé en bas de page.

Il serait bon également de le centrer. Mais n'essayez pas d'utiliser l'environnement `center`, ça ne marche pas. On peut s'en tirer en mettant le dessin dans un environnement `displaymath`, lorsque l'option de style `fleqn` n'est pas chargée.

On peut aussi modifier la taille du dessin grâce aux commandes

```
\epsfxsize=taille
```

et

```
\epsfysize=taille
```

Il vaut mieux ne pas utiliser les deux en même temps si l'on veut conserver les proportions. Pour ce qui est de la largeur du dessin (`\epsfxsize`), 14 cm est un maximum.

Voici donc le code type d'inclusion d'un dessin au format PostScript encapsulé avec L^AT_EX :

```
\documentstyle[11pt,A4,french,epsf]{report}

% ...

\begin{figure}[htbp]
\epsfxsize=14cm
$$
\epsfbox{dessin.eps}
$$
\caption{Dessin}
\label{fig:dessin}
\end{figure}
```

Le programme *xdvi* ne permet pas d'admirer ces dessins en PostScript encapsulé. Si vous voulez les voir, utilisez plutôt *xdvik* (mais il est beaucoup plus lent que *xdvi*).

8.3 Faire soi-même sa page de garde : l'environnement `titlepage`

Vous avez dû remarquer que la commande `\maketitle` génère une page de garde minimale. C'est dommage qu'un traitement de texte de la qualité de L^AT_EX doive se contenter de pages de garde aussi sobres.

Heureusement, il est possible de faire sa page de garde soi-même grâce à l'environnement `titlepage`. À l'intérieur, vous êtes totalement maître de la mise en page. Généralement, on utilise :

- l'environnement `center`, pour centrer le texte ;
- les commandes de changement de taille (`\large...`), pour avoir un gros titre ;
- les commandes d'espacement `\vspace` et `\hspace` pour aérer le texte ;
- on peut même inclure une image avec la commande `\epsfbox` (comme j'ai fait pour le logo ENSTA) ;

Seules contraintes, l'environnement `titlepage` doit se trouver juste après la commande `\begin{document}` et la page de garde doit tenir sur une page (mais vous vous en doutez, je pense).

8.4 Faire une bibliographie

Avec $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, il y a deux façons de faire une bibliographie :

- utiliser l’environnement `thebibliography` ;
- utiliser le programme `BIB $\text{T}_{\text{E}}\text{X}$` .

Comme je ne sais pas utiliser `BIB $\text{T}_{\text{E}}\text{X}$` , je me limiterai à l’environnement `thebibliography`.

8.4.1 L’environnement `thebibliography`

Voyons comment la bibliographie de ce manuel a été réalisée :

```
\clearpage

\addcontentsline{toc}{chapter}{Bibliographie}

\begin{thebibliography}{9}
  \bibitem{lampport:latex} Leslie \fsc{Lampport},
                        {\it \LaTeX: A Document Preparation System},
                        Addison-Wesley,
                        1986.
  \bibitem{fiter:intro} Bruno \fsc{Fiter},
                        {\it Introduction à \LaTeX},
                        1992.
\end{thebibliography}
```

Ce qui donne :

- [1] Leslie LAMPOR, *L A $\text{T}_{\text{E}}\text{X}$: A Document Preparation System*, Addison-Wesley, 1986.
- [2] Bruno FITER, *Introduction à L A $\text{T}_{\text{E}}\text{X}$* , 1992.

La commande `\clearpage` permet d’imprimer les tables et les figures qui n’ont pas encore pu l’être, et de débiter une nouvelle page.

Puis, on ajoute le titre « Bibliographie » à la table des matières (`toc`). Le second argument de la commande `\addcontentsline` précise le niveau du titre à ajouter, `chapter` pour le style `report`, `section` pour le style `article`.

L’environnement `thebibliography` s’utilise à peu près comme l’environnement `itemize`. Chaque élément de la bibliographie commence par la commande `\bibitem`, qui prend en argument une chaîne de caractères permettant, comme avec `\label`, de faire référence à l’ouvrage. Puis on met à la suite l’auteur de l’ouvrage, son titre (en italique), son éditeur et son année de parution.

Chaque élément de la bibliographie est repéré par un nombre entre crochets. Il est possible d’assigner une étiquette personnalisée grâce à l’argument optionnel de `\bibitem` :

```
\bibitem[Lampport 86]{lampport:latex} Leslie \fsc{Lampport}, ...
```


Ce qui donne :

```
[Lamport 86] Leslie LAMPORT, ...
```

L'argument de l'environnement `thebibliography` précise la taille maximale de ces étiquettes :

- lorsqu'on ne les personnalise pas, on met généralement 9 si l'on a moins de 10 ouvrages dans la bibliographie, 99 si l'on en a entre 10 et 99, etc. ;
- lorsqu'on les personnalise, on met l'étiquette la plus longue.

La commande `\cite` permet de faire référence à un élément de la bibliographie, elle imprime le numéro ou l'étiquette de l'ouvrage :

```
Voir \cite{lamport:latex} pour plus de précisions.
```

```
Voir [1] pour plus de précisions.
```

On peut indiquer des renseignements supplémentaires (page, chapitre...) grâce à l'argument optionnel de `\cite` :

```
Voir \cite[page 187]{lamport:latex} pour plus de précisions.
```

```
Voir [1, page 187] pour plus de précisions.
```

8.4.2 BIB_TE_X

Je ne sais malheureusement pas me servir de BIB_TE_X. Je serais reconnaissant à quiconque de m'expliquer comment ça marche.

8.5 Faire un index

La réalisation d'un index est très simple à faire avec L^AT_EX. Il suffit de placer dans le texte des commandes `\index` pour générer les entrées.

Avant cela, il faut modifier un peu votre document :

- si vous n'utilisez pas l'option de style `french`, chargez l'option de style `makeidx` (entre les crochets de la commande `\documentstyle`);
- ajoutez la commande `\makeindex` dans le préambule (c'est-à-dire entre les commandes `\documentstyle` et `\begin{document}`);
- mettez la commande `\printindex` où vous voulez voir apparaître l'index (c'est généralement à la fin du document, juste avant la commande `\end{document}`).

Il va falloir renoncer à utiliser *xlatex* (du moins si vous voulez générer un document avec index et table des matières à jour). En effet, pour un document avec index, la séquence de compilation est la suivante :

- une première compilation génère l'index dans un fichier *.idx* (c'est le rôle de la commande `\makeindex`);
- le programme *makeindex* formate l'index et le place dans un fichier *.ind*;
- une deuxième compilation inclut l'index et génère la table des matières (avec l'index);
- enfin, une troisième compilation inclut la table des matières.

L'utilisation de *makeindex* est expliquée dans la section 8.5.2 et l'automatisation de la compilation au moyen d'un *Makefile* dans la section 8.5.3. Vous pouvez également vous reporter à [5] pour une description plus complète de la réalisation d'un index avec L^AT_EX.

8.5.1 La commande `\index`

Pour générer une entrée dans l'index, utilisez la commande `\index` :

Le mot `babasse``\index{babasse}` est inclus dans l'index.

Ceci imprime :

Le mot `babasse` est inclus dans l'index.

et génère une entrée pour « `babasse` » dans l'index.

Il est possible de mettre toute une expression dans l'index :

Je mets l'expression `super top cool``\index{super top cool}`
dans l'index.

Pour générer des sous-entrées, il faut séparer l'entrée principale et la sous-entrée avec le caractère ! :

Ceci parle des sous-entrées de l'index`\index{index!sous-entrées}`

Enfin, on peut placer du texte à la position alphabétique repérée par un autre texte avec le caractère @ :

Je place le symbole `\alpha``\index{alpha@\alpha}` dans l'index,
à la place de « `alpha` » dans l'ordre alphabétique.

Il peut parfois être utile d'utiliser des accolades de groupement :

`taratata italique``\index{italique@{\it italique}}`

8.5.2 Formatage de l'index : le programme *makeindex*

Une fois l'index généré (dans le fichier *.idx*), il faut le mettre en forme avec le programme *makeindex*. Celui-ci produit un fichier *.ind* qui peut alors être inclus dans le document par la commande `\printindex`. *Makeindex* s'appelle par la commande :

```
makeindex toto.idx
```

Le format par défaut de *makeindex* est assez laid. Je vous conseille donc d'utiliser le fichier de format *index.ist* que vous pouvez trouver chez moi (*~babafou/tex/index.ist*). Recopiez-le dans votre répertoire de travail et appelez *makeindex* ainsi :

```
makeindex -s index.ist toto.idx
```

L'index sera alors formaté comme celui de ce manuel.

8.5.3 Automatiser la compilation du document en utilisant un *Makefile*

Il est facile d'automatiser la compilation en utilisant un *Makefile*. Pour un fichier *toto.tex*, le *Makefile* suivant réalise dans l'ordre :

- la compilation de génération de l'index ;
- le formatage de l'index ;
- les deux compilations de génération et d'inclusion de la table des matières ;
- la conversion du document au format PostScript.

```
toto.ps : toto.dvi
    dvips toto.dvi

toto.dvi : toto.tex toto.ind
    latex toto.tex
    latex toto.tex

toto.ind : toto.idx
    latex toto.tex
    makeindex -s index.ist toto.idx

clean :
    rm -f toto.aux toto.idx toto.ilg toto.ind \
        toto.lof toto.log toto.lot toto.toc
```

Il suffit de taper `make` et d'attendre un peu pour avoir un document prêt à imprimer. Accessoirement, `make clean` permet d'effacer tous les fichiers générés par L^AT_EX, à l'exception de *toto.tex* et *toto.ps*.

Annexe A

Règles typographiques françaises et anglaises

Les règles typographiques sont différentes en Français et en Anglais, surtout en ce qui concerne la ponctuation.

A.1 Règles typographiques françaises

- la virgule et le point sont collés au mot qui les précède et sont suivis par un espace ;
- le point-virgule, les deux points, le point d'exclamation et le point d'interrogation sont précédés et suivis d'un espace ;
- les points de suspension sont collés au mot qui les précède.

Voici un petit exemple de poésie téléphonique à la ponctuation légèrement retouchée pour servir mon propos :

J'avais un ami, mais il est parti. Ce sens à ma vie, il n'est plus en vie... Il m'a tout donné, puis s'est effacé, sans me déranger ; et je crois j'ai pleuré, j'ai pleuré. Tout au fond, l'air d'un con : envie de me casser. Voudrais bien essayer, voudrais bien continuer, continuer ! Un moment, un instant, j'ai cru oublier. En parlant, en marchant, à Paris, à minuit, dans ma ville ?

J'avais un ami, mais il est parti. Ce sens à ma vie, il n'est plus en vie... Il m'a tout donné, puis s'est effacé, sans me déranger ; et je crois j'ai pleuré, j'ai pleuré. Tout au fond, l'air d'un con : envie de me casser. Voudrais bien essayer, voudrais bien continuer, continuer ! Un moment, un instant, j'ai cru oublier. En parlant, en marchant, à Paris, à minuit, dans ma ville ?

Les « guillemets français » ne sont disponibles qu'avec l'option de style french (voir [7, page 3]).

A.2 Règles typographiques anglaises

- la virgule, le point-virgule, les deux points, le point d’exclamation et le point d’interrogation sont collés au mot qui les précède et sont suivis par un espace ;
- le point est collé au mot qui le précède et suivi de deux espaces (`LATEX` s’en moque, mais c’est l’usage) ;
- les points de suspension sont précédés et suivis d’un espace et sont plus espacés que trois points se suivant, c’est pourquoi `LATEX` dispose de la commande `\ldots` pour saisir les points de suspension anglais (n’oubliez pas la commande `\quad` pour l’espace suivant les points de suspension).

L’indicatif de fin de semaine (à la ponctuation modifiée) devrait vous montrer ça :

```
We all came out to Montreux, on the Lake Geneva shoreline, to
make records with a mobile: we didn't have much time. Franck
Zappa and the Mothers were at the best place around; but some
stupid with a flare gun burned the place to the ground \ldots\
Smoke on the water! Fire in the sky? Smoke on the water!
```

```
We all came out to Montreux, on the Lake Geneva shoreline, to make
records with a mobile: we didn't have much time. Franck Zappa and the
Mothers were at the best place around; but some stupid with a flare gun
burned the place to the ground ... Smoke on the water! Fire in the sky?
Smoke on the water!
```

Les guillemets se saisissent ‘ ‘ et ’ ’ (et pas "). Ce qui donne “ceci”.

Annexe B

SliT_EX

SliT_EX est une version de L_AT_EX qui permet de faire des transparents (*slides* en anglais, d'où le nom SliT_EX).

Pour pouvoir utiliser SliT_EX avec *x_latex*, il faut aller remplacer dans le champ *Façonneur* le nom `latex` par `slitex`.

SliT_EX a besoin de deux types de fichiers :

- un fichier de base ;
- un ou plusieurs fichiers de transparents.

B.1 Le fichier de base

Le fichier de base est très simple et ne contient généralement que :

```
\documentstyle[A4,french]{slides}

\begin{document}

\blackandwhite{transparentes.tex}

\end{document}
```

Ceci ressemble beaucoup à un document L_AT_EX, à quelques exceptions près :

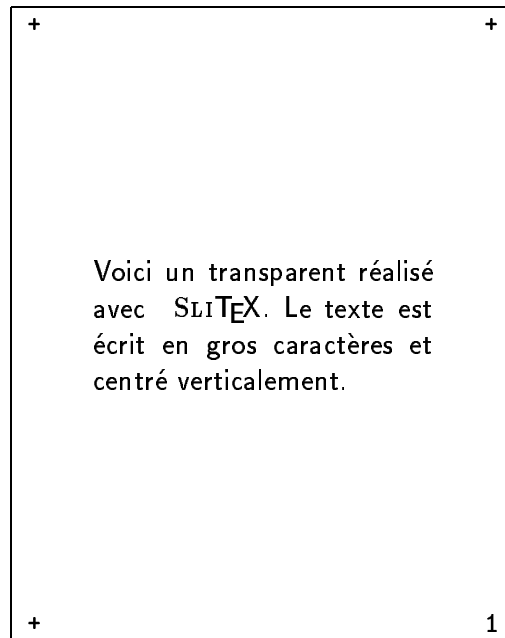
- le style du document est `slides` ;
- on ne charge pas les options de style `11pt` et `12pt`, mais on peut charger `A4`, `french`, `epsf`, etc. ;
- le corps du document ne contient que des commandes `\blackandwhite` dont chacune charge un fichier de transparents.

B.2 Les fichiers de transparents

Chaque fichier de transparents contient un ou plusieurs environnements `slide`, et chaque environnement `slide` contient le texte d'un transparent.

Dans l'environnement `slide`, le texte est écrit en gros caractères et centré verticalement. Chaque page est numérotée dans le coin inférieur droit et des signes « + » sont placés dans les trois autres coins :

```
\begin{slide}{}
Voici un transparent réalisé avec {\sc Sli}\TeX. Le texte est
écrit en gros caractères et centré verticalement.
\end{slide}
```



La commande `\pagestyle{plain}`, placée dans le préambule du fichier de base, permet de supprimer les « + » dans les coins. De même, la commande `\pagestyle{empty}` permet de supprimer les « + » et les numéros de page.

Vous pouvez utiliser avec SLIT_EX la plupart des commandes et environnements L^AT_EX qui peuvent servir à faire des transparents (par exemple, des commandes comme les commandes de sectionnement ne servent pas dans les transparents et ne sont donc pas connues de SLIT_EX). Je n'ai pas de liste des commandes L^AT_EX non disponibles dans SLIT_EX, donc, dans le doute, essayez !

Pour plus de précisions sur SLIT_EX, reportez-vous à [1, pages 132–138].

Annexe C

Reprogrammation des touches de fonction

Il est possible de redéfinir les caractères accentués obtenus directement grâce aux touches de fonction. En effet, le système de fenêtrage X permet de modifier la signification des touches du clavier. On peut alors leur assigner de nouveaux caractères accentués pour les textes écrits en espagnol, en italien, etc.

C.1 Comment X gère le clavier

La gestion du clavier sous X est très simple :

- à chaque touche est associé un nombre qu'on appelle un *keycode* et qui identifie la touche de manière unique (deux touches différentes ne peuvent avoir le même *keycode*) ;
- à chaque *keycode* (et donc à chaque touche) est associé un autre nombre qu'on appelle un *keysym* et qui représente le caractère affiché par cette touche.

Les programmes *xkeycaps* et *xev* permettent de connaître les *keycodes* et les *keysyms* des touches.

C.2 Modifier les associations *keycode/keysym*

Le programme *xmodmap* permet de modifier le *keysym* associé à un *keycode*. Pour cela, on utilise un fichier *xmodmaprc* dans lequel chaque ligne est de la forme

```
keycode keycode = keysym
```

puis on tape

```
xmodmap xmodmaprc
```

et le tour est joué!

Bibliographie

- [1] Leslie LAMPORT, *L_AT_EX: A Document Preparation System*, Addison-Wesley, 1986.
- [2] Bruno FITER, *Introduction à L_AT_EX*, 1992.
- [3] Gavin MALTBY, *An introduction to T_EX and friends*, 1992.
- [4] Claude GOMEZ, *Notes de cours L_AT_EX*, 1992.
- [5] Leslie LAMPORT, *MakeIndex: An Index Processor For L_AT_EX*, 1987.
- [6] Babafou, *Options de style pour L_AT_EX disponibles à l'ENSTA*, 1993.
- [7] Bernard GAULLE, *Notice d'utilisation du style french*, 1993.

Index

Symboles

<code>\-</code>	17
<code>%</code>	17
<code>-</code>	42
<code>^</code>	42
<code>~</code>	16
11pt	15
12pt	15

A

A4	15
accents	21
accolades de groupement ...	20, 23, 63
annexe	26
article	14

B

<code>\bibitem</code>	61
bibliographie	61
BIB _T E _X	61
book	14

C

commande	13
syntaxe	19
commentaire	17
<code>.cshrc</code>	8
césure	17

D

dessin	59
document	15
<code>\documentstyle</code>	13, 14

E

environnement	29
epsf	59

epsfbox	59
epsfxsize	59
epsfysize	59
espace insécable	16
exposant	42

F

fleqn	42
forme étoilée	
commande	19
environnement	29
fraction	49
french	15

I

index	62
<code>\index</code>	63
<i>index.ist</i>	64
indice	42
intégrale	50

L

L _A T _E X	5, 7
letter	14
lettres grecques	44

M

makeidx	62
<i>makeindex</i>	62, 64
<code>\makeindex</code>	62
matrice	53
modes mathématiques	41
environnement <code>displaymath</code> ..	42
environnement <code>math</code>	41
mot	16

N

- | | | | |
|------------------------------------|--------|---------------------------|----|
| <code>\newcommand</code> | 57 | <code>xdvi</code> | 11 |
| <code>\newenvironment</code> | 58 | <code>xfig</code> | 59 |
| note | | <code>xlatex</code> | 8 |
| note dans la marge | 28 | | |
| note de bas de page | 27 | | |
| O | | | |
| option de style | 13, 15 | | |
| <code>Overfull \hbox</code> | 17 | | |
| P | | | |
| page de garde | 26, 60 | | |
| paragraphe | 15 | | |
| PostScript encapsulé | 59 | | |
| <code>\printindex</code> | 62 | | |
| préambule | 15 | | |
| R | | | |
| racine | 49 | | |
| report | 14 | | |
| références croisées | 28 | | |
| S | | | |
| <code>slide</code> | 68 | | |
| <code>slides</code> | 67 | | |
| <code>SLiTeX</code> | 67 | | |
| somme | 50 | | |
| style | 13, 14 | | |
| symboles mathématiques | | | |
| flèches | 45 | | |
| opérateurs binaires | 45 | | |
| relations binaires | 45 | | |
| symboles divers | 47 | | |
| T | | | |
| table des matières | 26 | | |
| tableaux | 37 | | |
| tabulation | 35 | | |
| <code>tex_env</code> | 8 | | |
| <code>thebibliography</code> | 61 | | |
| W | | | |
| Word | 5, 7 | | |
| WYSISYG | 7 | | |
| X | | | |