Apprends LaTeX !

babafou

École nationale supérieure de techniques avancées

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 LaTeX tel qu'il y est installé. Il se peut que l'installation de LaTeX sur votre site soit quelque peu différente. En particulier, les caractéristiques suivantes peuvent être absentes :

- les extensions babel, indentfirst ou graphicx ;
- le programme *xfig* ;
- la reprogrammation des touches de fonction en caractères accentués.

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

Pour aller plus loin

Ce manuel ne prétend pas être exhaustif (je ne pense d'ailleurs pas qu'il existe de manuel exhaustif sur LaTeX et ses nombreuses extensions). Il décrit les aspects de LaTeX qui me paraissent les plus utiles et qui vous permettront de rédiger la plupart de vos documents. Si vous voulez aller plus avant dans votre connaissance de LaTeX, je vous conseille de lire *LaTeX: a document preparation system*, de Leslie Lamport ([1]). Un autre livre indispensable est le *LaTeX Companion*, de Michel Goossens, Frank Mittelbach et Alexander Samarin ([2]), qui décrit un grand nombre d'extensions de LaTeX.

Historique

La première version de ce manuel a été écrite au printemps 1994 et décrivait LaTeX 2.09. La présente édition a été adaptée en 1996 et 1997 à LaTeX 2_{ε}.

Où trouver ce manuel ?

L'arborescence source complète de ce manuel (contenant également les fichiers `.dvi` et PostScript) ainsi que le fichier PostScript seul sont disponibles par FTP anonyme sur les serveurs suivants :

- `<URL:ftp://ftp.agm-ita.ensta.fr/pub/babafou/>` (site de référence)
- `<URL:ftp://ftp.univ-lyon1.fr/pub/doc/french/texte/apprends_latex/>`
- `<URL:ftp://ftp.oleane.net/pub/doc/francais/apprends_latex/>`
- `<URL:ftp://ftp.lps.ens.fr/pub/users/babafou/>`

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

Vous pourrez également trouver une version HTML sur le serveur du LORIA :

- `<URL:http://www.loria.fr/services/tex/general.html>`

Conventions typographiques

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 noms des commandes et des environnements LaTeX, pour les noms de fichiers et pour les commandes UNIX tapées au clavier ;
- l'*italique* est utilisé lors de la définition de nouveaux termes ainsi que pour les options et les arguments des commandes LaTeX ;
- le style antique est utilisé pour les noms de classes et de leurs options ainsi que pour les noms d'extensions et de leurs options.

Remarques, commentaires et félicitations

Je serais heureux de savoir ce que vous pensez de mon manuel. Pour cela, vous pouvez m'envoyer un message à l'adresse électronique suivante :

<babafou@agm-ita.ensta.fr>

Chapitre 1

Introduction

Ce manuel est destiné à vous apprendre LaTeX[1] dans la joie et la bonne humeur. Après l'avoir lu, vous devriez être en mesure d'utiliser LaTeX pour réaliser tous vos documents.

1.1 Qu'est-ce que LaTeX ?

LaTeX est un formateur de texte d'une très grande puissance et qui produit 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 LaTeX est capable de faire, feuilletez ce manuel, il a été entièrement réalisé avec lui.

LaTeX est une extension d'un formateur de texte appelé TeX (le nom TeX vient du grec $\tau\acute{\epsilon}\chi\nu\eta$[2], qui veut dire « art »). TeX est extrêmement puissant et dispose d'un langage permettant de lui ajouter de nouvelles fonctions mais il est d'un abord assez ardu. LaTeX, en revanche, est bien plus simple et, comme il est une surcouche de TeX, il dispose du même langage, qui a servi à réaliser de très nombreuses extensions pour LaTeX.

Cependant, vous vous demandez certainement quel est l'intérêt de LaTeX par rapport à des logiciels de traitement de texte tels qu'Applix ou Word, qui s'utilisent d'une manière intuitive. Tout est affaire de philosophie.

1.2 La philosophie de LaTeX

LaTeX n'est pas un traitement de texte à proprement parler, comme Applix ou Word, c'est un *formateur de texte*. Un document LaTeX est un fichier texte tapé avec n'importe quel éditeur (*Emacs*, *vi*...) et contenant le texte

[1] Absolument rien à voir avec le caoutchouc. La fin de « LaTeX » se prononce comme dans le mot « varech ».

[2] D'où vient le mot « technique », ce qui explique la prononciation de LaTeX.

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 Applix et 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 pour les personnes habituées à voir le document au fur et à mesure de sa saisie. D'un autre côté, LaTeX décharge au maximum l'utilisateur des soucis de mise en page, ce qui lui permet de se concentrer sur la structure et le contenu de son document.

De plus, puisque qu'il traite le document globalement (et pas au fur et à mesure de la frappe), LaTeX peut en avoir une vue d'ensemble et faire de bons choix de mise en page.

En résumé, il faut bien comprendre que LaTeX traite le document tel qu'on le pense alors qu'Applix et Word ne traitent le document que tel qu'ils le voient. C'est cette intelligence qui fait tout l'intérêt de LaTeX.

Chapitre 2

Votre premier document

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

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

Nous expliquerons l'exemple de document LaTeX présenté ici dans les chapitres suivants.

2.1 Avant de commencer

Créez un répertoire `LaTeX`, dans lequel vous rangerez vos documents. Ceci étant fait une fois pour toutes, vous allez pouvoir réaliser votre premier document avec LaTeX.

2.2 Allons-y !

C'est parti, on commence.

2.2.1 L'édition

Éditez votre document au moyen de votre éditeur favori (*Emacs*, *vi*...) et tapez ceci :

```
\documentclass[a4paper,11pt]{article}

\usepackage[latin1]{inputenc}
```

```
\usepackage[T1]{fontenc}
\usepackage[francais]{babel}

\begin{document}

Voici mon premier document avec \LaTeX.

\end{document}
```

Ne vous inquiétez pas si vous ne comprenez rien à cet exemple, il sera expliqué dans les chapitres suivants.

Sauvez ce texte sous le nom `premier.tex` (les documents LaTeX ont, par convention, l'extension `.tex`).

2.2.2 La compilation

Compilez votre document au moyen de la commande

```
latex premier.tex
```

LaTeX affiche ce qu'il fait au fur et à mesure :

```
This is TeX, Version 3.14159 (Web2c 7.0)
(premier.tex
LaTeX2e <1997/06/01>
Hyphenation patterns for english, french, loaded.
(/usr/local/share/texmf/tex/latex/base/article.cls
Document Class: article 1997/06/16 v1.3v Standard LaTeX document class
(/usr/local/share/texmf/tex/latex/base/size11.clo))
(/usr/local/share/texmf/tex/latex/base/inputenc.sty beta test version
(/usr/local/share/texmf/tex/latex/base/latin1.def))
(/usr/local/share/texmf/tex/latex/base/fontenc.sty
(/usr/local/share/texmf/tex/latex/base/t1enc.def))
(/usr/local/share/texmf/tex/latex/babel/babel.sty
(/usr/local/share/texmf/tex/latex/babel/francais.ldf
(/usr/local/share/texmf/tex/latex/babel/babel.def)))
No file premier.aux.
[1] (premier.aux) )
Output written on premier.dvi (1 page, 352 bytes).
Transcript written on premier.log.
```

2.2.3 Les fichiers générés

Lorsqu'il compile un document, LaTeX génère un certain nombre de fichiers :

- un fichier `.dvi` (*device-independent*) qui est le résultat de la compilation, c'est ce fichier qui est directement visualisable et qui servira à l'impression du document (après avoir été converti en PostScript) ;
- un fichier `.aux` qui contient les références croisées ;
- un fichier `.log` qui contient une copie des messages affichés lors de la compilation.

Suivant la complexité de votre document, LaTeX peut également créer les fichiers :

- `.toc`, contenant la table des matières (*table of contents*) ;
- `.lot`, contenant la liste des tableaux (*list of tables*) ;
- `.lof`, contenant la liste des figures (*list of figures*) ;
- `.idx`, contenant l'index non formaté.

2.2.4 La détection et la correction des erreurs

Si vous vous êtes trompé quelque part, un message de ce genre s'affiche :

```
! Undefined control sequence.
l.9 Voici mon premier document avec \latex
                                          .
?
```

Lorsqu'il détecte une erreur, LaTeX vous indique :

- la nature de cette erreur (`Undefined control sequence`) ;
- la ligne où elle se situe (`l.9`, ligne 9) ;
- ainsi que l'endroit précis de cette erreur dans la ligne (pour vous l'indiquer, LaTeX passe à la ligne) : ici, la commande inconnue est `\latex`.

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

2.2.5 La visualisation

Pour visualiser votre document, tapez la commande

```
xdvi premier.dvi
```

Une fenêtre s'ouvre, vous montrant le document tel qu'il sera imprimé.

Grâce aux boutons placés dans la partie droite de la fenêtre, vous pouvez :

- vous déplacer dans le document (il est également possible de se déplacer grâce au clavier : la barre d'espace fait avancer d'une page et la touche BackSpace fait reculer d'une page) ;
- modifier le grossissement.

Il est également possible de zoomer en cliquant dans la fenêtre avec les boutons de la souris :

bouton de gauche	→	petit zoom
bouton du milieu	→	moyen zoom
bouton de droite	→	grand zoom

2.2.6 L'impression

Pour imprimer votre document, il faut tout d'abord le convertir en PostScript (c'est le langage de description de page utilisé par les imprimantes laser) au moyen de la commande

```
dvips premier.dvi
```

Il ne vous reste plus qu'à imprimer le fichier `premier.ps` généré au moyen des commandes *lpr* ou *lp* (suivant le système d'exploitation que vous utilisez).

Chapitre 3

Les concepts de base de LaTeX

Avant de pouvoir utiliser pleinement LaTeX et de profiter de sa puissance, il faut comprendre certains concepts. Ce chapitre vous initiera plus profondément à la philosophie et à l'esprit LaTeX et vous expliquera l'exemple du chapitre précédent.

3.1 Le document LaTeX

Un document LaTeX est un mélange de texte et de commandes de formatage.

Reprenons l'exemple précédent :

```
\documentclass[a4paper,11pt]{article}

\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[francais]{babel}

\begin{document}

Voici mon premier document avec \LaTeX.

\end{document}
```

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

```
\documentclass[a4paper,11pt]{article}
```

Cette commande est indispensable et doit figurer en première ligne de tout document LaTeX. Elle définit la *classe* du document (ici article) ainsi que les *options* à lui appliquer (ici a4paper et 11pt).

3.1.1 Syntaxe d'une commande LaTeX

On reconnaît ici la syntaxe générale d'une commande LaTeX :

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

3.1.2 La commande `\documentclass`

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

3.1.2.1 La classe d'un document

La classe, qui est l'argument obligatoire de la commande `\documentclass`, définit le type du document. Il existe cinq classes :

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

report est utilisé pour les documents plus longs qui doivent être découpés en chapitres (en effet, la classe article ne gère pas les chapitres) ;

book est utilisé pour les livres (il y a peu de différences avec la classe report) ;

letter permet d'écrire des lettres ;

slides permet de faire des transparents.

3.1.2.2 Les options de classe

Les options de classe, qui sont les arguments optionnels de la commande `\documentclass`, servent à modifier globalement la mise en page du document.

Expliquons quelque peu les options utilisées ici :

a4paper 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) ;

11pt porte la taille des caractères à onze points[1] (la taille normale, sans option, est de dix points mais je la trouve un peu petite ; il existe aussi une option 12pt).

3.1.3 Le préambule

Le *préambule* est la partie du document comprise entre les commandes `\documentclass` et `\begin{document}`. Dans notre exemple, il contient les commandes de chargement des extensions.

3.1.4 Les extensions

Les extensions (*packages* en anglais) servent à modifier la mise en page ou à définir de nouvelles commandes. Ce sont elles qui permettent d'ajouter de nouvelles fonctions à LaTeX. Elles sont chargées grâce à la commande `\usepackage`, utilisée dans le préambule du document.

Certaines de ces extensions sont livrées d'origine avec LaTeX, d'autres doivent être installées par l'administrateur LaTeX. Parmi les extensions standard, on trouve :

fontenc qui, utilisée avec l'option T1, demande à LaTeX d'utiliser le codage de caractères T1 (c'est le nouveau standard LaTeX pour le codage des caractères mais il n'est pas utilisé par défaut pour des raisons de compatibilité avec les anciens documents, d'où la nécessité d'utiliser cette extension) ;

inputenc qui permet, grâce à l'option latin1, d'utiliser des caractères ISO 8859-1 dans le document. Avec cette extension, vous pouvez saisir directement les caractères accentués dans vos documents, je vous recommande donc de l'utiliser systématiquement.

L'extension babel, quant à elle, si elle n'est pas standard, est très utile. Utilisée avec l'option francais, elle adapte LaTeX aux conventions typographiques françaises et redéfinit quelques noms en bon français, comme par

[1] Le point est une unité de mesure utilisée en typographie. Un point vaut 1/72e de pouce (1 pouce valant 2,54 cm).

exemple « Table des matières » ou « Chapitre » au lieu respectivement de « Table of contents » et « Chapter ».

Je vous conseille d'utiliser ces extensions dans tous vos documents.

Il existe beaucoup d'autres extensions. Nous en verrons quelques unes par la suite.

3.1.5 Le document

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

On remarque ici une structure très utilisée en LaTeX : la structure d'*environnement* (voir le chapitre 5 pour une description complète des environnements de LaTeX).

3.2 Structure du document

3.2.1 Les paragraphes

Un document LaTeX 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.
```

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.

Chaque paragraphe débute par un alinéa, sauf le paragraphe initial d'un chapitre, d'une section, etc. (en fait, tout paragraphe suivant une commande de hiérarchisation, voir 4.6). Pour que chaque paragraphe, quel qu'il soit, débute par un alinéa, utilisez l'extension indentfirst :

```
\usepackage{indentfirst}
```

3.2.2 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 une espace). Vous pouvez insérer entre deux mots autant d'espaces que vous voulez (mais une seule 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 une espace, profitez-en pour couper vos paragraphes en fin de ligne. En effet, avec un éditeur de texte, 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.

3.2.3 L'espace insécable : `~`

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

> Connaissez-vous la série télé 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 LaTeX ne les sépare jamais (c'est beau l'amour !). Ceci est possible grâce au caractère `~`. Ce caractère représente une *espace insécable*, qui est considérée comme une espace (et donc apparaîtra comme tel à l'impression), mais LaTeX ne séparera jamais en fin de ligne les deux mots qu'elle lie pour le meilleur et pour le pire :

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

> Connaissez-vous la série télé dans laquelle le héros est le nu-
> méro 6 ?

N'utilisez pas systématiquement l'espace insécable, cela nuit à la lisibilité du source. Attendez d'avoir complètement fini la rédaction de votre document pour y insérer éventuellement des espaces insécables, au cas par cas.

3.2.4 Les commentaires

LATEX 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.
```

3.3 Caractères spéciaux utilisés par LATEX

Vous avez remarqué que certains caractères ont une signification spéciale pour LATEX. Ainsi, le `\` indique une commande, le `~` une 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, LATEX ne fournit aucune commande permettant de les imprimer. Cependant, l'extension babel fournit les commandes `\tild` et `\circonflexe` permettant d'imprimer respectivement ˜ et ˆ.

Pour le \, il est nécessaire de définir soi-même une commande (voir la section 10.1.1) :

```
\newcommand{\bs}{\symbol{92}}
```

La commande `\bs` permet alors d'imprimer \.

3.4 Les problèmes de césure : `Overfull \hbox`

Il arrive parfois que LATEX 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 LATEX génère l'avertissement suivant :

```
Overfull \hbox (63.46103pt too wide) in paragraph at lines 99--99
```

L'avertissement `Overfull \hbox` signifie que quelque chose dépasse de la marge droite. LaTeX signale de combien le texte dépasse (`63.46103pt`) et à quel endroit se situe le problème (`lines 99-99`).

Rassurez-vous, cela n'arrive que très rarement. Dans ce cas, il est possible de préciser à LaTeX où il doit couper le mot. Pour cela, repérez l'endroit approprié grâce à *xdvi* et placez dans le texte la commande `\-` où vous voulez que LaTeX coupe le mot :

```
Je veux cou\-per ici.
```

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 d'autres avertissements de césure apparaissent ou disparaissent. C'est pourquoi je vous conseille de n'effectuer vos césures manuelles qu'après avoir terminé la rédaction de votre document, une par une, en vérifiant à chaque fois le résultat (on peut le faire visuellement avec *xdvi*, mais il suffit que le message `Overfull \hbox` ait disparu pour être sûr d'avoir bien effectué la césure).

Chapitre 4

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 structurer le document, etc.

4.1 Syntaxe d'une commande

Voici la syntaxe d'une commande LaTeX :

`\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.

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 :

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

4.2 Caractères spéciaux

Certains caractères, certains symboles, ne figurent pas sur les claviers. Pour les utilisateurs de claviers QWERTY, c'est en particulier le cas des voyelles accentuées, indispensables en français. Heureusement, LaTeX met à votre disposition une série de commandes palliant ce manque.

4.2.1 Symboles

LATEX définit des commandes permettant d'imprimer divers symboles (voir le tableau 4.1) ainsi que des symboles utilisés en français et dans des langues étrangères (voir le tableau 4.2).

†	`\dag`	§	`\S`	©	`\copyright`
‡	`\ddag`	¶	`\P`	£	`\pounds`

Tableau 4.1: Symboles divers

œ	`\oe`	å	`\aa`	ł	`\l`
Œ	`\OE`	Å	`\AA`	Ł	`\L`
æ	`\ae`	ø	`\o`	ß	`\ss`
Æ	`\AE`	Ø	`\O`		

Tableau 4.2: Symboles pour le français et les langues étrangères

Pour utiliser les commandes du tableau 4.2 au sein d'un mot, il est nécessaire de les entourer d'accolades afin d'éviter toute ambiguïté. En effet, `c\oeur` contient-il la commande `\oe`, `\oeu` ou `\oeur` ? Ainsi, « cœur » s'écrit `c{\oe}ur`.

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 (vous remarquerez au passage que ces commandes sont assez spéciales puisqu'elles ne commencent pas par un `\`).

LATEX définit également trois autres commandes permettant d'imprimer les logos TEX (`\TeX`), LATEX (`\LaTeX`) et LATEX 2ε (`\LaTeXe`).

Voyons donc ce que donne le texte suivant :

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

Voici le résultat :

Essai des commandes TEXet LATEXdans un texte.

Ce n'est pas exactement ce à quoi nous nous attendions, parce que les mots « et » et « dans » sont collés aux logos qui les précèdent. En effet, LATEX ignore toute espace (une ou plusieurs) suivant une commande sans argument (ce qui est le cas des commandes `\TeX`, `\LaTeX` et `\LaTeXe`). Il y a trois façons de s'en tirer :

- utiliser la commande `\␣` (backslash espace) qui imprime une 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 19) :

```
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 et donne un texte source plus facile à lire.

4.2.2 Les lettres accentuées

LaTeX définit aussi des commandes permettant d'imprimer toutes sortes de lettres accentuées (voir le tableau 4.3). Bien entendu, vous pouvez aussi taper directement les lettres accentuées présentes sur votre clavier (à condition d'utiliser l'extension inputenc avec l'option latin1).

è	`\'{e}`	ñ	`\~{n}`	ě	`\v{e}`	ç	`\c{c}`
é	`\'{e}`	ē	`\={e}`	e̋	`\H{e}`	ẹ	`\d{e}`
ê	`\^{e}`	ė	`\.{e}`	e͡e	`\t{ee}`	e̱	`\b{e}`
ë	`\"{e}`	ĕ	`\u{e}`				

Tableau 4.3: Accents

Mais tout ceci est loin d'être pratique. LaTeX permet donc d'utiliser, pour certaines de ces commandes, une forme simplifiée présentée dans le tableau 4.4.

è	`\'e`	ñ	`\~n`
é	`\'e`	ē	`\=e`
ê	`\^e`	ė	`\.e`
ë	`\"e`		

Tableau 4.4: Commandes simplifiées d'accentuation

Pour placer un accent sur un i ou un j, utilisez les commandes `\i` ou `\j`, qui produisent des lettres sans point, en argument des commandes du tableau 4.3.

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. Il est donc absolument indispensable de
conna\^{\i}tre les commandes d'accentuation.
```

Mais, pour les utilisateurs de claviers QWERTY, ce n'est tout de même pas très pratique. L'idéal serait de pouvoir taper directement les lettres accentuées au clavier. 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 4.5).

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

Tableau 4.5: 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.

Le ü n'est pas disponible par manque de touches. Ce n'est d'ailleurs pas très grave, puisqu'il n'est utilisé que dans le mot « capharnaüm » et dans quelques rares noms propres (Emmaüs, Ésaü...).

4.3 Déclarations et commandes de changement de style

LaTeX permet bien entendu de changer le style des caractères. On distingue :

- la *forme* (voir le tableau 4.6) ;

`{\upshape Upright shape}`	Upright shape
`{\itshape Italic shape}`	*Italic shape*
`{\slshape Slanted shape}`	*Slanted shape*
`{\scshape Small capitals}`	SMALL CAPITALS

Tableau 4.6: Déclarations de changement de forme

- la *graisse* (voir le tableau 4.7) ;

`{\mdseries Medium series}`	Medium series
`{\bfseries Boldface series}`	**Boldface series**

Tableau 4.7: Déclarations de changement de graisse

`{\rmfamily Roman family}`	Roman family
`{\sffamily Sans serif family}`	Sans serif family
`{\ttfamily Typewriter family}`	`Typewriter family`

Tableau 4.8: Déclarations de changement de famille

- la *famille* (voir le tableau 4.8). La famille *roman* s'appelle *romaine* en typographie française et la famille *sans serif* s'appelle *antique* ou *linéale* en typographie française.

Chacune de ces caractéristiques peut être changée indépendamment des autres.

Les commandes indiquées ici sont différentes des commandes que nous avons vues auparavant. Elles ne prennent pas d'argument et agissent sur le texte qui les suit. On les appelle des *déclarations*.

Afin de limiter la portée d'une déclaration, il faut utiliser des *accolades de groupement* :

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

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

On peut aussi mélanger les déclarations :

```
C'est bon le {\itshape gloubi{\bfseries boulga}}.
```

C'est bon le *gloubi**boulga***.

En passant, vous remarquerez que toute espace suivant une déclaration est ignorée (comme les espaces suivant les commandes `\TeX` et `\LaTeX`).

Il existe une commande correspondant à chacune de ces déclarations (voir le tableau 4.9).

Ces commandes modifient uniquement le style de leur argument, le texte qui suit n'est pas affecté :

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

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

Il est recommandé, dans la mesure du possible, d'employer les commandes de changement de style plutôt que les déclarations équivalentes.

De même que pour les déclarations, on peut aussi mélanger les commandes :

Déclaration	Commande
`\upshape`	`\textup`
`\itshape`	`\textit`
`\slshape`	`\textsl`
`\scshape`	`\textsc`
`\mdseries`	`\textmd`
`\bfseries`	`\textbf`
`\rmfamily`	`\textrm`
`\sffamily`	`\textsf`
`\ttfamily`	`\texttt`

Tableau 4.9: Commandes de changement de style

```
C'est bon le \textit{gloubi\textbf{boulga}}.
```

C'est bon le *gloubi**boulga***.

Signalons aussi l'existence de la commande `\emph`, qui sert à mettre en valeur son argument :

```
Je mets \emph{cette expression} en valeur.
```

Je mets *cette expression* en valeur.

On aborde ici l'un des aspects importants de la philosophie de LaTeX, la distinction entre la structure du document et son rendu visuel. Comme vous pouvez le voir, la commande `\emph` met son argument en valeur en l'affichant en italique. Si ce manuel avait été imprimé sur une imprimante ne disposant pas du style italique, l'argument de `\emph` aurait sans doute été souligné à la place. La commande `\emph` décrit donc non pas la façon dont il faut voir le texte, mais celle dont il faut le penser. Ce principe est la clé de voûte de la philosophie de LaTeX.

Imaginez que vous ayez à écrire un livre sur la botanique et que, dans ce livre, vous aimeriez que tous les noms latins des plantes soient en italique. Rien de plus simple, il suffit de les mettre en argument de `\textit`. Maintenant, alors que votre livre est terminé, votre éditeur vous demande de mettre les noms latins en gras plutôt qu'en italique, parce que cela correspond à sa charte graphique. Il vous faut donc remplacer tous les `\textit` par `\textbf` et encore, en faisant attention, car vous avez très bien pu utiliser l'italique pour autre chose que les noms latins. Bref, ce n'est pas très pratique...

LaTeX, conformément à sa philosophie, offre une solution élégante à ce problème. En effet, il est possible de définir ses propres commandes (pour le détail, reportez-vous au paragraphe 10.1.1). Il vous suffit donc de définir une commande `\latin` qui affichera son argument, en l'occurence un nom latin, dans le style de votre choix. Si les noms latins doivent être en gras, il suffit alors de modifier la définition de la commande `\latin`, c'est tout.

C'est pourquoi on n'utilise quasiment jamais les commandes de changement de style directement dans un document LaTeX. On préfère plutôt définir un ensemble de commandes reflétant la logique du texte et les utiliser à la place. Ainsi, pour écrire ce manuel, j'ai défini mes propres commandes :

- `\commande` permet d'écrire les noms de commandes en style machine à écrire ;
- `\classe` permet d'écrire les noms de classe en style antique ;
- et ainsi de suite...

Je vous encourage fortement à faire de même.

4.4 Déclarations de changement de corps

LaTeX dispose de déclarations (donc à utiliser avec des accolades de groupement) permettant de modifier le corps des caractères (voir le tableau 4.10).

corps	`\tiny`	corps	`\large`
corps	`\scriptsize`	corps	`\Large`
corps	`\footnotesize`	corps	`\LARGE`
corps	`\small`	corps	`\huge`
corps	`\normalsize`	corps	`\Huge`

Tableau 4.10: Déclarations de changement de corps

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

Quelques commandes plus ou moins utiles permettent de forcer la main à LaTeX pour la mise en page du document.

La commande `\newline` permet de forcer un saut de ligne. La commande `\\` est une abréviation pour `\newline` que vous rencontrerez souvent. N'abusez pas de `\\`. En particulier, cette commande ne sert pas à marquer la fin d'un paragraphe, une ligne blanche suffit.

La commande `\newpage` permet de commencer une nouvelle page. La commande `\clearpage` est équivalente mais elle force l'impression des tableaux et figures flottants (voir la section 5.10).

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

L'espace inséré par ces commandes est compressible, c'est-à-dire que LaTeX pourra insérer moins d'espace que demandé s'il le juge nécessaire (souvent pour des besoins de mise en page). Les formes étoilées `\hspace*{`*taille*`}` et `\vspace*{`*taille*`}` insèrent un espace incompressible, qui aura toujours la taille désirée.

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 pica (1/72$^{\text{e}}$ de pouce), c'est une unité typographique anglo-saxonne ;

`pc` pica (1 pica = 12 points), une autre unité typographique anglo-saxonne ;

`em` 1 em est égal à la largeur de la lettre « M » dans la police courante, c'est une unité propre à LaTeX ;

`ex` 1 ex est égal à la hauteur de la lettre « x » dans la police courante, c'est une 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 (avec de la vraie colle) un dessin ou un schéma. Ainsi, l'espace de 3 centimètres de la figure 4.1 a été obtenu grâce à la commande :

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

FIG. 4.1: Figure vide

Enfin, les commandes `\bigskip`, `\medskip` et `\smallskip` insèrent un espace vertical plus ou moins grand. La taille de l'espace qu'elles génèrent est fonction de la police utilisée.

4.6 Commandes de hiérarchisation du document

Hiérarchiser 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 4.11.

`\part{`*nom*`}`	`\subsection{`*nom*`}`
`\chapter{`*nom*`}`	`\subsubsection{`*nom*`}`
`\section{`*nom*`}`	

Tableau 4.11: Commandes de hiérarchisation

La commande `\chapter` n'est pas disponible dans la classe article. C'est pourquoi on utilise les classes report ou book lorsqu'on a besoin de faire des chapitres et la classe article sinon.

À noter que le paragraphe initial d'un chapitre, d'une section, etc. (en fait, tout paragraphe suivant une commande de hiérarchisation) ne débute pas par un alinéa, contrairement aux autres paragraphes. Pour que tous les paragraphes débutent par un alinéa, utilisez l'extension indentfirst.

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

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

Ce qui est très intéressant, c'est que c'est LaTeX 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`[1] et la seconde l'inclut dans le document.

Il existe également une commande `\listoftables`, qui imprime la liste des tableaux et une commande `\listoffigures`, qui imprime la liste des figures (voir section 5.10). Pour que ces listes soient à jour, il faut aussi deux compilations successives (mettant en jeu les fichiers `.lot`[2] et `.lof`[3]).

4.6.1 Forme étoilée

Chaque commande de hiérarchisation a une forme étoilée qui ne numérote pas le titre, ne modifie pas la numérotation des autres sections et ne produit pas d'entrée dans la table des matières. Ces formes étoilées sont principalement utilisées à la place des commandes normales dans les documents très courts et dans les cas où la numérotation n'est pas nécessaire.

[1] toc pour *table of contents.*
[2] `lot` pour *list of tables.*
[3] `lof` pour *list of figures.*

Par exemple, le titre de l'avant-propos a été obtenu ainsi :

```
\chapter*{Avant-propos}
```

On peut néanmoins faire apparaître le titre dans la table des matières grâce à la commande :

`\addcontentsline{toc}{`*niveau*`}{`*titre*`}`

qui fera apparaître *titre* dans la table des matières, au niveau *niveau* (qui peut être `part`, `chapter`, `section`, `subsection` ou `subsubsection`). Par exemple, pour inclure l'avant-propos dans la table des matières, j'ai utilisé :

```
\addcontentsline{toc}{chapter}{Avant-propos}
```

4.6.2 Annexes

Enfin, la commande `\appendix` permet de signaler à LaTeX que les chapitres (pour les classes report et book) ou les sections (pour la classe article) qui suivront doivent être considérées comme des annexes. Les annexes sont numérotées A, B, etc.

4.7 La page de garde

LaTeX 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.

`\date{`***date***`}` indique la date du document.

Les commandes `\title` et `\author` sont obligatoires. Si la commande `\date` n'est pas utilisée, la date du jour de la compilation est mise automatiquement.

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

```
\date{}
```

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

Voici un exemple de document avec une page de garde :

```
\documentclass[a4paper,11pt]{report}

\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[francais]{babel}

\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 classes report et book, la page de garde est imprimée seule au milieu d'une page alors que pour la classe article, elle est imprimée en haut de la première page. Dans ce cas, pour avoir la page de garde toute seule, il faut utiliser l'option de classe 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 10.3).

4.8 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.

4.8.1 Les notes de bas de page

Une note de bas de page[4] 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 ...
```

[4] Comme celle-ci.

4.8.2 Les notes dans la marge

ceci est une 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{ceci est une 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.

4.9 Les références croisées

Avec LaTeX, il est très facile 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 au numéro de 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 (4.9, n'est-ce pas ?) par la commande :

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

et à la page où elle débute (26, 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, `eqn` pour une équation...). 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 5

Les environnements de LaTeX

Les environnements sont des structures très puissantes et très utilisées en LaTeX.

5.1 Syntaxe d'un environnement

Un environnement est une partie du document délimitée par les commandes

`\begin{`*environnement*`}`

et

`\end{`*environnement*`}`

Tout comme les commandes, les environnements peuvent avoir des options et des arguments, auquel cas la commande de début d'environnement a 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*`*}`

Toute déclaration (changement de style ou de corps) effectuée à l'intérieur d'un environnement ne se propage pas à l'extérieur de cet environnement. Ainsi :

```
\begin{quote}
Je passe la suite du texte en \itshape 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.

5.2 L'environnement `em`

L'environnement `em` met son contenu en style `\emph`. Cela permet de mettre en relief un ou plusieurs paragraphes :

```
\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.*

5.3 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.

Ainsi,

```
Ceci est un paragraphe de texte tout ce qu'il y a de plus
normal, en dehors d'un quelconque environnement.
\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}
Ceci est un paragraphe de texte tout ce qu'il y a de plus
normal, en dehors d'un quelconque environnement.
```

donne :

Ceci est un paragraphe de texte tout ce qu'il y a de plus normal, en dehors d'un quelconque environnement.

> 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.

Ceci est un paragraphe de texte tout ce qu'il y a de plus normal, en dehors d'un quelconque environnement.

Alors que :

```
Ceci est un paragraphe de texte tout ce qu'il y a de plus
normal, en dehors d'un quelconque environnement.
\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}
Ceci est un paragraphe de texte tout ce qu'il y a de plus
normal, en dehors d'un quelconque environnement.
```

donne :

Ceci est un paragraphe de texte tout ce qu'il y a de plus normal, en dehors d'un quelconque environnement.

> 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.

Ceci est un paragraphe de texte tout ce qu'il y a de plus normal, en dehors d'un quelconque environnement.

5.4 L'environnement `verse`

LaTeX met l'environnement `verse` à la disposition des poètes :

- les marges sont augmentées comme avec l'environnement `quote` ;

- chaque vers doit être terminé par \\ pour passer au vers suivant, sauf le dernier de chaque strophe ;
- les strophes sont séparées par des lignes blanches.

Si LATEX avait existé à l'époque de Baudelaire :

```
\begin{verse}
Souvent, pour s'amuser, les hommes d'équipage \\
Prennent des albatros, vastes oiseaux des mers, \\
Qui suivent, indolents compagnons de voyage, \\
Le navire glissant sur les gouffres amers.

À peine les ont-ils déposés sur les planches, \\
Que ces rois de l'azur, maladroits et honteux, \\
Laissent piteusement leurs grandes ailes blanches \\
Comme des avirons traîner à côté d'eux.

Ce voyageur ailé, comme il est gauche et veule ! \\
Lui, naguère si beau, qu'il est comique et laid ! \\
L'un agace son bec avec un brûle-gueule, \\
L'autre mime, en boitant, l'infirme qui volait !

Le Poète est semblable au prince des nuées \\
Qui hante la tempête et se rit de l'archer ; \\
Exilé sur le sol au milieu des huées, \\
Ses ailes de géant l'empêchent de marcher.
\end{verse}
```

Souvent, pour s'amuser, les hommes d'équipage
Prennent des albatros, vastes oiseaux des mers,
Qui suivent, indolents compagnons de voyage,
Le navire glissant sur les gouffres amers.

À peine les ont-ils déposés sur les planches,
Que ces rois de l'azur, maladroits et honteux,
Laissent piteusement leurs grandes ailes blanches
Comme des avirons traîner à côté d'eux.

Ce voyageur ailé, comme il est gauche et veule !
Lui, naguère si beau, qu'il est comique et laid !
L'un agace son bec avec un brûle-gueule,
L'autre mime, en boitant, l'infirme qui volait !

Le Poète est semblable au prince des nuées
Qui hante la tempête et se rit de l'archer ;
Exilé sur le sol au milieu des huées,
Ses ailes de géant l'empêchent de marcher.

5.5 Les environnements `center`, `flushleft` et `flushright`

Ces environnements permettent de centrer du texte ou de ne l'aligner 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. LaTeX 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 aligné à gauche. \LaTeX\ se
débrouille tout seul pour faire la mise en page et
couper les lignes où il faut.

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

Voici un paragraphe aligné à gauche. LaTeX se débrouille tout seul pour faire la mise en page et couper les lignes où il faut.
Voici deux lignes
alignées à gauche.

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

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

Voici un paragraphe aligné à droite. LaTeX se débrouille tout seul pour faire la mise en page et couper les lignes où il faut.
Voici deux lignes
alignées à droite.

5.6 L'environnement `verbatim`

L'environnement `verbatim` est très simple : il ne fait rien ! Il se contente de mettre son contenu en style `\texttt`. 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 LATEX tel quel (en particulier, tous les exemples de ce manuel ont été réalisés avec l'environnement `verbatim`).

```
\begin{verbatim}

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

\end{verbatim}
```

donne :

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

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

Les environnements de liste sont très souvent employés par les utilisateurs de LATEX. Ils permettent :

- d'aérer le texte ;
- d'améliorer sa lisibilité ;
- de structurer ses idées.

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 ;
   \item de structurer ses idées.
\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 source plus lisible.

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

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

donne :

1. premier élément ;

2. deuxième 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 automatiquement 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 comportent qu'une phrase) chaque élément doit débuter par une minuscule et se terminer par un point-virgule, sauf le dernier, qui doit se terminer par un point. S'il y en a, les éléments des sous-listes doivent se terminer par une virgule, sauf le dernier qui doit se terminer par un point-virgule.

- Pour les listes dont un élément contient au moins deux phrases, chaque élément doit débuter par une majuscule et se terminer par un point.

5.8 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 5.9).

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 de chaque colonne qui semblent les plus longs (dans la première ligne qui n'a pas été imprimée à cause de la commande `\kill`).

Par malchance, le mot « Deuxième », quand il est imprimé, est légèrement plus long que le mot « Troisième », bien qu'il comporte moins de lettres. Le placement des tabulations n'est donc pas idéal. Une raison de plus pour ne pas utiliser l'environnement `tabbing`.

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 60–62 et 201–203].

5.9 Les tableaux : l'environnement `tabular`

C'est de loin l'environnement le plus puissant de LaTeX. Avec lui, vous pourrez faire tous les tableaux possibles et imaginables (ou presque).

Commençons par un exemple simple :

```
\begin{tabular}{lcr}
Sparc & SunOS  & 4.1.4 \\
HP    & HP-UX  & 10.20 \\
PC    & NetBSD & 1.2.1
\end{tabular}
```

ce qui donne :

Sparc	SunOS	4.1.4
HP	HP-UX	10.20
PC	NetBSD	1.2.1

On a donné à l'environnement `tabular` un argument contenant trois caractères. Cela signifie que le tableau comporte trois colonnes. Détaillons cet argument :

`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 (LATEX 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  & 4.1.4 \\
\hline
HP    & HP-UX  & 10.20 \\
\hline
PC    & NetBSD & 1.2.1 \\
\hline
\end{tabular}
```

Sparc	SunOS	4.1.4
HP	HP-UX	10.20
PC	NetBSD	1.2.1

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 lettres définissant l'alignement des 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{5cm}||*{2}{c|}|}
\hline
                           & Contenance & Quantité \\
\hline
\hline
\bfseries Blanche de Bruges &    33 cl   &    10    \\
\hline
\bfseries Guinness         &   1 pint   &     5    \\
\hline
\bfseries Kronenbourg      &    33 cl   &     0    \\
\hline
\end{tabular}
```

	Contenance	Quantité
Blanche de Bruges	33 cl	10
Guinness	1 pint	5
Kronenbourg	33 cl	0

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

`p{`***taille***`}` qui déclare une colonne de taille *taille* (le contenu de la colonne est aligné à gauche) ;

`*{`***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 corps dans une case au moyen d'une déclaration n'affecte pas les autres cases.

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 62–63 et 204–207] et [2, chapitre 5].

5.10 Les environnements `table` et `figure`

Ces environnements ne sont que des cadres, permettant de faire *flotter* (c'est-à-dire de les déplacer si nécessaire pour la mise en page) tableaux et figures, de leur donner un titre et de les numéroter.

Il est important de pouvoir faire flotter les tableaux et les figures car LATEX ne les coupe jamais en bas de page et il est donc possible que l'un d'eux dépasse de la marge inférieure de la page où il se trouve, voire de la page elle-même, ce qui n'est pas du meilleur effet.

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

table	tableaux
figure	dessins

Tableau 5.1: 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***) sur une page spéciale ne contenant pas de texte mais uniquement des tableaux et des figures.

L'ordre de préférence par défaut est `[tbp]`, c'est pourquoi il est préférable de préciser `[htbp]` si l'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`.

Enfin, vous pouvez faire figurer dans votre document la liste des tableaux au moyen de la commande `\listoftables` et la liste des figures au moyen de la commande `\listoffigures`. Pour que ces listes soient à jour, il faut effectuer deux compilations successives (mettant en jeu les fichiers `.lot`[1] et

[1] `lot` pour *list of tables*.

`.lof`[2]).

[2] `lof` pour *list of figures*.

Chapitre 6

LaTeX mathématique

L'un des atouts de LaTeX est certainement la facilité avec laquelle il permet de réaliser les textes scientifiques les plus complexes. Son mode mathématique standard convient à la plupart des besoins mais il peut s'avérer nécessaire d'utiliser une extension spécialisée dans certains cas, en particulier l'extension `amstex` (voir [2, chapitre 8]). Il existe également de nombreuses extensions pour représenter des molécules, des diagrammes utilisés en physique des particules, des circuits électriques, etc. (voir [3, chapitre 6]).

Les règles typographiques d'écriture des mathématiques sont, pour la plupart, respectées automatiquement par LaTeX. Il n'est cependant pas inutile de se reporter à [6, 107–116] pour en prendre connaissance.

6.1 Les modes mathématiques

LaTeX connaît un très grand nombre de symboles 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, LaTeX 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`.

6.1.1 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 `$ ... $`.

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

6.1.2 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, `\[ ... \]` et `$$ ... $$`. On aurait donc pu écrire :

```
Ce qui conduit à
\[
x + y + z = n
\]
```

ou :

```
Ce qui conduit à
$$
x + y + z = n
$$
```

On utilise généralement la forme `$$ ... $$`.

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

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

6.2 Structures couramment utilisées en mathématiques

6.2.1 Indices et exposants

En mathématiques, on rencontre les indices et les exposants un peu partout.

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

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

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

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

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

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

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

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

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

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.

6.2.2 Texte à l'intérieur d'une formule

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

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

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

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

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

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

Bien entendu, la commande `\mbox` est inutile avec l'environnement `math` :

```
Et donc $ f(x) > 1 $ si $ x < 3 $.
```

Et donc $f(x) > 1$ si $x < 3$.

6.3 Lettres, symboles et accents

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

L'utilisation de certains symboles nécessite le chargement de l'extension latexsym. Il sera indiqué par la suite où cette option est nécessaire.

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

6.3.1 Lettres grecques

Les commandes permettant d'obtenir les lettres grecques sont présentées dans le tableau 6.1.

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.

Minuscules

α	`\alpha`	θ	`\theta`	π	`\pi`	ϕ	`\phi`
β	`\beta`	ϑ	`\vartheta`	ϖ	`\varpi`	φ	`\varphi`
γ	`\gamma`	ι	`\iota`	ρ	`\rho`	χ	`\chi`
δ	`\delta`	κ	`\kappa`	ϱ	`\varrho`	ψ	`\psi`
ϵ	`\epsilon`	λ	`\lambda`	σ	`\sigma`	ω	`\omega`
ε	`\varepsilon`	μ	`\mu`	ς	`\varsigma`		
ζ	`\zeta`	ν	`\nu`	τ	`\tau`		
η	`\eta`	ξ	`\xi`	υ	`\upsilon`		

Majuscules

Γ	`\Gamma`	Λ	`\Lambda`	Σ	`\Sigma`	Ψ	`\Psi`
Δ	`\Delta`	Ξ	`\Xi`	Υ	`\Upsilon`	Ω	`\Omega`
Θ	`\Theta`	Π	`\Pi`	Φ	`\Phi`		

Tableau 6.1: Lettres grecques

Il n'existe pas de commandes permettant d'obtenir les lettres grecques majuscules qui ont une graphie identique dans notre alphabet. Par exemple, pour obtenir un alpha majuscule, utilisez simplement `$A$`.

Pour la même raison, il n'existe pas de commandes permettant d'obtenir la lettre omicron, il suffit d'utiliser `$o$` ou `$O$`.

Pour certaines lettres minuscules, il existe une deuxième forme dont la commande commence par `var`. En particulier, je vous recommande d'utiliser `\varepsilon` et `\varphi` plutôt que `\epsilon` et `\phi`, les premiers étant plus jolis.

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.

6.3.2 Symboles d'opérateurs binaires

Les commandes présentées dans le tableau 6.2 permettent d'obtenir des symboles d'opérateurs binaires.

6.3.3 Symboles de relations binaires

Les commandes présentées dans le tableau 6.3 permettent d'obtenir des symboles de relations binaires.

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

$\pm$	`\pm`	$\cap$	`\cap`	$\diamond$	`\diamond`	$\oplus$	`\oplus`
$\mp$	`\mp`	$\cup$	`\cup`	$\bigtriangleup$	`\bigtriangleup`	$\ominus$	`\ominus`
$\times$	`\times`	$\uplus$	`\uplus`	$\bigtriangledown$	`\bigtriangledown`	$\otimes$	`\otimes`
$\div$	`\div`	$\sqcap$	`\sqcap`	$\triangleleft$	`\triangleleft`	$\oslash$	`\oslash`
$\ast$	`\ast`	$\sqcup$	`\sqcup`	$\triangleright$	`\triangleright`	$\odot$	`\odot`
$\star$	`\star`	$\vee$	`\vee`	$\lhd$	`\lhd`[a]	$\bigcirc$	`\bigcirc`
$\circ$	`\circ`	$\wedge$	`\wedge`	$\rhd$	`\rhd`[a]	$\dagger$	`\dagger`
$\bullet$	`\bullet`	$\setminus$	`\setminus`	$\unlhd$	`\unlhd`[a]	$\ddagger$	`\ddagger`
$\cdot$	`\cdot`	$\wr$	`\wr`	$\unrhd$	`\unrhd`[a]	$\amalg$	`\amalg`

[a] Requiert l'extension latexsym.

Tableau 6.2: Symboles d'opérateurs binaires

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

[a] Requiert l'extension latexsym.

Tableau 6.3: Symboles de relations binaires

```
$ x \not\in F $
```

$x \not\in F$

6.3.4 Flèches

Les commandes présentées dans le tableau 6.4 permettent d'obtenir des flèches.

$\leftarrow$	`\leftarrow`	$\longleftarrow$	`\longleftarrow`	$\uparrow$	`\uparrow`
$\Leftarrow$	`\Leftarrow`	$\Longleftarrow$	`\Longleftarrow`	$\Uparrow$	`\Uparrow`
$\rightarrow$	`\rightarrow`	$\longrightarrow$	`\longrightarrow`	$\downarrow$	`\downarrow`
$\Rightarrow$	`\Rightarrow`	$\Longrightarrow$	`\Longrightarrow`	$\Downarrow$	`\Downarrow`
$\leftrightarrow$	`\leftrightarrow`	$\longleftrightarrow$	`\longleftrightarrow`	$\updownarrow$	`\updownarrow`
$\Leftrightarrow$	`\Leftrightarrow`	$\Longleftrightarrow$	`\Longleftrightarrow`	$\Updownarrow$	`\Updownarrow`
$\mapsto$	`\mapsto`	$\longmapsto$	`\longmapsto`	$\nearrow$	`\nearrow`
$\hookleftarrow$	`\hookleftarrow`	$\hookrightarrow$	`\hookrightarrow`	$\searrow$	`\searrow`
$\leftharpoonup$	`\leftharpoonup`	$\rightharpoonup$	`\rightharpoonup`	$\swarrow$	`\swarrow`
$\leftharpoondown$	`\leftharpoondown`	$\rightharpoondown$	`\rightharpoondown`	$\nwarrow$	`\nwarrow`
$\rightleftharpoons$	`\rightleftharpoons`	$\leadsto$	`\leadsto`[a]		

[a] Requiert l'extension latexsym.

Tableau 6.4: Flèches

6.3.5 Symboles divers

Les commandes présentées dans le tableau 6.5 permettent d'obtenir des symboles divers.

$\aleph$	`\aleph`	$\prime$	`\prime`	$\forall$	`\forall`	∞	`\infty`
$\hbar$	`\hbar`	$\emptyset$	`\emptyset`	$\exists$	`\exists`	$\Box$	`\Box`[a]
$\imath$	`\imath`	∇	`\nabla`	$\neg$	`\neg`	$\Diamond$	`\Diamond`[a]
$\jmath$	`\jmath`	$\surd$	`\surd`	$\flat$	`\flat`	$\triangle$	`\triangle`
ℓ	`\ell`	$\top$	`\top`	$\natural$	`\natural`	$\clubsuit$	`\clubsuit`
$\wp$	`\wp`	$\bot$	`\bot`	$\sharp$	`\sharp`	$\diamondsuit$	`\diamondsuit`
$\Re$	`\Re`	$\|$	`|`	$\backslash$	`\backslash`	$\heartsuit$	`\heartsuit`
$\Im$	`\Im`	$\angle$	`\angle`	∂	`\partial`	$\spadesuit$	`\spadesuit`
$\mho$	`\mho`[a]						

[a] Requiert l'extension latexsym.

Tableau 6.5: Symboles divers

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

6.3.6 Accents

Le tableau 6.6 présente les accents disponibles en mode mathématique.

$\hat{a}$	`\hat{a}`	$\acute{a}$	`\acute{a}`	$\bar{a}$	`\bar{a}`	$\dot{a}$	`\dot{a}`
$\check{a}$	`\check{a}`	$\grave{a}$	`\grave{a}`	$\vec{a}$	`\vec{a}`	$\ddot{a}$	`\ddot{a}`
$\breve{a}$	`\breve{a}`	$\tilde{a}$	`\tilde{a}`				

Tableau 6.6: 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} $
```

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

Les commandes `\overrightarrow` et `\overleftarrow` permettent de surmonter leur argument d'une grande flèche :

```
$ \overrightarrow{AB} $, $\overleftarrow{AB} $
```

$\overrightarrow{AB}$, $\overleftarrow{AB}$

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` :

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

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

6.3.7 Superposer deux symboles

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

`\stackrel{`*dessus*`}{`*dessous*`}`

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

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

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

```
$ \stackrel{\Rightarrow}{\sigma} =
  \Lambda ( \stackrel{\Rightarrow}{\varepsilon} ) $
```

$\stackrel{\Rightarrow}{\sigma} = \Lambda (\stackrel{\Rightarrow}{\varepsilon})$

6.4 Modifier l'espacement

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

`\,`	petite espace	`\!`	petite espace négative
`\:`	moyenne espace	`\quad`	petite espace interformules
`\;`	grande espace	`\qquad`	moyenne espace interformules

Tableau 6.7: Commandes d'espacement

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

6.5 Modifier le style et la taille

LaTeX permet de modifier le style des caractères en mode mathématique (voir le tableau 6.8).

Comme vous pouvez le constater, les commandes de changement de style n'agissent que sur les lettres, les nombres et les lettres grecques majuscules.

La commande `\mathcal` permet d'obtenir des majuscules calligraphiques :

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

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

`$ x + y + 2^n\Omega\cos\theta $`	$x + y + 2^n\Omega\cos\theta$
`$\mathit{x + y + 2^n\Omega\cos\theta}$`	$\mathit{x + y + 2^n\Omega\cos\theta}$
`$\mathrm{x + y + 2^n\Omega\cos\theta}$`	$\mathrm{x + y + 2^n\Omega\cos\theta}$
`$\mathbf{x + y + 2^n\Omega\cos\theta}$`	$\mathbf{x + y + 2^n\Omega\cos\theta}$
`$\mathsf{x + y + 2^n\Omega\cos\theta}$`	$\mathsf{x + y + 2^n\Omega\cos\theta}$
`$\mathtt{x + y + 2^n\Omega\cos\theta}$`	$\mathtt{x + y + 2^n\Omega\cos\theta}$

Tableau 6.8: Commandes de changement de style

Voici ce à quoi ressemblent ces majuscules :

$$\mathcal{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$$

Comme vous allez le voir par la suite, certains symboles n'ont pas la même taille dans l'environnement `math` et dans l'environnement `displaymath`. La commande `\displaystyle` permet, dans un environnement `math` par exemple, de forcer LATEX à utiliser des caractères et des symboles de la même taille qu'en environnement `displaymath`. Cette commande est à utiliser avec modération, car son utilisation risque de modifier l'interligne du paragraphe, pour les formules qui sont placées dans le texte.

De même, la commande `\textstyle` permet de forcer LATEX à utiliser des caractères et des symboles de la même taille qu'en environnement `math`.

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).

6.6 Fractions

Les fractions s'obtiennent grâce à la commande :

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

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

donne :

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

6.7 Racines

Les racines s'obtiennent grâce à la commande :

\sqrt[*ordre*]{*argument*}

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

donne :

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

6.8 Délimiteurs

Les *délimiteurs*, présentés dans le tableau 6.9, sont des symboles permettant d'encadrer des formules. Leur hauteur s'adapte à celle de la formule.

$($	`(`	$)$	`)`	$\uparrow$	`\uparrow`
$[$	`[`	$]$	`]`	$\downarrow$	`\downarrow`
$\{$	`\{`	$\}$	`\}`	$\updownarrow$	`\updownarrow`
$\lfloor$	`\lfloor`	$\rfloor$	`\rfloor`	$\Uparrow$	`\Uparrow`
$\lceil$	`\lceil`	$\rceil$	`\rceil`	$\Downarrow$	`\Downarrow`
$\langle$	`\langle`	$\rangle$	`\rangle`	$\Updownarrow$	`\Updownarrow`
$/$	`/`	$\backslash$	`\backslash`		
$\vert$	`|`	$\Vert$	`\\|`		

Tableau 6.9: Délimiteurs

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.` (`\right` suivi d'un point). En effet, tout `\left` doit avoir un `\right` correspondant, le `\right.` n'imprime rien, mais sert à fermer le `\left` qui lui correspond.

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

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

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

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

6.9 Symboles de taille variable

Les symboles présentés dans le tableau 6.10 sont de *taille variable* : il n'ont pas la même taille dans l'environnement `math` et dans l'environnement `displaymath`.

$\textstyle\sum \displaystyle\sum$	`\sum`	$\textstyle\bigcap \displaystyle\bigcap$	`\bigcap`	$\textstyle\bigodot \displaystyle\bigodot$	`\bigodot`
$\textstyle\prod \displaystyle\prod$	`\prod`	$\textstyle\bigcup \displaystyle\bigcup$	`\bigcup`	$\textstyle\bigotimes \displaystyle\bigotimes$	`\bigotimes`
$\textstyle\coprod \displaystyle\coprod$	`\coprod`	$\textstyle\bigsqcup \displaystyle\bigsqcup$	`\bigsqcup`	$\textstyle\bigoplus \displaystyle\bigoplus$	`\bigoplus`
$\textstyle\int \displaystyle\int$	`\int`	$\textstyle\bigvee \displaystyle\bigvee$	`\bigvee`	$\textstyle\biguplus \displaystyle\biguplus$	`\biguplus`
$\textstyle\oint \displaystyle\oint$	`\oint`	$\textstyle\bigwedge \displaystyle\bigwedge$	`\bigwedge`		

Tableau 6.10: Symboles de taille variable

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 (qui représentent leurs bornes) ne sont pas disposés de la même façon dans les deux environnements :

```
$  \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} $.
```

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

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

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

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

En cas d'usage intensif d'intégrales multiples, il peut être intéressant d'utiliser l'extension `amstex` (voir [2, chapitre 8]) qui définit des commandes spécifiques pour la saisie des intégrales multiples (voir [2, page 223]).

6.10 Fonctions et opérateurs

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

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

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 romain. Essayons donc ainsi :

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

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

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, LaTeX fournit les commandes présentées dans le tableau 6.11.

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

Tableau 6.11: 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 6.11 se comportent comme des symboles de taille variable en ce qui concerne le placement de leurs indices. 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} $
```

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

6.11 Les tableaux : l'environnement `array`

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

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

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

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

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

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

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

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

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

6.12 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}
```

$$e^{i\pi} + 1 = 0 \tag{6.1}$$

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 utilisant l'option de classe `leqno`

6.13 Les équations alignées : l'environnement `eqnarray`

Il peut être nécesssaire 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 \: \exp y
\end{eqnarray}
```

$$\ln xy = \ln x + \ln y \tag{6.2}$$

$$\exp(x+y) = \exp x \: \exp y \tag{6.3}$$

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}
```

$$\begin{aligned}\int_1^2 x^2 dx &= \left[\frac{x^3}{3}\right]_1^2 \\ &= \frac{2^3}{3} - \frac{1^3}{3} \\ &= \frac{8}{3} - \frac{1}{3} \\ &= \frac{7}{3} \qquad (6.4)\end{aligned}$$

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 il ne numérote pas les équations :

```
\begin{eqnarray*}
\int_0^\pi \sin x \, dx & = & \left[ - \cos x \right]_0^\pi \\
                        & = & - \cos \pi + \cos 0           \\
                        & = & 1 + 1                         \\
                        & = & 2
\end{eqnarray*}
```

$$\begin{aligned}\int_0^\pi \sin x \, dx &= \left[-\cos x\right]_0^\pi \\ &= -\cos\pi + \cos 0 \\ &= 1+1 \\ &= 2\end{aligned}$$

6.14 Théorèmes, définitions...

Théorème 1 (babafou) *LaTeX ne se limite pas aux formules et aux signes bizarres, 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
bizarres, 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, chacun ayant son propre compteur.

Chapitre 7

Bibliographie et index

Réaliser une bibliographie ou un index est généralement assez compliqué. LaTeX s'appuie sur deux autres programmes, BibTeX et *MakeIndex*, afin de rendre ce travail simple et efficace.

7.1 Faire une bibliographie

Avec LaTeX, il y a deux façons de faire une bibliographie :

- utiliser l'environnement `thebibliography` ;
- utiliser le programme BibTeX.

7.1.1 L'environnement `thebibliography`

Voici un exemple de bibliographie réalisée grâce à l'environnement `thebibliography` :

```
\clearpage

\addcontentsline{toc}{chapter}{Bibliographie}

\begin{thebibliography}{9}
\bibitem{lamport-latex}
   Leslie Lamport.
   \emph{\LaTeX: a document preparation system}.
   Addison-Wesley, Reading, Massachusetts,
   deuxième édition, 1994.
\bibitem{companion}
   Michel Goossens, Frank Mittelbach et Alexander Samarin.
   \emph{The \LaTeX\ Companion}.
   Addison-Wesley, Reading, Massachusetts,
   1994.
\end{thebibliography}
```

Ce qui donne :

Bibliographie

[1] Leslie Lamport. *LaTeX: a document preparation system.* Addison-Wesley, Reading, Massachusetts, deuxième édition, 1994.

[2] Michel Goossens, Frank Mittelbach et Alexander Samarin. *The LaTeX Companion.* Addison-Wesley, Reading, Massachusetts, 1994.

La commande `\clearpage` permet d'imprimer les tables et les figures qui n'ont pas encore pu l'être, et de commencer 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 la classe `report`, `section` pour la classe `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[Lamport 94]{lamport-latex} Leslie Lamport, ...
```

Ce qui donne :

[Lamport 94] Leslie Lamport, ...

L'argument de l'environnement `thebibliography` précise la taille maxinale 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 209]{lamport-latex} pour plus de
précisions.
```

Voir [1, page 209] pour plus de précisions.

7.1.2 BibTeX

L'environnement `thebibliography` convient parfaitement à la réalisation d'une bibliographie, mais il a certains inconvénients :

- il oblige à refaire une bibliographie pour chaque document, même si seulement quelques ouvrages diffèrent ;
- en cas de mise à jour d'un ouvrage, il faut modifier les bibliographies de tous les documents où il y est fait référence.

D'où l'idée, plutôt que d'avoir une bibliographie indépendante pour chaque document, de faire des bases de données bibliographiques et de générer automatiquement la bibliographie à partir de celles-ci en fonction des ouvrages dont il est fait référence dans le document.

7.1.2.1 La base de données bibliographique

Une base de données bibliographique est un fichier d'extension `.bib` qui contient un certain nombre d'entrées comme celle-ci :

```
@BOOK
{
   lamport-latex ,
   AUTHOR    = "Leslie Lamport" ,
   TITLE     = "{\LaTeX}: a document preparation system" ,
   PUBLISHER = "Addison-Wesley" ,
   ADDRESS   = "Reading, Massachusetts" ,
   EDITION   = "deuxième" ,
   YEAR      = "1994"
}
```

Chaque entrée correspond à un ouvrage dont le type est indiqué en premier, juste après le caractère `@`. Il s'agit ici d'un livre, donc de type `BOOK`.

On indique ensuite une clé, qui servira à repérer l'ouvrage dans vos bases de données bibliographiques, en tant qu'argument de la commande `\cite`. Le format de la clé est libre mais j'utilise généralement le nom de l'auteur, suivi de `-` et d'un mot permettant de caractériser l'ouvrage.

ATTENTION : les clés doivent être uniques au sein de toutes vos bases de données bibliographiques.

Viennent ensuite un certain nombre de champs qui définissent l'ouvrage (auteur, titre, etc.). Pour un type d'ouvrage donné, certains champs doivent obligatoirement être indiqués et d'autres sont facultatifs. Chaque champ est de la forme

`NOM_DU_CHAMP = "`*contenu du champ*`"`

Les champs sont séparés les uns des autres et de la clé par des virgules.

7.1.2.2 Les différents types d'entrées

Chaque entrée contient un ou plusieurs champs. Un champ peut être :

obligatoire
: Ce champ est nécessaire à BIBTEX pour générer la bibliographie.

optionnel
: Ce champ n'est pas nécessaire à BIBTEX pour générer la bibliographie, mais, s'il est indiqué, il pourra éventuellement l'utiliser pour la compléter.

ignoré
: Ce champ est inutile et son contenu ne sera jamais utilisé par BIBTEX pour générer la bibliographie.

Voici la liste des types d'entrées reconnus par BIBTEX. Pour chaque type sont indiqués les champs obligatoires et optionnels (les autres étant ignorés) dont la signification sera expliquée dans le paragraphe suivant.

`ARTICLE`
: Article paru dans un journal ou un magazine.
Champs obligatoires : `AUTHOR`, `TITLE`, `JOURNAL`, `YEAR`.
Champs optionnels : `VOLUME`, `NUMBER`, `PAGES`, `MONTH`, `NOTE`.

`BOOK`
: Livre avec un éditeur.
Champs obligatoires : `AUTHOR` ou `EDITOR`, `TITLE`, `PUBLISHER`, `YEAR`.
Champs optionnels : `VOLUME` ou `NUMBER`, `SERIES`, `ADDRESS`, `EDITION`, `MONTH`, `NOTE`.

`BOOKLET`
: Ouvrage qui est imprimé et distribué, mais sans éditeur.
Champ obligatoire : `TITLE`.
Champs optionnels : `AUTHOR`, `HOWPUBLISHED`, `ADDRESS`, `MONTH`, `YEAR`, `NOTE`.

`CONFERENCE`
: Identique à `INPROCEEDINGS`.

`INBOOK`
: Partie d'un livre, généralement sans titre, un chapitre, un paragraphe ou quelques pages.
Champs obligatoires : `AUTHOR` ou `EDITOR`, `TITLE`, `CHAPTER` et/ou `PAGES`, `PUBLISHER`, `YEAR`.
Champs optionnels : `VOLUME` ou `NUMBER`, `SERIES`, `TYPE`, `ADDRESS`, `EDITION`, `MONTH`, `NOTE`.

`INCOLLECTION`
: Partie d'un livre avec son propre titre.
Champs obligatoires : `AUTHOR`, `TITLE`, `BOOKTITLE`, `PUBLISHER`, `YEAR`.
Champs optionnels : `EDITOR`, `VOLUME` ou `NUMBER`, `SERIES`, `TYPE`, `CHAPTER`, `PAGES`, `ADDRESS`, `EDITION`, `MONTH`, `NOTE`.

`INPROCEEDINGS`
: Article d'actes de conférence.
Champs obligatoires : `AUTHOR`, `TITLE`, `BOOKTITLE`, `YEAR`.
Champs optionnels : `EDITOR`, `VOLUME` ou `NUMBER`, `SERIES`, `PAGES`, `ADDRESS`, `MONTH`, `ORGANIZATION`, `PUBLISHER`, `NOTE`.

`MANUAL`
: Documentation technique.
Champ obligatoire : `TITLE`.
Champs optionnels : `AUTHOR`, `ORGANIZATION`, `ADDRESS`, `EDITION`, `MONTH`, `YEAR`, `NOTE`.

`MASTERSTHESIS`
: Traité de maîtrise.
Champs obligatoires : `AUTHOR`, `TITLE`, `SCHOOL`, `YEAR`.
Champs optionnels : `TYPE`, `ADDRESS`, `MONTH`, `NOTE`.

`MISC`
: C'est le type à utiliser quand rien d'autre ne convient.
Champs obligatoires : aucun.
Champs optionnels : `AUTHOR`, `TITLE`, `HOWPUBLISHED`, `MONTH`, `YEAR`, `NOTE`.

`PHDTHESIS`
: Thèse de doctorat.
Champs obligatoires : `AUTHOR`, `TITLE`, `SCHOOL`, `YEAR`.
Champs optionnels : `TYPE`, `ADDRESS`, `MONTH`, `NOTE`.

`PROCEEDINGS`
: Actes d'une conférence.
Champs obligatoires : `TITLE`, `YEAR`.

Champs optionnels : `EDITOR`, `VOLUME` ou `NUMBER`, `SERIES`, `ADDRESS`, `MONTH`, `ORGANIZATION`, `PUBLISHER`, `NOTE`.

`TECHREPORT`
: Rapport publié par une école ou une institution.
Champs obligatoires : `AUTHOR`, `TITLE`, `INSTITUTION`, `YEAR`.
Champs optionnels : `TYPE`, `NUMBER`, `ADDRESS`, `MONTH`, `NOTE`.

`UNPUBLISHED`
: Document avec un auteur et un titre, mais qui n'a jamais été publié.
Champs obligatoires : `AUTHOR`, `TITLE`, `NOTE`.
Champs optionnels : `MONTH`, `YEAR`.

7.1.2.3 Les différents types de champs

`ADDRESS`
: Adresse de l'éditeur (`PUBLISHER`) ou de l'institution (`INSTITUTION`).

`ANNOTE`
: Annotation.

`AUTHOR`
: Nom de l'auteur (ou les noms des auteurs, dans le format décrit plus loin).

`BOOKTITLE`
: Titre d'une partie d'un livre ou d'un article dans les actes d'une conférence.

`CHAPTER`
: Numéro de chapitre (ou d'une quelconque partie).

`CROSSREF`
: Clé de l'élément vers lequel on veut établir une référence.

`EDITION`
: Édition d'un livre.

`EDITOR`
: Nom du rédacteur.

`HOWPUBLISHED`
: Façon dont l'ouvrage a été publié.

`INSTITUTION`
: Institution éditrice d'un rapport technique.

`JOURNAL`
: Nom d'un journal.

`KEY`
Utilisé pour créer une étiquette s'il n'y a pas d'auteur.

`MONTH`
Mois durant lequel l'ouvrage a été publié.

`NOTE`
Informations supplémentaires pouvant aider le lecteur.

`NUMBER`
Numéro d'un journal, d'un magazine.

`ORGANIZATION`
Organisation qui patronne une conférence ou publie un manuel.

`PAGES`
Un numéro de page ou plusieurs.

`PUBLISHER`
Nom de l'éditeur.

`SCHOOL`
Nom de l'école ou de l'université dans laquelle la thèse a été réalisée.

`SERIES`
Nom d'un ensemble de livres.

`TITLE`
Titre de l'ouvrage.

`TYPE`
Type d'un rapport technique.

`VOLUME`
Volume d'un journal ou d'un livre.

`YEAR`
Année de publication.

7.1.2.4 Quelques précisions sur certains champs

Si un champ contient une commande LaTeX, celle-ci doit être entourée d'accolades afin de ne pas dérouter BibTeX :

```
TITLE = "{\LaTeX}: a document preparation system"
```

Lorsqu'un champ `AUTHOR` ou `EDITOR` contient plusieurs noms, ils doivent être séparés les uns des autres par le mot `and`.

7.1.2.5 Générer la bibliographie avec BibTeX

Une fois que vous avez créé vos bases de données bibliographiques, le plus gros du travail est fait.

Maintenant, lorsque vous voulez faire référence à un ouvrage, utilisez la commande `\cite` avec pour argument la clé correspondante :

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

Il faut aussi indiquer à LaTeX où placer la bibliographie, au moyen de la commande `\bibliography`. Cette command prend en argument une liste de noms de fichiers de base de données bibliographique (mais sans l'extension `.bib`) :

```
\bibliography{fichier1,fichier2,fichier3}
```

Bien entendu, tous les ouvrages auxquels ils est fait référence par la commande `\cite` doivent se trouver dans ces fichiers.

Il faut également indiquer à BibTeX le style à utiliser pour mettre en page la bibliographie, grâce à la commande `\bibliographystyle`. L'argument de cette commande est le nom du style bibliographique à utiliser.

Les styles bibliographiques standard de LaTeX sont :

`plain`
: Les ouvrages sont triées dans l'ordre alphabétique et sont étiquetés par des nombres.

`unsrt`
: Identique au style `plain`, mais les ouvrages sont triés dans l'ordre de leur citation.

`alpha`
: Identique au style `plain`, mais les ouvrages sont étiquetés par des chaînes de caractères du style « Lam94 », formées à partir du nom de l'auteur et de l'année de publication.

`abbrv`
: Identique au style `plain`, mais les prénoms, noms de mois et de journaux sont abrégés.

Ils existe de nombreux autres styles bibliographiques.

Il faut aussi éventuellement inclure l'entrée de la bibliographie dans la table des matières au moyen de la commande `\addcontentsline`, ce qui n'est fait automatiquement ni par LaTeX ni par BibTeX :

```
\addcontentsline{toc}{chapter}{Bibliographie}
```

En résumé, mis à part les commandes `\cite` disséminées dans le document, l'inclusion de la bibliographie ressemble à ceci :

```
\addcontentsline{toc}{chapter}{Bibliographie}
\bibliographystyle{unsrt}
\bibliography{fichier1,fichier2,fichier3}
```

Si vous voulez inclure dans la bibliographie des ouvrages auxquels vous ne faites pas référence dans votre document, utilisez la commande `\nocite` (placée par exemple juste avant la commande `\bibliographystyle`), avec en argument la liste des clés de ces ouvrages, séparées par des virgules :

```
\nocite{lamport-latex,companion}
```

Il est aussi possible d'inclure dans la bibliographie tous les ouvrages répertoriés dans les bases de données bibliographiques en mettant `*` comme argument de `\nocite` :

```
\nocite{*}
```

Enfin, pour générer la bibliographie et l'inclure dans votre document, il faut :

- Compiler votre document avec LaTeX :

  ```
  latex document.tex
  ```

- Exécuter BibTeX sur votre document :

  ```
  bibtex document
  ```

  (ne mettez pas l'extension `.tex`, BibTeX n'aime pas ça)
  BibTeX génère alors un fichier `document.bbl`, qui contient la bibliographie mise en page selon le style indiqué par la commande `\bibliographystyle`.

- Recompiler deux fois votre document avec LaTeX afin d'y inclure la bibliographie et de mettre toutes les références à jour.

Pour plus d'informations au sujet de BibTeX, consultez [4], [1, pages 69–71 et 155–164] et [2, chapitre 13].

7.2 Faire un index

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

Avant cela, il faut modifier un peu votre document :

- chargez l'extension makeidx ;
- ajoutez la commande `\makeindex` dans le préambule ;

- mettez la commande `\printindex` où vous voulez voir apparaître l'index (c'est généralement à la fin du document, juste avant la table des matières).

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 7.2.2. Vous pouvez également vous reporter à [5], [1, pages 149–154] et [2, pages 345–370] pour une description plus complète de la réalisation d'un index avec LaTeX et *MakeIndex*.

7.2.1 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.
```

7.2.2 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 voici :

```
headings_flag          1
heading_prefix         "\n  \\item \\textbf{"
heading_suffix         "}"
symhead_positive       "Symboles"
symhead_negative       "symboles"
numhead_positive       "Nombres"
numhead_negative       "nombres"
delim_0                " \\dotfill\\ "
delim_1                " \\dotfill\\ "
delim_2                " \\dotfill\\ "
```

Maintenant, appelez *MakeIndex* ainsi :

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

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

Chapitre 8

Réaliser des transparents : la classe slides

La classe slides permet de réaliser des transparents très simples. Le corps du document contient un ou plusieurs environnements `slide`, dont chacun contient le texte d'un transparent :

```
\documentclass[a4paper]{slides}

\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[francais]{babel}

\begin{document}

\begin{slide}
Voici un transparent réalisé avec la classe slides.
Le texte est écrit en gros caractères, dans une
police antique et centré verticalement.
\end{slide}

\end{document}
```

Ce document produit un unique transparent (puisqu'il n'y a qu'un seul environnement `slide`, mais on aurait pu en mettre plusieurs à la suite). Vous pouvez en voir le résultat dans la figure 8.1.

Dans l'environnement `slide`, le texte est :

- écrit en gros caractères ;
- écrit dans une police antique ;
- centré verticalement.

Voici un transparent réalisé avec la classe slides. Le texte est écrit en gros caractères, dans une police antique et centré verticalement.

1

Fig. 8.1: Exemple de transparent réalisé avec la classe slides

et chaque page est numérotée dans le coin inférieur droit.

Vous pouvez utiliser avec la classe slides la plupart des commandes et des environnements LaTeX qui ont un sens dans ce cas (par exemple, des commandes comme les commandes de hiérarchisation ne servent pas dans les transparents et n'existent donc pas dans la classe slides). Je n'ai pas de liste des commandes LaTeX non disponibles dans la classe slides donc, dans le doute, essayez !

Pour plus de renseignements sur la classe slides, reportez-vous à [1, pages 80–84].

Chapitre 9

Écrire des lettres : la classe letter

Nous avons vu dans les chapitres précédents que LaTeX était très fort pour mettre en forme des documents complexes, avec une table des matières ou des formules mathématiques, mais il permet aussi de réaliser de simples lettres, au moyen de la classe letter. Le corps du document contient un ou plusieurs environnements `letter`, dont chacun contient le texte d'une lettre.

9.1 Les commandes du préambule

Certaines déclarations sont globales pour toutes les lettres d'un même document et les commandes correspondantes sont donc utilisées dans le préambule. Il s'agit des commandes `\address` et `\signature`.

9.1.1 La commande \address

La commande `\address` permet de définir votre adresse postale, qui sera imprimée en haut à droite de la lettre. Les différentes lignes de l'adresse sont séparées par \\ :

```
\address{babafou \\ Pays des babasses}
```

9.1.2 La commande \signature

La commande `\signature` permet de définir votre signature, qui sera imprimée en bas à droite de la lettre. Les différentes lignes de la signature sont séparées par \\ :

```
\signature{Ton petit babafou \\ qui t'aime}
```

9.2 L'environnement letter

Chaque lettre est contenue dans un environnement `letter` dont l'argument est l'adresse du destinataire :

```
\documentclass[a4paper,11pt]{letter}

\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[francais]{babel}

\address{babafou \\ Pays des babasses}

\signature{Ton petit babafou \\ qui t'aime}

\begin{document}

\begin{letter}{Père Noël \\ Dans les nuages}

\opening{Cher Père Noël,}

comme j'ai été bien sage toute l'année, je voudrais pour
Noël :
\begin{itemize}
   \item une babasse ;
   \item \LaTeX\ ;
   \item le manuel \emph{Apprends \LaTeX\ !} ;
   \item la bible de \LaTeX, le \emph{\LaTeX\ Companion}.
\end{itemize}

\closing{Joyeux Noël,}

\end{letter}

\end{document}
```

La commande `\opening` permet de commencer la lettre proprement dite et la commande `\closing` la termine par une formule de politesse. Entre ces deux commandes, se situe le texte de la lettre qui peut contenir n'importe quelle commande LaTeX (à l'exception des commandes qui n'ont pas de sens dans une lettre, comme `\section`, par exemple).

Le document peut contenir plusieurs lettres, chacune contenue dans un environnement `letter`. Toutes ces lettres auront la même adresse d'expéditeur et la même signature (commandes `\address` et `\signature` du préambule), mais il est possible de définir une adresse d'expéditeur ou une signature différente pour une lettre particulière en utilisant les commandes `\address` et `\signature` entre `\begin{letter}` et `\opening`.

Enfin, la commande `\makelabels`, utilisée dans le préambule, permet de générer une page spéciale, contenant uniquement les adresses des destinataires. Si l'on imprime cette page sur une feuille de papier autocollant, on a ainsi des étiquettes que l'on peut coller sur les enveloppes.

Pour plus de renseignements sur la classe letter, reportez-vous à [1, pages 84–86].

Chapitre 10

LaTeX avancé

Ce chapitre explique comment réaliser des fonctions avancées avec LaTeX. En particulier, vous allez apprendre à définir vos propres commandes, ce qui essentiel à la mise en œuvre de la philosophie LaTeX.

10.1 Définir de nouvelles commandes et de nouveaux environnements

10.1.1 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 occurence de `\nom` sera remplacée par *définition* avec substitution des éventuels arguments.

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 » :

```
C'est chouette la \ba\ !
```

C'est chouette la babasse !

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{\tenseur}[1]{\stackrel{\Rightarrow}{#1}}
```

La commande `\tenseur` permet d'imprimer une lettre surmontée d'une double flèche (bien entendu, elle n'a de sens qu'en mode mathématique) :

```
$$
\tenseur{\sigma} = \Lambda ( \tenseur{\varepsilon} )
$$
```

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

La commande `\tenseur` 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.).

10.1.2 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 LATEX 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} \itshape}
{\end{itemize}}
```

Ici, j'ai réparti la définition du nouvel environnement sur trois lignes pour la rendre plus lisible.

```
\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 `\newenvironnement` 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

10.2 Inclure des dessins avec *xfig*

Le programme de dessin vectoriel *xfig* permet de réaliser toutes sortes de dessins, figures et autres schémas.

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 LaTeX.

10.2.1 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 la fenêtre de sauvegarde ;
- donnnez 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 la fenêtre d’exportation ;
- puis cliquez sur `Export`.

Le dessin est alors enregistré en PostScript encapsulé avec l’extension `.eps`.

10.2.2 Inclure un dessin dans un document LaTeX

Afin d’inclure un fichier au format PostScript encapsulé dans un document LaTeX, il faut tout d’abord charger l’extension graphicx :

```
\usepackage{graphicx}
```

Pour afficher votre dessin, utilisez la commande

\includegraphics{*dessin.eps*}

Il est préférable de mettre le dessin dans une figure, ce qui 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 au moyen de l'environnement `center`.

On peut aussi indiquer la taille que l'on désire donner au dessin, en largeur :

\includegraphics[width=*largeur*]{*dessin.eps*}

ou en hauteur :

\includegraphics[height=*hauteur*]{*dessin.eps*}

Voici donc le code type d'inclusion d'un dessin au format PostScript encapsulé dans un document LATEX :

```
\documentclass[a4paper,11pt]{article}

\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[francais]{babel}
\usepackage{graphicx}

% ...

\begin{figure}[htbp]
\begin{center}
\includegraphics[width=10cm]{dessin.eps}
\end{center}
\caption{Dessin}
\label{fig:dessin}
\end{figure}
```

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

Vous avez dû remarquer que la commande `\maketitle` génère une page de garde plutôt minimale. C'est dommage qu'un formateur de texte de la qualité de LATEX 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 `\includegraphics` (comme j'ai fait pour le logo ENSTA sur la page de garde de ce document) ;

Seule contrainte, l'environnement `titlepage` doit se trouver juste après la commande `\begin{document}`.

Annexe A

Différences entre LaTeX 2.09 et LaTeX 2$_\varepsilon$

Les différences entre LaTeX 2.09 (version maintenant obsolète, décrite dans la première édition de ce manuel) et LaTeX 2$_\varepsilon$ (version décrite dans ce manuel) sont relativement minimes pour l'utilisateur (elles le sont en revanche beaucoup moins pour les concepteurs d'extensions). Cette annexe résume les principales différences entre ces deux versions.

A.1 Les commandes `\documentstyle`, `\documentclass` et `\usepackage`

La commande `\documentstyle` de LaTeX 2.09 a été remplacée en LaTeX 2$_\varepsilon$ par `\documentclass`. Les options de classe (arguments optionnels de `\documentclass`) correspondent aux options de style (arguments optionnels de `\documentstyle`) qui s'appliquent à la classe du document. Les autres options de styles (qu'on appelle désormais extensions) sont chargées grâce à la commande `\usepackage`.

A.2 Commandes de changement de style

Les commandes `\rm`, `\sf`, `\tt`, `\it`, `\sl`, `\sc` et `\bf` ont été remplacées respectivement par `\rmfamily`, `\sffamily`, `\ttfamily`, `\itshape`, `\slshape`, `\scshape` et `\bfseries`.

La commande `\em` a été remplacée par `\emph`. Les commandes `\textrm`, `\textsf`, `\texttt`, `\textit`, `\textsl`, `\textsc` et `\textbf` sont nouvelles en LaTeX 2$_\varepsilon$.

Ces nouvelles commandes prennent un argument et ne nécessitent donc pas d'accolades de groupement. De plus, il est possible de mélanger différents styles :

```
\textit{italique \textbf{gras}}
```

*italique **gras***

alors qu'en LaTeX 2.09, chaque commande annulait l'effet de la précédente :

```
{\it italique \bf gras}
```

italique **gras**

A.3 SLITEX et la classe slides

Le programme SLITEX n'existe plus, les transparents sont maintenant directement produits par LaTeX au moyen de la classe slides.

A.4 La commande `\LaTeXe`

Cette commande permet simplement d'imprimer le logo LaTeX 2ε (avec le 2ε). De même que pour la commande `\LaTeX`, il faut la faire suivre de `\␣` si l'on ne veut pas que le mot suivant lui soit collé.

Annexe B

Quelques règles typographiques françaises et anglaises

De même que l'écriture d'une phrase nécessite le respect de l'orthographe et de la grammaire, la composition d'un document nécessite le respect des règles typographiques. Certaines règles typographiques sont automatiquement respectées par LaTeX mais, pour réaliser des documents de bonne qualité, vous devez vous-même respecter toutes les autres. À cet effet, je vous recommande la lecture de l'excellent *Lexique des règles typographiques en usage à l'Imprimerie nationale* [6].

Les règles typographiques sont, pour la plupart, différentes selon les langues. Ici sont résumées et illustrées quelques règles élémentaires pour le français et l'anglais.

B.1 Règles typographiques françaises

- On met toujours les accents sur les majuscules.
- La virgule et le point sont collés au mot qui les précède et sont suivis par une espace.
- Le point-virgule, les deux points, le point d'exclamation et le point d'interrogation sont précédés et suivis d'une espace.
- Les points de suspension sont collés au mot qui les précède.
- Les « guillemets français » ne sont pas présents sur les claviers, mais ils se saisissent facilement sous Emacs grâce aux commandes `C-x 8 <` et `C-x 8 >`.

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 ?

B.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 une 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'une 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 `\␣` pour l'espace suivant les points de suspension).
- Les guillemets se saisissent `` et '' (et pas "). Ce qui donne “ceci”.

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!

Bibliographie

[1] Leslie Lamport. *LaTeX: a document preparation system*. Addison-Wesley, Reading, Massachusetts, deuxième édition, 1994.

[2] Michel Goossens, Frank Mittelbach et Alexander Samarin. *The LaTeX Companion*. Addison-Wesley, Reading, Massachusetts, 1994.

[3] Michel Goossens, Sebastian Rahtz et Frank Mittelbach. *The LaTeX Graphics Companion*. Addison-Wesley, Reading, Massachusetts, 1997.

[4] Oren Patashnik. BibT*E*X*ing*, février 1988.

[5] Leslie Lamport. *MakeIndex: An Index Processor For LaTeX*, février 1987.

[6] Imprimerie nationale. *Lexique des règles typographiques en usage à l'Imprimerie nationale*, 1990.

[7] Yves Perrousseaux. *Manuel de Typographie française élémentaire*. Atelier Perrousseaux éditeur, 1995.

Index

Liste des tableaux

Liste des figures

Table des matières