

Marc Baudoin

Apprends L^AT_EX !

quatrième édition

Copyright © 1994–2010 Marc Baudoin

Ce document est mis à disposition selon les termes de la License publique
du projet L^AT_EX (LPPL) :

■ <http://www.latex-project.org/lppl/>

Avant-propos

Au sujet de ce manuel

Ce manuel a été écrit à l'intention des élèves de l'École nationale supérieure de techniques avancées¹ (ENSTA ParisTech). Il est en effet essentiel que de futurs ingénieurs sachent produire efficacement des documents de qualité.

Néanmoins, il ne décrit aucune spécificité propre à l'environnement informatique de l'ENSTA ParisTech et pourra donc s'avérer utile à toute personne désirant apprendre à utiliser \LaTeX .

Il ne prétend pas être exhaustif et décrit les aspects de \LaTeX qui sont généralement les plus utiles et qui vous permettront de rédiger la plupart de vos documents.

Où trouver ce manuel ?

La version à jour de ce manuel (au format PDF) est disponible sur les sites suivants :

http://www.babafou.eu.org/Apprends_LaTeX/ (site de référence)
<http://www.ctan.org/tex-archive/info/apprends-latex/>
<http://tex.loria.fr/general.html>

Comment lire ce manuel ?

Les chapitres **1** à **5**, ainsi que le chapitre **12**, sont à lire absolument car ils expliquent les bases indispensables permettant d'utiliser \LaTeX . La lecture de l'annexe **A** est également enrichissante pour tous. Ceci ne fait au total qu'une cinquantaine de pages. Le chapitre **6** n'est à lire que par les personnes ayant à utiliser des expressions mathématiques. Il en va de même avec le chapitre **7** pour la réalisation d'une bibliographie ou d'un index. Les chapitres suivants sont à lire en fonction de ses besoins. Enfin, l'annexe **B** n'est à lire que par

1. <http://www.ensta.fr/>

les rares personnes ayant pratiqué la version 2.09 de \LaTeX (ou une version antérieure) et n'ayant jamais encore utilisé $\text{\LaTeX} 2_{\epsilon}$.

Historique

La première édition de ce manuel a été écrite au printemps 1994 et décrivait $\text{\LaTeX} 2.09$.

La deuxième édition a été adaptée en 1996 et 1997 à $\text{\LaTeX} 2_{\epsilon}$ et s'est vue rajouter quelques chapitres. De petites retouches ont été apportées dans les années suivantes, sans changement de fond.

La troisième édition, en 2009, représente principalement un travail de clarification de la présentation des exemples (qui sont désormais encadrés). Une description de la classe `beamer` a également été ajoutée. Par ailleurs, l'utilisation de `pdflatex` et du format PDF ont remplacé `latex` et les formats DVI et PostScript, de même que le codage de caractères UTF-8 a remplacé ISO 8859-1 dans les exemples.

La quatrième édition, en 2010, outre une réorganisation de certains chapitres existants, s'est vue rajouter la description de `xindy` et de `latexmk` dans le chapitre sur la réalisation d'une bibliographie et d'un index ainsi que la description de la classe `scrlltr2` dans le chapitre sur l'écriture d'une lettre. Le chapitre sur la rédaction d'un CV a été créé. Pour les personnes souhaitant explorer le monde de \LaTeX , le CTAN et la commande `texdoc` sont rapidement mentionnés, de même que l'extension `TikZ` (d'autant qu'elle dispose d'une excellente documentation en français) ainsi que l'ensemble de classes KOMA-Script et la classe `memoir`. Des descriptions des extensions `geometry` et `fancyhdr` ainsi que de nombreuses extensions permettant de changer la police de caractères ont également été rajoutées.

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 exemples d'arguments et d'options des commandes \LaTeX ;
- le style `linéal` est utilisé pour les noms de classes et leurs options ainsi que pour les noms d'extensions et leurs options.

Par ailleurs, les exemples de code \LaTeX sont encadrés afin de pouvoir les identifier plus facilement et beaucoup sont suivis de leur résultat, lui aussi encadré.

Remarques et commentaires

J'espère que ce manuel sera utile au plus grand nombre et je vous invite à me faire part de vos remarques et de vos commentaires le concernant par courriel à l'adresse [<babafou+latex@babafou.eu.org>](mailto:babafou+latex@babafou.eu.org).

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, vous pouvez feuilleter ce manuel, il a été entièrement réalisé avec lui.

\LaTeX est une surcouche d’un formateur de texte appelé \TeX (le nom \TeX vient du grec τέχνη, qui veut dire « art » et d’où vient le mot « technique »²). \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 , qui demande un apprentissage certain, par rapport à des logiciels de traitement de texte plus traditionnels tels qu’OpenOffice ou Word, qui s’utilisent paraît-il d’une manière intuitive et ergonomique. Tout est affaire de philosophie.

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

2. Ce qui explique la prononciation de \LaTeX .

1.2 La philosophie de L^AT_EX

L^AT_EX n'est pas un traitement de texte à proprement parler, comme OpenOffice et Word, c'est un *formateur de texte*. Un document L^AT_EX est un fichier texte tapé avec n'importe quel éditeur (Emacs, vi...) et contenant le texte du document mélangé avec des commandes indiquant sa structure. Ceci a une conséquence très importante : lors de la frappe, on ne voit pas le document tel qu'il sera imprimé. On dit que L^AT_EX n'est pas *WYSIWYG* (*what you see is what you get*), contrairement à OpenOffice et Word. Pour visualiser le document, il faut le compiler avec L^AT_EX, ce qui génère un fichier PDF qu'on peut alors visualiser. C'est certainement le gros inconvénient de L^AT_EX pour les personnes habituées à voir le document au fur et à mesure de sa saisie. D'un autre côté, L^AT_EX décharge au maximum l'utilisateur des soucis de mise en pages, 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), L^AT_EX peut en avoir une vue d'ensemble et faire de bons choix de mise en pages.

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

1.3 L^AT_EX et la typographie

La *typographie* est l'art d'assembler des caractères. Depuis l'invention de l'imprimerie, se sont développées des règles et des habitudes afin de rendre la présentation d'un texte agréable au regard. Malheureusement, rares sont les logiciels WYSIWYG à les respecter.

Par exemple, voyez-vous une différence entre les mots « efficace » et « efficace » ? Écrivons-les un peu plus gros pour bien voir ce qui change :

« efficace » et « efficace »

Dans le premier cas, les deux f et le i sont en quelque sorte soudés, la fin de la boucle du premier f allant caresser la naissance de celle du second, les deux barres des f se retrouvant dans le prolongement l'une de l'autre et finissant au sommet du i, dont le point se confond avec la fin de la boucle du second f. C'est ce qu'on appelle une *ligature*.

Dans le second cas, les lettres sont tout simplement distinctes, les unes à côté des autres.

Les ligatures ont pour objectif de gagner de la place en réduisant l'espacement entre les lettres mais elles ont également un intérêt esthétique. \LaTeX , dans la mesure où la police de caractères utilisée le permet, génère automatiquement les ligatures adéquates.

De même, comme indiqué précédemment, puisqu'il traite le document globalement, \LaTeX peut jouer sur l'espacement pour uniformiser le gris typographique, qui est ce qu'on perçoit d'un texte quand on la regarde d'assez loin.

D'autres aspects typographiques ne peuvent être traités de manière automatique par \LaTeX et nécessitent donc d'être pris en compte par l'auteur du document. Les ouvrages indiqués dans la bibliographie sont d'une lecture enrichissante pour qui aime le beau texte. Les règles typographiques les plus importantes sont indiquées dans l'annexe A. Celles s'appliquant spécifiquement à l'écriture des mathématiques sont indiquées au paragraphe 6.10.

1.4 Comment se procurer \LaTeX ?

Sur un système UNIX, \LaTeX fait souvent partie de l'installation de base. À défaut, il est toujours possible de l'installer grâce au système d'ajout de logiciels tiers propre à l'UNIX utilisé. Il existe également des distributions pour Mac OS X et Windows.

Si vous avez à installer \LaTeX , optez sans hésiter pour la distribution \TeX Live³, qui fournit un environnement complet comprenant \LaTeX et de nombreuses extensions, ainsi qu'un système de mise à jour de l'ensemble. Il s'agit d'un logiciel libre⁴ et téléchargeable gratuitement. Ce manuel a été réalisé avec la distribution \TeX Live.

3. <http://www.tug.org/texlive/>

4. http://fr.wikipedia.org/wiki/Logiciel_libre

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.

2.1 L’édition

Un document \LaTeX est un fichier texte qu’on saisit au moyen de son éditeur de texte favori (Emacs, vi...). Nous allons commencer avec un premier document très simple, que vous pouvez copier-coller (ou, au pire, retaper) dans votre éditeur de texte :

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

\usepackage[french]{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage{lmodern}
\usepackage{microtype}

\usepackage{hyperref}

\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 le chapitre suivant. Néanmoins, prenez bien garde à le recopier

sans erreur.

Enfin, sauvez ce fichier sous le nom `premier.tex` (les documents \LaTeX ont, par convention, l'extension `.tex`).

2.2 La compilation

La *compilation* est la transformation du fichier que vous venez de taper en un fichier PDF directement visualisable.

Sur un système UNIX¹, la compilation s'effectue dans un interpréteur de commandes au moyen de la commande `pdflatex` :

```
pdflatex premier.tex
```

\LaTeX affiche ce qu'il fait au fur et à mesure (il est assez bavard, aussi la partie centrale de son discours a été volontairement supprimée) :

```
This is pdfTeX, Version 3.1415926-1.40.10 (TeX Live 2009 NetBSD/i386)
entering extended mode
(./premier.tex
LaTeX2e <2009/09/24>
[...]
Output written on premier.pdf (1 page, 42535 bytes).
Transcript written on premier.log.
```

Si la compilation s'est correctement déroulée, les deux dernières lignes seront, au nombre d'octets près, identiques à ce qui figure ci-dessus.

2.3 Les fichiers générés

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

- un fichier `.pdf`, qui est le résultat de la compilation, c'est ce fichier PDF qui est directement visualisable ;
- un fichier `.aux`, contenant les références croisées ;
- un fichier `.log`, contenant une trace du processus de la compilation plus détaillée que ce qui a été affiché à l'écran.

Ces fichiers ont le même nom de base que celui qui a été compilé. Dans notre exemple, la compilation du fichier `premier.tex` a donc généré les fichiers `premier.pdf`, `premier.aux` et `premier.log`.

L'extension `hyperref` (nous en reparlerons au chapitre 3) génère pour sa part un fichier `.out` qui contient la table des matières du document PDF

1. Si vous utilisez un autre système d'exploitation, veuillez vous référer à la documentation de votre distribution \LaTeX pour savoir comment générer un fichier PDF.

(qui est affichée par la plupart des logiciels de visualisation à gauche du document).

Suivant la complexité de votre document, \LaTeX peut également créer des 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.

2.4 La détection et la correction des erreurs

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

```
! Undefined control sequence.
1.10 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 (1.10, ligne 10);
- 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` pour arrêter la compilation, corrigez l'erreur dans le texte, sauvez-le, puis recompilez.

2.5 La visualisation

Sous UNIX, la façon la plus simple de visualiser un document PDF est d'utiliser le logiciel `xpdf` mais on peut également utiliser `epdfview`, `evince`, `gv` ou Acrobat Reader.

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 expliquera l'exemple du chapitre précédent et vous initiera plus profondément à la philosophie et à l'esprit de \LaTeX .

3.1 Le document \LaTeX

Un document \LaTeX est un mélange de texte et de commandes. Reprenons l'exemple précédent :

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

\usepackage[french]{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage{lmodern}
\usepackage{microtype}

\usepackage{hyperref}

\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 », c'est-à-dire une seule ligne

sur les quatorze (en comptant les lignes blanches) que comporte le fichier. Mais attention, le reste n'est pas inutile, loin de là ! Ce sont les fameuses commandes. Examinons la première :

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

Cette commande est indispensable et doit figurer en première ligne de tout document L^AT_EX. 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 L^AT_EX

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

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

- une commande commence par une barre oblique inversée `\` ;
- celle-ci est suivie par le nom de la commande ;
- viennent ensuite, s'il y en a, les arguments optionnels, entre crochets, séparés par des virgules s'il y en a plusieurs ;
- puis les arguments obligatoires, entre accolades, séparés par des virgules s'il y en a plusieurs.

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 843]{the-latex-companion}`

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* est l'argument obligatoire de la commande `\documentclass`. Elle définit le type du document et influe sur sa mise en pages et les commandes qu'il est possible d'utiliser. Il existe plusieurs classes, les principales classes standards étant :

article qui est utilisée pour les documents courts (rapports de projet, de TP...);

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

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

letter qui permet d'écrire des lettres ;

slides qui 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 pages du document.

Expliquons quelque peu les options utilisées ici :

a4paper permet de régler les dimensions et 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 points¹ (il existe aussi une option **12pt** ; la taille normale, sans option, est de dix points ; il existe par cohérence une option **10pt** qui ne fait rien).

Il existe d'autres options de classe. Parmi elles, citons :

twocolumn qui permet de présenter le document sur deux colonnes ;

twoside qui permet de régler les marges du document pour une impression en recto-verso (dans ce cas, la marge intérieure (vers la reliure) est généralement plus petite que la marge extérieure).

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

Une *extension* (*package* en anglais) sert à modifier la mise en pages ou à définir de nouvelles commandes. Ce sont les extensions qui permettent d'ajouter de nouvelles fonctions à \LaTeX . Elles sont chargées grâce à la commande `\usepackage`, utilisée dans le préambule du document.

Expliquons les extensions utilisées dans l'exemple :

1. Le point est une unité de mesure utilisée en typographie. Un point vaut environ $1/72^{\circ}$ de pouce (1 pouce valant 2,54 cm).

babel, utilisée avec l’option `french`, adapte L^AT_EX aux conventions typographiques et aux règles de coupure des mots françaises et redéfinit quelques noms en bon français, comme par exemple « Table des matières » ou « Chapitre » au lieu respectivement de « *Contents* » et « *Chapter* ».

fontenc, utilisée avec l’option `T1`, demande à L^AT_EX d’utiliser le codage de caractères T1 (c’est le nouveau standard L^AT_EX pour le codage des caractères mais il n’est pas encore utilisé par défaut, d’où la nécessité d’utiliser cette extension).

inputenc indique, grâce à l’option `utf8`, que le fichier est codé en UTF-8 (il existe également une option `latin9` pour les fichiers codés en ISO 8859-15 et une option `latin1` pour les fichiers codés en ISO 8859-1). Signalons également l’existence d’une option `utf8x`. Malheureusement non maintenue depuis plusieurs années, elle peut néanmoins toujours être téléchargée² et permet d’utiliser un plus grand nombre de caractères UTF-8 que l’option `utf8`.

lmodern permet d’utiliser la police de caractères Latin Modern, mieux adaptée à la génération de fichiers PDF que la police par défaut (qui a par ailleurs exactement le même dessin). Cette police de caractères a vocation à devenir celle utilisée par défaut dans une future version de L^AT_EX.

microtype apporte des raffinements typographiques supplémentaires (en particulier l’ajustement de l’espacement permettant une meilleure coupure des mots).

hyperref permet de profiter des possibilités de navigation au sein d’un document PDF sans rien avoir à faire de spécial. Grâce à elle, les références croisées, références bibliographiques, notes et URL présentes dans le document deviendront des hyperliens menant vers l’élément pointé. Une table des matières navigable accompagnera également le document.

L’utilisation de ces extensions dans tous vos documents est vivement conseillée.

Par ailleurs, l’ordre dans lequel les extensions sont chargées est généralement sans importance. Aussi est-il astucieux de les charger dans l’ordre alphabétique (ce qui est fait dans l’exemple de document) afin de pouvoir s’y retrouver rapidement lorsque de nombreuses extensions sont utilisées. L’exception confirmant la règle, l’extension `hyperref` doit toujours être chargée en dernier parce qu’elle redéfinit des commandes provenant d’autres extensions. C’est pourquoi, dans notre exemple, elle figure en dernier et séparée des autres extensions par une ligne blanche, afin de bien souligner son caractère particulier.

2. <http://www.unruh.de/DniQ/latex/unicode/>

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

3.1.5 Le document

Le document à proprement parler 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.
```

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.

Comme un retour à la ligne est équivalent à une espace, profitez-en pour couper vos paragraphes en fin de ligne. Tout bon éditeur de texte est capable de faire cela automatiquement ainsi que de reformater au propre un paragraphe dont les lignes ont des longueurs disparates. 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 et il est plus agréable de lire un texte correctement présenté qu'une grande ligne dont certains mots commencent au bord droit de la fenêtre pour se poursuivre au bord gauche de la ligne suivante.

3.2.3 L'espace insécable : ~

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

Vous souvenez-vous de la série télé dans laquelle le héros porte le numéro 6 ?
--

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

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

Vous souvenez-vous de la série télé dans laquelle le héros porte le numéro~6 ?
--

Vous souvenez-vous de la série télé dans laquelle le héros porte le numéro 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.


```
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. L^AT_EX signale de combien le texte dépasse (63.46103pt, c'est précis) et à quel endroit du fichier source se situe le problème (lines 99-99).

Rassurez-vous, cela n'arrive que très rarement. Dans ce cas, il est possible de préciser à L^AT_EX où il doit couper le mot. Pour cela, repérez l'endroit approprié en visualisant le document PDF et placez dans le texte la commande `\-@` où vous voulez que L^AT_EX 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 coupure des mots apparaissent ou disparaissent. C'est pourquoi il est préférable de n'effectuer vos coupures 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 le document PDF, mais il suffit que le message `Overfull \hbox` ait disparu pour être sûr d'avoir bien effectué la césure).

Chapitre 4

Les commandes de L^AT_EX

Les *commandes* sont fréquemment utilisées avec L^AT_EX. Que ce soit pour structurer le document, pour produire certains caractères ne figurant pas sur le clavier, etc.

4.1 Syntaxe d'une commande

Voici la syntaxe d'une commande L^AT_EX :

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

- une commande commence par une barre oblique inversée \ ;
- celle-ci est suivie par le nom de la commande ;
- viennent ensuite, s'il y en a, les arguments optionnels, entre crochets, séparés par des virgules s'il y en a plusieurs ;
- puis les arguments obligatoires, entre accolades, séparés par des virgules s'il y en a plusieurs.

Certaines commandes ont 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 La page de titre

L^AT_EX permet d'imprimer une page de titre en début de document à partir d'informations contenues dans le préambule :

`\title{titre}` indique le titre du document.

`\author{nom}` indique le nom de l'auteur du document. S'il y a plusieurs auteurs, leurs noms doivent être séparés par `\and`.

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

On peut utiliser `\\` dans les arguments de ces commandes pour forcer un passage à la ligne si nécessaire.

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

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

```
\date{}
```

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

Voici un exemple de document avec une page de titre :

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

\usepackage[french]{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage{lmodern}
\usepackage{microtype}

\usepackage{hyperref}

\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 titre 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 titre toute seule, il faut utiliser l'option de classe `titlepage`.

Cette page de titre est, il est vrai, assez sobre. Néanmoins, il est possible de personnaliser sa mise en pages grâce à l’environnement `titlepage` (voir le paragraphe 5.11).

4.3 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 L^AT_EX. Pour cela, on utilise les commandes du tableau 4.1.

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

TABLE 4.1 – *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.

Les commandes de hiérarchisation affichent un numéro, généré automatiquement, suivi de leur argument dans un corps plus important que celui du texte et fonction de leur niveau hiérarchique (un titre de chapitre sera plus grand qu’un titre de section). 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 L^AT_EX qui gère tout seul la numérotation. On peut insérer des chapitres, des sections ou modifier leur ordre sans se soucier du décalage des numéros.

Avec la même simplicité, la commande `\tableofcontents` imprime la table des matières. Il faut deux compilations successives pour avoir une table des matières à jour. La première compilation enregistre la table des matières dans un fichier `.toc`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 le paragraphe 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).

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

4.3.1 Forme étoilée

Chaque commande de hiérarchisation a une forme étoilée qui ne numérote pas son titre, ne modifie pas la numérotation des autres titres de même niveau 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.

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`). Cette commande est à utiliser immédiatement après la forme étoilée. Par exemple, pour inclure l'avant-propos dans la table des matières, j'ai utilisé :

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

4.3.2 Annexes

Enfin, la commande `\appendix` permet de signaler à L^AT_EX 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.

4.4 Construire un document à partir de plusieurs fichiers

Un document volumineux, par exemple un livre comprenant de nombreux chapitres, peut tout à fait être contenu dans un seul fichier. Mais on peut également vouloir le construire à partir de plusieurs fichiers. On peut pour cela utiliser un fichier par chapitre et un fichier principal contenant la structure du document ainsi que les commandes permettant d'inclure les fichiers correspondant aux chapitres.

La commande `\include` permet d'inclure un fichier dans un document. Son argument est le nom du fichier à inclure, sans son extension `.tex` :

```
\include{fichier}
```

Le fichier principal de notre livre peut donc ressembler à :

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

[...]

\begin{document}

\maketitle

\include{introduction}
\include{these}
\include{antithese}
\include{synthese}
\include{conclusion}

\end{document}
```

Chacun des fichiers inclus contient un chapitre donc commence par la commande `\chapter`, suivie du contenu du chapitre.

La commande `\include` a une particularité, elle génère une nouvelle page avant d'inclure le contenu du fichier et une nouvelle page après. Ceci ne pose pas de problème si les fichiers inclus correspondent à des chapitres puisque ceux-ci sont mis en pages de cette façon. En revanche, cela peut être gênant si le document utilise par exemple la classe `article` et que les fichiers inclus correspondent à des sections.

Dans ce cas, on utilise plutôt la commande `\input` :

```
\input{fichier}
```

Comme la commande `\include`, la commande `\input` permet d'inclure le fichier dont le nom lui est passé en argument (avec ou sans son extension `.tex`) mais ne génère pas de nouvelle page avant ou après d'inclure le contenu du fichier.

4.5 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.

4.5.1 Les notes de bas de page

Une note de bas de page⁴ s'obtient très facilement grâce à la commande

```
\footnote{note}
```

```
Une note de bas de page\footnote{Ceci est une note de bas de page.} s'obtient [...]
```

\LaTeX s'occupe automatiquement de la numérotation.

4.5.2 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.

4.6 Les références croisées

Avec \LaTeX , il est très facile de faire référence à un objet du document (page, chapitre, tableau, équation...).

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

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

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

Ainsi, un repère a été placé sur cette section de la façon suivante :

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

4. Ceci est une note de bas de page.

On peut alors faire référence au numéro de ce paragraphe (4.6, n'est-ce pas ?) par la commande :

```
\ref{section:references}
```

```
On peut alors faire référence au numéro de ce paragraphe
(\ref{section:references}, n'est-ce pas ?) par la commande :
```

et à la page où elle débute (22, on dirait) par :

```
\pageref{section:references}
```

```
et à la page où elle débute (\pageref{section:references},
on dirait) par :
```

La séquence précédente a été réalisée sans trucages !

L'argument de la commande `\label` est une combinaison de lettres (en ASCII, sans accents), de chiffres et de signes de ponctuation. Il est pratique d'utiliser une chaîne de la forme *type:nom* où *type* représente le type de la chose référencée (*chapitre* pour un chapitre, *section* pour une section, *tableau* pour un tableau, *figure* pour une figure, *equation* pour une équation...) afin de distinguer tout de suite de quoi on parle et où *nom* fait directement référence à l'objet repéré. Ainsi, le texte avec lequel cette section est repérée est `section:references` parce que c'est une section parlant des références croisées. C'est clair.

4.7 Caractères spéciaux

Certains symboles, certains caractères ne figurent pas sur les claviers. Heureusement, \LaTeX met à votre disposition une série de commandes permettant de les imprimer.

4.7.1 Symboles

\LaTeX définit des commandes permettant d'imprimer divers symboles (voir le tableau 4.2) ainsi que des symboles utilisés en français et dans des langues étrangères (voir le tableau 4.3).

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

TABLE 4.2 – *Symboles divers.*

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

TABLE 4.3 – Symboles pour le français et les langues étrangères.

Pour utiliser les commandes des trois premières colonnes du tableau 4.3 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`, `\oer` ou `\oeur`? Ainsi, « cœur » s’écrit `c{\oe}ur`.

Notez que les deux commandes de la dernière colonne du tableau 4.3 sont assez particulières puisqu’elles ne commencent pas par un `\`.

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.

L^AT_EX définit également trois commandes permettant d’imprimer les logos T_EX (`\TeX`), L^AT_EX (`\LaTeX`) et L^AT_EX_{2 ϵ} (`\LaTeXe`).

Voyons donc ce que donne le texte suivant :

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

Essai des commandes `TeX` et `LaTeX` dans 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, L^AT_EX 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 corriger cela :

- utiliser la commande `_` (barre oblique inversée, 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*) :

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

Ces méthodes sont équivalentes (elles produisent exactement le même résultat) mais la première est plus élégante et donne un texte source plus facile à lire.

Enfin, l'extension `eurosym`⁵ définit la commande `\euro` qui affiche le symbole € :

```
Ce livre vaut 10 \euro\ et celui-ci en vaut 15.
```

```
Ce livre vaut 10 € et celui-ci en vaut 15.
```

ainsi que la commande `\EUR`, qui fait suivre son argument d'une espace fine insécable (il est donc assez pratique d'utiliser cette commande plutôt que de rajouter des espaces insécables si nécessaire) puis du symbole € :

```
Ce livre vaut \EUR{10} et celui-ci en vaut 15.
```

```
Ce livre vaut 10€ et celui-ci en vaut 15.
```

4.7.2 Les lettres accentuées

\LaTeX définit aussi des commandes permettant d'imprimer toutes sortes de lettres accentuées (voir le tableau 4.4).

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

TABLE 4.4 – *Commandes d'accentuation.*

Néanmoins, ces commandes sont d'un usage assez lourd. \LaTeX permet donc d'utiliser, pour certaines d'entre elles, les formes simplifiées présentées dans le tableau 4.5.

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

TABLE 4.5 – *Commandes simplifiées d'accentuation.*

5. <http://www.theiling.de/eurosym.html.en>

Pour placer un accent sur un i ou un j, utilisez les commandes `\i` (i) ou `\j` (j), qui produisent des lettres sans point, en argument des commandes du tableau 4.4.

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

```
Les caractères accentués sont très utilisés en
français. Il est donc absolument indispensable de
connaître les commandes d'accentuation.
```

Bien entendu, une saisie directe des caractères accentués est amplement préférable, lorsqu'elle est possible :

```
Les caractères accentués sont très utilisés en
français. Il est donc absolument indispensable de
connaître les commandes d'accentuation, sauf quand
on peut taper les caractères accentués directement.
```

4.8 Commandes et déclarations de changement de style

L^AT_EX permet de changer le style des caractères au moyen des commandes figurant dans le tableau 4.6. Le résultat des commandes de la première colonne est indiqué dans celle du milieu. Ces commandes prennent leur nom (pour les deux dernières lettres) de leur signification en anglais comme indiqué dans la dernière colonne.

Commande	Résultat	Signification
<code>\textup{droit}</code>	droit	upright
<code>\textit{italique}</code>	<i>italique</i>	<i>italic</i>
<code>\textsl{incliné}</code>	<i>incliné</i>	<i>slanted</i>
<code>\textsc{petites capitales}</code>	PETITES CAPITALES	SMALL CAPS
<code>\textmd{maigre}</code>	maigre	medium
<code>\textbf{gras}</code>	gras	boldface
<code>\textrm{romain}</code>	romain	roman
<code>\textsf{linéal}</code>	linéal	sans serif
<code>\texttt{machine à écrire}</code>	machine à écrire	typewriter

TABLE 4.6 – Commandes de changement de style.

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

4.8. Commandes et déclarations de changement de style

```
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 possible d'imbriquer les commandes de changement de style :

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

```
C'est bon le gloubi-boulga.
```

Attention néanmoins à ne abuser des changements de style, qui peuvent rapidement rendre le texte illisible :

```
Une telle COMPOSITION est dite lardée ou entrelardée.
```

Il existe également des *déclarations* permettant d'effectuer les mêmes changements de style, comme indiqué dans le tableau 4.7.

<code>\textup</code>	<code>\upshape</code>
<code>\textit</code>	<code>\itshape</code>
<code>\textsl</code>	<code>\slshape</code>
<code>\textsc</code>	<code>\scshape</code>
<code>\textmd</code>	<code>\mdseries</code>
<code>\textbf</code>	<code>\bfseries</code>
<code>\textrm</code>	<code>\rmfamily</code>
<code>\textsf</code>	<code>\sffamily</code>
<code>\texttt</code>	<code>\ttfamily</code>

TABLE 4.7 – Correspondance entre commandes et déclarations de changement de style.

Une déclaration est une commande ne prenant pas d'argument et agissent sur le texte qui la suit.

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.

Comme pour les commandes, on peut aussi imbriquer les déclarations de changement de style :

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

C'est bon le *gloubi-boulga*.

Remarquez que toute espace suivant une déclaration est ignorée (comme les espaces suivant les commandes `\TeX` et `\LaTeX`).

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

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 fondamentaux de la philosophie de L^AT_EX, 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 été mis en valeur autrement. 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 L^AT_EX.

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...

L^AT_EX, 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 4.11). Il vous suffit donc de définir une commande `\latin` qui affichera son argument, en l'occurrence 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 L^AT_EX. On préfère plutôt définir un ensemble de commandes reflétant la logique du texte et les utiliser à la place. Ainsi, pour écrire ce manuel, des commandes spécifiques ont été définies :

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

4.9 Déclarations de changement de corps

L^AT_EX dispose de déclarations (donc à utiliser avec des accolades de groupement) permettant de modifier le corps des caractères (voir le tableau 4.8).

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

TABLE 4.8 – Déclarations de changement de corps.

Je mets les mots suivants `{\Large en gros}`. La suite du texte est de taille normale.

Je mets les mots suivants `en gros`. La suite du texte est de taille normale.

Ces déclarations ne sont normalement pas utilisées dans le fil du texte car le rendu visuel d'un changement de corps dans le texte est désagréable. On les réserve plutôt à la personnalisation de la page de titre ou à certains tableaux.

4.10 Commandes de mise en pages

Quelques commandes plus ou moins utiles permettent de forcer la main à L^AT_EX pour la mise en pages du document.

La commande `\newline` permet de forcer un saut de ligne. La commande `\\` est une abréviation pour `\newline` que l'on utilise plus souvent⁶. 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 le paragraphe 5.10).

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 fonte utilisée.

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 ses propres besoins de mise en pages). 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 de mesure indiquées dans le tableau 4.9.

<code>mm</code>	millimètre
<code>cm</code>	centimètre
<code>em</code>	<code>1em</code> est égal à la largeur de la lettre « M » dans la fonte courante
<code>ex</code>	<code>1ex</code> est égal à la hauteur de la lettre « x » dans la fonte courante
<code>in</code>	pouce (<i>inch</i> en anglais), <code>1in = 2.54cm</code>
<code>pt</code>	point pica, <code>1in = 72.27pt</code>
<code>pc</code>	pica, <code>1pc = 12pt</code>

TABLE 4.9 – Unités de mesure utilisées par \LaTeX .

La commande `\hspace` est rarement utilisée, contrairement à la commande `\vspace` qui sert le plus souvent à laisser un espace dans un texte ou dans une figure.

4.11 Définir de nouvelles commandes

Il est possible de définir ses propres commandes grâce à la commande `\newcommand` :

$$\backslash\text{newcommand}\{\backslash\text{nom}\}[\text{nombre d'arguments}]\{\text{définition}\}$$

6. Nous l'avons déjà rencontrée dans la section sur la page de titre.

Par la suite, toute occurrence de `\nom` sera remplacée par *définition*, avec substitution des éventuels arguments.

Les définitions de commandes peuvent se trouver dans le préambule (c'est là où on les place le plus souvent) ou dans le corps du document.

Prenons un exemple simple, sans arguments :

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

Il est maintenant possible d'utiliser la commande `\ba`, qui imprimera donc « babasse » :

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

```
C'est chouette la babasse!
```

Néanmoins, il est bien plus fréquent de définir des commandes qui prennent des arguments. Pour cela, on en précise le nombre dans le paramètre *nombre d'arguments* de la commande `\newcommand` et l'on y fait référence dans *définition* en utilisant `#1` pour le premier argument, `#2` pour le deuxième, etc. :

```
\newcommand{\latin}[1]{\textit{#1}}
```

La commande `\latin` permet d'imprimer son argument en italique (comme nous en avons discuté au paragraphe 4.8) :

```
\latin{carpe diem}
```

```
carpe diem
```


Chapitre 5

Les environnements de L^AT_EX

Comme vu au chapitre précédent, les commandes agissent généralement sur des arguments de taille limitée, de quelques mots à une phrase entière, tout au plus. Lorsqu'il s'agit de traiter des données plus volumineuses, on utilise des environnements.

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.
```

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

Vous utilisez déjà l'environnement `document`, nous allons maintenant en découvrir d'autres.

5.2 L'environnement `em`

L'environnement `em` met son contenu dans le même style que la commande `\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}
```

```
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.

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.

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 \\
Preignent 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*

Les environnements *center*, *flushleft* et *flushright* permettent respectivement de centrer du texte ou de ne l'aligner que d'un côté (par défaut, le texte est justifié). 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 pages 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 pages 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 pages et
couper les lignes où il faut.

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

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

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

Voici un paragraphe aligné à droite. L^AT_EX se débrouille tout seul pour faire la mise en pages 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 `machine à écrire`. 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 L^AT_EX tel quel.

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

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

5.7 Les listes : *itemize*, *enumerate* et *description*

Les listes sont très souvent employés par les utilisateurs de L^AT_EX. Elles permettent :

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

La liste ci-dessus a été obtenue grâce à l'environnement *itemize* ainsi :

```
\begin{itemize}

\item de structurer ses idées ;

\item d'aérer le texte ;

\item d'améliorer sa lisibilité.

\end{itemize}
```

Chaque élément de la liste doit commencer par la commande `\item`, qui imprime un tiret.

Les lignes blanches dans cet exemple (ainsi que dans ceux qui suivent) ne sont pas obligatoires mais elles rendent le source plus lisible en permettant de bien différencier les éléments, surtout lorsqu'ils sont nombreux et longs. On peut également, si on le souhaite, indenter les éléments pour mieux refléter la structure de l'environnement.

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}
```

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}
```

```
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 évidemment imbriquer les listes :

Il faut 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}
```

Il faut 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 débiter 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 (c'est le cas dans l'exemple précédent).
- Pour les listes dont un élément contient au moins deux phrases, chaque élément doit débiter par une majuscule et se terminer par un point.

5.8 L'environnement *tabbing*

L'environnement `tabbing` permet d'aligner du texte selon la position de *tabulations* préalablement positionnées. Son utilisation est assez bizarre et

peu pratique, aussi on lui préfère généralement l'environnement `tabular` (voir le paragraphe 5.9).

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

```
\begin{tabbing}
Troisième ligne \= le plus long \= le plus long \kill
Première ligne \> court          \> le plus long \\
Deuxième ligne \> moins court   \> court          \\
Troisième ligne \> le plus long \> moins court
\end{tabbing}
```

```
Première ligne court          le plus long
Deuxième ligne moins court court
Troisième ligne le plus long moins court
```

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

- chaque tabulation est mise 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 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`).

L'alignement vertical des commandes `\=` et `\>` n'est pas nécessaire mais permet de se rapprocher de ce qui sera imprimé au final et de mieux appréhender le positionnement des tabulations.

Par malchance, le mot « Deuxième », quand il est imprimé, est légèrement plus long que le mot « Troisième », bien qu'il comporte moins de lettres. Il en est de même pour « moins court » et « plus long ». Le placement des tabulations n'est donc pas aussi simple qu'on pourrait le penser et nécessite une vérification systématique sur le document final. Il peut aussi varier en fonction de la fonte utilisée.

On peut redéfinir la position et le nombre des tabulations au milieu de l'environnement, si nécessaire :

```

\begin{tabbing}
ça c'est long \= ça c'est long \= ça c'est long \\
court          \> court          \> court          \\
ça c'est vraiment long \= moins long \\
tout petit          \> court
\end{tabbing}

```

```

ça c'est long ça c'est long ça c'est long
court          court          court
ça c'est vraiment long moins long
tout petit          court

```

Il existe d'autres commandes permettant de manipuler les tabulations, mais qui présentent peu d'intérêt. Pour plus d'informations sur l'environnement `tabbing`, reportez vous à [5, pages 60–62 et 201–203].

5.9 Les tableaux : l'environnement `tabular`

L'environnement `tabular` est de loin l'environnement le plus puissant de \LaTeX . Avec lui, il est possible de réaliser 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}

```

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 lettres. Cela signifie que le tableau comporte trois colonnes. Détaillons cet argument :

1 (*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 `&`.

Remarquez que les `&` sont placés les uns en dessous des autres, pour améliorer la lisibilité du tableau dans le source. De même les colonnes ont été alignées comme elles doivent être imprimées.

Voyons maintenant le même tableau en matérialisant les séparations entre lignes et colonnes :

```
\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

- 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`, placée entre chaque ligne du tableau (sans `\` après).

Encore un tableau :

```
\begin{tabular}{|p{5cm}|*{2}{c|}}
\hline
& Contenance & Quantité \\
\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 simplifier la définition du format des grands tableaux.

Vous remarquerez qu'on peut laisser une case vide et qu'une déclaration de changement de style (ou de corps) dans une case n'affecte pas les autres cases.

Modifions quelque peu le tableau précédent :

```
\begin{tabular}{|p{5cm}|*{2}{c|}}
\cline{2-3}
\multicolumn{1}{c|}{} & \multicolumn{2}{c|}{Commande} \\
\cline{2-3}
\multicolumn{1}{c|}{} & Contenance & Quantité \\
\hline
\bfseries Blanche de Bruges & 33 cl & 10 \\
\hline
\bfseries Guinness & 1 pint & 5 \\
\hline
\bfseries Kronenbourg & 33 cl & 0 \\
\hline
\end{tabular}
```

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

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*. Cette commande est généralement utilisée pour étendre un élément du tableau sur plusieurs colonnes (c'est le cas pour la case « Commande »), pour modifier l'alignement d'une case ou pour supprimer certains filets verticaux (c'est le cas pour les deux cases vides de gauche).

Pour approfondir vos connaissances sur l'environnement `tabular`, vous pouvez consulter [5, pages 62–63 et 204–207] et [10, chapitre 5].

5.10 Les environnements `table` et `figure`

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

Il est important de pouvoir faire flotter les tableaux et les figures car L^AT_EX ne les coupe 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]
\centering
\begin{tabular}{|l|l|}
\hline
table & figure \\
\hline
tableaux & dessins \\
\hline
\end{tabular}
\caption{Exemple d'environnement \environment{table}}
\label{tableau:exemple}
\end{table}
```

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

table	figure
tableaux	dessins

TABLE 5.1 – Exemple d'environnement `table`.

! demande à `LATEX` de faire tout son possible pour respecter l'ordre indiqué ensuite ;

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 document.

La commande `\centering` a été utilisée pour centrer le tableau. Elle est préférable à l'environnement `center` dans ce cas parce que celui-ci introduit une marge trop importante entre le tableau et sa légende.

Par ailleurs :

- la commande `\caption` permet de donner une légende au tableau ou à la figure ;
- la commande `\label`, qui doit obligatoirement être placée 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 `.lof`²).

5.11 Personnaliser la page de titre : l'environnement `titlepage`

La commande `\maketitle` génère une page de titre plutôt minimale. Il est regrettable qu'un logiciel de la qualité de `LATEX` doive se contenter de pages de titre aussi sobres.

Heureusement, il est possible de personnaliser la page de titre grâce à l'environnement `titlepage`. À l'intérieur de cet environnement, vous êtes totalement maître de la mise en pages. Généralement, on y utilise :

1. `.lot` pour *list of tables*.
 2. `.lof` pour *list of figures*.

- l’environnement `center`, pour centrer le texte ;
- les commandes de changement de taille (`\large`, etc.), pour obtenir un gros titre ;
- les commandes d’espacement `\vspace` et `\hspace` pour aérer le texte ;
- la commande `\includegraphics` (voir le paragraphe 8.6.1) pour inclure une image.

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

De nombreux exemples de pages de titre figurent dans [27].

5.12 Définir de nouveaux environnements

Il est possible de définir ses propres environnements grâce à la commande `\newenvironment` :

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

Les définitions d’environnements peuvent se trouver dans le préambule (c’est là où on les place le plus souvent) ou dans le corps du document.

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

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

Ici, la définition du nouvel environnement a été répartie sur trois lignes pour la rendre plus lisible. Il s’utilise ensuite ainsi :

```
\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 (c'est le cas ici).

Il est possible de créer des environnements qui prennent des arguments. Pour cela, on en précise le nombre dans le paramètre *nombre d'arguments* de la commande `\newenvironment` et l'on y fait référence dans *définition de début* en utilisant `#1` pour le premier argument, `#2` pour le deuxième, etc. :

```
\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

Chapitre 6

L^AT_EX mathématique

L'un des atouts de L^AT_EX 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. Pour aller plus loin, il peut s'avérer nécessaire d'utiliser des extensions spécialisées, en particulier l'extension `amsmath` (voir [10, chapitre 8]). Il existe également de nombreuses extensions pour représenter des molécules, des diagrammes utilisés en physique des particules, des circuits électriques, etc. (voir [8, chapitre 8]).

Les règles typographiques d'écriture des mathématiques sont, pour la plupart, respectées automatiquement par L^AT_EX. Il n'est cependant pas inutile de se reporter à [20, pages 107 à 116] pour en prendre connaissance. Les règles qui demandent une attention particulière seront indiquées au paragraphe 6.10.

6.1 Les modes mathématiques

L^AT_EX fournit 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 composer une formule mathématique. Pour cela, L^AT_EX dispose de deux modes mathématiques :

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

6.1.1 L'environnement `math`

L'environnement `math` permet de composer des formules mathématiques dans le corps du texte :

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

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 `$... $`, qui est la plus simple.

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`

L'environnement `displaymath` permet de composer des formules mathématiques 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 une forme plus simple, `\[... \]`. On aurait donc pu écrire :

Ce qui conduit à :

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

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

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 $`

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

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

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

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

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 simplement 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. Il n'y a pas vraiment de différence mathématique entre les deux formes mais la seconde est typographiquement plus précise.

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` :

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

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

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. Si l'on avait omis ces espaces, le résultat aurait été le suivant :

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

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

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 extension est nécessaire.

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

Majuscules							
Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>	Ψ	<code>\Psi</code>
Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Φ	<code>\Phi</code>		

TABLE 6.1 – Lettres grecques.

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.

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 `\AA`.

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, `\varepsilon` et `\varphi` sont à préférer à `\epsilon` et `\phi`.

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.

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

^a Requiert l'extension latexsym.

TABLE 6.2 – 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.

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

^a Requiert l'extension latexsym.

TABLE 6.3 – Symboles de relations binaires.

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

<code>\$ x \not\in F \$</code>

$x \notin F$

6.3.4 Flèches

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

\leftarrow	<code>\leftarrow</code>	\longleftarrow	<code>\longleftarrow</code>	\uparrow	<code>\uparrow</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>	\Uparrow	<code>\Uparrow</code>
\rightarrow	<code>\rightarrow</code>	\longrightarrow	<code>\longrightarrow</code>	\downarrow	<code>\downarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>	\Downarrow	<code>\Downarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>	\updownarrow	<code>\updownarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>	\nearrow	<code>\nearrow</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>	\searrow	<code>\searrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>	\swarrow	<code>\swarrow</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoondown	<code>\rightharpoondown</code>	\nwarrow	<code>\nwarrow</code>
\rightleftharpoons	<code>\rightleftharpoons</code>	\leadsto	<code>\leadsto</code> ^a		

^a Requiert l'extension latexsym.

TABLE 6.4 – *Flèches.*

6.3.5 Symboles divers

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

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

^a Requiert l'extension latexsym.

TABLE 6.5 – *Symboles divers.*

De plus, les commandes `\ldots` (...) et `\cdots` (...) permettent d'obtenir des points de suspension (respectivement alignés sur la base de la ligne ou

centrés verticalement).

6.3.6 Accents

Le tableau 6.6 présente les accents disponibles en mode mathématique.

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

TABLE 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 (voir le tableau 6.5).

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` :

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

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

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) $
```

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

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

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

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

6.3.8 Ensembles

Les extensions `bbm`, `bbold` et `dsfont` fournissent des polices de caractères et les commandes correspondantes permettant de représenter les symboles des ensembles usuels.

L'extension `bbm` fournit ainsi les commandes `\mathbbm`, `\mathbbmss` et `\mathbbmstt`. L'extension `bbold` fournit la commande `\mathbb`. L'extension `dsfont` fournit la commande `\mathds`. Ces commandes doivent être utilisées en mode mathématique.

L'extension `bbold` fournit également une commande `\textbb` destinée à être utilisée dans du texte normal.

Le choix de l'extension et de la commande à utiliser dépend des préférences de chacun quant à l'aspect des caractères. Le tableau 6.7 présente ce que produit chacune des commandes pour les ensembles usuels.

bbm			bbold	dsfont
<code>\mathbbm</code>	<code>\mathbbmss</code>	<code>\mathbbmtt</code>	<code>\mathbb</code>	<code>\mathds</code>
N	ℕ	ℕ	ℕ	ℕ
Z	ℤ	ℤ	ℤ	ℤ
D	℔	℔	℔	℔
Q	ℚ	ℚ	ℚ	ℚ
R	ℝ	ℝ	ℝ	ℝ
C	ℂ	ℂ	ℂ	ℂ

TABLE 6.7 – *Ensembles.*

6.4 Modifier l’espace

Dans certains cas, il peut être utile de modifier l’espace, afin d’éloigner ou de rapprocher certains symboles (voir le tableau 6.8).

<code>\,</code>	petite espace	<code>\!</code>	petite espace négative
<code>\:</code>	moyenne espace	<code>\quad</code>	petite espace interformules
<code>\;</code>	grande espace	<code>\qquad</code>	moyenne espace interformules

TABLE 6.8 – *Commandes d’espace.*

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

6.5 Modifier le style et la taille

L^AT_EX permet de modifier le style des caractères en mode mathématique (voir le tableau 6.9).

<code>\$ x + y + 2^n \Omega \cos \theta \$</code>	$x + y + 2^n \Omega \cos \theta$
<code>\$ \mathit{x + y + 2^n \Omega \cos \theta} \$</code>	$x + y + 2^n \Omega \cos \theta$
<code>\$ \mathrm{x + y + 2^n \Omega \cos \theta} \$</code>	$x + y + 2^n \Omega \cos \theta$
<code>\$ \mathbf{x + y + 2^n \Omega \cos \theta} \$</code>	$\mathbf{x + y + 2^n \Omega \cos \theta}$
<code>\$ \mathsf{x + y + 2^n \Omega \cos \theta} \$</code>	$x + y + 2^n \Omega \cos \theta$
<code>\$ \mathtt{x + y + 2^n \Omega \cos \theta} \$</code>	$x + y + 2^n \Omega \cos \theta$

TABLE 6.9 – *Commandes de changement de style.*

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

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

Soient \mathcal{C} le cercle de centre 0 et de rayon 1, \mathcal{D}_1 et \mathcal{D}_2 deux droites.

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

Voici ce à quoi ressemblent ces majuscules :

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Comme nous allons le voir par la suite (en particulier au paragraphe 6.9), certains symboles n'ont pas la même taille dans l'environnement `math` et dans l'environnement `displaymath`.

La déclaration `\displaystyle` permet, dans un environnement `math`, de forcer L^AT_EX à utiliser des caractères et des symboles de la même taille qu'en environnement `displaymath`. Cette déclaration 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 déclaration `\textstyle` permet de forcer L^AT_EX à utiliser, dans un environnement `displaymath`, des caractères et des symboles de la même taille qu'en environnement `math`.

Enfin, signalons aussi l'existence des deux déclarations `\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}`

`\[`
`\frac{x+1}{x+2}`
`\]`

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

6.7 Racines

Les racines s'obtiennent grâce à la commande :

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

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

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

6.8 Délimiteurs

Les délimiteurs, présentés dans le tableau 6.10, sont des symboles permettant d'encadrer des formules. Leur hauteur s'adapte à celle de la formule.

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

TABLE 6.10 – Délimiteurs.

Les délimiteurs doivent toujours aller par paire : 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 nécessaire.

Voici un exemple d'utilisation des délimiteurs (nous aborderons l'environnement `array` au paragraphe 6.12) :

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

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

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.

Il en va de même si l'on ne souhaite pas de délimiteur ouvrant, on utilise `\left.` (`\left` suivi d'un point).

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

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

6.9 Symboles de taille variable

Les symboles présentés dans le tableau 6.11 sont de *taille variable* : ils n'ont pas la même taille dans l'environnement `math` et dans l'environnement `displaymath`.

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

TABLE 6.11 – 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 $
\[ \sum_{i=0}^{+\infty} x_i \quad \int_a^b f(x) dx \]
```

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

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

Comme indiqué au paragraphe 6.5, on peut, en environnement `math`, utiliser la déclaration `\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 $\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$.

Dans les intégrales, on utilise souvent les commandes d'espacement vues au paragraphe 6.4 pour éloigner le dx ou pour rapprocher les symboles d'intégration des intégrales multiples :

```
\[
\int_a^b f(x) \, dx \quad \int \int \int_{\Omega} d\tau
\]
```

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

En cas d'usage intensif d'intégrales multiples, il peut être intéressant d'utiliser l'extension `amsmath` (voir [10, chapitre 8]) qui définit des commandes spécifiques pour la saisie des intégrales multiples (voir [10, page 492]).

6.10 Mathématiques et typographie

L^AT_EX respecte de lui-même une grande partie des règles typographiques d'écriture des mathématiques. On se reportera à [20, pages 107 à 116] pour en avoir une description complète. Néanmoins, un coup de pouce de la part de l'utilisateur est parfois nécessaire.

C'est le cas pour les intégrales, qu'il convient d'écrire comme indiqué à droite dans l'exemple suivant, c'est-à-dire avec un peu d'espace entre la fonction et la différentielle, celle-ci étant composée en romain :

```
\[
\int_a^b f(x) dx \quad \int_a^b f(x) \, \mathrm{d}x
\]
```

$$\int_a^b f(x)dx \quad \int_a^b f(x) \, dx$$

Les formules mathématiques contiennent majoritairement des variables et des fonctions et il est d'usage d'écrire celles-ci en italique. C'est pour cela que les formules mathématiques sont composées par L^AT_EX en italique.

En revanche, le symbole de différentielle se compose en romain, d'où l'utilisation de la commande `\mathrm` dans l'exemple précédent. On en profite également pour ajouter un peu d'espace entre la fonction et la différentielle, ce qui donne un rendu visuel plus agréable.

Si le document comporte beaucoup d'intégrales, il peut être utile de simplifier leur saisie en définissant une nouvelle commande permettant de représenter la différentielle et son espace :

```
\newcommand{\diff}{\, \mathrm{d}}
```

On peut alors écrire :

```
\[
\int_a^b f(x) \diff x
\]
```

$$\int_a^b f(x) \, dx$$

De même, les constantes se composent en romain. C'est en particulier le cas pour *e* et *i* :

```
\[
\mathrm{e}^{\mathrm{i}\pi} + 1 = 0
\]
```

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

Les constantes physiques se composent également en romain :

```
\[
E = m\mathrm{c}^2
\]
```

$$E = mc^2$$

En cas d'utilisation intensive de certaines constantes, il peut être utile de définir des commandes permettant de les représenter, comme il a été fait pour `\diff`.

6.11 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 l'opérateurs `lim` et la fonction `ln` ne sont pas en romain. Essayons donc ainsi :

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

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

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

Pour résoudre ces problèmes, L^AT_EX fournit les commandes présentées dans le tableau 6.12.

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

TABLE 6.12 – Fonctions et opérateurs.

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 6.12 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* :

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

$$c = a \bmod b \quad a \equiv b \pmod{n}$$

6.12 Les tableaux : l'environnement `array`

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

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

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

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

- l** (*left*) la colonne est alignée à gauche ;
- c** (*center*) la colonne est centrée ;
- r** (*right*) la colonne est alignée à droite.

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

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

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

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

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

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

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

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

6.13 Les équations numérotées : l'environnement `equation`

L'environnement `equation` s'utilise de la même façon que l'environnement `displaymath` mais il génère en plus un numéro d'équation :

```
\begin{equation} \label{eqn:euler}
\mathrm{e}^{\mathrm{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`.

6.14 Les équations numérotées et alignées : l'environnement `eqnarray`

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

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

$$\ln xy = \ln x + \ln y \quad (6.2)$$

$$\exp(x + y) = \exp x \exp y \quad (6.3)$$

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 \quad \nonumber \\
& = & \frac{2^3}{3} - \frac{1^3}{3} \quad \nonumber \\
& = & \frac{8}{3} - \frac{1}{3} \quad \nonumber \\
& = & \frac{7}{3}
\end{eqnarray}
```

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

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

```

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

6.15 Théorèmes, définitions...

Théorème 1 (babafou) *LaTeX ne se limite pas aux formules et aux signes bizarres, il permet aussi de composer 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 composer 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 dédié qui est incrémenté à chaque utilisation de l'environnement);

3. puis, éventuellement, son argument optionnel (c'est généralement l'auteur du théorème) entre parenthèses ;
4. enfin, il imprime son contenu, préalablement mis en italique.

On peut ainsi créer autant d'environnements que nécessaire, un pour les théorèmes, un pour les définitions, un pour les lemmes, un pour les hypothèses, etc., chacun ayant son propre compteur.

Chapitre 7

Bibliographie et index

Réaliser une bibliographie ou un index est généralement assez compliqué. Pour cela, \LaTeX s'appuie sur deux autres logiciels, \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 :

- en utilisant l'environnement `thebibliography` ;
- en utilisant le logiciel \BIBTeX .

7.1.1 L'environnement `thebibliography`

L'environnement `thebibliography` permet d'imprimer la liste des références bibliographiques :

```
\begin{thebibliography}{2}

\bibitem{latex-a-document-preparation-system}
  Leslie Lamport :
  \textit{\LaTeX: a document preparation system}.
  Addison-Wesley, second edition, 1994.

\bibitem{the-latex-companion}
  Frank Mittelbach et Michel Goossens :
  \textit{The \LaTeX\ Companion}.
  Addison-Wesley, second edition, 2004.

\end{thebibliography}
```

Ce qui donne :

Bibliographie

- [1] Leslie Lamport : *L^AT_EX : a document preparation system*. Addison-Wesley, second edition, 1994.
- [2] Frank Mittelbach et Michel Goossens : *The L^AT_EX Companion*. Addison-Wesley, second edition, 2004.

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 ou les auteurs de l'ouvrage, son titre (en italique, c'est l'usage), son éditeur, son année de parution et toute information devant figurer dans la bibliographie.

Dans la version imprimée, chaque élément de la bibliographie est repéré par un nombre entre crochets généré automatiquement. Il est possible d'assigner une étiquette personnalisée grâce à l'argument optionnel de `\bibitem` :

```
\bibitem[Lamport 1994]{latex-a-document-preparation-system}
  Leslie Lamport :
  \textit{\LaTeX: a document preparation system}.
  Addison-Wesley, second edition, 1994.
```

Ce qui donne :

```
[Lamport 1994] Leslie Lamport ...
```

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

- lorsqu'on ne les personnalise pas, on met généralement le nombre d'éléments de la bibliographie ;
- lorsqu'on les personnalise, on met l'étiquette la plus longue (qui n'est pas toujours évidente à déterminer comme nous l'avons vu avec l'environnement `tabbing`).

La commande `\cite` permet de faire référence à un élément de la bibliographie au sein du document. Elle imprime le numéro ou l'étiquette de l'ouvrage :

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

Voir [10] pour plus de précisions.

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

Voir `\cite[page 691]{the-latex-companion}` pour plus de précisions.

Voir [10, page 691] pour plus de précisions.

7.1.2 BibTeX

L'utilisation de l'environnement `thebibliography` convient très bien à la réalisation d'une bibliographie, mais elle a certains inconvénients :

- il est nécessaire de faire une bibliographie complète pour chaque document, avec les problèmes de redondance que cela implique si certains ouvrages sont cités dans plusieurs documents ;
- en cas de mise à jour d'un ouvrage (publication d'une nouvelle édition, par exemple), 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 dans chaque document, de regrouper l'ensemble des données bibliographiques dans une ou plusieurs bases de données et de générer automatiquement la bibliographie de chaque document à partir de ces bases de données, en fonction des ouvrages dont il est fait référence dans le document. C'est le rôle du logiciel BibTeX.

7.1.2.1 La base de données bibliographique

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

```
@BOOK
{
  latex-a-document-preparation-system ,
  AUTHOR    = "Leslie Lamport" ,
  TITLE     = "{\LaTeX}: a document preparation system" ,
  PUBLISHER = "Addison-Wesley" ,
  EDITION   = "Second" ,
  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.

Les caractéristiques de l'ouvrage sont indiquées à la suite, entre accolades.

Il faut tout d'abord spécifier une chaîne de caractères (pouvant contenir lettres, chiffres et signes de ponctuation), qu'on appelle *clé* et 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.

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" ,
```

Si une commande est utilisée dans un champ, elle doit être entourée d'accolades :

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

Les champs contenant une valeur numérique (YEAR, par exemple) peuvent se passer des guillemets :

```
YEAR = 1994 ,
```

La clé et les champs sont séparés les uns des autres par des virgules. La présentation adoptée dans l'exemple précédent est d'avoir une ligne pour la clé puis une ligne par champ donc chaque ligne se termine par une virgule. Remarquez que la dernière ligne avant l'accolade fermante se termine également par une virgule alors qu'elle n'est pas nécessaire. Néanmoins, dans la mesure où BIB_TE_X la tolère, ceci permet d'avoir une virgule à la fin de chaque ligne de l'entrée et donc d'éviter de l'oublier si jamais on rajoute un champ supplémentaire après l'actuel dernier.

7.1.2.2 Les différents types d'entrées

Le tableau 7.1 contient la liste des types d'entrées reconnus par BIB_TE_X. Pour chaque type sont indiqués sa signification ainsi que les champs obligatoires et optionnels (les autres, s'il y en a, étant ignorés) dont la signification sera expliquée dans le tableau 7.2.

Entrée	Signification, champs obligatoires et optionnels
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, il peut s'agir d'un chapitre, d'une section ou de 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

TABLE 7.1 – *Types d'entrées reconnus par BibTEX (à suivre).*

Entrée	Signification, champs obligatoires et optionnels
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 autre 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

TABLE 7.1 – Types d'entrées reconnus par BibTeX.

7.1.2.3 Les différents types de champs

Les champs utilisés par BibTeX, ainsi que leur signification, sont indiqués dans le tableau 7.2.

7.1.2.4 Quelques précisions sur certains champs

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

```
AUTHOR = "Frank Mittelbach and Michel Goossens" ,
TITLE = "The {\LaTeX} Companion" ,
```

Le champ MONTH doit contenir l'abréviation sur trois lettres du mois en anglais (`jan`, `feb`, `mar`, `apr`, `may`, `jun`, `jul`, `aug`, `sep`, `oct`, `nov` ou `dec`) sans guillemets :

```
MONTH = apr ,
```

ADDRESS	Adresse de l'éditeur (PUBLISHER) ou de l'institution (INSTITUTION)
ANNOTE	Annotation
AUTHOR	Nom de l'auteur (ou 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 du 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 ou plusieurs numéros de page
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

TABLE 7.2 – Champs utilisés par BIB_TE_X.

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{the-latex-companion} pour plus de précisions.
```

Il faut aussi indiquer à L^AT_EX où placer la bibliographie (généralement juste à la fin du document), au moyen de la commande `\bibliography`. Cette commande prend en argument une liste de noms de fichiers de base de données bibliographique (mais sans leur 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 pages la bibliographie, grâce à la commande `\bibliographystyle`, utilisée dans le préambule du document. L'argument de cette commande est le nom du style bibliographique à utiliser.

Les styles bibliographiques standards de L^AT_EX sont :

plain

Les ouvrages sont triés dans l'ordre alphabétique de leur auteur (ou premier auteur s'il y en a plusieurs) 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.

Selon la façon dont L^AT_EX est installé sur votre ordinateur, il est possible que les styles bibliographiques standards ne soient pas francisés. Ceci est facile à voir au fait que, pour les ouvrages ayant plusieurs auteurs, le mot clé **and** séparant les noms des auteurs dans la base de données bibliographique se retrouve sous la forme « and » dans la bibliographie au lieu de « et ». Dans ce cas, utilisez les styles **plain-fr**, **unsrt-fr**, **alpha-fr** ou **abbrv-fr**. S'ils ne sont pas installés, vous pouvez les télécharger ici :

■ <http://mirror.ctan.org/biblio/bibtex/contrib/bib-fr.zip>

Il existe de nombreux autres styles bibliographiques, dont beaucoup sont indiqués dans [10, pages 791–797].

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

```
% dans le préambule :
\bibliographystyle{plain} % ou plain-fr ou un autre style

[...]

\begin{document}

[...]

% à la fin du document :
\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 après la commande `\bibliography`), avec en argument la liste des clés de ces ouvrages, séparées par des virgules :

```
\nocite{guide-to-latex,bibtexing,makeindex}
```

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 :

```
pdflatex 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 `.bbl`, qui contient la bibliographie mise en pages (pour les curieux, cela est fait avec l’environnement `thebibliography`) selon le style indiqué par la commande `\bibliographystyle`.

- Recompiler deux fois votre document avec L^AT_EX afin d’y inclure la bibliographie et de mettre toutes les références à jour.

Pour plus d’informations au sujet de B_IB_TE_X, vous pouvez consulter [16], [5, pages 70, 71 et 155–164] et [10, chapitre 13].

7.2 Faire un index

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

Avant cela, il faut modifier un peu votre document :

- chargez l’extension `makeidx` ;
- ajoutez la commande `\makeindex` dans le préambule ;
- mettez la commande `\printindex` à l’endroit où vous voulez voir apparaître l’index (c’est généralement à la fin du document, après la bibliographie et avant la table des matières).

Le formatage de l’index nécessite l’utilisation d’un logiciel externe, qui va classer les entrées dans l’ordre alphabétique et fusionner les entrées identiques. Historiquement, on utilisait *MakeIndex* pour cela (voir le paragraphe 7.2.3). Malheureusement, celui-ci est incapable de traiter correctement les mots contenant des caractères accentués. Si l’on écrit un document dans une langue en utilisant (ce qui est notre cas), il est préférable d’utiliser `xindy` (voir le paragraphe 7.2.4).

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 », c’est-à-dire l’argument de la commande `\index`, 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.
```

Si l'on utilise *MakeIndex* plutôt que *xindy*, ceci peut servir à lui indiquer la position correcte d'un mot contenant des caractères accentués :

```
Un répertoire\index{repertoire@répertoire} permet de
regrouper des fichiers.
```

En effet, *MakeIndex* classe les caractères accentués avant tous les autres. Par exemple, il classe « répertoire » avant « racine ». L'utilisation de @ permet donc de le remettre dans le droit chemin, si nécessaire. Son utilisation ne doit pas être systématique, un mot contenant suffisamment de lettres avant son premier caractère accentué ayant toutes les chances d'être classé correctement.

7.2.2 Générer l'index

Que ce soit en utilisant *MakeIndex* ou *xindy*, la génération de l'index se fait selon la même méthode.

- Il faut tout d'abord compiler votre document avec L^AT_EX :

```
pdflatex document.tex
```

Ceci génère un fichier `.idx`, qui contient l'ensemble des entrées à faire figurer dans l'index.

- Il faut ensuite exécuter *MakeIndex* ou *xindy* sur ce fichier (la façon exacte de procéder sera abordée respectivement dans les paragraphes 7.2.3 et 7.2.4). Ceux-ci classent les entrées dans l'ordre alphabétique et fusionnent les entrées identiques. L'index ainsi mis au propre est placé dans un fichier `.ind`.
- Il faut enfin recompiler votre document avec L^AT_EX afin d'y inclure l'index (c'est-à-dire le contenu du fichier `.ind`).

7.2.3 Générer l'index avec *MakeIndex*

Pour générer l'index avec *MakeIndex*, on utilise la commande :

```
makeindex -c document.idx
```

L'option `-c` permet de ne pas tenir compte des espaces multiples pouvant figurer dans les entrées.

Le format par défaut de l'index est assez spartiate. Heureusement, il est possible d'en modifier la présentation. Pour cela, on utilise un fichier ayant par convention l'extension `.ist`. Vous pouvez essayer celui-ci, que l'on va appeler `index.ist` :

```
headings_flag          1
heading_prefix         "\n \\hfill \\textbf{\\large"
heading_suffix         "} \\hfill \\nopagebreak\n"
symhead_positive      "Symboles"
symhead_negative      "symboles"
delim_0                " \\dotfill\\ "
delim_1                " \\dotfill\\ "
delim_2                " \\dotfill\\ "
```

Pour l'utiliser, il faut exécuter *MakeIndex* en indiquant le nom de ce fichier après l'option `-s` ainsi :

```
makeindex -c -s index.ist document.idx
```

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

Pour plus d'informations au sujet de *MakeIndex*, vous pouvez consulter [15], [5, pages 150–154] et [10, § 11.2].

7.2.4 Générer l'index avec *xindy*

Pour générer l'index avec *xindy*, on utilise la commande :

```
texindy -L french -M texindy document.idx
```

La commande `texindy` permet à *xindy* de traiter un index généré par \LaTeX (car *xindy* est un logiciel générique de traitement d'index, qui peut être utilisé avec d'autres logiciels que \LaTeX).

L'option `-L` permet d'indiquer dans quelle langue est rédigé l'index. L'ordre de classement des caractères accentués est en effet différent selon les langues.

L'option `-M` permet d'utiliser le module dont le nom suit. Le module `texindy` permet de générer un index adapté à \LaTeX et semblable à ce que génère `MakeIndex`.

Il est possible de créer ses propres modules afin de personnaliser le format de l'index. Pour cela, on utilise un fichier ayant par convention l'extension `.xdy` et contenant des commandes en langage Lisp.

Par exemple, les termes figurant dans l'index sont par défaut directement suivis des numéros des pages où ils figurent. Pour avoir une présentation identique à celle de l'index de ce manuel, avec les numéros de pages à droite, précédés par des points de suspension, on peut utiliser le fichier `index.xdy` suivant :

```
(markup-locclass-list :open " \dotfill\ " :sep ", ")
```

Pour l'utiliser, il faut exécuter `texindy` en indiquant le nom de ce fichier après une autre option `-M` :

```
texindy -L french -M texindy -M index.xdy document.idx
```

On peut également personnaliser l'aspect des titres des rubriques de l'index grâce aux commandes suivantes, placées dans le préambule du document :

```
\newcommand*\lettergroupDefault[1]{\lettergroup{Symboles}}
\newcommand*\lettergroup[1]{\hfill\textbf{\large #1}\hfill\nopagebreak}
```

La première commande permet d'afficher « Symboles » avant les symboles non alphabétiques figurant dans l'index (s'il y en a). Par défaut, rien n'est affiché.

La seconde commande permet de centrer et d'afficher en taille `\large` les titres de chaque rubrique (par défaut, ils sont placés à gauche et sont de taille normale).

Pour plus d'informations au sujet de `xindy`, vous pouvez consulter [10, § 11.3] ainsi que le site :

■ <http://www.xindy.org/>

7.3 La commande `latexmk`

La réalisation d'un document complexe, contenant une bibliographie et un index, nécessite donc l'utilisation des commandes `pdflatex`, `bibtex`, `makeindex` ou `texindy` et une nouvelle fois `pdflatex` (voire une fois de plus si la bibliographie ou l'index modifient la pagination).

Non seulement, ceci est fastidieux mais, dans l'idéal, il ne faudrait régénérer la bibliographie que lorsqu'elle a changé. De même pour l'index. Ce qui est encore plus fastidieux...

Heureusement, la commande `latexmk` permet d'automatiser l'enchaînement de ces commandes dans l'ordre adéquat et uniquement lorsque cela est nécessaire. Elle s'utilise avec l'option `-pdf` pour générer un fichier PDF :

```
latexmk -pdf document.tex
```

Il est possible de configurer le comportement de `latexmk` grâce au fichier `~/ .latexmkrc`. Par exemple, pour permettre la génération de fichiers PDF sans avoir à utiliser l'option `-pdf`, il faut indiquer dans le fichier `~/ .latexmkrc` :

```
$pdf_mode = 1 ;
```

On peut alors faire simplement :

```
latexmk document.tex
```

Si *MakeIndex* est utilisé avec des options (par exemple `-c` ou `-s`), il faut l'indiquer dans le fichier `~/ .latexmkrc` :

```
$makeindex = 'makeindex -c -s index.ist %O -o %D %S' ;
```

De même, si vous utilisez `xindy` plutôt que *MakeIndex*, il faut l'indiquer dans le fichier `~/ .latexmkrc` :

```
$makeindex = 'texindy -L french -M texindy -M index.xdy -o %D %S' ;
```

Afin de stocker les données nécessaires à son fonctionnement, `latexmk` utilise un fichier, créé lors de son premier lancement, ayant le même nom de base que le fichier \LaTeX et l'extension `.fdb_latexmk`. Dans notre exemple, il s'agit donc du fichier `document.fdb_latexmk`. Ce fichier peut être supprimé au besoin, il sera recréé par `latexmk` lors de sa prochaine utilisation.

L'option `-c` (*clean up*) permet d'effacer ce fichier ainsi que tous les fichiers générés par \LaTeX , à l'exception du fichier PDF :

```
latexmk -c
```

7.4 Bibliographie et index dans la table des matières

Par défaut, la bibliographie et l'index ne figurent pas dans la table des matières générée par la commande `\tableofcontents`.

Pour les y faire apparaître, on peut utiliser l'extension `tocbibind`. Elle fera également apparaître la liste des tableaux, la table des figures, s'ils existent, ainsi que la table des matières elle-même.

Chapitre 8

Aller plus loin avec L^AT_EX

L'un des intérêts de L^AT_EX est qu'il est extensible. Il existe des centaines d'extensions et de classes permettant de réaliser quasiment tous les documents possibles et imaginables. Nous allons en présenter quelques unes parmi les plus utiles. Beaucoup d'autres sont documentées dans [10], dont la lecture est indispensable si l'on souhaite utiliser L^AT_EX à son plein potentiel.

8.1 CTAN : *the Comprehensive T_EX Archive Network*

The Comprehensive T_EX Archive Network (CTAN) est un réseau d'une centaine de serveurs répliqués permettant l'hébergement et la mise à disposition de fichiers relatifs à T_EX et L^AT_EX. Toutes les extensions, classes et documentations dont leurs auteurs souhaitent faire profiter la communauté y sont rassemblées.

Le site fédérant l'accès à ces serveurs dispose d'un moteur de recherche :

■ <http://www.ctan.org/>

L'ensemble logiciel T_EX Live (voir le paragraphe 1.4) intègre tous les fichiers disponibles sur le CTAN dont la licence l'autorise.

8.2 La commande `texdoc`

La commande `texdoc` permet d'afficher la documentation d'une extension ou d'une classe à partir de son nom.

Elle s'utilise dans un interpréteur de commandes avec le nom de l'extension ou de la classe comme argument :

```
texdoc babel
```

Il existe parfois plusieurs documentations. Par défaut, `texdoc` affiche la première. Pour pouvoir choisir celle à afficher parmi la liste des documentations disponibles, on utilise l'option `-l` :

```
texdoc -l babel
```

Bien entendu, la documentation de `texdoc` est également disponible :

```
texdoc texdoc
```

8.3 Modifier les marges ou le format de papier : l'extension `geometry`

Les marges par défaut des classes standards (voir le paragraphe 3.1.2.1) sont assez généreuses. Celles des classes KOMA-Script (voir le paragraphe 8.7.1) sont un peu plus réduites. Sans tomber dans l'excès qui consiste à les diminuer au-delà du raisonnable, il est parfois utile de réduire les marges d'un document.

L'extension `geometry` permet de modifier les marges. Elle se charge dans le préambule du document avec une ou plusieurs options permettant d'indiquer la taille des marges.

Utilisée sans option, l'extension `geometry` définit des marges plus réduites que les marges standards :

```
\usepackage{geometry}
```

L'option `margin` permet de fixer la même taille aux quatre marges (tête, pied, gauche et droite) du document :

```
\usepackage[margin=2cm]{geometry}
```

Les options `top` et `bottom` permettent respectivement de fixer la taille des marges de tête et de pied :

```
\usepackage[top=2cm,bottom=2cm]{geometry}
```

Les options `left` et `right` permettent respectivement de fixer la taille des marges gauche et droite :

```
\usepackage[left=2cm,right=2cm]{geometry}
```

8.4. Définir l'en-tête et le pied de page : l'extension fancyhdr

Lorsque le document est destiné à être mis en pages en recto-verso (grâce à l'option de classe `twoside`), la marge intérieure (vers la reliure) est généralement plus petite que la marge extérieure. On utilise dans ce cas les options `inner` et `outer` :

```
\usepackage[inner=2cm,outer=3cm]{geometry}
```

Il est bien entendu possible de fixer la taille de toutes les marges :

```
\usepackage[top=2cm,bottom=2cm,inner=2cm,outer=3cm]{geometry}
```

Enfin, l'extension `geometry` permet également de spécifier le format de papier lorsque celui-ci n'est pas standard. Dans ce cas, l'option de classe `a4paper` n'est pas utilisée et le format de papier est indiqué grâce aux options `paperwidth` et `paperheight` de l'extension `geometry` :

```
\usepackage[paperwidth=17cm,paperheight=24cm]{geometry}
```

Dans ce cas, les marges utilisées sont celles définies par défaut par l'extension `geometry`. Il est possible de les spécifier explicitement en plus des options `paperwidth` et `paperheight` comme vu précédemment.

8.4 Définir l'en-tête et le pied de page : l'extension fancyhdr

L'extension `fancyhdr` permet très simplement de définir le contenu de l'en-tête et du pied de page.

La figure 8.1 montre la structure que l'extension `fancyhdr` donne à l'en-tête et au pied de page.

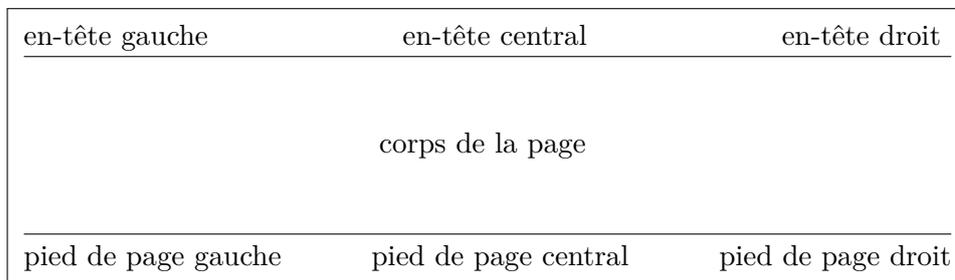


FIGURE 8.1 – Structure de l'en-tête et du pied de page avec l'extension `fancyhdr`.

L'en-tête et le pied de page sont constitués chacun d'une partie gauche, d'une partie centrale et d'une partie droite. Chacune de ces parties peut être définie indépendamment des autres. L'en-tête et le pied de page peuvent être séparés du corps du document par un filet.

La commande `\pagestyle`, utilisée dans le préambule du document avec l'argument `fancy`, permet d'utiliser l'extension `fancyhdr` pour la mise en pages de l'en-tête et du pied de page :

```
\pagestyle{fancy}
```

Ceci doit être complété, si nécessaire, par la définition de l'épaisseur des filets (voir le paragraphe 8.4.1). En fonction de la façon dont le document doit être mis en pages, on utilisera ensuite les commandes indiquées dans le paragraphe 8.4.2 pour un document mis en pages en recto seul ou les commandes indiquées dans le paragraphe 8.4.3 pour un document mis en pages en recto-verso.

Toutes les commandes décrites par la suite sont à utiliser dans le préambule du document.

8.4.1 Les filets

L'épaisseur du filet placé au-dessous de l'en-tête peut être définie au moyen de la commande :

```
\renewcommand{\headrulewidth}{0.4pt}
```

Fixer l'épaisseur du filet à `0pt` le supprime.

De même, l'épaisseur du filet placé au-dessus du pied de page peut être définie au moyen de la commande :

```
\renewcommand{\footrulewidth}{0.4pt}
```

Par défaut, le filet de l'en-tête a une épaisseur de `0.4pt` et celui du pied de page a une épaisseur de `0pt`.

8.4.2 En-tête et pied de page dans un document mis en pages en recto seul

Les commandes `\lhead`, `\chead` et `\rhead` permettent respectivement de définir la partie gauche, la partie centrale et la partie droite de l'en-tête.

Les commandes `\lfoot`, `\cfoot` et `\rfoot` permettent respectivement de définir la partie gauche, la partie centrale et la partie droite du pied de page.

L'extension `fancyhdr` ayant une définition par défaut de certaines parties, il est préférable de les définir toutes, en mettant un argument vide à chaque partie qui doit l'être :

```
\lhead{}
\chead{}
\rhead{\textsc{Le titre de mon document}}
\lfoot{}
\cfoot{\thepage}
\rfoot{}
```

La commande `\thepage` (qui est gérée automatiquement par \LaTeX) permet d'afficher le numéro de page.

8.4.3 En-tête et pied de page dans un document mis en pages en recto-verso

Dans un document mis en pages en recto-verso (grâce à l'option de classe `twoside`), il est d'usage d'échanger les parties gauche et droite de l'en-tête et du pied de page entre les pages paire et impaire se faisant face.

Pour cela, l'extension `fancyhdr` dispose des commandes `\fancyhead` et `\fancyfoot`, qui permettent de tenir compte de la parité de la page.

Par exemple, l'en-tête et le pied de page de ce manuel sont définis ainsi :

```
\fancyhead{}
\fancyhead[LE]{\textit{\nouppercase{\leftmark}}}
\fancyhead[RO]{\textit{\nouppercase{\rightmark}}}
\fancyfoot{}
\fancyfoot[C]{\thepage}
```

Les commandes `\fancyhead` et `\fancyfoot`, utilisées avec un argument vide, permettent de remettre à zéro la définition des différentes parties de l'en-tête et du pied de page.

La commande `\leftmark` (qui est gérée automatiquement par \LaTeX) permet d'afficher le numéro du chapitre courant suivi de son titre.

La commande `\rightmark` (qui est gérée automatiquement par \LaTeX) permet d'afficher le numéro de la section dans laquelle on se trouve au bas de la page suivi de son titre.

Ces titres sont par défaut affichés après avoir été convertis en majuscules. La commande `\nouppercase`, fournie par l'extension `fancyhdr`, permet de les afficher tels qu'ils ont été saisis dans le document.

L'argument optionnel des commandes `\fancyhead` et `\fancyfoot` permet de déterminer dans quelles conditions doit être imprimé l'argument obligatoire.

L'argument optionnel des commandes `\fancyhead` et `\fancyfoot` est constitué d'une combinaison des lettres indiquées dans le tableau 8.1 ou d'une liste de ces combinaisons, chaque élément étant séparé du suivant par une virgule.

L	partie gauche (<i>left</i>)
C	partie centrale (<i>center</i>)
R	partie droite (<i>right</i>)
E	pages paires (<i>even</i>)
O	pages impaires (<i>odd</i>)

TABLE 8.1 – *Arguments optionnels utilisables avec les commandes `\fancyhead` et `\fancyfoot`.*

Ainsi, LE s'applique à la partie gauche des pages paires, RO à la partie droite des pages impaires, LE,RO à la partie gauche des pages paires et à la partie droite des pages impaires et ainsi de suite.

Lorsque le numéro de page est placé dans l'en-tête, il l'est du côté extérieur des pages (c'est-à-dire à gauche des pages paires et à droite des pages impaires). Le titre courant est alors placé du côté intérieur (c'est-à-dire l'inverse). L'exemple suivant permet de réaliser cette mise en pages :

```
\fancyhead{}
\fancyhead[RE]{\textit{\nouppercase{\leftmark}}}
\fancyhead[LO]{\textit{\nouppercase{\rightmark}}}
\fancyhead[LE,RO]{\thepage}
\fancyfoot{}
```

À noter que les pages d'ouverture des chapitres ne tiennent pas compte de l'en-tête et du pied de page définis par l'extension `fancyhdr`. Une page d'ouverture de chapitre n'a pas d'en-tête et son pied de page ne comporte que le numéro de page en partie centrale.

8.5 Changer la police de caractères

Nous avons utilisé jusqu'ici l'extension `lmodern` afin de rédiger nos documents avec la police de caractères Latin Modern. De nombreuses autres polices de caractères sont utilisables avec L^AT_EX. Certaines nécessitent l'utilisation de commandes un peu compliquées que nous n'aborderons pas mais d'autres sont utilisables de la même façon que Latin Modern, c'est-à-dire en chargeant simplement une extension.

Le tableau 8.2 répertorie quelques extensions permettant de déterminer la police de caractères utilisée dans le document.

Extension	Option	Nom de la police	Figure
anttor	math	Antykwa Toruńska	8.2
arev		Arev Sans (Bitstream Vera Sans)	8.3
ccfonts		Concrete	8.4
cmbright		Computer Modern Bright	8.5
fourier		Fourier-GUTenberg (Adobe Utopia)	8.6
fouriernc		New Century Schoolbook	8.7
gfsartemis		Greek Font Society Artemisia	8.8
iwona	math	Iwona	8.9
kpfonts		Kp-Fonts	8.10
kurier	math	Kurier	8.11
lmodern		Latin Modern	8.12
mathdesign	charter	Bitstream Charter	8.13
mathdesign	utopia	Adobe Utopia	8.14
mathpazo		Adobe Palatino	8.15
mathptmx		Times	8.16

TABLE 8.2 – Extensions permettant de déterminer la police de caractères (avec mode mathématique).

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

$$\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \quad \int_a^b \frac{x}{\sqrt{x^2-1}} dx = \sqrt{b^2-1} - \sqrt{a^2-1}$$

FIGURE 8.2 – Exemple de document réalisé avec l'extension *anttor* (option *math*).

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

$$\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \quad \int_a^b \frac{x}{\sqrt{x^2-1}} dx = \sqrt{b^2-1} - \sqrt{a^2-1}$$

FIGURE 8.3 – Exemple de document réalisé avec l'extension *arev*.

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

$$\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \quad \int_a^b \frac{x}{\sqrt{x^2-1}} dx = \sqrt{b^2-1} - \sqrt{a^2-1}$$

FIGURE 8.4 – Exemple de document réalisé avec l'extension *ccfonts*.

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

$$\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \quad \int_a^b \frac{x}{\sqrt{x^2-1}} dx = \sqrt{b^2-1} - \sqrt{a^2-1}$$

FIGURE 8.5 – Exemple de document réalisé avec l'extension *cmbright*.

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

$$\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \quad \int_a^b \frac{x}{\sqrt{x^2-1}} dx = \sqrt{b^2-1} - \sqrt{a^2-1}$$

FIGURE 8.6 – Exemple de document réalisé avec l'extension *fourier*.

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

$$\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \quad \int_a^b \frac{x}{\sqrt{x^2-1}} dx = \sqrt{b^2-1} - \sqrt{a^2-1}$$

FIGURE 8.7 – Exemple de document réalisé avec l'extension *fouriernc*.

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu!

$$\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \quad \int_a^b \frac{x}{\sqrt{x^2-1}} dx = \sqrt{b^2-1} - \sqrt{a^2-1}$$

FIGURE 8.8 – Exemple de document réalisé avec l'extension *gfsartemis*ia.

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu!

$$\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \quad \int_a^b \frac{x}{\sqrt{x^2-1}} dx = \sqrt{b^2-1} - \sqrt{a^2-1}$$

FIGURE 8.9 – Exemple de document réalisé avec l'extension *iwona* (option *math*).

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu!

$$\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \quad \int_a^b \frac{x}{\sqrt{x^2-1}} dx = \sqrt{b^2-1} - \sqrt{a^2-1}$$

FIGURE 8.10 – Exemple de document réalisé avec l'extension *kpfonts*.

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

$$\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \quad \int_a^b \frac{x}{\sqrt{x^2-1}} dx = \sqrt{b^2-1} - \sqrt{a^2-1}$$

FIGURE 8.11 – Exemple de document réalisé avec l'extension *kurier* (option *math*).

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

$$\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \quad \int_a^b \frac{x}{\sqrt{x^2-1}} dx = \sqrt{b^2-1} - \sqrt{a^2-1}$$

FIGURE 8.12 – Exemple de document réalisé avec l'extension *lmodern*.

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

$$\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \quad \int_a^b \frac{x}{\sqrt{x^2-1}} dx = \sqrt{b^2-1} - \sqrt{a^2-1}$$

FIGURE 8.13 – Exemple de document réalisé avec l'extension *mathdesign* (option *charter*).

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

$$\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \quad \int_a^b \frac{x}{\sqrt{x^2-1}} dx = \sqrt{b^2-1} - \sqrt{a^2-1}$$

FIGURE 8.14 – Exemple de document réalisé avec l'extension *mathdesign* (option *utopia*).

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

$$\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \quad \int_a^b \frac{x}{\sqrt{x^2-1}} dx = \sqrt{b^2-1} - \sqrt{a^2-1}$$

FIGURE 8.15 – Exemple de document réalisé avec l'extension *mathpazo*.

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

$$\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \quad \int_a^b \frac{x}{\sqrt{x^2-1}} dx = \sqrt{b^2-1} - \sqrt{a^2-1}$$

FIGURE 8.16 – Exemple de document réalisé avec l'extension *mathptmx*.

Ces extensions fournissent la police de caractères utilisée pour le texte mais également une police mathématique assortie. Pour cela, avec les extensions `anttor`, `iwona` et `kurier`, il est nécessaire d'utiliser l'option `math`. L'extension `mathdesign`, quant à elle, s'utilise soit avec l'option `charter` pour utiliser la police Bitstream Charter soit avec l'option `utopia` pour utiliser la police Adobe Utopia.

Les extensions `arev`, `cmbright`, `gfsartemisia`, `kpfonts`, `lmodern` et `mathdesign` fournissent un ensemble cohérent de polices de caractères permettant de couvrir toutes les commandes de changement de style du tableau 4.6.

Les extensions `anttor`, `ccfonts`, `fourier`, `fouriernc`, `iwona`, `kurier`, `mathpazo` et `mathptmx`, en revanche, ont besoin d'être complétées pour pouvoir utiliser les commandes `\textsf` et `\texttt` (ou d'autres commandes ou environnements les utilisant). Les extensions le permettant sont indiquées dans le tableau 8.3.

Ainsi, il convient de choisir une extension correspondant à la commande `\textsf` et une autre correspondant à la commande `\texttt`. Pour compléter l'extension `fourier`, par exemple, on peut donc utiliser :

```
\usepackage{fourier}
\usepackage{helvet}
\usepackage{courier}
```

Par ailleurs, il est préférable d'utiliser l'extension `mathpazo` avec l'option `sc` ou avec l'option `osf`. Ces deux options font utiliser à l'extension `mathpazo` une police de meilleure qualité, avec de vraies petites capitales. En outre, l'option `osf` permet d'utiliser des chiffres français (0123456789) plutôt que des chiffres anglais (0123456789).

Les extensions répertoriées dans le tableau 8.4 permettent d'utiliser des polices de caractères ne disposant pas d'une police mathématique assortie. Si nécessaire, l'extension `eulervm` permet de leur ajouter une police mathématique générique (dont on peut voir un exemple dans la figure 8.24).

Les extensions `bera`, `bookman` et `newcent` fournissent un ensemble cohérent de polices de caractères permettant de couvrir toutes les commandes de changement de style du tableau 4.6.

Les extensions `chancery`, `charter`, `cyklop`, `tgbonum`, `tgchorus`, `tgpagella`, `tgschola` et `tgtermes`, en revanche, ont besoin d'être complétées par des extensions du tableau 8.3.

L'extension `libertine`, quant à elle, n'a besoin d'être complétée que par une extension correspondant à la commande `\texttt`.

Pour approfondir le sujet, vous pouvez lire [10, chapitre 7 et § 8.8] et consulter le site suivant, qui présente de nombreuses polices de caractères utilisables avec L^AT_EX :

Extension	Nom de la police	Commande	Figure
avant	Avant Garde	<code>\textsf</code>	8.17
helvet	Helvetica	<code>\textsf</code>	8.18
tgadventor	TeX Gyre Adventor	<code>\textsf</code>	8.19
tgheros	TeX Gyre Heros	<code>\textsf</code>	8.20
courier	Courier	<code>\texttt</code>	8.21
inconsolata	Inconsolata	<code>\texttt</code>	8.22
tgcursor	TeX Gyre Cursor	<code>\texttt</code>	8.23

TABLE 8.3 – *Extensions permettant de déterminer une police de caractères complémentaire.*

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

FIGURE 8.17 – *Exemple de document réalisé avec l'extension avant.*

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

FIGURE 8.18 – *Exemple de document réalisé avec l'extension helvet.*

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

FIGURE 8.19 – *Exemple de document réalisé avec l'extension tgadventor.*

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

FIGURE 8.20 – Exemple de document réalisé avec l'extension *tgheros*.

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

FIGURE 8.21 – Exemple de document réalisé avec l'extension *courier*.

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

FIGURE 8.22 – Exemple de document réalisé avec l'extension *inconsolata*.

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

FIGURE 8.23 – Exemple de document réalisé avec l'extension *tgcursor*.

$$\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \quad \int_a^b \frac{x}{\sqrt{x^2-1}} dx = \sqrt{b^2-1} - \sqrt{a^2-1}$$

FIGURE 8.24 – Exemple de document réalisé avec l'extension *eulervm*.

Extension	Nom de la police	Figure
bera	Bera	8.25
bookman	Bookman	8.26
chancery	Zapf Chancery	8.27
charter	Charter	8.28
cyklop	Cyklop	8.29
libertine	Linux Libertine	8.30
newcent	New Century Schoolbook	8.31
tgbonum	T _E X Gyre Bonum	8.32
tgchorus	T _E X Gyre Chorus	8.33
tgpagella	T _E X Gyre Pagella	8.34
tgschola	T _E X Gyre Schola	8.35
tgtermes	T _E X Gyre Termes	8.36

TABLE 8.4 – *Extensions permettant de déterminer la police de caractères (sans mode mathématique).*

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

FIGURE 8.25 – *Exemple de document réalisé avec l'extension bera.*

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

FIGURE 8.26 – *Exemple de document réalisé avec l'extension bookman.*

*Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier...
L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le
dire : ton Antoine commence à me les briser menu !*

FIGURE 8.27 – *Exemple de document réalisé avec l'extension chancery.*

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

FIGURE 8.28 – Exemple de document réalisé avec l'extension *charter*.

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

FIGURE 8.29 – Exemple de document réalisé avec l'extension *cyklop*.

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

FIGURE 8.30 – Exemple de document réalisé avec l'extension *libertine*.

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

FIGURE 8.31 – Exemple de document réalisé avec l'extension *newcent*.

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

FIGURE 8.32 – *Exemple de document réalisé avec l'extension tgbonum.*

*Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier...
L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire :
ton Antoine commence à me les briser menu !*

FIGURE 8.33 – *Exemple de document réalisé avec l'extension tgchorus.*

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

FIGURE 8.34 – *Exemple de document réalisé avec l'extension tgpagella.*

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

FIGURE 8.35 – *Exemple de document réalisé avec l'extension tgschola.*

Patricia, mon petit, je ne voudrais pas te paraître vieux jeu et encore moins grossier... L'homme de la pampa parfois rude, reste toujours courtois... Mais la vérité m'oblige à te le dire : ton Antoine commence à me les briser menu !

FIGURE 8.36 – Exemple de document réalisé avec l'extension *tgterms*.

■ <http://www.tug.dk/FontCatalogue/>

8.6 Inclure des graphiques

L'inclusion d'un graphique dans un document L^AT_EX peut se faire de deux façons :

- en faisant appel à une image stockée dans un fichier extérieur (voir le paragraphe 8.6.1) ;
- en décrivant le graphique au sein-même du document L^AT_EX au moyen d'un langage de description adapté (voir le paragraphe 8.6.2).

Cette seconde façon de faire présente l'avantage de pouvoir utiliser des expressions rédigées avec L^AT_EX pour le texte contenu dans le graphique. Ceci permet d'avoir la même police de caractères dans le graphique que dans le texte du document, ce qui offre une meilleure cohérence visuelle. Ceci permet également d'utiliser des expressions mathématiques dans le graphique. Un exemple de graphique réalisé de cette façon est présenté dans la figure 8.37.

Pour ces raisons, c'est souvent cette approche qui est retenue. En témoigne le grand nombre d'extensions de L^AT_EX permettant de décrire des graphiques au sein-même d'un document et dont les plus répandues sont présentées dans [8].

L'un des derniers arrivants dans ce domaine, malheureusement trop récent pour disposer d'une description dans [8] mais néanmoins très prometteur, s'appelle TikZ. Nous en discuterons rapidement au paragraphe 8.6.2 puisqu'il dispose d'une excellente documentation d'introduction en français à laquelle vous pourrez vous référer pour en savoir plus.

8.6.1 La commande `\includegraphics`

Il est possible d'inclure des fichiers aux formats PDF, PNG et JPEG dans un document L^AT_EX, grâce à l'extension `graphicx`.

Pour inclure le fichier, on utilise la commande `\includegraphics` :

```
\includegraphics{fichier}
```

Il est préférable de placer cette commande dans une figure, ce qui permet de donner à l'image une légende et un numéro et surtout de pouvoir la déplacer si elle risque d'être coupée en bas de page.

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

```
\includegraphics [width=largeur]{fichier}
```

ou en hauteur :

```
\includegraphics [height=hauteur]{fichier}
```

Voici donc le code type d'inclusion d'un fichier au format PDF dans un document L^AT_EX :

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

\usepackage[french]{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage{lmodern}
\usepackage{graphicx}
\usepackage{microtype}

\usepackage{hyperref}

\begin{document}

[...]

\begin{figure}[!htbp]
\centering
\includegraphics [width=10cm]{fichier.pdf}
\caption{Image}
\label{figure:image}
\end{figure}

[...]

\end{document}
```

8.6.2 L'extension TikZ

L'extension TikZ permet d'inclure des graphiques dans un document L^AT_EX au moyen d'un langage de description. Un exemple de graphique réalisé avec TikZ est présenté dans la figure 8.37.

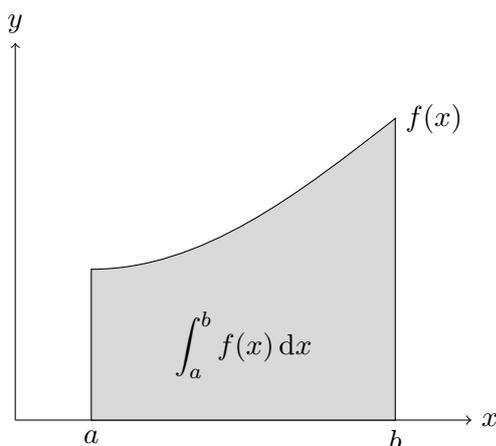


FIGURE 8.37 – Exemple de graphique réalisé avec l'extension TikZ.

L'utilisation de TikZ nécessite le chargement de l'extension tikz dans le préambule du document :

```
\usepackage{tikz}
```

L'inclusion d'un graphique se fait ensuite dans le corps du document au moyen de l'environnement `tikzpicture`, qu'on place généralement à l'intérieur d'une figure.

Pour vous donner une idée de la façon dont on décrit un graphique avec TikZ, voici le code source de celui correspondant à la figure 8.37 :

```

\begin{tikzpicture}

\draw[->] (0,0) -- (6,0) node[right] {$x$} ;
\draw[->] (0,0) -- (0,5) node[above] {$y$} ;

\filldraw[fill=gray!30]
  (1,0) node[below] {$a$}
  -- (1,2)
  cos (5,4) node[right] {$f(x)$}
  -- (5,0) node[below] {$b$}
  -- cycle ;

\draw (3,1) node {$\displaystyle \int_a^b f(x) \, \mathrm{d}x$} ;

\end{tikzpicture}

```

Nous n'irons pas plus loin ici car il existe une excellente documentation d'introduction à TikZ, en français. Vous pouvez donc vous référer à [19] pour en savoir plus sur TikZ.

Par la suite, la lecture de la documentation officielle de TikZ [18] vous permettra de parfaire vos connaissances.

Enfin, le site suivant présente de nombreux exemples de graphiques réalisés avec TikZ, accompagnés de leur code source, ce qui vous permettra de vous donner un large aperçu de ce qu'il est possible de faire avec TikZ :

■ <http://www.texample.net/tikz/>

Signalons également l'existence de l'extension PGFPLOTS [17], basée sur TikZ, qui permet de réaliser très simplement des graphiques à partir des coordonnées d'un ensemble de points :

■ <http://pgfplots.sourceforge.net/>

8.7 Utiliser d'autres classes que les classes standards

Il existe d'autres classes que les classes standards (qui sont indiquées au paragraphe 3.1.2.1). Certaines permettent de concevoir de nouveaux types de document, comme la classe `beamer` (voir le paragraphe 11.2) qui est utilisée pour concevoir des présentations destinées à être projetées avec un vidéoprojecteur, mais d'autres ont vocation à remplacer les classes standards. À cet effet, elles s'utilisent exactement de la même façon mais proposent des possibilités supplémentaires.

C'est le cas de l'ensemble KOMA-Script et de la classe `memoir`.

8.7.1 KOMA-Script

L'ensemble KOMA-Script fournit des classes `scrartcl`, `scrreprt` et `scrbook` destinées à remplacer respectivement les classes standards `article`, `report` et `book`, comme indiqué dans le tableau 8.5.

Classe standard	Classe KOMA-Script
<code>article</code>	<code>scrartcl</code>
<code>report</code>	<code>scrreprt</code>
<code>book</code>	<code>scrbook</code>

TABLE 8.5 – Correspondance entre classes standards et classes KOMA-Script.

Pour utiliser une classe KOMA-Script, il suffit de changer l'argument de la commande `\documentclass` dans un document existant ou de créer directement un nouveau document utilisant cette classe. Rien d'autre n'est à changer.

On peut omettre les options de classe `a4paper` et `11pt`, qui sont utilisées par défaut par les classes KOMA-Script.

KOMA-Script fournit également la classe `scrlettr2`, qui permet d'écrire des lettres (voir le paragraphe 9.2). À la différence des classes indiquées dans le tableau 8.5, la classe `scrlettr2` ne peut être directement utilisée en remplacement de la classe standard `letter` car elle utilise une syntaxe différente.

L'intérêt des classes KOMA-Script est qu'elles offrent des possibilités absentes des classes standards ou nécessitant l'utilisation d'extensions. Elles permettent en particulier de modifier les marges du document, la présentation des titres, des légendes, de la table des matières et bien d'autres choses.

La description des possibilités offertes par KOMA-Script dépasse le cadre de ce manuel. Pour en savoir davantage sur KOMA-Script, vous pouvez vous référer à sa documentation :

<http://mirror.ctan.org/macros/latex/contrib/koma-script/scrguien.pdf>

8.7.2 La classe `memoir`

La classe `memoir` est destinée à remplacer les classes standards `book` et `report`. Elle intègre par ailleurs les fonctionnalités d'un grand nombre d'extensions.

Pour qui doit rédiger un ouvrage volumineux et complexe, elle est particulièrement intéressante.

Là encore, la description des possibilités offertes par la classe `memoir` dépasse le cadre de ce manuel. Pour en savoir davantage sur la classe `memoir`, vous pouvez vous référer à sa documentation :

<http://mirror.ctan.org/macros/latex/contrib/memoir/memman.pdf>

Chapitre 9

Écrire une lettre : les classes `letter` et `scrlttr2`

Les chapitres précédents vous ont montré que `LATEX` était parfaitement adapté à la conception de documents complexes, avec des tableaux, des formules mathématiques, des graphiques, une bibliographie, un index et une table des matières, mais il permet aussi de réaliser des lettres, qui sont généralement des documents plus simples, au moyen des classes `letter` et `scrlttr2`. Cette dernière, qui offre plus de possibilités et respecte mieux les habitudes de présentation françaises, sera étudiée plus en détail.

9.1 La classe `letter`

Un document réalisé avec la classe `letter` peut contenir plusieurs lettres, chacune figurant dans un environnement `letter`. Celles-ci peuvent comporter la même adresse d'expéditeur et la même signature si ces éléments sont indiqués dans le préambule.

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.

La figure 9.1 montre le code source d'une lettre réalisée avec la classe `letter`, la figure 9.2 montre le document généré.

Pour plus d'informations sur la classe `letter`, vous pouvez consulter [5, pages 84–86].

9.1.1 Les commandes mutualisables

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

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

\usepackage[french]{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage{lmodern}
\usepackage{microtype}

% décommenter pour générer une page d'étiquettes
%\makelabels

\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 nouvelle babasse ;

\item le dernier \LaTeX\ ;

\item le manuel \textit{Apprends \LaTeX\ !} ;

\item la bible de \LaTeX, le \textit{\LaTeX\ Companion}.

\end{itemize}

\closing{Joyeux Noël,}

\end{letter}

\end{document}
```

FIGURE 9.1 – Exemple de lettre réalisée avec la classe `letter` (code source).

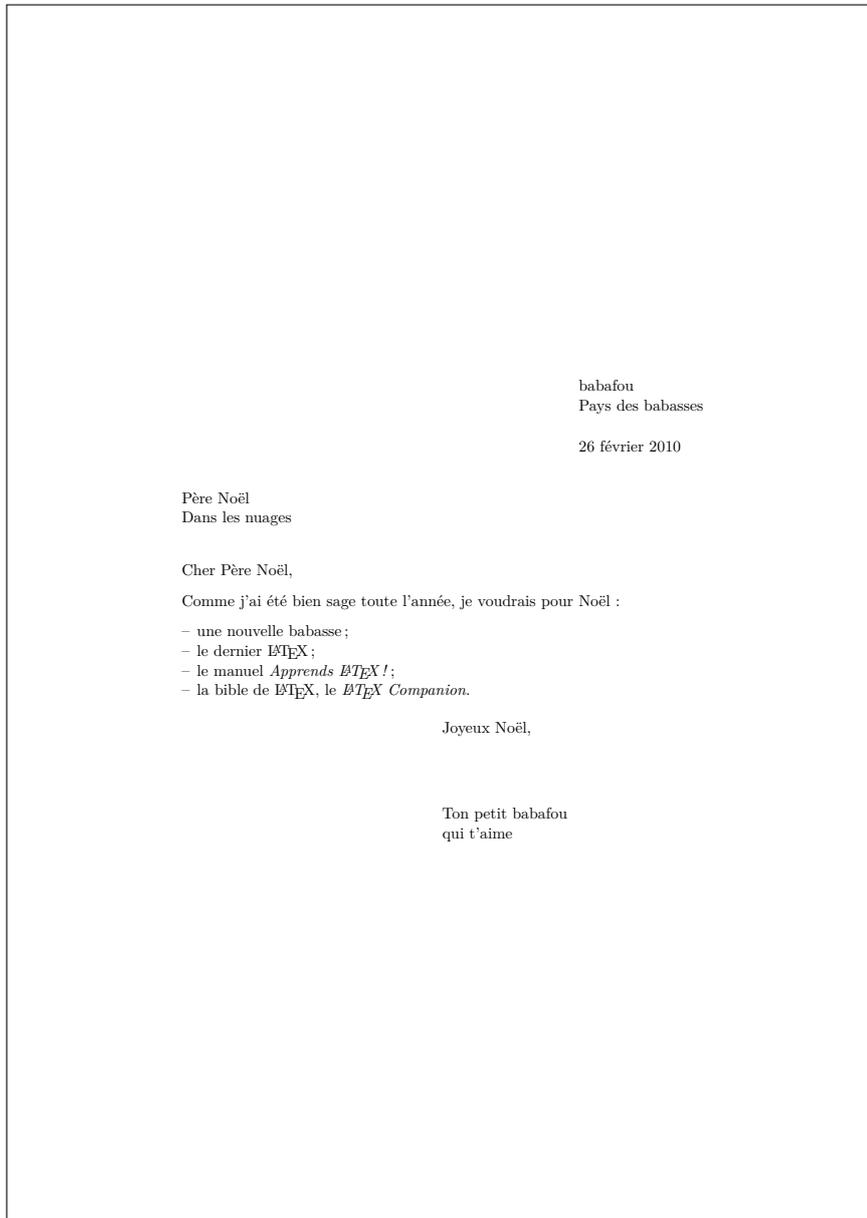


FIGURE 9.2 – Exemple de lettre réalisée avec la classe letter (document généré).

9.1.1.1 La commande `\address`

La commande `\address` permet de définir le nom et l'adresse postale de l'expéditeur, qui seront imprimés en haut à droite de la lettre. Les différentes lignes du nom et de l'adresse sont séparées par `\\` :

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

9.1.1.2 La commande `\signature`

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

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

9.1.2 L'environnement `letter`

Chaque lettre est contenue dans un environnement `letter` dont l'argument est l'adresse du destinataire. Les différentes lignes de l'adresse sont séparées par `\\`.

La commande `\opening` permet de commencer la lettre proprement dite en imprimant son argument. 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 figurant dans un environnement `letter`. Toutes ces lettres auront la même adresse d'expéditeur et la même signature (commandes `\address` et `\signature`) si celles-ci sont définies dans le préambule. Il est possible de définir une adresse d'expéditeur et une signature différentes pour une lettre particulière en utilisant les commandes `\address` et `\signature` entre `\begin{letter}` et `\opening`.

9.2 La classe `scrlttr2`

La classe `scrlttr2` fait partie de l'ensemble KOMA-Script (voir le paragraphe 8.7.1). À la différence des autres classes fournies par cet ensemble, la classe `scrlttr2` n'est pas un remplaçant direct de la classe `letter`, c'est-à-dire qu'on ne peut pas prendre une lettre écrite avec la classe `letter` et simplement remplacer l'argument de la commande `\documentclass` pour utiliser la classe `scrlttr2`. Les différences sont néanmoins minimes.

La classe `scrlttr2` offre plus de possibilités que la classe `letter` et respecte mieux les habitudes de présentation françaises.

La figure 9.3 montre le code source d’une lettre réalisée avec la classe `scr1ttr2`, la figure 9.4 montre le document généré.

9.2.1 L’option de classe **NF**

L’option de classe **NF** permet de générer une lettre respectant la norme française NF Z 11-001 :

```
\documentclass[NF]{scr1ttr2}
```

En particulier, ceci positionne l’adresse du destinataire au bon endroit pour qu’elle apparaisse correctement dans une enveloppe à fenêtre une fois pliée.

Rappelons que l’utilisation des options de classe `a4paper` et `11pt` n’est pas nécessaire avec les classes KOMA-Script (voir le paragraphe 8.7.1).

9.2.2 Les variables

Le paramétrage de la lettre se fait au moyen de *variables* qui sont spécifiées grâce à la commande `\setkomavar`, dont le premier argument indique la variable à renseigner et le second la valeur à lui attribuer :

```
\setkomavar{variable}{valeur}
```

La commande `\setkomavar` est généralement utilisée dans le préambule, auquel cas les variables renseignées de cette façon ont la même valeur pour toutes les lettres du document. Il est également possible de l’utiliser dans le corps du document, à l’extérieur des environnements `letter`, auquel cas les variables renseignées de cette façon ont la même valeur pour toutes les lettres qui suivent.

Si la valeur d’une variable doit être scindée en plusieurs lignes, celles-ci doivent être séparées par `\\`.

Le tableau 9.1 indique les plus employées des variables utilisables avec la classe `scr1ttr2` (il en existe un peu plus d’une trentaine).

Si la variable `date` n’est pas renseignée, la date du jour de la compilation est indiquée automatiquement dans la lettre.

Contrairement à la classe `letter`, dans laquelle la commande `\address` sert à définir le nom et l’adresse de l’expéditeur, la classe `scr1ttr2` utilise deux variables différentes, `fromname` pour le nom et `fromaddress` pour l’adresse.

Si la variable `signature` n’est pas renseignée, la valeur de la variable `fromname` est utilisée comme signature.

```
\documentclass[NF]{scrlettr2}

\usepackage[french]{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage{lmodern}
\usepackage{microtype}

\setkomavar{fromname}{babafou}
\setkomavar{fromaddress}{Pays des babasses}
\setkomavar{subject}{Liste pour Noël}
\setkomavar{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 nouvelle babasse ;

\item le dernier \LaTeX\ ;

\item le manuel \textit{Apprends \LaTeX\ !} ;

\item la bible de \LaTeX, le \textit{\LaTeX\ Companion}.

\end{itemize}

\closing{Joyeux Noël,}

\end{letter}

\end{document}
```

FIGURE 9.3 – Exemple de lettre réalisée avec la classe *scrlettr2* (code source).

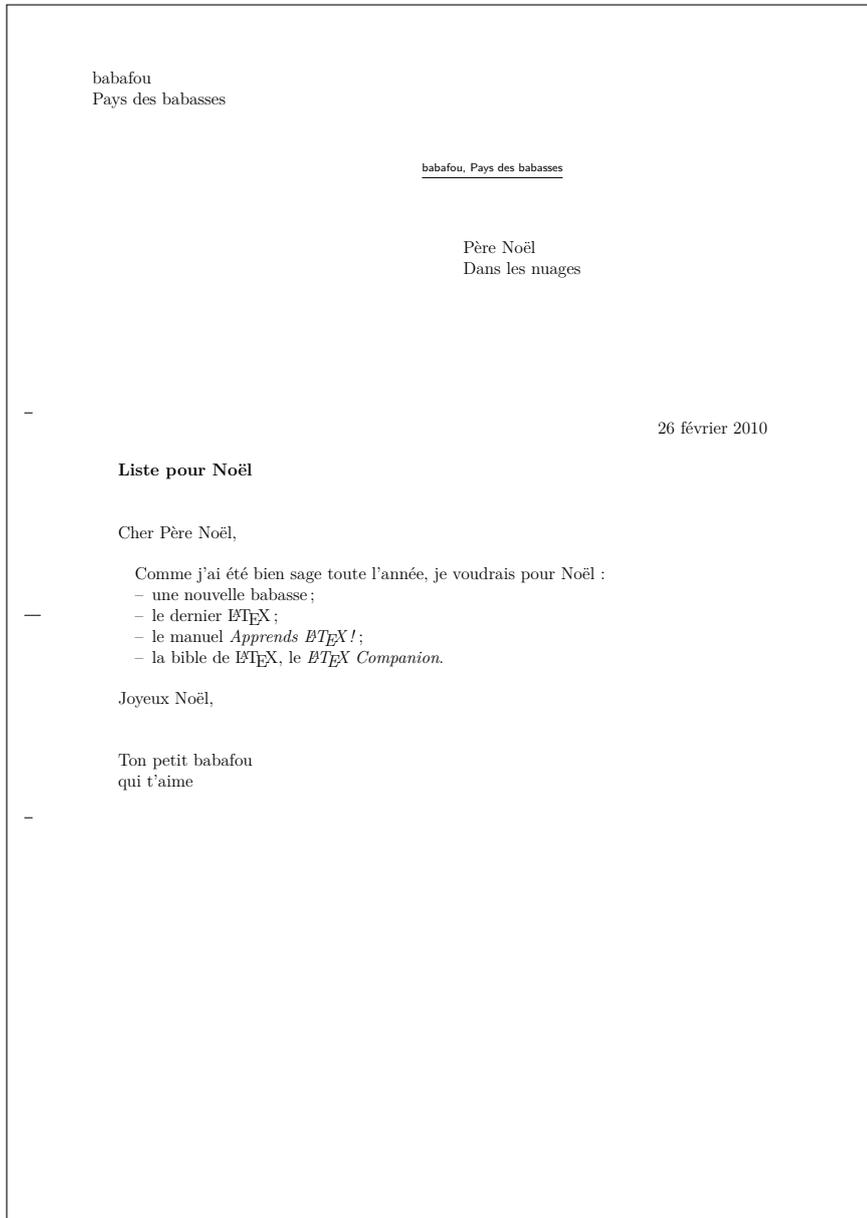


FIGURE 9.4 – Exemple de lettre réalisée avec la classe `scrlltr2` (document généré).

Variable	Signification
<code>date</code>	date
<code>fromaddress</code>	adresse de l'expéditeur
<code>fromname</code>	nom de l'expéditeur
<code>myref</code>	référence de la lettre
<code>signature</code>	signature
<code>subject</code>	objet de la lettre
<code>yourref</code>	référence de la lettre à laquelle on répond

TABLE 9.1 – Variables utilisables avec la classe *scrlltr2*.

9.2.3 Les options

Certains détails de mise en pages peuvent être modifiés grâce à la commande `\KOMAOPTIONS` (qui n'a pas été utilisée dans l'exemple de la figure 9.3). Employée dans le préambule du document, celle-ci impacte toutes les lettres du document. Il est également possible de limiter la portée de certaines options à une seule lettre (voir le paragraphe 9.2.4).

Il existe de nombreuses options. Nous n'en étudierons que deux. Vous pouvez vous reporter à la documentation de KOMA-Script pour avoir une description des autres.

L'option `backaddress` permet de supprimer l'adresse de retour écrite en petits caractères au-dessus de l'adresse du destinataire et ayant vocation à être visible dans la fenêtre de l'enveloppe :

```
\KOMAOPTIONS{backaddress=false}
```

L'option `foldmarks` permet de supprimer les marques facilitant le pliage de la lettre (qui n'ont par exemple pas de sens lorsque la lettre n'est pas envoyée par courrier postal mais par courriel sous forme de fichier PDF) :

```
\KOMAOPTIONS{foldmarks=hp}
```

Lorsqu'on utilise plusieurs options, il est possible de les regrouper dans la même commande `\KOMAOPTIONS` en séparant chaque option de la suivante par une virgule :

```
\KOMAOPTIONS{backaddress=false,foldmarks=hp}
```

qu'on peut également écrire :

```
\KOMAOptions
{
    backaddress      = false ,
    foldmarks        = hp
}
```

9.2.4 L'environnement `letter`

L'environnement `letter` de la classe `scrlltr2` s'utilise exactement comme celui de la classe `letter` (voir le paragraphe 9.1.2).

Pour appliquer certaines options (voir le paragraphe 9.2.3) à une lettre en particulier, on utilise l'argument optionnel de l'environnement `letter` avec la même syntaxe que la commande `\KOMAOptions` :

```
\begin{letter}[foldmarks=hp]{Père Noël \\ Dans les nuages}
```

9.3 Règles typographiques

9.3.1 Adresse

La Poste recommande :

- d'écrire une adresse sur six lignes au maximum ;
- de ne jamais mettre de virgule après le numéro de rue ;
- d'écrire en majuscules la dernière ligne, sans accents ni ponctuation (ce qui est contraire à toutes les règles typographiques mais facilite, paraît-il, la reconnaissance optique) et si possible l'ensemble de l'adresse.

Ce qui donne, par exemple :

```
Séraphin Lampion
51 rue des Bouffonneries
72530 YVRE L EVEQUE
```

<http://www.laposte.fr/Particulier/Envoyez-et-recevez-du-courrier/Les-offres-et-conseils-pas-a-pas/2.-Redaction-des-adresses>
<http://www.orthotypographie.fr/volume-I/accolade-allemand.html#Adresse>

9.3.2 Numéro de téléphone

Un numéro de téléphone (du moins en France) se compose de cinq groupes de deux chiffres séparés par des espaces (pas de point ni de tiret) :

01 23 45 67 89

Il en est de même pour un numéro de télécopieur.

Chapitre 10

Rédiger un CV

De nombreux recruteurs exigent sans vergogne une lettre de motivation et un CV au seul format Word. D'autres, plus ouverts d'esprit et en phase avec leur époque, acceptent volontiers le format PDF. C'est l'occasion de rédiger son CV avec \LaTeX . Nous étudierons d'abord la classe `moderncv`, qui produit probablement le résultat le plus joli, puis la classe `curve` et enfin l'extension `currvita`.

La lecture de l'article [26] complètera ce chapitre. Il contient quelques indications sur la façon de rédiger un CV ainsi qu'un tour d'horizon des classes permettant de rédiger un CV.

Par ailleurs, les règles typographiques indiquées au paragraphe 9.3 au sujet de l'adresse et du numéro de téléphone sont également valables dans un CV.

10.1 La classe `moderncv`

La classe `moderncv` permet très simplement de rédiger un CV d'aspect sobre et efficace, qu'on peut agrémenter d'une touche de couleur et d'une photo.

La figure 10.1 montre le code source d'un CV réalisé avec la classe `moderncv`, la figure 10.2 montre le document généré avec le thème `casual` et la figure 10.3 montre le document généré avec le thème `classic`.

10.1.1 Les thèmes

En effet, l'aspect de l'en-tête et du pied de page du CV dépend du *thème* utilisé. La commande `\moderncvtheme`, utilisée dans le préambule du document, permet d'en définir le thème grâce à son argument, qui peut être soit `casual` soit `classic` :

```
\moderncvtheme{classic}
```

```
1 \documentclass[a4paper,11pt]{moderncv}
2
3 \usepackage[french]{babel}
4 \usepackage[T1]{fontenc}
5 \usepackage[utf8]{inputenc}
6 \usepackage{lmodern}
7 \usepackage{microtype}
8
9 \usepackage{hyperref}
10
11 %\moderncvtheme{casual}
12 \moderncvtheme{classic}
13
14 %\setlength{\hintscolumnwidth}{45mm}
15 %\AtBeginDocument{\recomputelengths}
16
17 \firstname{S raphin}
18 \familyname{Lampion}
19 \title{Repr sentant en assurances}
20 \address{Place de la mairie}{99999 MOULINSART}
21 \mobile{06 02 03 04 05}
22 \phone{01 02 03 04 05}
23 \fax{01 02 03 04 05}
24 \email{lampion@mondass.fr}
25 \extrainfo{Mari  \\ beaucoup d'enfants}
26 \photo{seraphin-lampion.jpg}
27
28 \quote{Je vais vous figner un amour de petit contrat...}
29
30 \begin{document}
31
32 \maketitle
33
```

FIGURE 10.1 – Exemple de CV r alis  avec la classe *moderncv* (code source).

```

34 \section{Expérience professionnelle}
35
36 \cventry{1958 -- 1983}{Représentant}{Assurances Mondass}
37 {Paris}{}
38 {
39 \begin{itemize}
40 \item Démarchage à domicile.
41 \item Recherche d'opportunités.
42 \item Établissement de contrats.
43 \end{itemize}
44 }
45
46 \section{Références}
47
48 \subsection{Europe}
49
50 \cvlistdoubleitem{Château de Moulinsart}{Bianca Castafiore}
51
52 \subsection{International}
53
54 \cvlistitem{Général Alcazar}
55
56 \section{Méthodologies}
57
58 \cvcomputer{Humour}{Les histoires de mon oncle Anatole.}
59 {Enfumage}{Je parle beaucoup...}
60
61 \cvcomputer{Opportunisme}{Toutes les occasions sont bonnes à
62 prendre.}{}{}
63
64 \section{Formation}
65
66 \cventry{1955 -- 1958}{Diplôme de représentant}
67 {École de la vie}{Bruxelles}{option « embobinage »}
68 {Une formation ambitieuse qui développe l'assurance en soi.}
69

```

FIGURE 10.1 – Exemple de CV réalisé avec la classe moderncv (code source).

```
70 \section{Langues étrangères}
71
72 \cvlanguage{Belge}{courant}{langue maternelle}
73 \cvlanguage{Espagnol}{notions}{}
74
75 \section{Activités extra-professionnelles}
76
77 \cvline{Association}{Président du « Volant Club »}
78
79 \cvline{Association}{Président de la société philanthropique
80 « Les Joyeux Turlurons »}
81
82 \end{document}
```

FIGURE 10.1 – Exemple de CV réalisé avec la classe `moderncv` (code source).

Comme on peut le voir dans la figure 10.2, le thème `casual` place une partie des informations personnelles dans l’en-tête et le reste dans le pied de page. Le thème `classic`, en revanche, place toutes les informations personnelles dans l’en-tête, comme on peut le voir dans la figure 10.3. Il est plus utile pour un recruteur d’avoir toutes les informations personnelles au même endroit donc il est recommandé d’utiliser plutôt le thème `classic`.

L’argument optionnel de la commande `\moderncvtheme`, qui peut valoir `blue` (c’est la valeur par défaut), `green`, `grey`, `orange` ou `red`, permet de choisir la couleur des rubriques :

```
\moderncvtheme[green]{classic}
```

On peut y ajouter l’option `roman` pour utiliser dans les rubriques une police à empattement plutôt que la police sans empattement utilisée par défaut :

```
\moderncvtheme[green,roman]{classic}
```

10.1.2 Les informations personnelles

Les informations personnelles sont renseignées dans le préambule. On peut en voir un exemple dans la figure 10.1, de la ligne 16 à la ligne 27.

La commande `\firstname` (obligatoire) permet d’indiquer le prénom :

```
\firstname{Séraphin}
```



FIGURE 10.2 – Exemple de CV réalisé avec la classe moderncv, thème casual (document généré).

Place de la mairie
99999 MOULINSART
☎ 06 02 03 04 05
☎ 01 02 03 04 05
☎ 01 02 03 04 05
✉ lampion@mondass.fr
Marié
beaucoup d'enfants



Séraphin Lampion

Représentant en assurances

Je vais vous figoler un amour de petit contrat...

Expérience professionnelle

1958 – 1983 **Représentant, Assurances Mondass, Paris.**
–Démarchage à domicile.
–Recherche d'opportunités.
–Établissement de contrats.

Références

Europe

- Château de Moulinsart
- Bianca Castafiore

International

- Général Alcazar

Méthodologies

Humour Les histoires de mon oncle Anatole. Enfumage Je parle beaucoup...
Opportunisme Toutes les occasions sont bonnes à prendre.

Formation

1955 – 1958 **Diplôme de représentant, École de la vie, Bruxelles, option « embobinage ».**
Une formation ambitieuse qui développe l'assurance en soi.

Langues étrangères

Belge **courant** *langue maternelle*
Espagnol **notions**

Activités extra-professionnelles

Association Président du « Volant Club »
Association Président de la société philanthropique « Les Joyeux Turlurons »

FIGURE 10.3 – Exemple de CV réalisé avec la classe `moderncv`, thème `classic` (document généré).

La commande `\familyname` (obligatoire) permet d'indiquer le nom de famille :

```
\familyname{Lampion}
```

La commande `\title` (optionnelle) permet d'indiquer le titre du CV (qui correspond souvent au poste actuel ou au poste recherché), qui n'est affiché qu'avec le thème `classic` :

```
\title{Représentant en assurances}
```

La commande `\address` (optionnelle) permet d'indiquer l'adresse postale. Son premier argument contient le numéro et la rue, qu'on peut scinder en plusieurs lignes avec `\\` si nécessaire. Son second argument contient le code postal et la ville.

```
\address{Place de la mairie}{99999 MOULINSART}
```

La commande `\mobile` (optionnelle) permet d'indiquer le numéro de téléphone mobile :

```
\mobile{06 02 03 04 05}
```

La commande `\phone` (optionnelle) permet d'indiquer le numéro de téléphone fixe :

```
\phone{01 02 03 04 05}
```

La commande `\fax` (optionnelle) permet d'indiquer le numéro de télécopieur :

```
\fax{01 02 03 04 05}
```

La commande `\email` (optionnelle) permet d'indiquer l'adresse électronique :

```
\email{lampion@mondass.fr}
```

La commande `\extrainfo` (optionnelle) permet d'indiquer des informations supplémentaires, qu'on peut scinder en plusieurs lignes avec `\\` si nécessaire :

```
\extrainfo{Marié \\ beaucoup d'enfants}
```

La commande `\photo` (optionnelle) permet d'indiquer le nom du fichier au format PDF, PNG ou JPEG contenant la photo :

```
\photo{seraphin-lampion.jpg}
```

On peut en fixer la hauteur au moyen de l'argument optionnel de la commande `\photo` :

```
\photo[3cm]{seraphin-lampion.jpg}
```

La commande `\quote` (optionnelle) permet d'indiquer un texte qui sera affiché entre les informations personnelles et les rubriques du CV :

```
\quote{Je vais vous figoler un amour de petit contrat...}
```

Enfin, la commande `\maketitle`, utilisée au début du corps du document, permet d'imprimer ces informations, en fonction du thème choisi.

10.1.3 Les rubriques

Les rubriques du CV sont indiquées au moyen des commandes `\section` et `\subsection` (les autres commandes de hiérarchisation n'existent pas dans la classe `moderncv`).

Elles sont composées dans la couleur indiquée par l'argument optionnel de la commande `\moderncvtheme` (bleu par défaut). Les sections sont précédées d'une barre horizontale de la même couleur.

10.1.4 Le contenu des rubriques

Selon sa nature, le contenu des rubriques peut être indiqué à l'aide d'une des six commandes suivantes.

Quatre d'entre elles utilisent l'espace situé sous la barre horizontale des sections, qu'on appellera *partie gauche*. Toutes utilisent l'espace situé sous le titre des sections, qu'on appellera *partie droite*.

10.1.4.1 La commande `\cventry`

La commande `\cventry` est généralement utilisée pour décrire l'expérience professionnelle et la formation :

```
\cventry{1955 -- 1958}{Diplôme de représentant}
{École de la vie}{Bruxelles}{option « embobinage »}
{Une formation ambitieuse qui développe l'assurance en soi.}
```

La commande `\cventry` accepte six arguments. Certains d'entre eux peuvent ne pas être renseignés, auquel cas il convient de conserver néanmoins les couples d'accolades correspondants comme on peut le voir dans la figure 10.1 à la ligne 36.

Le premier argument est placé dans la partie gauche. Il est utilisé pour indiquer un intervalle temporel correspondant au poste ou à la formation.

Les arguments suivants sont placés dans la partie droite.

Le deuxième argument est composé en gras, suivi du troisième argument, composé en italique. L'un d'entre eux est utilisé pour indiquer le poste occupé ou le diplôme obtenu, l'autre pour indiquer l'entreprise ou l'organisme de formation.

Les quatrième et cinquième arguments sont composés à la suite et sont utilisés pour toute information complémentaire (option suivie dans le cas de la formation, par exemple).

Le sixième argument est composé sous la ligne formée par les précédents et permet de développer le poste occupé ou la formation suivie. On y utilise généralement l'environnement `itemize` comme on peut le voir dans la figure 10.1, de la ligne 37 à la ligne 43.

10.1.4.2 La commande `\cvcomputer`

La commande `\cvcomputer` est utilisée pour indiquer ses compétences en informatique mais on peut l'utiliser pour indiquer tout autre chose dans un domaine différent :

```
\cvcomputer{Humour}{Les histoires de mon oncle Anatole.}
{Enfumage}{Je parle beaucoup...}
```

La commande `\cvcomputer` accepte quatre arguments qui sont composés sous la forme d'un tableau de quatre colonnes. La première colonne est placée dans la partie gauche, les suivantes dans la partie droite. Les deux derniers arguments peuvent ne pas être renseignés, auquel cas il convient de conserver néanmoins les couples d'accolades correspondants comme on peut le voir dans la figure 10.1 à la ligne 61.

Les premier et troisième arguments font office de titre ou de catégorie pour l'argument suivant, les deuxième et quatrième de description.

10.1.4.3 La commande `\cvlanguage`

La commande `\cvlanguage` permet d'indiquer les langues étrangères :

```
\cvlanguage{Belge}{courant}{langue maternelle}  
\cvlanguage{Espagnol}{notions}{}
```

La commande `\cvlanguage` accepte trois arguments. Le premier argument, placé dans la partie gauche, indique la langue. Le deuxième argument, composé en gras, indique le niveau. Le troisième argument permet d'indiquer un commentaire qui est composé en italique. Ce dernier peut ne pas être renseigné, auquel cas il convient de conserver néanmoins le couple d'accolades correspondant comme on peut le voir dans la figure 10.1 à la ligne 72.

10.1.4.4 La commande `\cvline`

La commande `\cvline` affiche ses deux arguments sur une ligne, le premier dans la partie gauche et le second dans la partie droite :

```
\cvline{Association}{Président du « Volant Club »}
```

La commande `\cvline` peut par exemple être utilisée pour indiquer ses activités extra-professionnelles.

10.1.4.5 La commande `\cvlistitem`

La commande `\cvlistitem` affiche son unique argument précédé d'un gros point dans la partie droite :

```
\cvlistitem{Général Alcazar}
```

10.1.4.6 La commande `\cvlistdoubleitem`

La commande `\cvlistdoubleitem` affiche sur la même ligne ses deux arguments, chacun précédé d'un gros point, dans la partie droite :

```
\cvlistdoubleitem{Château de Moulinsart}{Bianca Castafiore}
```

10.1.5 Modifier la taille de la partie gauche

En fonction des informations figurant dans la partie gauche, il peut être nécessaire d'en augmenter la taille. Au contraire, il est parfois utile de la réduire. Ceci se fait au moyen des commandes suivantes, utilisées dans le préambule du document :

```
\setlength{\hintscolumnwidth}{45mm}
\AtBeginDocument{\recomputelengths}
```

La longueur 45mm est à adapter pour obtenir une partie gauche de la taille voulue.

10.2 La classe **curve**

La classe `curve` permet de réaliser facilement un CV d'apparence classique.

La figure 10.4 montre le code source d'un CV réalisé avec la classe `curve`, la figure 10.5 montre le document généré.

La classe `curve` s'utilise avec les options de classe habituelles, auxquelles on ajoute l'option de classe `french`. En effet, lorsqu'une rubrique est interrompue par une fin de page, le titre de cette rubrique est repris en haut de la page suivante en lui ajoutant « (suite) ». Pour que ceci soit affiché dans la bonne langue, il est donc nécessaire d'indiquer à la classe `curve` dans quelle langue le CV est rédigé (en plus de l'utilisation de l'extension `babel`).

Remarquez que l'extension `geometry` (voir le paragraphe 8.3) a été utilisée afin de réduire les marges de tête et de pied du document pour que le CV tienne sur une seule page.

10.2.1 Les informations personnelles

Les informations personnelles sont placées dans l'en-tête et le titre du CV.

La commande `\lefthead`, utilisée dans le préambule du document, permet de définir la partie gauche de l'en-tête du CV. Elle peut contenir plusieurs lignes séparées par `\\` :

```
\lefthead{Place de la mairie \\
          99999 MOULINSART \\
          téléphone : 01 02 03 04 05 \\
          courriel : lampion@mondass.fr}
```

Sur le même principe, la commande `\righthead`, utilisée dans le préambule du document, permet de définir la partie droite de l'en-tête du CV :

```
\righthead{Marié \\ beaucoup d'enfants}
```

La commande `\photo`, utilisée dans le préambule du document, permet d'indiquer le nom du fichier au format PDF, PNG ou JPEG contenant la photo :

```
1 \documentclass[a4paper,11pt,french]{curve}
2
3 \usepackage[french]{babel}
4 \usepackage[T1]{fontenc}
5 \usepackage[utf8]{inputenc}
6 \usepackage{lmodern}
7 \usepackage{microtype}
8 \usepackage[top=2cm,bottom=2cm]{geometry}
9
10 \leftheader{Place de la mairie \\  
11             99999 MOULINSART \\  
12             téléphone : 01 02 03 04 05 \\  
13             courriel : lampion@mondass.fr}
14
15 \rightheader{Marié \\  
16             beaucoup d'enfants}
17
18 \photo{seraphin-lampion.jpg}
19 \photoscale{0.12}
20
21 \title{Séraphin Lampion}
22 \subtitle{Représentant en assurances}
23
24 %\rubricalignment{1}
25
26 \prefix{}
27
28 \begin{document}
29
30 \makeheaders[t]
31 \maketitle
32
33 \makerubric{curve-experience}
34 \makerubric{curve-references}
35 \makerubric{curve-methodologies}
36 \makerubric{curve-formation}
37 \makerubric{curve-langues}
38 \makerubric{curve-extra}
39
40 \end{document}
```

FIGURE 10.4 – Exemple de CV réalisé avec la classe *curve* (code source).

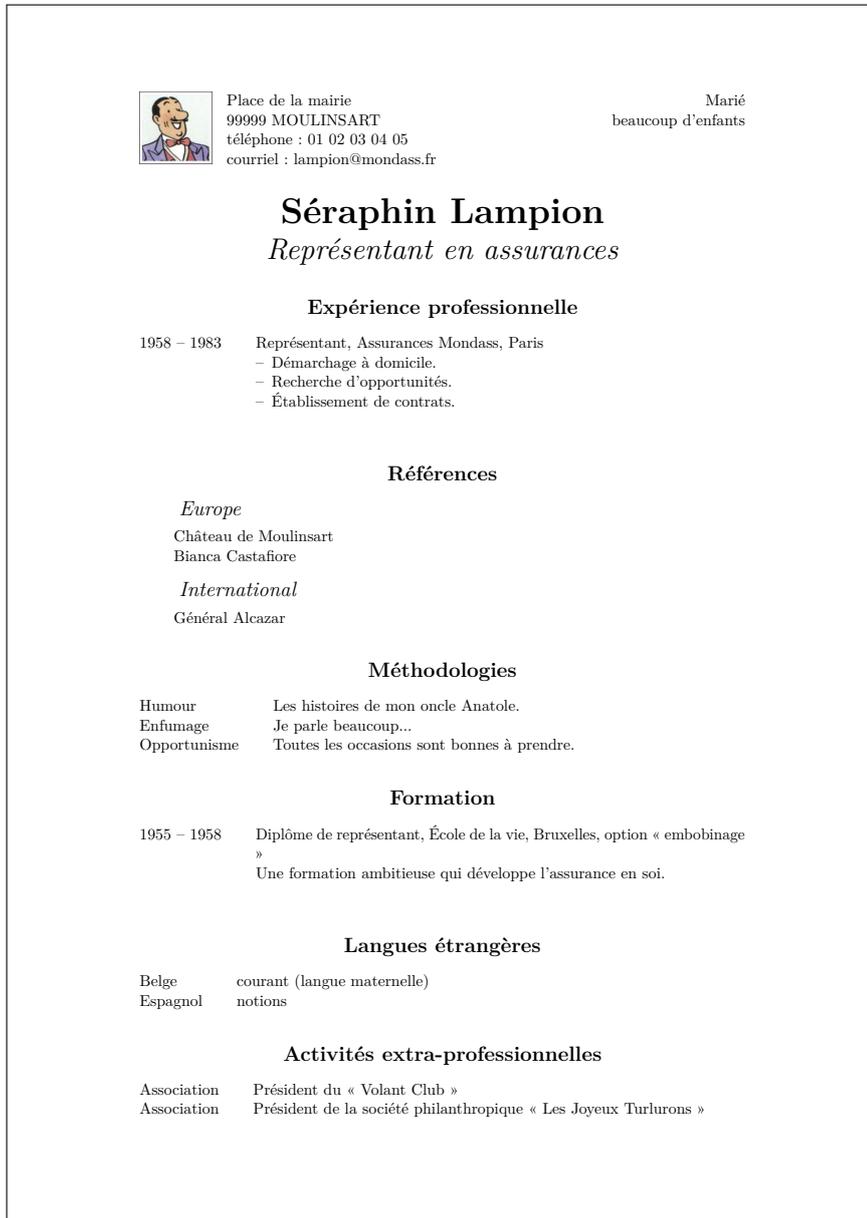


FIGURE 10.5 – Exemple de CV réalisé avec la classe curve (document généré).

```
\photo{seraphin-lampion.jpg}
```

La photo est par défaut placée à gauche, entre la marge et l'en-tête. L'argument optionnel de la commande `\photo` permet de la placer au centre s'il vaut `c` ou à droite, entre l'en-tête et la marge, s'il vaut `r` :

```
\photo[c]{seraphin-lampion.jpg}
```

La commande `\photoscale`, utilisée dans le préambule du document, permet d'indiquer la fraction de la largeur du CV alloué à la photo, entre 0 et 1 (elle est par défaut de 0,1 soit 10 %) :

```
\photoscale{0.12}
```

La commande `\title`, utilisée dans le préambule du document, permet d'indiquer le titre du CV (qu'on utilise généralement pour indiquer le prénom et le nom) :

```
\title{Séraphin Lampion}
```

La commande `\subtitle`, utilisée dans le préambule du document, permet d'indiquer le sous-titre du CV (qu'on utilise généralement pour indiquer le poste actuel ou le poste recherché) :

```
\subtitle{Représentant en assurances}
```

La commande `\makeheaders`, utilisée dans le corps du document, permet d'imprimer l'en-tête. Ses deux parties sont par défaut centrées verticalement. L'argument optionnel de la commande `\makeheaders` permet de aligner en haut s'il vaut `t` et en bas s'il vaut `b` :

```
\makeheaders[t]
```

La commande `\maketitle`, utilisée dans le corps du document, permet d'imprimer le titre et le sous-titre :

```
\maketitle
```

10.2.2 Les rubriques

Les rubriques du CV doivent être contenues chacune dans un fichier propre et appelées depuis le fichier principal au moyen de la commande `\makerubric`, dont l'argument est le nom du fichier contenant la rubrique, sans son extension `.tex` :

```
\makerubric{curve-experience}
\makerubric{curve-references}
\makerubric{curve-methodologies}
\makerubric{curve-formation}
\makerubric{curve-langues}
\makerubric{curve-extra}
```

Les figures 10.6 à 10.11 montrent les codes sources des rubriques.

```
1 \begin{rubric}{Expérience professionnelle}
2 \entry*[1958 -- 1983] Représentant, Assurances Mondass, Paris
3 \begin{itemize}
4 \item Démarchage à domicile.
5 \item Recherche d'opportunités.
6 \item Établissement de contrats.
7 \end{itemize}
8 \end{rubric}
```

FIGURE 10.6 – Exemple de CV réalisé avec la classe *curve* (code source du fichier *curve-experience.tex*).

```
1 \begin{rubric}{Références}
2 \subrubric{Europe}
3 \entry* Château de Moulinsart
4 \entry* Bianca Castafiore
5 \subrubric{International}
6 \entry* Général Alcazar
7 \end{rubric}
```

FIGURE 10.7 – Exemple de CV réalisé avec la classe *curve* (code source du fichier *curve-references.tex*).

Chaque rubrique est contenue dans un environnement `rubric`, dont l'argument indique le titre de la rubrique.

```
1 \begin{rubric}{Méthodologies}
2 \entry*[Humour] Les histoires de mon oncle Anatole.
3 \entry*[Enfumage] Je parle beaucoup...
4 \entry*[Opportunisme] Toutes les occasions sont bonnes à
5                       prendre.
6 \end{rubric}
```

FIGURE 10.8 – Exemple de CV réalisé avec la classe *curve* (code source du fichier *curve-methodologies.tex*).

```
1 \begin{rubric}{Formation}
2
3 \entry*[1955 -- 1958] Diplôme de représentant, École de la
4                       vie, Bruxelles, option « embobinage »
5 \entry*
6                       Une formation ambitieuse qui développe
7                       l'assurance en soi.
8 \end{rubric}
```

FIGURE 10.9 – Exemple de CV réalisé avec la classe *curve* (code source du fichier *curve-formation.tex*).

```
1 \begin{rubric}{Langues étrangères}
2 \entry*[Belge] courant (langue maternelle)
3 \entry*[Espagnol] notions
4 \end{rubric}
```

FIGURE 10.10 – Exemple de CV réalisé avec la classe *curve* (code source du fichier *curve-langues.tex*).

```
1 \begin{rubric}{Activités extra-professionnelles}
2 \entry*[Association] Président du « Volant Club »
3 \entry*[Association] Président de la société philanthropique
4                       « Les Joyeux Turlurons »
5 \end{rubric}
```

FIGURE 10.11 – Exemple de CV réalisé avec la classe *curve* (code source du fichier *curve-extra.tex*).

Les titres des rubriques sont centrés par défaut. Ceci peut être modifié au moyen de la commande `\rubricalignment`, utilisée dans le préambule, qui permet de les aligner à gauche si son argument vaut `l` ou à droite si son argument vaut `r` :

```
\rubricalignment{l}
```

Le contenu de chaque rubrique est constitué d'une ou plusieurs commandes `\entry*` (avec une étoile). Son argument optionnel, s'il est présent, est placé sur la gauche. Ce qui suit la commande `\entry*` est placé sur la droite.

Il est important de ne pas laisser de lignes blanches à l'intérieur de l'environnement `rubric`. En effet, en raison de la façon dont celui-ci est conçu, cela se traduit par des lignes blanches dans le document généré.

10.3 L'extension *currvita*

L'extension *currvita* permet de réaliser rapidement un CV plutôt dépouillé.

La figure 10.12 montre le code source d'un CV réalisé avec l'extension *currvita*, la figure 10.13 montre le document généré.

L'extension *currvita* s'utilise en complément d'une classe non spécialisée dans la rédaction de CV. Celle-ci peut être `article` ou `scrartcl` pour créer un CV autonome mais on peut également inclure un CV dans un document plus volumineux (par exemple un dossier de candidature créé avec une classe telle que `report` ou `scrreprt`).

L'extension *currvita* doit donc être chargée dans le préambule du document. Il est intéressant d'utiliser l'option `NoDate`, qui évite l'impression de la date de compilation à la suite du CV :

```
\usepackage[NoDate]{currvita}
```

Dans un CV autonome, il est inutile d'utiliser l'extension `hyperref` car l'extension *currvita* n'en tire pas parti.

L'environnement `cv` délimite le CV. Il s'utilise avec un argument qui contient le titre à placer en tête du CV (qu'on peut laisser vide si l'on n'en souhaite pas) :

```
\begin{cv}{Séraphin Lampion --- Représentant en assurances}
[... ]
\end{cv}
```

```
1 \documentclass[a4paper,11pt]{article}
2
3 \usepackage[french]{babel}
4 \usepackage[T1]{fontenc}
5 \usepackage[utf8]{inputenc}
6 \usepackage{lmodern}
7 \usepackage{microtype}
8 \usepackage[NoDate]{currvita}
9
10 \begin{document}
11
12 %\setlength{\cvlabelwidth}{25mm}
13
14 \begin{cv}{S raphin Lampion --- Repr sentant en assurances}
15
16 \begin{cvlist}{Informations personnelles}
17
18 \item S raphin Lampion \\
19     Place de la mairie \\
20     99999 MOULINSART
21
22 \item t l phone : 01 02 03 04 05 \\
23     courriel : lampion@mondass.fr
24
25 \end{cvlist}
26
27 \begin{cvlist}{Exp rience professionnelle}
28
29 \item[1958 -- 1983] Repr sentant, Assurances Mondass, Paris
30     \begin{itemize}
31         \item D marchage   domicile.
32         \item Recherche d'opportunit s.
33         \item  tablissement de contrats.
34     \end{itemize}
35
36 \end{cvlist}
37
```

FIGURE 10.12 – Exemple de CV r alis  avec l'extension *currvita* (code source).

```
38 \begin{cvlist}{Références}
39
40 \item Château de Moulinsart
41
42 \item Bianca Castafiore
43
44 \item Général Alcazar
45
46 \end{cvlist}
47
48 \begin{cvlist}{Methodologies}
49
50 \item[Humour] Les histoires de mon oncle Anatole.
51
52 \item[Enfumage] Je parle beaucoup...
53
54 \item[Opportunisme] Toutes les occasions sont bonnes à
55     prendre.
56
57 \end{cvlist}
58
59 \begin{cvlist}{Formation}
60
61 \item[1955 -- 1958] Diplôme de représentant, École de la vie,
62     Bruxelles, option « embobinage » \\
63     Une formation ambitieuse qui développe
64     l'assurance en soi.
65
66 \end{cvlist}
67
68 \begin{cvlist}{Langues étrangères}
69
70 \item[Belge] courant (langue maternelle)
71
72 \item[Espagnol] notions
73
74 \end{cvlist}
75
```

FIGURE 10.12 – Exemple de CV réalisé avec l'extension *currvita* (code source).

```
76 \begin{cvlist}{Activités extra-professionnelles}
77
78 \item[Association] Président du « Volant Club »
79
80 \item[Association] Président de la société philanthropique
81     « Les Joyeux Turlurons »
82
83 \end{cvlist}
84
85 \end{cv}
86
87 \end{document}
```

FIGURE 10.12 – Exemple de CV réalisé avec l’extension *curvita* (code source).

À l’intérieur de l’environnement `cv`, les rubriques du CV sont réalisées au moyen d’autant d’environnements `cvlist`, dont l’argument indique le titre de la rubrique :

```
\begin{cvlist}{Expérience professionnelle}
\item[1958 -- 1983] Représentant, Assurances Mondass, Paris
    \begin{itemize}
    \item Démarchage à domicile.
    \item Recherche d’opportunités.
    \item Établissement de contrats.
    \end{itemize}
\end{cvlist}
```

Le contenu de chaque rubrique est constitué d’une ou plusieurs commandes `\item`. Son argument optionnel, s’il est présent, est placé sur la gauche. Ce qui suit la commande `\item` est placé sur la droite.

Enfin, la commande suivante, placée dans le corps du document (pas dans le préambule) et avant d’utiliser l’environnement `cv`, permet de modifier la largeur de la partie gauche :

```
\setlength{\cvlabelwidth}{25mm}
```

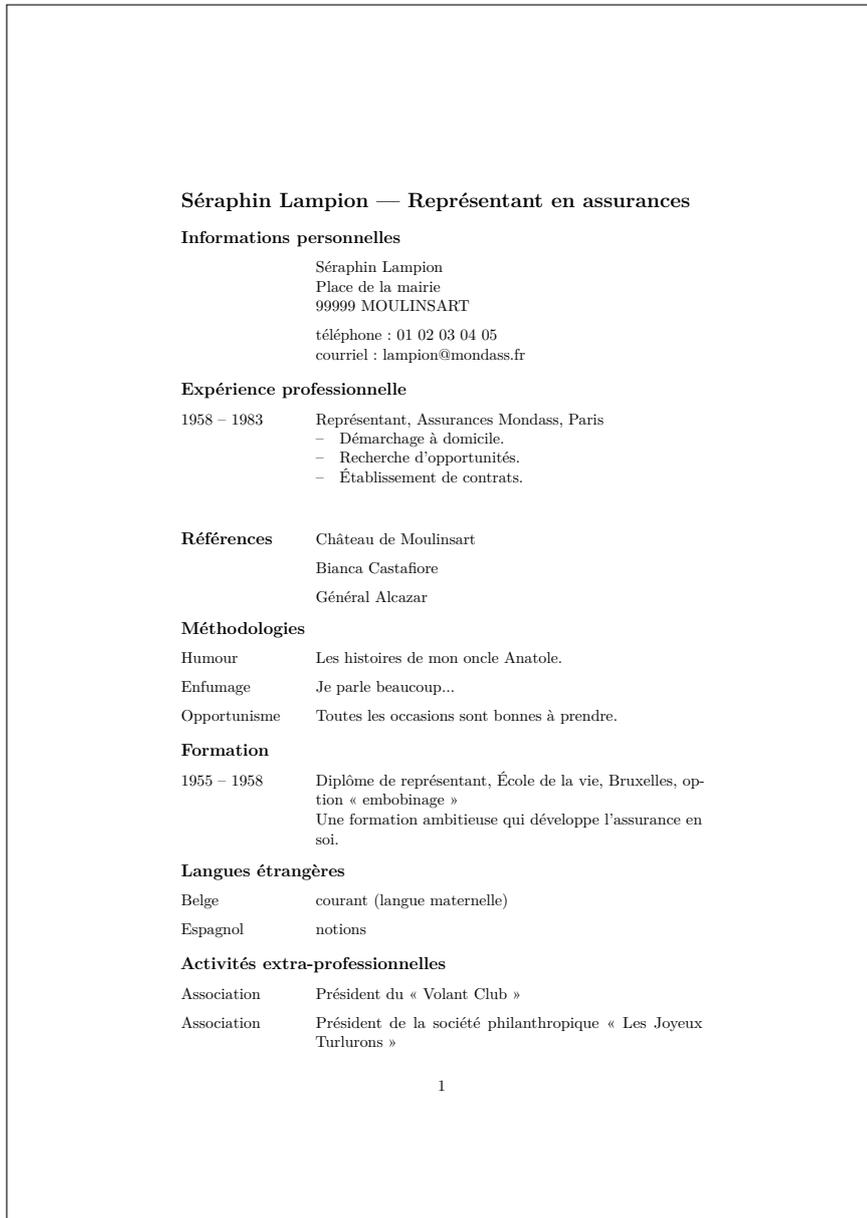


FIGURE 10.13 – Exemple de CV réalisé avec l'extension currvita (document généré).

Chapitre 11

Faire une présentation : les classes `slides` et `beamer`

En bureautique, beaucoup utilisent deux logiciels différents, un pour le traitement de texte et un autre pour les présentations. C'est regrettable parce que cela complique la réalisation conjointe d'un document écrit et de la présentation associée, cette dernière devant généralement emprunter certains éléments figurant dans le premier.

Avec \LaTeX , le problème ne se pose pas car il permet également de réaliser des présentations. Pour cela, on peut utiliser :

- la classe `slides`, qui est fournie en standard avec \LaTeX et qui est destinée à réaliser des transparents qu'on imprime et qu'on projette avec un rétroprojecteur ;
- la classe `beamer`, qui n'est pas standard mais qui figure dans la plupart des installations \LaTeX et qui est destinée à réaliser des présentations au format PDF pouvant être projetées avec un vidéoprojecteur (grâce à la fonction d'affichage en plein écran des logiciels de visualisations de fichiers PDF).

Signalons également l'existence des classes `seminar`¹ et `prosper`², qui n'ont aujourd'hui d'autre intérêt qu'historique. Sauf si vous avez besoin de modifier des présentations utilisant l'une de ces classes, il est préférable d'utiliser `beamer` pour réaliser de nouvelles présentations.

11.1 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 :

1. <http://www.tug.org/applications/Seminar/>
2. <http://prosper.sourceforge.net/prosper.html>

```
\documentclass[a4paper]{slides}

\usepackage[french]{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage{lmodern}
\usepackage{microtype}

\begin{document}

\begin{slide}

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

\end{slide}

\end{document}
```

Ce document produit un unique transparent (puisque'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 11.1.

Dans l'environnement `slide` :

- le texte est écrit en gros caractères ;
- il est écrit dans une police linéale ;
- il est centré verticalement ;
- chaque page est numérotée dans le coin inférieur droit.

Vous pouvez utiliser avec la classe `slides` la plupart des commandes et des environnements de \LaTeX qui ont un sens dans un transparent (par exemple, des commandes comme les commandes de hiérarchisation ne sont pas utiles dans les transparents et n'existent donc pas dans la classe `slides`). Il n'existe pas de liste des commandes et environnements de \LaTeX non disponibles dans la classe `slides` donc, dans le doute, essayez !

Pour plus de renseignements sur la classe `slides`, vous pouvez consulter [5, pages 80–84].

11.2 La classe `beamer`

De nos jours le vidéoprojecteur a quasiment partout remplacé le rétroprojecteur et les ternes transparents d'antan ont laissé la place à des

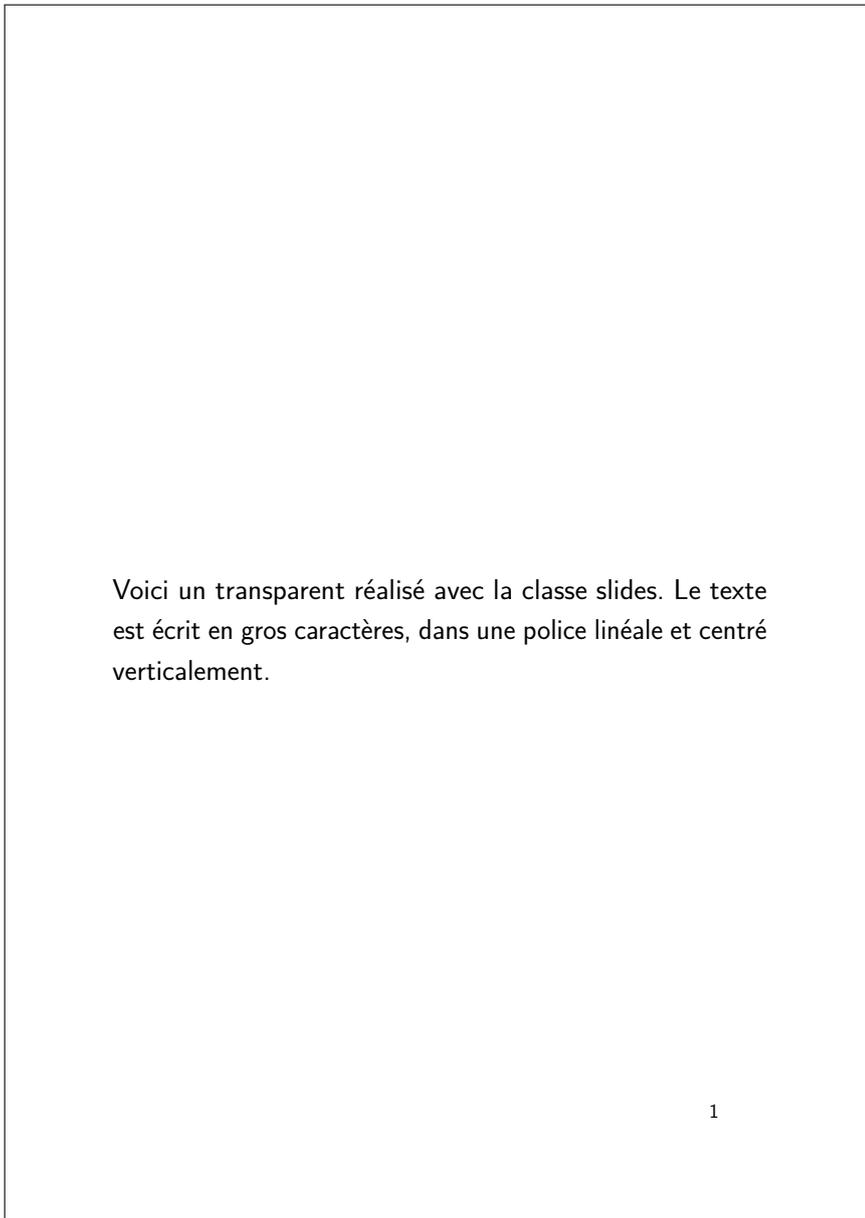


FIGURE 11.1 – *Exemple de transparent réalisé avec la classe slides.*

présentations multicolores. À cet effet, la classe `beamer` permet de réaliser des présentations au format PDF bénéficiant de toute la puissance de \LaTeX .

Utiliser le format PDF pour les présentations offre de nombreux avantages :

- il s’agit d’un format universel, visualisable aussi bien sur un ordinateur que sur un téléphone mobile évolué ;
- il s’agit d’un format ouvert, qui n’est pas lié à un logiciel spécifique ;
- la possibilité de visualiser les documents PDF directement dans un navigateur facilite leur diffusion sur la toile ;
- les documents PDF sont généralement de taille raisonnable.

L’aspect des présentations peut être modifié au moyen de thèmes, qui seront abordés au paragraphe 11.2.4. Ces thèmes peuvent également être personnalisés (modification des couleurs, ajout d’un logo), comme nous le verrons au paragraphe 11.2.5.

11.2.1 Structure d’une présentation

La structure d’une présentation réalisée avec la classe `beamer` est indiquée dans la figure 11.2.

Il est généralement inutile d’indiquer des options de classe avec `beamer`. En effet, la taille du papier n’a ici pas de sens et la fonte par défaut est suffisamment grande pour une projection sur écran.

Il est également inutile de charger l’extension `hyperref` car `beamer` le fait automatiquement.

Le corps du document contient une suite d’environnements `frame`, chacun contenant une page de la présentation.

11.2.2 La page de titre

La première page d’une présentation est la page de titre.

Elle utilise différentes indications figurant dans le préambule :

```
\title{Mon oncle Anatole}
\subtitle{Sa vie, son œuvre}
\author{Séraphin Lampion}
\institute{Assurances Mondass}
\date{23 octobre 2009}
```

- la commande `\title` indique le titre de la présentation ;
- la commande `\subtitle` indique le sous-titre de la présentation ;
- la commande `\author` indique l’auteur de la présentation ;
- la commande `\institute` indique l’employeur de l’auteur de la présentation ;
- la commande `\date` indique la date de la présentation.

```
\documentclass{beamer}

\usepackage[french]{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage{lmodern}
\usepackage{microtype}

% reste du préambule

\begin{document}

\begin{frame}

[...]

\end{frame}

\begin{frame}

[...]

\end{frame}

[...]

\end{document}
```

FIGURE 11.2 – Structure d'une présentation réalisée avec la classe beamer.

On peut utiliser `\\` dans les arguments de ces commandes pour forcer un passage à la ligne si nécessaire.

En fonction du thème utilisé, certains de ces éléments peuvent être repris dans les marges des pages intérieures de la présentation. S'ils sont trop longs, on peut indiquer une version courte en option de chacune des commandes. Celle-ci sera utilisée dans les marges alors que l'argument le sera dans la page de titre :

```
\title[ENSTA ParisTech]{École nationale supérieure  
de techniques avancées \\ (ENSTA ParisTech)}
```

On imprime ensuite la page de titre (dans le premier environnement `frame` de la présentation) au moyen de la commande `\titlepage` :

```
\begin{frame}  
  
\titlepage  
  
\end{frame}
```

Ceci donne au final la page de titre représentée dans la figure 11.3.



FIGURE 11.3 – Exemple de page de titre réalisée avec la classe *beamer*.

11.2.3 Les pages intérieures

Les pages suivant la page de titre sont elles-aussi contenues dans un environnement `frame` :

```
\begin{frame}{Ceci est le titre de la page}

La suite est le corps de la page.

\end{frame}
```

L'argument de l'environnement `frame` indique le titre de la page. On peut également ne pas lui mettre d'argument et utiliser à la place la commande `\frametitle` :

```
\begin{frame}

\frametitle{Ceci est le titre de la page}

La suite est le corps de la page.

\end{frame}
```

Le reste du contenu de l'environnement `frame` correspond au corps de la page et l'on peut y utiliser quasiment toutes les commandes et environnements vus dans les chapitres précédents.

Par ailleurs, `beamer` a étendu la syntaxe de \LaTeX pour permettre d'afficher certains éléments d'une page au fur et à mesure du discours, comme cela est habituel dans les présentations.

En pratique, chaque page (logique) du document est divisée en plusieurs pages (physiques) dans le document PDF généré en fonction du nombre d'éléments à faire apparaître. La succession des pages du document PDF donnera l'illusion de l'apparition des éléments sur une même page.

Considérons l'exemple suivant :

```
\begin{itemize}

\item<1-> premier élément

\item<2-> deuxième élément

\item<3-> troisième élément

\end{itemize}
```

Chaque commande `\item` est suivie d'une spécification entre chevrons. Celle-ci indique sur quelles pages du document PDF faire figurer l'élément en question.

On peut indiquer :

- une page précise : `<2>` ;
- une liste de pages : `<2,3,4>` ;
- un intervalle de pages : `<2-4>`, `<-3>` (de la première page jusqu'à la page 3), `<2->` (à partir de la page 2) ;
- et toute combinaison des spécifications précédentes.

Ainsi, dans notre exemple, le premier élément figurera sur la première page et les suivantes (la spécification `<1->` est d'ailleurs superflue puisque tout élément sans spécification figurera sur toutes les pages), le deuxième élément figurera sur la deuxième page et les suivantes et le troisième élément figurera sur la troisième page et les suivantes. La page logique (c'est-à-dire l'environnement `frame`) contenant la liste indiquée dans l'exemple générera donc trois pages dans le document PDF. La spécification `<3->` du troisième élément aurait d'ailleurs pu être simplifiée en `<3>` mais indiquer `<3->` est cohérent avec les spécifications des autres éléments et simplifie l'ajout éventuel d'un quatrième élément (puisque la spécification du troisième n'aura pas à être modifiée).

Le fait d'afficher certains éléments d'une page au fur et à mesure est surtout utilisé avec des listes, comme c'est le cas dans l'exemple précédent, mais il est possible de produire le même effet sur n'importe quel texte au moyen de la commande `\uncover`, qui s'utilise avec le même type de spécification :

```
\uncover<1->{La philanthropie de l'ouvrier charpentier.}
\uncover<2->{La Chine se dresse devant les Nippons.}
\uncover<3->{Les matières fiscales le détendent.}
```

Voire, au sein même d'un paragraphe :

```
\uncover<1->{La} \uncover<2->{muse} \uncover<3->{nous}
\uncover<4->{habite.}
```

Pour des portions de document plus importantes, on peut utiliser l'environnement `uncoverenv` :

```

\begin{uncoverenv}<1->
Paragraphe d'introduction :

\begin{itemize}
[...]
\end{itemize}
\end{uncoverenv}

\begin{uncoverenv}<2->
Paragraphe d'introduction :

\begin{itemize}
[...]
\end{itemize}
\end{uncoverenv}

```

Le texte qui n'est pas affiché dans une page PDF donnée est normalement invisible. Pour le faire apparaître en filigrane, on peut utiliser la commande `\setbeamercovered` dans le préambule :

```
\setbeamercovered{transparent}
```

Il peut être nécessaire d'adapter le niveau de transparence entre 0 et 100 (0 signifiant invisible et 100 opaque, le niveau par défaut est de 15) en fonction du vidéoprojecteur utilisé :

```
\setbeamercovered{transparent=20}
```

11.2.4 Modification de l'aspect des présentations : les thèmes

L'aspect des présentations peut facilement être modifié au moyen de *thèmes*. Pour cela, on utilise dans le préambule la commande `\usetheme` en indiquant le nom du thème souhaité en argument :

```
\usetheme{Montpellier}
```

Les thèmes utilisables sont indiqués dans le tableau [11.1](#).

Le thème `default` est celui qui est utilisé en l'absence de la commande `\usetheme` dans le préambule (mais on peut aussi le spécifier explicitement si on le souhaite). Les autres thèmes portent des noms de villes.

Les figures [11.4](#) à [11.30](#) montrent, pour chaque thème, à quoi ressemblent une page de titre à gauche et une page intérieure à droite.

default	Boadilla	Goettingen	Malmoe	Singapore
AnnArbor	CambridgeUS	Hannover	Marburg	Szeged
Antibes	Copenhagen	Ilmenau	Montpellier	Warsaw
Bergen	Darmstadt	JuanLesPins	PaloAlto	
Berkeley	Dresden	Luebeck	Pittsburgh	
Berlin	Frankfurt	Madrid	Rochester	

TABLE 11.1 – Thèmes utilisables avec *beamer*.

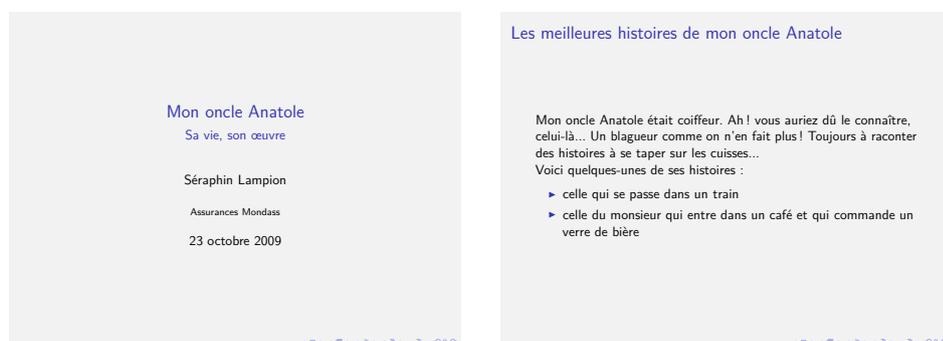


FIGURE 11.4 – Exemple de présentation réalisée avec la classe *beamer* et le thème *default*.



FIGURE 11.5 – Exemple de présentation réalisée avec la classe *beamer* et le thème *AnnArbor*.



FIGURE 11.6 – Exemple de présentation réalisée avec la classe *beamer* et le thème *Antibes*.



FIGURE 11.7 – Exemple de présentation réalisée avec la classe *beamer* et le thème *Bergen*.



FIGURE 11.8 – Exemple de présentation réalisée avec la classe *beamer* et le thème *Berkeley*.



FIGURE 11.9 – Exemple de présentation réalisée avec la classe beamer et le thème Berlin.



FIGURE 11.10 – Exemple de présentation réalisée avec la classe beamer et le thème Boadilla.



FIGURE 11.11 – Exemple de présentation réalisée avec la classe beamer et le thème CambridgeUS.



FIGURE 11.12 – Exemple de présentation réalisée avec la classe beamer et le thème Copenhagen.



FIGURE 11.13 – Exemple de présentation réalisée avec la classe beamer et le thème Darmstadt.



FIGURE 11.14 – Exemple de présentation réalisée avec la classe beamer et le thème Dresden.



FIGURE 11.15 – Exemple de présentation réalisée avec la classe *beamer* et le thème *Frankfurt*.



FIGURE 11.16 – Exemple de présentation réalisée avec la classe *beamer* et le thème *Goettingen*.

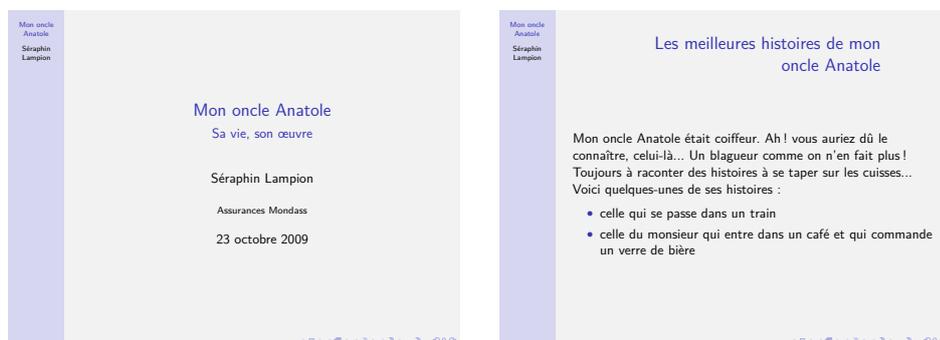


FIGURE 11.17 – Exemple de présentation réalisée avec la classe *beamer* et le thème *Hannover*.



FIGURE 11.18 – Exemple de présentation réalisée avec la classe *beamer* et le thème *Ilmenau*.



FIGURE 11.19 – Exemple de présentation réalisée avec la classe *beamer* et le thème *JuanLesPins*.



FIGURE 11.20 – Exemple de présentation réalisée avec la classe *beamer* et le thème *Luebeck*.



FIGURE 11.21 – Exemple de présentation réalisée avec la classe beamer et le thème Madrid.



FIGURE 11.22 – Exemple de présentation réalisée avec la classe beamer et le thème Malmo.



FIGURE 11.23 – Exemple de présentation réalisée avec la classe beamer et le thème Marburg.



FIGURE 11.24 – Exemple de présentation réalisée avec la classe *beamer* et le thème *Montpellier*.



FIGURE 11.25 – Exemple de présentation réalisée avec la classe *beamer* et le thème *PaloAlto*.

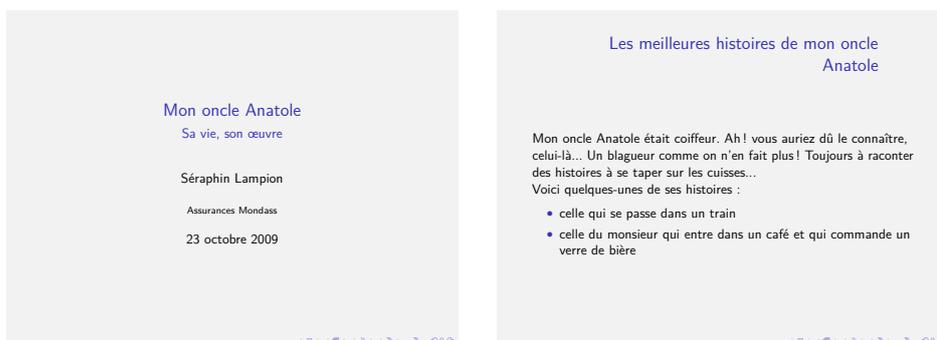


FIGURE 11.26 – Exemple de présentation réalisée avec la classe *beamer* et le thème *Pittsburgh*.



FIGURE 11.27 – Exemple de présentation réalisée avec la classe beamer et le thème Rochester.



FIGURE 11.28 – Exemple de présentation réalisée avec la classe beamer et le thème Singapore.



FIGURE 11.29 – Exemple de présentation réalisée avec la classe beamer et le thème Szeged.

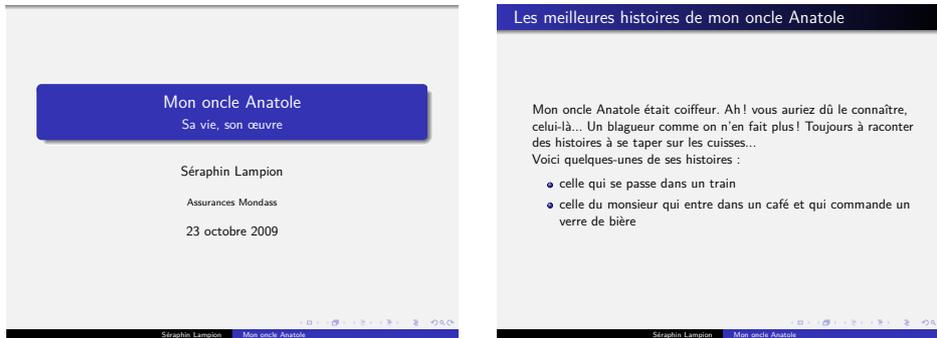


FIGURE 11.30 – Exemple de présentation réalisée avec la classe *beamer* et le thème *Warsaw*.

11.2.5 Personnalisation de l’aspect des présentations

Les thèmes offrent une grande diversité de mise en pages pour les présentations. Néanmoins, la dominante bleue de la plupart d’entre eux peut se révéler gênante. Il est heureusement possible de modifier les couleurs utilisées par son thème favori. Nous verrons également comment ajouter un logo.

11.2.5.1 Modification des couleurs

La commande `\usecolortheme`, utilisée dans le préambule, permet de modifier la couleur dominante utilisée par le thème courant :

```
\usecolortheme[RGB={64,141,183}]{structure}
```

Il est possible de spécifier la couleur de plusieurs manières :

- `rgb={r,g,b}` qui indique les valeurs de rouge, de vert et de bleu par des nombres décimaux entre 0 et 1, par exemple `rgb={0.3,0.5,0.7}` ;
- `RGB={r,g,b}` qui indique les valeurs de rouge, de vert et de bleu par des nombres entiers entre 0 et 255, par exemple `RGB={0,171,199}` ;
- `cmYk={c,m,y,k}` qui indique les valeurs de cyan, de magenta, de jaune et de noir par des nombres décimaux entre 0 et 1, par exemple `cmYk={0.76,0,0.15,0.11}` ;
- `cmY={c,m,y}` est identique à `cmYk` sauf que la composante noire n’est pas indiquée ;
- `hsb={h,s,b}` qui indique la teinte, la saturation et la valeur par des nombres décimaux entre 0 et 1, par exemple `hsb={0.3,0.5,0.7}` ;
- `named=nom` qui indique la couleur par son nom.

Pour `named`, il est possible d’utiliser les couleurs prédéfinies indiquées dans le tableau 11.2.

On peut aussi définir ses propres noms de couleurs au moyen de la commande `\definecolor`, utilisée dans le préambule :

black	cyan	green	orange	violet
blue	darkgray	lightgray	purple	white
brown	gray	magenta	red	yellow

TABLE 11.2 – Couleurs prédéfinies utilisables avec *beamer*.

```
\definecolor{nom}{type}{valeur}
```

qui définit une couleur *nom* spécifiée par *type* (qui peut être `rgb`, `RGB`, `cmk`, `cm` ou `hsb`, comme vu précédemment) et par *valeur* (qui doit être compatible avec *type*). Par exemple :

```
\definecolor{macouleur}{RGB}{0,171,199}  
\usecolortheme[named=macouleur]{structure}
```

11.2.5.2 Ajout d'un logo

La commande `\logo`, utilisée dans le préambule, permet d'insérer un logo dans la plupart des thèmes (tous sauf `Goettingen`, `Hannover` et `Marburg`). On peut utiliser pour cela la commande `\includegraphics` (voir le paragraphe 8.6.1) :

```
\logo{\includegraphics[width=1cm]{logo.pdf}}
```

Le logo contenu dans le fichier `logo.pdf` apparaîtra alors sur chaque page à l'endroit prévu par le thème. Sa taille (ici `1cm`) est à adapter au besoin.

Il peut également être nécessaire d'adapter la couleur de fond du logo pour qu'il s'en détache correctement. Pour cela, on utilise la commande `\setbeamercolor` dans le préambule :

```
\setbeamercolor{logo}{bg=white}
```

11.2.6 Pour aller plus loin

Nous n'avons couvert que quelques-uns des nombreux aspects de `beamer`. Pour plus de renseignements à son sujet, vous pouvez consulter [8, pages 752–796].

Les sites suivants sont également intéressants :

```
http://bitbucket.org/rivanvx/beamer/wiki/Home (site officiel)  
http://mirror.ctan.org/macros/latex/contrib/beamer/doc/  
beameruserguide.pdf (documentation officielle)
```

<http://fr.wikipedia.org/wiki/Beamer>
<http://mcclnews.free.fr/latex/introbeamer.php>
<http://www.math.umbc.edu/~rouben/beamer/>

Chapitre 12

Et maintenant ?

Si vous lisez ces lignes, c'est que vous avez probablement parcouru, voire lu attentivement, quelques uns des chapitres précédents. J'espère que, sans être nécessairement déjà convaincu de l'intérêt d'utiliser \LaTeX , vous avez en tout cas une bonne vision de ce qu'il permet de faire. Peut-être même l'utilisez-vous déjà pour réaliser vos documents.

12.1 Pour en apprendre davantage

Les possibilités offertes par \LaTeX et ses nombreuses extensions sont variées et ce manuel n'en décrit que quelques unes. Si vous souhaitez parfaire votre connaissance de \LaTeX , la lecture des ouvrages [5] et [10] de la bibliographie est un passage obligé. D'autres peuvent les compléter en fonction de vos besoins.

Plusieurs sites Web peuvent également se révéler intéressants.

Pour commencer, le site officiel de \LaTeX :

■ <http://www.latex-project.org/>

Le site de l'association GUTenberg (groupe francophone des utilisateurs de \TeX) :

■ <http://www.gutenberg.eu.org/>

Enfin, le site du LORIA¹ dédié à \LaTeX qui, même s'il n'est plus activement entretenu, contient toujours des références intéressantes :

■ <http://tex.loria.fr/>

1. LORIA : laboratoire lorrain de recherche en informatique et ses applications.

12.2 Obtenir de l'aide

Si vous vous posez une question, le premier réflexe est de consulter la foire aux questions (FAQ) :

■ <http://faq.tug.org/>

Si vous n'y trouvez pas de réponse satisfaisante, vous pouvez poser votre question dans les groupes de discussion USENET `fr.comp.text.tex` (en français) ou `comp.text.tex` (en anglais).

12.3 Conclusion

En guise de conclusion, il est utile de revenir sur certains points abordés dans ce manuel et trop souvent maltraités par les débutants en \LaTeX (voire, malheureusement, par des personnes l'utilisant depuis plus longtemps).

12.3.1 Il faut construire son document en pensant à sa structure

C'est la différence fondamentale entre \LaTeX et un traitement de texte WYSIWYG. Avec \LaTeX , on s'attache avant tout à la structure de son document (et les commandes et environnements disponibles incitent à le faire) alors qu'avec un traitement de texte WYSIWYG, on a plutôt tendance à se focaliser sur la présentation du document au détriment de son contenu.

Et, pour raisonner en termes de structure, il est indispensable de définir ses propres commandes et ses propres environnements.

Je vois régulièrement des documents utilisant directement de nombreuses commandes telles que celles des tableaux 4.6 et 4.8. C'est mal car ces commandes ne décrivent en rien la structure du document. La bonne façon de faire est, comme indiqué à la fin du paragraphe 4.8, de définir dans le préambule ses propres commandes, en rapport avec la nature de leurs arguments et la structure du document. Il est généralement difficile de tout prévoir à l'avance aussi ces commandes doivent-elle être définies au fur et à mesure de la rédaction du document, lorsqu'on prend conscience de leur nécessité.

L'utilisation dans le document de certaines commandes de mise en pages abordées au paragraphe 4.10 est également à proscrire, en particulier les commandes d'espacement vertical, qui sont fréquemment utilisées par les débutants pour détacher un paragraphe du texte qui l'entoure. Pour cela, il est préférable de définir un environnement dont la définition prendra en charge cet espacement.

Toujours au sujet des commandes de mise en pages, il est également fréquent de voir utiliser `\` ou `\newline` soit en fin de paragraphe, c'est-à-dire avec une ligne blanche à la suite, soit en lieu et place de cette ligne

blanche. Rappelons que ces deux commandes ne doivent être utilisées que dans des cas précis (comme, par exemple, pour marquer la fin d'un vers dans l'environnement `verse` ou d'une ligne dans l'environnement `tabular`) et ne doivent en aucun cas être utilisées pour marquer la fin d'un paragraphe. Pour cela, il faut une ligne blanche, qui a également l'intérêt d'aérer le fichier source. Les différents paragraphes, séparés les uns des autres par une ligne blanche, sont donc immédiatement identifiables dans le fichier source.

12.3.2 Il ne faut pas bidouiller

Il est légitime de vouloir personnaliser la mise en pages de son document. Pour cela, l'utilisateur débutant est souvent livré à lui-même. Il existe certainement une extension permettant de faire ce qu'il souhaite mais comment la trouver ? Elle n'est pas nécessairement répertoriée dans [10], le moteur de recherche du CTAN (voir le paragraphe 8.1) peut ne pas trouver l'extension désirée (souvent parce que la recherche n'est pas spécifiée correctement) et la FAQ peut également n'être d'aucun secours. De nombreux utilisateurs se tournent alors vers leur moteur de recherche favori qui, avec un peu de chance, leur donnera une réponse correcte mais risque fort de les orienter, dans la majorité des cas, vers une page contenant un bout de code incompréhensible réalisant tout ou partie de la fonction désirée, que l'utilisateur va bidouiller sans trop comprendre ce qu'il fait pour obtenir enfin quelque chose correspondant à ses attentes.

Je noircis peut-être le tableau mais ce genre de situation est extrêmement fréquent. Dans ce cas, si une recherche comme indiqué précédemment ne donne aucun résultat, il est préférable de demander conseil à des utilisateurs plus expérimentés, soit dans son entourage soit en utilisant les groupes de discussion USENET `fr.comp.text.tex` ou `comp.text.tex`, plutôt que de bidouiller quelque chose qui sera probablement bancal.

12.3.3 Il faut soigner le fichier source

Même si le but de \LaTeX est de produire des documents mis en pages, on passe l'essentiel de son temps à travailler sur le fichier source. Il convient donc d'en soigner la présentation afin qu'il soit facilement lisible (pour soi-même et, au besoin, pour d'autres) et qu'il reflète la structure du document. Par exemple, même si le document produit par \LaTeX sera identique, mettre une ligne blanche avant et après chaque commande `\section` permet de mieux les distinguer dans le fichier source.

La lisibilité est une notion assez subjective mais on peut dégager quelques règles de bon sens :

- séparer les blocs de texte de natures différentes par des lignes blanches ;
- éviter les paragraphes comprenant uniquement une seule longue ligne, qui sont vite illisibles car les mots sont généralement coupés au bord

de la fenêtre, et leur préférer les paragraphes composés de plusieurs lignes (mais sans ligne blanche) d'une longueur inférieure à la largeur de la fenêtre de l'éditeur de texte (habituellement, 80 caractères) ;

- indenter certaines parties du texte pour en refléter la structure (par exemple les listes imbriquées) ;
- aligner verticalement les éléments similaires (par exemple dans des tableaux).

12.4 À vous de jouer !

Vous avez maintenant toutes les cartes en main pour pouvoir utiliser \LaTeX et découvrir la richesse de son univers. C'est à vous de jouer !

Annexe A

Quelques règles typographiques françaises et anglaises

De même que l'écriture d'un texte 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, afin de réaliser des documents de bonne qualité, il faut connaître et respecter soi-même toutes les autres.

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.

En ce qui concerne la typographie française, l'ouvrage de référence est [20]. La lecture de [22] et de [21] est également enrichissante. Pour la typographie anglaise (américaine, pour être très précis), l'ouvrage de référence est [24].

A.1 Quelques règles typographiques françaises

Contrairement à une croyance répandue, une lettre accentuée écrite en majuscule doit conserver son accent (ou son tréma ou sa cédille). À cet effet, si votre éditeur de texte n'en permet pas la saisie directe, il est toujours possible d'utiliser les commandes des tableaux 4.4 et 4.5.

La commande `\up`, fournie par l'extension `babel` utilisée avec l'option `french`, permet de composer son argument en caractères supérieurs. En typographie, on ne parle ni d'exposant ni d'indice mais de caractères respectivement *supérieurs* et *inférieurs*. La commande `\up` peut par exemple être utilisée pour écrire les abréviations indiquées dans le tableau A.1.

madame	M ^{me}	M\up{me}
mesdames	M ^{mes}	M\up{mes}
mademoiselle	M ^{lle}	M\up{lle}
mesdemoiselles	M ^{lles}	M\up{lles}
monsieur	M.	M.
messieurs	MM.	MM.

TABLE A.1 – *Madame, mademoiselle, monsieur.*

<http://www.orthotypographie.fr/volume-I/echelle-extrait.html#Exposant>
http://www.orthotypographie.fr/volume-II/madame-mythologie.html#Madame_mademoiselle_monsieur

Les abréviations indiquées dans le tableau A.2, quant à elles, sont suffisamment communes pour disposer de commandes propres, également fournies par l’extension `babel` utilisée avec l’option `french`. Utiliser ces commandes évite d’avoir dans ses documents des horreurs comme « 3^{ème} », qu’on voit malheureusement bien trop souvent...

premier	1 ^{er}	1\ier
première	1 ^{re}	1\iere
premiers	1 ^{ers}	1\iers
premières	1 ^{res}	1\ieres
deuxième	2 ^e	2\ieme
deuxièmes	2 ^{es}	2\iemes
etc.		

TABLE A.2 – *Premier, deuxième, etc..*

L’extension `textcomp` fournit la commande `\oldstylenums` qui permet d’obtenir les chiffres français (0123456789) :

La prise de la Bastille a eu lieu le `\oldstylenums{14}` juillet `\oldstylenums{1789}`.

La prise de la Bastille a eu lieu le 14 juillet 1789.

Ceci ne fonctionne que si la police de caractères utilisée contient les signes nécessaires, ce qui est malheureusement rare.

<http://www.orthotypographie.fr/volume-I/cabochon-citation.html#Chiffres>

Concernant la ponctuation :

- la virgule, le point et les points de suspension 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 d’une espace fine (ce que l’extension `babel` utilisée avec l’option `french` gère automatiquement) et suivis d’une espace ;
- les « guillemets français » ne sont pas présents sur les claviers, mais ils se saisissent facilement avec Vim en utilisant `^K <<` et `^K >>` et avec Emacs grâce aux commandes `C-x 8 <` et `C-x 8 >` ; à défaut, des commandes `\og («)` et `\fg (»)` sont fournies par l’extension `babel` utilisée avec l’option `french`.

Voici un petit exemple de poésie téléphonique à la ponctuation légèrement retouchée pour illustrer ceci :

```
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 ?
```

A.2 Quelques règles typographiques anglaises

Concernant la ponctuation :

- 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”.

Un autre exemple de poésie profondément pourpre (à la ponctuation modifiée) permet d’illustrer ces règles :

```
We all came out to Montreux, on the Lake Geneva shoreline,
to make records with a mobile: we didn't have much
time. Frank 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. Frank 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!
```

Annexe B

Différences entre \LaTeX 2.09 et \LaTeX 2 ϵ

Les différences entre \LaTeX 2.09 (version maintenant obsolète, décrite dans la première édition de ce manuel) et \LaTeX 2 ϵ (version actuelle, décrite dans ce manuel, apparue en 1994 mais n'ayant cessé d'évoluer depuis, en attendant \LaTeX 3) sont relativement minimales 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.

B.1 Les commandes \documentstyle , \documentclass et \usepackage

La commande \documentstyle de \LaTeX 2.09 a été remplacée dans \LaTeX 2 ϵ 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 .

B.2 Déclarations et commandes de changement de style

Les déclarations de changement de style \it , \sl , \sc \bf , \rm , \sf et \tt de \LaTeX 2.09 ont été remplacées respectivement dans \LaTeX 2 ϵ par les déclarations \itshape , \slshape , \scshape , \bfseries , \rmfamily , \sffamily et \ttfamily .

La déclaration \em de \LaTeX 2.09 a été remplacée dans \LaTeX 2 ϵ par la commande \emph .

Les commandes de changement de style `\textup`, `\textit`, `\textsl`, `\textsc`, `\textmd`, `\textbf`, `\textrm`, `\textsf` et `\texttt` sont nouvelles dans L^AT_EX 2_ε.

Ces 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'avec L^AT_EX 2.09, chaque commande annulait l'effet de la précédente :

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

italique **gras**

B.3 SliT_EX et la classe slides

Le logiciel SliT_EX n'existe plus, les transparents sont maintenant directement produits par L^AT_EX au moyen de la classe slides.

B.4 La commande \LaTeXe

La commande `\LaTeXe` permet d'imprimer le logo L^AT_EX 2_ε (avec le 2_ε). 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.

Bibliographie

Outre les ouvrages indiqués dans cette bibliographie, vous pouvez également consulter les sites suivants :

<http://www.gutenberg.eu.org/texetlatex/article229.html>
<http://www.latex-project.org/guides/books.html>

Ouvrages généraux sur \LaTeX

- [1] Denis BITOUZÉ et Jean Côme CHARPENTIER : *Introduction à \LaTeX* . Pearson, juillet 2010, ISBN 2-7440-7451-9.
<http://www.pearson.fr/livre/?GCOI=27440100259650>.
- [2] Céline CHEVALIER : *\LaTeX pour l' impatient*. Minimax. H&K, troisième édition, novembre 2009, ISBN 2-351-41042-4.
http://www.h-k.fr/liens/tp/latex_pour_l_impatient.html.
- [3] Bernard DESGRAUPES : *\LaTeX : Apprentissage, guide et référence*. Vuibert, deuxième édition, février 2003, ISBN 2-7117-4809-X.
<http://www.vuibert.com/livre10910.html>.
- [4] Helmut KOPKA and Patrick W. DALY : *Guide to \LaTeX* . Tools and Techniques for Computer Typesetting. Addison-Wesley, fourth edition, 2004, ISBN 0-321-17385-6.
<http://www.informit.com/store/product.aspx?isbn=0321173856>.
- [5] Leslie LAMPORT : *\LaTeX : a document preparation system*. Addison-Wesley, second edition, 1994, ISBN 0-201-52983-1.
<http://www.informit.com/store/product.aspx?isbn=0201529831>.

L'ouvrage de référence, écrit par l'auteur de \LaTeX lui-même.

- [6] Ivan LAVALLÉE : *\LaTeX pas à pas*. Hermann, novembre 2009, ISBN 2-7056-6975-2.
<http://www.editions-hermann.fr/ficheproduit.php?lang=fr&menu=&ref=Informatique+LaTeX&prodid=775>.
- [7] Vincent LOZANO : *Tout ce que vous avez toujours voulu savoir sur \LaTeX sans jamais oser le demander*.

In Libro Veritas, octobre 2008, ISBN 2-352-09149-7.
<http://www.framabook.org/latex.html>.

Utilisation avancée de \LaTeX

- [8] Michel GOOSSENS, Frank MITTELBACH, Sebastian RAHTZ, Denis ROEGEL, and Herbert VOSS : *The \LaTeX Graphics Companion*. Tools and Techniques for Computer Typesetting. Addison-Wesley, second edition, 2008, ISBN 0-321-50892-0.
<http://www.informit.com/store/product.aspx?isbn=0321508920>.
- [9] Michel GOOSSENS and Sebastian RAHTZ : *The \LaTeX Web Companion: Integrating $T_{\text{E}}X$, HTML, and XML*. Tools and Techniques for Computer Typesetting. Addison-Wesley, 1999, ISBN 0-201-43311-7.
<http://www.informit.com/store/product.aspx?isbn=0201433117>.
- [10] Frank MITTELBACH and Michel GOOSSENS : *The \LaTeX Companion*. Tools and Techniques for Computer Typesetting. Addison-Wesley, second edition, 2004, ISBN 0-201-36299-6.
<http://www.informit.com/store/product.aspx?isbn=0201362996>.

Le compagnon indispensable de toute personne désirant utiliser \LaTeX à son plein potentiel. Ce livre décrit de très nombreuses extensions et astuces permettant de réaliser des documents correspondant exactement à vos attentes.

- [11] Frank MITTELBACH et Michel GOOSSENS : *\LaTeX Companion*. Pearson Education, deuxième édition, 2006, ISBN 2-7440-7182-X.
<http://www.pearson.fr/livre/?GCOI=27440100048330>.
Traduction française de *The \LaTeX Companion*.
- [12] Tanguy ORTOLO et Jeanne LEGRAND : *\LaTeX à 200 %*. O'Reilly, novembre 2006, ISBN 2-84177-270-5.
- [13] Herbert VOSS : *Typesetting mathematics with \LaTeX* . UIT Cambridge, September 2009.
<http://www.uit.co.uk/content/typesetting-mathematics-latex>.
- [14] Herbert VOSS : *Typesetting tables with \LaTeX* . UIT Cambridge, September 2009.
<http://www.uit.co.uk/content/typesetting-tables-latex>.

Bibliographie et index

- [15] Leslie LAMPORT : *MakeIndex: An Index Processor For \LaTeX* , February 1987.

<http://mirror.ctan.org/indexing/makeindex/doc/makeindex.pdf>.

- [16] Oren PATASHNIK : *BibTEXing*, February 1988.
<http://mirror.ctan.org/biblio/bibtex/contrib/doc/btxdoc.pdf>.

Graphisme

- [17] Christian FEUERSÄNGER : *Manual for Package PGFPLOTS*, January 2010.
<http://pgfplots.sourceforge.net/pgfplots.pdf>.

La documentation officielle de PGFPLOTS.

- [18] Till TANTAU : *The TikZ and PGF Packages*, February 2008.
<http://mirror.ctan.org/graphics/pgf/base/doc/generic/pgf/pgfmanual.pdf>.

La documentation officielle de TikZ.

- [19] Gérard TISSEAU et Jacques DUMA : *TikZ pour l' impatient*, novembre 2009.
<http://math.et.info.free.fr/TikZ/>.

Excellente documentation d'introduction à TikZ, en français.

Typographie

- [20] Imprimerie nationale : *Lexique des règles typographiques en usage à l'Imprimerie nationale*, troisième édition, 2002, ISBN 978-2-7433-0482-9.

L'ouvrage de référence sur la typographie française. À toujours avoir à portée de main.

- [21] Jean Pierre LACROUX : *Orthotypographie, Orthographe & Typographie françaises, Dictionnaire raisonné*, 2007.
<http://www.orthotypographie.fr/>.

- [22] Yves PERROUSSEAU : *Règles de l'écriture typographique du français*. Atelier Perrousseau éditeur, neuvième édition, 2009, ISBN 978-2-911220-028-9.
<http://newsletter.perrousseau.com/01/>.

- [23] Madeleine SAUVÉ : *Qu'est-ce qu'un livre ?* Éditions Fides, 2006, ISBN 2-7621-2614-2.
<http://www.editionsfides.com/livre.php?id=157>.

- [24] University of Chicago Press : *The Chicago Manual of Style*, fifteenth edition, 2003, ISBN 0226104036.
<http://www.chicagomanualofstyle.org/>.
L'ouvrage de référence sur la typographie de l'anglais américain.

Divers

- [25] Benjamin BAYART : *Joli manuel pour L^AT_EX*, décembre 1995.
<http://jmpl.fr.eu.org/>.
- [26] Lapo F. MORI and Maurizio W. HIMMELMANN : *Writing the curriculum vitæ with L^AT_EX*.
The PracT_EX Journal, (4), 2007.
<http://www.tug.org/pracjourn/2007-4/mori/>.
- [27] Peter WILSON : *Some Examples of Title Pages*, second edition, 2009.
<http://mirror.ctan.org/info/latex-samples/TitlePages/titlepages.pdf>, le code source est également disponible :
<http://mirror.ctan.org/info/latex-samples/TitlePages/titlepages.tex>.

Index

Symboles	
! (environnements <code>table</code> et <code>figure</code>)	25
47	
! (index)	82
\! (espacement)	60
! ' (i)	24
" (")	174
\" (ë)	25
\" (è)	25
# (caractère réservé)	15
\# (#)	15
#1	31, 49
\$ (caractère réservé)	15
\$ (environnement <code>math</code>)	52
\\$ (\$)	15
% (caractère réservé)	15
% (commentaire)	15
\% (%)	15
& (caractère réservé)	15
& (environnement <code>array</code>)	68
& (environnement <code>tabular</code>)	44
\& (&)	15
\' (é)	25
\' (é)	25
'' (")	174
((délimiteur)	62
\((environnement <code>math</code>)	52
) (délimiteur)	62
\) (environnement <code>math</code>)	52
* (environnement <code>tabular</code>)	45
\, (espacement)	60
\- (coupure des mots)	16
-- (—)	24
--- (—)	24
\. (è)	25
\. (è)	25
/ (délimiteur)	62
\: (espacement)	60
\; (espacement)	60
\= (ē)	25
\= (ē)	25
\= (environnement <code>tabbing</code>)	42
\> (environnement <code>tabbing</code>)	42
? ' (i)	24
@ (index)	83
[(délimiteur)	62
\[(environnement <code>displaymath</code>)	52
\ (caractère réservé)	15
\\ ...	18, 30, 36, 37, 42, 44, 68, 116, 117, 129, 133, 150, 168
\	17
] (délimiteur)	62
\] (environnement <code>displaymath</code>)	52
^ (caractère réservé)	15
^ (exposant)	53
\^ (ê)	25
\^ (è)	25
_ (caractère réservé)	15
_ (indice)	53
_ (—)	15
\^ (è)	25
\^ (è)	25
‘ ‘ (“)	174
~ (caractère réservé)	15
~ (espace insécable)	14
\~ (ñ)	25
\~ (ñ)	25
10pt (option de classe)	11

11pt (option de classe)	11	<code>\ast</code> (*)	56
12pt (option de classe)	11	<code>\asymp</code> (\asymp)	56
<code>\sq</code>	24	AUTHOR (bibliographie)	79
		<code>\author</code>	17, 148
		<code>.aux</code>	6
		avant (extension)	101

A

a4paper (option de classe)	11
<code>\AA</code> (Å)	24
<code>\aa</code> (å)	24
abbrv (bibliographie)	80
abbrv-fr (bibliographie)	80
accents	25
accolades de groupement	24, 27, 53
<code>\acute</code> (á)	58
<code>\addcontentsline</code>	20
ADDRESS (bibliographie)	79
<code>\address</code>	113, 116, 129
<code>\AE</code> (Æ)	24
<code>\ae</code> (æ)	24
<code>\aleph</code> (ℵ)	57
<code>\alpha</code> (α)	55
alpha (bibliographie)	80
alpha-fr (bibliographie)	80
<code>\amalg</code> (II)	56
amsmath (extension)	51, 64
<code>\and</code>	17
and	78
<code>\angle</code> (∠)	57
AnnArbor (thème beamer)	154
annexe	20
ANNOTE (bibliographie)	79
Antibes (thème beamer) ..	154, 155
anttora (extension)	95, 100
<code>\appendix</code>	20
<code>\approx</code> (≈)	56
<code>\arccos</code> (arccos)	67
<code>\arcsin</code> (arcsin)	67
<code>\arctan</code> (arctan)	67
arev (extension)	95, 100
<code>\arg</code> (arg)	67
array (environnement)	68
ARTICLE (bibliographie)	77
article (classe)	10, 110

B

b (environnements table et figure)	47
<code>\b</code> (e)	25
babel (extension)	12, 171–173
backaddress (option KOMA-Script)	120
<code>\backslash</code> (\)	57, 62
<code>\bar</code> (ā)	58
<code>.bb1</code>	81
bbm (extension)	59
bbold (extension)	59
beamer (classe)	148
<code>\begin</code>	33
bera (extension)	100, 103
Bergen (thème beamer) ...	154, 155
Berkeley (thème beamer)	154, 155
Berlin (thème beamer) ...	154, 156
<code>\beta</code> (β)	55
<code>\bfseries</code>	27
<code>.bib</code>	75
<code>\bibitem</code>	74
bibliographie	73
<code>\bibliography</code>	80
<code>\bibliographystyle</code>	80
BIBTEX	73, 75
<code>\bigcap</code> (∩)	63
<code>\bigcirc</code> (○)	56
<code>\bigcup</code> (∪)	63
<code>\bigodot</code> (⊙)	63
<code>\bigoplus</code> (⊕)	63
<code>\bigotimes</code> (⊗)	63
<code>\bigskip</code>	30
<code>\bigsqcup</code> (⊔)	63
<code>\bigtriangledown</code> (∇)	56
<code>\bigtriangleup</code> (△)	56
<code>\biguplus</code> (⊕)	63

<code>\bigvee</code> (\vee)	63	clé	76
<code>\bigwedge</code> (\wedge)	63	classe	10
<code>\bmod</code> (mod)	67	classic (thème moderncv)	123
Boadilla (thème beamer)	154, 156	<code>\clearpage</code>	30
BOOK (bibliographie)	77	<code>\cline</code>	46
book (classe)	11, 110	<code>\closing</code>	116
BOOKLET (bibliographie)	77	<code>\clubsuit</code> (\clubsuit)	57
bookman (extension)	100, 103	cm (unité de mesure)	30
BOOKTITLE (bibliographie)	79	cmbright (extension)	95, 100
<code>\bot</code> (\perp)	57	cmv (couleur)	163
<code>\bowtie</code> (\bowtie)	56	cmv (couleur)	163
<code>\Box</code> (\square)	57	commande	10, 17
<code>\breve</code> (\breve)	58	définir une nouvelle commande	
<code>\bullet</code> (\bullet)	56	30	
		forme étoilée	17
		syntaxe	17
		commentaire	15
		compilation	6
		CONFERENCE (bibliographie)	77
		<code>\cong</code> (\cong)	56
		Copenhagen (thème beamer)	154, 157
		<code>\coprod</code> (\coprod)	63
		<code>\copyright</code> (\copyright)	23
		corps	29
		<code>\cos</code> (cos)	67
		<code>\cosh</code> (cosh)	67
		<code>\cot</code> (cot)	67
		<code>\coth</code> (coth)	67
		coupure des mots	15
		courier (extension)	101
		CROSSREF (bibliographie)	79
		<code>\csc</code> (csc)	67
		CTAN	89
		<code>\cup</code> (\cup)	56
		currvita (extension)	139
		curve (classe)	133
		CV	123
		cv (environnement)	139
		<code>\cvcomputer</code>	131
		<code>\cventry</code>	130, 131
		<code>\cvlanguage</code>	132
		<code>\cvline</code>	132
		cvlist (environnement)	142
C			
<code>\mathbb{C}</code> (\mathbb{C})	60		
<code>\mathbb{m}{C}</code> (\mathbb{C})	60		
<code>\mathbb{mss}{C}</code> (\mathbb{C})	60		
<code>\mathbb{mtt}{C}</code> (\mathbb{C})	60		
<code>\mathds{C}</code> (\mathbb{C})	60		
c (environnement array)	68		
c (environnement tabular)	44		
<code>\c</code> (\c)	25		
CambridgeUS (thème beamer)	156		
<code>\cap</code> (\cap)	56		
<code>\caption</code>	47		
casual (thème moderncv)	123		
ccfonts (extension)	95, 100		
<code>\cdot</code> (\cdot)	56		
<code>\cdots</code> (\cdots)	57, 68		
center (environnement)	37		
<code>\centering</code>	47		
<code>\cfoot</code>	92		
chancery (extension)	100, 103		
CHAPTER (bibliographie)	79		
<code>\chapter</code>	19		
<code>\chapter*</code>	20		
charter (extension)	100, 103		
<code>\thead</code>	92		
<code>\check</code> (\check)	58		
<code>\chi</code> (χ)	55		
<code>\circ</code> (\circ)	56		
<code>\cite</code>	74, 76, 80		

<code>\cvlistdoubleitem</code>	132	<code>dsfont</code> (extension)	59
<code>\cvlistitem</code>	132	E	
<code>cyklop</code> (extension)	100, 103	<code>EDITION</code> (bibliographie)	79
D		<code>EDITOR</code> (bibliographie)	79
<code>\mathbb{D}</code> (\mathbb{D})	60	<code>\ell</code> (ℓ)	57
<code>\mathbb{m}</code> (\mathbb{D})	60	<code>em</code> (unité de mesure)	30
<code>\mathbb{mss}</code> (\mathbb{D})	60	<code>em</code> (environnement)	34
<code>\mathbb{mtt}</code> (\mathbb{D})	60	<code>\email</code>	129
<code>\mathds</code> (\mathbb{D})	60	<code>\emph</code>	28
<code>\d</code> (\d)	25	<code>\emptyset</code> (\emptyset)	57
déclaration	27, 33	<code>\end</code>	33
<code>\dag</code> (\dagger)	23	<code>\entry*</code>	139
<code>\dagger</code> (\dagger)	56	<code>enumerate</code> (environnement)	39
Darmstadt (thème beamer)	154, 157	environnement	33
<code>\dashv</code> (\dashv)	56	définir un nouvel environnement	
<code>\date</code>	18, 148	48	
date (variable KOMA-Script) ..	120	forme étoilée	33
<code>\ddag</code> (\ddagger)	23	syntaxe	33
<code>\ddagger</code> (\ddagger)	56	<code>\epsilon</code> (ϵ)	55
<code>\ddot</code> (\ddot{a})	58	<code>eqnarray</code> (environnement)	70
<code>\ddots</code> (\ddots)	68	<code>eqnarray*</code> (environnement)	70
default (thème beamer)	154	équation	69
<code>\definecolor</code>	163	equation (environnement)	69
<code>\deg</code> (\deg)	67	<code>\equiv</code> (\equiv)	56
<code>\Delta</code> (Δ)	55	espace	13
<code>\delta</code> (δ)	55	espace insécable	14
description (environnement) ..	40	<code>\eta</code> (η)	55
<code>\det</code> (\det)	67	<code>eulervm</code> (extension)	100
<code>\Diamond</code> (\diamond)	57	<code>\EUR</code> (€)	25
<code>\diamond</code> (\diamond)	56	<code>\euro</code> (€)	25
<code>\diamondsuit</code> (\diamond)	57	<code>eurosym</code> (extension)	25
<code>\dim</code> (\dim)	67	<code>ex</code> (unité de mesure)	30
<code>displaymath</code> (environnement) ..	51, 52	<code>\exists</code> (\exists)	57
<code>\displaystyle</code>	61, 64	<code>\exp</code> (\exp)	67
<code>\div</code> (\div)	56	exposant	53
document (environnement)	13	extension	11
<code>\documentclass</code>	10	<code>\extrainfo</code>	129
<code>\dot</code> (\dot{a})	58	F	
<code>\doteq</code> (\doteq)	56	<code>\familyname</code>	129
<code>\Downarrow</code> (\Downarrow)	57, 62	<code>\fancyfoot</code>	93
<code>\downarrow</code> (\downarrow)	57, 62	<code>fancyhdr</code> (extension)	91
Dresden (thème beamer) ..	154, 157	<code>\fancyhead</code>	93

-
- `\fax` 129
`.fdb_latexmk` 86
`\fg (»)` 173
`figure` 46
`figure` (environnement) 46
`\firstname` 126
`\flat (b)` 57
`fleqn` (option de classe) 53
`flushleft` (environnement) 37
`flushright` (environnement) ... 37
`foldmarks` (option KOMA-Script) .
120
`fontenc` (extension) 12
`\footnote` 22
`\footnotesize` 29
`\forall` 57
forme étoilée
 commande 17
 environnement 33
`fourier` (extension) 95, 100
`fouriernc` (extension) 95, 100
`\frac` 61
`fraction` 61
`frame` (environnement) ... 148, 150,
151
`\frametitle` 151
Frankfurt (thème beamer) 154, 158
`french` (option d'extension) 12,
171–173
`fromaddress` (variable KOMA-Script)
120
`fromname` (variable KOMA-Script)
120
`\frown (⤵)` 56
- G**
- `\Gamma (Γ)` 55
`\gamma (γ)` 55
`\gcd` (gcd) 67
`geometry` (extension) 90
`\geq (≥)` 56
`gfsartemisia` (extension) 95, 100
`\gg (≫)` 56
- Goettingen (thème beamer) .. 154,
158
`graphicx` (extension) 106
`\grave (à)` 58
`grec` 55
- H**
- `\H (ë)` 25
`h` (environnements table et figure)
47
Hannover (thème beamer) 154, 158
`\hat (â)` 58
`\hbar (ħ)` 57
`\heartsuit (♥)` 57
`helvet` (extension) 101
`\hline` 44
`\hom (hom)` 67
`\hookleftarrow (↵)` 57
`\hookrightarrow (↶)` 57
HOWPUBLISHED (bibliographie) .. 79
`hsb` (couleur) 163
`\hspace` 30
`\hspace*` 30
`\Huge` 29
`\huge` 29
`hyperref` (extension) 6, 12
- I**
- `\i (i)` 26
`.idx` 7, 83
`\ieme (e)` 172
`\iemes (es)` 172
`\ier (er)` 172
`\iere (re)` 172
`\ieres (res)` 172
`\iers (ers)` 172
Ilmenau (thème beamer) . 154, 159
`\Im (ℑ)` 57
`\imath (i)` 57, 58
`in` (unité de mesure) 30
`\in (∈)` 56
INBOOK (bibliographie) 77
`\include` 20, 21
`\includegraphics` 106, 164

INCOLLECTION (bibliographie) ..	77	l (environnement <code>array</code>)	68
inconsolata (extension)	101	l (environnement <code>tabular</code>)	43
.ind	83	\l (l)	24
index	82	\label	22, 47, 69
\index	82	\Lambda (Λ)	55
indice	53	\lambda (λ)	55
\inf (inf)	67	\langle (\langle)	62
\infty (∞)	57	\LARGE	29
INPROCEEDINGS (bibliographie) .	77	\Large	29
\input	21	\large	29
inputenc (extension)	12	L ^A T _E X	1
\institute	148	\LaTeX (L ^A T _E X)	24
INSTITUTION (bibliographie)	79	\LaTeXe (L ^A T _E X 2 _{ϵ})	24
\int (\int)	63	latexmk	86
intégrale	63	~/ .latexmkrc	86
\iota (ι)	55	latexsym (extension)	54, 56, 57
ISO 8859-1	12	latin1 (option d'extension)	12
ISO 8859-15	12	latin9 (option d'extension)	12
.ist	84	\lceil (\lceil)	62
\item	39, 40, 142	\ldots (...)	57
itemize (environnement)	39	\ldots (...)	174
\itshape	27	\leadsto (\leadsto)	57
iwona (extension)	95, 100	\left	62
J			
\j (j)	26	\left.	63
\jmath (j)	57, 58	\Leftarrow (\Leftarrow)	57
\Join (\Join)	56	\leftarrow (\leftarrow)	57
JOURNAL (bibliographie)	79	\leftharpoondown (\leftharpoondown)	57
JPEG	106	\leftharpoonup (\leftharpoonup)	57
JuanLesPins (thème beamer) .	154,	\leftheadr	133
159		\leftmark	93
K			
\kappa (κ)	55	\Leftrightarrow (\Leftrightarrow)	57
\ker (ker)	67	\leftrightharpoonup (\leftrightharpoonup)	57
KEY (bibliographie)	79	\leq (\leq)	56
\kill	42	leqno (option de classe)	69
KOMA-Script	110	letter (classe)	11, 110, 113
\KOMAsptions	120	letter (environnement) ..	113, 116,
kpfonts (extension)	95, 100	121	
kurier (extension)	95, 100	lettre	113
L			
\L (\L)	24	\lfloor (\lfloor)	62
		\lfoot	92
		\lg (lg)	67
		\lhd (\lhd)	56
		\lhead	92
		libertine (extension)	100, 103

<code>\sigma</code> (σ)	55	symboles mathématiques	
<code>\signature</code>	113, 116	accents	58
signature (variable KOMA-Script)		délimiteurs	62
120		ensembles	59
<code>\sim</code> (\sim)	56	flèches	57
<code>\simeq</code> (\simeq)	56	fonctions et opérateurs	66
<code>\sin</code> (\sin)	67	opérateurs binaires	56
Singapore (thème beamer)	154, 162	relations binaires	56
<code>\sinh</code> (\sinh)	67	symboles de taille variable ..	63
slide (environnement)	145	symboles divers	57
slides (classe)	11, 145	Szeged (thème beamer) ...	154, 162
<code>\slshape</code>	27		
<code>\small</code>	29	T	
<code>\smallskip</code>	30	t (environnements <code>table</code> et <code>figure</code>)	
<code>\smile</code> (\smile)	56	47	
somme	63	<code>\t</code> (êe)	25
<code>\spadesuit</code> (\spadesuit)	57	T1 (option d'extension)	12
<code>\sqcap</code> (\sqcap)	56	tabbing (environnement)	41
<code>\sqcup</code> (\sqcup)	56	table (environnement)	46
<code>\sqrt</code>	62	table des matières	19
<code>\sqsubset</code> (\sqsubset)	56	tableau	43, 46
<code>\sqsubseteq</code> (\sqsubseteq)	56	<code>\tableofcontents</code>	19, 87
<code>\sqsupset</code> (\sqsupset)	56	tabular (environnement)	43
<code>\sqsupseteq</code> (\sqsupseteq)	56	tabulation	41
<code>\ss</code> (β)	24	<code>\tan</code> (\tan)	67
<code>\stackrel</code>	59	<code>\tanh</code> (\tanh)	67
<code>\star</code> (\star)	56	<code>\tau</code> (τ)	55
subject (variable KOMA-Script) ..		TECHREPORT (bibliographie)	78
120		$\text{T}_{\text{E}}\text{X}$	1
<code>\subsection</code>	19, 130	<code>.tex</code>	6
<code>\subsection*</code>	20	$\text{T}_{\text{E}}\text{X}$ Live	3, 89
<code>\subset</code> (\subset)	56	texdoc	89
<code>\subseteq</code> (\subseteq)	56	texindy	84
<code>\subsubsection</code>	19	<code>\textasciicircum</code> (\textasciicircum)	15
<code>\subsubsection*</code>	20	<code>\textasciitilde</code> (\textasciitilde)	15
<code>\subtitle</code>	136, 148	<code>\textbackslash</code> (\textbackslash)	15
<code>\succ</code> (\succ)	56	<code>\textbb</code>	59
<code>\succeq</code> (\succeq)	56	<code>\textbf</code>	26, 27
<code>\sum</code> (\sum)	63	textcomp (extension)	172
<code>\sup</code> (\sup)	67	<code>\textit</code>	26, 27
<code>\supset</code> (\supset)	56	<code>\textmd</code>	26, 27
<code>\supseteq</code> (\supseteq)	56	<code>\textrm</code>	26, 27
<code>\surd</code> (\surd)	57	<code>\textsc</code>	26, 27
<code>\swarrow</code> (\swarrow)	57	<code>\textsf</code>	26, 27

`\vspace*` 30

W

Warsaw (thème beamer) ... 154, 163

`\wedge` (\wedge) 56

`\widehat` (\widehat{xyz}) 58

`\widetilde` (\widetilde{xyz}) 58

`\wp` (\wp) 57

`\wr` (\wr) 56

WYSIWYG 2

X

`.xdy` 85

`\Xi` (Ξ) 55

`\xi` (ξ) 55

`xindy` 84

Y

YEAR (bibliographie) 79

`yourref` (variable KOMA-Script) ..
120

Z

`\mathbb{Z}` (\mathbb{Z}) 60

`\mathbbm{Z}` (\mathbb{Z}) 60

`\mathbbmss{Z}` (\mathbb{Z}) 60

`\mathbbmtt{Z}` (\mathbb{Z}) 60

`\mathds{Z}` (\mathbb{Z}) 60

`\zeta` (ζ) 55

Liste des tableaux

4.1	Commandes de hiérarchisation	19
4.2	Symboles divers	23
4.3	Symboles pour le français et les langues étrangères	24
4.4	Commandes d’accentuation	25
4.5	Commandes simplifiées d’accentuation	25
4.6	Commandes de changement de style	26
4.7	Correspondance entre commandes et déclarations de changement de style	27
4.8	Déclarations de changement de corps	29
4.9	Unités de mesure utilisées par \LaTeX	30
5.1	Exemple d’environnement <code>table</code>	47
6.1	Lettres grecques	55
6.2	Symboles d’opérateurs binaires	56
6.3	Symboles de relations binaires	56
6.4	Flèches	57
6.5	Symboles divers	57
6.6	Accents	58
6.7	Ensembles	60
6.8	Commandes d’espace	60
6.9	Commandes de changement de style	60
6.10	Délimiteurs	62
6.11	Symboles de taille variable	63
6.12	Fonctions et opérateurs	67
7.1	Types d’entrées reconnus par \BibTeX	77
7.2	Champs utilisés par \BibTeX	79
8.1	Arguments optionnels utilisables avec les commandes <code>\fancyhead</code> et <code>\fancyfoot</code>	94
8.2	Extensions permettant de déterminer la police de caractères (avec mode mathématique)	95
8.3	Extensions permettant de déterminer une police de caractères complémentaire	101

8.4 Extensions permettant de déterminer la police de caractères (sans mode mathématique)	103
8.5 Correspondance entre classes standards et classes KOMA-Script	110
9.1 Variables utilisables avec la classe <code>scr1tr2</code>	120
11.1 Thèmes utilisables avec <code>beamer</code>	154
11.2 Couleurs prédéfinies utilisables avec <code>beamer</code>	164
A.1 Madame, mademoiselle, monsieur	172
A.2 Premier, deuxième, etc.	172

Table des figures

8.1	Structure de l'en-tête et du pied de page avec l'extension <code>fancyhdr</code>	91
8.2	Exemple de document réalisé avec l'extension <code>anttora</code> (option <code>math</code>)	95
8.3	Exemple de document réalisé avec l'extension <code>arev</code>	95
8.4	Exemple de document réalisé avec l'extension <code>ccfonts</code>	96
8.5	Exemple de document réalisé avec l'extension <code>cmbright</code>	96
8.6	Exemple de document réalisé avec l'extension <code>fourier</code>	96
8.7	Exemple de document réalisé avec l'extension <code>fouriernc</code>	96
8.8	Exemple de document réalisé avec l'extension <code>gfsartemis</code>	97
8.9	Exemple de document réalisé avec l'extension <code>iwona</code> (option <code>math</code>)	97
8.10	Exemple de document réalisé avec l'extension <code>kpfonts</code>	97
8.11	Exemple de document réalisé avec l'extension <code>kurier</code> (option <code>math</code>)	98
8.12	Exemple de document réalisé avec l'extension <code>lmodern</code>	98
8.13	Exemple de document réalisé avec l'extension <code>mathdesign</code> (option <code>charter</code>)	98
8.14	Exemple de document réalisé avec l'extension <code>mathdesign</code> (option <code>utopia</code>)	99
8.15	Exemple de document réalisé avec l'extension <code>mathpazo</code>	99
8.16	Exemple de document réalisé avec l'extension <code>mathptmx</code>	99
8.17	Exemple de document réalisé avec l'extension <code>avant</code>	101
8.18	Exemple de document réalisé avec l'extension <code>helvet</code>	101
8.19	Exemple de document réalisé avec l'extension <code>tgadventor</code>	101
8.20	Exemple de document réalisé avec l'extension <code>tgheros</code>	102
8.21	Exemple de document réalisé avec l'extension <code>courier</code>	102
8.22	Exemple de document réalisé avec l'extension <code>inconsolata</code>	102
8.23	Exemple de document réalisé avec l'extension <code>tgcursor</code>	102
8.24	Exemple de document réalisé avec l'extension <code>eulervm</code>	102
8.25	Exemple de document réalisé avec l'extension <code>bera</code>	103
8.26	Exemple de document réalisé avec l'extension <code>bookman</code>	103
8.27	Exemple de document réalisé avec l'extension <code>chancery</code>	103
8.28	Exemple de document réalisé avec l'extension <code>charter</code>	104

8.29	Exemple de document réalisé avec l'extension <code>cyklop</code>	104
8.30	Exemple de document réalisé avec l'extension <code>libertine</code> . . .	104
8.31	Exemple de document réalisé avec l'extension <code>newcent</code> . . .	104
8.32	Exemple de document réalisé avec l'extension <code>tgbonum</code> . .	105
8.33	Exemple de document réalisé avec l'extension <code>tgchorus</code> . .	105
8.34	Exemple de document réalisé avec l'extension <code>tgpagella</code> . .	105
8.35	Exemple de document réalisé avec l'extension <code>tgschola</code> . .	105
8.36	Exemple de document réalisé avec l'extension <code>tgtermes</code> . .	106
8.37	Exemple de graphique réalisé avec l'extension <code>TikZ</code>	108
9.1	Exemple de lettre réalisée avec la classe <code>letter</code> (code source)	114
9.2	Exemple de lettre réalisée avec la classe <code>letter</code> (document général)	115
9.3	Exemple de lettre réalisée avec la classe <code>scrlttr2</code> (code source)	118
9.4	Exemple de lettre réalisée avec la classe <code>scrlttr2</code> (document général)	119
10.1	Exemple de CV réalisé avec la classe <code>moderncv</code> (code source)	124
10.2	Exemple de CV réalisé avec la classe <code>moderncv</code> , thème <code>casual</code> (document général)	127
10.3	Exemple de CV réalisé avec la classe <code>moderncv</code> , thème <code>classic</code> (document général)	128
10.4	Exemple de CV réalisé avec la classe <code>curve</code> (code source) . .	134
10.5	Exemple de CV réalisé avec la classe <code>curve</code> (document général)	135
10.6	Exemple de CV réalisé avec la classe <code>curve</code> (code source du fichier <code>curve-experience.tex</code>)	137
10.7	Exemple de CV réalisé avec la classe <code>curve</code> (code source du fichier <code>curve-references.tex</code>)	137
10.8	Exemple de CV réalisé avec la classe <code>curve</code> (code source du fichier <code>curve-methodologies.tex</code>)	138
10.9	Exemple de CV réalisé avec la classe <code>curve</code> (code source du fichier <code>curve-formation.tex</code>)	138
10.10	Exemple de CV réalisé avec la classe <code>curve</code> (code source du fichier <code>curve-langues.tex</code>)	138
10.11	Exemple de CV réalisé avec la classe <code>curve</code> (code source du fichier <code>curve-extra.tex</code>)	138
10.12	Exemple de CV réalisé avec l'extension <code>currvita</code> (code source)	140
10.13	Exemple de CV réalisé avec l'extension <code>currvita</code> (document général)	143
11.1	Exemple de transparent réalisé avec la classe <code>slides</code>	147
11.2	Structure d'une présentation réalisée avec la classe <code>beamer</code>	149
11.3	Exemple de page de titre réalisée avec la classe <code>beamer</code> . .	150

11.4	Exemple de présentation réalisée avec la classe <code>beamer</code> et le thème <code>default</code>	154
11.5	Exemple de présentation réalisée avec la classe <code>beamer</code> et le thème <code>AnnArbor</code>	154
11.6	Exemple de présentation réalisée avec la classe <code>beamer</code> et le thème <code>Antibes</code>	155
11.7	Exemple de présentation réalisée avec la classe <code>beamer</code> et le thème <code>Bergen</code>	155
11.8	Exemple de présentation réalisée avec la classe <code>beamer</code> et le thème <code>Berkeley</code>	155
11.9	Exemple de présentation réalisée avec la classe <code>beamer</code> et le thème <code>Berlin</code>	156
11.10	Exemple de présentation réalisée avec la classe <code>beamer</code> et le thème <code>Boadilla</code>	156
11.11	Exemple de présentation réalisée avec la classe <code>beamer</code> et le thème <code>CambridgeUS</code>	156
11.12	Exemple de présentation réalisée avec la classe <code>beamer</code> et le thème <code>Copenhagen</code>	157
11.13	Exemple de présentation réalisée avec la classe <code>beamer</code> et le thème <code>Darmstadt</code>	157
11.14	Exemple de présentation réalisée avec la classe <code>beamer</code> et le thème <code>Dresden</code>	157
11.15	Exemple de présentation réalisée avec la classe <code>beamer</code> et le thème <code>Frankfurt</code>	158
11.16	Exemple de présentation réalisée avec la classe <code>beamer</code> et le thème <code>Goettingen</code>	158
11.17	Exemple de présentation réalisée avec la classe <code>beamer</code> et le thème <code>Hannover</code>	158
11.18	Exemple de présentation réalisée avec la classe <code>beamer</code> et le thème <code>Ilmenau</code>	159
11.19	Exemple de présentation réalisée avec la classe <code>beamer</code> et le thème <code>JuanLesPins</code>	159
11.20	Exemple de présentation réalisée avec la classe <code>beamer</code> et le thème <code>Luebeck</code>	159
11.21	Exemple de présentation réalisée avec la classe <code>beamer</code> et le thème <code>Madrid</code>	160
11.22	Exemple de présentation réalisée avec la classe <code>beamer</code> et le thème <code>Malmoe</code>	160
11.23	Exemple de présentation réalisée avec la classe <code>beamer</code> et le thème <code>Marburg</code>	160
11.24	Exemple de présentation réalisée avec la classe <code>beamer</code> et le thème <code>Montpellier</code>	161
11.25	Exemple de présentation réalisée avec la classe <code>beamer</code> et le thème <code>PaloAlto</code>	161

Table des figures

11.26	Exemple de présentation réalisée avec la classe beamer et le thème Pittsburgh	161
11.27	Exemple de présentation réalisée avec la classe beamer et le thème Rochester	162
11.28	Exemple de présentation réalisée avec la classe beamer et le thème Singapore	162
11.29	Exemple de présentation réalisée avec la classe beamer et le thème Szeged	162
11.30	Exemple de présentation réalisée avec la classe beamer et le thème Warsaw	163

Table des matières

Avant-propos	iii
1 Introduction	1
1.1 Qu'est-ce que \LaTeX ?	1
1.2 La philosophie de \LaTeX	2
1.3 \LaTeX et la typographie	2
1.4 Comment se procurer \LaTeX ?	3
2 Votre premier document	5
2.1 L'édition	5
2.2 La compilation	6
2.3 Les fichiers générés	6
2.4 La détection et la correction des erreurs	7
2.5 La visualisation	7
3 Les concepts de base de \LaTeX	9
3.1 Le document \LaTeX	9
3.1.1 Syntaxe d'une commande \LaTeX	10
3.1.2 La commande <code>\documentclass</code>	10
3.1.2.1 La classe d'un document	10
3.1.2.2 Les options de classe	11
3.1.3 Le préambule	11
3.1.4 Les extensions	11
3.1.5 Le document	13
3.2 Structure du document	13
3.2.1 Les paragraphes	13
3.2.2 Les mots	13
3.2.3 L'espace insécable : <code>~</code>	14
3.2.4 Les commentaires	15
3.3 Caractères spéciaux utilisés par \LaTeX	15
3.4 Les problèmes de coupure des mots : <code>Overfull \hbox</code>	15
4 Les commandes de \LaTeX	17
4.1 Syntaxe d'une commande	17
4.2 La page de titre	17

4.3	Commandes de hiérarchisation du document	19
4.3.1	Forme étoilée	20
4.3.2	Annexes	20
4.4	Construire un document à partir de plusieurs fichiers	20
4.5	Commandes d’annotation du document	21
4.5.1	Les notes de bas de page	22
4.5.2	Les notes dans la marge	22
4.6	Les références croisées	22
4.7	Caractères spéciaux	23
4.7.1	Symboles	23
4.7.2	Les lettres accentuées	25
4.8	Commandes et déclarations de changement de style	26
4.9	Déclarations de changement de corps	29
4.10	Commandes de mise en pages	29
4.11	Définir de nouvelles commandes	30
5	Les environnements de L^AT_EX	33
5.1	Syntaxe d’un environnement	33
5.2	L’environnement <code>em</code>	34
5.3	Les environnements <code>quote</code> et <code>quotation</code>	34
5.4	L’environnement <code>verse</code>	36
5.5	Les environnements <code>center</code> , <code>flushleft</code> et <code>flushright</code>	37
5.6	L’environnement <code>verbatim</code>	38
5.7	Les listes : <code>itemize</code> , <code>enumerate</code> et <code>description</code>	39
5.8	L’environnement <code>tabbing</code>	41
5.9	Les tableaux : l’environnement <code>tabular</code>	43
5.10	Les environnements <code>table</code> et <code>figure</code>	46
5.11	Personnaliser la page de titre : l’environnement <code>titlepage</code>	47
5.12	Définir de nouveaux environnements	48
6	L^AT_EX mathématique	51
6.1	Les modes mathématiques	51
6.1.1	L’environnement <code>math</code>	51
6.1.2	L’environnement <code>displaymath</code>	52
6.2	Structures couramment utilisées en mathématiques	53
6.2.1	Indices et exposants	53
6.2.2	Texte à l’intérieur d’une formule	54
6.3	Lettres, symboles et accents	54
6.3.1	Lettres grecques	55
6.3.2	Symboles d’opérateurs binaires	56
6.3.3	Symboles de relations binaires	56
6.3.4	Flèches	57
6.3.5	Symboles divers	57
6.3.6	Accents	58

6.3.7	Superposer deux symboles	59
6.3.8	Ensembles	59
6.4	Modifier l'espace	60
6.5	Modifier le style et la taille	60
6.6	Fractions	61
6.7	Racines	62
6.8	Délimiteurs	62
6.9	Symboles de taille variable	63
6.10	Mathématiques et typographie	65
6.11	Fonctions et opérateurs	66
6.12	Les tableaux : l'environnement <code>array</code>	68
6.13	Les équations numérotées : l'environnement <code>equation</code>	69
6.14	Les équations numérotées et alignées : l'environnement <code>eqnarray</code>	70
6.15	Théorèmes, définitions...	71
7	Bibliographie et index	73
7.1	Faire une bibliographie	73
7.1.1	L'environnement <code>thebibliography</code>	73
7.1.2	BIB _T E _X	75
7.1.2.1	La base de données bibliographique	75
7.1.2.2	Les différents types d'entrées	76
7.1.2.3	Les différents types de champs	78
7.1.2.4	Quelques précisions sur certains champs	78
7.1.2.5	Générer la bibliographie avec BIB _T E _X	80
7.2	Faire un index	82
7.2.1	La commande <code>\index</code>	82
7.2.2	Générer l'index	83
7.2.3	Générer l'index avec <code>MakeIndex</code>	84
7.2.4	Générer l'index avec <code>xindy</code>	84
7.3	La commande <code>latexmk</code>	85
7.4	Bibliographie et index dans la table des matières	87
8	Aller plus loin avec L^AT_EX	89
8.1	CTAN : <i>the Comprehensive T_EX Archive Network</i>	89
8.2	La commande <code>texdoc</code>	89
8.3	Modifier les marges ou le format de papier : l'extension <code>geometry</code>	90
8.4	Définir l'en-tête et le pied de page : l'extension <code>fancyhdr</code>	91
8.4.1	Les filets	92
8.4.2	En-tête et pied de page dans un document mis en pages en recto seul	92
8.4.3	En-tête et pied de page dans un document mis en pages en recto-verso	93
8.5	Changer la police de caractères	94

8.6	Inclure des graphiques	106
8.6.1	La commande <code>\includegraphics</code>	106
8.6.2	L'extension <code>TikZ</code>	108
8.7	Utiliser d'autres classes que les classes standards	109
8.7.1	KOMA-Script	110
8.7.2	La classe <code>memoir</code>	110
9	Écrire une lettre : les classes <code>letter</code> et <code>scrlttr2</code>	113
9.1	La classe <code>letter</code>	113
9.1.1	Les commandes mutualisables	113
9.1.1.1	La commande <code>\address</code>	116
9.1.1.2	La commande <code>\signature</code>	116
9.1.2	L'environnement <code>letter</code>	116
9.2	La classe <code>scrlttr2</code>	116
9.2.1	L'option de classe <code>NF</code>	117
9.2.2	Les variables	117
9.2.3	Les options	120
9.2.4	L'environnement <code>letter</code>	121
9.3	Règles typographiques	121
9.3.1	Adresse	121
9.3.2	Numéro de téléphone	121
10	Rédiger un CV	123
10.1	La classe <code>moderncv</code>	123
10.1.1	Les thèmes	123
10.1.2	Les informations personnelles	126
10.1.3	Les rubriques	130
10.1.4	Le contenu des rubriques	130
10.1.4.1	La commande <code>\cventry</code>	130
10.1.4.2	La commande <code>\cvcomputer</code>	131
10.1.4.3	La commande <code>\cvlanguage</code>	132
10.1.4.4	La commande <code>\cvline</code>	132
10.1.4.5	La commande <code>\cvlistitem</code>	132
10.1.4.6	La commande <code>\cvlistdoubleitem</code>	132
10.1.5	Modifier la taille de la partie gauche	132
10.2	La classe <code>curve</code>	133
10.2.1	Les informations personnelles	133
10.2.2	Les rubriques	137
10.3	L'extension <code>currvita</code>	139
11	Faire une présentation : les classes <code>slides</code> et <code>beamer</code>	145
11.1	La classe <code>slides</code>	145
11.2	La classe <code>beamer</code>	146
11.2.1	Structure d'une présentation	148

11.2.2 La page de titre	148
11.2.3 Les pages intérieures	151
11.2.4 Modification de l’aspect des présentations : les thèmes	153
11.2.5 Personnalisation de l’aspect des présentations	163
11.2.5.1 Modification des couleurs	163
11.2.5.2 Ajout d’un logo	164
11.2.6 Pour aller plus loin	164
12 Et maintenant ?	167
12.1 Pour en apprendre davantage	167
12.2 Obtenir de l’aide	168
12.3 Conclusion	168
12.3.1 Il faut construire son document en pensant à sa structure	168
12.3.2 Il ne faut pas bidouiller	169
12.3.3 Il faut soigner le fichier source	169
12.4 À vous de jouer!	170
A Quelques règles typographiques françaises et anglaises	171
A.1 Quelques règles typographiques françaises	171
A.2 Quelques règles typographiques anglaises	173
B Différences entre L^AT_EX 2.09 et L^AT_EX 2_ε	175
B.1 Les commandes <code>\documentstyle</code> , <code>\documentclass</code> et <code>\usepackage</code>	175
B.2 Déclarations et commandes de changement de style	175
B.3 S ^L I ^T E _X et la classe <code>slides</code>	176
B.4 La commande <code>\LaTeXe</code>	176
Bibliographie	177
Index	181
Liste des tableaux	193
Table des figures	195
Table des matières	199