

L<sup>A</sup>T<sub>E</sub>X de A à X

krystof

29 juin 2022



# Sommaire



## **Première partie**

# **Découvrir et installer L<sup>A</sup>T<sub>E</sub>X**



# Chapitre 1

## Présentation de L<sup>A</sup>T<sub>E</sub>X

### 1.1 L<sup>A</sup>T<sub>E</sub>X, c'est...

L<sup>A</sup>T<sub>E</sub>X (se prononce *latek*) est un langage créé originellement par des scientifiques qui cherchaient à rédiger des documents capables de gérer les mises en forme d'expression mathématiques, tout en offrant la possibilité d'y ajouter des extensions. Cette spécificité liée aux écritures mathématiques a rendu L<sup>A</sup>T<sub>E</sub>X populaire parmi la communauté scientifique.



FIGURE 1.1 – Logo de L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X est un langage de balisage qu'il est aisé d'apprendre, et s'avère très utile. Au regard du rendu et de l'élégance des documents produits (mise en page de façon professionnelle), il offre de larges possibilités pour la rédaction d'articles, de mémoires, de thèses, mais aussi de livres. C'est en fait un langage de description qui permet de respecter les normes éditoriales et typographiques.

Comparativement aux éditeurs de texte plus connus, tel que *LibreOffice*, L<sup>A</sup>T<sub>E</sub>X se montre bien plus efficace et aisé d'utilisation pour :

- la modification des styles de titres ;
- la gestion des notes ;
- la gestion des flottants (les figures que l'on insère dans les documents) ;
- la rédaction et la gestion des longs documents ;
- la hiérarchisation du texte (parties, chapitres, sections, etc.) ;
- la gestion des références internes au document ;
- la gestion des bibliographies, index et tables des matières.

Avec L<sup>A</sup>T<sub>E</sub>X tout est modifiable et paramétrable. De plus, nous pouvons à partir d'un document rédigé en L<sup>A</sup>T<sub>E</sub>X de générer, par une simple compilation du fichier source `.tex`, des documents en PDF.

## 1.2 La petite histoire de $\text{\LaTeX}$

Tout débute en 1977 par la création du langage  $\text{\TeX}$  par Donald Erwin KNUTH qui cherchait alors à accroître la lisibilité et à optimiser l’insertion des formules mathématiques dans les articles scientifiques.

En 1985, Leslie LAMPORT crée  $\text{\LaTeX}$ , bien plus aisé à utiliser que  $\text{\TeX}$ . Diverses tâches sont alors simplifiées à l’aide de macros intégrées au programme. Par la suite, suite à une évolution majeure, ce sera  $\text{\LaTeX 2}_{\epsilon}$  qui sera majoritairement utilisé.



## Chapitre 2

# Installer L<sup>A</sup>T<sub>E</sub>X

Une installation fonctionnelle comporte trois éléments :

**Une distribution L<sup>A</sup>T<sub>E</sub>X** : le logiciel comprenant toutes les composantes nécessaires de L<sup>A</sup>T<sub>E</sub>X.

**Un lecteur PostScript et/ou PDF** : afin de lire ses productions.

**Un éditeur L<sup>A</sup>T<sub>E</sub>X** : qui facilite la rédaction d'un document rédigé en L<sup>A</sup>T<sub>E</sub>X, même si un éditeur plus classique (exemple : *Vim*) peut suffire.

### Installation sous *Debian GNU/Linux*

Pour la distribution L<sup>A</sup>T<sub>E</sub>X on peut choisir d'installer le paquet *texlive* (distribution de base) ou le paquet *texlive-full* (distribution plus complète). On ajoutera le paquet *cm-super* afin de disposer de polices supplémentaires.

Pour lire et manipuler les fichiers .ps on installera *gv*, et pour les fichiers PDF le lecteur *evince* (bien intégré au bureau *Gnome* sera suffisant).

Enfin, concernant l'éditeur L<sup>A</sup>T<sub>E</sub>X cela est souvent une question de goût et/ou d'habitude personnelle. Pour ma part ma préférence va au logiciel *Texmaker*, qui dispose en plus d'une interface en français.



## Chapitre 3

# Mécanisme et structuration d'un document L<sup>A</sup>T<sub>E</sub>X

### 3.1 Un document à compiler

Un document L<sup>A</sup>T<sub>E</sub>X est avant tout du code qu'il est nécessaire de compiler pour obtenir un document lisible avec une mise en page élégante (du moins celle que l'on souhaite obtenir). Ce code se rédige dans un fichier `.tex` et la compilation, possible grâce à la distribution L<sup>A</sup>T<sub>E</sub>X nous permettra de lire notre document au format PS ou PDF.

Une compilation manuelle est tout à fait possible, mais pour un rendu quasi immédiat l'utilisation d'un éditeur L<sup>A</sup>T<sub>E</sub>X facilite grandement notre travail.

La rédaction d'un long document, tel un livre, nécessitera sûrement la création d'une bibliographie, d'un index, voire d'autres éléments complémentaires. Pour cela L<sup>A</sup>T<sub>E</sub>X stocke de telles informations dans divers fichiers aux extensions différentes. La compilation de notre fichier `.tex` de départ générera alors une multitude de fichiers répondant aux besoins du document.

### 3.2 Un premier document en guise d'exemple

A l'aide d'un éditeur de texte L<sup>A</sup>T<sub>E</sub>X saisir les lignes suivantes :

```
\documentclass{article}

\begin{document}
Bonjour, voici donc un premier document très simple.
\end{document}
```

Il suffit ensuite d'enregistrer le fichier au format `.tex` dans un répertoire dédié. Le fichier doit ensuite être compilé, soit à l'aide de l'éditeur, ou bien sous GNU/Linux grâce à la ligne de commande.

Imaginons que nous ayons enregistré notre document sous le nom `document_simple.tex`, voici comment le compiler et transformer les fichiers en celui voulu :

```
$ latex document_simple.tex : compilation qui permet d'obtenir le fichier document_simple.dvi.
$ xdvi document_simple.dvi : lecture du fichier grâce à la commande xdvi.
$ dvips document_simple.dvi -o : transformer le fichier .dvi en un fichier .ps grâce à la commande dvips.
$ ps2pdf document_simple.ps : transformation en un fichier PostScript, grâce à la commande ps2pdf.
```

\$ **pdflatex document\_simple.tex** : compilation directe en un fichier PDF grâce à la commande `pdflatex`.

\$ **xpdf document\_simple.pdf** : lecture du fichier PDF grâce à la commande `xpdf`.

### 3.3 Compilation et caractères spéciaux

Comme tout langage de programmation,  $\text{\LaTeX}$  utilise certains caractères pour son usage propre. Il en existe dix. Insérer l'un de ces caractères dans votre texte et il en résultera des erreurs de compilation. La parade à cela est d'insérer un *backslash* juste avant le caractère (Ex : `\%`). Pour le *backslash* lui-même nous utilisons la commande suivante : `\textbackslash{}`.

Les caractères spéciaux : `$ & % # _ ^ ~ \ { }`

Il existe un nombre encore conséquent de caractères spéciaux utilisés dans un environnement mathématique et autres<sup>1</sup>.

### 3.4 Structuration d'un document $\text{\LaTeX}$

Le texte de notre document doit impérativement s'insérer entre les deux lignes suivantes :

```
\begin{document}  
\end{document}
```

`\begin` et `\end` délimitent ce que l'on nomme un environnement. Avec  $\text{\LaTeX}$  il existe divers environnements que nous découvrirons au fur et à mesure.

#### Les divers types de documents

La commande `\documentclass{}` sert à indiquer à  $\text{\LaTeX}$  que le document que nous allons rédiger correspond à un certain type, et par conséquent  $\text{\LaTeX}$  va adapter sa mise en page au regard du type indiqué. Dans notre premier document simple donné en guise d'exemple, le type utilisé est `article`.

Principaux types de document usités :

**article** : Article

**book** : Livre

**letter** : Lettre

**report** : Rapport, thèse...

Syntaxe de la commande `documentclass{}` avec l'insertion d'options :

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

---

1. [https://tice.univ-irem.fr/lexique/res/Annexe\\_E\\_-\\_Liste\\_des\\_symboles\\_mathematiques\\_usuels\\_\\_LaTeX\\_.pdf](https://tice.univ-irem.fr/lexique/res/Annexe_E_-_Liste_des_symboles_mathematiques_usuels__LaTeX_.pdf)

# Chapitre 4

## Les packages

### 4.1 Que sont les packages ?

Les packages sont des outils additionnels que l'on insère afin d'implémenter des fonctionnalités supplémentaires aux fonctionnalités de base. IL faut savoir que quand on utilise  $\LaTeX$  nous avons fréquemment recours aux packages. Ces packages sont disponibles sous deux formes : soit le package est déjà présent dans la distribution  $\LaTeX$  et il ne reste plus qu'à s'en servir, soit le package est absent et il sera alors nécessaire de l'installer. Il est à noter que les packages considérés comme incontournables sont installés par défaut dans la distribution  $\LaTeX$  de base.

Pour utiliser un package on va se servir de la commande :

```
\usepackage[option]{type}
```

Cette commande se place juste après la commande `\documentclass{}`.

Exemple (noter l'utilisation du symbole `%` pour insérer des commentaires) :

```
\documentclass{book}

\usepackage[utf8]{inputenc}    % un premier package
\usepackage[T1]{fontenc}      % un deuxième package
\usepackage[french]{babel}    % un troisième package

\begin{document}
    Mon texte...
\end{document}
```

Dans notre exemple, les trois packages utilisés sont : *babel*<sup>1</sup> (Pour spécifier que le texte est écrit en français), *fontenc* et *inputenc* (pour utiliser tous les caractères du clavier). L'option `utf8` du package *inputenc* s'accompagne, pour les linuxiens, de l'installation du paquet *texlive-lang-french* de leur distribution GNU/Linux.

### 4.2 Installer un package non présent dans la distribution $\LaTeX$

Ces packages sont portent souvent les extensions `.ins` et `.sty`. Certains packages peuvent porter une extension autre, et dans ce cas il faudra se reporter au fichier `README` qui les accompagne qui vous guidera dans l'installation.

---

1. Pour en apprendre un peu plus sur ce package, nous renvoyons à la partie consacrée aux packages de ce livre.

**Au format .sty**

Pour l'installation d'un tel package, il suffit de copier le fichier dans le répertoire contenant le fichier `.tex` source. Lors de la compilation seront recherchés les fichiers `.sty`.

**Au format .ins**

L'installation se déroule en deux temps. Il faudra d'abord compiler le fichier `.ins`. Cette compilation va alors générer un fichier `.sty`. Placer ensuite ce fichier comme vu précédemment.

## **Deuxième partie**

# **Pour une utilisation basique de L<sup>A</sup>T<sub>E</sub>X**





## Chapitre 5

# Commençons par la mise en page

### 5.1 La structure d'un document

Nous allons ici apprendre à hiérarchiser notre document selon son type.

Élément de structure	Code $\text{\LaTeX}$
Partie	<code>\part{Titre de la partie}</code>
Chapitre	<code>\chapter{Titre du chapitre}</code>
Section	<code>\section{Titre de la section}</code>
Sous-section	<code>\subsection{Titre de la sous-section}</code>
Paragraphe	<code>\paragraph{Titre du paragraphe}</code>
Sous-paragraphe	<code>\subparagraph{Titre du sous-paragraphe}</code>

TABLE 5.1 – Les différents niveaux de hiérarchisation

A noter que fonction du type de document certains niveaux de hiérarchisation ne sont pas accessibles (Exemple : Le niveau *Chapitre* n'est pas accessible pour le type book).

Exemple <sup>1</sup> :

```
\documentclass{report}

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

\begin{document}
\part{Première partie}
\chapter{Premier chapitre}
\section{Première section}
\subsection{Première sous-section}
\LaTeX{} permet de rédiger des documents dont la mise en page
est réalisée automatiquement en se conformant du mieux
possible à des normes typographiques. Une fonctionnalité
distinctive de \LaTeX{} est son mode mathématique, qui permet
de composer des formules complexes...
\end{document}
```

---

1. Le texte de notre exemple est extrait de la page *Wikipedia* de  $\text{\LaTeX}$ .

## 5.2 Les annexes

Avec le type de document `book` ou `report`, nous avons fréquemment besoin d'ajouter des annexes.  $\LaTeX$  offre la possibilité d'une numérotation différente de celle des chapitres. Entre le contenu du texte (et à fortiori des divers chapitres) et les annexes, on va insérer la commande `\appendix`.

Exemple :

```
\chapter{Premier}
\chapter{Deuxième}
\chapter{Troisième}
\appendix % Les annexes débutent ici
\chapter{Une première annexe}
\chapter{Une seconde annexe}
```

Avec le type `article` la commande `\appendix` influera sur le titre des sections.

## 5.3 Modifier la numérotation

Il est possible de créer des chapitres sans numéro, ni lettre, en insérant une étoile dans la commande : `chapter*{Titre du chapitre}`. Cela fonctionne avec tous les éléments de structure.

Nous pouvons aussi modifier le comportement de la numérotation à l'aide des commandes suivantes :

**\frontmatter** : Positionnée juste après la commande `\begin{document}`, elle permet de numérotter le préambule en chiffres romains.

**\mainmatter** : Entre le préambule et le premier chapitre, elle permet de lancer la numérotation habituelle des pages (en chiffres arabes).

**\backmatter** : Positionnée avant le chapitre épilogue, les index et la bibliographie, elle stoppe la numérotation des chapitres, mais pas celle des pages.

## 5.4 Page de garde

Elle se compose de trois éléments :

**Le titre du document** : `\title{Titre du document}`

**Le nom de l'auteur** : `\author{Nom de l'auteur}`

**Une date** : `\date{La date voulue}`

Ces trois éléments sont insérés avec avant la commande `\begin{document}`, et la commande `\maketitle` placée juste après `\begin{document}` permet de composer la page de garde. Il est bien sûr possible de réaliser des pages de garde bien plus complexes.

## 5.5 Les alignements de texte

Avec  $\LaTeX$  les paragraphes sont naturellement justifiés. Ainsi pour tout autre type d'alignements les environnements suivants sont disponibles :

**L'environnement `flushright`** : alignement du texte à droite.

**L'environnement `center`** : pour centrer le texte.

**L'environnement `flushleft`** : alignement du texte à gauche.

## 5.6 Les sauts de lignes

Sauter deux lignes permet de créer un paragraphe :

```
\begin{document}
Un premier paragraphe

Un second paragraphe
\end{document}
```

Pour aller à la ligne sans créer de nouveau paragraphe on utilise la commande `\newline` ou bien `\\`. Pour faire un saut de page, c'est la commande `\newpage`.

## 5.7 La commande `\documentclass{}`

Description	Valeurs applicables	Valeur par défaut
Format du papier	a4paper, a5paper, letterpaper, b5paper...	letterpaper
Taille de la police principale	10pt, 11pt, 12pt	10pt
Alignement des équations	fleqn (à gauche)	Centrées par défaut
Colonnes	onecolumn, twocolumn	onecolumn
Première page des chapitres	openany, openright	openright
Recto verso	oneside, twoside	article et report : oneside & book : twoside

TABLE 5.2 – Options applicables à la commande `\documentclass{}`

Il est tout à fait possible d'insérer plusieurs options à la fois, il suffit pour cela de les séparer par des virgules.

## 5.8 Les marges

Pour modifier les marges nous allons tout d'abord utiliser la commande `\layout` du package *layout*. Saisir le code suivant et le compiler afin d'obtenir le *layout* du document.

```
\documentclass[a4paper,10pt]{book}

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

% packages additionnels
\usepackage{layout}
```

```
\begin{document}
```

```
\layout
```

```
\end{document}
```

Conjugué à un document saturé de texte nous pouvons visualiser le rendu avec les marges telles qu'elles sont définies. Nous pouvons alors modifier ces dernières à l'aide du package *geometry*. Exemple de modification des marges à la fois en haut, en bas, à droite et à gauche, avec la ligne suivante placée dans la préambule du document :

```
\usepackage[top=2.5cm, bottom=2.5cm, left=2.8cm, right=2.8cm]{geometry}
```

Il est toutefois possible d'intervenir encore plus finement en reprenant divers éléments du *layout* qu'il est alors possible de modifier. Il suffit pour cela de placer dans le préambule une ligne du type :

```
\setlength{nom_de_la_longueur}{longueur_dans_l'unité_voulue}
```

Un exemple :

```
\setlength{\marginparwidth}{2cm}
```

## 5.9 Les interlignes

Les interlignes personnalisés s'obtiennent à l'aide du package *setspace* et des commandes `\onehalfspacing` et `\doublespacing` qui permettent d'obtenir dans le document un interligne respectivement 1,5 et 2 fois plus grand que l'interligne habituel.

Illustrons cela par le code suivant, à l'aide des environnements `onehalfspace` et `doublespace` :

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

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

\usepackage{setspace}

\begin{document}
\section{Interligne simple}
\LaTeX{} est un langage et un système de composition de documents.
Il s'agit d'une collection de macro-commandes...

\section{Interligne intermédiaire (\textit{1,5})}
\begin{onehalfspace}
\LaTeX{} est un langage et un système de composition de documents. Il
s'agit d'une collection de macro-commandes...
\end{onehalfspace}

\section{Interligne double (\textit{2})}
\begin{doublespace}
\LaTeX{} est un langage et un système de composition de documents. Il
```

```
s'agit d'une collection de macro-commandes...
\end{doublespace}
\end{document}
```

## 5.10 Les listes

Il existe les *listes à puce*, les *listes numérotées* et les *listes descriptives*. Chacune de ces listes s'insère dans un environnement spécifique. Plutôt que de partir sur de grandes explications, le principe étant aisé à comprendre, je vous propose de tester le code suivant :

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

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

\begin{document}
\section{Les listes à puce}
\begin{itemize} % L'environnement des listes à puce
\item Puce 1
\item Puce 2
\item Puce 3
% Modifions les puces
\item[0] Puce 4
\item[5] Puce 5
\item[*] Puce 6
\item[\$] Puce 7
\item[\$] Puce 8
\end{itemize}

\section{Les listes numérotées}
\begin{enumerate} % L'environnement des listes numérotées
\item Première
\item Deuxième
\item Troisième
\end{enumerate}

\section{Les listes descriptives}
\begin{description} % L'environnement des listes de description
\item[Listes à puces]: avec l'environnement \textit{itemize}
\item[Listes numérotées]: avec l'environnement \textit{enumerate}
\item[Listes descriptives]: avec l'environnement \textit{description}
\end{description}
\end{document}
```

## 5.11 Les styles

Pour peaufiner un peu plus nos mises en page nous allons voir maintenant les en-têtes et pieds de pages. De base, L<sup>A</sup>T<sub>E</sub>X a été conçu avec trois modèles. Des packages proposent cependant des résultats plus aboutis, mais nous restons pour l'heure sur les modèles de base.

Ces trois modèles sont donc :

**plain** : Ce style insère un numéro de page au milieu du pied de page.

**headings** : Insère le nom du chapitre et le numéro de page en en-tête. Le pied de page demeure vide.

**empty** : En-tête et pied de pages demeurent vides.

Par défaut  $\text{\LaTeX}$  use du style `headings`. Pour modifier le style à une page en particulier, il suffit d'utiliser la commande `\pagestyle{nom_du_style}` au début de la page.

## 5.12 Quelques règles

Les commandes se terminant par des lettres doivent être suivies de la double accolade (`{}`) afin de pouvoir insérer une espace à leur suite (Exemple : `\LaTeX{}`).

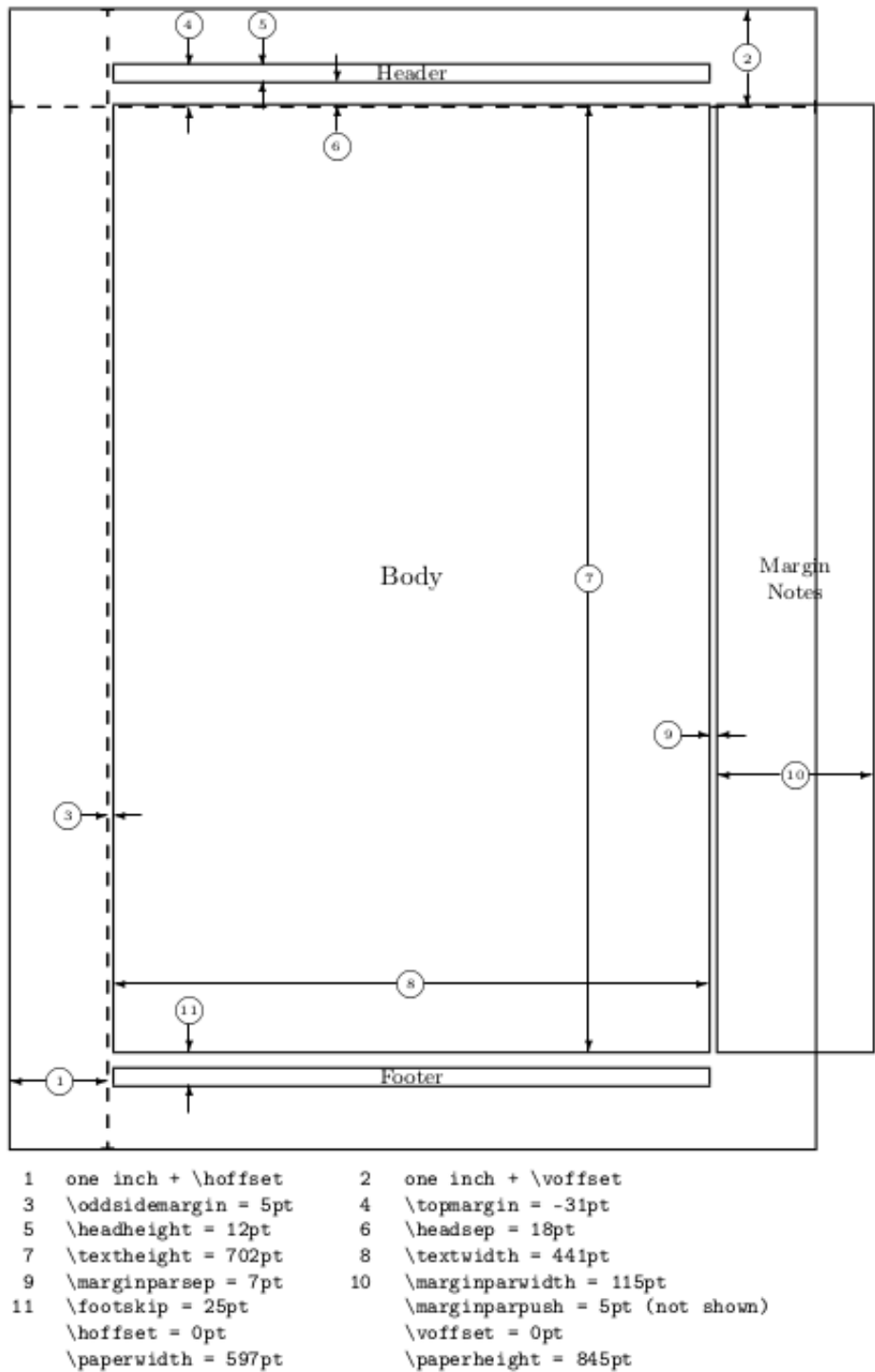


TABLE 5.3 – Le *layout*

## 1 Interligne simple

L<sup>A</sup>T<sub>E</sub>X est un langage et un système de composition de documents. Il s'agit d'une collection de macro-commandes destinées à faciliter l'utilisation du « processeur de texte » T<sub>E</sub>X de Donald Knuth. L<sup>A</sup>T<sub>E</sub>X permet de rédiger des documents dont la mise en page est réalisée automatiquement en se conformant du mieux possible à des normes typographiques. Une fonctionnalité distinctive de L<sup>A</sup>T<sub>E</sub>X est son mode mathématique, qui permet de composer des formules complexes. L<sup>A</sup>T<sub>E</sub>X est particulièrement utilisé dans les domaines techniques et scientifiques pour la production de documents de taille moyenne ou importante...

## 2 Interligne intermédiaire (1,5)

L<sup>A</sup>T<sub>E</sub>X est un langage et un système de composition de documents. Il s'agit d'une collection de macro-commandes destinées à faciliter l'utilisation du « processeur de texte » T<sub>E</sub>X de Donald Knuth. L<sup>A</sup>T<sub>E</sub>X permet de rédiger des documents dont la mise en page est réalisée automatiquement en se conformant du mieux possible à des normes typographiques. Une fonctionnalité distinctive de L<sup>A</sup>T<sub>E</sub>X est son mode mathématique, qui permet de composer des formules complexes. L<sup>A</sup>T<sub>E</sub>X est particulièrement utilisé dans les domaines techniques et scientifiques pour la production de documents de taille moyenne ou importante...

## 3 Interligne double (2)

L<sup>A</sup>T<sub>E</sub>X est un langage et un système de composition de documents. Il s'agit d'une collection de macro-commandes destinées à faciliter l'utilisation du « processeur de texte » T<sub>E</sub>X de Donald Knuth. L<sup>A</sup>T<sub>E</sub>X permet de rédiger des documents dont la mise en page est réalisée automatiquement en se conformant du mieux possible à des normes typographiques. Une fonctionnalité distinctive de L<sup>A</sup>T<sub>E</sub>X est son mode mathématique, qui permet de composer des formules complexes. L<sup>A</sup>T<sub>E</sub>X est particulièrement utilisé dans les domaines techniques et scientifiques pour la production de documents de taille moyenne ou importante...

TABLE 5.4 – Les interlignes



## Chapitre 6

# Les polices

Nous allons ici voir comment modifier la mise en forme d'un texte (en gars, en italique, surligné, etc...), changer la couleur d'un texte, et modifier aussi la police (ponctuellement ou définitivement).

### 6.1 La taille du texte

L<sup>A</sup>T<sub>E</sub>X propose dix commandes permettant d'augmenter ou de diminuer la taille d'un texte. On utilise ces commandes de deux manières :

`\commande{mon texte}`

Ou :

`{\commande mon texte}`

Commande	Taille du texte
<code>\tiny</code>	Minuscule
<code>\scriptsize</code>	Très très petit
<code>\footnotesize</code>	Très petit
<code>\small</code>	Petit
<code>\normalsize</code>	Normal - Défini par défaut
<code>\large</code>	Un peu plus grand que normal
<code>\Large</code>	Grand
<code>\LARGE</code>	Très grand
<code>\huge</code>	Très très grand
<code>\Huge</code>	Énorme !!!

TABLE 6.1 – Les commandes pour les tailles de texte

### 6.2 Graisse, italique, soulignement, etc.

Nous disposons pour réaliser toutes ces opérations sur le texte de trois méthodes :

- `\commande{mon texte}`.
- `{\comande mon texte}`.
- placer le texte dans un environnement.

Mise en forme	Commandes	Rendu	
Normal	<code>{\normalfont texte}</code> <code>\begin{rm}texte\end{rm}</code>	texte texte	
Gras	<code>\textbf{texte}</code> <code>{\bfseries texte}</code> <code>\begin{bf}texte\end{bf}</code>	<b>texte</b> <b>texte</b> <b>texte</b>	
Italique	<code>\textit{texte}</code> <code>{\itshape texte}</code> <code>\begin{it}texte\end{it}</code>	<i>texte</i> <i>texte</i> <i>texte</i>	
Penché	<code>\textsl{texte}</code> <code>{\slshape texte}</code> <code>\begin{sl}texte\end{sl}</code>	<i>texte</i> <i>texte</i> <i>texte</i>	
Machine à écrire	<code>\texttt{texte}</code> <code>{\ttfamily texte}</code> <code>\begin{tt}texte\end{tt}</code>	texte texte texte	
Petites capitales	<code>\textsc{texte}</code> <code>\bsc{texte}</code> <code>{\scshape texte}</code> <code>\begin{sc}texte\end{sc}</code>	TEXTE TEXTE TEXTE TEXTE	
Exposant	<code>texte\textsuperscript{texte}</code>	texte <sup>texte</sup>	
Encadré	<code>\fbox{texte}</code>	<table><tr><td>texte</td></tr></table>	texte
texte			
Soulignement <i>package soul</i>	<code>\ul{texte}</code>	<u>texte</u>	
Soulignement <i>package ulem</i>	<code>\uuline{texte}</code>	<u>texte</u>	
Soulignement <i>package ulem</i>	<code>\uwave{texte}</code>	<u>texte</u>	
Barrer <i>package soul</i>	<code>\st{texte}</code>	<del>texte</del>	

TABLE 6.2 – Les diverses mises en forme du texte

### 6.3 La commande `\emph{}`

Cette commande a un fonctionnement à part puisqu'elle permet d'indiquer à  $\text{\LaTeX}$  de mettre le texte en évidence (en emphase). C'est  $\text{\LaTeX}$  qui se chargera de choisir la meilleure manière de mettre le texte en valeur. Il est d'ailleurs préférable d'utiliser `\emph{}` à l'italique.

Nous pouvons cependant définir comment la commande `\emph{}` va mettre en évidence la portion de texte voulue, en plaçant dans le préambule la ligne suivante :

```
\renewcommand{\emph}{fonction_liée_à_la_commande}
```

Exemple pour une mise en évidence avec le style « machine à écrire » :

```
\renewcommand{\emph}{\texttt}
```

## 6.4 Mise en couleur

### Les couleurs par défaut

Elles sont au nombre de huit et l'usage de la couleur nécessite l'emploi du package *color*. Ces couleurs sont *black*, *white*, *red*, *green*, *blue*, *yellow*, *magenta* et *cyan*.

La commande est la suivante :

```
\textcolor{couleur}{mon texte}
```

### Création de nouvelles couleurs

Il est possible de créer de nouvelles couleurs avec la commande `\definecolor` qui se place dans le préambule, soit à partir de niveaux de gris ou d'un mélange de trois couleurs (rouge, vert et bleu). Ces nouvelles couleurs se verront attribuer un nom et elles pourront s'utiliser grâce à la commande `\textcolor`.

**Par niveaux de gris** : Le niveau de gris se trouve sur une échelle située entre 0 (le noir) et le 1 (le blanc). Choisir un niveau de gris va consister à prendre un nombre à deux décimales situé entre 0 et 1, que l'on va appliquer sur une des huit couleurs par défaut.

```
\definecolor{nomchoisi}{une des 8 couleurs}{niveau de gris}
```

**Par mélange des trois couleurs** : Il suffit de choisir tour à tour l'intensité de rouge, de vert et de bleu (entre 0 et 1).

```
\definecolor{nomchoisi}{rgb}{taux de rouge, de ver, de bleu}
```

## 6.5 Les packs de polices

Afin de pouvoir changer les polices de caractères des packs de polices ont été créés, tout en ayant dans l'idée de conserver une cohérence typographique à l'intégralité du texte. Un pack cohérent va comprendre quatre polices cohérentes :

- des caractères avec empattements ;
- des caractères sans empattements ;
- des caractères façon machine à écrire ;
- des caractères servant à écrire des formules mathématiques.

Par défaut  $\text{\LaTeX}$  fournit la police *Computer Modern*. Pour faire appel à autre pack c'est le même procédé que de faire appel à un package.

```
\usepackage{nom_du_pack}
```

Des modifications ponctuelles de police peuvent aussi être introduites grâce à la commande :

```
{\fontfamily{code_police}\selectfont mon texte}
```

Quelques packs : *bookman*, *charter*, *newcent*, *lmodern*, *mathpazo*, *mathptmx*, etc.

Code de la police	rendu
bch	Mon texte en Charter
cmr	Mon texte en Computer Modern
lmr	Mon texte en Latin Modern Roman
lmss	Mon texte en Latin Modern Sans Empattement
lmmsq	Mon texte en Latin Modern Sans Emp. Exp.
lmtt	Mon texte en Latin Modern Typewriter
pbk	Mon texte en Bookman

TABLE 6.3 – Quelques exemples de polices

## Chapitre 7

# Citations, notes et références

### 7.1 Les citations

Pour cela deux environnements sont proposés : `quote` et `quotation`.

`quote` sera utilisé pour une citation simple :

```
\begin{quote}
La vie c'est comme une bicyclette, il faut avancer pour ne pas
perdre l'équilibre.
\end{quote}
```

La vie c'est comme une bicyclette, il faut avancer pour ne pas perdre l'équilibre<sup>1</sup>.

`quotation` est fait pour de plus gros volumes de texte, et il introduit une tabulation comme pour un début de paragraphe.

```
\begin{quotation}
Ainsi l'homme est si malheureux qu'il s'ennuierait même sans aucune cause
d'ennui par l'état propre de sa complexion. Et il est si vain qu'étant
plein de mille causes essentielles d'ennui, la moindre chose comme un
billard et une balle qu'il pousse suffisent pour le divertir.
\end{quotation}
```

Ainsi l'homme est si malheureux qu'il s'ennuierait même sans aucune cause d'ennui par l'état propre de sa complexion. Et il est si vain qu'étant plein de mille causes essentielles d'ennui, la moindre chose comme un billard et une balle qu'il pousse suffisent pour le divertir<sup>2</sup>.

### 7.2 Les citations de code

#### La commande `\verb`

Cette commande permet d'insérer du code dans un paragraphe :

La commande `\verb|\verb|` pour insérer du texte.

---

1. Albert EINSTEIN

2. Blaise PASCAL

Ce qui donne : La commande `\verb` pour insérer du texte.

Le caractère `|` peut être remplacé par `(` ou `[` :

```
\verb[mon code[
\verb(mon code(
```

## L'environnement `verbatim`

Cet environnement s'accompagne du package *verbatim*. Il permet d'accompagner de plus gros volumes de code, que l'on écrit généralement sur plusieurs lignes. Un point de vigilance à avoir : Les tabulations sont remplacées par des espaces.

```
\begin{verbatim}
# Une boucle Python
i = 0
while i < 5:
    print(i + " X 5 = " + (i*5))
    i += 1
\end{verbatim}
```

L'environnement `verbatimtab`, que l'on trouve avec le package *moreverb*. La syntaxe est la suivante :

```
\begin{verbatimtab}[nombre_d'espaces_par_tabulation]
...le code...
\end{verbatimtab}
```

## L'environnement `lstlisting`

Cet environnement permet de mettre en forme du code avec de nombreuses et de façon colorée. Il est nécessaire d'utiliser le package *listings* pour cela, puis de faire appel à la commande `lstset` dans l'en-tête du document. Cette commande possède un grand nombre d'arguments paramétrables.

```
\lstset{
  language=nom_du_langage,      % choix du langage
  basicstyle=\footnotesize,     % taille de la police du code
  numbers=left,                  % numéro des lignes placé à gauche
  numbers=right,                 % numéro des lignes placé à droite
  numberstyle=\normalsize,      % taille de la police des numéros
  numbersep=7pt,                 % distance entre code et numérotation
  backgroundcolor=\color{white}, % couleur de fond (utilisation du package
                                % 'color' possible)
}
```

Les langages compatibles avec la commande sont constamment mis à jour sur la page *Wikibooks* consacrée au package *listings*<sup>3</sup>.

Le code à afficher s'insère dans l'environnement `lstlisting` comme ci-dessous avec un bout de code rédigé en langage *Python* :

3. [https://en.wikibooks.org/wiki/LaTeX/Source\\_Code\\_Listings#Supported\\_languages](https://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings#Supported_languages)

```

\usepackage{listings}

\lstset{
  language=Python,
  basicstyle=\footnotesize,
  numbers=left,
  numberstyle=\normalsize,
  numbersep=4pt,
}

\begin{document}

\begin{lstlisting}
# Une boucle while
i = 0
while i < 5:
    print(i + " X 5 = " + (i*5))
    i += 1
\end{lstlisting}

\end{document}

```

Ce qui donne à la compilation :

```

1  # Une boucle while
2  i = 0
3  while i < 5:
4      print(i + " X 5 = " + (i*5))
5      i += 1

```

### 7.3 Insérer une *url*

A l'aide du package *url* et de la commande éponyme :

```

\url{https://l'adresse_internet_à_faire_figurer}

```

### 7.4 Texte encadré et l'environnement minipage

L'environnement `minipage` et la commande `\fbox` permettent d'encadrer du texte et de le mettre en valeur. Mais attention à les utiliser avec sobriété.

#### Texte encadré avec `\fbox`

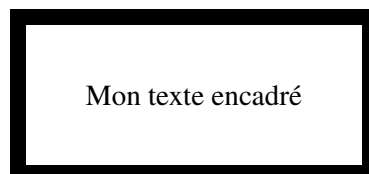
Avec `\fbox` il est possible de paramétrer diverses choses. Nous allons ici en utiliser deux : l'écart entre le texte et la bordure et l'épaisseur de cette dernière.

```

% Commande permettant de définir l'écart
\setlength{\fboxsep}{8mm}
% Commande permettant de définir l'épaisseur du trait
\setlength{\fboxrule}{2mm}
\fbox{Mon texte encadré}

```

Ce qui nous donne :



## L'environnement minipage

Une *minipage* est un encart de texte de largeur choisie, en quelque sorte une nouvelle page dans votre page. À l'intérieur de cet encart de texte, vous pourrez disposer et utiliser des environnements comme si cette *minipage* était un document à part entière. L'environnement *minipage* est dépendant de deux paramètres : la largeur et l'alignement vertical.

Illustrons cela :

```
Voici mon texte dans lequel on insère une minipage
\fbbox{ % pour encadrer notre minipage
\begin{minipage}[c]{5cm}
au sein de laquelle je peux utiliser d'autres environnements:
\begin{center}
\LaTeX{} c'est \textbf{bien} !
\end{center}
\end{minipage}
}
```

pour illustrer.

Voici mon texte dans lequel on insère une minipage  
illustrer.

au sein de laquelle je peux utiliser d'autres environnements :  $\LaTeX$ c'est <b>bien</b> !	pour
--	------

## 7.5 Notes de bas de page

### La commande \footnote

À l'endroit où l'on souhaite insérer la note de bas de page :

```
\footnote{Mon texte de bas de page}
```

### La commande \footnotemark

Tout d'abord, marquer tous les éléments concernés par des notes de bas de page à l'aide d'un numéro, puis on insère le texte de bas de page correspondant au numéro. Deux compilations seront alors nécessaires.

Exemple de code :



Voici mon code `\footnotemarck[1]` qui permet d'insérer `\footnotemark[2]` des notes de bas de page `\footnotemark[3]`.

```
% A insérer au niveau de la page où l'on souhaite voir figurer les
% notes de bas de page:
\footnotetext[1]{Ma première note}
\footnotetext[2]{Ma deuxième note}
\footnotetext[3]{Ma troisième note}
```

## 7.6 Les références internes

L<sup>A</sup>T<sub>E</sub>X permet d'écrire des références internes de façon simple à l'aide de trois commandes. La commande `\label{nom_choisi}` sert à marquer un endroit, et les commandes `\ref{nom_choisi}` et `\pageref{nom_choisi}` permettent d'appeler le numéro de page ou la référence de l'élément marqué avec la commande `\label{}`.



# Chapitre 8

## Les figures

### 8.1 Choix des fichiers

Avant d'entrer dans le vif du sujet voyons d'abord le type de compilation en fonction du fichier image utilisé.

**Utilisation de fichiers *.eps*** : obligation de compiler en *Post-Script* avant d'effectuer une conversion en *PDF*. Il sera impossible d'utiliser des fichiers *PNG*, *BMP*, *JPEG* ou *GIF* en parallèle.

**Utilisation des fichiers *PNG*, *BMP*, *JPEG* ou *GIF*** : obligation de compiler directement en *PDF*.

Il est bien sûr tout à fait possible de convertir un fichier image d'un format à une autre.

### 8.2 L'insertion d'image

Tout d'abord il est nécessaire de faire appel au package *graphicx*.

La commande pour insérer simplement une figure :

```
\includegraphics{chemin\du\fichier\image.xxx}
```

### 8.3 Taille d'une image

Il existe plusieurs méthodes pour indiquer à  $\text{\LaTeX}$  la taille de l'image à insérer :

- faire en sorte que l'image ait une largeur donnée, la hauteur sera automatiquement adaptée ;
- avec une hauteur donnée, c'est la largeur qui sera adaptée ;
- fixer à la fois la hauteur et la largeur (risque de déformation de l'image) ;
- choisir un coefficient de proportionnalité et l'image sera « retaillée » de façon cohérente.

Les quatre commandes correspondantes :

```
\includegraphics[width=200]{fichier_image.xxx}    % largeur
\includegraphics[height=200]{fichier_image.xxx}    % hauteur
\includegraphics[height=200, width=100]{fichier_image.xxx} % fixation
                                                    % hauteur et largeur
\includegraphics[scale=1.5]{fichier_image.xxx}    % Avec coefficient
```



FIGURE 8.1 – Illustration avec un angle de 45°

## 8.4 Rotation d'une image

On va pour cela utiliser la variable `angle` :

```
\includegraphics[angle=45]{fichier_image.xxx}
```

## 8.5 Intégration d'une image dans un paragraphe

Nous allons intégrer une image dans un texte de façon à ce que le texte contourne la figure à l'aide du package *wrapfig* (afin d'utiliser l'environnement *wrapfigure*. Noter cependant que nous n'avons pas la pleine maîtrise du résultat car c'est  $\text{\LaTeX}$  qui redéfinit les arrangements. Avec l'environnement *wrapfigure* il existe diverses variables :

- le nombre de lignes nécessaires à la bonne intégration de l'image ;
- la taille du dépassement autorisé dans la marge (0 pour garder des documents « propres ») ;
- la largeur de l'image ;
- l'alignement de l'image.

La syntaxe est alors la suivante :

```
\begin{wrapfigure}[nbre_de_lignes]{placement}{largeur_image_en_cm}
\includegraphics[width=largeur_en_cm]{fichier_image.xxx}
\end{wrapfigure}
Mon paragraphe sans saut de ligne après la commande \end{wrapfigure}...
```

Le placement se définit à l'aide des lettres suivantes :

- l** : image à gauche ;
- r** : image à droite ;
- o** : image à l'extérieur (à droite avec une page impaire, à gauche avec une page paire) ;
- i** : image à l'intérieur (à gauche pour une page impaire, à droite pour une page paire).

Exemple de code pour intégrer une image de 2.5 x 2.3 cm de large, qui occupe 6 lignes et située à droite du paragraphe :

```
\begin{wrapfigure}[6]{r}{2.5cm}
\includegraphics[width=2.3cm]{IMG/tux.png}
\end{wrapfigure}
Le paragraphe qui intégrera l'image...
```

Ce qui pourrait donner :

La mascotte de *Linux* est employée par de nombreuses applications en tant que logo, mais a aussi été modifiée par de nombreux particuliers et développeurs. On découvrira ainsi, parmi de nombreux autres, *Tux* en Sherlock Holmes, en Dracula, ou encore en Charlie Chaplin ou habillé avec des maillots de football. De nombreux logiciels libres, comme *TuxGuitar*, *Tux Paint* ou *Tux Racer* reprennent *Tux* dans leur intitulé et/ou dans leur logo. Le dessin du personnage a été choisi à l'issue d'un concours organisé en 1996 remporté par Larry EWING. Il utilisa *GIMP*, le logiciel de traitement d'image phare sur *GNU/Linux*. Il s'agit d'un personnage fictif représentant très approximativement un manchot pygmée dont l'idée a été suggérée par Alan COX<sup>1</sup> puis affinée par Linus TORVALDS, le créateur du noyau *Linux*. Certains déclarèrent de prime abord que cette mascotte était inappropriée car elle n'évoquait guère la puissance. Linus TORVALDS répondit que nulle personne poursuivie par un manchot pygmée, qui court vite et dispose d'un bec très dur, ne penserait cela. Les contradictions s'éteignirent...<sup>1</sup>



---

1. Extrait de la page *Wikipedia* consacrée à *Tux* : <https://fr.wikipedia.org/wiki/Tux>.



# Chapitre 9

## Les flottants

$\LaTeX$  propose une façon optimisée pour placer des images et des figures, en spécifiant exactement leur place. C’est d’ailleurs une des fonctions phares de  $\LaTeX$ . Nous allons pour cela utiliser des environnements dits « flottants ». En plus du positionnement ces environnements offrent la possibilité d’insérer des légendes à ces figures.

Voyons donc cela avec l’environnement `figure`.

### 9.1 La création d’un flottant

Très simplement, à l’aide de la commande `\includegraphics{}` et l’environnement `figure` :

```
\begin{figure}
\begin{center} % Pour centrer l'image
\includegraphics{fichier_image.xxx}
\end{center}
\end{figure}
```

### 9.2 Le placement

Il est possible de spécifier le type de placement à l’aide d’options insérées dans l’environnement `figure` :

```
\begin{figure}[option]
```

Les options de placement :

**t** : en haut de page ;

**b** : en bas de page ;

**p** : sur une page ne comportant que des flottants ;

**h** : placer de préférence dans la zone où l’on insère l’environnement ;

**H** : placer de façon insistante dans la zone où l’on insère l’environnement ;

**!b** : le « ! » pour insister encore plus ;

**bt** : de préférence en bas, mais en haut si en bas ce n’est pas possible.

### 9.3 Le placement par défaut

L<sup>A</sup>T<sub>E</sub>X place les flottants par défaut suivant les options prévues, mais il est possible de modifier ce comportement à l'aide la de commande suivante, fournie par le package float :

```
% Exemple avec un flottant de type 'figure'  
\floatplacement{figure}{t}
```

### 9.4 Les légendes

Pour légender des flottants on utilise la commande `\caption{}`. Elle s'utilise à la suite de l'environnement `center` et précède une éventuelle commande `\label{}`.

```
\begin{figure}  
  \begin{center}  
    \includegraphics{fichier_image.xxx}  
  \end{center}  
  \caption{Texte de la légende}  
  \label{référence}  
\end{figure}
```

### 9.5 Les sauts de page avec le flottants

`\clearpage` : saut de page en produisant une page remplie par tous les flottants non traités ;

`\cleardoublepage` : même effet, si ce n'est que la nouvelle page sera nécessairement une page impaire.



# Chapitre 10

## Les tableaux

Il faut savoir que  $\text{\LaTeX}$  considère les tableaux comme des objets flottants. Les tableaux font l'objet d'une documentation extrêmement fournie, à l'instar de la documentation des notations mathématiques. Dans ce chapitre nous allons juste nous contenter d'apprendre à :

- composer des tableaux simples ;
- fusionner des cellules ;
- paramétrer le placement allié à quelques détails de mise en page.

### 10.1 Structure type d'un tableau

#### Tableau sans bordure

Nous allons utiliser l'environnement `tabular`. Dans un premier temps il faut décider de l'alignement des cellules dans chaque colonne (options) :

**r** : à droite ;

**l** : à gauche ;

**c** : centré.

Exemple de squelette de tableau à deux colonnes :

```
\begin{tabular}{cc}
...code pour mon tableau...
\end{tabular}
```

Nous entrons ensuite ligne par ligne le contenu des cellules, séparées par le caractère `&`. Chaque ligne se termine par `\\` pour indiquer que nous changeons de ligne.

Exemple de tableau à deux colonnes et deux lignes :

```
\begin{tabular}{cc}
lig. 1 col. 1 & lig. 1 col. 2 \\
lig. 2 col. 1 & lig. 2 col. 2 \\
\end{tabular}
```

Ce qui donne :

lig. 1 col. 1	lig. 1 col. 2
lig. 2 col. 1	lig. 2 col. 2

## Tableau avec bordures

Pour obtenir une ligne horizontale on va utiliser la commande `\hline`. Pour marquer les colonnes, on va utiliser « | » lors de la spécification des alignements.

Exemple d'un tableau à deux colonnes et deux lignes :

```
\begin{tabular}{|c|c|}
\hline
lig. 1 col. 1 & lig. 1 col. 2 \\
\hline
lig. 2 col. 1 & lig. 2 col. 2 \\
\hline
\end{tabular}
```

Ce qui donne :

lig. 1 col. 1	lig. 1 col. 2
lig. 2 col. 1	lig. 2 col. 2

## 10.2 Fusion des cellules

### Fusion de colonnes

La commande est la suivante :

```
\multicolumn{nbre_colonnes_fusionnées}{alignements}{Texte}
```

La difficulté réside dans le choix d'un nouvel alignement pour la cellule fusionnée.

Illustrons cela :

```
\begin{tabular}{|c|c|c|c|c|c|c|c|c|c|c|c|}
\hline
Multiplié par & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & \\
\hline
1 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & \\
\hline
2 & 2 & 4 & 6 & 8 & 10 & 12 & 14 & 16 & 18 & 20 & \\
\hline
3 & 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 & 27 & 30 & \\
\hline
4 & 4 & 8 & 12 & 16 & 20 & 24 & 28 & 32 & 36 & 40 & \\
\hline
5 & 5 & 10 & 15 & 20 & 25 & 30 & 35 & 40 & 45 & 50 & \\
\hline
6 & 6 & 12 & 18 & 24 & 30 & 36 & 42 & 48 & 54 & 60 & \\
\hline
7 & 7 & 14 & 21 & 28 & 35 & 42 & 49 & 56 & 63 & 70 & \\
\hline
8 & 8 & 16 & 24 & 32 & 40 & 48 & 56 & 64 & 72 & 80 & \\
\hline
\end{tabular}
```

```

9 & 9 & 18 & 27 & 36 & 45 & 54 & 63 & 72 & 81 & 90 \\
\hline
10 & 10 & 20 & 30 & 40 & 50 & 60 & 70 & 80 & 90 & 100 \\
\hline
\end{tabular}

```

Multiplié par	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

TABLE 10.1 – Table de multiplication

## Fusion de lignes

Pour la fusion de lignes c'est la commande `\multirow` contenue dans le package du même nom.

```

\multirow{nbre de lignes fusionnées}{taille colonne en cm}{texte}
\multirow{nbre de lignes fusionnées}*{texte}

```

Observez bien le code suivant et le résultat qui s'en suit :

```

\begin{tabular}{|l|c|c|c|c|}
\hline
1 & \multicolumn{2}{c|}{2} & 3 & 4 \\
\hline
\multicolumn{2}{|l|}{5} & 6 & 7 & 8 \\
\hline
9 & 10 & \multicolumn{3}{c|}{11} \\
\hline
\multirow{2}{1cm}{12} & 13 & 14 & 15 & 16 \\
\cline{2-5}
& 17 & 18 & 19 & 20 \\
\hline
21 & 22 & \multirow{2}{23} & 24 & 25 \\
\cline{1-2} \cline{4-5}
26 & 27 & & 28 & 29 \\
\hline
\end{tabular}

```

1	2	3	4
5	6	7	8
9	10	11	
12	13	14	15
	17	18	19
21	22	23	24
26	27	28	29

TABLE 10.2 – Fusion de lignes et de colonnes

## 10.3 Autres paramètres applicables à un tableau

### Colonne de largeur paramétrée

`p{largeur de la colonne en centimètres}`

Cette option n’a aucune influence sur l’alignement du texte au sein des cellules.

Un exemple :

```
\begin{tabular}{|p{1cm}|p{2cm}|p{3cm}|p{4cm}|}
\hline
1cm & 2cm & 3cm & 4cm \\
\hline
\end{tabular}
```

Ce qui donne :

1cm	2cm	3cm	4cm
-----	-----	-----	-----

TABLE 10.3 – Cellules de longueur définie

### Créer une *slashbox*

Le package *slashbox* permet d’utiliser la commande :

```
\backslashbox{Texte dessous}{Texte dessus}
```

Cette commande sert à scinder en deux parties triangulaires de même aire une cellule initialement rectangulaire.

Démonstration :

```
\begin{tabular}{|c|p{1cm}|p{2cm}|}
\hline
\backslashbox{Debian}{Ubuntu} & 1cm & 2cm \\
\hline
\end{tabular}
```

Résultat :

Ubuntu	1cm	2cm
Debian		

TABLE 10.4 – Avec le package *slashbox*

## Changer les séparateurs

Cela est possible à l'aide des commandes `!\{séparateur\}` ou `@\{séparateur\}`, contenues dans le package *array*. A noter que ce package contient beaucoup de commandes utiles à la création des tableaux. La différence entre ces deux commandes réside dans le fait de pouvoir insérer une espace avant et après le séparateur.

```
\begin{tabular}{|c !{\#} c @{\#}c|}
\hline
3 + 1 & = & 4 \\
\hline
\end{tabular}
```

Ce qui a pour résultat :

3 + 1 # =#4
-------------

TABLE 10.5 – De nouveaux séparateurs

## 10.4 Des commandes et des environnements dans un tableau

Démonstration avec la table de multiplication :

```
\begin{tabular}{|>\begin{bf} c <\end{bf}|c|c|c|c|c|c|c|c|c|c|}
\hline
Multiplié par & \begin{bf}1\end{bf} & \begin{bf}2\end{bf} &
\begin{bf}3\end{bf} & \begin{bf}4\end{bf} & \begin{bf}5\end{bf} &
\begin{bf}6\end{bf} & \begin{bf}7\end{bf} & \begin{bf}8\end{bf} &
\begin{bf}9\end{bf} & \begin{bf}10\end{bf} \\
\hline
1 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
\hline
2 & 2 & 4 & 6 & 8 & 10 & 12 & 14 & 16 & 18 & 20 \\
\hline
3 & 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 & 27 & 30 \\
\hline
4 & 4 & 8 & 12 & 16 & 20 & 24 & 28 & 32 & 36 & 40 \\
\hline
5 & 5 & 10 & 15 & 20 & 25 & 30 & 35 & 40 & 45 & 50 \\
\hline
6 & 6 & 12 & 18 & 24 & 30 & 36 & 42 & 48 & 54 & 60 \\
\hline
7 & 7 & 14 & 21 & 28 & 35 & 42 & 49 & 56 & 63 & 70 \\
\hline
```

```

8 & 8 & 16 & 24 & 32 & 40 & 48 & 56 & 64 & 72 & 80 \\
\hline
9 & 9 & 18 & 27 & 36 & 45 & 54 & 63 & 72 & 81 & 90 \\
\hline
10 & 10 & 20 & 30 & 40 & 50 & 60 & 70 & 80 & 90 & 100 \\
\hline

\end{tabular}

```

Affichage :

<b>Multiplié par</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>
<b>1</b>	1	2	3	4	5	6	7	8	9	10
<b>2</b>	2	4	6	8	10	12	14	16	18	20
<b>3</b>	3	6	9	12	15	18	21	24	27	30
<b>4</b>	4	8	12	16	20	24	28	32	36	40
<b>5</b>	5	10	15	20	25	30	35	40	45	50
<b>6</b>	6	12	18	24	30	36	42	48	54	60
<b>7</b>	7	14	21	28	35	42	49	56	63	70
<b>8</b>	8	16	24	32	40	48	56	64	72	80
<b>9</b>	9	18	27	36	45	54	63	72	81	90
<b>10</b>	10	20	30	40	50	60	70	80	90	100

TABLE 10.6 – Mise en gras de certaines parties

## Colorer des cellules

Pour cela deux packages sont nécessaires : *color* et *colortbl*. Les commandes sont les suivantes :

```

\columncolor{couleur} % Pour colorer les colonnes
\rowcolor{couleur} % Pour colorer des lignes
\cellcolor{couleur} % Pour colorer les cellules

```

Exemple d'un tableau avec la première ligne et la première colonne sur fond jaune :

```

\begin{tabular}{>{\begin{bf} \columncolor{yellow}} c <{\end{bf}}ccccccccc}
\hline
\rowcolor{yellow}Multiplié par & \begin{bf}1\end{bf} & \begin{bf}2\end{bf} &
  \begin{bf}3\end{bf} & \begin{bf}4\end{bf} & \begin{bf}5\end{bf} &
  \begin{bf}6\end{bf} & \begin{bf}7\end{bf} & \begin{bf}8\end{bf} &
  \begin{bf}9\end{bf} & \begin{bf}10\end{bf} \\
\hline
1 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
\hline
2 & 2 & 4 & 6 & 8 & 10 & 12 & 14 & 16 & 18 & 20 \\
\hline
3 & 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 & 27 & 30 \\
\hline
4 & 4 & 8 & 12 & 16 & 20 & 24 & 28 & 32 & 36 & 40 \\
\hline
5 & 5 & 10 & 15 & 20 & 25 & 30 & 35 & 40 & 45 & 50

```

```

\hline
6 & 6 & 12 & 18 & 24 & 30 & 36 & 42 & 48 & 54 & 60 \\
\hline
7 & 7 & 14 & 21 & 28 & 35 & 42 & 49 & 56 & 63 & 70 \\
\hline
8 & 8 & 16 & 24 & 32 & 40 & 48 & 56 & 64 & 72 & 80 \\
\hline
9 & 9 & 18 & 27 & 36 & 45 & 54 & 63 & 72 & 81 & 90 \\
\hline
10 & 10 & 20 & 30 & 40 & 50 & 60 & 70 & 80 & 90 & 100 \\
\hline
\end{tabular}

```

Affichage :

Multiplié par	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

TABLE 10.7 – Tableau avec des couleurs

### L'environnement flottant `table`

Cela revient à insérer l'environnement `tabular` dans un environnement flottant. Celui-ci se nomme `table` et est similaire en tout point à l'environnement `figure`, avec l'utilisation de `caption`, `label`, `center`, etc.





# Chapitre 11

## Sommaire et index

### 11.1 Table des matières

#### Table des matières simple

Il suffit de placer la commande `\tableofcontents` là où vous souhaitez insérer la table des matières. La table est conçue telle que le prévoient les paramètres par défaut de  $\text{\LaTeX}$ . cela nécessite deux compilations : la première permet à  $\text{\LaTeX}$  de comprendre la structure du document et de lister les titres, et la seconde permet la mise en forme de la table avec les numéros de pages.

#### Paramétrage d'une table des matières

**Appeler la table « sommaire »** avec la commande suivante à insérer dans le corps du document, juste avant la commande `\tableofcontents` :

```
\renewcommand{\contentsname}{Sommaire}
```

**Raccourcir une ligne** Lors de la création d'un élément de structure il est possible de définir deux titres : l'un pour le document, l'autre pour la table des matières. On code cela de la manière suivante :

```
\section{titre table des matières}{titre dans le document}
```

**Quel niveau hiérarchique dans notre table des matières ?** Il est possible de définir le niveau hiérarchique de la table des matières (inclusion des sous-sections ou non, des sections, etc.). La commande à placer en préambule est alors la suivante :

```
\setcounter{tocdepth}{Choix du niveau (nombre)}
```

Concernant le niveau hiérarchique choisi, voir le tableau suivant :

A noter qu'avec la classe `book`, par défaut, les titres des paragraphes ne sont pas inclus.

### 11.2 Tables de figures et tableaux

Ces tables suivent à peu près les mêmes mécanismes que ceux de la table des matières. On utilise respectivement les commandes `\listoffigures` et `\listoftables`. Apparaissent dans ces tables, soit un titre spécialement conçu pour la table, soit la légende contenue dans la commande `\caption{}`. Voici les deux usages de la commande `\caption{}` :

!h

Niveau hiérarchique (inclus)	Valeur
Partie	-1
Chapitre	0
Section	1
Sous-section	2
Sous-sous-section	3
Paragraphe	4
Sous-paragraphe	5

TABLE 11.1 – Tableau des niveaux hiérarchiques de la table des matières

```
\caption{légende de la figure ou de la table}
\caption[légende courte]{légende de la figure ou de la table}
```

Là encore, deux compilations sont nécessaires.

### 11.3 Les index

La création d'un index nécessite le package *makeindex* et l'insertion de la commande `\makeindex` dans le préambule, puis on insère la commande `\printindex` à l'endroit où l'on décide de l'afficher. Les entrées d'index seront marqués à l'aide de la commande `\index{argument}` collée au mot que l'on souhaite indexé. L'utilisation d'index nécessite trois compilations. Dans l'index les *arguments* figureront dans l'ordre alphabétique.

Nous venons de voir la méthode la plus simple pour la création d'index, mais créer de véritables index s'avère beaucoup plus complexe.

Commande permettant de d'insérer un *argument* d'index avec une référence croisée :

```
\index{argument|see{référence croisée}}
```

## **Troisième partie**

# **Vers une connaissance plus approfondie de L<sup>A</sup>T<sub>E</sub>X**



# Chapitre 12

## La mise en page

### 12.1 Des en-têtes et pieds de pages personnalisés

Nous allons réaliser cela à l'aide de la commande `\pagestyle{fancy}` fournie par le package *fancyhdr*. Cette commande découpe chaque en-tête et pied de page en trois morceaux, de sorte que nous devons personnaliser six commandes différentes pour arriver à créer une mise en page cohérente des en-têtes et pieds de pages.

La commande	Son résultat
<code>\lhead{..}</code>	Gère le coin gauche de l'en-tête
<code>\chead{..}</code>	Gère le centre de l'en-tête
<code>\rhead{..}</code>	Gère le coin droit de l'en-tête
<code>\lfoot{..}</code>	Gère le coin gauche du pied de page
<code>\cfoot{..}</code>	Gère le centre du pied de page
<code>\rfoot{..}</code>	Gère le coin droit du pied de page
<code>\leftmark</code>	Nom du chapitre
<code>\rightmark</code>	Nom de la section
<code>\thepage</code>	Numéro de page
<code>\thechapter</code>	Numéro du chapitre
<code>\thesection</code>	Numéro de la section courante
<code>\renewcommand{\headrulewidth}{épaisseur}</code>	Épaisseur de la ligne entre l'en-tête et le corps du texte
<code>\renewcommand{\footrulewidth}{épaisseur}</code>	Épaisseur de la ligne entre le pied de page et le corps du texte
<code>\renewcommand{\headsep}{longueur}</code>	Espacement entre l'en-tête et le corps du document

TABLE 12.1 – Les commandes disponibles avec le package *fancyhdr*

Voici, à titre d'exemple, le style utilisé pour ce document (les lignes suivantes sont placés juste après la première ligne avec la commande `\part{}`) :

```
% Personnalisation en-têtes et pieds de pages
\pagestyle{fancy}
\renewcommand{\headrulewidth}{1pt}
\renewcommand{\headsep}{15pt}
\setlength{\headheight}{1cm}
\lhead{\textcolor{magenta}{Chapitre} \thechapter}
\rhead{\thepage}
\cfoot{}
```

## 12.2 Les unités de longueur

Le plus simple étant de préciser tout cela à l'aide du tableau ci-dessous :

Unité	Exprimée avec $\text{\LaTeX}$	Valeur en millimètres
Point	pt	0,35 mm
Millimètre	mm	1,00 mm
Pica	pc	4,21 mm
Cicéro	cc	4,53 mm
Centimètre	cm	10,00 mm
Inch	in	25.4 mm

TABLE 12.2 – Les unités de longueur

A titre d'indication une feuille A4 mesure 210 millimètres de large et 297 de haut.

## 12.3 Les espaces

Ceux-ci peuvent se définir (entre deux mots, entre deux parties de texte) à l'aides des commandes suivantes :

Commande	Résultat
<code>\hspace{longueur}</code>	Insère un espace horizontal de la longueur choisie.
<code>\hspace*{longueur}</code>	Idem, saut que l'espace n'est pas inséré s'il y a un retour à la ligne entre les deux mots concernés par l'espacement.
<code>\vspace{hauteur}</code>	Insère un espace vertical de la longueur choisie.
<code>\vspace*{hauteur}</code>	Ici, l'espace n'est pas inséré s'il y a un saut de page.

TABLE 12.3 – Les quatre principales commandes d'espacement

# Chapitre 13

## Les gabarits

Voyons ici quelques gabarits dont il est tout à fait possible de vous inspirer.

### 13.1 La classe article

```
\documentclass[a4paper,10pt]{article}
\usepackage[utf8x]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[french]{babel}
\usepackage{lmodern} % Pour changer le pack de police
\author{\textsc{Nom} Prénom}
\date{\today}
\title{Mon titre d'article}

\begin{document}

\maketitle

\begin{abstract}
Le résumé (abstract en anglais) de mon article.
\end{abstract}

% Le contenu
Bla bla bla

\end{document}
```

### 13.2 La classe report

```
\documentclass[a4paper,10pt]{report}
\usepackage[utf8x]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[french]{babel}
\usepackage{lmodern} % Pour changer le pack de police
\usepackage{makeidx}
\title{Le titre}
\author{\textsc{Nom} Prénom}
```

```

\date{} % Pour mettre la date du jour, tapez \today
\makeindex

\begin{document}

\maketitle

\newpage
\tableofcontents
\newpage

% Le contenu
Bla\index{bla} bla bla

\listoffigures
\listoftables
\printindex

\end{document}

```

### 13.3 La classe book

```

\documentclass{book}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[french]{babel}
\usepackage{lmodern} % Pour changer le pack de police
\usepackage{makeidx}
\makeindex
\title{Le titre}
\author{\textsc{Nom} Prénom}
\date{\today}

\begin{document}

\maketitle % Page de garde

\frontmatter

Pages introductives

\mainmatter

Contenu

\appendix

Chapitres annexes
% Certains mettent la bibliographie après backmatter.
\bibliographystyle{} % Le style est mis entre crochets.

```



```
\bibliography{bibli} % Mon fichier de base de données s'appelle bibli.bib.  
  
\backmatter  
  
Epilogue  
  
\tableofcontents  
\listoffigures  
\listoftables  
\printindex  
  
\end{document}
```

## 13.4 La classe letter

```
\documentclass{letter}  
  
\usepackage[utf8]{inputenc}  
\usepackage[T1]{fontenc}  
\usepackage[francais]{babel}  
\usepackage{lmodern}  
  
\signature{M. Laleloulilo (signature)}  
\address{Première ligne de l'adresse \\ Deuxième ligne \\ Troisième ligne}  
  
\begin{document}  
% Pas besoin d'insérer un champ de datation. Il est ajouté par défaut.  
\begin{letter}{Un destinataire\\Un autre \\ Son copain \& son lapin \\  
ligne d'adresse 1 \\ ligne d'adresse 2 \\ ligne d'adresse 3}  
\opening{Formule de politesse d'ouverture,}  
  
Le texte.  
  
\closing{Cordialement, (formule de politesse)}  
\ps{P.-S. : Votre petit ajout ! :-)}  
  
\end{letter}  
  
\end{document}
```



## Chapitre 14

# Les bibliographies

Tout bon texte d'étude rédigé doit nécessairement comporter une bibliographie.  $\LaTeX$  offre à ce sujet des possibilités étendues, qui ne sont pas toujours aisées à aborder, mais les maîtriser ne donnera que plus de valeur à vos publications.

### 14.1 Le principe - Approche théorique

Réaliser une bibliographie consiste à mettre en forme une liste d'ouvrages ou de publications. Il existe pour cela divers modèles bibliographiques<sup>1</sup>. Retenons l'idée qu'« *une bibliographie est une liste structurée de références de documents, généralement écrits* »<sup>2</sup>.

Concernant la réalisation d'une bibliographie avec  $\LaTeX$  il faut savoir que ce dernier utilise pour cela un programme externe du nom de  $\text{Bib}\TeX$ . Ce programme fournit une armada d'outils destinés à concevoir facilement sa liste (la base de données) contenant toutes les informations sur les ouvrages (auteur, date de parution, etc.). Chaque ouvrage se voit affecté une étiquette et il suffira d'intégrer cette étiquette dans le document `.tex` pour que l'ouvrage en question figure dans le document rédigé.

Les étapes de ce principe :

1. Création d'une base de données (listing des ouvrages et publications). Chaque ouvrage se voit affecté une étiquette.
2. Les étiquettes sont insérées dans le texte.
3. Les différents ouvrages appelés dans le manuscrit sont sélectionnés dans la base de données.
4. La bibliographie est créée dans le document.

### 14.2 La base de données

Cette base de données est constituée de paragraphes contenant chacun les informations et l'étiquette des ouvrages. Le fichier qui sera enregistré le sera au format `.bib`. La structuration des informations concernant un ouvrage dans une base de données est toujours la même :

```
@type_d'oeuvre{référence_courte,  
Titre_du_champ1={Texte},  
Titre_du_champ2={Texte},  
Titre_du_champ2={Texte},
```

---

1. Nous renvoyons le lecteur vers la page *Wikipedia* consacrée au sujet (<https://fr.wikipedia.org/wiki/Bibliographie>) et notamment la partie listant les divers types de bibliographies

2. Source *Wikipedia*

```

Titre_du_champ2={Texte},
}
Entre deux ouvrages, tout texte est considéré comme commentaire.
@type_d'oeuvre{référence_courte,
Titre_du_champ1={Texte},
Titre_du_champ2={Texte},
Titre_du_champ2={Texte},
Titre_du_champ2={Texte},
}

```

Type d'oeuvre	Correspondance	Champs disponibles
@article	Article	author, journal, title, year, month, pages, notes
@book	Livre	author, title, publisher, year
@manual	Document technique	title, author, year, organization
@misc	Divers	author, title, year, month, note
@phdthesis	Thèse de doctorat	author, title, school, year
@unpublished	Manuscrit non publié	author, title, note, year, month

TABLE 14.1 – Les types d'oeuvres et les champs disponibles, parmi les plus répandus.

Pour le champ author, les noms et les prénoms sont séparés par une virgule, et quand il y a plusieurs auteurs ils sont eux séparés par l'instruction AND.

Champs	Correspondance
author	Auteur
journal	Journal
month	Mois de publication au choix ( <b>j</b> an, feb, mar...)
note	Notes complémentaires
organization	Société ou organisation
pages	Pages concernées
publisher	Editeur
year	Année de publication

TABLE 14.2 – Les principaux champs.

Exemple pour article de revue :

```

@article{ref,
author={Durand, Patrick AND Durand, René},
journal={La gazette linuxienne},
title={La distribution qui va détrôner Windows...},
year={2022},
months={jun},
pages={8},
notes={Complément paramétrages}
}

```

Exemple pour un livre :

```
@book{ref,
author={Kerouac, Jack},
title={Sur la route},
publisher={Gallimard},
year={1976}
}
```

### 14.3 Mise en place de la bibliographie

Nous ferons référence à des éléments de la base de données via la commande :

```
\cite{référence courte}
```

que nous placerons à l'endroit désiré dans le texte. Cette commande est à allier avec les commandes :

```
\bibliographystyle{style}
\bibliography{nom fichier .bib (sans.bib)}
```

Exemple :

```
\begin{document}

Mon texte est issue de l'ouvrage \cite{ref} qui s'appuie lui même sur
le livre \cite{ref2}...

\bibliographystyle{un des styles}
\bibliography{ma_bibli}

\end{document}
```

`\bibliographystyle{style}` utilise divers styles que nous pouvons résumer dans le tableau ci-dessous :

Styles	Effets
plain	Classe les entrées par ordre alphabétique et les numérote en conséquence.
abbrv	Classe les entrées par ordre alphabétique, les numérote en conséquence et abrège certains éléments de la bibliographie.
unsrt	Trie les entrées par ordre d'apparition dans le texte.
alpha	Le repère n'est plus un chiffre, mais les trois premières lettres du nom de l'auteur accolées aux derniers chiffres de l'année de parution.

TABLE 14.3 – Les styles bibliographiques

## 14.4 La compilation

La compilation d'une bibliographie va se dérouler en plusieurs temps. Deux compilations du fichier `.tex` source seront d'abord nécessaire, avant la compilation du fichier `.bib`, puis à nouveau deux autres compilations du fichier `.tex`.

Sous *GNU/Linux* cela donne :

```
$ latex fichier_source.tex
$ latex fichier_source.tex
$ bibtex fichier.bib
$ latex fichier_source.tex
$ latex fichier_source.tex
```

## Chapitre 15

# Structurer son document avec plusieurs fichiers .tex

### 15.1 Le concept

Pour les longs documents (tel celui-ci) il est possible de les scinder en plusieurs fichiers .tex. Cela permet de travailler sur une partie sans toucher aux autres. Ce principe repose sur une simple commande :

```
\input{chemin_du_fichier}
```

Avec cette commande, L<sup>A</sup>T<sub>E</sub>X insère alors le contenu du fichier appelé.

### 15.2 Illustration par un exemple fictif

Il faut tout d'abord créer la structure de fichiers du projet de notre document :

Répertoire principal

```
|__ Fichier .tex principal
|
|__ Répertoire "Part1"
|    |__ Fichier "chapitre1.tex"
|    |__ Fichier "chapitre2.tex"
|
|__ Répertoire "Part2"
|    |__ Fichier "chapitre3.tex"
|    |__ Fichier "chapitre4.tex"
```

Le fichier .tex principal contiendra alors les lignes suivantes :

```
\documentclass[book]

% Insertion des lignes de préambule

\begin{document}

\part{Première partie}
\input{Part1/chapitre1.tex}
\input{Part1/chapitre2.tex}
```

```
\part{Seconde partie}
\input{Part2/chapitre3.tex}
\input{Part2/chapitre4.tex}

\end{document}
```

Voilà donc comment dissocier un long document en plusieurs fichiers `.tex`.



## **Quatrième partie**

### **Des packages pour encore plus de fonctionnalités**



Thématiques	Packages	Description
Packages de langue	babel fontenc inputenc	Indispensables pour écrire un document en français.
Création d'un layout	layout	Pour afficher le gabarit de mise en page d'un document.
Modification des marges	geometry	Ensemble de commandes qui permettent de régler finement les marges.
Interligne	setspace	
Soulignement	soul ulem	Avec le package <i>soul</i> il est également possible de barrer du texte
Symbole euro	eurosym	
Packs de police	bookman charter lmodern mathpazo mathptmx newcent	Il est fortement conseillé de n'en utiliser qu'un à la fois.
Citation d' <i>url</i>	url	
Citation de code	listings moreverb verbatim	Le package <i>verbatim</i> donne accès aux environnements <i>verbatim</i> et <i>verbtimtab</i> . Avec le package <i>listings</i> il ne faut pas oublier de paramétrer <code>\lstset</code> . De plus <i>lstlisting</i> permet de citer du code avec la coloration syntaxique.
En-têtes et pieds de pages personnalisés	fancyhdr	
Insertion d'images	graphicx wrapfig	Le package <i>graphicx</i> est primordial pour travailler sur des images. <i>wrapfig</i> permet d'insérer une image dans un paragraphe (à utiliser avec parcimonie).
Manipuler les couleurs	color colortbl	<i>color</i> va permettre de colorer du texte et <i>colortbl</i> de colorer du texte contenu dans un tableau.
Les objets flottants	float	
Les tableaux	array multirow	
Les index	makeindex	

TABLE 15.1 – Liste des packages selon leur thématique



## Chapitre 16

# Les packages de langue

### 16.1 babel

Le package *babel* permet l'utilisation de commandes propres à la langue choisie, comme par exemple pour le français, l'utilisation des guillemets français ou les petites capitales.

Aperçu	Commande
« »	<code>\og et \fg</code>
Exposant <sup>2</sup>	<code>\up{2}</code>
PETITES CAPITALES	<code>\bsc{Petites capitales}</code>
1 <sup>o</sup>	<code>\primo</code>
2 <sup>o</sup>	<code>\secundo</code>
3 <sup>o</sup>	<code>\tertio</code>
4 <sup>o</sup>	<code>\quarto</code>
11 <sup>o</sup>	<code>\FrenchEnumerate{11}</code>
n <sup>o</sup>	<code>\no</code>
N <sup>o</sup>	<code>\No</code>
n <sup>os</sup>	<code>\nos</code>
N <sup>os</sup>	<code>\Nos</code>
35 <sup>o</sup>	<code>35\degres</code>

TABLE 16.1 – Commandes introduites par le package *babel* pour le français

### 16.2 fontenc

### 16.3 inputenc



## **Chapitre 17**

### **Création d'un *layout***





## **Chapitre 18**

### **Modification des marges avec *geometry***



## **Chapitre 19**

### **Les interlignes avec *setspace***



## Chapitre 20

# Les soulignements

**20.1** *soul*

**20.2** *ulem*



## **Chapitre 21**

### **L'euro avec *eurosym***





## Chapitre 22

# Les packs de police

**22.1** *bookman*

**22.2** *charter*

**22.3** *lmodern*

**22.4** *mathpazo*

**22.5** *mathptmx*

**22.6** *newcent*



## **Chapitre 23**

### **Les citations d'*url***



## Chapitre 24

# Citation de code

**24.1** *listings*

**24.2** *moreverb*

**24.3** *verbatim*



## **Chapitre 25**

### **En-têtes et pieds de pages personnalisés avec *fancyhdr***





## Chapitre 26

# L'insertion d'images

26.1 *graphicx*

26.2 *wrapfig*



## **Chapitre 27**

# **Manipuler les couleurs**

**27.1** *color*

**27.2** *colortbl*



## **Chapitre 28**

### **Les objets flottants avec *float***



## **Chapitre 29**

# **Les tableaux**

### **29.1   *array***

### **29.2   *multirow***





## **Chapitre 30**

### **Les index avec *makeindex***



## **Cinquième partie**

# **Le monde de L<sup>A</sup>T<sub>E</sub>X**



## **Chapitre 31**

### ***Texmaker***



## Chapitre 32

# Des ressources complémentaires

### 32.1 Ressources du Web

Le tutoriel de formation L<sup>A</sup>T<sub>E</sub>X de l'ENS : <https://www.tuteurs.ens.fr/logiciels/latex/>

Le Wikibooks en français : <https://fr.wikibooks.org/wiki/LaTeX>

Le cours de Ukonline : <https://www.ukonline.be/programmation/latex/tutoriel/>

### 32.2 Bibliographie L<sup>A</sup>T<sub>E</sub>X

MAGUIS Noël-Arnaud , *Rédigez des documents de qualité avec L<sup>A</sup>T<sub>E</sub>X - L'outil des professionnels pour publier mémoires, thèses, rapports, articles scientifiques.*, Ed. Le Livre du Zéro, 2010.

LOZANO Vincent , *Tout ce que vous avez toujours voulu savoir sur LaTeX sans jamais oser le demander. Ou comment utiliser LaTeX quand on n'y connaît goutte*, Ed. Framasoft, 2013, <https://archives.framabook.org/tout-sur-latex/index.html>

### 32.3 Bibliographie typographie

ANDRÉ Jacques , *Petites leçons de typographie*, Ed. du Jobet, 1990, [jacques-andre.fr/faqtypo/lessons.pdf](http://jacques-andre.fr/faqtypo/lessons.pdf).





## **Table des figures**



## Liste des tableaux