

BIEN DÉBUTER EN **LaTeX**

RICHARD TAILLET

BIEN DÉBUTER EN LaTeX

Collection informatique

Brooks J.C., Grow C., Craig P., Short D., *Cybersécurité. Sécurisation des systèmes informatiques*

Desgraupes B., *LaTeX. Apprentissage, guide et référence*

Grimes R.A., *Hacking et contre-hacking. La sécurité informatique*

Lubanovic B., *Python. Comprendre les bases et maîtriser la programmation*

Monk S., *Programmation Arduino. Développez rapidement vos premiers programmes*

Taillet R., *Python pour la physique. Calcul, graphisme, simulation*

Thiry T., *Les pratiques de l'équipe agile. Définissez votre propre méthode*, 2^e éd.

Chez le même éditeur

Agashe A., Detroja P., Mehta N., *Blockchain : bulle ou révolution ?*

Cypel A., *Au cœur de l'intelligence artificielle. Des algorithmes à l'IA forte*

INF&RMTIQUE

BIEN DÉBUTER EN LaTeX

RICHARD TAILLET

Pour toute information sur notre fonds et les nouveautés dans votre domaine de spécialisation, consultez notre site web: **www.deboecksuperieur.com**

© De Boeck Supérieur s.a., 2022
Rue du bosquet 7, B - 1348 Louvain-la-Neuve

Il est interdit, sauf accord préalable et écrit de l'éditeur, de reproduire (notamment par photocopie) partiellement ou totalement le présent ouvrage, de le stocker dans une banque de données ou de le communiquer au public, sous quelque forme et de quelque manière que ce soit.

Dépôt légal :
Bibliothèque Nationale, Paris : août 2022
Bibliothèque royale de Belgique, Bruxelles : 2022/13647/118

ISBN 978-2-8073-3989-7

Table des matières

Avant-propos	11
Chapitre 1 Introduction	13
1.1 Introduction	13
1.1.1 L ^A T _E X, c'est quoi ?	13
1.1.2 Historique	14
1.2 Installation et utilisation	14
1.2.1 Plusieurs façons d'invoquer L ^A T _E X	14
1.2.2 Outils en ligne	15
1.2.3 Installation	15
1.2.4 Éditeurs adaptés	15
1.2.5 Commandes en ligne dans une console	16
1.3 Ce que fait un traitement de texte	19
1.4 Règles d'or	20
1.5 Documentation	21
Chapitre 2 Commandes de base	23
2.1 Commandes	23
2.1.1 Commandes, arguments obligatoires et optionnels	23
2.1.2 Préambule et corps du texte	24
2.2 Le préambule	24
2.2.1 Classes de document	24
2.2.2 Taille des pages	25
2.2.3 Orientation des pages	25
2.2.4 Recto-verso	25
2.3 Caractères, mots et paragraphes	25
2.3.1 Espaces, sauts de ligne et sauts de page	25
2.3.2 La notion de package	27

Table des matières

2.3.3	Caractères accentués	28
2.3.4	Autres caractères	30
2.3.5	Caractères spéciaux	31
2.3.6	Règles de typographie françaises	31
2.3.7	Gras, italique, etc.	33
2.3.8	Majuscules et minuscules	34
2.3.9	Indices et exposants	35
2.3.10	Taille des caractères	35
2.3.11	Les unités de longueur	37
2.3.12	Césure des mots	38
2.4	Chapitres, sections, sous-sections, etc.	39
2.4.1	Commandes de base	39
2.4.2	Sections non numérotées	44
2.4.3	Référencage automatique des chapitres, sections, etc.	46
2.5	Listes	47
2.5.1	Notion d’environnement	47
2.5.2	Environnements de listes	47
2.5.3	Référencage automatique des listes	49
2.5.4	Le package <code>enumitem</code>	50
2.6	Espaces	55
2.6.1	Espaces horizontaux	55
2.6.2	Variables de longueur	57
2.6.3	Indentation	58
2.6.4	Espaces verticaux	61
2.7	Couleurs	62
2.7.1	Le package <code>color</code>	62
2.7.2	Le package <code>xcolor</code>	64
2.8	Écrire sur plusieurs colonnes	65
2.9	Hyperliens	67
2.10	Commentaires	70
2.11	Partitionner le contenu avec <code>\include</code>	71
2.12	Contrôle plus fin de la page : le package <code>geometry</code>	72
2.13	Environnements prédéfinis	74
2.13.1	Les environnements <code>quote</code> , <code>quotation</code> et <code>verse</code>	74
2.13.2	Les environnements <code>center</code> , <code>flushleft</code> et <code>flushright</code>	75
2.13.3	L’environnement <code>verbatim</code>	76
Chapitre 3	Images, tables et compteurs	79
3.1	Insérer une image avec <code>\includegraphics</code>	79
3.2	L’environnement flottant <code>figure</code>	81
3.3	Insérer une table	82
3.4	L’environnement flottant <code>table</code>	83
3.5	Compteurs	84

Chapitre 4	Mathématiques	87
4.1	Mode mathématique	87
4.2	Écriture des fonctions mathématiques	89
4.3	Indices et exposants	90
4.4	Symboles mathématiques	91
4.4.1	Lettres grecques	91
4.4.2	Symboles	92
4.4.3	Les flèches	93
4.4.4	Lettres calligraphiées	94
4.4.5	Encadrer une équation	94
4.4.6	Éviter le bricolage!	95
4.4.7	Les packages d’extension	95
4.4.8	Vrai texte	97
4.5	Les structures mathématiques	98
4.5.1	Fractions	98
4.5.2	Racine carrées	100
4.5.3	Sommes, produits et intégrales	100
4.5.4	Limites	101
4.5.5	Empiler des éléments	101
4.5.6	Barrer des termes	101
4.5.7	Parenthèses adaptatives, délimiteurs	102
4.5.8	Vecteurs et barres	106
4.5.9	Matrices	106
4.5.10	Nombres à virgules	108
4.6	Manipuler les équations	108
4.6.1	Numérotation des équations	108
4.6.2	Alignement des équations	114
4.6.3	Regrouper des équations	116
4.6.4	Découper des équations	119
4.7	Trouver comment écrire un symbole avec $\LaTeX$	121
4.8	Code $\LaTeX$ sur internet	122
Chapitre 5	Notes, en-tête et pied de page	123
5.1	Notes	123
5.1.1	Notes en bas de page	123
5.1.2	Notes finales	124
5.1.3	Notes marginales	125
5.2	En-têtes et pieds de page	125
5.3	Le package <code>fancyhdr</code>	128
5.3.1	Commandes de base	128
5.3.2	Documents en double page	130
5.3.3	Personnalisation du filet	130
5.4	La classe <code>memoir</code>	132

Table des matières

Chapitre 6	Personnalisation	133
6.1	Nouvelles commandes	133
6.1.1	Nouvelle commande sans argument	133
6.1.2	Nouvelle commande avec un argument	135
6.1.3	Nouvelle commande avec deux arguments	136
6.2	Nouveaux environnements	136
6.2.1	La commande <code>\newenvironment</code>	136
6.2.2	Nouveaux environnements avec des arguments	137
6.2.3	Nouveaux environnements avec des arguments optionnels	138
6.3	Nouveaux compteurs	139
6.4	Exemple : proverbes numérotés	141
6.4.1	Version de base	141
6.4.2	Raffinement	142
6.5	Personnaliser l'apparence des titres avec <code>titlesec</code>	143
6.6	Personnaliser l'apparence des chapitres	146
6.6.1	Personnaliser les titres de chapitres avec <code>titlesec</code>	146
6.6.2	Personnaliser les titres de chapitres avec <code>fncychap</code>	146
6.6.3	Personnaliser les titres de chapitres avec la classe <code>memoir</code>	149
Chapitre 7	Références	151
7.1	Tables des matières	151
7.1.1	La commande <code>\tableofcontents</code>	151
7.1.2	Sections non numérotées	153
7.1.3	Personnalisation de la table des matières	156
7.1.4	Table des figures	156
7.1.5	Table des tables	156
7.2	Index	157
7.2.1	Créer un index	157
7.2.2	Mise en forme des numéros de page	157
7.2.3	Modifier l'affichage	159
7.2.4	Différents niveaux d'index	160
7.2.5	Prologue de l'index	160
7.3	Références bibliographiques	160
7.3.1	Environnement <code>thebibliography</code>	161
7.3.2	Utilisation de <code>bibtex</code>	163
Chapitre 8	Aller plus loin	167
8.1	Supports de présentations avec beamer	167
8.2	Dessiner avec Tikz	169
8.2.1	Tracer des traits	169
8.2.2	Dessiner une figure géométrique	171
8.3	LuaLaTeX	173

Chapitre 9	Divers	175
9.1	Musique	175
9.2	Échecs	176
9.3	Sudoku	177
9.4	QRcode	178
9.5	Listings de programmes informatiques	179
9.6	Cartes de jeux de rôles	180
9.7	Dessin de molécules	181
Chapitre 10	Exercices	183
10.1	Mode texte	183
	Exercice 1 Symboles en mode texte	183
	Exercice 2 Extrait de l’Almageste	184
	Exercice 3 Commande pour écrire les siècles	185
	Exercice 4 Liste personnalisée	185
	Exercice 5 Utilisation du signe %	186
	Exercice 6 Commande qui n’écrit rien	187
	Exercice 7 Délimiteurs	188
	Exercice 8 Texte à trous	188
	Exercice 9 Commande <code>\smash</code>	189
	Exercice 10 Visualisation de quelques longueurs avec <code>tikz</code>	190
	Exercice 11 Index avec des numéros de pages en couleur	191
	Exercice 12 Table	192
	Exercice 13 Chiffres en style ancien	194
	Exercice 14 Dessiner avec Tikz	194
	Exercice 15 En-têtes et pieds de page	195
10.2	Expressions mathématiques	196
	Exercice 16 Écriture d’une somme	196
	Exercice 17 Limite	196
	Exercice 18 Matrices imbriquées	196
	Exercice 19 Alignement dans une matrice	197
	Exercice 20 Intégrales sur des surfaces fermées	199
	Exercice 21 Taille de parenthèses	201
	Exercice 22 Écriture d’une commande mathématique	202
	Exercice 23 Utilisation de <code>cases</code>	203
	Exercice 24 Fraction continue	204
	Exercice 25 Vecteurs pointés	205
	Exercice 26 Vecteurs primés	205
	Exercice 27 Coefficient binomial	206
	Exercice 28 Quantificateurs	207
	Exercice 29 Commande pour écrire des dérivées partielles	207
	Exercice 30 Écriture mathématique en gras	208
	Exercice 31 Boîte de longueur nulle	208
	Exercice 32 Réaction chimique	209

Table des matières

	Exercice 33	Réaction nucléaire	210
10.3	Utilisation de divers packages		211
	Exercice 34	Package <code>stmaryrd</code>	211
	Exercice 35	Tableaux de young et package <code>ytableau</code>	212
	Exercice 36	Lettrine sophistiquée avec le package <code>lettrine</code>	213
	Exercice 37	Table des matières personnalisée, package <code>tocloft</code>	213
	Exercice 38	Utilisation du package <code>witharrows</code>	216
10.4	Utilisation du package <code>titlesec</code>		217
	Exercice 39	Mise en forme des sections avec <code>titlesec</code>	217
	Exercice 40	Chapitre personnalisé	218
	Exercice 41	Sections personnalisées	219
	Exercice 42	Plus de sections personnalisées	221
	Exercice 43	Encore plus de sections personnalisées	222
Chapitre 11 Compilation et messages d’erreur			229
11.1	Mécanisme de compilation		229
11.2	Quelques erreurs courantes		230
	11.2.1	« Undefined control sequence »	230
	11.2.2	« Missing } inserted »	231
	11.2.3	« Missing \$ inserted »	231
	11.2.4	Environnement ouvert mais jamais fermé	231
	11.2.5	« Command <code>\end{equation}</code> invalid in math mode »	232
	11.2.6	« Missing <code>\begin{document}</code> »	232
	11.2.7	Pas de <code>\end{document}</code>	233
Chapitre 12 Résumé des packages			235
12.1	Sélection de packages utiles		235
12.2	Les packages utilisés pour cet ouvrage		236

Avant-propos

L^AT_EX est un logiciel de traitement de texte qui a la réputation d'être difficile d'accès. J'ai eu l'occasion de donner, en premier cycle universitaire, des cours d'introduction à L^AT_EX et j'en ai tiré la conclusion qu'au bout d'une séance de deux heures, une personne intéressée par la découverte de ce logiciel pouvait commencer à rédiger des documents de façon autonome, voire y prendre goût. Deux heures, cela peut sembler long, si l'on compare aux quelques minutes qui semblent suffire pour apprendre à taper un courrier dans des logiciels comme Word. Toutefois, au bout de ces deux heures de L^AT_EX, on est en position de faire des choses beaucoup plus élaborées et précises et le bilan de cet investissement en temps est très positif.

La réputation de difficulté de L^AT_EX a plusieurs origines. D'une part, il s'agit d'un *langage* de mise en forme et il est moins intuitif de taper une commande pour mettre un mot en gras que de cliquer sur un bouton. Cette difficulté est réelle. D'autre part, L^AT_EX traîne quelques boulets liés à son histoire : ses premières versions rendaient pénibles des actions indispensables, par exemple l'utilisation des caractères accentués demandait quelques acrobaties. Ceci a énormément changé et quelqu'un qui découvre L^AT_EX aujourd'hui n'aura pas à s'encombrer de ces difficultés. Enfin, la procédure même consistant à laisser à un logiciel le soin de mettre en page un texte qu'on lui soumet peut donner l'impression que l'utilisateur perd le contrôle. C'est en fait l'inverse et il est beaucoup plus simple de faire exactement ce qu'on souhaite avec L^AT_EX qu'avec Word, par exemple.

L'objectif de cet ouvrage est de donner les clés aux lecteurs et lectrices pour se mettre à L^AT_EX, sans connaissance préalable sur le sujet. Je me suis appuyé pour cela sur un enseignement universitaire ciblant précisément ce public d'une part, et sur l'analyse attentive d'ouvrages dits « d'initiation » qui partent souvent dans des considérations techniques qui peuvent rebuter les débutants, en m'efforçant de ne pas tomber dans ce travers. J'ai tout de même souhaité faire figurer dans l'ouvrage quelques notions avancées, sans les développer, tant pour exciter la curiosité du lecteur que pour montrer la puissance de ce logiciel.

Avant-propos

L'ouvrage est construit en quatre parties :

- ▶ les chapitres 1 à 4 présentent les notions de base ;
- ▶ les chapitres 5, 6 et 7 présentent quelques compléments utiles, à un niveau qui reste celui d'un débutant (qui aurait lu les chapitres précédents) ;
- ▶ les chapitres 8 et 9 proposent des pistes pour aller plus loin, il s'agit davantage de vitrines destinées à donner envie d'aller plus loin (en particulier le chapitre 9) que de véritables introductions ;
- ▶ le chapitre 10 propose des exercices corrigés, prétextes également à présenter des façons d'utiliser $\text{\LaTeX}$.

Introduction

1.1 Introduction

1.1.1 $\text{\LaTeX}$, c'est quoi ?

$\text{\LaTeX}$ (prononcer « latèque ») est un langage de programmation typographique, développé depuis plusieurs décennies et utilisé comme traitement de texte. L'approche est très différente de celle de la plupart des autres traitements de texte, dans lesquels l'utilisateur entre son texte sur la page qu'il veut remplir, applique en direct des outils de mise en forme (on parle d'approche WYSIWYG pour « What You See Is What You Get ») en cliquant sur des boutons ou en piochant dans des menus. Avec $\text{\LaTeX}$, l'utilisateur décrit dans un fichier le contenu qu'il souhaite produire, par exemple

- ▶ du texte à mettre en forme, avec des titres de chapitres, de sections, etc. ;
- ▶ des images à inclure ;
- ▶ des notes finales, des notes de bas de page, des notes en marge ;
- ▶ des références bibliographiques ;
- ▶ etc.

puis fait interpréter ce fichier par un compilateur. Ceci peut sembler complexe au premier abord et de fait, l'utilisation de $\text{\LaTeX}$ demande une prise en main plus longue que Word, InDesign, Open Office ou Google Doc. Les possibilités offertes sont à la hauteur de cette complexité initiale, comme le montre un survol du chapitre 9 qui présente quelques applications avancées de $\text{\LaTeX}$.

Si $\text{\LaTeX}$ ne doit pas être vu comme un concurrent aux autres traitements de textes plus conventionnels et peut s'avérer inadapté à certains usages, en particulier pour les débutants, il peut s'agir d'une alternative très avantageuse dans certains cas. D'une part, la gestion de la typographie mathématique est de loin supérieure à celle de Word et Open Office (on peut noter à ce propos que ces logiciels commencent à intégrer des éléments de $\text{\LaTeX}$, sinon dans le rendu, en tout cas dans la syntaxe). D'autre part, le format du document source (un fichier de texte pur, sans caractères cachés) permet une transparence totale sur les instructions de page : une instruction donnée produit

un effet donné, prévisible et reproductible, sur le texte. Enfin, grâce à l'immense communauté d'utilisateurs de $\text{\LaTeX}$, à travers le monde entier, de très nombreuses ressources sont disponibles pour rédiger des documents de presque n'importe quel type, avec un rendu de qualité professionnelle, par exemples des articles scientifiques, des rapports, du courrier. Au-delà du niveau débutant, des extensions permettent d'écrire des partitions, de faire des graphiques, des dessins, etc.

L'objectif de cet ouvrage est de présenter une introduction à ce langage, afin d'en montrer quelques possibilités et préparer le lecteur intéressé à l'abord d'ouvrages plus avancés (voir la section 1.5).

1.1.2 Historique

En 1977, Donald Knuth, mathématicien et informaticien, commence à travailler sur l'écriture d'un logiciel de typographie et de mise en page qui permet de composer des documents techniques de bien meilleure qualité que ce qui existait alors. Ce logiciel s'appelle $\text{\TeX}$ et au bout d'une décennie d'améliorations successives, sa version numéro 3, publiée à la fin des années 1980, constitue sa version stable et définitive. Ce logiciel $\text{\TeX}$ s'appuie sur un langage de programmation, avec des commandes et des variables. La compilation d'un fichier source, contenant le programme, produit un document mis en page qui peut être envoyé à une imprimante ou visualisé sur un écran. L'exécutable `tex` qui permet de lancer ce programme produit par défaut un document dans un format aujourd'hui peu usité, le DVI. On préférera dans la plupart des situations utiliser `pdftex` qui produit un fichier pdf.

Le langage $\text{\TeX}$ est puissant et versatile, mais peut être difficile d'accès. Dès le début des années 1980, Leslie Lamport a conçu un ensemble de commandes qui facilitent l'usage de $\text{\TeX}$. Par exemple, là où $\text{\TeX}$ permet d'écrire un titre de section en spécifiant le numéro de la section, la mise en forme du titre, $\text{\LaTeX}$ propose une commande `\section{arg}` qui automatise ces tâches. Cette simplification contribua à la grande popularité de $\text{\LaTeX}$, qui est encore aujourd'hui une référence majeure parmi les solutions de mise en page et de typographie. L'exécutable qui permet de produire un document en pdf à partir d'un fichier sources s'appelle `pdflatex`.

1.2 Installation et utilisation

1.2.1 Plusieurs façons d'invoquer $\text{\LaTeX}$

Pour obtenir un document mis en page, il faut compiler un document en format texte, appelé fichier source. L'usage universel est d'utiliser le suffixe `.tex` pour ce fichier source. Nous allons considérer plusieurs façons de le faire, correspondant souvent à des affinités différents avec l'outil informatique :

- ▶ On peut utiliser des outils en ligne, comme **Overleaf**, **Papeeria** ou **LaTeX base**
- ▶ On peut faire appel à des solutions intégrées, comme **TexMaker**, **Miktex**, **TexStudio**, **TeXshop** (Mac OS) qui regroupent des outils d'édition des fichiers et de compilation ;

► On peut invoquer et utiliser $\text{\LaTeX}$ via une commande en ligne, dans une console ; Nous allons passer en revue ces solutions, de la première, de loin la plus simple pour débiter, à la dernière, qui demande de savoir utiliser les commandes en ligne du système d'exploitation utilisé.

1.2.2 Outils en ligne

Plusieurs sites proposent une solution intégrée pour écrire et compiler des fichiers $\text{\LaTeX}$, sans avoir à installer le logiciel sur son ordinateur, pourvu qu'on dispose d'une connexion internet et d'un navigateur. On peut notamment citer :

- <https://papeeria.com/>
- <https://latexbase.com/>
- <https://www.overleaf.com/>

Nous allons ici décrire Overleaf. La version gratuite permet aux débutants (et même à beaucoup d'autres) d'utiliser $\text{\LaTeX}$ sans avoir à l'installer sur son ordinateur. Les fichiers source sont eux-même hébergés sur le cloud (ils peuvent bien sûr être téléchargés à tout moment). Pour débiter, la solution [overleaf.com](https://www.overleaf.com/) présente plusieurs points forts :

- + Pas d'installation de logiciel ;
- + Accès à de nombreux packages, dans leur version la plus à jour ;
- + Éditeur de texte intégré, performant et ergonomique ;
- + Aide complète et claire ;
- + Démarrage facile grâce à des fichiers « template » fonctionnels ;
- + Possibilité de travail collaboratif.

Cette solution présente également quelques points faibles, notamment :

- Elle nécessite d'être connecté à internet ;
- Elle peut trouver ses limites pour des gros projets, demandant des ressources importantes en termes de mémoire et de temps de compilation.

1.2.3 Installation

Les deux autres méthodes font appel au logiciel $\text{\LaTeX}$ installé en local sur son ordinateur. Le logiciel $\text{\LaTeX}$ est installé par défaut sur de nombreux systèmes et s'il ne l'est pas, des outils d'installation fiables et ergonomiques sont disponibles pour tous (Windows, Mac, Linux). Le plus simple est d'utiliser la page <https://www.latex-project.org/get/>.

1.2.4 Éditeurs adaptés

Les distributions $\text{\LaTeX}$ proposent généralement des éditeurs de texte adaptés à cet usage. Sans remettre en cause les grandes forces du logiciel Word, ce n'est pas un éditeur de texte proprement dit mais un logiciel de mise en page. En revanche, les

1 Introduction / Installation et utilisation

éditeurs suivants (liste non exhaustive) sont recommandés, ils proposent une coloration syntaxique ainsi que quelques fonctionnalités utiles (voir plus loin) :

- ▶ TeXworks, inclus dans la distribution Miktex ;
- ▶ Texmaker, utilisé par l’auteur pour écrire cet ouvrage ;
- ▶ Emacs, qui demande une prise en main plus délicate, mais qui est également très adapté à l’édition de codes informatiques et qui possède des fonctionnalités de programmation puissantes ;
- ▶ Atom.

Ils permettent tous d’accéder à la compilation via un bouton cliquable ou une option de menu, une fois $\text{\LaTeX}$ installé sur l’ordinateur. Dans cette liste, nous conseillons pour débiter **Texmaker**, qui donne accès à, entre autres,

- ▶ la console de messages (avertissements et erreurs) ;
- ▶ à une visualisation pdf pour certains d’entre eux ;
- ▶ des outils permettant d’entrer facilement des symboles, des environnements, des commandes $\text{\LaTeX}$;
- ▶ un menu « Assistant » pour faciliter l’écriture de tableaux et de matrices ;
- ▶ une présentation synthétique et cliquable de la structure du fichier (chapitres, section, etc., voir plus loin) ;
- ▶ la possibilité de replier le contenu de certains environnements, comme le montrent les captures d’écran de la figure 1.1 ;
- ▶ des outils pour gérer les fichiers de bibliographie ;

1.2.5 Commandes en ligne dans une console

Cette dernière méthode sera familière aux utilisateurs de Linux, mais elle est également envisageable sur les autres systèmes d’exploitation (c’est ainsi qu’a été construit le présent ouvrage, sur Mac OS). Il faut localiser la console (ou « terminal ») sur votre ordinateur. Sur celui de l’auteur, l’ouverture de la console produit une fenêtre (figure de gauche) où **richardtaillet** est le nom de l’utilisateur et où le **\$** désigne le « prompt », un symbole indiquant qu’une entrée est attendue de la part de l’utilisateur. Ici, pour invoquer **pdflatex**, on entre **pdflatex** puis la touche « entrée ». On voit s’afficher (figure de droite) des informations sur la version du logiciel, puis le symbole ****** qui constitue le « prompt » de $\text{\LaTeX}$. Nous n’allons pas ici aborder les actions possibles à ce stade, mais uniquement indiquer comment quitter le logiciel : taper **x** puis « entrée ».

1 Introduction / Installation et utilisation

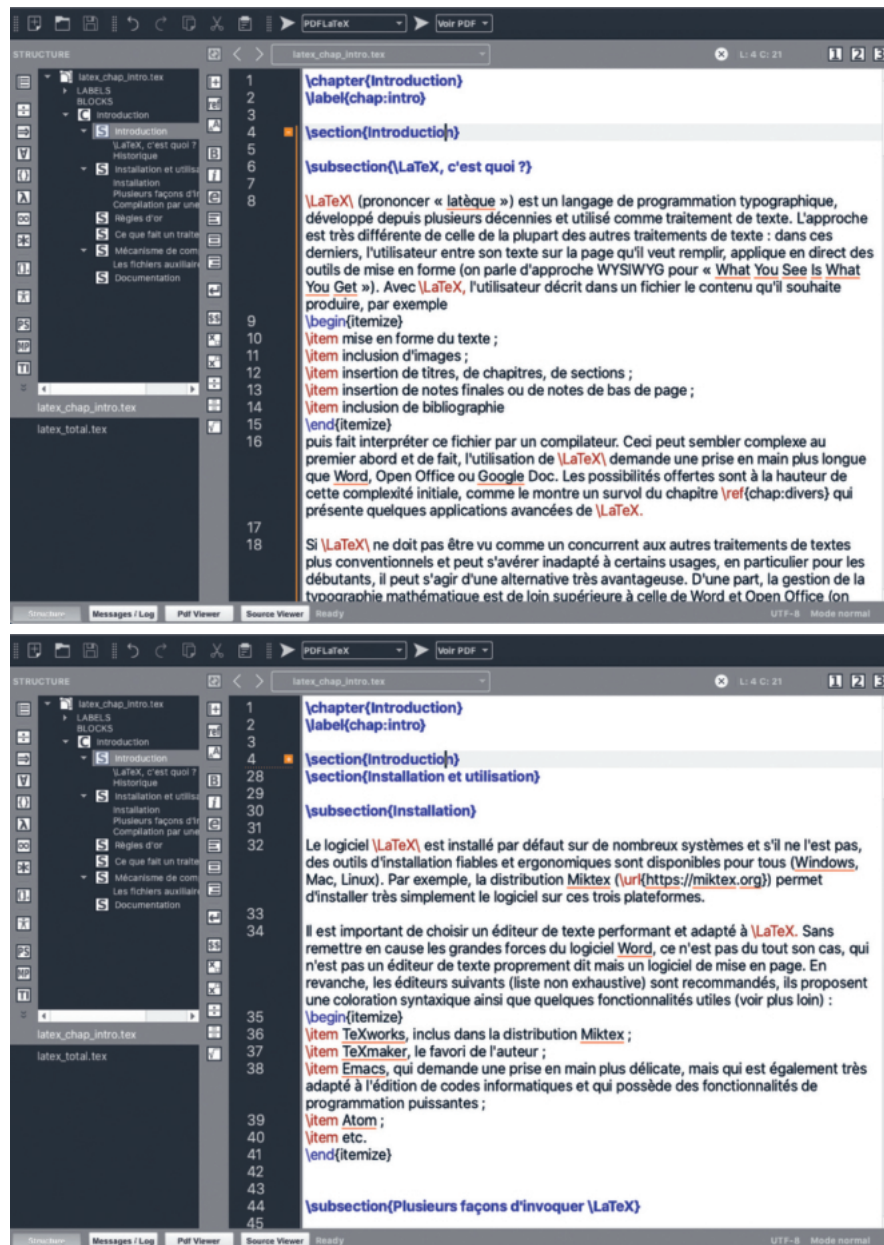
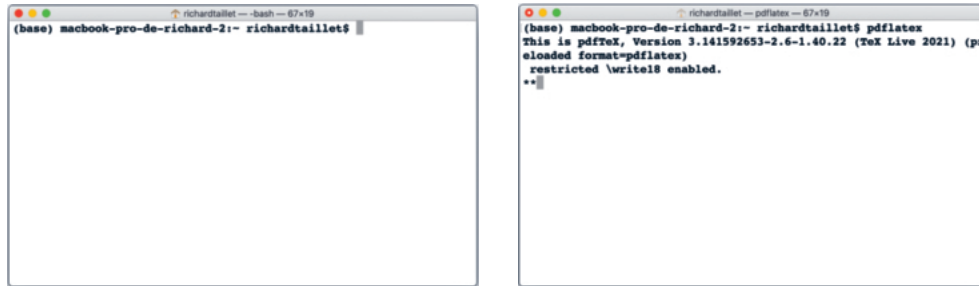


FIGURE 1.1 – Fonctionnalité de repliement des sections dans Texmaker : à gauche, un fichier $\text{\LaTeX}$ en train d'être édité. À droite, après avoir cliqué sur le , le contenu de la section concernée est replié, supprimé de l'affichage. Un nouveau clic sur le  qui apparaît alors restitue l'affichage complet du contenu de la section.

1 Introduction / Installation et utilisation



Considérons maintenant un fichier texte, nommé `mondocument.tex` qui contient :

```
1 \documentclass{minimal}
2 \begin{document}
3 Bonjour.
4 \end{document}
```

En entrant `pdflatex mondocument.tex` dans la console, on obtient

Console

```
(base) mbpderichard2-1:bouquin LaTeX richardtaillet$ pdflatex
mondocument.tex
This is pdfTeX, Version 3.141592653-2.6-1.40.22 (TeX Live 2021) (
preloaded format=pdflatex)
restricted \write18 enabled.
entering extended mode
(./mondocument.tex
LaTeX2e <2020-10-01> patch level 4
L3 programming layer <2021-02-18>
(/usr/local/texlive/2021/texmf-dist/tex/latex/base/minimal.cls
Document Class: minimal 2001/05/25 Standard LaTeX minimal class
) (/usr/local/texlive/2021/texmf-dist/tex/latex/l3backend/l3backend-
pdftex.def)

No file mondocument.aux.
[1{/usr/local/texlive/2021/texmf-var/fonts/map/pdftex/updmap/pdftex.
map}]
(./mondocument.aux) </usr/local/texlive/2021/texmf-dist/fonts/type1/
public/ams
fonts/cm/cmr10.pfb>
Output written on mondocument.pdf (1 page, 11377 bytes).
Transcript written on mondocument.log.
(base) mbpderichard2-1:bouquin LaTeX richardtaillet$
```

ainsi qu'un document pdf d'une page, appelé `mondocument.pdf` et situé dans le répertoire courant. Ce document contient le mot

Bonjour.

où le fond en couleur n'est pas présent dans le document produit, il sera destiné, dans cet ouvrage, à mettre en évidence ce qui est produit par la compilation.

Pour prendre un exemple un peu plus complexe, la page wikipedia qui lui est consacrée donne comme exemple le code suivant

```

1 \documentclass{minimal}
2
3 \begin{document}
4 Voici un exemple de \textbf{somme infinie} :
5 \[
6 \sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6}
7 \]
8 \end{document}

```

qui produit le texte suivant :

Voici un exemple de **somme infinie** :

$$\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

1.3 Ce que fait un traitement de texte

Un traitement de texte effectue de nombreuses tâches, souvent de manière discrète du point de vue de l'utilisateur. Le principe général est simple : à partir d'une suite de caractères contenus dans un fichier, le logiciel forme des mots, qu'il assemble pour former des lignes, elles-mêmes assemblées en paragraphes et en pages. Chacune de ces étapes peut receler des subtilités. Intéressons-nous par exemple au découpage du texte en lignes. En enchaînant des mots séparés par des espaces, il arrive un moment où l'ensemble devient trop long pour tenir sur une seule ligne. Où couper la ligne ? L'exemple suivant montre plusieurs choix qui peuvent être fait pour mettre en page une même phrase (la zone colorée permet de matérialiser la largeur de la colonne du texte)

La phrase est trop longue pour tenir sur une seule ligne et il faudrait revenir à la ligne.

La phrase est trop longue pour tenir sur une seule ligne et il faudrait revenir à la ligne.

La phrase est trop longue pour tenir sur une seule ligne et il faudrait revenir à la ligne.

La phrase est trop longue pour tenir sur une seule ligne et il faudrait revenir à la ligne.

La phrase est trop longue pour tenir sur une seule ligne et il faudrait revenir à la ligne.

La phrase est trop longue pour tenir sur une seule ligne et il faudrait revenir à la ligne.

1 Introduction / Règles d'or

Clairement, le premier n'est généralement pas acceptable, si on veut que le texte tienne dans les limites imposées. Le second non plus, le mot « ligne » étant coupé à un endroit non naturel. Si on veut que le texte soit justifié à droite (toutes les lignes complètes se terminant exactement au niveau de la marge de droite), les deux derniers choix sont acceptables. Les traitements de texte comportent des algorithmes qui permettent de proposer un choix optimal, à la fois du point de vue de la lisibilité et de celui du respect des règles typographiques. Dans le cas de L^AT_EX, tous les paramètres qui participent aux micro-décisions de couper ici ou là sont, en principe, paramétrables. Ceci dépasse le cadre de cet ouvrage d'introduction.

Au niveau qui précède celui de la composition des lignes, l'assemblage des lettres pour former des mots ne se résume pas à une simple juxtaposition : les fontes de caractères comportent des règles indiquant comment espacer les lettres entre elles, comment altérer leur apparence en fonction de leur environnement. Par exemple, les ligatures consistent à écrire certains enchaînement, comme « ff », « fi » et « fl », sous des formes spéciales :

$$\text{ff} \rightarrow \text{ff}, \text{fi} \rightarrow \text{fi}, \text{fl} \rightarrow \text{fl}$$

On voit que la forme du premier caractère est modifiée pour venir s'accrocher sur le second.

Au niveau suivant, le découpage en page doit éviter les « orphelines », c'est-à-dire des lignes seules en début de page. Les algorithmes tentent de resserrer le texte en amont (au détriment peut-être de la lisibilité) pour éviter la création de ces lignes solitaires. Ils décident aussi du placement des éléments dits « flottants » (voir le chapitre 3), figures et tables, pour garder une apparence équilibrée à l'ensemble du document.

1.4 Règles d'or

Lors de la rédaction d'un document, nous conseillons vivement de respecter les règles suivantes, qui se trouvent être facilitées par l'utilisation de L^AT_EX :

- **Règle 1 : donner du sens à la mise en forme typographique.** Par exemple, si certains éléments doivent être mis en gras parce que ce sont des mots importants, d'autres parce que ce sont des noms propres, il est déconseillé d'employer la mise en forme « gras » de façon indifférenciée. Il est de loin préférable de définir deux mises en forme (les utilisateurs des traitements de texte usuels parleront de « styles »), par exemple `\important{arg}` pour les premiers et `\nompropre{arg}` pour les seconds, même si *in fine* ces éléments auront la même apparence. Ceci permet de revenir d'un coup sur le choix typographique de tout un document, si on décide par exemple que finalement les noms propres seront écrits en italique plutôt qu'en gras ;
- **Règle 2 : faire confiance au comportement par défaut.** Avant de le modifier, se demander si c'est pertinent du point de vue de la forme et de la lisibilité. Par exemple on est tenté de redéfinir certains espaces, entre les paragraphes, ou

entre les éléments d'une équation, pour « faire plus joli ». Chacun reste le seul maître de son document, *in fine*, mais souvenez-vous que les règles par défaut résultent de plusieurs siècles d'expérience en typographie ;

- **Règle 3 : éviter le bricolage.** Il existe souvent des manières très simples et très propres de présenter les choses comme on le souhaite, grâce aux nombreux packages qui accompagnent $\text{\LaTeX}$. En effet, lorsqu'on cherche à réaliser une mise en forme particulière, il est probable que d'autres se soient posé la question et aient développé des solutions qui permettent de le faire de façon rigoureuse, puis les aient mises à disposition de la communauté d'utilisateurs sous la forme d'une extension de $\text{\LaTeX}$ (un « package »). Cet ouvrage présentera les plus importants pour débiter.

1.5 Documentation

Voici quelques autres ouvrages présentant $\text{\LaTeX}$, de façon plus ou moins technique. Ces ouvrages sont parfois redondants, mais plus souvent complémentaires. La liste est très loin d'être exhaustive.

- *The $\text{\TeX}$ book*, par Donald Knuth lui-même (1984, puis versions mises à jour)
- *$\text{\LaTeX}$, A document Preparation System*, par Leslie Lamport lui-même (seconde édition en 1994)
- *LaTeX Companion*, 3 éditions, par Frank Mittelbach et Michel Goossens (2023 pour la troisième édition)
- *LaTeX, apprentissage, guide et référence*, Bernard Desgraupes, chez De Boeck Supérieur (2019)

Internet regorge également de ressources permettant de bien débiter en $\text{\LaTeX}$. Les documentations des différents packages sont disponibles sur <https://ctan.org/>, nous préciserons à chaque fois l'adresse exacte à laquelle y trouver les ressources. Par ailleurs, le forum <https://tex.stackexchange.com/> est extrêmement riche en questions/réponses sur $\text{\LaTeX}$, il est possible d'y poser ses questions, après avoir vérifié qu'elles n'ont pas été déjà posées.

Avertissement : il convient de se méfier des bouts de code que l'on trouve tout faits sur internet, certains programmeurs $\text{\LaTeX}$ chevronnés n'ont pas suivi toutes les évolutions du langage et ratent parfois des solutions beaucoup plus performantes (et simples) que celles qu'ils ou elles proposent. Le cas des matrices est exemplaire à cet égard (voir la section 4.5.9 du chapitre 4).

Commandes de base

Ce chapitre a pour but de permettre au lecteur d’être capable de produire un document pdf avec L^AT_EX, en contrôlant l’apparence du texte, comme on le ferait avec un traitement de texte plus classique.

2.1 Commandes

2.1.1 Commandes, arguments obligatoires et optionnels

Un document L^AT_EX consiste, a minima, en un fichier au format texte, dont l’extension est `.tex`, et contenant du texte et des commandes. L’exemple minimal suivant produit un document pdf d’une page contenant le mot « bonjour » et un numéro de page :

```
1 \documentclass{report}
2 \begin{document}
3 Bonjour .
4 \end{document}
```

L^AT_EX utilise des commandes qui commencent par le caractère « `\` » appelé « backslash », « antislash » ou « trait de fraction inversé ». Il peut être utile de commencer par le localiser sur votre clavier. Ce caractère n’apparaît pas explicitement sur tous les claviers, pour les utilisateurs de Mac il est obtenu par `shift+option+\'`. Les mots commençant par « `\` » n’apparaissent pas dans le document compilé, ils servent à indiquer à L^AT_EX quoi faire apparaître et sous quelle forme. C’est tout l’objet de cet ouvrage. Les commandes peuvent avoir un ou plusieurs arguments, écrits à la suite de la commande, entre accolades « `{}` » pour les arguments obligatoires, entre crochets « `[]` » pour les arguments optionnels. Dans l’exemple qui précède, la commande `\begin{document}` indique à L^AT_EX l’endroit du fichier où commence le document à traiter.

2.1.2 Préambule et corps du texte

Dans toute la suite de cet ouvrage, on distinguera le « préambule », qui comprend la première partie du fichier, jusqu'à la commande `\begin{document}`, du reste du fichier appelé le « corps du texte ». Le préambule contient des instructions de mise en forme, des définitions générales et le corps du texte contient le texte proprement dit. Tout ce qui suit la commande `\end{document}` est ignoré par L^AT_EX (on peut y écrire du texte qu'on veut mettre de côté, des commentaires).

2.2 Le préambule

2.2.1 Classes de document

Le préambule contient les informations permettant de formater l'ensemble du document. A minima, il doit commencer par une ligne de la forme

```
1 \documentclass{arg}
```

où l'argument *arg*, par exemple un des mots-clé suivants (la liste n'est pas exhaustive),

- | | |
|------------------------|---------------------------|
| ▶ <code>article</code> | ▶ <code>letter</code> |
| ▶ <code>report</code> | ▶ <code>standalone</code> |
| ▶ <code>book</code> | ▶ <code>minimal</code> |
| ▶ <code>memoir</code> | |

indique la « classe » de document choisie. Ce choix prédétermine la disponibilité de certaines commandes et fixe certains choix typographiques, plus adaptés à l'écriture d'articles, de rapports, de livres, de mémoires ou de lettre respectivement, pour les premiers mots-clés donnés en exemple. Par exemple, les quatre premières classes permettent d'utiliser des commandes de structuration de sections, sous-sections, chapitres, etc. (voir la section 2.4).

La classe `minimal` comporte moins de fonctionnalités que les précédentes, elle est utilisée pour faire des tests en L^AT_EX. La classe `standalone` ne donne aucun accès à ces commandes non plus, cette classe est utile pour créer des documents pdf contenant simplement une formule, une phrase, un paragraphe ou une figure : la taille du document créé est adaptée à son contenu. Par exemple, le code

```
1 \documentclass{standalone}
2 \begin{document}
3 Bonjour !
4 \end{document}
```

produit une capsule pdf contenant juste le texte « Bonjour ! ». En remplaçant l'argument `standalone` par `article` dans la première ligne, on obtient un document d'une page contenant le même texte.

BIEN DÉBUTER EN LaTeX

Rédiger un texte sous LaTeX ? Cela devient simple avec ce livre, grâce à une présentation progressive et méthodique de ce langage réputé complexe.

LaTeX, langage très largement utilisé par les scientifiques, a su séduire de nombreuses communautés d'utilisateurs. Même si son abord est réputé difficile, l'auteur s'appuie sur son expérience d'enseignant et propose un livre qui s'adresse vraiment aux étudiants qui souhaitent débiter dans ce logiciel.

Il traite, à un niveau élémentaire, des fonctionnalités de base de LaTeX :

- comment démarrer,
- la mise en page élémentaires,
- les chapitres et sections,
- l'écriture des mathématiques,
- l'inclusion des figures,
- la conception de tables,
- les tables des matières,
- la construction d'un index,
- les notes marginales, etc.

Il présente également quelques applications originales, comme la conception de QR codes ou l'écriture de formules de chimie.

Il propose de nombreux exercices entièrement corrigés, qui permettent de s'entraîner sur les subtilités de LaTeX.

Richard Taillet est enseignant-chercheur en physique à l'université Savoie Mont Blanc. En plus de son activité académique, il est engagé dans de nombreuses actions visant à communiquer la science au grand public, en particulier le « Quart d'Heure Insolite ». Il est l'auteur de plusieurs ouvrages universitaires, dont le Dictionnaire de Physique (4 éditions), co-écrit avec Loïc Villain et Pascal Febvre, aux éditions De Boeck Supérieur.

**DU MÊME
AUTEUR**



ISBN : 978-2-8073-3989-7



9 782807 339897

deboeck
SUPÉRIEUR

www.deboecksuperieur.com