

Une courte (?) introduction à L^AT_EX 2_ε

ou L^AT_EX2e en 74 minutes

par Tobias Oetiker

Hubert Partl, Irene Hyna et Elisabeth Schlegl

traduit en français par Matthieu Herrb

Version 1.2
Janvier 1998

Ce document est « domaine public ». Il peut être imprimé et distribué gratuitement dans sa forme originale (comprenant la liste des auteurs). S'il est modifié ou que des extraits sont utilisés à l'intérieur d'un autre document, alors la liste des auteurs doit inclure tous les auteurs originaux et celui (ou ceux) qui a (qui ont) modifié le document. En cas d'usage commercial, le *copyleft* de GNU s'applique.

This document is “public domain”. It may be printed and distributed free of charge in its original form (including the list of authors). If it is changed or if parts of it are used within another document, then the author list must include all the original authors AND that author (those authors) who has (have) made the changes. When this document is used commercially, the GNU copyleft applies.

Merci !

Ce document est une traduction en français de « *The not so short introduction to LaTeX2e* » par Tobias Oetiker.

Une grande partie de ce document provient d'une introduction autrichienne à \LaTeX 2.09, écrite en allemand par :

Hubert Partl <partl@mail.boku.ac.at>

Zentraler Informatikdienst der Universität für Bodenkultur, Wien

Irene Hyna <Irene.Hyna@bmf.ac.at>

Bundesministerium für Wissenschaft und Forschung, Wien

Elisabeth Schlegl <no email>

in Graz

Si vous êtes intéressés par la version allemande, vous trouverez une version adaptée à \LaTeX 2 ϵ par Jörg Knappen sur :

`CTAN:/tex-archive/info/lkurz`

La version courante en anglais de ce document est disponible sur :

`CTAN:/tex-archive/info/lshort`

La version courante en français est disponible sur :

`ftp.laas.fr:/pub/Logiciels/latex/flshort/` en attendant d'être archivée sur CTAN.

Pour la préparation de ce document, l'aide des lecteurs du forum USENET `comp.text.tex` a été sollicité. De nombreuses personnes ont répondu et ont fourni des corrections, des suggestions et du texte pour améliorer ce document. Qu'ils en soient ici remerciés sincèrement. Ajoutons que je suis responsable de toutes les erreurs que vous pourriez trouver dans ce document.

Merci en particulier à :

Rosemary Bailey, David Carlisle, Chris McCormack, Wim van Dam, David Dureisseix, Elliot, David Frey, Robin Fairbairns, Alexandre Guimond, Cyril Goutte, Greg Gamble, Neil Hammond, Rasmus Borup Hansen, Martien Hulsen,

Eric Jacoboni, Alan Jeffrey, Byron Jones, David Jones,
Andrzej Kawalec, Alain Kessi, Christian Kern, Jörg Knappen,
Maik Lehradt, Martin Maechler, Claus Malten, Hubert Partl,
John Reffing, Mike Ressler, Brian Ripley, Young U. Ryu,
Chris Rowley, Craig Schlenter, Josef Tkadlec, Didier Verna,
and Rick Zaccone.

La version française a bénéficié des corrections des lecteurs du forum `fr.comp.text.tex` et en particulier de :

Vincent Zoonekynd, Christophe Dousson, Paul Gaborit,
Marie-Dominique Cabanne, Daniel Flipo, Sebastien Blondeel.

Note du traducteur : je tiens également à remercier chaleureusement les auteurs de ce document de le rendre publiquement utilisable et d'avoir ainsi rendu possible cette version française.

Préface

\LaTeX [1] est un logiciel de composition typographique adapté à la production de documents scientifiques et mathématiques de grande qualité typographique. Il permet également de produire toutes sortes d'autres documents, qu'il s'agisse de simples lettres ou de livres entiers. \LaTeX utilise \TeX [2] comme outil de mise en page.

Cette introduction décrit $\text{\LaTeX} 2_\epsilon$ et devrait se montrer suffisante pour la plupart des applications de \LaTeX . Pour une description complète du système \LaTeX , reportez-vous à [1, 3].

\LaTeX est disponible pour une vaste gamme de systèmes informatiques, du PC d'IBM aux plus gros CRAY. Dans de nombreuses universités, il est installé sur le réseau informatique, prêt à être utilisé. L'information nécessaire pour y accéder devrait être fournie dans le *Local Guide* [4]. Si vous avez des difficultés pour commencer, demandez de l'aide à la personne qui vous a donné cette brochure. Ce document *n'est pas* un guide d'installation du système \LaTeX . Son but est de vous montrer comment écrire vos documents afin qu'ils puissent être traités par \LaTeX .

Cette introduction est composée de quatre chapitres :

Le chapitre 1 présente la structure élémentaire d'un document $\text{\LaTeX} 2_\epsilon$.

Il vous apprendra également quelques éléments sur l'histoire de \LaTeX .

Après avoir lu ce chapitre, vous devriez avoir une vue générale de ce qu'est \LaTeX , pour pouvoir assimiler ce qui sera présenté dans les chapitres suivants.

Le chapitre 2 entre dans les détails du processus de codage d'un document. Il explique les commandes et les environnements essentiels de \LaTeX . Après avoir lu ce chapitre, vous serez capables de rédiger votre premier document.

Le chapitre 3 explique comment coder des formules mathématiques avec \LaTeX . De nombreux exemples y sont donnés pour mettre en valeur cet atout majeur de \LaTeX . À la fin de ce chapitre, vous trouverez des tableaux qui listent tous les symboles mathématiques disponibles.

Le chapitre 4 présente enfin divers outils de \LaTeX qui ne sont pas indispensables, mais cependant bien pratiques. Entre autres choses, vous apprendrez à insérer des figures PostScript dans vos documents et à gérer un index et une liste de références bibliographiques.

Il est important de lire ces chapitres dans l'ordre. Après tout, ce document n'est pas si long. Faites particulièrement attention aux exemples, car ils contiennent une bonne partie de l'information que vous pourrez trouver ici.

Si vous avez besoin de récupérer des fichiers relatifs à \LaTeX , utilisez l'une des archives ftp CTAN. En France elles sont sur les sites `ftp.loria.fr` et `ftp.ibp.fr`. Aux États-Unis, il s'agit de `ftp.shsu.edu`, en Allemagne de `ftp.dante.de` et au Royaume-Uni de `ftp.tex.ac.uk`. Si vous n'êtes pas dans l'un de ces pays, choisissez le site le plus proche de chez vous.

Si vous avez des suggestions concernant ce qui pourrait être ajouté, supprimé ou modifié dans ce document, contactez soit directement l'auteur de la version originale, soit moi-même, le traducteur. Nous sommes particulièrement intéressés par des retours d'utilisateurs débutants en \LaTeX au sujet des passages de ce livre qui devraient être mieux expliqués.

Tobias Oetiker <oetiker@ee.ethz.ch>
Department of Electrical Engineering,
Swiss Federal Institute of Technology, Zurich.

Matthieu Herrb <matthieu@laas.fr>
Laboratoire d'Analyse et d'Architecture des Systèmes,
Centre National de la Recherche Scientifique, Toulouse.

Table des matières

Merci !	iii
Préface	v
1 Ce qu'il faut savoir	1
1.1 Le nom de la bête	1
1.1.1 T _E X	1
1.1.2 L ^A T _E X	1
1.2 Les bases	2
1.2.1 Auteur, éditeur et typographe	2
1.2.2 Choix de la mise en page	2
1.2.3 Avantages et inconvénients	3
1.3 Fichiers source L ^A T _E X	4
1.3.1 Espaces	4
1.3.2 Caractères spéciaux	4
1.3.3 Commandes L ^A T _E X	5
1.3.4 Commentaires	6
1.4 Structure du fichier source	6
1.5 La mise en page du document	6
1.5.1 Classes de documents	8
1.5.2 Extensions	10
1.5.3 Styles de page	11
1.6 Gros documents	12
2 Mise en page	13
2.1 Sauts de ligne et de page	13
2.1.1 Paragraphes justifiés	13
2.1.2 Césure	14
2.2 Caractères spéciaux et symboles	16
2.2.1 Guillemets	16
2.2.2 Tirets	16
2.2.3 Points de suspension (. . .)	16
2.2.4 Ligatures	17

2.2.5	Accents et caractères spéciaux	17
2.3	L ^A T _E X international	18
2.4	L'espace entre les mots	19
2.5	Titres, chapitres et sections	19
2.6	Références croisées	21
2.7	Notes de bas de page	22
2.8	Souligner l'importance d'un mot	22
2.9	Environnements	22
2.9.1	Listes, énumérations et descriptions	23
2.9.2	Alignements à gauche, à droite et centrage	23
2.9.3	Citations et vers	24
2.9.4	Impression <i>verbatim</i>	25
2.9.5	Tableaux	26
2.10	Objets flottants	27
2.11	Nouvelles commandes et nouveaux environnements	30
2.11.1	Nouvelles commandes	31
2.11.2	Nouveaux environnements	31
3	Formules Mathématiques	33
3.1	Généralités	33
3.2	Groupements en mode mathématique	35
3.3	Éléments d'une formule mathématique	35
3.4	Espacement en mode mathématique	39
3.5	Alignements verticaux	40
3.6	Taille des polices mathématiques	41
3.7	Insertion de texte en mode mathématique	42
3.8	Décrire des variables	43
3.9	Théorèmes, propositions, etc.	43
3.10	Symboles gras	44
3.11	Liste des symboles mathématiques	46
4	Compléments	53
4.1	Polices et tailles	53
4.2	Espacement	55
4.2.1	Entre les lignes	55
4.2.2	Mise en page d'un paragraphe	56
4.2.3	Espacement horizontal	56
4.2.4	Espacement vertical	57
4.3	Disposition d'une page	58
4.4	Références bibliographiques	58
4.5	Index	60
4.6	Figures PostScript	62
	Bibliographie	65

TABLE DES MATIÈRES

ix

Index

67

Table des figures

1.1	Un fichier \LaTeX minimal	7
1.2	Exemple d'un article de revue réaliste	7
4.1	Paramètres de la disposition d'une page	59

Liste des tableaux

1.1	Classes de documents	8
1.2	Options de classes de document	9
1.3	Quelques extensions fournies avec \LaTeX	10
1.4	Les styles de page de \LaTeX	11
2.1	Accents et caractères spéciaux	18
2.2	Placements possibles	28
3.1	Accents en mode mathématique	46
3.2	Alphabet grec minuscule	46
3.3	Alphabet grec majuscule	46
3.4	Relations binaires	47
3.5	Opérateurs binaires	47
3.6	Opérateurs n-aires	48
3.7	Flèches	48
3.8	Délimiteurs	48
3.9	Grands délimiteurs	48
3.10	Symboles divers	49
3.11	Symboles non-mathématiques	49
3.12	Délimiteurs de l'AMS	49
3.13	Caractères grecs et hébreux de l'AMS	49
3.14	Relations binaires de l'AMS	50
3.15	Flèches de l'AMS	50
3.16	Négations des relations binaires et des flèches de l'AMS	51
3.17	Opérateurs binaires de l'AMS	51
3.18	Symboles divers de l'AMS	52
3.19	Polices mathématiques	52
4.1	Polices	54
4.2	Tailles des polices	54
4.3	Polices mathématiques	54
4.4	Unités \TeX	57
4.5	Exemples de clefs d'index	61
4.6	Clefs pour l'extension <code>graphicx</code>	63

Chapitre 1

Ce qu'il faut savoir

Dans la première partie de ce chapitre vous trouverez une rapide présentation de la philosophie et de l'histoire de $\LaTeX 2_{\epsilon}$. La deuxième partie met l'accent sur les structures fondamentales d'un document \LaTeX . Après avoir lu ce chapitre, vous devriez avoir une idée d'ensemble du fonctionnement de \LaTeX qui vous aidera à mieux comprendre les chapitres suivants.

1.1 Le nom de la bête

1.1.1 \TeX

\TeX est un programme écrit par Donald E. Knuth [2]. Il est conçu pour la composition de textes et d'équations mathématiques.

\TeX se prononce « Tech », avec un « ch » comme dans le mot écossais « Loch ». En alphabet phonétique cela donne [tex]... Dans un environnement ASCII, \TeX devient TeX.

1.1.2 \LaTeX

\LaTeX est un ensemble de macros qui permettent à un auteur de mettre en page son travail avec la meilleure qualité typographique en utilisant un format professionnel pré-défini. \LaTeX a été écrit par Leslie Lamport [1]. Il utilise \TeX comme outil de mise en page.

Récemment, \LaTeX a été mis à jour par l'équipe $\LaTeX 3$, menée par Frank Mittelbach, afin de réaliser certaines améliorations demandées depuis longtemps et de fusionner toutes les variantes qui s'étaient développées depuis la sortie de $\LaTeX 2.09$ il y a quelques années. Pour distinguer cette nouvelle version des précédentes, elle est appelée $\LaTeX 2_{\epsilon}$. Ce document est relatif à $\LaTeX 2_{\epsilon}$.

\LaTeX se prononce [latex]. Si vous voulez faire référence à \LaTeX dans un environnement ASCII, utilisez LaTeX. $\LaTeX 2_{\epsilon}$ se prononce [latex døzø] et s'écrit LaTeX2e.

En anglais, cela donne `[latex]` et `[latex tu: i:]`.

1.2 Les bases

1.2.1 Auteur, éditeur et typographe

Pour publier un texte, un auteur donne son manuscrit à une maison d'édition. L'éditeur décide alors de la mise en page du document (largeur des colonnes, polices de caractères, présentation des en-têtes,...). L'éditeur note ses instructions sur le manuscrit et le passe à un technicien typographe qui réalise la mise en pages en suivant ces instructions.

Un éditeur humain essaye de comprendre ce que l'auteur veut mettre en valeur et décide de la présentation en fonction de son expérience professionnelle et du contenu du manuscrit.

Dans un environnement \LaTeX , celui-ci joue le rôle de l'éditeur et utilise \TeX comme typographe pour la composition. Mais \LaTeX n'est qu'un programme et a donc besoin de plus de directives. L'auteur doit en particulier lui fournir la structure logique de son document. Cette information est insérée dans le texte sous la forme de « commandes \LaTeX ».

C'est une approche totalement différente de l'approche WYSIWYG¹ utilisée par les traitements de texte modernes tels que *Word pour Windows* ou *FrameMaker*. Avec ces programmes, l'auteur définit la mise en page du document de manière interactive pendant la saisie du texte. Tout au long de cette opération, il voit à l'écran à quoi ressemblera le document final une fois imprimé.

Avec \LaTeX , il n'est normalement pas possible de voir le résultat final durant la saisie du texte. Mais celui-ci peut être pré-visualisé après traitement du fichier par \LaTeX . Des corrections peuvent alors être apportées avant d'envoyer la version définitive vers l'imprimante.

1.2.2 Choix de la mise en page

La typographie est un métier (un art?). Les auteurs inexpérimentés font souvent de graves erreurs en considérant que la mise en page est avant tout une question d'esthétique : « si un document est beau, il est bien conçu ». Mais un document doit être lu et non accroché dans une galerie d'art. La lisibilité et la compréhensibilité sont bien plus importantes que le « look ». Par exemple :

- La taille de la police et la numérotation des en-têtes doivent être choisies afin de mettre en évidence la structure des chapitres et des sections.
- Les lignes ne doivent pas être trop longues pour ne pas fatiguer la vue du lecteur, tout en remplissant la page de manière harmonieuse.

1. What you see is what you get – Ce que vous voyez est ce qui sera imprimé.

Avec un logiciel WYSIWYG, l'auteur produit généralement des documents esthétiquement plaisants (quoi que...) mais très peu ou mal structurés. \LaTeX empêche de telles erreurs de formatage en forçant l'auteur à décrire la structure logique de son document et en choisissant lui-même la mise en page la plus appropriée.

1.2.3 Avantages et inconvénients

Un sujet de discussion qui revient souvent quand des gens du monde WYSIWYG rencontrent des utilisateurs de \LaTeX est le suivant : « les avantages de \LaTeX par rapport à un traitement de texte classique » ou bien le contraire. La meilleure chose à faire quand une telle discussion démarre, est de garder son calme, car souvent cela dégénère. Mais parfois on ne peut y échapper...

Les principaux avantages de \LaTeX par rapport à un traitement de texte traditionnel sont :

- mise en page professionnelle qui donne aux documents l'air de sortir de l'atelier d'un imprimeur.
- composition des formules mathématiques de manière pratique.
- il est seulement nécessaire de connaître quelques commandes de base pour décrire la structure logique du document. Il n'est pas nécessaire de se préoccuper de la mise en page.
- des structures complexes telles que des notes de bas de page, des renvois, la table des matières ou les références bibliographiques sont produites facilement.
- pour la plupart des tâches de la typographie qui ne sont pas directement gérées par \LaTeX , il existe des extensions supplémentaires gratuites. Par exemple pour inclure des figures POSTSCRIPT ou pour formater une bibliographie selon un standard précis. La majorité de ces extensions sont décrites dans *The \LaTeX Companion* [3].
- \LaTeX encourage les auteurs à écrire des documents bien structurés, parce que c'est ainsi qu'il fonctionne (en décrivant la structure).
- \TeX , l'outil de formatage de $\LaTeX 2_{\epsilon}$, est réellement portable et gratuit. Ainsi il est disponible sur quasiment toutes les machines existantes.

\LaTeX a également quelques inconvénients :

- il consomme plus de ressources (mémoire, espace disque, puissance de calcul) qu'un traitement de texte normal. Mais les choses changent,

puisque Word 6.0 pour Windows consomme plus de place disque qu'un système $\text{\LaTeX} 2_{\epsilon}$ normal. En ce qui concerne la puissance de calcul, \LaTeX bat tous les traitements de texte WYSIWYG, puisqu'il ne consomme de temps CPU que durant le formatage, alors qu'un logiciel WYSIWYG recalcule toute la page après chaque frappe de caractère.

- bien que quelques paramètres des mises en page pré-définies puissent être personnalisés, la mise au point d'une présentation entièrement nouvelle est difficile et demande beaucoup de temps².

1.3 Fichiers source \LaTeX

L'entrée de \LaTeX est un fichier texte ASCII. Vous pouvez le créer avec l'éditeur de texte de votre choix. Il contient le texte de votre document ainsi que les commandes qui vont dire à \LaTeX comment mettre en page le texte. On appelle ce fichier *fichier source*.

1.3.1 Espaces

Les caractères d'espacement, tels que les blancs ou les tabulations sont traités de manière unique comme « espace » par \LaTeX . Plusieurs blancs *consécutifs* sont considérés comme *une seule* espace. Les espaces en début d'une ligne sont en général ignorés et un seul retour à la ligne est traité comme une espace.

Une ligne vide entre deux lignes de texte définit la fin d'un paragraphe. *Plusieurs* lignes vides sont considérées comme *une seule* ligne vide. Le texte ci-dessous est un exemple. Sur la droite se trouve le contenu du fichier source et sur la gauche le résultat formaté.

Saisir un ou plusieurs espaces entre les mots n'a pas d'importance.	Saisir un ou plusieurs espaces entre les mots n'a pas d'importance.
Une ligne vide commence un nouveau paragraphe.	Une ligne vide commence un nouveau paragraphe.

1.3.2 Caractères spéciaux

Les symboles suivants sont des caractères réservés qui, soit ont une signification spéciale dans \LaTeX , soit ne sont pas disponibles dans toutes les polices. Si vous les saisissez directement dans votre texte, ils ne seront pas

² La rumeur dit que c'est un des points qui devrait être améliorés dans la future version $\text{\LaTeX}3$

imprimés mais forceront L^AT_EX à faire des choses que vous n'avez pas voulues.

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

Comme vous le voyez, certains de ces caractères peuvent être utilisés dans vos document en les préfixant par un antislash :

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

Les autres et bien d'autres encore pourront être obtenus avec des commandes spéciales à l'intérieur de formules mathématiques ou comme accents.

1.3.3 Commandes L^AT_EX

Les commandes L^AT_EX sont sensibles à la casse des caractères (majuscules ou minuscules) et utilisent l'un des deux formats suivants :

- soit elles commencent par un antislash `\` et ont un nom composé uniquement de lettres. Un nom de commande est terminé par une espace, un chiffre ou tout autre caractère qui n'est pas une lettre.
- soit elles sont composées d'un antislash et d'un caractère spécial (non-lettre) exactement.

L^AT_EX ignore les espaces après les commandes. Si vous souhaitez obtenir un blanc après une commande, il faut ou bien insérer `{ }` suivi d'un blanc ou bien utiliser une commande d'espacement spécifique de L^AT_EX. La séquence `{ }` empêche L^AT_EX d'ignorer les blancs après une commande.

J'ai lu que Knuth classe les gens qui utilisent T_EX en T_EXniciens et en T_EXperts. Aujourd'hui nous sommes le 3 février 1998.

J'ai lu que Knuth classe les gens qui utilisent `\TeX{}` en `\TeX{niciens}` et en `\TeX perts.` Aujourd'hui nous sommes le `\today`.

Certaines commandes sont suivies d'un paramètre qui est mis entre accolades `{ }`. Certaines commandes supportent des paramètres optionnels qui suivent le nom de la commande entre crochets `[]`. L'exemple suivant montre quelques commandes L^AT_EX. Ne vous tracassez pas pour les comprendre, elles seront expliquées plus loin.

You can *lean* on me!

You can `\textsl{lean}` on me!

Please, start a new line right here!
Thank you!

Please, start a new line
right here!`\linebreak[3]`
Thank you!

1.3.4 Commentaires

Quand \LaTeX rencontre un caractère `%` en analysant le fichier source, il ignore le reste de la ligne en cours. C'est utile pour ajouter des notes qui n'apparaîtront pas dans la version imprimée.

```
C'est un exemple pour tout le monde    C'est un exemple % stupide
                                         % Mieux: instructif <----
                                         pour tout le monde
```

1.4 Structure du fichier source

Quand \LaTeX analyse un fichier source, il s'attend à y trouver une certaine structure. C'est pourquoi chaque fichier source doit commencer par la commande :

```
\documentclass{...}
```

Elle indique quel type de document vous voulez écrire. Après cela vous pouvez insérer des commandes qui vont influencer le style du document ou vous pouvez charger des extensions qui ajoutent de nouvelles fonctions au système \LaTeX . Pour charger une extension, utilisez la commande :

```
\usepackage{...}
```

Quand tout le travail de préparation est fait³, vous pouvez commencer le corps du texte avec la commande :

```
\begin{document}
```

Maintenant vous pouvez saisir votre texte et y insérer des commandes \LaTeX . À la fin de votre document, utilisez la commande

```
\end{document}
```

pour dire à \LaTeX qu'il en a fini. Tout ce qui suivra dans le fichier source sera ignoré.

La figure 1.1 montre le contenu d'un document \LaTeX 2_ε minimum. Un fichier source plus complexe est présenté sur la figure 1.2.

1.5 La mise en page du document

3. La partie entre `\documentclass` et `\begin{document}` est appelée le *préambule*.

```
\documentclass{article}
\begin{document}
Small is beautiful.
\end{document}
```

FIG. 1.1: *Un fichier L^AT_EX minimal*

```
\documentclass[a4paper,11pt]{article}
\usepackage{latexsym}
\author{H. ~Partl}
\title{Minimalism}
\frenchspacing
\begin{document}
\maketitle
\tableofcontents
\section{Start}
Well and here begins my lovely article.
\section{End}
\ldots{} and here it ends.
\end{document}
```

FIG. 1.2: *Exemple d'un article de revue réaliste*

1.5.1 Classes de documents

La première information dont L^AT_EX a besoin en traitant un fichier source est le type de document que son auteur est en train de créer. Ce type est spécifié par la commande `\documentclass`.

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

Ici *classe* indique le type de document à créer. Le tableau 1.1 donne la liste des classes de documents présentées dans cette introduction. L^AT_EX 2_ε fournit d'autres classes pour d'autres types de documents, notamment des lettres et des transparents. Le paramètre *options* permet de modifier le comportement de la classe de document. Les options sont séparées par des virgules. Les principales options disponibles sont présentées dans le tableau 1.2.

Exemple : un fichier source pour un document L^AT_EX pourrait commencer par la ligne

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

elle informe L^AT_EX qu'il doit formater ce document comme un *article* avec une taille des caractères de base de *onze points* et qu'il devra produire une mise en page pour une impression *double face* sur du papier *A4*.

TAB. 1.1: *Classes de documents*

article	pour des articles dans des revues scientifiques, des présentations, des rapports courts, des documentations, des invitations,...
report	pour des rapports plus longs contenant plusieurs chapitres, des petits livres, des thèses,...
book	pour des vrais livres ;
slides	pour des transparents. Cette classe utilise de grands caractères sans serif.

TAB. 1.2: *Options de classes de document*

<code>10pt, 11pt, 12pt</code>	définit la taille de la police principale du document. Si aucune option n'est présente, la taille par défaut est de 10pt.
<code>a4paper, letterpaper, ...</code>	définit la taille du papier. Le papier par défaut est <code>letterpaper</code> , le format standard américain. Les autres valeurs possibles sont : <code>a5paper</code> , <code>b5paper</code> , <code>executivepaper</code> , et <code>legalpaper</code> .
<code>fleqn</code>	aligne les formules mathématiques à gauche au lieu de les centrer.
<code>leqno</code>	place la numérotation des formules à gauche plutôt qu'à droite.
<code>titlepage, notitlepage</code>	indique si une nouvelle page doit être commencée après le titre du document ou non. La classe <code>article</code> continue par défaut sur la même page contrairement aux classes <code>report</code> et <code>book</code> .
<code>twocolumn</code>	demande à L ^A T _E X de formater le texte sur deux colonnes.
<code>twoside, oneside</code>	indique si la sortie se fera en recto-verso ou en recto simple. Par défaut, les classes <code>article</code> et <code>report</code> sont en simple face alors que la classe <code>book</code> est en double-face.
<code>openright, openany</code>	fait commencer un chapitre sur la page de droite ou sur la prochaine page. Cette option n'a pas de sens avec la classe <code>article</code> qui ne connaît pas la notion de chapitre. Par défaut, la classe <code>report</code> commence les chapitres sur la prochaine page vierge alors que la classe <code>book</code> les commence toujours sur une page de droite.

1.5.2 Extensions

TAB. 1.3: *Quelques extensions fournies avec L^AT_EX*

<code>doc</code>	permet de documenter des programmes pour L ^A T _E X. Dérite dans <code>doc.dtx</code> ^a et dans <i>The L^AT_EX Companion</i> [3].
<code>exscale</code>	fournit des versions de taille paramétrable des polices mathématiques étendues. Dérite dans <code>ltxscale.dtx</code> .
<code>fontenc</code>	spécifie le codage des polices de caractère que L ^A T _E X va utiliser. Dérite dans <code>ltoutenc.dtx</code> .
<code>ifthen</code>	fournit des commandes de la forme 'if ... then do ... otherwise do' Dérite dans <code>ifthen.dtx</code> et dans <i>The L^AT_EX Companion</i> [3].
<code>latexsym</code>	permet l'utilisation de la police des symboles L ^A T _E X. Dérite dans <code>latexsym.dtx</code> et dans <i>The L^AT_EX Companion</i> [3].
<code>makeidx</code>	fournit des commandes pour réaliser un index. Dérite dans ce document, section 4.5 et dans <i>The L^AT_EX Companion</i> [3].
<code>syntonly</code>	analyse un document sans le formater. Dérite dans <code>syntonly.dtx</code> et dans <i>The L^AT_EX Companion</i> [3]. Utile pour une vérification rapide de la syntaxe.
<code>inputenc</code>	permet de spécifier le codage des caractères utilisé dans le fichier source, parmi ASCII, ISO Latin-1, ISO Latin-2, 437/850 IBM code pages, Apple Macintosh, Next, ANSI-Windows ou un codage défini par l'utilisateur. Dérite dans <code>inputenc.dtx</code> .

^a Ce fichier devrait être installé sur votre système et vous devriez être capable de le formater avec la commande `latex doc.dtx`. Il en est de même pour les autres fichiers cités dans ce tableau.

En rédigeant votre document, vous remarquerez peut-être qu'il y a des domaines où les commandes de base de L^AT_EX ne permettent pas d'exprimer ce que vous voudriez. Si vous voulez inclure des graphiques, du texte en couleur ou du code d'un programme dans votre document, il faut augmenter

les possibilités de L^AT_EX grâce à des extensions. Une extension est chargée par la commande

```
\usepackage[options]{extension}
```

Extension est le nom de l'extension et *options* une liste de mots-clés qui déclenchent certaines fonctions de l'extension. Certaines extensions font partie de la distribution standard de L^AT_EX 2_ε (Reportez-vous au tableau 1.3). D'autres sont distribuées à part. Le *Local Guide* [4] peut vous fournir plus d'informations sur les extensions installées sur votre site. *The L^AT_EX Companion* [3] est la principale source d'information au sujet de L^AT_EX 2_ε. Ce livre contient la description de centaines d'extensions ainsi que les informations nécessaires pour écrire vos propres extensions à L^AT_EX 2_ε.

1.5.3 Styles de page

L^AT_EX propose trois combinaisons d'en-têtes et de pieds de page, appelées styles de page et définies par le paramètre *style* de la commande :

```
\pagestyle{style}
```

Le tableau 1.4 donne la liste des styles prédéfinis.

TAB. 1.4: *Les styles de page de L^AT_EX*

plain imprime le numéro de page au milieu du pied de page. C'est le style par défaut.

headings imprime le titre du chapitre courant et le numéro de page dans l'en-tête de chaque page et laisse le pied de page vide. C'est à peu près le style utilisé dans ce document.

empty Laisse l'en-tête et le pied de page vides.

On peut changer le style de la page en cours grâce à la commande

```
\thispagestyle{style}
```

Dans *The L^AT_EX Companion* [3], on vous explique comment créer vos propres en-têtes et pieds de pages.

1.6 Gros documents

Lorsque l'on travaille sur des gros documents, il peut être pratique de couper le fichier source en plusieurs morceaux. \LaTeX a deux commandes qui vous permettent de gérer plusieurs fichiers sources.

```
\include{nom de fichier.tex}
```

Vous pouvez utiliser cette commande dans le corps de votre document pour insérer le contenu d'un autre fichier source. Remarquez que \LaTeX va sauter une page pour traiter le contenu de *nom de fichier*.

La seconde commande peut être utilisée dans le préambule. Elle permet de dire à \LaTeX de n'inclure que certains des fichiers désignés par les commandes `\include`.

```
\includeonly{nom de fichier, nom de fichier, ... }
```

Après avoir rencontré cette commande dans le préambule d'un document, seules les commandes `\include` dont les fichiers sont cités en paramètre de la commande `\includeonly` seront exécutées. Attention, il ne doit pas y avoir d'espace entre les noms de fichiers et les virgules.

La commande `\include` saute une page avant de commencer le formatage du texte inclus. Ceci est utile lorsqu'on utilise `\includeonly`, parce qu'ainsi les sauts de pages ne bougeront pas, même si certains morceaux ne sont pas inclus. Parfois ce comportement n'est pas souhaitable. Dans ce cas, vous pouvez utiliser la commande :

```
\input{nom de fichier}
```

qui insère simplement le fichier indiqué sans aucun traitement sophistiqué.

Attention : certaines extensions redéfinissent parfois certains caractères spéciaux⁴. Ils ne peuvent plus alors être utilisés dans les *noms de fichiers*.

4. Par exemple le caractère « souligné » : `~`.

Chapitre 2

Mise en page

Après la lecture du chapitre précédent vous connaissez maintenant les éléments de base qui constituent un document \LaTeX . Dans ce chapitre, nous allons compléter vos connaissances afin de vous rendre capables de créer des documents réalistes.

2.1 Sauts de ligne et de page

2.1.1 Paragraphes justifiés

Bien souvent les livres sont composés de lignes qui ont toutes la même longueur ; on dit qu'elles sont justifiées à droite. \LaTeX insère des retours à la ligne et des espacements entre les mots de manière à optimiser la présentation de l'ensemble d'un paragraphe. En cas de besoin, il coupe les mots qui ne tiennent pas en entier sur une ligne. La présentation exacte d'un paragraphe dépend de la classe de document¹. Normalement la première ligne d'un paragraphe est en retrait par rapport à la marge gauche et il n'y a pas d'espace verticale particulière entre deux paragraphes (Cf. section 4.2.2).

Dans certains cas particuliers, il peut être nécessaire de demander à

1. et des règles typographiques propres de chaque pays *NdT*.

L^AT_EX de couper une ligne :

`\` ou `\newline`

commence une nouvelle ligne sans commencer un nouveau paragraphe.

`\`*

empêche un saut de page après le saut de ligne demandé.

`\newpage`

provoque un saut de page.

`\linebreak[n]`, `\nolinebreak[n]`, `\pagebreak[n]` et `\nopagebreak[n]`

font ce que suggère leur nom anglais. Ces commandes permettent à l'auteur de paramétrer leur action par l'intermédiaire du paramètre optionnel n . Il peut prendre une valeur entre zéro et quatre. En donnant à n une valeur inférieure à quatre, vous laissez à L^AT_EX la possibilité de ne pas tenir compte de votre commande si cela devait rendre le résultat réellement laid.

L^AT_EX essaye toujours de trouver les meilleurs endroits pour les retours à la ligne. S'il ne trouve pas de solution pour couper les lignes de manière conforme à ses normes de qualité, il laisse dépasser un bout de ligne sur la marge droite du paragraphe. L^AT_EX émet alors le message d'erreur « overfull hbox² ». Cela se produit surtout quand L^AT_EX ne trouve pas de point de césure dans un mot. En utilisant alors la commande `\sloppy`, vous pouvez demander à L^AT_EX d'être moins exigeant. Il ne produira plus de lignes trop longues en ajoutant de l'espace entre les mots du paragraphe, même si ceux-ci finissent trop espacés selon ses critères. Dans ce cas le message « underfull hbox³ » est produit. Souvent, malgré tout, le résultat est acceptable. La commande `\fussy` agit dans l'autre sens, au cas où vous voudriez voir L^AT_EX revenir à ses exigences normales.

2.1.2 Césure

L^AT_EX coupe les mots si nécessaire. Si l'algorithme de césure⁴ ne trouve pas l'endroit correct pour couper un mot, vous pouvez utiliser les commandes suivantes pour informer T_EX de l'exception.

2. débordement horizontal

3. boîte horizontale pas assez pleine

4. Hyphenation en anglais

La commande :

```
\hyphenation{liste de mots}
```

permet de ne couper les mots cités en argument qu'aux endroits indiqués par « - ». Cette commande doit être placée dans le préambule et ne doit contenir que des mots composés de lettres normales. La casse des caractères n'est pas prise en compte. L'exemple ci-dessous permet à « anticonstitutionnellement » d'être coupé, ainsi qu'à « Anticonstitutionnellement ». Mais il empêche toute césure de « FORTRAN », « Fortran » ou « fortran ». Ni les caractères spéciaux ni les symboles ne sont autorisés dans cette commande.

Exemple:

```
\hyphenation{FORTRAN}
\hyphenation{An-ti-cons-ti-tu-tion-nel-le-ment}
```

La commande `\hyphenation{liste de mots}` a un effet *global* sur toutes les occurrences des mots de la liste. Si l'exception ne concerne qu'une occurrence d'un mot on utilise la commande `\-` qui insère un point de césure potentiel dans un mot. Ces positions deviennent alors les *seuls* points de césure possibles pour cette occurrence mot.

Normalement, en français, on ne coupe pas la dernière syllabe d'un mot si elle est muette, mais il arrive qu'on soit obligé de le faire, par exemple si on travaille sur des textes étroits (cas de colonnes multiples).

Exemple: on pourra coder `ils ex\pri\ment` pour autoriser *exceptionnellement* le rejet à la ligne suivante de la syllabe muette `ment`.

Cette commande est surtout utile pour les mots contenant des caractères spéciaux. En effet \LaTeX ne fait pas la césure des mots contenant des caractères spéciaux. La plupart des versions américaines de \TeX considèrent les accents comme des caractères spéciaux. Une version assez répandue hors des États-Unis, $\MIT\TeX$ ⁵, supprime cette restriction gênante pour les langues européennes. L'extension `inputenc` avec le codage T1 et la nouvelle famille de polices EC aboutissent au même résultat pour les versions récentes de $\LaTeX 2_{\epsilon}$.

I think this is: supercalifragilisticexpialidocious	I think this is: su\per\cal\-% i\frag\i\lis\tic\ex\pi\-% al\i\do\cious
---	--

Plusieurs mots peuvent être maintenus ensemble sur une ligne avec la commande :

```
\mbox{texte}
```

Elle a pour effet d'interdire toute coupure de ligne dans *texte*.

5. Développée par Michael J. Ferguson et Bernd Raichle

Mon numéro de téléphone va changer. À partir du 18 octobre, ce sera le 0561 336 330.

Le paramètre *nom du fichier* de la commande `input` contient le nom du fichier à lire.

Mon num\’ero de t\’el\’ephone va changer. \‘A partir du 18 octobre, ce sera le \mbox{0561 336 330}.

Le param\’etre
`\mbox{\emph{nom du fichier}}`
 de la commande `\texttt{input}`
 contient le nom du fichier
 \‘a lire.

2.2 Caractères spéciaux et symboles

2.2.1 Guillemets

Pour insérer des guillemets n’utilisez pas le caractère " comme sur une machine à écrire. En typographie, il y a des guillemets ouvrants et fermants spécifiques. Avec \LaTeX , utilisez deux ‘ pour les guillemets ouvrants et deux ’ pour les guillemets fermants.⁶

“Please press the ‘x’ key.”
 « Appuyez sur la touche ‘x’. »

‘‘Please press the ‘x’ key.’’
 << Appuyez sur la touche ‘x’. >>

2.2.2 Tirets

\LaTeX connaît quatre types de tirets. Trois d’entre eux sont accessibles en juxtaposant un nombre variable de tirets simples. Le quatrième n’est pas réellement un tiret : il s’agit du signe mathématique moins :

belle-fille,
 pages 13–67
 oui—ou non ?
 0, 1 et –1

belle-fille, \\
 pages 13--67\\
 oui---ou non ? \\
 \$0\$, \$1\$ et \$-1\$

2.2.3 Points de suspension (...)

Sur une machine à écrire, une virgule ou un point occupent la même largeur que les autres lettres. En typographie professionnelle, le point occupe très peu de place et il est placé tout près du caractère qui le précède. Il n’est

6. En français, avec l’extension `french`, utilisez `<<` et `>>` *NdT*.

donc pas possible d'utiliser trois points consécutifs pour créer des points de suspension⁷. À la place on utilise la commande spécifique :

\ldots

Non pas comme ça ... mais ainsi :	Non pas comme <code>\c{c}a ...</code>
New York, Tokyo, Budapest, ...	mais ainsi : <code>\</code>
	New York, Tokyo, Budapest, <code>\ldots</code>

2.2.4 Ligatures

Certaines séquences de lettres ne sont pas composées simplement en juxtaposant les différentes lettres les unes à la suite des autres, mais en utilisant des symboles spéciaux.

ff fi fl ffi ... à la place de ff fi fl ffi ...

Ces ligatures peuvent être désactivées en insérant un `\mbox{}` entre les lettres en question. Cela peut s'avérer utile pour certains mots composés⁸

Not shelfful	Not <code>shelfful</code> <code>\</code>
but shelfful	but <code>shelf</code> <code>\mbox{}</code> <code>ful</code>

2.2.5 Accents et caractères spéciaux

\LaTeX permet l'utilisation d'accents et de caractères spéciaux issus de nombreuses langues. Le tableau 2.1 montre tous les accents que l'on peut ajouter à la lettre o. Ils s'appliquent naturellement aux autres lettres.

Pour placer un accent sur un i ou un j, il faut supprimer leur point. Ceci s'obtient en tapant `\i` et `\j`.

Hôtel, naïve, élève,	<code>H\^otel, na\"i ve, \ 'el\ 'eve,\</code>
smørrebrød, ¡Señorita!,	<code>sm\o rrebr\o d, ! 'Se\~norita!,\</code>
Schönbrunner Schloß Straße	<code>Sch\"onbrunner Schlo\ss\}</code>
	<code>Stra\ss e</code>

⁷. sauf avec l'extension `french` qui rend la commande `\ldots` inutile.

⁸. Je n'ai pas trouvé d'exemple en français. *NdT*.

2.3 L^AT_EX international

Pour composer des documents dans des langues autres que l'anglais, L^AT_EX doit pouvoir s'adapter aux règles typographiques et aux règles de césure propres à chaque langue.

Pour de nombreuses langues, ces règles différentes peuvent être obtenues en utilisant l'extension `babel` maintenue par Johannes Braams.

Si votre système est configuré pour supporter cette extension, vous pouvez sélectionner la langue à utiliser avec la commande :

```
\usepackage[langue]{babel}
```

après la commande `\documentclass`. Les *langues* supportées par votre système sont normalement listées dans votre *Local Guide* [4].

Pour certaines langues `babel` définit également de nouvelles commandes qui simplifient la saisie des caractères spéciaux.

Pour le français, il existe une extension spécifique : `french`, développé par Bernard Gaulle [10]. En plus des règles de césure spécifiques du français, il adapte L^AT_EX aux règles de la typographie française [11].

Certains systèmes informatiques vous permettent de saisir directement les caractères accentués ou les symboles spécifiques d'une langue (Soit par des touches spéciales, soit par des combinaisons programmées). L^AT_EX 2_ε sait gérer plusieurs codages de ces caractères spéciaux grâce à l'extension `inputenc`⁹. En utilisant cette extension, il faut cependant prendre garde au fait que tous les systèmes informatiques n'utilisent pas le même codage des caractères spéciaux. Il est nécessaire d'y prendre garde si l'on souhaite échanger ses documents.

9. MIT_EX est une autre possibilité pour gérer directement les caractères spéciaux. *NdT*.

TAB. 2.1: *Accents et caractères spéciaux*

ò	\'o	ó	\'o	ô	\^o	õ	\~o
ō	\=o	ò	\.o	ö	\"o	ç	\c c
ö	\u o	ö	\v o	ő	\H o	q	\c o
ø	\d o	ø	\b o	öo	\t oo		
œ	\oe	Œ	\OE	æ	\ae	Æ	\AE
å	\aa	Å	\AA				
ø	\o	Ø	\O	l	\l	L	\L
ı	\i	ı	\j	ı	!'	ı	?'

Terminons ce paragraphe en notant qu'il existe plusieurs outils capables de convertir le codage des accents d'un document source d'un système à l'autre. Pour échanger des documents sur *internet*, il est conseillé d'utiliser le codage standard T_EX (\'e) qui est transmis sans risque d'erreur entre tous les systèmes.

2.4 L'espace entre les mots

Pour obtenir une marge droite alignée, L^AT_EX insère des espaces plus ou moins larges entre les mots.

Un tilde « ~ » produit une espace interdisant tout saut de ligne (dit espace *insécable*). ~ est à utiliser pour éviter les coupures indésirables.

La commande \, permet d'insérer un demi-espace insécable.

```
1234567                1\,234\,567
```

Terminons par une subtilité de la typographie anglo-saxonne : après la ponctuation finale d'une phrase, les règles de la typographie anglo-saxonne veulent que l'on insère une espace plus large. Mais si un point suit une lettre majuscule, L^AT_EX considère qu'il s'agit d'une abréviation et insère alors une espace normale. La commande \@ avant un point indique que celui-ci termine une phrase, même lorsqu'il suit une lettre majuscule.

```
M. Dupont vous remercie      M.~Dupont vous remercie\\
cf. Fig. 5                    cf.~Fig.~5\\
I like basic. Do you?        I like basic\@. Do you?
```

L'ajout d'espace supplémentaire à la fin d'une phrase peut être supprimé par la commande :

```
\frenchspacing
```

qui est active par défaut avec les extensions french ou babel. Dans ce cas, la commande \@ n'est pas nécessaire.

2.5 Titres, chapitres et sections

Pour aider le lecteur à suivre votre pensée, vous souhaitez séparer vos documents en chapitres, sections ou sous-sections. L^AT_EX utilise pour cela des commandes qui prennent en argument le titre de chaque élément. C'est à vous de les utiliser dans l'ordre.

Dans la classe de document `article`, les commandes de sectionnement suivantes sont disponibles :

<code>\section{...}</code>	<code>\paragraph{...}</code>
<code>\subsection{...}</code>	<code>\subparagraph{...}</code>
<code>\subsubsection{...}</code>	<code>\appendix</code>

Dans les classes `report` et `book`, la commande

```
\chapter{...}
```

est également reconnue¹⁰ (elle s'intercale entre `\part` et `\section`).

L'espace entre les sections, la numérotation et le choix de la police et de la taille des titres sont gérés automatiquement par \LaTeX .

Deux commandes de sectionnement ont un comportement spécial :

- la commande `\part` ne change pas la numérotation des chapitres.
- la commande `\appendix` ne prend pas d'argument. Elle bascule simplement la numérotation des chapitres¹¹ en lettres.

\LaTeX peut ensuite créer la table des matières en récupérant la liste des titres et de leur numéro de page d'une exécution précédente (fichier `.toc`). La commande :

```
\tableofcontents
```

imprime la table des matières. Un document doit être traité (on dit aussi « compilé ») deux fois par \LaTeX pour avoir une table des matières correcte. Dans certains cas, un troisième passage est même nécessaire. \LaTeX vous indique quand cela est nécessaire.

Toutes les commandes citées ci-dessus existent dans une forme « étoilée » obtenue en ajoutant une étoile `*` au nom de la commande. Ces commandes produisent des titres de sections qui n'apparaissent pas dans la table des matières et qui ne sont pas numérotés. On peut ainsi remplacer la commande `\section{Introduction}` par `\section*{Introduction}`.

Par défaut, les titres de section apparaissent dans la table des matières exactement comme ils sont dans le texte. Parfois il n'est pas possible de faire tenir un titre trop long dans la table des matières. On peut donner un titre spécifique pour la table des matières en argument optionnel avant le titre principal :

```
\chapter[Le LAAS du CNRS]{Le Laboratoire
d'Analyse et d'Architecture
des Systèmes du Centre National
de la Recherche Scientifique}
```

10. Puisque la classe `article` ne connaît pas les chapitres, il est facile par exemple de regrouper des articles en tant que chapitres d'un livre en remplaçant le `\title` de chaque article par `\chapter`.

11. Pour la classe `article`, elle change la numérotation des sections

Le titre du document est obtenu par la commande :

```
\maketitle
```

Le titre lui-même est défini par les commandes :

```
\title{...}, \author{...} et éventuellement \date{...}
```

qui doivent être appelées avant `\maketitle`. Dans l'argument de la commande `\author`, vous pouvez citer plusieurs auteurs en séparant leurs noms par des commandes `\and`.

Vous trouverez un exemple des commandes citées ci-dessus sur la figure 1.2, page 7.

En plus des commandes de sectionnement expliquées ci-dessus, $\text{\LaTeX} 2_{\epsilon}$ a introduit trois nouvelles commandes destinées à être utilisées avec la classe `book` :

```
\frontmatter, \mainmatter et \backmatter
```

Elles permettent de séparer en trois parties votre ouvrage. Ces commandes modifient les titres des chapitres et la numérotation des pages comme cela se fait dans un livre.

2.6 Références croisées

Dans les livres, rapports ou articles, on trouve souvent des références croisées vers des figures, des tableaux ou des passages particuliers du texte. \LaTeX dispose des commandes suivantes pour faire des références croisées :

```
\label{marque}, \ref{marque} et \pageref{marque}
```

où *marque* est un identificateur choisi par l'utilisateur. \LaTeX remplace `\ref` par le numéro de la section, de la sous-section, de la figure, du tableau, ou du théorème où la commande `\label` correspondante a été placée. L'utilisation de références croisées rend nécessaire de compiler deux fois le document : à la première compilation les numéros correspondant aux étiquettes `\label{}` sont inscrits dans le fichier `.aux`, et à la compilation suivante `\ref{}` et `\pageref{}` peuvent imprimer ces numéros.

Une référence à cette section ressemble à : « voir section 2.6, page 21. »

```
Une r\ef\erence \a cette
section\label{ma-section}
ressemble \a :
<< voir section~\ref{ma-section},
page~\pageref{ma-section}. >>
```

2.7 Notes de bas de page

La commande :

```
\footnote{texte}
```

imprime une note de bas de page en bas de la page en cours.

Les notes de bas de page^a sont très prisées par les utilisateurs de L^AT_EX.

Les notes de bas de page%
`\footnote{ceci est une note de bas de page.}`
sont tr[\]'es pris[\]'ees par les utilisateurs de `\LaTeX{}`.

^a ceci est une note de bas de page.

2.8 Souligner l'importance d'un mot

Dans un manuscrit réalisé sur une machine à écrire, les mots importants sont soulignés. Dans un ouvrage imprimé, ils sont *mis en valeur*¹². La commande de mise en valeur est :

```
\emph{texte}
```

Son argument est le texte à mettre en valeur. En général, la police *italique* est utilisée pour la mise en valeur, sauf si le texte est déjà en italique. Ce comportement dépend de la définition de la classe de document.

Si vous voulez souligner l'importance d'un mot dans un passage déjà mis en valeur, L^AT_EX utilise alors une police droite.

```
\emph{Si vous voulez souligner l'importance d'un mot dans un passage d'ej mis en valeur, \LaTeX{ utilise alors une police droite.}}
```

2.9 Environnements

Pour composer du texte dans des contextes spécifiques, L^AT_EX définit des environnements différents pour divers types de mise en page :

```
\begin{nom} texte \end{nom}
```

nom est le nom de l'environnement. Les environnements peuvent être imbriqués .

```
\begin{aaa}... \begin{bbb}... \end{bbb}... \end{aaa}
```

12. Emphasized en anglais.

Dans les sections suivantes tous les environnements importants sont présentés.

2.9.1 Listes, énumérations et descriptions

L'environnement `itemize` est utilisé pour des listes simples, `enumerate` est utilisé pour des énumérations (listes numérotées) et `description` est utilisé pour des descriptions.

- | | |
|--|--|
| <p>1. vous pouvez imbriquer les environnements à votre guise :</p> <ul style="list-style-type: none"> • mais cela peut ne pas être très beau • ni facile à suivre. <p>2. Souvenez-vous :</p> <p>Clarté : les faits ne vont pas devenir plus sensés parce qu'ils sont dans une liste.</p> <p>Synthèse : cependant une liste peut très bien résumer des faits.</p> | <pre> \begin{enumerate} \item vous pouvez imbriquer les environnements \`a votre guise : \begin{itemize} \item mais cela peut ne pas \^etre tr\`es beau \item ni facile \`a suivre. \end{itemize} \item Souvenez-vous : \begin{description} \item[Clart\`e :] les faits ne vont pas devenir plus sens\`es parce qu'ils sont dans une liste. \item[Synth\`ese :] cependant une liste peut tr\`es bien r\`e-su-mer des faits. \end{description} \end{enumerate} </pre> |
|--|--|

Notez que l'extension `french` utilise une présentation des listes simples qui respecte les règles typographiques françaises :

- | | |
|---|--|
| <p>Une liste simple française :</p> <ul style="list-style-type: none"> – voici un élément – puis un autre. <p>An english list:</p> <ul style="list-style-type: none"> • one item • an other one | <p>Une liste simple fran\c{c}aise :</p> <pre> \begin{itemize} \item voici un \`el\`ement \item puis un autre. \end{itemize} An english list: \begin{itemize} \item one item \item an other one \end{itemize} </pre> |
|---|--|

2.9.2 Alignements à gauche, à droite et centrage

Les environnements `flushleft` et `flushright` produisent des textes alignés à gauche ou à droite. L'environnement `center` produit un texte

centré. Si vous n'utilisez pas la commande `\\` pour indiquer les sauts de ligne, ceux-ci sont calculés automatiquement par `LATEX`.

Ce texte est aligné à gauche. <code>L^AT_EX</code> n'essaye pas d'aligner la marge droite.	<code>\begin{flushleft}</code> Ce texte est\\ <code>align\`e \`a gauche.</code> <code>\LaTeX{}</code> n'essaye pas d'aligner la marge droite. <code>\end{flushleft}</code>
--	---

Ce texte est aligné à droite. <code>L^AT_EX</code> n'essaye pas d'aligner la marge gauche.	<code>\begin{flushright}</code> Ce texte est\\ <code>align\`e \`a droite.</code> <code>\LaTeX{}</code> n'essaye pas d'aligner la marge gauche. <code>\end{flushright}</code>
--	---

Au centre de la terre.	<code>\begin{center}</code> Au centre de la terre. <code>\end{center}</code>
------------------------	--

2.9.3 Citations et vers

L'environnement `quote` est utile pour les citations, les phrases importantes ou les exemples.

Une règle typographique simple pour la longueur des lignes :	Une règle typographique simple pour la longueur des lignes :
Une ligne ne devrait pas comporter plus de 66 caractères.	<code>\begin{quote}</code> Une ligne ne devrait pas comporter plus de 66~caract\`eres.
C'est pourquoi les pages composées par <code>L^AT_EX</code> ont des marges importantes.	C'est pourquoi les pages compos\`ees par <code>\LaTeX</code> ont des marges importantes.
Cela explique pourquoi les journaux utilisent souvent plusieurs colonnes.	<code>\end{quote}</code> Cela explique pourquoi les journaux utilisent souvent plusieurs colonnes.

Il existe deux autres environnements comparables : `quotation` et `verse`. L'environnement `quotation` est utile pour des citations plus longues, couvrant plusieurs paragraphes. L'environnement `verse` est utilisé pour la poésie, là où les retours à la ligne sont importants. Les vers sont séparés par des commandes `\\` et les strophes par une ligne vide.

Voici le début d'une fugue de Boris Vian :

```
Les gens qui n'ont plus
rien à faire
Se suivent dans la rue
comme
Des wagons de chemin
de fer.
Fer fer fer
Fer fer fer
Fer quoi faire
Fer coiffeur.
```

```
Voici le d\ebut d'une
fugue de Boris Vian\,:
\begin{flushleft}
\begin{verse}
Les gens qui n'ont plus
rien \a faire\\
Se suivent dans la rue comme\\
Des wagons de chemin de fer.

Fer fer fer\\
Fer fer fer\\
Fer quoi faire\\
Fer coiffeur.\\
\end{verse}
\end{flushleft}
```

2.9.4 Impression *verbatim*

Tout texte inclus entre `\begin{verbatim}` et `\end{verbatim}` est imprimé tel quel, comme s'il avait été tapé à la machine, avec tous les retours à la ligne et les espaces, sans qu'aucune commande \LaTeX ne soit exécutée.

À l'intérieur d'un paragraphe, une fonctionnalité équivalente peut être obtenue par

`\verb+texte+`

Le caractère + est seulement un exemple de caractère séparateur. Vous pouvez utiliser n'importe quel caractère, sauf les lettres, * ou l'espace. La plupart des exemples de commandes \LaTeX dans ce document sont réalisés avec cette commande.

La commande `\ldots` ...

```
10 PRINT "HELLO WORLD ";
20 GOTO 10
```

La commande `\verb|\ldots| \ldots`

```
\begin{verbatim}
10 PRINT "HELLO WORLD ";
20 GOTO 10
\end{verbatim}
```

La_{version}\‘etoile\‘e_{de}
l'environnement_{verbatim}
met_{les}es_{espaces}
\‘evidence

```
\begin{verbatim*}
La version \‘etoile\‘e de
l'environnement verbatim
met les espaces en
\‘evidence
\end{verbatim*}
```

La commande `\verb` peut également être utilisée avec une étoile :

```
comme_ceci:-) | \verb*|comme ceci :-) |
```

L'environnement `verbatim` et la commande `\verb` ne peuvent être utilisés à l'intérieur des paramètres d'autres commandes.

2.9.5 Tableaux

L'environnement `tabular` permet de réaliser des tableaux avec ou sans lignes de séparation horizontales ou verticales. \LaTeX ajuste automatiquement la largeur des colonnes.

L'argument *description du tableau* de la commande :

```
\begin{tabular}{description du tableau}
```

définit le format des colonnes du tableau. Utilisez un `l` pour une colonne de texte aligné à gauche, `r` pour du texte aligné à droite et `c` pour du texte centré. `p{largeur}` permet de réaliser une colonne de texte justifié à droite sur plusieurs lignes, et enfin `|` permet d'obtenir une ligne verticale.

À l'intérieur de l'environnement `tabular`, le caractère `&` est le séparateur de colonnes, `\\` commence une nouvelle ligne et `\hline` insère une ligne horizontale.

7C0	hexadécimal
3700	octal
11111000000	binaire
1984	décimal

```
\begin{tabular}{|r|l|}
\hline
7C0 & hexad\`ecimal \\
3700 & octal \\
11111000000 & binaire \\
\hline \hline
1984 & d\`ecimal \\
\hline
\end{tabular}
```

Bienvenue dans ce cadre. Merci de votre visite.
--

```
\begin{tabular}{|p{4.7cm}|}
\hline
Bienvenue dans ce
cadre.\\
Merci de votre visite.\\
\hline
\end{tabular}
```

La construction `@{...}` permet d'imposer le séparateur de colonnes. Cette commande supprime l'espacement inter-colonnes et le remplace par ce qui est indiqué entre les crochets. Une utilisation courante de cette commande est présentée plus loin comme solution au problème de l'alignement

des nombres décimaux. Une autre utilisation possible est de supprimer l'espace dans un tableau avec `@{}`.

sans espace	<pre>\begin{tabular}{@{} l @{}} \hline sans espace\\ \hline \end{tabular}</pre>
-------------	---

avec espaces	<pre>\begin{tabular}{l} \hline avec espaces\\ \hline \end{tabular}</pre>
--------------	--

Comme il n'y a pas de commande prévue¹³ pour aligner les nombres sur le point décimal (ou la virgule si on respecte les règles françaises) nous pouvons « tricher » et réaliser cet alignement en utilisant deux colonnes : la première alignée à droite contient la partie entière et la seconde alignée à gauche contient la partie décimale. La commande `\@{.}` dans la description du tableau remplace l'espace normale entre les colonnes par un simple « . », donnant l'impression d'une seule colonne alignée sur le point. N'oubliez pas de remplacer dans votre tableau le point par un séparateur de colonnes (`&`) ! Un label peut être placé au-dessus de cette colonne en utilisant la commande `\multicolumn`.

expression	Valeur	<pre>\begin{tabular}{c r @{,} l} expression & & \\ \multicolumn{2}{c}{Valeur} & \\ \hline \$\pi\$ & 3&1416 & \\ \$\pi^{\pi}\$ & 36&46 & \\ \$(\pi^{\pi})^{\pi}\$ & 80662&7 & \\ \end{tabular}</pre>
------------	--------	---

2.10 Objets flottants

De nos jours, la plupart des publications contiennent un nombre important de figures et de tableaux. Ces éléments nécessitent un traitement particulier car ils ne peuvent être coupés par un changement de page. On pourrait imaginer de commencer une nouvelle page chaque fois qu'une figure ou un tableau ne rentrerait pas dans la page en cours. Cette façon de faire

13. Si les extensions de l'ensemble « tools » sont installées sur votre système, jetez un œil sur l'extension `dcolumn` faite pour résoudre ce problème.

laisserait de nombreuses pages à moitié blanches, ce qui ne serait réellement pas beau.

La solution est de laisser « flotter » les figures et les tableaux qui ne rentrent pas sur la page en cours, vers une page suivante et de compléter la page avec le texte qui suit l'objet « flottant ». \LaTeX fournit deux environnements pour les objets flottants adaptés respectivement aux figures (`figure`) et aux tableaux (`table`). Pour faire le meilleur usage de ces deux environnements, il est important de comprendre comment \LaTeX traite ces objets flottants de manière interne. Dans le cas contraire ces objets deviendront une cause de frustration intense car \LaTeX ne les placera jamais à l'endroit où vous souhaitiez les voir.

Commençons par regarder les commandes que \LaTeX propose pour les objets flottants :

Tout objet inclus dans un environnement `figure` ou `table` est traité comme un objet flottant. Les deux environnements flottants ont un paramètre optionnel :

`\begin{figure}[placement]` ou `\begin{table}[placement]`

appelé *placement*. Ce paramètre permet de dire à \LaTeX où vous autorisez l'objet à flotter. Un *placement* est composé d'une chaîne de caractères représentant des *placements possibles*. Reportez-vous au tableau 2.2.

TAB. 2.2: *Placements possibles*

Caractère	Emplacement pour l'objet flottant ...
<code>h</code>	<i>here</i> , ici, à l'emplacement dans le texte où la commande se trouve. Utile pour les petits objets.
<code>t</code>	<i>top</i> , en haut d'une page
<code>b</code>	<i>bottom</i> , en bas d'une page
<code>p</code>	<i>page</i> , sur une page à part ne contenant que des objets flottants.
<code>!</code>	ici, sans prendre en compte les paramètres internes ^a qui pourraient empêcher ce placement.

^a tels que le nombre maximum d'objets flottants sur une page

Un tableau peut commencer par exemple par la ligne suivante :

```
\begin{table}[!hbp]
```

L'emplacement `[!hbp]` permet à \LaTeX de placer le tableau soit sur place (`h`), soit en bas de page (`b`) soit enfin sur une page à part (`p`), et tout cela même si les règles internes de \LaTeX ne sont pas toutes respectées (`!`).

Si aucun placement n'est indiqué, les classes standard utilisent `[tbp]` par défaut.

\LaTeX place tous les objets flottants qu'il rencontre dans l'ordre en suivant les indications fournies par l'auteur. Si un objet ne peut être placé sur la page en cours, il est placé soit dans la file des figures soit dans la file des tableaux¹⁴. Quand une nouvelle page est entamée, \LaTeX essaye d'abord de voir si les objets en tête des deux files pourraient être placés sur une page spéciale, à part. Si cela n'est pas possible, les objets en tête des deux files sont traités comme s'ils venaient d'être trouvés dans le texte : \LaTeX essaye de les placer selon les placements possibles restants. Tous les nouveaux objets flottants rencontrés dans la suite du texte sont ajoutés à la queue des files. \LaTeX respecte scrupuleusement l'ordre d'apparition des objets flottants. C'est pourquoi un objet flottant qui ne peut être placé dans le texte repousse toutes les autres à la fin du document.

D'où la règle :

Si \LaTeX ne place pas les objets flottants comme vous le souhaitez, c'est souvent à cause d'un seul objet trop grand qui bouche l'une des deux files d'objets flottants.

Après avoir rapidement expliqué cette difficulté, voici quelques éléments supplémentaires qu'il est bon de connaître sur les environnements `table` et `figure`.

Avec la commande :

```
\caption{texte de la légende}
```

vous définissez une légende pour l'objet. Un numéro (incrémenté automatiquement) et le mot « Figure » ou « Table »¹⁵ sont ajoutés par \LaTeX .

Les deux commandes :

```
\listoffigures et \listoftables
```

fonctionnent de la même manière que la commande `\tableofcontents` ; elles impriment respectivement la liste des figures et des tableaux. Dans ces listes, la légende est reprise en entier. Si vous désirez utiliser des légendes longues, vous pouvez en donner une version courte entre crochets qui sera utilisée pour la table.

```
\caption[courte]{LLLLLoooooonnnnnggggguuuueee}
```

Avec `\label` et `\ref` vous pouvez faire référence à votre objet à l'intérieur de votre texte.

14. Il s'agit de files FIFO (*First In, First Out*) : premier arrivé, premier servi.

15. Avec l'extension `french`, la présentation des légendes est modifiée pour obéir aux règles françaises.

L'exemple suivant dessine un carré et l'insère dans le document. Vous pouvez utiliser cette commande pour réserver de la place pour une illustration que vous allez coller sur le document terminé.

```
La figure~\ref{blanche} est un exemple de Pop-Art.
\begin{figure}[!hbp]
\makebox[\textwidth]{\framebox[5cm]{\rule{0pt}{5cm}}}
\caption{Cinq centimètres sur cinq} \label{blanche}
\end{figure}
```

Dans l'exemple ci-dessus¹⁶ \LaTeX va s'acharner (!) à placer la figure là où se trouve la commande (h) dans le texte. S'il n'y arrive pas, il essaiera de la placer en bas (b) de la page. Enfin s'il ne peut la placer sur la page courante, il essaiera de créer une page à part avec d'autres objets flottants. S'il n'y a pas suffisamment de tableaux en attente pour remplir une page spécifique, \LaTeX continue et, au début de la page suivante, réessayera de placer la figure comme si elle venait d'apparaître dans le texte.

Dans certains cas il peut s'avérer nécessaire d'utiliser la commande :

`\clearpage` ou même `\cleardoublepage`

Elle ordonne à \LaTeX de placer tous les objets en attente immédiatement puis de commencer une nouvelle page. `\cleardoublepage` commence une nouvelle page de droite.

À la section 4.6, vous apprendrez à inclure une figure POSTSCRIPT dans votre document $\LaTeX 2_{\epsilon}$.

2.11 Nouvelles commandes et nouveaux environnements

Dans le premier chapitre, nous avons vu que \LaTeX avait besoin de connaître la structure logique d'un document pour pouvoir le formater correctement. Ceci est une idée intéressante, mais en pratique on atteint assez vite ses limites quand on ne trouve pas dans \LaTeX la commande spécialisée pour exprimer ce que l'on voudrait.

Une solution serait de combiner plusieurs commandes \LaTeX pour obtenir le résultat désiré. Ponctuellement, cela ne pose pas de problème ; mais si cela se reproduit, répéter chaque fois la même combinaison peut devenir long et fastidieux. De plus, si vous souhaitez plus tard modifier la présentation, il faudra aller modifier chaque combinaison de commandes tout au long du texte.

Pour résoudre ce problème, \LaTeX vous permet de définir vos propres commandes et vos propres environnements.

¹⁶. En supposant que la file des figures soit vide.

2.11.1 Nouvelles commandes

Pour ajouter de nouvelles commandes, utilisez la commande :

```
\newcommand{nom}[num]{définition}
```

Cette commande prend principalement deux arguments : le *nom* de la commande à créer et sa *définition*. L'argument *num* entre crochets est optionnel. Il indique le nombre de paramètres qu'utilisera la nouvelle commande (jusqu'à 9).

Les deux exemples ci-dessous vous aiderons à saisir le principe. Le premier exemple définit une nouvelle commande appelée `\ucil` qui est une abréviation d'« une courte introduction à \LaTeXe ». Une telle commande pourrait être utile si vous aviez à citer de nombreuses fois le titre de ce livre.

```
Une courte (?) introduction à \newcommand{\ucil}
\LaTeXe ... Une courte (?) intro- {Une courte (?)
duction à \LaTeXe                introduction \‘a \LaTeXe}
                                % dans le document :
                                \ucil \ldots{} \ucil
```

L'exemple suivant montre comment utiliser l'argument *num*. La séquence `#1` est remplacée par le paramètre que vous passez. Pour utiliser plus d'un paramètre, continuez avec `#2`, etc.

```
- Une courte introduction à \newcommand{\uil}[1]
  \LaTeXe                    {Une \emph{#1}
                              introduction \‘a \LaTeXe}
- Une petite introduction à \begin{itemize}
  \LaTeXe                    \item \uil{courte}
                              \item \uil{petite}
                              \end{itemize}
```

\LaTeX ne vous permet pas de créer une nouvelle commande si celle-ci existe déjà. Si vous voulez explicitement remplacer une commande existante, utilisez `\renewcommand`. Elle utilise la même syntaxe que `\newcommand`. Dans certains cas, vous pouvez avoir besoin de `\providecommand`. Cette commande fonctionne comme `\newcommand`, mais si la nouvelle commande est déjà définie, \LaTeXe ignore la nouvelle définition.

2.11.2 Nouveaux environnements

De manière analogue à la commande `\newcommand`, il existe une commande pour définir de nouveaux environnements. Pour rédiger ce document, elle a été utilisée plusieurs fois pour définir des environnements utilisés

dans cette présentation : « exemples », « morceaux de code » et « boîtes de définition de commandes ». La commande `\newenvironment` se présente de la manière suivante :

```
\newenvironment{nom}[num]{avant}{après}
```

De même que pour `\newcommand`, vous pouvez utiliser `\newenvironment` avec un paramètre optionnel ou sans. Le contenu de l'argument *avant* est exécuté avant que le contenu de l'environnement ne soit traité. Le contenu de l'argument *après* est traité lorsque l'on rencontre la commande `\end{nom}`.

L'exemple ci-dessous illustre l'utilisation de `\newenvironment`.

```
Texte encadré et centré!
```

```
\newenvironment{cboxed}%
  {\begin{center}
   \begin{tabular}{|l|} \hline}%
  {\ \ \hline \end{tabular}
   \end{center}}
%% dans le texte :
\begin{cboxed}
Texte encadr\'e et centr\'e !
\end{cboxed}
```

L'argument *num* est utilisé de la même façon que pour la commande `\newcommand`. \LaTeX vous empêche de redéfinir un environnement qui existe déjà. Si jamais vous vouliez redéfinir un environnement existant, utilisez `\renewenvironment` qui utilise la même syntaxe que `\newenvironment`.

Chapitre 3

Formules Mathématiques

Vous êtes prêts! Dans ce chapitre nous allons aborder l'atout majeur de \TeX : la composition de formules mathématiques. Mais attention, ce chapitre ne fait que décrire les commandes de base. Bien que ce qui est expliqué ici soit suffisant pour la majorité des utilisateurs, ne désespérez pas si vous n'y trouvez pas la solution à votre problème de mise en forme d'une équation mathématique. Il y a de fortes chances pour que la solution se trouve dans l'une des extensions de AMS¹- \TeX .

3.1 Généralités

\TeX dispose d'un mode spécial pour la mise en page de formules mathématiques. Des maths à l'intérieur d'un paragraphe sont saisies entre $\backslash($ et $\backslash)$, entre $\$$ et $\$$ ou entre $\backslash\text{begin}\{\text{math}\}$ et $\backslash\text{end}\{\text{math}\}$.

Ajoutez a au carré et b au carré pour obtenir c au carré. Ou, en utilisant une approche plus matheuse: $c^2 = a^2 + b^2$

Ajoutez $\$a\$$ au carré et $\$b\$$ au carré pour obtenir $\$c\$$ au carré. Ou, en utilisant une approche plus matheuse : $\$c^{\{2\}}=a^{\{2\}}+b^{\{2\}}\$$

100 m³ d'eau

100 $\backslash\text{m}\$^{\{3\}}\$$ d'eau

J'♥ \TeX

J' $\backslash\text{\heartsuit}\$ \backslash\text{LaTeX}\{\}$

Il vaut mieux composer les équations ou les formules plus importantes « *hors-texte* », c'est-à-dire sur des lignes à part. Pour cela, on les inclut entre $\backslash[$ et $\backslash]$ ou entre $\backslash\text{begin}\{\text{displaymath}\}$ et $\backslash\text{end}\{\text{displaymath}\}$. On obtient

1. American Math Society = Société Américaine de Mathématiques. De nombreuses extensions présentés dans ce chapitre sont développées par l'AMS.

ainsi des formules qui ne sont pas numérotées. Si vous voulez qu'elles soient numérotées par L^AT_EX, utilisez l'environnement `equation`.

Ajoutez a au carré et b au carré pour obtenir c au carré. Ou, en utilisant une approche plus matheuse :

$$c^2 = a^2 + b^2$$

Le mot de la fin.

Ajoutez `a` au carré et `b` au carré pour obtenir `c` au carré. Ou, en utilisant une approche plus matheuse :

```
\begin{displaymath}
c^2=a^2+b^2
\end{displaymath}
Le mot de la fin.
```

Avec `\label` et `\ref`, vous pouvez faire référence à une équation.

$$\epsilon > 0 \quad (3.1)$$

L'équation (3.1) nous donne...

```
\begin{equation} \label{eq}
\epsilon > 0
\end{equation}
L'équation (\ref{eq})
nous donne...
```

Remarquez que les expressions mathématiques sont formatées différemment selon qu'elles sont composées « en ligne » ou « hors texte ». Comparez :

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

```
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
```

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

```
\begin{displaymath}
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
\end{displaymath}
```

Il y a des différences entre le mode *mathématique* et le mode *texte*. Par exemple, en mode *mathématique* :

1. la plupart des espaces et des retours à la ligne n'ont aucune signification. Les espaces sont déduites de la logique de la formule ou indiqués à l'aide de commandes spécifiques telles que `\,`, `\quad` ou `\qquad`.

$$\forall x \in \mathbf{R} : \quad x^2 \geq 0 \quad (3.2)$$

```
\begin{equation}
\forall x \in \mathbf{R} :
\quad x^2 \geq 0
\end{equation}
```

2. les lignes vides ne sont pas autorisées. Un seul paragraphe par formule.

3. chaque lettre est considérée comme étant le nom d'une variable et sera imprimée comme tel. Pour insérer du texte normal (police et espacement standard) dans une formule, il faut utiliser la commande `\textrm{...}`.

$$x^2 \geq 0 \quad \text{pour tout } x \in \mathbf{R} \quad (3.3)$$

```

\begin{equation}
x^{2} \geq 0 \quad \text{pour tout } x \in \mathbf{R}
\end{equation}

```

Une mode récente et contestable pousse à utiliser la police « blackboard bold » (Gras Tableau Noir, ainsi appelée car c'est par le doublement des verticales des lettres que l'on simule le gras typographique lorsqu'on ne peut faire autrement) qui est obtenue par la commande `\mathbb` de l'extension `amsfonts` ou `amssymb` pour désigner les ensembles de nombres entiers, réels, etc.

L'exemple précédent devient :

$$x^2 \geq 0 \quad \text{pour tout } x \in \mathbb{R}$$

```

\begin{displaymath}
x^{2} \geq 0 \quad \text{pour tout } x \in \mathbb{R}
\end{displaymath}

```

3.2 Groupements en mode mathématique

La plupart des commandes du mode mathématique ne s'applique qu'au caractère suivant. Pour qu'une commande s'applique à un ensemble de caractères, il faut les grouper en utilisant des accolades : `{...}`.

$$a^x + y \neq a^{x+y} \quad (3.4)$$

```

\begin{equation}
a^{x+y} \neq a^{x+y}
\end{equation}

```

3.3 Éléments d'une formule mathématique

Dans cette section nous allons voir les commandes les plus importantes du mode mathématique. Pour une liste de tous les symboles disponibles, voyez la section 3.11, page 46.

Les lettres **grecques minuscules** sont saisies de la manière suivante : `\alpha`, `\beta`, `\gamma`, etc. Les lettres **grecques majuscules**² sont saisies ainsi : `\Gamma`, `\Delta`, etc.

$\lambda, \xi, \pi, \mu, \Phi, \Omega$ `\lambda, \xi, \pi, \mu, \Phi, \Omega`

Les **indices et exposants** sont positionnés en utilisant les caractères `_` et `^`.

a_1 x^2 $e^{-\alpha t}$ a_{ij}^3 `a_{1} \quad x^{2} \quad $e^{-\alpha t}$ \quad a^3_{ij} \quad`
 $e^{x^2} \neq (e^x)^2$ `$e^{x^2} \neq (e^x)^2$`

La **racine carrée** est saisie ainsi : `\sqrt`, la racine $n^{\text{ième}}$ est produite par la commande `\sqrt[n]`. La taille du symbole racine est calculée par L^AT_EX. Pour obtenir le symbole seul, utilisez `\surd`.

\sqrt{x} $\sqrt{x^2 + \sqrt{y}}$ $\sqrt[3]{2}$ `\sqrt{x}` `\sqrt{x^2 + \sqrt{y}}` `\sqrt[3]{2}`
 $\sqrt{x^2 + y^2}$ `\sqrt{x^2 + y^2}` `\surd[x^2 + y^2]`

Les commandes `\overline` et `\underline` créent un **trait horizontal** au-dessus ou au-dessous d'une expression.

$\overline{m+n}$ `\overline{m+n}`

Les commandes `\overbrace` et `\underbrace` créent une grande **accolade horizontale** au-dessus ou au-dessous d'une expression.

$\underbrace{a + b + \dots + z}_{26}$ `\underbrace{ a+b+\cdots+z }_{26}`

Pour ajouter des accents mathématiques tels que des flèches ou des tildes, vous pouvez utiliser les commandes du tableau 3.1. Les chapeaux et les tildes larges, couvrant plusieurs caractères, sont produits par les commandes `\widetilde` et `\widehat`. La commande `'` produit un prime.

$y = x^2$ $y' = 2x$ $y'' = 2$ `\begin{displaymath}`
`y=x^{2}\quad y'=2x\quad y''=2`
`\end{displaymath}`

2. Il n'y a pas de Alpha majuscule dans L^AT_EX 2_ε parce que c'est le même caractère que le A romain. Lorsque le nouveau codage mathématique sera terminé, cela changera.

Les **vecteurs** sont en général marqués en ajoutant une flèche au-dessus du nom de la variable. Ceci est obtenu par la commande `\vec`. Pour coder le vecteur de A à B , les commandes `\overrightarrow` et `\overleftarrow` sont bien utiles.

$$\vec{a} \quad \overrightarrow{AB}$$

```
\begin{displaymath}
\vec{a} \quad \overrightarrow{AB}
\end{displaymath}
```

Les noms des fonctions telles que `log` doivent être imprimés à l'aide d'une police droite et non en italique comme les variables. \LaTeX fournit donc les commandes suivantes pour les fonctions les plus utilisées :

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

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

```
\[\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1\]
```

Pour la fonction modulo, il y a deux commandes possibles : `\bmod` pour l'opérateur binaire et `\pmod` pour l'opérateur unaire :

$$a \bmod b \quad x \equiv a \pmod{b}$$

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

Un trait de **fraction** est produit par la commande :

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

La forme utilisant un *slash* ($1/2$) est souvent préférable pour des petits éléments.

$$1\frac{1}{2} \text{ hours} \quad \frac{x^2}{k+1} \quad x^{\frac{2}{k+1}} \quad x^{1/2}$$

```
\$1\frac{1}{2}\$~hours
\begin{displaymath}
\frac{x^2}{k+1} \quad x^{\frac{2}{k+1}} \quad x^{1/2}
\end{displaymath}
```

Pour imprimer des coefficients binomiaux (à l'américaine) ou d'autres structures semblables, on peut utiliser la commande `\binom` de l'extension `amsmath` :

$$\binom{n}{k} C_n^k$$

```
\begin{displaymath}
\binom{n}{k} \quad \mathrm{C}_n^k
\end{displaymath}
```

Les **intégrales** sont produites par la commande `\int`, les **sommes** par la commande `\sum`. Les limites inférieures et supérieures sont indiquées avec `_` et `^` comme pour les indices et les exposants.

$$\sum_{i=1}^n \int_0^{\frac{\pi}{2}}$$

```
\begin{displaymath}
\sum_{i=1}^n \quad \int_0^{\frac{\pi}{2}}
\end{displaymath}
```

Pour superposer des indices, l'extension `amsmath` propose la commande `\substack` et l'environnement `subarray` qui permet d'aligner les indices à gauche au lieu de les centrer.

$$\sum_{\substack{0 < i < n \\ 1 < j < m}} P(i, j) = \sum_{\substack{i \in I \\ 1 < j < m}} Q(i, j)$$

```
\begin{displaymath}
\sum_{\substack{0 < i < n \\ 1 < j < m}} P(i, j) =
\sum_{\substack{i \in I \\ 1 < j < m}} Q(i, j)
\end{displaymath}
```

Pour les **crochets et les autres délimiteurs**, il existe toutes sortes de symboles en `TEX` (par exemple `[< || ↓`). Les parenthèses et les crochets sont obtenus avec les caractères correspondants, les accolades avec `\{`, et tous les autres délimiteurs sont obtenus par des commandes spéciales (par exemple `\updownarrow`). Pour une liste de tous les délimiteurs disponibles, reportez-vous au tableau 3.8, page 48.

$$a, b, c \neq \{a, b, c\}$$

```
\begin{displaymath}
\{a, b, c\} \neq \{a, b, c\}
\end{displaymath}
```

Si vous ajoutez la commande `\left` avant un délimiteur ouvrant ou `\right` avant un délimiteur fermant, `TEX` détermine automatiquement la taille appropriée pour ce caractère. Remarquez qu'il est nécessaire de fermer chaque délimiteur ouvrant (`\left`) avec un délimiteur fermant (`\right`). Si vous ne voulez pas de délimiteur fermant, utilisez le délimiteur invisible `\right.!`

$$1 + \left(\frac{1}{1-x^2} \right)^3$$

```
\begin{displaymath}
1 + \left( \frac{1}{1-x^2} \right)^3
\end{displaymath}
```

Dans certains cas, il est nécessaire d'indiquer la taille exacte des délimiteurs mathématiques à la main. Vous pouvez alors utiliser les commandes `\big`, `\Big`, `\bigg` et `\Bigg` comme préfixes des commandes qui impriment les délimiteurs³.

$$\left((x+1)(x-1) \right)^2$$

```
$$\Big( (x+1) (x-1) \Big)^2$$\
$\big(\Big(\bigg(\Bigg(\quad
\big)\Big)\bigg)\Big)\quad
\big\|\Big\|\bigg\|\Big\|\$
```

Pour saisir des **points de suspension** dans une formule, vous pouvez utiliser plusieurs commandes. `\ldots` imprime les points sur la base de la ligne, `\cdots` les imprime au milieu. En plus il y a les commandes `\vdots` pour les imprimer verticalement et `\ddots` pour les imprimer en diagonale. Vous trouverez un autre exemple dans la section 3.5.

$$x_1, \dots, x_n \quad x_1 + \cdots + x_n$$

```
\begin{displaymath}
x_{1}, \ldots, x_{n} \quad x_{1} + \cdots + x_{n}
\end{displaymath}
```

3.4 Espacement en mode mathématique

Si l'espacement choisi par $\text{T}_{\text{E}}\text{X}$ dans une formule n'est pas satisfaisant, il peut être ajusté en insérant des commandes d'espacement. Les plus importantes sont : `\`, pour une petite espace, `_` pour une espace de taille moyenne (`_` représente le caractère « espace »), `\quad` et `\qquad` pour des espaces plus larges et `\!` qui réduit une espace.

3. Ces commandes ne fonctionnent pas correctement après une commande de changement de taille ou si les options `11pt` ou `12pt` sont utilisées. Les extensions `exscale` ou `amsmath` permettent d'avoir le comportement attendu.

$$\iint_D g(x, y) \, dx \, dy$$

plutôt que

$$\int \int_D g(x, y) \, dx \, dy$$

```

\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\int\!\!\!\int_{D} g(x,y)
\ , \ \ud x\ , \ \ud y
\end{displaymath}
plutôt que
\begin{displaymath}
\int\int_{D} g(x,y)\ud x \ud y
\end{displaymath}

```

Remarquez que le ‘d’ de l’élément différentiel est traditionnellement imprimé en caractères romains par la commande `\ud`.

3.5 Alignements verticaux

Pour imprimer des **matrices**, utilisez l’environnement `array`. Il fonctionne de manière similaire à l’environnement `tabular`. La commande `\\` est utilisée pour séparer les lignes.

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots \\ x_{21} & x_{22} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

```

\begin{displaymath}
\mathbf{X} =
\left( \begin{array}{ccc}
x_{11} & x_{12} & \ldots \\
x_{21} & x_{22} & \ldots \\
\vdots & \vdots & \ddots
\end{array} \right)
\end{displaymath}

```

L’environnement `array` peut également être utilisé pour imprimer des expressions qui ont un délimiteur invisible obtenu par la commande `\right. :`

$$y = \begin{cases} a & \text{si } d > c \\ b + x & \text{le matin} \\ l & \text{la journée} \end{cases}$$

```

\begin{displaymath}
y = \left\{ \begin{array}{ll}
a & \text{\texttrm{si } $d>c$} \\
b+x & \text{\texttrm{le matin}} \\
l & \text{\texttrm{la journée}}
\end{array} \right.
\end{displaymath}

```

Pour les formules qui prennent plusieurs lignes ou pour des systèmes d’équations, utilisez les environnements `eqnarray` et `eqnarray*` plutôt que `equation`. Avec `eqnarray` chaque ligne est numérotée, alors que la variante `eqnarray*` ne produit aucun numéro.

Les environnements `eqnarray` et `eqnarray*` se comportent comme un tableau à trois colonnes de la forme `{rcl}`, où la colonne centrale peut être

utilisée pour le signe égal, ou tout autre opérateur relationnel de votre choix. La commande `\` sépare les lignes.

$$\begin{array}{rcl}
 f(x) & = & \cos x \quad (3.5) \\
 f'(x) & = & -\sin x \quad (3.6) \\
 \int_0^x f(y) dy & = & \sin x \quad (3.7)
 \end{array}$$

```

\begin{eqnarray}
f(x) & = & \cos x \\
f'(x) & = & -\sin x \\
\int_0^x f(y) dy & = & \sin x
\end{eqnarray}

```

On peut considérer qu'il y a trop d'espace de part et d'autre de la colonne centrale, autour des signes égal. Il peut être réduit par la commande `\setlength\arraycolsep{2pt}` comme dans l'exemple suivant.

Les **équations longues** ne sont pas découpées automatiquement en morceaux harmonieux. L'auteur doit indiquer où les couper et comment indenter la suite. Les deux méthodes ci-dessous sont les plus courantes pour obtenir le résultat attendu.

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \quad (3.8)$$

```

{\setlength\arraycolsep{2pt}
\begin{eqnarray}
\sin x & = & x - \frac{x^3}{3!} \\
& & + \frac{x^5}{5!} - \\
& & \frac{x^7}{7!} + \dots
\end{eqnarray}}

```

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots \quad (3.9)$$

```

\begin{eqnarray}
\lefteqn{\cos x = 1} \\
& & - \frac{x^2}{2!} + \\
& & \frac{x^4}{4!} - \\
& & \frac{x^6}{6!} + \dots
\end{eqnarray}

```

La commande `\nonumber` empêche \LaTeX de produire un numéro pour cette équation.

Il est parfois difficile d'obtenir des équations alignées verticalement correctement avec cette méthode. L'extension `amsmath` offre d'autres possibilités plus puissantes : voir notamment les environnements `multline`, `split` et `align` décrits dans [9] et [3].

3.6 Taille des polices mathématiques

En mode mathématique \TeX choisit la taille de la police en fonction du contexte. Les exposants, par exemple, sont imprimés avec une police plus petite.

Malgré tout, il peut être nécessaire d'indiquer à \LaTeX la taille exacte. En mode mathématique, la taille de la police est déterminée par les quatre commandes :

`\displaystyle (123)`, `\textstyle (123)`, `\scriptstyle (123)` and `\scriptscriptstyle (123)`.

Changer de style modifie également la façon dont les limites sont affichées.

$$\text{corr}(X, Y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\left[\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2 \right]^{1/2}}$$

```

\begin{displaymath}
\mathop{\mathrm{corr}}(X, Y) =
\frac{\displaystyle \sum_{i=1}^n (x_i - \overline{x})
(y_i - \overline{y})}{\displaystyle \biggl[
\sum_{i=1}^n (x_i - \overline{x})^2
\sum_{i=1}^n (y_i - \overline{y})^2
\biggr]^{1/2}}
\end{displaymath}

```

Cet exemple est un cas où on a besoin de crochets plus petits que ceux qui seraient produits par les commandes standards `\left[\right]`.

3.7 Insertion de texte en mode mathématique

La commande `\mathrm` permet d'insérer du texte normal (en caractères romains) dans une formule. La taille du texte est ajustée automatiquement, mais on est limité à un mot et les accents sont interdits. La commande `\textrm` qui fait passer en mode *texte* permet l'utilisation des lettres accentuées et respecte les espaces. L'extension `amsmath` fournit une commande `\text` également très pratique pour insérer du texte dans une équation.

$2^{\text{ième}}$	pour tout x	<code>\begin{eqnarray*}</code>
		<code>2^{\textrm{i'eme}}</code>
2^{nd}	pour tout x	<code>&&\textrm{pour tout }x\backslash</code>
		<code>2^{\mathrm{nd}}</code>
$2^{\text{ième}}$	pour tout x	<code>&&\mathrm{pour tout }x\backslash</code>
		<code>2^{\text{i'eme}}</code>
		<code>&&\text{pour tout }x</code>
		<code>\end{eqnarray*}</code>

3.8 Décrire des variables

Après certaines formules, on souhaite parfois ajouter une section qui décrit les variables utilisées dans l'expression. L'exemple suivant devrait vous aider à réaliser cela.

$$a^2 + b^2 = c^2$$

Où : a , b – sont les côtés adjacents à l'angle droit du triangle rectangle.

c – est l'hypoténuse du triangle.

```
\begin{displaymath}
a^2+b^2=c^2
\end{displaymath}
```

```
{\settowidth{\parindent}{0u :\ }
\makebox[0pt][r]{0\'u :\ }
$a$, $b$ -- sont
les c\^ot\'es adjacents \a
l'angle droit du triangle
rectangle.
```

```
$c$ -- est l'hypot\'enuse
du triangle.}
```

Si vous utilisez souvent des constructions telles que celle-ci, il est temps d'essayer la commande `\newenvironnement` pour créer un environnement dédié à la description des variables.

3.9 Théorèmes, propositions, etc.

En rédigeant des documents mathématiques on a besoin d'un moyen de présenter des lemmes, des définitions, des axiomes et d'autres structures similaires. \LaTeX prend cela en charge avec la commande :

```
\newtheorem{nom}[compteur]{texte}[section]
```

L'argument *nom* est un mot-clef utilisé pour identifier le théorème. L'argument *texte* définit le nom réel du théorème tel qu'il sera imprimé.

Les arguments entre crochets sont optionnels. Ils servent à indiquer la numérotation à utiliser sur le théorème. Avec *compteur* vous indiquez le *nom* d'un théorème déjà déclaré. Le nouveau théorème sera alors numéroté dans la même séquence. Avec *section* vous indiquez dans quel niveau de sectionnement vous voulez numéroté votre théorème.

Après avoir exécuté `\newtheorem` dans le préambule de votre document, vous pouvez utiliser la commande suivante :

```
\begin{nom}[texte]
Ceci est mon premier théorème
\end{nom}
```

Voilà pour la théorie. Les exemples qui suivent devraient montrer montrer clairement que l'environnement `\newtheorem` facile à utiliser.

Loi 1 <i>Le chef a raison.</i>	% definitions dans le preambule <code>\newtheorem{loi}{Loi}</code>
Commandement 2 (Important) <i>Le chef a toujours raison.</i>	<code>\newtheorem{commandement}[loi]{Commandement}</code> %dans le document <code>\begin{loi} \label{chef}</code> Le chef a raison. <code>\end{loi}</code>
Loi 3 <i>Si le chef a tort, voir la loi 1.</i>	<code>\begin{commandement}[Important]</code> Le chef a toujours raison. <code>\end{commandement}</code> <code>\begin{loi}</code> Si le chef a tort, voir la loi~\ref{chef}. <code>\end{loi}</code>

Le théorème « commandement » utilise le même compteur que le théorème « loi ». C'est pourquoi il a un numéro dans la même séquence que les autres « lois ». L'argument entre crochets permet de spécifier un titre ou quelque chose de ce genre pour le théorème.

Murphy 3.9.1 <i>Tout ce qui peut aller mal, ira mal.</i>	<code>\newtheorem{mur}{Murphy}[section]</code> <code>\begin{mur} Tout ce qui peut aller mal, ira mal.\end{mur}</code>
---	--

Le théorème « Murphy » est numéroté à l'intérieur de la section en cours. On aurait pu utiliser un autre niveau tel que `chapter` ou `subsection`.

3.10 Symboles gras

Il est relativement difficile d'obtenir des symboles gras avec \LaTeX ; cela est sans doute fait exprès car les typographes amateurs ont tendance à en abuser. La commande `\mathbf` permet d'obtenir des caractères gras, mais ce sont des caractères romains (droits), alors que les caractères mathématiques sont normalement en italique.

Les commandes `\mathversion{bold}` et `\mathversion{normal}`, utilisables *en mode texte uniquement* permettent de changer le style par défaut utilisé par le mode mathématique.

μ, M

```
\mathversion{bold}
\[ \mu, M ]
```

μ, M	μ, \mathbf{M}	<code>\mathversion{normal}</code>
		<code>\[\mu, M \quad</code>
		<code>\mathbf{\mu}, \mathbf{M} \]</code>

L'extension `amsmath` permet d'utiliser le gras pour un élément seulement dans une formule. Elle fournit la commande `\boldsymbol` et un « gras du pauvre⁴ » `\pmb` qui agit par superposition sur les systèmes où ne sont pas installées les polices nécessaires pour les symboles gras.

μ, M	$\boldsymbol{\mu}, \boldsymbol{M}$	μ, M	$\boldsymbol{\mu}, \boldsymbol{M}$	<code>\[\mu, M \quad</code>
				<code>\boldsymbol{\mu}, \boldsymbol{M} \quad</code>
				<code>\quad \pmb{\mu}, \pmb{M} \]</code>

4. Poor man's bold en anglais. *NdT*.

3.11 Liste des symboles mathématiques

Dans les tableaux suivants, vous trouverez tous les symboles accessibles en mode *mathématique*.

Pour utiliser des symboles présents dans les tables 3.12 à 3.16⁵, l'extension `amssymb` doit être chargée dans le préambule du document et les polices mathématiques de l'AMS doivent être installées sur votre système. Si les extensions et les polices de l'AMS ne sont pas installées sur votre système, vous pouvez les récupérer sur

CTAN:/tex-archive/macros/latex/packages/amslatex

TAB. 3.1: *Accents en mode mathématique*

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

TAB. 3.2: *Alphabet grec minuscule*

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

TAB. 3.3: *Alphabet grec majuscule*

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

5. Ces tables sont dérivées du fichier `symbols.tex` de David Carlisle et modifiées selon les suggestions de Josef Tkadlec

TAB. 3.4: *Relations binaires*

Vous pouvez produire la négation de ces relations en préfixant ces commandes par `\not`.

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

TAB. 3.5: *Opérateurs binaires*

$+$	<code>+</code>	$-$	<code>-</code>	\triangleleft	<code>\triangleleft</code>
\pm	<code>\pm</code>	\mp	<code>\mp</code>	\triangleleft	<code>\triangleleft</code>
\cdot	<code>\cdot</code>	\div	<code>\div</code>	\triangleright	<code>\triangleright</code>
\times	<code>\times</code>	\setminus	<code>\setminus</code>	\star	<code>\star</code>
\cup	<code>\cup</code>	\cap	<code>\cap</code>	\ast	<code>\ast</code>
\sqcup	<code>\sqcup</code>	\sqcap	<code>\sqcap</code>	\circ	<code>\circ</code>
\vee	<code>\vee</code> , <code>\lou</code>	\wedge	<code>\wedge</code> , <code>\land</code>	\bullet	<code>\bullet</code>
\oplus	<code>\oplus</code>	\ominus	<code>\ominus</code>	\diamond	<code>\diamond</code>
\odot	<code>\odot</code>	\oslash	<code>\oslash</code>	\uplus	<code>\uplus</code>
\otimes	<code>\otimes</code>	\bigcirc	<code>\bigcirc</code>	\amalg	<code>\amalg</code>
\triangleup	<code>\triangleup</code>	\triangledown	<code>\triangledown</code>	\dagger	<code>\dagger</code>
\triangleleft	<code>\lhd</code> ^a	\triangleright	<code>\rhd</code> ^a	\ddagger	<code>\ddagger</code>
\triangleleft	<code>\unlhd</code> ^a	\triangleright	<code>\unrhd</code> ^a	\wr	<code>\wr</code>

^aUtilisez l'extension `latexsym` pour avoir accès à ces symboles

TAB. 3.6: *Opérateurs n-aires*

\sum	<code>\sum</code>	\bigcup	<code>\bigcup</code>	\bigvee	<code>\bigvee</code>	\bigoplus	<code>\bigoplus</code>
\prod	<code>\prod</code>	\bigcap	<code>\bigcap</code>	\bigwedge	<code>\bigwedge</code>	\bigotimes	<code>\bigotimes</code>
\coprod	<code>\coprod</code>	\bigsqcup	<code>\bigsqcup</code>			\bigodot	<code>\bigodot</code>
\int	<code>\int</code>	\oint	<code>\oint</code>			\biguplus	<code>\biguplus</code>

TAB. 3.7: *Flèches*

\leftarrow	<code>\leftarrow</code> ou <code>\gets</code>	\longleftarrow	<code>\longleftarrow</code>	\uparrow	<code>\uparrow</code>
\rightarrow	<code>\rightarrow</code> ou <code>\to</code>	\longrightarrow	<code>\longrightarrow</code>	\downarrow	<code>\downarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>	\updownarrow	<code>\updownarrow</code>
\Leftarrow	<code>\Leftarrow</code>	\Lleftarrow	<code>\Lleftarrow</code>	\Uparrow	<code>\Uparrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Rrightarrow	<code>\Rrightarrow</code>	\Downarrow	<code>\Downarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Llongleftrightarrow	<code>\Llongleftrightarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>	\nearrow	<code>\nearrow</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>	\searrow	<code>\searrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>	\swarrow	<code>\swarrow</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoondown	<code>\rightharpoondown</code>	\nwarrow	<code>\nwarrow</code>
\rightleftharpoons	<code>\rightleftharpoons</code>	\iff	<code>\iff</code> (plus d'espace)	\leadsto	<code>\leadsto</code> ^a

^aUtilisez l'extension `latexsym` pour obtenir ces symboles

TAB. 3.8: *Délimiteurs*

$($	<code>(</code>	$)$	<code>)</code>	\uparrow	<code>\uparrow</code>	\Uparrow	<code>\Uparrow</code>
$[$	<code>[</code> ou <code>\lbrack</code>	$]$	<code>] ou \rbrack</code>	\downarrow	<code>\downarrow</code>	\Downarrow	<code>\Downarrow</code>
$\{$	<code>\{</code> ou <code>\lbrace</code>	$\}$	<code>\} ou \rbrace</code>	\updownarrow	<code>\updownarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\langle	<code>\langle</code>	\rangle	<code>\rangle</code>	$ $	<code> </code> ou <code>\vert</code>	$\ $	<code>\ </code> ou <code>\Vert</code>
\lfloor	<code>\lfloor</code>	\rfloor	<code>\rfloor</code>	\lceil	<code>\lceil</code>	\rceil	<code>\rceil</code>
$/$	<code>/</code>	\backslash	<code>\backslash</code>				

TAB. 3.9: *Grands délimiteurs*

$\left($	<code>\lgroup</code>	$\right)$	<code>\rgroup</code>	\int	<code>\lmoustache</code>	$\}$	<code>\rmoustache</code>
$\left $	<code>\arrowvert</code>	$\right $	<code>\Arrowvert</code>	$\{$	<code>\bracevert</code>		

TAB. 3.10: *Symboles divers*

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

^aUtilisez l'extension `latexsym` pour obtenir ces symboles

TAB. 3.11: *Symboles non-mathématiques*

Ces symboles peuvent également être utilisés en mode *texte*.

\dagger	<code>\dag</code>	\S	<code>\S</code>	\copyright	<code>\copyright</code>
\ddagger	<code>\ddag</code>	\P	<code>\P</code>	\pounds	<code>\pounds</code>

TAB. 3.12: *Délimiteurs de l'AMS*

\ulcorner	<code>\ulcorner</code>	\urcorner	<code>\urcorner</code>	\llcorner	<code>\llcorner</code>	\lrcorner	<code>\lrcorner</code>
-------------	------------------------	-------------	------------------------	-------------	------------------------	-------------	------------------------

TAB. 3.13: *Caractères grecs et hébreux de l'AMS*

\digamma	<code>\digamma</code>	\varkappa	<code>\varkappa</code>	\beth	<code>\beth</code>	\daleth	<code>\daleth</code>	\gimel	<code>\gimel</code>
------------	-----------------------	-------------	------------------------	---------	--------------------	-----------	----------------------	----------	---------------------

TAB. 3.14: Relations binaires de l'AMS

\triangleleft	<code>\lessdot</code>	\triangleright	<code>\gtrdot</code>	\doteq	<code>\doteqdot</code> ou <code>\Doteq</code>
\leqslant	<code>\leqslant</code>	\geqslant	<code>\geqslant</code>	$\dot{=}$	<code>\risingdotseq</code>
\leqslantless	<code>\leqslantless</code>	\geqslantgtr	<code>\geqslantgtr</code>	\fallingdotseq	<code>\fallingdotseq</code>
\leqq	<code>\leqq</code>	\geqq	<code>\geqq</code>	\equiv	<code>\eqcirc</code>
\lll ou \llless	<code>\lll</code> ou <code>\llless</code>	\ggg ou \gggtr	<code>\ggg</code> ou <code>\gggtr</code>	\circ	<code>\circeq</code>
\lesssim	<code>\lesssim</code>	\gtrsim	<code>\gtrsim</code>	\triangleq	<code>\triangleq</code>
\lessapprox	<code>\lessapprox</code>	\gtrapprox	<code>\gtrapprox</code>	\bumpeq	<code>\bumpeq</code>
\lessgtr	<code>\lessgtr</code>	\gtrless	<code>\gtrless</code>	\Bumpeq	<code>\Bumpeq</code>
\lesseqgtr	<code>\lesseqgtr</code>	\gtreqless	<code>\gtreqless</code>	\sim	<code>\thicksim</code>
\lesseqqgtr	<code>\lesseqqgtr</code>	\gtreqqless	<code>\gtreqqless</code>	\approx	<code>\thickapprox</code>
\preccurlyeq	<code>\preccurlyeq</code>	\succcurlyeq	<code>\succcurlyeq</code>	\approx	<code>\approxeq</code>
\curlyeqprec	<code>\curlyeqprec</code>	\curlyeqsucc	<code>\curlyeqsucc</code>	\backsimeq	<code>\backsimeq</code>
\precsim	<code>\precsim</code>	\succsim	<code>\succsim</code>	\backsimeq	<code>\backsimeq</code>
\precapprox	<code>\precapprox</code>	\succapprox	<code>\succapprox</code>	\vDash	<code>\vDash</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\Vdash	<code>\Vdash</code>
\Subset	<code>\Subset</code>	\Supset	<code>\Supset</code>	\Vvdash	<code>\Vvdash</code>
\sqsubset	<code>\sqsubset</code>	\sqsupset	<code>\sqsupset</code>	\backepsilon	<code>\backepsilon</code>
\therefore	<code>\therefore</code>	\because	<code>\because</code>	\propto	<code>\varpropto</code>
\shortmid	<code>\shortmid</code>	\shortparallel	<code>\shortparallel</code>	\between	<code>\between</code>
\smallsmile	<code>\smallsmile</code>	\smallfrown	<code>\smallfrown</code>	\pitchfork	<code>\pitchfork</code>
\vartriangleleft	<code>\vartriangleleft</code>	\vartriangleright	<code>\vartriangleright</code>	\blacktriangleleft	<code>\blacktriangleleft</code>
\trianglelefteq	<code>\trianglelefteq</code>	\trianglerighteq	<code>\trianglerighteq</code>	\blacktriangleright	<code>\blacktriangleright</code>

TAB. 3.15: Flèches de l'AMS

\dashleftarrow	<code>\dashleftarrow</code>	\dashrightarrow	<code>\dashrightarrow</code>	\multimap	<code>\multimap</code>
\leftleftarrows	<code>\leftleftarrows</code>	\rightrightarrows	<code>\rightrightarrows</code>	\Uparrow	<code>\upuparrows</code>
\leftrightarrows	<code>\leftrightarrows</code>	\rightleftarrows	<code>\rightleftarrows</code>	\Downarrow	<code>\downdownarrows</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Rrightarrow	<code>\Rrightarrow</code>	\Uparrow	<code>\upharpoonleft</code>
\twoheadleftarrow	<code>\twoheadleftarrow</code>	\twoheadrightarrow	<code>\twoheadrightarrow</code>	\Uparrow	<code>\upharpoonright</code>
\leftarrowtail	<code>\leftarrowtail</code>	\rightarrowtail	<code>\rightarrowtail</code>	\Downarrow	<code>\downharpoonleft</code>
\leftrightharpoons	<code>\leftrightharpoons</code>	\rightleftharpoons	<code>\rightleftharpoons</code>	\Downarrow	<code>\downharpoonright</code>
\Lsh	<code>\Lsh</code>	\Rsh	<code>\Rsh</code>	\rightsquigarrow	<code>\rightsquigarrow</code>
\looparrowleft	<code>\looparrowleft</code>	\looparrowright	<code>\looparrowright</code>	\leftrightarrow	<code>\leftrightsquigarrow</code>
\curvearrowleft	<code>\curvearrowleft</code>	\curvearrowright	<code>\curvearrowright</code>		
\circlearrowleft	<code>\circlearrowleft</code>	\circlearrowright	<code>\circlearrowright</code>		

TAB. 3.16: *Négations des relations binaires et des flèches de l'AMS*

\nless	\ngtr	\varsubsetneqq
\lneq	\gneq	\varsupsetneqq
\nleq	\ngeq	\nsubseteqeq
\nleqslant	\ngeqslant	\nsupseteqeq
\lneqq	\gneqq	\nmid
\lvertneqq	\gvertneqq	\nparallel
\nleqq	\ngeqq	\nshortmid
\lnsim	\gnsim	\nshortparallel
\lnapprox	\gnapprox	\nsim
\nprec	\nsucc	\ncong
\npreceq	\nsucceq	\nvdash
\nprecneqq	\nsuccneqq	\nvDash
\nprecnsim	\succnsim	\nVdash
\nprecnapprox	\succnapprox	\nVDash
\subsetneq	\supsetneq	\ntriangleleft
\varsubsetneq	\varsupsetneq	\ntriangleright
\nsubseteq	\nsupseteq	\ntrianglelefteq
\nsubseteqeq	\supseteqeq	\ntrianglerighteq
\nleftarrow	\rightarrow	\nleftrightarrow
\nLeftarrow	\Rightarrow	\nLeftrightarrow

TAB. 3.17: *Opérateurs binaires de l'AMS*

\dotplus	\centerdot	\intercal
\ltimes	\rtimes	\divideontimes
\Cup ou \doublecup	\Cap ou \doublecap	\smallsetminus
\veebar	\barwedge	\doublebarwedge
\boxplus	\boxminus	\circleddash
\boxtimes	\boxdot	\circledcirc
\leftthreetimes	\rightthreetimes	\circledast
\curlyvee	\curlywedge	

TAB. 3.18: *Symboles divers de l'AMS*

\hbar	<code>\hbar</code>	\hbar	<code>\hslash</code>	\mathbb{k}	<code>\Bbbk</code>
\square	<code>\square</code>	\blacksquare	<code>\blacksquare</code>	\textcircled{S}	<code>\circledS</code>
\triangle	<code>\vartriangle</code>	\blacktriangle	<code>\blacktriangle</code>	\complement	<code>\complement</code>
∇	<code>\triangledown</code>	\blacktriangledown	<code>\blacktriangledown</code>	\Game	<code>\Game</code>
\lozenge	<code>\lozenge</code>	\blacklozenge	<code>\blacklozenge</code>	\bigstar	<code>\bigstar</code>
\sphericalangle	<code>\angle</code>	\sphericalangle	<code>\measuredangle</code>	\sphericalangle	<code>\sphericalangle</code>
\diagup	<code>\diagup</code>	\diagdown	<code>\diagdown</code>	\backprime	<code>\backprime</code>
\nexists	<code>\nexists</code>	\Finv	<code>\Finv</code>	\varnothing	<code>\varnothing</code>
\eth	<code>\eth</code>	\mho	<code>\mho</code>		

TAB. 3.19: *Polices mathématiques*

Exemple	Commande	Extension à utiliser
ABCdef	<code>\mathrm{ABCdef}</code>	
ABCdef	<code>\mathit{ABCdef}</code>	
\mathnormal{ABCdef}	<code>\mathnormal{ABCdef}</code>	
\mathcal{ABC}	<code>\mathcal{ABC}</code>	
\mathcal{QRS}	<code>\mathcal{ABC}</code>	eucal avec l'option : <code>mathcal</code> ou
	<code>\mathscr{ABC}</code>	eucal avec l'option : <code>mathscr</code>
\mathfrak{ABCdef}	<code>\mathfrak{ABCdef}</code>	eufrak
\mathbb{ABC}	<code>\mathbb{ABC}</code>	amsfonts ou amssymb

Chapitre 4

Compléments

Ne lisez pas ce chapitre ! Si vous vous sentez à l'aise, vous pouvez écrire vos premiers documents avec \LaTeX dès maintenant. Le but de ce chapitre est d'ajouter un peu de piment à votre connaissance de \LaTeX . Une description bien plus complète des possibilités et des améliorations possibles se trouve dans le *LaTeX Manual* [1] et dans *The LaTeX Companion* [3].

4.1 Polices et tailles

\LaTeX choisit la police de caractères et sa taille en fonction de la structure logique du document (sections, notes de bas de page,...). Dans certains cas, on voudrait pouvoir changer la taille de la police à la main. Pour cela, utilisez les commandes listées dans les tableaux 4.1 et 4.2. La taille exacte de chaque police est un choix qui dépend de la classe de document et de ses options.

Les romains petits et **gras** régnerent
sur toute la belle et grande
Italie.

```
{\small Les romains  
petits et \textbf{gras}  
r'\eg\n\`erent sur}  
\Large toute la belle  
et grande \textit{Italie}.}
```

Une caractéristique importante de $\LaTeX 2_{\epsilon}$ est que les différents attributs d'une police sont indépendants. Cela signifie que vous pouvez exécuter des commandes de changement de taille ou même de changement de police tout en conservant l'attribut gras ou italique. Cela peut paraître évident à quelqu'un qui débute avec $\LaTeX 2_{\epsilon}$, mais ça ne l'est certainement pas pour quelqu'un qui était habitué à $\LaTeX 2.09$.

En mode *mathématique*, vous pouvez utiliser les commandes de changement de police pour quitter provisoirement le mode math et saisir du texte normal. Pour changer les attributs de la police en mode mathématique, il existe un jeu de commandes spéciales. Reportez-vous au tableau 4.3.

TAB. 4.1: *Polices*

<code>\textrm{...}</code>	roman	<code>\textsf{...}</code>	sans serif
<code>\texttt{...}</code>	typewriter		
<code>\textmd{...}</code>	medium	<code>\textbf{...}</code>	bold face
<code>\textup{...}</code>	upright	<code>\textit{...}</code>	<i>italic</i>
<code>\textsl{...}</code>	<i>slanted</i>	<code>\textsc{...}</code>	SMALL CAPS
<code>\emph{...}</code>	<i>emphasised</i>	<code>\textnormal{...}</code>	document font

TAB. 4.2: *Tailles des polices*

<code>\tiny</code>	minuscule	<code>\Large</code>	plus grand
<code>\scriptsize</code>	très petit	<code>\LARGE</code>	très grand
<code>\footnotesize</code>	assez petit		
<code>\small</code>	petit	<code>\huge</code>	énorme
<code>\normalsize</code>	normal		
<code>\large</code>	grand	<code>\Huge</code>	géant

TAB. 4.3: *Polices mathématiques*

<i>Commande</i>	<i>Exemple</i>	<i>Résultat</i>
<code>\mathcal{...}</code>	<code>\$\$\mathcal{B}=c\$</code>	$\mathcal{B} = c$
<code>\mathrm{...}</code>	<code>\$\$\mathrm{K}_2\$</code>	K_2
<code>\mathbf{...}</code>	<code>\$\$\sum x=\mathbf{v}\$</code>	$\sum x = \mathbf{v}$
<code>\mathsf{...}</code>	<code>\$\$\mathsf{G\times R}\$</code>	$G \times R$
<code>\mathtt{...}</code>	<code>\$\$\mathtt{L}(b,c)\$</code>	$L(b, c)$
<code>\mathnormal{...}</code>	<code>\$\$\mathnormal{R_{19}}\neq R_{19}\$</code>	$R_{19} \neq R_{19}$
<code>\mathit{...}</code>	<code>\$\$\mathit{ffi}\neq ffi\$</code>	$\mathit{ffi} \neq ffi$

En lien avec les commandes de changement de taille, les accolades jouent un rôle important. Elles sont utilisées pour construire des *groupes* qui limitent la portée de la plupart des commandes L^AT_EX.

Il aime les grands et les petits caractères.	Il aime les <code>{\LARGE grands et {\small les petits} caract\`eres}</code> .
--	--

Les commandes de changement de taille modifient également l'interligne, mais seulement si le paragraphe se termine dans la portée de la commande de changement de taille. L'accolade fermante ne doit donc pas être placée trop tôt. Remarquez la position de la commande `\par` dans les deux exemples suivants :

Ne lisez pas ceci ! Ce n'est pas vrai ! Croyez-moi !	<code>{\Large Ne lisez pas ceci ! Ce n'est pas vrai ! Croyez-moi !\par}</code>
--	--

Ce n'est pas vrai. Mais n'oubliez pas que je suis un menteur.	<code>{\Large Ce n'est pas vrai. Mais n'oubliez pas que je suis un menteur.}\par</code>
---	---

Pour conclure cette promenade au pays des commandes de changement de police, voici un petit conseil :

N'oubliez pas ! *Plus* **vOus** *utilisez* de polices **dans un** document *Plus* il DEVIENT *lisible et bien présenté*¹.

4.2 Espacement

4.2.1 Entre les lignes

Pour utiliser un interligne plus grand pour un document, vous pouvez utiliser la commande

`\linespread{facteur}`

dans le préambule de votre document. Utilisez `\linespread{1.3}` pour un interligne « un et demi » et `\linespread{1.6}` pour un « double » interligne. L'interligne par défaut est 1.

Ce changement a souvent des effets indésirables dans les notes de bas de page, les tableaux, etc. C'est assez compliqué à gérer, mais l'extension `setspace` est prévue à cet effet.

1. Attention, c'est du second degré, j'espère que vous l'aurez compris.

4.2.2 Mise en page d'un paragraphe

Il y a deux paramètres qui jouent sur l'apparence d'un paragraphe. En insérant une définition telle que :

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{1ex plus 0.5ex minus 0.2ex}
```

dans le préambule, vous supprimez le retrait des débuts de paragraphe (1^{re} définition) et vous augmentez l'espace entre deux paragraphes (2^e définition). Attention, la deuxième définition a également une influence sur la table des matières.

Pour indenter un paragraphe qui ne l'est pas, utilisez la commande :

```
\indent
```

au début du paragraphe². Bien sûr cela ne marche que si `\parindent` n'est pas nul.

Pour créer un paragraphe sans indentation, utilisez :

```
\noindent
```

en tête du paragraphe.

l'extension `french` modifie ici aussi les règles par défaut de `LATEX` pour s'adapter aux règles françaises. Elle propose en plus la commande :

```
\lettrine{texte}
```

qui permet de commencer le premier paragraphe d'un document par une lettrine telle que celle de la page [iii](#).

4.2.3 Espacement horizontal

`LATEX` détermine l'espacement entre les mots et les phrases automatiquement. Pour ajouter de l'espacement horizontal, utilisez :

```
\hspace{longueur}
```

Si une telle espace doit être conservée, même s'elle tombe en début ou en fin de ligne, utilisez `\hspace*`. Dans le cas le plus simple, *longueur* est simplement un nombre suivi d'une unité. Les unités les plus importantes sont listées dans le tableau [4.4](#).

2. Pour indenter le premier paragraphe après le titre d'une section, utilisez l'extension `indentfirst` de l'ensemble `tools`.

TAB. 4.4: Unités $T_{E}X$

mm	millimètre	□
cm	centimètre = 10 mm	□
in	pouce ^a = 25.4 mm	□
pt	point $\approx 1/72$ pouce ≈ 0.35 mm	□
em	largeur d'un m dans la police courante	□
ex	hauteur d'un x dans la police courante	□

^aInch en anglais. *NdT.*

Ceci est une espace de 1.5 cm. Ceci\hspace{1.5cm}est une espace de 1.5 cm.

La commande :

`\stretch{n}`

produit une espace élastique. Elle s'étend jusqu'à ce que tout l'espace libre sur la ligne soit occupé. Si deux commandes `\hspace{\stretch{n}}` sont exécutées sur la même ligne, les deux espaces s'étendent en fonction du facteur d'élasticité n .

x x x x\hspace{\stretch{1}}%
x\hspace{\stretch{3}}x

4.2.4 Espacement vertical

L'espacement vertical entre les paragraphes, les sections, les sous-sections,... est déterminé automatiquement par $L_{A}T_{E}X$. En cas de besoin, de l'espace supplémentaire *entre deux paragraphes* peut être inséré avec la commande :

`\vspace{longueur}`

Cette commande doit normalement être utilisée entre deux lignes vides. Si l'espacement doit être conservé en haut ou en bas d'une page, utilisez la version étoilée de la commande `\vspace*`.

La commande `\stretch` en association avec `\pagebreak` permet d'imprimer du texte sur la dernière ligne d'une page ou de centrer verticalement du texte sur une page.

Du texte \ldots

```
\vspace{\stretch{1}}
Ceci sera imprimé sur la dernière ligne.\pagebreak
```

De l'espace supplémentaire entre deux lignes du *même* paragraphe ou à l'intérieur d'une table peut être obtenu par la commande :

```
\[\longueur]
```

4.3 Disposition d'une page

L^AT_EX 2_ε permet d'indiquer la taille du papier en paramètre de la commande `\documentclass`. Il définit ensuite automatiquement les marges les mieux adaptées. Parfois, on peut ne pas être satisfait par les valeurs prédéfinies et vouloir les modifier. La figure 4.1 montre tous les paramètres qui peuvent être modifiés. Cette figure a été réalisée avec l'extension `layout` de l'ensemble `tools`.

L^AT_EX dispose de deux commandes pour modifier ces paramètres. Elles sont généralement utilisées dans le préambule.

La première commande affecte une valeur fixe à l'un quelconque de ces paramètres :

```
\setlength{paramètre}{longueur}
```

La deuxième commande ajoute une longueur à l'un des paramètres.

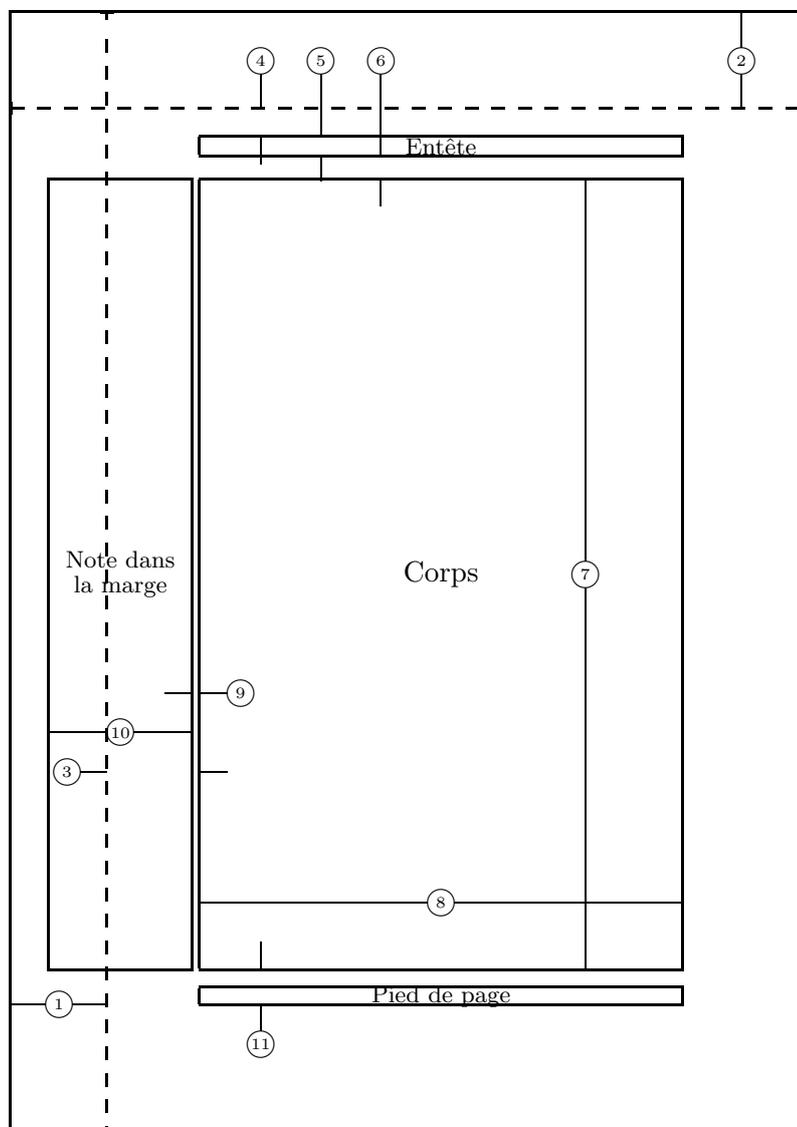
```
\addtolength{paramètre}{longueur}
```

La deuxième commande est en pratique plus utile que `\setlength`, parce qu'elle permet de travailler en relatif par rapport à la taille par défaut. Pour ajouter un centimètre à la largeur du texte, nous utiliserions les commandes suivantes dans le préambule :

```
\addtolength{\hoffset}{-0.5cm}
\addtolength{\textwidth}{1cm}
```

4.4 Références bibliographiques

L'environnement `thebibliography` permet de produire une liste de références bibliographiques. Cet environnement est conçu pour être produit automatiquement par BiB_TE_X. Il est cependant possible de construire de petites



1 un pouce + <code>\hoffset</code>	2 un pouce + <code>\voffset</code>
3 <code>\evensidemargin = 70pt</code>	4 <code>\topmargin = 22pt</code>
5 <code>\headheight = 13pt</code>	6 <code>\headsep = 19pt</code>
7 <code>\textheight = 595pt</code>	8 <code>\textwidth = 360pt</code>
9 <code>\marginparsep = 7pt</code>	10 <code>\marginparwidth = 106pt</code>
11 <code>\footskip = 27pt</code>	<code>\marginparpush = 5pt</code> (non affiché)
<code>\hoffset = 0pt</code>	<code>\voffset = 0pt</code>
<code>\paperwidth = 597pt</code>	<code>\paperheight = 845pt</code>

FIG. 4.1: Paramètres de la disposition d'une page

bibliographies à la main. Chaque référence commence par

```
\bibitem{marque}
```

La *marque* est utilisée pour citer la référence dans le document.

```
\cite{marque}
```

La numérotation des références est produite automatiquement. Le paramètre qui suit `\begin{thebibliography}` définit la largeur du décrochement utilisé pour placer ces numéros.

Partl [1] propose que ...

```
Partl~\cite{pa}
propose que \ldots
```

```
{\small
\begin{thebibliography}{99}
\bibitem{pa} H.~Partl:
\emph{German \TeX},
TUGboat Vol.~9, No.~1 ('88)
\end{thebibliography}
}
```

Bibliographie

- [1] H. Partl: *German T_EX*, TUGboat Vol. 9, No. 1 ('88)

4.5 Index

L'index est une caractéristique fort utile de nombreux ouvrages. L^AT_EX et le programme associé `makeindex`³ permettent de créer des index assez facilement. Dans cette introduction, seules les commandes élémentaires de gestion d'un index sont présentées. Pour une description plus détaillée, reportez-vous à *The L^AT_EX Companion* [3].

Pour utiliser cette fonctionnalité, l'extension `makeidx` doit être chargé dans le préambule⁴ avec :

```
\usepackage{makeidx}
```

3. sur les systèmes qui ne supportent pas les noms de fichiers de plus de huit caractères, ce programme s'appelle `makeidx`.

4. sauf avec l'extension `french` qui gère directement les index.

TAB. 4.5: Exemples de clefs d'index

Exemple	Résultat	Commentaires
<code>\index{hello}</code>	hello, 1	Entrée normale
<code>\index{hello!Peter}</code>	Peter, 3	Sous-entrée de 'hello'
<code>\index{Sam@\textsl{Sam}}</code>	<i>Sam</i> , 2	Entrée formatée
<code>\index{Lin@\textbf{Lin}}</code>	Lin , 7	idem
<code>\index{Jenny textbf}</code>	Jenny, 3	Numéro de page formaté
<code>\index{Joe textit}</code>	Joe, <i>5</i>	idem

Les commandes de création de l'index doivent être activées par la commande :

```
\makeindex
```

placée dans le préambule.

Le contenu de l'index est défini par une série de commandes :

```
\index{clef}
```

où *clef* est un mot-clef de l'index. Vous insérez des commandes `\index` aux endroits du texte que vous voulez voir référencés par l'index. Le tableau 4.5 explique la syntaxe de l'argument *clef* avec plusieurs exemples.

Quand le fichier source est traité par \LaTeX , chaque commande `\index` crée une entrée adaptée contenant le numéro de la page en cours dans le fichier qui porte le même nom de base que le fichier source, mais avec le suffixe `.idx`. Ce fichier est ensuite traité par le programme `makeindex`.

```
makeindex nom de fichier
```

Le programme `makeindex` crée un index trié dans le fichier `.ind`. Ensuite, la prochaine fois que le fichier source sera traité, le contenu du fichier `.ind` sera inclus à l'endroit où \LaTeX rencontrera la commande :

```
\printindex
```

L'extension `showidx` permet de visualiser les entrées de l'index dans la marge gauche du texte. Cela permet la relecture et la mise au point de l'index.

Le programme `makeindex` standard ne traite malheureusement pas correctement les caractères accentués dans les clefs : il les place systématiquement en tête de l'ordre alphabétique.

4.6 Figures PostScript

Avec les environnements `figure` et `table`, L^AT_EX fournit les mécanismes de base pour travailler avec des objets tels que des images ou des graphiques.

Il existe également plusieurs possibilités pour produire des graphiques avec des commandes L^AT_EX ou des extensions de L^AT_EX. Malheureusement, la plupart des utilisateurs trouvent ces commandes difficiles à mettre en œuvre. Pour plus d'informations à leur sujet, reportez-vous à *The L^AT_EX Companion* [3] et au *L^AT_EX Manual* [1].

Un moyen bien plus simple d'inclure des graphiques dans un document est de les produire à l'aide d'un logiciel spécialisé⁵ puis d'inclure le résultat dans le document. Ici encore L^AT_EX offre de nombreuses solutions. Dans cette introduction, seule l'utilisation de graphiques PostScript Encapsulé (EPS) sera traitée. En effet, c'est la solution la plus simple et la plus utilisée. Pour pouvoir utiliser des graphiques au format EPS, il faut disposer d'une imprimante acceptant les documents PostScript.

Un ensemble de commandes bien adaptées à l'insertion de graphiques est fourni par l'extension `graphicx`, développée par D. P. Carlisle. Elle fait partie d'un ensemble d'extensions appelé « `graphics` ». ⁶

En supposant maintenant que vous travaillez sur un système où l'extension `graphicx` est installée et qui dispose d'une imprimante PostScript, voici la marche à suivre pour inclure une figure dans un document :

1. Exportez la figure de votre logiciel graphique au format EPS.
2. Chargez l'extension `graphicx` dans le préambule de votre fichier source avec :

```
\usepackage[pilote]{graphicx}
```

pilote est le nom du programme de conversion « DVI vers PostScript » que vous utilisez⁷. Cette information est nécessaire car l'insertion de la figure est réalisée par le pilote au moment de l'impression. Cela permet d'inclure les bonnes commandes dans le fichier `.dvi` pour réaliser cette insertion.

3. Utilisez la commande :

```
\includegraphics[clef=valeur, ...]{fichier}
```

pour insérer *fichier* dans votre document. Le paramètre optionnel est une liste de paires de *clefs* et de *valeurs* séparées par des virgules. Les

5. Tel que XFig, idraw, CorelDraw!, FreeHand,...

6. CTAN:/tex-archive/macros/latex/packages/graphicx

7. Le programme le plus répandu est `dvips`.

clefs permettent de modifier la largeur, la hauteur, ou l'angle de rotation de la figure. Le tableau 4.6 présente les clefs les plus importantes.

TAB. 4.6: *Clefs pour l'extension graphicx*

<code>width</code>	définit la largeur de la figure
<code>height</code>	définit la hauteur de la figure
<code>angle</code>	(en degrés) tourne la figure dans le sens des aiguilles d'une montre

L'exemple suivant devrait illustrer le fonctionnement de la commande :

```
\begin{figure}
\begin{center}
\includegraphics[angle=90, width=10cm]{test.eps}
\end{center}
\end{figure}
```

Cette commande inclut la figure stockée dans le fichier `test.eps`. La figure est *d'abord* tournée de 90 degrés puis ajustée pour que sa largeur finale soit de 10 cm. Le ratio final est de 1.0, puisqu'aucune hauteur n'est spécifiée.

Pour plus d'informations, reportez vous à [8].

Bibliographie

- [1] Leslie Lamport. *L^AT_EX: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, second edition, 1994, ISBN 0-201-52983-1.
- [2] Donald E. Knuth. *The T_EXbook*, Volume A of *Computers and Typesetting*, Addison-Wesley Publishing Company, 1984, ISBN 0-201-13448-9.
- [3] Michel Goossens, Frank Mittelbach and Alexander Samarin. *The L^AT_EX Companion*. Addison-Wesley, Reading, Massachusetts, 1994, ISBN 0-201-54199-8.
- [4] Chaque installation de L^AT_EX devrait fournir un document appelé *L^AT_EX Local Guide*, qui explique les particularités de cette installation. Malheureusement certains administrateurs système paresseux ne fournissent pas ce document. Dans ce cas, demandez de l'aide aux autres utilisateurs autour de vous ou au gourou local de L^AT_EX.
- [5] L^AT_EX3 Project Team. *L^AT_EX 2_ε for authors*. Distribué avec L^AT_EX 2_ε dans `usrguide.tex`.
- [6] L^AT_EX3 Project Team. *L^AT_EX 2_ε for Class and Package writers*. Distribué avec L^AT_EX 2_ε dans `clsguide.tex`.
- [7] L^AT_EX3 Project Team. *L^AT_EX 2_ε Font selection*. Distribué avec L^AT_EX 2_ε dans `fntguide.tex`.
- [8] D. P. Carlisle. *Packages in the 'graphics' bundle*. Distribué avec les extensions « graphics » dans `grfguide.tex`.
- [9] American Mathematical Society *A_MS-L^AT_EX Version 1.2 User's guide*. Distribué avec les extensions A_MS-L^AT_EX dans `amsldoc.tex`.
- [10] B. Gaulle. *Notice d'utilisation du style french multilingue*. Disponible avec le style `french` sur <ftp.univ-rennes1.fr:/pub/GUTenberg/french>.
- [11] Y. Perrousseau. *Manuel de typographie française élémentaire*. Ateliers Perrousseau éditeur, 1995, ISBN 2-911220-00-5.

Index

Symboles	
"	16
\$	35
'	38
\(35
\)	35
\,	36, 41
-	16
—	16
\-	15
—	16
—	16
., espace après	19
...	17
\@	19
\[35
\]	14, 24–26, 60
*	14
\]	35
^	38
_	38
~	19
A	
A4 (papier)	9
A5 (papier)	9
accent	17
aigu	18
circonflexe	18
grave	18
accolade	57
accolades	5, 40
\addtolength	60
æ	18
alignement décimal	27
amsmath	37, 54
amssymb	37, 48, 54
\and	21
antislash	5
\appendix	20
array	42
article (classe)	8
\author	21
avantages de L ^A T _E X	3
B	
B5 (papier)	9
babel	18, 19
\backmatter	21
backslash	5
\begin	22
\bibitem	62
\Big	41
\big	41
\Bigg	41
\bigg	41
\binom	39
blackboard bold	37
blancs	4
\bmod	39
bold face	56
\boldsymbol	47
book (classe)	8
C	
\caption	29
caractères réservés	4
caractères spéciaux	17
\cdots	41
center	24
césure	14, 18

<code>\chapter</code>	20	<code>\frenchspacing</code>	19
<code>\cite</code>	62	<code>\frontmatter</code>	21
<code>\cleardoublepage</code>	30	<code>\fussy</code>	14
<code>\clearpage</code>	30	<code>\hline</code>	26
codage	10	<code>\hspace</code>	58
commandes		<code>\Huge</code>	56
<code>\(</code>	35	<code>\huge</code>	56
<code>\)</code>	35	<code>\hyphenation</code>	15
<code>\,</code>	36, 41	<code>\include</code>	12
<code>\-</code>	15	<code>\includegraphics</code>	64
<code>\@</code>	19	<code>\includeonly</code>	12
<code>\[</code>	35	<code>\indent</code>	58
<code>\]</code>	14, 24–26, 60	<code>\index</code>	63
<code>*</code>	14	<code>\input</code>	12
<code>\]</code>	35	<code>\int</code>	40
<code>\addtolength</code>	60	<code>\item</code>	23
<code>\and</code>	21	<code>\label</code>	21, 36
<code>\appendix</code>	20	<code>\LARGE</code>	56
<code>\author</code>	21	<code>\Large</code>	56
<code>\backmatter</code>	21	<code>\large</code>	56
<code>\begin</code>	22	<code>\ldots</code>	17, 41
<code>\bibitem</code>	62	<code>\left</code>	40
<code>\Big</code>	41	<code>\lettrine</code>	58
<code>\big</code>	41	<code>\linebreak</code>	14
<code>\Bigg</code>	41	<code>\linespread</code>	57
<code>\bigg</code>	41	<code>\listoffigures</code>	30
<code>\binom</code>	39	<code>\listoftables</code>	30
<code>\bmod</code>	39	<code>\mainmatter</code>	21
<code>\boldsymbol</code>	47	<code>\makeindex</code>	63
<code>\caption</code>	29	<code>\maketitle</code>	21
<code>\cdots</code>	41	<code>\mathbb</code>	37
<code>\chapter</code>	20	<code>\mathbf</code>	56
<code>\cite</code>	62	<code>\mathcal</code>	56
<code>\cleardoublepage</code>	30	<code>\mathit</code>	56
<code>\clearpage</code>	30	<code>\mathnormal</code>	56
<code>\date</code>	21	<code>\mathrm</code>	44, 56
<code>\ddots</code>	41	<code>\mathsf</code>	56
<code>\displaystyle</code>	44	<code>\mathtt</code>	56
<code>\documentclass</code>	8	<code>\mathversion</code>	46
<code>\emph</code>	22, 56	<code>\mbox</code>	15, 17
<code>\end</code>	22	<code>\multicolumn</code>	27
<code>\footnote</code>	22	<code>\newcommand</code>	31
<code>\footnotesize</code>	56	<code>\newenvironment</code>	32
<code>\frac</code>	39	<code>\newline</code>	14

<code>\newpage</code>	14	<code>\text</code>	44
<code>\newtheorem</code>	45	<code>\textbf</code>	56
<code>\noindent</code>	58	<code>\textit</code>	56
<code>\nolinebreak</code>	14	<code>\textmd</code>	56
<code>\nonumber</code>	43	<code>\textnormal</code>	56
<code>\nopagebreak</code>	14	<code>\textrm</code>	44, 56
<code>\normalsize</code>	56	<code>\textsc</code>	56
<code>\overbrace</code>	38	<code>\textsf</code>	56
<code>\overleftarrow</code>	39	<code>\textsl</code>	56
<code>\overline</code>	38	<code>\textstyle</code>	44
<code>\overrightarrow</code>	39	<code>\texttt</code>	56
<code>\pagebreak</code>	14	<code>\textup</code>	56
<code>\pageref</code>	21	<code>\thispagestyle</code>	11
<code>\pagestyle</code>	11	<code>\tiny</code>	56
<code>\paragraph</code>	20	<code>\title</code>	21
<code>\parindent</code>	58	<code>\ud</code>	42
<code>\parskip</code>	58	<code>\underbrace</code>	38
<code>\part</code>	20	<code>\underline</code>	38
<code>\pmb</code>	47	<code>\usepackage</code>	11, 18
<code>\pmod</code>	39	<code>\vdots</code>	41
<code>\printindex</code>	63	<code>\vec</code>	39
<code>\providecommand</code>	32	<code>\verb</code>	25, 26
<code>\qqquad</code>	36, 41	<code>\vspace</code>	59
<code>\quad</code>	36, 41	<code>\widehat</code>	38
<code>\ref</code>	21, 36	<code>\widetilde</code>	38
<code>\renewcommand</code>	32	commentaires	6
<code>\renewenvironment</code>	33	crochets	5, 40
<code>\right</code>	40	cédille	18
<code>\right.</code>	40, 42		
<code>\scriptscriptstyle</code>	44	D	
<code>\scriptsize</code>	56	<code>\date</code>	21
<code>\scriptstyle</code>	44	<code>dcolumn</code>	27
<code>\section</code>	19	<code>\ddots</code>	41
<code>\setlength</code>	58, 60	<code>description</code>	23
<code>\sloppy</code>	14	deux colonnes	9
<code>\small</code>	56	dimensions	58
<code>\sqrt</code>	38	<code>displaymath</code>	35
<code>\stretch</code>	59	<code>\displaystyle</code>	44
<code>\subparagraph</code>	20	disposition d'une page	60
<code>\subsection</code>	20	<code>\documentclass</code>	8
<code>\substack</code>	40	double interligne	57
<code>\subsubsection</code>	20	double-face	9
<code>\sum</code>	40	délimiteurs	40
<code>\tableofcontents</code>	20		

- E**
- EC 15
 - \emph 22, 56
 - emplacement 28
 - empty 11
 - en-tête 11
 - \end 22
 - enumerate 23
 - environnement 22
 - environnements
 - array 42
 - center 24
 - description 23
 - displaymath 35
 - enumerate 23
 - eqnarray 42
 - equation 36
 - figure 28, 29, 64
 - flushleft 24
 - flushright 24
 - itemize 23
 - math 35
 - quotation 25
 - quote 24
 - subarray 40
 - table 28, 29
 - tabular 26
 - thebibliography 60
 - verbatim 25
 - verse 25
 - eqnarray 42
 - equation 36
 - équations
 - longues 43
 - système d' 42
 - espace 4
 - après une commande 5
 - en début de ligne 4
 - espace insécable 19
 - espacement
 - horizontal 58
 - mathématique 41
 - vertical 59
 - eucal 54
 - eufрак 54
 - executive (papier) 9
 - exposant 38
 - exscale 10, 41
 - extension 6, 10
 - extensions
 - amsfonts 37, 54
 - amsmath .. 39–41, 43, 44, 47
 - amssymb 37, 48, 54
 - babel 18, 19
 - dcolumн 27
 - eucal 54
 - eufрак 54
 - exscale 10, 41
 - fontenc 10
 - french 16–19, 23, 58, 62
 - graphicx 64
 - ifthen 10
 - indentfirst 58
 - inputenc 10, 15, 18
 - latexsym 10
 - layout 60
 - makeidx 10, 62
 - mathrsfs 54
 - setspace 57
 - showidx 63
 - syntonly 10
- F**
- fichier source 4
 - figure 28, 29, 64
 - fleqn 9
 - flushleft 24
 - flushright 24
 - fontenc 10
 - \footnote 22
 - \footnotesize 56
 - formules 35
 - \frac 39
 - fraction 39
 - french 16–19, 23, 58, 62
 - \frenchspacing 19
 - \frontmatter 21
 - \fussy 14

-
- G**
- graphicx 64
 - graphiques 64
 - grec
 - alphabet 37
 - groupe 57
 - guillemets 16
- H**
- headings 11
 - \hline 26
 - horizontal
 - accolade 38
 - trait 38
 - \hspace 58
 - \Huge 56
 - \huge 56
 - Hyphenation 14
 - \hyphenation 15
- I**
- i et j sans points 18
 - ifthen 10
 - \include 12
 - \includegraphics 64
 - \includeonly 12
 - \indent 58
 - indentfirst 58
 - index 62
 - \index 63
 - indice 38
 - \input 12
 - inputenc 10, 15, 18
 - \int 40
 - interligne 57
 - international 18
 - internet 19
 - intégrale 40
 - italic 56
 - \item 23
 - itemize 23
- J**
- justification 13
- K**
- Knuth, Donald E. 1
- L**
- \label 21, 36
 - Lamport, Leslie 1
 - \LARGE 56
 - \Large 56
 - \large 56
 - L^AT_EX 2.09 1
 - L^AT_EX 2_ε 1
 - L^AT_EX3 1
 - L^AT_EX3. 4
 - latexsym 10
 - layout 60
 - \ldots 17, 41
 - \left 40
 - legal (papier) 9
 - leqno 9
 - letter (papier) 9
 - \lettrine 58
 - ligature 17
 - \linebreak 14
 - \linespread 57
 - \listoffigures 30
 - \listoftables 30
 - livre 8
- M**
- \mainmatter 21
 - makeidx 10, 62
 - \makeindex 63
 - makeindex 63
 - \maketitle 21
 - marges 60
 - math 35
 - \mathbb 37
 - \mathbf 56
 - \mathcal 56
 - \mathit 56
 - \mathnormal 56
 - \mathrm 44, 56
 - mathrsfs 54
 - \mathsf 56

-
- `\mathtt` 56
`\mathversion` 46
 mathématiques 35
 accents 38
 espaces 41
 fonctions 39
`\mbox` 15, 17
 Mittelbach, Frank 1
 modulo 39
 moins (signe) 16
`\multicolumn` 27

N
`\newcommand` 31
`\newenvironment` 32
`\newline` 14
`\newpage` 14
`\newtheorem` 45
`\noindent` 58
`\nolinebreak` 14
`\nonumber` 43
`\nopagebreak` 14
`\normalsize` 56
 notitlepage 9

O
 objets flottants 28
 œ 18
 oneside 9
 openany 9
 openright 9
 option 8
`\overbrace` 38
 overfull hbox 14
`\overleftarrow` 39
`\overline` 38
`\overrightarrow` 39

P
`\pagebreak` 14
`\pageref` 21
`\pagestyle` 11
 papier
 A4 9
 A5 9
 letter 9
 taille du 9
`\paragraph` 20
 paramètre 5
 paramètres optionnels 5
 parenthèses 40
`\parindent` 58
`\parskip` 58
`\part` 20
 pieds de page 11
 plain 11
`\pmb` 47
`\pmod` 39
 point 16
 points de suspension 17, 41
 diagonaux 41
 verticaux 41
 police 55
 PostScript Encapsulé 64
 prime 38
`\printindex` 63
`\providecommand` 32
 préambule 6

Q
`\qqquad` 36, 41
`\quad` 36, 41
 quotation 25
 quote 24

R
 racine carrée 38
 rapport 8
 recto simple 9
 recto-verso 9
`\ref` 21, 36
`\renewcommand` 32
`\renewenvironment` 33
 report (classe) 8
`\right` 40
`\right.` 40, 42
 roman 56
 références croisées 21

S

-
- sans serif 56
 - scandinaves (caractères) 18
 - `\scriptscriptstyle` 44
 - `\scriptsize` 56
 - `\scriptstyle` 44
 - `\section` 19
 - `\setlength` 58, 60
 - setspace 57
 - showidx 63
 - simple face 9
 - slanted 56
 - slides (classe) 8
 - `\sloppy` 14
 - `\small` 56
 - small caps 56
 - somme 40
 - `\sqrt` 38
 - `\stretch` 59
 - style de page
 - empty 11
 - headings 11
 - plain 11
 - subarray 40
 - `\subparagraph` 20
 - `\subsection` 20
 - `\substack` 40
 - `\subsubsection` 20
 - `\sum` 40
 - symboles
 - gras 46
 - mathématiques 48
 - syntonly 10
 - systèmes d'équations 42

 - T**
 - T1 15
 - table 28, 29
 - tableau 28
 - `\tableofcontents` 20
 - tabular 26
 - taille
 - de la police par défaut 9
 - des polices 55
 - des polices mathématiques .. 43
 - du papier 9
 - prédéfinies 56
 - `\text` 44
 - `\textbf` 56
 - `\textit` 56
 - `\textmd` 56
 - `\textnormal` 56
 - `\textrm` 44, 56
 - `\textsc` 56
 - `\textsf` 56
 - `\textsl` 56
 - `\textstyle` 44
 - `\texttt` 56
 - `\textup` 56
 - thebibliography 60
 - `\thispagestyle` 11
 - tilde 19, 38
 - `\tiny` 56
 - tiret 16
 - `\title` 21
 - titlepage 9
 - titre du document 9, 21
 - transparents 8
 - twocolumn 9
 - twoside 9

 - U**
 - `\ud` 42
 - umlaut 18
 - `\underbrace` 38
 - underfull hbox 14
 - `\underline` 38
 - unités 58, 59
 - upright 56
 - `\usepackage` 11, 18

 - V**
 - `\vdots` 41
 - `\vec` 39
 - vecteurs 39
 - `\verb` 25, 26
 - verbatim 25

verse	25
vertical	
espacement	59
points de suspension	41
virgule	16
\vspace	59
W	
\widehat	38
\widetilde	38
WYSIWYG	2, 3