

Rédigez des documents de qualité avec LaTeX

Par Laleloulilo



www.openclassrooms.com

*Licence Creative Commons 6 2.0
Dernière mise à jour le 29/11/2012*

Sommaire

Sommaire	2
Lire aussi	4
Rédigez des documents de qualité avec LaTeX	6
Partie 1 : Découverte de LaTeX	8
Qu'est-ce que LaTeX ?	8
LaTeX, quésaco ?	8
Word versus LaTeX	8
Comparatif général	9
Les avantages de LaTeX	10
LaTeX et la rétrocompatibilité	10
LaTeX et les publications	11
LaTeX dans les publications scientifiques	11
LaTeX dans le monde de l'éducation	11
LaTeX et les présentations	11
Historique rapide	11
En résumé	12
Installer LaTeX	14
De quels logiciels avons-nous besoin ?	14
L'installation typique	14
Installation sur Windows	15
MiKTeX, la distribution LaTeX	15
Ghostscript et Adobe Reader, vos lecteurs de PostScript et de PDF	17
TeXnicCenter, l'éditeur LaTeX	17
Installation sur Mac OS	18
Installation sur Linux	20
En résumé	21
Structurer son premier document	21
La compilation	22
Rédaction d'un document simple	23
Rédaction du fichier .tex	23
Compilation	24
Les résultats	26
Types de documents et caractères spéciaux	26
Les caractères spéciaux	26
Types de documents et portions de texte	27
En résumé	29
Les packages	30
La philosophie des packages	30
Comment s'en servir ?	30
Comment installer un package ?	32
MiKTeX et son installation à la volée	32
Autres distributions	34
En résumé	35
Partie 2 : Utilisation basique de LaTeX	35
Maîtriser sa mise en page (1/2)	36
Structure des documents	36
Généralités	36
Modification de la numérotation des éléments de structure	38
Page de garde	40
La théorie	40
La pratique	40
Alignements de texte et sauts	41
Alignements de texte	41
Sauts	42
En résumé	43
Maîtriser sa mise en page (2/2)	45
La forme de votre publication	45
La commande \documentclass{}	45
Exemples pratiques	45
Marges et interlignes	46
Marges	46
Interlignes	48
Les listes	49
Listes à puces	49
Listes numérotées	50
Listes de description	51
Les styles	52
En résumé	53
Les polices	53
Graisie, style, taille	54
Taille de texte	54
Graisie, soulignement	55
La commande \emph, un cas à part	56
Couleur	57

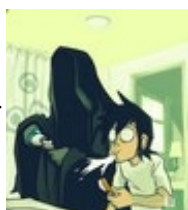
Les huit couleurs par défaut	57
Création de nouvelles couleurs	57
Les packs de polices	58
En résumé	62
Les notes	62
Citation, code brut et URL	63
Les citations simples	63
Les citations de code	64
Gestion des URL	66
Minipage et texte encadré	67
La commande \fbox	67
L'environnement minipage	67
Notes de bas de page, références internes	68
Notes de bas de page	68
Les références internes	70
En résumé	71
TP 1 – Mise en page de la Déclaration universelle des droits de l'homme	72
Les consignes	72
Les contraintes de l'exercice	72
La correction	77
Étape 1 : le type de document	77
Étape 2 : la langue	77
Étape 3 : les autres packages	78
Étape 4 : la page de garde	78
Étape 5 : le squelette complet	79
Étape 6 : le contenu	81
Étape 7 : rassembler les morceaux	91
Conclusion	96
Aller plus loin	96
Commandes et espaces	96
Marges et layout	97
Personnalisation de la commande \emph	101
En-têtes et pieds de pages personnalisés	102
En résumé	103
Partie 3 : Utilisation avancée de LaTeX	104
Les figures	104
Les différents formats d'images	104
Insertion d'images et de figures	105
Insertion simple	106
Intégration d'une image dans un paragraphe	110
Les flottants	111
L'environnement figure	111
Les légendes	113
Sauts de page	114
En résumé	114
Les tableaux	114
Structure type d'un tableau	115
La théorie	115
Exercice	117
Fusion de cellules	118
Fusion de colonnes	118
Fusion de lignes	118
Autres paramètres applicables à un tableau	119
Modifications mineures	120
Exercice	121
Utilisation de commandes et d'environnements dans un tableau	121
Colorer des cellules	123
Exercice	123
Un environnement flottant : table	124
En résumé	126
Les mathématiques	126
Saisir une expression mathématique	127
Fonctions usuelles	128
Fonctions usuelles à arguments	128
Fonctions usuelles sans argument	130
Les intégrales	131
Preamble technique	131
Intégrales et expressions plus complexes	132
Les systèmes d'équations	135
L'environnement eqnarray	136
L'environnement align	137
L'environnement array	138
Les matrices	140
Les différents environnements	140
Exemple pratique	143
La commande 	144
Mise en forme de théorèmes, lemmes... ..	145
Généralités	145
Pratiquons !	145
Flèches, symboles étirables et espaces	147
Les espaces	148
Les accents et chapeaux divers	149

En résumé	151
Sommaire et index	151
Tables des matières	152
Table des matières simple	152
Paramétrage d'une table des matières	153
Tables des figures et tableaux	158
Les index	160
Théorie	160
Pratique	162
En résumé	163
La bibliographie	163
Principe d'une bibliographie	164
La base de données	165
Mise en place de la bibliographie	168
Concept	168
Exemples	168
Compilation	170
En résumé	171
TP 2 – Rédaction d'un article présentant le théorème de Pythagore	173
Les consignes	173
La structure	173
Les contenus	173
Petite aide	176
La correction	176
Être plus à l'aise avec LaTeX	180
Écrire un document dans plusieurs fichiers .tex	180
L'option draft	182
Les unités de longueur	182
Les espaces	183
En résumé	183
Partie 4 : Annexes	183
Les caractères spéciaux	184
Environnement mathématique	184
Lettres grecques	184
Opérateurs extensibles	185
Fonctions	185
Flèches	186
Opérateurs	187
Accents	192
Polices mathématiques	192
Caractères de structure	193
Divers	194
Autres caractères spéciaux	195
Liste des packages	197
La liste	197
Packages de langue	197
Création d'un layout	197
Modification des marges	197
Interligne	197
Soulignement	197
Symbole euro	197
Pack de polices	197
Citation d'URL	198
Citation de code	198
Citation de code coloré	198
En-têtes et pieds de pages personnalisés	198
Insertion d'images	198
Insertion d'une image dans un paragraphe	198
Manipuler les couleurs et colorer du texte	198
Colorer du texte contenu dans un tableau	198
Insertion d'expressions scientifiques	199
Création d'index	199
Petite astuce	199
Les gabarits	199
La classe article	200
La classe report	200
La classe book	201
La classe letter	201
Mémento	202
Les essentiels	203
Squelette minimal	203
Appel d'un package	203
Introduction d'un environnement	203
Inclusion d'un fichier	203
La commande \documentclass	203
Les références	204
Les éléments de mise en page	204
Éléments de structure	204
Page de garde	204
Alignements de texte	205
Sauts	205

Interlignes	205
Marges	205
Listes	206
Les polices	207
Les tailles	207
Les couleurs	207
Changement de police ponctuel	208
Les packs de polices	209
Les effets	209
Les notes	210
Citations simples	210
La commande \fbox	210
Création de minipage	210
Les figures	211
Insertion simple	211
Rotation	211
Inclusion dans un paragraphe	211
Découpage d'une image	211
Image flottante	211
Les tableaux	212
Structure type	212
Fusion de lignes	212
Fusion de colonnes	212
Coloration de cellules	212
Les mathématiques	213
Les formules simples	213
Les systèmes d'équation	214
Les matrices	215
Les théorèmes, lemmes, etc.	215
Les espaces	216
Les tables	216
Index	216
Bibliographie	216
Base de données	216
Insertion de la bibliographie	218
Aller plus loin	218
Typographie et encodage	219
Latin-1 et UTF-8	219
L'intégriste est partout	219
De l'art de la présentation	219
Les cours d'approfondissement	219
Les incontournables	220
Beamer	220
Les interfaces graphiques alternatives	220



Rédigez des documents de qualité avec LaTeX

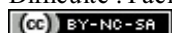


Par

Laleloulilo

Mise à jour : 29/11/2012

Difficulté : Facile Durée d'étude : 15 jours

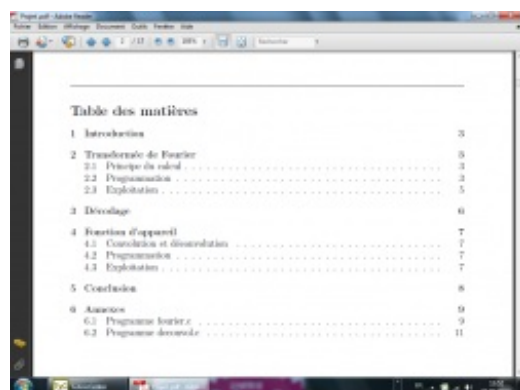


LaTeX ? Mais que vient faire le LaTeX sur un site de tutoriel informatique ? C'est un langage ? A quoi sert il ?

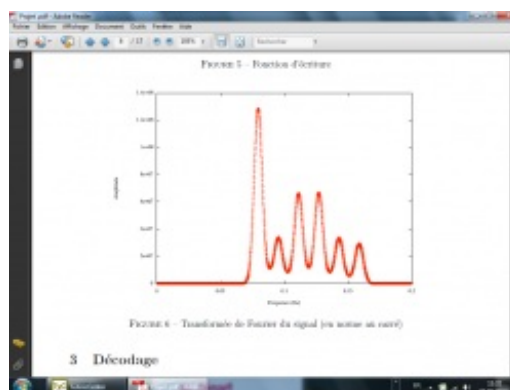
Telles sont les questions que peuvent éveiller chez vous la lecture du titre de ce tutoriel. LaTeX est un langage informatique permettant de mettre en page des documents de la manière la plus professionnelle qui soit. Ce langage a été conçu pour rendre la création de document facile pour l'auteur d'un côté, et produire des documents lisibles et clairs du côté des lecteurs. Les documents produits sont la plupart du temps au format pdf.

Les documents créés en LaTeX respectent les standards de l'édition et de l'imprimerie, tout en étant optimisés pour l'affichage de formule mathématiques, faisant de ce langage un outil de choix pour les scientifiques et les éditeurs (il est à noter que dans certains laboratoires de recherche, Word est interdit et seul LaTeX est utilisé).

Les documents produits sous LaTeX sont souvent sobres mais extrêmement lisibles, en voici quelques exemples :

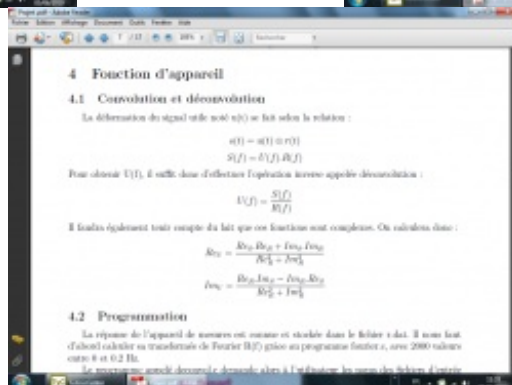


un sommaire en LaTeX



insertion

d'image en LaTeX



Maths en LaTeX

Ce cours est conçu pour vous apprendre le langage LaTeX à partir de zéro et faire de vous des Jedis de la mise en page, il comportera trois parties durant lesquelles vous apprendrez à :

- **installer** LaTeX et créer votre premier document ;
- **modifier** les réglages du programme ;
- **insérer** images, tableaux, listes à puces...

Toutes les fonctions utiles seront détaillées, le but étant de vous permettre d'être réellement à l'aise avec LaTeX à la fin de ce tutoriel, raison pour laquelle ce tutoriel sera ponctué d'exercices variés.

Je vous attends avec mon sabre laser et mon clavier dans le premier chapitre. 🤖



Ce cours vous plaît ?

Si vous avez aimé ce cours, vous pouvez retrouver le livre "*Rédigez des documents de qualité avec LaTeX*" du même auteur, en vente [sur le Site du Zéro](#), en librairie et dans les boutiques en ligne. Vous y trouverez ce cours adapté au format papier avec une série de chapitres inédits.

[Plus d'informations](#)

Partie 1 : Découverte de LaTeX

Cette partie est la plus importante du cours. Elle développe l'utilisation globale, l'installation de LaTeX, et vous permettra de créer votre premier document.

En somme, tout un programme. 😊

Qu'est-ce que LaTeX ?

LaTeX, vous en avez déjà entendu parler, mais ce nom vous intrigue. À quoi sert-il ? Est-il réellement utile ? Quels sont ses avantages et inconvénients ? Pourquoi a-t-il été créé, et quelle est son histoire ?

Toutes ces questions trouveront leurs réponses dans ce chapitre. Du rôle de LaTeX dans l'avancée du journalisme scientifique à ses caractéristiques techniques, en passant par un court portrait de ses créateurs, cette introduction vous apportera d'intéressantes informations culturelles et techniques. Bien entendu, nous verrons quelques exemples de la mise en page qu'il permet, afin de mieux comprendre les capacités de ce langage (car c'en est un). Et tout vous sera expliqué à partir de... zéro.

LaTeX, quésaco ?



LaTeX se prononce « latec » ou « latèque », mais certainement pas « latex ». Il est important de vous en souvenir, sinon vous risquerez de provoquer de nombreux quiproquos avec vos collègues lorsque vous parlerez de LaTeX.

L
A
T
E
X

Le logo de LaTeX

LaTeX est un langage créé pour séparer le fond de la forme lors de la création d'un document ou d'une publication. Plus clairement, l'auteur tape des instructions dans une sorte de bloc-notes et structure son texte grâce à des mots et des commandes propres à LaTeX. Par exemple, l'auteur peut indiquer à LaTeX de placer la première partie de son texte en gras, et une autre en italique. En somme, il **décrit** comment il veut hiérarchiser l'information. Ensuite, son code est traité par un logiciel : LaTeX choisit alors les meilleurs agencements et la disposition optimale pour chacun des éléments du document.

En résumé, LaTeX est un **langage de description** donnant à l'auteur les moyens d'obtenir des documents mis en page de façon professionnelle sans avoir à se soucier de leur forme. La priorité est donnée à l'essentiel : le contenu.



LaTeX est une arme à double tranchant, car s'il a tendance à générer une mise en page presque parfaite, il est souvent très difficile d'agencer les éléments de façon plus exotique. LaTeX est conçu pour créer des documents lisibles et beaux. Bien souvent, quand il essaiera de changer votre mise en page (car pour lui, elle n'est pas adaptée), il aura raison.

Un soin tout particulier a été consacré à l'élaboration de commandes permettant d'écrire très facilement d'impeccables formules mathématiques. La figure suivante est un exemple de rendu d'une formule compliquée, qui n'a pourtant requis qu'une seule ligne de code LaTeX. Cette spécificité et cette facilité d'écriture des formules font de LaTeX un outil très diffusé dans le monde scientifique.

$$\sum_{k=2}^{47} k + 1$$

Une formule

Word versus LaTeX

Nous allons maintenant développer les différences entre LaTeX et Word à travers un comparatif global de leurs caractéristiques, suivi d'un résumé traitant des avantages que présente LaTeX.

Comparatif général

Que ce soit dans le cadre de votre travail ou de celui de vos études, vous avez sûrement déjà été confronté à la lourde et ingrate tâche que représentent l'écriture et la mise en page de longs rapports. Rédiger un document de plus de 15 pages demande déjà un grand travail et, bien souvent, la mise en page sur Word donne du fil à retordre.

La cause principale ? L'utilisateur doit gérer à la fois le fond et la forme sur un logiciel fonctionnant comme Word. Trop souvent, il s'enlise dans ces manipulations et perd du temps.



Word reste l'outil le plus développé à l'heure actuelle. La concurrence est rude entre les suites logicielles de bureautique, et il y a fort à parier qu'au fil des ans, LaTeX et Word se rapprocheront énormément, tant au niveau de la facilité d'utilisation que de l'optimisation de la prise en main.

LaTeX fait peur aux débutants à cause de son interface austère, mais Word n'aide pas ces derniers à réellement apprendre l'emploi correct de ses outils. Ces deux problèmes et la manière dont ils seront résolus constitueront des facteurs prépondérants dans la progression respective des deux logiciels sur le marché de l'informatique.

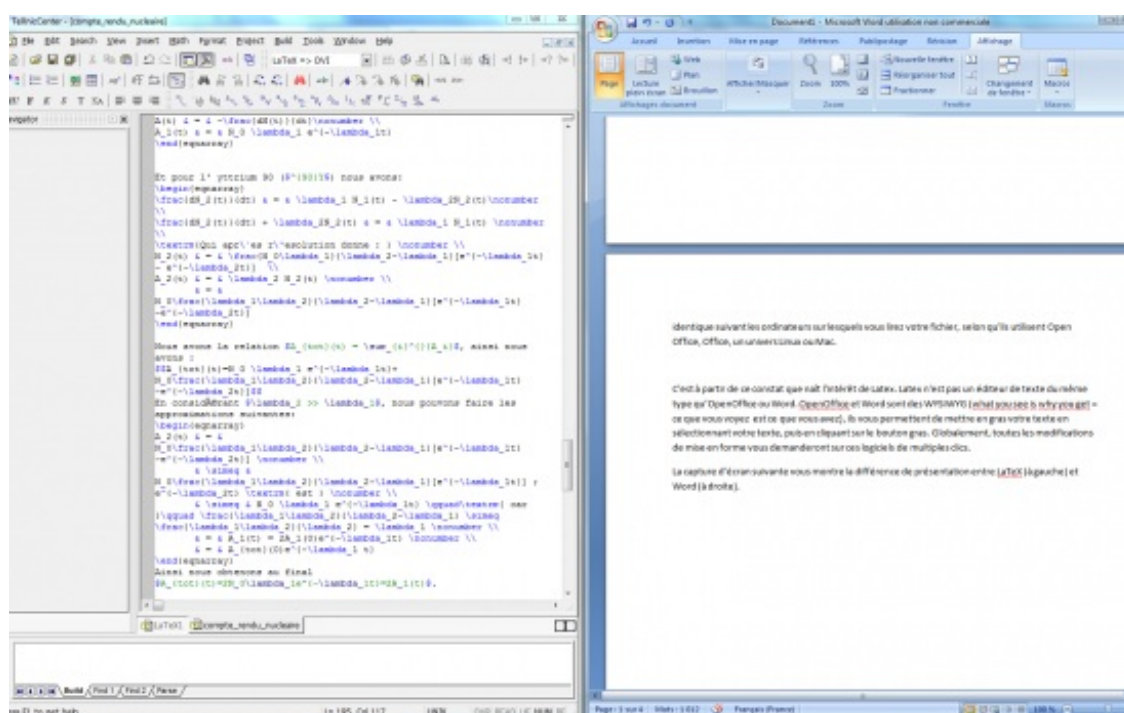
Concrètement, les modifications deviennent vite un cauchemar une fois qu'on a sélectionné et modifié des dizaines de petits éléments. Cela n'est qu'un exemple, mais dans vos publications, vous avez sûrement déjà eu besoin de :

1. mettre en page des images ;
2. légender les figures ;
3. écrire des formules mathématiques (sous Word, c'est extrêmement fastidieux) ;
4. dessiner des schémas ;
5. créer des tableaux.

Autant de tâches indispensables, mais qui prennent du temps (même si elles sont faites dans les règles de l'art), à un point tel que nous pouvons avancer que, pour un utilisateur expérimenté, le temps consacré à la rédaction représente moins de 50 % du temps passé sur un long document.

C'est à partir de ce constat que naît l'intérêt de LaTeX. LaTeX n'est pas un traitement de texte du même type qu'OpenOffice et Word. OpenOffice et Word sont ce qu'on appelle des WYSIWYG (*what you see is what you get* = ce que vous voyez est ce que vous obtenez). Ils vous permettent de mettre en gras votre texte en le sélectionnant et en cliquant sur le bouton gras : vous voyez alors votre texte en gras.

La figure suivante vous montre la différence de présentation entre LaTeX (à gauche) et Word (à droite).



LaTeX vs Word

J'insiste là-dessus : LaTeX ne se comporte et ne s'utilise pas de la même façon que Word. Une fois que vous aurez choisi quel

type de publication vous écrivez (livre, article, C.V., lettre...), vous n'aurez plus qu'à taper votre texte au kilomètre dans son éditeur, en mettant en avant les paragraphes, les titres, ou les mots importants de la manière suivante (ici, un morceau de texte est mis en italique) :

Code : TeX

```
La liberté consiste à pouvoir faire \textit{tout ce qui ne nuit pas}
à autrui...
```

LaTeX, à partir de votre texte, créera une mise en page lisible et adaptée, mettra en valeur vos mots en gras, vos titres, vos images et légendes. De façon plus large, c'est un langage créé pour respecter les normes éditoriales et typographiques, qui vous permet d'écrire sans avoir à vous soucier de la mise en page. L'idée de taper des balises et des commandes en plein milieu du texte peut sembler étrange au début, mais vous découvrirez très vite qu'elle simplifie la vie.



L'optimisation de la création de documents typographiquement propres est l'argument principal jouant en faveur de l'utilisation de LaTeX. En deuxième rang, nous pourrions citer la gratuité, bien qu'OpenOffice soit également gratuit.

Une attention toute particulière a été apportée à la simplification de l'écriture des formules mathématiques et autres expressions scientifiques, souvent enfantées dans la douleur avec des outils comme l'éditeur d'équations de Word. Nous apprendrons bien entendu à maîtriser cette partie de LaTeX.



Information : il existe des éditeurs d'équations performants permettant de recopier des formules au sein de Word. Néanmoins, LaTeX intègre nativement ce genre d'outil et il n'est pas nécessaire d'installer de programme annexe pour effectuer des opérations sur un schéma ou une expression mathématique.

Les avantages de LaTeX

Il est possible que dans un élan de folie, un auteur estime un élément de mise en page peu adapté ou souhaite simplement changer la police d'un titre. Le voilà donc face à un problème déjà suggéré plus haut : comment modifier un style de titre présent sur 60, 100 voire 200 pages ? Si certains outils WYSIWYG ont eux aussi apporté une solution à ce problème, sachez qu'avec LaTeX, il vous sera possible de changer les caractéristiques de tous vos titres en une fois, de sorte que votre modification soit appliquée sur toutes les pages.

En réalité, avec LaTeX, tout est modifiable, tout est paramétrable. Le logiciel vous donnera une base par défaut propre et structurée pour mettre en page vos publications, mais vous serez libre de modifier les moindres marges, alignements et styles selon vos désirs. Que du bonheur !

Si nous devons lister d'autres qualités de LaTeX, nous pourrions citer en vrac :

- sa gestion des notes ;
- sa gestion des flottants (dont nous parlerons dans la troisième partie du livre) ;
- sa gestion des longs documents ;
- la possibilité qu'il offre de hiérarchiser facilement du texte en parties, chapitres, sections... ;
- la possibilité qu'il offre de faire des références (à un paragraphe, une figure...) de façon très simple et souple ;
- la possibilité qu'il offre de générer automatiquement des bibliographies, index et tables des matières.

LaTeX et la rétrocompatibilité

Rétrocompatibilité quoi ? La rétrocompatibilité, c'est la capacité qu'a un logiciel de produire des fichiers lisibles par ses versions précédentes qui n'ont pas été mises à jour. Vous avez sûrement déjà été confronté au problème causé par une différence de version avec une présentation sur PowerPoint. Si vous n'avez jamais eu le souci sous PowerPoint, vous l'avez probablement rencontré sous Word : un changement de PC peut miraculeusement décaler tous les paragraphes de votre cher rapport et rendre tous les schémas illisibles (l'expérience prouve que l'individu lambda se sent très bête dans ce genre de cas). Ces deux problèmes sont dits de rétrocompatibilité, causés par des logiciels ne produisant pas des fichiers compatibles avec leurs versions précédentes.

Sur LaTeX, ce genre de complication ne survient que très rarement. Un document écrit sur un ordinateur A aura, dans la grande majorité des cas, exactement le même rendu sur un ordinateur B, quelle que soit la version de LaTeX utilisée et le système d'exploitation (Linux, Windows, Mac OS...).

Ajoutons à cela qu'il est souvent possible de produire avec LaTeX des documents en PDF, un format international de mise en page, connu pour ne pas déformer les documents et présenter le même rendu à peu près sur toutes les machines.

Suffisamment appréciable pour être noté, n'est-ce pas ?



Il peut arriver qu'une extension du logiciel, que nous appellerons **package**, entre en conflit avec une de ses versions précédentes. Néanmoins, cela reste exceptionnel.



Les suites logicielles classiques de traitement de texte intègrent depuis peu la fonction de création de PDF. L'initiative de LaTeX semble avoir eu du succès.

LaTeX et les publications

Nous venons de traiter les points qui font de LaTeX un langage de qualité, à savoir :

1. la rétrocompatibilité ;
2. la création de PDF ;
3. la possibilité de lire ses documents sur tous les systèmes d'exploitation ;
4. la présence d'outils permettant la modification rapide de l'ensemble des éléments formant la mise en page ;
5. la possibilité de passer (enfin) plus de temps à écrire du contenu qu'à le mettre en forme.

Autant d'avantages qui ont fait de LaTeX le chouchou du monde de l'édition et du monde scientifique... Tant et si bien que bon nombre d'ouvrages sont édités en LaTeX, des professionnels de ce langage se faisant payer par des maisons d'édition pour mettre en page des livres de manière irréprochable.

Nous allons nous pencher sur les applications de ce langage dans les publications et le milieu scientifique.

LaTeX dans les publications scientifiques



Les Livres du Zéro sont écrits en LaTeX

Point important : les scientifiques communiquent essentiellement entre eux grâce à des articles. Ces articles sont la plupart du temps écrits en LaTeX. Ces publications sont légion et chaque semaine, des dizaines de revues scientifiques trient, sélectionnent et mettent en page des articles traitant des innovations et découvertes récentes. En somme, si la science fait avancer le monde, LaTeX est à l'heure actuelle son format de prédilection.

D'ailleurs sachez que les « Livres du Zéro » sont écrits en LaTeX (y compris ce livre que vous tenez dans les mains !) (figure suivante), comme la majorité des fascicules et livres de cours que vous avez pu consulter par le passé.

LaTeX dans le monde de l'éducation

Je suis actuellement étudiant à l'université de Bordeaux 1 : sachez qu'à partir de la première année de Master, dans certaines matières (notamment la physique et les maths), il est obligatoire de rendre ses rapports de stage, mémoires et autres textes au format LaTeX. Il est également utilisé de façon extrêmement fréquente dans les laboratoires de recherche grâce à l'aspect pratique du format PDF (il est facile à exporter et à lire sur les ordinateurs de tous les pays).

LaTeX et les présentations

Notez que LaTeX permet non seulement de faire des rapports, mais aussi de créer des présentations contenant des *slides* (diapositives) de qualité incomparable, et bien plus encore ! Mais nous en parlerons plus loin dans ce livre.

Historique rapide

Difficile d'enseigner un langage informatique sans survoler rapidement les raisons de sa création et les C.V. (généralement fort bien remplis) de ses pères. Nous allons ici développer l'historique de LaTeX.

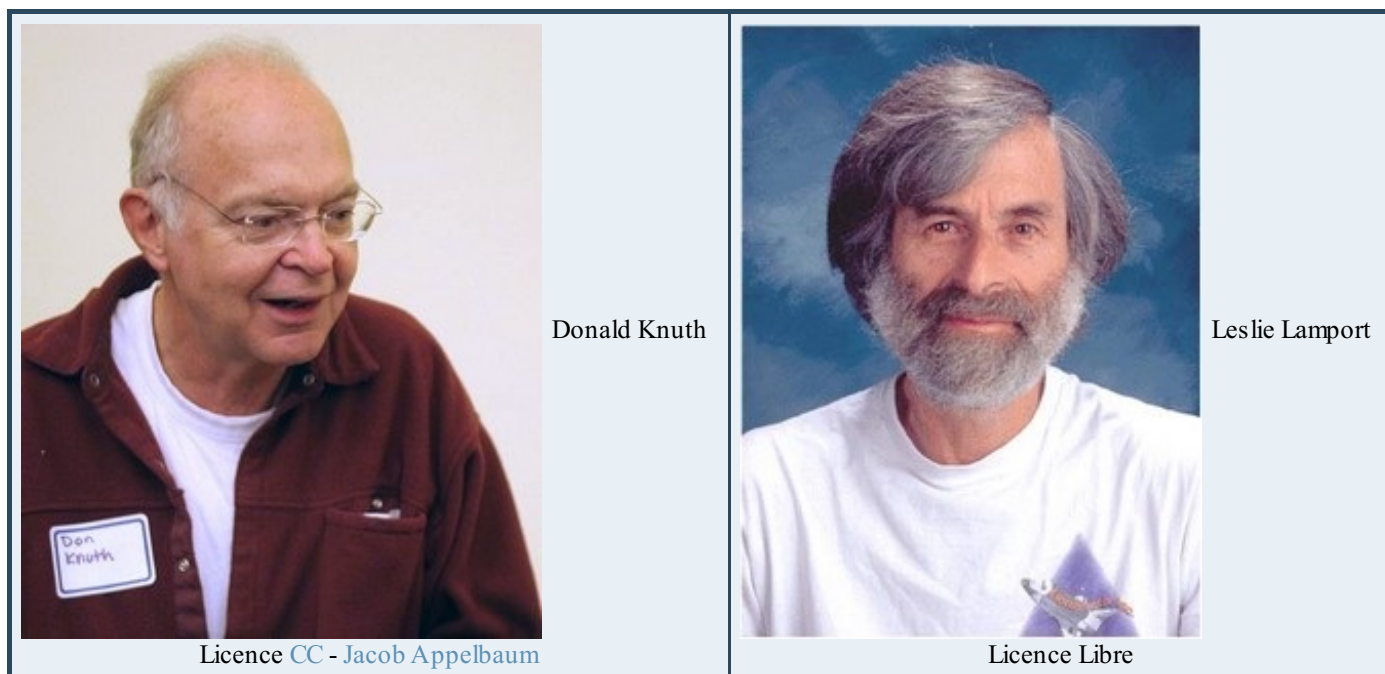
Tout commence avec la création du langage TeX par Donald Erwin Knuth (chercheur américain en informatique, né en 1938), déclenchée en 1977 par un ras-le-bol de l'inélégante mise en page de ses articles mathématiques publiés par l'*American Mathematical Society*.

Comme souvent dans le monde informatique, un besoin a été la raison de l'innovation, en l'occurrence : TeX. Le langage avait été conçu non pas pour produire de beaux documents, mais pour accroître la lisibilité et optimiser l'insertion de formules mathématiques.

Nous ne sommes pas des typographes, ni des experts de la mise en page, mais ceux qui ont participé à la création de TeX le sont, et ils ont créé un ensemble cohérent de commandes permettant à l'utilisateur d'obtenir un résultat professionnel.

LaTeX est une surcouche de TeX et n'a été créé que plus tard, plus exactement en 1985 par Leslie Lamport (chercheur en informatique américain, né en 1941). TeX, bien que révolutionnaire, était relativement compliqué à utiliser. LaTeX permit une simplification énorme de bon nombre de tâches grâce à des **macros** intégrées au programme.

Les pères de LaTeX : Donald Knuth à gauche, Leslie Lamport à droite.



Une macro est une commande qui vous permet de lancer une série d'actions à votre place. Cela peut aller du double-clic à des processus bien plus compliqués. Ici, nous n'allons pas nous en préoccuper : les macros sont intégrées à LaTeX pour nous simplifier la vie, ne nous la compliquons pas.

La dernière évolution majeure est LaTeX_E, c'est celle avec laquelle nous allons travailler dans ce cours.

LaTeX est un logiciel libre, vous pouvez donc à votre guise le distribuer, vous amuser à modifier son code (pour autant que cela vous amuse), vendre des millions de copies dans l'optique de dominer le monde... du moment que votre logiciel ne reprend pas le nom « TeX » (c'est écrit dans la licence de TeX).



Ce côté « bidouillable » de LaTeX a entraîné l'apparition de nombreux logiciels permettant la création de documents, abondance qui laisse parfois le néophyte perplexe. J'en ai choisi quelques-uns pour vous et vous guiderai dans leur installation, que vous soyez sous Mac OS, Windows, ou Linux ! Tout cela se passe dans le prochain chapitre.

En résumé

- LaTeX est un langage de description libre et gratuit. Il permet de concevoir des documents de qualité professionnelle sans connaissances en typographie et mise en page.
- Contrairement à un traitement de texte comme Word, LaTeX vous permet de vous concentrer sur le contenu de votre

document. Tout le reste est généré automatiquement par LaTeX. Pas besoin de se préoccuper de la numérotation des pages, de la création d'un sommaire, de la numérotation des figures ou encore des marges et alinéas !

- LaTeX permet de produire des documents PDF qui s'affichent de la même façon sur tous les ordinateurs, qu'ils soient sous Windows, Mac OS ou Linux.
- C'est un langage très populaire dans les études supérieures, chez les scientifiques et dans le monde de l'édition. Il excelle en particulier dans l'écriture de formules mathématiques, domaine dans lequel il fait figure d'outil de référence.
- Ce livre a été écrit en LaTeX. 😊

Installer LaTeX

Le chapitre précédent vous a permis de vous familiariser avec les caractéristiques de LaTeX et quelques-unes des raisons qui ont fait son succès.

Ce deuxième chapitre traite de façon détaillée de l'installation des logiciels permettant d'utiliser LaTeX sur votre machine. LaTeX étant libre, de nombreux environnements graphiques ont été développés, le choix est vaste et vous pourrez trouver des dizaines de méthodes et de logiciels permettant d'installer un éditeur LaTeX.

N'ayez crainte, les environnements sont différents, mais à l'intérieur, ces logiciels utilisent le même noyau dur (TeX, dont nous avons parlé dans le chapitre 1). En conséquence, un texte tapé sous Windows via un éditeur quelconque pourra être lu et modifié avec d'autres logiciels et environnements (par exemple sur des logiciels conçus pour Mac ou Linux).

De quels logiciels avons-nous besoin ?

Dans cette sous-partie, notre regard se porte sur ce que nous souhaitons obtenir grâce à LaTeX. En l'occurrence, nous souhaitons (et parviendrons à) créer des documents en PDF et en PostScript.



Qu'est-ce que PostScript ?

C'est un format de fichier rendant possible sa lecture par des imprimantes sans l'intermédiaire d'un PC, chose que les PDF ou les documents générés par Word ne permettent pas. Offrant beaucoup d'avantages, il est très utilisé au sein des laboratoires de recherche. Vous n'aurez normalement pas grand besoin d'utiliser des fichiers au format PostScript pour vos publications, néanmoins, certains chercheurs et éditeurs en sont de grands fans et ne jurent que par lui. Nous apprendrons donc à manipuler ce format.

L'installation typique

Une installation classique de LaTeX comporte trois éléments.

- **Une distribution LaTeX** : c'est un logiciel comportant toutes les composantes de LaTeX, c'est lui qui va transformer via LaTeX votre code tapé dans l'éditeur LaTeX en un document PDF ou PostScript. Vous l'aurez compris, c'est le morceau le plus important.
- **Un ou plusieurs lecteurs de PostScript et de PDF**, pour pouvoir lire et imprimer vos productions.
- **Un éditeur LaTeX** : c'est un logiciel vous permettant de taper votre texte, de mettre en couleur les différents codes utilisés, de numéroté vos lignes, etc. En somme, choisir un éditeur LaTeX vous simplifie la vie, mais **il n'est pas indispensable** : il est possible de taper du LaTeX dans un simple éditeur de texte (même si cela est moins pratique). Il propose aussi bon nombres de raccourcis et boutons utiles à la création de document (Ces boutons et raccourcis diffèrent d'un éditeur à l'autre).

Comme dit ci-dessus, vous pouvez choisir de ne pas utiliser d'éditeur LaTeX et d'employer à la place un éditeur de type Bloc-notes, ou de faire usage d'outils de coloration de texte plus évolués. Les éditeurs LaTeX colorent le code (ce qui le rend beaucoup plus lisible) et vous permettent, via des raccourcis et des fenêtres vous informant des bugs, de faciliter la production de documents.

De la figure suivante à la figure suivante, vous pouvez voir le même code, tapé respectivement dans un éditeur de texte simple (Bloc-notes), dans un éditeur de texte avec coloration (Notepad++, avec lequel les habitués de la programmation web auront sûrement beaucoup d'affinités) et pour finir, dans un éditeur LaTeX (TeXnicCenter).



Bloc-notes



Notepad++



TeXnicCenter

Par la suite, nous utiliserons un éditeur LaTeX.

Installation sur Windows

Notre installation logicielle pour LaTeX sur Windows se déroulera en trois temps, je vais vous les détailler.

MiKTeX, la distribution LaTeX

Tout d'abord nous allons installer **MiKTeX**, une des distributions LaTeX disponibles pour Windows. Pour l'installer, la procédure est plutôt simple, mais elle vous demandera de la **minutie**.

- Créez un dossier appelé `Latex` sur votre bureau ou dans votre dossier `Mes documents`, il servira à ranger la masse de fichiers que MiKTeX utilise pour son fonctionnement interne.
- Allez sur le site de MiKTeX : [Site web de MiKTeX](http://www.miktex.org).

Dans la colonne de gauche, vous trouverez un titre « **Download** » (figure suivante) avec en dessous un lien vers la page de téléchargement de la dernière version de MiKTeX, **cliquez dessus**.



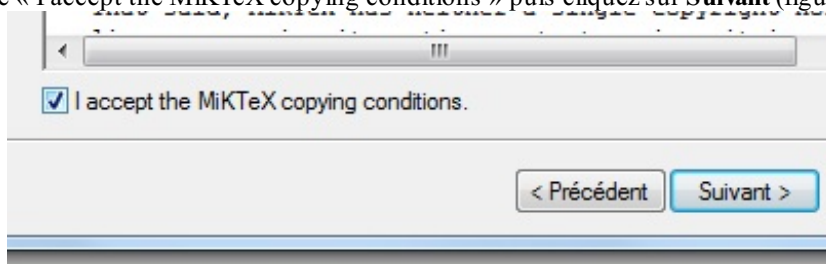
Lien de téléchargement de MiKTeX

- Dans la nouvelle page, vers le bas, vous trouverez un paragraphe intitulé « Installing the complete MiKTeX system » comportant un bouton **download**, vous devez cliquer dessus et enregistrer le fichier dans votre dossier `Latex`. Télécharger ce fichier devrait être rapide, il pèse environ 3 Mo.



Le fichier que vous avez téléchargé est ce qu'on appelle un gestionnaire d'installation. Dans notre cas, il va dans un premier temps télécharger MiKTeX, et dans un second temps il l'installera.

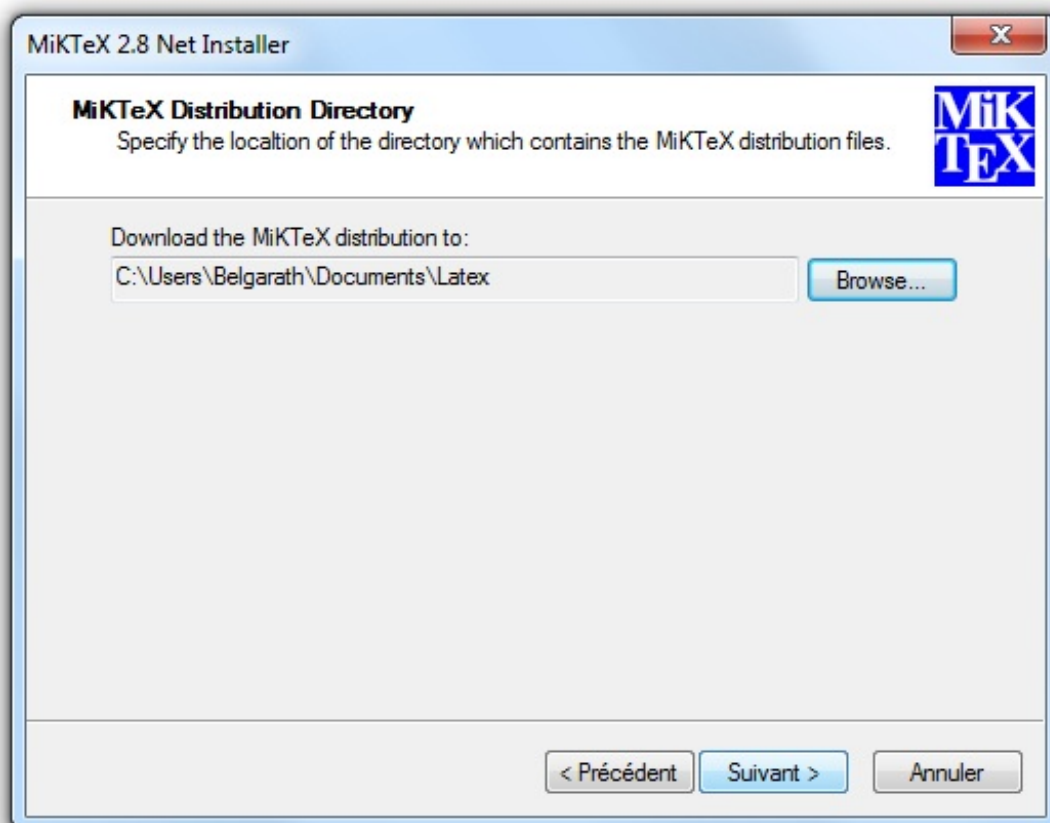
- **Lancez le gestionnaire d'installation** (normalement vous l'avez enregistré dans votre dossier `Latex`, si vous l'avez enregistré ailleurs, pas d'inquiétude, ça ne change rien à la procédure) en **double cliquant** dessus.
- **Cochez** la case « I accept the MiKTeX copying conditions » puis cliquez sur **Suivant** (figure suivante).



Case à cocher

- **Cochez** « Download MiKTeX » puis cliquez sur **Suivant**.

- **Cochez** « Complete MiKTeX » puis cliquez sur **Suivant**.
- **Choisissez** un serveur de téléchargement dans un pays plutôt proche du vôtre (par exemple, j'ai sélectionné un serveur français au hasard). Plus le serveur de téléchargement choisi est proche, plus vous chargerez rapidement. Cliquez sur **Suivant**.
- **ÉTAPE IMPORTANTE !** MiKTeX vous demande alors où vous voulez qu'il enregistre tout son attirail. C'est ici que le dossier `Latex` préalablement créé montre tout son intérêt. **Cliquez sur Browse** (Parcourir, en français), cherchez votre dossier `Latex` puis cliquez sur **OK**. Normalement, votre écran doit ressembler à la capture d'écran montrée en figure suivante.
- Cliquez sur **Suivant**, vous aurez droit à un petit écran récapitulatif comprenant le nom du serveur que vous avez choisi pour télécharger et le dossier de destination des fichiers de MiKTeX (ici le dossier `Latex`).
- Appuyez sur **Start** pour lancer le téléchargement.



Choix du

dossier de destination



MiKTeX doit télécharger plusieurs centaines de Mo de fichiers, et cela peut prendre pas mal de temps. Profitez donc de ces instants de répit pour tuer des zombies dans un jeu quelconque ou refaire le monde autour d'un café. La folle aventure de l'installation ne reprendra que dans quelques minutes. Soyez rassuré, le plus dur est derrière vous.



Il est possible qu'à un moment ou à un autre, l'installation de MiKTeX plante. Il vous faudra alors tout recommencer depuis le début (ça m'est arrivé lors de la rédaction de ce tutoriel). Ce n'est pas la grande joie d'appuyer cinquante fois sur **Suivant**, mais une bonne installation vous permettra de travailler sereinement plus tard.

- La longue période de chargement est terminée, l'assistant d'installation vous félicite pour votre patience. **Fermez** la fenêtre et **relancez le gestionnaire d'installation**. Une nouvelle fois, vous allez **cocher** la case « I accept the MiKTeX copying conditions » puis cliquer sur **Suivant**. Ensuite, vous allez choisir « **Install MiKTeX** » (enfin) et cliquer sur **Suivant**.
- Sur l'écran qui s'affiche alors, le gestionnaire d'installation vous demande si vous souhaitez installer la version basique

ou complète. Vu que nous avons téléchargé la version complète, nous cocherons donc « **Complete MiKTeX** » avant de cliquer sur **Suivant**.

- L'écran suivant vous demande si vous souhaitez installer MiKTeX pour vous seul ou sur tous les comptes présents sur votre machine. Faites selon vos souhaits puis cliquez sur **Suivant**.
- **ÉTAPE IMPORTANTE : MiKTeX** vous demande où sont rangés tous les fichiers que vous avez téléchargés plus tôt. **Vérifiez** bien que le bon dossier est sélectionné avant de cliquer sur **Suivant**. L'écran qui suit vous montre où **MiKTeX** va s'installer (les fichiers que vous avez téléchargés dans la première partie sont ceux dont il se sert pour l'installation). Vous pouvez changer la destination si vous voulez avant de cliquer sur **Suivant** et d'atterrir sur un écran vous demandant le format de papier par défaut que vous utiliserez (ici A4, vu que nos imprimantes impriment surtout en A4, mais vous pouvez le modifier si vous avez des besoins très spécifiques). Cet écran vous proposera aussi d'activer la demande de permission à chaque fois que le programme devra charger un package manquant. Laissez le réglage par défaut et appuyez sur **Suivant**.
- Un écran vous résume vos choix, et une fois que vous aurez cliqué sur **Start**, la procédure s'achèvera et le logiciel s'installera tranquillement. N'oubliez pas de cliquer une dernière fois sur **Suivant** pour voir encore un merveilleux écran de félicitations. :D



Cette étape est à la fois la plus cruciale et la plus ennuyeuse de l'apprentissage de LaTeX. Lorsque j'ai dû l'installer la première fois, me repérer entre toutes les distributions et les outils n'a pas été évident. C'est pour cela que je vous ai détaillé au maximum la procédure d'installation, pour que vous ne viviez pas le même cauchemar. Si vous voyez d'autres utilisateurs de LaTeX sur Windows utiliser d'autres logiciels, pas de panique, rappelez-vous que LaTeX étant libre, il existe des multitudes de logiciels pour l'utiliser.

Ghostscript et Adobe Reader, vos lecteurs de PostScript et de PDF

Il est temps d'installer un lecteur de PostScript ainsi qu'un lecteur de PDF sur votre ordinateur, de sorte que vous puissiez :

- manipuler les fichiers PS (diminutif de PostScript) ;
- transformer les fichiers .ps en .pdf ;
- lire les .ps ;
- lire les .pdf.

Ghostscript

Pour ce qui est des PS, téléchargez le logiciel **Ghostscript** sur telecharger.com : [Télécharger Ghostscript](#).

Lancez le gestionnaire d'installation, cliquez sur **Setup**, vérifiez que l'emplacement où il propose de s'installer vous plaît puis cliquez sur **Install**.

Adobe Reader

Les PDF se lisent la plupart du temps avec **Adobe Reader** (mais il existe beaucoup d'autres logiciels capables de lire des PDF). Ce logiciel est souvent présent par défaut sur les ordinateurs de bureau. Si ce n'est pas le cas, vous pourrez l'obtenir lui aussi sur telecharger.com : [Télécharger Adobe Reader](#).

Sa procédure d'installation est aussi simple que celle de **Ghostscript**, elle ne comporte aucun piège.

Bravo ! Vous venez d'installer Ghostscript et Adobe Reader.

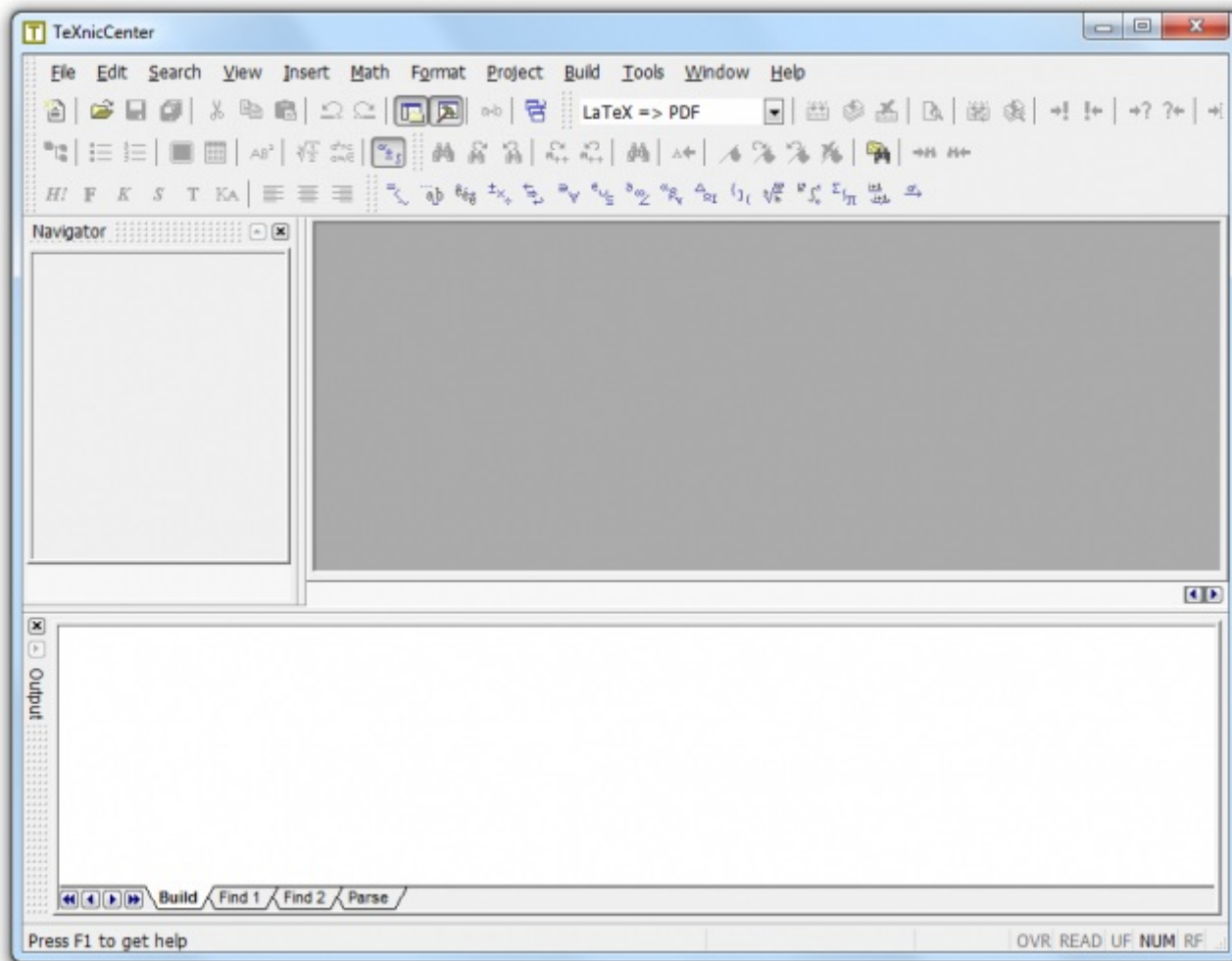
TeXnicCenter, l'éditeur LaTeX

TeXnicCenter sera votre éditeur LaTeX, vous avez d'ailleurs pu voir une capture d'écran de ce programme dans le paragraphe traitant des logiciels nécessaires. Pour l'installer, il vous faudra d'abord aller le chercher sur le site officiel : [Télécharger TeXnicCenter](#).

C'est le premier lien, nommé « **TeXnicCenter Installer** » qui nous intéresse, **cliquez** dessus, il vous permettra de lancer le

téléchargement du gestionnaire d'installation. Une fois que c'est fait, démarrez-le.

- À l'écran d'accueil, appuyez sur **Next**.
- Cochez « **I accept the agreement** » puis cliquez sur **Next**.
- **TeXnicCenter** vous propose de s'installer dans un dossier, changez-le s'il ne vous convient pas puis cliquez sur **Next**.
- Laissez l'installation sur **Typical** et cliquez sur **Next**.
- Il vous propose de créer un raccourci dans le menu « Démarrer », modifiez selon vos souhaits puis cliquez sur **Next**.
- Il vous propose de créer une icône sur le bureau, modifiez si besoin est avant de cliquer sur **Next**.
- S'affiche alors le résumé, vous pouvez cliquer sur **Install** après avoir vérifié les informations qu'il contient. Le petit écran de félicitations apparaît et avec lui, la possibilité de cliquer sur le bouton **Finish**. 😊



TeXnicCenter



L'installation est achevée, vous faites désormais cohabiter sur votre PC les trois composants nécessaires à votre apprentissage de LaTeX. Installation fastidieuse, mais importante pour la suite ! Nous ferons nos premiers tests dans le chapitre suivant et configurerons TeXnicCenter.

Installation sur Mac OS

Utilisateurs de Mac, vous êtes de grands chanceux.

Votre système d'exploitation remporte la palme de l'installation LaTeX la plus simple et agréable. Les détenteurs de Windows viennent de souffrir environ trois heures pour installer toutes les composantes de LaTeX séparées en trois alors que pour vous, tout est rassemblé dans un seul fichier !

Le code web fourni ci-dessous mène au site du projet **MacTeX**, au sein duquel se trouve un énorme fichier comportant tous les outils utiles à la réalisation d'un projet LaTeX. Le fichier à télécharger se trouve en plein milieu de la page (figure suivante), téléchargez-le sur votre Mac et lancez le gestionnaire d'installation avant de suivre la procédure d'une simplicité enfantine qui vous est proposée : [Télécharger MacTeX](#).

The MacTeX-2009 Distribution

[for Mac OS 10.3, 10.4, 10.5, and 10.6 — PPC and Intel]

[See FAQ below if updating to Snow Leopard]

The current distribution is MacTeX 2009.

To obtain the distribution, click the link below.

MacTeX.mpkg.zip

[approximately 1.3G - 12 March 2010]

Once you have the package, double click it to install.

If the previous link fails, use one of the specific sites on [this mirror page](#).

The two links below lead to other optional download packages:

MacTeXtras.zip [312M - 18 May 2010]

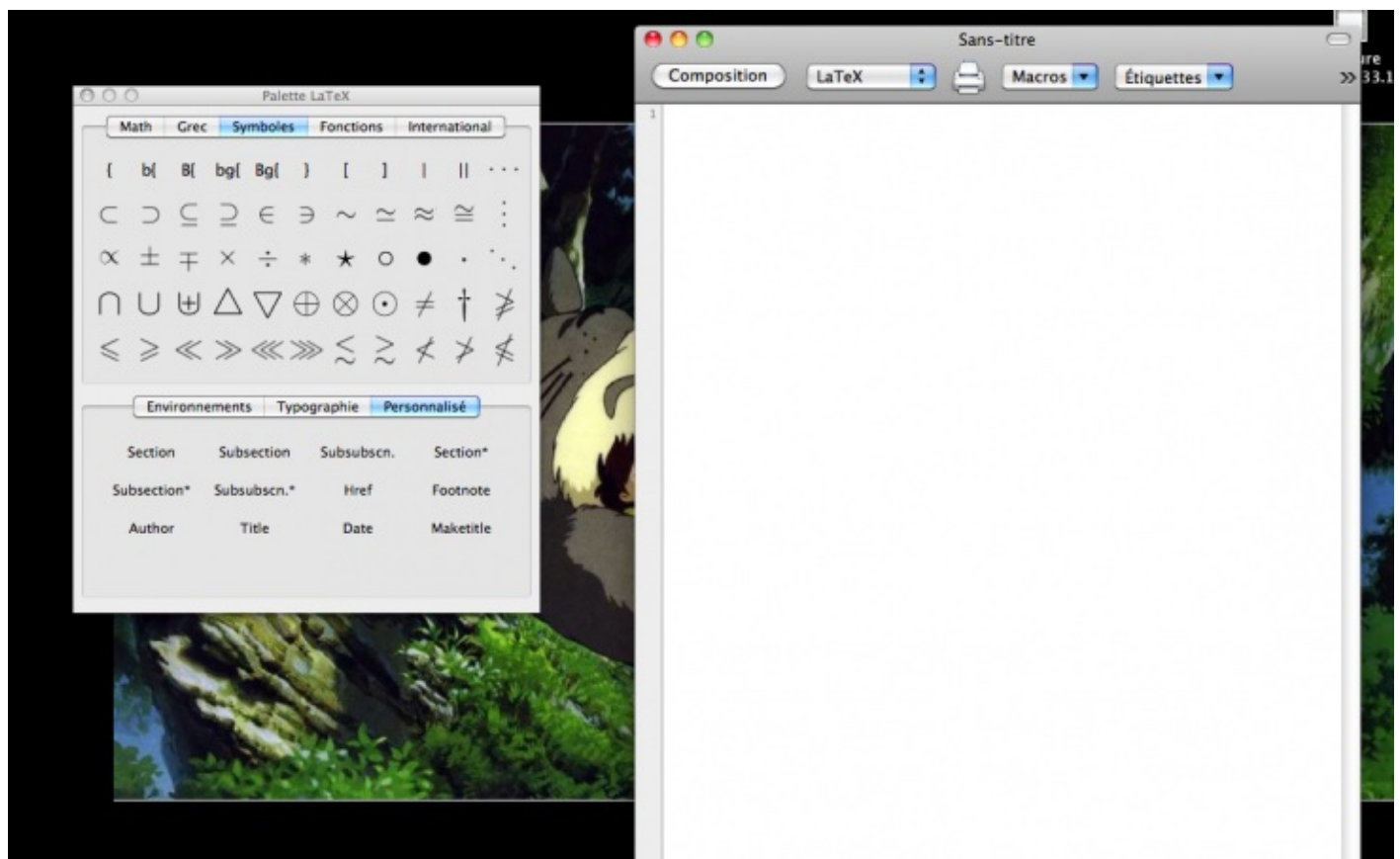
Smaller Packages [for users with slow download speed]

MacTeX

Installation de

Une fois cela fait, vous pourrez démarrer TeXShop, l'éditeur LaTeX pour Mac que nous utiliserons par la suite. Il devrait s'être installé dans votre dossier Applications, libre à vous de faire des raccourcis sur votre bureau.

Une capture d'écran de ce à quoi pourra ressembler votre TeXShop vous est montrée à la page suivante (figure suivante).



TeXShop



Avec MacTeX, vous avez installé une suite logicielle complète centrée autour de LaTeX. Elle vous permettra par exemple d'écrire des équations en LaTeX que vous pourrez inclure par copier-coller dans des documents Word. Outils pratiques, ils ne seront pas abordés ici mais ils méritent que vous les testiez lorsque vous vous sentirez à l'aise avec LaTeX.

Installation sur Linux

Dans le cadre de cet ouvrage, l'éditeur LaTeX choisi sera **Kile** (figure suivante) et la distribution LaTeX se nomme **TeX Live**. Les méthodes d'installation sont pour ce système-ci extrêmement variées et les retours sur la facilité d'installation sur Linux ont été très mitigés. La majeure partie des étudiants utilisateurs de LaTeX sous Linux ont témoigné avoir trouvé beaucoup plus évidente et adaptée l'installation via le gestionnaire de paquets (grâce auquel vous téléchargerez dans un premier temps **TeX Live** puis **Kile**). Nous installerons dans le même temps de quoi lire vos fichiers PDF ou PostScript produits.



Beaucoup, à leurs débuts sous Linux, ont eu du mal avec l'installation de LaTeX. N'hésitez pas à chercher dans vos groupes de travail ou votre entourage des personnes prêtes à vous aider. Il sera parfois plus évident pour vous de travailler dans les mêmes environnements que vos collaborateurs, et ce que vous lisez ici s'applique à n'importe quel éditeur LaTeX, seul l'emplacement des boutons changera si vous utilisez un éditeur LaTeX différent.

Comme mentionné plus haut, nous allons utiliser le gestionnaire de paquets (figure suivante).

Dans un premier temps, **ouvrez-le**. Recherchez et installez **tous les paquets** contenant les mots suivants :

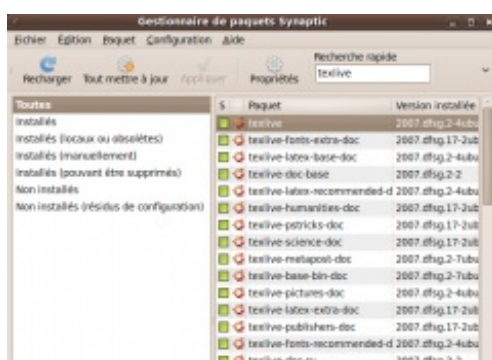
- texlive ;
- ec-fonts-mftraced (uniquement si vous êtes sous Mandriva) ;
- cm-super (pack de polices) ;
- xdvi (pas pour Ubuntu) ;
- gv (pour lire et manipuler les .ps) ;
- spdf ;
- kile (toujours en dernier).

Une fois ces multiples chargements effectués, vous pouvez vous reposer : votre installation fonctionne.

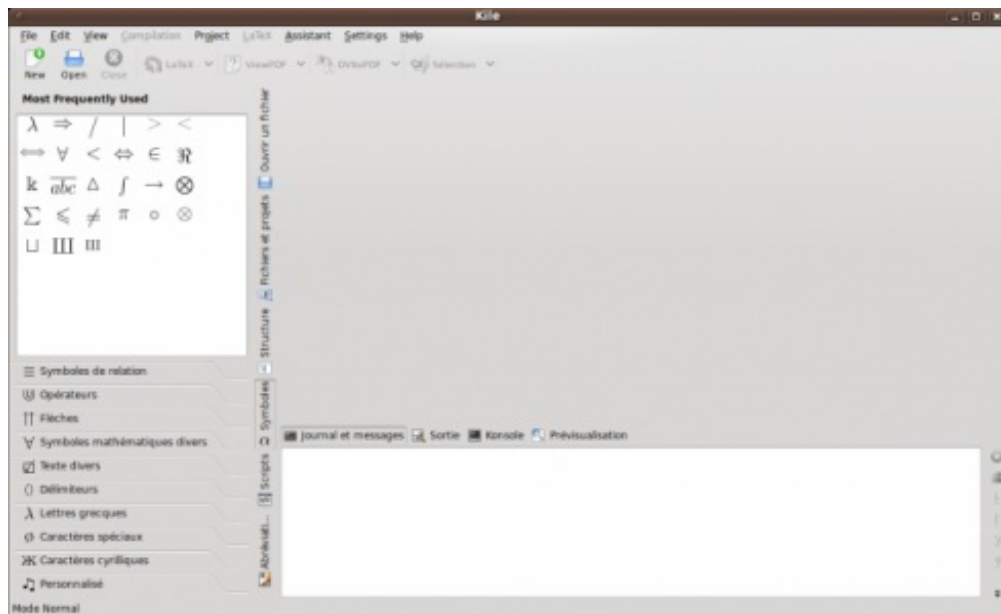
Voici une capture d'écran du gestionnaire de paquets et une seconde de **Kile**, l'éditeur LaTeX avec lequel nous allons nous exercer et apprendre à utiliser LaTeX.



Kile est un éditeur LaTeX facilitant l'écriture et la compilation de fichiers LaTeX. Il est néanmoins possible de manipuler via les lignes de commande des fichiers LaTeX tapés dans un éditeur de texte. Nous apprendrons à nous servir de celles-ci dans le prochain chapitre (l'apprentissage des deux méthodes vous permettra par la suite d'utiliser selon vos souhaits la console ou l'interface graphique).



Gestionnaire de paquets



Kile



En cas de problème, veuillez consulter les sites internet de TeX Live ainsi que de Kile afin d'obtenir les informations spécifiques à votre installation Linux. Ce sont des outils très utiles qui vous permettront d'adapter la procédure d'installation à votre distribution Linux si jamais elle est un tant soit peu exotique.

En résumé

- Il existe trois grandes familles de logiciels utilisés pour concevoir des documents en LaTeX : **les distributions, les éditeurs LaTeX et les lecteurs.**
- Un éditeur LaTeX n'est pas indispensable à l'apprentissage de LaTeX (mais en utiliser un simplifie énormément l'apprentissage grâce à la coloration du code et aux différents outils disponibles).
- Le format de prédilection que nous choisirons par la suite sera le **PDF** : il est lisible sur quasiment tous les ordinateurs récents. Nous nous souviendrons néanmoins de l'attrait que suscite le format de fichier **PostScript** pour certains laboratoires de recherche.
- **Kile, TeXShop et TeXnicCenter** sont actuellement très utilisés dans le monde de la recherche. Nous utiliserons ces éditeurs LaTeX sur des ordinateurs équipés respectivement de Linux, Mac OS et Windows.

Structurer son premier document

Durant les deux premiers chapitres de ce cours, vous avez découvert l'histoire de LaTeX ainsi que sa nature, et comment effectuer son installation. Ce nouveau chapitre sera pour vous l'occasion d'un premier contact avec les logiciels que vous venez d'installer.

Soyez sûr d'une chose : le plus dur est derrière vous.

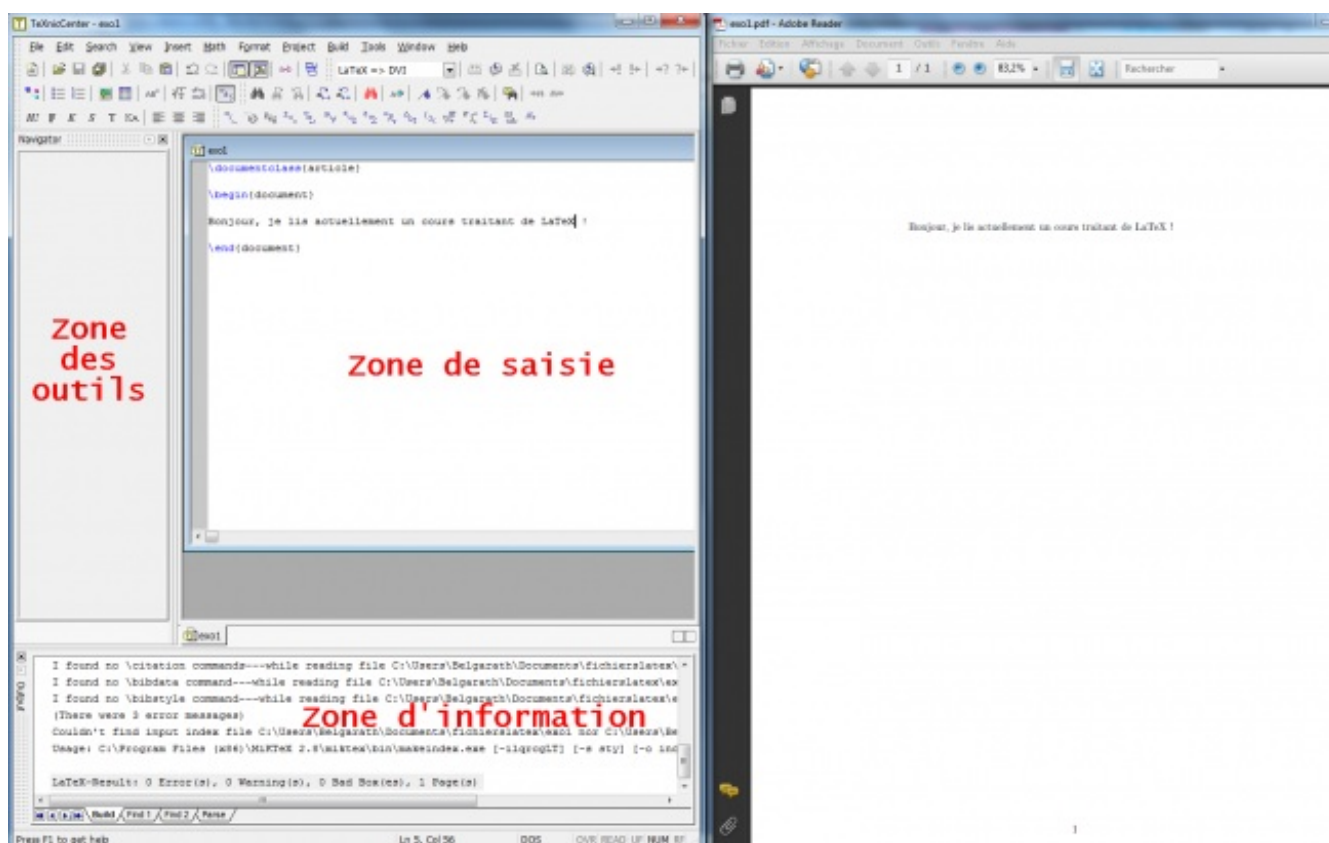
Ce chapitre vous permettra d'écrire votre premier document, de l'imprimer et de comprendre comment LaTeX transforme un code informatique ingrat en un document chatoyant.

C'est un chapitre extrêmement important, car il explique les mécanismes essentiels de LaTeX.

La compilation

Cette partie développe le concept de compilation et vous permettra d'appréhender le fonctionnement global de LaTeX.

Cette portion du cours s'appuie sur un constat : des lignes de code se transforment en une mise en page. Vous pouvez voir sur la capture d'écran ci-dessous (figure suivante) un document écrit via un éditeur LaTeX ainsi que son résultat une fois passé à la moulinette par votre distribution LaTeX.



Éditeur LaTeX et lecteur PDF



Que s'est-il passé ?

Souvenons-nous du chapitre précédent : nous y avons appris que les éditeurs LaTeX n'étaient pas indispensables, ils ne jouent qu'un rôle pratique. En revanche, nous avons fortement insisté sur le caractère incontournable de l'installation d'une distribution LaTeX. La distribution est la pièce centrale, elle permet de transformer un fichier de lignes de code (ces fichiers sont des `.tex`) en un fichier imprimable de type PS ou PDF.

Cette transformation s'appelle une **compilation**. Votre distribution LaTeX transforme votre fichier `.tex` en un fichier que vous pourrez par la suite utiliser pour vos travaux et publications.

Résumons tout cela.

1. Le code est écrit dans l'éditeur LaTeX.

2. La distribution LaTeX le compile.
3. Le fichier `.tex` de départ se voit transformé en un fichier `.ps`, `.pdf` ou `.dvi` (un format très peu utilisé).

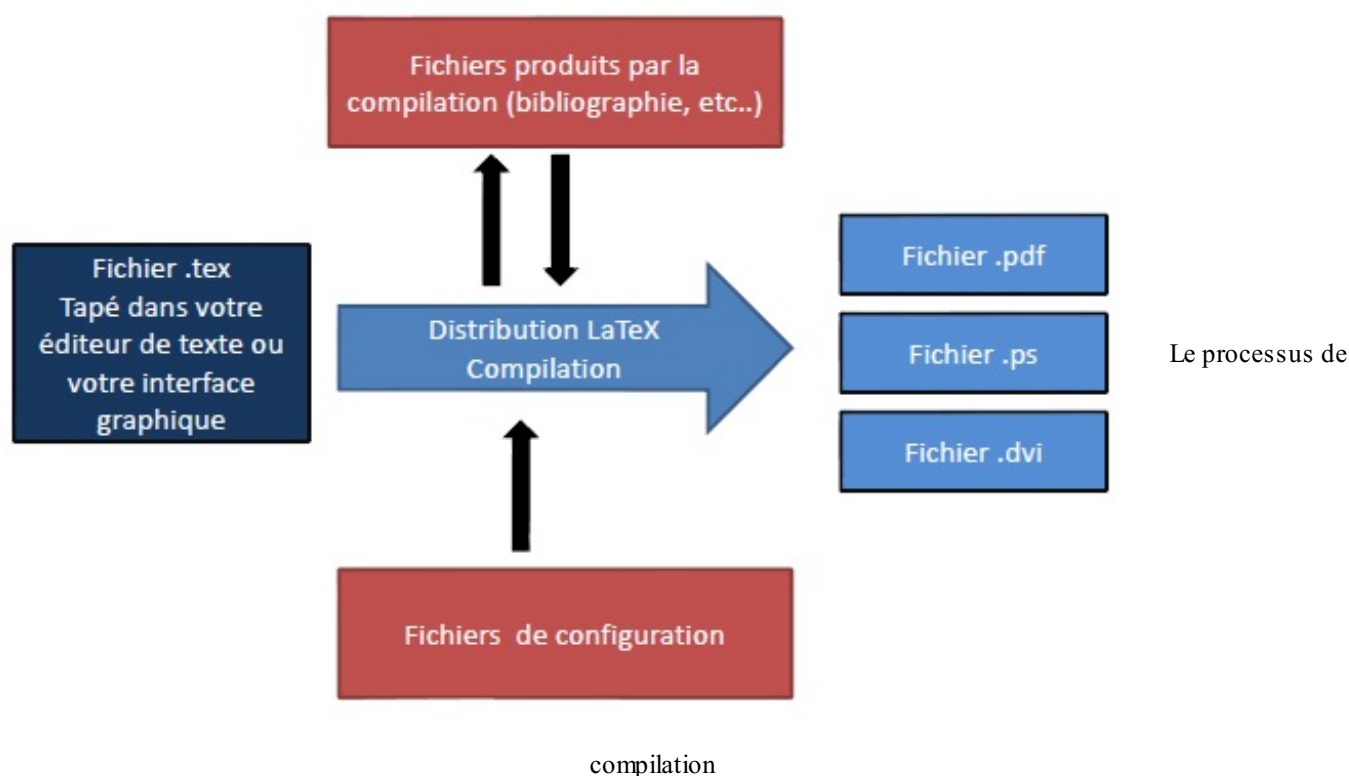
Le processus de création n'est pas plus compliqué que cela. Pour créer un document, il vous suffira d'écrire votre code au sein de l'éditeur LaTeX et votre distribution LaTeX se chargera du travail de transformation. Pour que le travail d'écriture du code et de compilation soit plus simple, les éditeurs LaTeX sont dotés d'une zone réservée aux raccourcis (pour les caractères spéciaux, par exemple), qui vous permettront d'insérer facilement dans vos écrits des morceaux de code dont vous vous servirez souvent (Nous verrons par la suite que certains accents doivent être traduits en code, et recopiés de façon très fréquente). Une seconde zone, dédiée aux informations liées à la compilation, vous permet de suivre les compilations et d'obtenir des informations sur d'éventuels bugs présents dans vos compositions. En somme, LaTeX communique avec vous pour vous aider à travailler.



Manipulerons-nous uniquement des fichiers `.tex`, `.dvi`, `.ps` et `.pdf` avec LaTeX ?

Pour rédiger un livre, il faut une bibliographie, un index et tout un tas d'éléments que LaTeX stocke dans des fichiers aux extensions différentes. Pendant la compilation, votre fichier `.tex` de départ créera donc une multitude de fichiers correspondant à ces besoins. Ne prenez pas peur si votre dossier ne contenant au départ qu'un `.tex` se voit tout à coup envahi par de multiples fichiers aux extensions étranges.

Nous allons résumer toutes ces informations via un schéma (figure suivante). Je vous conseille vivement de bien le lire, il est souvent pratique de l'avoir à l'esprit.



Si vous avez compris ce schéma, sachez que LaTeX n'a pas un fonctionnement plus compliqué que cela. Il ne nous reste plus qu'à prendre en main les logiciels et à apprendre comment mettre en forme nos textes en utilisant les commandes et le formalisme de LaTeX.

Rédaction d'un document simple

Dans cette partie, je vais vous aider et vous guider dans l'écriture de votre premier document. Il sera très simple et ne comportera qu'une phrase : « Bonjour, je lis actuellement un cours traitant de LaTeX ! »

Rédaction du fichier `.tex`

Dans un premier temps, il vous faut ouvrir votre **éditeur LaTeX** (Kile, TeXShop, TeXnicCenter ou autre). Il est possible que celui-ci vous pose quelques questions simples (pour connaître l'emplacement de votre lecteur de PDF, par exemple), répondez-y.



Si TeXnicCenter vous demande le chemin d'installation de MiKTeX, vous devrez fournir un chemin du type
 C:\Program Files (x86)\MiKTeX 2.8\miktex\bin.

Votre éditeur est à présent ouvert et vous vous trouvez face à une fenêtre scindée en trois parties : une partie console dans laquelle apparaissent les messages d'information (ainsi que les messages d'erreur), une zone réservée aux barres d'outils, et un espace réservé à la rédaction de votre fichier.

Grâce aux raccourcis disponibles dans la barre de menus, créez un nouveau document.
 Cliquez à l'intérieur de la zone réservée à l'édition et recopiez-y le paragraphe ci-dessous.

Code : TeX

```
\documentclass{article}

\begin{document}
Bonjour, je lis actuellement un cours traitant de LaTeX !
\end{document}
```

Enregistrez ce fichier dans le dossier vide de votre choix en utilisant la barre de raccourcis présente en haut de votre éditeur.
 Nous l'appellerons `bonjour.tex`.

Il est interdit de mettre des espaces ou des accents dans le nom de vos fichiers. Votre compilateur n'apprécierait pas l'originalité et aurait tendance à vous renvoyer des erreurs.

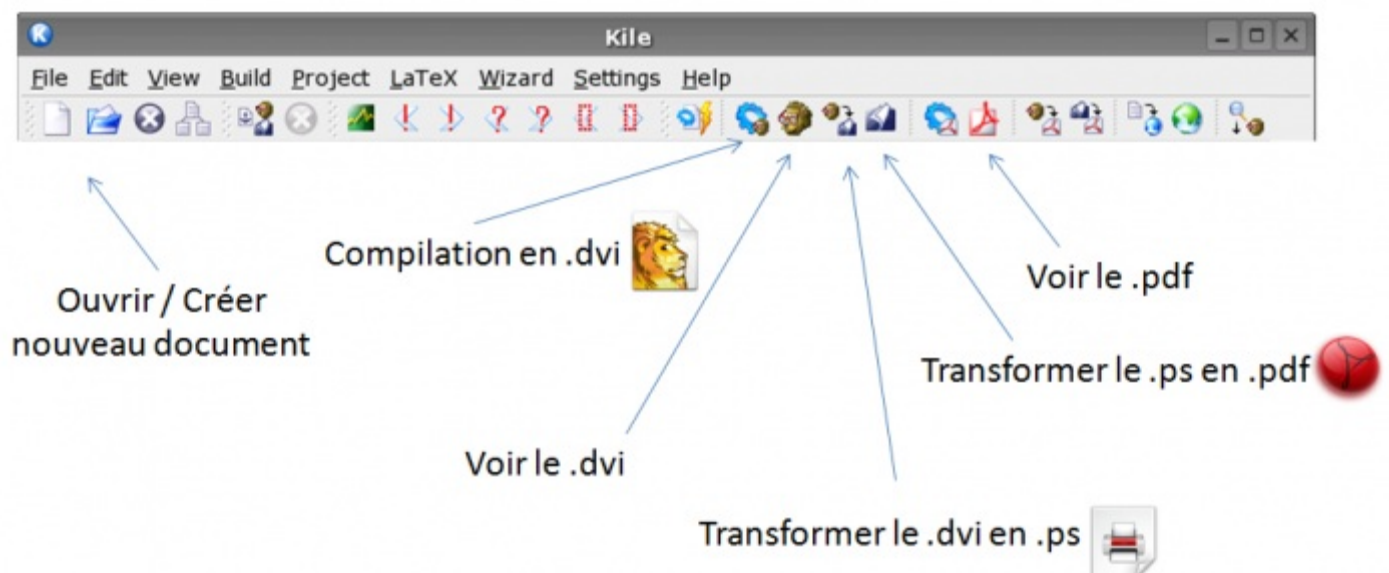


Autre subtilité : le compilateur compile du texte brut et non pas enrichi comme peut en fournir Word (un texte enrichi est un texte auquel une mise en page a été ajoutée). Si jamais l'envie vous prend de copier-coller un texte depuis Word, faites d'abord un copier-coller dans le Bloc-notes ou un logiciel du même genre. Vous éviterez ainsi beaucoup de problèmes.

Compilation

Nous allons maintenant compiler votre fichier `.tex` pour le transformer en un autre format (PostScript, PDF ou DVI). Ici, trois cas de figure se présentent selon que vous soyez sous Windows, Mac ou Linux.

Compiler avec Kile (Linux)



Compilation avec Kile (pour les anciennes versions)

Vous pouvez voir sur la capture d'écran ci-dessus (figure suivante) la barre d'outils à votre disposition dans Kile. Pour vous, tout commence par la création d'un fichier DVI que vous transformerez ensuite en fichier PostScript, puis en PDF selon vos besoins.

Pas de manipulation compliquée ici, votre éditeur LaTeX se charge de tout et les fichiers produits par la compilation se trouvent dans le dossier où vous avez enregistré votre fichier `.tex`.



J'ai choisi d'insérer ici des captures d'écran de l'ancienne version de Kile ; ce n'est pas une erreur de ma part (Il est encore possible de travailler sur des ordinateurs utilisant l'ancienne version, même s'ils sont de plus en plus rares), mais un choix pédagogique. La nouvelle version comporte simplement quelques menus déroulants extrêmement intuitifs (Ils sont du même type que ceux de TeXShop.).

Compiler en ligne de commande (Linux)

Nous avons vu dans le chapitre précédent que l'éditeur LaTeX n'était pas indispensable. Il vous est possible de créer votre fichier `.tex` dans un éditeur de texte (Emacs, Vim...) avant de le compiler en ligne de commande.

La procédure est assez simple. Une fois votre fichier enregistré dans un dossier, ouvrez la console et placez-vous dans ce dossier avant d'utiliser l'une ou l'autre des commandes ci-après.

Dans un premier temps, vous pouvez compiler votre fichier `bonjour.tex` en un fichier `bonjour.dvi`. Le fichier créé apparaît alors dans le dossier où vous avez enregistré votre fichier d'origine en `.tex`.

Code : Console

```
latex bonjour.tex
```

Vous pouvez ensuite lire votre fichier `bonjour.dvi` grâce à la commande `xdvi` et l'imprimer grâce à la commande `dvips`.

Code : Console

```
xdvi bonjour.dvi  
dvips bonjour.dvi
```

Vous pourrez par la suite choisir de transformer votre fichier `bonjour.dvi` en un fichier `.ps` grâce à la commande `dvips` (en y ajoutant `-o` comme vous pourrez le voir plus bas dans l'exemple). La commande `ps2pdf` vous permettra de transformer votre fichier PostScript en un fichier PDF.

Il vous est tout à fait possible de compiler directement votre fichier `bonjour.tex` en un fichier `bonjour.pdf` grâce à la commande `pdflatex`. La lecture de PDF se fait grâce à la commande `xpdf`.

Résumons ces commandes par quelques exemples pratiques. Testez ces commandes, elles vous serviront tout au long du cours.

Code : Console

```
dvips bonjour.dvi -o  
ps2pdf bonjour.ps  
pdflatex bonjour.tex  
xpdf bonjour.pdf
```

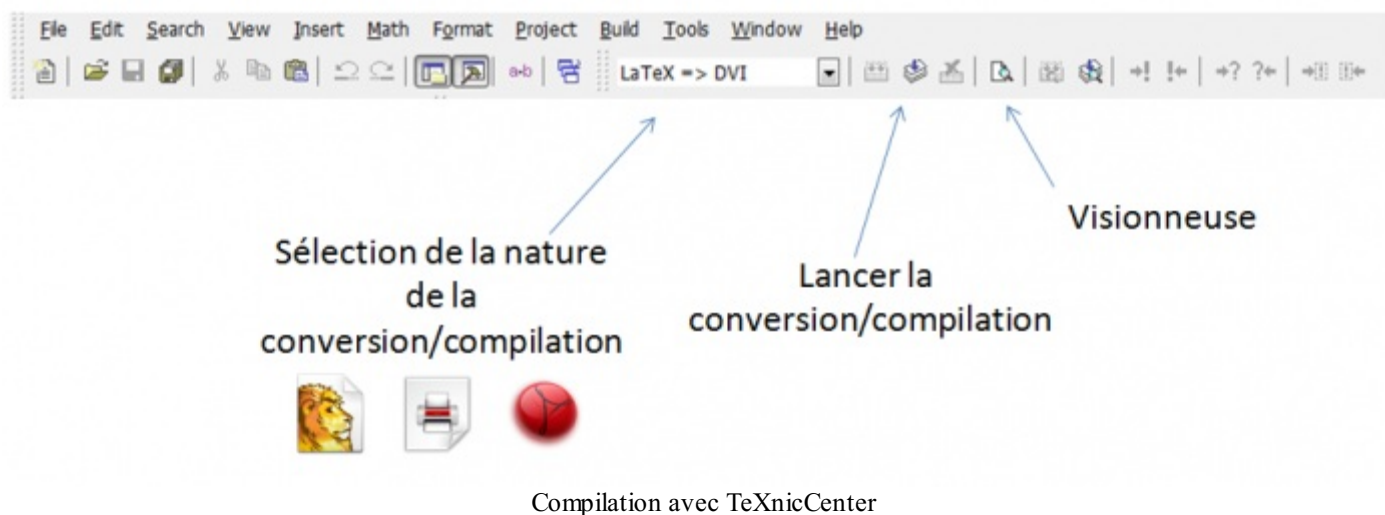
Compiler avec TeXShop (Mac)

Une fois de plus, c'est vous, détenteurs de Mac, qui possédez l'interface la plus minimaliste et la plus simple : il vous suffit de cliquer sur « Composition » tout en ayant sélectionné « LaTeX » dans le menu déroulant à droite du bouton.

Votre éditeur LaTeX se charge de tout et les conversions sont faites de façon automatique.

Sans plus attendre, vous pouvez admirer le résultat de votre compilation dans la fenêtre qui vient de s'ouvrir, ou en allant fouiller dans le dossier où est enregistré votre fichier `.tex`.

Compiler avec TeXnicCenter (Windows)



Dans votre barre de menu, vous constatez la présence d'un menu déroulant (figure suivante). Ce menu vous permet de choisir entre la compilation d'un fichier `.tex` en un `.dvi` et la conversion de ce fichier `.dvi` en un autre format de fichier.

Le bouton à sa droite sert à lancer les compilations / conversions et le bouton encore plus à droite vous donne les aperçus de vos fichiers.

Les fichiers créés apparaissent dans le dossier où vous avez préalablement enregistré votre `.tex`.

Les résultats

Ici, deux possibilités s'offrent à vous : soit votre compilation a réussi, vous venez de découvrir comment écrire un document en LaTeX (libre à vous de l'imprimer) et vos logiciels fonctionnent, soit votre éditeur LaTeX vous a renvoyé une erreur... je suis alors au regret de vous annoncer que vous vous êtes trompé lors de l'installation et que vous devez la recommencer.



Avant de vous résigner si jamais la compilation n'a pas fonctionné, essayez de copier une nouvelle fois le code source. Il serait dommage de recommencer l'installation à cause d'un « \ » oublié.

Vous venez de créer votre premier document, félicitations !

Types de documents et caractères spéciaux

Avant toute chose, sachez que la compilation de documents LaTeX fonctionne toujours de la même manière, vous n'avez plus rien à apprendre de ce côté-là.

Nous allons revenir ici sur le code que nous avons entré dans l'éditeur LaTeX et parler des caractères spéciaux.

Les caractères spéciaux

Comme tout langage de programmation, LaTeX utilise certains caractères pour son usage propre. Dans notre premier document, nous avons utilisé le « \ » pour lui envoyer des ordres. En somme, le « \ » n'est pas simplement recopié par LaTeX, mais interprété.

Ce n'est pas le seul caractère à avoir cette spécificité : ils sont dix et figurent dans le tableau ci-dessous.

Tentez de mettre l'un de ces caractères dans votre texte, il en résultera des erreurs de compilation. Pour signaler à LaTeX que nous ne lui envoyons pas une commande ou que nous ne communiquons pas avec lui en utilisant ces caractères, il suffit de faire précéder les signes d'un backslash (le fameux « \ »). Pour le backslash lui-même, nous utiliserons une commande.

Table des caractères réservés	
\$	\\$
&	\&
%	\%
#	\#
_	_
{	\{
}	\}
	\
^	\^
\	\textbackslash{}

Testons ce tableau ! Entrez le code suivant dans votre éditeur LaTeX, puis enregistrez le `.tex` avant de le compiler et de le convertir en PDF.

Code : TeX

```
\documentclass{article}

\begin{document}
Bonjour, je lis actuellement un cours traitant de LaTeX !
\$ \& \% \# \_ \{ \} \ \^ \textbackslash{}

\end{document}
```

Vous devez obtenir la phrase suivante : « Bonjour, je lis actuellement un cours traitant de LaTeX ! \$ & % # _ { } ^ \ ».

Ce tableau est le premier, mais il y en aura d'autres, pour les accents par exemple. L'ensemble des tableaux de caractères spéciaux est rassemblé au sein d'une annexe volumineuse disponible à la fin du tutoriel (Elle comporte surtout des caractères destinés aux mathématiques).

Types de documents et portions de texte

Nous allons ici décortiquer les trois lignes de code dont nous n'avons pas encore parlé.

Les portions de texte

Dans nos deux premiers exercices, nous avons entouré notre texte d'un couple de commandes.

Code : TeX

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

Ces deux commandes délimitent le début et la fin de ce que nous appellerons un **environnement**. Ici, cela revient à dire que nous commencerons **toujours** un document par `\begin{document}` et que nous écrirons du texte avant de terminer le document par `\end{document}`.

De façon plus large, nous pourrions vouloir insérer un poème ou une recette dans un texte. Il nous suffirait alors d'écrire `\begin{recette}` puis la recette avant de fermer avec `\end{recette}`. Ce qui pourrait nous donner le code ci-dessous (ce n'est qu'un exemple fantaisiste).

Code : TeX

```
\begin{document}
\begin{recette}
Ma recette, bla, bla bla.
\end{recette}
\begin{texte}
Mon joli texte, bla, bla bla.
\end{texte}
\end{document}
```

La commande `\begin` ouvre un environnement tandis que `\end` le ferme. Le type d'environnement se voit entouré d'accolades et les options qui lui sont rattachées sont entre crochets (nous en utiliserons tout au long de ce tutoriel).

En résumé, délimiter un environnement revient à écrire un code du même type que celui ci-dessous.

Code : TeX

```
\begin{type}[options]
Bla,bla
\end{type}
```

Nous apprendrons les noms et l'utilité des différents environnements au fil du cours.

Les types de documents

La première ligne de nos deux exercices n'a pas encore été analysée : elle contient pourtant une information capitale. Tout d'abord, réaffichons cette ligne, je vous expliquerai tout ensuite.

Code : TeX

```
\documentclass{article}
```

Une traduction approximative de la commande nous donne quelque chose qui ressemble à ceci : ce document est un article, probablement scientifique. Ici aussi, nous écrivons le type entre les accolades.

Cette commande sert en fait à expliquer à LaTeX que la publication que nous sommes en train de produire est un article. Ceci dans le but que LaTeX mette en page l'intégralité du contenu afin qu'il respecte les normes de typographie et de mise en page d'un article parfaitement présenté.



Pour écrire un roman, je fais comment ?

Tout simplement, vous remplacez le type de document par `book`. Il existe plusieurs types de documents, je vous cite ici les principaux dans un tableau.

Type de document	
article	Article (sous-entendu scientifique)
book	Livre
letter	Lettre
report	Rapport (stage, thèse...)



Peut-on là aussi spécifier des options ?

Absolument, nous verrons dans la suite du cours certaines options dédiées à certains types de document. La formulation des options se fait selon la syntaxe suivante :

Code : TeX

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

En résumé

- Pour transformer un fichier LaTeX (`.tex`) en PDF (`.pdf`), on passe par une étape dite de **compilation**.
- La compilation d'un document se déclenche grâce à des raccourcis présents dans l'éditeur LaTeX. Il est aussi possible d'utiliser la ligne de commande. Il suffit de taper : `pdflatex fichier.tex`.
- Certains caractères spéciaux (comme `$`, `#`, `&`, `%`...) doivent être précédés d'un *backslash* (`\textbackslash`) pour être insérés dans un texte (L'oubli d'un *backslash* devant ces caractères spéciaux peut entraîner de multiples erreurs et bugs lors d'une compilation.).
- Un document LaTeX peut être de type `article`, `book`, `letter` ou `report` selon le type de document que vous souhaitez écrire.
- Nous écrirons le contenu de notre document à l'intérieur de l'environnement `document`, c'est-à-dire entre les commandes `\begin{document}` et `\end{document}`.

Les packages

Votre apprentissage des mécanismes élémentaires de LaTeX est terminé. Vous savez à présent ouvrir et rédiger des fichiers source en `.tex` et les compiler de façon à obtenir des documents aux formats PDF, PostScript et DVI.

Nous avons également pu constater que pour expliquer à LaTeX ce que nous attendons de lui, nous utilisons des commandes et des lignes de code. Ces différentes commandes et leurs applications seront étudiées dans les parties 2 et 3.

Ce chapitre est le dernier chapitre théorique. Nous allons ici répondre à une question importante : si LaTeX ne sait pas faire quelque chose (mettre du texte en couleur, par exemple), comment changer cela ?

Nous allons répondre à cette question en introduisant la notion de « package », notion centrale et omniprésente dans l'univers LaTeX.

La philosophie des packages

Les habitués de l'informatique connaissent assez bien la signification des mots « **extension** », « **package** » et « **plugin** ». Ces éléments sont présents dans de nombreux programmes et dans cette partie, nous allons tenter de développer le concept de « **package** » pour ceux à qui il n'est pas familier.

Imaginons votre installation LaTeX comme une cuisine équipée. Vous pouvez cuire des aliments, poser des objets sur le plan de travail, etc. En bref, cette pièce peut vous rendre certains **services** tout comme LaTeX nativement vous permet d'écrire du texte simple (c'est ce que nous avons pu voir dans le chapitre précédent).

Vous vous préparez un plat et avez besoin d'un couteau pour le cuisiner, or ce couteau ne fait pas partie des **outils** fournis par la personne qui a conçu votre cuisine. C'est un problème. Il est nécessaire pour vous d'aller chercher quelque part cet outil, afin de l'apporter dans votre cuisine pour vous permettre d'atteindre votre objectif : découper ce que vous vouliez découper.

Dans le cas de LaTeX, des outils ont été créés dans le but de pallier certains manques et d'implanter de nouvelles fonctions. Grâce à ces derniers, l'univers LaTeX permet de réaliser chaque jour de nouvelles choses. Ces outils sont appelés « packages », et tout un chacun peut en créer ainsi que les diffuser.

Si jamais un package vient à manquer, vous irez le chercher sur l'internet et l'utiliserez ensuite (gardez la métaphore du couteau en tête, **quand un outil manque, il suffit d'aller le chercher**).



Dans le cas de logiciels de traitement de texte, cette logique peut vous apparaître comme une faiblesse. En effet, pourquoi ce logiciel n'embarque-t-il pas d'emblée toutes les fonctions qu'on attend de lui ? Tout simplement pour conserver la possibilité d'évoluer facilement via sa communauté.

En tant qu'utilisateur de LaTeX, vous serez fréquemment amené à utiliser des packages. En résumé, deux possibilités s'offrent à vous lorsque vous souhaitez en manipuler un :

- le **package est déjà présent** dans votre installation LaTeX et il ne vous reste plus qu'à vous en servir (c'est comme si votre couteau était déjà dans un des tiroirs de votre cuisine) ;
- le **package est absent** de votre installation LaTeX et vous devrez aller le chercher et l'installer (comme si vous alliez au magasin chercher le couteau pour le mettre dans un tiroir).

Ces tâches sont très simples et nous allons apprendre dans la suite de ce chapitre à installer et utiliser les packages de LaTeX.

Comment s'en servir ?

Même si nous allons ici apprendre à installer des packages, ce n'est pas l'objectif principal de ce chapitre. Des millions d'utilisateurs avant vous ont conçu des documents à l'aide de LaTeX et rendu certains packages incontournables. C'est la raison pour laquelle ils sont aujourd'hui souvent installés par défaut avec votre distribution LaTeX (de la même manière que vous ne vous installeriez pas dans une maison sans meubles, les utilisateurs de LaTeX préfèrent installer dès le début les éléments essentiels leur permettant de travailler confortablement).

Cette partie va vous apprendre à manipuler les packages.

Pour utiliser un package avec LaTeX, nous procéderons en deux étapes. Dans un premier temps, nous communiquerons à LaTeX, au moyen d'une commande, que nous travaillerons avec tel ou tel package et dans un second temps, nous pourrons les utiliser dans nos publications.

La commande `\usepackage[option]{type}` permet d'appeler un package. Elle se place juste après la ligne dédiée à la commande `\documentclass`.



La zone entre `\documentclass` et `\begin{document}` est le préambule du code. C'est la zone où nous placerons les appels de packages et diverses commandes. Il ne faut cependant pas confondre le préambule du code avec le préambule d'un livre, ce n'est pas du tout la même chose.

Regardez l'exemple ci-dessous que nous allons commenter ensemble.

Code : TeX

```
\documentclass{report}

\usepackage[latin1]{inputenc} % un package
\usepackage[T1]{fontenc}      % un second package
\usepackage[français]{babel}  % un troisième package
\begin{document}

J'écris mon 3e document avec \log \LaTeX{} \fg.
\LaTeX{} est un langage créé par Leslie \bsc{Lamport}.

\end{document}
```



Ici, nous constatons que les annotations dans le code se font avec la syntaxe suivante : `% Mon commentaire`. Dans la suite du cours je me servirai des commentaires pour annoter des morceaux de code et vous aider à vous repérer. N'hésitez pas à utiliser les commentaires dans votre travail pour mieux vous y retrouver.

Trois packages ont été utilisés dans l'exemple ci-dessus :

- **babel** est utilisé pour spécifier à LaTeX que vous écrivez en français ;
- **fontenc** et **inputenc** vous permettent d'utiliser tous les caractères de votre clavier.



Petit test pour nos amis linuxiens. Créez avec votre bloc-notes un fichier contenant la phrase « Je maîtrise déjà les bases du LaTeX » avant de l'enregistrer sous le nom `codage.txt` et de taper dans votre console la ligne suivante : `file codage.txt`. Si la console inscrit ISO-8859, tout va bien. Si elle inscrit UTF-8 (UTF-8 et ISO-8859 désignent des processus d'encodage de caractères, c'est-à-dire la manière dont votre machine va coder les lettres. Bien que nous ne puissions pas voir directement la différence, se tromper entre UTF-8 et Latin-1 peut tout faire planter), vous devrez taper `\usepackage[utf8]{inputenc}` à la place de `\usepackage[latin1]{inputenc}` dans vos prochains documents.

Utilisateur sous Mac OS : il est important que vous enregistriez vos documents en Latin-1. TeXShop vous proposera dans le menu Préférences → Documents → Encodage de choisir l'encodage par défaut. Sélectionnez celui-ci : Occidental (ISO Latin 1).



`\up{}`, `\og` et `\fg` sont-elles de nouvelles commandes ?

Effectivement, ces commandes sont nouvelles. Le package **babel** permet l'utilisation de commandes propres à la langue choisie (Nous n'utiliserons ici que les commandes de babel liées au français.). Entre autres, les guillemets français et les petites capitales en ce qui concerne le français. Ce package permet donc à LaTeX d'exécuter de nouvelles tâches et d'être plus efficace pour écrire en langue française.

Ci-dessous, un tableau vous montrant les commandes introduites par le package **babel**. N'hésitez pas à les essayer et à vous faire la main avec.

Package babel

«	<code>\og</code>
»	<code>\fg</code>
Exposant	<code>\up{ }</code>
Petites Majuscules	<code>\bsc{Lamport}</code>
1 ^{er}	<code>1\ier</code>
2 ^e	<code>2\ieme</code>
1 ^o	<code>\primo</code>
2 ^o	<code>\secundo</code>
3 ^o	<code>\tertio</code>
4 ^o	<code>\quarto</code>
8 ^o (ou n'importe quel autre chiffre)	<code>\FrenchEnumerate{8}</code>
n ^o	<code>\no</code>
N ^o	<code>\No</code>
n ^{os}	<code>\nos</code>
N ^{os}	<code>\Nos</code>
40° (le caractère degré)	<code>40\degres</code>

Des centaines de packages existent, chacun rendant LaTeX plus performant. Ils vous permettent d'ajouter, à la carte, des fonctions à votre distribution. Souvent, lorsque vous avez envie de faire quelque chose d'original avec LaTeX, quelqu'un a déjà fabriqué un package adapté (Une recherche sur votre moteur de recherche favori suffit souvent à trouver la perle rare.) et il ne vous reste plus qu'à l'installer puis à l'utiliser.

Comment installer un package ?

L'installation par défaut de LaTeX sur votre machine inclut déjà de nombreux packages. Il est cependant possible que vous ayez un jour besoin d'installer un package absent sur votre machine. Cette partie est là pour vous expliquer comment faire.

Notons que si vous avez installé LaTeX comme expliqué dans le chapitre 3, **il est peu probable qu'un package utilisé dans ce cours ne soit pas présent sur votre machine.**



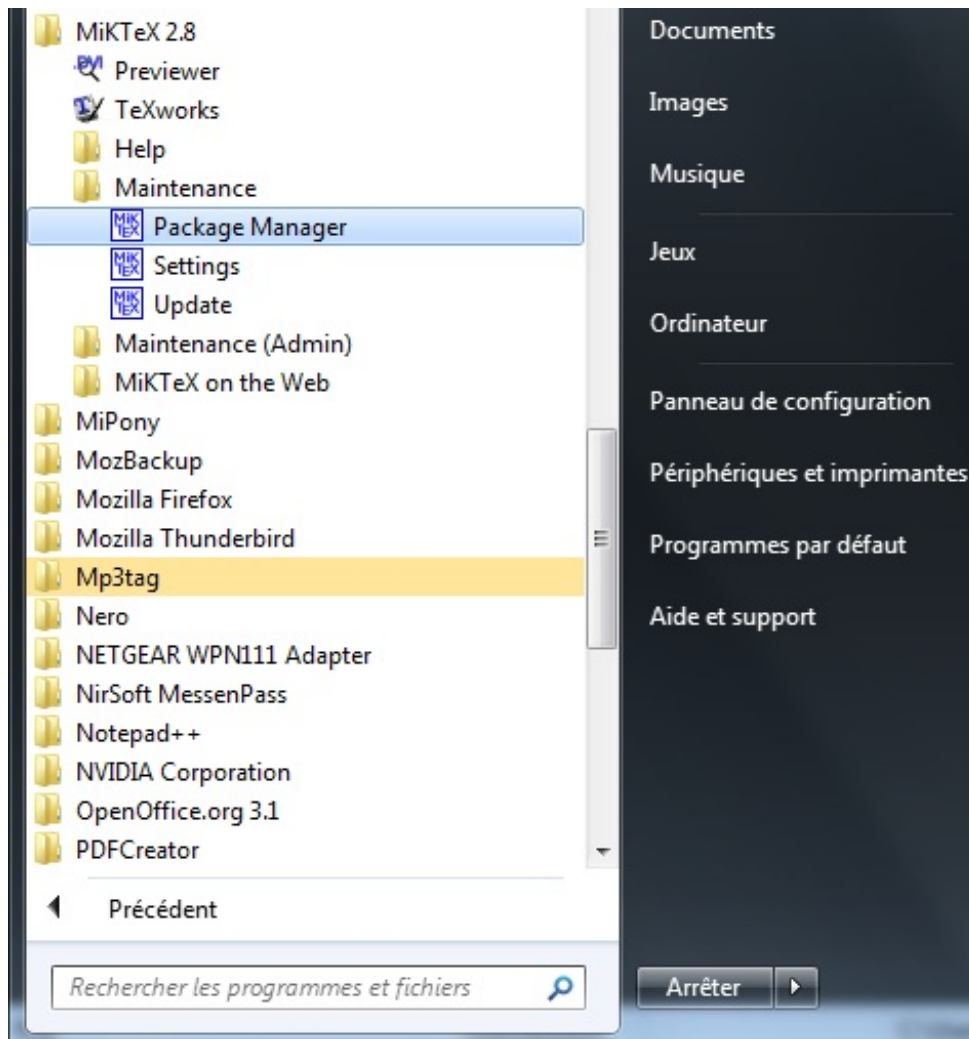
Cette sous-partie n'étant utile que si vous tentez d'installer un package non présent dans votre distribution, vous pouvez ne pas la lire et passer au chapitre suivant sans crainte.

MiKTeX et son installation à la volée

Les utilisateurs de MiKTeX (Windows) ont de la chance en ce qui concerne les packages, car MiKTeX installe tout seul comme un grand les packages appelés lors de la compilation qui ne sont pas présents sur leur machine.

Si vous souhaitez savoir si un package se trouve dans votre installation MiKTeX, rien de plus simple, il vous suffit d'aller dans votre menu Démarrer et d'aller chercher le **Package Manager** (figure suivante).

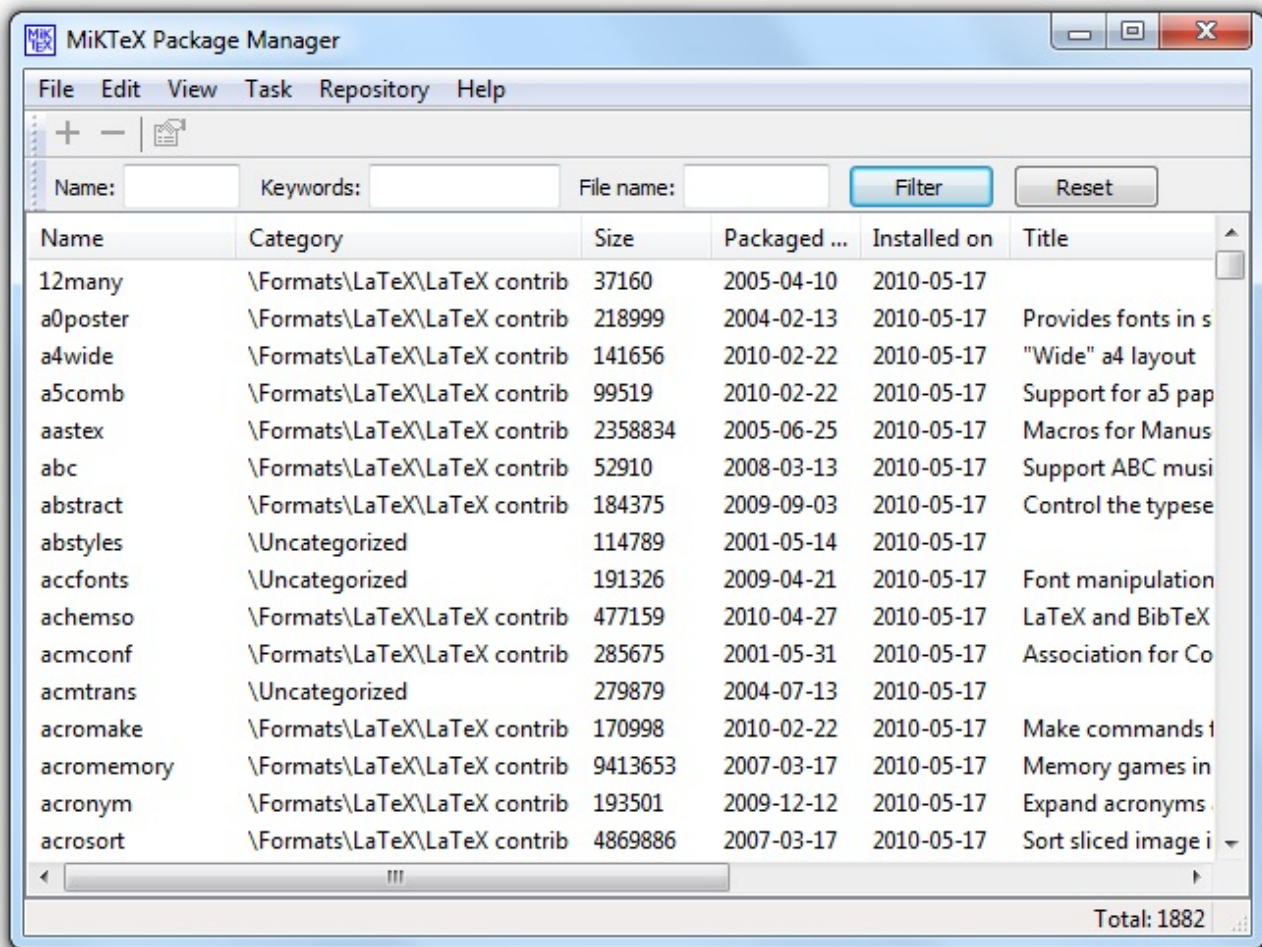
Chemin du Package Manager : MiKTeX → Maintenance → Package Manager.



Trouver le Package Manager dans le

menu

Le Package Manager vous permettra de voir la liste des packages installés dans votre distribution (figure suivante).



Package Manager

Autres distributions

Il existe plusieurs méthodes d'installation de package avec LaTeX. J'ai sélectionné pour vous les deux plus faciles à mon sens. Elles devraient vous permettre d'utiliser la quasi-totalité des packages.

Les deux méthodes développées ici diffèrent légèrement, suivant que votre package est un fichier `.ins` ou `.sty`.

Dans de rares cas, les packages sont fournis sous d'autres extensions, mais ils sont alors accompagnés d'un fichier README vous guidant lors de leur installation.

Les packages en `.sty`, méthode simple

Si votre package est de la forme `nom_de_package.sty`, rien de plus simple pour l'utiliser : il suffit de le copier dans le dossier contenant votre source `.tex`. Lorsque votre distribution compilera le fichier `.tex`, elle recherchera dans ce dossier les fichiers `.sty` des packages manquants, et le tour sera joué.

Résumons, la commande `\usepackage{nom_de_package}` demande à LaTeX d'utiliser un package installé ou, s'il ne l'est pas, d'aller chercher le fichier `nom_de_package.sty` dans le dossier de travail.

Simple, n'est-ce pas ?

Les packages en `.ins`, méthode en deux temps

Les packages contenus dans un fichier `.ins` doivent être traités en deux étapes. Premièrement, mettez votre fichier `nom_de_package.ins` dans un répertoire et compilez-le : il enfantera (sans douleur) un fichier `nom_de_package.sty`.

Ce fichier `nom_de_package.sty` doit être traité selon le processus développé dans le paragraphe « Les packages en `.sty`, méthode simple ».

En résumé

- Les packages sont des outils permettant à LaTeX d'exécuter de nouvelles tâches : coloration du texte, règles typographiques, lettrines, encadrements... Grâce à eux, il est possible d'étendre les possibilités de LaTeX.
- Il existe des milliers de packages. Sous Windows, MiKTeX les télécharge et les installe automatiquement à la volée. Sous Mac OS et Linux il faut parfois télécharger manuellement ces fichiers.
- Pour utiliser un package, il suffit de l'appeler au début de votre document avec la commande `\usepackage{nompacage}`.
- La communauté des utilisateurs (dont vous faites désormais partie) peut ajouter autant de fonctions qu'elle le souhaite à LaTeX via la création de packages. Créez-en un qui prépare le café et vous ferez fortune. :)

Partie 2 : Utilisation basique de LaTeX

La première partie a été l'occasion pour vous de vous familiariser avec LaTeX et les notions théoriques utiles à son usage. Dans cette deuxième partie, vous allez apprendre des commandes et faire vos armes sur votre premier tp.

Maîtriser sa mise en page (1/2)

Les deux premiers chapitres de cette partie traiteront de mise en page. Nous y aborderons de nombreuses fonctions vous permettant de hiérarchiser votre contenu et de le présenter selon vos goûts.

Ce premier chapitre vous apprendra, entre autres, à :

- structurer vos documents ;
- maîtriser les différentes numérotations disponibles dans un document de classe `book` ;
- justifier et espacer vos paragraphes ;
- composer des préambules et annexes ;
- élaborer une page de garde.

Structure des documents Généralités

Tableau des éléments de structure

Vous avez appris dans le chapitre 4 à indiquer à LaTeX que vous alliez écrire en français. Il est maintenant temps d'apprendre à hiérarchiser des informations sous forme de **parties**, **chapitres**, **sections**, etc. Votre distribution LaTeX se servira de la hiérarchisation de vos informations pour mettre en page votre contenu proprement. LaTeX se base sur différentes règles de typographie et de mise en page professionnelles pour donner à chaque élément un placement optimal et une lisibilité accrue.

Ci-dessous, un tableau résumant les différents niveaux de hiérarchisation du contenu (de la grande partie jusqu'au tout petit sous-paragraphes).

Commandes des éléments de structure	
Partie	<code>\part{nom de la partie}</code>
Chapitre	<code>\chapter{nom du chapitre}</code>
Section	<code>\section{nom de la section}</code>
Sous-section	<code>\subsection{nom de la sous section}</code>
Sous-sous-section	<code>\subsubsection{nom de la sous sous section}</code>
Paragraphe	<code>\paragraph{nom du paragraphe}</code>
Sous-paragraphe	<code>\subparagraph{nom du sous paragraphe}</code>

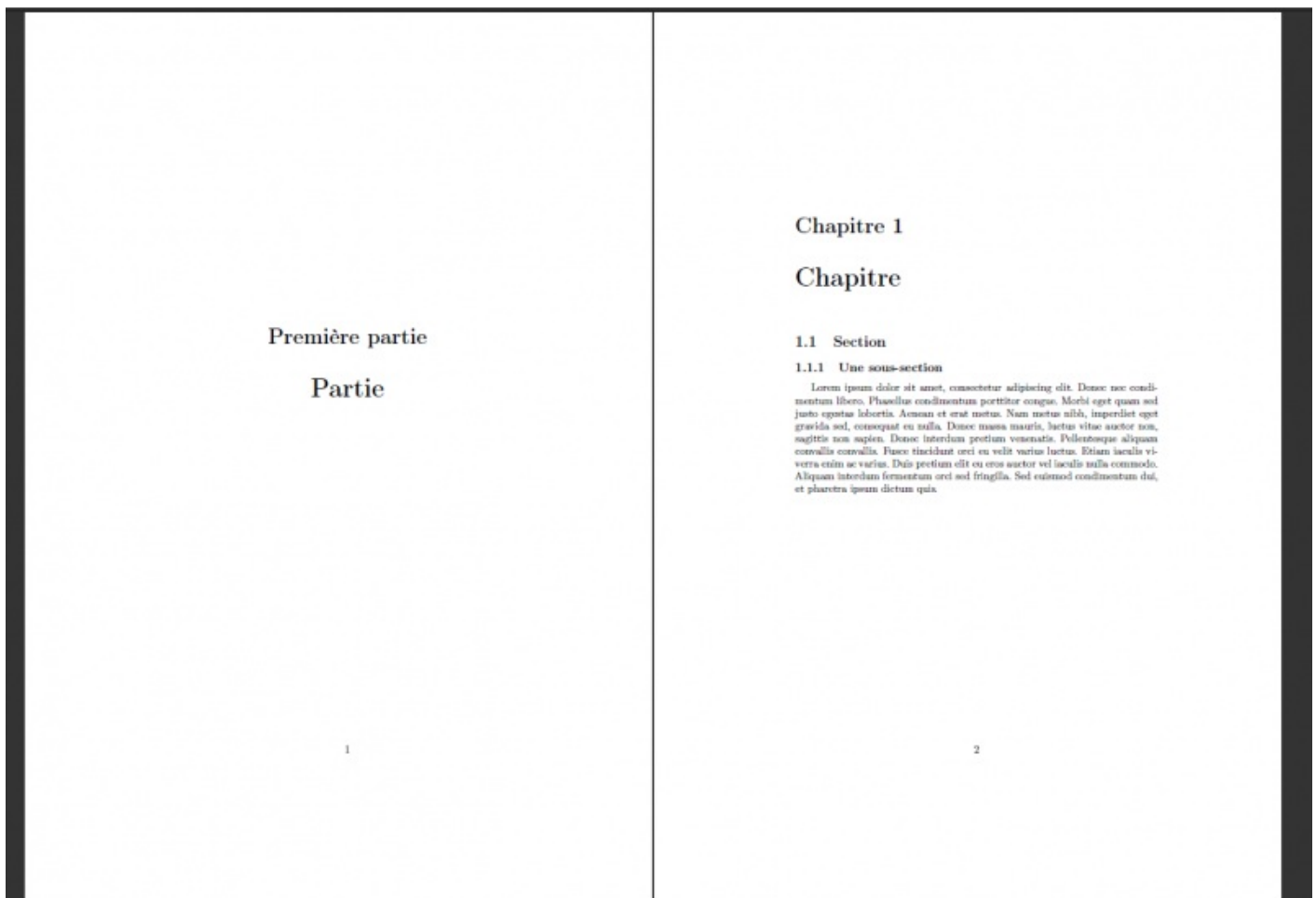
Notez que `\chapter` n'est pas disponible pour les classes de document `article` et `letter`.

Exercice

J'ai choisi volontairement de vous présenter le tableau précédent avant de vous donner un exemple pratique. L'heure de l'exercice a sonné !

Essayez d'écrire un document de classe `report`, en français et contenant les éléments de structure suivants : une partie, un chapitre, une section et une sous-section dans laquelle vous écrirez quelques lignes.

Votre résultat doit ressembler à la figure suivante disponible à la page suivante.



Résultat de l'exercice



Le texte latin utilisé pour cet exercice est un « faux texte », plus couramment appelé **lipsum**, généré sur l'internet. Ce type de texte permet de remplir un document d'informations factices afin de voir à quoi ressemblera le document final. C'est un outil précieux pour les webdesigners et les autres professionnels du graphisme et de la mise en page (générateur de texte).

Si ceci est le premier document que vous composez par vous-même, il est normal que cela vous demande de rechercher des notions dans les chapitres précédents ; de même, il est normal de trouver quelques coquilles dans votre code. Néanmoins, votre code final doit être proche de celui-ci (même si vous pouvez avoir choisi des noms de chapitres ou de sections différents) :

Code : TeX

```
\documentclass{report}

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

\begin{document}

\part{Partie}
\chapter{Chapitre}
\section{Section}
\subsection{Une sous-section}
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Donec nec condimentum libero. Phasellus condimentum porttitor
congue.
Morbi eget quam sed justo egetas lobortis. Aenean et erat metus.
Nam metus nibh, imperdiet eget gravida sed, consequat eu nulla.
Donec massa mauris, luctus vitae auctor non, sagittis non sapien.
Donec interdum pretium venenatis. Pellentesque aliquam convallis
```

```
convallis.
Fusce tincidunt orci eu velit varius luctus. Etiam iaculis viverra
enim ac varius.
Duis pretium elit eu eros auctor vel iaculis nulla commodo. Aliquam
interdum fermentum orci sed fringilla.
Sed euismod condimentum dui, et pharetra ipsum dictum quis.

\end{document}
```

Modification de la numérotation des éléments de structure

Numéroter les annexes

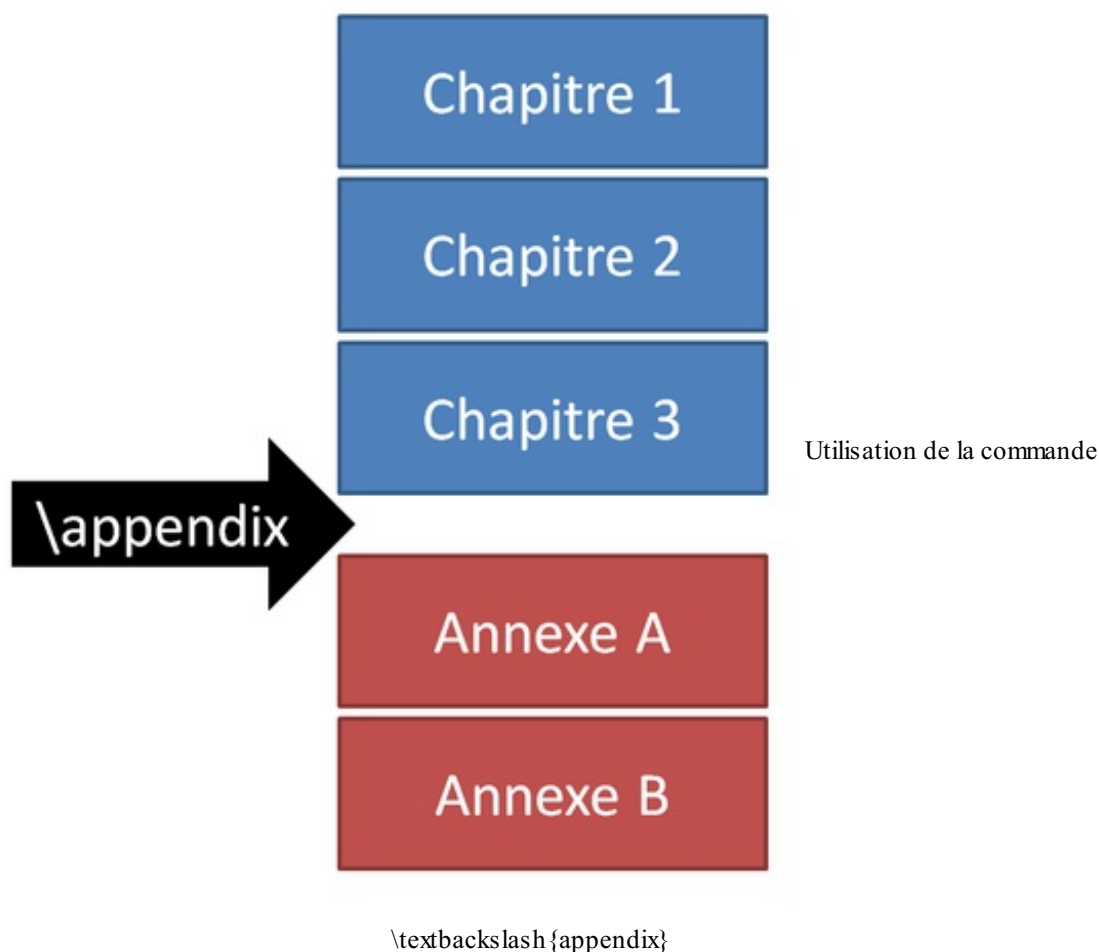
Rendre un rapport} nécessite souvent de mettre en annexe un certain nombre de textes et de photos (Cela permet, par exemple, de ne pas encombrer le contenu principal avec d'immenses tableaux). LaTeX permet à l'auteur de numéroter ses chapitres de contenu en chiffres et d'entamer une nouvelle numérotation des chapitres annexes avec des lettres.

La commande permettant de faire ceci est la commande `\appendix`. Elle se place entre le contenu et les annexes, comme vous pouvez le constater dans le prochain code source.

Code : TeX

```
\documentclass{report}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[français]{babel}
\begin{document}
\part{Partie}
\chapter{premier chapitre}
\chapter{second chapitre}
\chapter{troisième chapitre}
\chapter{quatrième chapitre}
\chapter{cinquième chapitre}
\chapter{sixième chapitre}
\appendix
\chapter{Un schéma}
% Pas besoin de changer de commande pour transformer un chapitre en
annexe grâce à appendix
\chapter{Un article annexe}
\chapter{Un canard}
\end{document}
```

Résumons ceci par un schéma (figure suivante).



Dans un article, étant donné que la commande `\chapter` ne peut pas être utilisée (elle n'est utilisable que pour les rapports et les livres), `\appendix` jouera sur la numérotation des sections.



Il est possible de créer des chapitres sans numéro ni lettre en tapant `\chapter*{nom du chapitre}`. L'ajout d'étoile en fin de mot fonctionne avec tous les éléments de structure donnés plus haut. Vous pouvez donc choisir de créer des parties sans numéro. Il en va de même pour les paragraphes et les sections.

Autres numérotations particulières

Dans le cas d'un livre, il peut être intéressant d'organiser et de numérotter de façons différentes les pages suivantes :

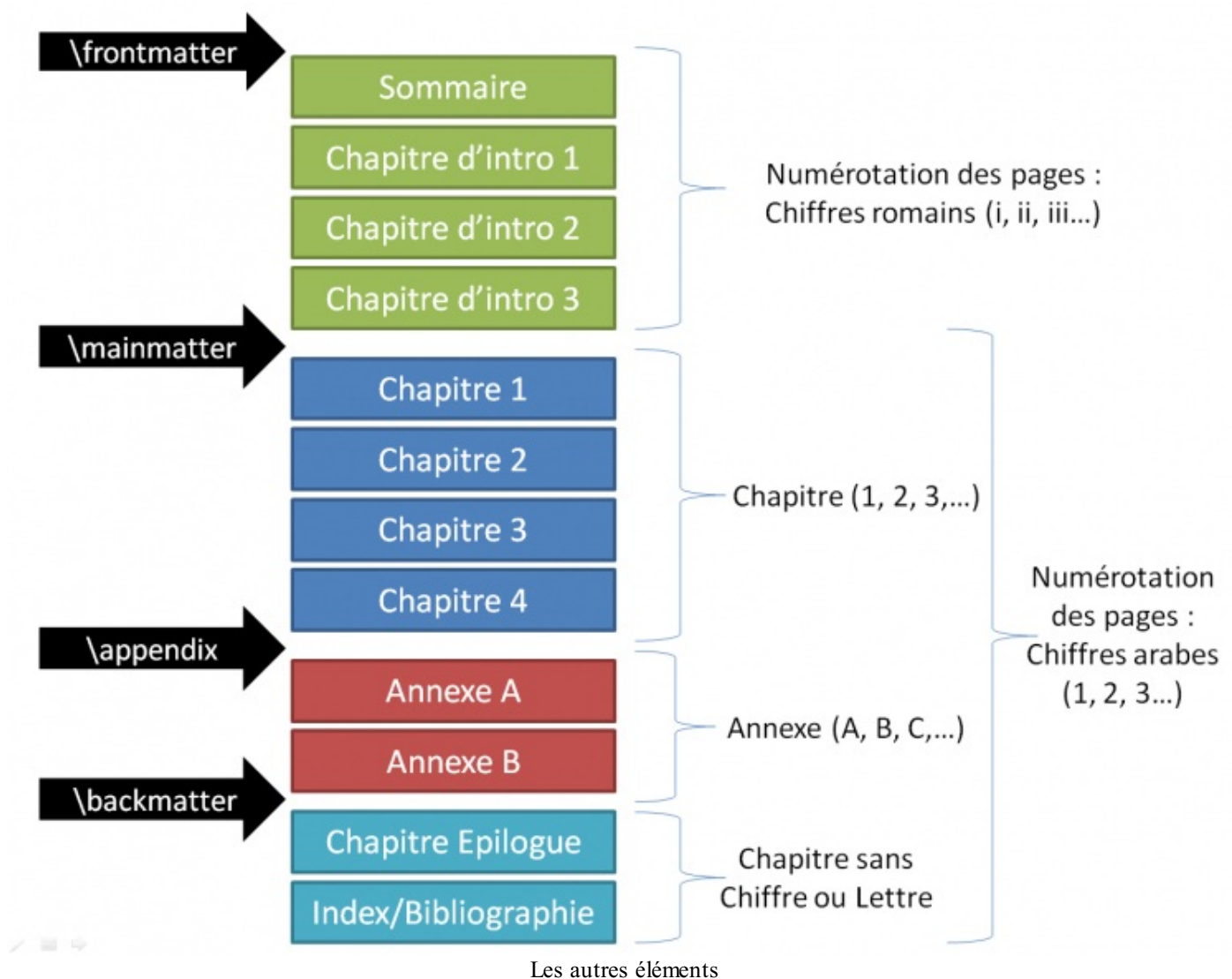
- les préambules et avant-propos ;
- les chapitres de contenu ;
- les annexes ;
- les pages d'index et de bibliographie (Nous apprendrons dans les chapitres 13 et 14 à créer des index ainsi que des bibliographies).

Ainsi, de nouvelles commandes apparaissent, fonctionnant de la même manière que la commande `\appendix`. Cependant, elles n'agissent pas uniquement sur la numérotation des chapitres et sections, mais également sur la numérotation des pages. Tout d'abord, la commande `\frontmatter`, qui se positionne juste après `\begin{document}`, permet de numérotter le préambule en chiffres romains.

Ensuite, `\mainmatter` se place entre le préambule et le premier chapitre. Cette commande permet de lancer la numérotation arabe habituelle des pages (1, 2, 3, etc.).

Enfin, `\backmatter` se place avant le chapitre épilogue, les index et bibliographies (que nous apprendrons à créer par la suite). Il a pour effet de stopper la numérotation des chapitres, mais pas la numérotation des pages.

La figure suivante résume tout ce qui a été dit.



Page de garde La théorie

Nous venons de créer une organisation hiérarchique des données présentes dans votre document. Néanmoins, il manque une page extrêmement importante : la page de garde.

La page de garde est composée de trois éléments :

- le titre du document (commande `\title{votre titre}`);
- l'auteur (commande `\author{les noms des auteurs}`);
- la date (commande `\date{la date que vous souhaitez}`).

Ces trois éléments sont introduits avant la commande `\begin{document}`, et une quatrième commande `\maketitle` se place juste après `\begin{document}` afin de faire comprendre à LaTeX que vous souhaitez composer une page de garde avec les trois éléments cités plus haut.

La pratique

La création d'une page de garde est assez simple et vous avez toutes les cartes en main pour en créer une. Vous allez donc effectuer un exercice dans lequel je souhaite que vous réalisiez un article avec une page de garde dans les conditions suivantes :

- l'auteur s'appelle John Doe ;
- il a écrit le livre le 3 mai 2010 ;

- « Les Lapins » est le titre de son livre ;
- le livre comporte une seule partie nommée « Éloge des lapins », qui comportera un paragraphe de faux texte.

Normalement, vous voici avec une belle page de garde (figure suivante) et un code ressemblant, au paragraphe près, à celui cité ci-dessous :

Code : TeX

```
\documentclass{book}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[français]{babel}

\title{Les Lapins}
\author{John \bsc{Doe}}
\date{3 mai 2010}
\begin{document}

\maketitle

\part{Éloge des lapins}

Votre gros paragraphe.

\end{document}
```

Les Lapins

John DOE

Page de garde

3 mai 2010

Alignements de texte et sauts

Alignements de texte

LaTeX justifie naturellement les paragraphes, il vous sera donc nécessaire d'utiliser différents environnements si vous souhaitez que votre texte soit centré, aligné à gauche ou à droite.



Rappel sur les environnements : un environnement commence par la commande `\begin{nom de l'environnement}` et se termine par `\end{nom de l'environnement}`.

Ici, trois environnements différents sont utilisables suivant vos besoins :

- l'environnement `flushright` pour aligner votre texte à droite ;
- l'environnement `center` pour centrer votre texte ;
- l'environnement `flushleft` pour aligner votre texte à gauche.

Une petite démonstration pratique de l'utilisation de ces trois environnements se trouve à la figure suivante (essayez de produire le même résultat : si vous n'y arrivez pas, reprenez le chapitre introduisant les environnements).

0.1 texte justifié

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut fringilla tempor diam, at dignissim enim pellentesque in. Phasellus mi arcu, fermentum euismod hendrerit sit amet, ultrices eu urna. Vestibulum congue velit id quam condimentum nec molestie nunc imperdiet. Maecenas ac mi ipsum, ut dictum nisi. In tincidunt vehicula dolor quis sagittis. Vestibulum nulla nibh, accumsan nec volutpat nec, faucibus a justo. Aenean aliquam aliquet rutrum. Suspendisse rhoncus sapien ac elit bibendum facilisis congue ante rutrum. Phasellus vestibulum porttitor luctus.

0.2 texte aligné à droite

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut fringilla tempor diam, at dignissim enim pellentesque in. Phasellus mi arcu, fermentum euismod hendrerit sit amet, ultrices eu urna. Vestibulum congue velit id quam condimentum nec molestie nunc imperdiet. Maecenas ac mi ipsum, ut dictum nisi. In tincidunt vehicula dolor quis sagittis. Vestibulum nulla nibh, accumsan nec volutpat nec, faucibus a justo. Aenean aliquam aliquet rutrum. Suspendisse rhoncus sapien ac elit bibendum facilisis congue ante rutrum. Phasellus vestibulum porttitor luctus.

Les

0.3 texte centré

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut fringilla tempor diam, at dignissim enim pellentesque in. Phasellus mi arcu, fermentum euismod hendrerit sit amet, ultrices eu urna. Vestibulum congue velit id quam condimentum nec molestie nunc imperdiet. Maecenas ac mi ipsum, ut dictum nisi. In tincidunt vehicula dolor quis sagittis. Vestibulum nulla nibh, accumsan nec volutpat nec, faucibus a justo. Aenean aliquam aliquet rutrum. Suspendisse rhoncus sapien ac elit bibendum facilisis congue ante rutrum. Phasellus vestibulum porttitor luctus.

0.4 texte aligné à gauche

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut fringilla tempor diam, at dignissim enim pellentesque in. Phasellus mi arcu, fermentum euismod hendrerit sit amet, ultrices eu urna. Vestibulum congue velit id quam condimentum nec molestie nunc imperdiet. Maecenas ac mi ipsum, ut dictum nisi. In tincidunt vehicula dolor quis sagittis. Vestibulum nulla nibh, accumsan nec volutpat nec, faucibus a justo. Aenean aliquam aliquet rutrum. Suspendisse rhoncus sapien ac elit bibendum facilisis congue ante rutrum. Phasellus vestibulum porttitor luctus.

différents alignements

Sauts

Ici nous allons apprendre à sauter des lignes et créer des paragraphes. Rien de compliqué à cela, vous allez voir.

Tout d'abord, **pour créer un paragraphe**, il vous suffit de sauter deux lignes. Rien de plus, juste deux sauts de lignes, comme vous pouvez le voir ci-dessous.

Code : TeX

```
\begin{document}  
Un paragraphe.  
  
Un second paragraphe.  
\end{document}
```

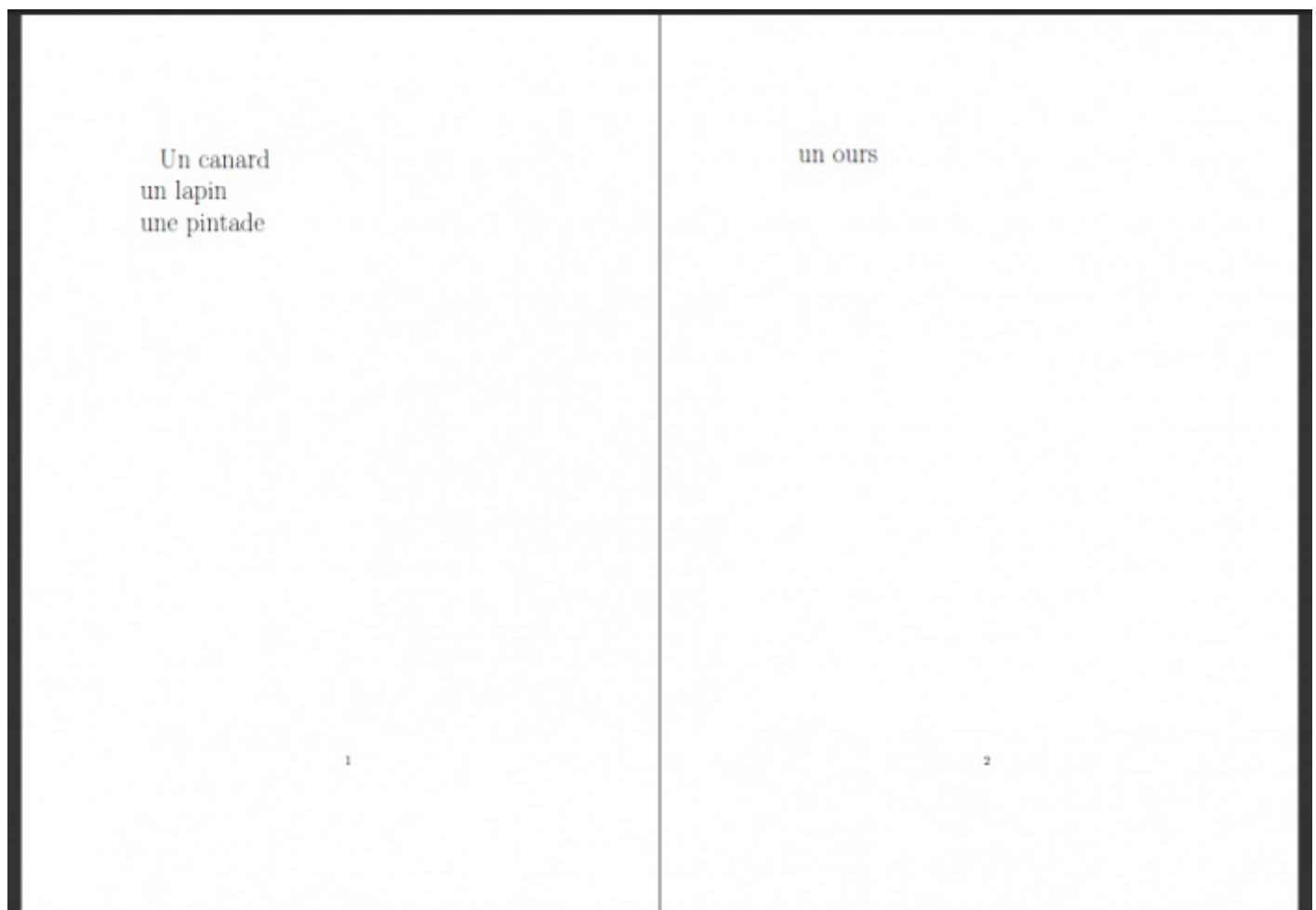
Pour **aller à la ligne sans créer de nouveau paragraphe** il vous faudra taper la commande `\newline` ou la commande `\\`.

Pour **faire un saut de page** il vous faudra taper la commande `\newpage`. Cette commande ferme le paragraphe en cours et en crée un nouveau au début de la page suivante.

Ci-dessous un exemple d'utilisation de ces nouvelles commandes ainsi que son rendu.

Code : TeX

```
Un canard\\un lapin\noline une pintade\nepage un ours
```



Rendu des différents types de sauts

En résumé

- Les documents LaTeX respectent une hiérarchie très précise : une partie contient des chapitres, scindés en sections, elles-mêmes divisées en sous-sections...
- Un document de classe book (livre) propose en plus un découpage global avec un préambule, un corps, des annexes et

des chapitres épiques



Dans ce tutoriel, nous utiliserons le terme « épiques » pour les parties situées entre `\backmatter` et `\end{document}`, parce que LaTeX les numérote de manière différente et que cette particularité peut être utile à certains. Cependant, dans la plupart des ouvrages francophones, il n'y a bien entendu qu'un seul épique.

- Les différentes parties d'un livre n'ont pas la même numérotation, ni au niveau des titres, ni au niveau des numéros de pages.
- Une page de garde simple (Il est bien sûr possible de faire une page de garde bien plus compliquée) comporte un titre, le nom du ou des auteurs et une date.
- Les environnements `flushright`, `center` et `flushleft` permettent respectivement d'aligner à droite, de centrer ou d'aligner à gauche du texte. Par défaut, LaTeX justifie le texte.
- Pour créer un nouveau paragraphe, il suffit de sauter deux lignes. Si vous le souhaitez, il est aussi possible d'utiliser `\newline` ou `\\` qui permettent d'effectuer un simple retour à la ligne. Enfin, `\newpage` engendre un saut de page.

Maîtriser sa mise en page (2/2)

Dans ce second chapitre sur la mise en page, nous n'allons pas voir comment structurer les informations, mais plutôt apprendre à mieux les présenter.

Au programme de ce chapitre :

- la modification de la forme de votre document ;
- les marges ;
- la suppression de la numérotation de vos pages ;
- les listes numérotées et non numérotées ;
- les en-têtes et pieds de page.

Ces éléments sont cruciaux pour une bonne mise en page. Les marges vous permettront par exemple de relier vos documents plus facilement.

La forme de votre publication

La commande `\documentclass{}`

Nous avons vu dans la première partie qu'un document commence forcément par la commande `\documentclass[options]{type du document}`. Les types de documents utilisés dans ce livre sont énoncés dans le tableau ci-dessous (déjà présent dans le chapitre 3).

Type de document	
article	Article (sous-entendu scientifique)
book	Livre
letter	Lettre
report	Rapport (stage, thèse...)

Par le biais du type de document, nous avons défini la nature de la publication que nous composons : livre, lettre ou autre. Néanmoins, certaines informations manquent :

- le format du papier sur lequel sera imprimée la publication (A4, A5...) ;
- la taille de la police principale ;
- l'alignement des équations (que nous apprendrons à écrire dans la partie 3) ;
- le positionnement de la première page de chaque chapitre (il est possible de faire en sorte que chaque chapitre commence sur la page de droite) ;
- l'information permettant de savoir si le document créé sera recto verso ou non.

Un second tableau fait donc naturellement son apparition, listant les options applicables à la commande `\documentclass{}`.

Options applicables à la commande <code>\documentclass{}</code>		
Description	Valeurs applicables	Valeur par défaut
Format du papier	a4paper, a5paper, letterpaper, b5paper...	letterpaper
Taille de la police principale	10pt, 11pt, 12pt	10pt
Alignement des équations	fleqn (à gauche)	centrées par défaut
Colonnes	onecolumn, twocolumn	onecolumn
Première page des chapitres	openany, openright	openright
Recto verso	oneside, twoside	article et report : oneside, book : twoside

Exemples pratiques

Le tableau que vous venez de lire est un outil précieux, mais encore faut-il savoir s'en servir. Dans un premier temps, copiez et collez le code source ci-dessous dans un fichier `test.tex`.

Code : TeX

```
\documentclass{book}

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

\begin{document}
\section{Un fort beau chapitre}
cinq ou six gros paragraphes de faux texte.
\end{document}
```

Tentez de remplacer `\documentclass{book}` par l'une des lignes ci-dessous (une seule à la fois : il ne faut pas deux commandes `\documentclass{ }` dans une même publication).

Code : TeX

```
\documentclass[twocolumn]{book}
\documentclass[twocolumn]{article}
\documentclass[12pt]{book}
\documentclass[11pt]{article}
\documentclass[twocolumn, 12pt]{book}
```

Essayez ces lignes (ajoutez un peu de texte pour mieux voir les différences, si vous préférez).

Retenez de cette sous-partie que la commande `\documentclass{ }` peut être personnalisée grâce à des options contenues entre crochets et séparées par des virgules de la façon suivante : `\documentclass[option1, option2, option3]{type}`.

Marges et interlignes

Marges

Vous venez d'apprendre à créer des publications adaptées à la taille de votre papier et aux principales exigences d'un auteur (choisir la taille de la police principale est essentiel, par exemple).

Voyns maintenant comment modifier les marges d'un document.

Cette modification se fait en trois temps. Tout d'abord, nous allons créer un document dans lequel nous appelons le package `layout`, puis nous allons utiliser la commande `\layout` introduite par celui-ci afin d'obtenir un gabarit qui nous donnera une idée de la taille de chaque marge.

En parallèle, nous imprimerons un document saturé de texte afin de voir le rendu produit avec ces tailles de marges.

Le code à utiliser pour la création du `layout` est donné ci-dessous. Il est accompagné d'une capture d'écran d'un document saturé en texte, ainsi que d'une capture d'écran d'un `layout` (figure suivante).

Code : TeX

```
\documentclass{article}
```



```

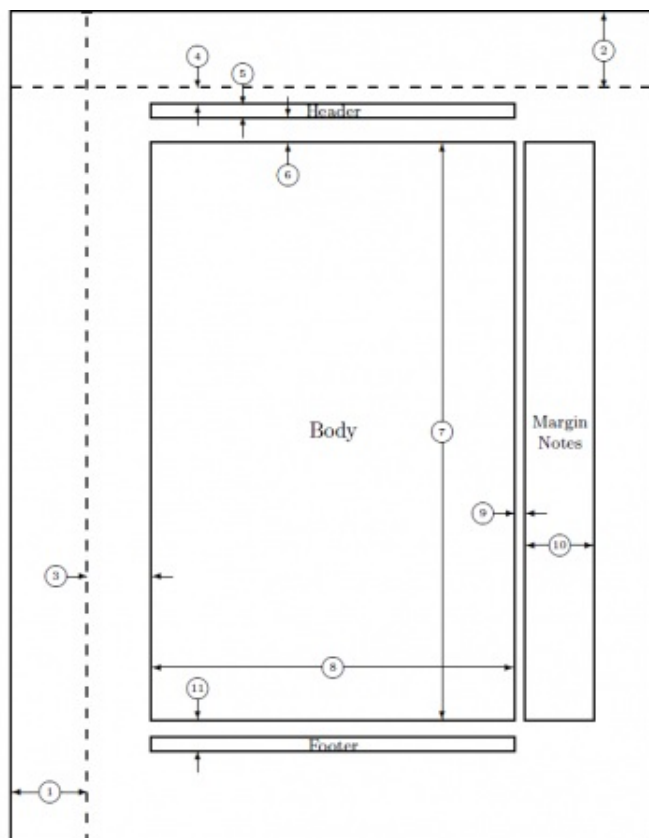
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[français]{babel}
\usepackage{layout}

\begin{document}

\layout

\end{document}

```



1 one inch + \hoffset	2 one inch + \voffset
3 \oddsidemargin = 62pt	4 \topmargin = 16pt
5 \headheight = 12pt	6 \headsep = 25pt
7 \textheight = 550pt	8 \textwidth = 345pt
9 \marginparsep = 11pt	10 \marginparwidth = 65pt
11 \footskip = 30pt	\marginparpush = 5pt (not shown)
\hoffset = 0pt	\voffset = 0pt
\paperwidth = 614pt	\paperheight = 794pt

1 section

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec nec condimentum libero. Phasellus condimentum porttitor congue. Morbi eget quam sed justo egestas lobortis. Aenean et erat metus. Nam metus nisl, imperdiet eget gravida sed, consequat eu nulla. Donec massa mauris, luctus vitae auctor non, sagittis non sapien. Donec interdum pretium venenatis. Pellentesque aliquam convallis convallis. Fusce tincidunt orci eu velit varius luctus. Etiam laeule vi-vera enim ac varius. Duis pretium elit eu eros auctor vel laeule nulla commodo. Aliquam interdum fermentum orci sed fringilla. Sed euismod condimentum dui, et pharetra ipsum dictum quis.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec nec condimentum libero. Phasellus condimentum porttitor congue. Morbi eget quam sed justo egestas lobortis. Aenean et erat metus. Nam metus nisl, imperdiet eget gravida sed, consequat eu nulla. Donec massa mauris, luctus vitae auctor non, sagittis non sapien. Donec interdum pretium venenatis. Pellentesque aliquam convallis convallis. Fusce tincidunt orci eu velit varius luctus. Etiam laeule vi-vera enim ac varius. Duis pretium elit eu eros auctor vel laeule nulla commodo. Aliquam interdum fermentum orci sed fringilla. Sed euismod condimentum dui, et pharetra ipsum dictum quis.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec nec condimentum libero. Phasellus condimentum porttitor congue. Morbi eget quam sed justo egestas lobortis. Aenean et erat metus. Nam metus nisl, imperdiet eget gravida sed, consequat eu nulla. Donec massa mauris, luctus vitae auctor non, sagittis non sapien. Donec interdum pretium venenatis. Pellentesque aliquam convallis convallis. Fusce tincidunt orci eu velit varius luctus. Etiam laeule vi-vera enim ac varius. Duis pretium elit eu eros auctor vel laeule nulla commodo. Aliquam interdum fermentum orci sed fringilla. Sed euismod condimentum dui, et pharetra ipsum dictum quis.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec nec condimentum libero. Phasellus condimentum porttitor congue. Morbi eget quam sed justo egestas lobortis. Aenean et erat metus. Nam metus nisl, imperdiet eget gravida sed, consequat eu nulla. Donec massa mauris, luctus vitae auctor non, sagittis non sapien. Donec interdum pretium venenatis. Pellentesque aliquam convallis convallis. Fusce tincidunt orci eu velit varius luctus. Etiam laeule vi-vera enim ac varius. Duis pretium elit eu eros auctor vel laeule nulla commodo. Aliquam interdum fermentum orci sed fringilla. Sed euismod condimentum dui, et pharetra ipsum dictum quis.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec nec condimentum libero. Phasellus condimentum porttitor congue. Morbi eget quam sed justo egestas lobortis. Aenean et erat metus. Nam metus nisl, imperdiet eget gravida sed, consequat eu nulla. Donec massa mauris, luctus vitae auctor non, sagittis non sapien. Donec interdum pretium venenatis. Pellentesque aliquam convallis convallis. Fusce tincidunt orci eu velit varius luctus. Etiam laeule vi-vera enim ac varius. Duis pretium elit eu eros auctor vel laeule nulla commodo. Aliquam interdum fermentum orci sed fringilla. Sed euismod condimentum dui, et pharetra ipsum dictum quis.

Le duo layout / page saturée en texte

Ces deux éléments conjugués nous permettent de mieux visualiser les marges et leurs grandeurs respectives.

Afin de modifier celles-ci, nous utiliserons le package `geometry` dans lequel nous spécifierons les tailles des marges en haut, en bas, à gauche puis à droite.

Tout ceci s'écrit avec la syntaxe suivante :

Code : TeX

```

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

```

Bien sûr, vous pouvez modifier à votre guise les différentes longueurs.

Résumons les étapes :

- création du layout ;
- création d'une page **saturée** en texte (afin de mieux visualiser les marges) ;
- modification des marges (via le package `geometry`) ;
- création d'une page remplie de texte pour voir le rendu des marges (facultatif).

Interlignes

Dans certains types de documentation, un interlignage une fois et demie ou deux fois supérieur à la normale est requis. Nous allons voir ensemble comment utiliser ces derniers dans un document. Il est également possible de créer des interlignes d'une taille définie par l'auteur, mais ce n'est pas notre souhait ici. Afin d'obtenir des interlignes personnalisés, il faut utiliser le package `setspace` ainsi que les commandes `\onehalfspacing` et `\doublespacing` dans le préambule, qui permettent d'obtenir dans tout le document un interligne respectivement 1,5 et 2 fois plus grand que l'interligne habituel.

Pour ne changer les interlignes que dans des petits morceaux de votre composition, les environnements `onehalfspace` et `doublespace` sont adaptés. L'exemple ci-dessous illustre ce cas (figure suivante).

Code : TeX

```
\documentclass[10pt]{article}

\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[français]{babel}
\usepackage{setspace}

\begin{document}

\section{interligne simple}

Un paragraphe.

\section{interligne intermédiaire}

\begin{onehalfspace}
Un autre paragraphe.
\end{onehalfspace}

\section{interligne double}

\begin{doublespace}
Encore un autre paragraphe.
\end{doublespace}

\end{document}
```

1 interligne simple

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed vitae est eget urna luctus posuere vel ac metus. Aliquam pharetra mi in nibh molestie ac. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed vitae est eget urna luctus.

2 interligne intermédiaire

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed vitae est eget urna luctus posuere vel ac metus. Aliquam pharetra mi in nibh molestie ac. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed vitae est eget urna luctus.

Interlignes

3 interligne double

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed vitae est eget urna luctus posuere vel ac metus. Aliquam pharetra mi in nibh molestie ac. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed vitae est eget urna luctus.

Les listes

Nous allons maintenant apprendre à créer des listes.

Les listes sont très utiles dans un document et peuvent être selon vos souhaits :

- des listes à puces (comme celle-ci) ;
- des listes numérotées ;
- des listes de description (beaucoup moins fréquentes).

Listes à puces

La création d'une liste à puces se fait en trois temps :

- ouverture de l'environnement `itemize` ;
- écriture de chaque élément de la liste précédé de la commande `\item` ;
- fermeture de l'environnement `itemize`.

Rien de compliqué là-dedans, vous pouvez voir ci-dessous un exemple pratique présentant un code de liste à puces, ainsi que son rendu (figure suivante).

Petit bonus, le changement de la forme de la puce est introduit dans le code.

Code : TeX

```
\begin{document}

\begin{itemize}

\item Un canard.
```

```

\item Un mammouth.
\item Un canard.
\item Un mammouth.
\item Un canard.
\item Un mammouth.
\item[@] Une pintade. % En plaçant un @ entre crochets après \item,
j'ai transformé la puce en @
\item[0] Un lapin.

\end{itemize}

```

- un canard
- un mammouth
- un canard
- un mammouth
- un canard
- un mammouth
- @ une pintade
- 0 un lapin

Liste à puces

Listes numérotées

La création d'une liste numérotée se fait également en trois temps (seul le nom de l'environnement change) :

- ouverture de l'environnement `enumerate` ;
- écriture de chaque élément de la liste précédé de la commande `\item` ;
- fermeture de l'environnement `enumerate`.

Si vous avez compris comment construire des listes à puces, vous n'aurez aucune difficulté à créer des listes numérotées.

Cependant, un exemple de code et un rendu en image (figure suivante) ne peuvent pas faire de tort.

Code : TeX

```

\documentclass{article}
\begin{document}
\begin{enumerate}
\item un canard
\item un mammouth
\item un canard
\item un mammouth
\item un canard
\item un mammouth
\item[@] une pintade % En plaçant un @ entre crochets après \item,
j'ai transformé la puce en @
\item[0] un lapin
\end{enumerate}
\end{document}

```

1. un canard
2. un mammouth
3. un canard
4. un mammouth
5. un canard
6. un mammouth
7. une pintade

Liste numérotée

Listes de description

Si vous souhaitez écrire une série de définitions, l'environnement `description` est fait pour vous. Il vous permet de remplacer les puces par des expressions de votre choix qui seront mises en gras.

Ces mots en gras devront être placés entre crochets juste après la commande `\item` dans le code source.

Démonstration ci-dessous (figure suivante) :

Code : TeX

```
\documentclass{article}
\begin{document}
\begin{description}
\item[Un canard :] bestiole qui fait coin.
\item[Un poulpe :] bestiole qui fait bloub.
\item[Un ornithorynque :] bestiole qui fait rire.
\item[Un ours :] bestiole qui fait mal.
\end{description}
\end{document}
```

un canard : bestiole qui fait coin

un ornithorynque : bestiole qui fait rire

Liste de description

un ours : bestiole qui fait mal

Les listes n'ont rien de sorcier : en faire usage aérera vos documents en plus de vous permettre, par exemple, de présenter simplement des processus, des manipulations ou des protocoles opératoires (ce ne sont que des exemples, vous pouvez très bien vous en servir pour lister des ingrédients dans la recette des crêpes).

Les styles

Nous venons d'apporter de nombreuses modifications à nos publications. Elles ont été mises en forme, structurées, listées.

Il nous reste un élément essentiel à personnaliser : les **en-têtes** et **pieds de pages**.

Ces éléments ont eux aussi été étudiés par les concepteurs de LaTeX. Il existe bien sûr des packages permettant d'en faire des choses originales. Néanmoins, les pères de LaTeX ont choisi trois couples en-têtes / pieds de page qui leur semblaient les plus adaptés à un usage professionnel (nous utiliserons d'autres packages et des mises en forme élaborées plus loin dans le cours).

Pour utiliser un couple en-têtes / pieds de page, il vous suffira de choisir ce que LaTeX appelle un **style**. Ces trois styles sont :

- le style `plain` : il permet d'insérer le numéro de page au milieu du pied de page ;
- le style `headings` : il permet d'insérer le nom du chapitre et le numéro de page en en-tête. Le pied de page est vide ;
- le style `empty` : l'en-tête et le pied de page sont vides.

Dans les pages suivantes, vous trouverez des captures d'écran de documents composés respectivement avec les commandes `plain` (figure suivante), `headings` (figure suivante) et enfin `empty` (figure suivante).

Pour donner un style à une page en particulier, il suffit d'utiliser la commande `\pagestyle{nom du style}` que nous insérerons au début de la page à modifier.

Rien de bien compliqué, n'est-ce pas ?

Dans le premier TP, nous apprendrons à personnaliser de manière bien plus approfondie nos en-têtes et pieds de page.

erat metus. Nam metus nullo, imperdiet eget gravida sed, consequat eu nulla. Donec massa massa, luctus vitae auctor non, sagittis non sapien. Donec interdum pretium venenatis. Pellentesque aliquam convallis convallis. Fusce tincidunt orci eu velit varius luctus. Etiam iaculis viverra enim ac varius. Duis pretium elit eu eros auctor vel iaculis nulla commodo. Aliquam interdum fermentum orci sed fringilla. Sed euismod condimentum dui, et pharetra ipsum dictum quis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec nec condimentum libero. Phasellus condimentum porttitor congue. Morbi eget quam sed justo egetas lobortis. Aenean et erat metus. Nam metus nullo, imperdiet eget gravida sed, consequat eu nulla. Donec massa massa, luctus vitae auctor non, sagittis non sapien. Donec interdum pretium venenatis. Pellentesque aliquam convallis convallis. Fusce tincidunt orci eu velit varius luctus. Etiam iaculis viverra enim ac varius. Duis pretium elit eu eros auctor vel iaculis nulla commodo. Aliquam interdum fermentum orci sed fringilla. Sed euismod condimentum dui, et pharetra ipsum dictum quis.

4

Le style plain

4 CHAPITRE 1. HEADINGS

erat metus. Nam metus nullo, imperdiet eget gravida sed, consequat eu nulla. Donec massa massa, luctus vitae auctor non, sagittis non sapien. Donec interdum pretium venenatis. Pellentesque aliquam convallis convallis. Fusce tincidunt orci eu velit varius luctus. Etiam iaculis viverra enim ac varius. Duis pretium elit eu eros auctor vel iaculis nulla commodo. Aliquam interdum fermentum orci sed fringilla. Sed euismod condimentum dui, et pharetra ipsum dictum quis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec nec condimentum libero. Phasellus condimentum porttitor congue. Morbi eget quam sed justo egetas lobortis. Aenean et erat metus. Nam metus nullo, imperdiet eget gravida sed, consequat eu nulla. Donec massa massa, luctus vitae auctor non, sagittis non sapien. Donec interdum pretium venenatis. Pellentesque aliquam convallis convallis. Fusce tincidunt orci eu velit varius luctus. Etiam iaculis viverra enim ac varius. Duis pretium elit eu eros auctor vel iaculis nulla commodo. Aliquam interdum fermentum orci sed fringilla. Sed euismod condimentum dui, et pharetra ipsum dictum quis.

Le style headings

erat metus. Nam metus nihil, impedit eget gravis sed, consequat ex nulla. Donec massa mauris, luctus vitae auctor non, sagittis non sapiens. Donec interdum pretium venenatis. Pellentesque aliquam convallis convallis. Fusce tincidunt orci eu velit varius luctus. Etiam iaculis viverra enim ac varius. Duis pretium elit eu eros auctor vel iaculis nulla conmodo. Aliquam interdum fermentum orci sed fringilla. Sed euismod condimentum dui, et pharetra ipsum dictum quis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec nec condimentum libero. Phasellus condimentum porttitor congue. Morbi eget quam sed justo egetas lobortis. Aenean et erat metus. Nam metus nihil, impedit eget gravis sed, consequat ex nulla. Donec massa mauris, luctus vitae auctor non, sagittis non sapiens. Donec interdum pretium venenatis. Pellentesque aliquam convallis convallis. Fusce tincidunt orci eu velit varius luctus. Etiam iaculis viverra enim ac varius. Duis pretium elit eu eros auctor vel iaculis nulla conmodo. Aliquam interdum fermentum orci sed fringilla. Sed euismod condimentum dui, et pharetra ipsum dictum quis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec nec condimentum libero. Phasellus condimentum porttitor congue. Morbi eget quam sed justo egetas lobortis. Aenean et erat metus. Nam metus nihil, impedit eget gravis sed, consequat ex nulla. Donec massa mauris, luctus vitae auctor non, sagittis non sapiens. Donec interdum pretium venenatis. Pellentesque aliquam convallis convallis. Fusce tincidunt orci eu velit varius luctus. Etiam iaculis viverra enim ac varius. Duis pretium elit eu eros auctor vel iaculis nulla conmodo. Aliquam interdum fermentum orci sed fringilla. Sed euismod condimentum dui, et pharetra ipsum dictum quis.

Le style empty

En résumé

- On peut personnaliser la forme de son document en ajoutant des paramètres à la commande `\documentclass{}`. Cela nous permet d'indiquer la taille du papier (A4, A5...), la taille de la police (10pt, 11pt...), le nombre de colonnes, etc.
- Il est possible de paramétrer les marges d'un document. On fait appel pour cela au package `geometry`. Ici, nous n'avons fait qu'un travail grossier, mais le premier TP vous apprendra à être beaucoup plus précis (en pratique vous constaterez que les tailles de marges par défaut sont souvent suffisantes).
- Il existe différents types de listes, chacune adaptée au genre de données listées : liste normale (`itemize`), liste numérotée (`enumerate`), liste de description (`description`).
- La commande `\pagestyle` nous permet de choisir comment l'en-tête et le pied de page seront affichés : avec le numéro de page en pied de page (`plain`), avec le nom du chapitre et le numéro de page en en-tête (`headings`) ou sans en-tête ni pied de page (`empty`).

Les polices

Ce troisième chapitre traitera des polices et de toutes les façons de modifier du texte. La bonne maîtrise de ces outils vous permettra de facilement mettre en avant des mots ou des phrases, rendant ainsi vos documents plus lisibles et structurés.

Quelques conseils seront distillés, en rapport avec les chapitres précédents (en particulier à propos de la commande `\documentclass` et de la taille de police par défaut).

À la fin de ce chapitre, vous pourrez sans problème :

- mettre en forme du texte (gras, italique, soulignement...);
- changer la couleur d'un texte ;
- changer de façon ponctuelle ou définitive la police d'un document.

Rien de sorcier dans les commandes, leur manipulation ne devrait pas poser de problème. Ce chapitre étant le plus facile du livre, profitez-en pour vous entraîner.

Graisse, style, taille Taille de texte

Nous allons dans ce paragraphe comprendre comment la taille du texte peut être modifiée, de façon à écrire certains mots en plus grand que les autres.

LaTeX propose dix commandes différentes permettant d'augmenter ou de diminuer la taille de votre texte selon votre envie. Ces commandes s'utilisent très simplement et possèdent deux syntaxes différentes :

Code : TeX

```
\commande{mon bout de texte} % rien d'inhabituel
{\commande mon bout de texte} % nouvelle façon
```

Dans le cas des modifications de tailles de texte, ces deux méthodes fonctionnent. Néanmoins, je vous conseille d'utiliser la syntaxe `\commande{mon bout de texte}` car elle est similaire à la majorité des autres syntaxes utilisées dans LaTeX.

Maman ?
Oui chéri !
il me faut...
de quoi as-tu besoin mon lapin ?
il me faut de l'eau !
Pardon ? !
De l'eau !
Il me faut de l'eau !
de l'eaaaaaaau !
de l'eaaauuuuuuuuu !

Les tailles de texte

Revenons à nos commandes. Comme dit plus haut, le texte peut subir dix transformations différentes au niveau de la taille. Cela nous donne donc une échelle de taille de texte assez large (figure suivante).

Une fois n'est pas coutume, un beau tableau devrait nous aider à y voir plus clair dans ce tas de commandes. Testez ces dernières, vous serez surpris des changements qu'elles effectueront dans vos mises en page.

Commande	Taille de texte
<code>\tiny</code>	Minuscule
<code>\scriptsize</code>	Très très petite
<code>\footnotesize</code>	Très petite
<code>\small</code>	Petite
<code>\normalsize</code>	Normale (définie dans <code>\documentclass</code>)
<code>\large</code>	Légèrement plus grande que la normale
<code>\Large</code>	Grande
<code>\LARGE</code>	Très grande
<code>\huge</code>	Très très grande
<code>\Huge</code>	Énorme !



Point important : lors de l'appel de `\documentclass`, vous avez choisi une taille de police pour votre document (10 pt par défaut, que vous pouvez augmenter jusqu'à 12 pt). Il serait maladroit d'encadrer l'intégralité d'un document dans une commande `\large` au lieu de simplement augmenter la taille de police par défaut.

Graisse, soulignement...

Nous venons d'apprendre à rendre plus petite ou plus grosse une portion de texte. Nous allons à présent apprendre à mettre ledit texte en gras, à le souligner ainsi que moult commandes de modifications.

Il existe ici trois méthodes différentes pour appliquer des transformations de type graisse et soulignement à un texte :

- la méthode habituelle `\commande{mon bout de texte}` ;
- la méthode `{\commande mon bout de texte}` ;
- les environnements.

A la différence des commandes de tailles de textes s'utilisant indifféremment à l'intérieur ou à l'extérieur des accolades, il existe ici des commandes différentes suivant que vous utilisiez l'une ou l'autre des deux méthodes.

Pour vous épargner de multiples paragraphes, les exemples et commandes sont synthétisés ci-dessous sous forme de tableau (à imprimer et à garder sous la main).

Mise en forme		
Modification	Commande	Rendu
Normal	<code>{\normalfont un lapin}</code>	un lapin
	<code>\begin{rm}un lapin \end{rm}</code>	un lapin
Gras	<code>\textbf{un lapin}</code>	un lapin
	<code>{\bfseries un lapin}</code>	un lapin
	<code>\begin{bf}un lapin \end{bf}</code>	un lapin

Italique	<code>\textit{un lapin}</code>	<i>un lapin</i>
	<code>{\itshape un lapin}</code>	<i>un lapin</i>
	<code>\begin{it}un lapin \end{it}</code>	<i>un lapin</i>
Penché	<code>\textsl{un lapin}</code>	<i>un lapin</i>
	<code>{\slshape un lapin}</code>	<i>un lapin</i>
	<code>\begin{sl}un lapin \end{sl}</code>	<i>un lapin</i>
Machine à écrire	<code>\texttt{un lapin}</code>	un lapin
	<code>{\ttfamily un lapin}</code>	un lapin
	<code>\begin{tt}un lapin \end{tt}</code>	un lapin
Petites majuscules	<code>\textsc{un lapin}</code>	UN LAPIN
	<code>{\scshape un lapin}</code>	UN LAPIN
	<code>\begin{sc}un lapin \end{sc}</code>	UN LAPIN
Exposant (En mode texte)	Un canard <code>un lapin</code>	Un canard ^{un lapin}
Encadrer (Paramétrable)	<code>\fbox{un lapin}</code>	<div style="border: 1px solid black; padding: 2px;">un lapin</div>
Soulignement (Package soul)	<code>\ul{un lapin}</code>	<u>un lapin</u>
Soulignement (Package ulem)	<code>\uuline{un lapin}</code>	<u><u>un lapin</u></u>
	<code>\uwave{un lapin}</code>	<u>un lapin</u>
Barrer (Package soul)	<code>\st{un lapin}</code>	un lapin



Gardez à l'esprit qu'un trop grand nombre de mots mis en forme peut nuire à votre document et le rendre totalement illisible.

Utilisez ces commandes avec parcimonie.

La commande `\emph`, un cas à part

Nous venons de voir une vingtaine de commandes ayant chacune un usage bien défini. Certaines grossissent le texte, d'autres le barrent ou le soulignent. En bref, chacune a une fonction simple et facile à comprendre.

La commande `\emph{texte}` est différente et permet tout simplement de dire à LaTeX « *ce bout de texte est important, mets-le en valeur* ». LaTeX se chargera alors tout seul de choisir la meilleure façon de le mettre en valeur.

Petit exemple, un mot mis en valeur dans un texte en italique, et un mot mis en valeur dans un texte normal (figure suivante).

Code : TeX

```
\documentclass{book}
\begin{document}
\textit{Lorem ipsum dolor sit amet, \emph{consectetuer} adipiscing elit.}
Lorem ipsum dolor sit amet, \emph{consectetuer} adipiscing elit.
\end{document}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
 Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

La commande `\emph{}`

Vous constatez que `\emph` change de comportement suivant le contexte ; en cela réside sa force.

Conclusion : utiliser l'italique peut souvent être une erreur, mieux vaut lui préférer `\emph`. Dans le premier TP, nous apprendrons à faire en sorte que `\emph` mette en gras les mots importants.

Couleur

Nous allons ici apprendre à colorer du texte et à le rendre tel un arc-en-ciel au cœur de vos publications. Soulignons tout de même la dangerosité avérée d'une portion de texte rose au sein d'une thèse et la probabilité élevée que cette portion cause pertes et fracas à l'heure du jury.

Les huit couleurs par défaut

Pour la colorisation du texte, nous utiliserons le package `color`.

Ce package propose par défaut huit couleurs (`black`, `white`, `red`, `green`, `blue`, `yellow`, `magenta` et `cyan`), que vous pourrez utiliser à votre guise pour colorer du texte via la commande suivante :

Code : TeX

```
\textcolor{couleur}{texte en couleur}
```

Création de nouvelles couleurs

Avec huit couleurs, vous ne pouvez guère aller loin, raison pour laquelle de bonnes âmes ont conçu un système vous permettant d'en composer de nouvelles.

Pour ce faire, une commande est disponible : `\definecolor`.

Cette commande permet de créer des couleurs à partir de niveaux de gris ou d'un mélange de trois couleurs (rouge, vert et bleu).

Les couleurs créées recevront chacune un nom que vous choisirez et s'utiliseront de la même manière que précédemment, toujours grâce à la commande `\textcolor` et selon la syntaxe vue plus haut :

Code : TeX

```
\textcolor{le nom de couleur choisi}{texte en couleur}
```

Niveaux de gris

Définir une couleur en niveaux de gris revient à choisir une couleur entre le blanc et le noir. Ici, nous estimerons que le gris se trouve sur une échelle située entre 0 (le noir) et 1 (le blanc). Choisir un niveau de gris consistera à prendre un nombre à deux décimales (nous utiliserons un point entre le 0 et ses décimales) situé entre 0 et 1.

La syntaxe de `\definecolor` dans le cas des nuances de gris sera :

Code : TeX

```
\definecolor{le nom choisi sans espace}{gray}{le nombre à deux décimales}
```

Mélange de rouge, vert et bleu

Afin de créer de nouvelles couleurs, nous mélangerons trois couleurs initiales dans des proportions choisies. De la même manière que pour les nuances de gris, nous choisirons tour à tour l'intensité de rouge, de vert et de bleu que contiendra la couleur que nous voulons créer.

La syntaxe de `\definecolor` dans le cas du mélange rouge, vert, bleu sera :

Code : TeX

```
\definecolor{le nom de couleur choisi}{rgb}{taux de rouge entre 0 et 1, taux de vert, taux de bleu}
```

Les packs de polices

Nous avons appris à changer la taille d'un morceau de texte et à lui faire subir de multiples transformations, de l'italique au soulignement. C'est bien, mais il nous reste encore une chose à apprendre à changer : **les polices de caractères**.

Il y a fort à parier que depuis le début de cet ouvrage, au fur et à mesure de vos compilations, l'envie vous a titillé de changer la police de vos textes.

Pour ce qui est des polices, LaTeX a des avantages à double tranchant. Rappelons-nous que LaTeX a été développé suite à l'exaspération de chercheurs face à la mise en page plutôt passable de leurs textes. Ils ont souhaité lors de la création du langage que l'ensemble des éléments d'une publication, qu'ils soient en gras, en majuscules ou dans une formule, soient cohérents entre eux.

Cette réflexion profonde a entraîné une innovation : la création des packs de polices.

L'idée centrale des packs de polices est de donner une cohérence au niveau de la typographie à l'intégralité du texte présent dans un document. Résultat : un pack comprenant quatre polices cohérentes correspondant respectivement à :

- des caractères avec empattements ;
- des caractères sans empattements ;
- des caractères façon machine à écrire (à chasse fixe) ;
- des caractères servant à écrire des formules mathématiques.

Bien qu'il existe un pack par défaut fourni dans LaTeX (utilisant la police Computer Modern) vous pouvez changer de pack de polices via l'utilisation des packages cités ci-dessous.

Il vous suffira de les appeler via la commande `\usepackage{nom_du_pack}`.

Une fois n'est pas coutume, voici une longue liste de noms avec les captures d'écran correspondantes.

Texte avec empattements

Texte sans empattements

Par défaut

Texte type machine à écrire

$$\int_4^8 kx dx \sum_{i=0}^n y_n^7 + 4569852321$$

Texte avec empattements

Texte sans empattements

Pack bookman

Texte type machine à écrire

$$\int_4^8 kx dx \sum_{i=0}^n y_n^7 + 4569852321$$

Texte avec empattements

Texte sans empattements

Pack charter

Texte type machine à écrire

$$\int_4^8 kx dx \sum_{i=0}^n y_n^7 + 4569852321$$

Texte avec empattements

Texte sans empattements

Pack newcent

Texte type machine à écrire

$$\int_4^8 kx dx \sum_{i=0}^n y_n^7 + 4569852321$$

Texte avec empattements

Texte sans empattements

Pack lmodern (le classique)

Texte type machine à écrire

$$\int_4^8 kx dx \sum_{i=0}^n y_n^7 + 4569852321$$

Texte avec empattements

Texte sans empattements

Pack mathpazo

Texte type machine à écrire

$$\int_4^8 kx dx \sum_{i=0}^n y_n^7 + 4569852321$$

Texte avec empattements

Texte sans empattements

Pack mathptmx

Texte type machine à écrire

$$\int_4^8 kx dx \sum_{i=0}^n y_n^7 + 4569852321$$

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

Code : TeX

```
{\fontfamily{code_de_la_police}\selectfont mon bout de texte}
```

Différentes polices sont utilisables, j'en ai sélectionné pour vous et vous les livre dans un nouveau tableau (toujours avec des exemples de rendu).

Code de la police	Exemple	
bch	Charter	bch
cmr	Computer Modern	bcr
lmr	Latin Modern Roman	lmr
lmss	Latin Modern Sans Empattement	lmss
lmssq	Latin Modern Sans Emp. Expansé	lmssq
lmtt	Latin Modern Typewriter	
pag	Avant Garde	
pcr	Courier	pcr

pbk	Bookman	pbk
phv	Helvetica	phv
pnc	New Century Schoolbook	pnc
ppl	Palatino	ppl
ptm	Times	ptm
pzc	<i>Zapf Chancery</i>	pzc



Il est souvent tentant de changer en permanence de police. Néanmoins, un document sobre sera bien plus lisible pour un correcteur ou un lecteur qu'un document nécessitant en permanence de se réadapter à la police. Utilisez ces commandes sans lourdeur, sans quoi vos publications deviendront indigestes.

En résumé

- La taille du texte peut être paramétrée avec des commandes comme `\small`, `\normal`, `\large`, `\huge`...
- De nombreuses mises en forme sont utilisables via les commandes appropriées. Ainsi, il est notamment possible de passer le texte en souligné (`\ul`), italique (`\textit`), gras (`\textbf`)...
- Les créateurs de LaTeX ont privilégié des **packs** cohérents typographiquement, aux polices indépendantes. Il est possible de changer de pack de polices : il faut charger le package correspondant au pack choisi.

Les notes

Dans ce chapitre, les différents types de notes et de références possibles avec LaTeX seront abordés. C'est un chapitre facile, mais important, car il vous permettra de citer proprement tout ou une partie d'un texte, et de créer des notes en bas de certaines de vos pages.

Vous apprendrez à :

- citer des textes ;
- insérer des paragraphes de code brut dans vos documents ;
- créer des minipages ;
- composer des notes de bas de page ;
- utiliser les références internes au sein de votre publication.

Citation, code brut et URL

Cette première partie développe simplement les manières de créer des citations de texte, de code ainsi que la façon d'obtenir une URL cliquable.

Les citations simples

Fréquemment, vos rapports nécessiteront que vous citiez des propos. Pour vous permettre cela, deux environnements vous sont proposés : `quote` et `quotation`.

Ce code source vous montre la différence entre `quote` et `quotation` : le nombre de tabulations avant le premier mot (`quotation` est fait pour de gros volumes de texte, tandis que `quote` est plus adapté pour une ligne isolée).

Code : TeX

```

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Fusce condimentum tempor risus cursus aliquet.
\begin{quote}
Tout individu a droit à la vie, à la liberté et à la sûreté de sa
personne. \end{quote}
consectetur adipiscing elit.
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Nulla volutpat metus ut magna aliquam nec porttitor neque sodales.
\begin{quotation}
Tout individu a droit à la vie, à la liberté et à la sûreté de sa
personne. \end{quotation}
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Fusce condimentum tempor risus cursus aliquet.
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce condimentum
 tempor risus cursus aliquet. Nulla porttitor arcu in dolor tristique euismod. Cras
 arcu mi, adipiscing non ornare id, tristique ut ante.

Tout individu a droit à la vie, à la liberté et à la sûreté de sa personne.
 consectetur adipiscing elit. Lorem ipsum dolor sit amet, consectetur adipiscing
 elit. Nulla volutpat metus ut magna aliquam nec porttitor neque sodales.

Tout individu a droit à la vie, à la liberté et à la sûreté de sa
 personne.
 sit amet suscipit tortor pulvinar sed. Proin laoreet, orci iaculis ullamcorper
 pulvinar, lectus sapien accumsan nulla, ut condimentum orci diam sit amet
 justo.

les citations

simples

Les citations de code

Trois méthodes majeures permettent de citer des extraits de code.

La commande `\verb`

Pour insérer du code dans un paragraphe, le plus simple reste d'utiliser la commande `\verb`. Le texte que mettra en forme la commande `\textbackslashash{verb}` sera délimité par le caractère venant juste après son appel, généralement `|`.

Code : TeX

```
\verb| Mon bout de code {} |
```

Il se peut que vous utilisiez le caractère `|` dans certaines zones de votre code, et vous allez être confronté au problème que montrent le code source et le paragraphe ci-dessous (lisez-les attentivement).

Code : TeX

```
\verb[lapin & lapin {} [
\verb|lapin & lapin {} |
\verb(lapin & lapin {} (
\verb&lapin & lapin {} &
```

Les trois premières lignes de code donnent le même résultat tandis que la quatrième ne fonctionne pas ; le `&` est le délimiteur dans la quatrième ligne et, étant donné qu'il y a déjà un `&` dans le code, LaTeX comprendra mal l'instruction et n'encadrera que le premier mot, « lapin ». Cette subtilité rend assez risquée l'utilisation de `\verb` et peut parfois déclencher des dizaines d'erreurs ou de mauvais rendus.

L'environnement `verbatim(tab)`

L'environnement `verbatim`, accompagné du package du même nom, permet d'encadrer de gros volumes de code. Petit souci : il remplace les tabulations par des espaces.

Code : TeX

```
\documentclass{article}

\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[français]{babel}
\usepackage{verbatim}
\begin{document}

\begin{verbatim}
votre code
\end{verbatim}
\end{document}
```

Afin d'éviter ce désagrément et de choisir librement le nombre d'espaces correspondant à une tabulation, vous pouvez utiliser l'environnement `verbatimtab` fourni par le package supplémentaire `moreverb`.

Sa syntaxe est la suivante :

Code : TeX

```
\begin{verbatim}[nombre_d'espaces_par_tabulation]
votre code
\end{verbatim}
```

Dans l'exemple ci-dessous, chaque tabulation introduite dans le code se verra remplacée par 10 espaces une fois le document compilé.

Code : TeX

```
\documentclass{article}

\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[français]{babel}
\usepackage{verbatim}
\usepackage{moreverb}
\begin{document}

\begin{verbatim}[10]
votre code
\end{verbatim}
\end{document}
```

L'environnement `lstlisting`

L'environnement `lstlisting` permet de mettre en forme de façon colorée et d'utiliser de nombreuses options pour afficher du code.

Dans un premier temps, vous devez appeler le package `listings` dans l'en-tête du document, avant d'utiliser la commande `\lstset` et de la paramétrer pour qu'elle mette en forme votre code selon vos souhaits.

La commande `\lstset` se place avant `\begin{document}` et possède un grand nombre d'arguments paramétrables.

Code : TeX

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

Les langages compatibles avec la commande sont constamment mis à jour sur [la page de Wikibooks consacrée au package listings](#).

Jetez-y un coup d'œil à chaque fois que vous utiliserez ce package.

Le code à afficher s'insère dans votre document via l'environnement `lstlisting`.

Ci-dessous, un petit exemple du traitement d'un code C (extrait du tutoriel C du Site du Zéro) avec `lstlisting` (figure suivante).

Code : TeX

```
\documentclass{article}

\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[français]{babel}
\usepackage{listings}
\lstset{
  language=C,
  basicstyle=\footnotesize,
  numbers=left,
  numberstyle=\normalsize,
  numbersep=7pt,
}

\begin{document}

\begin{lstlisting}
if (age == 2)
{
  printf("Salut bebe !");
}
else if (age == 6)
{
  printf("Salut gamin !");
}
else if (age == 12)
{
  printf("Salut jeune !");
}
\end{lstlisting}

\end{document}
```

```
1  if (age == 2)
2  {
3    printf("Salut_bebe_!");
4  }
5  else if (age == 6)
6  {
7    printf("Salut_gamin_!");
8  }
9  else if (age == 12)
10 {
11   printf("Salut_jeune_!");
```

Citation de code

Gestion des URL

LaTeX gère les URL de façon très efficace, vous permettant de rendre vos documents interactifs.

Pour insérer une URL, il vous suffit d'utiliser le package `url` ainsi que la commande `\url{adresse}`, rien de plus compliqué que cela.

Minipage et texte encadré

Il peut parfois être utile de faire ressortir une portion de texte de façon singulière. L'environnement `minipage` et la commande `\fbox` vous permettent d'encadrer du texte et de le mettre en valeur.

Répetons-le une fois de plus, ce n'est pas parce que vous saurez faire des cadres après ce paragraphe qu'il faut en mettre partout.

Restez sobre, vos lecteurs vous en remercieront.

La commande `\fbox`

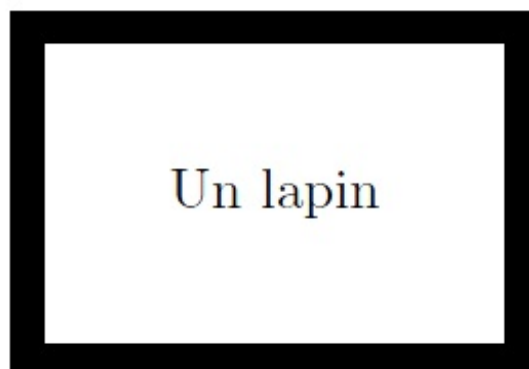
La commande `\fbox`, permettant d'encadrer des portions de texte, a déjà été vue dans le chapitre précédent.

Cependant, nous n'avons pas encore vu les paramètres qui peuvent lui être appliqués. `\fbox` est une commande puissante, pour laquelle il est possible de paramétrer des tas de choses. Néanmoins, dans ce cours, nous n'utiliserons `\fbox` que pour encadrer des textes.

Les deux paramètres qui nous intéressent ici sont l'écart entre le texte et la bordure ainsi que l'épaisseur de cette dernière. Ci-dessous une démonstration de `\fbox` et son résultat après compilation (figure suivante).

Code : TeX

```
% Commande permettant de définir l'écart
\setlength{\fboxsep}{8mm}
% Commande permettant de définir l'épaisseur du trait
\setlength{\fboxrule}{2mm}
\fbox{Un lapin}
```



Texte encadré

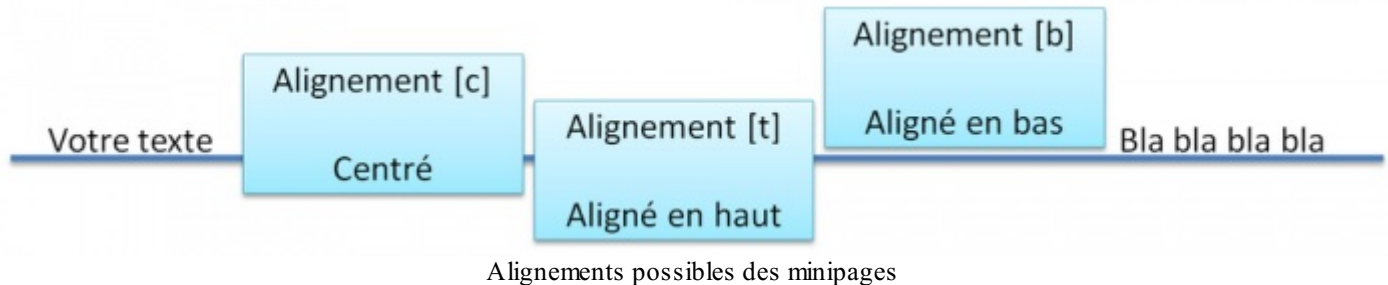
L'environnement minipage

Une minipage est un encart de texte de largeur choisie, en quelque sorte une nouvelle page dans votre page (Des utilisateurs avancés de LaTeX peuvent réussir à agencer les minipages de manière surprenante, tout comme les boîtes encadrées). À l'intérieur de cet encart de texte, vous pourrez disposer et utiliser des environnements comme si cette minipage était un document à part entière.

Une minipage est en quelque sorte un document dans un document. Il vous est même possible d'insérer une minipage dans une minipage (mais nous allons éviter de faire des choses qui ne servent à rien 🤪).

L'environnement `minipage` est dépendant de deux paramètres : **la largeur** et **l'alignement vertical** de la minipage à créer.

Pour ce qui est de la largeur, je pense que vous avez deviné à quoi servait le paramètre correspondant (Il s'exprime généralement en centimètres). Le paramètre d'alignement est quant à lui plus difficile à comprendre. L'alignement est vertical et défini par rapport au niveau du texte. Trois alignements sont proposés, et un schéma vaut mieux qu'un long discours pour visualiser les trois alignements disponibles.



Ici, une démonstration d'une minipage (volontairement mise en plein milieu d'un long texte) de 5 centimètres de large centrée verticalement (figure suivante) obtenue via la commande :

Code : TeX

```
\fbox{ %fbox est utilisé pour voir les bords de la minipage
\begin{minipage}[c]{5cm}
Un petit paragraphe.
\end{minipage}
}
```

Aenean eu tristique lectus. Aenean velit lorem, aliquam vitae mattis eget, imperdiet eu lorem. Aenean posuere sollicitudin hendrerit. Nullam sed adipiscing neque. Nulla sapien justo, feugiat et ultricies at, volutpat at nibh. Mauris placerat ornare urna, sit amet dapibus libero bibendum vitae. Nunc venenatis tristique augue non semper. Donec et ultricies ante. Nulla facilisi. Mauris eu urna

et augue dictum consequat in vitae nibh.

Suspendisse potenti. Sed quis nisi scelerisque eros posuere tempor. Aliquam euismod tristique odio, in congue diam condimentum vitae. Nulla facilisi.

Exemple de minipage

Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut libero risus, ultrices non imperdiet vel, sodales ut metus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Sed placerat mollis est, ut dapibus eros mattis at. Suspendisse nisi diam, rhoncus sit amet fringilla eu, ultricies sit amet nunc. Aenean sit amet mauris elit. Nunc ma-

Notes de bas de page, références internes

Cette partie traite des notes de bas de page ainsi que des références internes au sein des documents.

Ces deux éléments, souvent négligés, montrent à votre lectorat que votre texte est structuré et bien pensé.

Notes de bas de page

Il existe deux écoles pour l'insertion de notes de bas de page.

La commande `\footnote`

La première consiste à utiliser la commande `\footnote{Texte de la note.}` tout simplement à l'endroit où vous voulez insérer un renvoi vers une note de bas de page.

Cette commande est très intuitive, la seule chose à paramétrer ici est l'espacement entre l'exposant renvoyant à la note et le mot ou le groupe de mots concerné. Trois solutions sont possibles :

- coller l'exposant au mot concerné ;
- utiliser une espace (L'espace typographique est féminin), qui peut parfois être trop large ;
- insérer une espace plus fine via la commande `\espace`.

Code : TeX

```
un canard\footnote{bestiole qui fait coin}
un ornithorynque\footnote{bestiole qui fait rire}
un ours\footnote{bestiole qui fait mal}
```

un canard¹ un ornithorynque² un ours³

-
1. bestiole qui fait coin
 2. bestiole qui fait rire
 3. bestiole qui fait mal

Notes de bas de page (méthode

simple)

La commande `\footnotemark`

Ici, l'opération se déroule en deux temps. Dans un premier temps, vous marquez tous les éléments concernés par des notes de bas de page personnalisées via la commande `\footnotemark[n\degre_de_la_note]`, puis vous indiquerez les notes de bas de page correspondant aux numéros précédemment choisis grâce à la commande `\footnotetext[n\degre_de_la_note]{Texte}`.

Deux compilations sont nécessaires : une pour que LaTeX liste les marques, et une pour qu'il associe les notes de bas de page aux marques. Cette méthode est souvent salvatrice lorsque des erreurs apparaissent avec `\footnote`.

Ci-dessous, un code source de démonstration ainsi qu'une capture d'écran prise après deux compilations.

Code : TeX

```
un canard\footnotemark[1] \\
un ornithorynque\footnotemark[18] \\
un ours\footnotemark[32]

\footnotetext[1]{bestiole qui fait coin}
\footnotetext[18]{bestiole qui fait rire}
\footnotetext[32]{bestiole qui fait mal}
```

un canard¹
un ornithorynque¹⁸
un ours³²

Notes de bas de page (méthode avancée)

-
1. bestiole qui fait coin
 18. bestiole qui fait rire
 32. bestiole qui fait mal

Les références internes

LaTeX vous permet d'écrire des références internes de façon simple. Pour ce faire, trois commandes sont à connaître. La commande `\label{nom_choisi}` sert à marquer un endroit, et les commandes `\ref{nom_choisi}` et `\pageref{nom_choisi}` vous permettent d'appeler le numéro de page ou la référence de l'élément marqué dans une autre zone de votre document (figure suivante).

Démonstration :

Code : TeX

```
\documentclass{report}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[français]{babel}

\begin{document}

\part{Partie}
\chapter{Chapitre}
\section{Section}
\subsection{Une sous-section}
\label{patate}
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Donec nec condimentum libero. Phasellus condimentum porttitor
congue.
Morbi eget quam sed justo egestas lobortis. Aenean et erat metus.
\chapter{Chapitre}
\section{Section}
\subsection{Une sous-section}
Ici nous parlerons de la sous-section \ref{patate} vue à la page
\pageref{patate}.
\end{document}
```

2.1 Section

2.1.1 Une sous-section

Ici nous parlerons de la sous section 1.1.1 vue à la page 2 .

En résumé

- Les environnements `quote` et `quotation` permettent de citer du texte.
- On peut insérer du code source avec la commande `\verb` (dans un paragraphe), l'environnement `verbatim` (pour les codes plus longs) ou `lstlistings` (plus d'options). `lstlistings` offre de nombreuses options permettant notamment de colorer le code et de numérotter les lignes).
- Il est possible de créer des boîtes de texte encadrées en combinant la commande `\fbox` et l'environnement `minipage`.
- Les notes de bas de page sont généralement insérées via `\footnote` mais peuvent être agencées plus finement grâce à `\footnotemark` et `\footnotetext`.
- Des marqueurs invisibles (`\label`) peuvent être disposés un peu partout dans les publications et être utilisés afin de créer des références internes (Nous apprendrons un peu plus loin à insérer des marqueurs sur des images ou des tableau) avec `\ref` (indique le numéro de section où se trouve le marqueur) et `\pageref` (indique le numéro de page où se trouve le marqueur).

TP 1 – Mise en page de la Déclaration universelle des droits de l'homme

Il y a de ça huit chapitres, LaTeX envahissait votre vie. Vous avez appréhendé tous les concepts de base du langage, de la mise en page jusqu'à la hiérarchisation du contenu en passant par la personnalisation du texte et la production de citations élégantes. Toutes ces notions vous ont été enseignées de façon indépendante, mais il ne faut pas oublier la finalité de l'apprentissage de LaTeX : la création de documents réels.

Ce premier TP est l'occasion pour vous de tester votre savoir-faire et de vous entraîner. L'intégralité du contenu des chapitres vus jusqu'ici sera mise à profit et il y a fort à parier que vous aurez besoin d'aller les consulter pour répondre aux problématiques de l'exercice.

Comme vous avez dû vous en douter, le titre du chapitre ne laissant aucun suspense, je vous annonce que nous allons mettre en page une publication sur la Déclaration universelle des droits de l'homme et du citoyen. Texte fondateur de notre civilisation moderne, il vous permettra de conjuguer apprentissage et culture.

Les consignes

Dans ce TP, vous allez travailler avec trois textes différents, en utiliser certains morceaux choisis et les mettre en page de façon à en faire un petit livre de 17 pages. Le nombre peut faire peur, mais ne vous inquiétez pas, ce travail devrait vous prendre moins d'une heure et demie si vous avez suivi les précédents chapitres avec attention.

Le document produit à l'issue de ce TP est téléchargeable [ici](#). Il est sobre, mais élégant, de quoi donner le sourire à vos lecteurs.

Le but ici est de vous faire utiliser des packages ainsi que de vous faire revoir les chapitres de la deuxième partie. Suite à l'exercice pratique, de multiples pistes vous seront données pour vous permettre d'aller plus loin dans votre apprentissage de LaTeX.

Autre chose à noter : **ce TP clôt la deuxième partie du cours (et vous avez survécu jusqu'ici).**

La partie III, qui suit ce TP, développe les commandes avancées de LaTeX telles que celles qui permettent de manipuler les images, les tableaux, les notations mathématiques et les tables. Bien qu'accessibles, ces notions nécessitent d'être à l'aise avec les notions de commandes, d'environnements et de packages. L'importance de travailler ce TP ne s'en trouve que renforcée.

Faites chauffer votre clavier : les consignes vous attendent dans le paragraphe ci-dessous. Elles sont longues, mais comportent une grande quantité de texte à prendre depuis les articles originaux de Wikipédia.



Les mots en gras ne le sont pas pour faire joli, mais pour vous aider !

Les contraintes de l'exercice

Contenu hiérarchisé de notre document

Votre travail consistera ici à créer un **livre** en **français** intitulé « **Les droits de l'homme** ».

Ce livre utilisera des extraits de textes issus de quatre sources différentes :

- l'article « Droits de l'Homme » de Wikipédia ;
- l'article « Déclaration des droits de l'Homme et du Citoyen de 1789 » de Wikipédia ;
- l'article « Révolution française » de Wikipédia ;
- le texte descriptif de la licence Creative Commons 3.0.

Sur la **page de garde** seront écrits :

- le titre du livre ;
- vos **nom** et prénom ;
- la date du jour.

Un **chapitre** nommé « Introduction » sera composé en préambule **gros et en italique**.

Il contiendra le texte suivant :

Citation

Les droits de l'homme sont un concept selon lequel tout être humain possède des droits universels, inaliénables, quel que soit le droit positif en vigueur ou les autres facteurs locaux tels que l'ethnie, la nationalité, l'orientation sexuelle ou la religion.

Selon cette philosophie, combattue ou éclipsée aux 19^{ème} siècle, 20^{ème} siècle et 21^{ème} siècle par d'autres doctrines, l'homme, en tant que tel, et indépendamment de sa condition sociale, a des droits « inhérents à sa personne, inaliénables et sacrés », et donc opposables en toutes circonstances à la société et au pouvoir. Ainsi le concept de droits de l'homme est-il par définition universaliste et égalitaire, incompatible avec les systèmes et les régimes fondés sur la supériorité ou la « vocation historique » d'une caste, d'une race, d'un peuple, d'une classe ou d'un quelconque groupe social ; incompatible tout autant avec l'idée que la construction d'une société meilleure justifie l'élimination ou l'oppression de ceux qui sont censés faire obstacle à cette édification.

Les droits de l'homme, types de prérogatives dont sont titulaires les individus, sont généralement reconnus dans les pays occidentaux par la loi, par des normes de valeur constitutionnelle ou par des conventions internationales, afin que leur respect soit assuré, si besoin est même contre l'État. L'existence, la validité et le contenu des droits de l'homme sont un sujet permanent de débat en philosophie et en sciences politiques.

Le **corps du livre** sera composé d'une partie intitulée « Déclaration des droits de l'homme et du citoyen de 1789 » contenant **trois chapitres**. Le **premier chapitre**, intitulé « Histoire », contiendra le texte ci-dessous :

Citation

La Déclaration des Droits de l'Homme a été écrite dans un temple protestant. L'assemblée réunie à Versailles par la convocation des États généraux pour trouver une solution fiscale au déficit de l'État, se déclare Assemblée nationale en réunissant les trois ordres, dont elle décide l'abolition, puis s'institue Assemblée nationale constituante, et décide de rédiger une déclaration des principes fondamentaux à partir desquels sera établie une nouvelle Constitution. Elle se réunit pour cela, après avoir pris les décrets des 4 et 11 août 1789 sur la suppression des droits féodaux, qu'elle reprendra dans l'article premier de la Déclaration.

La Déclaration des Droits de l'Homme et du Citoyen a été débattue par l'Assemblée nationale française sous la présidence du marquis de Mirabeau à partir d'un des trois projets proposés : celui de 24 articles, rédigé par le VI^e bureau dirigé par Jérôme Champion de Cicé. L'attribution du texte primitif à La Fayette inspiré par la Déclaration d'indépendance des États-Unis est donc erronée. L'abbé Grégoire propose que la Déclaration des Droits de l'Homme et du Citoyen soit accompagnée de celle des devoirs.

La discussion débute le 9 juillet et débouche sur un vote le 26 août 1789, sous l'influence des leaders du tiers-état et de la noblesse. Le texte est peu modifié, mais est enrichi d'un préambule. Il est ratifié en partie le soir du 5 octobre 1789 par Louis XVI à Versailles, sur l'exigence de l'Assemblée, qui utilisa la pression d'une foule vindicative venue de Paris, initialement pour d'autres revendications.

Entièrement promulguée par le Roi à Paris, le 3 novembre 1789, la Déclaration des Droits est la dernière ordonnance royale. Elle servira de Préambule à la première Constitution de la Révolution française.

Le **deuxième chapitre**, nommé « Le texte », est plus complexe et hiérarchisé. Sa première **section**, intitulée « Introduction », contient l'extrait ci-dessous :

Citation

Les représentants du peuple français, constitués en Assemblée nationale, considérant que l'ignorance, l'oubli ou le mépris des droits de l'homme sont les seules causes des malheurs publics et de la corruption des gouvernements, ont résolu d'exposer, dans une déclaration solennelle, les droits naturels, inaliénables et sacrés de l'homme, afin que cette déclaration, constamment présente à tous les membres du corps social, leur rappelle sans cesse leurs droits et leurs devoirs ; afin que les actes du pouvoir législatif et ceux du pouvoir exécutif, pouvant être à chaque instant comparés avec le but de toute institution politique, en soient plus respectés ; afin que les réclamations des citoyens, fondées désormais sur des principes simples et incontestables, tournent toujours au maintien de la Constitution et au bonheur de tous.

Sa seconde **section**, intitulée « Les articles », contient 17 **paragraphes**. Le premier paragraphe a pour titre « Article premier », les suivants s'appellent « Article 2 », « Article 3 »... « Article 17 ». Ci-dessous le texte respectif de chaque article. Prenez le temps de le lire, ne serait-ce que par intérêt culturel.

Citation

Article premier - Les hommes naissent et demeurent libres et égaux en droits. Les distinctions sociales ne peuvent être fondées que sur l'utilité commune.

Article 2 - Le but de toute association politique est la conservation des droits naturels et imprescriptibles de l'homme. Ces droits sont la liberté, la propriété, la sûreté et la résistance à l'oppression.

Article 3 - Le principe de toute souveraineté réside essentiellement dans la Nation. Nul corps, nul individu ne peut exercer d'autorité qui n'en émane expressément.

Article 4 - La liberté consiste à pouvoir faire tout ce qui ne nuit pas à autrui : ainsi, l'exercice des droits naturels de chaque homme n'a de bornes que celles qui assurent aux autres membres de la société la jouissance de ces mêmes droits. Ces bornes ne peuvent être déterminées que par la loi.

Article 5 - La loi n'a le droit de défendre que les actions nuisibles à la société. Tout ce qui n'est pas défendu par la loi ne peut être empêché, et nul ne peut être contraint à faire ce qu'elle n'ordonne pas.

Article 6 - La loi est l'expression de la volonté générale. Tous les citoyens ont droit de concourir personnellement ou par leurs représentants à sa formation. Elle doit être la même pour tous, soit qu'elle protège, soit qu'elle punisse. Tous les citoyens, étant égaux à ses yeux, sont également admissibles à toutes dignités, places et emplois publics, selon leur capacité et sans autre distinction que celle de leurs vertus et de leurs talents.

Article 7 - Nul homme ne peut être accusé, arrêté ou détenu que dans les cas déterminés par la loi et selon les formes qu'elle a prescrites. Ceux qui sollicitent, expédient, exécutent ou font exécuter des ordres arbitraires doivent être punis ; mais tout citoyen appelé ou saisi en vertu de la loi doit obéir à l'instant ; il se rend coupable par la résistance.

Article 8 - La loi ne doit établir que des peines strictement et évidemment nécessaires, et nul ne peut être puni qu'en vertu d'une loi établie et promulguée antérieurement au délit, et légalement appliquée.

Article 9 - Tout homme étant présumé innocent jusqu'à ce qu'il ait été déclaré coupable, s'il est jugé indispensable de l'arrêter, toute rigueur qui ne serait pas nécessaire pour s'assurer de sa personne doit être sévèrement réprimée par la loi.

Article 10 - Nul ne doit être inquiété pour ses opinions, mêmes religieuses, pourvu que leur manifestation ne trouble pas l'ordre public établi par la loi.

Article 11 - La libre communication des pensées et des opinions est un des droits les plus précieux de l'homme ; tout citoyen peut donc parler, écrire, imprimer librement, sauf à répondre de l'abus de cette liberté dans les cas déterminés par la loi.

Article 12 - La garantie des droits de l'homme et du citoyen nécessite une force publique ; cette force est donc instituée pour l'avantage de tous, et non pour l'utilité particulière de ceux à qui elle est confiée.

Article 13 - Pour l'entretien de la force publique, et pour les dépenses d'administration, une contribution commune est indispensable ; elle doit être également répartie entre les citoyens, en raison de leurs facultés.

Article 14 - Les citoyens ont le droit de constater, par eux-mêmes ou par leurs représentants, la nécessité de la contribution publique, de la consentir librement, d'en suivre l'emploi, et d'en déterminer la quotité, l'assiette, le recouvrement et la durée.

Article 15 - La société a le droit de demander compte à tout agent public de son administration.

Article 16 - Toute société dans laquelle la garantie des droits n'est pas assurée ni la séparation des pouvoirs déterminée, n'a point de Constitution.

Article 17 - La propriété étant un droit inviolable et sacré, nul ne peut en être privé, si ce n'est lorsque la nécessité publique, légalement constatée, l'exige évidemment, et sous la condition d'une juste et préalable indemnité.

Le **troisième chapitre**, intitulé « Sources », contient le texte suivant.

Citation

La question des sources de la Déclaration française a suscité une controverse empreinte de nationalisme au sein de l'historiographie. Dans une brochure de 1895, l'historien allemand Georg Jellinek présentait l'œuvre française comme une simple héritière des Déclarations anglo-saxonnes (Pétition des droits, Déclaration des droits), elles-mêmes inspirées du Protestantisme luthérien. Traduite en français en 1902, dans un contexte de montée des tensions entre la France et

l'Allemagne, elle donnera lieu à une réplique aussi peu nuancée, portée par Emile Boutmy : la Déclaration des Droits de l'Homme et du Citoyen n'aurait de source que dans la tradition philosophique et humaniste des Lumières.

Le Préambule, ajouté au projet, a été rédigé sous l'influence de Mirabeau et de Jean-Joseph Mounier, député du Tiers qui avait fait adopter le serment du Jeu de Paume, tous deux monarchiens, c'est-à-dire partisans d'une Monarchie constitutionnelle à l'anglaise.

L'invocation à « l'Être suprême » a été rajoutée au cours des séances pour tenir compte des convictions chrétiennes de presque tous les citoyens.

Le texte de l'Article premier, « Tous les Hommes naissent et demeurent libres et égaux en droits », synthétise la Loi du 4 août 1789 abolissant la société d'ordres hiérarchisés.

L'article 16, associant constitution et organisation de la séparation des pouvoirs, est un principe antérieurement admis avec la séparation des ordres spirituel, politique et économique. Mais les trois pouvoirs politiques auxquels renvoie implicitement cet article, à savoir le législatif, l'exécutif et le judiciaire, est la conception proposée par Montesquieu depuis 1748 dans *De l'Esprit des Lois*.

L'article 3, qui attribue la souveraineté à la Nation, s'inspire des thèmes des remontrances des Parlements, portées par les nombreux membres du club des Amis de la Constitution, plus connu sous le nom de Club des Jacobins, mais aussi du célèbre pamphlet de l'abbé Sieyès, qui pose l'équation : peuple = Tiers-État, c'est-à-dire que les députés du Tiers-État sont les représentants légitimes de l'ensemble du peuple.

L'article 6, directement inspiré du philosophe Rousseau, a été proposé par Talleyrand. Lu à la tribune du comité de constitution le 12 septembre 1789, ce qui deviendra l'article 6 de la Déclaration des droits prenait la forme suivante : « La loi étant l'expression de la volonté générale, tous les citoyens ont droit de concourir personnellement ou par représentation à sa formation ; elle doit être la même pour tous ».

Les autres articles reprennent des principes généraux du droit ou de la procédure qui sont déjà établis, comme la positivité du droit, le caractère contradictoire des procédures, la non-rétroactivité des lois, etc.

Son idéal est l'individualisme libéral. C'est une œuvre de circonstance, une proclamation générale, un texte tourné vers le passé avec pour objectif d'en finir avec l'Ancien Régime ; mais également un texte tourné vers l'avenir en promouvant la philosophie des Lumières et son idéal rationaliste.

L'annexe A, « La révolution française » nous raconte une belle histoire via ce texte :

Citation

La Révolution française est la période de l'histoire de France comprise entre la convocation des États généraux en 1789 et le coup d'État du 18 brumaire (9-10 novembre 1799) de Napoléon Bonaparte. C'est un moment fondamental de l'histoire de France, marquant la fin de l'Ancien Régime, et le passage à une monarchie constitutionnelle puis à la Première République. Elle a mis fin à la royauté, à la société d'ordres et aux privilèges. Justifiée par la Déclaration des Droits de l'Homme et du Citoyen, qui proclamait l'égalité théorique des citoyens devant la loi, les libertés fondamentales et la souveraineté de la Nation, apte à se gouverner au travers des représentants élus, cette période causa la mort de plusieurs milliers de personnes et la terreur pour la majorité.

Le chapitre épilogue sera nommé « Sources et licences », il contiendra **deux sections non numérotées**. La première section (« Sources ») contiendra une **liste de description**. La seconde (« Licence Creative Commons 3.0 ») comportera deux paragraphes. **Les mentions « Paternité » et « Partage des conditions initiales à l'identique » devront être signalées comme importantes.**

La mise en forme finale de la page devra être similaire à celle de la figure suivante, avec bien sûr les adresses internet réelles et cliquables.

Sources et licences

Sources

Droits de l'homme (Wikipedia) : `http://adresse.com`

Déclaration de 1789 (Wikipedia) : `http://adresse.com`

Déclaration de 1793 (Wikipedia) : `http://adresse.com`

Déclaration de 1795 (Wikipedia) : `http://adresse.com`

La révolution française (Wikipedia) : `http://adresse.com`

Mise en forme

Licence Creative Commons 3.0

Vous êtes libres :

- de reproduire, distribuer et communiquer cette création au public
- de modifier cette création

Selon les conditions suivantes :

- *Paternité* : vous devez citer le nom de l'auteur original de la manière indiquée par l'auteur de l'œuvre ou le titulaire des droits qui vous confère cette autorisation (mais pas d'une manière qui suggérerait qu'ils vous soutiennent ou approuvent votre utilisation de l'œuvre).
- *Partage des conditions initiales à l'identique* : si vous transformez ou modifiez cette œuvre pour en créer une nouvelle, vous devez la distribuer selon les termes du même contrat ou avec une licence similaire ou compatible.

attendue

Consignes additionnelles et pistes

- Le pack de police utilisé sera `lmodern`.
- Les noms propres seront écrits en petites capitales (convention française).
- Les en-têtes et pieds de pages seront de type `headings` (réglage par défaut pour la classe `book`).
- La marge de droite sera de 3 cm.
- La marge de gauche sera de 6 cm.
- La marge du bas sera de 5 cm.
- La marge du haut sera de 5 cm.
- Les adresses internet ne devront pas tenir sur plus d'une ligne, mais vous avez le droit d'utiliser un réducteur d'URL comme `bit.ly` qui créera une adresse courte à partir d'une adresse initiale.

Les consignes s'arrêtent (enfin) ici !

Afin de travailler dans de meilleures conditions, je vous conseille de compiler votre code très fréquemment pour régler les problèmes un par un. Quelques petits pièges se trouvent enfouis dans les textes de ce TP, à vous de les déjouer.

Tout est faisable avec vos connaissances actuelles, il suffit que vous preniez votre temps.

Au fait ! Ce tableau de caractères spéciaux devrait vous être utile...

Accents et signes											
Ó	\'{o}	Ö	\. {o}	Œ	\b{o}	ò	_{o}	ø	\o	ö	\" {o}
Ö	\u{o}	ō	\={o}	Œ	\d{o}	õ	\~{o}	ô	\^{o}	ő	\H{o}
å	\aa	š	\v s	‡	\ddag	†	\dag	ôô	\t{oo}	ß	\ss
Ø	\o	æ	\ae	Æ	\AE	Œ	\c{o}	l	\i	¶	\P
©	\copyright	œ	\oe	Œ	\OE	§	\d s	š	\r s	Š	\H s
J	\j	Å	\AA	Œ	\t s	§	\s	£	\pounds		



Dans l'encodage choisi, les apostrophes penchées provoquent des erreurs. À vous de trouver une solution...

Que la force soit avec vous !

La correction

L'heure est venue de corriger tout votre travail, de regarder les points qui ont pu poser des problèmes. Notez que chaque étape fait l'objet d'au moins une compilation dans le but de pouvoir corriger les erreurs pas à pas.

Étape 1 : le type de document

Le type de document a une importance prépondérante pour nous, il définit un squelette de code source. La classe `book` n'a pas été choisie par hasard ; souvenez-vous, nous avons traité ensemble les problématiques de la création d'annexes et de préambules.

La première étape du travail était de mettre en forme ce squelette, qui devait ressembler à peu de choses près à celui-ci :

Code : TeX

```
\documentclass{book}

\begin{document}

  \frontmatter
  \mainmatter
  \appendix
  \backmatter

\end{document}
```

Étape 2 : la langue

La consigne disait d'écrire le texte en français et il était visible que nombre d'accents allaient apparaître. Les trois packages habituels de gestion de caractères étaient donc indispensables. Il fallait les insérer dans le préambule du document, comme vous l'avez appris dans le chapitre traitant des packages.

Code : TeX

```
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[français]{babel}
```

Étape 3 : les autres packages

Nous venons d'appeler les packages nous permettant d'écrire en français, il est de bon ton de survoler les consignes pour comprendre quels packages nous seront utiles pour continuer l'exercice. Ici, trois autres packages étaient nécessaires : `url`, `geometry` et `lmodern`.

Code : TeX

```
\usepackage{url} % Pour écrire des adresses cliquables.
\usepackage{lmodern} % Pour changer le pack de police.
\usepackage[top=5cm, bottom=5cm, left=6cm, right=3cm]{geometry} %
Les marges.
```

Étape 4 : la page de garde

La création de page de garde a été abordée dans les chapitres de mise en page. Celle demandée dans l'exercice est simple, elle comporte les trois mentions classiques : titre, nom des auteurs (les noms sont en petites capitales) et date du jour. Les trois paramètres sont définis dans le préambule et la commande `\maketitle` est insérée après `\begin{document}` afin de créer la page de garde (figure suivante).

Code : TeX

```
\title{Les droits de l'homme}
\author{\textsc{Laleloulilo} - \textsc{Zozor}}
\date{\today} % Pour mettre la date du jour, tapez \today.

\begin{document}

\maketitle % Page de garde.
```

Les droits de l'homme

LALELOULILO - ZOZOR

6 juillet 2010

Page de garde

Étape 5 : le squelette complet

Les packages sont listés, le travail préparatoire est presque terminé. Il nous reste les éléments de structure secondaires à lister (chapitres, sections...). Certains n'étant pas numérotés, nous n'oublions pas le symbole *.

Code : TeX

```
\documentclass{book}

\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[français]{babel}
\usepackage{url} % Pour écrire des adresses cliquables.
\usepackage{lmodern} % Pour changer le pack de police.
\usepackage[top=5cm, bottom=5cm, left=6cm, right=3cm]{geometry} %
Les marges.

\title{Les droits de l'homme}
\author{\textsc{Laleloulilo} - \textsc{Zozor}}
\date{\today} % Pour mettre la date du jour, tapez \today.

\begin{document}

\maketitle % Page de garde.

\frontmatter

\chapter{Introduction}

\mainmatter

\part{Déclaration des Droits de l'Homme et du Citoyen de 1789}

\chapter{Histoire}
\chapter{Le texte}

\section{Introduction}
\section{Les articles}

\paragraph{Article premier}
\paragraph{Article 2}
\paragraph{Article 3}
\paragraph{Article 4}
\paragraph{Article 5}
\paragraph{Article 6}
\paragraph{Article 7}
\paragraph{Article 8}
\paragraph{Article 9}
\paragraph{Article 10}
\paragraph{Article 11}
\paragraph{Article 12}
\paragraph{Article 13}
\paragraph{Article 14}
\paragraph{Article 15}
\paragraph{Article 16}
\paragraph{Article 17}

\chapter{Sources}

\appendix

\chapter{La révolution française}

\backmatter

\chapter{Sources et licences}

\section*{Sources}

\begin{description}
```

```

\item
\item
\item
\item
\item
\end{description}

\section*{Licence Creative Commons 3.0}

\paragraph{Vous êtes libres :}

\begin{itemize}
\item
\item
\end{itemize}

\paragraph{Selon les conditions suivantes :}

\begin{itemize}
\item \emph{Paternité} :
\item \emph{Partage des conditions initiales à l'identique} :
\end{itemize}

\end{document}

```

Si vous avez réussi à aller jusqu'ici, c'est déjà très bien. Félicitations, vous commencez à vous sentir à l'aise avec LaTeX et ses concepts fondamentaux !

Étape 6 : le contenu

Les choses sérieuses commencent. Ici se trouvent les plus grosses sources d'erreurs d'étourderie. Chaque chapitre en est rempli l'un après l'autre. Ne pas oublier d'insérer les commandes de mise en exposant, de création de noms en petites capitales, d'introduction de guillemets français ainsi que de remplacer par la commande `\oe` le caractère spécial `œ` présent un peu partout dans les paragraphes.

Autre piège, certaines apostrophes sont penchées et font planter LaTeX (mais seulement si vous aviez tapé votre code en `latin1`). Il fallait que vous vous en rendiez compte et que vous les remplaciez par des apostrophes droites (Ce genre de problème arrive de façon extrêmement fréquente au quotidien).

Le chapitre d'introduction

Ce chapitre contenait des apostrophes penchées, des exposants ainsi que des guillemets français. De plus, il était souhaité qu'il soit gros et en italique (figures suivante et suivante).

Code : TeX

```

\begin{it}
\Large{
  Les droits de l'homme sont un concept selon lequel tout être
  humain possède des droits universels, inaliénables, quel que soit le
  droit positif en vigueur ou les autres facteurs locaux tels que
  l'ethnie, la nationalité, l'orientation sexuelle ou la religion.

  Selon cette philosophie, combattue ou éclipsée aux
  19\textsuperscript{ième} siècle, 20\textsuperscript{ième} siècle et
  21\textsuperscript{ième} siècle par d'autres doctrines, l'homme, en
  tant que tel, et indépendamment de sa condition sociale, a des
  droits \og inhérents à sa personne, inaliénables et sacrés \fg, et
  donc opposables en toutes circonstances à la société et au pouvoir.
  Ainsi le concept de droits de l'homme est-il par définition
  universaliste et égalitaire, incompatible avec les systèmes et les
  régimes fondés sur la supériorité ou la \og vocation historique \fg

```

d'une caste, d'une race, d'un peuple, d'une classe ou d'un quelconque groupe social ; incompatible tout autant avec l'idée que la construction d'une société meilleure justifie l'élimination ou l'oppression de ceux qui sont censés faire obstacle à cette édification.

Les droits de l'homme, types de prérogatives dont sont titulaires les individus, sont généralement reconnus dans les pays occidentaux par la loi, par des normes de valeur constitutionnelle ou par des conventions internationales, afin que leur respect soit assuré, si besoin est même contre l'état. L'existence, la validité et le contenu des droits de l'homme sont un sujet permanent de débat en philosophie et en sciences politiques.

```
}  
\end{it}
```


Introduction

Les droits de l'homme sont un concept selon lequel tout être humain possède des droits universels, inaliénables, quel que soit le droit positif en vigueur ou les autres facteurs locaux tels que l'ethnie, la nationalité, l'orientation sexuelle ou la religion.

Selon cette philosophie, combattue ou éclipsée aux 19^{ième} siècle, 20^{ième} siècle et 21^{ième} siècle par d'autres doctrines, l'homme, en tant que tel, et indépendamment de sa condition sociale, a des droits « inhérents à sa personne, inaliénables et sacrés », et donc opposables en toutes circonstances à la société et au pouvoir. Ainsi le concept de droits de l'homme est-il par définition universaliste et égalitaire, incompatible avec les systèmes et les régimes fondés sur la supériorité ou la « vocation historique » d'une caste, d'une race, d'un peuple, d'une classe ou d'un quelconque groupe social ; incompatible tout autant avec l'idée que la construction d'une société meilleure justifie l'élimination ou l'oppression de ceux qui sont censés faire obstacle à cette édification.

Les droits de l'homme, types de prérogatives dont sont titulaires les individus, sont généralement reconnus dans



Chapitre d'introduction (la suite)

Les chapitres « Histoire », « Le texte », « Sources » et l'annexe « La Révolution française »

Rien de transcendant, les commandes utilisées ici l'ont déjà été dans le paragraphe précédent. La clé du succès était juste de prendre votre temps et de compiler souvent afin de mieux voir vos erreurs (figures suivante à suivante).

Code : TeX

```
\part{Déclaration des droits de l'homme et du citoyen de 1789}
\chapter{Histoire}
```

La Déclaration des Droits de l'Homme a été écrite dans un temple protestant. L'assemblée réunie à Versailles par la convocation des États généraux pour trouver une solution fiscale au déficit de l'État, se déclare Assemblée nationale en réunissant les trois ordres, dont elle décide l'abolition, puis s'institue Assemblée nationale constituante, et décide de rédiger une déclaration des principes fondamentaux à partir desquels sera établie une nouvelle Constitution. Elle se réunit pour cela, après avoir pris les décrets des 4 et 11 août 1789 sur la suppression des droits féodaux, qu'elle reprendra dans l'article premier de la Déclaration.

La Déclaration des droits de l'Homme et du Citoyen a été débattue par l'Assemblée nationale française sous la présidence du marquis de `\textsc{Mirabeau}` à partir d'un des trois projets proposés, celui de 24 articles rédigé par le VI^e bureau, dirigé par Jérôme `\textsc{Champion de Cicé}`. L'attribution du texte primitif à `\textsc{La Fayette}` inspiré par la Déclaration d'indépendance des États-Unis est donc erronée. L'abbé Grégoire propose que la Déclaration des droits de l'Homme et du Citoyen soit accompagnée de celle des devoirs.

La discussion débute le 9 juillet et débouche sur un vote le 26 août 1789, sous l'influence des leaders du tiers-état et de la noblesse. Le texte est peu modifié, mais est enrichi d'un préambule. Il est ratifié en partie le soir du 5 octobre 1789 par Louis XVI à Versailles, sur l'exigence de l'Assemblée, qui utilisa la pression d'une foule vindicative venue de Paris, initialement pour d'autres revendications.

Entièrement promulguée par le Roi à Paris, le 3 novembre 1789, la Déclaration des Droits est la dernière ordonnance royale. Elle

servira de Préambule à la première Constitution de la Révolution française.

```
\chapter{Le texte}
\section{Introduction}
```

Les représentants du peuple français, constitués en Assemblée nationale, considérant que l'ignorance, l'oubli ou le mépris des droits de l'homme sont les seules causes des malheurs publics et de la corruption des gouvernements, ont résolu d'exposer, dans une déclaration solennelle, les droits naturels, inaliénables et sacrés de l'homme, afin que cette déclaration, constamment présente à tous les membres du corps social, leur rappelle sans cesse leurs droits et leurs devoirs ; afin que les actes du pouvoir législatif et ceux du pouvoir exécutif, pouvant être à chaque instant comparés avec le but de toute institution politique, en soient plus respectés ; afin que les réclamations des citoyens, fondées désormais sur des principes simples et incontestables, tournent toujours au maintien de la Constitution et au bonheur de tous.

```
\section{Les articles}
```

```
\paragraph{Article premier}
```

Les hommes naissent et demeurent libres et égaux en droits. Les distinctions sociales ne peuvent être fondées que sur l'utilité commune.

```
\paragraph{Article 2}
```

Les hommes naissent et demeurent libres et égaux en droits. Les distinctions sociales ne peuvent être fondées que sur l'utilité commune.

```
\paragraph{Article 3}
```

Le principe de toute souveraineté réside essentiellement dans la Nation. Nul corps, nul individu ne peut exercer d'autorité qui n'en émane expressément.

```
\paragraph{Article 4}
```

La liberté consiste à pouvoir faire tout ce qui ne nuit pas à autrui : ainsi, l'exercice des droits naturels de chaque homme n'a de bornes que celles qui assurent aux autres membres de la société la jouissance de ces mêmes droits. Ces bornes ne peuvent être déterminées que par la loi.

```
\paragraph{Article 5}
```

La loi n'a le droit de défendre que les actions nuisibles à la société. Tout ce qui n'est pas défendu par la loi ne peut être empêché, et nul ne peut être contraint à faire ce qu'elle n'ordonne pas.

```
\paragraph{Article 6}
```

La loi est l'expression de la volonté générale. Tous les citoyens ont droit de concourir personnellement ou par leurs représentants à sa formation. Elle doit être la même pour tous, soit qu'elle protège, soit qu'elle punisse. Tous les citoyens, étant égaux à ces yeux, sont également admissibles à toutes dignités, places et emplois publics, selon leur capacité et sans autre distinction que celle de leurs vertus et de leurs talents.

```
\paragraph{Article 7}
```

La loi est l'expression de la volonté générale. Tous les citoyens ont droit de concourir personnellement ou par leurs représentants à

sa formation. Elle doit être la même pour tous, soit qu'elle protège, soit qu'elle punisse. Tous les citoyens, étant égaux à ces yeux, sont également admissibles à toutes dignités, places et emplois publics, selon leur capacité et sans autre distinction que celle de leurs vertus et de leurs talents.

`\paragraph{Article 8}`

La loi ne doit établir que des peines strictement et évidemment nécessaires, et nul ne peut être puni qu'en vertu d'une loi établie et promulguée antérieurement au délit, et légalement appliquée.

`\paragraph{Article 9}`

Tout homme étant présumé innocent jusqu'à ce qu'il ait été déclaré coupable, s'il est jugé indispensable de l'arrêter, toute rigueur qui ne serait pas nécessaire pour s'assurer de sa personne doit être sévèrement réprimée par la loi.

`\paragraph{Article 10}`

Nul ne doit être inquiété pour ses opinions, mêmes religieuses, pourvu que leur manifestation ne trouble pas l'ordre public établi par la loi.

`\paragraph{Article 11}`

La libre communication des pensées et des opinions est un des droits les plus précieux de l'homme ; tout citoyen peut donc parler, écrire, imprimer librement, sauf à répondre de l'abus de cette liberté dans les cas déterminés par la loi.

`\paragraph{Article 12}`

La garantie des droits de l'homme et du citoyen nécessite une force publique ; cette force est donc instituée pour l'avantage de tous, et non pour l'utilité particulière de ceux à qui elle est confiée.

`\paragraph{Article 13}`

Pour l'entretien de la force publique, et pour les dépenses d'administration, une contribution commune est indispensable ; elle doit être également répartie entre les citoyens, en raison de leurs facultés.

`\paragraph{Article 14}`

Les citoyens ont le droit de constater, par eux-mêmes ou par leurs représentants, la nécessité de la contribution publique, de la consentir librement, d'en suivre l'emploi, et d'en déterminer la quotité, l'assiette, le recouvrement et la durée.

`\paragraph{Article 15}`

La société a le droit de demander compte à tout agent public de son administration.

`\paragraph{Article 16}`

Toute société dans laquelle la garantie des droits n'est pas assurée ni la séparation des pouvoirs déterminée, n'a point de Constitution.

`\paragraph{Article 17}`

La propriété étant un droit inviolable et sacré, nul ne peut en être privé, si ce n'est lorsque la nécessité publique, légalement constatée, l'exige évidemment, et sous la condition d'une juste et préalable indemnité.

`\chapter{Sources}`

La question des sources de la Déclaration française a suscité une controverse empreinte de nationalisme au sein de l'historiographie. Dans une brochure de 1895, l'historien allemand Georg `\textsc{Jellinek}` présentait l'`\oe`uvre française comme une simple héritière des Déclarations anglo-saxonnes (Pétition des droits, Déclaration des droits), elles-mêmes inspirées du Protestantisme luthérien. Traduite en français en 1902, dans un contexte de montée des tensions entre France et Allemagne, elle donnera lieu à une réplique aussi peu nuancée, portée par Emile `\textsc{Boutmy}` : la Déclaration des droits de l'homme et du citoyen n'aurait de source que dans la tradition philosophique et humaniste des Lumières.

Le Préambule, ajouté au projet, a été rédigé sous l'influence de `\textsc{Mirabeau}`, et de Jean-Joseph `\textsc{Mounier}`, député du Tiers qui avait fait adopter le serment du Jeu de Paume, tous deux monarchiens, c'est-à-dire partisans d'une Monarchie constitutionnelle à l'anglaise.

L'invocation à l'`\og` Être suprême `\fg` a été rajoutée au cours des séances pour tenir compte des convictions chrétiennes de presque tous les citoyens.

Le texte de l'article Un, `\og` Tous les Hommes naissent et demeurent libres et égaux en droit `\fg`, synthétise la Loi du 4 août 1789 abolissant la société d'ordres hiérarchisés.

L'article 16, associant constitution et organisation de la séparation des pouvoirs, est un principe antérieurement admis avec la séparation des ordres spirituel, politique et économique. Mais les trois pouvoirs politiques auxquels renvoie implicitement cet article, à savoir le législatif, l'exécutif et le judiciaire, est la conception proposée par Montesquieu depuis 1748 dans `\emph{De l'Esprit des Lois}`.

L'article 3, qui attribue la souveraineté à la Nation, s'inspire des thèmes des remontrances des Parlements, portées par les nombreux membres du club des Amis de la Constitution, plus connu sous le nom de Club des Jacobins, mais aussi du célèbre pamphlet de l'abbé `\textsc{Sieyès}`, qui pose l'équation : peuple = Tiers-État, c'est-à-dire que les députés du Tiers-État sont les représentants légitimes de l'ensemble du peuple.

L'article 6, directement inspiré du philosophe `\textsc{Rousseau}`, a été proposé par `\textsc{Talleyrand}`. Lu à la tribune du comité de constitution le 12 septembre 1789, ce qui deviendra l'article 6 de la déclaration des droits prenait la forme suivante : `\og` La loi étant l'expression de la volonté générale, tous les citoyens ont droit de concourir personnellement ou par représentation à sa formation ; elle doit être la même pour tous `\fg`.

Les autres articles reprennent des principes généraux du droit ou de la procédure qui sont déjà établis, comme la positivité du droit, le caractère contradictoire des procédures, la non rétroactivité des lois, etc.

Son idéal est l'individualisme libéral. C'est une `\oe`uvre de circonstance, une proclamation générale, un texte tourné vers le passé avec pour objectif d'en finir avec l'Ancien Régime ; mais également un texte tourné vers l'avenir en promouvant la philosophie des lumières et son idéal rationaliste.

`\appendix`

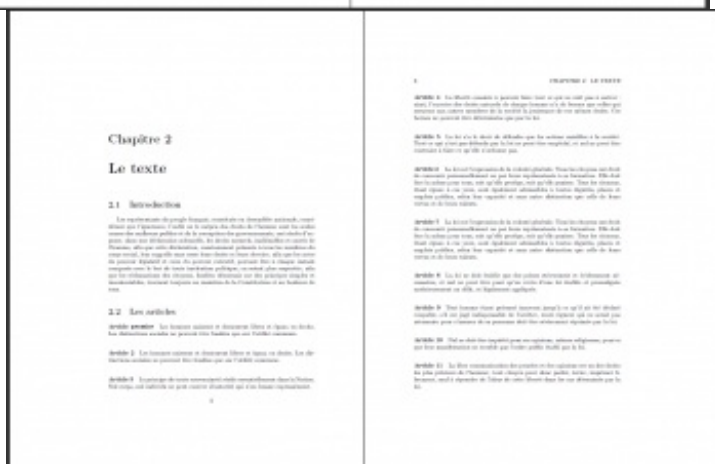
`\chapter{La révolution française}`

La Révolution française est la période de l'histoire de France comprise entre la convocation des États généraux en 1789 et le coup d'État du 18 brumaire (9-10 novembre 1799) de Napoléon

`\textsc{Bonaparte}`. C'est un moment fondamental de l'histoire de France, marquant la fin de l'Ancien Régime, et le passage à une monarchie constitutionnelle puis à la Première République. Elle a mis fin à la royauté, à la société d'ordres et aux privilèges. Justifié par la Déclaration des droits de l'homme et du citoyen, qui proclamait l'égalité théorique des citoyens devant la loi, les libertés fondamentales et la souveraineté de la Nation, apte à se gouverner au travers des représentants élus, cette période causa la mort de plusieurs milliers de personnes et la terreur pour la majorité.



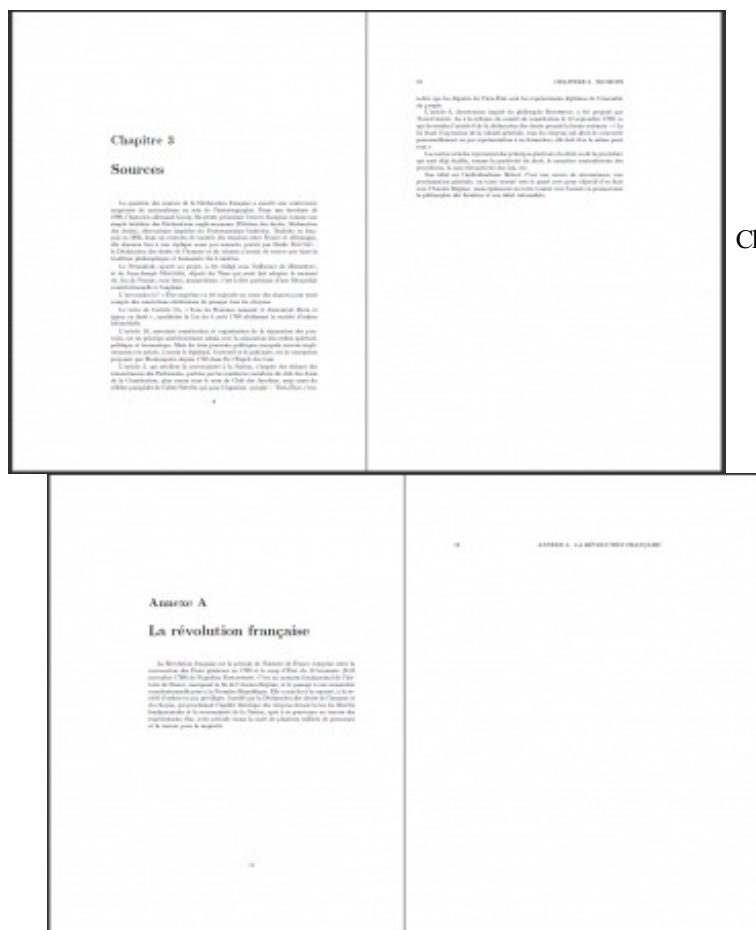
Chapitre Histoire



Chapitre Texte



Chapitre Texte (bis)



Chapitre Sources

Annexe A

Le chapitre épilogue « Sources et Licences »

Pour ce chapitre, il fallait dans un premier temps utiliser une liste de description ainsi que la commande `\url` accompagnée d'URL réduites via `bit.ly` (C'est un célèbre réducteur d'URL). Dans un second temps, deux listes à puces toutes simples faisaient leur apparition. Dans l'une d'elles, la commande `\emph` sera manipulée pour signifier l'importance des termes « Paternité » et « Partage des conditions initiales à l'identique » (figure suivante).

Code : TeX

```
\chapter{Sources et licences}

\section*{Sources}

\begin{description}

\item[Droits de l'homme (Wikipedia):] \url{http://bit.ly/9nYoSU}
\item[Déclaration de 1789 (Wikipedia):]
\url{http://bit.ly/1DJqQg}
\item[Déclaration de 1793 (Wikipedia):] \url{http://bit.ly/SeRyG}
\item[Déclaration de 1795 (Wikipedia):]
\url{http://bit.ly/b6sdRI}
\item[La révolution française (Wikipedia):]
\url{http://bit.ly/cUd3SF}

\end{description}

\section*{Licence Creative Commons 3.0}

\paragraph{Vous êtes libres :}
```



```

\begin{itemize}

\item de reproduire, distribuer et communiquer cette création
au public
\item de modifier cette création

\end{itemize}

\paragraph{Selon les conditions suivantes :}

\begin{itemize}

\item \emph{Paternité} : vous devez citer le nom de l'auteur
original de la manière indiquée par l'auteur de l'\oe uvre ou le
titulaire des droits qui vous confère cette autorisation (mais pas
d'une manière qui suggérerait qu'ils vous soutiennent ou approuvent
votre utilisation de l'\oe uvre).
\item \emph{Partage des conditions initiales à l'identique} :
si vous transformez ou modifiez cette oeuvre pour en créer une
nouvelle, vous devez la distribuer selon les termes du même contrat
ou avec une licence similaire ou compatible.

\end{itemize}

```

Sources et licences

Sources

Droits de l'homme (Wikipedia) : <http://bit.ly/9nYoSU>

Déclaration de 1789 (Wikipedia) : <http://bit.ly/1DJqQg>

Déclaration de 1793 (Wikipedia) : <http://bit.ly/SeRyG>

Déclaration de 1795 (Wikipedia) : <http://bit.ly/b6sdRI>

La révolution française (Wikipedia) : <http://bit.ly/cUd3SF>

Sources

Licence Creative Commons 3.0

Vous êtes libres :

- de reproduire, distribuer et communiquer cette création au public
- de modifier cette création

Selon les conditions suivantes :

- *Paternité* : vous devez citer le nom de l'auteur original de la manière indiquée par l'auteur de l'œuvre ou le titulaire des droits qui vous confère cette autorisation (mais pas d'une manière qui suggérerait qu'ils vous soutiennent ou approuvent votre utilisation de l'œuvre).
- *Partage des conditions initiales à l'identique* : si vous transformez ou modifiez cette oeuvre pour en créer une nouvelle, vous devez la distribuer selon les termes du même contrat ou avec une licence similaire ou compatible.

Étape 7 : rassembler les morceaux

L'étape 7 est tout simplement celle au cours de laquelle nous allons regrouper tous les éléments du fichier (je l'ai appelé `tp.tex`, libre à vous de le nommer autrement).

Une fois tous les morceaux rassemblés, le code source est complet. J'ai là aussi choisi de vous donner un code web plutôt que 20 pages de code illisibles.



La numérotation en première page d'un chapitre est différente de celle utilisée dans les pages qui suivent. C'est un réglage par défaut de LaTeX : ce n'est ni une erreur, ni un bug.

Code : TeX

```
\documentclass{book}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[français]{babel}
\usepackage{url} % Pour écrire des adresses cliquables.
\usepackage{lmodern} % Pour changer le pack de police.
\usepackage[top=5cm, bottom=5cm, left=6cm, right=3cm]{geometry} %
Les marges.
\title{Les droits de l'homme}
\author{\textsc{Laleloulilo} - \textsc{Zozor}}
\date{\today} % Pour mettre la date du jour, tapez \today.
\begin{document}

\maketitle % Page de garde.

\frontmatter

\chapter{Introduction}
\begin{it}
\Large{
  Les droits de l'homme sont un concept selon lequel tout être
  humain possède des droits universels, inaliénables, quel que soit le
  droit positif en vigueur ou les autres facteurs locaux tels que
  l'ethnie, la nationalité, l'orientation sexuelle ou la religion.

  Selon cette philosophie, combattue ou éclipsée aux 19\ieme{} siècle,
  20\ieme{} siècle et 21\ieme{} siècle par d'autres doctrines,
  l'homme, en tant que tel, et indépendamment de sa condition sociale,
  a des droits \og inhérents à sa personne, inaliénables et sacrés
  \fg, et donc opposables en toutes circonstances à la société et au
  pouvoir. Ainsi le concept de droits de l'homme est-il par définition
  universaliste et égalitaire, incompatible avec les systèmes et les
  régimes fondés sur la supériorité ou la \og vocation historique \fg
  d'une caste, d'une race, d'un peuple, d'une classe ou d'un
  quelconque groupe social ; incompatible tout autant avec l'idée que
  la construction d'une société meilleure justifie l'élimination ou
  l'oppression de ceux qui sont censés faire obstacle à cette
  édification.

  Les droits de l'homme, types de prérogatives dont sont titulaires
  les individus, sont généralement reconnus dans les pays occidentaux
  par la loi, par des normes de valeur constitutionnelle ou par des
  conventions internationales, afin que leur respect soit assuré, si
  besoin est même contre l'état. L'existence, la validité et le
  contenu des droits de l'homme sont un sujet permanent de débat en
  philosophie et en sciences politiques.

}
\end{it}
```

```
\mainmatter
```

```
\part{Déclaration des droits de l'homme et du citoyen de 1789}
\chapter{Histoire}
```

La Déclaration des Droits de l'Homme a été écrite dans un temple protestant. L'assemblée réunie à Versailles par la convocation des États généraux pour trouver une solution fiscale au déficit de l'État, se déclare Assemblée nationale en réunissant les trois ordres, dont elle décide l'abolition, puis s'institue Assemblée nationale constituante, et décide de rédiger une déclaration des principes fondamentaux à partir desquels sera établie une nouvelle Constitution. Elle se réunit pour cela, après avoir pris les décrets des 4 et 11 août 1789 sur la suppression des droits féodaux, qu'elle reprendra dans l'article premier de la Déclaration. La Déclaration des Droits de l'Homme et du Citoyen a été débattue par l'Assemblée nationale française sous la présidence du marquis de Mirabeau à partir d'un des trois projets proposés : celui de 24 articles, rédigé par le VI^e bureau dirigé par Jérôme Champion de Cicé. L'attribution du texte primitif à La Fayette inspiré par la Déclaration d'indépendance des États-Unis est donc erronée. L'abbé Grégoire propose que la Déclaration des droits de l'Homme et du Citoyen soit accompagnée de celle des devoirs.

La discussion débute le 9 juillet et débouche sur un vote le 26 août 1789, sous l'influence des leaders du tiers-état et de la noblesse. Le texte est peu modifié, mais est enrichi d'un préambule. Il est ratifié en partie le soir du 5 octobre 1789 par Louis XVI à Versailles, sur l'exigence de l'Assemblée, qui utilisa la pression d'une foule vindicative venue de Paris, initialement pour d'autres revendications.

Entièrement promulguée par le Roi à Paris, le 3 novembre 1789, la Déclaration des Droits est la dernière ordonnance royale. Elle servira de Préambule à la première Constitution de la Révolution française.

```
\chapter{Le texte}
\section{Introduction}
```

Les représentants du peuple français, constitués en Assemblée nationale, considérant que l'ignorance, l'oubli ou le mépris des droits de l'homme sont les seules causes des malheurs publics et de la corruption des gouvernements, ont résolu d'exposer, dans une déclaration solennelle, les droits naturels, inaliénables et sacrés de l'homme, afin que cette déclaration, constamment présente à tous les membres du corps social, leur rappelle sans cesse leurs droits et leurs devoirs ; afin que les actes du pouvoir législatif et ceux du pouvoir exécutif, pouvant être à chaque instant comparés avec le but de toute institution politique, en soient plus respectés ; afin que les réclamations des citoyens, fondées désormais sur des principes simples et incontestables, tournent toujours au maintien de la Constitution et au bonheur de tous.

```
\section{Les articles}
```

```
\paragraph{Article premier}
```

Les hommes naissent et demeurent libres et égaux en droits. Les distinctions sociales ne peuvent être fondées que sur l'utilité commune.

```
\paragraph{Article 2}
```

Les hommes naissent et demeurent libres et égaux en droits. Les distinctions sociales ne peuvent être fondées que sur l'utilité commune.

```
\paragraph{Article 3}
```

Le principe de toute souveraineté réside essentiellement dans la Nation. Nul corps, nul individu ne peut exercer d'autorité qui n'en émane expressément.

`\paragraph{Article 4}`

La liberté consiste à pouvoir faire tout ce qui ne nuit pas à autrui : ainsi, l'exercice des droits naturels de chaque homme n'a de bornes que celles qui assurent aux autres membres de la société la jouissance de ces mêmes droits. Ces bornes ne peuvent être déterminées que par la loi.

`\paragraph{Article 5}`

La loi n'a le droit de défendre que les actions nuisibles à la société. Tout ce qui n'est pas défendu par la loi ne peut être empêché, et nul ne peut être contraint à faire ce qu'elle n'ordonne pas.

`\paragraph{Article 6}`

La loi est l'expression de la volonté générale. Tous les citoyens ont droit de concourir personnellement ou par leurs représentants à sa formation. Elle doit être la même pour tous, soit qu'elle protège, soit qu'elle punisse. Tous les citoyens, étant égaux à ces yeux, sont également admissibles à toutes dignités, places et emplois publics, selon leur capacité et sans autre distinction que celle de leurs vertus et de leurs talents.

`\paragraph{Article 7}`

La loi est l'expression de la volonté générale. Tous les citoyens ont droit de concourir personnellement ou par leurs représentants à sa formation. Elle doit être la même pour tous, soit qu'elle protège, soit qu'elle punisse. Tous les citoyens, étant égaux à ces yeux, sont également admissibles à toutes dignités, places et emplois publics, selon leur capacité et sans autre distinction que celle de leurs vertus et de leurs talents.

`\paragraph{Article 8}`

La loi ne doit établir que des peines strictement et évidemment nécessaires, et nul ne peut être puni qu'en vertu d'une loi établie et promulguée antérieurement au délit, et légalement appliquée.

`\paragraph{Article 9}`

Tout homme étant présumé innocent jusqu'à ce qu'il ait été déclaré coupable, s'il est jugé indispensable de l'arrêter, toute rigueur qui ne serait pas nécessaire pour s'assurer de sa personne doit être sévèrement réprimée par la loi.

`\paragraph{Article 10}`

Nul ne doit être inquiété pour ses opinions, mêmes religieuses, pourvu que leur manifestation ne trouble pas l'ordre public établi par la loi.

`\paragraph{Article 11}`

La libre communication des pensées et des opinions est un des droits les plus précieux de l'homme ; tout citoyen peut donc parler, écrire, imprimer librement, sauf à répondre de l'abus de cette liberté dans les cas déterminés par la loi.

`\paragraph{Article 12}`

La garantie des droits de l'homme et du citoyen nécessite une force

publique ; cette force est donc instituée pour l'avantage de tous, et non pour l'utilité particulière de ceux à qui elle est confiée.

`\paragraph{Article 13}`

Pour l'entretien de la force publique, et pour les dépenses d'administration, une contribution commune est indispensable ; elle doit être également répartie entre les citoyens, en raison de leurs facultés.

`\paragraph{Article 14}`

Les citoyens ont le droit de constater, par eux-mêmes ou par leurs représentants, la nécessité de la contribution publique, de la consentir librement, d'en suivre l'emploi, et d'en déterminer la quotité, l'assiette, le recouvrement et la durée.

`\paragraph{Article 15}`

La société a le droit de demander compte à tout agent public de son administration.

`\paragraph{Article 16}`

Toute société dans laquelle la garantie des droits n'est pas assurée ni la séparation des pouvoirs déterminée, n'a point de Constitution.

`\paragraph{Article 17}`

La propriété étant un droit inviolable et sacré, nul ne peut en être privé, si ce n'est lorsque la nécessité publique, légalement constatée, l'exige évidemment, et sous la condition d'une juste et préalable indemnité.

`\chapter{Sources}`

La question des sources de la Déclaration française a suscité une controverse empreinte de nationalisme au sein de l'historiographie. Dans une brochure de 1895, l'historien allemand Georg `\textsc{Jellinek}` présentait l'`\oe`uvre française comme une simple héritière des Déclarations anglo-saxonnes (Pétition des droits, Déclaration des droits), elles-mêmes inspirées du Protestantisme luthérien. Traduite en français en 1902, dans un contexte de montée des tensions entre France et Allemagne, elle donnera lieu à une réplique aussi peu nuancée, portée par Emile `\textsc{Boutmy}` : la Déclaration des droits de l'homme et du citoyen n'aurait de source que dans la tradition philosophique et humaniste des Lumières.

Le Préambule, ajouté au projet, a été rédigé sous l'influence de `\textsc{Mirabeau}`, et de Jean-Joseph `\textsc{Mounier}`, député du Tiers qui avait fait adopter le serment du Jeu de Paume, tous deux monarchiens, c'est-à-dire partisans d'une Monarchie constitutionnelle à l'anglaise.

L'invocation à l' `\og` Être suprême `\fg` a été rajoutée au cours des séances pour tenir compte des convictions chrétiennes de presque tous les citoyens.

Le texte de l'article Un, `\og` Tous les Hommes naissent et demeurent libres et égaux en droit `\fg`, synthétise la Loi du 4 août 1789 abolissant la société d'ordres hiérarchisés.

L'article 16, associant constitution et organisation de la séparation des pouvoirs, est un principe antérieurement admis avec la séparation des ordres spirituel, politique et économique. Mais les trois pouvoirs politiques auxquels renvoie implicitement cet article, à savoir le législatif, l'exécutif et le judiciaire, est la conception proposée par Montesquieu depuis 1748 dans `\emph{De l'Esprit des Lois}`.

L'article 3, qui attribue la souveraineté à la Nation, s'inspire des thèmes des remontrances des Parlements, portées par les nombreux membres du club des Amis de la Constitution, plus connu sous le nom de Club des Jacobins, mais aussi du célèbre pamphlet de l'abbé `\textsc{Sieyès}`, qui pose l'équation : peuple = Tiers-État, c'est-à-dire que les députés du Tiers-État sont les représentants légitimes de l'ensemble du peuple.

L'article 6, directement inspiré du philosophe `\textsc{Rousseau}`, a été proposé par `\textsc{Talleyrand}`. Lu à la tribune du comité de constitution le 12 septembre 1789, ce qui deviendra l'article 6 de la déclaration des droits prenait la forme suivante : `\og` La loi étant l'expression de la volonté générale, tous les citoyens ont droit de concourir personnellement ou par représentation à sa formation ; elle doit être la même pour tous`\fg`.

Les autres articles reprennent des principes généraux du droit ou de la procédure qui sont déjà établis, comme la positivité du droit, le caractère contradictoire des procédures, la non rétroactivité des lois, etc.

Son idéal est l'individualisme libéral. C'est une `\oe`uvre de circonstance, une proclamation générale, un texte tourné vers le passé avec pour objectif d'en finir avec l'Ancien Régime ; mais également un texte tourné vers l'avenir en promouvant la philosophie des lumières et son idéal rationaliste.

`\appendix`

`\chapter{La révolution française}`

La Révolution française est la période de l'histoire de France comprise entre la convocation des États généraux en 1789 et le coup d'État du 18 brumaire (9-10 novembre 1799) de Napoléon `\textsc{Bonaparte}`. C'est un moment fondamental de l'histoire de France, marquant la fin de l'Ancien Régime, et le passage à une monarchie constitutionnelle puis à la Première République. Elle a mis fin à la royauté, à la société d'ordres et aux privilèges. Justifié par la Déclaration des droits de l'homme et du citoyen, qui proclamait l'égalité théorique des citoyens devant la loi, les libertés fondamentales et la souveraineté de la Nation, apte à se gouverner au travers des représentants élus, cette période causa la mort de plusieurs milliers de personnes et la terreur pour la majorité.

`\backmatter`

`\chapter{Sources et licences}`

`\section*{Sources}`

`\begin{description}`

```

\item[Droits de l'homme (Wikipédia):] \url{http://bit.ly/9nYoSU}
\item[Déclaration de 1789 (Wikipédia):]
\url{http://bit.ly/1DJqQg}
\item[Déclaration de 1793 (Wikipédia):] \url{http://bit.ly/SeRyG}
\item[Déclaration de 1795 (Wikipédia):]
\url{http://bit.ly/b6sdRI}
\item[La révolution française (Wikipédia):]
\url{http://bit.ly/cUd3SF}

```

`\end{description}`


```

\section*{Licence Creative Commons 3.0}

\paragraph{Vous êtes libres :}

\begin{itemize}

\item de reproduire, distribuer et communiquer cette création
au public ;
\item de modifier cette création.

\end{itemize}

\paragraph{Selon les conditions suivantes :}

\begin{itemize}

\item \emph{Paternité} : vous devez citer le nom de l'auteur
original de la manière indiquée par l'auteur de l'\oe uvre ou le
titulaire des droits qui vous confère cette autorisation (mais pas
d'une manière qui suggérerait qu'ils vous soutiennent ou approuvent
votre utilisation de l'\oe uvre).
\item \emph{Partage des conditions initiales à l'identique} :
si vous transformez ou modifiez cette oeuvre pour en créer une
nouvelle, vous devez la distribuer selon les termes du même contrat
ou avec une licence similaire ou compatible.

\end{itemize}

\end{document}

```

Conclusion

Vous voici arrivé à la fin de cette correction. Vous avez pu constater qu'une bonne organisation permettait d'avancer vite dans la création de documents avec LaTeX. Seules 7 étapes plutôt simples ont été nécessaires pour venir à bout de ce TP.

Gardez ces étapes en tête, elles vous aideront à créer de longs rapports sans vous embrouiller (Cela peut très vite arriver : lors de la rédaction du TP, je me suis aussi emmêlé les pinceaux). Il est maintenant temps de vous présenter de **nouveaux outils**. Ces outils sont un peu plus complexes que ceux utilisés jusqu'à maintenant, mais vous seront utiles pour concevoir des documents à la mise en page plus fine.

Aller plus loin

Ce TP a été l'occasion de vous entraîner à manipuler des concepts simples. Dans cette sous-partie, nous allons voir quelques subtilités de LaTeX qui pourront vous servir lors de la conception de vos publications.

Commandes et espaces

Nous allons ici apprendre à mettre une espace après une commande LaTeX.

Petit test, compilez le code suivant (donnant la figure suivante). La commande `\LaTeX` sert à écrire le mot LaTeX.

Code : TeX

```

\emph{canard} lapin\\
\LaTeX lapin\\
\LaTeX{} lapin\\

```


canard lapin
L^AT_EXlapin
L^AT_EX lapin

Les espaces

Constatez que dans la deuxième ligne, il n'y a pas d'espace entre les deux mots alors que vous en aviez inséré une dans le code. À la troisième ligne, vous remarquez que l'insertion de `{ }` vient pallier ce problème. Nous allons donc formuler ici une règle simple qui vous permettra d'insérer des espaces après les commandes **se terminant par des lettres**.

Les commandes se terminant par des lettres doivent être suivies de `{ }` afin de pouvoir insérer une espace à leur suite.

Règle toute simple, elle vous sauvera dans de nombreuses situations. Je n'ai pas utilisé cette règle dans le TP, étant donné que nous ne l'avions pas encore vue. Nous apprendrons à gérer les espaces de manière beaucoup plus fine dans le second TP.

Marges et layout

Lors du TP, nous avons paramétré les marges de notre document de sorte qu'il ait des marges définies en haut, en bas, à gauche et à droite.

Il est temps d'apprendre à faire les choses de manière plus fine. Lors de la création d'un layout (Créé via la commande `\layout` que nous avons déjà vue dans le chapitre concernant les marges), LaTeX vous montre toute une liste de paramètres. Chaque paramètre d'un layout peut être modifié et nous allons apprendre ici à le faire.



Ces informations sont données à titre de précision : si vos besoins en termes de mise en page n'exigent pas un niveau de détail exceptionnel, vous pouvez très bien vous contenter de définir les marges en haut, en bas, à droite et à gauche comme nous l'avons fait jusqu'à maintenant.

Avant toute manipulation, remémorons-nous l'apparence d'un layout ainsi que le code source nécessaire à sa création (figure suivante) :

Code : TeX

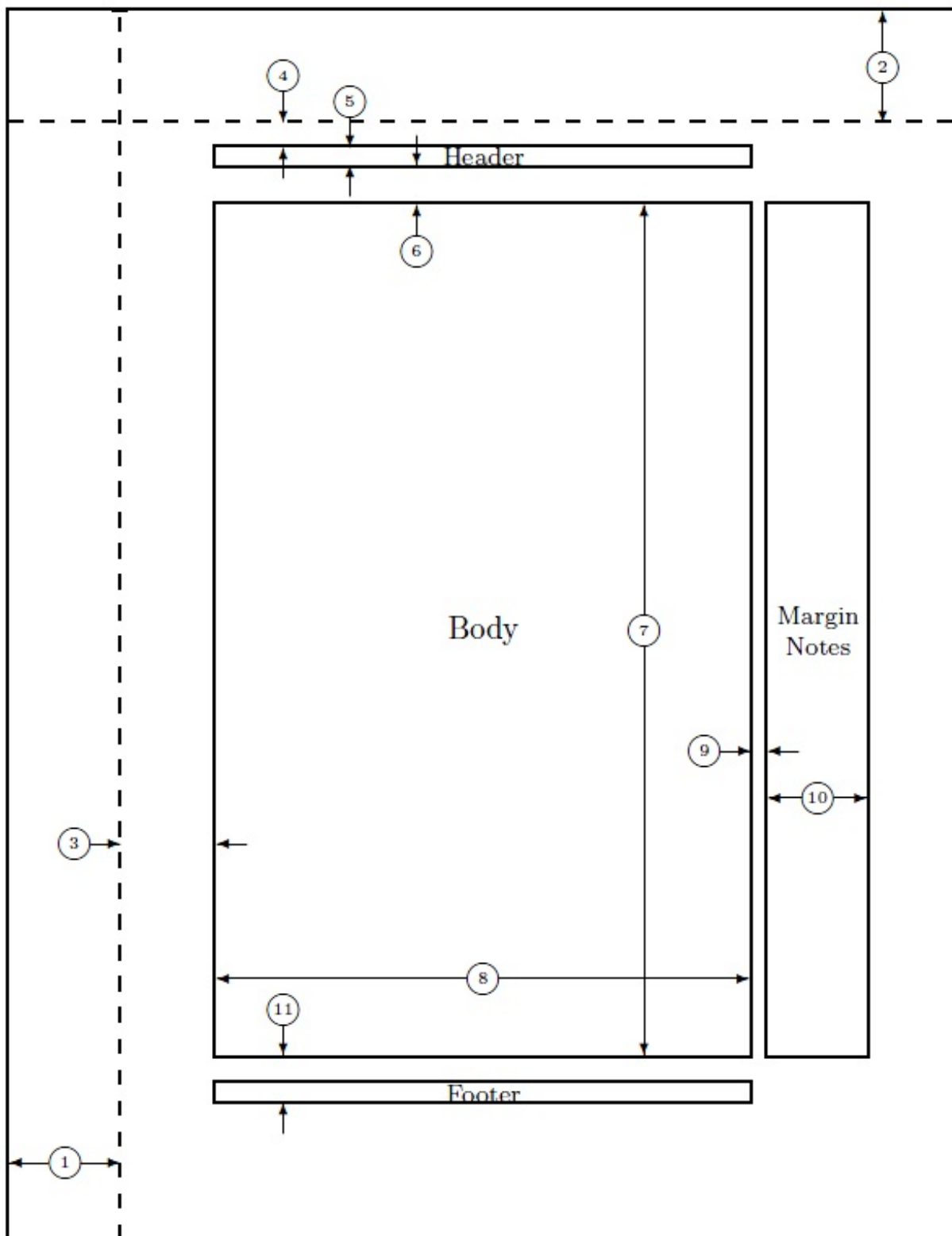
```
\documentclass{article}

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

\begin{document}

\layout

\end{document}
```



Layout par

1	<code>one inch + \hoffset</code>	2	<code>one inch + \voffset</code>
3	<code>\oddsidemargin = 62pt</code>	4	<code>\topmargin = 16pt</code>
5	<code>\headheight = 12pt</code>	6	<code>\headsep = 25pt</code>
7	<code>\textheight = 550pt</code>	8	<code>\textwidth = 345pt</code>
9	<code>\marginparsep = 11pt</code>	10	<code>\marginparwidth = 65pt</code>
11	<code>\footskip = 30pt</code>		<code>\marginparpush = 5pt (not shown)</code>
	<code>\hoffset = 0pt</code>		<code>\voffset = 0pt</code>
	<code>\paperwidth = 614pt</code>		<code>\paperheight = 794pt</code>
			défaut

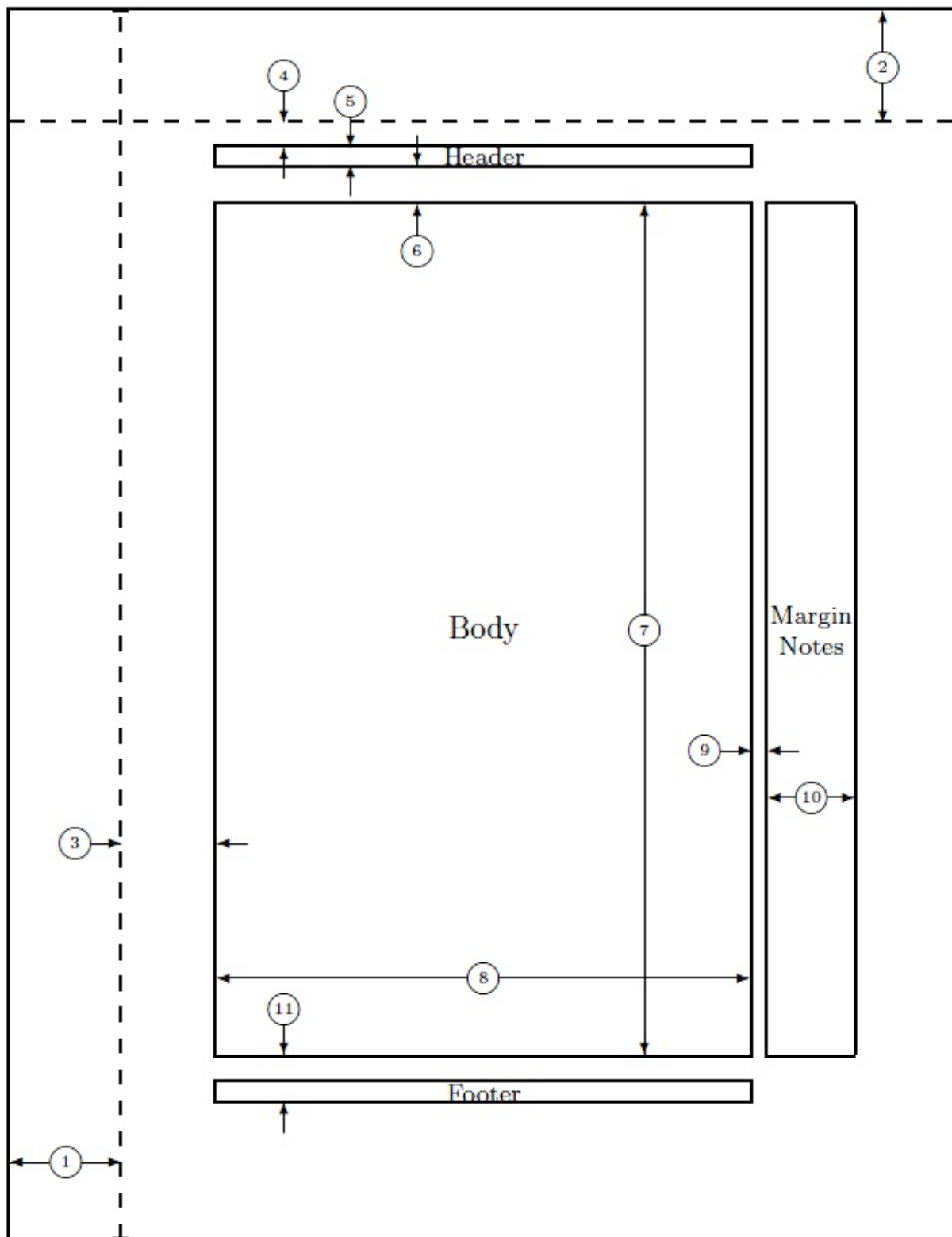
Chaque longueur est associée à une commande et à un repère sur le layout, de sorte que vous puissiez facilement identifier les

différentes longueurs et les associer à leur nom.

Je vous montre ici comment modifier une longueur du layout. Prenons la longueur `\marginparwidth` repérée par le nombre 10 dans le layout (figure suivante). Pour la modifier (figure suivante), il suffit d'utiliser la commande suivante dans le préambule du document :

Code : TeX

```
\setlength{nom de la longueur}{longueur dans l'unité qui vous plaît}  
\setlength{\marginparwidth}{2cm} % Les centimètres seront  
automatiquement convertis.
```



Layout modifi

1	<code>one inch + \hoffset</code>	2	<code>one inch + \voffset</code>
3	<code>\oddsidemargin = 62pt</code>	4	<code>\topmargin = 16pt</code>
5	<code>\headheight = 12pt</code>	6	<code>\headsep = 25pt</code>
7	<code>\textheight = 550pt</code>	8	<code>\textwidth = 345pt</code>
9	<code>\marginparsep = 11pt</code>	10	<code>\marginparwidth = 56pt</code>
11	<code>\footskip = 30pt</code>		<code>\marginparpush = 5pt (not shown)</code>
	<code>\hoffset = 0pt</code>		<code>\voffset = 0pt</code>
	<code>\paperwidth = 614pt</code>		<code>\paperheight = 794pt</code>

é

Ce mécanisme est applicable à toutes les longueurs figurant dans le layout.



La liste des unités de longueur vous sera fournie dans l'annexe \og Mémento \fg.

Personnalisation de la commande `\emph`

Nous avons vu dans le chapitre traitant du texte et des polices que la commande `\emph` ne servait non pas à mettre du texte en italique, mais à mettre des termes en emphase (c'est-à-dire à signaler leur importance).

Ce TP est l'occasion pour vous d'apprendre à modifier la manière dont `\emph` met en valeur du texte.

Nous souhaitons ici que la commande transforme le texte en texte type machine à écrire plutôt qu'en italique. Le résultat serait le suivant (capture d'écran du dernier paragraphe du TP sur la figure suivante) :

Code : TeX

```
\begin{itemize}

  \item \emph{Paternité} : vous devez citer le nom de l'auteur
    originalde la manière indiquée par l'auteur de l'\oe uvre ou le
    titulaire desdroits qui vous confère cette autorisation (mais pas
    d'une manière qui suggérerait qu'ils vous soutiennent ou approuvent
    votre utilisation del'\oe uvre).

  \item \emph{Partage des conditions initiales à l'identique} :
    sivous transformez ou modifiez cette œuvre pour en créer une
    nouvelle, vousdevez la distribuer selon les termes du même contrat
    ou avec une licencesimilaire ou compatible.

\end{itemize}
```

Selon les conditions suivantes :

- **Paternité** : vous devez citer le nom de l'auteur original de la manière indiquée par l'auteur de l'œuvre ou le titulaire des droits qui vous confère cette autorisation (mais pas d'une manière qui suggérerait qu'ils vous soutiennent ou approuvent votre utilisation de l'œuvre).
- **Partage des conditions initiales à l'identique** : si vous transformez ou modifiez cette oeuvre pour en créer une nouvelle, vous devez la distribuer selon les termes du même contrat ou avec une licence similaire ou compatible.

Mise en valeur du texte

Comment en sommes-nous arrivés là ? Tout simplement en redéfinissant la commande `\emph` grâce à la commande suivante insérée dans le préambule du document :

Code : TeX

```
\renewcommand{nom de la commande}{fonctions liées à cette commande}
\renewcommand{\emph}{\texttt}
```

Ici la commande est `\emph`, elle est associée à la transformation « machine à écrire » (`\texttt`). N'ayant pas spécifié que le texte devait être en italique, il ne le sera pas. En revanche, des problèmes pourront se poser si elle est insérée dans un texte en

italique (elle ne jouera plus son rôle).

La commande `\renewcommand` a de nombreuses applications, nous en verrons d'autres par la suite.

En-têtes et pieds de pages personnalisés

Il y a de ça quelques chapitres, vous étiez jeune et innocent et utilisiez la commande `\pagestyle` avec l'une des trois options prédéfinies : `plain`, `empty` ou `headings`.

Vous voici prêt à apprendre à créer des en-têtes et pieds de pages personnalisés. La commande `\pagestyle{fancy}` permet de faire cela (elle est fournie par le package `fancyhdr`).

`\pagestyle{fancy}` découpe chaque en-tête et pied de page en trois morceaux, de sorte que nous devons personnaliser six commandes différentes pour arriver à créer une mise en page cohérente des en-têtes et pieds de pages.

- `\lhead{...}` gère le coin gauche de l'en-tête.
- `\chead{...}` gère le centre de l'en-tête.
- `\rhead{...}` gère le coin droit de l'en-tête.
- `\lfoot{...}` gère le coin gauche du pied de page.
- `\cfoot{...}` gère le centre du pied de page.
- `\rfoot{...}` gère le coin droit du pied de page.

Diverses autres commandes sont utilisables :

- `\leftmark` : nom du chapitre ;
- `\rightmark` : nom de la section ;
- `\thepage` : numéro de page ;
- `\thechapter` : numéro du chapitre ;
- `\thesection` : numéro de la section courante ;
- `\renewcommand{\headrulewidth}` {épaisseur dans l'unité de votre choix} : épaisseur de la ligne entre l'en-tête et le corps du texte ;
- `\renewcommand{\footrulewidth}` {épaisseur dans l'unité de votre choix} : épaisseur de la ligne entre le pied de page et le corps du texte ;
- `\renewcommand{\headsep}` {longueur dans l'unité de votre choix} : espacement entre l'en-tête et le corps du document.

Faites le test sur le document que nous venons de créer dans le TP en remplaçant les premières lignes par les suivantes.

Code : TeX

```
\documentclass{book}

\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[français]{babel}
\usepackage{url}
\usepackage{lmodern}
\usepackage{fancyhdr}
\usepackage[top=5cm, bottom=5cm, left=6cm, right=3cm]{geometry}

\title{Les droits de l'homme}
\author{\textsc{Laleloulilo} - \textsc{Zozor}}
\date{\today} % Pour mettre la date du jour, tapez \today.
\begin{document}

\pagestyle{fancy}

\renewcommand{\headheight}{13pt}
\renewcommand{\headrulewidth}{2pt}
\renewcommand{\footrulewidth}{2pt}
\renewcommand{\headsep}{15pt}
\lhead{\rightmark}
\chead{}
\rhead{\leftmark}
\lfoot{Section \thesection}
\cfoot{Chapitre \thechapter}
\rfoot{\thepage}
```

```
\maketitle % Page de garde.
```

Toutes les commandes ont été utilisées ici. Avouons-le, certaines pages ne sont pas des modèles de grâce. Qu'à cela ne tienne, vous pouvez changer le style des en-têtes et pieds de pages à la carte via la commande `\thispagestyle{ }` qui peut se voir attribuer elle aussi les options `plain`, `empty` ou `headings`. Cette commande se place n'importe où dans le document, elle prendra effet sur la page où le texte qui l'entoure est mis en page.

En résumé

- Écrire un petit livre en LaTeX n'est pas difficile, mais demande beaucoup d'organisation. Il faut respecter la hiérarchie imposée par LaTeX.
- La commande `\emph{ }` indique que le texte doit être mis en valeur. Celui-ci peut être écrit en italique ou d'une façon différente selon le contexte.
- Les en-têtes et pieds de pages peuvent être paramétrés grâce à `\pagestyle{fancy}` et la foule de commandes annexes qui lui sont dédiées.
- Les commandes se terminant par des lettres doivent être suivies de `{ }` si l'on veut insérer une espace à leur suite.

Partie 3 : Utilisation avancée de LaTeX

La partie 3 contient des chapitres potentiellement difficiles à maîtriser. Aucune notion présentée ici n'est évidente. La maîtrise de vos interfaces et du concept de compilation est crucial afin d'appréhender les notions introduites.

Les figures

Ce premier chapitre de la troisième partie de cet ouvrage introduit les manipulations de figures. Les concepts et packages présentés ici, bien qu'assez faciles à prendre en main dans des cas simples, peuvent se révéler extrêmement agaçants dans des situations plus exotiques. Les plus expérimentés connaissent la difficulté rencontrée lors de l'insertion de certaines figures dans des documents à la mise en page complexe.

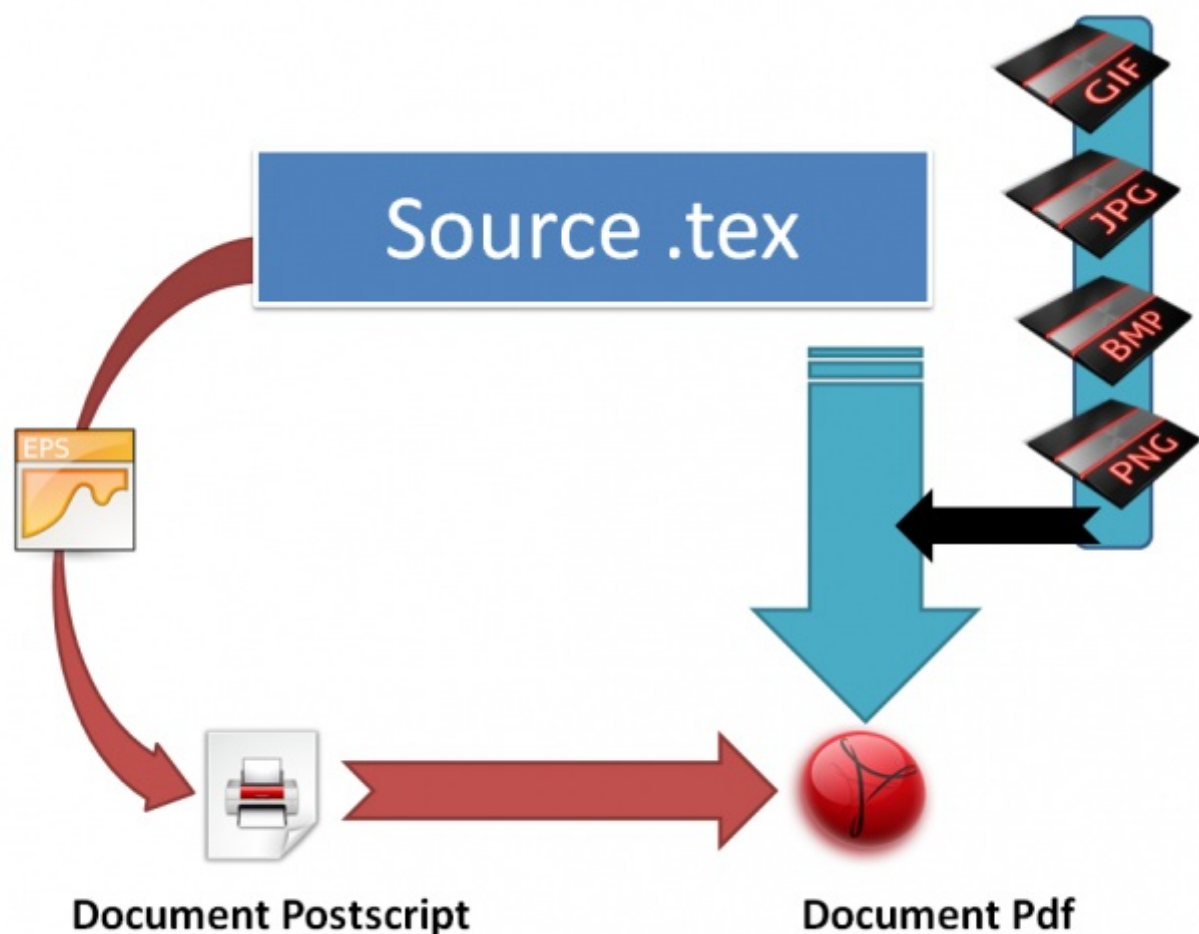
Ne vous inquiétez pas, nous allons découvrir ensemble et sans empressement les notions liées aux figures. Un nouveau concept fera également son apparition dans ce livre : **le flottant**. Nous prendrons le temps de l'étudier.

Les différents formats d'images

LaTeX, avec ses packages, vous permet d'insérer des figures et des images de divers formats dans votre document et de les arranger selon vos souhaits. Une chose importante est à noter : **tous les formats d'images ne sont pas compatibles entre eux**.

Pourquoi ? À cause de la compilation. Nous avons appris dans le troisième chapitre qu'un fichier `.tex` pouvait, selon vos besoins, être compilé en fichiers de différents formats (nous avons parlé des formats PS et PDF). Selon la méthode de compilation, il vous sera permis d'utiliser certains formats d'images.

Avant de donner des explications techniques, jetons un œil au schéma de la figure suivante.



La gestion des formats d'images

Nous nous intéresserons ici à l'insertion de plusieurs formats courants d'images et de figures (EPS, PNG, GIF, JPEG, BMP). Nous constatons sur ce schéma qu'un fichier PDF s'obtient de deux manières bien distinctes : soit il est le produit d'une compilation

directe du fichier `.tex`, soit il est le résultat de la conversion d'un fichier PostScript (ces possibilités de conversion et de compilation ont été abordées dans le chapitre 3 en détail, revenez-y si elles ne sont pas claires dans votre esprit).

Cette différence nous intéresse, parce que le schéma montre qu'une compilation directe en PDF ne permet pas l'insertion de figures d'extension `.eps`, alors qu'une compilation d'un fichier `.tex` en un fichier PostScript l'autorise.



Pour des raisons de compréhension, j'appelle « figures » la famille des PostScript et « images » la famille des PNG. Même si le choix est assez litigieux, au moins il m'est plus facile de vous faire comprendre quels formats sont autorisés (ou ne le sont pas) dans une publication en cours.

Nous sommes ici face à une originalité technique embêtante : tous les formats d'images ne peuvent pas être utilisés simultanément dans un document. Conséquence logique, un choix entre l'utilisation de figures (EPS et PS) ou d'images (BMP, GIF, PNG, JPEG) s'impose dès la conception d'un document.



Une image ou une figure peut se convertir facilement d'un format à un autre en l'ouvrant avec un logiciel de traitement d'images (Gimp, Photoshop\dots) pour l'enregistrer sous un autre format.

Votre choix doit se faire selon le type de fichier que vous manipulez. Des chercheurs auront plutôt tendance à travailler avec le format PostScript, alors que les autres préféreront probablement utiliser des images au format PNG. Quel que soit votre choix (En cas d'hésitation, je vous conseille plutôt les PNG.), les packages et les commandes utilisés par la suite restent les mêmes.

Résumons tout ceci !

- Si vous souhaitez insérer des figures en `.eps`, il est **obligatoire** de compiler en PostScript avant d'effectuer une conversion en PDF (vous pouvez également conserver votre fichier en PS). Il vous sera alors interdit d'utiliser des formats tels que le PNG, le BMP, le JPEG ou le GIF.
- Si vous souhaitez insérer des images (GIF, BMP, JPEG, PNG...), il est **obligatoire** de compiler en PDF **directement**. Il vous sera interdit d'utiliser les figures au format PS ou EPS.
- Une image peut facilement se convertir en figure et vice-versa.

Insertion d'images et de figures

L'épineux problème de la cohabitation entre les fichiers `.eps` et les autres formats d'images ayant été traité, nous pouvons aborder la pratique.

Nous allons apprendre à insérer une ou plusieurs images dans un document. Les opérations effectuées dans cette partie nécessiteront toutes l'utilisation du package `graphicx`. Chargez-le via la commande `\usepackage{graphicx}` à chaque fois que vous aurez à manipuler des images.

Pour travailler une image, il nous faut tout d'abord... une image. Voici donc la Wonder Poulpy (Mascotte officielle de tous les jours et unique modèle de poulpe de compagnie.), qui a accepté de se faire photographier en exclusivité afin que nous puissions manipuler sa photo et l'inclure dans nos documents.



Poulpy



Téléchargez cette image grâce au code web fourni ci-dessus. Pour ce faire, effectuez un clic droit sur l'image avant de l'enregistrer sous le nom de votre choix (évitez les accents et les espaces) **dans le dossier où vous composez vos documents LaTeX**. Pour ma part, je l'ai appelée `poulpy.png`.

Insertion simple

Une image est insérée dans un document via la commande suivante :

Code : TeX

```
\includegraphics{chemin de l'image}
```

Le chemin de l'image est l'adresse de l'image sur votre ordinateur. Vous pouvez spécifier un chemin absolu ou un chemin relatif, au choix.



Le chemin absolu est le chemin exact à parcourir dans vos dossiers et répertoires permettant de retrouver votre image, il est du type : `C:\Cassoulet\Documents\fichierslatex\chapitre9\poulpy.png`.

Le chemin relatif, en revanche, est le chemin à parcourir depuis le fichier source pour retrouver l'image. Ainsi, si le fichier `.tex` se trouve dans le répertoire `fichierslatex` mentionné précédemment, le chemin relatif de l'image sera `chapitre9\poulpy.png`.

Comme nous avons enregistré l'image de Poulpy dans le dossier où se trouve notre fichier `.tex`, son chemin relatif se résume à `poulpy.png`. C'est celui-ci que nous utiliserons dans l'exemple qui suit :

Code : TeX

```
\documentclass[12pt]{report}

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

\begin{document}
\includegraphics{poulpy.png}
\end{document}
```

Le résultat attendu s'affiche : l'image est à sa taille d'origine et positionnée en haut à gauche du document (figure suivante).



insertion de poulpy

Taille d'une image

Vous avez plusieurs possibilités pour indiquer à LaTeX la taille de l'image que vous souhaitez insérer. Vous pouvez indifféremment lui demander de :

- faire en sorte qu'une image ait une largeur donnée, il adaptera la hauteur afin de conserver ses proportions ;
- faire en sorte qu'une image ait une hauteur donnée, il adaptera la largeur de la même façon ;
- fixer à la fois la hauteur et la largeur, quitte à déformer l'image ;
- choisir un coefficient de proportionnalité permettant de réduire de façon cohérente et simultanée la hauteur et la largeur de l'image.

La largeur et la hauteur sont fixées respectivement par les variables `width` et `height`. L'échelle, quant à elle, se fixe grâce à la variable `scale` et un coefficient. Ce dernier vaut 1 pour la taille par défaut de l'image, à vous de le diminuer ou de l'augmenter selon vos désirs. Voici plusieurs exemples de codes insérant une image redimensionnée.

Code : TeX

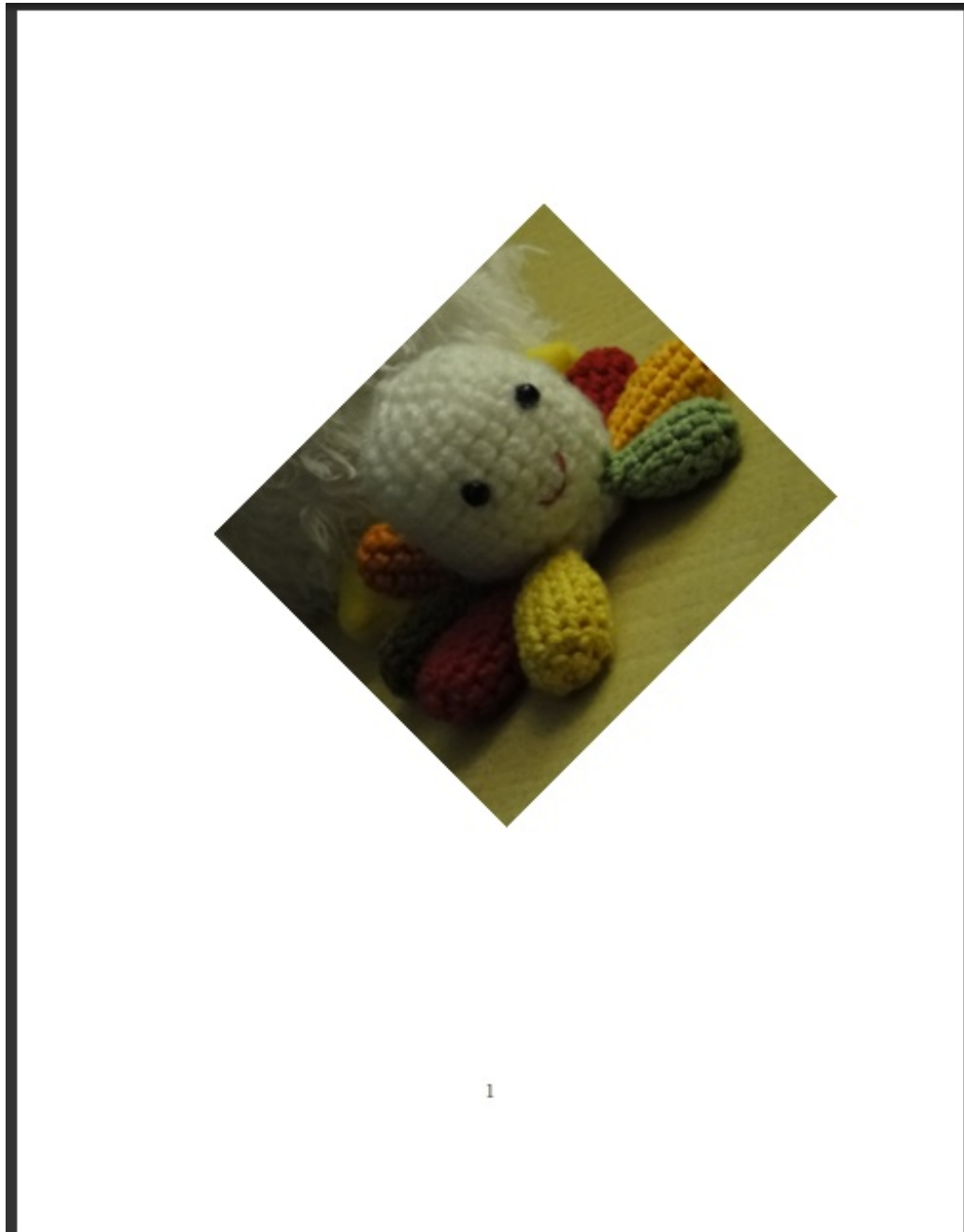
```
\includegraphics[width=200]{poulpy.png}
\includegraphics[height=200]{poulpy.png}
\includegraphics[height=200, width=600]{poulpy.png}
% Ici, Poulpy est un peu plate
\includegraphics[scale=1.5]{poulpy.png}
% Ici, Poulpy est plutôt grande
```

Rotation d'une image

La variable `angle` nous permet de faire tourner une image dans un document. Le rendu sur des images rectangulaires n'est pas très esthétique, mais cela peut être utile avec des images ayant une autre forme. La figure suivante vous montre l'utilisation de cette option.

Code : TeX

```
\includegraphics[angle=45]{poulpy.png} % Poulpy en biais
```



Poulpy en biais

Insertion d'une portion d'image

Il est souvent agréable de pouvoir ajouter uniquement une portion de l'image originale dans un document sans avoir à passer par un logiciel d'édition. Nous allons ici apprendre à isoler une portion rectangulaire d'une image de grande taille.



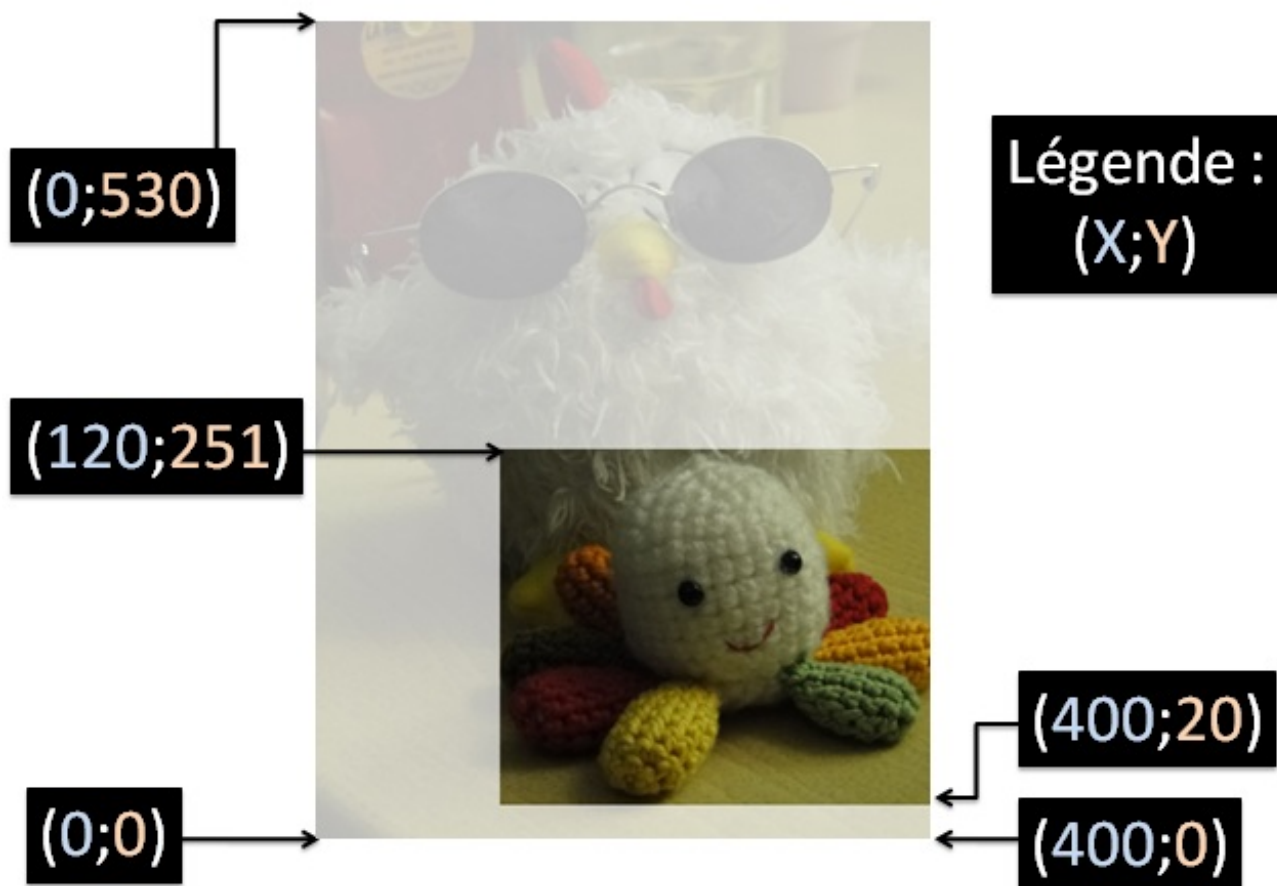
Les subtilités développées dans ce paragraphe ne fonctionnent (facilement) qu'avec les fichiers PostScript.

Pour notre exemple, nous allons nous exercer sur une photo de Wonder Poulpy et Monsieur Poule que vous voyez en figure suivante. Vous pouvez la télécharger au format EPS en [cliquant ici](#).



Wonder Poulpy et Monsieur Poule

Ici, notre objectif va être d'isoler Poulpy, afin de pouvoir insérer sa photo dans un nouveau document sans son compagnon à plumes. Chaque pixel de l'image représente un point. À chaque point sont associées une abscisse et une ordonnée, l'origine se trouvant sur le coin inférieur gauche de l'image. Si nous souhaitons isoler Poulpy, notre photo de départ devient similaire à celle visible en figure suivante.



Code : TeX

```
\includegraphics*[abscisse du coin inférieur gauche,ordonnée du coin
inférieur gauche][abscisse du coin supérieur droit,ordonnée du coin
supérieur droit du rectangle]{chemin de l'image}
```

Tentez d'isoler Poulpy !

Je vais tout de même donner la solution à ceux qui n'auraient pas le courage de faire l'exercice (il est un peu difficile pour les moins expérimentés).

Code : TeX

```
\includegraphics*[120,20][400,251]{poulpy_et_mr_poule.eps}
```

Intégration d'une image dans un paragraphe

Nous allons maintenant nous pencher sur l'intégration d'une image dans du texte de façon à ce que le texte contourne la figure.



Soyons francs, LaTeX n'aime pas trop ça et peut de temps en temps gérer les choses de manière extrêmement exotique.

Le package utilisé ici est `wrapfig`. Il est nécessaire de l'appeler lorsque vous souhaitez inclure l'environnement `wrapfigure` servant à encadrer des images ou des figures par le texte d'un paragraphe. Plusieurs variables font leur apparition avec l'utilisation de l'environnement `wrapfigure` :

- le nombre de lignes nécessaires à la bonne intégration de l'image ;
- la taille du dépassement autorisé dans la marge (nous laisserons cette variable à 0 afin de garder des publications propres) ;
- la largeur de l'image ;
- l'alignement de l'image.

La syntaxe à observer est la suivante :

Code : TeX

```
\begin{wrapfigure}[nombre de lignes]{placement}{largeur de l'image
en cm}
\includegraphics[width=largeur en cm]{votre image}
\end{wrapfigure}
Votre paragraphe, sans saut de ligne entre \end et le paragraphe
```

Le placement se définit à partir des lettres :

- `l` pour placer l'image à gauche ;
- `r` pour placer l'image à droite ;
- `o` pour placer l'image à l'extérieur, c'est-à-dire à droite pour une page impaire et à gauche pour une page paire ;
- `i` pour placer l'image à l'intérieur, c'est-à-dire à gauche pour une page impaire et à droite pour une page paire.

Petit exercice : reprenez l'image de Poulpy donnée en début de chapitre et tentez de l'insérer dans un paragraphe de faux texte. L'image devra faire 4 cm de large, occuper 8 lignes et se trouver à droite de votre paragraphe.

La solution de l'exercice se trouve ci-dessous, illustrée avec la figure suivante.

Code : TeX


```

\documentclass[12pt]{report}

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

\usepackage{wrapfig}
\usepackage{graphicx}
\begin{document}

\begin{wrapfigure}[8]{r}{4cm}
\includegraphics[width=4cm]{poulpy.png}
\end{wrapfigure}
Gros paragraphe.
\end{document}

```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut sit amet lectus a odio condimentum porttitor ac cursus orci. Aenean at sapien turpis. Fusce sollicitudin dictum tellus placerat porta. Curabitur lacinia consequat quam. Cras dapibus, sem vitae posuere facilisis, turpis sem facilisis arcu, quis ornare urna risus quis justo. Nunc sagittis blandit lectus sit amet ultrices. Curabitur lacinia consequat quam. Cras dapibus, sem vitae posuere facilisis, turpis sem facilisis arcu, quis ornare urna risus quis justo. Nunc sagittis blandit lectus sit amet ultrices.



Rendu de l'insertion

Les flottants

Les techniques pour insérer des images développées dans les paragraphes précédents sont des solutions simples, mais malheureusement peu utilisées. LaTeX propose aux auteurs d'optimiser le placement des images et des figures à leur place (C'est l'une de ses fonctions phares.).

Afin de laisser à LaTeX la possibilité de positionner des images à sa guise, nous allons placer ces images dans des environnements dits « flottants ». Le contenu de ces environnements sera placé de la manière que LaTeX estimera la plus judicieuse dans votre mise en page.

Bien entendu, vous pourrez indiquer à LaTeX votre préférence pour un type de placement en particulier.

Après avoir vu comment placer des flottants insérant des images, nous apprendrons à les légender. Ces légendes seront également traitées ultérieurement.

L'environnement figure

La création d'un flottant

Pour mettre une image en flottant, nous allons tout simplement insérer la commande habituelle `\includegraphics` dans l'environnement `figure`. Afin de rendre le tout plus élégant dans le document final, nous centrerons l'image.



Notons que l'environnement `center` utilisé ici n'est pas un simple élément facultatif de mise en page : il nous permettra par la suite de rendre cohérents les alignements des images (et des figures) avec les alignements des légendes.

Code : TeX

```

\begin{figure}

\begin{center}
\includegraphics{poulpy.png}
\end{center}

\end{figure}

```



Un flottant

Le placement

Il est possible de spécifier à l'environnement `figure` (et par extension à LaTeX) votre préférence en termes de placement.

Vous pouvez souhaiter que votre flottant soit plutôt :

- placé en haut de page (option `t`) ;
- placé en bas de page (option `b`) ;
- sur une page ne comportant que des flottants (option `p`).

Il existe encore deux options. Vous pouvez indiquer à LaTeX que vous souhaitez faire figurer l'image dans la zone où vous avez tapé la commande de manière informative (option `h`) ou bien de manière insistante (option `H`).

`H` et `h` sont des outils intéressants, mais que je vous conseille d'employer avec modération. Laissez LaTeX faire son travail avant de lui imposer des contraintes.

Les options de placement sont spécifiées dans la commande d'ouverture de l'environnement `figure` selon la syntaxe que voici.

Code : TeX

```

\begin{figure}[les options non séparées par des virgules]
Quelques démonstrations ci-dessous.
Le « ! » est utilisé ici pour faire comprendre
à LaTeX que nous insistons « énormément » sur une option.
\begin{figure}[b] %nous voulons le flottant en bas.
\begin{figure}[!b] %nous voulons le flottant en bas (avec
insistance).
\begin{figure}[bt] %nous voulons le flottant en bas, ou en haut
s'il ne peut pas être en bas.
\begin{figure}[h] %nous voulons le flottant ici.
\begin{figure}[H] %nous voulons le flottant ICI !
\begin{figure}[hb] %nous voulons le flottant ici, ou en bas si cela

```

n'est pas possible.

Les placements par défaut

LaTeX, selon votre distribution et votre configuration, place les flottants par défaut suivant l'une ou l'autre des options développées ci-dessus. La commande `\floatplacement{type de flottant}{options}` s'utilise afin de changer le comportement par défaut de LaTeX.

Dans notre cas, le type de flottant est `figure` et la commande s'utilise par exemple comme ceci :

Code : TeX

```
\floatplacement{figure}{t}
```



Cette commande est fournie par le package `float`.

Les légendes

Légender les figures et les images est indispensable en LaTeX. Nous allons à présent voir comment ajouter des légendes à nos flottants. C'est le rôle de la commande `\caption{ma légende}`. Elle s'utilise à la suite de l'environnement `center` et précède une éventuelle commande `\label`.

Un démonstration s'impose (figure suivante) :

Code : TeX

```
\begin{figure}
\begin{center}
\includegraphics{poulpy.png}
\end{center}
\caption{Poulpy est multicolore}
\label{Poulpy est multicolore}
\end{figure}
```



Figure légendée

FIGURE 1 – Poulpy est multicolore

Sauts de page

Nous avons appris dans les chapitres traitant de la mise en page à dire à LaTeX d'effectuer un saut de page, grâce à la commande `\newpage`. Avec les flottants, deux nouvelles commandes apparaissent :

- `\clearpage`, qui réalise un saut de page tout en imposant à LaTeX de mettre tous les flottants en page (il les traitera tous, et produira des pages remplies par les flottants non traités) ;
- `\cleardoublepage`, qui a le même effet, si ce n'est qu'il reprend la nouvelle page sur une page impaire.

Comprenez bien l'utilité de ces commandes. Si vous écrivez un rapport en deux grandes parties, il est primordial qu'aucune image de la première partie ne se retrouve dans la seconde à cause d'un choix inopportun de LaTeX. Ces deux commandes s'utilisent exactement de la même manière que `\newpage`.

En résumé

- Il faut choisir entre l'élaboration d'un document comprenant des figures (PS et EPS) ou des images (JPEG, PNG, BMP...).
- Pour insérer des images, il faut utiliser le package `graphicx` et faire ensuite appel à la commande `\includegraphics`.
- Il est possible d'insérer précisément des images ou figures n'importe où dans un document, mais il est fortement conseillé de laisser LaTeX choisir à notre place leur emplacement. On utilise pour cela l'environnement `figure` qui rend l'image flottante : LaTeX sélectionne alors l'emplacement le plus approprié pour l'image (C'est parfois sur la page suivante !).
- Pour afficher la légende d'une image, on utilise la commande `\caption`.

Les tableaux

En sciences, les tableaux sont des éléments essentiels. Les créateurs de LaTeX ont su les mettre à l'honneur et apporter une somme colossale de packages et d'environnements sachant les gérer. Massifs et encombrants, comme des images ou des figures, les tableaux sont des flottants et LaTeX peut les intégrer comme tels.

Les tableaux, tout comme les notations mathématiques, font l'objet d'une documentation extrêmement fournie. Ce chapitre traitera plusieurs points utiles à la création des tableaux en vous apprenant à :

- composer des tableaux simples ;
- fusionner des cellules ;
- paramétrer le placement des flottants ainsi que quelques autres détails de mise en page.

Bien que difficile, ce chapitre sera extrêmement riche en exemples.

Structure type d'un tableau

Dans cette première partie, nous allons faire nos premiers pas avec des tableaux. Notre but sera de réaliser une grande table de multiplication.

La théorie

Tableau sans bordure

Le tableau sans bordure est le plus simple à créer. Ici, tout se passe dans l'environnement `tabular`, et nous souhaitons créer le tableau visible en figure suivante. Sa création va se dérouler en trois temps.

1	2
3	4

Tableau simple

Dans un premier temps, nous devons décider de l'alignement des cellules dans chaque colonne. Elles peuvent être...

- à droite (option `r`) ;
- à gauche (option `l`) ;
- centrées (option `c`).

Dans un deuxième temps, l'une des trois options sera choisie pour chaque colonne et insérée entre accolades à l'ouverture de l'environnement `tabular`.

Ici, nous souhaitons créer un tableau contenant deux colonnes centrées. Notre squelette de départ sera donc le suivant :

Code : TeX

```
\begin{tabular}{cc}  
\end{tabular}
```

Pour finir, nous rentrons ligne par ligne le contenu des cellules, séparé par le caractère « `&` ». Chaque ligne se terminera par `\\` (c'est un moyen d'indiquer à LaTeX que l'on veut changer de ligne).

Voici donc le code source de notre tableau simple :

Code : TeX

```
\begin{tabular}{cc}
1 & 2 \\
3 & 4 \\
\end{tabular}
```



Bien que peu intuitif au début, l'environnement `tabular` est très pratique.

Tableau avec bordures

Nous souhaitons maintenant ajouter des bordures visibles. Afin de faire apparaître les lignes, il faut placer la commande `\hline` entre deux lignes consécutives, et plus généralement partout où vous souhaitez obtenir une ligne horizontale (dans un tableau).

Démonstration :

Code : TeX

```
\begin{tabular}{cc}
\hline
1 & 2 \\
\hline
3 & 4 \\
\hline
\end{tabular}
```

1	2
3	4

Les bordures, première étape

Les colonnes, quant à elles, sont un peu plus subtiles à mettre en place. Pour marquer la séparation entre deux colonnes consécutives par une bordure verticale, il faut insérer le caractère «`|`» (Le caractère «`|`» est présent en haut de votre clavier. Sur un clavier français, vous pouvez l'insérer avec la combinaison de touches `AltGr + 6`.) entre elles lors de la spécification des alignements (figure suivante).

Ici, `{cc}` devient `{|c|c|}`.

Démonstration :

Code : TeX

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

1	2
3	4

Bordures, seconde étape

Exercice

Il est temps pour vous de souffrir. Dans cet exercice, vous devez reproduire en LaTeX l'affreux tableau ci-dessous (figure suivante). Prenez le temps de réfléchir, il n'y a pas de piège. Recopier ce tableau devrait vous prendre moins de dix minutes.

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

Table de multiplication

La solution :

Code : TeX

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



```

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

\end{tabular}

```

Fusion de cellules

Nous allons apprendre à fusionner des lignes et des colonnes. Les commandes exposées ici ne sont pas faciles à prendre en main et demandent un peu d'entraînement et de gymnastique intellectuelle. La difficulté réside dans la mémorisation des commandes. Une fois que vous les connaîtrez par cœur, les choses seront beaucoup plus faciles.

Fusion de colonnes

La commande de fusion de colonnes est la suivante :

Code : TeX

```

\multicolumn{nombre de colonnes fusionnées}{c, l ou r accompagné des
bordures éventuelles}{le texte de la cellule}

```

La difficulté réside dans la possibilité de choisir de nouveau un alignement pour la cellule fusionnée. Faites bien attention à ce que l'alignement et les bordures de votre cellule soient cohérents avec celles du reste du tableau.

La figure suivante met en pratique la commande de fusion de colonnes (regardez bien les différentes syntaxes utilisées) :

Code : TeX

```

\begin{tabular}{|c|c|c|c|c|}
\hline
1 & \multicolumn{2}{c|}{2} & 3 & 4 \\
\hline
\multicolumn{2}{c|}{5} & 6 & 7 & 8 \\
\hline
9 & 10 & \multicolumn{3}{c|}{11} \\
\hline
\end{tabular}

```

1	2	3	4
5	6	7	8
9	10	11	

Fusion de colonnes

Fusion de lignes

Pour fusionner deux lignes entre elles, on utilise la commande `\multirow`, contenue dans le package du même nom. Elle est dépendante de trois paramètres (L'un des paramètres est remplaçable par une étoile.) :

Code : TeX

```
\multirow{nombre de lignes fusionnées}{taille de la colonne en
cm}{votre texte}
\multirow{nombre de lignes fusionnées}*{votre texte}
```

Vous pouvez voir ci-dessous la démonstration de l'utilisation de `\multirow`. Notez que la commande `\cline` a été introduite. Elle vous permet selon vos souhaits de ne mettre une ligne horizontale qu'entre deux colonnes choisies par vous (Les colonnes sont numérotées de la gauche vers la droite.). Prenez le temps de regarder le code et la figure suivante afin de mieux comprendre l'utilité de `\cline`.

Code : TeX

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

1	2		3	4
5		6	7	8
9	10	11		
12	13	14	15	16
	17	18	19	20
21	22	23	24	25
26	27		28	29

Fusion de lignes et de colonnes

Autres paramètres applicables à un tableau

De nombreuses options de personnalisation sont disponibles sous LaTeX. Nous allons ici étudier les plus courantes avant d'apprendre à nous servir de l'environnement `table`, permettant de faire flotter nos tableaux. Nous les rendrons plus élégants et plus lisibles.

Modifications mineures

Colonne de largeur paramétrée

Nous avons appris précédemment à créer des colonnes en spécifiant simplement l'alignement de leur contenu. Il existe des alternatives au trio `r`, `c` et `l`. Je vais vous en présenter trois.

La nouvelle instruction possible est :

Code : TeX

```
p{largeur de la colonne en centimètres}
```

À la différence de `r`, `c` et `l`, cette instruction n'a pas d'influence sur l'alignement horizontal du texte au sein des cellules. La figure suivante illustre son fonctionnement.

Code : TeX

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

1cm	2cm	3cm	4cm	Cellules de
-----	-----	-----	-----	-------------

longueurs définies

Nous apprendrons plus loin dans ce chapitre à utiliser des environnements. Cela vous permettra de choisir l'alignement du contenu dans des colonnes à largeur fixe.

Créer une slashbox

Le package `slashbox` permet d'utiliser la commande `\backslashslashbox{Texte dessous}{Texte dessus}`. Cette commande sert à scinder en deux parties triangulaires de même aire une cellule initialement rectangulaire (figure suivante).

Démonstration :

Code : TeX

```
\begin{tabular}{|c|p{1cm}|p{2cm}|}
\hline
\backslashslashbox{Patate}{Carotte} & 1cm & 2cm \\
\hline
\end{tabular}
```

Patate	Carotte	1cm	2cm
--------	---------	-----	-----

Le package slashbox

Changer les séparateurs

Le séparateur habituel lors de la création d'un tableau en LaTeX est le caractère « | ». Il symbolise la création d'une ligne verticale. Il est néanmoins possible d'utiliser de nouveaux séparateurs via les commandes `!{votre séparateur}` ou `@{votre séparateur}`. Ces commandes sont contenues dans le package `array` (comme beaucoup de commandes utiles à la création de tableaux).

La commande `!{ }` a l'avantage, contrairement à `@{ }`, d'insérer une espace avant et après le séparateur. Visuellement, la différence est flagrante (figure suivante) :

Code : TeX

```
\begin{tabular}{|c !{bah} c @{bah}c|}
\hline
texte & texte & texte \\
\hline
\end{tabular}
```

texte bah textebahtexte

Exemple d'utilisation de `!{ }` et `@{ }`

Exercice

Tentez de recréer vous-même le tableau mis en avant sur la figure suivante (le symbole euro est disponible via la commande `\euro{ }`, du package `eurosym`). Il vous faudra bien sûr utiliser les commandes qui viennent d'être introduites.

bonbon	0,50 €
table	14,99 €

Forme de la solution

Solution :

Code : TeX

```
\begin{tabular}{|c| c @{,} c !{\euro{}}|}
\hline
bonbon & 0 & 50 \\
table & 14 & 99 \\
\hline
\end{tabular}
```

Utilisation de commandes et d'environnements dans un tableau

La conception de tableaux telle que nous venons de l'apprendre ne permet pas l'insertion de commandes ou même d'environnements dans les colonnes. Avouons-le, c'est fâcheux. Pour pouvoir agir sur les colonnes, il vous faudra utiliser `<{votre commande ou votre \begin{..}\}}` avant l'instruction de création de colonne (`l`, `r`, `c` ou `\p{}`) et `>{votre commande ou votre \end{..}}` juste après si vous le souhaitez.

Démonstration avec la table de multiplication que nous avons utilisée en début de chapitre (figure suivante) :

Code : TeX

```
\begin{tabular}{|>{\begin{bf}} c <{\end{bf}}|c|c|c|c|c|c|c|c|c|c|}

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

\end{tabular}
```

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

Mise en gras de certaines parties

Colorer des cellules

Nous allons ici apprendre à colorer des lignes, des cellules ainsi que des colonnes. Deux packages sont à utiliser : `color` (dont nous nous sommes déjà servi) et `colortbl`.

Les commandes sont les suivantes :

- `\columncolor{couleur}` pour colorer les colonnes ;
- `\rowcolor{couleur}` pour colorer les lignes ;
- `\cellcolor{couleur}` pour colorer les cellules.

Exercice

Réalisez le tableau visible sur la figure suivante, en colorant la première ligne et la première colonne sur fond jaune (Vous avez le droit de vous servir des codes précédemment vus dans le chapitre.) :

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

Tableau à reproduire

Solution :

Code : TeX

```

\begin{tabular}{>{\begin{bf} \columncolor{yellow}} c
<{\end{bf}}cccccccc}

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

```



Vous constaterez que les tableaux ne sont pas difficiles à réaliser mais demandent de la rigueur lors de la conception.

Un environnement flottant : table

Faire flotter un tableau revient à insérer l'environnement `tabular` dans un nouvel environnement qui, lui, est flottant. Celui-ci s'appelle `table` et son fonctionnement interne est similaire à celui de l'environnement `figure` vu précédemment, avec `caption`, `label`, `center`, etc. Rien ne diffère dans leur fonctionnement.

Tentez de faire **flotter** le tableau vu dans le paragraphe précédent tout en écrivant en légende « Tables de multiplication ». Rappelez-vous de la syntaxe ci-dessous et tout ira bien (figure suivante).

Code : TeX

```

\begin{table}
\begin{center}
\begin{tabular}
Votre tableau
\end{tabular}
\end{center}
\caption{votre légende}
\label{votre référence}

```



```
\end{table}
```

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

Tableau flottant

TABLE 1 – Tables de multiplication

La solution :

Code : TeX

```

\begin{table}[h]
\begin{center}
\begin{tabular}{>{\begin{bf}} \columncolor{yellow}} c
<{\end{bf}} ccccccccc}

\rowcolor{yellow}Multiplié par & \begin{bf}1\end{bf} &
\begin{bf}2\end{bf} & \begin{bf}3\end{bf} & \begin{bf}4\end{bf} &
\begin{bf}5\end{bf} & \begin{bf}6\end{bf} & \begin{bf}7\end{bf} &
\begin{bf}8\end{bf} & \begin{bf}9\end{bf} & \begin{bf}10\end{bf} \\
1 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
2 & 2 & 4 & 6 & 8 & 10 & 12 & 14 & 16 & 18 & 20 \\
3 & 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 & 27 & 30 \\
4 & 4 & 8 & 12 & 16 & 20 & 24 & 28 & 32 & 36 & 40 \\
5 & 5 & 10 & 15 & 20 & 25 & 30 & 35 & 40 & 45 & 50 \\
6 & 6 & 12 & 18 & 24 & 30 & 36 & 42 & 48 & 54 & 60 \\
7 & 7 & 14 & 21 & 28 & 35 & 42 & 49 & 56 & 63 & 70 \\
8 & 8 & 16 & 24 & 32 & 40 & 48 & 56 & 64 & 72 & 80 \\
9 & 9 & 18 & 27 & 36 & 45 & 54 & 63 & 72 & 81 & 90 \\
10 & 10 & 20 & 30 & 40 & 50 & 60 & 70 & 80 & 90 & 100 \\
\end{tabular}
\end{center}
\caption{Tables de multiplication}
\label{mes belles tables}
\end{table}

```

En résumé

- Un tableau nécessite un formalisme et un code propre.
- Un tableau se crée dans l'environnement `tabular`.
- Les cellules d'une même ligne sont séparées par le symbole `&`. Pour passer à la ligne suivante, on insère un retour à la ligne avec `\\`.
- Il est possible d'utiliser les environnements et les commandes de mises en forme de texte au sein d'un tableau.
- Pour rendre un tableau flottant, et laisser LaTeX gérer sa position, il faut l'entourer de l'environnement `table`.

Les mathématiques

Vous venez de survivre aux tableaux, c'était le chapitre le plus difficile de ce cours. Il est maintenant temps d'apprendre à écrire des expressions mathématiques, des formules, des intégrales. C'est le domaine de prédilection de LaTeX.

Nous traiterons dans l'ordre :

- des manières d'introduire des maths dans un document ;
- des équations simples ;
- des systèmes d'équations, ainsi que de tout ce qui ressemble de près ou de loin à une accolade ;
- des matrices et des théorèmes ;
- des flèches et symboles étirables.

Il va de soi que certains d'entre vous n'ont pas les connaissances mathématiques ou physiques nécessaires pour saisir l'utilité d'une partie des outils mathématiques (discriminants, matrices, exponentielles...) et des notations qui leur sont associées. Ne prenez pas peur, ceci n'est pas un cours de maths, mais un cours destiné à vous apprendre à écrire des expressions mathématiques. Il vous suffit de sauter les paragraphes qui ne vous sont pas utiles.

Saisir une expression mathématique

Cette première partie présente les éléments vous permettant d'insérer des expressions mathématiques ou scientifiques au sein d'un texte. Ces commandes et environnements ont pour effet de passer de la police standard du document (Nous avons appris à la modifier dans le chapitre traitant des polices.) à la police définie pour les maths (Il est également possible de la modifier grâce aux packs de polices.).

Ce changement de police est aussi accompagné d'un changement global des règles de typographie. Par exemple, il n'y aura pas la même espace de chaque côté du signe « - » selon que vous êtes dans l'environnement de texte ou dans celui de l'écriture scientifique.

Deux possibilités vous sont offertes par LaTeX. Vous pouvez choisir d'écrire des expressions scientifiques au milieu de votre texte, ou bien le faire à part, dans un espace où elles seront plus lisibles.

Bien sûr, ce serait trop simple s'il n'y avait qu'une seule méthode, nous allons donc en examiner plusieurs tout au long de ce chapitre.

- Pour écrire des expressions mathématiques au sein d'un texte il faudra entourer les expressions par des « \$ », ou utiliser l'environnement `math`. Beaucoup de commandes ne donnent pas le même résultat lorsqu'elles sont utilisées au sein d'un texte. Ce n'est pas une erreur de LaTeX mais une optimisation. Ne vous étonnez donc pas si le rendu d'une formule mathématique est parfois différent lorsque celle-ci est insérée dans un paragraphe et non en dehors.
- Pour écrire des expressions mathématiques en dehors d'un paragraphe, il faudra les entourer par les signes `\[` (à leur commencement) et `\]` (à leur clôture), ou utiliser l'environnement `equation` (Plutôt spécifique aux équations, il permet de numéroté ces dernières.). Il est aussi possible de remplacer chacun de ces signes par `$$`
- Pour écrire du texte dans une expression mathématique, il faudra utiliser la commande `\text{mon bout de texte}`.

N'oublions pas que la bonne lisibilité d'un document met de bonne humeur son lecteur. Nous privilégierons donc la seconde solution : écrire les expressions scientifiques en dehors des paragraphes.

Une démonstration de toutes ces commandes se trouve ci-dessous. Les commandes et environnements sont volontairement noyés dans des paragraphes afin que vous puissiez mieux voir leurs effets (figure suivante).

Code : TeX

```

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Vitae faucibus augue velit ac elit. Aliquam ullamcorper, lectus non
tincidunt pharetra, $x+y+z=4$ nisi sapien ornare felis, ut tristique
tellus justo id ipsum.

```

```

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Vitae faucibus augue velit ac elit. Aliquam ullamcorper, lectus non
tincidunt pharetra, \[x+y+z=4\] nisi sapien ornare felis, ut
tristique tellus justo id ipsum.

```

```

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Vitae faucibus augue velit ac elit. Aliquam ullamcorper, lectus non
tincidunt pharetra, \begin{math}x+y+z=4\end{math} nisi sapien ornare
felis, ut tristique tellus justo id ipsum.

```

```

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Vitae faucibus augue velit ac elit. Aliquam ullamcorper, lectus non
tincidunt pharetra, \begin{equation}x+y+z=4\end{equation} nisi sapien
ornare felis, ut tristique tellus justo id ipsum.

```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vitae faucibus augue velit ac elit. Aliquam ullamcorper, lectus non pharetra, $x + y + z = 4$ nisi sapien ornare felis, ut tristique tellus justo id ipsum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vitae faucibus augue velit ac elit. Aliquam ullamcorper, lectus non tincidunt pharetra,

$$x + y + z = 4$$

nisi sapien ornare felis, ut tristique tellus justo id ipsum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vitae faucibus augue velit ac elit. Aliquam ullamcorper, lectus non tincidunt pharetra, $x + y + z = 4$ nisi sapien ornare felis, ut tristique tellus justo id ipsum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vitae faucibus augue velit ac elit. Aliquam ullamcorper, lectus non tincidunt pharetra,

$$x + y + z = 4 \tag{1}$$

nisi sapien ornare felis, ut tristique tellus justo id ipsum.

Insertion d'équations sommaires

Ces commandes et environnements sont des incontournables et ils savent montrer leur utilité dans de nombreux cas. Les pages suivantes vont introduire un grand nombre d'exemples et d'objets mathématiques.

Fonctions usuelles

Cette partie traite de fonctions assez faciles à écrire. Si quelques-unes d'entre elles sont des commandes qui prennent des arguments (mais n'utilisent pas forcément le `\` habituel), vous pourrez constater que d'autres s'introduisent exactement de la même manière qu'un caractère spécial, comme le cosinus par exemple (*cos* pour les intimes).

Fonctions usuelles à arguments

Nous allons commencer en douceur, avec ci-dessous une série de fonctions qui prennent des arguments, chacune d'elles étant accompagnée d'une capture et d'un exemple. Essayez de les utiliser, elles ne mordent pas.

Fraction

Code : TeX

```

\[\frac{numérateur}{denominateur}\]

```

$$\frac{numérateur}{denominateur}$$

Fraction

*Fractions en cascade (package **amsmath**)*

Code : TeX

```
\[\cfrac{num1}{den1} + \cfrac{num2}{den2}\]
```

$$\frac{num1}{den1 + \frac{num2}{den2}}$$

Fraction complexe

Racine

Code : TeX

```
\[\sqrt[15]{nombre}\]
```

$$\sqrt[15]{nombre}$$

Racine

Limite

Code : TeX

```
\[\lim_{nbr1 \to nbr2} x+42\]
```

$$\lim_{nbr1 \rightarrow nbr2} x + 42 \quad \text{Limite}$$

Exposant

Code : TeX

```
\[nombre^{exposant}\]
```

$$nombre^{exposant} \quad \text{Exposant}$$

Indice

Code : TeX

```
\[nombre_{indice}\]
```

$$nombre_{indice} \quad \text{Indice}$$

Fonctions usuelles sans argument

Le tableau suivant contient des commandes s'utilisant comme des caractères spéciaux. Par exemple, plutôt qu'une hypothétique commande `\cos{argument}` pour le cosinus, il y a le caractère spécial `\cos`, qui peut être suivi de ce que vous voulez (un exposant, un θ ou n'importe quoi d'autre). Ci-dessous, une petite démonstration des possibilités que cela offre, suivi d'un gros tableau (Les caractères grecs sont présentés dans l'annexe A.).

Code : TeX

```
\[ \cos \]  
\[ \cos^{42} (\lambda) \]
```

COS

Cosinus dans tous ses états

$\cos^{42}(\lambda)$

Voilà donc la preuve par l'exemple que l'utilisation du cosinus est alors identique à celle des commandes élémentaires, ce qui rend les choses plus simples, non ?

Fonctions usuelles sans argument			
Nom	Code	Nom	Code
cos	<code>\cos</code>	sin	<code>\sin</code>
arccos	<code>\arccos</code>	arcsin	<code>\arcsin</code>
cos	<code>\cos</code>	sin	<code>\sin</code>
cosh	<code>\cosh</code>	sinh	<code>\sinh</code>
tan	<code>\tan</code>	arg	<code>\arg</code>
arctan	<code>\arctan</code>	ln	<code>\ln</code>
tanh	<code>\tanh</code>	log	<code>\log</code>
exp	<code>\exp</code>	dim	<code>\dim</code>
min	<code>\min</code>	max	<code>\max</code>

Le tableau que voilà contient des commandes permettant d'écrire du texte en caractères romains plutôt qu'en italique mathématique. En effet, s'il existe une commande fondamentale servant à mettre n'importe quel texte en caractères romains, `\mathrm{votre fonction}`, celle-ci est fastidieuse à utiliser, et de bonnes âmes ont créé des raccourcis tels que `\cos` pour vous éviter d'avoir à écrire `\mathrm{cos}` à chaque cosinus que vous employez. Nous utiliserons la commande `\mathrm{fonction}` lorsqu'il nous faudra écrire du texte ou une fonction ne disposant pas d'un tel raccourci.

Les intégrales

Préambule technique

À partir de ce paragraphe, les choses se compliquent un peu (mais pas beaucoup, rassurez-vous). Vous allez faire cohabiter dans vos expressions des symboles tels que des intégrales, des fonctions et plein d'autres petites choses bien sympathiques. Le problème auquel vous aurez à faire face est qu'à force de tout mélanger, vous finirez par ne plus vraiment savoir quels packages appeler.

Alors, soyons simples et allons-y comme des brutes ! Nous allons tout simplement appeler dès le préambule les trois principaux packages nécessaires à l'écriture d'expressions scientifiques, et nous n'aurons plus à nous soucier du package auquel nous faisons appel pour utiliser l'un ou l'autre symbole (Vous imaginez une table de 300 symboles avec à côté de chacun le nom du package ? Ce serait tout simplement ignoble.). Ici, je les ai classés par ordre alphabétique, mais cela n'a strictement aucune importance à notre niveau (Dans une utilisation plus avancée de LaTeX, les modifications apportées par les packages sur les commandes peuvent interférer entre elles (deux packages qui définissent la même commande, par exemple). Généralement, le dernier package introduit a le dernier mot. Ici, nous ne nous intéressons pas à ces cas atypiques.).

Code : TeX

```
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{mathrsfs}
```


Souvent, vous tomberez sur des documentations indiquant des commandes LaTeX sans préciser les packages qui les fournissent. Ces trois-là devraient vous sortir de bien des situations désagréables.

Intégrales et expressions plus complexes

Les commandes simples ayant été introduites, nous allons apprendre à construire des expressions un peu plus ambitieuses. Nous allons maintenant nous pencher sur la rédaction de formules comportant des intégrales (simples, doubles, triples et même plus !), des sommes ou des produits.

Un exemple étant beaucoup plus facile à comprendre qu'un long discours, j'ai dressé pour vous une liste vous présentant des formules générales ainsi que des utilisations de celles-ci. Vous serez confronté à la commande `\` suivie d'une espace : elle sert simplement à insérer une espace dans un environnement mathématique.



Remarquez qu'il existe des commandes dérivées de `\int{ }` (la commande pour le symbole d'intégration) permettant de générer des intégrales doubles ou triples (respectivement, `\iint{ }` et `\iiint{ }`). Celles-ci fonctionnent de la même manière que `\int{ }`.

Intégrale simple

Modèle

Code : TeX

```
\[ \int {contenu} \]  
\[ \int_{borne inférieure}^{borne supérieure} {contenu} \]  
\[ \int \limits_{borne inférieure}^{borne supérieure} {contenu} \]
```

Démonstration

Code : TeX

```
\[ \int {x^2 dx} \]  
\[ \int_1^3 {x^2 dx} \]  
\[ \int \limits_1^3 {x^2 dx} \]
```

$$\int x^2 dx$$

$$\int_1^3 x^2 dx$$

Intégrale simple

$$\int \limits_1^3 x^2 dx$$

*Intégrale sur une courbe fermée***Modèle****Code : TeX**

```
\[\oint {contenu}\]
```

Démonstration**Code : TeX**

```
\[\oint {x^2\ dx}\]
```

$$\oint x^2 dx$$

Intégrale curviligne

*Symbole de la somme***Modèle****Code : TeX**

```
\[\sum_{en dessous}^{au dessus} contenu\]
```

Démonstration**Code : TeX**

```
\[\sum_{k=2}^{47} k+1\]
```

$$\sum_{k=2}^{47} k + 1$$

Somme

Symbole du produit

Modèle**Code : TeX**

```
\[\prod_{\text{en dessous}}^{\text{au dessus}} \text{contenu}\]
```

Démonstration**Code : TeX**

```
\[\prod_{k=2}^{47} k+1\]
```

$$\prod_{k=2}^{47} k + 1$$

Produit

En réalité, les intégrales doubles et triples peuvent également s'écrire avec des intégrales simples à l'intérieur d'autres intégrales simples. Vous trouverez des exemples ci-dessous, illustrant des cas plus ou moins exotiques.

Code : TeX

```
\[\int\int {x^2+y\ dx\ dy}\]\]
\[\int\int_1^3 {x^2+y\ dx\ dy}\]\]
\[\int\int\limits_1^3 {x^2+y\ dx\ dy}\]\]
\[\int_0^4\{\int_1^3 {x^2+y\ dx\ dy}\}\]\]
\[\int\limits_0^4 {\int\limits_1^3 {x^2+y\ dx\ dy}}\]\]
\[\int\int\int {x^2+y+z\ dx\ dydz}\]\]
\[\int_{x=0}^{x=5}\{\int_{y=0}^{y=4}\{\int_{z=1}^{z=3} {x^2+y+z\ dx\ dydz}\}\}\]\]
\[\int\limits_{x=0}^{x=5} {\int\limits_{y=0}^{y=4} {\int\limits_{z=1}^{z=3} {x^2+y+z\ dx\ dydz}}}\]\]
```

$$\int \int x^2 + y \, dx dy$$

$$\int \int_1^3 x^2 + y \, dx dy$$

$$\int \int_1^3 x^2 + y \, dx dy$$

Intégrales doubles

$$\int_0^4 \int_1^3 x^2 + y \, dx dy$$

$$\int_0^4 \int_1^3 x^2 + y \, dx dy$$

Si le besoin d'écrire plusieurs lignes sous un opérateur se fait sentir, utilisez la commande `\substack{ }`. Il faut alors séparer chaque ligne par les habituels « `\\` ».

Code : TeX

```
\[ \sum_{\substack{k=0 \\ i=0 \\ j=0}}^n i+j+k \]
```

```
\[ \prod_{\substack{k=0 \\ i=0 \\ j=0}}^n i+j+k \]
```

$$\sum_{\substack{k=0 \\ i=0 \\ j=0}}^n i + j + k$$

La commande `\substack{ }`

$$\prod_{\substack{k=0 \\ i=0 \\ j=0}}^n i + j + k$$

Les systèmes d'équations

Les systèmes d'équations sont simples à mettre en place : ils requièrent l'utilisation d'environnements ayant des fonctionnalités

similaires à celles de `tabular`, si ce n'est qu'ils mettent en forme... des équations.

L'environnement `eqnarray`

Notre choix va ici se porter sur l'environnement `eqnarray`, qui offre la possibilité de créer un système d'équations proprement aligné, grâce au caractère `&`. Notez que l'environnement `eqnarray` numérote les équations, mais que son homologue `eqnarray*` ne le fait pas.

Démonstration :

Code : TeX

```
\begin{eqnarray}
y &=& x - y + z \\
x &=& y \\
z &=& y
\end{eqnarray}

\begin{eqnarray*}
y &=& x - y + z \\
x &=& y \\
z &=& y
\end{eqnarray*}

\begin{eqnarray}
y &=& x - y + z \\
x &=& y \\
z &=& y
\end{eqnarray}
```

$$y = x - y + z \quad (1)$$

$$x = y \quad (2)$$

$$z = y \quad (3)$$

$$y = x - y + z$$

$$x = y$$

$$z = y$$

Systèmes d'équations

$$y = x - y + z \quad (4)$$

$$x = y \quad (5)$$

$$z = y \quad (6)$$

Vous ne manquerez pas de remarquer que la numérotation des équations se poursuit le long du document. C'est très pratique pour guider vos lecteurs, et un lecteur heureux est un lecteur qui vous aime. Important, n'est-ce pas ?

L'environnement `align`

Sans doute aurez-vous souvent recours à l'environnement `eqnarray`. Néanmoins, celui-ci pose quelques problèmes d'alignement et d'espacement dans des cas particuliers. Il est alors plus adapté de choisir l'environnement `align` (Vous vous demandez sûrement pourquoi je vous ai d'abord présenté une notion « bancal »). C'est tout simplement parce que vous la croiserez de façon extrêmement fréquente dans des codes sources LaTeX.). La commande `align` fonctionne de la même manière que la commande `eqnarray`; de même, vous avez la possibilité d'utiliser `align*`, il y a juste une petite différence dans la syntaxe que vous pouvez remarquer dans le code source ci-dessous.

Code : TeX

```
\begin{align}
y &= x - y + z \\
x &= y \\
z &= y
\end{align}

\begin{align*}
y &= x - y + z \\
x &= y \\
z &= y
\end{align*}

\begin{align}
y &= x - y + z \\
x &= y \\
z &= y
```

```
\end{align}
```

Les grands acharnés peuvent lire [ce document](#). Il développe les différences entre `eqnarray` et `align`.

L'environnement `array`











Notre prochaine mission est de positionner, dans un premier temps, des parenthèses autour de notre système d'équations et dans un second temps, une accolade ouvrante à son extrémité gauche. Pour ce faire, nous allons utiliser la syntaxe suivante (nous utilisons ici `array` pour aligner les équations et choisir l'alignement de chaque colonne) :

Code : TeX

```
\[
\left code_du_délimiteur_de_gauche
\begin{array}{r c l}
y &=& x - y + z \\
x &=& y \\
z &=& y
\end{array}
\right code_du_délimiteur_de_droite
\]
```

Pour pouvoir installer nos parenthèses et notre accolade, il nous faut jeter un œil au tableau des délimiteurs contenant les codes correspondant à chacun d'entre eux (le point correspond à une absence de délimiteur lors de l'utilisation de `\right` et `\left`).

Code	Rendu	Code	Rendu
Les délimiteurs			
(()))
[[]]]
\{	{ }	\}	}
\uparrow	↑	\downarrow	↓
\updownarrow	↕	\Uparrow	↗
\Downarrow	↘	\Updownarrow	↕

<code>\lfloor</code>		<code>\rfloor</code>	
<code>\lceil</code>		<code>\rceil</code>	
<code>\langle</code>		<code>\rangle</code>	
<code>/</code>		<code>\backslash</code>	
<code> </code>		<code>\ </code>	

Vous avez l'exercice, vous avez les codes des délimiteurs... À vous de jouer !

La solution :

Code : TeX

```

\left(
\begin{array}{r c l}
y &=& x - y + z \\
x &=& y \\
z &=& y
\end{array}
\right)

\left\{
\begin{array}{r c l}
y &=& x - y + z \\
x &=& y \\
z &=& y
\end{array}
\right.

```

$$\begin{pmatrix} y & = & x - y + z \\ x & = & y \\ z & = & y \end{pmatrix}$$

Solution

$$\begin{cases} y & = & x - y + z \\ x & = & y \\ z & = & y \end{cases}$$

Les matrices

Vous vous figurez sans doute qu'une matrice — ou l'expression de son déterminant — se conçoit à l'aide de la commande `array`, eh bien c'est raté ! Cela est certes possible, mais des environnements ont été spécialement conçus pour vous aider à écrire des matrices dans vos documents.

Les différents environnements

Vous n'allez pas être dépaycé : la syntaxe est toujours la même que celle que l'on connaît pour les tableaux, la seule chose qui change est le nom des différents environnements. À nouveau, nous allons passer en revue la liste des différentes options qui vous sont proposées.

L'environnement `matrix`

Code : TeX

```
\[
\begin{matrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{matrix}
\]
```

$$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$$

L'environnement `pmatrix`

Code : TeX

```
\[  
\begin{pmatrix}  
1 & 2 & 3 \\  
4 & 5 & 6 \\  
7 & 8 & 9 \\  
\end{pmatrix}  
\]
```

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

L'environnement `vmatrix`

Code : TeX

```
\[  
\begin{vmatrix}  
1 & 2 & 3 \\  
4 & 5 & 6 \\  
7 & 8 & 9 \\  
\end{vmatrix}  
\]
```

$$\begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix}$$

L'environnement `Vmatrix`

Code : TeX

```
\[  
\begin{Vmatrix}  
1 & 2 & 3 \\  
4 & 5 & 6 \\  
7 & 8 & 9 \\  
\end{Vmatrix}  
\]
```

$$\left\| \begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array} \right\|$$

L'environnement `bmatrix`

Code : TeX

```
\[  
\begin{bmatrix}  
1 & 2 & 3 \\  
4 & 5 & 6 \\  
7 & 8 & 9 \\  
\end{bmatrix}  
\]
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

L'environnement *Bmatrix*

Code : TeX

```
\[
\begin{Bmatrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{Bmatrix}
\]
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Exemple pratique

Petite application de l'environnement `matrix` et de ses pairs, voici un exemple de matrice contenant des séries de trois points horizontaux (commande `\cdots`), verticaux (commande `\vdots`) ou diagonaux (commande `\ddots`). Effet garanti (Vous pouvez vous servir de ces trois commandes dans d'autres contextes si vous le souhaitez).

Code : TeX

```
\[
\begin{pmatrix}
a_{11} & \cdots & a_{1k} \\
\vdots & \ddots & \vdots \\
a_{k1} & \cdots & a_{kk}
\end{pmatrix}
\]
```

$$\begin{pmatrix} 1 & \dots & k \\ \vdots & \ddots & \vdots \\ k & \dots & n \end{pmatrix}$$

Matrice et dots

La commande ``

Voici une autre petite astuce : la commande ``, qui permet d'insérer un espace de la longueur du texte contenu entre ses crochets. Dans le contexte des matrices, elle est très utile, car elle permet d'insérer des blancs et d'aligner les différents éléments. Cette commande n'est cependant pas spécifique aux matrices : elle peut servir dans beaucoup d'autres cas.

Code : TeX

```
\[ % Sans \phantom{}
\begin{Vmatrix}
1 & 12345 & 3 \\
94 & 5 & -6 \\
7 & 8 & 9
\end{Vmatrix}
\]

\[ % Avec \phantom{}
\begin{Vmatrix}
\phantom{9}1 & 12345 & \phantom{-}3 \\
94 & \phantom{1234}5 & -6 \\
\phantom{9}7 & \phantom{1234}8 & \phantom{-}9
\end{Vmatrix}
\]
```

$$\left| \begin{array}{ccc} 1 & 12345 & 3 \\ 94 & 5 & -6 \\ 7 & 8 & 9 \end{array} \right|$$

La commande `phantom{}`

$$\left| \begin{array}{ccc} 1 & 12345 & 3 \\ 94 & 5 & -6 \\ 7 & 8 & 9 \end{array} \right|$$

Mise en forme de théorèmes, lemmes...

À l'instar des citations, les théorèmes, corollaires, lemmes et autres affreuses petites choses écrites par de vieux barbus doivent respecter leurs conventions typographiques. Nous allons apprendre à mettre tout cela en place (eh non, vos souffrances ne sont pas terminées).

Généralités



Tout d'abord, clarifions un point. Un abus de langage assez affreux consiste à dire qu'on « écrit un théorème » à chaque fois que l'on emploie la syntaxe dont je vais vous parler dans les lignes qui suivent. Si vous êtes physicien, personne ne vous en voudra (La preuve, c'est que je suis physicien, et je ne vous en voudrai pas.). En revanche, si vous faites des maths, évitez de dire à tout va « j'écris un théorème » quand vous mettez en page un corollaire : vous éviterez ainsi les multiples railleries de vos fourbes collègues.

Revenons à nos moutons ! Nous cherchons à mettre en page différentes choses : des corollaires, des lois, des théorèmes... Il faut bien sûr que chaque type d'entité ait une numérotation qui lui soit propre au sein du document (si dans une page, vous tapez un théorème et un corollaire, le théorème devra s'appeler « théorème 1 » et le corollaire « corollaire 1 », pas « corollaire 2 »).

Petit plus, nous souhaiterions pouvoir numéroter les différentes entités par rapport aux chapitres, sections et autres éléments structuraux. Ainsi, si un théorème est le troisième du chapitre cinq, il portera le nom de « théorème 5.3 » (même idée pour les sections et les autres éléments de structure).

Tout ceci se fait très facilement grâce à LaTeX, mais en deux temps. Dans un premier temps, il nous faudra indiquer à LaTeX dans le préambule qu'il faut créer les différentes entités utilisant la mise en page dédiée au théorème. Dans le corps du texte, nous pourrons ensuite simplement utiliser la commande permettant d'inclure des théorèmes, des lois et d'autres joyeusetés.

Pratiquons !

Si vous avez lu les paragraphes précédents, vous avez dû garder en tête la méthode en deux temps. La première chose à faire est donc d'appeler dans le préambule la commande `\newtheorem` (déclinée de plusieurs façons) avant d'utiliser un nom

d'environnement de votre choix dans la suite du document.

Code : TeX

```

\documentclass{book}

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

\usepackage{amsthm}
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{mathrsfs}

\newtheorem*{petit_nom}{Lemme} % La petite étoile enlève la
numérotation, maisnécessite le package amsthm
\newtheorem{petit_nom1}{Proposition}
\newtheorem{petit_nom2}{Définition}[chapter] % Le [chapter] peut par
exemple êtreremplacé par [section], il permet de numéroté les
éléments par rapport aux numéros de chapitre
\newtheorem{petit_nom3}{Définition}[section]

\begin{document}
\chapter{Les ours}
\chapter{Les lapins}
\section{les lapins nains}

\begin{petit_nom}
ils aiment les carottes
\end{petit_nom}

\begin{petit_nom}[des lapins] % Argument optionnel, j'ajoute le nom
de la loi entre crochets
ils aiment les carottes
\end{petit_nom}

\begin{petit_nom1}
ils aiment les carottes
\end{petit_nom1}

\begin{petit_nom1}[des lapins] % Argument optionnel, j'ajoute le
nom entre crochets
ils aiment les carottes
\end{petit_nom1}

\begin{petit_nom2}
ils aiment les carottes
\end{petit_nom2}

\begin{petit_nom2}[des lapins] % Argument optionnel, j'ajoute le
nom entre crochets
ils aiment les carottes
\end{petit_nom2}

\section{les autres}

\begin{petit_nom3}
ils aiment les carottes
\end{petit_nom3}

\begin{petit_nom3}[des lapins] % Argument optionnel, j'ajoute le
nom entre crochets
ils aiment les carottes
\end{petit_nom3}

\end{document}

```

Chapitre 2

Les Lapins

2.1 les lapins nains

Lemme. *ils aiment les carottes*

Lemme (des lapins). *ils aiment les carottes*

Les

Proposition 1. *ils aiment les carottes*

Proposition 2 (des lapins). *ils aiment les carottes*

Définition 2.1. *ils aiment les carottes*

Définition 2.2 (des lapins). *ils aiment les carottes*

2.2 les autres

Définition 2.2.1. *ils aiment les carottes*

Définition 2.2.2 (des lapins). *ils aiment les carottes*

théorèmes

L'exemple est très explicite : la façon dont vous utiliserez la commande `\newtheorem` est importante, mais il n'y a rien de plus à savoir utiliser. Sachez que dans de nombreux cas, des commandes complexes en LaTeX vous seront présentées par le biais d'un code commenté et rien d'autre. Prenez le temps de le décortiquer : vous verrez alors que tout deviendra plus simple (enfin, tout ce qui concerne LaTeX). C'est la raison pour laquelle tout ce qui est nécessaire à la compréhension de ce concept est intégré au code précédent et ne figure pas dans un paragraphe explicatif.

Flèches, symboles étirables et espaces

Cette avant-dernière partie recense de nombreuses commandes à la fois paramétrables et utiles en LaTeX. Cela va des différents types d'espaces aux accolades, en passant par les flèches et les accents.

Ici ne sont listées **que** les commandes paramétrables, toutes les autres se trouvent dans la partie mathématique de l'annexe traitant des caractères spéciaux (sinon, ce chapitre deviendrait vraiment trop long).

Les espaces

Différents types d'espaces sont disponibles dans un environnement mathématique. Notez que l'une d'elles est négative et permet de rapprocher des éléments. Un exemple de rendu est montré à la figure suivante.

Espace	Code
négative	<code>\!</code>
fine	<code>\,</code>
normale	<code>\</code> suivi d'une espace
moyenne	<code>\;</code>
large	<code>\:</code>
cadratin	<code>\quad</code>
double cadratin	<code>\qquad</code>

Code : TeX

```

\ [1\!2\]
\ [1\,2\]
\ [1\ 2\]
\ [1\;2\]
\ [1\:2\]
\ [1\quad2\]
\ [1\qquad2\]

```

12

1 2

1 2

1 2

Les espaces

1 2

1 2

1 2

Les accents et chapeaux divers

Les expressions scientifiques imposent parfois de coiffer une lettre ou une série de lettres de flèches (pour représenter des vecteurs) ou de divers accents (Dans le cas de `\stackrel`, nous remplacerons l'accent par une série de lettres.). Voici un tableau les présentant.

Code	Rendu
<code>\hat{a}</code>	\hat{a}
<code>\acute{a}</code>	\acute{a}
<code>\bar{a}</code>	\bar{a}
	.

<code>\dot{a}</code>	\dot{a}
<code>\breve{a}</code>	\breve{a}
<code>\check{a}</code>	\check{a}
<code>\grave{a}</code>	\grave{a}
<code>\vec{a}</code>	\vec{a}
<code>\ddot{a}</code>	\ddot{a}
<code>\tilde{a}</code>	\tilde{a}
<code>\overleftarrow{aze}</code>	\overleftarrow{aze}
<code>\overrightarrow{aze}</code>	\overrightarrow{aze}
<code>\overbrace{aze}</code>	\overbrace{aze}
<code>\underbrace{aze}</code>	\underbrace{aze}
<code>\overline{aze}</code>	\overline{aze}
<code>\underline{aze}</code>	\underline{aze}
<code>\stackrel{dessus}{dessous}</code>	$\stackrel{dessus}{dessous}$
<code>\widehat{aze}</code>	\widehat{aze}

<code>\widetilde{aze}</code>	\widetilde{aze}	
<code>\xrightarrow[dehors]{dessus}</code>	$\xrightarrow[dehors]{dessus}$	
<code>\xleftarrow[dehors]{dessus}</code>	$\xleftarrow[dehors]{dessus}$	
<code>\overset{a}{X}</code>	$\overset{a}{X}$	Commande Overset
<code>\underset{b}{X}</code>	$\underset{b}{X}$	Commande underset
<code>\overset{a}{\underset{b}{X}}</code>	$\overset{a}{\underset{b}{X}}$	Mix entre les deux fonctions
<code>\overbrace{\sin^2(x) + \cos^2(x)}^{\text{Merveilleux}}=1</code>	$\overbrace{\sin^2(x) + \cos^2(x)}^{\text{Merveilleux}} = 1$	Commande overbrace
<code>\underbrace{\sin^2(x) + \cos^2(x)}_{\text{Épique}}=1</code>	$\underbrace{\sin^2(x) + \cos^2(x)}_{\text{Épique}} = 1$	Commande underbrace

En résumé

- Une expression mathématique peut être insérée au sein d'un paragraphe (il faut écrire `$expression$`) ou en dehors d'un paragraphe (il faut cette fois écrire `\[expression\]`).
- On utilise ensuite des commandes spécifiques pour mettre en forme les formules. Ainsi, `\frac{numérateur}{dénominateur}` permet de créer une fraction.
- Pour écrire un système d'équations, on utilise l'environnement `eqnarray` ou `align`.
- Il existe des centaines de caractères spéciaux dédiés à l'écriture d'expressions scientifiques. Reportez-vous à l'annexe de ce tutoriel pour en avoir la liste.

Sommaire et index

Prenons le temps de faire un bilan : les tableaux, les images, la mise en forme du texte et des pages... Nous serions tentés de nous dire que tous les grands domaines de LaTeX ont été abordés. Eh bien non, il nous en manque encore deux, qui nous occuperont pendant deux courts chapitres. Je pense ici aux différents types de tables d'index ainsi qu'aux bibliographies. Ces deux éléments, bien que ne représentant pas une difficulté insurmontable, sont le cauchemar de bon nombre d'étudiants et une intarissable source de blocage. Ne vous inquiétez pas, nous allons apprendre à utiliser simplement les tables et index dans ce chapitre (Les bibliographies seront pour le chapitre suivant.). Préparez vos neurones !

Nous aborderons ici les tables de tableaux, de figures et bien sûr les tables des matières. Nous apprendrons aussi les mécanismes élémentaires nécessaires à la création d'un index simple.

Tables des matières

Les tables sont censées aider le lecteur à mieux se repérer dans vos publications. Leur propreté (et accessoirement la vôtre) sera pour le correcteur ou le lecteur un gage de sérieux.

Dans de nombreux cas en édition, ce qui est jugé est votre table des matières. Mal la présenter revient à vous tirer, dès le départ, une balle dans le pied. C'est pour cela que tout projet sérieux requiert une table des matières réfléchie et bien fournie. Il est aussi parfois nécessaire d'enlever des éléments superflus à une table des matières, ou de ne pas aller trop loin dans le détail.

Imaginez un document où vous présentez à la fois vos chapitres, sections, sous-sections et paragraphes dans votre table des matières. Ne serait-ce pas un peu trop ? Nous apprendrons ici à enlever certains éléments de la table en demandant à LaTeX de ne pas y inclure les éléments hiérarchiques de trop bas niveau.

Résumons en trois mots les qualités d'une bonne table des matières :

- cohérence ;
- pertinence (vis-à-vis du contenu réel de la publication) ;
- propreté.

Table des matières simple

Avant de passer aux choses très subtiles, nous devons apprendre à inclure une table des matières dans un document. Pour ce faire, il vous suffit d'insérer la commande `\tableofcontents` à l'endroit où vous souhaitez afficher une table des matières dans votre document.

Je vous assure, ça n'a rien de plus compliqué. En une ligne de code, vous disposerez d'une table conçue avec les paramètres par défaut de LaTeX, ce qui suffit à la majorité des petits rapports. La seule subtilité réside dans le fait qu'il faille compiler deux fois. La première permet à LaTeX de comprendre la structure du document et de lister les titres. La seconde permet la mise en forme de la table, avec les numéros de pages.

Petite démonstration sur le code suivant (figure suivante) :

Code : TeX

```
\documentclass{book}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[français]{babel}
\begin{document}

\part{Partie 1}
\chapter{Chapitre 1}
\section*{Une section}
\section{Une section avec un nom méga
\mmligne mais alors vraiment méga trop giga long qui dépasse}
\subsection{Une sous-section}
\subsection{Une sous-section}
\chapter{Chapitre 2}
\section{Une section}
\subsection{Une sous-section}
\paragraph{Un paragraphe}
\subsection{Une sous-section}
\paragraph{Un paragraphe}
```



```

\section{Une section}
\chapter{Chapitre 3}
\section{Une section}
\section{Une section}

\tableofcontents

\end{document}

```

Table des matières

I	Partie 1	1
1	Chapitre 1	3
1.1	Une section avec un nom méga mais alors vraiment méga trop giga long qui dépasse	3
1.1.1	Une sous section	3
1.1.2	Une sous section	3
2	Chapitre 2	5
2.1	Une section	5
2.1.1	Une sous section	5
2.1.2	Une sous section	5
2.2	Une section	5
3	Chapitre 3	7
3.1	Une section	7
3.2	Une section	7

Table des matières

La table obtenue apparaît après deux compilations et plusieurs éléments sont à noter :

- il manque les paragraphes ;
- le titre de l'une des sections est trop long ;
- il manque une section ;
- vous aimeriez appeler cette table des matières « Sommaire » et la placer en tête de document.

La seule de ces cinq choses que nous sachions faire pour l'instant est de mettre en tête de document la table des matières. Il suffit d'écrire `\tableofcontents` après la commande de création d'une éventuelle page de garde, ou encore juste après votre `\begin{document}`.

Examinons de plus près les autres éléments.

Paramétrage d'une table des matières

Appeler votre table « Sommaire »

Avec les packages francisant votre document, le nom par défaut de votre table des matières est « Table des matières ». Ce nom me semble peu adapté si la table est présente en début de manuscrit. L'appellation « Sommaire » peut paraître plus appropriée.

Voici la commande :

Code : TeX

```
\renewcommand{\contentsname}{Votre nouveau titre} % Dans le corps du
document, avant la commande \tableofcontents.
```

La figure suivante montre le résultat avec le mot « Sommaire ».

Sommaire

I	Partie 1	1
1	Chapitre 1	3
1.1	Une section avec un nom méga mais alors vraiment méga trop giga long qui dépasse	3
1.1.1	Une sous section	3
1.1.2	Une sous section	3
2	Chapitre 2	5
2.1	Une section	5
2.1.1	Une sous section	5
2.1.2	Une sous section	5
2.2	Une section	5
3	Chapitre 3	7
3.1	Une section	7
3.2	Une section	7

Changement de titre

La méthode développée ici permet tout aussi bien de changer le nom en « Sommaire » qu'en autre chose. Elle est applicable aussi à peu près à tous les éléments de structure d'un document (pour changer l'appellation « Partie » en autre chose, il vous faudra remplacer `\contentsname` par `\partname` par exemple.).

Raccourcir une ligne

Nous pouvons voir que dans notre sommaire, une des lignes est trop longue et dépasse. Heureusement, la communauté LaTeX a pensé à permettre aux utilisateurs de définir deux titres différents lors de la création d'un élément de structure : l'un pour le document, l'autre pour la table des matières.

La commande `\section{titre dans le document}` se transforme en `\section[titre dans la table des matières]{titre dans le document}`.

Dans le cas qui nous intéresse, remplaçons `\section{Une section avec un nom méga mais alors vraiment méga trop giga long qui dépasse}` par `\section[un titre plus court]{ Une section avec un nom méga mais alors vraiment méga trop giga long qui dépasse}` et observons le résultat sur la figure suivante.

Sommaire

I	Partie 1	1
1	Chapitre 1	3
1.1	un titre plus court	3
1.1.1	Une sous section	3
1.1.2	Une sous section	3
2	Chapitre 2	5
2.1	Une section	5
2.1.1	Une sous section	5
2.1.2	Une sous section	5
2.2	Une section	5
3	Chapitre 3	7
3.1	Une section	7
3.2	Une section	7

Les titres courts

Magnifique, n'est-ce pas ? Et cela fonctionne avec tous les éléments de structure.

Inclure les paragraphes dans le sommaire

Par défaut, LaTeX n'inclut pas les titres des paragraphes dans la table des matières de la classe `book`. Nous souhaiterions qu'ils apparaissent.

Pour ce faire, nous allons nous dire que chaque élément de hiérarchie correspond à un chiffre. Nous démarrons par les parties à 1 (je sais, c'est bizarre), puis les chapitres à 0 et ainsi de suite, cela nous donnant un tableau sympathique.

Élément hiérarchique	Nombre
Partie	-1
Chapitre	0
Section	1
Sous-section	2
Sous-sous-section	3
Paragraphe	4
Sous-paragraphe	5

La commande qui nous intéresse est `\setcounter{tocdepth}{Nombre de votre choix}`. Elle sert à demander à LaTeX de composer une table des matières contenant tous les éléments hiérarchiques jusqu'à ceux définis par le nombre que vous aurez choisi. Si vous choisissez -1 il n'y aura que les parties, si vous choisissez 5 il y aura tout. Tentons le 4 pour créer une table des matières comportant les titres des paragraphes :

Code : TeX

```
\setcounter{tocdepth}{4}
```

Le résultat est visible sur la figure suivante.

Table des matières

I	Partie 1	1
1	Chapitre 1	3
1.1	un titre plus court	3
1.1.1	Une sous section	3
1.1.2	Une sous section	3
2	Chapitre 2	5
2.1	Une section	5
2.1.1	Une sous section	5
	un paragraphe	5
	un paragraphe	5
	un paragraphe	5
2.1.2	Une sous section	5
	un paragraphe	5
	un paragraphe	5
	un paragraphe	5
2.2	Une section	5
3	Chapitre 3	7
3.1	Une section	7
3.2	Une section	7

Utilisation de `\setcounter`

Inclure un élément de structure fictif

Lorsqu'un élément de structure est suivi de `*` (par exemple `\chapter*{un chapitre}`), il n'y a pas de référence vers celui-ci dans la table des matières. Néanmoins, il est aussi possible de forcer LaTeX à ajouter une entrée.

Le `{toc}` est un élément que nous ne développerons pas, dites-vous juste que sans lui, la compilation peut échouer. La commande est `\addcontentsline{toc}{élément de structure sans son \ ni ses accolades et crochets}{titre apparaissant dans le sommaire}`.

Petit exemple (figure suivante):

Code : TeX

```
\documentclass{book}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[français]{babel}
\begin{document}
```

```

\addcontentsline{toc}{part}{Le Monde}
\addcontentsline{toc}{chapter}{L'Eurasie}
\addcontentsline{toc}{section}{L'Europe}
\addcontentsline{toc}{subsection}{La France}
\addcontentsline{toc}{subsubsection}{L'Aquitaine}
\addcontentsline{toc}{paragraph}{La Gironde}
\addcontentsline{toc}{subparagraph}{Bordeaux}

\setcounter{tocdepth}{5}
\tableofcontents
\end{document}

```

Table des matières

Le Monde	1
L'Eurasie	1
L'Europe	1
La France	1
L'aquitaine	1
La Gironde	1
Bordeaux	1

Éléments de structure fictifs

Chose importante ici : étant donné que la commande `\addcontentsline` n'est qu'un marqueur et ne représente rien de réel dans le document, il est normal que le sommaire considère que tous les éléments sont en première page. Vous pouvez bien sûr les placer où vous le souhaitez dans votre document selon vos besoins.

Tables des figures et tableaux

Les tables des figures et des tableaux permettent au lecteur de mieux se retrouver dans un texte. Ces deux tables ne sont pas incontournables, mais suivent les mêmes mécanismes, à peu de chose près, que les tables des matières.

Elles s'insèrent respectivement via les commandes `\listoffigures` et `\listoftables` ; elles font apparaître, au choix, soit un titre spécialement conçu pour la table, soit la légende contenue dans `\caption`. Nous n'avons pas encore appris à indiquer des titres conçus pour les tables dans la commande `\caption`, pourtant c'est très simple. Les deux syntaxes possibles ci-dessous vous sont présentées.

Code : TeX

```

\caption[titre court]{légende normale de l'image}
\caption{légende normale de l'image}

```

Vous vous attendiez à quelque chose de plus compliqué ? Désolé de vous décevoir.



Ne vous trompez pas lors de la compilation. N'oubliez pas qu'il faut compiler deux fois. J'ai reçu de nombreux messages de personnes pensant être face à un bug de compilation dans leurs tables, alors qu'elles avaient simplement oublié la seconde compilation (Il y avait aussi quelques cas de mauvaise installation sur Windows).

Je vous ai fabriqué un exemple massif (figure suivante), mais qui illustre bien la manière de créer des tables recensant les tableaux ainsi que les figures. Prenez le temps de le décortiquer, il n'est pas très difficile, mais très utile. J'ai réutilisé ici l'image de Poulpy, mascotte officielle du livre.

Code : TeX

```

\documentclass[12pt]{report}

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

\usepackage{graphicx}
\begin{document}

\begin{table}
\begin{center}
\begin{tabular}{|c|c|}
\hline
1 & 2 \\
\hline
3 & 4 \\
\hline
\end{tabular}
\end{center}
\caption[Un tableau]{Mon beau tableau}
\end{table}

\begin{figure}
\begin{center}
\includegraphics{poulpy.png}
\end{center}
\caption{Poulpy est multicolore}
\end{figure}

\begin{figure}
\begin{center}
\includegraphics{poulpy.png}
\end{center}
\caption[Chatoyante]{Poulpy est chatoyante}
\end{figure}

\begin{figure}
\begin{center}
\includegraphics{poulpy.png}
\end{center}
\caption{Poulpy est inestimable}
\end{figure}

\begin{table}
\begin{center}
\begin{tabular}{|c|c|}
\hline
1 & 2 \\
\hline
3 & 4 \\
\hline
\end{tabular}
\end{center}
\caption{Mon beau tableau}
\end{table}

\begin{figure}
\begin{center}
\includegraphics{poulpy.png}
\end{center}
\caption[Poulpesque]{Poulpy est poulpesque}

```



```

\end{figure}

\listoftables
\listoffigures

\end{document}

```

Liste des tableaux

1	Un tableau	1
2	Mon beau tableau	1

Table des figures et Table des

Table des figures

1	Poulpy est multicolore	1
2	Chatoyante	2
3	Poulpy est inestimable	2
4	Poulpesque	3

tableaux

Les index

À mon sens, les index constituent la limite en difficulté à aborder dans ce livre, et n'ont finalement pas un intérêt énorme dans des rapports ou de petites publications. J'ai tout de même choisi de vous en parler un peu, ne serait-ce que par souhait de proposer une ouverture.

Prenez conscience tout de même que je vous expose ici la méthode permettant d'élaborer des index élémentaires. Les choses peuvent vite devenir très difficiles à mettre en place.

En résumé, je vous explique ici le concept pour que vous ne soyez pas paniqué plus tard si vous devez approfondir la chose, mais nous ne nous y attarderons pas (Beaucoup de documentations n'abordent même pas le sujet.).

Théorie

Commençons par les choses simples. La création d'un index nécessite le package `makeidx` et d'insérer la commande `\makeindex` dans le préambule, puis `\printindex` à l'endroit où vous souhaitez afficher celui-ci (exactement comme vous l'avez fait pour les tables).

Pour mettre en place les entrées dans l'index, il faut les marquer avec la commande `\index{argument}` en collant celle-ci au mot concerné. C'est à partir d'ici que les choses se gâtent.

Code : TeX

```
J'aime les patates\index{patate}.
```

Après trois compilations, vous vous retrouvez normalement avec un index indiquant la position du marqueur « patate » dans

votre document.



Les utilisateurs de TeXnicCenter n'auront besoin de compiler que trois fois pour créer un index. Néanmoins, ce n'est pas aussi simple pour tout le monde. Les utilisateurs de Linux devront compiler une fois avant de taper `makeindex nom_du_fichier.tex` dans leur console (ou cliquer sur Build → Compile → Makeindex dans Kile) et compiler de nouveau.

Les utilisateurs de TeXShop trouveront quant à eux la commande « Makeindex » dans l'un de leurs menus déroulants. Ils devront eux aussi compiler normalement une fois, sélectionner « Makeindex » et compiler dans ce mode, puis compiler à nouveau en mode LaTeX.

Si nous insérons un marqueur « pâté » et un marqueur « patate », LaTeX placerait dans l'ordre alphabétique « pâté » avant « patate ». Difficile à croire, et pourtant (figure suivante)...

Code : TeX

```
J'aime le pâté\index{pâté} et les patates\index{patate}.
```

Index

Index simple. Petit problème d'ordre alphabétique...

pâté, 1
patate, 1

L'idée derrière tout ça c'est que LaTeX a un peu (voire beaucoup) de mal avec les accents. Il faut donc lui donner en argument une version non accentuée de chaque mot comportant des accents (c'est aussi valable pour les mots avec majuscules). Ces deux versions devront être séparées par un @. Oui, je sais : c'est tordu. Mais il faut faire avec, c'est le formalisme de LaTeX.

Code : TeX

```
J'aime le pâté\index{pate@pâté} et les patates\index{patate@Patate}.
```

Index

Tout s'arrange :)

Patate, 1
paté, 1

La figure suivante montre le résultat du code vu plus haut. L'ordre alphabétique est de nouveau respecté.

Si vous souhaitez créer des sous-entrées dans l'index, il vous suffit de séparer les entrées et sous-entrées (Notez que créer des sous-sous-entrées est possible.) par des « ! ». L'exemple est très parlant ici (figure suivante).

Code : TeX

```
J'aime le pâté\index{pate@pâté} et les patates\index{patate}, ainsi  
que les patates nouvelles\index{patate!patates nouvelles}.
```

Index

Sous-entrée dans un index

patate, 1
 patates nouvelles, 1
paté, 1

Pour finir, vous pouvez faire des références croisées au sein de vos index. « Références croisées », ce n'est pas vraiment très parlant comme terme, nous allons donc partir d'un exemple pour expliquer ce qu'elles sont. Faites le test, insérez `|see{ }`. Cette commande permet à l'index de lier deux mots entre eux. Ainsi, écrire `\index{poulpy|see{bestiole mythologique}}` donnera dans l'index « poulpy, voir bestiole mythologique ».

Bien entendu, ici le mot « see » est traduit par « voir » lors de la compilation, mais LaTeX peut très bien compiler dans d'autres langues.

Pratique

Votre mission, si vous l'acceptez, est de créer l'index ci-dessous (figure suivante) à partir de la phrase « Poulpy est chatoyante, merveilleuse et multicolore bien que son inestimable charisme ne cache qu'une fourberie sans nom, son être tout entier est abyssal et inestimable. »

Index

Poulpy, 1	
charisme, 1	Exercice
fourberie, 1	
chatoyante, 1	
être, 1	
abyssal, 1	
inestimable, 1	
merveilleuse, 1	
multicolore, 1	

La solution

Code : TeX

```
Poulpy\index{poulpy@Poulpy} est
chatoyante\index{poulpy@Poulpy!chatoyante},
merveilleuse\index{poulpy@Poulpy!merveilleuse} et
multicolore\index{poulpy@Poulpy!multicolore} bien que son
inestimable charisme\index{poulpy@Poulpy!charisme} ne cache qu'une
fourberie\index{poulpy@Poulpy!charisme!fourberie} sans nom, son
être\index{poulpy@Poulpy!etre@être} tout entier est
abyssal\index{poulpy@Poulpy!etre@être!abyssal} et
inestimable\index{poulpy@Poulpy!etre@être!inestimable}.
```

Si vous en êtes venu à bout sans problème, vous pouvez vous estimer à l'aise avec LaTeX.

En résumé

- Une table des matières s'insère très facilement avec la commande `\tableofcontents`. De même, il est possible d'insérer la table des figures (`\listoffigures`) et des tables (`\listoftables`).
- Il faut compiler deux fois pour que ces tables puissent être générées.
- On utilise la commande `\index` pour ajouter des éléments à l'index. Pour compiler celui-ci, il faut appeler le programme `makeindex`.
- LaTeX place mal les mots accentués dans son index, mais il est possible d'y remédier en indiquant le même mot sans accent à LaTeX.

La bibliographie

Cher lecteur, vous qui avez suivi avec attention jusque-là, je dois vous annoncer que vous arrivez au dernier chapitre de cours de ce livre.

Pour finir en beauté, nous allons apprendre l'une des notions qui donnent des boutons à bien des débutants (Les étudiants que j'ai interrogés m'ont pour la plupart avoué que ça leur faisait peur à tel point qu'ils n'avaient même pas essayé.) en LaTeX : la bibliographie. Cet élément, bien que créé pour être maniable, est souvent mal compris par les novices et passé sous silence.

Ce chapitre est consacré entièrement à l'une des méthodes permettant de créer des bibliographies. Elle sera développée et expliquée en long, en large et en travers.

Conséquence logique du problème : les débutants n'insèrent pas de bibliographie dans leurs publications et ne citent ni leurs sources ni leurs ressources documentaires. C'est une grave erreur...

Principe d'une bibliographie

Avant d'aborder la technique et LaTeX, j'aimerais parler de la définition du mot « bibliographie » et de son sens. Rien que sur la signification du mot, deux écoles se distinguent.

D'un côté, nous avons ceux qui estiment que la bibliographie comporte la liste de toutes les œuvres, qu'elles soient sous format papier ou numérique, qui ont servi de près ou de loin à concevoir une production. De l'autre, il y a ceux pour qui une bibliographie est un rassemblement d'œuvres répondant à un besoin particulier. Je vous cite ci-dessous un extrait de la définition du mot « bibliographie » sur Wikipédia, qui vous permettra de vous faire une idée des différents types de bibliographie existants.

Citation

Une bibliographie thématique regroupe des documents sur un même sujet ou un même thème. Inversement, une bibliographie systématique (parfois générale) regroupe des documents parus dans un même lieu et / ou à une même date.

Une bibliographie signalétique ne donne que les caractéristiques essentielles du document. Une bibliographie analytique en donne une description qui se veut neutre. Une bibliographie critique émet un jugement sur les documents qui y sont signalés.

Une bibliographie rétrospective décrit des documents publiés dans le passé.

Une bibliographie exhaustive décrit tous les livres publiés dans le cadre thématique, chronologique ou géographique retenu ; elle s'oppose à une bibliographie sélective. Une bibliographie sommaire propose une description plus réduite des ouvrages concernés.

La forme globale et la présentation d'une bibliographie sont **variables selon les besoins et les auteurs de contenu**. C'est ce que nous devons retenir ici.

À ce stade, normalement, les choses doivent vous sembler un peu tordues. Vous pensiez savoir ce qu'est une bibliographie... et au final vous vous rendez compte que les bibliographies peuvent prendre de multiples formes. Néanmoins, ces multiples formes ont un point commun : **la mise en forme d'une liste d'œuvres**.

Restons sur cette idée-là : « Une bibliographie est une liste d'un certain nombre d'œuvres choisies, répondant toutes à une consigne (regrouper les sources, des œuvres thématiques, etc.), mise en page selon les souhaits d'un (ou de plusieurs) auteur(s) ».

Tout de suite, nous y voyons déjà un peu plus clair.

La démarche de l'auteur est donc de choisir, dans la longue liste des œuvres qu'il connaît, celles qui seront publiées dans sa bibliographie. La mise en forme, LaTeX sait très bien la faire. La liste proprement dite est créée par un programme externe : BibTeX.



BibTeX fournit à l'auteur une armada d'outils destinés à concevoir facilement sa liste (que nous appellerons base de données) contenant toutes les informations sur les œuvres dont il dispose (auteur, date de parution, etc.).

L'auteur affectera une petite étiquette à chaque œuvre. Il lui suffira ensuite, dans son document LaTeX, de faire référence à certaines étiquettes pour intégrer aussitôt les œuvres concernées dans la bibliographie de sa publication.

Résumons tout ceci dans un schéma (figure suivante).

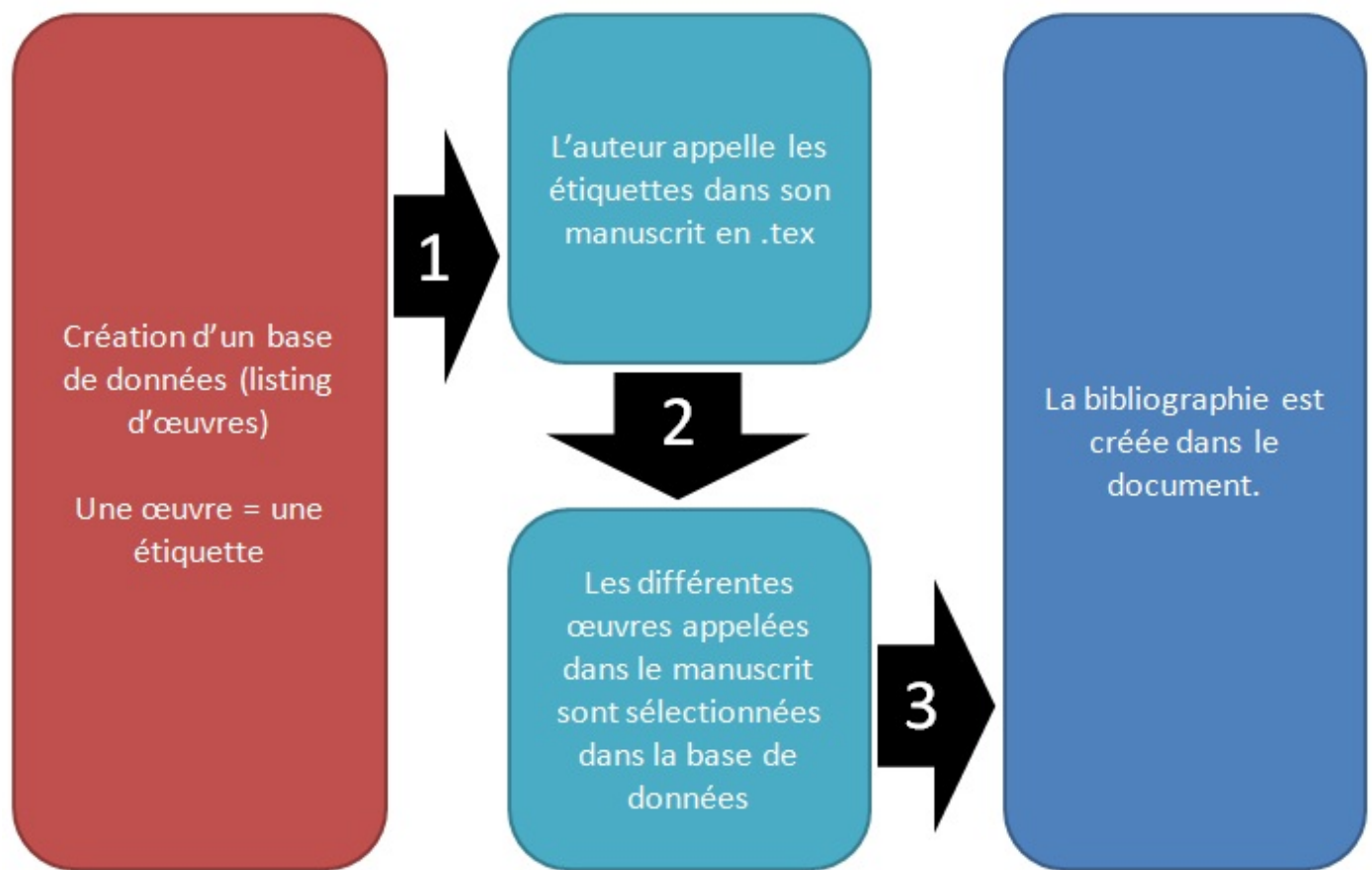


Schéma résumé

L'idée générale étant maintenant fixée, entrons dans le vif du sujet. Dans un premier temps, nous créerons la base de données, avant d'apprendre à faire des références à celle-ci dans nos documents `.tex` pour créer la bibliographie.

La base de données

Créer la base de données est la partie la plus facile. Elle est constituée de paragraphes contenant chacun les informations et l'étiquette des œuvres.



Le fichier que nous enregistrerons ici, qui contiendra la base de données, ne sera pas un `.tex` mais un `.bib`.

La structure des informations concernant une œuvre dans la base de données est toujours la même. Je vous la montre ci-dessous.

Code : TeX

```

@type_d'œuvre{référence_courte,
Titre_du_champ1={texte de votre choix},
Titre_du_champ2={texte de votre choix},
Titre_du_champ3={texte de votre choix},
Titre_du_champ4={texte de votre choix},
}
Entre deux œuvres, mon texte est considéré comme commentaire.
@type_d'œuvre{référence_courte,
Titre_du_champ1={texte de votre choix},
Titre_du_champ2={texte de votre choix},
Titre_du_champ3={texte de votre choix},
Titre_du_champ4={texte de votre choix},
}
  
```

Vous pouvez enchaîner autant d'œuvres que vous le souhaitez dans le fichier.

Il faut néanmoins respecter certaines règles pour que votre base de données fonctionne correctement :

- ne pas oublier les virgules ;
- ne pas choisir au hasard des types d'œuvres ou des titres de champs. Ils sont prédéfinis, vous ne pouvez pas en inventer.

Voilà ce qui rebute les débutants : le formalisme un peu triste et strict. En réalité, par rapport à la mise en forme d'un tableau, il est beaucoup plus simple d'écrire et de relire ce genre de code.

Revenons à notre code. Je vous ai dit que les types d'œuvres (Le type `@misc` est utilisé pour les sites web et ce qui ne trouve pas sa place dans les autres catégories.) et les titres des champs étaient prédéfinis. Il est temps pour vous de jeter un œil à leurs deux tableaux récapitulatifs.

Type d'oeuvre		
Type d'oeuvre	Commentaires	Champs
<code>@article</code>	Article	author journal title year month pages notes
<code>@book</code>	Livre	author title publisher year
<code>@manual</code>	Document technique	title author year organization
<code>@misc</code> (utilisé pour les sites web et ce qui ne trouve pas sa place ailleurs)	Divers	author title month year note
<code>@phdthesis</code>	Thèse de doctorat	title author school year
<code>@unpublished</code>	Manuscrit non publié	author title note year month

Il en existe bien sûr d'autres, mais ceux-ci sont les plus répandus.

Champs	Correspondance
Champs	
author	auteur
journal	Journal

month	mois de publication, au choix (jan, feb, mar, apr
note	notes complémentaires
organization	société ou organisation
pages	pages concernées
publisher	éditeur
year	année de publication

Nous pouvons, à l'aide des deux tableaux, créer notre première petite base de données bibliographique. Nous l'appellerons `biblio.bib`.

Code : TeX

Pour `author`, les noms et prénoms sont séparés par une virgule, et les différents auteurs par `AND`

```
@Article{ref,
author={Durand, Patrick AND Durand, René},
journal={Le beau journal},
title={Les tomates tueuses},
year={2007},
month={jan},
pages={24},
notes={sympathique}
}
```

```
@book
{ref2,
author={Dupont, Fernand},
title={Les choux farcis},
publisher={Un gros éditeur},
year={2004}
}
```

```
@manual
{ref3,
title={Réparer son vaisseau},
author={Dupont, Nestor},
year={2009},
organization={L'Alliance}
}
```

```
@misc
{ref4,
author={Les Zéros},
title={Le site du zéro},
month={jun},
year={2009},
note={www.siteduzero.com}
}
```

```
@phdthesis
{ref5,
title={La belle thèse},
author={Barbu, Monsieur},
school={Université},
year={2002}
}
```

```
@unpublished
{ref6,
author={Mauvais, Monsieur},
```

```
title={Mon roman inachevé},
note={il est chouette mon roman},
year={2000},
month={feb}
}
```

Mise en place de la bibliographie

La base de données bibliographique que vous venez d'apprendre à créer peut contenir autant d'éléments que vous le souhaitez. L'idée centrale est d'y regrouper les œuvres sur lesquelles vous vous basez et celles que vous citerez dans vos documents.

Ici nous avons créé un fichier `.bib`. Certains n'en utilisent qu'un seul dans lequel ils vont piocher (L'intérêt d'une base de données est justement de pouvoir piocher à l'intérieur sans être obligé d'utiliser l'intégralité des informations qu'elle stocke.), d'autres en créent un par publication, c'est selon les goûts de chacun. De mon côté, j'aime assez l'idée de disposer, en un seul endroit, de la liste des ressources utilisées dans tous mes écrits (En partant du principe que le geek a tendance à être un peu désordonné, disséminer des fichiers `.bib` partout est dangereux).

Concept

Apprenons maintenant à nous servir de cette base. Nous ferons des références à des éléments de la base de données via la commande `\cite{référence_courte}` que nous placerons où nous le voudrons dans le texte. Deux autres commandes viennent s'y ajouter : `\bibliographystyle{ }` et `\bibliography{nom du fichier.bib sans l'extension .bib}`. La première donne à LaTeX le style de mise en page souhaité pour la bibliographie, la seconde lui précise l'emplacement de la base de données.

Les éléments de la base de données auxquels les commandes `\cite{référence_courte}` font référence seront listés avec leurs informations respectives dans une page « Bibliographie » ou un encart « Références », suivant les classes de vos documents. Cette page se trouvera au même emplacement que celui où vous choisirez de faire apparaître les commandes `\bibliographystyle{ }` et `\bibliography{ }` (Leur fonctionnement est similaire à celui des tables.).

Exemples

Beaucoup d'informations vous ont été données. Il est temps de vous montrer quelques exemples, illustrant les différents styles de bibliographies possibles, avant de vous expliquer plus en détail comment se passe la compilation sous Mac OS, Linux ou Windows.

Code : TeX

```
\documentclass{article}

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

\begin{document}

J'insère le premier \cite{ref}, le second \cite{ref2}, le troisième
\cite{ref3}, le quatrième \cite{ref4}, le cinquième \cite{ref5} et
le sixième \cite{ref6}.

\bibliographystyle{ } % Le style est mis entre accolades.
\bibliography{bibli} % mon fichier de base de données s'appelle
bibli.bib

\end{document}
```

`\bibliographystyle{plain}` classe les entrées par ordre alphabétique et les numérote en conséquence (figure

suivante).

J'insère le premier [4], le second [2], le troisième [3], le quatrième [6], le cinquième [1] et le sixième [5].

Références

- [1] Monsieur Barbu. *La Belle thèse*. PhD thesis, Université, 2002.
- [2] Fernand Dupont. *Les choux farcis*. Un gros éditeur, 2004.
- [3] Nestor Dupont. *réparer son vaisseau*. l'Alliance, 2009.
- [4] Patrick Durand and René Durand. Les tomates tueuses. *Le beau journal*, page 24, jan 2007.
- [5] Monsieur Mauvais. Mon roman inachevé. il est chouette mon roman, feb 2000.
- [6] Les Zéros. Le site du zéro, jun 2009. www.siteduzero.com.

`\bibliographystyle{plain}`

`\bibliographystyle{abbrv}` classe les entrées par ordre alphabétique, les numérote en conséquence et abrège certains éléments de la bibliographie (figure suivante).

J'insère le premier [4], le second [2], le troisième [3], le quatrième [6], le cinquième [1] et le sixième [5].

Références

- [1] M. Barbu. *La Belle thèse*. PhD thesis, Université, 2002.
- [2] F. Dupont. *Les choux farcis*. Un gros éditeur, 2004.
- [3] N. Dupont. *réparer son vaisseau*. l'Alliance, 2009.
- [4] P. Durand and R. Durand. Les tomates tueuses. *Le beau journal*, page 24, jan 2007.
- [5] M. Mauvais. Mon roman inachevé. il est chouette mon roman, feb 2000.
- [6] L. Zéros. Le site du zéro, jun 2009. www.siteduzero.com.

`\bibliographystyle{abbrv}`

`\bibliographystyle{unsrt}` trie les entrées par ordre d'apparition dans le texte (figure suivante).

J'insère le premier [1], le second [2], le troisième [3], le quatrième [4], le cinquième [5] et le sixième [6].

Références

- [1] Patrick Durand and René Durand. Les tomates tueuses. *Le beau journal*, page 24, jan 2007.
- [2] Fernand Dupont. *Les choux farcis*. Un gros éditeur, 2004.
- [3] Nestor Dupont. *réparer son vaisseau*. l'Alliance, 2009.
- [4] Les Zéros. Le site du zéro, jun 2009. www.siteduzero.com.
- [5] Monsieur Barbu. *La Belle thèse*. PhD thesis, Université, 2002.
- [6] Monsieur Mauvais. Mon roman inachevé. il est chouette mon roman, feb 2000.

`\bibliographystyle{unsrt}`

`\bibliographystyle{alpha}` : le repère n'est plus un chiffre, mais les trois premières lettres du nom de l'auteur accolées aux deux derniers chiffres de l'année de parution (figure suivante).

J'insère le premier [DD07], le second [Dup04], le troisième [Dup09], le quatrième [Zér09], le cinquième [Bar02] et le sixième [Mau00].

Références

- [Bar02] Monsieur Barbu. *La Belle thèse*. PhD thesis, Université, 2002.
- [DD07] Patrick Durand and René Durand. Les tomates tueuses. *Le beau journal*, page 24, jan 2007.
- [Dup04] Fernand Dupont. *Les choux farcis*. Un gros éditeur, 2004.
- [Dup09] Nestor Dupont. *réparer son vaisseau*. l'Alliance, 2009.
- [Mau00] Monsieur Mauvais. Mon roman inachevé. il est chouette mon roman, feb 2000.
- [Zér09] Les Zéros. Le site du zéro, jun 2009. www.siteduzero.com.

`\bibliographystyle{alpha}`

Compilation

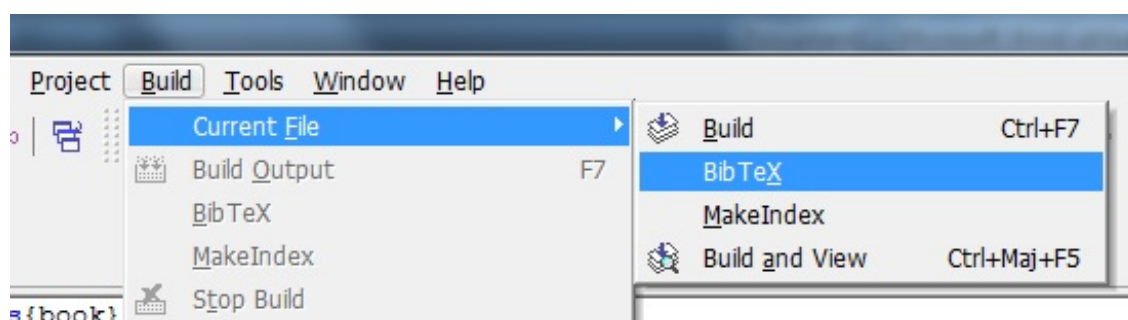
La compilation d'un document comportant une bibliographie se passe d'une manière légèrement inhabituelle. Détaillons les trois façons de venir à bout de celle-ci.

Gardons à l'esprit qu'il faudra d'abord compiler notre document LaTeX (afin d'obtenir la liste des références à rechercher dans la base de données) avant d'appeler BibTeX (pour piocher dans la base de données) et de compiler une nouvelle fois notre document LaTeX.

Dans la pratique, des cas exotiques peuvent survenir (Typiquement, des créations de sommaires, de notes de bas de page et autres petites choses sympathiques.) : j'ai pris l'habitude de compiler deux fois au lieu d'une les sources LaTeX.

Sous MiKTeX

Compilez deux fois votre document, comme vous le feriez d'habitude, avant de cliquer sur le bouton « BibTeX » contenu dans Build → Current File (figure suivante) et de compiler deux nouvelles fois.



Le bouton bibtex de

MiKTeX

Le processus est un peu fastidieux, mais une fois maîtrisé il ne pose pas de problème : il s'agit juste d'une nouvelle routine à intégrer.

Sous Linux

Pour vous, cela se passe dans la console avec les commandes suivantes :

Code : Console

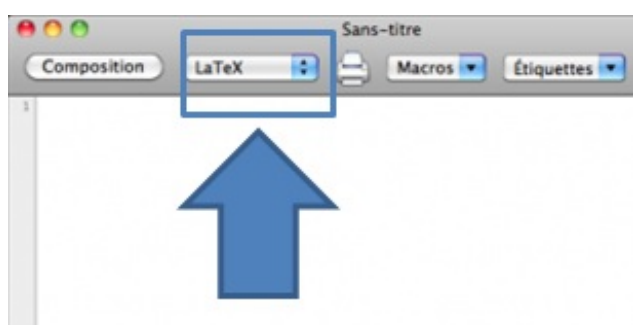
```
latex nom_de_fichier.tex
latex nom_de_fichier.tex

bibtex nom_de_fichier

latex nom_de_fichier.tex
latex nom_de_fichier.tex
```

TeXShop

Sous Mac OS, compilez deux fois comme à votre habitude dans un premier temps. Sélectionnez ensuite, dans le menu déroulant approprié (voir figure suivante), l'option « BibTeX » au lieu de « LaTeX » et compilez. Cela fait, sélectionnez à nouveau LaTeX dans votre menu déroulant, compilez deux nouvelles fois et tout ira bien.



Le bouton BibTeX de TeXShop

En résumé

- Créer une bibliographie nécessite une nouvelle composante de notre installation LaTeX : BibTeX.
- Une bibliographie se construit en deux temps : d'abord, il faut élaborer une base de données en créant un fichier `.bib`. Ensuite, on fait appel aux références dans le texte avec la commande `\cite{}`.
- La bibliographie est insérée avec la commande `\bibliography{}`.
- Il existe quatre façons différentes de présenter une bibliographie. La commande `\bibliographystyle{}` permet de choisir entre ces différentes options de mise en forme.
- Une bibliographie **n'est pas un élément facultatif** dans un rapport ou une thèse, votre crédibilité peut en dépendre. Ne la négligez pas !

TP 2 – Rédaction d'un article présentant le théorème de Pythagore

Vous voici arrivé au deuxième et dernier TP de cet ouvrage. Il a été pensé pour qu'une majeure partie des notions abordées précédemment soit revue et que vous puissiez vous rendre compte de vos lacunes éventuelles. Nous utiliserons ici un article de Wikipédia traitant du théorème de Pythagore et deux tables issues du même site.

Le livre final issu de ce TP fait 23 pages (dont 9 blanches) et ne comporte pas de piège vicieux. J'ai préféré vous faire utiliser les commandes usuelles, à vous de voir si vous voulez ajouter des fioritures ou faire des essais (Je vous conseille d'être curieux et de tenter des choses.).

Les consignes

J'ai choisi pour cet exercice de ne plus vous tenir la main, il vous faudra réussir à suivre les consignes données, tenter de recréer une structure et la remplir avec le contenu indiqué, ceci sans oublier de respecter les standards et le formalisme de LaTeX.

Terrifiant, n'est-ce pas ?

Ne vous inquiétez pas, tout va bien se passer.

La structure

Nous allons nous préoccuper en premier lieu de la structure du **livre** que nous rédigeons. Il comportera :

- une page de garde dont le titre sera « Le théorème de Pythagore » et comportant votre nom en **petites capitales** et la **date du jour** ;
- un **sommaire** ;
- un **chapitre d'introduction** que l'on appellera « Introduction » ;
- une **première partie** nommée « Théorème de Pythagore » contenant deux **chapitres**, l'un nommé « Énoncé du théorème » et l'autre « Réciproque » ;
- le **chapitre** « Énoncé du théorème » comportera deux **sections** appelées respectivement « Théorie » et « Exemple » ;
- une **seconde partie**, du nom d'« Annexes et tables » contiendra : un **chapitre** « Table d'addition », un **chapitre** « Table de multiplication », une **table des figures**, une **liste des tableaux** ainsi qu'une **bibliographie**.

Vous devrez choisir où placer les différentes commandes de structure relatives aux livres selon vos souhaits. Essayez de rester le plus pertinent possible dans vos choix.

L'article « Théorème de Pythagore » sur Wikipédia (duquel est issu le texte que nous utiliserons) est disponible [ici](#).

Les contenus

Chapitre d'introduction

Le chapitre d'introduction comporte un paragraphe et une image (dont la légende sera « Une version géométrique du théorème »).

Le texte qu'il doit contenir est le suivant et l'image (figure suivante) est placée sur la page de l'article.

Citation

Le théorème de Pythagore est un théorème de géométrie euclidienne qui énonce que dans un triangle rectangle (qui possède un angle droit) le carré de la longueur de l'hypoténuse (côté opposé à l'angle droit) est égal à la somme des carrés des longueurs des deux autres côtés. Ce théorème est nommé d'après Pythagore de Samos, mathématicien, philosophe et astronome de la Grèce antique.

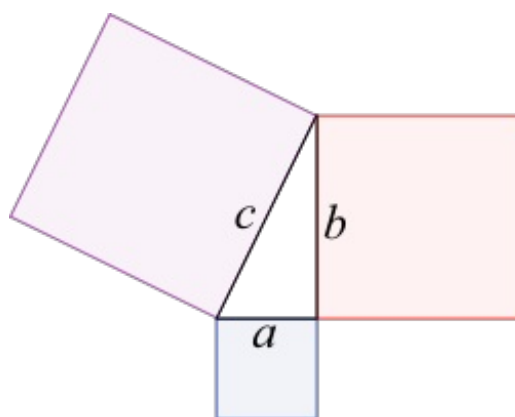


Image de l'introduction

Chapitre « Énoncé du théorème »

Nous utiliserons ici pour la première fois dans notre document une référence à la bibliographie, ainsi que la mise en page liée au théorème (figure suivante). Les mathématiques sont omniprésentes et un flottant fait son apparition (figure suivante). Bien que déclaré après les paragraphes de texte, ce flottant est inséré dans la section « Théorie ».

Le texte provient de l'article « Théorème de Pythagore » de Wikipédia et la mise en page doit respecter les captures ci-dessous (eh non, je ne vous donne pas l'image, vous allez devoir la prendre sur la page du wiki et la convertir vous-même en PNG).

Chapitre 1

Énoncé du théorème

1.1 Théorie

La forme la plus connue du théorème de Pythagore §§ est la suivante :

Théorème (de Pythagore). Dans un triangle rectangle, le carré de la longueur de l'hypoténuse (côté opposé à l'angle droit) est égal à la somme des carrés des longueurs des côtés de l'angle droit.

Note : Le terme « longueur », généralement oublié, est très important. En effet, la longueur est un nombre réel sur lequel l'opération d'élevation au carré est parfaitement définie ; l'hypoténuse est un segment de droite, objet géométrique pour lequel l'élevation au carré n'a pas de sens.

Cependant, il est parfois retenu afin de ne pas compliquer l'apprentissage du théorème (La notion de longueur étant sous-entendue).

Dans un triangle ABC rectangle en C, AB étant l'hypoténuse, où $AB = c$, $AC = b$ et $BC = a$ (cf. figure ci-dessous), on aura donc : $BC^2 + AC^2 = AB^2$ ou encore $a^2 + b^2 = c^2$.

Le théorème de Pythagore permet ainsi de calculer la longueur d'un des côtés d'un triangle rectangle si on connaît les deux autres.

1.2 Exemple

Avec les notations ci-dessus, soit le triangle rectangle de côtés $a = 3$ et $b = 4$; alors la longueur du troisième côté, c , est donnée par : $a^2 + b^2 = 3^2 + 4^2 = 25 = c^2$. Les longueurs étant des réels positifs, on obtient $c = 5$. Un triplet de nombres entiers tel que $(3, 4, 5)$, représentant la longueur des côtés d'un triangle rectangle s'appelle un triplet pythagoricien.

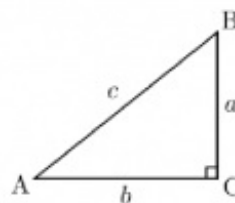


FIGURE 1.1 – Triangle rectangle

Les textes

Le flottant

Chapitre « Réciproque »

Si vous avez réussi l'épreuve du chapitre « Énoncé du théorème », celui-ci ne vous posera aucun problème (figure suivante).

Chapitre 2

Réciproque

La réciproque du théorème de Pythagore est également vraie :

Réciproque (Théorème de Pythagore). Si dans un triangle, la somme des carrés de deux côtés est égale au carré du plus grand côté, alors ce triangle est rectangle.

Le théorème de Pythagore est donc une propriété caractéristique des triangles rectangles. Formulé autrement, si dans un triangle ABC on a $BC^2 + AC^2 = AB^2$, alors ce triangle est rectangle en C .

Chapitre « Réciproque »

5

Tables d'addition et de multiplication

Ces deux annexes (figures suivante et suivante) nécessitent la création de tableaux, l'utilisation d'une couleur en niveaux de gris et des références bibliographiques. Les choses se corsent un peu...

Annexe A

Table d'addition

Table issue de Wikipédia [1].

Additionné à	1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10	11
2	3	4	5	6	7	8	9	10	11	12
3	4	5	6	7	8	9	10	11	12	13
4	5	6	7	8	9	10	11	12	13	14
5	6	7	8	9	10	11	12	13	14	15
6	7	8	9	10	11	12	13	14	15	16
7	8	9	10	11	12	13	14	15	16	17
8	9	10	11	12	13	14	15	16	17	18
9	10	11	12	13	14	15	16	17	18	19
10	11	12	13	14	15	16	17	18	19	20

TABLE A.1 – Table d'addition

9

Annexe B

Table de multiplication

Table issue de Wikipédia [2].

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

TABLE B.1 – Table de multiplication

11

La bibliographie

Une bibliographie sera utile dans ce document, je vous montre son allure sur la figure suivante. À vous de créer la base de données correspondante (Elle ne comporte aucun piège.).

Bibliographie

- [1] Wikipédia. Table d'addition, jul 2010. fr.wikipedia.org.
- [2] Wikipédia. Table de multiplication, jul 2010. fr.wikipedia.org.
- [3] Wikipédia. Théorème de pythagore, jul 2010. fr.wikipedia.org.

La bibliographie

Au boulot (normalement, vous ne devriez pas avoir besoin de plus d'une bonne heure) !

Petite aide

Vous avez eu toutes les informations nécessaires à la création du document, néanmoins, je me doute qu'il n'est pas forcément évident de visualiser le travail dans sa globalité. Si (et seulement si) vous avez essayé de faire l'exercice, mais que vous vous sentez trop perdu, vous pouvez consulter le document au format PDF [ici](#).

La correction

Compiler souvent, reprendre les explications des chapitres précédents et les exemples, telles étaient les clés du succès. Je ne vous donnerai pas ici de technique détaillée pour parvenir au document final (C'est exactement le même type de processus que pour le premier TP).

Les concepts avancés de LaTeX s'entremêlent tous lors de la composition d'un document, de sorte que vous pouvez indifféremment constituer votre bibliographie pendant ou après l'écriture de votre contenu. L'important, c'est d'être organisé et de ne rien oublier. Lors de la création du TP, j'ai utilisé la méthode suivante pour réaliser le document :

- création de la structure d'un livre (avec `\appendix`, `\begin{document}`, etc.) ;
- compilation et correction des erreurs ;
- page de garde et listing des `\usepackage{}` ;
- compilation et correction des erreurs ;
- création des éléments de structure interne (parties, chapitres, sections) ;
- compilation et correction des erreurs ;
- création du sommaire ;
- double compilation et correction des erreurs ;
- remplissage des différents chapitres ;
- compilation et correction des (nombreuses) erreurs ;
- création des flottants ;
- compilation et correction des erreurs ;
- création des tables de flottants ;
- compilation ;
- création de la bibliographie ;
- cycle habituel : deux compilations, appel de BibTeX puis deux nouvelles compilations.

Le trait est ici à peine grossi, et la méthode est ponctuée par de nombreuses étapes de correction. Cette méthode a l'avantage de pouvoir être transposée à la quasi-totalité des ébauches de publications que vous rédigerez (Évidemment, des étapes peuvent venir se greffer ou ne pas s'avérer nécessaires.). Suivant votre efficacité, vous pourrez sauter l'une ou l'autre des compilations.

Pour ma part, j'aime assez savoir d'où viennent mes erreurs et compiler après chaque étape me permet de ne pas avoir besoin de fouiller très longtemps, même avec des documents complexes.

Les deux seules difficultés de cet exercice sont la présence d'une bibliographie à remplir avec des pages web (donc, des entrées de type @misc) et les tableaux un peu complexes (qui ont déjà été vus dans le chapitre sur les tableaux). Je vous livre ci-dessous les deux codes source, celui de bibliographie.bib, ainsi que celui de tp2.tex. Prenez le temps de les lire si vous avez bloqué dessus et réessayez : vous y arriverez.

Code : TeX

```
\documentclass{book}

\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[français]{babel}
\usepackage{graphicx}
\usepackage{amsthm}
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{mathrsfs}
\usepackage{color}
\usepackage{colortbl}

\definecolor{grisclair}{gray}{0.8}

\newtheorem*{theo}{Théorème}
\newtheorem*{reci}{Réciproque}

\title{Le théorème de Pythagore}
\author{\textsc{Laleloulilo}}
\date{\today}

\begin{document}

\renewcommand{\contentsname}{Sommaire}

\maketitle % Page de garde.

\frontmatter
\tableofcontents

\chapter{Introduction}

Le théorème de Pythagore est un théorème de géométrie euclidienne
qui énonce que dans un triangle rectangle (qui possède un angle
droit) le carré de la longueur de l'hypoténuse (côté opposé à l'angle
droit) est égal à la somme des carrés des longueurs des deux autres
côtés. Ce théorème est nommé d'après Pythagore de Samos,
mathématicien, philosophe et astronome de la Grèce antique.

\begin{figure}
\begin{center}
\includegraphics{intro.png}
\end{center}
\caption{Une version géométrique du théorème}
\label{Une version géométrique du théorème}

\end{figure}

\mainmatter
\part{Théorème de Pythagore}
\chapter{Énoncé du théorème}

\section{Théorie}

La forme la plus connue du théorème de Pythagore \cite{theo} est la
suivante :

\begin{theo}[de Pythagore]
Dans un triangle rectangle, le carré de la longueur de l'hypoténuse
```

(côté opposé à l'angle droit) est égal à la somme des carrés des longueurs des côtés de l'angle droit.

`\end{theo}`

Note : Le terme « longueur », généralement oublié, est très important. En effet, la longueur est un nombre réel sur lequel l'opération d'élevation au carré est parfaitement définie ; l'hypoténuse est un segment de droite, objet géométrique pour lequel l'élevation au carré n'a pas de sens.

Cependant, il est parfois retiré afin de ne pas compliquer l'apprentissage du théorème (la notion de longueur étant sous-entendue).

```
\begin{figure}
\begin{center}
\includegraphics{triangle.png}
\end{center}
```

```
\caption{Triangle rectangle}
\label{Triangle rectangle}
\end{figure}
```

Dans un triangle ABC rectangle en C, AB étant l'hypoténuse, où $AB = c$, $AC = b$ et $BC = a$ (cf. figure ci-dessus), on aura donc : $BC^2 + AC^2 = AB^2$ ou encore $a^2 + b^2 = c^2$.

Le théorème de Pythagore permet ainsi de calculer la longueur d'un des côtés d'un triangle rectangle si on connaît les deux autres.

`\section{Exemple}`

Avec les notations ci-dessus, soit le triangle rectangle de côtés $a = 3$ et $b = 4$; alors la longueur du troisième côté, c , est donnée par : $a^2 + b^2 = 3^2 + 4^2 = 25 = c^2$. Les longueurs étant des réels positifs, on obtient $c = 5$. Un triplet de nombres entiers tel que $(3, 4, 5)$, représentant la longueur des côtés d'un triangle rectangle s'appelle un triplet pythagoricien.

`\chapter{Réciproque}`

La réciproque du théorème de Pythagore est également vraie :

```
\begin{reci}[Théorème de Pythagore]
Si dans un triangle, la somme des carrés de deux côtés est égale au
carré du plus grand côté, alors ce triangle est rectangle.
\end{reci}
```

Le théorème de Pythagore est donc une propriété caractéristique des triangles rectangles. Formulé autrement, si dans un triangle ABC on a $BC^2 + AC^2 = AB^2$, alors ce triangle est rectangle en C .

```
\appendix
\part{Annexes et Tables}
```

`\chapter{Table d'addition}`

Table issue de Wikipédia `\cite{addi}`.

```
\begin{table}
\begin{center}
```

```
\begin{tabular}{>{\begin{bf} \columncolor{grisclair}} c
<{\end{bf}}cccccccc}
```

```
\rowcolor{grisclair}Additionné à & \begin{bf}1\end{bf} &
\begin{bf}2\end{bf} & \begin{bf}3\end{bf} & \begin{bf}4\end{bf} &
\begin{bf}5\end{bf} & \begin{bf}6\end{bf} & \begin{bf}7\end{bf} &
\begin{bf}8\end{bf} & \begin{bf}9\end{bf} & \begin{bf}10\end{bf} \\
```

```

1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & \\
2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & \\
3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & \\
4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & \\
5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & \\
6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & \\
7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & \\
8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & \\
9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & \\
10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & \\

\end{tabular}
\end{center}
\caption{Table d'addition}
\label{Table d'addition}
\end{table}

\chapter{Table de multiplication}

Table issue de Wikipédia \cite{mult}.

\begin{table}
\begin{center}

\begin{tabular}{>{\begin{bf}} \columncolor{grisclair}} c
<{\end{bf}} ccccccccc}

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

\end{tabular}
\end{center}
\caption{Table de multiplication}
\label{Table de multiplication}
\end{table}
\backmatter

\listoffigures
\listoftables

```

```
\bibliographystyle{plain}
\bibliography{bibliographie}

\end{document}
```

Code : TeX

```
@misc
{theo,
author={Wikipédia},
title={Théorème de Pythagore},
month={jul},
year={2010},
note={fr.wikipedia.org}
}

@misc
{mult,
author={Wikipédia},
title={Table de multiplication},
month={jul},
year={2010},
note={fr.wikipedia.org}
}

@misc
{addi,
author={Wikipédia},
title={Table d'addition},
month={jul},
year={2010},
note={fr.wikipedia.org}
}
```

Être plus à l'aise avec LaTeX

Ce second TP est un résumé des notions développées dans cet ouvrage. Mine de rien, vous savez déjà beaucoup de choses sur LaTeX si vous en êtes à cette page, et ce n'est pas fini ! J'ai rassemblé ici quelques nouvelles commandes, des notions et des outils qui ne vont pas révolutionner votre utilisation de LaTeX, mais la rendre plus facile et plus intuitive.

Écrire un document dans plusieurs fichiers .tex

Le concept

Le livre que vous tenez entre vos mains est le fruit d'une compilation d'un fichier nommé `latex.tex`, je vous laisse imaginer la longueur du code nécessaire à la création d'un petit pavé de ce type... Ce sera long, très long même. Fort heureusement, LaTeX sait gérer les longs documents et aide les auteurs à scinder une même publication en plusieurs parties.

L'utilité de la chose ? Permettre à chacun de travailler sur une partie sans toucher au reste. Ainsi, au moment où j'écris ces lignes, des correcteurs sont en train de retoucher un autre chapitre et nous ne nous dérangeons pas mutuellement. Nous utilisons en plus de cela un logiciel qui nous permet de vérifier que nous n'écrasons pas le travail des autres, au cas où deux personnes travailleraient sur un même fichier.

La commande sur laquelle repose ce principe est toute simple, c'est la suivante :

Code : TeX

```
\input{chemin_de_mon_fichier}
```


Lorsque vous utilisez `\input`, LaTeX remplace simplement la commande par le contenu du fichier appelé. Un exemple ne ferait pas de mal, n'est-ce pas ?

Exemple

Nous allons créer quatre fichiers, le premier s'appelle `livre.tex` et contiendra le code ci-dessous :

Code : TeX

```
\documentclass{book}

\begin{document}

\input{chapitre1.tex}

\input{chapitre2.tex}

\input{partie2/chapitre3.tex}

\end{document}
```

Les trois autres fichiers (`chapitre1.tex`, `chapitre2.tex` et `chapitre3.tex` — ce dernier étant enregistré dans un dossier nommé `partie 2`) contiennent les codes ci-dessous.

Code : TeX

```
\chapter{Le premier}
```

Code : TeX

```
\chapter{Le second}
Poulpy est une déesse.
```

Code : TeX

```
\chapter{Le dernier}
```

Normalement, l'arborescence de vos fichiers doit être la suivante :

- dans un dossier quelconque, votre fichier `livre.tex`, ainsi que `chapitre1.tex`, `chapitre2.tex` et un dossier nommé `partie2` ;
- dans le dossier `partie2`, le fichier `chapitre3.tex`.

Le résultat

La commande `\input` va faire son travail et lors de la compilation, elle va être remplacée par le contenu du fichier vers lequel elle pointe (Tout ceci n'est pas visible, c'est la petite cuisine interne du compilateur).

Au cours de la compilation, votre fichier va ressembler à ça :

Code : TeX

```

\documentclass{book}

\begin{document}

% Début du contenu de chapitrel.tex
\chapter{Le premier}
% Fin du contenu de chapitrel.tex
% Début du contenu de chapitre2.tex
\chapter{Le second}
Poulpy est une déesse.
% Fin du contenu de chapitre2.tex
% Début du contenu de chapitre3.tex
\chapter{Le dernier}
% Fin du contenu de chapitre3.tex

\end{document}

```

Ce n'est pas plus compliqué que ça. Vous savez maintenant tout ce qu'il faut savoir sur la commande `\input`, vous pouvez donc l'utiliser pour dissocier les différents morceaux de vos publications.

L'option draft

Il fut un temps où vous étiez jeune et innocent, et où vous ne connaissiez pas les classes de document. Vous avez utilisé à travers les divers exemples les options liées aux tailles de polices par défaut et d'autres petites choses. Il est temps de vous apprendre un nouvel argument pour `\documentclass` : l'argument `draft`.

Le mot *draft* est un terme anglais qui signifie « brouillon » ou « ébauche ». Pour nous, cela veut dire deux choses :

- les images sont remplacées par des cadres blancs contenant leur chemin (absolu ou relatif) ;
- les problèmes de dépassement dans les marges peuvent être facilement repérés en vue de les corriger (Le draft insère de gros carrés noirs là où il y a dépassement, ce qui permet de les repérer extrêmement facilement et de modifier les phrases problématiques.).

La conséquence logique, c'est que les images ne sont pas vraiment chargées et donc, que la compilation est plus rapide. À titre d'exemple, ce livre a compilé en 65 secondes en mode normal et en 50 secondes en `draft`. Je recommande grandement son utilisation en dernière relecture pour vérifier que tous les mots rentrent dans les marges, et son gain en temps de compilation le rend intéressant quand un utilisateur ne travaille pas sur les images, mais qu'il compile souvent.

Faites le test sur ce TP, remplacez la première ligne de code par la suivante et constatez le changement.

Code : TeX

```

\documentclass[draft]{book}

```

Ici aussi, c'est une option toute simple, mais qui peut changer pas mal de choses et vous apporter énormément de confort.

Les unités de longueur

À de nombreuses reprises, des notions de longueurs ont été abordées, mais sans vraiment les définir. Quelle longueur fait un point par rapport à un centimètre ? Comment faire les conversions ? Pourquoi Poulpy est-elle tellement magnifique ?

Toutes les réponses sont ici !

Trève de plaisanteries, je vous ai concocté un tableau tout simple comportant le nom des unités de longueur, leur abréviation dans LaTeX (par exemple pour 1 centimètre entre crochets, il faut écrire `{1cm}`) ainsi que leur conversion en millimètres.



Gardez à l'esprit qu'une feuille A4 mesure 210 millimètres de large et 297 de haut. Cela permet souvent de choisir les bonnes longueurs à appliquer aux commandes.

Les unités de longueur		
Nom	Abréviation	Valeur en millimètres
Point	pt	0,35 mm
Millimètre	mm	1,00 mm
Pica	pc	4,21 mm
Cicéro	cc	4,53 mm
Centimètre	cm	10,00 mm
Inch	in	25,4 mm

Les espaces

Le dernier point que nous allons aborder est celui concernant l'espacement hors du mode scientifique. Il se peut que vous ayez besoin à certains moments d'insérer des espaces horizontaux ou verticaux de tailles précises entre deux mots ou deux morceaux d'un document.

Plutôt que de vous donner toute une gamme de commandes aussi indigeste que difficile à mémoriser, je vais seulement vous en donner quatre. Elles prennent toutes en argument une longueur dans l'unité de votre choix, la seule chose qui change étant leur effet. La combinaison de ces quatre commandes vous permet de faire à peu près n'importe quoi. Comprenez bien qu'il est parfois préférable de modérer l'usage de ces commandes, aussi puissantes que visuellement dangereuses. 😊

- `\hspace{longueur}` insère un espace horizontal de la longueur choisie.
- `\hspace*{longueur}` : idem, sauf que l'espace n'est pas inséré s'il y a un retour à la ligne entre les deux mots concernés par l'espacement.
- `\vspace{hauteur}` insère un espace vertical de la longueur choisie.
- `\vspace*{hauteur}` : ici, l'espace n'est pas inséré s'il y a un saut de page.

En résumé

- Une approche méthodique aide à venir à bout de gros projets LaTeX.
- Il est possible, et même recommandé, de séparer un gros document LaTeX en plusieurs fichiers. Cela facilite un travail organisé et collaboratif. Pour inclure un fichier `.tex` dans un autre, on utilise la commande `\input`.
- Le mode `draft` aide à compiler plus rapidement un document.
- LaTeX fournit des commandes aidant à l'insertion d'espacements de tailles et d'orientations variables : `\hspace` (espace horizontal) et `\vspace` (espace vertical).
- Citer vos sources dans une bibliographie est un gage de sérieux.

Partie 4 : Annexes

Les quelques annexes ci-dessous vous donnent les armes pour mieux comprendre le cours.

Les caractères spéciaux

Les caractères spéciaux sont très nombreux et très utiles lors de la création de documents. J'ai choisi de vous les lister dans cette annexe afin que vous puissiez les avoir sous la main à tout moment.

Les packages utilisés ici ont été les suivants (rien ne change avec l'UTF-8) :

Code : TeX

```
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[français]{babel}
\usepackage{lmodern}
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{mathrsfs}
```

Environnement mathématique

Lettres grecques

<i>ligne1</i>	α	β	γ	δ	ϵ
<i>ligne2</i>	ζ	η	θ	ι	κ
<i>ligne3</i>	λ	μ	ν	ξ	π
<i>ligne4</i>	ρ	σ	τ	υ	ϕ
<i>ligne5</i>	χ	ψ	ω		
<i>ligne6</i>	Λ	Θ	Γ	Δ	Ξ
<i>ligne7</i>	Π	Σ	Υ	Φ	Ψ
<i>ligne8</i>	Ω				

Lettres grecques

Code : TeX

```
$$
\begin{array}{c c c c c c}
ligne 1 & \alpha & \beta & \gamma & \delta & \epsilon \\
ligne 2 & \zeta & \eta & \theta & \iota & \kappa \\
ligne 3 & \lambda & \mu & \nu & \xi & \pi \\
ligne 4 & \rho & \sigma & \tau & \upsilon & \phi \\
ligne 5 & \chi & \psi & \omega & & \\
ligne 6 & \Lambda & \Theta & \Gamma & \Delta & \Xi \\
ligne 7 & \Pi & \Sigma & \Upsilon & \Phi & \Psi \\
ligne 8 & \Omega & & & & 
\end{array}
```

```
\end{array}
$$
```

Opérateurs extensibles

<i>ligne1</i>	Σ	\int	\biguplus	\bigoplus	\bigvee	Opérateurs extensibles
<i>ligne2</i>	\prod	\oint	\bigcap	\bigotimes	\bigwedge	
<i>ligne3</i>	\coprod	\iint	\bigcup	\bigodot	\bigsqcup	

Code : TeX

```
$$
\begin{array}{c c c c c c}
\text{ligne 1} & \& \backslash\text{sum} & \& \backslash\text{int} & \& \backslash\text{biguplus} & \& \backslash\text{bigoplus} & \& \backslash\text{bigvee} \\
\text{ligne 2} & \& \backslash\text{prod} & \& \backslash\text{ooint} & \& \backslash\text{bigcap} & \& \backslash\text{bigotimes} & \& \backslash\text{bigwedge} \\
\text{ligne 3} & \& \backslash\text{coprod} & \& \backslash\text{iint} & \& \backslash\text{bigcup} & \& \backslash\text{bigodot} & \& \backslash\text{bigsqcup} \\
\end{array}
$$
```

Fonctions

<i>ligne1</i>	arccos	cos	csc	exp	ker
<i>ligne2</i>	lim sup	min	sinh	arcsin	cosh
<i>ligne3</i>	deg	gcd	lg	ln	Pr
<i>ligne4</i>	sup	arctan	cot	det	hom
<i>ligne5</i>	lim	log	sec	tan	arg
<i>ligne6</i>	coth	dim	inf	lim inf	max
<i>ligne7</i>	sin	tanh			

Fonctions

Code : TeX

```


$$\begin{array}{c c c c c c}
\text{ligne 1} & \backslash \arccos & \backslash \cos & \backslash \csc & \backslash \exp & \backslash \ker \\
\text{ligne 2} & \backslash \limsup & \backslash \min & \backslash \sinh & \backslash \arcsin & \backslash \cosh \\
\text{ligne 3} & \backslash \deg & \backslash \gcd & \backslash \lg & \backslash \ln & \backslash \Pr \\
\text{ligne 4} & \backslash \sup & \backslash \arctan & \backslash \cot & \backslash \det & \backslash \hom \\
\text{ligne 5} & \backslash \lim & \backslash \log & \backslash \sec & \backslash \tan & \backslash \arg \\
\text{ligne 6} & \backslash \coth & \backslash \dim & \backslash \inf & \backslash \liminf & \backslash \max \\
\text{ligne 7} & \backslash \sin & \backslash \tanh & & & \\
\end{array}$$


```

Flèches

ligne1	\leftarrow	\longleftarrow	\uparrow	\rightrightarrows	\Leftrightarrow
ligne2	\Uparrow	\rightarrow	\longrightarrow	\downarrow	\Rightarrow
ligne3	\Rightarrow	\Downarrow	\Leftrightarrow	\longleftrightarrow	\Updownarrow
ligne4	\Leftrightarrow	\longleftrightarrow	\Updownarrow	\mapsto	$\overrightarrow{\hspace{1cm}}$
ligne5	\nearrow	\curvearrowleft	\curvearrowright	\searrow	\lrcorner
ligne6	\rightarrowtail	\swarrowtail	$\negthickspace\rightarrowtail$	\rightarrowtail	\nwarrowtail
ligne7	\rightleftharpoons	\rightsquigarrow	\dashrightarrow	\dashleftarrow	\rightleftarrows
ligne8	\rightleftarrows	\leftrightsquigarrow	\leftrightharpoons	\leftrightsquigarrow	\leftrightsquigarrow
ligne9	\rightleftharpoons	\curvearrowleft	\circlearrowleft	\curvearrowright	\Uparrow
ligne10	\updownarrow	\downarrow	\rightarrowtail	\rightsquigarrow	\Rightarrow
ligne11	\Leftrightarrow	\Rightarrow	\Leftrightarrow	\Rightarrow	\rightarrowtail
ligne12	\rightarrowtail	\Rightarrow	\curvearrowright	\circlearrowright	\rightarrowtail
ligne13	\Downarrow	\uparrow	\downarrow	\rightsquigarrow	\leftrightsquigarrow
ligne14	\rightarrowtail	\leftrightsquigarrow	\Rightarrow	\Leftrightarrow	\leftrightsquigarrow

Flèches

Code : TeX

```

$$
\begin{array}{c c c c c c}
\text{ligne 1} & \& \leftarrow & \longrightarrow & \uparrow & \Leftarrow & \\
& \& \Longleftarrow & \& \\
\text{ligne 2} & \& \Uparrow & \rightarrow & \longrightarrow & \downarrow & \\
& \& \Rightarrow & \& \\
\text{ligne 3} & \& \longrightarrow & \Downarrow & \leftrightsquigarrow & \\
& \& \longleftarrow & \updownarrow & \& \\
\text{ligne 4} & \& \leftrightsquigarrow & \Longleftarrow & \Updownarrow & \\
& \& \mapsto & \longmapsto & \& \\
\text{ligne 5} & \& \nearrow & \hookrightarrow & \hookrightarrow & \searrow & \\
& \& \leftharpoonup & \& \\
\text{ligne 6} & \& \rightharpoonup & \swarrow & \leftharpoondown & \\
& \& \rightharpoondown & \nwarrow & \& \\
\text{ligne 7} & \& \rightleftharpoons & \leadsto & \dashrightarrow & \\
& \& \dashleftarrow & \leftleftarrows & \& \\
\text{ligne 8} & \& \leftrightarrows & \Lleftarrow & \twoheadleftarrow & \\
& \& \leftarrowtail & \looparrowleft & \& \\
\text{ligne 9} & \& \leftrightharpoons & \curvearrowleft & \circlearrowleft & \\
& \& \Lsh & \upuparrows & \& \\
\text{ligne 10} & \& \upharpoonleft & \downharpoonleft & \multimap & \\
& \& \leftrightsquigarrow & \rightrightarrows & \& \\
\text{ligne 11} & \& \rightleftarrows & \rightrightarrows & \rightleftarrows & \\
& \& \twoheadrightarrow & \rightarrowtail & \& \\
\text{ligne 12} & \& \looparrowright & \rightleftharpoons & \curvearrowright & \\
& \& \circlearrowright & \Rsh & \& \\
\text{ligne 13} & \& \downdownarrows & \upharpoonright & \downharpoonright & \\
& \& \rightsquigarrow & \nleftarrow & \& \\
\text{ligne 14} & \& \nrightarrow & \nLeftarrow & \nrightarrow & \\
& \& \nleftarrow & \nLeftarrow & \& \\
& \& \end{array}
$$

```

Opérateurs

<i>ligne1</i>	$*$	\pm	\cap	\triangleleft	\star
<i>ligne2</i>	\mp	\cup	\triangleright	\cdot	\amalg
<i>ligne3</i>	\uplus	\triangleleft	\circ	\odot	\sqcap
<i>ligne4</i>	\triangleright	\bullet	\ominus	\sqcup	\unlhd
<i>ligne5</i>	\bigcirc	\oplus	\wedge	\unrhd	\diamond
<i>ligne6</i>	\oslash	\vee	\bigtriangledown	\times	\otimes
<i>ligne7</i>	\dagger	\triangle	\div	\wr	\ddagger
<i>ligne8</i>	\setminus	\centerdot	\Box	$\bar{\wedge}$	\veebar
<i>ligne9</i>	\circledast	\boxplus	\curlywedge	\curlyvee	\circledcirc
<i>ligne10</i>	\boxminus	\Cap	\Cup	\circleddash	\boxtimes

Opérateurs (partie 1)

Code : TeX

```


$$\begin{array}{cccccc}
\text{ligne 1} & \& \backslash\text{ast} & \& \backslash\text{pm} & \& \backslash\text{cap} & \& \backslash\text{lhs} & \& \backslash\text{star} \\
\text{ligne 2} & \& \backslash\text{mp} & \& \backslash\text{cup} & \& \backslash\text{rhd} & \& \backslash\text{cdot} & \& \backslash\text{amalg} \\
\text{ligne 3} & \& \backslash\text{uplus} & \& \backslash\text{triangleleft} & \& \backslash\text{circ} & \& \backslash\text{odot} & \& \backslash\text{sqcap} \\
\text{ligne 4} & \& \backslash\text{triangleright} & \& \backslash\text{bullet} & \& \backslash\text{ominus} & \& \backslash\text{sqcup} & \& \backslash\text{unlhd} \\
\text{ligne 5} & \& \backslash\text{bigcirc} & \& \backslash\text{oplus} & \& \backslash\text{wedge} & \& \backslash\text{unrhd} & \& \backslash\text{diamond} \\
\text{ligne 6} & \& \backslash\text{oslash} & \& \backslash\text{vee} & \& \backslash\text{bigtriangledown} & \& \backslash\text{times} & \& \backslash\text{otimes} \\
\text{ligne 7} & \& \backslash\text{dagger} & \& \backslash\text{bigtriangleup} & \& \backslash\text{div} & \& \backslash\text{wr} & \& \backslash\text{ddagger} \\
\text{ligne 8} & \& \backslash\text{setminus} & \& \backslash\text{centerdot} & \& \backslash\text{Box} & \& \backslash\text{barwedge} & \& \backslash\text{veebar} \\
\text{ligne 9} & \& \backslash\text{circledast} & \& \backslash\text{boxplus} & \& \backslash\text{curlywedge} & \& \backslash\text{curlyvee} & \& \backslash\text{circledcirc} \\
\text{ligne 10} & \& \backslash\text{boxminus} & \& \backslash\text{Cap} & \& \backslash\text{Cup} & \& \backslash\text{circleddash} & \& \backslash\text{boxtimes}
\end{array}$$


```

<i>ligne1</i>	\perp	\top	$\dot{+}$	\boxdot	\intercal
<i>ligne2</i>	\rightthreetimes	\divideontimes	\square	$\overline{\wedge}$	\backslash
<i>ligne3</i>	\equiv	\leq	\geq	\perp	\cong
<i>ligne4</i>	\prec	\succ	\mid	\neq	\preceq
<i>ligne5</i>	\succeq	\parallel	\sim	\ll	\gg
<i>ligne6</i>	\bowtie	\simeq	\subset	\supset	\Join
<i>ligne7</i>	\approx	\subseteq	\supseteq	\ltimes	\asymp
<i>ligne8</i>	\sqsubset	\sqsupset	\rtimes	\doteq	\sqsubseteq
<i>ligne9</i>	\sqsupseteq	\smile	\propto	\dashv	\vdash
<i>ligne10</i>	\frown	\models	\in	\ni	\notin

Opérateurs (partie 2)

Code : TeX

```


$$\begin{array}{cccccc}
\text{ligne 1} & \& \backslash bot & \& \top & \& \dotplus & \& \boxdot & \& \intercal \\
\text{ligne 2} & \& \rightthreetimes & \& \divideontimes & \& \square & \& \overline{\wedge} & \& \backslash \\
\text{ligne 3} & \& \equiv & \& \leq & \& \geq & \& \perp & \& \cong \\
\text{ligne 4} & \& \prec & \& \succ & \& \mid & \& \neq & \& \preceq \\
\text{ligne 5} & \& \succeq & \& \parallel & \& \sim & \& \ll & \& \gg \\
\text{ligne 6} & \& \bowtie & \& \simeq & \& \subset & \& \supset & \& \Join \\
\text{ligne 7} & \& \approx & \& \subseteq & \& \supseteq & \& \ltimes & \& \asymp \\
\text{ligne 8} & \& \sqsubset & \& \sqsupset & \& \rtimes & \& \doteq & \& \sqsubseteq \\
\text{ligne 9} & \& \sqsupseteq & \& \smile & \& \propto & \& \dashv & \& \vdash \\
\text{ligne 10} & \& \frown & \& \models & \& \in & \& \ni & \& \notin
\end{array}$$


```

<i>ligne1</i>	\approx	\leq	\geq	\lessgtr	\thicksim
<i>ligne2</i>	\lesssim	\gtrsim	\lesseqgtr	\backsim	\lessapprox
<i>ligne3</i>	\gtrapprox	\lesseqgtr	\backsimeq	\lll	\ggg
<i>ligne4</i>	\gtreqless	\triangleq	\lessdot	\gtrdot	\gtreqless
<i>ligne5</i>	\circeq	\lesssim	\gtrsim	\gtrless	\bumpeq
<i>ligne6</i>	\eqslantless	\eqslantgtr	\backepsilon	\Bumpeq	\precsim
<i>ligne7</i>	\succsim	\between	\doteqdot	\precapprox	\succapprox
<i>ligne8</i>	\pitchfork	\thickapprox	\Subset	\Supset	\shortmid
<i>ligne9</i>	\fallingdotseq	\subteqq	\supseteq	\smallfrown	\risingdotseq
<i>ligne10</i>	\sqsubset	\sqsupset	\smallsmile	\varpropto	\preccurlyeq

Opérateurs (partie 3)

Code : TeX

```

$$
\begin{array}{c c c c c c}
\text{ligne 1} & \& \backapprox & \& \leqq & \& \geqq & \& \lessgtr & \& \thicksim \\
\text{ligne 2} & \& \lesssim & \& \gtrsim & \& \lesseqgtr & \& \backsim & \& \lessapprox \\
\text{ligne 3} & \& \gtrapprox & \& \lesseqgtr & \& \backsimeq & \& \lll & \& \ggg \\
\text{ligne 4} & \& \gtreqless & \& \triangleq & \& \lessdot & \& \gtrdot & \& \gtreqless \\
\text{ligne 5} & \& \circeq & \& \lesssim & \& \gtrsim & \& \gtrless & \& \bumpeq \\
\text{ligne 6} & \& \eqslantless & \& \eqslantgtr & \& \backepsilon & \& \Bumpeq & \& \precsim \\
\text{ligne 7} & \& \succsim & \& \between & \& \doteqdot & \& \precapprox & \& \succapprox \\
\text{ligne 8} & \& \pitchfork & \& \thickapprox & \& \Subset & \& \Supset & \& \shortmid \\
\text{ligne 9} & \& \fallingdotseq & \& \subteqq & \& \supseteq & \& \smallfrown & \& \risingdotseq \\
\text{ligne 10} & \& \sqsubset & \& \sqsupset & \& \smallsmile & \& \varpropto & \& \preccurlyeq
\end{array}
$$

```

<i>ligne1</i>	\succcurlyeq	\Vdash	\therefore	\curlyeqprec	\curlyeqsucc
<i>ligne2</i>	\vdash	\because	\blacktriangleleft	\blacktriangleright	\equiv
<i>ligne3</i>	\Hbar	\trianglelefteq	\trianglerighteq	$=$	\neq
<i>ligne4</i>	\triangleleft	\triangleright	\nmid	\ncong	\nleq
<i>ligne5</i>	\ngeq	\nsubseteq	\nmid	\nleqq	\ngeqq
<i>ligne6</i>	\nsupseteq	\nparallel	\nleqslant	\ngeqslant	\nsubseteqq
<i>ligne7</i>	\nshortmid	\nless	\ngtr	\nsupseteqq	\nshortparallel
<i>ligne8</i>	\nprec	\nsucc	\subsetneq	\nsim	\npreceq
<i>ligne9</i>	\nsucceq	\supsetneq	\nVDash	\preccurlyeq	\succcurlyeq
<i>ligne10</i>	\subsetneqq	\nVDash	\precnsim	\succnsim	\supsetneqq
<i>ligne11</i>	\nvdash	\lnapprox	\gnapprox	\varsubsetneq	\ntriangleleft
<i>ligne12</i>	\lneq	\gneq	\varsupsetneq	\ntrianglelefteq	\lneqq
<i>ligne13</i>	\gneqq	\varsubsetneqq	\ntriangleright	\lnsim	\gnsim
<i>ligne14</i>	\nsubseteq	\nsubseteq	\nsubseteq	\nsubseteq	\nsubseteq

Opérateurs (partie 4)

Code : TeX

```

 $\$$ 
\begin{array}{c c c c c c}
ligne 1 & \succcurlyeq & \Vdash & \therefore & \curlyeqprec & \curlyeqsucc \\
ligne 2 & \vdash & \because & \blacktriangleleft & \blacktriangleright & \equiv \\
ligne 3 & \Hbar & \trianglelefteq & \trianglerighteq & = & \neq \\
ligne 4 & \triangleleft & \triangleright & \nmid & \ncong & \nleq \\
ligne 5 & \ngeq & \nsubseteq & \nmid & \nleqq & \ngeqq \\
ligne 6 & \nsupseteq & \nparallel & \nleqslant & \ngeqslant & \nsubseteqq \\
ligne 7 & \nshortmid & \nless & \ngtr & \nsupseteqq & \nshortparallel \\
ligne 8 & \nprec & \nsucc & \subsetneq & \nsim & \npreceq \\
ligne 9 & \nsucceq & \supsetneq & \nVDash & \preccurlyeq & \succcurlyeq \\
ligne 10 & \subsetneqq & \nVDash & \precnsim & \succnsim & \supsetneqq \\
ligne 11 & \nvdash & \lnapprox & \gnapprox & \varsubsetneq & \ntriangleleft \\
ligne 12 & \lneq & \gneq & \varsupsetneq & \ntrianglelefteq & \lneqq \\
ligne 13 & \gneqq & \varsubsetneqq & \ntriangleright & \lnsim & \gnsim \\
ligne 14 & \nsubseteq & \nsubseteq & \nsubseteq & \nsubseteq & \nsubseteq

```

```

ligne 14 & \varsupsetneqq & \ntrianglerighteq & \lvertneqq &
\gvertneqq & \\
\end{array}
$$

```

Accents

<i>ligne1</i>	á	ā	acute{a}	ā	ă
<i>ligne2</i>	ǎ	ă	ǎ	ä	à
<i>ligne3</i>	ä	à	à	â	à
<i>ligne4</i>	â	ã	→	ã	→

Accents (Maths)

Code : TeX

```

$$
\begin{array}{c c c c c c}
ligne 1 & \acute{a} & \bar{a} & \Acute{\Acute{a}} & \Bar{\Bar{a}} & \breve{a} \\
ligne 2 & \check{a} & \Breve{\Breve{a}} & \Check{\Check{a}} & \ddot{a} & \dot{a} \\
ligne 3 & \Ddot{\Ddot{a}} & \Dot{\Dot{a}} & \grave{a} & \hat{a} & \Grave{\Grave{a}} \\
ligne 4 & \Hat{\Hat{a}} & \tilde{a} & \vec{a} & \Tilde{\Tilde{a}} & \Vec{\Vec{a}}
\end{array}
$$

```

Polices mathématiques

Démonstration

Code : TeX

```

\[A B C D E F G H I J K L M N O P Q R S T U V W X Y Z\]
\[ \mathcal{A B C D E F G H I J K L M N O P Q R S T U V W X Y Z} \]
\[ \mathbb{A B C D E F G H I J K L M N O P Q R S T U V W X Y Z} \]
\[ \mathfrak{A B C D E F G H I J K L M N O P Q R S T U V W X Y Z} \]
\[ \mathsf{A B C D E F G H I J K L M N O P Q R S T U V W X Y Z} \]
\[ \mathbf{A B C D E F G H I J K L M N O P Q R S T U V W X Y Z} \]

```

*Rendu**ABCDEFGHIJKLMNOPQRSTUVWXYZ**ABCDEFGHIJKLMNOPQRSTUVWXYZ*

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Polices mathématiques

Caractères de structure

$\$$ $\&$ $\%$ $\#$ $\{$

Caractères de structure

 $\}$ $_$ \sim \wedge \backslash **Code : TeX**

```
$$ \ $ $ $  
$$ \& $ $  
$$ \% $ $  
$$ \# $ $  
$$ \{ $ $  
$$ \} $ $  
$$ \_ $ $  
$$ \sim $ $  
$$ \wedge $ $  
$$ \backslash $ $
```

Divers

<i>ligne1</i>	∞	\forall	\Bbbk	\wp	∇
<i>ligne2</i>	\exists	\bigstar	\angle	∂	\nexists
<i>ligne3</i>	\diagdown	\measuredangle	\eth	\emptyset	\diagup
<i>ligne4</i>	\sphericalangle	\clubsuit	\varnothing	\Diamond	\complement
<i>ligne5</i>	\diamondsuit	\imath	\Finv	\triangledown	\heartsuit
<i>ligne6</i>	\jmath	\vartriangleright	\triangle	\spadesuit	ℓ
<i>ligne7</i>	\hbar	\triangleleft	\cdots	$\int\int\int\int$	\hslash
<i>ligne8</i>	\blacklozenge	\vdots	$\int\int\int$	\Diamond	\blacksquare
<i>ligne9</i>	\ldots	$\int\int$	\mathcal{U}	\blacktriangle	\ddots
<i>ligne10</i>	\sharp	\prime	\textcircled{S}	\natural	\flat
<i>ligne11</i>	\square	\backslash	\Re	\Im	\surd

Divers

Code : TeX

```


$$\begin{array}{cccccc}
\text{ligne 1} & \&\infty & \&\forall & \&\Bbbk & \&\wp & \&\nabla \\
\text{ligne 2} & \&\exists & \&\bigstar & \&\angle & \&\partial & \&\nexists \\
\text{ligne 3} & \&\diagdown & \&\measuredangle & \&\eth & \&\emptyset & \&\diagup \\
\text{ligne 4} & \&\sphericalangle & \&\clubsuit & \&\varnothing & \&\Diamond & \&\complement \\
\text{ligne 5} & \&\diamondsuit & \&\imath & \&\Finv & \&\triangledown & \&\heartsuit \\
\text{ligne 6} & \&\jmath & \&\vartriangleright & \&\triangle & \&\spadesuit & \&\ell \\
\text{ligne 7} & \&\hbar & \&\triangleleft & \&\cdots & \&\int\int\int\int & \&\hslash \\
\text{ligne 8} & \&\blacklozenge & \&\vdots & \&\int\int\int & \&\Diamond & \&\blacksquare \\
\text{ligne 9} & \&\ldots & \&\int\int & \&\mathcal{U} & \&\blacktriangle & \&\ddots \\
\text{ligne 10} & \&\sharp & \&\prime & \&\textcircled{S} & \&\natural & \&\flat \\
\text{ligne 11} & \&\square & \&\backslash & \&\Re & \&\Im & \&\surd
\end{array}$$


```

Autres caractères spéciaux



Ces commandes ne s'utilisent pas dans le mode d'écriture scientifique, même si certaines d'entre elles fonctionnent correctement dans un environnement de ce type.

ligne 1	ó	ö	ô	\	õ	Divers (Mode texte)
ligne 2	ō	ș	ò	ǒ	ö	
ligne 3	ôo	q	o	š	o	
ligne 4	Å	å	ß	ı	ı	
ligne 5	š	ø	š	š	Ø	
ligne 6	¶	§	æ	Æ	†	
ligne 7	‡	©	£	{	}	
ligne 8	%	\$	&	#	_	
ligne 9	^	...				

Code : TeX

```

\begin{tabular}{c c c c c c}
ligne 1 & \'{o} & \"{o} & ^{o} & \textbackslash{} & ~{o} \\
\\
ligne 2 & ={} & \d s & \.{} & \u{} & \H{} \\
\\
ligne 3 & \t{oo} & \c{} & \d{} & \r s & \b{} \\
\\
ligne 4 & \AA & \aa & \ss & \i & \j \\
\\
ligne 5 & \H s & \o & \t s & \v s & \O \\
\\
ligne 6 & \P & \S & \ae & \AE & \dag \\
\\
ligne 7 & \ddag & \copyright & \pounds & \{ & \} \\
\\
ligne 8 & \% & \$ & \& & \# & \_ \\
\\
ligne 9 & ^{} & \dots & & & & \\
\\
\end{tabular}

```

Liste des packages

Depuis le début du livre, vous avez appris à utiliser quelques packages utiles à la création de documents simples sous LaTeX. Cette annexe va reprendre, thème par thème, le nom et le rôle de chacun des packages utilisés dans ce cours, avec, pour chacun d'entre eux, un petit commentaire sur son utilité et le besoin auquel il répond.

Garder cette liste sous la main est un bon moyen d'éviter les erreurs d'étourderie (ou tout simplement les fautes de frappe) lors de la saisie de vos préambules. Gardez tout de même à l'esprit qu'un package s'insère via la commande `\usepackage{ }`.

La liste

Packages de langue

Ces trois packages sont indispensables pour écrire un document en français.

- `inputenc`
- `fontenc`
- `babel`

Création d'un layout

Le layout permet d'afficher le gabarit de mise en page d'un document.

- `layout`

Modification des marges

Les commandes de ce package permettent de régler finement les marges.

- `geometry`

Interligne

- `setspace`

Soulignement

Le package `soul` peut également servir à barrer du texte.

- `soul`
- `ulem`

Symbole euro

- `eurosym`

Pack de polices

N'en inclure qu'un à la fois (il est fortement conseillé d'en utiliser un).

- `bookman`
- `charter`

- newcent
- lmodern
- mathpazo
- mathptmx

Citation d'URL

- url

Citation de code

Le package `verbatim` donne accès aux environnements `verbatim` et `verbatimab`.

- verbatim
- moreverb

Citation de code coloré

Ne pas oublier de paramétrer correctement `\lstset`.

- listings

En-têtes et pieds de pages personnalisés

- fancyhdr

Insertion d'images

Primordial pour travailler sur des images.

- graphicx

Insertion d'une image dans un paragraphe

À utiliser en dernier recours uniquement...

- wrapfig

Manipuler les couleurs et colorer du texte

À employer avec prudence et retenue.

- color

Colorer du texte contenu dans un tableau

Pour l'impression, il vaut mieux utiliser uniquement des niveaux de gris.

- colortbl

Insertion d'expressions scientifiques

Le package `amsthm` n'est utile que si votre document comprend des théorèmes.

- `amsmath`
- `amssymb`
- `mathrsfs`
- `amsthm`

Création d'index

- `makeidx`

Petite astuce

Voici une petite astuce toute simple pour ne pas avoir à vous souvenir de chacun des noms des vingt-huit packages utilisés ici : il suffit de créer un gros bout de code à insérer en préambule et dont chaque ligne est précédée par un signe `%`. Il ne reste qu'à enlever le signe pour qu'une ligne ne soit plus considérée comme un commentaire.

N'hésitez pas à utiliser le code ci-dessous dans vos préambules.

Code : TeX

```
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[français]{babel}
%\usepackage{layout}
%\usepackage{geometry}
%\usepackage{setspace}
\usepackage{soul}
\usepackage{ulem}
%\usepackage{eurosym}
%\usepackage{bookman}
%\usepackage{charter}
%\usepackage{newcent}
%\usepackage{lmodern}
%\usepackage{mathpazo}
%\usepackage{mathptmx}
%\usepackage{url}
%\usepackage{verbatim}
%\usepackage{moreverb}
%\usepackage{listings}
%\usepackage{fancyhdr}
%\usepackage{wrapfig}
%\usepackage{color}
%\usepackage{colortbl}
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{mathrsfs}
%\usepackage{amsthm}
%\usepackage{makeidx}
```

Les gabarits

Si vous l'avez lu en entier, ce livre devrait vous avoir convaincu, grâce aux exemples, de la puissance de LaTeX. Néanmoins, un petit problème se pose : comment retenir les commandes élémentaires ainsi que les éléments de structure à insérer dans un document `.tex` ? Cette annexe rassemble pour vous quatre gabarits de documents afin de faciliter vos compositions de rapports, lettres, livres et articles.

Les différents codes web sont là pour que vous téléchargiez les gabarits sur votre ordinateur et les utilisiez lorsque le besoin s'en fera sentir.

La classe article

Notons ici qu'un article scientifique peut parfois faire plus d'une dizaine de pages (c'est même assez courant). Il est souvent primordial de l'accompagner d'une bibliographie, voire d'une table des matières (Qu'il me semblerait plus judicieux de placer en début de publication dans le cas d'un document de la classe `article`.).

Code : TeX

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

\begin{document}

\maketitle

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

Bla bla bla

\end{document}
```

La classe report

Code : TeX

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

\title{Mon titre d'article}
\date{}
\author{Laleloulilo}
\makeindex
\begin{document}

\maketitle

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

```

\tableofcontents

Bla\index{bla} bla bla

\listoffigures
\listoftables
\printindex
\end{document}

```

La classe book

Code : TeX

```

\documentclass{book}

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

\begin{document}

\maketitle % Page de garde

\frontmatter

Pages introductives

\mainmatter

Contenu

\appendix

Chapitres annexes
\bibliographystyle{} % Le style est mis entre crochets.
\bibliography{bibli} % Mon fichier de base de données s'appelle
bibli.bib.

\backmatter

Epilogue

\tableofcontents
\listoffigures
\listoftables
\printindex

\end{document}

```

La classe letter

Nous avons peu parlé de la classe `letter`, celle-ci peut pourtant s'avérer très utile lors de la rédaction de courriers officiels (ou plus généralement de courriers un peu sérieux). Certains lui préféreront la classe `lettre`, que je trouve un peu moins élégante.

Code : TeX

```

\documentclass{letter}

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

```



```
\usepackage{lmodern}

\signature{M. Laleloulilo (signature)}
\address{Première ligne de l'adresse \\ Deuxième ligne \\ Troisième
ligne}
\begin{document}
\begin{letter}{Un destinataire\\Un autre \\ Son copain & son lapin
\\
ligne d'adresse 1 \\ ligne d'adresse 2 \\ ligne d'adresse 3}
\opening{Formule de politesse d'ouverture,}

Le texte.

\closing{Cordialement, (formule de politesse)}
\ps{P.-S. : Votre petit ajout ! :-)}

\end{letter}

\end{document}
```

Mémento

Nombreux furent les environnements et commandes étudiés tout au long de ce livre. LaTeX en compte au moins cent, voire mille fois plus et pourtant, vous avez déjà du mal à vous souvenir de celles qui ont été vues. De plus, vous n'allez pas vous amuser à relire et à feuilleter les 200 pages d'explications pour retrouver une commande ou une syntaxe.

Pour cette raison, j'ai écrit pour vous cette annexe « Mémento » qui rassemble la majeure partie des commandes vues précédemment ainsi que quelques bonus. Certains éléments ne peuvent que difficilement être résumés sans perdre en précision, dans ce cas un renvoi est fait vers une page de cours.

Les structures des différentes classes de document ne sont pas rappelées ici, elles font l'objet de l'annexe « Gabarits ».

Les essentiels

Squelette minimal

Dans le cas d'une publication en français.

Code : TeX

```
\documentclass{report}
\usepackage[utf8x]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[français]{babel}

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

Appel d'un package

Code : TeX

```
\usepackage{nom du package}
```

Introduction d'un environnement

Code : TeX

```
\begin{nom de l'environnement}
\end{nom de l'environnement}
```

Inclusion d'un fichier

Code : TeX

```
\input{chemin du fichier}
```

La commande \documentclass

Options applicables à la commande <code>\documentclass</code>		
Description	Valeurs applicables	Valeur par défaut
Format du papier	a4paper, a5paper, letterpaper, b5paper...	letterpaper
Mode brouillon	draft	Pas en draft par défaut
Taille de la police principale	10pt, 11pt, 12pt	10pt
Alignement des équations	fleqn (à gauche)	centrées par défaut
Colonnes	onecolumn, twocolumn	onecolumn
Première page des chapitres	openany, openright	openright
Recto verso	oneside, twoside	article et report : oneside, book : twoside

Les références

Les marqueurs sont des éléments invisibles permettant de créer des références (placez d'abord un marqueur avec `\label` puis faites-y référence ailleurs dans le texte).

Code : TeX

```
\label{historique} % Placer un marqueur nommé "historique".
\ref{historique} % Faire une référence dans un texte.
\pageref{historique} % Faire une référence à la page du marqueur.
```

Les éléments de mise en page

Éléments de structure

Commandes des éléments de structure	
Partie	<code>\part{nom de la partie}</code>
Chapitre	<code>\chapter{nom du chapitre}</code>
Section	<code>\section{nom de la section}</code>
Sous-section	<code>\subsection{nom de la sous-section}</code>
Sous-sous-section	<code>\subsubsection{nom de la sous-sous-section}</code>
Paragraphe	<code>\paragraph{nom du paragraphe}</code>
Sous-paragraphe	<code>\subparagraph{nom du sous paragraphe}</code>

Page de garde

Code : TeX

```
\title{Le titre}
\author{Prénom \bsc{Nom}}
\date{\today}
\begin{document}
\maketitle
```

Alignements de texte

Par défaut le texte est justifié. Vous pouvez utiliser :

- l'environnement `flushright` pour aligner votre texte à droite ;
- l'environnement `center` pour centrer votre texte ;
- l'environnement `flushleft` pour aligner votre texte à gauche.

Sauts

Code : TeX

```
\newline % Changement de ligne.  
\\ % Changement de ligne.  
% Changement de page.  
\clearpage % Changement de page après gestion des flottants.  
\cleardoublepage % Idem, mais redémarre sur une page impaire.
```

Interlignes

Pour un interligne 1,5 fois plus grand que celui par défaut, utilisez `\onehalfspacing` dans le préambule ou bien l'environnement `onehalfspace`.

Pour un interligne 2 fois plus grand que celui par défaut, utilisez `\doublespacing` dans le préambule ou bien l'environnement `doublespace`.

Marges

Modification grossière des marges

Exemple avec 3 cm de marge partout :

Code : TeX

```
\usepackage[top=3cm, bottom=3cm, left=3cm, right=3cm]{geometry}
```

Création d'un layout

Code : TeX

```
\documentclass{article}  
  
\usepackage[latin1]{inputenc}  
\usepackage[T1]{fontenc}  
\usepackage[français]{babel}  
\usepackage{layout}  
  
\begin{document}
```

```
\layout  
\end{document}
```

Modifications fines d'un layout

Les noms des longueurs sont disponibles sur le layout.

Code : TeX

```
\setlength{nom de la longueur}{longueur dans l'unité qui vous plait}  
\setlength{\marginparwidth}{2cm}
```

Listes

Listes à puces

Code : TeX

```
\begin{document}  
  
\begin{itemize}  
  
  \item Un canard.  
  \item Un mammouth.  
  \item Un canard.  
  \item Un mammouth.  
  \item Un canard.  
  \item Un mammouth.  
  \item[@] Une pintade. % En plaçant un @ entre crochets après \item,  
    j'ai transformé la puce en @.  
  
\end{itemize}
```

Listes numérotées

Code : TeX

```
\begin{enumerate}  
  
  \item Un canard.  
  \item Un mammouth.  
  \item Un canard.  
  \item Un mammouth.  
  \item Un canard.  
  \item Un mammouth.  
  \item Une pintade.  
  
\end{enumerate}
```

Listes de description

Code : TeX

```

\begin{description}

\item[Un canard :] bestiole qui fait coin.
\item[Un poulpe :] bestiole qui fait bloub.
\item[Un ornithorynque :] bestiole qui fait rire.
\item[Un ours :] bestiole qui fait mal.

\end{description}

```

Les polices

Les tailles

Commande	Taille de texte
<code>\tiny</code>	Minuscule
<code>\scriptsize</code>	Très très petite
<code>\footnotesize</code>	Très petite
<code>\small</code>	Petite
<code>\normalsize</code>	Normale (définie dans <code>\documentclass</code>)
<code>\large</code>	Légèrement plus grande que la normale
<code>\Large</code>	Grande
<code>\LARGE</code>	Très grande
<code>\huge</code>	Très très grande
<code>\Huge</code>	Énorme !

Les couleurs

Code : TeX

```

\textcolor{couleur}{texte en couleur}

```

Création de couleur en niveaux de gris :

Code : TeX

```

\definecolor{le nom choisi sans espace}{gray}{le nombre à deux
décimales}

```

Création de couleur via un mélange de rouge, de vert et de bleu :

Code : TeX

```

\definecolor{nom}{rgb}{taux de rouge entre 0 et 1,taux de vert,taux

```

```
de bleu}
```

Changement de police ponctuel

Code : TeX

```
{\fontfamily{code_de_la_police}\selectfont mon bout de texte}
```

Code de la police	Exemple
bch	Charter
cmr	Computer Modern
lmr	Latin Modern Roman
lmss	Latin Modern Sans Empattement
lmssq	Latin Modern Sans Emp. Expansé
lmtt	Latin Modern Typewriter
pag	Avant Garde
pcr	Courier
pbk	Bookman
phv	Helvetica
pnc	New Century Schoolbook

ppl	Palatino
ptm	Times
pzc	<i>Zapf Chancery</i>

Les packs de polices

- bookman
- charter
- newcent
- lmodern
- mathpazo
- mathptmx

Les effets

Mise en forme		
Modification	Commande	Rendu
Normal	<code>{\normalfont un lapin}</code>	un lapin
	<code>\begin{rm}un lapin \end{rm}</code>	un lapin
Gras	<code>\textbf{un lapin}</code>	un lapin
	<code>{\bfseries un lapin}</code>	un lapin
	<code>\begin{bf}un lapin \end{bf}</code>	un lapin
Italique	<code>\textit{un lapin}</code>	<i>un lapin</i>
	<code>{\itshape un lapin}</code>	<i>un lapin</i>
	<code>\begin{it}un lapin \end{it}</code>	<i>un lapin</i>
Penché	<code>\textsl{un lapin}</code>	<i>un lapin</i>
	<code>{\slshape un lapin}</code>	<i>un lapin</i>
	<code>\begin{sl}un lapin \end{sl}</code>	<i>un lapin</i>
Machine à écrire	<code>\texttt{un lapin}</code>	un lapin
	<code>{\ttfamily un lapin}</code>	un lapin

	<code>\begin{tt}un lapin \end{tt}</code>	un lapin
Petites majuscules	<code>\textsc{un lapin}</code>	UN LAPIN
	<code>{\scshape un lapin}</code>	UN LAPIN
	<code>\begin{sc}un lapin \end{sc}</code>	UN LAPIN
Exposant (En mode texte)	Un canard <code>un lapin</code>	Un canard ^{un lapin}
Encadrer (Paramétrable)	<code>\fbox{un lapin}</code>	<div style="border: 1px solid black; padding: 2px;">un lapin</div>
Soulignement (Package soul)	<code>\ul{un lapin}</code>	<u>un lapin</u>
Soulignement (Package ulem)	<code>\uuline{un lapin}</code>	<u><u>un lapin</u></u>
	<code>\uwave{un lapin}</code>	<u><u>un lapin</u></u>
Barrer (Package soul)	<code>\st{un lapin}</code>	un lapin

Les notes

Citations simples

L'environnement `quote` pour des citations courtes et `quotation` pour des citations de plusieurs lignes.

Citations d'URL :

Code : TeX

```
\url{adresse}
```

La commande `\fbox`

Code : TeX

```
% Commande permettant de définir l'écart.
\setlength{\fboxsep}{8mm}
% Commande permettant de définir l'épaisseur du trait.
\setlength{\fboxrule}{2mm}
\fbox{Un lapin}
```

Création de minipage

Code : TeX

```
\begin{minipage}[c]{5cm}
Un petit paragraphe.
```

```
\end{minipage}
```

Les figures

Insertion simple

Code : TeX

```
\includegraphics{chemin de l'image}  
\includegraphics[width=300]{chemin de l'image}  
\includegraphics[height=250]{chemin de l'image}  
\includegraphics[height=250, width=300]{chemin de l'image}  
\includegraphics[scale=1.2]{chemin de l'image}
```

Rotation

Code : TeX

```
\includegraphics[angle=52]{chemin de l'image}
```

Inclusion dans un paragraphe

Code : TeX

```
\begin{wrapfigure}[nombre de lignes]{placement}{largeur de l'image  
en cm}  
\includegraphics[width=largeur en cm]{chemin de l'image}  
\end{wrapfigure}  
Votre paragraphe, sans saut de ligne entre \end et le paragraphe.
```

Découpage d'une image

Code : TeX

```
\includegraphics*[abscisse du coin inférieur gauche,ordonnée du coin  
inférieur gauche][abscisse du coin supérieur droit,ordonnée du coin  
supérieur droit du rectangle]{chemin de l'image}
```

Image flottante

Code : TeX

```
\begin{figure}  
\begin{center}  
\includegraphics{chemin de l'image}  
\end{center}  
\caption{Légende}
```

```
\label{Référence}
\end{figure}
```

ou

Code : TeX

```
\begin{figure}
\center
\includegraphics{chemin de l'image}
\caption{Légende}
\label{Référence}
\end{figure}
```

Les tableaux

Structure type

Code : TeX

```
\begin{table}
\begin{center}
\begin{tabular}{|c|c|}
\hline
1 & 2 \\
\hline
3 & 4 \\
\hline
\end{tabular}
\end{center}
\caption{votre légende}
\label{votre référence}
\end{table}
```

Fusion de lignes

Code : TeX

```
\multirow{nombre de lignes fusionnées}{taille de la colonne en
cm}{votre texte}
\multirow{nombre de lignes fusionnées}*{votre texte}
```

Fusion de colonnes

Code : TeX

```
\multicolumn{nbre de col fusionnées}{c, l ou r }{le texte}
```

Coloration de cellules

Utilisable avec les couleurs personnalisées.

- `\columncolor{couleur}` pour colorer les colonnes.
- `\rowcolor{couleur}` pour colorer les lignes.
- `\cellcolor{couleur}` pour colorer les cellules.

Les mathématiques

Les formules simples

Fraction

Code : TeX

```
\[\frac{numérateur}{dénominateur}\]
```

Fractions en cascade

Code : TeX

```
\[\cfrac{num1}{den1} + \cfrac{num2}{den2}\]
```

Racine

Code : TeX

```
\[\sqrt[15]{nombre}\]
```

Limite

Code : TeX

```
\[\lim_{nbr1 \to nbr2} x+42\]
```

Exposant

Code : TeX

```
\[nombre^{exposant}\]
```

Indice

Code : TeX

```
\[nombre_{indice}\]
```

Intégrales simples

Code : TeX

```
\[ \int {contenu} \]  
\[ \int_{borne inférieure}^{borne supérieure} {contenu} \]  
\[ \int\limits_{borne inférieure}^{borne supérieure} {contenu} \]
```

Pour faire des intégrales doubles ou triples, utilisez respectivement les commandes `\iint` et `\iiint`. Vous pouvez mettre plusieurs `\int` à la suite, mais mathématiquement cela ne représente pas la même chose.

Intégrale sur une courbe fermée

Code : TeX

```
\[ \oint {contenu} \]
```

Symbole de la somme

Code : TeX

```
\[ \sum_{en dessous}^{au dessus} contenu \]
```

Symbole du produit

Code : TeX

```
\[ \prod_{en dessous}^{au dessus} contenu \]
```

Les systèmes d'équation

Code : TeX

```
\begin{align}  
y &= x - y + z \\ x &= y \\ z &= y  
\end{align}  
  
\begin{align*} % Sans numérotation.  
y &= x - y + z \\ x &= y \\ z &= y  
\end{align*}
```

```
\begin{align}
y &=& x - y + z \\
x &=& y \\
z &=& y \\
\end{align}
```

Les matrices

Utilisation classique

Code : TeX

```
\[
\begin{matrix}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{matrix}
\]
```

Il est aussi possible d'utiliser les environnements `pmatrix`, `vmatrix`, `Vmatrix`, `bmatrix` et `Bmatrix` selon le type de délimiteur souhaité.

La commande `\phantom`

Code : TeX

```
\[
\begin{matrix}
\phantom{9}1 & 12345 & \phantom{-}3 \\
94 & \phantom{1234}5 & -6 \\
\phantom{9}7 & \phantom{1234}8 & \phantom{-}9
\end{matrix}
\]
```

Les théorèmes, lemmes, etc.

Dans le préambule :

Code : TeX

```
\newtheorem*{petit_nom}{Réciproque}
\newtheorem{petit_nom1}{Définition}
\newtheorem{petit_nom2}{Proposition}[chapter]
\newtheorem{petit_nom3}{Corollaire}[section]
```

Ensuite chaque nouvel environnement défini est utilisé :

Code : TeX

```
\begin{petit_nom}
\end{petit_nom}
```

ou

Code : TeX

```
\begin{petit_nom}[de quelque chose]
\end{petit_nom}
```

Les espaces

Espace	Code
négative	<code>\!</code>
fine	<code>\,</code>
normale	<code>\</code> suivi d'une espace
moyenne	<code>\;</code>
large	<code>\:</code>
cadratin	<code>\quad</code>
double cadratin	<code>\qquad</code>

Les tables

Code : TeX

```
\tableofcontents % Table des matières.
\listoffigures % Table des figures.
\listoftables % Liste des tableaux.
```

Index

Il faut insérer la commande `\makeindex` dans le préambule ainsi que `\printindex` à l'endroit où vous souhaitez mettre en page votre index.

Code : TeX

```
\index{mot} % ajoute l'entrée « mot »
\index{pate@pâté} % « pâté » correctement placé alphabétiquement
\index{mot|see{lapin}} % « mot, voir lapin »
```

Bibliographie

Base de données

Code : TeX

```
@type_d'œuvre{référence_courte,
Titre_du_champ1={texte de votre choix},
Titre_du_champ2={texte de votre choix},
Titre_du_champ3={texte de votre choix},
Titre_du_champ4={texte de votre choix},
```


}

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

Champs	
Champs	Correspondance
author	Auteur
journal	Journal
month	Mois de publication, au choix (jan, feb, mar, apr...)
note	Notes complémentaires
organization	Société ou organisation
pages	Pages concernées
publisher	Éditeur
year	Année de publication

Insertion de la bibliographie

Code : TeX

```
\bibliographystyle{} % Le style  
\bibliography{fichier sans extension} % Le fichier de base de  
données.
```

- `\bibliographystyle{plain}` classe les entrées par ordre alphabétique et les numérote en conséquence.
- `\bibliographystyle{abbrv}` classe les entrées par ordre alphabétique, les numérote en conséquence et abrège certains éléments de la bibliographie.
- `\bibliographystyle{unsrt}` trie les entrées par ordre d'apparition dans le texte.
- `\bibliographystyle{alpha}` : le repère n'est plus un chiffre, mais les trois premières lettres du nom de l'auteur accolées aux deux derniers chiffres de l'année de parution.

Aller plus loin

Avant de vous quitter, j'ai souhaité mettre à votre disposition une annexe d'ouverture, une sorte de conclusion vous permettant de poursuivre votre apprentissage de LaTeX facilement et ce, sans dépenser un sou de plus.

L'idée ici est de vous orienter vers différentes pistes et de vous présenter des idées et points de vue qui vous permettront de vous perfectionner dans la pratique de LaTeX.

Typographie et encodage Latin-1 et UTF-8

Un débat a fait rage à diverses étapes de la composition de ce livre : quel encodage de caractères vous apprendre à utiliser ? UTF-8 ou Latin-1 ? Comment vous transmettre les bonnes pratiques sans pour autant vous rendre la vie impossible ? Telles étaient les questions, je vais ici y répondre.

D'abord, si vous étiez endormi à la lecture du passage concerné, sachez que l'encodage est, grossièrement, le nom donné au processus qui transforme vos caractères en code informatique. Plusieurs processus ont en effet été développés, et le standard actuel est l'UTF-8 car il gère très bien plusieurs dizaines de langues ainsi que des centaines de caractères spéciaux.

Vous en déduirez que le Latin-1 a des lacunes dans ces domaines (Comme vous avez pu le lire dans le TP 1, il n'y a pas besoin de remplacer le caractère œ par une commande en Latin-1.) et commencerez doucement à vous demander pourquoi je l'ai recommandé aux utilisateurs de Windows et de Mac.

La raison est simple : généralement, les utilisateurs de Mac et de Windows échangent des foules de fichiers et Windows a encore beaucoup de mal avec l'UTF-8. La probabilité est donc extrêmement forte que vous ayez à travailler avec des personnes utilisant encore le Latin-1, même si c'est mal.



Je suis sous Windows et une personne me communique un document en UTF-8 à modifier, que faire ?

Tout simplement télécharger la version 2 de TeXnicCenter, qui fonctionne très bien, même si elle n'est pas encore totalement stable. Une fois qu'il sera en version finale, ce programme comblera le retard pris par Windows et permettra à tous les utilisateurs de LaTeX de travailler en UTF-8. Bien sûr, cela prendra du temps, beaucoup de temps...

Si vous voulez le tester, cliquez [ici](#).

L'intégriste est partout

Vous aimez les débats qui tachent, le sang qui coule, et comprendre les luttes mesquines entre pro et anti `eqnarray` vous intéresse ? J'ai sélectionné pour vous [un document](#) qui liste les méfaits de cet environnement ainsi que de nombreux autres. Il s'agit d'un texte à lire pour se sentir malin dans un cadre scientifique et ne pas se faire avoir en produisant ses premiers écrits sérieux. Ce document explique aussi pourquoi « `$$` » n'est plus beaucoup employé pour écrire des expressions scientifiques.

De l'art de la présentation

Continuons dans l'ouverture : je vous propose ici deux manuscrits de typographie ([celui-ci](#) et [celui-là](#)) qui vous permettront, l'un comme l'autre, de comprendre quelles sont les choses à faire (et à ne surtout pas faire) pour préserver notre langue et être crédible typographiquement face à des magazines ou un lectorat très rigoureux sur l'écriture et le respect des conventions.

Les cours d'approfondissement

Voici une petite sélection de divers cours à lire si vous voulez approfondir vos connaissances. À votre niveau actuel, aucun ne présente de difficulté particulière (LaTeX utilise toujours les mêmes notions de commandes, d'environnements et de packages ; il n'y a pas grand-chose à apprendre de plus. Si vous aimez le challenge, vous pouvez tenter de créer vos propres commandes et packages.). Dans l'ordre des codes web, je vous ai mis :

1. un exemple de [cours sous forme de code commenté](#) ;
2. le [tutoriel de formation à LATEX de l'ENS](#) (un bon niveau général est requis) ;
3. le [Framabook](#) Tout ce que vous avez toujours voulu savoir sur LATEX sans jamais oser le demander ;
4. le [Wikibooks en français](#), très bien construit, traitant de LaTeX. Il donne des exemples intéressants d'utilisation du

- langage en chimie ;
- 5. le [Wikibooks en anglais](#), un peu plus fourni ;
- 6. le [cours de Ukonline](#), très répandu chez les débutants ;
- 7. une [courte introduction à LaTeX](#), véritable perle de pédagogie.

Les incontournables

Quatre outils fort utiles :

1. [Le conjugueur](#), véritable Bescherelle informatique ;
2. [un aide-mémoire supplémentaire](#) ;
3. une [table](#) listant pratiquement tous les caractères spéciaux ;
4. le guide "[Détection et résolution des problèmes](#)", outil salvateur lorsqu'une erreur de compilation inconnue fait surface.

Beamer

Si vous avez eu le courage de pratiquer, la classe de documents `beamer` vous tend les bras. Cette classe est conçue pour produire des diapositives et se présente comme un concurrent très sérieux de PowerPoint dans la sphère scientifique.

1. [La page Beamer](#) sur Wikipedia.
2. [Introduction à Beamer](#) sous Linux.
3. [Cours plus complet en anglais](#).

Les interfaces graphiques alternatives

Si vous êtes lassé de Kile, de TeXnicCenter ou de TeXShop, prenez le temps de tester les trois logiciels présentés ci-dessous.

1. [L'éditeur LaTeX](#) en ligne de Google.
2. [Lyx](#), le presque traitement de texte en LaTeX.
3. [Emacs](#), la console par excellence.

Au plaisir de vous lire un jour prochain, car il ne faut pas l'oublier, LaTeX n'est qu'un écran pour vos créations.

Ce tutoriel comportera 3 grandes parties et sera écrit en partie au sein de L'IMB (institut de mathématique de Bordeaux) situé à l'intérieur de l'université de Bordeaux I. Cette fac a la spécificité d'être depuis quelques années classée par l'université de Shanghai comme faisant partie de ce qui se fait de mieux dans le domaine informatique (elle a atteint le 8ème rang mondial).

Je garderai un pc sous la main pour vous parler via twitter ([@laleloulilo](#)), alors n'hésitez pas, vos remarques seront le plus possible prises en compte.

En plus, écrire en papotant, c'est plus agréable. 😊