



Rédigez des documents de qualité avec LaTeX

Par Laleloulilo



www.siteduzero.com

Sommaire

Sommaire	1
Lire aussi	2
Rédigez des documents de qualité avec LaTeX	4
Partie 1 : Découverte de LaTeX	5
Qu'est ce que LaTeX ?	6
LaTeX, quésaco ?	6
Word Vs LaTeX	6
Comparatif Général	6
Les avantages de LaTeX	8
LaTeX et la rétro-compatibilité	8
LaTeX et les publications	9
LaTeX dans les publications scientifiques	9
LaTeX dans le monde de l'éducation	9
LaTeX et les présentations	9
Historique rapide	9
Installer LaTeX	11
De quels logiciels avons nous besoin ?	11
Installation sous Windows	12
MiKTeX, la distribution LaTeX	12
Ghostscript et Adobe Reader, vos lecteurs de postscript et de pdf	14
TeXnicCenter, L'interface graphique	14
Installation sous Mac	15
Installation sous Linux	17
Structurer son premier document	18
La compilation	19
Rédaction d'un document simple	20
Rédaction du fichier .Tex	20
Compilation	21
Les résultats	23
Type de document et caractères spéciaux	23
Les caractères spéciaux	23
Types de documents et portions de texte	24
Les packages	26
La philosophie des packages	27
Comment s'en servir?	27
Comment installer un package ?	29
MiKTeX et son installation à la volée	29
Autres distributions	31
Partie 2 : Utilisation basique de LaTeX	32
Maitriser sa mise en page (1/2)	33
Structure des documents	33
Généralités	33
Modification de la numérotation des éléments de structure	35
Page de garde	37
La théorie	37
La pratique	37
Alignements de texte et sauts	38
Alignements de texte	38
Sauts	39
Maitriser sa mise en page (2/2)	41
La forme de votre publication	42
La commande <code>\documentclass{}</code>	42
Exemples pratiques	43
Marges et interlignes	43
Marges	43
Interlignes	45
Les listes	46
listes à puces	46
listes numérotées	47
listes de description	48
Les styles	49
Les polices	49
Graisie, style, taille	50
Taille de texte	50
Graisie, soulignement	51
La commande <code>\emph{}</code> , un cas à part	52
Couleur	53
Les 8 couleurs par défaut	53
Création de nouvelles couleurs	54
Les packs de polices	54
Les notes	58
Citation, code brut et url	59
Les citations simples	59
Les citations de code	60

Gestion des urls	62
Minipage et texte encadré	62
La commande \fbox	62
L'environnement minipage	63
Notes de bas de pages, références internes	64
Notes de bas de pages	64
Les références internes	65
TP : Mise en page de la Déclaration Universelle des Droits de l'Homme en Latex	66
Les consignes	67
Les contraintes de l'exercice	67
La correction	72
Etape 1 : Le type de document	72
Etape 2 : la langue	72
Etape 3 : les autres packages	73
Etape 4 : la page de garde	73
Etape 5 : le squelette complet	74
Etape 6 : le contenu	76
Etape 7 : rassembler les morceaux	84
Conclusion	90
Aller plus loin	90
Commandes et espaces	90
Marges et Layout	90
Personnalisation de la commande \emph{}	95
En-têtes et pieds de pages customisés	95
Partie 3 : Utilisation avancée de LaTeX	98
Les figures	98
Les différents formats d'images	98
Insertion d'images et de figures	99
Insertion simple	99
Intégration d'une image dans un paragraphe	104
Les flottants	105
L'environnement figure	105
Les légendes	107
Sauts de page	107
Les tableaux	108
Structure type d'un tableau	109
La théorie	109
Exercice	111
Fusion de cellules	112
Fusion de colonnes	112
Fusion de lignes	112
Autres paramètres applicables à un tableau	113
Modifications mineures	113
Utilisations de commandes et d'environnements dans un tableau	115
Colorer des cellules	116
L'environnement table, un environnement flottant	117
Les Mathématiques	119
Saisir une expression mathématique	120
Fonctions usuelles	121
Fonctions usuelles à arguments	121
Fonctions usuelles sans argument	122
Les intégrales	123
Préambule technique	123
Intégrales et formules complexes	124
Les systèmes d'équations	126
Mise en place d'un système	126
Les délimiteurs	127
Les délimiteurs : usage avancé	129
Les matrices	130
Mise en forme type théorème	133
Généralités	133
Pratiquons !	134
Flèches, symboles étirables et espaces	136
Les espaces	137
Les accents et chapeaux divers	138
Sommaire et index	140
Tables des matières	141
Table des matières simple	141
Paramétrage d'une table des matières	143
Tables des figures et tableaux	146
Les index	148
Théorie	149
Pratique	150
La Bibliographie	151
Principe d'une bibliographie	152
La base de données	153
Mise en place de la bibliographie	156
Concept	156
Exemples	156
Compilation	158
TP 2 : Rédaction d'un article présentant le théorème de Pythagore	160

Les consignes	161
La structure	161
Les contenus	161
Petite aide	164
La correction	164
Être plus à l'aise avec LaTeX !	168
Écrire un document dans plusieurs fichiers .tex	168
L'option draft	170
Les unités de longueur	170
Les espaces	171
Partie 4 : Annexes	171
Les caractères spéciaux	172
Lettres grecques (Mode mathématique)	172
Opérateurs extensibles (Mode mathématique)	173
Fonctions (Mode mathématique)	173
Flèches (Mode mathématique)	174
Accents (Mode mathématique)	175
Opérateurs (Mode mathématique)	175
Polices mathématiques (Mode mathématique)	180
Caractères de structure (Mode mathématique)	180
Divers (Mode mathématique)	181
Caractères spéciaux (Mode texte)	182
Liste des packages	183
La liste	184
Petite astuce	184
Aller plus loin	186
Typographie et encodage	186
Latin-1 et UTF-8	186
L'intégriste est partout	186
De l'art de la présentation	186
Les cours d'approfondissement	186
Les incontournables	187
Beamer	187
Les interfaces graphiques alternatives	187



Rédigez des documents de qualité avec LaTeX



Par

Laleloulo

Mise à jour : 15/10/2010

Difficulté : Facile  Durée d'étude : 15 jours

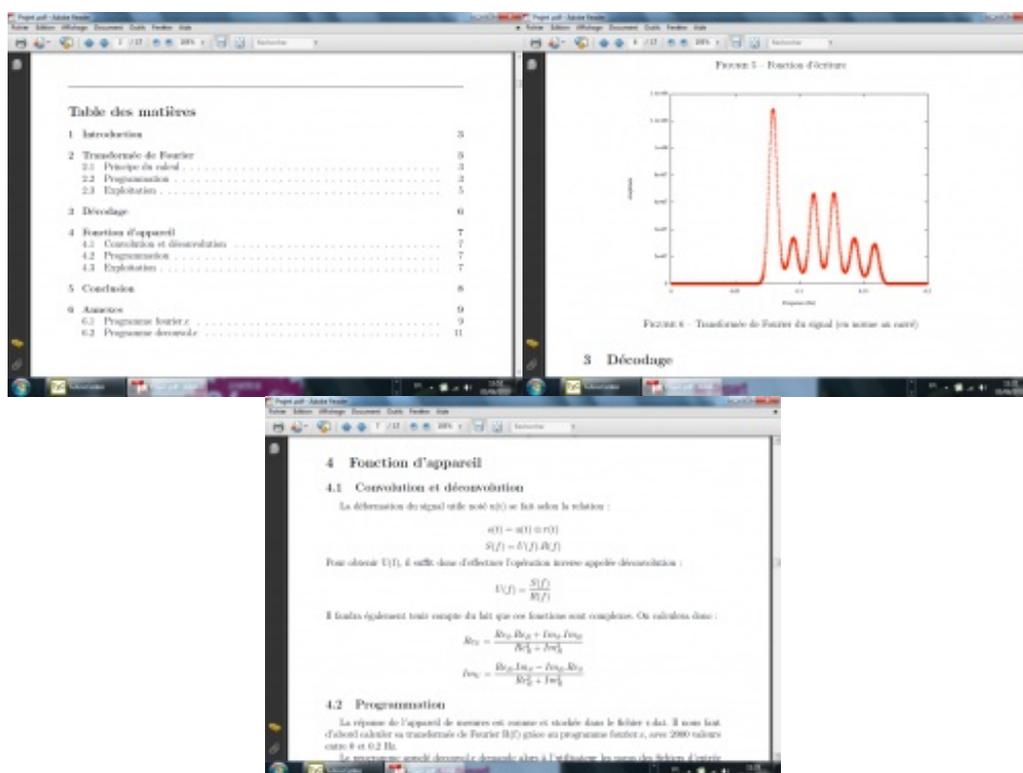
17 515 visites depuis 7 jours, classé 19/777

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 :



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 ?

Bonjour et bienvenue dans ce premier chapitre du tutoriel Latex.

Je m'appelle Laleloulilo, auteur tout neuf du site du zéro, je serai votre guide à travers ce cours.

Présenter de façon professionnelle des documents avec Latex demande un apprentissage, mais pas de grandes connaissances préalables. Ce tutoriel démarrera de zéro et toutes les notions nécessaires à votre apprentissage seront développées, avec bien entendu des exercices et des TP venant ponctuer le cours et vous permettre de vous confronter à votre progression.

Lire ce tutoriel et vous former à Latex devrait vous prendre environ 2 semaines, l'important étant d'aller à votre rythme. Peu importe que l'apprentissage soit plus rapide ou plus lent, prenez votre temps, personne n'est pressé. 😊

Ce premier chapitre est une présentation générale de Latex et de ses usages.

LaTeX, qué-saco ?



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

Un soin tout particulier a été consacré à l'élaboration de commandes permettant d'écrire très facilement d'impeccables formules mathématiques. Une formule compliquée demande rarement plus d'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.

Word Vs LaTeX

Nous allons ici développer les différences entre LaTeX et Word à travers un comparatif global de leurs caractéristiques, suivi par un résumé traitant des avantages ainsi que de la rétrocompatibilité de 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 et demeure l'outil le plus développé à l'heure actuelle. La concurrence est rude entre les suites logicielles et il y a fort à parier qu'au fil des ans LaTeX et Word se rapprocheront énormément au niveau de la facilité d'utilisation et de l'optimisation de la prise en main.





LaTeX fait peur aux débutants avec son interface austère et Word n'aide pas les débutants à réellement apprendre à se servir correctement de ses outils.

Ces deux problèmes et leurs gestions seront prépondérants dans la progression respective des deux logiciels sur le marché informatique.

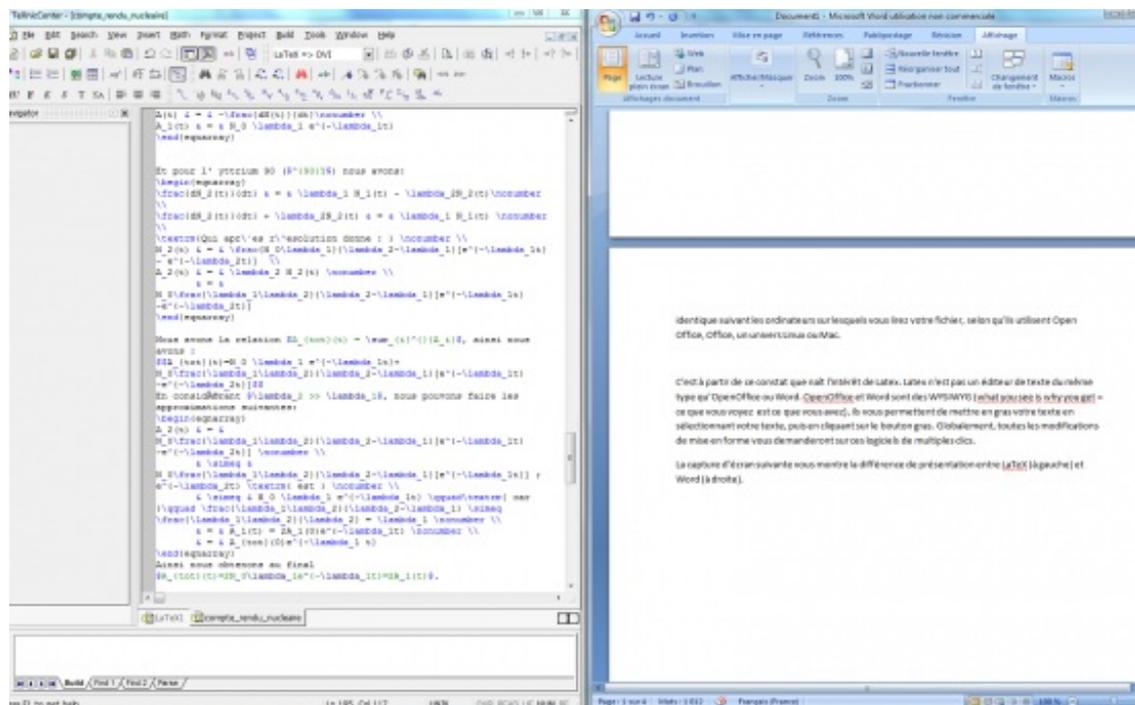
Concrètement, les modifications deviennent vite un cauchemar une fois qu'on a sélectionné et modifié des dizaines de petits éléments. Cela n'est qu'un exemple, mais dans vos publications, vous avez sûrement déjà eu besoin de :

- mettre en page des images
- insérer des légendes
- écrire des formules mathématiques (extrêmement fastidieux)
- dessiner des schémas
- créer des tableaux

Autant de tâches indispensables mais qui malgré tout prennent du temps (même si elles sont faites dans les règles de l'art). A un point tel que nous pouvons estimer que l'écriture représente moins de 50% du temps passé sur un long document pour un utilisateur avancé.

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 ou Word. OpenOffice et Word sont des WYSIWYG (what you see is what you get = ce que vous voyez est ce que vous avez), ils vous permettent de mettre en gras votre texte en sélectionnant votre texte, puis en cliquant sur le bouton gras. Globalement, toutes les modifications de mise en forme vous demanderont sur ces logiciels de multiples clics.

La capture d'écran suivante vous montre la différence de présentation entre LaTeX (à gauche) et Word (à droite).



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 créez (livre, article, lettre, cv...), 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 importantes 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, placera 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, vous permettant de créer du contenu sans avoir à vous soucier de la mise en page. L'idée de taper des balises et des commandes en plein milieu du texte au kilomètre peut sembler étrange au début, mais vous découvrirez très vite qu'elle vous simplifie la vie.



L'optimisation de la création de documents typographiquement propres reste le principal argument d'utilisation de LaTeX. En deuxième rang nous pourrions citer la gratuité, bien qu'OpenOffice puisse s'en vanter lui aussi.

Le logo de LaTeX est le suivant :

Une attention toute particulière a été apportée à la facilité d'écriture des formules mathématiques et autres écrits scientifiques, souvent enfantés dans la douleur avec des outils du type de l'éditeur d'équation de Word. Nous apprendrons bien entendu à maîtriser cette partie de LaTeX.



Information : il existe des éditeurs d'équation performants permettant de copier coller des formules au sein de Word, néanmoins LaTeX intègre nativement ce genre d'outil et il n'est pas nécessaire d'installer plusieurs programmes différents pour réaliser toutes les opérations souhaitées sur un schéma ou une formule.

Les avantages de LaTeX

Il est possible que dans un élan de folie un utilisateur trouve un élément de mise en page peu adapté ou souhaite tout simplement changer la typographie d'un titre. Nous nous retrouvons donc dans une problématique similaire à celle développée tout à l'heure : comment modifier un style de titre présent sur 60 voir 100 ou 200 pages de texte sans avoir à y passer des heures ? LaTeX vous permet de changer les caractéristiques de tous vos titres en une fois, ceci ayant pour effet d'appliquer votre modification sur toutes vos pages.

Sachez que sur LaTeX, tout est modifiable et paramétrable. Le logiciel vous donnera une base 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. 😊

Une discussion avec les zéros (visible en commentaire) rajoute de nouveaux arguments à l'utilisation de LaTeX.

Citons en vrac :

- la gestion des notes
- la gestion des flottants (dont nous parlerons dans la troisième partie du cours)
- la possibilité de hiérarchiser facilement du texte en parties, chapitres, sections...
- la gestion des longs documents
- les bibliographies, index et tables des matières faciles à créer.

LaTeX et la rétro-compatibilité

Rétro-compatibiliqoui? 😬. La rétro-compatibilité désigne la capacité pour un logiciel de produire des fichiers lisibles par ses versions précédentes. Vous avez sûrement déjà été confronté au problème du plantage d'une présentation sur PowerPoint causé par une différence de version. Si ça n'est pas le cas vous aurez sûrement déjà pu constater qu'un changement de pc pouvait

miraculeusement décaler tous les paragraphes de votre cher rapport et rendre tous les schémas illisibles (l'expérience démontre que l'individu lambda se sent très bête dans ce genre de cas). Ces deux problèmes sont des problèmes de non rétro-compatibilité, causés par des logiciels ne produisant pas des fichiers lisibles par leurs versions précédentes.

Sur LaTeX, ce genre de problème ne se pose pas. Un document écrit sur un Pc A aura exactement le même rendu sur un pc B, quelle que soit la version de LaTeX utilisée et le système d'exploitation (Linux, Windows, Mac). Ajoutons à cela qu'il est souvent possible de produire des documents en pdf avec LaTeX (le pdf est un format international de mise en page).

Suffisamment rare pour être noté n'est ce pas ?



Les suites logicielles classiques de traitement de texte intègrent depuis peu la fonction de création de pdf. L'initiative de LaTeX semble avoir contaminé de nouveaux territoires. 😊

LaTeX et les publications

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

- i. la rétro-compatibilité
- ii. la création de pdf
- iii. la possibilité de pouvoir lire ses documents sur tous les systèmes d'exploitation
- iv. des outils permettant la modification rapide de l'ensemble des éléments formant la mise en page
- v. un programme donnant la possibilité de pouvoir (enfin) passer plus de temps à écrire du contenu que de temps à le mettre en forme.

Plein d'avantages qui ont permis à LaTeX d'être promu chouchou du monde de l'édition et du monde scientifique. Tant et si bien que bon nombre de livres sont édités en LaTeX, des professionnels de ce langage se faisant payer pour mettre en page des livres de manière professionnelle au sein des maisons d'édition.

Nous allons voir ici les applications de ce langage dans les publications et le milieu scientifique.

LaTeX dans les publications scientifiques

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 légions et chaque semaine des dizaines de revues scientifiques trient et mettent en page les articles traitant des nouvelles innovations et découvertes. En somme si la science fait avancer le monde, LaTeX en est à l'heure actuel son support écrit de prédilection.

D'ailleurs si vous avez déjà lu l'un des « Livres du zéro » sachez qu'ils sont écrits en LaTeX, comme la majorité des fascicules et livres de cours que vous avez pu consulter par le passé.



LaTeX dans le monde de l'éducation

Actuellement je suis étudiant à l'université des sciences de Bordeaux 1 et sachez qu'à partir de la première année de Master, il est obligatoire dans certaines matières (surtout la physique et les maths) de rendre ses rapports de stage, mémoires et autres textes au format LaTeX. Il est aussi utilisé de façon extrêmement fréquente dans les laboratoires de recherche grâce à la facilité qu'il a d'être exporté et lu sur les ordinateurs de tous pays.

LaTeX et les présentations

Sachez que LaTeX ne permet pas seulement de faire des rapports, il peut par exemple vous permettre de créer une présentation contenant des slides (diapositives) de qualité incomparable, et bien plus encore, mais nous en parlerons plus loin dans ce cours.

Historique rapide

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

Tout débute en ce qui nous concerne dans ce cours par l'invention par Donald Erwin Knuth (chercheur américain en informatique né en 1938) du langage Tex, invention déclenchée en 1977 par le ras le bol suscité par la mise en page plutôt laide de ses articles mathématiques publiés par l'American Mathematical Society.

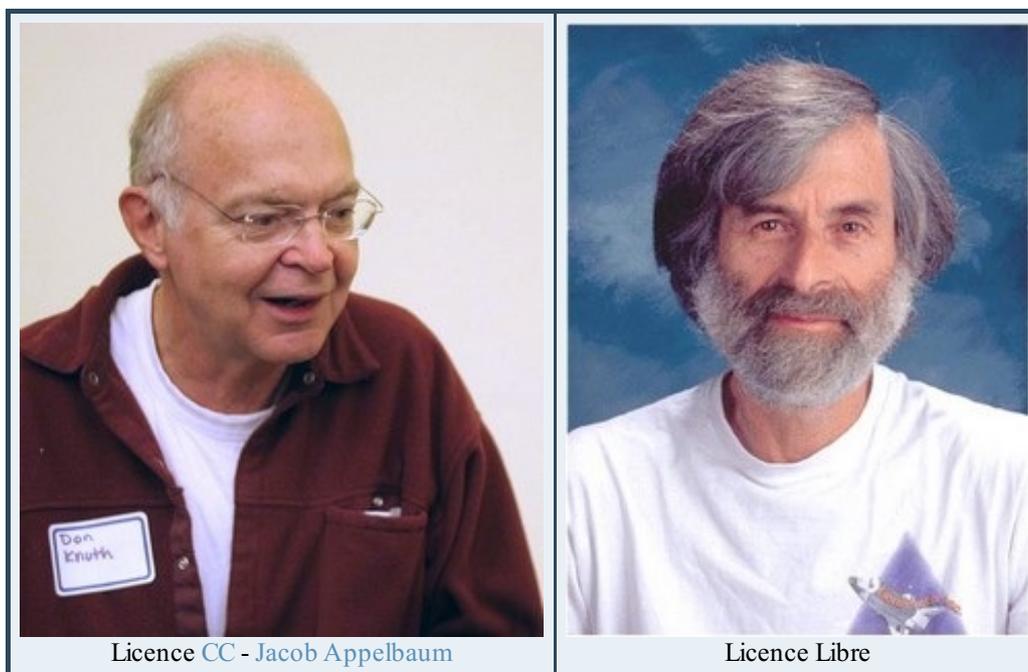
Comme souvent dans le monde informatique, un besoin a permis la création d'une innovation, en l'occurrence : Tex. Le langage avait été conçu non pas pour produire de beaux documents, mais plutôt pour accroître la lisibilité et optimiser l'insertion de

formules mathématiques.

Nous ne sommes pas des typographes ou des experts en mise en page, mais les participants à la création de Tex le sont et ont créé un ensemble cohérent de commandes, permettant à l'utilisateur d'obtenir un résultat professionnel.

LaTeX est une sur-couche de Tex qui n'a été créée 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 et LaTeX permet 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 enchainées à votre place, cela peut aller d'un double clic jusqu'à des processus bien plus compliqués. Ici nous ne les regarderons pas, les macros sont intégrées à LaTeX pour nous simplifier la vie, nous n'allons pas nous embêter à la compliquer. 😊

La dernière évolution majeure est LaTeX2e, c'est celle sur laquelle nous allons ~~souffrir~~ 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 fasse rire) et 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 entraîne la présence de nombreux logiciels permettant la création de documents, qui laisse parfois le néophyte perplexe. J'en ai choisi quelques uns pour vous, et vous guiderai dans leur installation selon que vous soyez sur Mac, Windows, ou Linux, tout cela se passe dans le prochain chapitre.

Dans ce chapitre, nous avons pu développer les origines, les avantages de LaTeX ainsi que son historique. Les prochains chapitres nous ferons rentrer dans le vif du sujet et vous permettrons de vous familiariser avec l'outil.

Nous garderons en tête quelques qualificatifs de LaTeX dans la suite du cours :

- rétro-compatible
- libre et gratuit
- création facile de pdf
- langage très populaire chez les scientifiques
- langage répandu dans le monde de l'édition.

Rendez vous dans le prochain chapitre pour procéder à l'installation de LaTeX sur votre ordinateur.

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.

Il est temps de rentrer dans le vif du sujet et d'installer la bête (il paraît que vous n'attendez que ça). 😊

Ce second 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 craintes, 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 sur Windows via une interface quelconque pourra être lu et modifié avec d'autres logiciels et environnements (par exemple sur des logiciels pour Mac ou Linux).

Cette profusion de logiciels est un des atouts du monde des logiciels libres. Elle permet à tout un chacun de trouver des outils taillés sur mesure pour ses besoins.

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 nous y arriverons) 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 .doc générés par Word ne savent pas faire. Il a beaucoup d'avantages et est très utilisé au sein des laboratoires de recherches. Vous n'aurez normalement pas grand besoin d'utiliser des fichiers sous format PostScript pour vos publications néanmoins certains chercheurs et éditeurs en sont des grands fans et ne jurent que par lui. Nous apprendrons donc à les manipuler.

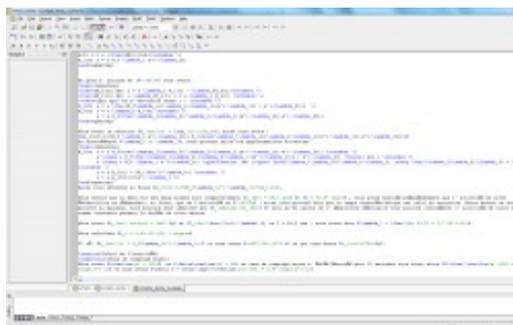
Une installation type de Latex comporte 3 éléments.

- **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'interface graphique 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.
- **Une interface graphique**: c'est un logiciel vous permettant de taper votre texte, de mettre en couleur les différents codes utilisés, de numéroter vos lignes... En somme, choisir une interface graphique vous simplifie la vie, mais **elle n'est pas indispensable** : il est possible de taper du Latex en ligne de code blanche sur fond noir (mais c'est un peu triste). 😊

Comme dit ci-dessus, vous pouvez au choix ne pas utiliser d'interface graphique et utiliser à la place un éditeur de type Bloc-notes ou des outils plus évolués de colorisation de texte. Les interfaces graphiques colorent le code (ce qui le rend beaucoup plus lisible) et vous permettent via des raccourcis et des fenêtres vous informant des bogues, de faciliter la production de document.

De la gauche vers la droite, vous pouvez voir ci-dessous le même code tapé : dans un éditeur de texte simple (Bloc-notes), dans un éditeur de texte avec colorisation (notepad++) et pour finir dans une interface graphique (TeXnicCenter).





Dans la suite du cours, nous utiliserons une interface graphique.

Installation sous 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**, c'est une des distributions Latex disponibles sur Windows. Pour l'installer la procédure est plutôt simple, mais elle demande d'être minutieux.

- Créez un dossier appelé "**Latex**" sur votre bureau ou dans votre dossier "Mes documents", il servira à ranger la masse de fichier que MiKTeX utilise pour son fonctionnement interne.
- Allez sur internet, à l'adresse <http://miktex.org/> et dans la colonne de gauche, vous trouverez un titre « **Download** » avec en dessous un lien vers la page de téléchargement de la dernière version de MiKTeX, **cliquez dessus**.

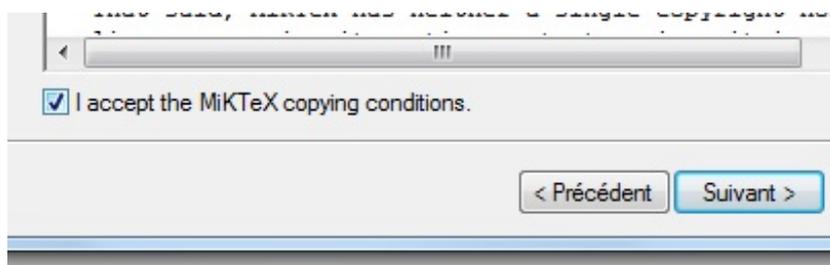


- Dans la nouvelle page, vers le bas de la page, vous trouverez un paragraphe intitulé « Installing the complete MiKTeX system » comportant un bouton **download**, vous devez cliquer dessus et enregistrer le fichier dans votre dossier « Latex ». Télécharger ce fichier devrait être rapide, il pèse environ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

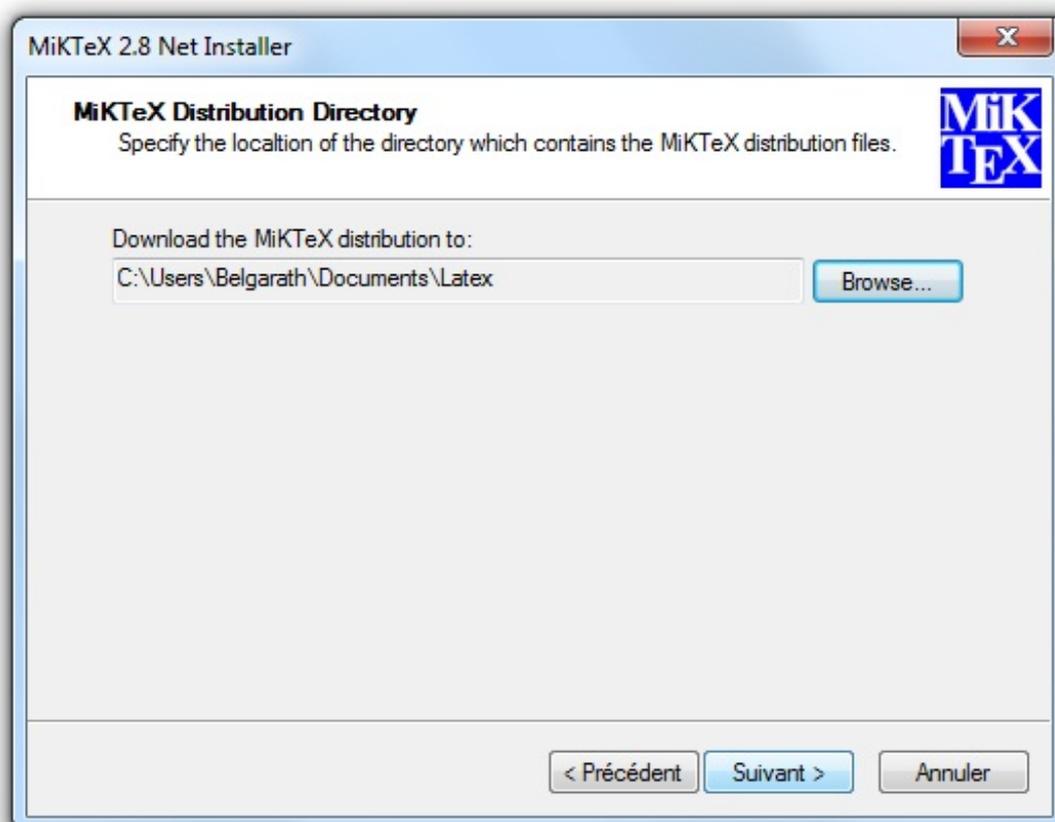


Le fichier que vous avez téléchargé est ce qu'on appelle un installateur, dans notre cas il va dans un premier temps télécharger MiKTeX et dans un second temps il l'installera.

- **Lancez l'installateur** (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**.



- **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 votre (par exemple j'en ai choisi un français au hasard). Plus le serveur de téléchargement choisi est proche, plus vous chargerez rapidement. Ensuite cliquez sur **suivant**.
- **ÉTAPE IMPORTANTE!** MiKTeX vous demande ici dans quel dossier il enregistrera tout son attirail, c'est ici que le dossier « Latex » créé préalablement 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 ci-dessous.



- Cliquez sur **Suivant**, vous aurez droit à un petit écran récapitulatif résumant 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.



MiKTeX doit télécharger environ 250 Mo de fichiers, et cela peut prendre pas mal de temps. Profitez donc de la pause pour tuer des zombies sur 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 où à un autre l'installation de MiKTeX plante, il vous faudra alors tout recommencer depuis le début (ça m'est arrivé pendant l'écriture du tutoriel). C'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 l'installateur**. 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 allez cliquer sur **suivant**.
- Sur le nouvel écran, il vous demande si vous souhaitez installer la version basique ou complète, vu que nous avons téléchargé la 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 ouverts 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 téléchargé plus tôt. **Vérifiez** bien que le bon dossier est sélectionné avant de cliquer sur **Suivant**. L'écran suivant 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 le changer ou le laisser avant de cliquer sur Suivant et d'atterrir sur un écran vous demandant votre format de papier par défaut (ici A4 vu que nos imprimantes impriment surtout en A4, mais vous pouvez le changer si vous avez des besoins très spécifiques) et vous proposant de vous demander gentiment la 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 félicitation. 



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 un 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, ne paniquez pas, LaTeX étant libre, il y a des multitudes de logiciels pour le pratiquer.

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 : [Télécharger le logiciel](#).

Lancez l'installateur, 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 le trouver à l'adresse suivante : [Télécharger le logiciel](#).

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. 

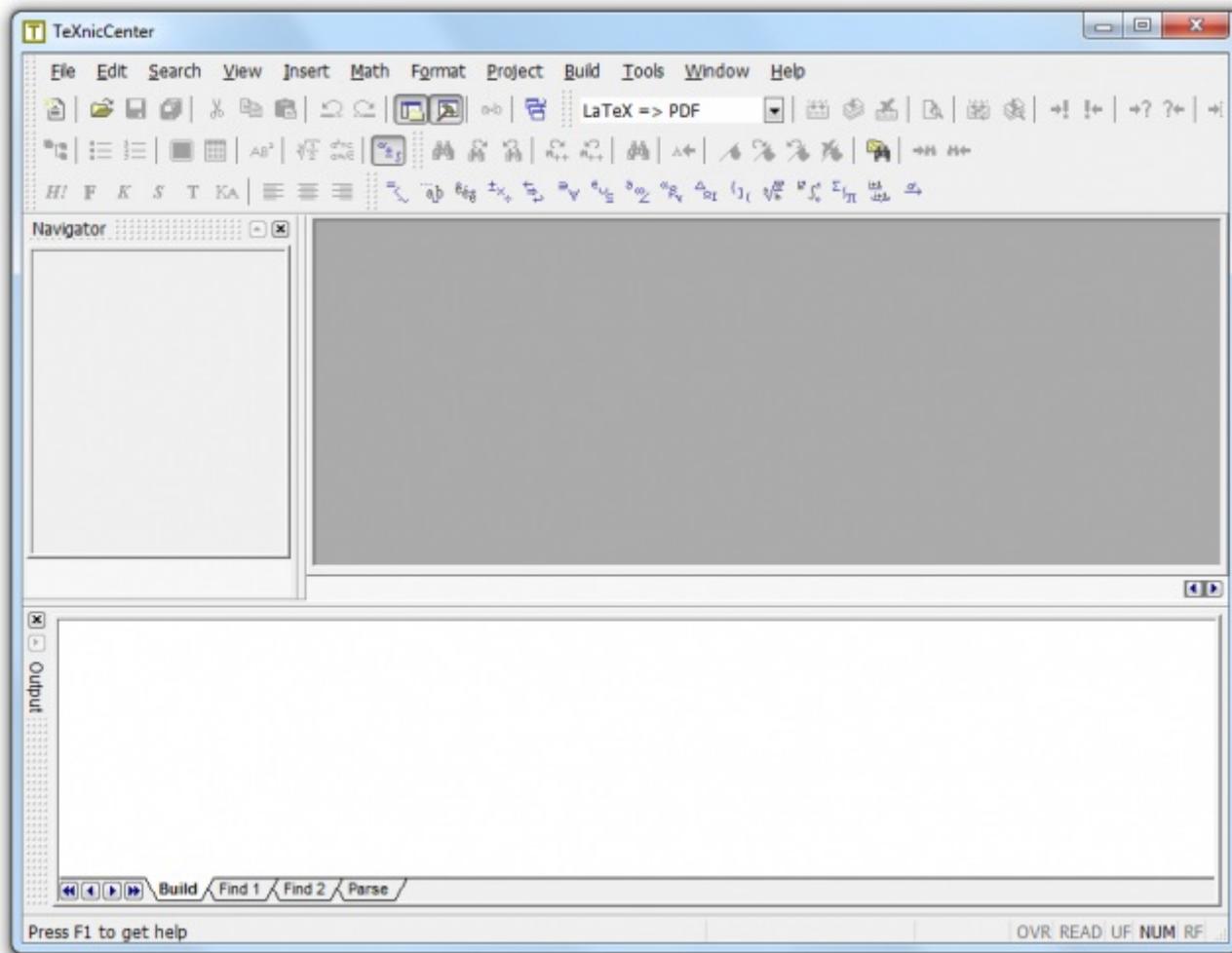
TeXnicCenter, L'interface graphique

TeXnicCenter sera votre interface graphique, vous avez pu voir une de ses captures d'écrans dans la paragraphe traitant des logiciels nécessaires. Pour l'installer, il vous faudra tout d'abord aller la chercher à l'adresse suivante :

<http://www.texniccenter.org/resources/downloads/29>. C'est le premier lien, nommé "TeXnicCenter Installer" qui nous intéresse, **cliquez** dessus, il vous permettra de lancer le téléchargement de l'installateur.

Une fois n'est pas coutume, lancez l'installateur.

- A 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 si 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 si besoin puis cliquez sur **Next**.
- Il vous propose de créer une icône sur le bureau, modifiez si besoin puis cliquez sur **Next**.
- Petite fiche résumé, vous pouvez cliquer sur **Install** après avoir vérifié les informations qu'elle contient. S'en suis le petit écran de félicitation et la possibilité de cliquer sur le bouton **Finish**. 😊



L'installation est achevée, vous avez en présence sur votre pc les trois composantes nécessaires à votre apprentissage de LaTeX. Installation fastidieuse mais néanmoins importante pour la suite, nous ferons nos premiers tests dans le chapitre suivant et configurerons TeXnicCenter. 😊

Installation sous Mac

Possesseurs de Mac, vous êtes des grands chanceux.

Oh que oui! 😊

C'est vous, possesseurs de Mac qui détenez la palme de l'installation LaTeX la plus simple et agréable. Les possesseurs de Windows viennent de souffrir environ trois heures pour installer toutes les composantes de LaTeX séparées en Trois bouts, alors que pour vous, tout est rassemblé dans un seul fichier!

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:

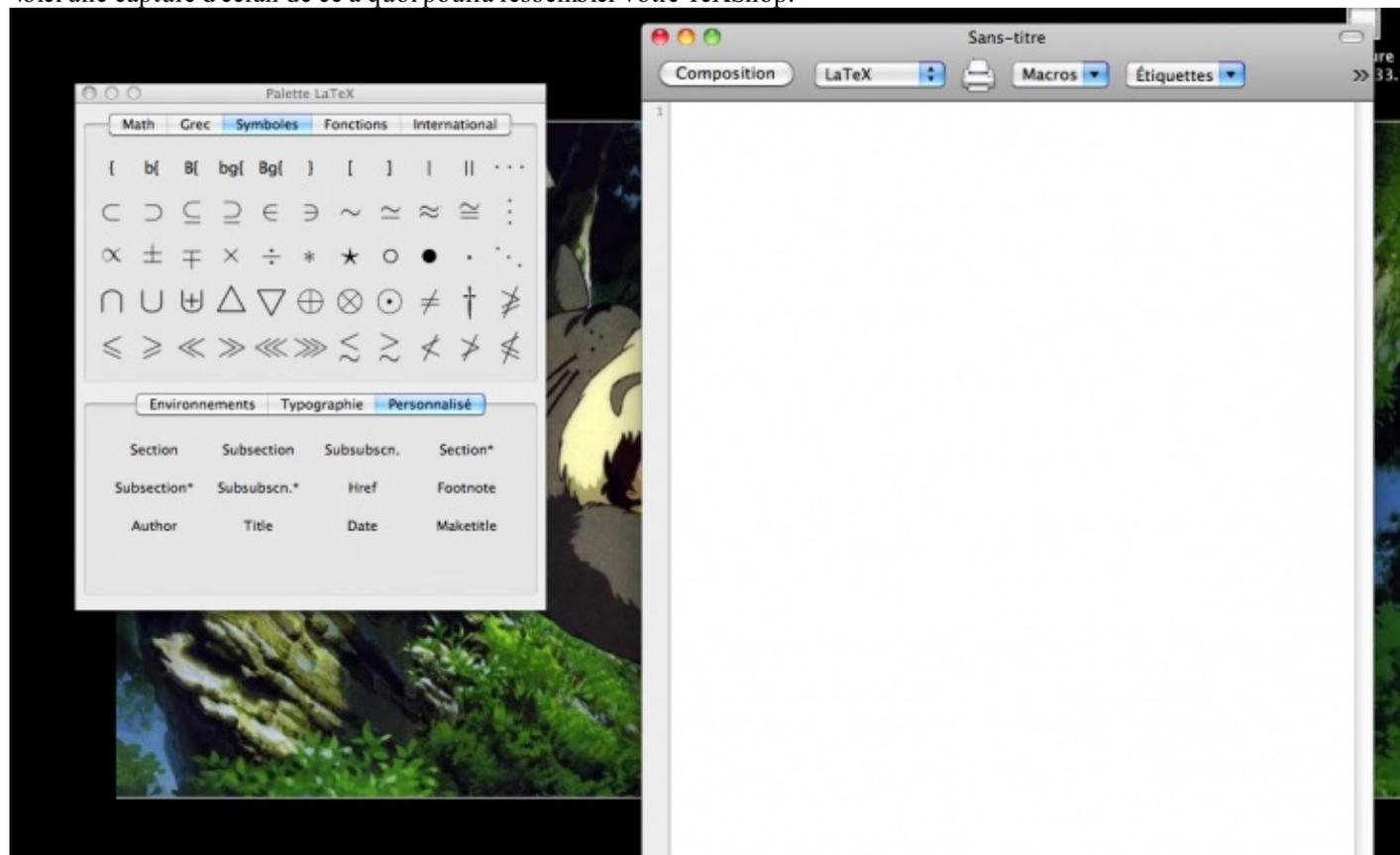
MacTeXtras.zip [312M - 18 May 2010]

Smaller Packages [for users with slow download speed]

Tout d'abord [allez sur ce site](#), c'est le site du projet MacTeX, rassemblant en son sein 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, téléchargez le sur votre Mac et lancez l'installateur avant de suivre la procédure simple qui se propose à vous.

Une fois fait, vous pourrez utiliser à loisir TeXShop, l'interface graphique pour Mac que nous utiliserons dans la suite du cours. Il devrait s'être installé dans votre dossier « applications », libre à vous d'en faire des raccourcis sur votre bureau.

Voici une capture d'écran de ce à quoi pourra ressembler votre TeXShop.



Lors de cette installation, 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 méritent que vous les testiez lorsque vous vous sentirez à l'aise avec LaTeX.

Installation sous Linux

Dans le cadre de ce cours, l'interface graphique choisie sera Kile et la distribution LaTeX se nomme TeXLive. Les méthodes d'installations 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 paquet (dans lequel vous téléchargerez dans un premier temps TeXlive puis Kile). Nous installerons dans le même temps de quoi à lire vos fichiers produits, de type pdf et postscript.



Beaucoup à leurs débuts sur linux ont eu du mal avec l'installation de LaTeX. N'hésitez pas à chercher dans votre amphi où dans vos groupes de travail des personnes prêtes à vous aider. Il sera parfois plus évident pour vous de travailler dans les mêmes environnements que vos collaborateurs, et le cours que vous lisez ici s'applique à n'importe quelle interface graphique, seul l'emplacement des boutons changera si vous utilisez une interface graphique différente.

Comme dit plus haut, nous allons utiliser le gestionnaire de paquet.

Dans un premier temps, **ouvrez-le**. Vous **allez rechercher et installer tous les paquets contenant les mots suivants** (via le champ de recherche) :

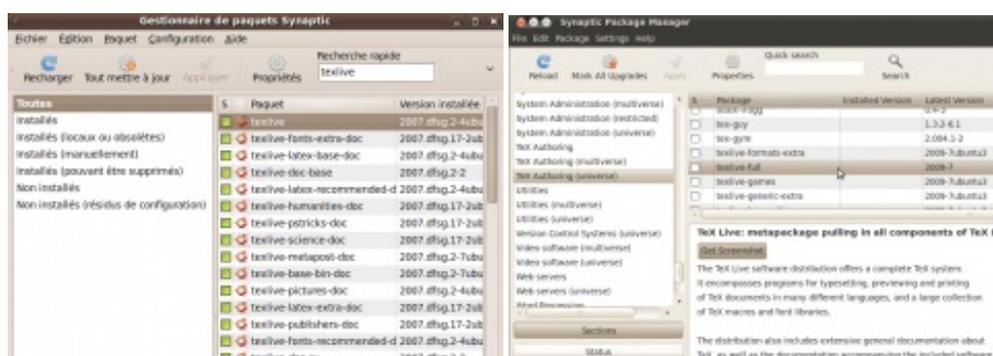
- texlive
- ec-fonts-mftraced (uniquement si vous êtes sous mandriva)
- cm-super (pack de polices)
- xdvi (pour utiliser les .dvi que nous utiliserons dans la suite du cours)
- gv (pour lire et manipuler les .ps)
- spdf
- kile (toujours en dernier).

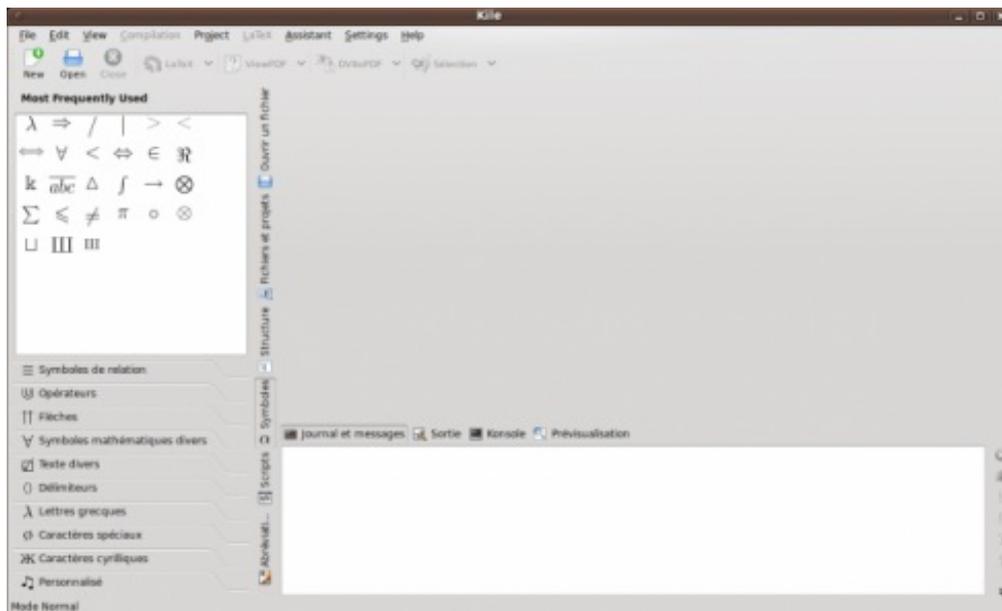
Une fois ces multiples chargements effectués vous pouvez vous reposer, et votre installation fonctionne.

Ici deux captures d'écran du gestionnaire de paquet et une troisième de Kile, l'interface graphique sur laquelle nous allons nous exercer et apprendre à utiliser LaTeX.



Kile est une interface graphique facilitant l'écriture et la compilation de fichiers LaTeX. Il est possible néanmoins de manipuler des fichiers LaTeX, tapés dans un éditeur de texte, via les lignes de commandes. 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).





Si soucis il y a, veuillez consultez les site internet de [TeXLive](#) ainsi que de [Kile](#) afin d'obtenir les informations spécifiques à votre installation Linux. Ce sont des outils très utiles et qui vous permettront d'adapter la procédure d'installation à votre distribution Linux si jamais elle est un tant soit peu exotique.

Vous voilà arrivé à la fin de l'installation de votre nouvel outil, nous allons apprendre dans le chapitre suivant à créer notre premier document. 😊

L'installation est la partie la plus embêtante avec LaTeX, en particulier avec Linux. Ayez conscience qu'après ce petit moment de souffrance, le reste sera plus simple, car vous aurez sur votre ordinateur un outil complet.

Structurer son premier document

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

Soyez sûrs d'une chose, le plus dur est derrière vous.

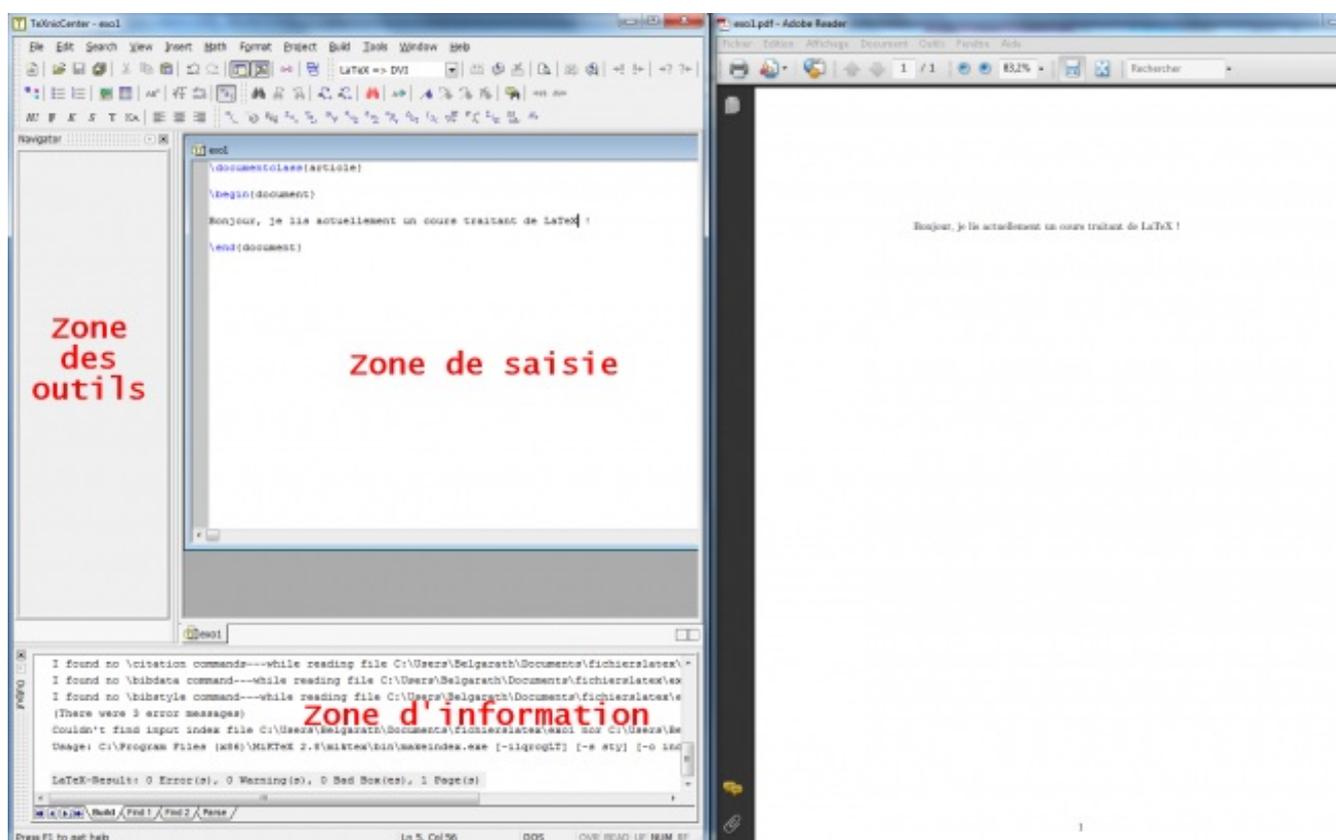
Ce chapitre vous permettra d'écrire votre premier document, de l'imprimer, et de comprendre comment LaTeX transforme un ingrat code informatique en un document chatoyant. 😊

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

La compilation

Cette sous-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 codes se transforment en une mise en page. Vous pouvez voir dans la capture d'écran ci-dessous un document écrit dans une interface graphique ainsi que son résultat une fois passé par la moulinette de votre distribution LaTeX.



Que s'est-il passé ?

Souvenons-nous du chapitre précédent. Nous y avons appris que les interfaces graphiques ne sont pas indispensables. Elles ne jouent qu'un rôle esthétique. En revanche, nous avons appuyé fortement sur le caractère incontournable de l'installation d'une distribution LaTeX. Les distributions sont les pièces centrales, elles permettent de transformer un fichier de ligne 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 utiliser par la suite pour vos travaux et vos publications.

Résumons tout cela.

- Le code est rédigé dans l'interface graphique.
- La distribution LaTeX le compile.

- Le fichier .tex de départ se voit transformé en un fichier .ps, .pdf ou .dvi (un format très peu utilisé).

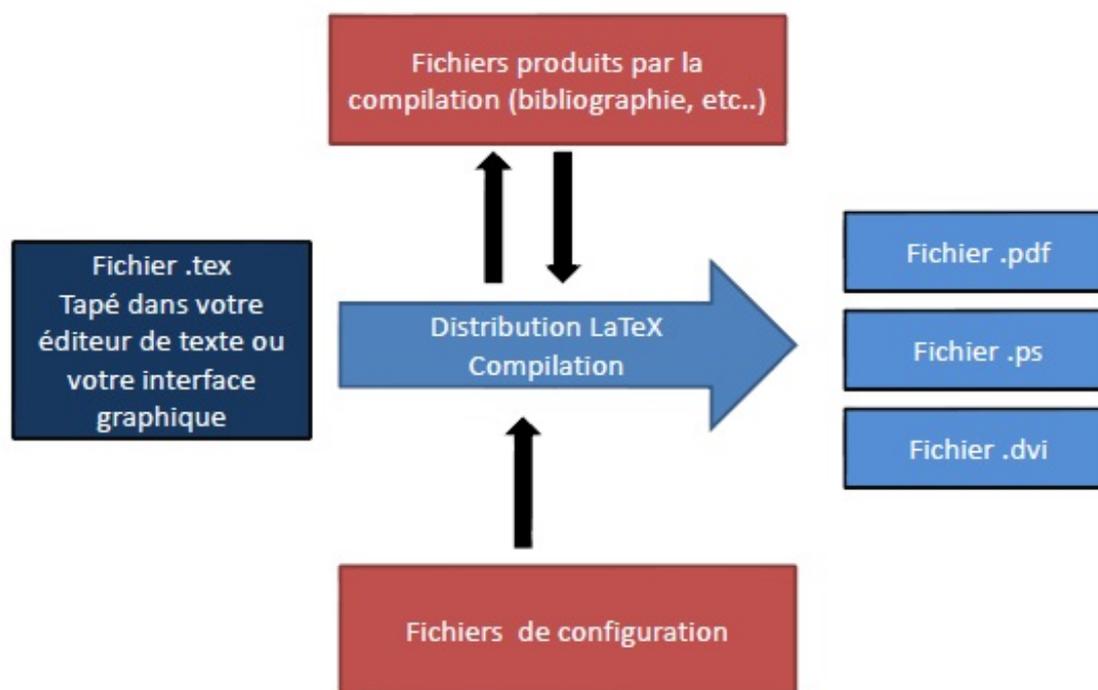
Le processus de création n'est pas plus compliqué que cela. Pour écrire un document il vous suffira d'écrire votre code au sein de l'interface graphique et votre distribution LaTeX se chargera du travail de transformation. Pour que le travail d'écriture du code et de compilation soit plus simple, les interfaces graphiques sont dotées d'une zone réservée aux raccourcis (type caractère spéciaux) qui vous permettront d'insérer facilement dans vos écrits des bouts de codes dont vous vous servez 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 bogues 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. Votre fichier .tex de départ créera donc après compilation une multitude de fichiers correspondants à ces besoins. Ne prenez pas peur si votre dossier ne contenant au départ qu'un .tex se voit tout d'un coup envahi par de multiples fichiers aux extensions étranges.

Nous allons résumer toutes ces informations via un schéma. Je vous conseille vivement de l'imprimer, il est souvent pratique de le garder 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 écrire des textes et les mettre en forme dans le formalisme LaTeX.

Rédaction d'un document simple

Dans cette sous-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 interface graphique (Kile, TeXShop, TeXnicCenter ou autre). Il est possible que celle-ci vous pose quelques questions simples (pour connaître l'emplacement où vous avez installé 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 avec une partie console où apparaissent les messages d'information (ainsi que les messages d'erreurs), une zone réservée aux barres d'outils, et une zone réservée à la rédaction de votre fichier.

Grâce aux raccourcis disponibles dans la barre de menu, ouvrez un nouveau document. Cliquez à l'intérieur de la zone réservée à l'édition et recopiez 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 un dossier vide de votre choix grâce à la barre de raccourcis présente en haut de votre interface. Nous l'appellerons bonjour.tex.

Il est interdit de mettre des espaces ou des accents dans vos noms de fichiers. Votre compilateur n'apprécie pas l'originalité et aura tendance à vous renvoyer des erreurs.

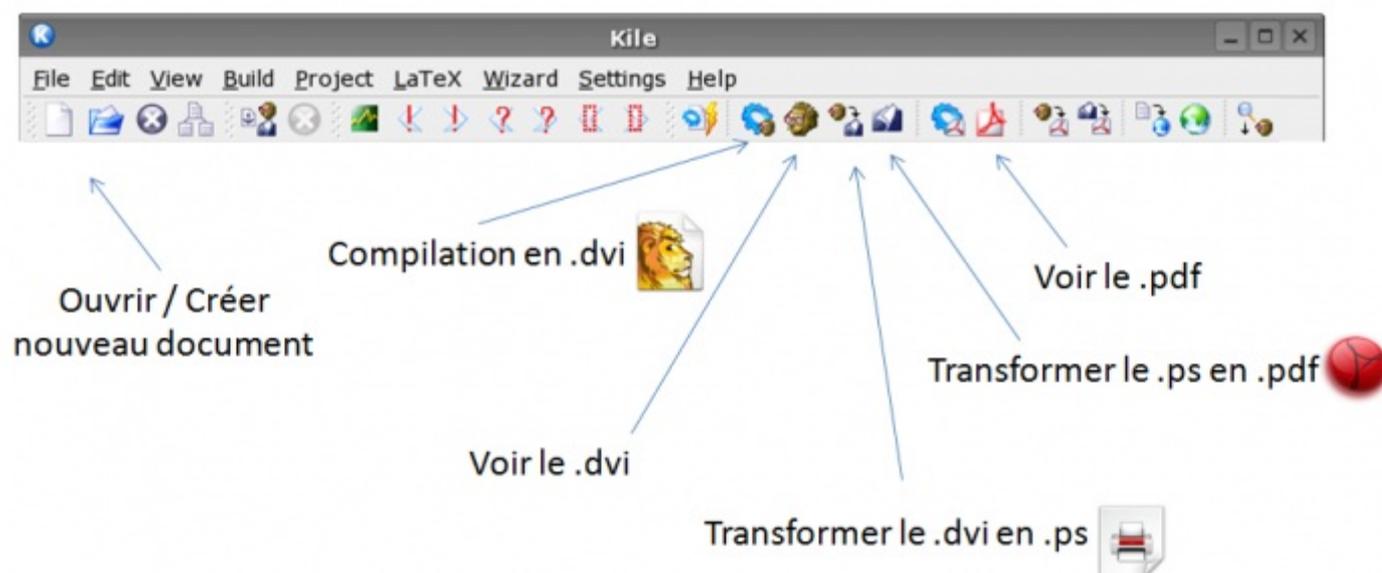


Autre subtilité, le compilateur compile du texte brut et non pas du texte enrichi comme peut en fournir Word (un texte enrichi est un texte auquel nous avons ajouté une mise en page). Si jamais l'envie vous prends de copier-coller un texte depuis word, faites d'abord un copier-coller dans le bloc note ou un logiciel du même genre. Vous vous éviterez ainsi beaucoup de problèmes.

Compilation

Nous allons maintenant compiler votre fichier .tex en un autre format (postscript, pdf ou dvi). Ici trois cas de figures se présentent pour vous suivant que vous soyez sur windows, sous mac ou encore linux.

Compiler avec Kile (Linux)



Vous pouvez voir sur la capture d'écran ci-dessus 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 fichier pdf selon vos besoins.

Pas de manipulation compliquées ici, votre interface graphique se charge de tout et vos fichiers produits se trouvent dans le dossier où vous avez enregistré votre fichier .tex.

Compiler en ligne de commande

Nous avons vu dans le chapitre précédent que l'interface graphique n'est 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 suivantes.

Dans un premier temps, vous pouvez compiler votre fichier `bonjour.tex` en un fichier en `bonjour.dvi`. Le fichier créé apparaît 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 décrite `dvips` (en y rajoutant `-o` comme vous pourrez le voir dans l'exemple). La commande `ps2pdf` vous permettra de transformer votre fichier au format postscript en 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 le long du cours.

Code : Console

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

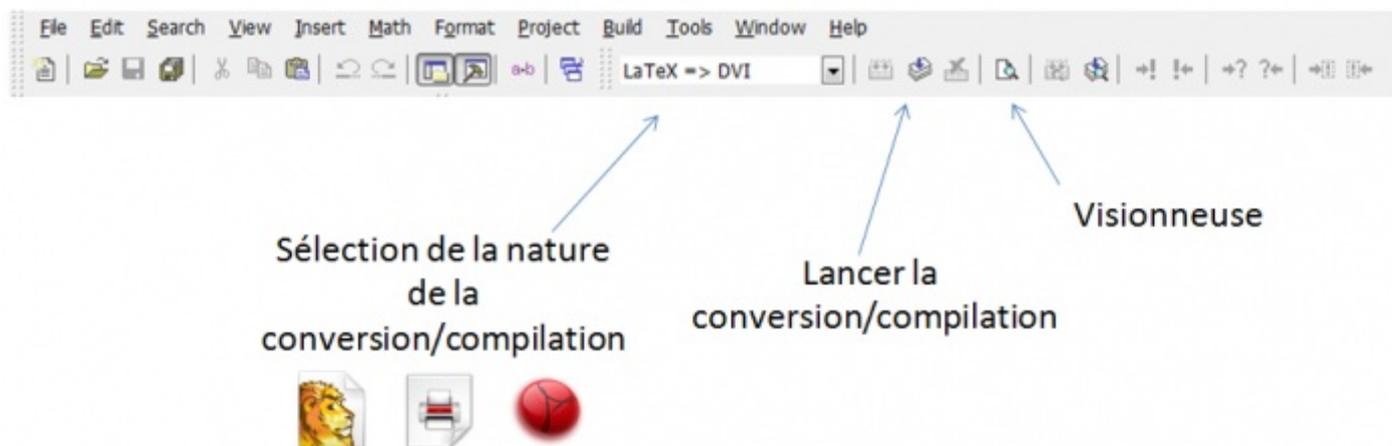
Compiler avec TeXShop (Mac)

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

Votre interface graphique se charge de tout et les conversions sont faites de façon automatique.

Sans plus attendre vous pouvez admirer le résultat de votre compilation en allant fouiller dans le dossier dans lequel vous avez enregistré votre fichier .tex.

Compiler avec TeXnicCenter (Windows)



Sur votre barre de menu vous constatez la présence d'un menu déroulant. Ce menu vous permet de choisir entre la compilation d'un fichier .tex en un .dvi ou la conversion de ce fichier .dvi en un autre format de fichier.

Le bouton à sa droite vous sert à lancer les compilations/conversions et le boutons encore plus à droite vous donne des 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 vous avez réussi votre compilation et vous venez de découvrir comment écrire un document en LaTeX (libre à vous de l'imprimer) et vos logiciels fonctionnent. Autre cas de figure, votre interface graphique vous a renvoyé une erreur et je suis dans le regret de vous annoncer que vous avez fait une erreur lors de l'installation et que vous devez la recommencer.



Avant de vous résigner si jamais la compilation n'a pas fonctionné, essayez de recopier une nouvelle fois le code source (pas de copier-coller). Il serait dommage de recommencer l'installation pour un \ oublié.

Type de document et caractères spéciaux

Vous venez de créer votre premier document, félicitations ! 😊

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'interface graphique et allons 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 10 et ce sont les suivants : \$ & % # _ { } ~ ^ \.

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 avec ces signes, il suffit de faire précéder les signes par un backslash (le fameux « \ » s'appelle un backslash). Pour le backslash lui-même, nous utiliserons une commande.

Table des caractères réservés	
\$	\\$

&	\&
%	\%
#	\#
_	_
{	\{
}	\}
~	\~
^	\^{}
\	\textbackslash{}

Testons ce tableau ! Entrez le code suivant dans votre interface graphique, 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}
```

Le résultat obtenu doit ressembler à la capture ci-dessous.

Bonjour, je lis actuellement un cours traitant de LaTeX !
\$ & % # - ^ \ { }

Ce tableau est le premier de ce cours, mais il y en aura d'autres, par exemple pour les accents. Lorsque le cours sera plus avancé, je synthétiserai pour vous tous les tableaux au sein d'une annexe volumineuse que vous pourrez imprimer et garder sous la main.

Types de documents et portions de texte

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

Portions de texte

Nous avons entouré dans nos deux premiers exercices notre texte par 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 commençons un document par `\begin{document}`, que nous écrivons du texte avant de fermer 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 refermer par `\end{recette}`. Ce qui pourrait nous donner le code ci-dessous (ici ce n'est qu'un exemple fantaisiste).

Code : TeX

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

`\begin` ouvre un environnement et `\end` le ferme. Le type d'environnement se voit encadré par des accolades et les options qui lui sont rattaché sont entre crochets (nous en utiliserons dans la suite du cours).

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 les utilités des différents environnements au fil du cours.

Les types de documents

La première ligne de nos 2 exercices n'a pas encore été analysée, et elle contient une information capitale. Tout d'abord, ré-affichons cette ligne, je vous explique tout ensuite.

Code : TeX

```
\documentclass{article}
```

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

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 aux types de documents. La formulation des options se fait selon la syntaxe suivante.

Code : TeX

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

Vous venez de réaliser votre premier document avec LaTeX et de commencer à toucher le code. Résumons ici ce que nous avons appris.

- La compilation d'un document se déclenche grâce à des raccourcis présents dans l'interface graphique.
- Certains caractères doivent être précédés d'un backslash pour être insérés dans un texte.
- Le contenu d'un document est encadré par une commande spécifiant le type du document ainsi que des commandes ouvrant et fermant l'environnement document.

Prochain chapitre : les packages...

Les packages

Votre apprentissage des mécanismes élémentaires de LaTeX est terminé. Vous avez appris à ouvrir et rédiger des fichiers sources.tex et à les compiler de façon à obtenir des documents aux formats pdf, postscript et dvi.

Nous avons pu aussi 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 pratiques seront étudiées dans les parties 2 et 3.

Ce chapitre est le dernier chapitre théorique. Nous allons ici répondre à la question suivante : si LaTeX ne sait pas faire quelque chose (par exemple mettre du texte en couleur) comment rendre possible un changement de couleur de mon texte ?

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

Ce court chapitre est le dernier de la partie 1. La suite du cours sera essentiellement pratique.

La philosophie des packages

Les habitués du monde informatique connaissent assez bien la signification des mots « extensions », « packages » ou « plugins ». Ces éléments sont présents dans de nombreux programmes et nous allons tenter de développer le concept de « package » pour ceux à qui il n'est pas familier dans cette sous-partie.

Imaginons votre installation LaTeX comme une cuisine équipée. Vous pouvez cuire des choses, 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 (comme nous avons pu le voir dans le chapitre précédent).

Vous vous préparez un plat et avez besoin d'un couteau pour le cuisiner, et ce couteau ne fait pas parti 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'amener dans votre cuisine pour vous permettre d'atteindre votre but de départ : 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, de telle sorte que, grâce à leur création, l'univers LaTeX permet de réaliser chaque jour de nouvelles choses. Ces outils sont appelés « packages » et tout un chacun peut en créer et en diffuser un.

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

Pour vous en tant qu'utilisateur de LaTeX, vous serez amené fréquemment à utiliser des packages. En résumé : deux possibilités s'offrent lorsque que 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 (comme si votre couteau était dans un des tiroirs de votre cuisine)
- il est absent de votre installation LaTeX et vous devrez aller le chercher et l'installer.

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

Comment s'en servir?

Bien que nous allons ici apprendre à installer des packages, ce n'est pas le point central de ce chapitre. Des millions d'utilisateurs avant vous ont conçu des documents sous 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 un bureau sans table, les utilisateurs de LaTeX préfèrent installer dès le début les éléments leur permettant de travailler confortablement).

Cette sous-partie va vous apprendre à utiliser les packages.

Pour utiliser un package sous LaTeX nous procéderons en deux étapes. Dans un premier temps nous dirons à LaTeX, via une commande, que nous utiliserons tel ou tel package, dans un second temps nous pourrons les utiliser dans nos publications.

La commande permettant d'appeler un package est la commande `\usepackage[option]{type}`. Elle se place juste après la ligne dédiée à la commande `\documentclass`. Regardez l'exemple ci-dessous, que nous commenterons 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 3\up{ième} document avec \og 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 `% suivi d'un 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 note un fichier contenant la phrase «j'aime latex» avant de l'enregistrer sous le nom amour.txt et de taper dans votre console la ligne suivante : `file amour.txt`. Si la console inscrit Iso-8859 tout va bien. Si elle inscrit utf-8 vous devrez taper `\usepackage[utf8]{inputenc}` à la place de `\usepackage[latin1]{inputenc}` dans vos prochains documents.



Utilisateur sous mac : il est important que vous enregistriez vos documents en latin 1. TeXShop vous proposera dans le menu Préférences>Document>Encodage de choisir l'encodage par défaut. Choisissez le suivant : Occidental Latin 1.

Utf-8 et iso-8859 désignent des processus d'encodage de caractère, c'est-à-dire la manière dont votre machine va coder les lettres. Bien que nous ne puissions pas voir la différence à l'œil, se tromper entre utf8 et latin 1 peut tout faire planter.



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

Effectivement, ces commandes sont nouvelles. Le package babel (lorsqu'il est utilisé pour le français) permet l'utilisation de commandes propres à la langue choisie. Entre autres les guillemets français et les petites majuscules 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 permettant de voir les commandes introduites par le package babel. Essayez les et entraînez vous 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°	<code>\secundo</code>
3°	<code>\tertio</code>
4°	<code>\quarto</code>
8° (ou n'importe quel autre chiffre)	<code>\FrenchEnumerate{8}</code>
n°	<code>\no</code>
N°	<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 et chacun rend LaTeX plus performant et vous permet d'ajouter, à la carte, des fonctions à votre distribution. Souvent, lorsque vous aurez envie de faire quelque chose d'original avec LaTeX, quelqu'un aura déjà fabriqué un package adapté et il ne vous restera plus qu'à l'installer ainsi qu'à l'utiliser.

Comment installer un package ?

Votre installation faite, de nombreux packages ont été installés sur votre machine. Il est cependant possible que vous ayez un jour besoin d'installer un package particulier non-présent sur votre machine. Cette sous-partie est là pour vous expliquer comment faire.

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



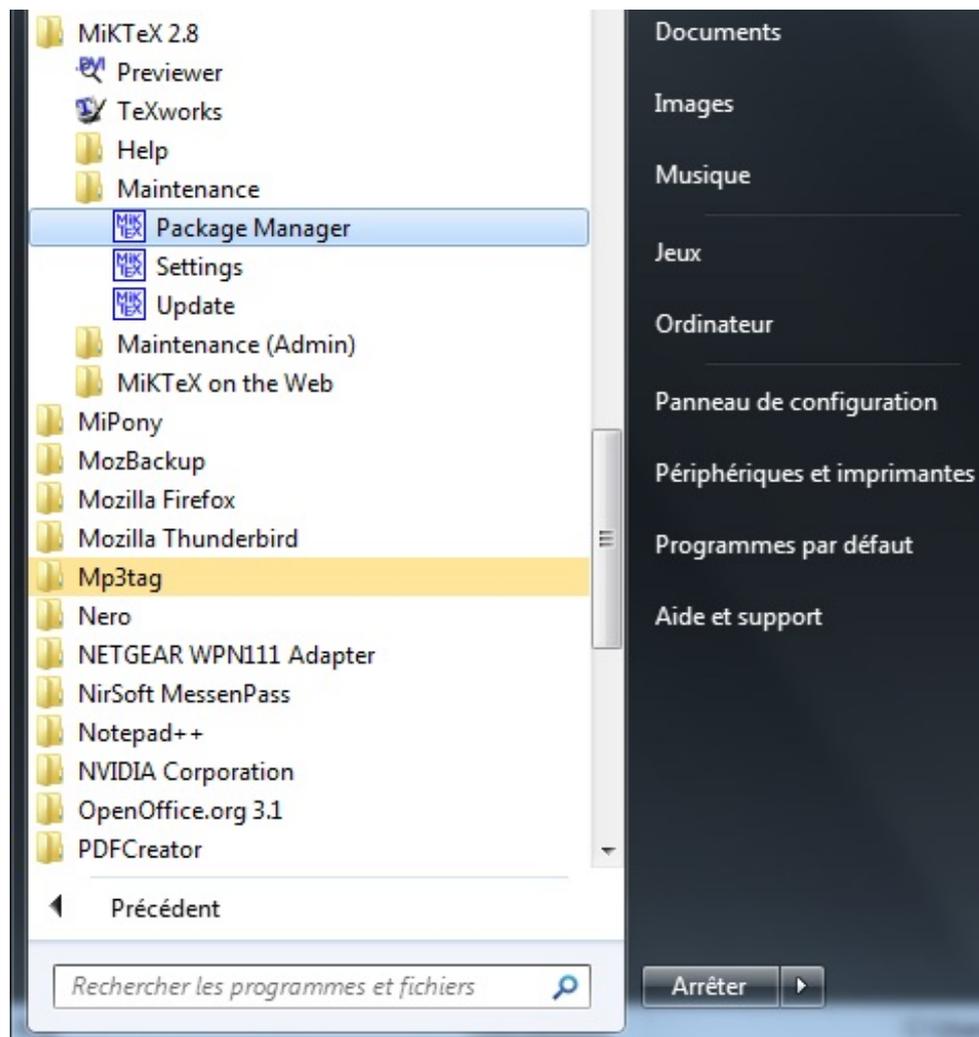
Cette sous-partie n'étant utile que si vous tentez d'installer un package non présent dans votre distribution, vous pouvez ne pas la lire et passer au chapitre suivant sans crainte si vous n'en avez pas besoin.

MiKTeX et son installation à la volée

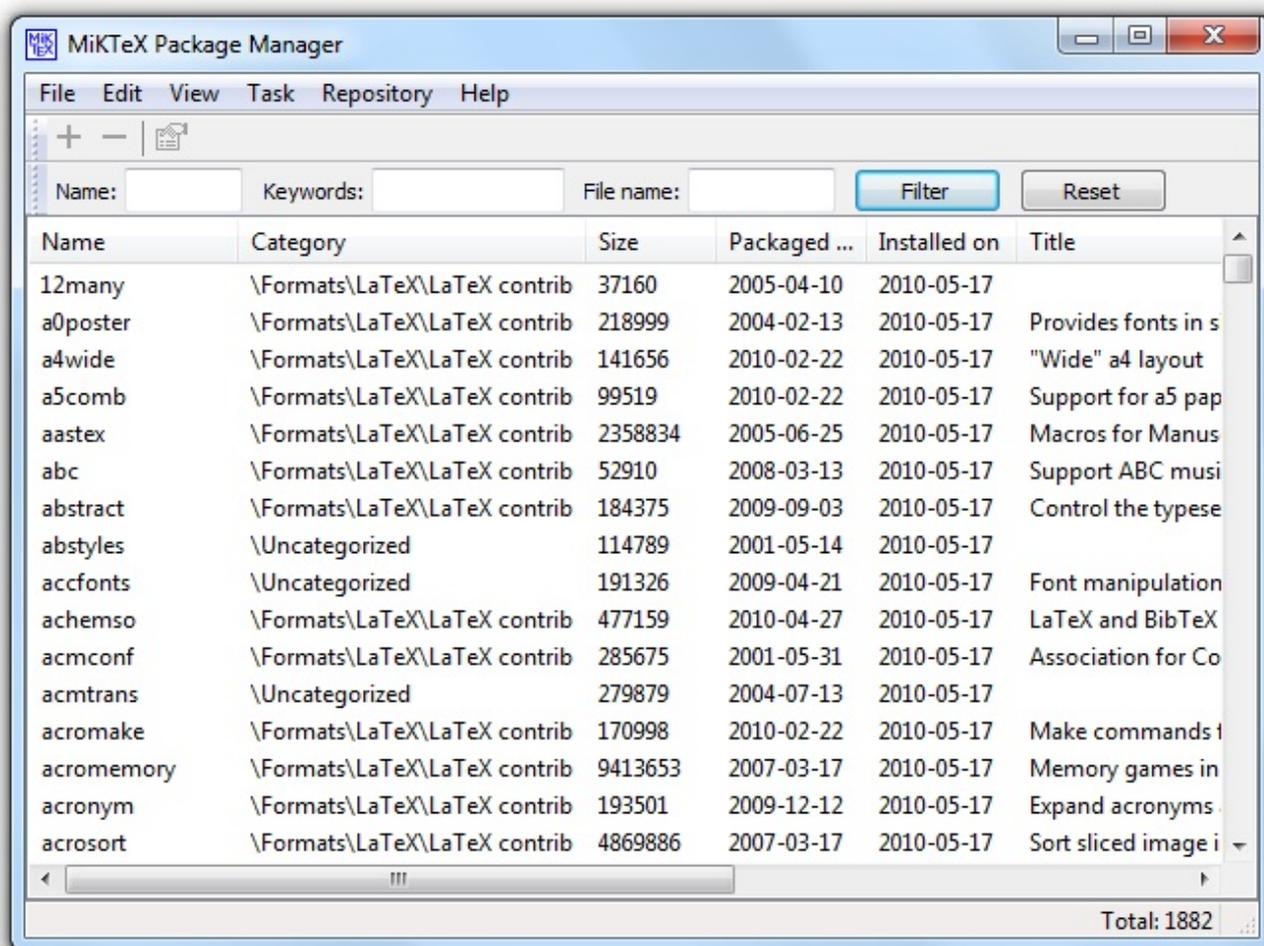
Les utilisateurs de MiKTeX (Windows) ont de la chance en ce qui concerne les packages, car MiKTeX installe tout seul comme un grand les packages appelés lors de vos compilations.

Si jamais 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.

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



Le package manager vous permettra de voir la liste des packages installés dans votre distribution.



Autres distributions

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

Deux méthodes sont disponibles suivant que votre package soit sous la forme d'un fichier .dtx ou .sty.

Dans de rares cas, les packages sont fournis sous d'autres formes, mais ils sont alors accompagnés d'un fichier Readme vous guidant dans 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 votre fichier .tex elle recherchera les fichiers .sty des packages manquant dans votre dossier et le tour sera joué.

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

Pas très compliqué, n'est-ce pas ?

Les packages en .ins, méthode en deux temps

Les packages contenus dans un fichiers .ins doivent subir 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 ».



Les méthodes développées ici vous permettent d'installer et d'utiliser pour un document précis des packages. Il existe des méthodes permettant d'installer pour de bon sur votre machine des packages. Ces méthodes bien qu'efficaces ne sont pas évidentes pour les utilisateurs débutants, raison pour laquelle j'ai préféré vous présenter des méthodes plus faciles.

Vous avez ici eu votre premier contact avec les packages. Ils sont omniprésents dans l'univers LaTeX et très utiles, nous en utiliserons quelques-uns dans la suite du cours.

Nous garderons en mémoire que :

- il existe des milliers de packages LaTeX
- installer un package ne prend pas plus de dix minutes
- la communauté des utilisateurs (dont vous faites maintenant partie) peut ajouter autant de fonction qu'elle le souhaite à LaTeX via la création de package.

Ce chapitre clos la partie 1, partie très théorique qui vous a permis de vous familiariser avec le langage LaTeX et ses logiciels. Les chapitres qui viennent vous permettront de pratiquer et d'apprendre à mettre en page des lettres, des rapports, d'écrire des formules mathématiques...

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.

Maitriser sa mise en page (1/2)

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

Ce premier chapitre vous permettra d'apprendre entre autres à :

- structurer vos documents
- justifier et espacer vos paragraphes
- 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 formes de parties, chapitres, sections, etc... Votre distribution LaTeX se servira de la hiérarchisation de vos données pour mettre en page de façon propre vos informations.

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 (non disponible pour les classes de document article et letter)	<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>

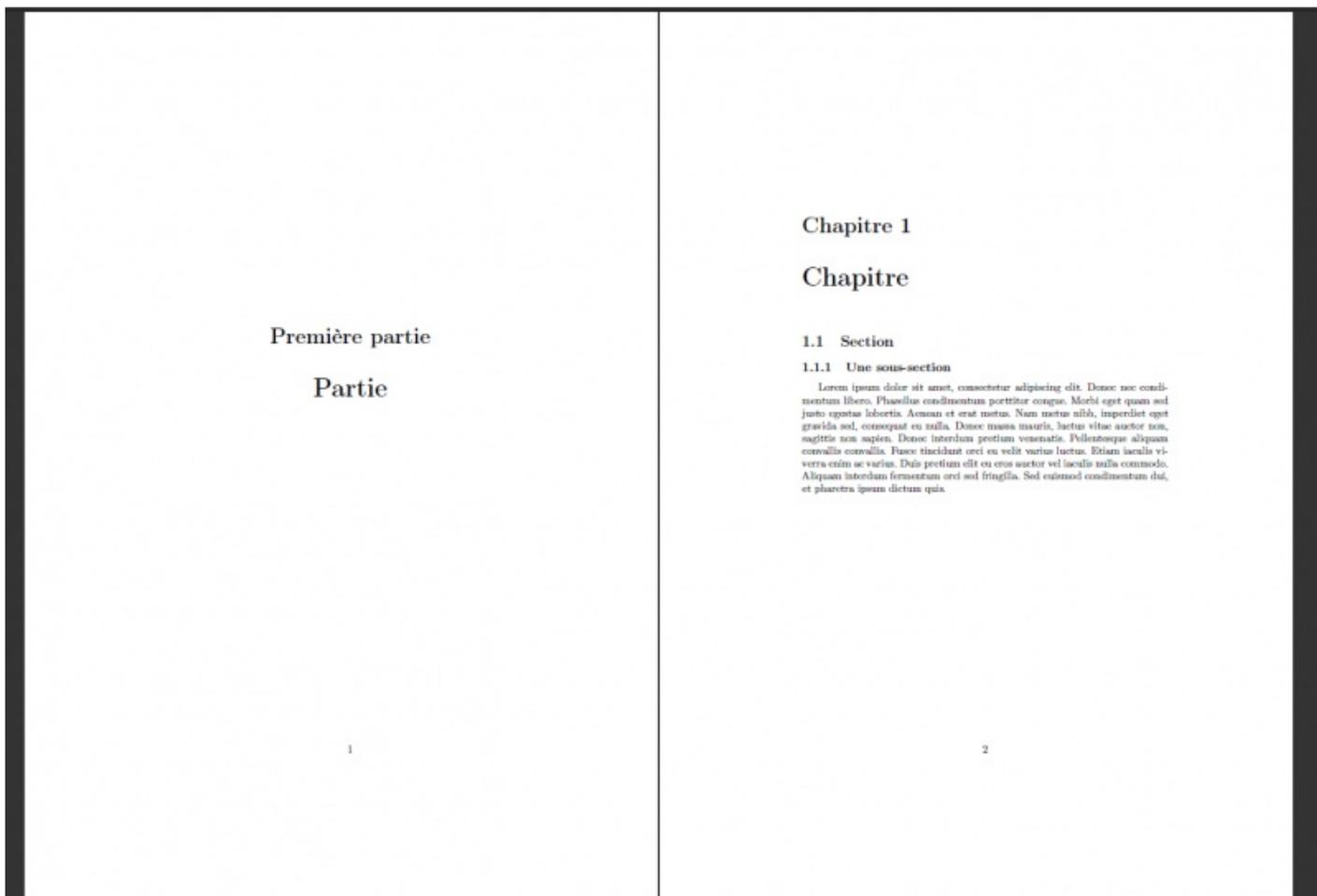
Exercice

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, contenant : une partie, un chapitre, une section et une sous-section dans laquelle vous écrirez quelques lignes.

Votre résultat doit ressembler à la capture d'écran ci-dessous.



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 tout comme la présence de quelques coquilles dans votre code. Néanmoins, votre code final doit être proche de celui-ci (même si vous pouvez avoir 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 egestas 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}
```



Le texte latin utilisé ci-dessus est un « faux texte » généré par le site internet lipsum.com. 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 métiers du graphisme et de la mise en page.

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. 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` et elle se place entre le contenu et les annexes, comme vous pouvez le constater dans le code source ci-dessous.

Code : TeX

```
\documentclass{report}

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

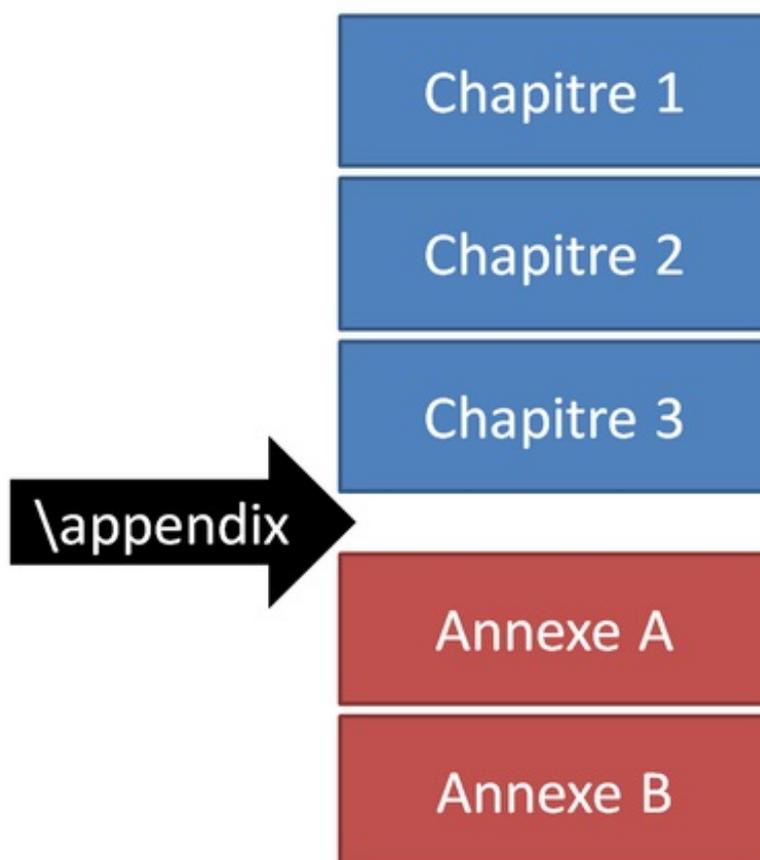
\begin{document}

\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 grace à appendix
\chapter{Un article annexe}
\chapter{Un canard}

\end{document}
```

Résumons ceci par un schéma :



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 numéros/lettres 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 numéros, idem 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 :

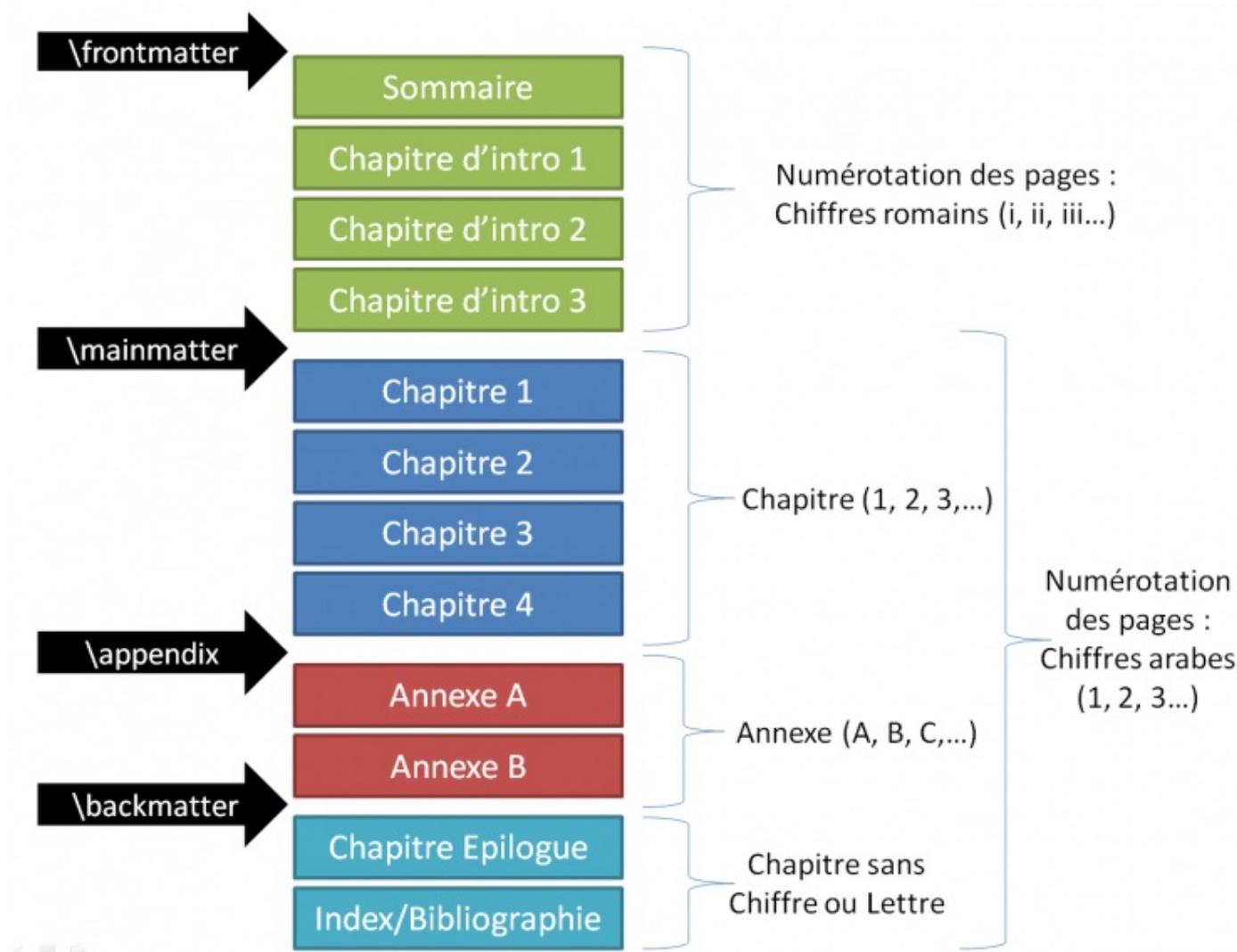
- les préambules et avant-propos
- les chapitres de contenu
- les annexes
- les pages d'index et de bibliographie.

Ainsi de nouvelles commandes apparaissent, fonctionnant de la même manière que la commande `\appendix`, mais n'agissant pas uniquement sur la numérotation des chapitres/sections mais également sur la numérotation des pages. Tout d'abord `\frontmatter`, qui se positionne juste après `\begin{document}` et 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 d'épilogue et les index/bibliographies (que nous apprendrons à créer dans la suite du cours), et a pour effet de stopper la numérotation des chapitres, mais pas la numérotation des pages.

Résumons ceci par un nouveau schéma :



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 mains pour en créer une. Vous allez donc subir 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 « Eloge des lapins », qui comportera un paragraphe de faux texte.

Normalement vous voici avec une belle page de garde 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{Eloge des lapins}

Votre gros paragraphe.

\end{document}
```

Les Lapins

John DOE

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

Petite démonstration pratique de l'utilisation de ces trois environnements ci-dessous (essayez de reproduire 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.

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.

Sauts

Dans cette sous-partie 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.

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

Code : TeX

```
Un canard\\un lapin\negline une pintade \newpage un ours
```

<p>Un canard un lapin une pintade</p> <p style="text-align: center;">1</p>	<p>un ours</p> <p style="text-align: center;">2</p>
--	---

Ce chapitre vous a permis de comprendre comment hiérarchiser votre contenu. Avouez-le, LaTeX n'est pas si compliqué que ça une fois la théorie passée. 🤖

Nous avons parlé de beaucoup de choses différentes ici, retenez que toutes les commandes explicitées dans ce chapitre permettent à LaTeX d'adapter une mise en page suivant le contenu, le niveau hiérarchique des informations ainsi leur emplacement dans votre publication.

Dans le prochain chapitre, nous n'allons pas suggérer à LaTeX des mises en page comme nous l'avons fait ici en hiérarchisant le contenu.

LaTeX se pliera à votre volonté et, par exemple, modifiera les marges des documents selon vos moindres désirs.

Ça ne rigole plus. 🐻

Maitriser sa mise en page (2/2)

Dans ce second chapitre de mise en page, nous n'allons pas structurer de l'information mais plutôt apprendre à mieux la 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 des éléments centraux dans une mise en page. Les marges vous permettront par exemple de relier vos documents plus facilement.

Les commandes sont simples à appréhender, tout comme celles du chapitre précédent.

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 cours sont ceux présentés dans le tableau ci-dessous (déjà vu dans le chapitre 3).

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

Nous avons, par le biais du type de document, défini la nature de la publication que nous composons : livre, lettre ou autre chose. 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 créer dans la partie 3)
- le positionnement des première pages de chapitres (il est possible de faire en sorte que chaque chapitre démarre sur une page de droite)
- l'information permettant de savoir si le document créé sera ou ne sera pas recto-verso.

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

Options applicables à la commande <code>\documentclass{}</code>		
Descriptions	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 rapport : oneside livre : twoside

Exemples pratiques

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

Code : TeX

```

\documentclass{book}

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

\begin{document}
\section{Un fort beau chapitre}
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Pellentesque aliquet, mauris in iaculis accumsan, tellus lacus
dictum risus, ut lacinia erat mauris ut libero.
Vestibulum suscipit massa vitae nunc semper eu aliquet ligula
sodales. Nulla vulputate mi vitae odio lobortis mattis.
Praesent sagittis turpis sit amet magna iaculis et mattis libero
blandit. Duis at diam sem.
Curabitur fermentum, risus nec aliquam dictum, nibh mauris dignissim
enim, et euismod odio ipsum sed purus.
Maecenas orci urna, consequat nec lacinia eget, fermentum mollis
diam. Fusce quis nulla non ipsum malesuada laoreet at non lacus.
Morbi a magna felis, eu fermentum mi.
Duis convallis pellentesque nibh at fermentum.
Praesent lobortis lobortis turpis, at semper augue accumsan a.
Praesent vestibulum consectetur interdum. Curabitur tristique
posuere sapien, id suscipit erat auctor ut.
Duis rutrum malesuada tortor et dapibus. Pellentesque ac tellus at
justo vestibulum hendrerit id vel dui.
\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}

```

Testez ces lignes, et si besoin rajoutez du faux texte pour mieux voir les différences.

Retenez de cette sous-partie que la commande `\documentclass{}` peut être personnalisée via 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 par exemple essentiel).

Nous allons ici apprendre à 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`, et utilisons la commande `\layout` introduite par celui-ci afin d'obtenir un gabarit nous donnant une idée de la taille de chaque marge.

Simultanément, nous imprimons un document saturé de texte afin de voir le rendu actuel des 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.

Code : TeX

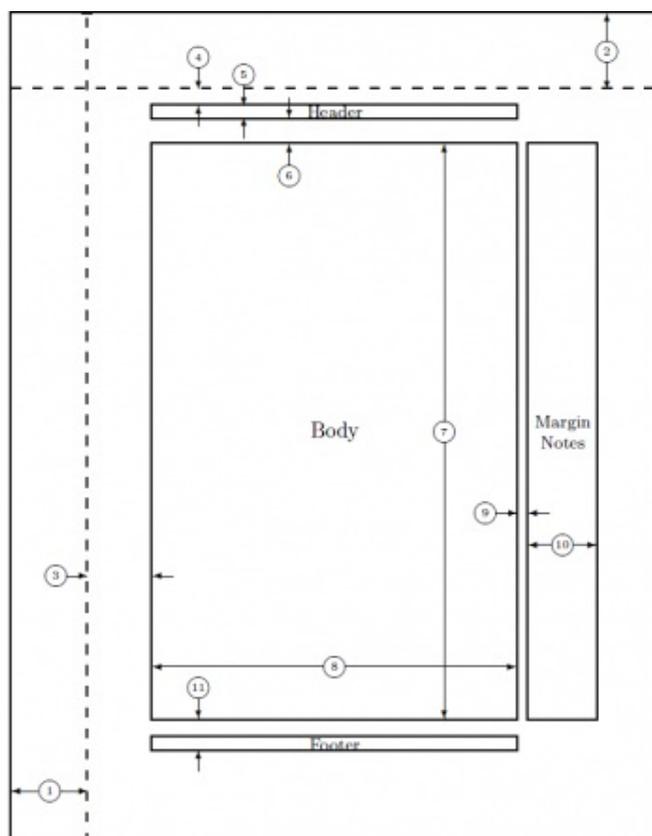
```
\documentclass{article}

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

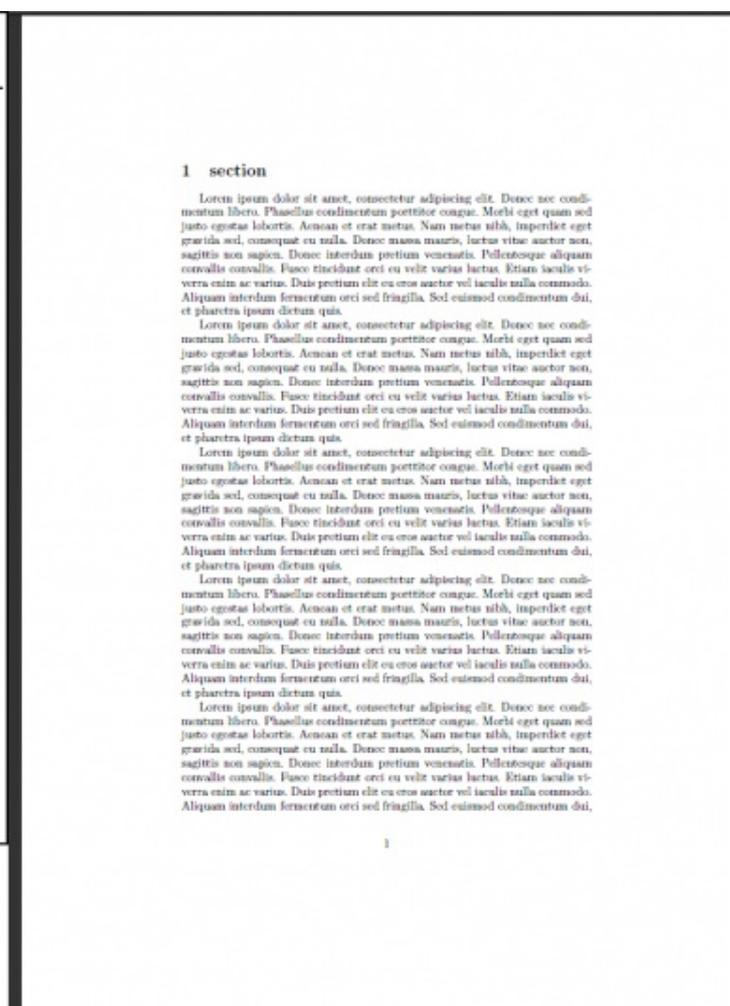
\begin{document}

\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



Ces deux éléments conjugués nous permettent de mieux visualiser les marges et leurs longueurs respectives les unes par rapport aux autres.

Afin de changer les marges, 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 selon la syntaxe suivante :

Code : TeX

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

Bien sûr, vous pouvez changer selon vos souhaits 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 de texte saturé pour voir le rendu des marges (facultatif).

Interlignes

Dans certains types de documentation, utiliser un interlignage une fois et demi ou deux fois plus grand que la normale est demandé. Nous allons voir ensemble comment ceux-ci peuvent être utilisés dans un document (il est aussi possible de faire des interlignes d'une taille définie par l'auteur, mais ce n'est pas notre souhait dans ce cours). Afin d'obtenir des interlignes personnalisés il faut utiliser le package `setspace` ainsi les commandes `\onehalfspacing` et `\doublespacing` dans le préambule, permettant respectivement d'avoir un interligne 1,5 fois et 2 fois plus grand que l'interligne habituel dans tout le document.

Pour ne changer les interlignes que dans des petits morceaux de votre composition, les environnements `onehalfspace` et `doublespace` sont adaptés. La démonstration ci-dessous les utilise.

Code : TeX

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

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

\begin{document}

\section{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.

\section{interligne intermédiaire}

\begin{onehalfspace}
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.
\end{onehalfspace}
```

```
\section{interligne double}

\begin{doublespace}
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.
\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.

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 ici 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 fréquent).

listes à puces

La création d'une liste à puce 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 et son rendu, de liste à puce.

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

listes numérotées

La création d'une liste numérotée se fait aussi en trois temps (seul le nom de 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.

Tout de même, un exemple codé et un rendu en image ne peuvent pas faire de mal.

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}
```

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

listes de description

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

Les mots écrits en gras lors de la compilation sont placés entre crochets juste après la commande `\item` dans le code source.

Démonstration ci-dessous :

Code : TeX

```
\begin{description}

\item[un canard :] bestiole qui fait coin
\item[un ornithorynque :] bestiole qui fait rire
\item[un ours :] bestiole qui fait mal

\end{description}
```

un canard : bestiole qui fait coin

un ornithorynque : bestiole qui fait rire

un ours : bestiole qui fait mal

Les listes n'ont rien de sorcier, les utiliser aérera vos documents et vous permettra par exemple de présenter 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.

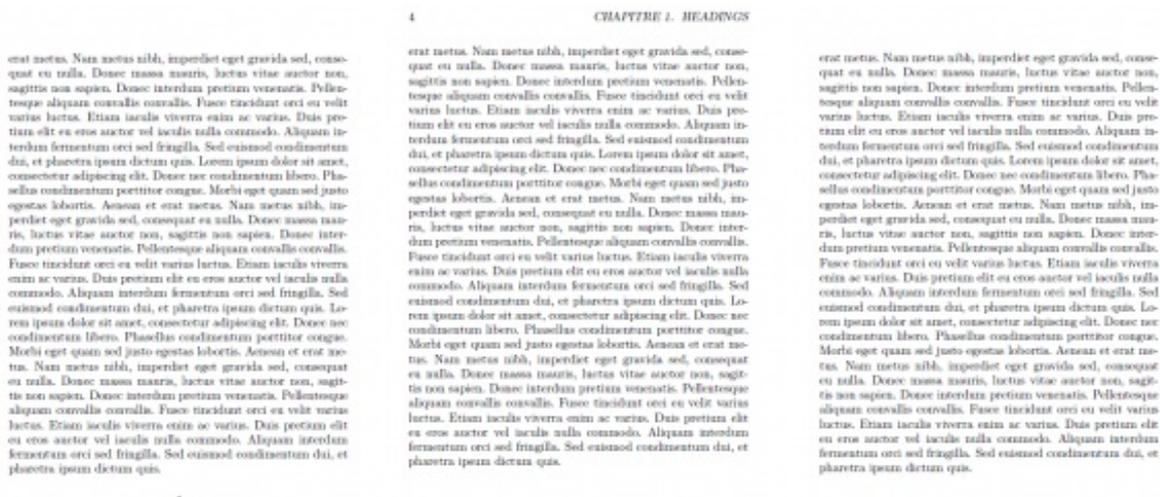
Il nous reste un élément essentiel à personnaliser : les en-têtes et pieds de pages.

Ces éléments ont été eux aussi 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 les packages et les mises en forme élaborées plus loin dans le cours).

Afin de changer les couples en-tête/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 un numéro de page dans le pied de page. Ce numéro de page est centré.
- 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.

Ci-dessous, les captures d'écrans de documents composés respectivement avec les balises plain, headings et enfin empty.



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

Rien de bien compliqué n'est ce pas ?

Lors du premier TP, nous apprendrons à personnaliser de manière bien plus approfondie nos entêtes et pieds de page. Les deux chapitres de mise en page sont terminés. Si vous avez pris le temps de refaire les exemples, vous devriez vous sentir familiarisé avec les commandes et la compilation.

Les chapitres de la partie 2 n'introduisent pas de commandes trop compliquées, néanmoins à partir du premier TP les choses vont commencer à se corser un peu et il sera plus facile pour vous de progresser si vous avez testé par vous-même les commandes données dans le cours.

Nous retiendrons de ce chapitre que :

- il est possible de paramétrer totalement les marges d'un document
- le faux texte permet de remplir une mise en page
- LaTeX simplifie la création des pages de garde
- les styles permettent de changer facilement les en-têtes et pieds de pages, mais ils ne permettent pas de les personnaliser

Le prochain chapitre traitera des polices et des options de mise en forme du texte (gras, italique...).

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).

A 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 par défaut d'un document.

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

Graisse, style, taille

Taille de texte

Nous allons dans ce paragraphe comprendre la façon dont la taille du texte peut être modifiée, de façon à pouvoir grossir certains mots ou certaines expressions par rapport à d'autres.

LaTeX propose dix commandes différentes permettant à votre texte de grossir ou de maigrir selon votre envie. Ces commandes s'utilisent très simplement et apparaissent sous 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 de rester sur la syntaxe `\commande{mon bout de texte}` car elle est similaire à la majorité des autres syntaxes utilisées dans LaTeX.

Revenons à nos commandes. Comme dit plus haut, le texte peut subir dix transformations différentes au niveau de la taille, cela nous donnant une échelle de taille de texte assez large.

```
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'eaaaauuuuuuuuu !
```

Une fois n'est pas coutume, un beau tableau devrait nous aider à y voir plus clair dans ce tas de commandes.

Taille de texte	
<code>\tiny</code>	Minuscule
<code>\scriptsize</code>	Très très petit
<code>\footnotesize</code>	Très petit
<code>\small</code>	Petit
<code>\normalsize</code>	Normale (définie dans <code>\documentclass</code>)
<code>\large</code>	Légèrement plus gros que la normale
<code>\Large</code>	Gros
<code>\LARGE</code>	Très gros
<code>\huge</code>	Très Très gros
<code>\Huge</code>	Enorme !

Testez ces commandes, vous serez surpris des changements qu'elles donneront dans vos mises en page.



Point important : Lors de l'écriture de `\documentclass` vous avez choisi une taille de police pour votre document (10 pt par défaut, que vous pouvez monter jusqu'à 12pt). 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	Un canard <code>un lapin</code>	Un canard ^{un lapin}
Encadrer	<code>\fbox{un lapin}</code>	un lapin
Soulignement Package soul	<code>\ul{un lapin}</code>	<u>un lapin</u>
Soulignement Package ulem	<code>\uuline{un lapin}</code>	<u>un lapin</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 barre ou le souligne. 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 compte, met-le en valeur ». LaTeX se chargera tout seul comme un grand de choisir la manière utilisée pour 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 sans l'italique.

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.

Vous constatez que `\emph{ }` change de comportement suivant le contexte, et en cela réside sa force.

Conclusion : utiliser l'italique peut souvent être une erreur, il vaut mieux 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 ici 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 8 couleurs par défaut

Nous utiliserons pour la colorisation du texte le package `color`.

Le package `color` propose par défaut 8 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}

```

EXERCICE : Reproduisez le texte ci-dessous grâce à la commande que vous venez d'apprendre, cela devrait être sans aucune difficulté.

J'écris un texte **chatoyant**.

SOLUTION :

Code : TeX

```

\documentclass{article}

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

J'écris un texte
\textcolor{red}{c}\textcolor{green}{h}\textcolor{cyan}{a}\textcolor{black}{t}\te

\end{document}

```

Création de nouvelles couleurs

Avec 8 couleurs, vous ne pouvez aller guère loin, raison pour laquelle de bonnes âmes ont créé un système vous permettant de créer d'en fabriquer 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 précédemment :

Code : TeX

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

Niveaux de gris

Définir une couleur en niveau de gris revient à choisir une couleur entre le blanc et le noir. Ici nous estimerons que le gris se trouve sur échelle située entre 0 (le noir) et 1 (le blanc). Choisir un niveau de gris consistera à choisir 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 dans la couleur que nous créerons.

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 ensemble 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 du cours vous avez compilé des choses et que l'envie vous a titillée de changer la police de vos textes.

Pour ce qui est des polices, LaTeX a des avantages à doubles tranchants. Rappelons-nous que LaTeX a été créé suite à l'exaspération de chercheurs face à la mise en page plutôt minable de leur texte. 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 police.

L'idée centrale des packs de police 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 :

- du texte avec empâtements
- du texte sans empâtements
- du texte façon machine à écrire
- du texte mathématique.

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

Il vous suffira d'appeler les packages souhaités via la commande `\usepackage{nom_du_pack}` .

Une fois n'est pas coutume, voici un joli tableau :

Pack de police (nom du package)	Exemple
Par défaut	<p>Texte avec empâtements</p> <p>Texte sans empâtements</p> <p>Texte type machine à écrire</p> $\int_4^8 kx dx \sum_{i=0}^n y_n^7 + 4569852321$
bookman	<p>Texte avec empâtements</p> <p>Texte sans empâtements</p> <p>Texte type machine à écrire</p> $\int_4^8 kx dx \sum_{i=0}^n y_n^7 + 4569852321$

charter	<p>Texte avec empattements</p> <p>Texte sans empattements</p> <p>Texte type machine à écrire</p> $\int_4^8 kx dx \sum_{i=0}^n y_n^7 + 4569852321$
newcent	<p>Texte avec empattements</p> <p>Texte sans empattements</p> <p>Texte type machine à écrire</p> $\int_4^8 kx dx \sum_{i=0}^n y_n^7 + 4569852321$
lmodern	<p>Texte avec empattements</p> <p>Texte sans empattements</p> <p>Texte type machine à écrire</p> $\int_4^8 kx dx \sum_{i=0}^n y_n^7 + 4569852321$

mathpazo	<p>Texte avec empattements</p> <p>Texte sans empattements</p> <p>Texte type machine à écrire</p> $\int_4^8 kx dx \sum_{i=0}^n y_n^7 + 4569852321$
mathptmx	<p>Texte avec empattements</p> <p>Texte sans empattements</p> <p>Texte type machine à écrire</p> $\int_4^8 kx dx \sum_{i=0}^n y_n^7 + 4569852321$

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
cmr	Computer Modern
lmr	Latin Modern Roman
lms	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>



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 sinon vos publications deviendront indigestes.

Les modifications de texte n'ont plus de secrets pour vous et vous pouvez selon vos souhaits :

- changer les styles
- grossir du texte
- changer les packs de police
- modifier ponctuellement les polices.

Le prochain chapitre traitera des notes et vous permettra de faire des citations et des notes de bas de page.

Celui-ci achèvera la partie 2 de ce cours.

Les notes

Nous allons aborder ensemble le chapitre traitant des différentes notes et références possibles dans LaTeX.

Ce chapitre vous permettra de :

- citer des textes
- insérer des paragraphes de code brut dans vos documents
- créer des minipages
- composer des notes de bas de page.

Les références internes seront abordées elles aussi dans ce chapitre.

Ce court chapitre est le dernier de la partie 2, c'est le dernier chapitre présentant des commandes « simples ». La partie 3 et le tp introduisant des commandes aux utilisations plus ardues, prenez le temps de vous entraîner lors de ce chapitre et reproduisez les exemples donnés afin de continuer de vous familiariser avec les commandes et les packages.

LaTeX n'est pas un langage difficile mais il s'acquiert par la pratique, ne la négligez pas.

Citation, code brut et url

Cette première sous-partie développe simplement les manières de créer des citations de texte, de code ainsi que la façon de compiler 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.

Code : TeX

```

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.
\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}
sit amet suscipit tortor pulvinar sed. Proin laoreet,
orci iaculis ullamcorper pulvinar, lectus sapien accumsan nulla,
ut condimentum orci diam sit amet justo.

```

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.

Le code source donné ci-dessus vous montre la différence entre quote et quotation : le nombre de tabulations avant le premier

mot.

Les citations de code

Trois méthodes majeures se partagent les différents types de citations de code.

La commande `\verb`

Pour insérer du code dans un paragraphe, le plus simple reste d'utiliser la commande `\verb`. Cette commande a la particularité de délimiter le texte concerné par son application par le caractère venant juste après `verb`.

Les trois premières lignes de codes donnent donc le même résultat et 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») :

Code : TeX

```
\verb[lapin & lapin {}] [
\verb|lapin & lapin {}| %les usagers utilisent la majeure parti du
temps le signe |
\verb(lapin & lapin {})(
\verb&lapin & lapin {}&
```

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 le nombre d'espaces liés à une tabulation, vous pouvez utiliser l'environnement `\begin{verbatimtab}[nombre_d'espace_par_tabulation]` fourni par le package supplémentaire `moreverb`.

Code : TeX

```
\documentclass{article}

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

\begin{verbatimtab}[10]
votre code
\end{verbatimtab}
```

```
\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 dans l'en-tête du document, vous devez appeler le package `listings` 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 les numéros de lignes à
droite (right) ou à gauche (left)
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 mis à jour constamment sur [la page de wikibooks consacrée au package listings](#). Allez y jetez 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`.

Petit exemple ci-dessous de traitement de code C (tiré du tutoriel de C du site du zéro) avec `lstlisting`.

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_!");
```

Gestion des urls

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

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 être parfois 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.

Répétons 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{mon bout de texte}` a déjà été vue précédemment. Elle permet d'encadrer des portions de texte.

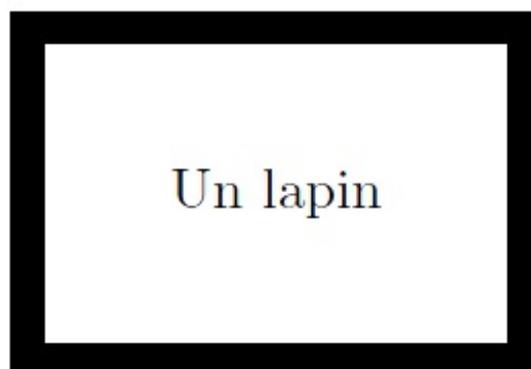
Bien que nous ayons vu la commande dans le chapitre précédent, nous n'avons pas vu les paramètres applicables à celle-ci. `\fbox` est une commande puissante, pour laquelle il est possible de paramétrer des tas de choses. Néanmoins nous limiterons l'usage de `\fbox` à l'encadrement de texte dans ce cours.

Les deux paramètres qui nous intéressent ici sont l'écart entre le texte et la bordure ainsi que l'épaisseur de cette dernière.

Petite démonstration de la méthode permettant de changer ces deux paramètres et capture d'écran du résultat ci-dessous.

Code : TeX

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



L'environnement minipage

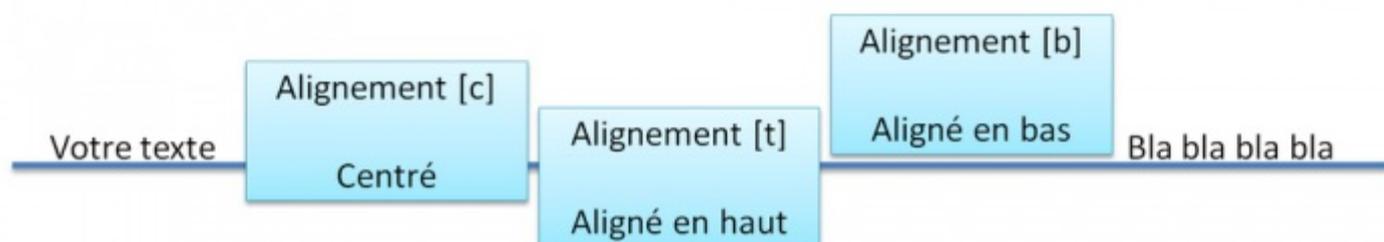
Une minipage est un encart de texte de largeur choisie encadré en noir. A l'intérieur de cet encart de texte, vous pourrez disposer et utiliser des environnements comme si cette minipage était un document à part entière.

En quelque sorte, une minipage est 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 : sa largeur et son alignement.

Pour ce qui est de la largeur, il est facile de comprendre le rôle du paramètre, il s'exprime en cm. Le paramètre d'alignement quant à lui est plus fin à comprendre. L'alignement est vertical et défini par rapport au niveau du texte. Trois alignements vous sont proposés.

Un schéma vaut mieux qu'un long discours pour visualiser les trois alignements proposés.



Ici, une démonstration d'une minipage centrée de 5cm de large et centrée verticalement obtenue via la commande :

Code : TeX

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

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 pages, références internes

Cette sous-partie traite des notes de bas de pages ainsi que des références internes dans les documents.

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

Notes de bas de pages

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 aux emplacements où renvoyer vers une note de bas de page est souhaité.

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 concernés, trois solutions sont disponibles :

- coller l'exposant au mot concerné
- utiliser une espace, qui peut parfois être trop large
- insérer via la commande `\`, une espace plus fin.

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

La commande `\footnotemark`

Ici l'opération se passe en deux temps. Dans un premier temps vous marquez les éléments concernés par des notes de bas de pages via la commande `\footnotemark[numéro_de_la_note]`, les notes de bas de pages correspondantes aux numéros sont données par la commande `\footnotetext[numéro_de_la_note]{Texte correspondant}`.

Ici deux compilations sont nécessaires, l'une pour que LaTeX liste les marques, l'autre pour que LaTeX associe les notes de bas de pages aux marques. Cette méthode est souvent salvatrice lorsque des erreurs apparaissent avec `\footnote`. **Utilisez-la préférentiellement.**

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

-
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.

Les commandes `\ref{nom_choisi}` et `\pageref{nom_choisi}` vous permettent d'appeler la page ou la référence de l'élément marqué dans une autre zone de votre document.

Petite démonstration (avec rendu en suivant) :

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}
```

Sur la page où la référence est faite, nous pouvons lire :

2.1 Section

2.1.1 Une sous-section

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

Les **citations** et **références** diverses n'ont maintenant plus aucun secret pour vous. Vous pourrez dès à présent mettre en valeur de façon forte certaines parts de vos publications via des **minipages** et relier les pages entres elles via des **références internes**.

Ce chapitre est le dernier de la partie deux, nous allons attaquer ensemble notre premier gros exercice d'application : la composition d'un livre (court, rassurez-vous) contenant l'intégralité de la déclaration universelle des droits de l'homme et du citoyen.

Jeune Zéro, après cette épreuve tu seras un Padawan ! 😊

TP : Mise en page de la Déclaration Universelle des Droits de l'Homme en Latex

LaTeX a envahi votre vie il y a de ça huit chapitres, et 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 indépendantes, 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 mis à profit et il y a fort à parier que vous aurez besoin d'aller les consulter pour répondre aux problématiques de l'exercice.

Le titre du chapitre ne laissant aucun suspens, je vous annonce que nous allons mettre en page une publication centrée autour de la déclaration universelle des droits de l'homme. Texte fondateur de notre civilisation moderne, il vous permettra de conjuguer apprentissage et culture.

Les consignes

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

Le document produit dans ce tp est téléchargeable en [cliquant ici](#). Il est sobre mais élégant, de quoi donner le sourire à vos correcteurs. 😊

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 et de progresser dans votre maîtrise de LaTeX.

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

La partie trois suivant ce TP développe les commandes avancées de LaTeX telles que les images, les tableaux, les 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 de grandes quantités de textes à copier-coller.

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 disponible [ici](#).

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

- le titre du livre
- vos **noms** et prénoms
- la date du jour.

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

Il contiendra le texte suivant :

Citation : Wikipédia

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 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 : Wikipédia

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.

La première **section**, intitulée « Introduction », contient l'extrait ci-dessous :

Citation : Wikipédia

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.

La 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 : Wikipédia

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 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.

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 ci-dessous :

Citation : Wikipédia

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 France et 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 Un, « Tous les Hommes naissent et demeurent libres et égaux en droit », 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 le texte suivant :

Citation : Wikipédia

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. 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.

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) portera en son sein 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 ci-dessous (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>

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.

Consignes additionnelles et pistes

- Le pack de police utilisé est lmodern.
- Les noms propres seront écrits en petites majuscules (convention française).
- Les entêtes et pieds de pages seront de type "headings" (réglage par défaut pour book).
- La marge de droite est de 3 cm.
- La marge de gauche est de 6 cm.
- La marge du bas est de 5 cm.
- La marge du haut est de 5 cm.
- Les adresses internet ne doivent pas tenir sur plus d'une ligne, mais vous avez le droit d'utiliser un réducteur d'url de type 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 afin de régler les erreurs une par une. 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 vous suffit de prendre votre temps.

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

Accents et signes											
Ó	<code>\'{}o</code>	Ô	<code>\.{}o</code>	Œ	<code>\b{}o</code>	Ò	<code>\{}o</code>	Ø	<code>\o</code>	Ö	<code>\"{}o</code>
Û	<code>\u{}o</code>	Ō	<code>\={}o</code>	Ŧ	<code>\d{}o</code>	Õ	<code>\~{}o</code>	Ô	<code>\^{}o</code>	Ŏ	<code>\H{}o</code>
Å	<code>\aa</code>	Š	<code>\v s</code>	‡	<code>\ddag</code>	†	<code>\dag</code>	ŌŌ	<code>\t{}oo</code>	ß	<code>\ss</code>
Ø	<code>\o</code>	æ	<code>\ae</code>	Æ	<code>\AE</code>	Œ	<code>\c{}o</code>	l	<code>\i</code>	¶	<code>\P</code>
©	<code>\copyright</code>	œ	<code>\oe</code>	Œ	<code>\OE</code>	Œ	<code>\d s</code>	Š	<code>\r s</code>	Š	<code>\H s</code>
J	<code>\j</code>	Å	<code>\AA</code>	Š	<code>\t s</code>	§	<code>\S</code>	£	<code>\pounds</code>		

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 pas à pas les erreurs.

Etape 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 chose près à celui-ci :

Code : TeX

```
\documentclass{book}

\begin{document}

  \frontmatter
  \mainmatter
  \appendix
  \backmatter

\end{document}
```

Etape 2 : la langue

Le texte demandait d'écrire en français et il était visible que nombres 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}
```

Etape 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 serons utiles pour continuer l'exercice. Ici, trois autres packages étaient demandés : `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}
%pour les marges
```

Etape 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 majuscules) 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.

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

Etape 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 leur « * ».

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}
%pour 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 jusque ici, c'est déjà très bien. Félicitations, vous commencez à vous sentir à l'aise avec LaTeX et ses concepts élémentaires ! 😊

Etape 6 : le contenu

Les choses sérieuses commencent. Ici se trouvent les plus grosses sources d'erreurs d'étourderies. 😊. 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 majuscules, 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. 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 présentait des apostrophes penchées, des exposants ainsi que des guillemets français. De plus il était souhaité qu'il soit gros et en italique.

Code : TeX

```

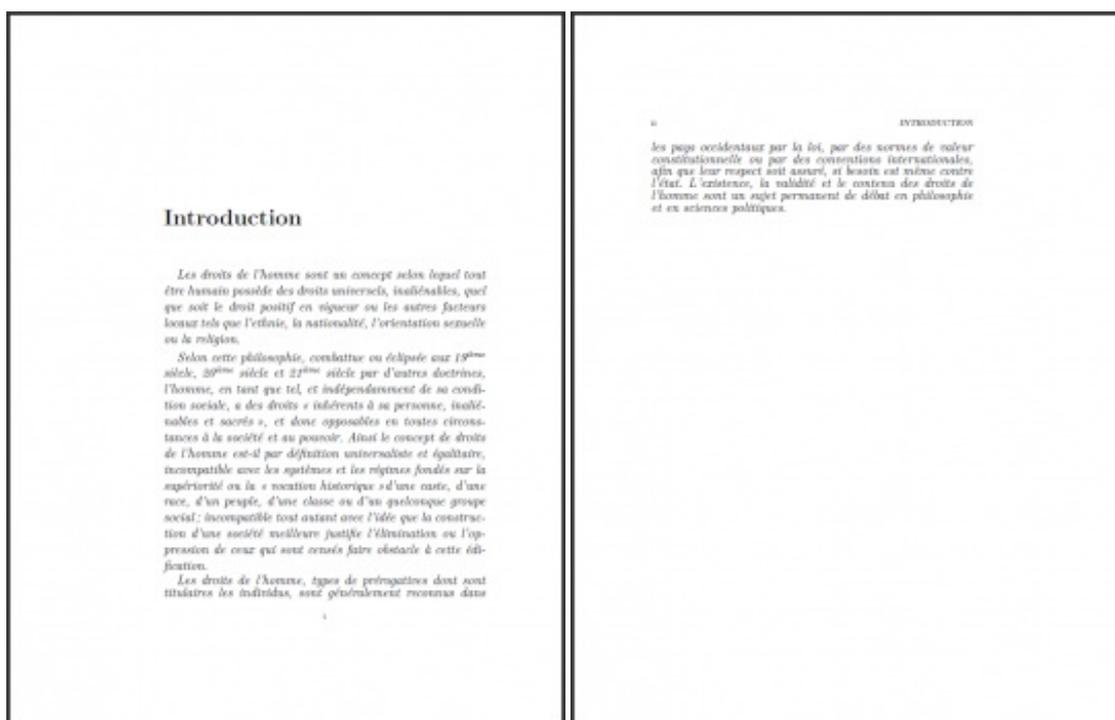
\begin{it}
\Large{
  Les droits de l'homme sont un concept selon lequel tout être
  humain possède des droits universels, inaliénables, quel que soit le
  droit positif en vigueur ou les autres facteurs locaux tels que
  l'ethnie, la nationalité, l'orientation sexuelle ou la religion.
}
\end{it}

```

Selon cette philosophie, combattue ou éclipsée aux 19^{èmes} siècle, 20^{èmes} siècle et 21^{èmes} 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.

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



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

Rien de transcendant, les commandes utilisées ici ont été aussi utilisées dans le paragraphe précédent. La clé du succès était juste de prendre votre temps et de compiler souvent afin de mieux voir vos erreurs.

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 `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.



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

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>

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 l'étape durant laquelle nous allons regrouper tous les éléments du fichier (je l'ai appelé tp.tex, libre à vous de l'appeler autrement).

Une fois tous les morceaux rassemblés, le code source est le suivant:

Code : TeX

```
\documentclass{book}

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

\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 19¹⁹ siècles, 20²⁰ siècle et 21²¹ 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 **\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.

```
\backmatter
```

```
\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}
```

```
\end{document}
```

Conclusion

Vous voici arrivé à la fin de cette correction. Vous avez pu constater qu'une bonne organisation permet d'avancer vite dans la création de documents avec LaTeX. 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 (et cela peut très vite arriver, lors de la création 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 ils 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 apprendre 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 (la commande `\LaTeX` sert à écrire le mot LaTeX) :

Code : TeX

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

canard lapin
L^AT_EX lapin
L^AT_EX lapin

Constatez que dans la deuxième ligne il n'y a pas d'espace entre les deux mots alors que vous en aviez mis un dans le code. La troisième ligne, via l'insertion de `{ }` vient palier 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 un espace à leur suite.

Règle toute simple, elle vous sauvera dans de nombreuses situations (je n'ai pas utilisé cette règle dans le TP, vu 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. LaTeX lors de la création d'un layout (via la commande `\layout` que nous avons déjà vu dans le chapitre concernant les marges) vous montre toute une liste de paramètres.

Chaque paramètre d'un layout est paramétrable et nous allons apprendre ici même à les modifier.



Ces informations sont données à titre de précision, si vos besoins en terme 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 :

Code : TeX

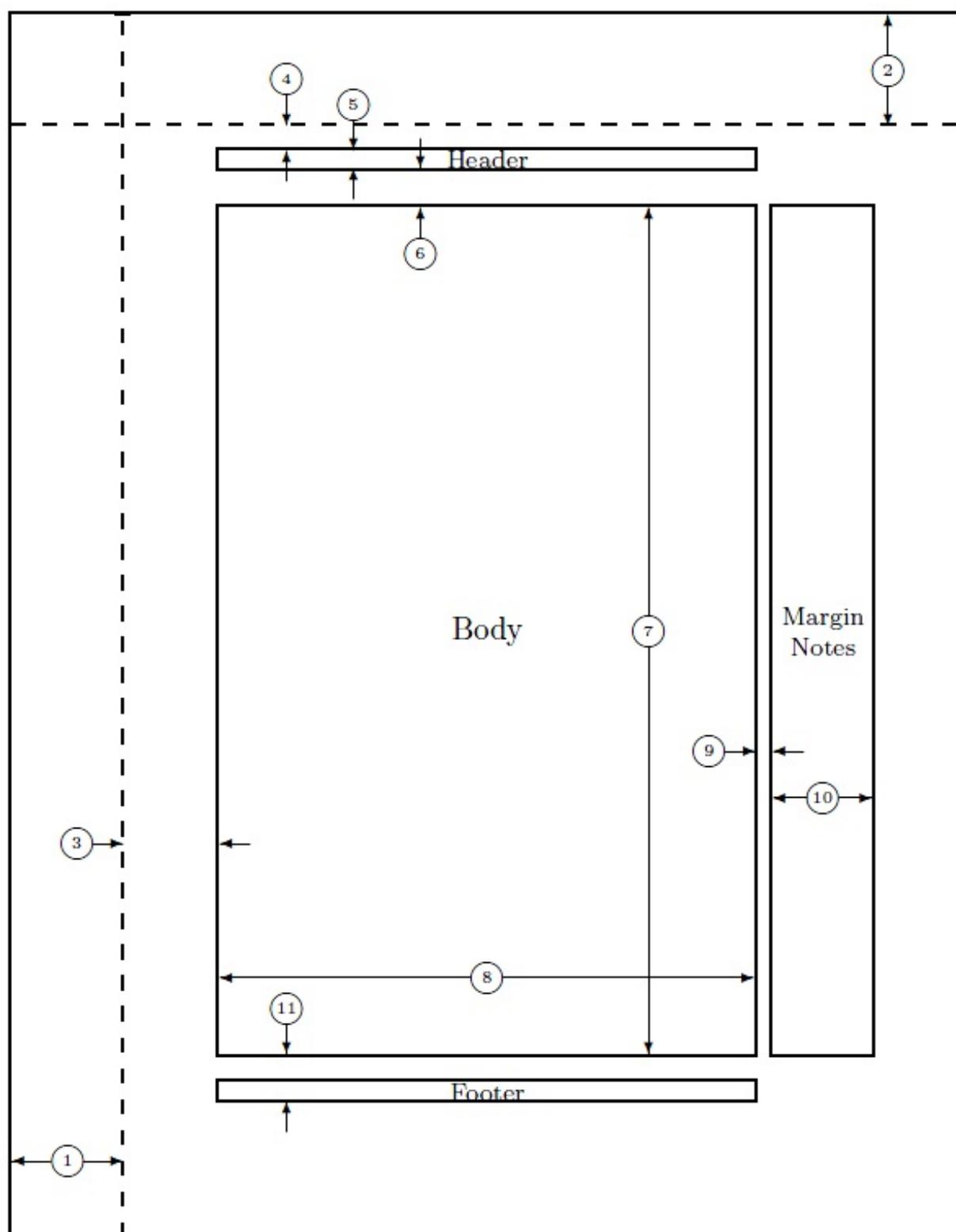
```
\documentclass{article}

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

\begin{document}

\layout

\end{document}
```



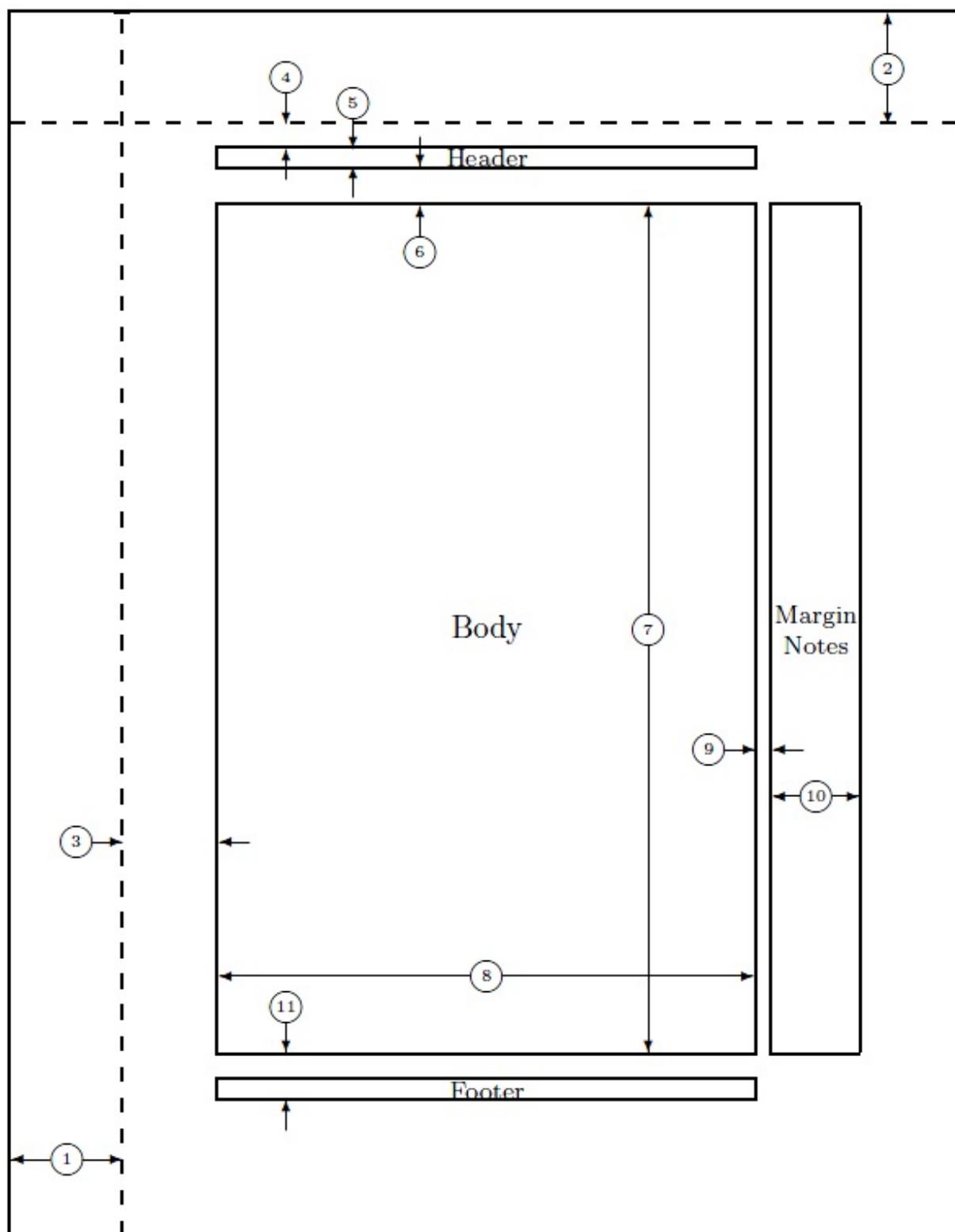
1	<code>one inch + \hoffset</code>	2	<code>one inch + \voffset</code>
3	<code>\oddsidemargin = 62pt</code>	4	<code>\topmargin = 16pt</code>
5	<code>\headheight = 12pt</code>	6	<code>\headsep = 25pt</code>
7	<code>\textheight = 550pt</code>	8	<code>\textwidth = 345pt</code>
9	<code>\marginparsep = 11pt</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>

Chaque longueur est associée à une commande et à un repère sur le layout, de sorte que vous pouvez facilement identifier les différentes longueurs et les associer à leurs noms.

Je vous montre ici comment modifier une longueur du layout. Prenons la longueur `\marginparwidth` repérée par le nombre « 10 » dans le layout, pour la modifier 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 plait}  
\setlength{\marginparwidth}{2cm} %les centimètres seront  
automatiquement convertis
```



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>

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



La liste des unités de longueur vous sera fournie dans l'une des annexes.

Personnalisation de la commande `\emph{}`

Nous avons vu lors du chapitre traitant du texte ainsi que des polices que la commande `emph{}` sert non pas à mettre en italique du texte, mais à signaler son 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 les commandes `emph{}` transforment le texte en texte type machine à écrire plutôt qu'en italique. Le résultat en serait le suivant (capture d'écran du dernier paragraphe du TP) :

Code : TeX

```
\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}
```

La capture d'écran :

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.

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` et 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 customisés

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

Jeune zéro, te voici padawan et 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 pieds de page en trois morceaux, de sorte que nous devons personnaliser six commandes différentes pour arriver à créer une mise en page cohérente des entêtes 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 pieds de page.
- `\cfoot{...}` gère le centre du pieds de page.
- `\rfoot{...}` gère le coin droit du pieds de page.

Différentes 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'entête et le corps du texte.
- `\renewcommand{\footrulewidth}` { épaisseur dans l'unité de votre choix } : épaisseur de la ligne entre le pieds 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 page à la carte via la commande `\thispagestyle{}` qui peut se voir attribuer lui 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.

Ce Tp a été pour vous votre première expérience complète de mise en page et de rédaction. Les choses ne sont pas évidentes si elles sont prises toutes ensemble mais faciles à appréhender si le travail est fait avec méthode.

J'ai choisi d'introduire de nombreux éléments d'ouverture dans ce Tp afin de vous permettre de composer des documents plus complexes et à la mise en page plus subtile si vous le souhaitez.

Lorsque l'écriture du cours sera achevée, je rajouterai d'autres éléments ici mêmes et dans les chapitres précédents afin de vous aider à être plus efficaces et plus performants. L'idée sous-jacente aux sous-parties et à l'annexe « aller plus loin » est de vous faire comprendre que tout dans LaTeX peut être approfondi et paramétré.

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 du cours introduit les manipulations de figures. Les concepts et packages introduits ici, bien qu'assez faciles à prendre en main dans des cas simples, peuvent se révéler extrêmement agaçants dans des situations plus exotiques, et les plus expérimentés connaissent la difficulté réelle rencontrée lors du placement 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 nouvel objet fera son apparition dans ce tutoriel : Le flottant. Nous prendrons le temps de l'étudier.

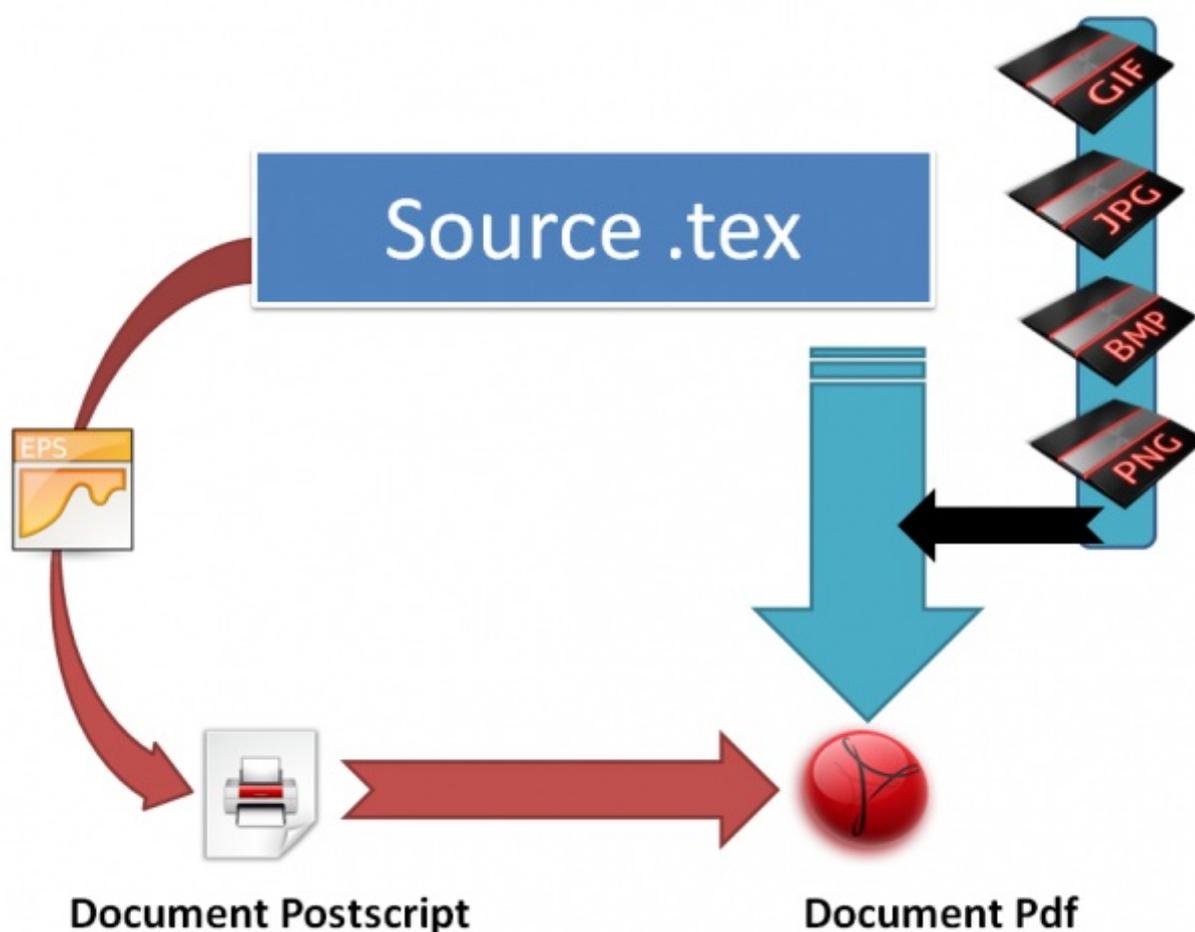
Les différents formats d'images

LaTeX et ses packages vous permettent d'insérer des figures et des images de divers formats afin de les mettre en page selon vos souhaits.

Une chose importante est à noter : **tous les formats d'images ne sont pas compatibles entre eux.**

Raison invoquée : la compilation. Nous avons vu lors du troisième chapitre du cours qu'un fichier .tex pouvait selon les besoins être compilé en des fichiers aux extensions variées (nous avons vu les ps et pdf). Selon les méthodes de compilation, vous pourrez ou ne pourrez pas utiliser les différents formats d'image.

Avant de donner des explications techniques, un schéma s'impose.



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 conversions et de compilation ont été abordées dans le chapitre 3 en détail, revenez-y si elles ne sont pas claires pour vous).

Cette différence nous intéresse car le schéma montre qu'une compilation directe en pdf ne permet pas l'insertion de figures aux formats .eps alors qu'une compilation d'un fichier .tex en un fichier postscript l'autorise.

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, ps) ou d'images (bmp, gif, png, jpeg) se pose dès la conception d'un document.



Une image/figure peut se convertir facilement d'un format à un autre en l'ouvrant avec des logiciels d'édition (Gimp, Photoshop...) avant de l'enregistrer de nouveau sous un nouveau format.

Votre choix doit se faire selon le type de fichier que vous manipulez. Des chercheurs auront plutôt tendance à travailler avec des postscript alors qu'un étudiant préférera utiliser des images au format png. Quel que soit votre choix, les packages ainsi que les commandes utilisées par la suite resteront 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 aussi conserver votre fichier en ps). Il vous sera interdit d'utiliser des formats tels que png, bmp, jpeg ou 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 et 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 dans la sous partie précédente, nous pouvons maintenant aborder la pratique.

Nous allons apprendre à insérer une ou plusieurs images dans un document. Les opérations effectuées dans cette sous-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.

Afin de travailler une image, il nous faut tout d'abord...une image. Voici donc la wonder Poulpy, qui a accepté de se faire photographier en exclusivité afin que nous puissions inclure et manipuler sa photo.



Téléchargez cette image en effectuant un clic droit avant de l'enregistrer sous le nom de votre choix (sans accents et sans espace) dans le dossier où vous composez vos documents LaTeX. Pour ma part je l'ai appelé 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 au choix spécifier un chemin absolu ou un chemin relatif.



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 par rapport au fichier source .tex pour retrouver une image. Si le fichier .tex se trouve dans le répertoire « fichierslatex », le chemin relatif de l'image sera chapitre9/poulpy.png.

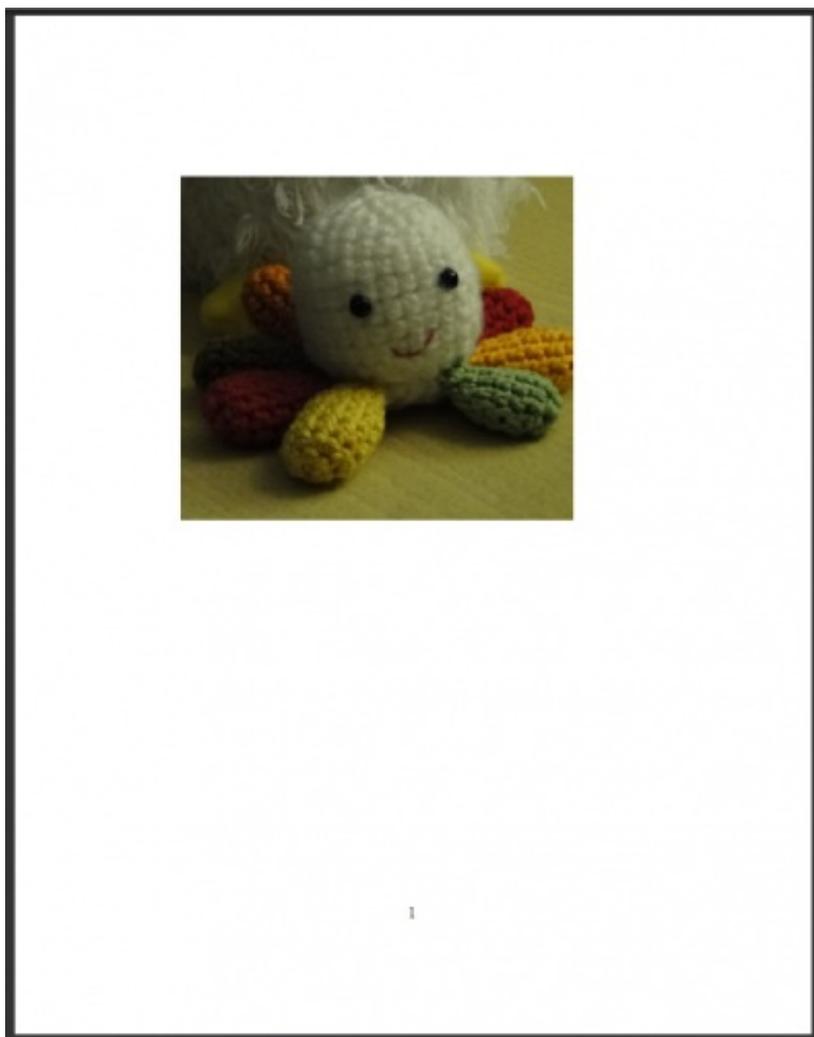
Code : TeX

```
\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 par défaut et calée en haut à gauche du document.



Taille d'une image

Plusieurs solutions se proposent à vous pour spécifier à LaTeX votre choix de taille pour une image donnée. Vous pouvez selon vos souhaits lui demander de :

- faire en sorte qu'une image ait une certaine largeur et s'adapte en hauteur afin de conserver ses proportions
- faire en sorte qu'une image ait une certaine hauteur et s'adapte en largeur
- fixer à la fois la hauteur et la largeur, quitte à déformer l'image
- choisir un coefficient, une échelle, permettant de réduire de façon cohérente et simultanée la hauteur et la largeur de l'image.

La longueur et la hauteur sont fixées respectivement par les variables `width` et `height`. L'échelle quant à elle se fixe via la variable `scale` et un coefficient. Ce coefficient est de 1 pour la taille par défaut de l'image, à vous de le diminuer ou de l'augmenter selon vos désirs. Ci-dessous plusieurs codes sources de démonstration.

Code : TeX

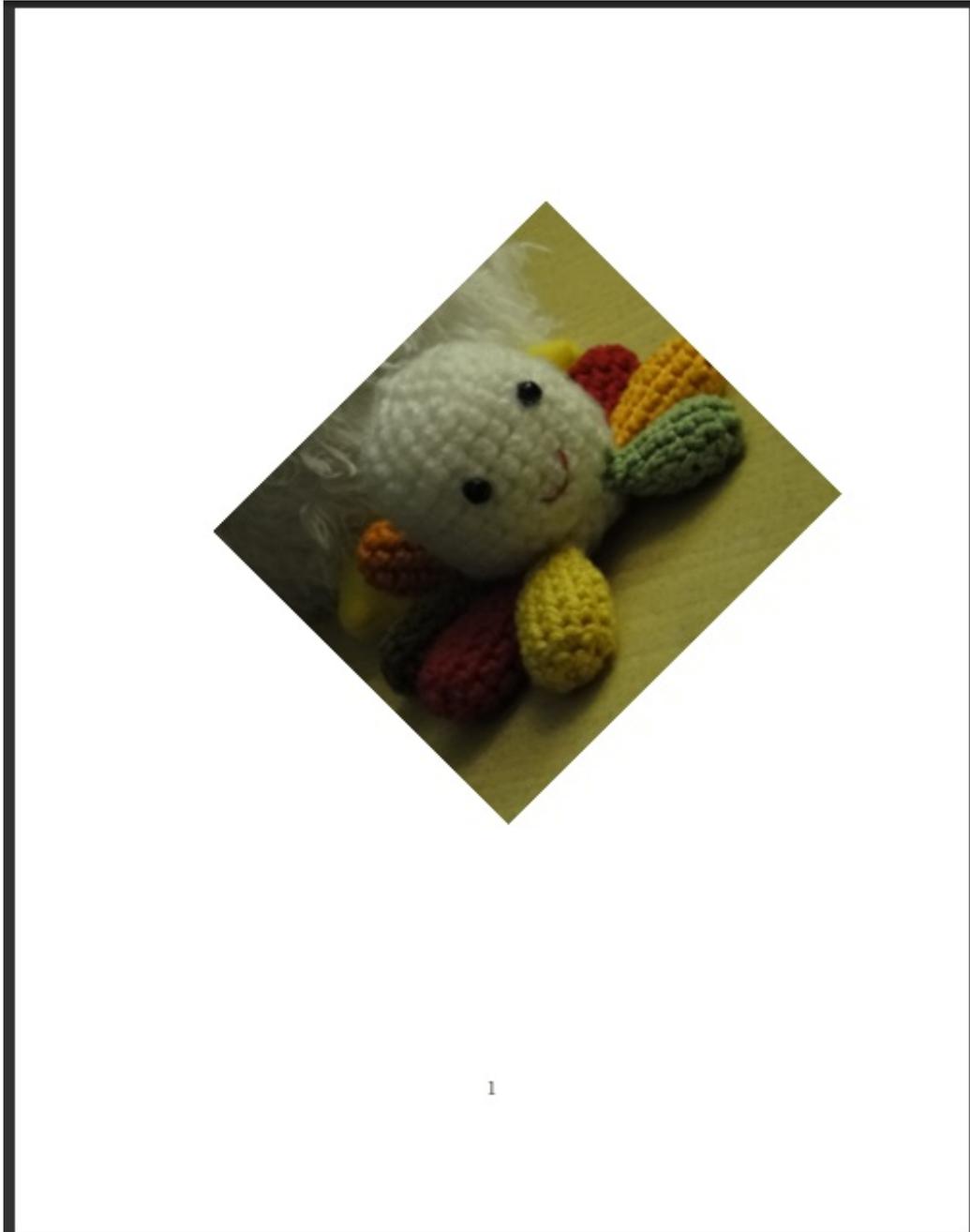
```
\includegraphics[width=200]{poulpy.png}
\includegraphics[height=200]{poulpy.png}
\includegraphics[height=200, width=600]{poulpy.png} %Ici poulpy
est un peu plate
\includegraphics[scale=1.5]{poulpy.png} %Ici poulpy est plutôt
grande
```

Rotation d'une image

La variable `angle` nous permet de faire tourner une image dans un document. Le rendu sur des images rectangulaires est assez laid mais pour d'autres types d'images, cela peut être utile. Démonstration :

Code : TeX

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



Insertion d'une portion d'image

Il peut être agréable d'insérer une portion d'image 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.

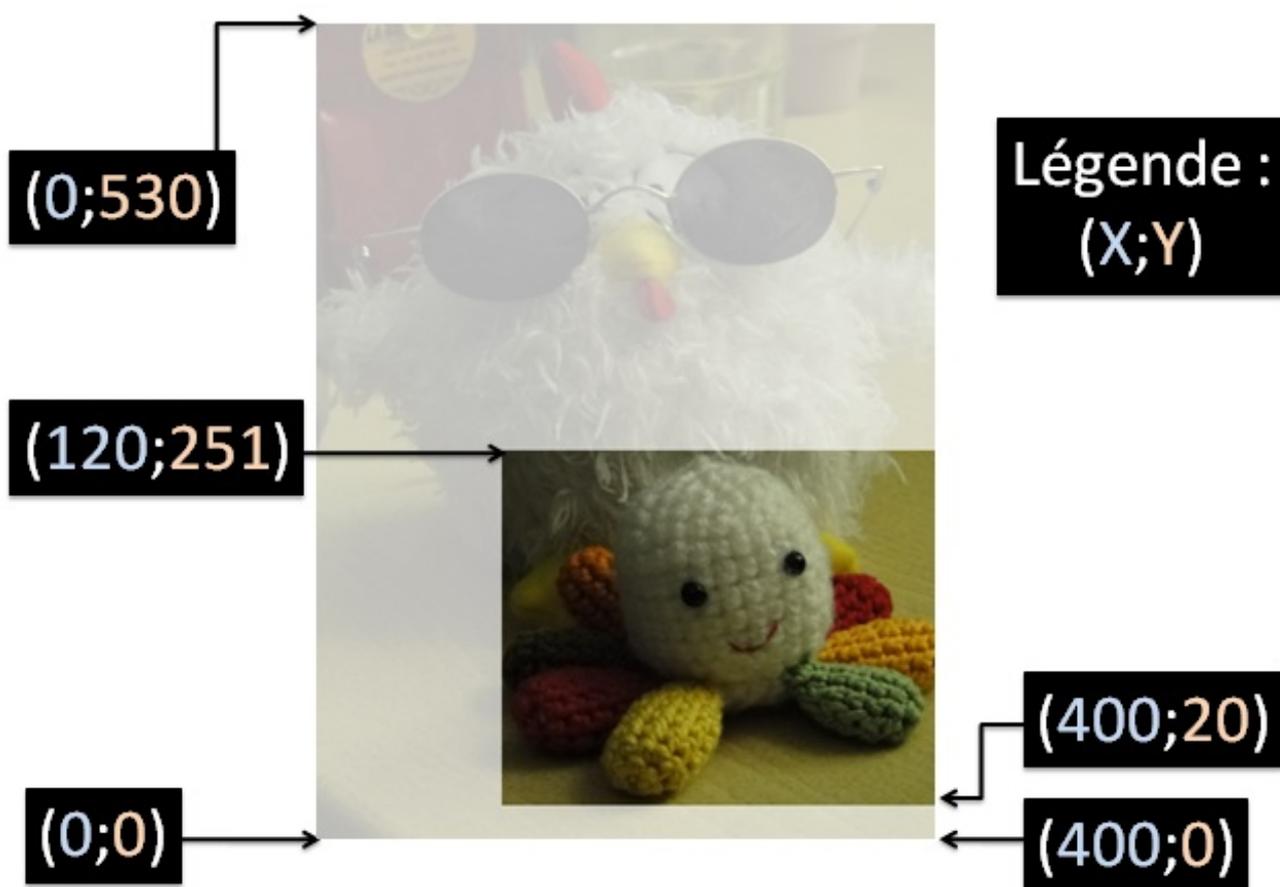


Les subtilités développées dans ce paragraphe ne fonctionnent (facilement) qu'avec les fichiers postscript.

Pour ce faire nous allons nous exercer sur la photo suivante, que vous pouvez télécharger au format eps en [cliquant ici](#) :



Notre but va être ici d'isoler Poulpy afin d'insérer sa photo dans un nouveau document sans son compagnon à plume. Chaque pixel de l'image représente un point. A chaque point sont associés une abscisse et une ordonnée, l'origine se trouvant au coin inférieur gauche de l'image. Si nous souhaitons isoler Poulpy, notre photo de départ devient la suivante.



La commande à utiliser afin de découper convenablement un rectangle dans une image est la suivante :

Code : TeX

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

Tentez d'isoler Poulpy, je donne la solution à ceux qui n'aurait 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 ici apprendre à intégrer une image dans du texte tout en faisant en sorte 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 `wrapfig`. Il est nécessaire de l'appeler lorsque vous souhaitez inclure l'environnement `wrapfigure` afin d'encadrer des images/figures par le texte présent dans un paragraphe. Plusieurs variables font leur apparition lors de 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 ligne]{placement}{largeur de l'image en cm}
\includegraphics[width=largeur en cm]{votre image}
\end{wrapfigure}
Votre paragraphe sans sauter de ligne entre \end et le paragraphe
```

Le placement se définit à partir des lettres :

- l pour placer à gauche
- r pour placer à droite
- o pour placer à l'extérieur, c'est-à-dire à droite pour une page impaire et à gauche pour une page paire
- i pour placer 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 et occuper 8 lignes de haut à droite de votre paragraphe.

Solution de l'exercice ci-dessous.

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}
\end{document}
```

```

\end{wrapfigure}
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.
\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.



Les flottants

Les techniques d'insertion d'images citées dans les paragraphes précédents sont des solutions simples mais néanmoins peu utilisées. LaTeX propose aux auteurs d'optimiser le placement des images et des figures à leur place.

Afin de laisser à LaTeX la possibilité de placer les images à sa guise, nous allons placer les images dans des environnements dits « flottants ». Le contenu de ces environnements sera placé de la manière qui semblera à LaTeX la plus judicieuse dans votre mise en page.

Bien sûr, vous pourrez suggérer à LaTeX votre préférence pour tel ou tel type de placement.

Après avoir appris à placer des flottants contenant des images, nous apprendrons à les légender. Ces mêmes légendes seront traitées dans des chapitres ultérieurs.

C'est parti ! 🤖

L'environnement figure

La création d'un flottant

Afin de créer un 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 une simple fioriture de mise en page, il nous permettra un peu plus tard 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}
```



Le placement

Une option permet de spécifier à l'environnement figure (et par extension à LaTeX) vos préférences en terme de placement.

Vous pouvez souhaiter que votre flottant soit :

- plutôt placé en haut de page (option t)
- plutôt placé en bas de page (option b)
- sur une page ne comportant que des flottants (option p).

Deux autres options s'offrent à vous. Vous pouvez spécifier à LaTeX votre souhait de 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'utiliser avec modération. Laissez LaTeX faire son travail avant de lui imposer des contraintes.

Les options de placements sont spécifiées dans la balise d'ouverture de l'environnement figure selon la syntaxe suivante.

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 aimerions « énormément » 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 selon l'une des options développées ci dessus. La commande `\floatplacement{type de flottant}{options}` s'utilise afin de changer les

comportements de LaTeX par défaut.

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 ses figures et images est indispensable en LaTeX. Nous allons ici apprendre à apposer des légendes sous nos flottants. La commande `\caption{ma légende}` sert à cela, elle s'utilise à la suite de l'environnement `center` et précède une éventuelle commande `\label` .

Démonstration pratique :

Code : TeX

```
\begin{figure}
\begin{center}
\includegraphics{poulpy.png}
\end{center}
\caption{Poulpy est multicolore}
\label{Poulpy est multicolore}

\end{figure}
```



FIGURE 1 – Poulpy est multicolore

Sauts de page

Nous avons appris dans les chapitres de mise en page à demander 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 en page tous les flottants (il les traitera tous et produira des pages remplies par les flottants non traités).
- `\cleardoublepage` , qui a le même effet hormis le fait qu'il redémarre 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 intéressant et primordial qu'aucune image de la première partie ne se retrouve dans la seconde à cause d'un choix de LaTeX. Ces deux commandes s'utilisent exactement de la même manière que `\newpage` .

Ce chapitre comportait beaucoup de nouveautés et de concepts. Mine de rien, son contenu cause de nombreux problèmes de mise en pages même aux grands acharnés de LaTeX. Ne soyez pas surpris si dans un premier temps les flottants ne vous obéissent que de façon relative, leur maîtrise vient avec l'expérience et de nombreux essais. 🤔

Nous retiendrons de ce chapitre qu'un choix est à faire entre le format postscript et les autres formats d'images (ce choix entraînant aussi une possible modification de la méthode de compilation). De même, nous nous souviendrons des commandes permettant de manipuler des flottants.

Prochain chapitre : les tableaux, un nouveau type de flottant. C'est le chapitre le plus difficile du cours mais je vous aiderai à mieux les comprendre par de multiples exemples. Pas d'inquiétudes, nous allons prendre le temps de comprendre. 😊

Les tableaux

En science, 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 gérer comme tels.

Les tableaux, tout comme les maths (que nous étudierons dans le prochain chapitre), font l'objet d'une documentation extrêmement fournie (ainsi que d'un nombre de commandes impressionnant). Ce chapitre traitera plusieurs points utiles à la création des tableaux en vous apprenant à :

- créer 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 sous-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 suivant (la création va se dérouler en trois temps) :

1	2
3	4

Nous devons décider dans un premier temps de l'alignement des cellules dans chaque colonne. Trois choix s'offrent à nous :

- à droite (option `r`)
- à gauche (option `l`)
- centré (option `c`)

Dans un second temps, l'une des trois lettres « `l` », `r` » ou `c` » 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 sera terminée par `\\` (c'est un moyen de signifier à LaTeX notre souhait de changer de ligne).

Pour notre tableau simple, voici le code source :

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 bordure

Nous souhaitons maintenant rajouter des bordures visibles. Afin de faire apparaître les lignes, la commande `\hline` est à placer 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 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 « `|` » entre elles lors de la spécification des alignements.

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

Exercice

Il est temps pour vous de **souffrir**. 🐱. Dans cet exercice, vous devez reproduire en LaTeX l'afreux tableau ci-dessous. 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

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
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 ainsi que des colonnes. Les commandes exposées ici ne sont pas faciles à prendre en main et demandent un peu d'entraînement et de gymnastique intellectuelle.

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.

Mise en pratique de 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 lignes

Fusionner deux lignes entre elles nécessite une commande introduite par le package `multirow`. Cette commande s'appelle également `\multirow` et est dépendante de trois paramètres (dont l'un 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 l'image 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}{1cm}{12} & 13 & 14 & 15 & 16 \\
\cline{2-5}
& 17 & 18 & 19 & 20 \\
\hline
21 & 22 & \multirow{2}{*}{23} & 24 & 25 \\
\cline{1-2} \cline{4-5}
26 & 27 & & 28 & 29 \\
\hline
\end{tabular}
```

1	2		3	4
5		6	7	8
9	10	11		
12	13	14	15	16
	17	18	19	20
21	22	23	24	25
26	27		28	29

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,l ». Je vais vous en présenter trois dans ce paragraphe.

La nouvelle instruction possible est :

Code : TeX

```
p{largeur de la colonne en centimètre}
```

A la différence de ses consœurs « r,c,l », cette instruction n'a pas d'influence sur l'alignement horizontal du texte au sein des cellules.

Code : TeX

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

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

Nous apprendrons plus loin dans ce chapitre à utiliser des environnements. Ceci vous permettra de choisir l'alignement du contenu dans des colonnes à largeur fixe.

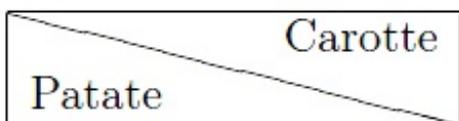
Créer une slashbox

Le package `slashbox` permet d'introduire la commande `\backslashslashbox{Texte dessous}{Texte dessus}`. Cette commande sert à scinder en deux parties triangulaires de même aire une cellule initiale rectangulaire.

Démonstration :

Code : TeX

```
\begin{tabular}{|c|p{1cm}|p{2cm}|}
\hline
\backslashslashbox{Patate}{Carotte} & 1cm & 2cm \\
\hline
\end{tabular}
```

	1cm	2cm
---	-----	-----

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 possible d'utiliser de nouveaux séparateurs via les commandes `!{votre séparateur}` ou `@{votre séparateur}`. Ces commandes sont introduites par le package `array` (comme beaucoup de commandes utiles à la création de tableaux).

La commande `!{ }` présente l'avantage, par rapport à `@{ }`, d'insérer une espace avant et après le séparateur. Visuellement, la différence est flagrante :

Code : TeX

```
\begin{tabular}{|c !{bah} c @{bah}c|}
\hline
texte & texte & texte \\
\hline
\end{tabular}
```

texte bah texte

Exercice : tentez de recréer vous-même le tableau suivant (le symbole euro est disponible via la commande `euro{ }` disponible dans le package `eurosym`) :

bonbon	0,50 €
table	14,99 €

Solution :

Code : TeX

```
\begin{tabular}{|c| c @{,} c !{\euro} |}
\hline
bonbon & 0 & 50 \\
table & 14 & 99 \\
\hline
\end{tabular}
```

Utilisations de commandes et d'environnements dans un tableau

La conception de tableau telle que nous venons de l'apprendre ne permet pas l'insertion de commande 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 :

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

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 : tentez de réaliser le tableau ci-dessous (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

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.



L'environnement `table`, un environnement flottant

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. `Caption`, `label`, `center` ... Rien ne change 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.

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

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 \\
\end{tabular}
\end{center}
\end{table}
```

```

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}

```

Ce chapitre vous a présenté de façon globale les tableaux. Gardez à l'esprit qu'il existe de multiples subtilités dans leur utilisation et que vous trouverez sûrement dans vos recherches futures un package ou une commande répondant à chacune de vos envies.

Le choix éditorial fait dans ce tutoriel est de vous présenter les grands ensembles de LaTeX et toutes les fonctions essentielles à la mise en place rapide de rapports ou de compte-rendus écrits. Nous approfondirons un peu les tableaux lors du tp et une annexe « aller plus loin » sera disponible en annexe pour les plus acharnés !

Retenons de ce chapitre que :

- un tableau nécessite un formalisme et un code propre
- un tableau est créé dans l'environnement `tabular`
- un tableau est flottant si celui-ci est entouré par l'environnement `table`
- tout est possible avec les tableaux. 🤖

Chapitre suivant : Les mathématiques.

C'est un très gros morceau, essentiel en LaTeX.

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 (le domaine de prédilection de LaTeX), des formules et des intégrales.

Résumer l'intégralité des outils permettant d'écrire des maths reviendrait à écrire un pavé de quelques centaines de pages. Nous nous attarderons ici sur les éléments essentiels tout en vous permettant d'aller plus loin si cela est votre souhait.

Il est notable que certains zéros ont un niveau en math ou en physique qui ne leur permet pas de comprendre l'utilité de certaines formes mathématiques (discriminants, matrices, exponentielles...). Ne prenez pas peur, ceci n'est pas un cours de maths mais un cours visant écrire des expressions mathématiques. Il vous suffira de sauter les paragraphes qui ne vous seront pas utiles. 😊

C'est un chapitre long mais qui tout de même possède une certaine cohérence, nous traiterons dans l'ordre :

- la manière de saisir des maths
- les équations simples
- les formalismes d'équation un peu plus violents (type intégrale)
- les systèmes d'équations ainsi que tout ce qui ressemble de près ou de loin à une accolade
- les matrices
- les théorèmes
- les flèches et symboles étirables

Comme je vous le disais plus haut, la liste est longue mais ne comporte pas d'éléments réellement affreux. Les maths ne font que combiner différentes méthodes et syntaxes vues précédemment (avec des noms de commandes et d'environnements différents bien entendu).

Saisir une expression mathématique

Cette première sous partie présente les éléments vous permettant d'insérer au sein d'un texte des expressions mathématiques ou scientifiques. Ces commandes et environnements ont pour effet de passer de la police standard du document (que nous avons appris à modifier dans le chapitre traitant des polices) à la police définie pour les maths (que nous avons elle aussi appris à modifier via les 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 « - » suivant que vous soyez dans le mode texte ou dans le mode écriture scientifique.

Deux possibilités vous sont offertes par LaTeX. Vous pouvez selon vos désirs choisir d'écrire des expressions scientifiques au sein de votre texte ou plutôt à part, dans un espace où il sera plus lisible.

Bien sûr, tout serait trop simple s'il n'existait qu'une méthode, nous allons donc en examiner plusieurs autres au cours 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`.
- Pour écrire des expressions mathématiques en dehors d'un paragraphe, il faudra entourer les expressions par « `\[` » (à l'ouverture) et « `\]` » (à la clôture) ou des « `$$` » ou utiliser l'environnement `equation` (qui est plutôt spécifique aux équations et permet de numéroter ces dernières).
- 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. 😊

Démonstration de toutes ces commandes ci-dessous. Les commandes et environnements sont volontairement noyés dans des paragraphes afin que vous puissiez mieux voir leurs effets.

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.

Ces commandes et environnements sont des incontournables et savent montrer leur utilité dans de nombreux cas. La sous-partie suivante va introduire un grand nombre d'exemples et de formules.

Cela nous change des tableaux et de leur syntaxe affreuse. 😊

Fonctions usuelles

Cette sous partie traite des fonctions assez faciles à écrire. Quelques-unes (tableau 1) sont des commandes avec arguments (mais qui n'utilisent pas forcément le « \ » rituel) alors que dans un second tableau, vous pourrez par exemple constater que cosinus (cos pour les intimes) s'introduit exactement de la même manière qu'un caractère spécial.

Fonctions usuelles à arguments

Dans le premier tableau visible ci-dessous, la liste des fonctions à arguments avec pour chacune, une capture et un exemple.

Nom de la fonction	Code	Exemple
--------------------	------	---------

Fonctions usuelles		
Fraction	<code>\[\frac{numérateur}{denominateur}\]</code>	$\frac{\textit{numérateur}}{\textit{denominateur}}$
Fraction complexe Package amsmath	<code>\[\cfrac{num1}{den1 + \cfrac{num2}{den2}}\]</code>	$\frac{\textit{num1}}{\textit{den1} + \frac{\textit{num2}}{\textit{den2}}}$
Racine	<code>\[\sqrt[15]{nombre}\]</code>	$\sqrt[15]{\textit{nombre}}$
Limite	<code>\[\lim_{nbr1 \to nbr2} x+42\]</code>	$\lim_{\textit{nbr1} \rightarrow \textit{nbr2}} \textit{x} + 42$
Exposant	<code>\[nombre^{exposant}\]</code>	$\textit{nombre}^{\textit{exposant}}$
Indice	<code>\[nombre_{indice}\]</code>	$\textit{nombre}_{\textit{indice}}$

Fonctions usuelles sans argument

Le prochain tableau contient des commandes s'utilisant comme des caractères spéciaux. Par exemple, au lieu d'avoir une hypothétique commande `\cos{argument}` pour cosinus, il y aura le caractère spécial `\cos` suivi de ce que vous voudrez (un exposant, un théta ou autre chose). Petite démonstration ci-dessous et ensuite un gros tableau.

Code : TeX

```
\[\cos\]
\[\cos^{42}(\lambda)\] %Les caractères spéciaux grecs sont dans
l'une des annexes
```

COS

$$\cos^{42}(\lambda)$$

Vous venez de constater qu'utiliser cosinus revient à assembler des commandes élémentaires simples entre elles (comme ça tout est plus facile). 😊

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>

Tout ce tableau ne contient que des commandes permettant d'écrire le texte en romain plutôt qu'en italique mathématique. La fonction normalement utilisée pour faire ça est `\mathrm{votre fonction}` mais pour éviter d'écrire sans arrêt `\mathrm{cos}` des bonnes âmes ont créés des raccourcis de types `\cos`. La commande `\mathrm{fonction}` nous servira à mettre en caractères romains le texte de n'importe quelle fonction.

Les intégrales

Préambule technique

A partir de ce paragraphe, les choses se complexifient un peu (mais pas beaucoup, rassurez-vous). 😊. Vous allez faire côtoyer dans vos expressions des symboles ainsi que des intégrales, des fonctions et pleins de petites choses sympathiques. Le problème de fond est qu'à force de tout mélanger vous allez finir par ne plus vraiment savoir quels packages appeler.

Soyons simples et allons-y comme des brutes ! Nous allons tout simplement appeler les trois packages majeurs utiles à l'écriture d'expression scientifique dès le préambule de notre document et ne plus nous embêter à chercher à quel package nous faisons appel pour tel ou tel symbole (vous vous imaginez une table de 300 symboles avec à coté de chacun le nom du package, ça serait juste ignoble). Ici je les ai mis par ordre alphabétique mais l'ordre n'a strictement aucune importance à notre niveau.

Code : TeX

```
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{mathrsfs}
```

Souvent vous tomberez sur des docs donnant des commandes LaTeX sans les packages qui vont avec. Ces trois-là devraient vous sortir de bon nombre de situations désagréables. 😊

Intégrales et formules complexes

Suite à ce petit préambule, nous allons apprendre à composer des équations d'une nature un peu plus ambitieuses que celles vues précédemment. Nous allons ici voir comment écrire des formules comportant des intégrales (simples, doubles, triples et autres) ainsi que des sommes ou des produits.

Un exemple étant beaucoup plus facile à comprendre qu'un long discours, j'ai dressé pour vous un tableau vous montrant des formules générales ainsi que des exemples. La commande `\` suivie d'une espace est introduite, elle sert à insérer une espace dans un environnement mathématique.

Nom	Modèle	Exemple	Rendu
Fonctions complexes			
Intégrale simple	Code : TeX <pre> \[\int {contenu}\] \[\int_{borne inférieure}^{borne supérieure} {contenu}\] \[\int \limits_{borne inférieure}^{borne supérieure} {contenu}\] </pre>	Code : TeX <pre> \[\int {x^2 dx}\] \[\int_{1}^{3} {x^2 dx}\] \[\int \limits_{1}^{3} {x^2 dx}\] </pre>	$\int x^2 dx$ $\int_1^3 x^2 dx$ $\int_1^3 x^2 dx$
Intégrale curviligne	Code : TeX <pre> \[\oint{Contenu}\] </pre>	Code : TeX <pre> \[\oint{x^2\ dx}\] </pre>	$\oint x^2 dx$
Somme	Code : TeX <pre> \[\sum_{en dessous}^{au dessus} contenu\] </pre>	Code : TeX <pre> \[\sum_{k=2}^{47} k+1\] </pre>	$\sum_{k=2}^{47} k + 1$
Produit	Code : TeX <pre> \[\prod_{en dessous}^{au dessus} contenu\] </pre>	Code : TeX <pre> \[\prod_{k=2}^{47} k+1\] </pre>	$\prod_{k=2}^{47} k + 1$

Les intégrales doubles et triples ne sont que des intégrales à l'intérieur d'autres intégrales. Petites démonstrations ci-dessous, dans 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}}\]
```

$$\int \int x^2 + y \, dx dy$$

$$\int \int_1^3 x^2 + y \, dx dy$$

$$\int \int_1^3 x^2 + y \, dx dy$$

$$\int_0^4 \int_1^3 x^2 + y \, dx dy$$

$$\int_0^4 \int_1^3 x^2 + y \, dx dy$$

$$\int \int \int x^2 + y + z \, dx dy dz$$

$$\int_{x=0}^{x=5} \int_{y=0}^{y=4} \int_{z=1}^{z=3} x^2 + y + z \, dx dy dz$$

$$\int_{x=0}^{x=5} \int_{y=0}^{y=4} \int_{z=1}^{z=3} x^2 + y + z \, dx dy dz$$

Si jamais un jour le besoin se fait sentir d'écrire plusieurs lignes sous un opérateur, la commande `\substack{}` est conçue pour et permet d'écrire plusieurs lignes en dessous d'un opérateur en séparant chaque ligne par les habituels `\\`. Démonstration ci-dessous sur la somme et le produit.

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\\
```

$$\sum_{\substack{k=0 \\ i=0 \\ j=0}}^n i + j + k$$

$$\prod_{\substack{k=0 \\ i=0 \\ j=0}}^n i + j + k$$

Les systèmes d'équations

Mise en place d'un système

Les systèmes d'équations sont simples à mettre en place. Ils nécessitent d'utiliser des environnements ayant des fonctions similaires à celles de `tabular`, mais mettant en forme des formules.

Notre choix va ici se porter sur l'environnement `eqnarray`, qui offre la possibilité de créer via le caractère `&` un système d'équation aligné bien proprement. L'environnement `eqnarray` numérote les équations, mais son homologue `eqnarray*` ne les numérote 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}
```

$$y = x - y + z \quad (1)$$

$$x = y \quad (2)$$

$$z = y \quad (3)$$

$$y = x - y + z$$

$$x = y$$

$$z = y$$

$$y = x - y + z \quad (4)$$

$$x = y \quad (5)$$

$$z = y \quad (6)$$

Première constatation, la numérotation des équations n'est pas interne à un environnement, elle se poursuit dans le document. C'est très pratique pour guider vos lecteurs, et un lecteur heureux est un lecteur qui vous aime. Important n'est ce pas ? 😊

Les délimiteurs

Notre mission suivante est de positionner dans un premier temps des parenthèses autour du système et dans un second temps, de positionner une accolade sur le coté gauche de notre système d'équation. Pour ce faire, nous allons utiliser la syntaxe suivante (nous utilisons ici `array` pour créer l'alignement des 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 le tableau des délimiteurs avec les codes correspondants 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	\uparrow	\downarrow	\downarrow
\updownarrow	\updownarrow	\Uparrow	\Uparrow
\Downarrow	\Downarrow	\Updownarrow	\Updownarrow
\lfloor	\lfloor	\rfloor	\rfloor
\lceil	\lceil	\rceil	\rceil
\langle	\langle	\rangle	\rangle
/	$/$	\backslash	\backslash
	$ $	\	$\ $

Vous avez l'exercice, vous avez les codes des délimiteurs... A vous de jouer ! 😊

La solution :

Code : TeX

```

\[
\left(
\begin{array}{r c l}
y &=& x - y + z \\
x &=& y \\
z &=& y
\end{array}
\right)
\]

```

```


$$\begin{array}{r c l} y & = & x - y + z \\ x & = & y \\ z & = & y \end{array}$$


```

$$\left(\begin{array}{r c l} y & = & x - y + z \\ x & = & y \\ z & = & y \end{array} \right)$$

$$\left\{ \begin{array}{r c l} y & = & x - y + z \\ x & = & y \\ z & = & y \end{array} \right.$$

Les délimiteurs : usage avancé

Sij'écris $\left[\left(\frac{\int \int (x^2+y) dx dy}{1 + \frac{\pi}{2}} \right) \right]$, c'est correct mais plutôt laid vu que les parenthèses n'entourent pas réellement complètement le contenu. Le mieux serait de pouvoir utiliser les délimiteurs comme nous l'avons fait pour les systèmes d'équations.

Ici les délimiteurs fonctionnent de la même manière que précédemment avec les systèmes d'équations. `\left` avant le délimiteur de gauche, `\right` avant le délimiteur de droite, tout en n'oubliant pas que le point symbolise l'absence de délimiteur (donc si vous voulez un délimiteur à droite et pas à gauche vous pouvez très bien écrire `\left (mon contenu \right .)`).

Mise en pratique :

Code : TeX

```

\left[ \left( \frac{\int \limits_{x=0}^{x=5} \int \limits_{y=0}^{y=4} \int \limits_{z=1}^{z=3} \{x^2+y+z\} dx dy dz}{1 + \frac{\pi}{2}} \right) \right]
\left[ \left( \frac{\int \limits_{x=0}^{x=5} \int \limits_{y=0}^{y=4} \int \limits_{z=1}^{z=3} \{x^2+y+z\} dx dy dz}{1 + \frac{\pi}{2}} \right) \right]

```

```

\right) \]
\[ \sqrt{\left( \frac{x}{x+y} \right) \times \left[ \frac{x}{x+y} \right]} = \left| \frac{x}{x+y} \right|
+ y \right] } = \left| \frac{x}{x+y} \right| \]
% Très pratique pour les dérivées partielles !
\[ C_p = \left. \frac{\delta H}{\delta T} \right|_P \]

```

Rendu :

$$\left(\frac{\int_{x=0}^{x=5} \int_{y=0}^{y=4} \int_{z=1}^{z=3} x^2 + y + z \, dx dy dz}{1 + \frac{\Pi}{2}} \right)$$

$$\left(\frac{\int_{x=0}^{x=5} \int_{y=0}^{y=4} \int_{z=1}^{z=3} x^2 + y + z \, dx dy dz}{1 + \frac{\Pi}{2}} \right)$$

$$\sqrt{\left(\frac{x}{x+y} \right) \times \left[\frac{x}{x+y} \right]} = \left| \frac{x}{x+y} \right|$$

$$C_p = \left. \frac{\delta H}{\delta T} \right|_P$$

Les matrices

Vous vous dites sûrement qu'une matrice ou un déterminant se conçoit à l'aide de `array`, eh bien non ! Bien que cela soit possible, des environnements ont été créés spécialement pour vous aider à concevoir des matrices et les mettre en place dans vos documents.

Les syntaxes utilisées ressemblent toujours à celles utilisées pour créer des tableaux, la seule chose qui change est le nom des différents environnements. Une nouvelle fois, un tableau vous résume les différentes options qui vous sont proposées :

Nom de l'environnement	Exemple	Rendu
Les matrices		

matrix	<p>Code : TeX</p> <pre> \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix} </pre>	$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$
pmatrix	<p>Code : TeX</p> <pre> \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} </pre>	$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$
vmatrix	<p>Code : TeX</p> <pre> \begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix} </pre>	$\begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix}$
Vmatrix	<p>Code : TeX</p> <pre> \begin{Vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{Vmatrix} </pre>	$\begin{Vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{Vmatrix}$

bmatrix	<p>Code : TeX</p> <pre> \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix> </pre>	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$
Bmatrix	<p>Code : TeX</p> <pre> \begin{Bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{Bmatrix> </pre>	$\begin{Bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{Bmatrix}$

Petite application pratique de l'environnement `matrix` et de ses pairs, la création de matrices 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}
1 & \cdots & k \\
\vdots & \ddots & \vdots \\
k & \cdots & n
\end{pmatrix>

```

$$\begin{pmatrix} 1 & \cdots & k \\ \vdots & \ddots & \vdots \\ k & \cdots & n \end{pmatrix}$$

Autre petite astuce, la commande `` permettant d'insérer un espace de la longueur du texte contenu dans la commande. Dans le cadre des matrices, c'est très utile pour insérer des blancs et aligner les différents éléments. Cette commande

n'est pas spécifique aux matrices, elle peut servir dans beaucoup d'autres cas.

Code : TeX

```
\[ %sans \phantom{}
\begin{Vmatrix}
1 & 12345 & 3 \\
94 & 5 & -6 \\
7 & 8 & 9
\end{Vmatrix}
\]

\[ %avec \phantom{}
\begin{Vmatrix}
\phantom{9}1 & 12345 & \phantom{-}3 \\
94 & \phantom{1234}5 & -6 \\
\phantom{9}7 & \phantom{1234}8 & \phantom{-}9
\end{Vmatrix}
\]
```

$$\left| \begin{array}{ccc} 1 & 12345 & 3 \\ 94 & 5 & -6 \\ 7 & 8 & 9 \end{array} \right|$$

$$\left| \begin{array}{ccc} 1 & 12345 & 3 \\ 94 & 5 & -6 \\ 7 & 8 & 9 \end{array} \right|$$

Mise en forme type théorème

Tout comme les citations, les théorèmes ainsi que les corollaires et autres petites choses affreuses écrites par des vieux barbus ont leur typographie propre. Nous allons apprendre à la mettre en place (et oui, vos souffrances ne sont pas terminées). 😊

Généralités



Tout d'abord, clarifions une chose. Il existe un abus de langage assez affreux qui consiste à dire « j'écris un théorème » à chaque fois que vous allez utiliser la syntaxe dont je vais vous parler dans les lignes qui suivent. Si vous êtes physicien, personne ne vous en voudra (preuve en est, 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 collègues fourbes. 😊

Revenons à nos moutons ! Nous cherchons à mettre en page différentes choses : des corollaires, des lois, des théorèmes... Nous souhaitons que chacune de ces différentes entités ait une numérotation propre au sein du document (si dans une page vous tapez un théorème et un corollaire, le théorème s'appellera « théorème 1 » et le corollaire s'appellera « corollaire 1 » et non « corollaire 2 »).

Petit plus, nous souhaiterions pouvoir numéroter les différentes entités par rapport aux chapitres, sections et autres éléments structuraux. De sorte que si un théorème est le troisième du chapitre cinq, il puisse porter le nom « 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 demander à LaTeX dans le préambule de créer les différentes entités utilisant la mise en page dédiée au théorème. Dans le texte, nous pourrons ensuite utiliser simplement la commande permettant d'inclure les théorèmes, lois et autres joyeusetés.

C'est parti ! 😊

Pratiquons !

Si vous avez lu les paragraphes précédents, vous avez du garder en tête la méthode en deux temps. La première chose à faire est d'appeler dans le préambule la commande `\newtheorem` (déclinée de plusieurs façons) avant d'utiliser un nom d'environnement choisi par vous-même 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, mais nécessite le package amsthm
\newtheorem{petit_nom1}{Proposition}
\newtheorem{petit_nom2}{Définition}[chapter] %le [chapter] peut par
exemple être remplacé par [section], il permet de numéroter 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, je rajoute 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, je rajoute 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, je rajoute 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, je rajoute 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*

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*

L'exemple est très explicite : la façon dont vous utiliserez la commande `\newtheorem` est importante, mais il n'y a rien d'autre à savoir utiliser. Prenez conscience aussi que vos professeurs vous montreront certaines fonctions pas évidentes en LaTeX via des exemples de code commenté sans même ajouter de texte complémentaire. Si vous prenez le temps de décortiquer le code, la vie sera plus belle pour vous (ou au moins le morceau concernant LaTeX). 😊

Flèches, symboles étirables et espaces

Cette avant-dernière sous partie recense de nombreuses commandes à la fois paramétrables et utiles en LaTeX. Elles vont 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 le 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 fourni en suivant.

Espace	Code
négative	<code>\!</code>
fine	<code>\,</code>
normale	<code>\</code> suivie d'une espace
moyenne	<code>\;</code>
large	<code>\:</code>
cadratin	<code>\quad</code>
double quadratin	<code>\qquad</code>

Code : TeX

```
$$1\!2$$  
$$1\,2$$  
$$1\ 2$$  
$$1\;2$$  
$$1\:2$$  
$$1\quad2$$  
$$1\qquad2$$
```

$$12$$

$$1\ 2$$

$$1\ 2$$

$$1\ 2$$

$$1\ 2$$

$$1\ 2$$

$$1\ 2$$

Les accents et chapeaux divers

Les environnements scientifiques nécessitent parfois de coiffer une lettre ou une série de lettres de flèches (représentant des vecteurs) ou de divers accents (dans le cas de `\stackrel` nous remplacerons l'accent par une série de lettres). Ils sont listés ci-dessous.

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[dessous]{dessus}</code>	$\xrightarrow[dessous]{dessus}$
<code>\xleftarrow[dessous]{dessus}</code>	$\xleftarrow[dessous]{dessus}$
<code>\overset{a}{X}</code>	$\overset{a}{X}$
<code>\underset{b}{X}</code>	$\underset{b}{X}$
<code>\overset{a}{\underset{b}{X}}</code>	$\overset{a}{\underset{b}{X}}$
<code>\overbrace{\sin^2(x) + \cos^2(x)}^{\text{Merveilleux}}=1</code>	$\overbrace{\sin^2(x) + \cos^2(x)}^{\text{Merveilleux}} = 1$
<code>\underbrace{\sin^2(x) + \cos^2(x)}_{\text{Épique}}=1</code>	$\underbrace{\sin^2(x) + \cos^2(x)}_{\text{Épique}} = 1$

Les mathématiques et la typographie des expressions scientifiques méritent à elles seules un ouvrage entier et un tutoriel de plusieurs centaines de pages approfondies. Ces notions représentent ici près d'un quart en volume du tutoriel (en comptant les tables de caractères spéciaux qui viendront se greffer en annexe et les exercices pratiques en TP).

Comme vous avez pu le voir, les syntaxes apprises ici ne sont que des combinaisons de syntaxes vues précédemment. En somme, rien de bien méchant.

Vous avez sûrement aussi remarqué que dans certains exemples je n'utilise pas de \$\$, et pour cause, certains environnements n'en nécessitent pas (et malheureusement pour le savoir il faut faire le test, il n'y a pas vraiment de règle générale).

Gardez tout cela à l'esprit et préparez-vous pour le prochain chapitre : les tables.

Sommaire et index

Les maths sont derrière nous. Il est temps de souffler... 😊

Prenons le temps de faire un bilan : les tableaux, les images, la mise en forme de 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 source de blocage intarissable. 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 apprendrons ici à manier les tables de tableaux, de figures et bien sur 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 la vôtre accessoirement) sera pour le correcteur un gage de sérieux.

Dans de nombreux cas en édition, ce qui est jugé est votre table des matières. Mal la présenter revient à partir avec une balle dans le pied, c'est pourquoi tout projet sérieux nécessite une table des matières réfléchiée 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 et à demander à LaTeX de ne pas considérer les éléments hiérarchiques de trop bas niveaux.

Résumons en trois mots une bonne table des matières :

- cohérence
- pertinence (vis-à-vis du contenu réel de la publication)
- propreté.

Table des matières simple

Avant de commencer à faire des choses très subtiles, nous devons apprendre à insérer une table des matières dans un document. Pour ce faire il vous suffit d'insérer la commande `\tableofcontents` à l'endroit où vous souhaitez insérer une table des matières dans votre document.

Je vous assure, ça n'a rien de plus compliqué. En une ligne de code vous vous retrouverez avec 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 la seconde d'annexer les numéros de pages.

Petite démonstration sur le code suivant :

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 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

La table obtenue apparait après 2 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 « Tables des matières ». Ce nom me semble peu adapté si la table est présente en début de manuscrit. L'appellation « Sommaire » peut paraître plus adaptée.

Voici la commande :

Code : TeX

```
\renewcommand{\contentsname}{Votre nouveau titre} %dans le corps du
document, avant la commande \tableofcontents .
```

Ci-dessous, 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

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 (il vous faudra pour changer l'appellation « Partie » en autre chose 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.

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

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 paragraphes dans la table des matières de la classe `book`. Nous souhaiterions qu'ils le soient.

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, ceci nous donnant un tableau sympathique. 😊

Élément hiérarchique	Nombre
Parties	-1
Chapitres	0

Sections	1
Sous section	2
Sous sous section	3
Paragraphe	4
Sous paragraphe	5

La commande qui nous intéresse est `{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 avoir une table des matières comportant les titres des paragraphes :

Code : TeX

```
\setcounter{tocdepth}{4}
```

Le résultat :

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

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 rajouter une entrée.

La commande est `\addcontentsline{toc}{élément de structure sans son \ ni ses accolades et crochets}{titre apparaissant dans le sommaire}`.

Petit exemple :

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

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 affiche 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 tables des tableaux permettent au lecteur de mieux s'y retrouver dans un texte. Ces deux tables ne sont pas incontournables, mais suivent les mêmes mécanismes à peu de choses près que les tables des matières.

Elles s'insèrent respectivement via les commandes `\listoffigures` et `\listoftables` et font apparaître au choix un

titre spécialement conçu pour la table, soit la légende contenue dans `\caption`. Nous n'avons pas encore appris à mettre de titres conçus pour les tables dans la commande `\caption`, pourtant cela est très simple. Ci-dessous vous sont présentées les deux syntaxes possibles.

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 2 fois. Je suis certain qu'il y en a un qui va l'oublier et poser la question en commentaire. 😊

Je vous ai fabriqué un exemple massif 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 vraiment pas méchant mais très utile. J'ai réutilisé ici l'image de Poulpy, mascotte officielle du tutorial. 😊

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

1	Poulpy est multicolore	1
2	Chatoyante	2
3	Poulpy est inestimable	2
4	Poulpesque	3

Les index

A mon sens, les index sont la limite en difficulté de ce tutoriel et n'ont finalement pas un intérêt énorme dans des rapports ou des petites publications. J'ai tout de même choisi de vous en parler un peu, ne serait-ce que par souhait de faire une ouverture.

Prenez conscience tout de même que je vous expose ici la méthode permettant d'élaborer des index élémentaires mais que les choses peuvent extrêmement vite devenir très difficiles à mettre en place (la syntaxe de certains index me rappelle un peu celle des regex en php en terme de difficulté).

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 (pas mal de documentations n'abordent même pas le sujet).

Théorie

Commençons par les choses simples. 😊. La création d'un index nécessite d'utiliser le package `makeidx`, d'insérer la commande `\makeindex` dans le préambule et `\printindex` à l'endroit où vous souhaitez insérer 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 vous donnant la position du marqueur « patate » dans votre document.

Les utilisateurs de TeXnicCenter n'auront besoin que de compiler 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_de_fichier.tex` dans leur console (ou cliquer sur Build>Compile>Makeindex dans Kile) et recompiler de nouveau.



Les utilisateurs de TeXShop trouveront quant à eux la commande `Makeindex` dans l'un de leur menu déroulant et devront eux aussi compiler normalement une fois, sélectionner `Makeindex` et compiler dans ce mode, avant de compiler à nouveau en mode LaTeX.

Si nous faisons le test de mettre un marqueur « paté » et un marqueur « patate » LaTeX placerait dans l'ordre alphabétique « paté » avant « patate ». Difficile à croire et pourtant...

Code : TeX

```
J'aime le paté\index{paté} et les patates\index{patate}
```

Index

paté, 1
patate, 1

L'idée derrière tout ça c'est que LaTeX a un peu (beaucoup) de mal avec les accents et il faut lui mettre en argument une version non accentuée de chaque mot accentué (c'est aussi valable pour les mots avec majuscules). Ces deux versions devront être séparées par un @. Oui, je sais, c'est tordu. 😊

Code : TeX

```
J'aime le paté\index{pate@paté} et les patates\index{patate@Patate}
```

Index

Patate, 1
paté, 1

C'est déjà plus joli.

Si vous souhaitez créer des sous-entrées dans l'index, il vous suffit de séparer les entrées et sous entrées par des « ! ». L'exemple est très parlant ici. Notez que créer des sous-sous-entrées est possible.

Code : TeX

```
J'aime le paté\index{pate@paté} et les patates\index{patate}, ainsi  
que les patates nouvelles\index{patate!patates nouvelles}.
```

Index

patate, 1
 patates nouvelles, 1
paté, 1

Pratique

Votre mission si vous l'acceptez est de créer l'index ci-dessous 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
 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.

Les tables et Index ont donné des sueurs froides à bien des étudiants et vous y avez survécu. Félicitations. 😊. Il n'y a pas de grande conclusion à faire ici, je vous conseillerai juste de rester méthodique et de garder le cours sous le coude lors de vos premières compositions.

Ne faites pas les tables et index en premier, gardez-les plutôt pour la fin lors de la création d'un document. Il est toujours plus facile d'y voir plus clair dans la réalisation et l'organisation d'un document une fois qu'il est réellement écrit.

Le prochain chapitre vous apprendra à utiliser les bibliographies.

La Bibliographie

Chers zéros, vous qui avez suivi avec attention jusque là, je dois vous annoncer que vous arrivez au dernier chapitre de cours du tutoriel.

Je sais, c'est triste. 😞

Pour finir en beauté, nous allons apprendre l'une des notions qui donnent des boutons à bien des débutants en LaTeX : la bibliographie. Cet élément, qui est créé malgré tout pour être maniable, n'est pas souvent bien compris par les novices et passé sous silence.

Conséquence logique du problème : les novices n'utilisent pas de bibliographie dans leurs publications et ne citent pas leurs sources et leurs ressources documentaires. Sachez une chose, en science ce genre d'oubli est grave. En journalisme, je ne vous en parle même pas, ça peut être un bon motif pour prendre la porte pour incompétence notoire. 😊

Ce chapitre est consacré entièrement à l'une des méthodes permettant de créer des bibliographies. Elle vous sera développée, expliquée sous toutes ses coutures.

Après ce chapitre, nous entamerons l'ultime épreuve du TP 2 qui fera de vous un Jedi. 😊

Principe d'une bibliographie

Avant de parler 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, ceux pour qui une bibliographie est un rassemblement d'œuvres qui répond à un besoin particulier. Je vous cite ci-dessous un petit morceau de la définition du mot bibliographie sur Wikipédia, qui vous permettra de vous faire une idée des différents types de bibliographie existantes.

Citation : Wikipédia : bibliographie

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.

Arrivé ici, normalement, les choses doivent vous sembler un peu tordues. Vous pensiez savoir ce qu'est une bibliographie et au final vous vous rendez compte qu'elles peuvent prendre de multiples formes. Néanmoins ces multiples formes ont un point commun : **mettre en forme une liste d'œuvres**.

Restons sur cette idée là : « Une bibliographie est un listing d'un certain nombre d'œuvres choisies, répondant toutes à une consigne (regrouper les sources, regrouper toutes des œuvres thématiques...), mise en page selon les souhaits d'un (ou de plusieurs) auteur(s) ».

Tout de suite, nous y voyons déjà un peu plus clair. 😊

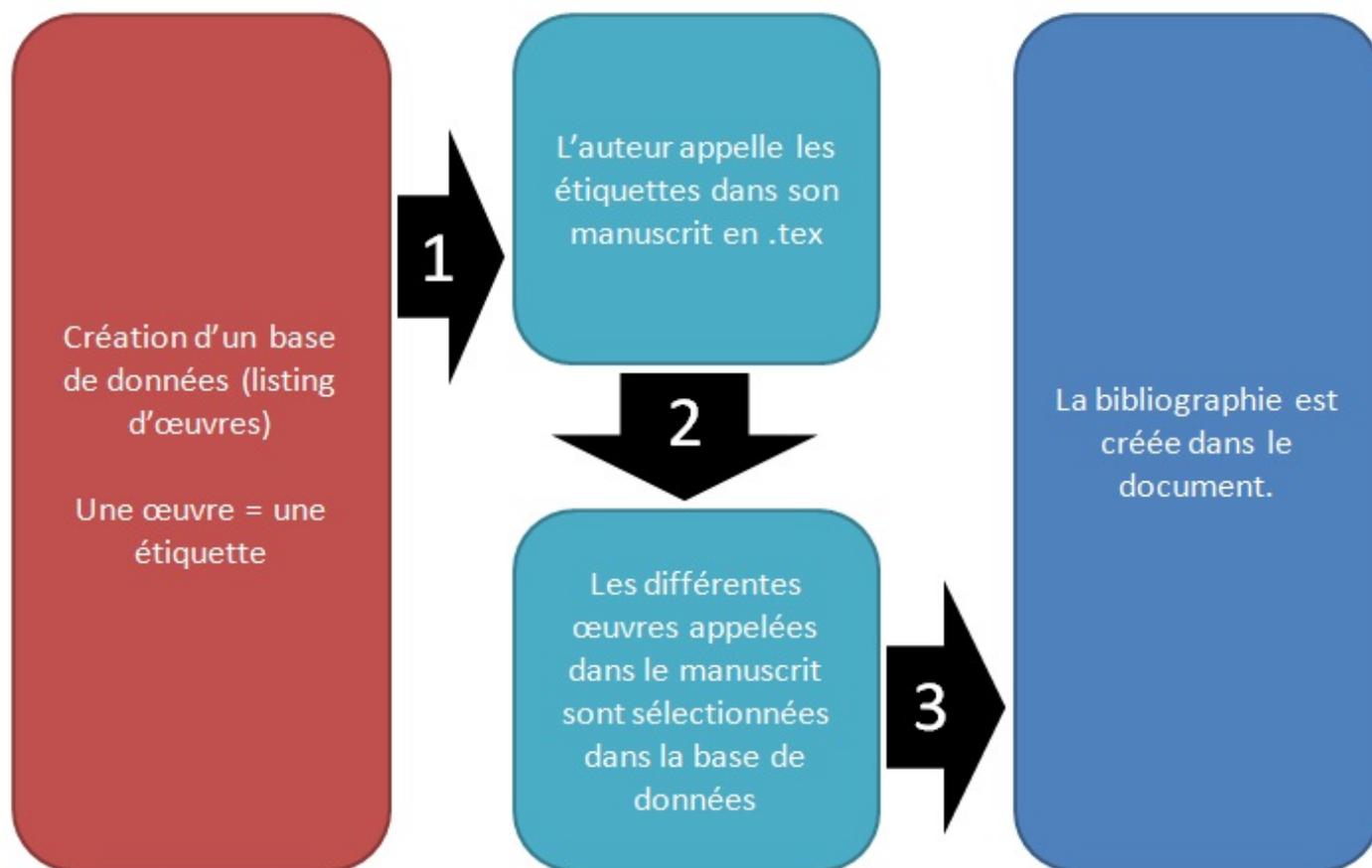
La démarche de l'auteur est donc de choisir dans la longue liste des œuvres qu'il connaît celles qui seront publiées dans sa bibliographie. Ceci, LaTeX sait très bien le faire, et le listing se crée grâce à un programme externe : bibtex.



Bibtex fournit une armada d'outils à l'auteur dans le but de concevoir facilement son listing (que nous appellerons base de données) contenant toutes les informations sur les œuvres qu'il connaît (auteur, date d'écriture...).

L'auteur donnera une petite étiquette à chaque œuvre, et il lui suffira par la suite dans son document LaTeX de faire référence à certaines étiquettes pour intégrer aussitôt les œuvres concernées dans la bibliographie de sa publication.

Nous allons résumer tout ceci dans un schéma. 😊



L'idée générale étant maintenant fixée, nous allons maintenant rentrer 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 afin de créer la bibliographie.

La base de données

Créer la base de données est la partie la plus facile dans une bibliographie. La base se constitue de paragraphes contenant chacun les informations et l'étiquette des œuvres.



Le fichier que nous enregistrerons ici et qui contiendra la base de données ne sera pas un .tex mais un .bib

La structure des informations données à propos d'une œuvre dans la base de données est toujours exactement 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. Certaines règles sont néanmoins à respecter afin 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, il y en a des prédéfinis, vous ne pouvez pas en inventer.

Voilà ce qui rebute les débutants, le formalisme un peu tristounet. 😊. En réalité, par rapport à la mise en forme d'un tableau, c'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 et les titres étaient prédéfinis, il est temps pour vous d'utiliser leurs deux tableaux récapitulatifs.

Type d'oeuvre		
Type d'oeuvre	Commentaires	Champs
@article	Article	author journal title year month pages notes
@book	Livre	author title publisher year
@manual	Document technique	title author year organization
@misc (utilisé pour les sites web et ce qui ne trouve pas sa place ailleurs)	Divers	author title month year note
@phdthesis	Thèse de doctorat	title author school year
@unpublished	Manuscrit non publié	author title note year month

Il en existe d'autres bien sûr, 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



Utiliser le package **url** pour la saisie d'adresse permet d'afficher les URL correctement dans la bibliographie et de s'affranchir des erreurs de compilation lorsque qu'une URL contient un caractère tel que &, #, %, etc... (ajout d'un membre fort pertinent ! Grand merci à lui)

Nous pouvons en utilisant les différents éléments ci-dessus créer notre première petite base de donnée 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={sympatique}
}
```

```
@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 étant d'y regrouper les œuvres que vous utilisez ainsi que 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 aller piocher, d'autres en créent un par publication, c'est selon les goûts de chacun. De mon côté j'aime assez l'idée de pouvoir avoir en un seul endroit toute la liste des ressources utilisées dans toutes mes créations (en partant du principe que le geek a tendance à être un peu désordonné, commencer à mettre des fichiers .bib partout c'est un peu du suicide). 🤪

Concept

Apprenons maintenant à nous servir de cette base de données. Nous allons devoir faire des références à des éléments de la base de données via la commande `\cite{référence_courte}` que nous placerons où nous le voudrions dans le texte. Deux autres commandes, `\bibliographystyle{ }` et `\bibliography{nom du fichier.bib sans l'extension .bib}` servent respectivement à donner à LaTeX le style de mise en page souhaitée pour la bibliographie ainsi que la localisation 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 document utilisées. Cette page se trouvera à l'emplacement où vous aurez choisi de faire apparaître les commandes `\bibliographystyle{ }` et `\bibliography{ }` (le fonctionnement est similaire à celui des tables).

Exemples

Beaucoup d'informations vous ont été données, je vous montre quelques exemples, illustrant les différents styles de bibliographies possibles, avant de vous expliquer plus en détail comment se passe la compilation suivant que vous soyez sous Mac, 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 crochets.
\bibliography{bibli} % mon fichier de base de donnée s'appelle
bibli.bib

\end{document}

```

`\bibliographystyle{plain}` classe les entrées par ordre alphabétique et les numérote en conséquence.

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{abbrv}` classe les entrées par ordre alphabétique, les numérote en conséquence et abrège certains éléments de la bibliographie.

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{unsrt}` trie les entrées par ordre d'apparition dans le texte.

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{alpha}` le repère n'est plus un chiffre mais les trois premières lettres du nom de l'auteur ainsi que les deux derniers chiffres de l'année de parution.

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.

Compilation

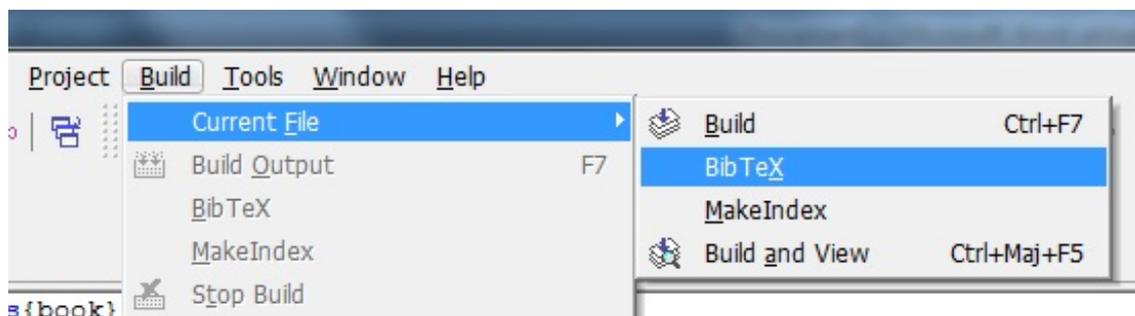
La compilation d'un document comportant une bibliographie se passe d'une manière légèrement inhabituelle, détaillons ici les trois manières d'arriver à bout de celle-ci.

Gardons à l'esprit que nous souhaitons tout d'abord compiler notre document LaTeX (pour obtenir la liste des références à aller rechercher dans la base de données) avant d'appeler BibTeX (pour piocher dans la base de données) et de recompiler une nouvelle fois notre document LaTeX.

Dans la pratique, des cas exotiques peuvent arriver (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 et de recompiler deux nouvelles fois.



Le processus est un peu fastidieux mais une fois maîtrisé il ne pose pas de problème, c'est une nouvelle routine à intégrer.

Sous Linux

Pour vous, nous allons nous y prendre à la console, les choses seront plus simples ainsi. 😊

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

Les utilisateurs sous mac devront dans un premier temps compiler deux fois comme à leur habitude, avant de sélectionner dans l'un de leurs menus déroulants l'option BibTeX au lieu de LaTeX et de compiler. Re-sélectionnez LaTeX dans votre menu déroulant, compilez deux nouvelles fois et tout devrait aller. 😊



La compilation, le côté très laborieux des bases de données et le formalisme inhabituel sont autant de caractéristiques qui peuvent vous faire frémir de peur au début mais qui bien vite vont devenir des réflexes.

Il n'y a pas vraiment de conclusion technique ici. Restez juste conscient de l'importance de citer vos sources dans vos

publications. Il en ira souvent de votre réputation et de votre crédibilité si vous fondez des propos sur des références fausses ou absentes (à mes débuts en tant que blogueurs ça m'est arrivé et c'est très désagréable). 🤔

Ce chapitre est le dernier du cours, nous allons maintenant pouvoir nous lancer dans le second TP !

TP 2 : Rédaction d'un article présentant le théorème de Pythagore

Vous voici arrivés au dernier second et dernier TP du cours. Je l'ai composé de façon à ce qu'une majeure partie des notions abordées précédemment soient revues et que vous puissiez vous rendre compte d'éventuelles lacunes.

Nous utiliserons ici un article de wikipédia traitant du théorème de Pythagore ainsi qu'une table de multiplication et une table d'addition (qui peuvent aussi être retrouvées sur wikipédia).

Le livre final issu de ce TP fait 23 pages (dont 9 blanches) et il n'y a pas vraiment de pièges fourbes ici. J'ai préféré vous faire utiliser les commandes usuelles, à vous de voir si vous voulez rajouter des fioritures ou faire des essais. 😊

La sous-partie d'ouverture sera riche de nombreuses nouvelles idées et commandes vous permettant d'améliorer vos documents et votre maîtrise de LaTeX.

Les consignes

J'ai choisi pour cet exercice de ne plus vous tenir la main, il vous faudra tenter de réussir à suivre les consignes données, recréer une structure et la remplir avec le contenu donné, 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 majuscules** et la **date du jour**
- un **sommaire**
- un **chapitre d'introduction** (nommé « **Introduction** »)
- une **première partie** nommée « Théorème de Pythagore » contenant un **chapitre** nommé « Énoncé du théorème » et un **chapitre** « Réciproque »
- le **chapitre** « Énoncé du théorème » portera en son sein deux **sections** nommées respectivement « **Théorie** » et « **Exemple** »
- une **seconde partie** nommée « 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 balises de structure caractéristiques des livres selon vos souhaits. Essayez de rester le plus pertinent possible dans vos choix. 😊

Les contenus

La bibliographie

Une bibliographie sera utile dans ce document, je vous montre ci-dessous son allure. A vous de créer la base de données correspondante (je n'ai rien mis de difficile dedans).

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.

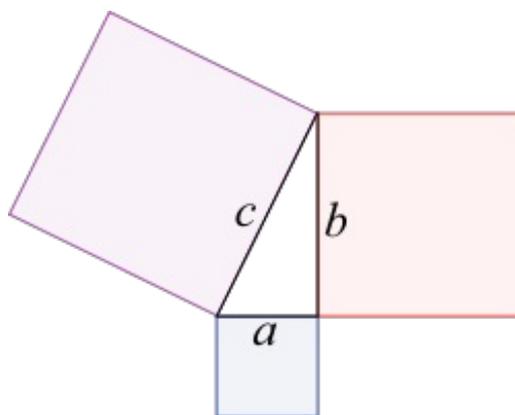
Chapitre d'introduction

Le chapitre d'introduction comporte un paragraphe et une image (dont la légende sera « Une version géométrique du théorème »

Le texte :

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.

L'image :



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. Les mathématiques sont omniprésentes et un flottant fait son apparition. Bien qu'après les paragraphes de texte, ce flottant est inséré dans la section « Théorie ».

Le texte est donné sur l'article "Théorème de Pythagore" de wikipédia et la mise en page doit respecter les captures ci-dessous (et oui, je ne vous donne pas l'image, vous allez devoir la prendre sur la page du wiki et la convertir en PNG). 🤖

Chapitre 1

Énoncé du théorème

1.1 Théorie

La forme la plus connue du théorème de Pythagore [8] 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 tenté sinon 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, 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.

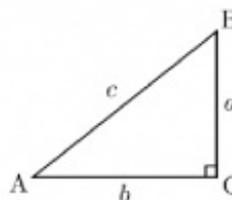


FIGURE 1.1 – Triangle rectangle

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.

3

Chapitre « Réciproque »

Si vous avez réussi l'épreuve du chapitre « Énoncé du théorème » il ne vous posera aucun problème.

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.

5

Tables d'addition et de multiplication

Ces deux annexes nécessitent la création de tableaux, l'utilisation d'une couleur en niveau 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

Au boulot (vous devriez avoir besoin d'une heure normalement) ! 🐼

Petite aide

Vous avez eu toutes les informations nécessaires à la création du document néanmoins je me doute que certains d'entre vous n'arrivent pas vraiment à visualiser le travail dans sa globalité. Si vous avez essayé (et seulement si vous avez essayé) de faire l'exercice mais que vous vous sentez trop perdu, vous pouvez consulter le document en pdf fourni [ici](#).

La correction

Compiler souvent, reprendre le cours et relire 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 lors du tp1).

Les concepts avancés de LaTeX s'entremêlent tous lors de la création d'un document, de sorte que vous pouvez très bien faire votre bibliographie pendant ou après l'écriture de votre contenu. L'important reste 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}` ...)
- compilation et correction des erreurs
- page de garde et listing des `\usepackage{}`
- compilation et correction des erreurs
- création des éléments de structure internes (parties, chapitres, section)
- 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 : 2 compilations, appel de bibtex puis 2 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 s'appliquer à la quasi-totalité des projets de publication (des étapes peuvent venir se greffer ou s'enlever) et suivant votre efficacité vous pouvez 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 sur des documents complexes.

Les deux seules difficultés présentes dans l'exercice sont la présence d'une bibliographie à remplir avec des pages web (donc des éléments `@misc`) et les tableaux complexes (qui ont déjà été vus dans le chapitre sur les tableaux). Je vous livre ci-dessous les deux codes sources, celui de bibliographie.bib ainsi que celui de tp2.tex. Prenez le temps de les lire si jamais vous avez bloqué et retentez, 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} & \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}}cccccccc}

\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} & \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 & \\

\end{tabular}
\end{center}
\end{table}

```

```

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 issu de ce cours 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. . . C'est 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.

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

Création des différents fichiers. 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 `partie2`)) contiennent les codes ci-dessous.

`chapitre1.tex` :

Code : TeX

```
\chapter{Le premier}
```

`chapitre2.tex` :

Code : TeX

```
\chapter{Le second}
Poulpy est une déesse.
```

`chapitre3.tex` :

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. Au cours de la compilation, votre fichier va ressembler à ceci :

Code : TeX

```
\documentclass{book}
\begin{document}
% Début du contenu de chapitre1.tex
\chapter{Le premier}
% Fin du contenu de chapitre1.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.

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, le livre issu de ce cours 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 !

Trêve 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.

- `\hspace{longueur}` insère un espace horizontal de la longueur choisie.
- `\hspace*{longueur}` : idem, sauf que l'espace n'est pas insérée 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.

Résumons l'ensemble des points vus lors de ce TP !

- Une approche méthodique aide à venir à bout de gros projets LaTeX.
- 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.
- LaTeX convertit automatiquement les unités de grandeur utilisées, vous avez donc l'embarras du choix.
- Citer vos sources dans une bibliographie est un gage de sérieux.
- Il est possible de séparer un document LaTeX en plusieurs fichiers, et donc de faciliter un travail organisé et collaboratif.

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 en annexes pour que vous puissiez les garder sous la main à tout moment.

La majorité d'entre elles n'ont pas été vu dans le cours néanmoins les introduire dans un document ne pose aucun problème particulier. J'ai préféré dans cette version web vous donner des tableaux et leurs codes sources correspondants juste en dessous (sinon j'aurais été obligé de faire plusieurs centaines de mini-captures d'écrans et cela aurait été aussi abrutissant qu'inutile).

Chaque ligne comporte 5 symboles, et les packages utilisés ici ont été les suivants (rien ne change avec l'utf8) :

Code : TeX

```
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[français]{babel}
\usepackage{lmodern}
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{mathrsfs}
```

S'il manque des caractères spéciaux, n'hésitez pas à m'envoyer un mp, je les rajouterai.

Lettres grecques (Mode mathématique)

<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				

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}
```

```

ligne 6 & \Lambda & \Theta & \Gamma & \Delta & \Xi \\
ligne 7 & \Pi & \Sigma & \Upsilon & \Phi & \Psi \\
ligne 8 & \Omega & & & & \\
\end{array}
$$

```

Opérateurs extensibles (Mode mathématique)

<i>ligne1</i>	Σ	\int	\uplus	\oplus	\vee
<i>ligne2</i>	\prod	\oint	\cap	\otimes	\wedge
<i>ligne3</i>	\amalg	\iint	\cup	\odot	\sqcup

Code : TeX

```

$$
\begin{array}{c c c c c c}
ligne 1 & \sum & \int & \biguplus & \bigoplus & \bigvee \\
ligne 2 & \prod & \oint & \bigcap & \bigotimes & \bigwedge \\
ligne 3 & \amalg & \iint & \bigcup & \bigodot & \bigsqcup \\
\end{array}
$$

```

Fonctions (Mode mathématique)

<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			

Code : TeX

```

$$
\begin{array}{c c c c c c}
ligne 1 & \arccos & \cos & \csc & \exp & \ker \\
ligne 2 & \limsup & \min & \sinh & \arcsin & \cosh \\
ligne 3 & \deg & \gcd & \lg & \ln & \Pr \\
ligne 4 & \sup & \arctan & \cot & \det & \hom \\
ligne 5 & \lim & \log & \sec & \tan & \arg \\
ligne 6 & \coth & \dim & \inf & \liminf & \max \\
ligne 7 & \sin & \tanh & & & \\
\end{array}
$$

```

Flèches (Mode mathématique)

<i>ligne1</i>	\leftarrow	\longleftarrow	\uparrow	\Leftarrow	\Lleftarrow
<i>ligne2</i>	\Uparrow	\rightarrow	\longrightarrow	\downarrow	\Rightarrow
<i>ligne3</i>	\Longrightarrow	\Downarrow	\leftrightarrow	\longleftrightarrow	\Updownarrow
<i>ligne4</i>	\Leftrightarrow	\Longleftrightarrow	\Updownarrow	\mapsto	\longmapsto
<i>ligne5</i>	\nearrow	\hookrightarrow	\hookrightarrow	\searrow	\lrcorner
<i>ligne6</i>	\rightharpoonup	\swarrow	\lrcorner	\rightharpoonright	\nearrow
<i>ligne7</i>	\rightleftharpoons	\rightsquigarrow	\dashrightarrow	\dashleftarrow	\rightleftarrows
<i>ligne8</i>	\Leftrightarrow	\Lleftarrow	\Leftarrow	\leftleftarrows	\Lleftarrow
<i>ligne9</i>	\Leftrightarrow	\curvearrowleft	\circlearrowleft	\curvearrowright	\Uparrow
<i>ligne10</i>	\upharpoonleft	\downharpoonleft	\multimap	\leftrightsquigarrow	\rightleftarrows
<i>ligne11</i>	\leftrightarrow	\Rightarrow	\Leftrightarrow	\rightarrow	\rightharpoonright
<i>ligne12</i>	\curvearrowright	\rightleftharpoons	\curvearrowright	\circlearrowright	\rightharpoonup
<i>ligne13</i>	\Downarrow	\lrcorner	\lrcorner	\rightsquigarrow	\leftrightsquigarrow
<i>ligne14</i>	\rightarrow	\Lleftarrow	\Lleftarrow	\leftrightarrow	\Lleftarrow

Code : TeX

```


$$\begin{array}{c c c c c c}
\text{ligne 1} & \leftarrow & \longleftarrow & \uparrow & \Leftarrow & \Lleftarrow \\
\text{ligne 2} & \Uparrow & \rightarrow & \longrightarrow & \downarrow & \Rightarrow \\
\text{ligne 3} & \Longrightarrow & \Downarrow & \leftrightarrow & \longleftrightarrow & \Updownarrow \\
\text{ligne 4} & \Leftrightarrow & \Longleftrightarrow & \Updownarrow & \mapsto & \longmapsto \\
\text{ligne 5} & \nearrow & \hookrightarrow & \hookrightarrow & \searrow & \lrcorner \\
\text{ligne 6} & \rightharpoonup & \swarrow & \lrcorner & \rightharpoonright & \nearrow \\
\text{ligne 7} & \rightleftharpoons & \rightsquigarrow & \dashrightarrow & \dashleftarrow & \rightleftarrows \\
\text{ligne 8} & \Leftrightarrow & \Lleftarrow & \Leftarrow & \leftleftarrows & \Lleftarrow \\
\text{ligne 9} & \Leftrightarrow & \curvearrowleft & \circlearrowleft & \curvearrowright & \Uparrow \\
\text{ligne 10} & \upharpoonleft & \downharpoonleft & \multimap & \leftrightsquigarrow & \rightleftarrows \\
\text{ligne 11} & \leftrightarrow & \Rightarrow & \Leftrightarrow & \rightarrow & \rightharpoonright \\
\text{ligne 12} & \curvearrowright & \rightleftharpoons & \curvearrowright & \circlearrowright & \rightharpoonup \\
\text{ligne 13} & \Downarrow & \lrcorner & \lrcorner & \rightsquigarrow & \leftrightsquigarrow \\
\text{ligne 14} & \rightarrow & \Lleftarrow & \Lleftarrow & \leftrightarrow & \Lleftarrow
\end{array}$$


```

```

ligne 12 & \looparrowright & \rightleftharpoons & \curvearrowright &
\circlearrowright & \Rsh \\
ligne 13 & \downdownarrows & \upharpoonright & \downharpoonright &
\rightsquigarrow & \nleftarrow \\
ligne 14 & \nrightharpoonup & \nLeftarrow & \nrightarrow &
\leftrightharpoonup & \Leftrightarrow
\end{array}
$$

```

Accents (Mode mathématique)

<i>ligne1</i>	á	ā	á	ā	ǎ
<i>ligne2</i>	ǎ	ǎ	ǎ	ä	ä
<i>ligne3</i>	ä	ä	à	â	à
<i>ligne4</i>	â	ã	→	ã	→

Code : TeX

```

$$
\begin{array}{c c c c c c}
ligne 1 & \acute{a} & \bar{a} & \acute{a} & \bar{a} & \check{a} \\
ligne 2 & \check{a} & \check{a} & \check{a} & \ddot{a} & \ddot{a} \\
ligne 3 & \ddot{a} & \dot{a} & \grave{a} & \hat{a} & \grave{a} \\
ligne 4 & \hat{a} & \tilde{a} & \vec{a} & \tilde{a} & \vec{a} \\
\end{array}
$$

```

Opérateurs (Mode mathématique)

<i>ligne1</i>	*	±	∩	◁	★
<i>ligne2</i>	∓	∪	▷	·	∏
<i>ligne3</i>	⊕	◁	○	⊙	∏
<i>ligne4</i>	▷	●	⊖	⊔	⊑
<i>ligne5</i>	◯	⊕	∧	⊔	◇
<i>ligne6</i>	⊘	∨	▽	×	⊗
<i>ligne7</i>	†	△	÷	∫	‡
<i>ligne8</i>	\	▪	□	∧	∇
<i>ligne9</i>	⊛	⊞	∩	∫	⊙
<i>ligne10</i>	⊞	∩	∪	⊖	⊠

Code : TeX

```

$$
\begin{array}{c c c c c c}
ligne 1 & \ast & \pm & \cap & \lhd & \star \\
ligne 2 & \mp & \cup & \rhd & \cdot & \amalg \\
ligne 3 & \uplus & \triangleleft & \circ & \odot & \sqcap \\
ligne 4 & \triangleright & \bullet & \ominus & \sqcup & \unlhd \\
ligne 5 & \bigcirc & \oplus & \wedge & \unrhd & \diamond \\
ligne 6 & \oslash & \vee & \bigtriangledown & \times & \otimes \\
ligne 7 & \dagger & \bigtriangleup & \div & \int & \ddagger \\
ligne 8 & \backslash & \cdot & \square & \bar{\wedge} & \nabla \\
ligne 9 & \circledast & \boxplus & \curlywedge & \curlyvee & \circledcirc \\
ligne 10 & \boxminus & \Cap & \Cup & \circleddash & \boxtimes \\
\end{array}
$$

```

<i>ligne1</i>	\perp	\top	$\dot{+}$	\boxdot	\intercal
<i>ligne2</i>	\sphericalangle	\ast	\square	$\overline{\wedge}$	\sphericalangle
<i>ligne3</i>	\equiv	\leq	\geq	\perp	\cong
<i>ligne4</i>	\prec	\succ	\mid	\neq	\preceq
<i>ligne5</i>	\succ	\parallel	\sim	\ll	\gg
<i>ligne6</i>	\bowtie	\subseteq	\subset	\supset	\supseteq
<i>ligne7</i>	\approx	\supseteq	\supseteq	\bowtie	\asymp
<i>ligne8</i>	\sqsubset	\sqsupset	\rtimes	\doteq	\sqsubset
<i>ligne9</i>	\sqsupseteq	\smile	\propto	\dashv	\vdash
<i>ligne10</i>	\frown	\models	\in	\ni	\notin

Code : TeX

```


$$\begin{array}{c c c c c c}
\text{ligne 1} & \& \backslash\text{bot} & \& \backslash\text{top} & \& \backslash\text{dotplus} & \& \backslash\text{boxdot} & \& \backslash\text{intercal} & \backslash\backslash \\
\text{ligne 2} & \& \backslash\text{rightthreetimes} & \& \backslash\text{divideontimes} & \& \backslash\text{square} & \& \backslash\text{doublebarwedge} & \& \backslash\text{leftthreetimes} & \backslash\backslash \\
\text{ligne 3} & \& \backslash\text{equiv} & \& \backslash\text{leq} & \& \backslash\text{geq} & \& \backslash\text{perp} & \& \backslash\text{cong} & \backslash\backslash \\
\text{ligne 4} & \& \backslash\text{prec} & \& \backslash\text{succ} & \& \backslash\text{mid} & \& \backslash\text{neq} & \& \backslash\text{preceq} & \backslash\backslash \\
\text{ligne 5} & \& \backslash\text{succeq} & \& \backslash\text{parallel} & \& \backslash\text{sim} & \& \backslash\text{ll} & \& \backslash\text{gg} & \backslash\backslash \\
\text{ligne 6} & \& \backslash\text{bowtie} & \& \backslash\text{simeq} & \& \backslash\text{subset} & \& \backslash\text{supset} & \& \backslash\text{Join} & \backslash\backslash \\
\text{ligne 7} & \& \backslash\text{approx} & \& \backslash\text{subseteq} & \& \backslash\text{supseteq} & \& \backslash\text{times} & \& \backslash\text{asympt} & \backslash\backslash \\
\text{ligne 8} & \& \backslash\text{sqsubset} & \& \backslash\text{sqsupset} & \& \backslash\text{rtimes} & \& \backslash\text{doteq} & \& \backslash\text{sqsubse} & \backslash\backslash \\
\text{ligne 9} & \& \backslash\text{sqsupseteq} & \& \backslash\text{smile} & \& \backslash\text{propto} & \& \backslash\text{dashv} & \& \backslash\text{vdash} & \backslash\backslash \\
\text{ligne 10} & \& \backslash\text{frown} & \& \backslash\text{models} & \& \backslash\text{in} & \& \backslash\text{ni} & \& \backslash\text{notin} & \backslash\backslash \\
\end{array}$$


```

<i>ligne1</i>	\approx	\leq	\geq	\lessgtr	\thicksim
<i>ligne2</i>	\lessslant	\geqslant	\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	\supseteqq	\smallfrown	\risingdotseq
<i>ligne10</i>	\sqsubset	\sqsupset	\smallsmile	\varpropto	\preccurlyeq

Code : TeX

```


$$\begin{array}{c c c c c c}
\text{ligne 1} & \approx & \leq & \geq & \lessgtr & \thicksim \\
\text{ligne 2} & \lessslant & \geqslant & \lesseqgtr & \backsim & \lessapprox \\
\text{ligne 3} & \gtrapprox & \lesseqgtr & \backsimeq & \lll & \ggg \\
\text{ligne 4} & \gtreqless & \triangleq & \lessdot & \gtrdot & \gtreqless \\
\text{ligne 5} & \circeq & \lesssim & \gtrsim & \gtrless & \bumpeq \\
\text{ligne 6} & \eqslantless & \eqslantgtr & \backepsilon & \Bumpeq & \precsim \\
\text{ligne 7} & \succsim & \between & \doteqdot & \precapprox & \succapprox \\
\text{ligne 8} & \pitchfork & \thickapprox & \Subset & \Supset & \shortmid \\
\text{ligne 9} & \fallingdotseq & \subteqq & \supseteqq & \smallfrown & \risingdotseq \\
\text{ligne 10} & \sqsubset & \sqsupset & \smallsmile & \varpropto & \preccurlyeq
\end{array}$$


```

<i>ligne1</i>	\succcurlyeq	\Vdash	\therefore	\curlyeqprec	\curlyeqsucc
<i>ligne2</i>	\vDash	\because	\blacktriangleleft	\blacktriangleright	\Vdash
<i>ligne3</i>	\Hbar	\triangleleft	\triangleright	$=$	\neq
<i>ligne4</i>	\triangleleft	\triangleright	\neq	\neq	\neq
<i>ligne5</i>	\neq	\neq	\neq	\neq	\neq
<i>ligne6</i>	\neq	\neq	\neq	\neq	\neq
<i>ligne7</i>	\neq	\neq	\neq	\neq	\neq
<i>ligne8</i>	\neq	\neq	\neq	\neq	\neq
<i>ligne9</i>	\neq	\neq	\neq	\neq	\neq
<i>ligne10</i>	\neq	\neq	\neq	\neq	\neq
<i>ligne11</i>	\neq	\neq	\neq	\neq	\neq
<i>ligne12</i>	\neq	\neq	\neq	\neq	\neq
<i>ligne13</i>	\neq	\neq	\neq	\neq	\neq
<i>ligne14</i>	\neq	\neq	\neq	\neq	\neq

Code : TeX

```

 $\$$ 
\begin{array}{c c c c c c}
ligne 1 & \succcurlyeq & \Vdash & \therefore & \curlyeqprec & \curlyeqsucc \\
ligne 2 & \vDash & \because & \blacktriangleleft & \blacktriangleright & \Vdash \\
ligne 3 & \Hbar & \triangleleft & \triangleright & = & \neq \\
ligne 4 & \triangleleft & \triangleright & \neq & \neq & \neq \\
ligne 5 & \neq & \neq & \neq & \neq & \neq \\
ligne 6 & \neq & \neq & \neq & \neq & \neq \\
ligne 7 & \neq & \neq & \neq & \neq & \neq \\
ligne 8 & \neq & \neq & \neq & \neq & \neq \\
ligne 9 & \neq & \neq & \neq & \neq & \neq \\
ligne 10 & \neq & \neq & \neq & \neq & \neq \\
ligne 11 & \neq & \neq & \neq & \neq & \neq \\
ligne 12 & \neq & \neq & \neq & \neq & \neq \\
ligne 13 & \neq & \neq & \neq & \neq & \neq \\
ligne 14 & \neq & \neq & \neq & \neq & \neq
\end{array}

```

```
ligne 14 & \varsupsetneqq & \ntrianglerighteq & \lvertneqq &
\gvertneqq & \\
\end{array}
$$
```

Polices mathématiques (Mode mathématique)

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Code : TeX

```
$$A B C D E F G H I J K L M N O P Q R S T U V W X Y Z$$
$$\mathcal{A B C D E F G H I J K L M N O P Q R S T U V W X Y Z}$$
$$\mathbb{A B C D E F G H I J K L M N O P Q R S T U V W X Y Z}$$
$$\mathfrak{A B C D E F G H I J K L M N O P Q R S T U V W X Y Z}$$
$$\mathsf{A B C D E F G H I J K L M N O P Q R S T U V W X Y Z}$$
$$\mathbf{A B C D E F G H I J K L M N O P Q R S T U V W X Y Z}$$
```

Caractères de structure (Mode mathématique)

Ici sont listés les caractères utiles à la création de code en LaTeX et qui ne peuvent pas être insérés simplement dans un document LaTeX.

\$

&

%

#

{

}

—

~

^

\

Code : TeX

```
$$ \$ $$  
$$ & $$  
$$ % $$  
$$ # $$  
$$ { $$  
$$ } $$  
$$ _ $$  
$$ \sim $$  
$$ \wedge $$  
$$ \backslash $$
```

Divers (Mode mathématique)

<i>ligne1</i>	∞	\forall	\mathbb{k}	\wp	∇
<i>ligne2</i>	\exists	\star	\angle	∂	\nexists
<i>ligne3</i>	\diagdown	\sphericalangle	\eth	\emptyset	\diagup
<i>ligne4</i>	\sphericalangle	\clubsuit	\varnothing	\diamond	\complement
<i>ligne5</i>	\diamond	\imath	\Finv	\triangledown	\heartsuit
<i>ligne6</i>	\jmath	\Game	\triangle	\spadesuit	ℓ
<i>ligne7</i>	\hbar	\vartriangle	\cdots	\iiint	\hslash
<i>ligne8</i>	\blacklozenge	\vdots	\iiint	\lozenge	\blacksquare
<i>ligne9</i>	\ldots	\iint	\mho	\blacktriangle	\ddots
<i>ligne10</i>	\sharp	\prime	\circledS	\natural	\flat
<i>ligne11</i>	\square	\backprime	\Re	\Im	\surd

Code : TeX

```


$$\begin{array}{c c c c c c}
\text{ligne 1} & \&\infty & \&\forall & \&\mathbb{k} & \&\wp & \&\nabla \\
\text{ligne 2} & \&\exists & \&\star & \&\angle & \&\partial & \&\nexists \\
\text{ligne 3} & \&\diagdown & \&\sphericalangle & \&\eth & \&\emptyset & \&\diagup \\
\text{ligne 4} & \&\sphericalangle & \&\clubsuit & \&\varnothing & \&\diamond & \&\complement \\
\text{ligne 5} & \&\diamond & \&\imath & \&\Finv & \&\triangledown & \&\heartsuit \\
\text{ligne 6} & \&\jmath & \&\Game & \&\triangle & \&\spadesuit & \&\ell \\
\text{ligne 7} & \&\hbar & \&\vartriangle & \&\cdots & \&\iiint & \&\hslash \\
\text{ligne 8} & \&\blacklozenge & \&\vdots & \&\iiint & \&\lozenge & \&\blacksquare \\
\text{ligne 9} & \&\ldots & \&\iint & \&\mho & \&\blacktriangle & \&\ddots \\
\text{ligne 10} & \&\sharp & \&\prime & \&\circledS & \&\natural & \&\flat \\
\text{ligne 11} & \&\square & \&\backprime & \&\Re & \&\Im & \&\surd
\end{array}$$


```

Caractères spéciaux (Mode texte)

ligne 1	ó	ö	ô	\	õ
ligne 2	ō	ş	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 & \={o} & \d s & \. {o} & \u{o} & \H{o}
\\
ligne 3 & \t{oo} & \c{o} & \d{o} & \r s & \b{o}
\\
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 vos débuts dans ce tutoriel, 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 préambule. Gardez tout de même à l'esprit qu'un package s'insère via la commande `\usepackage{}`. 😊

La liste

Utilité	Les packages	Commentaires
Packages de langue	inputenc fontenc babel	Ces trois packages sont indispensables pour écrire un document en français.
Création d'un layout	layout	Le layout permet de connaître le gabarit de mise en page d'un document.
Modification des marges	geometry	Il permet de régler finement les marges.
Interligne	setspace	-
Soulignement	soul ulem	Le package soul peut aussi s'utiliser pour barrer du texte.
Symbole €	eurosym	-
Pack de police	bookman charter newcent lmodern mathpazo mathptmx	N'en utiliser qu'un à la fois (il est fortement conseillé d'en utiliser un).
Citation d'url	url	-
Citation de code	verbatim moreverb	verbatim donne accès aux environnements verbatim et verbtimtab
Citation de code coloré	listings	ne pas oublier de paramétrer correctement <code>\lstset</code>
En-têtes et pieds de pages personnalisés	fancyhdr	<code>\pagestyle{fancy}</code>
Insertion d'images	graphicx	Primordial pour travailler sur des images.
Insertion d'une image dans un paragraphe	wrapfig	A utiliser en dernier recours uniquement.
Manipuler les couleurs et colorer du texte	color	A utiliser avec prudence et retenue. 😊
Colorer du texte contenu dans un tableau	colortbl	Utilisez plutôt des couleurs en niveau de gris.
Insertion d'expressions scientifiques	amsmath amssymb mathrsfs amsthm	amsthm n'est utile que si il y a des théorèmes.
Création d'index	makeidx	-

Petite astuce

Petite astuce toute simple pour ne pas avoir à vous souvenir de chacun des noms des 29 packages utilisés ici, il vous suffit de créer un gros bout de code à insérer en préambule dont chaque ligne est précédée par un signe %. Il vous suffit d'enlever le signe pour qu'une ligne ne soit plus considérée comme commentaire.

N'hésitez pas à copier-coller 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{graphicx}
%\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{asmthm}
%\usepackage{makeidx}
```

Nous venons de passer au moins une dizaine d'heures ensemble pour à peine entrevoir le potentiel de 28 packages et LaTeX en comporte des centaines. Ce potentiel et ce vivier font la force du programme, néanmoins si vous arrivez ne serait-ce qu'à utiliser de façon intelligente 10% des packages de LaTeX vous pourrez vous estimer expert.

Qui sait, peut être qu'un jour vous apprendrez à fabriquer vos propres packages... 🤔

Aller plus loin

Cette annexe épilogue est la vôtre et vous permettra d'ajouter des petits points au tuto. Pour ce faire, il vous suffit **de m'envoyer un message privé** dans lequel vous écrirez ce que vous souhaitez rajouter au tuto.

Actuellement, en me basant sur ce que m'ont envoyés les zéros, cette annexe contient les éléments suivants :

- description rapide de beamer (c'est quoi ? A quoi ça sert ?)
- règles élémentaires de typographie (j'ai pas grand-chose de ce côté-là, hésitez pas à m'envoyer des trucs)
- des mises en avant de certaines interfaces graphiques
- une petite biblio avec des ressources utiles à la création de documents en LaTeX (et disponibles gratuitement).

A vos mails !

Typographie et encodage Latin-1 et UTF-8

Un débat a fait rage à diverses étapes de la composition de ce cours : 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 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, Vous pouvez le télécharger en cliquant sur ce lien : [TeXnicCenter](#).

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.

[Voir le document.](#)

De l'art de la présentation

Continuons dans l'ouverture : je vous propose ici deux manuscrits de typographie qui vous permettront, l'un comme l'autre, de comprendre quelles sont les choses à 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.

[Premier manuscrit](#), [deuxième manuscrit](#).

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. Je vous ai mis :

- un exemple de [cours sous forme de code commenté](#) ;
- le [tutoriel de formation à LATEX de l'ENS](#) (un bon niveau général est requis) ;
- le [Framabook](#) Tout ce que vous avez toujours voulu savoir sur LATEX sans jamais oser le demander ;
- le [Wikibooks en français](#), très bien construit, traitant de LaTeX. Il donne des exemples intéressants d'utilisation du langage en chimie ;
- le [Wikibooks en anglais](#), un peu plus fourni ;
- le [cours de Ukonline](#), très répandu chez les débutants ;
- une [courte introduction à LaTeX](#), véritable perle de pédagogie.

Les incontournables

Quatre outils fort utiles :

- [Le conjugueur](#), véritable Bescherelle informatique ;
- [un aide-mémoire supplémentaire](#) ;
- une [table](#) listant pratiquement tous les caractères spéciaux ;
- le guide "[Détecter et résoudre les 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.

- [La page Beamer](#) sur Wikipedia.
- [Introduction à Beamer](#) sous Linux.
- [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.

- [L'éditeur LaTeX](#) en ligne de Google.
- [Lyx](#), le presque traitement de texte en LaTeX.
- [Emacs](#), la console par excellence.

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. 😊