

Écrivez de vrais documents

Une introduction au logiciel LaTeX

29 septembre 2014



Introduction

Si vous êtes habitués à utiliser Word depuis l'école primaire, vous n'avez peut-être jamais pensé que l'on puisse utiliser *autre chose* que Word pour rédiger un document. Pourtant, aucun des livres que vous pouvez trouver dans une librairie n'a été écrit avec Word. En fait, quasiment personne dans le monde de l'édition n'utilise Word. Les professionnels utilisent en général d'autres outils, qui sont plus adaptés à leurs besoins. Deux logiciels sont particulièrement utilisés par les éditeurs : LaTeX et Adobe InDesign.

LaTeX sert à écrire des documents plutôt classiques, comme des romans ou des livres documentaires. InDesign laisse aux designers plus de libertés pour la mise en page, il est donc très utilisé pour les brochures et le magazines.

Il se trouve que LaTeX est libre, gratuit, relativement facile à utiliser et très puissant. C'est donc à ce logiciel que nous allons nous intéresser dans ce guide.

Objectifs de ce guide

Ce guide ne fait qu'une vingtaine de pages, mais après l'avoir lu vous devriez être capables de :

- Gagner du temps en structurant votre document correctement,
- Écrire des documents d'une grande qualité, similaire à celle d'un vrai livre,
- Travailler en groupe facilement avec d'autres personnes,
- Avoir de bonnes bases pour trouver facilement les fonctionnalités dont vous aurez besoin.

1 Qu'est ce que LaTeX ?

1.1 Un logiciel automatique

La première chose qu'il est important de comprendre est que *LaTeX fonctionne sur une logique complètement différente de celle de Word*.

Word est un logiciel où il faut tout faire à la main. Vous sélectionnez ce que vous voulez modifier, puis vous appuyez sur un bouton, et la sélection prend l'apparence que vous souhaitez.

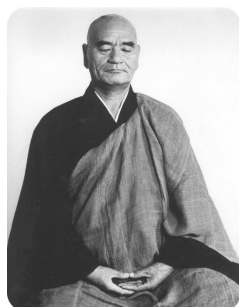
LaTeX est quant à lui un logiciel automatique. Vous lui donnez un texte dans lequel vous insérez des instructions, et il génère tout seul un document (en pdf) à partir de vos indications. Autrement dit, vous décrivez le document et LaTeX le crée pour vous.

Comme sur les forums

Si vous utilisez des forums de discussion, vous avez peut-être l'habitude d'écrire `[b]mot[/b]` pour mettre un mot en caractère gras. En fait, LaTeX fonctionne un peu sur le même principe.

Pour résumer, vous contrôlez Word en appuyant sur des petits boutons, alors qu'avec LaTeX vous commandez le logiciel à lui en écrivant des instructions dans un vrai langage.

Vous vous rendrez bientôt compte que cela est beaucoup plus rapide.



« Il faut dix minutes pour comprendre LaTeX, mais il faut toute une vie pour en maîtriser les pouvoirs. »

— Deshimaru Sensei

1.2 À l'origine du projet

Le projet LaTeX a été lancé par Donald Knuth, un mathématicien de l'Université de Stanford. On lui doit, entre autres, la théorie de l'analyse des algorithmes et il est considéré comme l'un des pères de l'informatique moderne.

Par la suite, de nombreux contributeurs y ont ajouté des composants et de nombreuses fonctionnalités. Il n'est donc pas développé par une entreprise comme Microsoft, mais par une communauté de développeurs qui l'améliorent selon leurs besoins. C'est aussi pour cette raison qu'il est gratuit.

Mais pourquoi personne ne l'utilise ?

On ne vous a probablement pas parlé de LaTeX au collège, mais il y a quand même de nombreuses personnes qui s'en servent. Les scientifiques, en particulier dans les domaines de la physique et des mathématiques, utilisent généralement tous LaTeX. En effet, pendant longtemps, LaTeX a été le seul moyen d'écrire des équations correctement.

Il dispose aussi d'outils très perfectionnés pour gérer les citations bibliographiques, d'où sa popularité pour écrire des articles scientifiques.

2 De bonnes raisons pour l'utiliser

Voyons quelques-uns des avantages que l'on peut avoir à utiliser LaTeX à la place de Word.

2.1 Il est compatible avec tout

Déjà, c'est un logiciel libre, gratuit et multi-plateforme. Cela veut dire qu'il fonctionnera partout, sur n'importe quel ordinateur, et que vous pouvez faire ce que vous voulez avec. Vous pouvez l'installer sur un ordinateur, mais il existe aussi des compilateurs en ligne où vous copiez-collez votre texte source dans un formulaire web, et ils vous renvoient votre document mis en forme au format pdf. Vous pourriez taper du LaTeX sur votre téléphone ou sur votre grille-pain, cela fonctionnera pareil. Il est même possible de l'utiliser depuis un smartphone.

Il n'affichera pas votre document n'importe comment sous prétexte que vous êtes sous Mac, ou que votre collègue utilise la version 2010 plutôt que 2011. Quand vous aurez utilisé LaTeX, vous serez étonné de vous rendre compte à quel point Word ne marchait pas.

.doc, .odf et .docx

À l'origine, Word utilisait le format `.doc`. C'était un format propriétaire, c'est-à-dire que personne en dehors de Microsoft ne savait comment le lire, et on ne pouvait donc l'ouvrir qu'avec Word.

Puis, OpenOffice/LibreOffice sont apparus, avec leur propre format : le `.odf`. Cette fois, il s'agit d'un format libre, avec toute la documentation nécessaire pour que les logiciels concurrents puissent le lire et le modifier comme ils le veulent.

Cela n'était pas à l'avantage de Microsoft, car si tout le monde commençait à utiliser le `.odf`, il n'y aurait plus de problèmes d'incompatibilité, et Microsoft perdrait leur monopole puisque chacun pourrait utiliser le logiciel qu'il souhaite.

Ils ont donc développé le format `.docx`. Il s'agit lui aussi d'un format libre, mais il est beaucoup plus compliqué que le format `odf`. Sa documentation fait 6546 pages, il est donc presque impossible pour les concurrents de l'implémenter parfaitement. Comme cela, Microsoft s'assurent que les problèmes persisteront et que leur monopole ne sera pas dérangé.

Comme il est open-source, de nombreux outils ont pu être développés pour le travail collaboratif. Nous en verrons quelques-uns à la section 10, pour que vous puissiez travailler en équipe.

2.2 Il donne des documents homogènes

Avec LaTeX, vous vous contentez de dire « ça, c’est un titre », « ça, c’est une sous-section » et LaTeX s’occupe du reste. Tous les titres auront donc la même apparence, la même police etc. Il n’est donc plus nécessaire de passer des heures à nettoyer le document pour que tout ait la même mise en forme (figure 1).

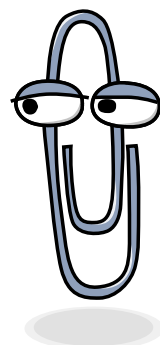


FIGURE 1 – « Héhé... Il semblerait que vous soyez en train d’essayer de régler les marges de la page en bougeant les petits curseurs... »

2.3 La numérotation

Un autre avantage appréciable de LaTeX est qu’il sait compter. Que ce soient les numéros de parties, l’ordre des figures, les notes en pied de page ou les citations bibliographiques, il s’occupe de tout et vous n’aurez plus à vous préoccuper de la numérotation.

Si ces arguments vous ont donné envie de l’essayer, il ne vous reste plus qu’à l’installer.

3 Installation

3.1 TeXLive

LaTeX est un logiciel modulaire. Ce n’est pas un gros logiciel avec une liste de fonctionnalités déterminées. Il faut plutôt imaginer cela comme un noyau central, auquel s’ajoutent des milliers de petits modules indépendants qui ont chacun une fonction spécifique (il y a des modules pour écrire des partitions de musique, pour dessiner des formules chimiques ou encore pour décrire des parties d’échecs).

Si vous n’installiez que LaTeX tout seul, vous ne pourriez faire que le strict minimum. Et comme vous débutez, vous risqueriez d’avoir du mal à choisir les composants que vous voulez. Nous allons donc installer un kit qui contient tous les modules dont vous pourriez possiblement avoir besoin. Ce kit s’appelle TeXLive.

3.1.1 Avec Linux

Comme d’habitude, il suffit d’installer TeXLive avec le gestionnaire de logiciels. En général, le paquet s’appelle tout simplement `texlive`.

Sous Arch, c’est `texlive-most`, parce que les développeurs n’ont pas tout mis (ils ont par exemple retiré des packages qui servent à écrire en croate ou en arménien. Si vous les voulez vraiment, installez aussi `texlive-lang`).

3.1.2 Avec Mac

Nous allons installer MacTex, qui est une distribution gigantesque qui contient entre autres TeXLive. Vous la trouverez ici :

<http://www.tug.org/mactex>

L'avantage, c'est que vous aurez *tout* installé en un seul clic.

3.1.3 Avec Windows

Nous n'allons pas installer TexLive mais MiKTeX, une autre distribution similaire qui est plus légère. Cela ne devrait pas changer grand chose pour vous. Téléchargez MiKTeX ici :

<http://www.miktex.org>

Et suivez la marche à suivre décrite sur le site.

3.2 Une interface digne de ce nom

En théorie, n'importe quel logiciel capable d'écrire du texte suffit à créer des documents en LaTeX, puisque le code source n'est rien d'autre qu'un simple texte. Cependant, il existe de très nombreux logiciels spécialement conçus pour écrire plus facilement vos documents.

Ces logiciels savent faire plusieurs choses très utiles :

- Compléter les commandes pour que vous n'ayez pas à les taper,
- Afficher un aperçu du pdf,
- Afficher les commandes en couleurs pour une meilleure lisibilité,
- Corriger l'orthographe,
- Afficher un sommaire pour se déplacer plus vite dans le document...

Nous allons en passer plusieurs en revue, il ne vous restera que l'embarras du choix. N'hésitez pas à en essayer plusieurs pour choisir votre préféré.

3.2.1 Gummi

Gummi est une interface pour LaTeX simple et très efficace. Elle permet d'afficher un aperçu en temps réel du document final, qui se met à jour dès que vous arrêtez de taper.

Gummi n'est hélas disponible que sous Linux.

Maintenez Ctrl enfoncée et cliquez sur l'aperçu à l'endroit que vous voulez modifier. Cela vous emmènera directement à l'endroit correspondant dans le code source.

3.2.2 TexShop

TexShop est disponible sous Mac, et il est lui aussi très bien pour les débutants. Il est inclus dans MacTex, vous l'avez donc déjà installé. Si vous tapez le début d'une commande et que vous appuyez sur TAB, il cherchera à la compléter tout seul. Vous pouvez appuyer sur TAB jusqu'à trouver la bonne commande. C'est assez pratique.

3.2.3 TexMaker

TexMaker est assez similaire à Gummi. Il a l'avantage de fonctionner sur n'importe quel système d'exploitation.

3.2.4 Lyx

Cet éditeur est un peu spécial, car il reproduit le fonctionnement de Word, c'est-à-dire une interface avec des boutons. Avec Lyx, le code est masqué et les titres apparaissent directement en gros et en gras comme dans le document final. On perd donc une partie des avantages de LaTeX, mais il peut toujours dépanner si vous devez travailler avec quelqu'un qui ne sait pas se servir de LaTeX.

3.2.5 Vim et Emacs

Si vous êtes habitués à l'un de ces éditeurs de texte pour programmeurs, sachez qu'il est aussi possible de les utiliser pour écrire du LaTeX.

3.3 Première compilation

Pour bien comprendre le fonctionnement des choses, je vous propose de faire une première compilation à la main. Vous pouvez sans problème passer cette étape, car c'est un peu technique.

1. Ouvrez un éditeur de texte brut (le bloc-note, gedit, nano...) et recopiez ceci sans vous tromper :

```
\documentclass[11pt]{article}
\begin{document}
Bonjour le monde.
\end{document}
```

2. Sauvegardez ce fichier sous le nom test.tex
3. Lancez un terminal et déplacez-vous à l'emplacement de test.tex
4. Lancez la commande « `pdflatex test.tex` »
5. Regardez ce qui est apparu dans le dossier. En principe, il devrait y avoir entre autres un fichier pdf avec le texte « Bonjour le monde. ».

Et voilà, vous avez fait fonctionner LaTeX pour la première fois. Par la suite, c'est votre éditeur qui s'occupera de tout ça (ouf).

3.4 Un modèle de document vide

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

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

\begin{document}

\end{document}
```

Voici un modèle pour un document vide en français. Ce n'est pas grave si vous ne comprenez pas à quoi servent les trois `\usepackage`, dites-vous simplement qu'ils permettent de gérer tous nos accents et autres bizarreries du français.

Gardez bien ce modèle sous la main, car nous allons l'utiliser tout le temps pour la suite de ce guide.

Le code source d'un document LaTeX est divisé en deux parties :

- Tout ce qui se trouve avant `\begin{document}` se nomme la *préambule*. C'est là que prendront place les réglages sur l'apparence et la mise en page du document.
- Le vrai contenu du document sera à mettre entre `\begin{document}` et `\end{document}`.

La première ligne détermine la classe du document. Est-ce un article, un livre, une lettre ? Ici, nous utilisons la classe `article`.

La classe article

En pratique, la classe « article » convient pour la plupart des documents courants. Nous allons nous y tenir dans ce guide.

Maintenant que vous avez un modèle prêt à l'emploi, vous pouvez essayer de taper le texte de votre choix dans le corps du document, et de générer le pdf correspondant avec votre éditeur.

4 Comment donner des instructions

4.1 Les commandes de LaTeX

Avec LaTeX, toutes les instructions commencent par le signe `\`. Cela s'appelle une contre-oblique (ou backslash). Ce n'est PAS un slash, il penche vers la gauche.

Quand LaTeX rencontrera ce caractère, il saura que vous voulez lui donner une indication sur ce qu'il doit faire.

Avec un clavier AZERTY, on obtient ce caractère avec AltGr + 8. Souvenez-vous-en, vous allez en avoir besoin tout le temps.

Si vous utilisez Mac

Il faut taper « Alt + Majuscule + / ». Faites attention à ne pas abîmer vos ongles soigneusement manucurés en appuyant sur les touches.

Une commande peut accepter différents paramètres (ou « arguments »). Les arguments obligatoires s'écrivent entre accolades. Il s'agit en général du texte qui sera affecté par la commande. Les arguments facultatifs s'écrivent entre crochets. Ils permettent de faire des réglages, par exemple pour définir l'épaisseur d'un trait.

Voilà quelques exemples de ce à quoi une commande LaTeX complète peut ressembler :

```
\nom{argument}
\nom{argument 1}{argument 2}
\nom[argument facultatif]{argument}
```

Mais il est temps pour vous d'apprendre votre première vraie commande : la commande `\emph`.

La commande `\emph{texte}` permet de faire une *emphase*. Elle aura pour effet de mettre le texte entre accolades en italique.

```
Ceci est \emph{particulièrement} important.
```

Donnera : Ceci est *particulièrement* important.

5 La structure du document

5.1 Section, subsection, subsubsection

Nous allons voir les commandes qui permettent de structurer le texte. Ce sont de loin les plus importantes.

- `\section{Titre de la section}` indique un titre de section.
- `\subsection{Titre de la sous-section}` indique un titre de sous-section.
- `\subsubsection{Titre de la sous-sous-section}` indique un titre de sous-sous-section.

Ce n'est pas si compliqué, n'est-ce pas ?

Maintenant, reprenez le modèle que je vous ai donné tout à l'heure, et écrivez ceci dans le corps du texte (c'est-à-dire dans la partie entre `\begin{document}` et `\end{document}`).

```

\section{Première partie}

\subsection{Sous-partie}
Contenu de la sous-partie.

\subsection{Autre sous-partie}
Contenu de la deuxième sous-partie.

\section{Deuxième partie}
Contenu de la deuxième grande partie.

```

Générez maintenant le pdf. Et voilà, remarquez comme LaTeX a tout pris en charge, et tout numéroté correctement.

Rien qu’avec ces différentes commandes, vous pouvez déjà faire la plupart de vos documents habituels. Évidemment, pour l’instant vous ne pourrez pas donner à votre article le design exact que vous imaginez, et vous devrez vous contenter du thème par défaut. Mais avec la pratique, vous progresserez petit à petit sans vous en rendre compte.

5.2 Il faut vraiment taper tout ça à chaque fois ?

Bien sûr que non. Il existe tout plein de méthodes pour ne pas avoir à tout taper à la main. En fait, il n’est même pas nécessaire de se souvenir d’une commande pour l’utiliser. Mais nous verrons ces méthodes un peu plus tard dans le livre. Pour l’instant, vous êtes encore en train d’apprendre, vous devrez donc taper les commandes en entier pour bien comprendre comment ça fonctionne. Ensuite seulement, vous pourrez chercher à gagner du temps en utilisant des raccourcis.

5.3 Les « environnements »

Ces commandes sont un peu particulières, puisqu’elles affectent non pas ce qui est entre les accolades, mais tout ce qui se trouve entre les deux commandes. On appelle cela un *environnement*. Par exemple, voici comment centrer du texte :

```

\begin{center}
Votre paragraphe centré ici.
\end{center}

```

Nous verrons d’autres environnements un peu plus tard.

5.4 Faire des listes

Faire des listes en LaTeX s’avère beaucoup moins aléatoire qu’avec Word. En LaTeX, les listes sont des environnements. Les trois principaux sont `itemize`, `enumerate` et `description`, ils correspondent aux trois types de liste les plus communs. À l’intérieur de l’environnement, on écrit les items de la liste avec la commande `\item`.

Mais quelques exemples valent mieux qu’une longue explication :

5.4.1 Itemize

Une liste normale, avec des tirets.


```
\begin{itemize}
\item Premier item,
\item Deuxième item,
\item Troisième item.
\end{itemize}
```

- Premier item,
- Deuxième item,
- Troisième item.

5.4.2 Description

Celle-là permet d'écrire des définitions, comme dans un dictionnaire.

```
\begin{description}
\item[Mot :] Définition,
\item[Mot :] Définition,
\item[Mot :] Définition.
\end{description}
```

Mot : Définition,
Mot : Définition,
Mot : Définition.

5.4.3 Enumerate

Enfin, *enumerate* sert à faire des listes numérotées.

```
\begin{enumerate}
\item Premier item,
\item Deuxième item,
\item Troisième item.
\end{enumerate}
```

1. Premier item,
2. Deuxième item,
3. Troisième item.

5.5 La page de titre

Avant de créer un titre, nous allons déjà donner au logiciel les informations nécessaires, en écrivant les commandes suivantes dans le préambule :

```
\title{Votre titre}
\author{Votre nom}
\date{La date}
```

Si vous ne mettez pas la commande `\date`, la date d'aujourd'hui sera utilisée.

Ensuite, à l'endroit où vous voulez placer votre titre, utilisez la commande `\maketitle`. Cela fera apparaître un grand titre.

Cette commande permet de créer un titre simple et élégant, mais elle ne permet pas beaucoup de customisation. Si vous voulez faire un titre plus sophistiqué, mieux vaut le faire à la main et ne pas du tout utiliser `\maketitle`.

5.6 Faire des paragraphes

Les retours à la ligne simples ne sont pas pris en compte par LaTeX. Vous pouvez donc aérer votre code source. Beaucoup de gens appuient sur Entrée après chaque phrase, voire après chaque idée distincte. C'est un bon moyen d'avoir les idées plus claires.

```
Aaa
Bbb
Ccc.
```

Aaa Bbb Ccc.

Par conséquent, pour avoir un *retour à la ligne* dans le document, il faut *passer une ligne* dans le code source.

```
Aaa.
```

```
Bbb.
```

Aaa.

Bbb.

Le double-antislash `\\` donne le même résultat.

```
Aaa.\\
```

```
Bbb.
```

Aaa.

Bbb.

Mais alors, comment passer une ligne dans le document ? Il suffit d'utiliser la commande `\\` puis de passer une ligne dans le code source. Cela peut paraître un peu confus, mais on s'y habitue.

```
Aaa.\\
```

```
Bbb.
```

Aaa.

Bbb.

5.7 Charger des packages

Les packages, ce sont comme des extensions qui vous permettent de vous simplifier la vie. Avant de s'en servir, il faut dire à LaTeX que l'on veut l'utiliser en écrivant cette commande *dans le préambule* (avant `\begin{document}`):

```
\usepackage{nom du package}  
ou bien  
\usepackage[options]{nom du package}
```

Certains package prennent effet dès qu'on les appelle (essayez avec le package `cmbright`, qui change l'apparence du texte) :

```
\usepackage{cmbright}
```

D'autres rendent disponibles de nouvelles commandes (par exemple `chemfig` donne accès à des commandes pour écrire des formules chimiques).

```
\usepackage{chemfig}
```

Puis, dans le corps du document, essayez par exemple d'écrire ceci :

```
\chemfig{HOOC-*6(-==-=)}
```

Vous obtiendrez une belle molécule d'acide benzoïque.

6 Des images, des figures et des tableaux

6.1 Des images

Commencez par charger le package *graphicx* dans le préambule (`\usepackage{graphicx}`).

La commande pour inclure une image est `\includegraphics`. Entre les accolades, il faut écrire le nom du fichier de l'image, qui doit se trouver dans le même dossier que le fichier tex. Les formats d'image reconnus sont png, jpg, pdf et eps. Il n'est pas nécessaire d'écrire l'extension du nom de fichier : si plusieurs images portent le même nom dans des formats différents, LaTeX choisira la meilleure.

Par exemple, si vous mettez une image nommée `lapin.png` à côté de votre fichier tex, voici comment l'afficher :

```
\includegraphics{lapin}
```

Le nom du fichier ne doit pas contenir d'espace, utilisez plutôt des traits d'union (`petit-lapin.jpg`).

Vous pouvez régler les dimensions de l'image en utilisant les paramètres `width` et `height`. Attention, si vous mentionnez les deux l'image apparaîtra déformée.

```
\includegraphics[width=40mm]{lapin}
\includegraphics[height=30mm]{lapin}

\includegraphics[width=40mm,height=30mm]{lapin}
```

N'oubliez pas de mettre une unité (mm, cm, px...). Pour les nombres décimaux, il faut mettre un point à la place de la virgule (`width=4.2cm` par exemple).

Une commande très utile à ce moment est `\linewidth`, qui correspond à la largeur du texte. Vous pouvez mettre un coefficient avant : `0.5\linewidth` correspond à la moitié d'une ligne, par exemple.

```
\includegraphics[width=\linewidth]{lapin}
\includegraphics[width=0.5\linewidth]{lapin}
```

Images vectorielles ou matricielles ?

Puisque nous parlons d'inclure des images, voici un conseil important (qui est valable aussi avec Word/Writer). Vous devriez utiliser des images *vectorielles*.

Les images que vous avez l'habitude d'utiliser (jpg, png, etc.) enregistrent la couleur de chaque pixel. On dit qu'elles sont *matricielles*.

Au contraire, les images vectorielles enregistrent la forme, la couleur et l'épaisseur des traits. Les principaux formats sont *pdf*, *svg* et *eps*. L'avantage, c'est que l'on peut changer leur taille n'importe comment, la qualité restera la même. Le résultat sera par conséquent bien meilleur à l'impression (figure 2). Les images vectorielles sont aussi bien plus légères.

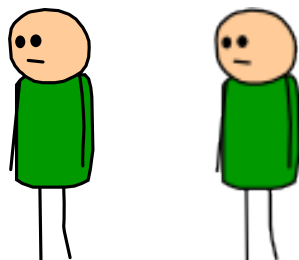


FIGURE 2 – La même image en vectoriel (gauche) et en matriciel (droite). Essayez de zoomer dessus, vous verrez la différence.

Un bon logiciel pour faire vos schémas en vectoriel est *Inkscape*. Sinon, il y a Adobe Illustrator. Excel et Calc permettent d'exporter leurs graphiques en vectoriel, au format pdf. Notez que LaTeX ne connaît pas le svg, si vous en avez il faudra le convertir en pdf.

Préférez toujours des images vectorielles, quand c'est possible.

6.2 Des tableaux

Autant vous prévenir tout de suite, les commandes de LaTeX pour faire des tableaux sont un peu compliquées. Plutôt que des les construire à la main, je vous conseille de les générer automatiquement :

- Vous pouvez par exemple utiliser un éditeur de tableaux en ligne, tel que l'excellent www.tablesgenerator.com/
- Il existe des plug-ins pour exporter du code LaTeX depuis Excel, LibreOffice et même R. Il suffit de le googler.
- Regardez dans votre éditeur LaTeX, la plupart possèdent des outils pour construire des tableaux (avec Gummi, le bouton se trouve en bas à gauche).

Un exemple de ce que ça donne :

```
\begin{tabular}{|l|l|l|l|}  
\hline  
Voici & Le & Type\\  
\hline  
De & Code & Que\\  
\hline  
Vous & Allez & Obtenir\\  
\hline  
\end{tabular}
```

Voici	Le	Type
De	Code	Que
Vous	Allez	Obtenir

Si vous voulez comprendre comment tout cela fonctionne, c'est ici :
<https://fr.wikibooks.org/wiki/LaTeX/Tableaux>.

6.3 Les figures

Tout cela est très bien, mais vous n'allez tout de même pas balancer vos images et vos tableaux en plein milieu du texte comme des barbares.

Nous allons les mettre dans des *flottants*. Le principe d'un flottant, c'est qu'il flotte. Le contenu du flottant n'apparaîtra pas directement là où vous l'avez mis : LaTeX va plutôt chercher le meilleur endroit pour le caser à proximité, en général entre deux paragraphes. Vous pourrez ensuite l'appeler dans le texte en disant par exemple « voir figure 1 ». Vous pourrez aussi générer facilement une liste des figures et une liste des tables, dans le cas d'un mémoire.

Pour faire un flottant, il suffit d'utiliser l'environnement `figure` pour les images, et `table` pour les tableaux.

```
\begin{figure}[h]  
\includegraphics[width=\linewidth]{votre-image}  
\end{figure}
```

Le petit `[h]` sert à indiquer à quel endroit vous voulez que le flottant soit. `[h]` signifie *here*, donc LaTeX le placera le plus près possible de l'endroit où la commande est écrite.

Les autres possibilités sont `[t]` (*top*, en haut de la page), `[b]` (*bottom*, en bas) et `[p]`, qui indique à LaTeX de faire une page à part où il mettra toutes les figures.

Pour les tableaux, c'est exactement le même fonctionnement. Il suffit de remplacer le mot `figure` par `table`.

6.3.1 Une légende

La commande `\caption{votre légende ici}` permet d'ajouter une légende. Placez là quelque part dans le flottant.

Par convention, vous devez placer la légende en dessous pour une image, et au dessus pour un tableau.

6.3.2 Appeler une figure dans le texte

Et maintenant, le plus difficile : faire référence à une figure dans le texte. Nous allons faire cela grâce à deux commandes qui vont ensemble, `\label` et `\ref`.

Ces commandes fonctionnent avec des mots clés. En mettant `\label{mot-clé}` dans la figure, vous définissez un mot clé associé à une figure précise. En mettant `\ref{mot-clé}` dans le texte, vous récupérez le numéro de la figure correspondant. Voyons un exemple :

```
\begin{figure}[h]

UNE PHOTO DE PINGOUIN
\caption{Votre légende.}
\label{pingouin}

\end{figure}

...comme on peut le voir sur la figure \ref{pingouin}.
```

Il faut générer le pdf deux fois de suite pour que ces commandes fonctionnent.

Ces deux commandes ne sont pas limitées aux figures. Utilisées ailleurs, elles permettent de renvoyer à une section particulière.

7 D'autres commandes indispensables

7.1 Écrire des commentaires dans le code source

En LaTeX, le caractère `%` permet d'insérer des commentaires dans le code. Tout ce qui se trouve après ce caractère jusqu'à la fin de la ligne ne sera pas pris en compte lors de la compilation du document.

Utilisez les commentaires pour laisser des petits messages cachés aux personnes avec qui vous travaillez.

7.2 Note en pied de page

La commande `\footnote{Votre note ici}` permet d'écrire des notes en pied de page¹. Il suffit de l'insérer à l'endroit voulu dans le texte, LaTeX se charge de la numérotation tout seul².

1. Comme cela.

2. Voyez plutôt.

7.3 Quelques caractères spéciaux

7.3.1 Pourcentages

Nous avons vu que % servait à écrire des commentaires. Si vous voulez parler d'un pourcentage, utilisez la commande `\%` qui permet d'obtenir ce caractère comme si de rien n'était.

7.3.2 Dollars et euros

De même, \$ sert à écrire des formules de maths, il faut donc utiliser la commande `\$` pour afficher la caractère \$ dans un texte. Pour les euros, c'est la commande `\euro`, qui nécessite d'abord d'importer le package *eurosym*.

7.3.3 Lettres grecques

Pour taper des lettres grecques, c'est un peu compliqué. Il faut importer le package *upgreek* dans le préambule, puis taper `\upa\alpha`, `\upbeta`, `\upgamma` etc. Pour les lettres grecques majuscules, c'est la même chose mais avec une majuscule, par exemple `\Updelta`.

Il faut entourer la commande avec des \$ car, à l'origine, ces lettres sont censées être utilisées dans des formules de maths. Les développeurs de LaTeX n'ont manifestement jamais eu à écrire un texte sur les récepteurs de l'adrénaline.

7.4 Numéros de page

- `\thispagestyle{empty}` enlève le numéro de la page où la commande est invoquée,
- `\pagestyle{empty}` enlève tout bonnement les numéros de page de tout de le document, à partir de la commande.

7.5 Écrire en exposant

La commande pour écrire « en haut » est `\up{...}`.

`1\up{er}`, `2\up{e}`, `3\up{e}`.

1^{er}, 2^e, 3^e.

7.6 Et pour aller plus loin...

Si vous êtes habitué à Word, vous vous demandez sûrement « comment on fait pour changer la couleur d'un mot », « comment on change la police d'une phrase » et autres questions du même genre. En vérité, on a rarement besoin de faire ce genre de chose quand on écrit un document sérieux. Mais si vous voulez vraiment le faire, regardez ici :

https://fr.wikibooks.org/wiki/LaTeX/Mise_en_forme_du_texte

8 L'apparence du document

8.1 Vous avez probablement tort

Il est bien sûr possible de customiser chaque pixel de vos documents, mais c'est rarement une bonne idée. LaTeX fait des documents très classiques, qui respectent les standards de la typographie et de l'imprimerie. N'oubliez jamais ceci : *Le style par défaut convient*.

Vous pouvez le changer, mais vous allez y passer du temps et le résultat sera probablement moins bien qu'au départ.

Un exemple de chose à ne pas faire

La commande `textbf` permet de mettre du texte en gras. Cependant, les titres seront déjà mis en caractères gras automatiquement, et il est typographiquement incorrect d'utiliser du gras en plein paragraphe. Vous n'aurez donc jamais besoin d'employer cette commande en pratique. Mais allez faire comprendre cela à un utilisateur de Word.

Voici quand même quelques réglages qui pourront se révéler utiles dans certaines situations.

8.2 Taille du texte

Certains réglages qui concernent tout le document prennent place dès la première ligne du code, entre les crochets de la commande `\documentclass`.

Les paramètres possibles sont 10pt, 11pt et 12pt. Exemple :

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

Les autres tailles ne sont pas disponibles par cette méthode. En règle générale, si LaTeX ne veut pas faire quelque chose, c'est qu'il ne faut pas le faire.

8.3 La police d'écriture

En règle générale, on préfère utiliser une seule et unique police pour l'ensemble du document, que ce soit pour les titres, les équations ou les légendes de schémas. Nous allons donc utiliser des packages spéciaux qui changent uniformément la police de tout le document. Par exemple, ajouter `\usepackage{times}` dans le préambule mettra tout le document en Times New Roman.

Par conséquent, seules les polices permettant d'écrire des équations mathématiques sont disponibles sous forme de packages. Mais rassurez-vous, il existe suffisamment de packages pour couvrir tous vos besoins.

Voici quelques autres packages de polices incluses dans TeXLive :

- `mathpazo` : Correspond à la police *Palatino*. Très bien pour un document à imprimer.
- `cmbright` : Une police sans sérif très lumineuse.
- `concmath` : Une police assez vintage.
- `gentium` : C'est la police que vous avez devant les yeux.

Vous pourrez trouver un catalogue de polices d'écriture avec les packages correspondants ici :

<http://www.tug.dk/FontCatalogue>

Et le Comic Sans MS ?

Quelqu'un s'est amusé à créer un package pour cela, mais il n'est pas inclus dans TeXLive. Vous devrez donc l'installer vous-même pour connaître le plaisir d'écrire des équations mathématiques en Comic Sans MS.

<http://www.ctan.org/tex-archive/fonts/comicsans>

8.4 Géométrie

Le package `geometry` permet de faire tous les réglages imaginables par rapport à la taille des marges, du papier, des colonnes, et bien plus encore. Voici quelques exemples.

Donner à toutes les marges une largeur de 20 mm avec `margin=` :

```
\usepackage[margin=20mm]{geometry}
```

Régler chaque marge séparément avec `top`, `bottom`, `right` et `left` :

```
\usepackage[top=5px,bottom=5px]{geometry}
```

N'oubliez pas de mettre une unité aux longueurs.

8.5 Domptez LaTeX

Pour finir, voilà quelques commandes extrêmement utiles. Elles permettent de régler finement la position des différents éléments, dans le cas où LaTeX ne les aurait pas placés au bon endroit.

`\vspace{15mm}` laisse un espace vide vertical de la longueur souhaitée. Vous pourrez donc régler au millimètre près la distance entre deux paragraphes, etc. En mettant une longueur négative, vous pouvez aussi remonter un élément.

`\hspace{15mm}` correspond à un espace horizontal.

`\hfill` pousse ce qui se trouve après lui au maximum vers la droite. Exemple :

```
À gauche.\hfill À droite.
```

À gauche.

À droite

`\noindent` supprime l'indentation si vous le mettez au début d'un paragraphe ou avant une image.

`\clearpage` permet de continuer faire un saut de page.

9 Comment aller plus vite

Les commandes de LaTeX sont souvent un peu longues à taper. Heureusement, les éditeurs permettent de gagner du temps en les complétant pour vous.

9.1 Les *snippets*

Globalement, tous les éditeurs fonctionnent de la même façon. Vous commencez à taper une commande, vous appuyez sur TAB, et l'éditeur rajoute la fin. La plupart des éditeurs permettent de créer vos propres raccourcis, de sorte que vous n'ayez qu'à taper un mot d'appel puis appuyer sur TAB, et le logiciel rajoute une section de code définie à l'avance. Par exemple, vous pourrez faire en sorte que, si vous écrivez « fig » puis appuyez sur TAB, le code suivant tout entier apparaisse :

```
\begin{figure}  
\includegraphics[width=\linewidth]{}  
\caption{}  
\label{}  
\end{figure}
```

Et vous n'aurez plus qu'à compléter les différents éléments. Il faut reconnaître que c'est assez pratique, surtout que cela évite de mémoriser les commandes les plus compliquées. N'hésitez pas à fouiller dans votre éditeur pour trouver les options correspondantes.

Avec Gummi, c'est particulièrement simple. Vous pouvez créer des *snippets* de ce genre en allant dans Édition / Préférences / Éditeur / Configurer les extraits de code. Vous pouvez alors choisir un mot d'appel et écrire le code correspondant. Insérez ensuite des repères « \$1 », « \$2 », « \$3 » et ainsi de suite à l'emplacement des différents éléments à compléter. Quand vous utiliserez votre *snippet*, vous n'aurez qu'à appuyer sur TAB à nouveau pour passer de l'un à l'autre et remplir rapidement tous les champs.

9.2 D'autres langages et des convertisseurs

Une autre façon de procéder est de rédiger votre document en utilisant un autre langage descriptif plus simple que LaTeX. Il en existe plein, je vous conseille particulièrement Markdown ou Mediawiki (qui est la syntaxe utilisée pour écrire des articles de Wikipédia), car ils sont très faciles à apprendre.

Quand vous aurez écrit votre document, il vous suffira d'utiliser un convertisseur comme Pandoc, qui le traduira automatiquement en LaTeX. Il existe de nombreux autres convertisseurs, je vous laisse faire votre choix.

Pandoc fonctionne en ligne de commande, pour convertir un fichier de mediawiki vers LaTeX il faut taper ceci dans un terminal :

```
pandoc -f mediawiki -t latex votre_fichier
```

Version originale (mediawiki)

```
=Section=  
==Sous-section==  
Liste :  
*a  
*b  
*c
```

Version LaTeX après conversion

```
\section{Section}  
\subsection{Sous-section}  
Liste :  
\begin{itemize}  
\item a  
\item b  
\item c  
\end{itemize}
```

Le code mediawiki est quand même bien plus simple et plus lisible. Cela peut d'ailleurs être un bon moyen de convaincre vos collaborateurs de faire des documents en LaTeX.

10 Travailler en groupe

Maintenant que vous savez rédiger des documents en LaTeX, vous serez sans doute amenés à travailler avec d'autres personnes sur un même document. Heureusement, il existe plein de façons différentes de le faire et vous devriez en trouver une qui convient à vos collègues (même s'ils ne connaissent pas très bien LaTeX).

10.1 ShareLaTeX

Voici un outil que je vous recommande fortement si vous souhaitez travailler à plusieurs. Il s'agit d'un éditeur LaTeX en ligne, qui fonctionne exactement comme Google Docs, mais avec LaTeX.

Vous le trouverez à cette adresse : www.sharelatex.com

Il vous permet de travailler à plusieurs en même temps sur un même fichier (vous voyez en direct les modifications de vos collègues). ShareLaTeX comprend aussi un compilateur en ligne (basé sur TeXLive) qui permet de télécharger le fichier pdf final même sur un ordinateur où LaTeX n'est pas installé.

Bref, ShareLaTeX c'est bien.

10.2 Input et Subfiles

La commande `\input` permet tout simplement de récupérer le contenu d'un autre fichier texte, et de le mettre à la place de `\input`.

```
\input{autre-fichier.tex}
```

Ainsi, vous pouvez par exemple mettre le préambule, la bibliographie ou les figures dans un fichier séparé, de façon à vous y retrouver plus facilement. Si vous travaillez sur un projet où chaque personne doit rédiger une section, vous pouvez faire un fichier général qui ne contiendra que le préambule et des `\input` vers les différentes parties. Chaque personne pourra alors travailler parallèlement sur sa partie, sans qu'il y ait besoin de gérer les différentes versions du document final.

Mieux encore, le package `subfiles` est spécialement prévu pour le travail en groupe. Il fonctionne d'une façon similaire à `\input`, mais permet de compiler les sous-fichiers indépendamment pour avoir un aperçu.

fichier-principal.tex

```
\usepackage{subfiles}

\begin{document}

\subfile{fichier-secondaire1.tex}
\subfile{fichier-secondaire2.tex}
\subfile{fichier-secondaire3.tex}

\end{document}
```

fichier-secondaire1.tex

```
\documentclass[fichier-principal.tex]{subfiles}

\begin{document}

\section{La première section}
Contenu de la section.

\end{document}
```

10.3 Git, Subversion...

Ces logiciels sont beaucoup utilisés par les développeurs de logiciels pour travailler à plusieurs sur un seul code source. Ils sont tout à fait adaptés pour LaTeX et vous pouvez vous en servir. En revanche, si vous n'êtes pas informaticiens il y a peu de chances pour que vos collègues soient prêts à s'en servir...

11 Aller plus loin

Pour finir, voici quelques informations qui ne vous seront peut-être pas utiles tout de suite, mais qui pourront le devenir quand vous serez habitués à LaTeX.

11.1 Une table des matières incluse dans le pdf

Certains lecteurs de pdf permettent d'afficher un sommaire dans un panneau, de façon à pouvoir naviguer dans le document en cliquant sur les titres de sections. Pour cela, il faut d'abord que le sommaire en question soit inclu dans le pdf sous forme de *méta-données*. Avec LaTeX, il suffit d'utiliser le package `hyperref` avec l'option `hidelinks`.

```
\usepackage[hidelinks]{hyperref}
```

Le package `hyperref` sait aussi faire tout plein de bonnes choses, comme faire fonctionner les liens hypertextes, rendre cliquable la table des matières, etc. N'hésitez pas à l'utiliser.

11.2 Corriger les fautes

Pour corriger l'orthographe de votre document, il y a souvent un correcteur inclus dans votre éditeur. Mais ces correcteurs ne sont en général pas très avancés, et il peut être utile de vérifier votre document avec un outil plus élaboré. Pour cela, nous allons d'abord enlever tout le code LaTeX, pour ne garder que le texte.

Cela peut se faire facilement en utilisant un script comme `opendetex` :

code.google.com/p/opendetex

Écrivez « `detex votre_fichier` » dans un terminal pour obtenir le texte brut. Maintenant, vous pouvez chercher les fautes avec l'outil de votre choix :

www.reverso.net/orthographe/correcteur-francais

www.bonpatron.com

Vous pouvez même copier-coller votre texte dans Word, après vous être assuré que personne ne vous regarde.

11.3 Des commandes personnalisées

La commande `\newcommand` sert à créer vos propres commandes. C'est très pratique, car cela vous permet de remplacer une série de commandes compliquées par une seule commande simple. Elle prend deux arguments : d'abord le nom de votre nouvelle commande (n'oubliez pas de commencer par un `\`), et ensuite la série de commandes à remplacer. Par exemple, si vous n'avez pas envie de taper μ à chaque fois que vous écrivez une quantité en microlitres, vous pouvez créer une commande pour cela :

```
\newcommand{\ul}{ $\mu$ l }
```

Prélevez 5\ul de solution.

Prélevez 5 μ l de solution.

En fait, il est possible de programmer des commandes beaucoup plus complexes, vous trouverez tous les détails pour cela ici :

<https://en.wikibooks.org/wiki/LaTeX/Macros>

11.4 Créer des diaporamas

Il est possible de faire des diaporamas impeccables avec LaTeX, en utilisant la classe `beamer` (figure 3). Je ne vais pas m'étendre sur le sujet, l'Internet est déjà parsemé de tutoriels à ce propos.

Le principal avantage de Beamer, c'est qu'il permet de gagner énormément de temps quand on a un peu l'habitude de travailler avec LaTeX. Si vous savez quoi écrire, votre diaporama sera prêt en quelques minutes.

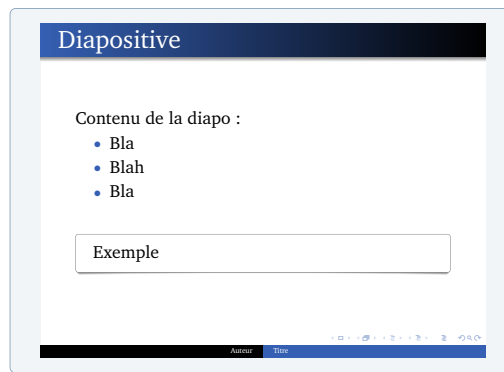


FIGURE 3 – Un exemple de diapositive créée avec Beamer.

`qpdfpresenterconsole`

Quand on utilise PowerPoint, il est possible de présenter un diaporama sur un projecteur, tout en affichant sur votre ordinateur un aperçu de la diapo suivante, un chronomètre, et d'autres éléments utiles. Le logiciel `qpdfpresenterconsole` permet de faire la même chose avec des diaporamas en pdf, ce qui en fait un outil génial si vous utilisez Beamer.

12 Améliorez ce guide

Ainsi s'achève cette introduction à LaTeX. Ce n'est qu'une introduction, pour que vous puissiez rapidement faire vos premiers projets sans faire n'importe quoi. Si vous voulez en savoir plus sur une commande ou une autre, vous pourrez trouver toutes sortes d'informations détaillées sur en.wikibooks.org/wiki/LaTeX (en anglais) ou fr.wikibooks.org/wiki/LaTeX (en français). N'oubliez pas que c'est à l'usage que vous progresserez, petite recherche google après petite recherche google, et dans quelques mois vous serez capables de faire à peu près n'importe quel document.

Encore une chose, ce guide est sous license *Creative Common*. Cela signifie que vous êtes libres de le redistribuer, de le faire tourner à vos amis et collègues et de le poster partout sur Internet.

Cela veut aussi dire que vous avez le droit de le modifier pour l'améliorer. Il suffit pour cela d'aller sur la page GitHub :

<https://github.com/Nenuphar/latex-guide-fr>

N'hésitez pas à faire des modifications, ne serait-ce que pour corriger des fautes d'orthographe. Il peut aussi être intéressant pour vous de jeter un œil au code source, pour avoir un exemple de vrai document LaTeX.

13 Aide-mémoire

Voici un document complet qui récapitule l'essentiel des commandes à connaître en LaTeX.

```
\documentclass[12pt]{article} % Article avec une police de 12 points
\usepackage[french]{babel} % Pour un document en français
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{graphicx} % Pour afficher des images

\title{Titre}
\author{Auteur}

\begin{document}

\maketitle % Titre
\clearpage % Saut de page

\section{Section 1}
\subsection{Sous-section 1}
Texte en \emph{italique}.\ \ % Pour passer une ligne

Note \footnote{Commentaire.} en pied de page.

\begin{center}
Texte centré.
\end{center}
Fin de la 1ère sous-section.

\vspace{14mm}
Il y a une espace de 14 mm entre ce texte et le paragraphe suivant.

\subsection{Sous-section 2}
\begin{itemize}
\item Liste classique
\end{itemize}

\begin{enumerate}
\item Liste numérotée
\end{enumerate}

\begin{description}
\item[Liste] de définitions
\end{description}

\section{Section 2}
Référence à la figure \ref{mon-image}.

\begin{figure}[h]
\includegraphics[width=0.4\linewidth]{image} % image.pdf, image.jpg, image.png...
\caption{Image occupant 40 \% de la largeur de la page.}
\label{mon-image} % Mot-clé correspondant à la figure.
\end{figure} % Remplacer figure par tableau pour un tableau.

\end{document}
```