

Petit tutoriel LaTeX pour le TIPE

MPSI Lycée Camille Jullian

4 juin 2023

LaTeX est un logiciel de traitement de texte « scientifique » créé par Leslie Lamport, qui n'est en fait qu'une amélioration du TeX initialement créé par Donald Knuth (une légende vivante de l'informatique, auteur entre autres du grand classique *The Art of computer programming*). Le principe de LaTeX est donc de pouvoir écrire des textes à la typographie soignée (notamment en ce qui concerne les formules mathématiques), mais il s'agit surtout d'un traitement de texte qui est **compilé**, contrairement aux logiciels plus classiques du type Word. Concrètement, quand on tape du texte avec Word, on interagit directement avec un affichage qui correspond exactement à ce que verra le lecteur du fichier final, et les modifications (de texte, de mise en page, etc) se font instantanément. Avec LaTeX, c'est très différent : on crée en fait un fichier texte dans lequel on va taper une liste de commandes (pas seulement le texte lui-même donc, mais aussi des indications de mise en page) à partir desquelles un programme va créer l'affichage final. De ce point de vue, le fonctionnement de LaTeX est en fait très proche de celui d'un langage de programmation comme Python : on lui dit ce qu'il y aura à faire, puis on exécute le programme pour avoir le résultat attendu... ou pas (si on fait des erreurs de syntaxe, comme en Python, le logiciel nous râlera dessus !). Mieux, LaTeX étant (comme Python) un logiciel libre, on peut créer soi-même de nouvelles commandes ou profiter de celles déjà programmées par d'autres utilisateurs plus compétents (l'équivalent des modules en Python, qu'on appelle plutôt **packages** en LaTeX).

Pour se lancer dans LaTeX.

Si vous souhaitez tenter de taper un document LaTeX sur les ordi du lycée, il vous suffit de lancer le programme TeXmaker, qui est un éditeur de texte pour LaTeX. Si vous voulez faire du LaTeX chez vous, il vous faudra installer **deux** programmes, une « distribution LaTeX » (en gros le programme qui maîtrise le langage et qui fera la compilation de ce que vous avez écrit et le traduira en un beau document pdf) et donc un éditeur de texte qui vous aidera à taper plus facilement vos documents (pour garder le parallèle avec Python, la distribution sera l'équivalent du Python que vous installez sur votre ordi, et l'éditeur est l'équivalent du Spyder ou Pyzo ou n'importe quel autre logiciel que vous utilisez pour disposer d'une interface sympathique quand vous tapez vos programmes). Les fichiers manipulés par cet éditeur sont de tout bêtes fichiers texte, affublés d'une extension .tex pour les distinguer des autres (mais vous pouvez les ouvrir sans problème avec votre Bloc-notes, ça ne contient que du texte !). L'avantage de la présence d'un éditeur spécifique est qu'il vous donnera accès à plein de raccourcis pour gagner du temps sur les commandes les plus fréquemment utilisées (cf ci-dessous, vous allez taper du `\section` ou du `\enumerate` suffisamment souvent pour ne pas avoir envie de le faire systématiquement à la main), ainsi qu'à un système de surlignage en couleur permettant de visualiser plus facilement ce que vous faites (voire de repérer rapidement des erreurs de syntaxe).

Préambule d'un document LaTeX.

Avant de taper quoi que ce soit dans votre document, il sera en pratique indispensable de donner quelques commandes générales (qui constituent ce que j'appelle préambule) pour indiquer à votre programme les lignes directrices de mise en page. Oui, même si vous n'en avez pas conscience, les règles générales en la matière diffèrent pas mal d'un pays à un autre, et il vaut mieux fixer les choses immédiatement pour que votre LaTeX préféré sache que votre document va contenir plein de caractères accentués du type « é » ou « à », voire même des « œ » mais probablement moins de « å » (mais, comme vous pouvez le constater, rien ne vous empêchera d'en mettre quand même dans vos documents!), qu'un nouveau chapitre s'appelle « Chapitre » et pas « Chapter », ou qu'il faut commencer chaque nouveau paragraphe par un alinéa de 2.37 cm, ni plus ni moins. En pratique, je vous conseille de mettre les quelques lignes suivantes au début de tous vos documents LaTeX :

```
\documentclass[a4paper,11pt]{article}  
\usepackage[latin1]{inputenc}  
\usepackage[T1]{fontenc}  
\usepackage[french]{babel}
```

Quelques petits commentaires : le backslash `\` débutant chaque ligne est tout simplement le symbole introduisant un nom de commande en LaTeX. Si jamais vous avez vraiment envie d'afficher un backslash dans votre texte, vous taperez donc très logiquement « `\backslash` ». La première ligne, avec le `documentclass`, est une introduction obligatoire qui précise le type de document que vous allez taper. Ce type de document est indiqué entre accolades, j'ai choisi par défaut le type **article** qui est le plus fréquent et vous suffira largement, mais il en existe d'autres comme **book** pour des documents plus conséquents. Les indications entre crochets sont optionnelles, celles que j'ai précisées donnent le format du document produit et la taille des caractères (testez avec 8pt ou 15pt au lieu de 11pt, vous verrez la différence).

Les trois lignes suivantes servent à importer divers packages (exactement comme les importations de modules en Python) pour gérer correctement l'encodage et les spécificités de mise en page en français. Les plus aventureux pourront remplacer l'option `latin1` de la deuxième ligne par `utf8` mais il y a encore parfois des problèmes avec les caractères accentués selon l'option choisie.

On peut ajouter à ce préambule des instructions pour faire un joli titre dans le document. Par exemple, pour le document que vous êtes en train de lire, j'ai ajouté les lignes suivantes :

```
\title{Petit tutoriel LaTeX pour le TIPE}  
\date{4 juin 2023}  
\author{MPSI Lycée Camille Jullian}
```

Je pense que vous comprendrez facilement les commandes en question. On peut indiquer dans la commande `date` un `\today` qui sera automatiquement remplacé par la date du jour à chaque nouvelle compilation du fichier. Une fois le préambule terminé, on commence réellement l'écriture du document.

Sectionnement et style.

Première chose à faire obligatoirement, indiquer le début du document par la commande « `\begin{document}` ». De même, on ferme le document par un « `\end{document}` ». Si on souhaite afficher un titre en début de document, on enchaîne immédiatement avec la commande « `\maketitle` », qui affichera un titre conforme aux instructions données dans le préambule. Si on souhaite afficher un résumé du document en-dessous du titre, on ajoutera dans le préambule une commande « `\abstract{Du blabla pour résumer le texte.}` ».

La structure du reste du document sera automatiquement gérée par LaTeX à partir de commandes permettant d'insérer un nouveau chapitre, une nouvelle partie, ou un nouveau paragraphe. En particulier, la numérotation de ces sections est complètement automatique (si on insère une nouvelle section au milieu du document, tout sera renuméroté correctement), tout comme l'est par exemple la numérotation des théorèmes et autres définitions dans mes polycopiés de cours. On peut même ajouter des commandes permettant de référencer un élément du type théorème indépendamment de sa numérotation (ainsi, si la numérotation est modifiée et qu'elle est citée ailleurs dans le texte, la modification y sera également effectuée). Pour ce que vous aurez à taper, vous ne devriez pas avoir besoin de plus de commandes que « `\section{Titre de la section}` » qui va automatiquement créer une nouvelle partie numérotée (si on ne veut **pas** de numérotation, on ajoute une `*` derrière `section`) avec un titre en gras correspondant à ce qu'on a indiqué entre accolades dans la commande, et « `\subsection{Titre}` » qui fonctionne de la même façon (avec numérotation interne à la section dont font partie les sous-sections). Pour ceux qui veulent vraiment des structures très détaillées, on peut faire des `\subsubsection` (numérotées), puis des `\paragraph` et des `\subparagraph` (non numérotés).

Comme déjà précisé plus haut, la mise en page du texte tapé au sein de ces différentes sections sera automatique, en particulier les alinéas en début de paragraphes (ici il s'agit de paragraphes au sens usuel du terme, présents même si on n'a pas utilisé la commande `\paragraph` qui les affublerait d'un titre). Si on souhaite commencer un nouveau paragraphe pour notre texte, il faut sauter une ligne dans le fichier texte (si on se contente de passer à la ligne, LaTeX continuera le même paragraphe). Il existe des commandes spécifiques pour ajuster l'écart vertical entre deux paragraphes, par exemple `\bigskip` qui crée un espace supplémentaire, mais aussi des commandes permettant de supprimer par exemple les alinéas : si on débute un nouveau paragraphe par un `\noindent`, il sera justifié à gauche comme le reste du texte, sans alinéa.

Commandes spécifiques, mode `maths`.

Puisque j'ai commencé à évoquer des commandes spécifiques à utiliser au sein du texte qu'on est en train de taper, soyons un peu plus précis sur ce sujet. Il existe des milliers de commandes en LaTeX, dont certaines ne sont disponibles qu'après avoir chargé un certain package. Ainsi, quand je fais des tableaux de variation dans mes corrigés, j'utilise un package spécial et des commandes toutes faites. De façon plus générale, tous les environnements spéciaux, comme les listes, les tableaux ou les figures, sont gérés par des commandes spéciales, à l'intérieur d'un cadre qui doit être délimité par un `\begin{environnement}` et un `\end{environnement}`. Je ne détaillerai pas ici tous les environnements disponibles, notamment les tableaux qui sont un peu plus complexes, mais les listes, qui seront probablement les seuls que vous aurez envie d'utiliser rapidement, sont très faciles à gérer. Une liste non numérotée est gérée en LaTeX par l'environnement **`enumerate`**, on crée un nouvel élément numéroté par la commande « `\item` ». Ainsi, si on tape les lignes de code suivantes :

```
\begin{enumerate}
\item J'aime le chocolat.
\item Et les frites.
\item Mais pas les deux mélangés.
\end{enumerate}
```

On obtiendra l'affichage suivant :

1. J'aime le chocolat.
2. Et les frites.
3. Mais pas les deux mélangés.

Si on souhaite faire des listes non numérotées (les items seront délimités par des tirets par défaut, mais on peut modifier cela facilement), on utilisera simplement l'environnement **`itemize`** au lieu d'**`enumerate`**. On utilise toujours la commande « `\item` » pour créer les nouveaux éléments.

Si on souhaite taper des formules mathématiques à l'intérieur du document, on a tout d'abord intérêt à ajouter dans le préambule l'importation de packages dédiés aux symboles mathématiques, notamment le package **amsmaths** (personnellement, j'ajoute systématiquement aussi les packages **amsfonts**, **amsthm** et **amssymb** mais j'avoue très franchement que je ne sais plus du tout ce qu'ils gèrent exactement). De plus, toutes les formules mathématiques seront délimitées par des symboles \$ (un en début de formule, un pour en marquer la fin), qui va modifier sensiblement l'affichage (essayez par exemple d'afficher du simple texte entre \$ et vous verrez que ce n'est pas la même police). La plupart des commandes mathématiques ne fonctionnent **que** dans cet environnement particulier (qu'on appelle en général « mode maths »). Quelques exemples de commandes mathématiques classiques (un élément à placer en indice sera toujours précédé d'un underscore, un élément à placer en exposant d'un accent circonflexe) : `\frac{a}{b}` qui affichera $\frac{a}{b}$, `\sqrt[n]{x}` qui affichera $\sqrt[n]{x}$, `\sum_{i=1}^n a_i` qui affichera $\sum_{i=1}^n a_i$, `\int_0^1 f(x) dx` qui affichera $\int_0^1 f(x)dx$, ou encore `\overrightarrow{u}` qui affichera \overrightarrow{u} . Comme la liste des commandes est bien trop étendue pour que j'en fasse le tour, deux documents sur Internet qui peuvent vous servir de référence : la version courte devrait être largement suffisante

<https://www.cmor-faculty.rice.edu/~heinken/latex/symbols.pdf>

Mais pour les plus intéressés, voici la version complète :

<https://tug.ctan.org/info/symbols/comprehensive/symbols-a4.pdf>

Ah, une dernière remarque que j'ai oubliée : si on veut afficher du texte en gras ou en italique (ou souligné ou n'importe quoi d'autre), il existe bien sûr des commandes spécifiques, notamment `\textit{Texte à mettre en italique}` et `\textbf{Texte à mettre en gras}`.

Bibliographie en LaTeX.

Il existe des commandes spécifiques pour gérer une bibliographie dans un document LaTeX. La version simple consiste à ajouter dans le document un environnement **thebibliography** (introduit comme d'habitude par un `begin` et terminé par un `end`) à l'intérieur duquel les différents éléments seront introduits par la commande `\bibitem{Hippopotame}` et numérotés automatiquement (entre crochets). Le nom (ici « Hippopotame ») donné à la référence est celui qui sera utilisé quand on voudra citer la référence à l'intérieur du document. Il suffira de placer dans le texte un `\cite{Hippopotame}` pour faire apparaître dans le texte le numéro de la référence correspondante. Bien sûr, à l'intérieur du `bibitem` de chaque référence, on indiquera les éléments souhaités (nom des auteurs, titre, éditeur, etc) avec le style qu'on veut (italique, gras), en veillant bien à rester cohérent dans toutes les références créées.