

# Mini-guide d'introduction à LaTeX

Durant ce semestre, il y aura des exercices à rendre en format PDF, obtenus à partir de "LaTeX", qui est un langage permettant de rédiger des documents mathématiques de façon très commode et élégante. Dans ce mini-guide, nous présentons quelques fonctionnalités de base afin que vous puissiez rédiger vos premiers documents en LaTeX !

## 1 Logiciels et plateformes pour LaTeX

Commençons par expliquer comment on peut installer LaTeX. Il y a deux possibilités :

- Télécharger et installer une distribution TeX ainsi qu'un éditeur (logiciel) LaTeX. Pour la distribution, nous conseillons [MikTeX](#) pour Windows, [TeXLive](#) pour Linux, ou [MacTeX](#) pour MacOSX. Pour le logiciel, nous pouvons conseiller TexMaker qui est disponible gratuitement à la fois sur Mac OS X, Windows et Linux : voir [leur site web](#). (Vous pouvez utiliser tout autre éditeur qui vous conviendrait mieux, comme TeXworks).
- L'autre possibilité est d'utiliser une plateforme en ligne, comme [OverLeaf](#). L'inscription est gratuite, et vous permet de créer des projets et des fichiers LaTeX depuis votre ordinateur ou votre tablette (ou même depuis votre téléphone portable).

Dans les deux cas, le principe est le même. On écrit le "code" LaTeX, et on peut à tout moment le compiler, ce qui va automatiquement générer un fichier PDF qui donne le résultat du code.

En fait, la compilation va générer d'autres fichiers accessoires (comme ".aux", etc.). Vous n'avez pas à vous en soucier : seuls les fichiers ".tex" et ".pdf" vous intéressent. (Toutefois, il peut être recommandé de créer un dossier pour chaque projet LaTeX que vous allez écrire, afin de rassembler dans un seul et même dossier le ".tex", ".pdf" et ces documents accessoires créés lors de la compilation).

## 2 Comment écrire un document en LaTeX ?

### 2.1 Principe général

Vous trouverez sur Moodle un *template* que vous pourrez utiliser pour écrire vos rendus d'exercices.

En haut de ce document, vous verrez ceci :

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
\documentclass[11pt, a4paper, twoside]{article}  
\usepackage[utf8]{inputenc}  
\usepackage[T1]{fontenc}  
\usepackage[français]{babel}  
\usepackage{lmodern}  
  
\usepackage{amsmath}  
\usepackage{amssymb}  
\usepackage{amsthm}  
  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Il s'agit de "packages" nécessaires au bon fonctionnement de la compilation et vous permettant d'inclure plein de symboles mathématiques dans votre rédaction. Mais peu importe : vous n'avez pas besoin de modifier quoi que ce soit de ce préambule, ni de comprendre ce que ces *packages* veulent dire.

C'est dans la partie suivante que nous allons rédiger notre texte :

```
\begin{document}
\title{Série 1}
\author{Évariste Galois}
\maketitle

%[La résolution des exercices ici...] %commentaire

\end{document}
```

Il y a d'abord "`\begin{document}`", puis trois lignes qui permettent d'afficher un titre, votre nom, et la date d'aujourd'hui.

En-dessous de ces quelques lignes (mais avant le "`\end{document}`"), vous pouvez taper n'importe quel texte, et dès que vous cliquerez sur "compilation" (soit dans le logiciel LaTeX, soit sur la plateforme en ligne), votre texte s'affichera sur le PDF ! C'est aussi simple que cela. Notez que dans l'image ci-dessus, le passage en gris est un commentaire : il suffit d'utiliser le symbole % pour mettre un passage en commentaire dans votre fichier, c'est-à-dire qu'il n'apparaîtra pas dans le PDF final.

Ainsi, si vous compilez le template donné sur Moodle, sans rien modifier, votre fichier PDF devrait ressembler à ceci (mis à part la date qui sera différente) :

Série 1  
Évariste Galois  
12 septembre 2020

1

## 2.2 Écrire des équations

Il sera très utile de savoir comment on écrit des symboles mathématiques en  $\text{\LaTeX}$ . De façon générale, si on veut écrire une équation dans un texte, comme  $x = 2$ , il suffit d'encadrer l'équation entre deux symboles \$. Ainsi, `$y_1 = 2^5$` produit l'équation  $y_1 = 2^5$ .

Si on veut au contraire écrire une équation centrée à la ligne, il existe plusieurs possibilités :

- On peut utiliser des "double dollars" : par exemple, écrire `$$x + 2 = 8$$` produit l'équation

$$x + 2 = 8$$

à la ligne.

- Il est possible d'utiliser des crochets avec un backslash devant. Si on écrit `\[ x + 2 < -3 \]`, cela produit l'inéquation

$$x + 2 < -3$$

à la ligne.

- On peut utiliser l'environnement `"align"`, qui est très pratique surtout lorsqu'il y a une suite de plusieurs équations à la ligne (mais il fonctionne aussi avec une seule équation bien sûr). Il permet en plus de numéroter les équations automatiquement.

Par exemple, le code

```
\begin{align}
  a &= b && \\
    &= c && \\
    &= d+2 && \\
\end{align}
```

va donner le résultat

$$a = b \tag{1}$$

$$= c \tag{2}$$

$$= d + 2 \tag{3}$$

Le symbole `"&"` permet d'aligner les équations correctement (ici elles sont alignées avec le signe égal), et les double backslash `\\` indiquent un retour à la ligne.

On peut également faire un lien vers une équation numérotée, lorsqu'on utilise l'environnement `"align"`. D'abord, on indique à l'aide de la commande `\label{...}` comment l'on veut nommer l'équation :

```
\begin{align}
  x^2 &= 2 && \label{eq1} \\
  x &= \pm \sqrt{2} && \label{eq2} \\
\end{align}
```

va simplement produire

$$x^2 = 2 \tag{4}$$

$$x = \pm\sqrt{2} \tag{5}$$

(c'est-à-dire que les `"labels"` sont juste pour nous en interne, ils ne s'affichent pas).

Ensuite, on utilise `\eqref{...}` dans la suite du texte, pour que le numéro de l'équation s'affiche :

L'équation `\eqref{eq1}` implique `\eqref{eq2}` parce que...

va afficher le texte « L'équation (4) implique (5) parce que... ». (Il faut parfois compiler deux fois de suite pour mettre à jour les numéros : la première fois qu'on compile, il y aura des points d'interrogations `??` au lieu des numéros).

Signalons ici quelques problèmes qui peuvent surgir lors de la compilation. Par exemple, imaginons que vous écrivez une formule avec un dollar ouvrant mais vous oubliez d'insérer un dollar fermant (typiquement vous écrivez `"...donc $x = 2, ce qui implique..."`). Lorsque vous compilerez, vous obtiendrez une "erreur", vous indiquant la ligne où se produit l'erreur (en l'occurrence `Missing $ inserted`). Il ne faut pas s'inquiéter lorsqu'une erreur se produit, il s'agit parfois d'un simple oubli, ou alors d'un problème plus délicat au sujet duquel les assistants pourront vous aider si besoin. Enfin, le compilateur produit parfois des `"warnings"`, mais vous pouvez les ignorer complètement.

## 2.3 Symboles mathématiques

Vous savez déjà les principaux principes de  $\text{\LaTeX}$ . Il nous reste à discuter de la façon dont sont implémentés les différents symboles mathématiques.

Nous donnons ci-dessous une liste non-exhaustive, mais qui contient la plupart des symboles qui vous seront utiles.

Il n'est pas nécessaire de les apprendre par coeur ! En effet, certains éditeurs possèdent des listes de symboles, sur lesquels on peut cliquer afin que la commande correspondante s'écrive directement (dans  $\text{\TeX}$ Maker, ces listes sont visibles en cliquant sur "Structure" en bas à gauche, il y a alors des icônes tout à gauche qui rassemblent des familles de symboles). Par ailleurs, [ce site Internet](#) est très utile pour identifier une commande à partir d'un dessin du symbole.

Ceci dit, avec de la pratique et de l'expérience, vous finirez par connaître ces commandes sur le bout de vos doigts.

<code>\to</code>	<code>\mapsto</code>	<code>\implies</code>	<code>\iff</code>	<code>\in</code>	<code>\subset</code>	<code>\not \in</code>	<code>\cup</code>	<code>\cap</code>	<code>\vee</code>	<code>\perp</code>	<code>f \circ g</code>
$\rightarrow$	$\mapsto$	$\implies$	$\iff$	$\in$	$\subset$	$\notin$	$\cup$	$\cap$	$\vee$	$\perp$	$f \circ g$
<code>\bigcap_{i=1}^n</code>	<code>\bigcup_{x \in X}</code>	<code>\sum_{n=0}^9</code>	<code>\geq</code>	<code>\leq</code>	<code>&lt;</code>	<code>&gt;</code>	<code>\cong</code>	<code>\blacksquare</code>			
$\bigcap_{i=1}^n$	$\bigcup_{x \in X}$	$\sum_{n=0}^9$	$\geq$	$\leq$	$<$	$>$	$\cong$	$\blacksquare$			
<code>\{ \}</code>	<code>[...]</code>	<code>\frac{1}{3}</code>	<code>\dfrac{\pi}{4}</code>	<code>\pm</code>	<code>\lim_{x \to 0}</code>	<code>\lfloor x \rfloor</code>	<code>\times</code>				
$\{ \}$	$[...]$	$\frac{1}{3}$	$\frac{\pi}{4}$	$\pm$	$\lim_{x \rightarrow 0}$	$\lfloor x \rfloor$	$\times$				
<code>\int_{-1}^1</code>	<code>\sqrt[3]{2}</code>	<code>\mathbb{N}</code>	<code>\mathbb{R}</code>	<code>\alpha</code>	<code>\beta</code>	<code>\gamma</code>	<code>\delta</code>				
$\int_{-1}^1$	$\sqrt[3]{2}$	$\mathbb{N}$	$\mathbb{R}$	$\alpha$	$\beta$	$\gamma$	$\delta$				
<code>\Delta</code>	<code>\epsilon</code>	<code>\phi</code>	<code>\chi</code>	<code>\psi</code>	<code>\omega</code>	<code>\Omega</code>	<code>\infty</code>	<code>\overline{a}</code>	<code>\tilde{f}</code>		
$\Delta$	$\epsilon$	$\phi$	$\chi$	$\psi$	$\omega$	$\Omega$	$\infty$	$\bar{a}$	$\tilde{f}$		
<code>\vert</code>	<code>\forall</code>	<code>\exists</code>	<code>\cdot</code>	<code>\varnothing</code>	<code>\star</code>	<code>\equiv</code>	<code>\sim</code>	<code>\leftrightarrow</code>			
$ $	$\forall$	$\exists$	$\cdot$	$\emptyset$	$\star$	$\equiv$	$\sim$	$\leftrightarrow$			

Par exemple, la formule

$$\forall n \geq 1, \quad \sum_{k=1}^n k = \frac{k \cdot (k+1)}{2} \in \mathbb{N}$$

s'obtient avec le code

```
\[ \forall n \geq 1, \quad \sum_{k=1}^n k = \dfrac{k \cdot (k+1)}{2} \in \mathbb{N} \]
```

(Ici, la commande `\quad` permet d'obtenir un espace horizontal).

Pour écrire plusieurs symboles (y compris des nombres avec plus que 2 chiffres) en exposant ou en indice, il faut utiliser des accolades. Par exemple,  $x_{n_k}^{20}$  s'obtient à l'aide du code  `$\$x_{n_k}^{20}\$$` .

Signalons également que la plupart des fonctions classiques en mathématiques, comme  $\sin$ ,  $\cos$ ,  $\tan$  ou encore  $\min$ ,  $\inf$ ,  $\deg$ , s'obtiennent simplement à l'aide des commandes `\sin`, `\cos`, `\tan`, `\min`, `\inf`, `\deg`, etc.

Voici quelques remarques **facultatives** :

- Certains symboles s'affichent un peu différemment selon qu'ils sont *inline* (dans un texte), ou dans une équation centrée à la ligne. Par exemple, on constate la différence entre  $\sum_{i=0}^n$  et

$$\sum_{i=0}^n$$

que donnent les codes `\sum_{i=0}^n` et `$$\sum_{i=0}^n$$` respectivement.

- On peut signaler aussi le fait que des commandes du type `\left\{ \right\}` permettent d'avoir des accolades qui s'adaptent au contenu : comparer le résultat de `\left\{ \dfrac{1}{3} \right\}` et de `\left\{ \dfrac{1}{3} \right\}` qui donnent respectivement  $\left\{ \frac{1}{3} \right\}$  et  $\left\{ \frac{1}{3} \right\}$ . La même remarque est valide pour `\left[ \dots \right]`.
- *Si vous le souhaitez*, vous pouvez abrégier certaines commandes en leur donnant des noms personnalisés. Par exemple, si vous écrivez (plutôt avant le `\begin{document}`, c'est-à-dire à la fin du préambule des *packages*) la commande

```
\newcommand{\R}{\mathbb{R}}
```

alors cela vous permettra d'écrire `$ x \in \R $` qui donnera  $x \in \mathbb{R}$ . (La commande `\mathbb{R}` va toujours donner  $\mathbb{R}$  comme résultat, on a juste une nouvelle façon plus courte d'écrire  $\mathbb{R}$ ).

De plus, on peut créer de nouveaux opérateurs. Par exemple, si l'on écrit la commande `\DeclareMathOperator{\Sym}{Sym}` (à la fin du préambule des *packages*), alors on pourra utiliser `\Sym(X)` dans le document pour afficher  $\text{Sym}(X)$  (comparer avec `\Sym(X)` qui donne  $\text{Sym}(X)$ ).

On mentionne ces manoeuvres ici car cela peut être pratique, mais il n'est ni nécessaire, ni exigé que cette notion soit maîtrisée — c'est juste pour votre information.

- Il y a plein d'autres choses que vous pouvez faire avec  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  que nous n'avons pas présentées ici ! Par exemple, on peut écrire des matrices (avec l'environnement *pmatrix*), créer des tableaux (avec l'environnement *tabular*), faire des distinctions de cas<sup>1</sup>, inclure des images (avec la commande `\includegraphics[scale=0.5]{filename}`), etc.

## 2.4 Autres notions typographiques

Si vous souhaitez créer des listes, il y a au moins deux possibilités. La première est d'utiliser l'environnement *itemize* : par exemple, le code

```
\begin{itemize}
\item Mon premier truc
\item Mon deuxième truc
\end{itemize}
```

va afficher

- Mon premier truc
- Mon deuxième truc

---

<sup>1</sup>On entend par là des formules du style  $f(x) = \begin{cases} 0 & \text{si } x > 0 \\ 1 & \text{sinon} \end{cases}$ . Cela se crée avec l'environnement *cases*.

Une deuxième possibilité est d'utiliser l'environnement *enumerate* : par exemple, le code

```
\begin{enumerate}
\item Si  $f = 0$ , alors ...
\item Si  $f > 0$ , alors ...
\end{enumerate}
```

va afficher

1. Si  $f = 0$ , alors ...
2. Si  $f > 0$ , alors ...

Par ailleurs, pour écrire en *italique*, on peut utiliser `\textit{...}`, et pour écrire en **gras**, on peut utiliser `\textbf{...}`. De plus, pour centrer un texte, on peut utiliser l'environnement *center*, c'est-à-dire

```
\begin{center}
...
\end{center}
```

Enfin, si vous souhaitez créer des sections (numérotées), vous pouvez utiliser la commande `\section{Mon titre}`. De même, la commande `\subsection{Mon sous-titre}` permet de créer une sous-section.

## 2.5 Autres questions et références

Les assistants seront à votre disposition durant les séances d'exercices pour répondre à toute question concernant LaTeX. Par ailleurs, de nombreux sites proposent de l'aide et de la documentation supplémentaires :

- Cette "[cheat-sheet](#)" rassemble les symboles mathématiques les plus fréquemment employés.
- Un aide-mémoire très complet et assez avancé se trouve [ici](#).
- De nombreuses autres fonctionnalités peuvent s'obtenir en ajoutant de nouveaux "packages", que l'on peut télécharger sur [ce site](#). On peut ainsi obtenir des documents personnalisés à souhait (mais bien sûr, cela ne sera aucunement nécessaire pour ce cours).
- Plein d'autres références (en anglais) sont disponibles [ici](#).
- Enfin, [ce forum](#) est également très fourni en informations et questions de toutes sortes.