

Guide d'utilisation de \LaTeX (version Lite)

Le KI'022



Préambule

Ce document a pour but de présenter les fonctionnalités de base et les plus utiles de LaTeX, ce langage étant devenu la norme pour produire des documents scientifiques aujourd'hui. Si vous ne connaissez absolument rien à ce langage pas de panique ! Ce guide est fait pour vous, il vous donnera les bases qui vous permettront d'écrire tous vos rapports et autres pendant vos années aux Ponts et Chaussées.

Mais d'abord je vais essayer de répondre à la question "A quoi sert LaTeX ?" où plutôt : "Pourquoi me donner du mal à apprendre un langage compliqué si je peux faire plus ou moins la même chose avec Word ?". Le premier truc qu'on vous dira, c'est que LaTeX permet de faire des jolies maths, et c'est probablement ce qui vous servira le plus mais il y a d'autres manières de répondre à cette question :

- LaTeX vous permet de vous concentrer uniquement sur le **contenu** de votre document et non sur la forme qui est gérée automatiquement.
- LaTeX demande un certain temps d'adaptation, surtout si vous êtes habitués à Word mais les documents produits sont de plus grande qualité et font plus "scientifique".
- Sans parler des maths, LaTeX procure une grande facilité pour ce qui est de référencer ses figures, ses graphiques, faire une bibliographie ou une table des matières. Il sera plus facile et surtout plus rapide de produire un rapport scientifique avec LaTeX car tout ce que j'ai cité se fera (presque) tout seul.

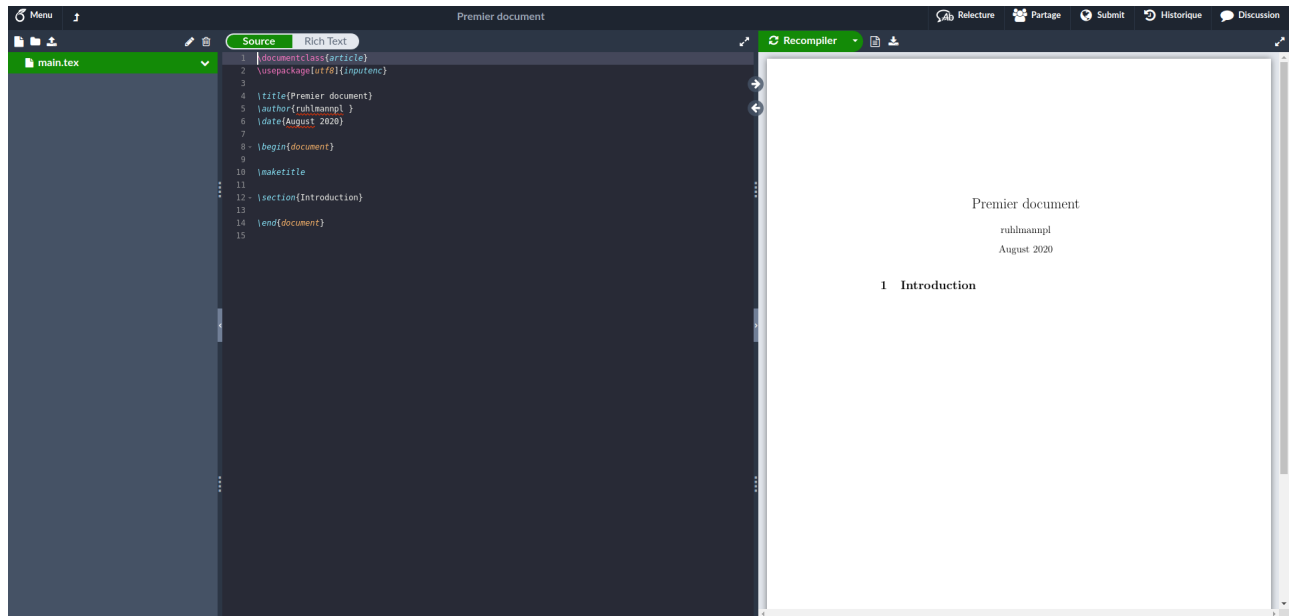
Bonne lecture !

Commencer L^AT_EX rapidement

L'installation

Avant d'écrire un document LaTeX, il vous faut une distribution de LaTeX. Le plus simple, rapide et efficace est d'utiliser le site *Overleaf*. Il suffit de se connecter et de créer un nouveau document et... c'est tout. Il n'y a pas plus simple, le seul désavantage¹ étant que vous allez avoir besoin d'une connexion Internet, ce qui sera rarement un soucis.

Pour commencer un nouveau document il vous suffit de cliquer sur "Nouveau projet" → "Projet vide" et vous devriez tomber sur quelque chose qui ressemble à l'image ci-dessous.



Pour le moment, effacez tout le contenu de la page de gauche, nous allons repartir de zéro pour cette fois. Appuyez ensuite sur le bouton "Recompiler" et vous devriez avoir une page blanche sur la droite. Voilà une des principales caractéristiques de LaTeX, contrairement à Word ou d'autres logiciels de traitements de texte, ce n'est pas du temps réel. C'est à dire que vous allez écrire du code et ensuite compiler ce code pour obtenir un document PDF.

Je vous conseille de souvent compiler votre code pour vous rendre compte de ce que vous écrivez, vous pouvez le faire soit en appuyant sur le bouton "Recompiler" ou avec les touches Ctrl+Enter.

Un autre avantage d'Overleaf est qu'il vous permettra de travailler en collaboration sur un même document, chose impossible sur un éditeur classique. Vous trouverez dans le guide complet des explications pour ceux qui voudraient avoir une distribution de LaTeX sur leur PC (Mac Linux et Windows).

1. Il y en a d'autres, mais peu conséquents par rapport à la facilité d'utilisation procurée par Overleaf

Les bases d'un document L^AT_EX

Pour débiter un document L^AT_EX il faut utiliser un certain nombre de commandes et inclure des paquets². Voici ce à quoi chacun de vos documents devrait ressembler (commandes obligatoires en rouge et obligatoires pour un document en français en bleu) :

```
\documentclass[options]{type}
\usepackage[utf8]{inputenc}
\usepackage[french]{babel}
\usepackage[T1]{fontenc}

\usepackage[options]{nom}

\title{titre}
\author{auteur}
\date{date}

\begin{document}
\maketitle
\tableofcontents

\section{titre de la section}  (\section* pour éviter la numérotation)
\subsection{titre de la sous-section}
Votre texte

\subsection{titre de la sous-section}
Votre texte

\end{document}
```

Vous pouvez déjà essayer de reproduire ce schéma (sans le `\usepackage` en noir pour le moment) avec `\documentclass[11pt]{article}`, remplacez juste ce qui est en italique par le titre, auteur, date..., appuyez sur Ctrl+Enter et vous devriez avoir un document qui ressemble à ceci :

Un premier document		
Le KI		
4 août 2020		
Table des matières		
1	Première section	1
1.1	Première sous-section	1
1.2	Deuxième sous-section	1
1 Première section		
1.1 Première sous-section		
	Du texte	
1.2 Deuxième sous-section		
	Du texte	

Pas mal non ? En quelques lignes vous avez déjà un document prêt à être rendu.

2. Le template fourni lors de la formation regroupe les paquets les plus utilisés en pratique.

→ Vous avez d'autres commandes pour faire des sections, comme `\part` `\chapter` `\subsubsection` mais certaines sont spécifiques à certaines classes (ici on utilise la classe article, voir le guide complet pour en savoir plus sur les classes).

Les commandes classiques de mise en page

Pour modifier le texte, il y a les commandes :

Commande	Rendu	Raccourci	→ Selon les éditeurs (Overleaf ou autre), il peut y avoir plusieurs raccourcis que ceux-ci, néanmoins il est toujours possible de les utiliser façon "Word", c'est à dire sélectionner le texte puis appuyer sur Ctrl+I pour le mettre en italique par exemple .
<code>\textbf{<i>texte</i>}</code>	→ texte	Ctrl + B	
<code>\textit{<i>texte</i>}</code>	→ <i>texte</i>	Ctrl + I	
<code>\texttt{<i>texte</i>}</code>	→ texte		
<code>\underline{<i>texte</i>}</code>	→ <u>texte</u>		
<code>\emph{<i>texte</i>}</code>	→ <i>texte</i>		
<code>\textsc{<i>texte</i>}</code>	→ TEXTE		
<code>\fbox{<i>texte</i>}</code>	→ texte		

Les listes à puces (respectivement numérotées) se font avec les commandes suivantes :

<code>\begin{itemize}</code>	ou	<code>\begin{enumerate}</code>	→ Pour des listes avec des • plutôt que des tirets, mettre la commande <code>\frenchbsetup{StandardLists=true}</code> en début de document (avant le <code>\begin{document}</code>). On peut également "empiler" plusieurs listes en cascade, elles seront indentées au fur et à mesure.
<code>\item <i>texte</i></code>		<code>\item <i>texte</i></code>	
<code>\item <i>texte</i></code>		<code>\item <i>texte</i></code>	
<code>\end{itemize}</code>		<code>\end{enumerate}</code>	

→ En LaTeX, le `\begin{}` suivi d'un `\end{}` s'appelle un environnement, cela permet de formater ce que vous allez mettre entre ces deux commandes. **Faire attention à bien mettre le `\begin` et le `\end` sinon ça ne marchera pas !**

Les citations se font de la façon suivante :

```
\begin{quote}
texte
\end{quote}
```

La commande `\footnote{texte}` crée une note en bas de page.

La position du texte sur la page se modifie avec les environnements `flushleft`, `center`, `flushright`. C'est certainement le 2e qui vous sera le plus utile :

Et pour des citations de plusieurs lignes :

```
\begin{quotation}
texte
\end{quotation}
```

```
\begin{center}
texte
\end{center}
```

Pour les sauts de lignes et paragraphes :

Vous l'avez peut-être remarqué mais en LaTeX, pour revenir à la ligne vous ne pouvez pas juste appuyer sur la touche **Enter**, il vous faut soit mettre un double backslash `\\` à la fin de votre ligne ou utiliser la commande `\newline`.

Vous pouvez aussi appuyer deux fois sur **Enter** mais cela créera un nouveau paragraphe, c'est à dire qu'il sera indenté par défaut (décalé vers la droite). Pour éviter cela il suffit de mettre `\noindent` en début de ligne ou, pour le faire pour tout le document, mettre la commande `\setlength{\parindent}{0pt}`

en début de document (avant le `\begin{document}`).

Si vous voulez sauter une ligne, il faut alors mettre un `\\` à la fin de votre ligne puis laisser une ligne blanche comme pour faire un nouveau paragraphe.

Pareillement, les espaces ne se font pas en appuyant 40 fois sur la touche espace, il faut utiliser des commandes telles que `\quad` et `\qquad`.

→ Je vous conseil d'utiliser le package `geometry` suivi de la commande

`\geometry{top=1.5cm, bottom=2cm, left=2.5cm, right=2.5cm}`, cela vous permettra de gérer vos marges en faisant varier les valeurs `top`, `bottom`, `left` ou `right`.

Les images, tableaux et autres flottants

Un des points les plus importants dans un document est l'inclusion d'images, leur référencement et les disposer intelligemment pour ne pas perturber la lecture. Sur ce point, il peut être très compliqué de vouloir faire exactement ce que l'on a en tête en LaTeX car contrairement aux logiciels Office, on ne peut pas déplacer l'image avec la souris. Mais pour la plupart de vos documents, LaTeX va très bien gérer votre placement d'images.

Commençons par le plus basique, le `\includegraphics[options]{chemin d'accès à l'image}`. Disons que vous ayez une image `image.png` dans le même dossier que votre fichier `.tex` (sur Overleaf vous pouvez directement importer vos images avec un petit bouton en haut à gauche). Pour mettre cette image dans votre document il y a juste besoin d'écrire deux choses :

- `\usepackage{graphicx}` en début de document.
- `\includegraphics[scale=x]{image.png}` (avec x le facteur d'échelle entre 0 et 1) là où vous voulez insérer votre image.

Bien sur cette commande fonctionne dans les environnements cités ci-dessus, par exemple si vous voulez centrer votre image :

<code>\begin{center}</code>	→ D'autres paramètres sont possibles plutôt que
<code>\includegraphics[scale=x]{image.png}</code>	le <code>scale</code> , voir le guide complet pour en savoir plus.
<code>\end{center}</code>	

Les flottants

La méthode présentée ci-dessus permet de dire à LaTeX précisément où doivent être insérées vos images. Mais il peut être parfois plus judicieux de laisser LaTeX organiser vos images, pour cela on utilise des flottants (non ce n'est pas pareil que des flottants en info...). Les flottants tiennent leur nom du fait que LaTeX va les déplacer à sa convenance pour faire le meilleur document possible (selon lui, libre à vous de changer ensuite).

Prenons l'exemple de l'image précédente, nous allons maintenant la mettre dans un environnement supplémentaire nommé **figure** :

```
\begin{figure}[h]
\centering (Equivalent à un center)
\includegraphics[scale=x]{image.png}
\caption{légende}
\end{figure}
```

→ Sans le `\centering`, l'image sera à gauche.

→ L'argument "h" indique que l'on veut que notre image soit ici ("here"), c'est à dire là où on écrit le texte. On peut aussi mettre b ou t pour "top" et "bottom" par exemple, ainsi notre image se retrouvera en haut ou en bas de la page peut importe où l'on a écrit la commande.



FIGURE 1 – Le logo du KI

Voici une image incorporée dans un document avec les commandes ci-dessus. Il y a plein de manière de modifier la légende et l'image à votre convenance, comme l'encadrer, l'inclure dans du texte (package `wrapfig`) et d'autres. Plus de détails sont donnés dans le guide complet.

→ Attention cependant, LaTeX peut décider de ne pas respecter votre argument entre crochet [] si cela conduit l'image à déborder de la page ou à trop éloigner le titre d'une section de son texte.

Un autre exemple de flottants : **les tableaux**, ceux-ci ont une syntaxe bien précise mais il existe certains sites comme [tablesgenerator](#) qui vous simplifie la vie, vous avez juste à rentrer vos données et il vous génère le code automatiquement.

Ecrire des mathématiques

On aborde ici une grosse partie de ce que vous allez faire en LaTeX, au début cela va être un peu laborieux, car vous passerez votre temps à regarder la magnifique fiche mémo faite par le KI (disponible [ici](#)) avec entre autre les commandes mathématiques. Mais ces commandes sont assez naturelles et vous les mémoriserez assez vite.

⇒ Avant toute chose, assurez vous d'avoir mis `\usepackage{amsmath}` en préambule.

Pour écrire des maths, c'est très simple, il suffit de se placer dans un environnement mathématique, obtenu avec `\(\)` ou `\[\]`.

Ainsi le code suivant :

Le théorème de Pythagore `\(x^2 + y^2 = z^2 \)` a été prouvé faux pour des puissances supérieures. Cela veut dire que l'équation suivante n'a pas de solutions entières : `\[x^n + y^n = z^n \]`

Devient sur le pdf :

Le théorème de Pythagore $x^2 + y^2 = z^2$ a été prouvé faux pour des puissances supérieures. Cela veut dire que l'équation suivante n'a pas de solutions entières :

$$x^n + y^n = z^n$$

Observez la différences entre les deux, le `\(\)` permet d'écrire des maths sur la même ligne que le texte en cours alors que `\[\]` centre vos maths en sautant une ligne (**inline** vs **display** mode). Vous pouvez aussi utiliser le \$ plutôt que `\(` et `\)` pour un environnement inline.

Si vous voulez numéroté vos équations alors il faut utiliser l'environnement `equation` :

$$x^2 + y^2 = z^2 \tag{1}$$

Vous pouvez inclure toutes sortes de symboles dans un environnement mathématique comme des lettres grecques α β γ , des signes logiques et quantificateurs \forall \exists \cup \sum ainsi que les notations ensemblistes $\mathbb{R}\mathbb{Q}\mathbb{S}\mathbb{Z}$, \mathcal{CBTL} (requiert le package `amssymb`) La plupart de ces symboles sont répertoriés dans la fiche mémo distribuée lors de la formation ou disponible sur [ce site](#).

Puissance et indice

Pour écrire des puissances et des indices, il faut utiliser `^` et `_`.

Le code suivant :

`\[a_1^2 + a_2^2 = a_3^2 \]`

En utilisant des `{ }` lorsque les exposants/indices ne sont pas réduits à un seul symbole :

`\[x^{2 \alpha} - 1 = y_{ij} + y_{ji} \]`

Plus compliqué :

`\[\sum_{i=1}^{\infty} \frac{1}{n^s} = \prod_p \frac{1}{1 - p^{-s}} \]`

Devient sur le pdf :

$$a_1^2 + a_2^2 = a_3^2$$

$$x^{2\alpha} - 1 = y_{ij} + y_{ji}$$

$$\sum_{i=1}^{\infty} \frac{1}{n^s} = \prod_p \frac{1}{1 - p^{-s}}$$

Parenthèses et accolades

Vous aurez souvent besoin d'écrire des normes, des produits scalaires ou même d'encadrer une grosse expression entre parenthèse. LaTeX est fait pour gérer tout ça, j'ai listé ici quelques exemples :

Le code suivant :

`\{x+y\}`

`\langle x+y \rangle`

`\| x+y \|`

`\left(\frac{m_1 m_2}{r^2} \right)`

Devient sur le pdf :

$\{x + y\}$

$\langle x + y \rangle$

$\|x + y\|$

$F = G \left(\frac{m_1 m_2}{r^2} \right)$

Les commandes `\left` et `\right` permettent d'ajuster la taille de vos parenthèses (marche avec `[] {} ...`) à l'expression entre les deux. Ces commandes sont indissociables, l'une ne peut venir sans l'autre, ainsi si vous voulez juste mettre une parenthèse à gauche il vous faudra mettre un `\right`. (le point est inclus) à la fin de votre expression et un `\left(` au début.

→ Si jamais `\left` et `\right` ne donnent pas le résultat souhaité, vous pouvez essayer `\big` `\Big` `\bigg` `\Bigg` (ordre de taille croissante).

Espaces en mode math

En mode math (c'est à dire entre `$ $` ou `\[\]`) on ne peut pas espacer les caractères, symboles ou expression en appuyant simplement sur la touche espace, LaTeX fera comme si tout fait partie d'un même mot. Pour manuellement mettre un espace dans une expression il vous faudra donc utiliser un des symboles suivants :

Commande	Code	Rendu
<code>\!</code>	<code>\$x^2\! +3x\! +2\$</code>	$x^2 + 3x + 2$
Espace	<code>\$x^2 + 3 x+ 2\$</code>	$x^2 + 3x + 2$
<code>\,</code>	<code>\$x^2\,, +3x\,, +2\$</code>	$x^2 + 3x + 2$
<code>\</code>	<code>\$x^2\ +3x\ +2\$</code>	$x^2 + 3x + 2$
<code>\quad</code>	<code>\$x^2\quad +3x\quad +2\$</code>	$x^2 + 3x + 2$

Equations

En LaTeX vous pouvez présenter vos équations de la manière suivante :

<code>\begin{equation}</code>	→ Il faut utiliser <code>\begin{equation*}</code> si vous ne
<code>\label{eq1}</code>	voulez pas numéroté votre équation. Le <code>\label</code>
<code>f(x) = 3x(x+1)</code>	quant à lui permet de faire des références à l'équa-
<code>\end{equation}</code>	tion par <code>\ref{eq1}</code> .

Si vous voulez couper votre équation sur plusieurs lignes, il faut utiliser l'environnement `split`.

<code>\begin{equation} \label{eq1}</code>	
<code>\begin{split}</code>	
<code>f(x) &= 3x(x+1) \\</code>	$f(x) = 3x(x + 1)$
<code>&= 3x^2 + 3x</code>	$= 3x^2 + 3x$
<code>\end{split}</code>	
<code>\end{equation}</code>	(2)

Si maintenant vous voulez aligner plusieurs lignes, soit pour plusieurs équations ou juste pour développer un calcul, l'environnement `align` est parfait :

<code>\begin{align} (Ou align*)</code>	→ Le <code>&</code> sert à aligner l'équation verticalement,
<code>2x - 5y &= 8 \\</code>	on le met généralement au niveau du signe égal,
<code>3x + 9y &= -12</code>	à noter que vous pouvez en mettre plusieurs sur
<code>\end{align}</code>	une même ligne.

Les matrices s'écrivent d'ailleurs avec une syntaxe similaire, il faut utiliser l'environnement `pmatrix` et séparer les coefficients avec des `&` et utiliser des `\\` pour changer de ligne.

Voilà pour le principal, il y a bien sur une multitude d'autres environnement qui permettent d'augmenter la complexité, vous trouverez dans le guide complet une présentation de ces fonctionnalités. J'aimerais ici finir par quelque chose qui vous sera sûrement utile : **définir une fonction en mathématiques**.

<code>\[f: \left</code>	
<code>\begin{array}{rcl}</code>	
<code>\mathbb{R} &\longrightarrow & \mathbb{R} \\</code>	$f : \left \begin{array}{l} \mathbb{R} \longrightarrow \mathbb{R} \\ x \longmapsto \begin{cases} x^2/2 & \text{si } x \geq 0 \\ \frac{\pi^2 + 1}{2} & \text{si } x < 0 \end{cases} \end{array} \right.$
<code>x &\longmapsto & \begin{cases} \frac{x^2}{2} & \text{si } x \geq 0 \\ \frac{\pi^2 + 1}{2} & \text{si } x < 0 \end{cases}</code>	
<code>\textrm{si } x \geq 0 \\</code>	
<code>\frac{\pi^2 + 1}{2} & \textrm{si } x < 0</code>	
<code>\end{cases}</code>	
<code>\end{array} \right. \]</code>	

Bien sur cet exemple est un degré de complexité au dessus de ce que vous pourriez être amenés à faire, mais il vous suffit de le modifier à votre convenance.

Inclure du code

Dans la plupart de vos TPs, vous aurez à compléter un code Python ou même juste à le faire tourner pour faire une approximation numérique, un calcul de déformations, etc. Il vous faudra donc inclure dans votre compte-rendu la partie du code la plus importante que vous avez écrite. Pour cela, LaTeX vous propose le package `listings`.

En n'oubliant pas de charger ce package en préambule de votre document, vous pourrez inclure du code de cette façon :

```
import math as m
import numpy as np
import scipy as sp
import matplotlib.pyplot as plt

def OpenPtsFile(filename): #function to import data in Pts file
    f = open(filename, 'r')
    print ('Reading file ' + filename)
    s = f.readline()
    vals = s.split("\t")
    npts = int(vals[0])
    nraw = int(vals[1])
    ncol = int(vals[2])
    print(npts, 'points,', nraw, 'rows,', ncol, 'columns')
    Pts = np.zeros(shape=(nraw, ncol, 2))
```

Ce résultat est celui obtenu en faisant un simple copier-coller du code source, vous pouvez d'ailleurs éviter cette démarche par la commande :

`\lstinputlisting[language=Python, firstline=2, lastline=23]{moncode.py}` qui va inclure votre code source de la ligne `firstline` (=2) à la ligne `lastline` (=23) puis le mettre en forme comme ci-dessus.

Rendre le code plus joli :

Il est possible de modifier l'aspect visuel du code dans votre document en définissant un `style` par l'intermédiaire de la commande `\lstdefinestyle`. Les possibilités étant très nombreuses, je ne vais pas vous décrire toutes les options possibles, je vous montre plutôt un exemple de ce qu'il est possible de faire. Une petite recherche sur internet³ vous permettra d'en savoir plus.

```
1 import math as m
2 import numpy as np
3 import scipy as sp
4 import matplotlib.pyplot as plt
5
6 def OpenPtsFile(filename): #function to import data in Pts file
7     f = open(filename, 'r')
8     print ('Reading file ' + filename)
9     s = f.readline()
10    vals = s.split("\t")
11    npts = int(vals[0])
12    nraw = int(vals[1])
13    ncol = int(vals[2])
14    print(npts, 'points,', nraw, 'rows,', ncol, 'columns')
15    Pts = np.zeros(shape=(nraw, ncol, 2))
```

3. Vous pouvez aussi consulter la documentation du package `listings` disponible [ici](#)

Faire une bibliographie

Cette section est un peu plus technique car elle requiert l'usage d'un fichier tiers qui vous permettra de faire votre bibliographie. En premier lieu, il vous faut le package `biblatex`, importez-le, puis suivez les étapes suivantes :

- Étape 1 : Sur Overleaf, cliquez sur le bouton "New File" en haut à gauche de l'écran (sous le bouton menu). Écrivez "exemple.bib" dans la zone de texte puis créez votre fichier.
- Étape 2 : Ensuite dans votre fichier .tex, il faut ajouter la ligne `\addbibresource{exemple.bib}` en dessous de votre `\usepackage{biblatex}`.
- Étape 3 : Dans votre fichier "exemple.bib", copiez-collez le texte suivant :

```
@book{guidelatex,
  author = {Le KI'022},
  title = {Guide d'utilisation de \LaTeX},
  year = {2020},
}
```

- Étape 4 : Maintenant, il vous faut écrire un petit texte pour citer le guide, par exemple :
"J'utilise le guide LaTeX du KI `\cite{guidelatex}`⁴ pour pouvoir écrire mes rapports en première année des Ponts."
Puis ajoutez simplement la commande `\printbibliography` à la fin de votre document.
- Bonus : Vous pouvez écrire `\printbibliography[heading=bibintoc, title=Bibliographie]` pour ajouter votre bibliographie à la table des matières (nécessite le `\tableofcontents`).

Table des matières	
1 Introduction	1
Bibliographie	1
1 Introduction	
J'utilise le guide LaTeX du KI [1] pour pouvoir écrire mes rapports en première année des Ponts.	J'utilise le guide LaTeX du KI [1] pour pouvoir écrire mes rapports en première année des Ponts.
Références	Bibliographie
[1] Le KI'022. <i>Guide d'utilisation de L^AT_EX</i> . 2020.	[1] Le KI'022. <i>Guide d'utilisation de L^AT_EX</i> . 2020.
(a) Sans la table des matières	(b) Avec la table des matières

→ Voilà la marche à suivre, en fonction des œuvres à citer, il vous faudra juste changer l'étape 3 en mettant d'autres œuvres comme des articles `@article`, des thèses `@thesis`,...

4. L'argument du `\cite` est le premier argument de `\@book{...}`.

Créer vos propres commandes

Cette section n'apportera rien de plus à vos documents LaTeX, elle vous fera simplement gagner du temps à l'écriture. Mais lorsque vous devrez écrire des rapports de TP en AnaCS par exemple, il vous faudra écrire beaucoup de normes, intégrales et autres expressions qui partagent beaucoup de caractéristiques communes.

Voyons un exemple pour illustrer mes propos :

Code	PDF	
<code>\$ \ u \ _{\mathcal{H}^1_0}</code> <code>\left(\Omega \right)</code>	$\ u\ _{\mathcal{H}^1_0(\Omega)}$	→ L'idéal serait d'avoir une commande <code>\norme{u}</code> qui donnerait le même rendu que le code de gauche.

Heureusement, nous pouvons en LaTeX définir nos propres commandes de façon assez simple. Avant de s'attaquer à l'exemple ci-dessus, voyons-en un plus simple.

Par exemple, la commande `\mathbb{R}` peut s'avérer redondante lorsque l'on doit faire beaucoup intervenir ce symbole (\mathbb{R}).

Pour cela, il vous suffit d'écrire en préambule de votre document (avant le `\begin{document}`) :

<code>\newcommand{\R}{\mathbb{R}}.</code>	→ Le <code>\R</code> est le nom de votre commande tandis que le <code>\mathbb{R}</code> est ce que votre commande va effectuer. Voilà, maintenant, pour écrire \mathbb{R} il vous suffira d'écrire <code>\R</code> .
---	--

Maintenant il faudrait que votre commande puisse accepter des paramètres, c'est à dire de la forme `\maccommande{argument}`. Il faut donc modifier la commande précédente ainsi :

<code>\newcommand{\bb}[1]{\mathbb{#1}}</code>	→ Le <code>\bb</code> est le nom de votre commande tandis que <code>[1]</code> est le nombre d'arguments. Enfin <code>\mathbb{#1}</code> est ce que fait votre commande, le <code>#1</code> sera remplacé par le texte que vous mettrez dans <code>\bb{...}</code> ⁵ .
---	---

Voilà, nous pouvons maintenant écrire la commande vue au début de la section, nous voulons un seul paramètre, la fonction `u`, tout le reste doit être fixé :

`\newcommand{\norme}[1]{\| #1 \|_{\mathcal{H}^1_0} \left(\Omega \right)}`

Ainsi, si vous êtes dans un environnement math, la commande `\norme{u}` vous produira $\|u\|_{\mathcal{H}^1_0(\Omega)}$.

Conclusion

Merci d'avoir lu ce guide, j'espère qu'il vous aura été utile et que les explications étaient assez claires. Si jamais vous voulez en savoir plus, sur n'importe quel aspect de ce qui a été présenté ici, je vous invite à consulter le guide complet disponible sur [ce site](#). Et surtout n'oubliez pas que **Google est votre ami**, la plupart de vos questions ont déjà été posées et répondues sur plein de forums.

N'hésitez pas à me faire part de vos remarques ou commentaires, soit par mail à l'adresse pierre-louis.ruhmann@eleves.enpc.fr ou en contactant le KI par Facebook ou sur [son site](#).

5. Si le nombre d'arguments est supérieur à 1, vous y ferez référence avec `#1`, `#2`, etc. Et pour utiliser votre commande vous devrez écrire `\maccommande{argument 1}{argument 2}...{argument n}`.