

Débuter en L^AT_EX*

Maxime SENOT

16 novembre 2020

Table des matières

1	L'environnement de travail	2
1.1	Compiler avec un terminal	2
1.2	Compiler avec un éditeur plus ou moins spécifique	2
2	Commencer un document	2
3	Mise en forme du document	3
3.1	Mise en forme générale du document	3
3.2	Les listes	4
4	Inclure des images	4
5	Écrire un algorithme	5
6	Labels, références, notes de bas de page	5
7	Les maths	6
8	Ressources	7

Résumé

L^AT_EX est un langage de programmation pour faire de la mise en page de texte, que ce soit un article sur une ou deux colonnes, un livre, des transparents (rendez-vous en avril pour la soutenance !)...

L^AT_EX n'est pas un éditeur WYSIWYG (contrairement à Word : *What You See Is What You Get*) : comme pour tout langage de programmation, les fichiers sources doivent être compilés.

Dans L^AT_EX, tout est automatisé (entête, numérotation des parties, de la biblio, des notes de bas de page, etc). Le langage est très riche (formules mathématiques, bibliographie, références, table des matières, figures, pseudo-codes, etc). La base d'utilisateurs est gigantesque (documentations, faq, librairies).

*d'après un document de Lucie Martinet et Damien Stehlé.

1 L'environnement de travail

Un document \LaTeX a une extension `.tex`.

C'est le fichier `.tex` qui va être compilé. On peut écrire du \LaTeX avec n'importe quel éditeur de texte (gedit, emacs,...). Pour compiler, il faut soit utiliser un terminal soit un éditeur qui permet de compiler avec des raccourcis.

1.1 Compiler avec un terminal

Il faut d'abord se placer dans le répertoire où se trouve votre `.tex` (comme d'habitude) et lancer l'une des commandes suivantes :

```
latex PremierTex.tex
pdflatex PremierTex.tex
```

La première commande permet d'obtenir un `.dvi`. On pourra ensuite l'exporter en `.pdf` avec la commande `dvipdf PremierTex.dvi`, ou en `.ps` avec la commande `dvips -o PremierTex.ps PremierTex.dvi`.

La seconde commande est plus pratique, car elle permet d'avoir un `.pdf` directement. Mais il peut y avoir des incompatibilités en particulier dans le cas d'une insertion de figures dans d'autres formats que `.pdf`.

Pour que la table des matières (qui s'obtient avec la commande `\tableofcontents`, placée dans le corps du document \LaTeX) soit à jour, il faut compiler deux fois de suite. C'est aussi le cas pour les références, ou pour la bibliographie.

1.2 Compiler avec un éditeur plus ou moins spécifique

Ce document a été initialement compilé avec TeXworks (disponible sous windows). Dans la plupart des éditeurs spécifiques à \LaTeX , il existe un bouton permettant d'effectuer la compilation du fichier `.tex` directement en fichier `.pdf`. De nombreux autres éditeurs de textes spécifiques à \LaTeX existent : Kile, TexMaker, TeXnicCenter (Windows)...

2 Commencer un document

Comme pour Python, C, Java, ou tout autre langage de programmation, un programme \LaTeX commence par des entêtes et une liste de packages à importer.

L'une des plus courtes formes se présente de la manière suivante :

```
\documentclass[10pt]{TypeDeDocument}
\usepackage[utf8]{inputenc}
\usepackage[frenchb]{babel}
\usepackage[T1]{fontenc}
```

```
\date{}
```

```
\begin{document}  
\section{Titre de la première section}  
\section{Titre de la deuxième section}  
\end{document}
```

Avant `\begin{document}`, nous pouvons aussi ajouter nos propres commandes en les déclarant préalablement.

Ce qui arrive après la commande `\end{document}` n'est pas pris en compte lors de la compilation.

3 Mise en forme du document

3.1 Mise en forme générale du document

Pour faire des sections numérotées avec un titre, on utilise la commande suivante :

```
\section{Titre}
```

Pour faire des sous-sections :

```
\subsection{Sous-Titre}
```

Pour faire des sous-sous-sections :

```
\subsubsection{Sous-Sous-Titre}
```

Et ça s'arrête là. Pour supprimer la numérotation des sections, il faut ajouter une `*` comme suit :

```
\section*{Titre}  
\subsection*{Sous-Titre}  
\subsubsection*{Sous-Sous-Titre}
```

L'indentation se fait automatiquement à chaque nouveau paragraphe. Un nouveau paragraphe est créé quand il y a un saut de ligne dans le code `LATEX`. On peut également utiliser les commandes suivantes :

```
\\      %pour passer à la ligne suivante  
%ou  
\par    %pour commencer un nouveau paragraphe
```

3.2 Les listes

Il est possible de faire des listes, des énumérations, et même de changer les numéros et la forme des « tirets ». Par exemple :

Ma liste de courses :

— Jus de fruits

⇒ Viande

- Une nouvelle classe de L1

Ou par ordre de priorité :

1. Une nouvelle classe de L1
2. Jus de fruits
3. Viande

L^AT_EX permet également de faire des tableaux :

case 1	case 4	case 7
case 2	c	case 8
case 3	case 6	case 9

On peut aussi centrer du texte, ou n'importe quoi. Par exemple :

case 1	case 4	case 7
case 2	case 5	case 8
case 3	case 6	case 9

4 Inclure des images

Pour faire cela, vous devez inclure un package qui le permet :

```
\usepackage{graphicx}
```

A l'endroit où vous voulez inclure votre image, vous devez utiliser la commande suivante :

```
\includegraphics{nom_de_la_figure}
```

Par exemple :



Bon, c'est pas super... Pour arranger un peu l'affichage, il faut jouer avec les options, et tester !

En général, lorsqu'on ajoute une image au sein d'un texte, on aime bien pouvoir lui donner une légende et y faire référence. Pour cela, il existe un environnement flottant, nommé *figure*. Un environnement flottant signifie, comme son nom l'indique, qu'il peut être déplacé. En particulier, L^AT_EX placera cet environnement à l'endroit qu'il juge le plus adéquat, souvent, à l'endroit qui permet de minimiser la taille du document.

```
\begin{figure}[!h]
  \includegraphics{nom_de_la_figure}
  \caption{La légende que vous voulez.}
\end{figure}
```

L’option `[!h]` permet de forcer \LaTeX à placer notre figure à cet endroit du document et pas ailleurs. Malgré cette instruction, \LaTeX n’obéit pas toujours, mais c’est rare.

L’instruction `\caption{La légende que vous voulez}` permet d’ajouter une légende à la figure.

Vous pouvez aussi ranger vos figures dans un répertoire séparé, par exemple un sous-répertoire. Cependant, vous devrez veiller à écrire le bon chemin.

Voici ce que l’on obtient avec la figure précédente, avec quelques options :

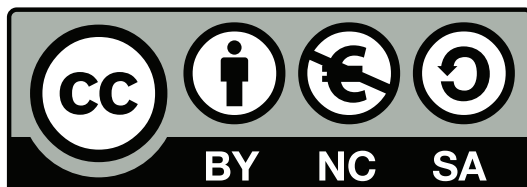


FIGURE 1 – Logo de la licence libre CC BY-NC-SA.

5 Écrire un algorithme

Il existe différents packages pour présenter des algorithmes en \LaTeX : `algorithm2e`, `algorithmics`... L’algorithme « Faire un rapport \LaTeX » est un exemple d’algorithme obtenu grâce au package `algorithm2e`.

6 Labels, références, notes de bas de page

Tout cela peut se faire automatiquement, sans avoir à chercher les numéros qui vont bien manuellement. Les commandes importantes pour cela sont :

```
\label{}      % pour créer un label
\ref{}        % pour lui faire référence
\cite{}       % idem \ref, mais pour la biblio
\footnote{}   % note de bas de page
```

Comme j’ai créé des références dans mon fichier source, je peux maintenant parler de la Figure 1 de la Section 4. Je peux également faire référence à l’entrée [1] de la bibliographie.^{1 2}

-
1. Ne pas abuser des notes de bas de page
 2. Qu’est-ce que je viens juste de dire ?

Algorithme 1 : Faire un rapport en L^AT_EX.

input : un rapport de projet à écrire.
output : le rapport rédigé en L^AT_EX.

```
// voir la doc d'algorithm2e pour définir des  
    mots-clés et instructions en français  
 $n \leftarrow$  nombre de personnes dans l'équipe du projet  
Rédiger_rapport(param1 =  $n$  élèves ; param2 = consignes)  
Compiler_rapport  
while (errors) ou (warnings) do  
    Lire_messages  
    Corriger_erreurs  
    Compiler_rapport  
if Accord(param_opt = toute l'équipe) or date=deadline then  
    Rendre_rapport  
else  
    Effectuer_retouches
```

7 Les maths

Pour faire des maths, on utilise un environnement spécial. On a le choix entre deux environnements possibles :

`$mon expression mathématiques$`

ou

`\[mon expression mathématiques\]`

La première version permet d'écrire des maths au sein d'une ligne de texte sans revenir à la ligne. Un petit exemple valant mieux qu'un long discours, on peut écrire

ceci : $\sum_{i=0}^{max} i^2$, avec la commande `$\displaystyle\sum_{i=0}^{\max} i^2$`.

La deuxième crée un environnement qui sera séparé du texte par un retour à la ligne et des sauts de lignes. Par exemple, l'expression `\[\sum_{i=0}^n \prod_{j=i}^m k_{ij}^2\]` donnera :

$$\sum_{i=0}^n \prod_{j=i}^m k_{ij}^2$$

On peut aussi créer des commandes L^AT_EX pour les formules que l'on utilise souvent, et pour rendre le fichier source plus lisible. Par exemple, avec les commandes créées au début du source .tex de ce .pdf : ε et $\langle \vec{b}_1, \vec{b}_2 \rangle$.

8 Ressources

Quelques liens très utiles :

`www.tuteurs.ens.fr/logiciels/latex`

`http://melusine.eu.org/syracuse/doc/faq-tex-french/faq-tex-french.html`

`http://detexify.kirelabs.org/classify.html`

`www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf`

Références

- [1] Leslie Lamport, *ΛT_EX: A Document Preparation System*. Addison Wesley, Massachusetts, 2nd Edition, 1994.
- [2] The Comprehensive TeX Archive Network. `http://www.ctan.org/`
- [3] Donald E. Knuth, *The T_EX book*. Addison Wesley, 1986.