

Fiche d'aide L^AT_EX

Rémy Detobel

13 novembre 2015

Table des matières

1	Introduction	3
1.1	Les avantages	3
1.2	Les IDE	3
1.3	Les balises	3
1.4	Les packages.	3
1.5	Le corps du texte.	4
1.6	Document déjà fait	4
2	Éléments basique	4
2.1	Type de document	4
2.2	Structure du contenu	4
2.2.1	Titre	4
2.2.2	Table des matières.	5
2.2.3	Marges.	5
2.2.4	Paragraphe	5
2.3	Mise en forme	5
3	Liste	5
3.1	Introduction	5
3.2	Liste numéroté	5
3.3	Liste à point	5
3.4	Personnalisation	6
4	Tableau	6
4.1	Introduction	6
4.2	Créer un tableau	6
4.3	Bordure	6
5	Image	6
6	Math	6
6.1	Introduction	6
6.1.1	Code sur une ligne	6
6.1.2	Code centre	6
6.2	Résumé des balises	6
6.3	Lettres grec	6
6.4	Graphes	7
6.4.1	Exemple	7
6.4.2	Introduction	7
6.4.3	Crée l'environnement	7
6.4.4	Créer un point	7
6.5	Créer une ligne	8
6.6	Matrice	8
6.6.1	Introduction	8
6.6.2	Matrice classique	8
6.6.3	Déterminant	8

1 Introduction

1.1 Les avantages

\LaTeX permet de créer un document PDF via des balises. Il est très pratique pour écrire un rapport, des formules mathématiques, une table des matières... Bref, ce qui (en général) est compliqué sur des éditeurs tel que Word devient “simple” ici (lorsque l’on a pris l’habitude de l’utiliser). Sachez également que les profs aiment voir de jolis rapports en \LaTeX .

1.2 Les IDE

Les IDE viennent de “Integrated Development Environment”, ici on ne fait pas de développement... On devrait dire ILE “Integrated \LaTeX Environment. Mais entre nous, je vous propose de garder IDE. Ces IDE vous permettent de créer un document \LaTeX plus facilement que via un simple éditeur de texte. Votre éditeur vous permet d’ajouter une balise juste en poussant sur un bouton, fait de l’autocomplétion, ... Bref, pleins de petites choses qui vous permettent d’aller plus vite.

Nombre de sondés : 9

- texWork : 2
- Kile : 1
- LateXila : 1
- TexMaker : 1
- TexStudio : 2
- sublime texte : 1
- vim : 1
- notepad++ : 1

Les IDE intègrent en général un compilateur permettant de créer le document PDF. Si vous n’utilisez pas d’IDE vous pouvez tout de même écrire du \LaTeX dans un simple éditeur de texte et compiler ensuite votre “code” pour produire le document PDF.

1.3 Les balises

\LaTeX utilise des balises permettant de préciser comment votre texte doit être compris par l’éditeur. Ces balises commencent en général par un backslash (\backslash). Il y a deux types de balises : les balises qui doivent être ouvertes et fermées et les simples balises. Par exemple, pour définir le contenu de votre document vous devez le mettre entre les balises `\begin{document}` et `\end{document}`. En fait vous spécifiez le début et la fin. Ces balises “begin” et “end” permettent de définir un environnement. Vous retrouverez ces balises dans les parties à venir.

Les balises simples sont beaucoup utilisées pour les caractères spéciaux tel que les symboles mathématiques ou les caractères UTF8 non disponibles de base sur un clavier. Par exemple, pour mettre un backslash vous ne pouvez pas taper `\` car il sera interprété comme le début d’une balise. Vous devez donc taper : `\textbackslash`.

1.4 Les packages.

Par défaut, \LaTeX intègre des balises (comme en python, en C++ ou en Java). Cependant, si vous voulez faire des choses plus complexes, il vous faudra importer des packages (des sortes de librairies). Les packages vous permettent donc d’utiliser de nouvelles balises. Les packages doivent être importés en début de document via la balise `\usepackage{nom du package}`. Sachez que certains packages se trouvent déjà sur votre ordinateur et ne sont pas intégrés par défaut dans \LaTeX alors que d’autres doivent être téléchargés en plus. Internet vous renseignera mieux que nous sur comment installer un package et sur tous les packages existants.

1.5 Le corps du texte.

1.6 Document déjà fait

Voici un “code” avec déjà tout pour que vous puissiez écrire vos documents. A noté que le symbole “%” permet de commencer un commentaire. C’est à dire du texte qui ne sera pas lu par L^AT_EX et donc simplement ignoré.

```
\documentclass[a4paper,11pt]{article}
\usepackage[french]{babel} % Mise en page Français
\usepackage[utf8]{inputenc} % Mode d’encodage
\usepackage[T1]{fontenc}
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{hyperref} %Permet d’avoir des liens automatique dans votre table des matières
\usepackage{graphicx}
\usepackage[top=2cm, bottom=2cm, left=2cm, right=2cm]{geometry}

\title{Votre Titre}
\author{L’auteur}

\begin{document} % Début du document

\maketitle % Création d’un titre automatiquement

\newpage % Nouvelle page
\tableofcontents % Table des matières
\newpage % Nouvelle page

% Votre Texte

\end{document} % Fin du document
```

2 Éléments basique

2.1 Type de document

2.2 Structure du contenu

2.2.1 Titre

Vous pouvez créé des séparation, des titres, des chapitres, ... Bref, plusieurs méthodes existent pour structurer votre document. En fonction du type de document (voir paragraphe ci dessus) il y a différentes balises à utiliser. Voici la liste des balises les plus utilisé permettant d’organiser son code (trié par leur ordre d’importance) :

1. `\chapter{titre}` Chapitre (disponible dans les documents article)
2. `\section{titre}` Section
3. `\subsection{titre}` Sous-section
4. `\subsubsection{titre}` Sous-sous-section

Par exemple, “Éléments basique” est une **section** et “Structure du contenu” est une **subsubsection**.

2.2.2 Table des matières.

La création d'une table des matières sur L^AT_EX se fait de manière automatique. Celle-ci sera mise en place via l'instruction suivante :

```
\tableofcontents
```

Celle-ci va reprendre l'ensemble des structures utilisées dans votre script L^AT_EX (`chapitre`, `section`, `subsection`, `subsubsection`) et les mettre en page.

2.2.3 Marges.

Si vous avez déjà tenté de compiler votre script L^AT_EX une fois, vous vous serez sûrement demandé comment réduire la taille de ces imposantes marges. Pour cela rien de plus simple. Il vous suffit d'importer le package `geometry` et d'y inclure la taille de vos marges comme suit :

```
\usepackage[top=2cm, bottom=2cm, left=2cm, right=2cm]{geometry}
```

2.2.4 Paragraphe

2.3 Mise en forme

Il est toujours possible de mettre en page le texte dans vos paragraphes. Notez que la mise en page se fait automatiquement pour les titres.

<code>\textit{texte}</code>	<i>I</i>	Italique
<code>\underline{texte}</code>	<u>U</u>	Souligne
<code>\textbf{texte}</code>	G	Gras

Il est également possible d'aligner votre texte au centre en le mettant entre des balises `\begin{center}` et `\end{center}`.

3 Liste

3.1 Introduction

L^AT_EX intègre plusieurs types de listes. Nous vous en présenterons deux dans ce document.

3.2 Liste numérotée

Il s'agit simplement d'une liste avec des numéros devant chaque point.

```
\begin{enumerate}
  \item Premier point
  \item Second point
\end{enumerate}
```

1. Premier point
2. Second point

3.3 Liste à point

Encore plus basique, il s'agit d'une liste avec simplement des points.

```
\begin{itemize}
  \item Premier point
  \item Second point
\end{itemize}
```

- Premier point
- Second point

3.4 Personnalisation

Il vous est possible de personnaliser les caractères mis au début de chaque nouveau point. Pour cela il vous suffit de mettre entre crochet ([]) les caractères que vous voulez. Par exemple • ou un +

```
\begin{itemize}
  \item[\textbullet] Premier point
  \item[+] Second point
  \item[$-$] Troisième point
\end{itemize}
```

• Premier point
+ Second point
– Troisième point

4 Tableau

4.1 Introduction

Faire un tableau en L^AT_EX n'est pas toujours simple. Nous vous expliquons ici les balises que nous utilisons le plus souvent. Sachez cependant que pour des tableaux complexes le plus simple reste d'utiliser un outil disponible sur internet à l'adresse : http://tablesgenerator.com/latex_tables

4.2 Créer un tableau

4.3 Bordure

5 Image

6 Math

6.1 Introduction

Il existe plusieurs manières d'écire des formules mathématiques. Quoiqu'il en soit il vous faut rentrer dans un environnement math. Il existe plusieurs environnements mathématiques et plusieurs moyens d'y rentrer.

6.1.1 Code sur une ligne

Pour écrire vos formules mathématiques dans une phrase vous devez les mettre **entre** des balises dollar (\$) ou entre la suite de caractères “\(" et “\)” pour fermer la balise. Voici un petit exemple : $\forall a \in \mathbb{R}, \sum_{i=1}^{10} (a + i)$

6.1.2 Code centre

A d'autre moment, vous aurez plus envie de mettre en évidence une formule et de la mettre au centre spécialement sur une ligne. Pour cela vous devez mettre vos balises **entre** des doubles dollars (\$\$) soit via la balise “\[” et “\]”

$$\forall a \in \mathbb{R}, \sum_{i=1}^{10} (a + i)$$

6.2 Résumé des balises

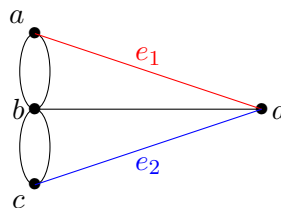
6.3 Lettres grec

Pour utiliser les lettres grecques il vous suffit de mettre leur nom. Par exemple α (\alpha), β (\beta), ... Certaines lettres grecques peuvent être écrites en majuscule comme par exemple sigma. Pour se faire, il suffit de mettre la première lettre de la commande en majuscule. Exemple : Σ (\Sigma) (à la place σ (\sigma)).

Balise	Symbole	Description
<code>\forall</code>	\forall	Pour tout
<code>\in</code>	\in	Appartient à
<code>\mathbb{Lettre}</code>	\mathbb{N}	Ensemble (remplacer “Lettre” par la lettre de l’ensemble voulu)
<code>\neq</code>	\neq	Différent de..
<code>\subset</code>	\subset	Inclus dans...
<code>\supset</code>	\supset	Contient...
<code>\subseteq</code>	\subseteq	Inclus ou égal à...
<code>\supseteq</code>	\supseteq	Contient ou égal à...
<code>\exists</code>	\exists	Il existe ...
<code>\leq</code>	\leq	Plus petit ou égal
<code>\geq</code>	\geq	Plus grand ou égal
<code>\Sigma</code>	Σ	Somme
<code>\log</code>	\log	Logarithme
<code>\int</code>	\int	Intégral
<code>\infty</code>	∞	Infini
<code>\phi</code>	ϕ	Vide
<code>\sqrt{valeur}</code>	\sqrt{valeur}	Racine de valeur
<code>\frac{x}{y}</code>	$\frac{x}{y}$	Fraction (division) de x par y

6.4 Graphes

6.4.1 Exemple



6.4.2 Introduction

Nous ne présenterons ici qu’une toute petite partie de ce module simplement pour vous expliquer que cela existe et pour vous expliquer rapidement. Sachez également qu’il est possible de faire des repères cartésien, ... Nous vous invitons à consulter internet pour plus d’information.

6.4.3 Créé l’environnement

```
\begin{minipage}{\textwidth}
\center
\begin{tikzpicture}
```

Avec ces 4 balises vous serez dans un environnement de dessin. La seule chose que je vous propose de modifier ici est `{\textwidth}`. Cela définit la taille que va prendre votre dessin. Si vous voulez par exemple que le graphique prenne la moitié de la page (en largeur) mettez : `{0.5\textwidth}`.

6.4.4 Créer un point

Exemple : `\draw (0,2) node[anchor=south east]{a} node{\textbullet};`
 Passons rapidement en revue cette instruction (l’ordre des nodes n’a pas d’importance). Nous avons donc l’instruction `\draw` suivit des coordonnées où placer le point.
 Vient ensuite `node[anchor=south east]{a}` qui décrit le label à placé à côté du point. Ici on précise

(via `anchor` que le point sera placé au sud est du label.

Enfin, on dit de placer un `\textbullet` à cette coordonnées (ce qui correspond à \bullet).

6.5 Créer une ligne

Exemple : `\draw (0,1) -- (3,1);`

On retrouve de nouveau les coordonnées. La syntaxe est assez simple, on a les 2 coordonnées à relier lié par 2 tirets représentant la ligne. Ces deux tirets peuvent être changé par ellipse ou autre (la syntaxe des coordonnées deva peut-être un peu changé ou précisé).

6.6 Matrice

6.6.1 Introduction

Nous parlerons ici de seulement deux sortes de matrices : $()$ et $| |$.

6.6.2 Matrice classique

Pour faire une matrice, il vous faut rentrer dans l'environnement math (avec des $\$$ par exemple). Ensuite vous devez spécifié que vous voulez une matrice, ici nous utiliseront "pmatrix". Nous allons donc faire `\begin{pmatrix}`. Ensuite, cela va se passer comment un tableau. C'est à dire un "&" pour séparer le colonne et "\\" pour les lignes. Cela donne donc :

<code>\begin{pmatrix}</code>	
<code>a & b \\\</code>	$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$
<code>c & d</code>	
<code>\end{pmatrix}</code>	

6.6.3 Déterminant

Il vous suffit de suivre la même syntaxe pour faire des matrices déterminante. Un bon exemple vaut mieux que des explications :

<code>\begin{vmatrix}</code>	
<code>a & b \\\</code>	$\begin{vmatrix} a & b \\ c & d \end{vmatrix}$
<code>c & d</code>	
<code>\end{vmatrix}</code>	