

# Fiche d'aide L<sup>A</sup>T<sub>E</sub>X

Rémy Detobel & Denis Hoornaert

2 février 2018

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
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
<b>2</b>	<b>Éléments basique</b>	<b>4</b>
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
<b>3</b>	<b>Liste</b>	<b>5</b>
3.1	Introduction . . . . .	5
3.2	Liste numérotée . . . . .	5
3.3	Liste à points . . . . .	6
3.4	Personnalisation . . . . .	6
<b>4</b>	<b>Tableau</b>	<b>6</b>
4.1	Introduction . . . . .	6
4.2	Créer un tableau . . . . .	6
4.3	Bordure . . . . .	6
<b>5</b>	<b>Image</b>	<b>6</b>
<b>6</b>	<b>Math</b>	<b>6</b>
6.1	Introduction . . . . .	6
6.1.1	Code intégré à une ligne . . . . .	6
6.1.2	Code centré . . . . .	6
6.2	Résumé des balises . . . . .	7
6.3	Opérateurs et bornes . . . . .	7
6.4	Lettres grecques . . . . .	8
6.5	Graphes . . . . .	8
6.5.1	Exemple . . . . .	8
6.5.2	Introduction . . . . .	8
6.5.3	Crée l'environnement . . . . .	8
6.5.4	Créer un point . . . . .	8
6.6	Créer une ligne . . . . .	8
6.7	Matrice . . . . .	9
6.7.1	Introduction . . . . .	9
6.7.2	Matrice classique . . . . .	9
6.7.3	Déterminant . . . . .	9
<b>7</b>	<b>Haut et bas de page</b>	<b>9</b>
7.1	Introduction . . . . .	9
7.2	Initialiser . . . . .	9
7.3	Mettre du texte et des images . . . . .	9
7.4	Ajouter des lignes . . . . .	10

# 1 Introduction

## 1.1 Les avantages

L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X.

## 1.2 Les IDE

Les IDE veulent dire “Integrated Development Environment”, ici on ne fait pas de développement... On devrait dire ILE “Integrated L<sup>A</sup>T<sub>E</sub>X Environment. Mais entre nous, je vous propose de garder IDE.

Ces IDE vous permettent de créer un document L<sup>A</sup>T<sub>E</sub>X 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’auto-complétion, etc. Bref, plein de petites choses qui vous permettront d’aller plus vite.

Nombre de sondés : 9

- texWork : 2
- Kile : 1
- LateXila : 1
- TexMaker : 1
- TexStudio : 2
- Sublime Text : 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 L<sup>A</sup>T<sub>E</sub>X dans un simple éditeur de texte et compiler ensuite votre “code” pour produire le document PDF.

## 1.3 Les balises

L<sup>A</sup>T<sub>E</sub>X utilise des balises permettant de préciser comment votre texte doit être compris par l’éditeur. Ces balises commencent en général par un backslash (\). Il y a deux types de balises les balises qui doivent être ouverte et fermée 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 tels que les symboles mathématiques ou les caractères UTF8 non disponible 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, L<sup>A</sup>T<sub>E</sub>X 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 bibliothèques). Les packages vous permettent donc d’utiliser de nouvelles balises. Les packages doivent être importé en début de document via la balise `\usepackage{nom du package}`. Sachez que certains packages se trouvent déjà sur votre ordinateur et ne sont juste pas intégrés par défaut dans L<sup>A</sup>T<sub>E</sub>X alors que d’autres doivent être téléchargées en plus. Internet vous renseignera mieux que nous sur comment installer un package et sur tous les packages existant.

## 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 noter que le symbole “%” permet de commencer un commentaire. C’est à dire du texte qui ne sera pas lu par L<sup>A</sup>T<sub>E</sub>X et donc simplement ignoré.

```
\documentclass[a4paper,11pt]{article}
\usepackage[french]{babel} % Mise en page Français
\usepackage[utf8]{inputenc} % Mode d’encodage du fichier source
\usepackage[T1]{fontenc} % Mode d’encodage de police dans le fichier destination
\usepackage{amsmath} % 3 packages pour la mise en page des formules mathématiques
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{hyperref} % Permet d’avoir des liens automatique dans votre table des matières
\usepackage{graphicx} % Permet d’inclure des images
\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 (page de garde)

\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 créer des séparations, des titres, des chapitres, etc. 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 utilisées 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**.

À noter que ces commandes-ci ne sont pas définies sur base de **begin** et **end** : un chapitre (respectivement une section) se termine lorsqu’un autre chapitre (respectivement une autre section) est déclaré.

### 2.2.2 Table des matières.

La création d'une table des matières sur L<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X (`chapter`, `section`, `subsection`, `subsubsection`) et les mettre en page.

### 2.2.3 Marges.

Si vous avez déjà tenté de compiler votre script L<sup>A</sup>T<sub>E</sub>X 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}
```

Ces données peuvent toutes être changées pour obtenir des marges de la taille désirée.

Une alternative au package `geometry` est le package `fullpage` qui s'utilise ainsi :

```
\usepackage{fullpage}
```

et qui trouve des marges automatiques.

### 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>	<b>G</b>	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<sup>A</sup>T<sub>E</sub>X 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 points

Encore plus basique, il s'agit d'une liste avec simplement des points (appelé liste à puces).

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

— Premier point  
— Second point

### 3.4 Personnalisation

Il vous est possible de personnaliser le caractère mis au début de chaque nouveau point. Pour cela il vous suffit de mettre entre crochet ([ ]) le 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<sup>A</sup>T<sub>E</sub>X n'est pas toujours simple. Nous vous expliquons ici les balises que nous utilisons le plus souvent. Sachez cependant que pour des tableau complexes, le plus simple reste d'utiliser un outil disponible sur internet à l'adresse : [http://tablesgenerator.com/latex\\_tables](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'écrire des formules mathématiques. Quoi qu'il en soit, il vous faut rentrer dans un environnement math. Il existe plusieurs environnement mathématiques et plusieurs moyens d'y rentrer.

#### 6.1.1 Code intégré à une ligne

Pour écrire vos formules mathématique 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 :  $\forall a \in \mathbb{R}, \sum_{i=1}^{10} (a + i)$ . Le code utilisé est le suivant :

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

#### 6.1.2 Code centré

A d'autres moments, 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 “\[” et “\]”

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

Le code utilisé est le suivant :

```
\[ \forall a \in \mathbb{R}, \sum_{i=1}^{10} (a+i). \]
```

Vous pouvez également remarquer que l’affichage dans ces deux modes est différent : une formule centrée prendra plus de place et une formule intégrée à une ligne tentera de prendre le moins de place possible pour ne pas casser l’alignement du paragraphe.

## 6.2 Résumé des balises

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>\to</code>	$\rightarrow$	Flèche vers la droite
<code>\Rightarrow</code>	$\Rightarrow$	Flèche d’implication logique
<code>\iff</code>	$\Leftrightarrow$	Double implication (si et seulement si)
<code>\exists</code>	$\exists$	Il existe ...
<code>\leq</code>	$\leq$	Plus petit ou égal
<code>\geq</code>	$\geq$	Plus grand ou égal
<code>\sum</code>	$\sum$	Somme
<code>\prod</code>	$\prod$	Produit
<code>\log</code>	$\log$	Logarithme
<code>\int</code>	$\int$	Intégral
<code>\infty</code>	$\infty$	Infini
<code>\lim</code>	$\lim$	Limite
<code>\emptyset</code>	$\emptyset$	Vide
<code>\sqrt{valeur}</code>	$\sqrt{valeur}$	Racine de valeur
<code>\frac{x}{y}</code>	$\frac{x}{y}$	Fraction (division) de x par y

## 6.3 Opérateurs et bornes

Certaines notations mathématiques attendent des bornes (lors d’une sommation  $\sum$  par exemple, ou pour une limite). Ces bornes sont soit des bornes supérieures, soit inférieures. Par exemple, une borne qui doit être mise en dessous d’un opérateur se note entre accolades, juste après un underscore :

```
\lim_{n \to \infty}
```

pour représenter :

$$\lim_{n \rightarrow \infty}$$

De même, une borne qui doit être mise au dessus se fera de la même manière en changeant l’underscore par un circonflexe :

```
\sum_{i=1}^{10} f(i)
```

qui donne :

$$\sum_{i=1}^{10} f(i)$$

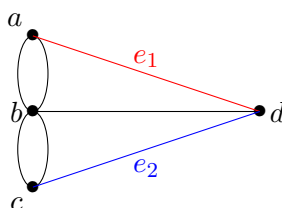
Cependant, ces notations ne sont pas réservées aux opérateurs particuliers. Il est commun de devoir mettre un exposant dans une formule (par exemple  $O(n^2)$ ) pour dénoter une complexité. L'exposant se met de la même manière : grâce au circonflexe. Également, ajouter un indice (par exemple  $x_1, x_2, \dots$  pour dénoter une suite) se fait par un underscore.

## 6.4 Lettres grecques

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

## 6.5 Graphes

### 6.5.1 Exemple



### 6.5.2 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.

### 6.5.3 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}`.

### 6.5.4 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  $\bullet$ ).

## 6.6 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 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).



## 6.7 Matrice

### 6.7.1 Introduction

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

### 6.7.2 Matrice classique

Pour faire une matrice, il vous faut rentrer dans l'environnement math (avec des \$ par exemple). Ensuite vous devez spécifier 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 les colonnes et "\\" pour les lignes. Cela donne donc :

```

\begin{pmatrix}
a & b \\
c & d
\end{pmatrix}

```

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Pour noter une matrice entre crochets et pas entre parenthèses, il suffit de remplacer `pmatrix` par `bmatrix` (pour *parenthesis matrix* et *bracket matrix* respectivement).

### 6.7.3 Déterminant

Il vous suffit de suivre la même syntaxe pour noter le déterminant d'une matrice. Un bon exemple vaut mieux que des explications :

```

\begin{vmatrix}
a & b \\
c & d
\end{vmatrix}

```

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix}$$

## 7 Haut et bas de page

### 7.1 Introduction

L<sup>A</sup>T<sub>E</sub>X permet de faire plein de choses par défaut de manière très rapide. Par exemple ajouter des informations au dessus ou en dessous de la page. Vous pouvez donc remarquer que ce document a le logo de l'ULB, le titre du document et le nom de ses auteurs en haut de la page. En bas de page il y a le numéro de la page. Il s'agit d'une librairie qui permet de faire ces effets. Il en existe plusieurs, nous allons vous apprendre quelques balises du package `fancyhdr`.

### 7.2 Initialiser

Pour utiliser le package `fancyhdr` nous allons commencer par importer le package et l'initialiser :

```

\usepackage{fancyhdr}
\pagestyle{fancy}

```

### 7.3 Mettre du texte et des images

Dans votre header, vous pouvez mettre du texte mais également des images. Il suffit d'écrire ces informations comme n'importe où dans votre document L<sup>A</sup>T<sub>E</sub>X. Nous allons donc commencer par définir l'image au milieu du header :

```

\fancyhead[C]{\includegraphics[scale=0.2]{./logo.jpg}}

```

Décortiquons cette ligne. Le premier mot "fancyhead" permet de dire que l'on va définir un header (du packet fancy). Si l'on veut définir des informations pour le footer, ce sera "fancyfoot". La lettre entre crochets définit la position que l'on configure. Il y a donc "C" pour center, "R" pour Right et "L" pour Left.

pour Left. Enfin, vous pouvez mettre entre crochets n'importe quel texte ou image (comme dans le cas ci-dessus).

## 7.4 Ajouter des lignes