

# Premiers pas avec LaTeX : Introduction

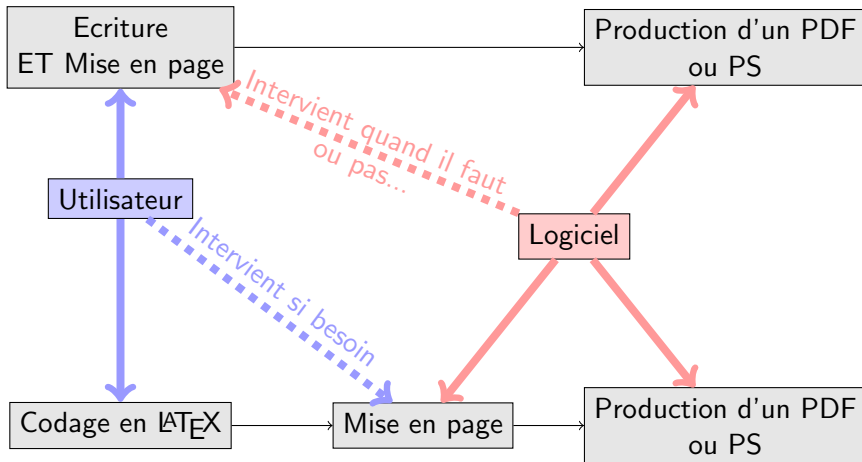
Jean-Baptiste Boni

19 mars 2021

## Definition

WYDIWYG (What You Do Is What You Get) comprend les logiciels de traitements de textes comme MS-Word, OpenOffice ou encore LibreOffice. C'est-à-dire que la mise en page doit être faite par l'utilisateur via une interface graphique.

$\text{\LaTeX}$  n'en est pas un !

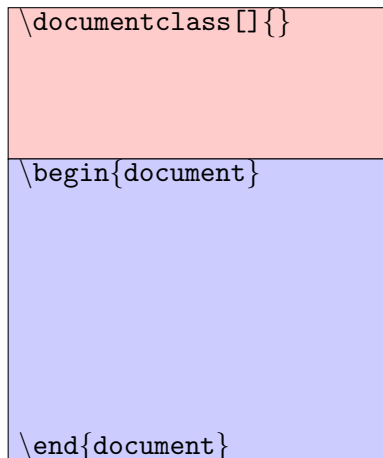


La mise en page n'étant pas faite par l'utilisateur, elle est faite par des “*packages*” codés par la communauté. Il faut donc un compilateur pour les utiliser tous ensemble. Deux grands compilateurs sont disponibles :

- ▶ *MikTeX*, qui est un compilateur très simple et qui contient le minimum syndical pour compiler un code LaTeX pas trop complexe.
- ▶ *TeXlive*, qui lui est beaucoup plus complet et contient un grand nombre d'options.

Par ailleurs, il est intéressant, bien que pas obligatoire, d'utiliser un éditeur de code LaTeX qui vous permettra de bien structurer votre code et d'être plus efficace.

Solution en ligne : *OverLeaf* (<https://www.overleaf.com/>)



Le préambule du document :  
il contient l'appel à tous les packages que vous voudrez utiliser (ou pas) et les options dont ils ont besoin pour fonctionner

Le corps du document :  
ici, vous écrivez votre texte et faites appel aux différents packages qui dirigeront la mise en page

## Les classes

Les classes de documents sont des normes de mise en page de natures différentes. Les normes de mise en page ne sont pas les mêmes qu'on écrive un article, un livre ou une lettre ! MikTeX propose plusieurs classes basiques qui font très bien le boulot. On retiendra article, report, book et thesis.

Une classe doit **TOUJOURS** être déclarée en première dans votre code par la commande :

```
\documentclass[<options de la classe>]{<nom de la classe>}
```

## Les packages

Un package est en fait du code TeX qui dirige la mise en page et la définition de certaines fonctions. Des exemples de packages usuels :

**graphicx** permet d'inclure des images dans votre document,

**tabularx** permet de faire des tableaux,

**babel** permet de faire une typographie spécifique à une langue,

**ulem** permet de souligner du texte...

Beaucoup de packages sont déjà chargés lors de la déclaration de la classe de document que vous utilisez. Pour charger un package manuellement, on écrira :

```
\usepackage[<options du package>]{<nom du package>}
```

## Les environnements

Un environnement est une partie de texte sur laquelle on applique une mise en page particulière (police en gras, en couleur, soulignée, itemisée...). Un environnement sera toujours encapsulé entre deux balises :

```
\begin{<environnement>}
\end{<environnement>}
```

Certains environnements simples peuvent être abrégés par des fonctions entre crochets :

```
\textbf{<texte>}
{\bfseries <texte>}
```



## Chapitres, sections, sous-sections...

Pour structurer un document sous formes de sections, des fonctions prédéfinies sont disponibles :

```
\chapter{<titre>} (sauf pour la classe article)
\section{<titre>}
\subsection{<titre>}
\subsubsection{...}
```

Ils permettront de créer des chapitres et sections qui apparaîtront dans la table des matières. Si vous ne voulez pas faire apparaître une section dans la table des matières, il suffit de rajouter une \* :

```
\section*{<texte>}
```

Elle apparaîtra alors dans le corps de texte mais non numérotée.

# Bibliographie et citations

La gestion de la bibliographie et de la citation dans un script Tex se fait via un fichier BibTeX. Ce format est normé et est se présente comme un listing des propriétés d'une entrée bibliographique... Bref, sous Mendeley, sélectionnez les articles que vous souhaitez citer puis sélectionnez "Exporter" puis choisissez le format BibTeX. Le fichier se retrouve au format *.bib*

**Attention !** Ce fichier doit se trouver dans le même dossier que votre script TeX où vous souhaitez citer vos entrées.

On dit ensuite au compilateur quel fichier de biblio est utilisé par la fonction *bibliography*.

```
\bibliography{<nom du fichier>}
```

Il faut ensuite choisir le style des citations et de la table des références en utilisant *bibliographystyle*. Je vous laisse regarder les différents styles par défaut (plain, alpha, abbr...). Certains journaux ont créé leurs styles de référence...

```
\bibliographystyle{<nom du style>}
```

## Definition

Un environnement de figure ou de table est un **flottant**, c'est-à-dire que TeX considère tout ce qui est dans cet environnement comme "bougeable" d'un bloc. Il choisira donc toujours, selon lui, le meilleur endroit pour ce flottant.

Les environnements de figure, de table et d'équations sont, par exemple, flottants mais permettent aussi d'autres choses : comptage des figures, gestion de la légende, des sous-figures, des références...

## Exemple d'insertion d'une image dans un environnement figure

```
\begin{figure}
    \includegraphics{imagefile}
    \caption{La légende de l'image}
    \label{fig:label pour y faire référence}
\end{figure}
```

A noter que la fonction *includegraphics* est disponible par le package *graphicx*.

## Comment construire le tableau ?

*tabular* est très simpliste, il gère tout seul la largeur des colonnes, il suffit de lui dire l'alignement des colonnes que l'on souhaite. Ces alignements s'écrivent par des abréviations :

- ▶ *c* pour centré
- ▶ *l* pour aligné à gauche
- ▶ *r* pour aligné à droite

On indiquera ces abréviations autant de fois que l'on veut de colonnes.

Il faut aussi lui indiquer si l'on veut des lignes séparatrices entre les colonnes. Pour cela, on lui ajoutera le symbole `|` entre les précédentes abréviations.

Donc si je veux un tableau de 5 colonnes dont la première est centrée et les autres alignées à gauche avec des séparations entre chaque colonne, j'écrirai :

```
\begin{tabular}{|c|l|l|l|l|}  
    contenu...  
\end{tabular}
```

Un tableau se construit grâce à la commande *tabular*. Pour qu'il soit dans un environnement **flottant**, il faut l'encapsuler dans un environnement *table*.

```
\begin{table}
  \begin{tabular}{options}
    contenu...
  \end{tabular}
\end{table}
\caption{La légende du tableau}
\label{tab:label pour y faire référence}
```



Pour remplir un tableau, on délimite les colonnes par des `&` et chaque ligne de code correspond à une ligne, on la termine par `\\`. Pour tracer une ligne horizontale on utilise la commande `hline`.

```
\begin{tabular}{|c|c|c|}
  1 & 2 & 3\\
  \hline
  4 & 5 & 6\\
  \hline
\end{tabular}
```

1	2	3
4	5	6

Le package *standalone* permet de séparer votre code en plusieurs scripts et les compiler indépendamment. Je vous conseille de l'utiliser régulièrement pour éviter d'avoir les codes de vos graphes dans votre script principal. Je vous renvoie à l'exemple.

## Definition

**Tikz** (et par extension **PGFplot**) permet de dessiner dans LaTeX, tracer des formes etc. PGFplot est un enfant de ce plus gros package qui est spécifique au tracé de graphes (2D ou 3D).