

# Séminaire L<sup>A</sup>T<sub>E</sub>X

*Ou pourquoi Word c'est le mal*

Pigeon Moelleux

21 novembre 2022

# Outline

- 1 Introduction
- 2 Pause installation
- 3 Pourquoi  $\text{\LaTeX}$  est mieux que Word
- 4 Structure
- 5 Macros et environnements
  - Macros
  - Environnements
- 6 Packages
- 7 Où trouver de la documentation
- 8 Mini-TP

# De T<sub>E</sub>X à L<sup>A</sup>T<sub>E</sub>X

## T<sub>E</sub>X

- Logiciel libre de composition de documents
- Créé en 1977 par Donal Knuth
- Utilisé notamment dans les domaines scientifiques

# De T<sub>E</sub>X à L<sup>A</sup>T<sub>E</sub>X

## T<sub>E</sub>X

- Logiciel libre de composition de documents
- Créé en 1977 par Donal Knuth
- Utilisé notamment dans les domaines scientifiques

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

- Aussi un logiciel libre et système de composition de documents
- Collection de macro-commandes facilitant l'utilisation de T<sub>E</sub>X

# De T<sub>E</sub>X à L<sup>A</sup>T<sub>E</sub>X

## T<sub>E</sub>X

- Logiciel libre de composition de documents
- Créé en 1977 par Donal Knuth
- Utilisé notamment dans les domaines scientifiques

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

- Aussi un logiciel libre et système de composition de documents
- Collection de macro-commandes facilitant l'utilisation de T<sub>E</sub>X
- **On prononce « tek » et non « teks » !**

# Exemples d'utilisation

## Lettre de motivation

Pigeon Moelleux  
Adresse quelconque

Gif-sur-Yvette,  
30 minutes avant la présentation

Madame, Monsieur,  
Souhaitant faire un séminaire de  
 $\text{\LaTeX}$ , je suis en train de trouver des  
idées de choses à rédiger mais je n'y  
arrive malheureusement pas beau-  
coup.

## Article scientifique

Article scientifique random  
MOI MÊME  
Aujourd'hui

### I. Introduction

#### 1. Définitions préliminaires

**Lemme 1** : J'apprécie les fruits au sirop

On a par récurrence immédiate : 
$$\sum_{k=0}^n k = \frac{n(n+1)}{2}$$

# Exemples d'utilisation

## Lettre de motivation

Pigeon Moelleux  
Adresse quelconque

Gif-sur-Yvette,  
30 minutes avant la présentation

Madame, Monsieur,  
Souhaitant faire un séminaire de  
 $\LaTeX$ , je suis en train de trouver des  
idées de choses à rédiger mais je n'y  
arrive malheureusement pas beau-  
coup.

## Article scientifique

Article scientifique random  
MOI MÊME  
Aujourd'hui

### I. Introduction

#### 1. Définitions préliminaires

**Lemme 1** : J'apprécie les fruits au sirop

On a par récurrence immédiate :  $\sum_{k=0}^n k = \frac{n(n+1)}{2}$

*Alors normalement c'est plus beau mais je voulais tout faire en un document et sans utiliser d'images*

# Outline

- 1 Introduction
- 2 Pause installation**
- 3 Pourquoi  $\text{\LaTeX}$  est mieux que Word
- 4 Structure
- 5 Macros et environnements
  - Macros
  - Environnements
- 6 Packages
- 7 Où trouver de la documentation
- 8 Mini-TP



# Installation

## Ubuntu/Debian

```
sudo apt install texlive-full
```

## Windows et MacOS

Plusieurs distributions sont disponibles (voir TeXLive ou MiKTeX)

## Archlinux, VoidLinux, NixOS

Vous devrez être capable de trouver ^^

## En ligne

Overleaf : <https://overleaf.com> (utile pour les projets de groupe)

# Outline

- 1 Introduction
- 2 Pause installation
- 3 Pourquoi L<sup>A</sup>T<sub>E</sub>Xest mieux que Word**
- 4 Structure
- 5 Macros et environnements
  - Macros
  - Environnements
- 6 Packages
- 7 Où trouver de la documentation
- 8 Mini-TP

# Points généraux

## Concepts clefs de $\LaTeX$

- Langage balisé (comme le HTML ou le XML)
- Permet de concevoir des documents de haute qualité (en particulier des PDF (ou des DVI si vous êtes vieux))

# Points généraux

## Concepts clefs de $\text{\LaTeX}$

- Langage balisé (comme le HTML ou le XML)
- Permet de concevoir des documents de haute qualité (en particulier des PDF (ou des DVI si vous êtes vieux))

## Philosophie

- Simple à utiliser après une courte période d'apprentissage
- Indépendant de l'éditeur de texte
- Facilité de partage

# Points généraux

## Concepts clefs de $\text{\LaTeX}$

- Langage balisé (comme le HTML ou le XML)
- Permet de concevoir des documents de haute qualité (en particulier des PDF (ou des DVI si vous êtes vieux))

## Philosophie

- Simple à utiliser après une courte période d'apprentissage
- Indépendant de l'éditeur de texte
- Facilité de partage
- **WYSIWYM** : *What you see is what you mean*

# Points généraux

## Concepts clefs de L<sup>A</sup>T<sub>E</sub>X

- Langage balisé (comme le HTML ou le XML)
- Permet de concevoir des documents de haute qualité (en particulier des PDF (ou des DVI si vous êtes vieux))

## Philosophie

- Simple à utiliser après une courte période d'apprentissage
- Indépendant de l'éditeur de texte
- Facilité de partage
- **WYSIWYM** : *What you see is what you mean*  
Pas besoin de se concentrer sur la mise en page : il suffit de se concentrer sur le fond, contrairement à d'autres moyens de mise en forme de documents de type **WYSIWYG** (comme Word ou Libre Office)

# Outline

- 1 Introduction
- 2 Pause installation
- 3 Pourquoi  $\text{\LaTeX}$  est mieux que Word
- 4 Structure**
- 5 Macros et environnements
  - Macros
  - Environnements
- 6 Packages
- 7 Où trouver de la documentation
- 8 Mini-TP

# Enfin du code !

Bon on blablate beaucoup, mais on a pas encore vu de code !



# Enfin du code !

Bon on blablate beaucoup, mais on a pas encore vu de code !

## Exemple minimal

```
\documentclass{minimal}

\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[french]{babel}

\begin{document}
  Hello world!
\end{document}
```

# Enfin du code !

Bon on blablate beaucoup, mais on a pas encore vu de code !

## Exemple minimal

```
\documentclass{minimal}

\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[french]{babel}

\begin{document}
  Hello world!
\end{document}
```

## Compilation du code

```
pdflatex <fichier>.tex
```

# Syntaxe générale (1/3)

Parties du code

# Syntaxe générale (1/3)

## Parties du code

- Ce qui n'est pas affiché : définition de la classe de document, importation des packages, définition des commandes, ...

# Syntaxe générale (1/3)

## Parties du code

- Ce qui n'est pas affiché : définition de la classe de document, importation des packages, définition des commandes, ...
- Ce qui est affiché : tout ce qui se trouve entre `\begin{document}` et `\end{document}` est affiché directement (sauf les commandes et les caractères spéciaux qui ont une signification particulière)

# Syntaxe générale (1/3)

## Parties du code

- Ce qui n'est pas affiché : définition de la classe de document, importation des packages, définition des commandes, ...
- Ce qui est affiché : tout ce qui se trouve entre `\begin{document}` et `\end{document}` est affiché directement (sauf les commandes et les caractères spéciaux qui ont une signification particulière)

# Syntaxe générale (2/3)

## Caractères spéciaux

Symbole	Affichage	Signification
<code>\</code>	<code>\textbackslash</code>	Début de commande
<code>#</code>	<code>\#</code>	Début d'argument dans les déclarations
<code>@</code>	<code>@</code>	C'est le bordel
<code>\$</code>	<code>\\$</code>	Environnement mathématique
<code>%</code>	<code>\%</code>	Commentaire
<code>^ et _</code>	<code>\^ et \_</code>	Réservé aux environnements mathématiques
<code>~</code>	<code>\~</code>	Espace insécable
<code>{ et }</code>	<code>\{ et \}</code>	Début et fin de groupe

# Syntaxe générale (3/3)

## Commandes

- Sont présentes absolument partout en  $\text{\LaTeX}$
- Commencent par un  $\backslash$  (antislash en français, ou *backslash* en anglais, ou *fordított perjel* en hongrois)



# Syntaxe générale (3/3)

## Commandes

- Sont présentes absolument partout en  $\text{\LaTeX}$
- Commencent par un `\` (antislash en français, ou *backslash* en anglais, ou *fordított perjel* en hongrois)

## Exemple

```
\textbf{Ceci est en gras}, \textit{et ceci est en italique}.
```

# Syntaxe générale (3/3)

## Commandes

- Sont présentes absolument partout en  $\text{\LaTeX}$
- Commencent par un  $\backslash$  (antislash en français, ou *backslash* en anglais, ou *fordított perjel* en hongrois)

## Exemple

```
\textbf{Ceci est en gras}, \textit{et ceci est en italique}.
```

**Ceci est en gras**, *et ceci est en italique.*

# Exemple de document (1/2)

## Code

```
\documentclass[a4paper, 12pt]{article}

\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[french]{babel}

\title{Ceci n'est pas un document}
\author{Moi-même}
\date{Demain}

\begin{document}
  \maketitle
  \section{Ceci est une partie}
    \subsection{Ceci est une sous-partie}
      La numérotation est automatique.

    \subsection{Et une autre sous-partie}
      Les sous-parties existent aussi !
\end{document}
```

# Exemple de document (2/2)

## Rendu

Ceci n'est pas un document

Moi-même

Demain

### **1 Ceci est une partie**

#### **1.1 Ceci est une sous-partie**

La numérotation est automatique.

#### **1.2 Et une autre sous-partie**

Les sous-parties existent aussi!

# Outline

- 1 Introduction
- 2 Pause installation
- 3 Pourquoi  $\text{\LaTeX}$  est mieux que Word
- 4 Structure
- 5 Macros et environnements**
  - Macros
  - Environnements
- 6 Packages
- 7 Où trouver de la documentation
- 8 Mini-TP

# Introduction aux macros

## Macros

- Rappel :  $\LaTeX$  est un ensemble de macros de  $\TeX$
- Équivalent d'une fonction définie par l'utilisateur
- Utilisation : `\commande[option1, ..., optionN]{arg1}...{argM}`

# Introduction aux macros

## Macros

- Rappel :  $\LaTeX$  est un ensemble de macros de  $\TeX$
- Équivalent d'une fonction définie par l'utilisateur
- Utilisation : `\commande[option1, ..., optionN]{arg1}...{argM}`

## Exemples d'utilisations

Code $\LaTeX$	Résultat
<code>\textit{Texte en italique}</code>	<i>Texte en italique</i>
<code>\$\$\frac{\partial f}{\partial x}(x) = x\$</code>	$\frac{\partial f}{\partial x}(x) = x$
<code>\fbox{Texte encadré}</code>	<div style="border: 1px solid black; padding: 2px; display: inline-block;">Texte encadré</div>

# Crée ses propres macros

## Syntaxe de la création de macros

```
\newcommand{\macro}[nb d'arguments]{Ce que ça fait}
```

On peut alors désigner l'argument n<sup>o</sup>*i* par #*i*.



# Crée ses propres macros

## Syntaxe de la création de macros

```
\newcommand{\macro}[nb d'arguments]{Ce que ça fait}
```

On peut alors désigner l'argument n<sup>o</sup>*i* par #*i*.

## Exemples

```
\newcommand{\jump}{\vspace{1\baselineskip}}
```

# Crée ses propres macros

## Syntaxe de la création de macros

```
\newcommand{\macro}[nb d'arguments]{Ce que ça fait}
```

On peut alors désigner l'argument n<sup>o</sup>*i* par #*i*.

## Exemples

```
\newcommand{\jump}{\vspace{1\baselineskip}}
```

```
\newcommand{\derivate}[2]{\frac{\partial #1}{\partial #2}}
```

# Crée ses propres macros

## Syntaxe de la création de macros

```
\newcommand{\macro}[nb d'arguments]{Ce que ça fait}
```

On peut alors désigner l'argument n<sup>o</sup>*i* par #*i*.

## Exemples

```
\newcommand{\jump}{\vspace{1\baselineskip}}
```

```
\newcommand{\derivate}[2]{\frac{\partial #1}{\partial #2}}
```

$\$ \backslash \text{derivate} \{ f \} \{ x \} \$ \longrightarrow \frac{\partial f}{\partial x}$  et  $\$ \backslash \text{derivate} \{ g \} \{ y \} \$ \longrightarrow \frac{\partial g}{\partial y}$

# Introduction aux environnements

## Environnements

- Omniprésents en  $\text{\LaTeX}$

# Introduction aux environnements

## Environnements

- Omniprésents en  $\text{\LaTeX}$
- Débutent avec `\begin{...}` et terminent avec `\end{...}`

# Introduction aux environnements

## Environnements

- Omniprésents en  $\text{\LaTeX}$
- Débutent avec `\begin{...}` et terminent avec `\end{...}`
- Servent à baliser le code (texte en italique, en gras, début et fin d'une partie spécifique, d'une liste, ...)

# Introduction aux environnements

## Environnements

- Omniprésents en  $\text{\LaTeX}$
- Débutent avec `\begin{...}` et terminent avec `\end{...}`
- Servent à baliser le code (texte en italique, en gras, début et fin d'une partie spécifique, d'une liste, ...)

## Exemples

```
\begin{itemize}
  \item Ceci est une liste à puces
  \item Ceci est un second point
\end{itemize}
```

# Introduction aux environnements

## Environnements

- Omniprésents en  $\text{\LaTeX}$
- Débutent avec  $\text{\begin}\{\dots\}$  et terminent avec  $\text{\end}\{\dots\}$
- Servent à baliser le code (texte en italique, en gras, début et fin d'une partie spécifique, d'une liste, ...)

## Exemples

```
 $\text{\begin}\{itemize}\$   
   $\text{\item}$  Ceci est une liste à puces  
   $\text{\item}$  Ceci est un second point  
 $\text{\end}\{itemize}\$ 
```

- Ceci est une liste à puces
- Ceci est un second



# Exemple courant : les environnements mathématiques

## Environnements mathématiques

Deux types d'environnements mathématiques usuels : inline et displayed.

# Exemple courant : les environnements mathématiques

## Environnements mathématiques

Deux types d'environnements mathématiques usuels : inline et displayed.

### Inline

- Délimité par  $\backslash(\dots\backslash)$  ou par  $\$ \dots \$$
- Insert des formules mathématiques au milieu du texte

# Exemple courant : les environnements mathématiques

## Environnements mathématiques

Deux types d'environnements mathématiques usuels : inline et displayed.

### Inline

- Délimité par  $\backslash(\dots\backslash)$  ou par  $\$\dots\$\$
- Insert des formules mathématiques au milieu du texte

### Displayed

- Délimité par  $\backslash[\dots\backslash]$  ou par  $\$\$\dots\$\$\$
- Insert des formules mathématiques de manière détachée du texte

## Exemples

Ceci est une fraction :  $\$\frac{1}{2}\$$  !

Ceci est une fraction :  $\frac{1}{2}$  !

# Outline

- 1 Introduction
- 2 Pause installation
- 3 Pourquoi  $\text{\LaTeX}$  est mieux que Word
- 4 Structure
- 5 Macros et environnements
  - Macros
  - Environnements
- 6 Packages**
- 7 Où trouver de la documentation
- 8 Mini-TP

# Packages

## Importer un package

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

# Packages

## Importer un package

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

## Créer ses propres packages

Il est possible de créer ses propres packages facilement, et de les utiliser dans ses documents (prochain séminaire?).

# Packages

## Importer un package

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

## Créer ses propres packages

Il est possible de créer ses propres packages facilement, et de les utiliser dans ses documents (prochain séminaire?).

## Exemples de packages très utiles

- **tikz** : *TOUT* ce qui se rapproche de près ou de loin à un graphique
- **listings** : afficher du code proprement
- **hyperref** : créer des hyperliens (comme ici ^^)

# Outline

- 1 Introduction
- 2 Pause installation
- 3 Pourquoi  $\text{\LaTeX}$  est mieux que Word
- 4 Structure
- 5 Macros et environnements
  - Macros
  - Environnements
- 6 Packages
- 7 Où trouver de la documentation
- 8 Mini-TP



# Documentation L<sup>A</sup>T<sub>E</sub>X

- CTAN : <https://ctan.org/> : pour la documentation des packages
- WikiBooks LaTeX : <https://en.wikibooks.org/wiki/LaTeX> : regroupe beaucoup de pages documentant les fonctionnalités de L<sup>A</sup>T<sub>E</sub>X
- StackExchange : <https://tex.stackexchange.com> : sauve la vie
- Overleaf : <https://fr.overleaf.com/learn> : tutoriels pour tous les niveaux !

# Outline

- 1 Introduction
- 2 Pause installation
- 3 Pourquoi  $\text{\LaTeX}$  est mieux que Word
- 4 Structure
- 5 Macros et environnements
  - Macros
  - Environnements
- 6 Packages
- 7 Où trouver de la documentation
- 8 Mini-TP

# À vous d'écrire

## On commence doucement

Ceci est du texte, et si  $x = 1$ , alors  $2 \times x = 2$ .

## On fait des listes aussi

- Ceci est un point
- $\frac{1}{2} = 0.5$  ce qui montre que je n'ai **plus du tout** d'idée
- J'aime bien les choux-fleurs

## Et même des tableaux

Colonne 1	<b>coucou</b>	Aligné à droite
Nom de ligne 1	$\sqrt{4} = 2$	yoyo
$e^{i\pi} + 1 = 0$	Voici du rien	$1 + 1 = 2$