

# LaTeX

Par Dr. CHAIBI Hasna

# Histoire de LaTeX

- Dans les années **70**, un informaticien du nom de **Donald Knuth** trouvait que les logiciels à sa disposition pour écrire du texte avec des formules mathématiques n'étaient pas assez bons.
- Il décide alors (avec l'aide d'autres personnes) d'écrire le sien, qui sortira en **1978** sous le nom de **TeX**.
- Le but était de fournir un outil simple à utiliser pour composer des textes d'une grande qualité typographique. **TeX** comporte environ 300 commandes. On les appelle des **primitives**.
- Voyant que certaines fonctions étaient compliquées à utiliser, **Donald Knuth** complète **TeX** en créant des commandes pour faciliter son utilisation.

# Histoire de LaTeX

- Le langage ainsi obtenu (combinaison de TeX et de **commandes**) est appelé un **format**. Il s'agit du format **Plain TeX**.
- Malgré cela, le format **Plain TeX** reste compliqué et les utilisateurs ont commencé à écrire des extensions pour simplifier la composition des textes et éviter de tout réécrire depuis zéro.
- Au début des années **80**, **Leslie Lamport** fusionne un grand nombre d'extensions afin d'obtenir un langage de plus haut niveau: **LaTeX**.

# Qu'est ce que LATEX?

- **LaTeX** permet de rédiger des documents dont la **mise en page** est réalisée **automatiquement** en se conformant du mieux possible à des normes typographiques. Une fonctionnalité distinctive de **LaTeX** est, son mode **mathématique**, qui permet de composer des **formules complexes**.
- **LaTeX** est particulièrement utilisé dans les domaines techniques et scientifiques pour la production de documents de taille moyenne (tels que des **articles**, **CVs ou des présentations** ) ou importante (**thèses**, **mémoires** ou **livres**, par exemple).

# Typographie et LaTeX

- La **typographie** désigne l'art et la technique de la mise en forme des textes écrits. Elle inclut tout ce qui concerne l'apparence des lettres et des mots, comme :
- **Les polices de caractères** (Arial, Times New Roman, etc.)
- **Les tailles de police** (petit, moyen, grand)
- **L'espacement** entre les lettres, les mots et les lignes
- **L'alignement** (justifié, centré, à gauche, à droite)
- **Les styles** (gras, italique, souligné, etc.)
- **L'hyphénation** (coupure des mots en fin de ligne)

# Typographie et LaTeX

- En **LaTeX**, la typographie est très soignée automatiquement.  
Par exemple :
  - Les espaces et l'alignement sont bien gérés.
  - Les césures des mots en fin de ligne sont optimisées.
  - Les polices sont bien ajustées pour un rendu professionnel.
- la typographie, c'est tout ce qui rend un texte **beau, clair et agréable à lire !**

# Principe

- **LATEX** peut être considéré comme un langage de programmation évolué dans le sens où il s'appuie sur **TeX** qui est un langage de plus bas niveau. Cela signifie que le document que l'on veut créer doit être décrit dans un fichier source (mon\_fichier.tex par exemple), composé d'un ensemble de commandes (balises) **LATEX** puis doit être compilé.
- Le compilateur **LATEX** fournit en sortie un fichier DVI (device independent) (mon\_fichier.dvi). Ce fichier peut alors lui même être converti en format Postscript ou PDF pour être imprimé/exporté. La plupart des commandes **LATEX** se caractérisent par le fait qu'elles commencent par un "**backslash**" \, que leurs arguments obligatoires apparaissent entre accolades ({ et }) et que leurs arguments optionnels apparaissent entre crochets ([ et ]). Ex : \documentclass[12pt]{report}

# Installation

- Pour commencer avec LaTeX, voici ce que vous devez installer :
1. **Un moteur LaTeX** C'est le cœur du système, il transforme votre code LaTeX en un document PDF. Vous avez plusieurs choix :
    - **TeX Live** (recommandé sous Linux et Windows)
    - **MiKTeX** (alternative pour Windows)
    - **MacTeX** (pour macOS).
  2. **Un éditeur LaTeX** Pour écrire et compiler vos documents facilement, vous pouvez utiliser :
  3. **TeXworks** : inclus avec MiKTeX et TeX Live
  4. **TeXstudio** : très complet et recommandé
  5. **Overleaf** : éditeur en ligne, pas besoin d'installation

# Installation

- Lien de téléchargement:
- Distribution: <https://miktex.org/download>
- Editeur 1 - TexStudio: <https://www.texstudio.org/>

# Les commandes, la base de LaTeX

- Les **commandes** sont le cœur de LaTeX. Elles permettent par exemple d'indiquer que telle chose est un titre, qu'une autre est importante, etc.

```
1 \commande
```

- Une commande commence par un anti-slash suivi de son nom. L'éditeur de texte choisi est censé mettre les commandes en évidence. De plus, une commande peut prendre des arguments. Il y a deux types d'arguments, les arguments obligatoires et les arguments optionnels.

# Les commandes, la base de LaTeX

- Par exemple, une commande `\titre` pourrait prendre en argument obligatoire le titre de notre document et en argument facultatif son sous-titre s'il y en a un.
- On les indique de cette manière: les arguments optionnels (ou options) suivent le nom de la commande et sont placés entre crochets. Il peut également y en avoir plusieurs. Les arguments obligatoires, eux, sont entre accolades.
- Voyons maintenant des exemples de commande.

```
1 \commande
2 \commande{argument}
3 \commande[option]
4 \commande[option]{argument}
5 \commande[option]{argument}{autre_argument}
```

# Les commandes, la base de LaTeX

- \commande (Commande sans argument ni option)

```
\newpage % Insère une nouvelle page
```

- \commande{argument} (Commande avec un argument obligatoire)

```
\textbf{Texte en gras} % Met "Texte en gras" en gras
```

- \commande[option]{argument} (Commande avec une option et un argument obligatoire)

```
\includegraphics[width=15cm, height=10cm]{super_logo.png}
```

- \commande {argument}{autre\_argument} (Commande avec deux arguments obligatoires)

```
\textcolor{red}{Texte en rouge}
```

# Les packages, un système d'extensions

- **LaTeX** en lui-même est très simple. Par exemple, à moins de l'écrire nous même, il n'existe pas de commande pour charger des images. Pour pallier ce manque, nous pouvons étendre les fonctionnalités de **LaTeX** à l'aide d'extensions appelées **packages**.
- Pour charger un **package**, on utilise la commande **\usepackage**. Elle prend en paramètre obligatoire le nom du package à charger. Certains packages ont des options, qu'on peut charger en passant des arguments facultatifs à la commande.

```
1 \usepackage[option]{package}
```

- Il est possible de charger **plusieurs packages** avec une seule commande. Pour cela, il suffit de séparer leurs noms par une virgule.

```
1 \usepackage{package, autre_package}
```

Il est déconseillé de le faire lorsque l'un des packages nécessite des options

# Structure générale d'un document LATEX

```
\documentclass[11pt,twoside,a4paper]{article}
%===== En-Tete =====
%--- Insertion de paquetages (optionnel) ---

\usepackage[utf8]{inputenc} % Encodage UTF-8

%===== Corps =====
\begin{document} % Marque le début du texte à composer
    Le corps de document ...
\end{document} % Marque le Fin du document
```

# Structure générale d'un document LATEX

- Un document LATEX commence donc toujours par une commande `document class` permettant de spécifier la classe de document (entre accolades). Les classes les plus utilisées sont **article**, **report**, **letter**, et **book**.
- Les options de cette commande sont déclarées entre crochets. Parmi les plus courantes, on mentionnera **10pt**, **11pt**, **12pt** (qui déterminent la taille nominale des polices),
- **a4paper** (pour les dimensions de la feuille de papier), **french**, **twocolumn** (pour la disposition du texte sur deux colonnes)

# Format de papier

- Le format de papier est une option qui peut s'avérer intéressante dans le cas où nous voulons un document spécifique. Le papier par défaut est soit au format **A4**, soit au format **letter** (le format par défaut dépend de notre installation). Nous disposons néanmoins de plusieurs autres formats.

Format	Taille en mm	Utilisation courante
a4paper	210 × 297	Standard en Europe
a5paper	148 × 210	Petits livres, carnets
b5paper	176 × 250	Livres et brochures
letterpaper	216 × 279	Standard aux USA
legalpaper	216 × 356	Documents légaux (USA)

# Texte sur Deux Colonnes en LaTeX

- En LaTeX, il est possible d'écrire un document sur **deux colonnes** en utilisant l'option **twocolumn** dans la classe de document ou avec l'environnement **multicols**.

1-Avec l'option **twocolumn** dans **\documentclass**

```
\documentclass[twocolumn]{article}
```

2- Avec le package **multicol** (Flexible et personnalisable)

```
\begin{multicols}{2} % Début du texte sur deux colonnes  
    Texte à mettre en deux colonnes  
\end{multicols} % Fin du texte en colonnes
```

# Structure générale d'un document LATEX

- En pratique, l'entête permet de définir un certain nombre de configurations, notamment les packages utilisés (par la commande `\usepackage`), les redéfinitions de commandes, le titre, le style bibliographique etc. Ce que nous écrirons avant `\begin{document}` ou après `\end{document}` ne sera pas affiché.

# Structure générale d'un document LATEX

## ✓ Les accents posent des problèmes!

- Cela est dû à ce que l'on appelle **l'encodage**. L'encodage est «la façon de transcrire un texte grâce aux codes des caractères qui le composent, selon un jeu de caractères donné». Il faut donc indiquer à **LaTeX** dans quel encodage notre fichier est enregistré.
- **Il existe plusieurs encodages.** Les encodages par défaut diffèrent suivants les systèmes, mais certains encodages sont, bien heureusement, disponibles sous tous les systèmes. Nous conseillons d'utiliser **UTF-8** partout. Il a l'avantage d'être disponible sur la majorité des systèmes et de gérer de nombreux caractères (en particulier les accents).

# Les packages inputenc et fontenc

❑ \usepackage[utf8]{inputenc} : Gestion de l'encodage des fichiers source

- Ce package **définit l'encodage du fichier .tex**.
- Il permet à LaTeX de comprendre correctement les caractères spéciaux (accents, symboles...).
- Par défaut, LaTeX utilise l'encodage ASCII, qui ne supporte pas directement les caractères accentués comme é, è, à, etc.
- Avec utf8, vous pouvez écrire directement en UTF-8 sans erreurs.
- **Sans inputenc, vous devriez écrire : \E au lieu de É.**

# Les packages inputenc et fontenc

## ❑ \usepackage[T1]{fontenc} : Gestion des polices et de l'affichage des caractères

- Ce package **définit l'encodage des polices** utilisées pour générer le document PDF.
- Il améliore la typographie et la gestion des accents dans les PDF.
- L'encodage standard de LaTeX (OT1) ne gère pas bien certains caractères accentués, ce qui peut poser problème pour le copier-coller et l'hyphénation des mots accentués.
- T1 améliore ces aspects en stockant les accents comme des caractères distincts.

## Exemple sans T1 :

- Le mot "**économie**" pourrait être mal césuré en fin de ligne.
- En copiant un mot accentué depuis le PDF, l'accent pourrait disparaître.
- Avec T1, ces problèmes disparaissent.

# Les paragraphes et l'écriture

- Le bloc principal que nous allons utiliser est le paragraphe. Et il s'agit bien d'un bloc aussi bien dans notre code que dans le document pdf. Pour le rédacteur, un changement de paragraphe correspond à un changement d'idée (la plupart du temps).
- Les deux règles essentielles qui concernent les paragraphes.
  - Un retour à la ligne n'est pas interprété comme un changement de paragraphe, mais comme une espace. Pour changer de paragraphe, il faut sauter une ligne.
  - Sauter plusieurs lignes revient à sauter une ligne, donc à changer de paragraphes. En fait, deux retours à la ligne consécutifs sont équivalents à la primitive `\par`. Elle permet d'indiquer explicitement à Tex la fin d'un paragraphe

# Les paragraphes et l'écriture

- En LaTeX, vous pouvez indiquer des paragraphes de plusieurs manières :

**1- En laissant une ligne vide entre deux blocs de texte** (Méthode recommandée)

```
%===== Corps =====
\begin{document} % Marque le début du texte à composer
    Ceci est le premier paragraphe.
    Ceci est le deuxième paragraphe.
\end{document} % Marque le Fin du document
```



# Les paragraphes et l'écriture

- 2. Avec la commande \par

```
\begin{document} % Marque le début du texte à composer  
Ceci est un premier paragraphe. \par  
Ceci est un deuxième paragraphe.  
\end{document} % Marque le Fin du document
```

- 3. Avec l'environnement \begin{paragraph}... \end{paragraph}

```
\begin{paragraph}{Introduction}  
Ceci est un paragraphe qui commence par un titre en  
gras.  
\end{paragraph}
```

# Des caractères réservés

- Essayons maintenant d'écrire un document avec quelques symboles.

```
%===== Corps =====
▼ \begin{document} % Marque le début du texte à composer
  Dans mon document, je peux écrire de smileys :), ^, :\ et
  même
  devenir riche$.
✖ \end{document} % Marque le Fin du document
```

- On obtient une erreur. Notre code ne compile pas. Cela est dû au fait que certains caractères sont réservés. Cela veut dire que nous ne pouvons pas les taper tels quels dans le code pour les afficher dans le document final.

# Caractères réservés

Caractère	Description	Comment l'afficher en LaTeX
#	Symbole de dièse (hash)	\#
\$	Symbole dollar (utilisé pour les formules mathématiques)	\\$
%	Pourcentage (utilisé pour les commentaires)	\%
&	Symbol "et commercial" (utilisé dans les tableaux)	\&
_	Tiret bas (underscore, utilisé dans les indices)	\_
{ }	Accolades (utilisées pour grouper des commandes)	\{ \}
\	Anti-slash (utilisé pour les commandes LaTeX)	\textbackslash ou \\$\backslash\\$
~	Espace insécable (préserve l'espace entre mots)	\textasciitilde ou \\$\sim\\$
^	Exposant (utilisé en mathématiques)	\textasciicircum ou \\$^\wedge\\$

# L'espace insécable

- En LaTeX, l'**espace insécable** (~) est utilisé pour empêcher une coupure de ligne entre deux éléments. Il garantit que les mots ou les symboles de part et d'autre restent ensemble sur la même ligne.
  - L'espace insécable est utile dans plusieurs cas :
    - 1- **Empêcher une coupure entre un mot et un numéro**

**Référez-vous au chapitre~3.**
- **Sans ~ → "3"** pourrait être envoyé à la ligne suivante.
  - **Avec ~ → "chapitre 3"** restera toujours ensemble.

# L'espace insécable

## 2- Maintenir l'unité entre un nombre et son unité

**La distance est de 10~km.**

- Assure que "10 km" ne sera pas coupé sur deux lignes.

## 3- Éviter une séparation dans une abréviation

**M.~Dupont est absent.**

- Garantit que "M. Dupont" reste ensemble.

# Les ligature

- En **typographie**, une **ligature** est la fusion de deux ou plusieurs lettres pour améliorer l'esthétique et la lisibilité du texte. **LaTeX** prend en charge automatiquement certaines ligatures courantes en fonction de la police utilisée. (par exemple la ligature « œ » dans le mot œuf).
- Si une commande sans argument est suivie d'un espace, celui-ci n'est pas pris en compte. Ainsi, en écrivant `\oeuf`, on obtient « œuf » et non « œ uf »
- Pour prendre en compte l'espace, il y a deux solutions:
  - mettre une paire d'accolades vide après la commande: `\oe{}uf`;
  - entourer la commande d'accolades: `{\oe{}} uf`.
- La solution que nous croiserons le plus est celle de l'antislash

# Caractères spéciaux

- Les caractères codés en ISO88591 sont compris par le compilateur : les caractères accentués usuels peuvent être saisis directement. Il existe en revanche un certain nombre de caractères spéciaux résumés dans le tableau suivant.

ö	\\"{o}	ö	\.{o}	ő	\u{o}	č	\v{c}
ő	\H{o}	őo	\t{oo}	ç	\c{c}	ő	\d{o}
ø	\b{o}	ő	\={o}				
œ, œ	\oe, \OE	†	\dag	æ, Æ	\ae, \AE	‡	\ddag
å, Å	\aa, \AA	§	\S	ø, Ø	\o, \O	¶	\P
ł, Ł	\l, \L	©	\copyright	ß	\ss	£	\pounds
¿	?	ı	\i	¡	! ‘	J	\j
#	\#	\$	\\$	%	\%	&	\&
-	\_	{	\{	}	\}	\	\textbackslash

# Les commentaires

- En **LaTeX**, un **commentaire** est une ligne de texte qui est ignorée par le compilateur. Il permet d'ajouter des notes explicatives dans le code sans affecter le rendu final du document. Un commentaire commence par le signe %.
- **Comment commenter plusieurs lignes en LaTeX ?**
- **LaTeX ne permet pas** directement les commentaires multilignes comme /\* ... \*/ en C. Voici quelques solutions :
  - Ajouter % à chaque ligne
  - Utiliser \iffalse ... \fi pour commenter un bloc entier
  - Utiliser l'environnement **comment** (avec le package **comment**)
  - Si vous devez désactiver temporairement un grand bloc, utilisez le package **comment** : \usepackage{comment}

# Les commandes de hiérarchisation

- Nous aimerais sûrement pouvoir découper notre document en chapitres, en sections, en sous sections. Le sectionnement d'un texte est très important et permet d'organiser des idées et de ne pas perdre le lecteur. Il nous faut alors savoir que c'est l'une des choses qui se fait le plus facilement en **LaTeX**.
- **LaTeX** dispose en effet de quelques commandes pour nous aider dans cette tâche. Ces commandes correspondent à plusieurs niveaux de titres (**chapitres, sections, sous-sections, paragraphes...**) bien qu'on ne les utilise pas toujours suivant la signification de leur nom

# Les commandes de hiérarchisation

- Le nombre de commandes est plus ou moins long en fonction de la classe que l'on a choisi. La classe **article** que nous étudions ne permet pas, par exemple, d'avoir de **chapitre**. Voici les commandes disponibles avec la classe **article** par ordre d'importance.

## 1. Sections principales (Obligatoires pour organiser un document)

```
\section{Titre de la Section}
```

## 2. Sous-sections (Pour détailler davantage une section)

```
\subsection{Titre de la Sous-section}
```

# Les commandes de hiérarchisation

## 3. Sous-sous-sections (Encore plus de précision)

```
\subsubsection{Titre de la Sous-sous-section}
```

## 4. Paragraphes et sous-paragraphes (Texte structuré finement)

```
\paragraph{Titre du Paragraphe}
```

```
\ subparagraph{Titre du Sous-paragraphe}
```

## 5. Versions non numérotées (Pour titres sans numérotation)

```
\section*{Titre sans Numérotation}
```

# Les commandes de hiérarchisation

## 6. Table des matières (Facultatif, mais recommandé)

`\tableofcontents`

Génère automatiquement une table des matières basée sur les sections.

## 7. Annexes

Nous avons aussi la possibilité de créer une annexe. Pour cela, il nous faut utiliser la commande `\appendix`. Toutes les commandes `\section` qui seront utilisées après cette commande seront considérées comme faisant partie de l'annexe et seront donc numérotées différemment (des lettres A, B plutôt que des chiffres).

# Les commandes de hiérarchisation

## ❖ Titres courts

- Par défaut, le nom de **partie** qui apparaît dans la **table des matières** est le nom donné à la partie. Avec la commande **\section{Un nom de section vraiment très très long}**, le nom qui apparaîtra est très long. **Ce serait fabuleux de pouvoir mettre un titre un peu plus court dans la table des matières et de malgré tout garder le titre complet dans l'article.** Non seulement la table des matières sera plus lisible, et en plus la recherche sera facilitée. Pour faire cela, il suffit de passer à la commande de sectionnement le titre plus court en option:

```
\section[Nom court]{Génie Informatique, Réseaux-Télécoms, Management  
et Finance, Génie Industriel ,Logistique et Energétique Systèmes Embarqués}
```

# Les commandes de hiérarchisation

- lorsque l'on utilise la variante étoilée d'une commande de sectionnement, le titre n'apparaît pas dans la table des matières. Très embêtant pour nos introductions et autres avant-propos. Pour le faire apparaître, il faut utiliser la commande **\addcontentsline** après la commande étoilée. Cette commande prend trois arguments:
  - le premier sera toujours **toc** pour «**Table Of Contents**»;
  - le deuxième est le niveau de titre;
  - le troisième est le titre que l'on veut faire apparaître. Pour faire apparaître l'introduction, on utilisera donc ce code:

```
\section*{Introduction}
\addcontentsline{toc}{section}{Introduction}
```

# Des documents français

## ❑ babel, un package international

- Si le titre de la table des matières est «Contents», là où nous aimerais avoir «Table des matières». Nous aimerais quand même voir des documents français. Pour résoudre ce problème, nous allons utiliser le package **babel**. Il s'agit de l'un des principaux packages de LaTeX.
- Il a pour rôle de gérer les langues dans les documents. Il se charge donc de traduire certains mots, de modifier les règles de typographie, et il propose même des commandes pour faciliter l'écriture dans certaines langues:

```
\usepackage[french]{babel}
```

Avec l'option **french**, **babel** ajoute automatiquement les espaces insécables là où il le faut. Nous n'avons donc pas besoin de le faire manuellement.

# Des documents français

- Une autre méthode est de donner le langage à utiliser en option au **documentclass**. Les autres packages pourront alors détecter la langue utilisée et s'y adapter. C'est cette dernière méthode que nous allons adopter.
- Le package **babel** offre de plus de nombreuses fonctions pour accorder notre document à la langue utilisée. Par exemple, la typographie française veut que les noms de famille soient écrits en petite capitale. **babel** nous donne la commande **\textsc** qui prend en paramètre un texte et se charge de le mettre en petites capitales. De plus elle met ce texte dans une boite insécable ce qui veut dire que le nom ne pourra pas être coupé.

# Gestion du titre

## □ La commande `\maketitle`

LaTeX nous propose la commande `\maketitle` pour faire des titres. Elle se charge de mettre en place le titre, d'écrire le nom de l'auteur, tout ça sans que nous n'ayons à nous en mêler. Cette commande est à utiliser juste après avoir écrit `\begin{document}` afin d'avoir le titre en première page.

# Gestion du titre

- En fait, les éléments du titre sont définis grâce à d'autres commandes à utiliser avant la commande **maketitle** (elles peuvent être utilisées dans le préambule).
- La commande **\title** prend en paramètre le titre du document. Elle est obligatoire pour utiliser la commande **\maketitle**.
- La commande **\author** prend en paramètre les auteurs du document (s'il y en a plusieurs, il faut les séparer par la commande **\and**). Pour ne pas avoir d'auteurs, nous pouvons ne pas utiliser la commande **author**. Cependant, il vaut mieux transmettre à argument vide (utiliser **\author{ }** ) pour ne pas avoir d'avertissemens.
- La commande **\date** prend en paramètre la date du document. Nous pouvons lui passer ce que nous voulons. Pour avoir la date du jour, nous pouvons soit ne pas utiliser **\date**, soit lui passer **\today** comme argument. Pour ne pas avoir de date, il suffit de transmettre à **\date** un argument vide.

# Gestion du titre

- Nous pouvons utiliser la commande **\thanks** dans les commandes **\title** et **\author** pour obtenir une note de bas de page (elle prend en argument obligatoire la note à afficher). Elle est surtout utilisée dans les articles pour donner des informations complémentaires telles que l'adresse courriel.

□ Exemple:

```
title{Titre de l'article \thanks{Cette recherche a été financée par  
l'Agence de la Recherche. Contactez l'auteur principal à :  
auteur@example.com.} }
```

# Gestion du titre

- Nous avons maintenant vu comment mettre en page le titre. En essayant de compiler un code avec la commande **\maketitle**, nous pouvons remarquer que le texte commence tout de suite après le titre. Nous pouvons changer cela et dédier une page au titre. Cela se fait grâce aux options du **documentclass**:
  - l'option **titlepage** page permet d'avoir le titre seul sur une page;
  - l'option **notitlepage** permet d'avoir le titre en haut de la première page.

# De la sémantique

## □ L'emphase

- la commande `\emph` sert à mettre le texte en italique. Elle sert à indiquer que le texte passé en paramètre à la commande est important. L'italique n'est qu'une conséquence de cela. **LaTeX** met ce qui est important en italique (car ce sont les conventions typographiques), mais il aurait tout autant pu le mettre en gras ou en violet. `\emph` n'est pas une commande de police, c'est une commande **sémantique**.

➤ Exemple:

Ce texte est `\emph{\textbf{extrêmement important}}`.

# De la sémantique

## □ Les listes

Il y a trois types de listes.

- **Les listes à puces**
- **Les listes numérotées**
- **Les descriptions**

# De la sémantique

## ➤ Les listes à puces

- Pour les faire, il faut utiliser l'environnement commande **itemize**. À l'intérieur de celui-ci, il faut utiliser la **\item** pour une nouvelle entrée. Cette entrée peut tenir sur plusieurs paragraphes.

```
\begin{itemize}
    \item Système d'exploitation Unix
    \item Langage Python
    \item Transmission de données
\end{itemize}
```

# De la sémantique

## ➤ Les listes numérotées

- Les listes numérotées se construisent de la même manière que les listes à puces grâce à la commande **\item**. Seul l'environnement change: pour les listes numérotées, il faut utiliser l'environnement **enumerate**. La numérotation est faite automatiquement.

```
\begin{enumerate}
    \item Management des Entreprises
    \item Management des Relations Internationales
    \item Management des Ressources humaines
\end{enumerate}
```

# De la sémantique

## ➤ Les descriptions

Cet environnement permet de mettre en valeur une partie du texte (par défaut en gras).

```
\begin{description}
```

```
    \item[2020] Baccalauréat en sciences expérimentales à LS.
```

```
    \item[2023] Licence en mathématiques et informatique à FSR.
```

```
    \item[2025] Master mathématiques appliquées à FSR.
```

```
\end{description}
```

# Annotations et références

## □ Notes

- La commande `\thanks` permet d'obtenir une note de bas de page afin de donner des informations complémentaires à propos de l'auteur et du titre.
- Néanmoins, cette commande ne sert justement qu'à ça et l'utiliser en dehors des commandes `\author` et `\title` ne produira pas le résultat escompté. Nous allons maintenant voir comment faire pour obtenir des notes.

# Annotations et références

## □ Notes de bas de pages: La commande \footnote

- Pour écrire des notes de bas de page, nous pouvons utiliser la commande `note`. Il suffit de la placer après le mot (ou le groupe de mots) à annoter en lui passant en paramètre la note `\footnote` en question. Cette note se retrouvera en bas de la page du texte annoté. **LaTeX** s'occupe de numérotier chacune des notes.

### ➤ Exemple:

```
\begin{document}
```

Voici un exemple d'utilisation de la commande  
footnote `\footnote{Ceci est une note de bas de page.}` dans  
un document LaTeX.

```
\end{document}
```

# Annotations et références

## □ Notes de bas de pages: La commande `\footmark`

- La commande `\footmark` est à placer à côté du texte à annoter (comme `\footnote`), mais on ne lui passe pas le texte à annoter en argument. On lui passe en paramètre facultatif le **numéro** que l'on veut donner à la note.
- Pour indiquer à **LaTeX** le texte associé à ce numéro, il faut utiliser la commande `\footnotetext`. Elle prend en paramètre le texte à annoter et en paramètre facultatif le numéro de la note.
- De cette manière, `\footmark` permet de choisir l'emplacement de la note, et `\footnotetext` son contenu.

# Annotations et références

- Un des avantages de **\footnotemark** est de pouvoir faire référence plusieurs fois à la même note de la manière suivante.

□ Exemple:

```
\begin{document}
```

Dans ce document, la même note de bas de page est  
ici **\footnotemark[1]**, mais aussi ici **\footnotemark[1]**.

```
\footnotetext[1]{voici la note}
```

```
\end{document}
```

# Annotations et références

## □ Notes de marges

Les notes de marges consistent simplement à placer du texte dans la marge. Pour les utiliser, il faut utiliser la commande **\marginpar**. Elle prend en paramètre le texte à afficher dans la marge. Les notes de marges ne sont pas numérotées. Elles sont simplement placées dans la marge à côté de l'endroit où elles ont été appelées. Il est conseillé de ne faire que des notes courtes (et de réserver les notes de marges aux documents dont les marges sont grandes), ceci afin de pouvoir bien identifier où la note a été appelée dans le document et de ne pas avoir un document trop chargé

# Le système de références

- **LaTeX** dispose d'un système de références très puissant. Il permet de faire des références croisées très facilement et donc de renvoyer à un autre passage du document sans trop de soucis. Il permet ainsi de faire référence à une section, à une image et à d'autres parties du document. En fait, il permet même de rappeler la page où se trouve l'objet référencé. Ceci est utile notamment dans le cas de documents à imprimer. Dans le cas de documents destinés à rester numérique, on préférera placer des liens hypertextes plutôt que de donner la page.

# Le système de références

- En **LaTeX**, le système d'**étiquettes** (*labels*) est un mécanisme permettant de référencer facilement des éléments numérotés comme des sections, figures, tableaux, équations, etc. Cela est très utile pour éviter d'insérer manuellement les numéros et garantir leur mise à jour automatique si la structure du document change.

## 1- Déclaration d'une étiquette : `\label{mon_etiquette}`

- L'étiquette est définie avec la commande `\label{nom_du_label}` et doit être placée immédiatement après l'élément que vous souhaitez référencer.

```
\section{Introduction}  
\label{sec:intro}
```

# Le système de références

**2- Référencement d'une étiquette : `\ref{mon_etiquette}` et `\pageref{mon_etiquette}`**

Une fois l'étiquette définie, vous pouvez la référencer avec :

- `\ref{mon_etiquette}` : pour afficher le numéro de l'élément référencé.
- `\pageref{mon_etiquette}` : pour afficher le numéro de la page où se trouve l'élément.

## □ Exemple

Comme expliqué dans la section `\ref{sec:intro}`, nous allons détailler...

`\usepackage{hyperref}`

% Pour des références cliquables

# Le système de références

## □ Application aux figures, tableaux et équations

- Les labels peuvent être utilisés pour les **figures, tableaux et équations**.

```
\begin{figure}
    \centering
    \includegraphics[width=0.5\textwidth]{supmti_logo.png}
    \caption{Exemple d'image}
    \label{fig:image}
\end{figure}
```

La figure `\ref{fig:image}` illustre notre propos.

# Le système de références

## □ Tableaux :

```
\begin{table}
    \centering
    \begin{tabular}{c|c}
        A & B \\
        \hline
        1 & 2
    \end{tabular}
    \caption{Exemple de tableau}
    \label{tab:exemple}
\end{table}
```

Comme montré dans le tableau `\ref{tab:exemple}`, nous avons...

# Le système de références

□ Équations :

```
\begin{equation}
```

$$E = mc^2$$

```
\label{eq:einstein}
```

```
\end{equation}
```

L'équation `\ref{eq:einstein}` est bien connue.

# Le système de références

- Commande **\autoref** (package **hyperref**)
- Si vous utilisez le package **hyperref**, vous pouvez utiliser **\autoref{}** pour que LaTeX affiche automatiquement le type de l'élément référencé

```
\begin{document}
```

```
\section{Introduction}  
\label{sec:intro}
```

Dans **\autoref{sec:method}**, nous présentons la méthode utilisée.

```
\section{Méthodologie}  
\label{sec:method}
```

Cette section décrit la méthodologie de l'étude.

```
\end{document}
```



**\autoref{sec:intro}** affichera automatiquement "Section 2" au lieu de simplement « 2 ».

# Le système de références

- Pour générer automatiquement **la table des matières, la table des figures et la table des tableaux** du document, on utilise ces trois commandes:

```
\tableofcontents    % Génère la table des matières  
\listoffigures    % Génère la table des figures  
\listoftables     % Génère la table des tableaux
```

# Système d'index en LaTeX

- Un index est une liste alphabétique de mots-clés présents dans un document, souvent accompagnée de numéros de page où ces mots apparaissent. En LaTeX, on peut générer un index en utilisant le package **makeidx**.

# Système d'index en LaTeX

- Un **système d'index** en LaTeX permet de créer un index alphabétique à la fin d'un document, référençant des mots-clés et les pages où ils apparaissent. Pour ce faire, LaTeX utilise le package **makeidx**

**Étapes pour créer un index en LaTeX :**

## 1. Charger le package **makeidx**

Il faut ajouter dans le préambule du document :

```
\usepackage{makeidx}  
\makeindex % Active le mécanisme d'indexation
```

# Système d'index en LaTeX

## 2. Marquer les mots à indexer

Pour ajouter un mot à l'index, utilisez la commande `\index{mot}` dans le texte :

- Voici un exemple de  
`\index{LaTeX}LaTeX.`
- Cela ajoutera "LaTeX" à l'index, associé à la page où il apparaît.

## 3. Générer et insérer l'index

À l'endroit où vous voulez insérer l'index (généralement à la fin), ajoutez :

`\printindex`

# Création de commandes

- En **LaTeX**, il est possible de créer ses propres commandes pour simplifier l'écriture et améliorer la lisibilité du code. Voici les principales méthodes de création de commandes :
  1. **Création de nouvelles commandes avec \newcommand**
- La commande **\newcommand {nom\_commande} [nombre\_arguments] {commande}** permet de définir une nouvelle commande.

```
\documentclass{article}
\newcommand{\HelloWorld}{Bonjour le monde !}

\begin{document}
    \HelloWorld
\end{document}
```

# Création de commandes

## 2. Création de commandes avec arguments

- On peut aussi créer des commandes avec un ou plusieurs arguments.

```
\documentclass{article}
\newcommand{\addition}[2]{La somme de #1 et #2 est égale à \textbf{#1 + #2}.}
```

```
\begin{document}
    \addition{3}{5}
\end{document}
```

# Création de commandes

## 3. Modification de commandes existantes avec \renewcommand

- Si une commande existe déjà et qu'on souhaite la modifier, on utilise **\renewcommand**.

### □% Modification de la commande \section

- **\renewcommand{\section}[1]{\vspace{10pt} \noindent \textcolor{blue}{\Large \textbf{#1}} \vspace{5pt} \par}**
- **\renewcommand{\section}[1]{...}** : On redéfinit la commande \section pour modifier son apparence.
- **\vspace{10pt}** : Ajoute un espace avant la section.
- **\noindent** : Supprime l'indentation automatique.
- **\textcolor{blue}{...}** : Change la couleur du texte en bleu.
- **\Large** : Augmente la taille du texte.
- **\textbf{#1}** : Met le titre de section en gras.
- **\par** : Termine le paragraphe proprement.

# Insertion de fichiers

- Lorsque l'on rédige un long document en LaTeX, il est recommandé de diviser son contenu en plusieurs fichiers afin de faciliter la gestion, la navigation et la compilation. Voici une explication détaillée des différentes commandes utilisées pour inclure des fichiers externes.

## 1. \input{file}

Cette commande insère le contenu du fichier **file.tex** directement à l'endroit où elle est appelée. Elle est équivalente à copier-coller le texte du fichier dans le document principal

# Insertion de fichiers

## 2. \include{file}

Cette commande insère aussi un fichier externe, mais elle est plus puissante que \input car :

- Elle force un saut de page avant et après l'insertion du fichier.
- Elle ne peut être utilisée que dans le corps du document (pas dans l'entête).
- Elle permet d'utiliser \includeonly{} pour n'inclure que certains fichiers lors de la compilation.

# Insertion de fichiers

## 3. \subimport{dir/}{file} et \subincludefrom{dir/}{file} (via le package import)

- Lorsque les fichiers sont organisés dans des sous-dossiers, \input et \include ne gèrent pas bien les chemins relatifs. Le package import permet d'inclure correctement des fichiers situés dans des sous-répertoires.

# Insertion de fichiers

Commande	Utilisation	Particularités
\input{file}	Inclut file.tex	Peut être utilisé partout, pas de saut de page
\subimport{dir/}{file}	Inclut file.tex d'un sous-dossier	Nécessite le package import
\include{file}	Inclut file.tex avec saut de page	Ne fonctionne pas dans l'entête, compatible avec \includeonly{}}
\subincludefrom{dir/}{file}	Comme \include, mais pour un sous-dossier	Nécessite le package import

# Insertion de fichiers

## □ Insertion de fichier PDF

- Pour insérer des fichiers PDF (par exemple des attestations, certificats, annexes...) dans un document, notamment dans la **section des annexes**

:

`\includepdf[options]{chemin/vers/le/fichier.pdf}`

Assurez-vous d'avoir inclus ce package dans le préambule :



`\usepackage{pdffpages}`

# Insertion de fichiers

Option	Description
pages=-	Inclut <b>toutes les pages</b> du PDF
pages={1}	Inclut <b>seulement la page 1</b>
pages={1,3}	Inclut les pages <b>1 et 3</b>
pages={1-2}	Inclut les <b>pages 1 à 2</b>
pagecommand={}	Ne met <b>aucun en-tête ou pied de page</b> sur les pages PDF insérées
fitpaper=true	Ajuste le PDF pour <b>remplir la page</b>
scale=0.9	<b>Redimensionne</b> le PDF à 90% de la taille
addtotoc={page,section,...}	<b>Ajoute le PDF au sommaire</b> (table des matières)

# Insertion de fichiers

- Insérer un fichier PDF entier :  
`\includepdf[pages=-, pagecommand={}, fitpaper=true] {certificat.pdf}`
- Insérer uniquement la première page du PDF :  
`\includepdf[pages=1, pagecommand={}, fitpaper=true]{attestation_stage.pdf}`

# Insertion de fichiers

- Insérer un fichier en conservant la numérotation des pages LaTeX
- \includepdf[pages=-]{Annexes/rapport\_final.pdf}
- Insérer avec un titre dans la table des matières :

```
\includepdf[  
    pages=-,  
    pagecommand={},  
    addtotoc={1,section,1,Attestation de présence}  
]{attestation_presence.pdf}
```

# Mise en page

## □ Taille de caractère

- La taille des caractères est en premier lieu définie par les options de style spécifiées au début d'un document. En plus de cela, il existe 10 tailles que l'on peut spécifier par les commandes du tableau :

\tiny	Ce texte est tiny
\scriptsize	Ce texte est scriptsize
\footnotesize	Ce texte est footnotesize
\small	Ce texte est small
\normalsize	Ce texte est normalsize
\large	Ce texte est large
\Large	Ce texte est Large
\LARGE	Ce texte est LARGE
\huge	Ce texte est huge
\HUGE	Ce texte est Huge

Huge



# Mise en page

## □ Polices de caractères

- La police par défaut de LaTeX est la police roman. D'autres polices sont également disponibles pour mettre en évidence certaines parties de texte. Tout est résumé dans le tableau. On notera qu'il existe deux types de commande :
  1. Les commandes à un argument, qui permettent de spécifier la mise en forme de brefs extraits (par `\emph{test}`);
  2. Les commandes sans argument, qui fonctionnent comme les précédentes mais qui peuvent être utilisées pour de plus longs extraits

# Mise en page

## □ Polices de caractères

\textnormal{...}	{\normalfont ...}	Normal
\textbf{...}	{\bfseries ...}	<b>Gras</b>
\textit{...}	{\itshape ...}	<i>Italic</i>
\textrm{...}	{\rmfamily ...}	Police Roman
\textsf{...}	{\sffamily ...}	Police Sans serif
\texttt{...}	{\ttfamily ...}	Police Typewriter
\emph{...}	{\em ...}	<i>Emphasize</i>
\textup{...}	{\upshape ...}	Upright
\textsl{...}	{\slshape ...}	<i>Slanted</i>
\textsc{...}	{\scshape ...}	PETITES CAPITALE
\textmd{...}	{\mdseries ...}	Medium series

# Mise en page

## □ Alignements

- En LaTeX, les alignements peuvent être réalisés de deux manières :
  - 1. En ligne** : Utilisation directe dans une ligne de texte.
  - 2. Dans un environnement** : Utilisation d'environnements spécifiques pour gérer l'alignement sur plusieurs lignes.

{\centering ...}	\begin{center}... \end{center}	centré
{\raggedright ...}	\begin{flushleft}... \end{flushleft}	aligné à gauche
{\raggedleft ...}	\begin{flushright}... \end{flushright}	aligné à droite

# Mise en page

## □ Texte souligné

- On souligne un texte au moyen de la commande `\underline{...}`. Ainsi `\underline{Texte souligné}` produit Texte souligné.

## □ Espacements et sauts

□	espace	,	petit espace
~	espace insécable	\\"	saut de ligne
<code>\newline</code>	saut de ligne	\\"[ <i>h</i> ]	à la ligne avec <i>h</i> d'interligne
<code>\hspace{<i>l</i>}</code>	espace horizontal de <i>l</i> , ignoré sur un saut de ligne	<code>\vspace{<i>h</i>}</code>	espace vertical de <i>h</i> , ignoré sur un saut de page
<code>\hspace*{<i>l</i>}</code>	espace horizontal de <i>l</i>	<code>\vspace*{<i>h</i>}</code>	espace vertical de <i>h</i>
<code>\smallskip</code>	petit espace vertical	<code>\medskip</code>	espace vertical moyen
<code>\bigskip</code>	grand espace vertical		
<code>\hfill</code>	espace élastique horizontal	<code>\vfill</code>	espace élastique vertical
<code>\rulefill</code>	ligne élastique horizontale	<code>\dotfill</code>	ligne de points élastiques horizontaux

# Mise en page

## □ Espaces horizontaux et insécables

### 1.   (Espace standard)

Exemple :

- Ceci est un texte avec un espace standard.

### 2. ~ (Espace insécable)

Empêche un retour à la ligne à cet endroit.

- M.~Dupont est un enseignant.

"M." et "Dupont" resteront sur la même ligne.

### 3. \newline (Saut de ligne)

Force un retour à la ligne.

- Ceci est une phrase.\newline
- Ceci est une nouvelle ligne.

# Mise en page

## □ Espaces horizontaux

### 4. **\hspace{l}** (Espace horizontal de l)

Ajoute un espace de l (ex. : 1cm) sur la ligne.

- Mot1 \hspace{1cm} Mot2

Un espace de 1 cm est ajouté entre "Mot1" et "Mot2".

### 5. **\hspace\*{l}** (Espace horizontal de l, même en début de ligne)

Comme \hspace, mais fonctionne aussi en début de ligne.

- \hspace\*{2cm} Texte aligné avec un décalage

# Mise en page

## □ Sauts de ligne

### 6. \, (Petit espace)

Ajoute un petit espace entre deux mots.

- Mot1\,Mot2

Un petit espace est ajouté entre "Mot1" et "Mot2".

### 7. \\ (Saut de ligne)

Passe à la ligne suivante.

- Ligne 1 \\
- Ligne 2

### 8. \\[h] (Saut de ligne avec espace h)

Ajoute un saut de ligne avec une distance spécifique.

- Ligne 1 \\[1cm]

Ligne 2

"Ligne 2" est affichée 1 cm sous "Ligne 1".

# Mise en page

## □ Espaces verticaux

### 9. `\vspace{h}` (Espace vertical de h, ignoré en début de page)

Ajoute un espace vertical.

- **Texte avant**`\vspace{1cm}`

- **Texte après**

### 10. `\vspace*{h}` (Espace vertical forcé)

Ajoute un espace vertical même en début de page.

- `\vspace*{2cm}`

- **Texte affiché avec un décalage vertical.**

### 11. `\smallskip` (Petit espace vertical)

Ajoute un petit espace vertical.

**Texte avant**`\smallskip`

- **Texte après**

# Mise en page

- Espaces verticaux

- 12. \medskip (Espace vertical moyen)

- Ajoute un espace vertical moyen.

- Texte avant \medskip

- Texte après

- 13. \bigskip (Grand espace vertical)

- Ajoute un grand espace vertical.

- Texte avant\bigskip

- Texte après

# Mise en page

## □ En-tête et pied de page

LaTeX offre plusieurs styles prédéfinis pour les en-têtes et pieds de page

- **empty** : aucun en-tête ni pied de page.
- **plain** : aucun en-tête, avec le numéro de page centré en pied de page (style par défaut).
- **headings** : en-têtes contenant le titre du chapitre, de la section ou de la sous-section et le numéro de la page.
- **myheadings** : permet à l'utilisateur de définir lui-même le contenu des en-têtes via les commandes **\markboth** et **\markright**.

# Mise en page

- Pour une personnalisation avancée des en-têtes et pieds de page, le package fancyhdr est couramment utilisé.
  - **Chargement du package :** \usepackage{fancyhdr}
  - **Activation du style personnalisé :** \pagestyle{fancy}
  - **Configuration des en-têtes et pieds de page :**
    - ❖ Pour définir le contenu des en-têtes : \fancyhead[zone]{contenu}
    - ❖ Pour définir le contenu des pieds de page : \fancyfoot[zone]{contenu}
- Les zones sont spécifiées par :
- **L** : gauche (Left), **C** : centre, **R** : droite (Right)

# Tableaux et figures en LATEX

- LaTeX offre une flexibilité remarquable pour la création de tableaux, permettant de structurer et de présenter des données de manière claire et professionnelle. Voici un aperçu des principales méthodes et packages pour concevoir des tableaux en LaTeX.

- 1. Environnement tabular**
- 2. Environnement table**
- 3. Packages Utiles pour les Tableaux**

# Tableaux et figures en LATEX

## 1. Environnement tabular

L'environnement `tabular` est la méthode de base pour créer des tableaux en LaTeX. Il est utilisé pour insérer des tableaux dans le corps du texte. Voici un exemple simple :

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

Dans cet exemple :

- `|c|c|c|` spécifie trois colonnes centrées avec des barres verticales entre elles.
- `\hline` ajoute une ligne horizontale.
- Chaque ligne de données est séparée par `\backslash`, et les colonnes sont séparées par `&`.

# Tableaux et figures en LATEX

- **Environnement table**

Pour inclure un titre (**caption**) et gérer la position du tableau dans le document, l'environnement **table** est utilisé en combinaison avec **tabular** :

```
\begin{table}[h]
    \centering
    \caption{Titre du tableau}
    \begin{tabular}{|c|c|c|}
        \hline
        Colonne 1 & Colonne 2 & Colonne 3 \\
        \hline
        Donnée 1 & Donnée 2 & Donnée 3 \\
        Donnée 4 & Donnée 5 & Donnée 6 \\
        \hline
    \end{tabular}
\end{table}
```

[h] indique que le tableau doit être placé "ici" dans le texte.  
\centering centre le tableau sur la page.  
\caption{Titre du tableau} ajoute un titre au tableau.

# Tableaux et figures en LATEX

- Plusieurs packages étendent les fonctionnalités de base pour la création de tableaux :
- **array** : Permet des améliorations dans la définition des colonnes, comme l'ajustement de la largeur et l'alignement du texte.
- **booktabs** : Fournit des commandes pour créer des tableaux avec une présentation professionnelle, en utilisant des lignes horizontales de différentes épaisseurs sans barres verticales.
- Exemple d'utilisation de booktabs :

# Tableaux et figures en LATEX

```
\usepackage{booktabs}  
...  
\begin{tabular}{ccc}  
  \toprule  
  Colonne 1 & Colonne 2 & Colonne 3 \\  
  \midrule  
  Donnée 1 & Donnée 2 & Donnée 3 \\  
  Donnée 4 & Donnée 5 & Donnée 6 \\  
  \bottomrule  
\end{tabular}
```

# Tableaux et figures en LATEX

- **tabularx** : Utilisé pour créer des tableaux dont la largeur s'ajuste automatiquement à la largeur de la page, en définissant des colonnes de largeur proportionnelle. Exemple d'utilisation de tabularx :

```
\usepackage{tabularx}  
...  
\begin{tabularx}{\textwidth}{|X|X|X|}  
 \hline  
 Colonne 1 & Colonne 2 & Colonne 3 \\  
 \hline  
 Donnée 1 & Donnée 2 & Donnée 3 \\  
 Donnée 4 & Donnée 5 & Donnée 6 \\  
 \hline  
\end{tabularx}
```

# Figures en LATEX

## □ Insertion de figures en LATEX

- L'insertion d'images en LATEX se fait généralement avec le package **graphicx**.

### 1. Chargement du package

- Avant d'insérer des images, il faut charger le package **graphicx** dans le préambule :

# Figures en LATEX

```
\documentclass[a4paper,12pt]{article}
\usepackage{graphicx}
\usepackage{float} % Pour [H]
\begin{document}
\begin{figure}[H]
    \centering
    \includegraphics[width=0.5\textwidth]{supmti_logo.png}
    \caption{Une image d'exemple}
    \label{fig:exemple}
\end{figure}
\end{document}
```

## Positionnement des figures

- [h] = here (ici)
- [t] = top (en haut de la page)
- [b] = bottom (en bas)
- [p] = page spéciale de figures
- [H] = exact (nécessite float)

# Figures en LATEX

- Plusieurs images dans la même figure
- Utilisation de `\usepackage{subfigure}` ou `\usepackage{subcaption}`

```
\begin{figure}[H]
    \centering
    \begin{subfigure}[b]{0.4\textwidth}
        \includegraphics[width=\textwidth]{r1.jpeg}
        \caption{Image 1}
    \end{subfigure}
    \hfill
    \begin{subfigure}[b]{0.4\textwidth}
        \includegraphics[width=\textwidth]{R3.jpeg}
        \caption{Image 2}
    \end{subfigure}
    \caption{Deux images côte à côte}
\end{figure}
```

# Les maths en LATEX

## □ Le mode mathématiques

- Pour écrire des formules et/ou des symboles mathématiques, il faut d'abord passer en mode mathématiques. On distingue principalement deux méthodes
  1.  $\$...$$  ou  $\backslash(...\backslash)$ : la formule fait partie d'un texte.
  2.  $\backslash[...\backslash]$ : la formule est mise en évidence : il y a un saut de ligne et la formule est centrée.

# Les maths en LATEX

- **Police de caractères spécifiques ; \usepackage{amsmath, amssymb}**

\mathbb{...}	Alphabet <i>blackboard</i> : ABC...N...Z
\mathcal{...}	Alphabet <i>calligraphié</i> : ABC...
\mathfrak{...}	Alphabet <i>Euler Fraktur</i> : ABC...abc...123...
\mathbf{...}	<b>Gras</b>
\mathit{...}	<i>Italic</i>
\mathrm{...}	Police roman
\mathsf{...}	Police Sans serif
\mathtt{...}	Police Typewriter

# Les maths en LATEX

## □ Symboles spéciaux

- Caractères et constructions spéciaux

$\hat{a}$ \hat{a}	$\acute{a}$ \acute{a}	$\bar{a}$ \bar{a}	$\dot{a}$ \dot{a}	$\check{a}$ \check{a}
$\check{a}$ \check{a}	$\grave{a}$ \grave{a}	$\vec{a}$ \vec{a}	$\ddot{a}$ \ddot{a}	$\tilde{a}$ \tilde{a}

$\overleftarrow{abc}$ \overleftarrow{abc}	$\overrightarrow{abc}$ \overrightarrow{abc}
$\overline{abc}$ \overline{abc}	$\underline{abc}$ \underline{abc}
$\widetilde{abc}$ \widetilde{abc}	$\widehat{abc}$ \widehat{abc}
$\xrightarrow{abc}$ \xrightarrow{\text{abc}}	$\xleftarrow{abc}$ \xleftarrow{\text{abc}}
$\overbrace{abc}$ \overbrace{abc}	$\underbrace{abc}$ \underbrace{abc}

# Les maths en LATEX

## □ Les lettres grecques

- Elles sont obtenues à partir du nom de la lettre précédé par un antislash \. Certaines possèdent une variante préfixée par \var.

$\alpha$	\alpha	$\theta$	\theta	$\pi$	\pi	$\phi$	\phi
$\beta$	\beta	$\vartheta$	\vartheta	$\varpi$	\varpi	$\varphi$	\varphi
$\gamma$	\gamma	$\iota$	\iota	$\rho$	\rho	$\chi$	\chi
$\delta$	\delta	$\kappa$	\kappa	$\varrho$	\varrho	$\psi$	\psi
$\epsilon$	\epsilon	$\lambda$	\lambda	$\sigma$	\sigma	$\omega$	\omega
$\varepsilon$	\varepsilon	$\mu$	\mu	$\varsigma$	\varsigma		
$\zeta$	\zeta	$\nu$	\nu	$\tau$	\tau		
$\eta$	\eta	$\xi$	\xi	$\upsilon$	\upsilon		

# Les maths en LATEX

## □ Les lettres grecques

- Certaines lettres existent sous forme majuscule

$\Gamma$	<code>\Gamma</code>	$\Xi$	<code>\Xi</code>	$\Phi$	<code>\Phi</code>
$\Delta$	<code>\Delta</code>	$\Pi$	<code>\Pi</code>	$\Psi$	<code>\Psi</code>
$\Theta$	<code>\Theta</code>	$\Sigma$	<code>\Sigma</code>	$\Omega$	<code>\Omega</code>
$\Lambda$	<code>\Lambda</code>	$\Upsilon$	<code>\Upsilon</code>		

# Les maths en LATEX

## □ Symboles mathématiques

- Symboles divers.

$\aleph$	<code>\aleph</code>	$\prime$	<code>\prime</code>	$\forall$	<code>\forall</code>
$\hbar$	<code>\hbar</code>	$\emptyset$	<code>\emptyset</code>	$\exists$	<code>\exists</code>
$i$	<code>\imath</code>	$\nabla$	<code>\nabla</code>	$\neg$	<code>\neg</code> ou <code>\lnot</code>
$j$	<code>\jmath</code>	$\surd$	<code>\surd</code>	$\flat$	<code>\flat</code>
$\ell$	<code>\ell</code>	$\top$	<code>\top</code>	$\natural$	<code>\natural</code>
$\wp$	<code>\wp</code>	$\bot$	<code>\bot</code>	$\sharp$	<code>\sharp</code>
$\Re$	<code>\Re</code>	$\mid$	<code>\mid</code>		<code>\clubsuit</code>
$\Im$	<code>\Im</code>	$\angle$	<code>\angle</code>		<code>\diamondsuit</code>
$\partial$	<code>\partial</code>	$\triangle$	<code>\triangle</code>		<code>\heartsuit</code>
$\infty$	<code>\infty</code>	$\backslash$	<code>\backslash</code>		<code>\spadesuit</code>

# Les maths en LATEX

## □ Symboles de tailles variables et délimiteurs

$\sum$	<code>\sum</code>	$\oint$	<code>\oint</code>	$\bigvee$	<code>\bigvee</code>	$\bigodot$	<code>\bigodot</code>
$\prod$	<code>\prod</code>	$\bigcap$	<code>\bigcap</code>	$\bigwedge$	<code>\bigwedge</code>	$\bigotimes$	<code>\bigotimes</code>
$\coprod$	<code>\coprod</code>	$\bigcup$	<code>\bigcup</code>	$\bigoplus$	<code>\bigoplus</code>		
$\int$	<code>\int</code>	$\bigsqcup$	<code>\bigsqcup</code>	$\biguplus$	<code>\biguplus</code>		
$\{$	<code>\{</code>	$($	<code>(</code>	$\ $	<code>\ </code>	$\{$	<code>\lbrace</code>
$\}$	<code>\}</code>	$)$	<code>)</code>	$ $	<code> </code>	$\}$	<code>\rbrace</code>
$[$	<code>\lbrack</code>	$\lfloor$	<code>\lfloor</code>	$\lceil$	<code>\lceil</code>	$\langle$	<code>\langle</code>
$]$	<code>\rbrack</code>	$\rfloor$	<code>\rfloor</code>	$\rceil$	<code>\rceil</code>	$\rangle$	<code>\rangle</code>

- Les délimiteurs encadrent une expression entre `\left` délimiteur et `\right` délimiteur ( le délimiteur «.» sert alors de délimiteur vide).

# Les maths en LATEX

## □ Opérations binaires

$\pm$	<code>\pm</code>	$\cap$	<code>\cap</code>	$\vee$	<code>\vee</code> ou <code>\lor</code>
$\mp$	<code>\mp</code>	$\cup$	<code>\cup</code>	$\wedge$	<code>\wedge</code> ou <code>\land</code>
$\setminus$	<code>\setminus</code>	$\oplus$	<code>\uplus</code>	$\oplus$	<code>\oplus</code>
$\cdot$	<code>\cdot</code>	$\sqcap$	<code>\sqcap</code>	$\ominus$	<code>\ominus</code>
$\times$	<code>\times</code>	$\sqcup$	<code>\sqcup</code>	$\otimes$	<code>\otimes</code>
$*$	<code>*</code>	$\triangleleft$	<code>\triangleleft</code>	$\oslash$	<code>\oslash</code>
$\star$	<code>\star</code>	$\triangleright$	<code>\triangleright</code>	$\odot$	<code>\odot</code>
$\diamond$	<code>\diamond</code>	$\wr$	<code>\wr</code>	$\dagger$	<code>\dagger</code>
$\circ$	<code>\circ</code>	$\bigcirc$	<code>\bigcirc</code>	$\ddagger$	<code>\ddagger</code>
$\bullet$	<code>\bullet</code>	$\bigtriangleup$	<code>\bigtriangleup</code>	$\amalg$	<code>\amalg</code>
$\div$	<code>\div</code>	$\bigtriangledown$	<code>\bigtriangledown</code>		

# Les maths en LATEX

## □ Relations

$\leq$	<code>\leq ou \le</code>	$\geq$	<code>\geq ou \le</code>	$\equiv$	<code>\equiv</code>
$\prec$	<code>\prec</code>	$\succ$	<code>\succ</code>	$\sim$	<code>\sim</code>
$\preceq$	<code>\preceq</code>	$\succeq$	<code>\succeq</code>	$\simeq$	<code>\simeq</code>
$\ll$	<code>\ll</code>	$\gg$	<code>\gg</code>	$\asymp$	<code>\asymp</code>
$\subset$	<code>\subset</code>	$\supset$	<code>\supset</code>	$\approx$	<code>\approx</code>
$\subseteq$	<code>\subseteq</code>	$\supseteq$	<code>\supseteq</code>	$\cong$	<code>\cong</code>
$\sqsubset$	<code>\sqsubset</code>	$\sqsupset$	<code>\sqsupset</code>	$\bowtie$	<code>\bowtie</code>
$\in$	<code>\in</code>	$\ni$	<code>\ni ou \owns</code>	$\propto$	<code>\propto</code>
$\vdash$	<code>\vdash</code>	$\dashv$	<code>\dashv</code>	$\models$	<code>\models</code>
$\smile$	<code>\smile</code>	$\mid$	<code>\mid</code>	$\doteq$	<code>\doteq</code>
$\frown$	<code>\frown</code>	$\parallel$	<code>\parallel</code>	$\perp$	<code>\perp</code>

# Les maths en LATEX

## □ Relations contraires.

- Elles sont généralement obtenues en préfixant la relation par \not.

$\not<$	$\not>$	$\not=$ ou $\neq$
$\not\leq$	$\not\geq$	$\not\equiv$ ou $\not\equiv$
$\not\prec$	$\not\succ$	$\not\sim$
$\not\preceq$	$\not\succeq$	$\not\simeq$
$\not\subset$	$\not\supset$	$\not\approx$
$\not\subseteq$	$\not\supseteq$	$\not\cong$
$\not\sqsubset$	$\not\sqsupset$	$\not\asymp$
$\not\sqsubseteq$	$\not\sqsupseteq$	

# Les maths en LATEX

## □ Flèches et pointillés

a \leftarrow b \qquad a \longleftarrow b \qquad

$\leftarrow$	<code>\leftarrow</code> ou <code>\gets</code>	$\rightarrow$	<code>\rightarrow</code> ou <code>\to</code>
$\leftarrow\!\leftarrow$	<code>\longleftarrow</code>	$\rightarrow\!\rightarrow$	<code>\longrightarrow</code>
$\Leftarrow$	<code>\Leftarrow</code>	$\Rightarrow$	<code>\Rightarrow</code>
$\Longleftarrow$	<code>\Longleftarrow</code>	$\Longrightarrow$	<code>\Longrightarrow</code>
$\leftrightarrow$	<code>\leftrightarrow</code>	$\Leftrightarrow$	<code>\Leftrightarrow</code>
$\longleftrightarrow$	<code>\longleftrightarrow</code>	$\Longleftrightarrow$	<code>\Longleftrightarrow</code> ou <code>\iff</code>
$\hookleftarrow$	<code>\hookleftarrow</code>	$\hookrightarrow$	<code>\hookrightarrow</code>
$\leftharpoonup$	<code>\leftharpoonup</code>	$\rightharpoonup$	<code>\rightharpoonup</code>
$\leftharpoondown$	<code>\leftharpoondown</code>	$\rightharpoondown$	<code>\rightharpoondown</code>
$\uparrow$	<code>\uparrow</code>	$\downarrow$	<code>\downarrow</code>
$\Uparrow$	<code>\Uparrow</code>	$\Downarrow$	<code>\Downarrow</code>
$\updownarrow$	<code>\updownarrow</code>	$\Updownarrow$	<code>\Updownarrow</code>
$\nearrow$	<code>\nearrow</code>	$\nwarrow$	<code>\nwarrow</code>
$\searrow$	<code>\searrow</code>	$\swarrow$	<code>\swarrow</code>
$\mapsto$	<code>\mapsto</code>	$\longmapsto$	<code>\longmapsto</code>
$\leftrightharpoons$	<code>\leftrightharpoons</code>	$\rightleftharpoons$	<code>\rightleftharpoons</code>
$\curvearrowleft$	<code>\curvearrowleft</code>	$\curvearrowright$	<code>\curvearrowright</code>
$\ldots$	<code>\ldots</code>	$\cdots$	<code>\cdots</code>
$\vdots$	<code>\vdots</code>	$\ddots$	<code>\ddots</code>

# Les maths en LATEX

## □ Fonctions standards (sin, cos etc...)

- Certaines abréviations /fonctions standards existent déjà en LATEX.  
Ainsi, on peut obtenir:

$$\cos(\theta + \phi) = \cos \theta \cos \phi - \sin \theta \sin \phi$$

- En tapant

```
\[
\cos(\theta + \phi) = \cos \theta \cos \phi - \sin \theta \sin \phi
]
```

# Les maths en LATEX

- Voici la liste des séquences à utiliser

\arccos	\cos	\csc	\exp	\ker	\limsup	\min	\sinh
\arcsin	\cosh	\deg	\gcd	\lg	\ln	\Pr	\sup
\arctan	\cot	\det	\hom	\lim	\log	\sec	\tan
\arg	\coth	\dim	\inf	\liminf	\max	\sin	\tanh

# Les maths en LATEX

## □ Structures courantes

Indices et exposants

Un indice s'obtient à l'aide de `_` et un exposant avec `^`.

Exemple:

$$\begin{array}{c|c} x_i & x_{\_i} \\ x_{i-1}^{j-1} & x_{\_{i-1}}^{\_{j-1}} \end{array} \quad \begin{array}{c|c} x^i & x^{\_i} \\ x_{u_n} & x_{\_{u\_n}} \end{array}$$

# Les maths en LATEX

## □ Fractions

- La commande `\frac{num}{den}` produit une fraction de numérateur num et de dénominateur den.
- Exemple:

`\begin{itemize}`

`\item $\frac{1}{2}$ : une demi`

`\item $\frac{a+b}{c+d}$ : fraction avec des expressions`

`\item $\frac{x^2 + 1}{x - 1}$ : fraction algébrique`

`\item $\frac{\sqrt{2}}{2}$ : racine carrée sur 2`

`\item $\frac{\int_0^1 x^2 dx}{\int_0^1 x dx}$ : rapport d'intégrales`

`\end{itemize}`

# Introduction à Beamer

# Introduction à Beamer

## □ Structure de base d'un document Beamer

- Thèmes et personnalisation
- Création de diapositives (frames)
- Éléments courants
- Effets de transition & incrustations
- Intégration de figures, tableaux et codes
- Références bibliographiques

# Introduction à Beamer

## □Définition

- **Beamer** est une classe de document LaTeX utilisée pour créer des **présentations sous forme de diaporamas** au format PDF.
- Elle fonctionne comme un équivalent LaTeX de PowerPoint.
- Adaptée à un public académique et scientifique.
- Génère des présentations **professionnelles, propres et typographiquement correctes**.

# Introduction à Beamer

- Pourquoi utiliser Beamer ?

Avantages	Détails
Qualité typographique	Texte net, homogène, professionnel
Intégration facile	Avec les maths, BibTeX, TikZ, etc.
Reproductibilité	Fichier source en .tex, versionnable
Professionnalisme	Idéal pour soutenances, conférences
Thèmes et blocs	Nombreux styles disponibles

# Ce qu'on peut faire avec Beamer

- Créer des diapositives avec titres, listes, colonnes.
- Ajouter des figures, graphiques, tableaux, équations.
- Gérer des incrustations (éléments qui apparaissent progressivement).
- Ajouter des transitions douces avec \pause.
- Structurer les présentations avec sections et sous-sections.
- Intégrer des citations et des références bibliographiques.
- Ajouter un sommaire dynamique qui se met à jour automatiquement.

# Comparaison avec PowerPoint

Critère	Beamer	PowerPoint
Éditeur	Code LaTeX	Interface graphique
Apparence	Minimaliste, propre	Libre, souvent chargé
Collaboratif	Oui (ex. Overleaf, Git)	Moins adapté
Mathématiques	Excellent	Plus limitées
Courbe d'apprentissage	Moyenne à élevée	Faible

# Quand utiliser Beamer ?

- Présentations scientifiques (soutenance, conférence, exposé)
- Cours magistraux
- Défense de projet ou rapport technique

# Ce qu'il faut pour commencer

- Distribution LaTeX (TeXLive, MiKTeX)
- Éditeur : Overleaf, TeXstudio, VS Code avec extension LaTeX
- Structure minimale d'un fichier :

# Structure de base d'un document Beamer

- Structure générale

```
\documentclass{beamer} % 1. Déclaration de la classe beamer

\title{Titre de la présentation} % 2. Titre
\author{Votre nom} % 3. Auteur
\date{\today} % 4. Date (peut être personnalisée)

\begin{document} % 5. Début du document

    \frame{\titlepage} % 6. Diapositive de titre

    \begin{frame}{Titre de la diapo} % 7. Début d'une diapositive
        Contenu de la diapositive % 8. Contenu : texte, liste, image, etc.
    \end{frame} % 9. Fin de la diapositive

\end{document} % 10. Fin du document
```

# Exemple de base complet

```
\documentclass{beamer}

\title{Introduction à LaTeX}
\author{Dr. Nom Prénom}
\date{Avril 2025}

\begin{document}

\frame{\titlepage}

\begin{frame}{Plan de la séance}
\begin{itemize}
\item Introduction
\item Structure d'un fichier
\item Commandes de base
\end{itemize}
\end{frame}

\begin{frame}{Conclusion}
Merci pour votre attention !
\end{frame}

\end{document}
```

# Structure de base d'un document Beamer

- Extensions utiles (optionnel mais conseillé)
- **Ajouter un thème** : `\usetheme{Madrid}`
- **Changer les couleurs** : `\usecolortheme{dolphin}`
- **Mettre un logo** : `\logo{\includegraphics[height=0.8cm]{logoSupMTI.png}}`

# Qu'est-ce qu'un thème dans Beamer ?

- Un **thème Beamer** définit l'apparence visuelle générale de votre présentation :
- Police, couleur, arrière-plan, style des titres, encadrés...
- Similaire aux « modèles de diapositives » dans PowerPoint

# Types de thèmes dans Beamer

- Beamer permet de personnaliser ta présentation avec plusieurs niveaux de thèmes :

Type de thème	Commande	Rôle
Thème principal	<code>\usetheme{}</code>	Structure globale, disposition des éléments
Thème de couleur	<code>\usecolortheme{}</code>	Jeu de couleurs utilisé
Thème de police	<code>\usefonttheme{}</code>	Apparence de la police (taille, graisse...)
Thème d'arrière-plan	<code>\useinnertheme{}/\useoutertheme{}</code>	Contrôle les blocs, titres, barres latérales

# Exemples de thèmes disponibles

- `\usetheme{Madrid}` % Clair, épuré, professionnel
- `\usetheme{CambridgeUS}` % Formelle, avec barre latérale
- `\usetheme{Berkeley}` % Avec une structure verticale à gauche
- `\usetheme{AnnArbor}` % Colorée avec une barre horizontale
- `\usetheme{Darmstadt}` % Design moderne avec titres soulignés
- `\usetheme{Frankfurt}` % Similaire à Madrid avec quelques variations

# Thèmes de couleur :

- \usecolortheme{dolphin}
- \usecolortheme{seahorse}
- \usecolortheme{whale}
- \usecolortheme{orchid}

# Thèmes de police :

- \usefonttheme{default}
- \usefonttheme{serif}
- \usefonttheme{structurebold}
- \usefonttheme{professionalfonts}

# Création de diapositives (frames)

- Types de contenu dans une frame

Type de contenu	Exemple
Texte	<b>Texte simple ou paragraphes</b>
Listes	<b>itemize ou enumerate</b>
Tableaux	<b>Environnement tabular</b>
Figures / images	<b>Avec \includegraphics</b>
Mathématiques	<b>Formules en mode \$...\$ ou \[...\]</b>
Colonnes	<b>Disposition parallèle avec columns</b>
Blocs	<b>block, alertblock, exampleblock</b>

# Exemple de frame avec liste, image et formule

```
\begin{frame}{Introduction à Beamer}
    Voici une liste d'éléments :
    \begin{itemize}
        \item Premier point
        \item Deuxième point
    \end{itemize}

    \includegraphics[width=0.3\textwidth]{logo.png}

    Voici une formule : \[ E = mc^2 \]
\end{frame}
```

# Affichage progressif (animations simples)

- Avec `\pause`, vous pouvez faire apparaître les éléments un par un :

```
\begin{frame}{Étapes}
  \begin{itemize}
    \item Étape 1
    \pause
    \item Étape 2
    \pause
    \item Étape 3
  \end{itemize}
\end{frame}
```

# Utilisation des blocs (mise en forme stylée)

```
\begin{frame}{Encadrement}
    \begin{block}{Définition}
        Un bloc classique pour encadrer une information.
    \end{block}

    \begin{alertblock}{Attention !}
        Un bloc rouge pour les avertissements.
    \end{alertblock}

    \begin{exampleblock}{Exemple}
        Un bloc vert pour illustrer.
    \end{exampleblock}
\end{frame}
```

# Frame avec colonnes

```
\begin{frame}{Colonnes}
    \begin{columns}
        \column{0.5\textwidth}
        Contenu colonne 1
        \column{0.5\textwidth}
        Contenu colonne 2
    \end{columns}
\end{frame}
```

# Frames spéciales : titre + sommaire

```
\section{Introduction}
\begin{frame}{Introduction}
    Texte d'introduction ici.
\end{frame}
\section{Conclusion}
\begin{frame}{Conclusion}
    Texte de Conclusion ici.
\end{frame}
\begin{frame}
    \tableofcontents[currentsection]
\end{frame}
```

# Bonnes pratiques de structuration

- Utilisation des commandes `\section{}` et `\subsection{}` pour structurer logiquement la présentation.
- Utilisation de `\AtBeginSection` pour afficher automatiquement un sommaire à chaque début de section.

# Utiliser les commandes `\section{}` et `\subsection{}`

- Beamer permet de structurer logiquement une présentation comme un document :

```
\begin{document}

  \frame{\titlepage}
  \section{Introduction}
  \subsection{Contexte}
  \subsection{Objectifs}
```

# Générer un sommaire dynamique

- Insérer un sommaire au début ou au fil des sections :

```
\begin{frame}
  \frametitle{Plan de la présentation}
  \tableofcontents
\end{frame}
```

Pour qu'il affiche uniquement les sections principales :

```
\tableofcontents[hideallsubsections]
```