

Introduction au LATEX

DR. FATMA BELABED

2022-2023

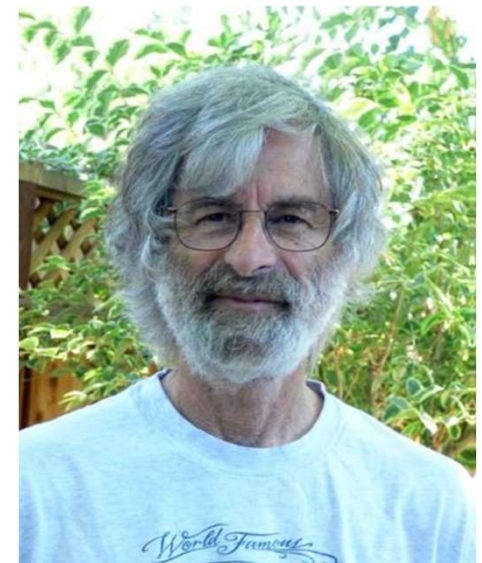


Introduction

TEX est un logiciel d'édition développé par Donald KNUTH, puis modifié par Leslie LAMPORT (LATEX) permettant de produire des documents de qualité digne de la publication professionnelle.



Donald Knuth



Leslie Lamport

LATEX, pour faire quoi?

- **Easy** Des rapports de stage, mémoire, thèses. . .
 - **Professional** Des présentations orales avec transparents
 - **Hardcore** Des “posters” pour présentation orale
 - **Out of control** Du grec, de la chimie, des partitions musicales. . .
- ❖ **Anecdotes**
- Le «X» de LaTeX est en fait un chi grec et se prononce donc comme un K.

Installation sous Windows

- Installer MikTeX
 - <http://miktex.org/>

- Installer Texmaker
 - http://www.xmlmath.net/texmaker/index_fr.html

❖ Remarque

- Il vous faut les deux, car **MikTeX** est une version du programme LATEX, tandis que **Texmaker** est simplement un éditeur contenant plein de fonctions utiles!

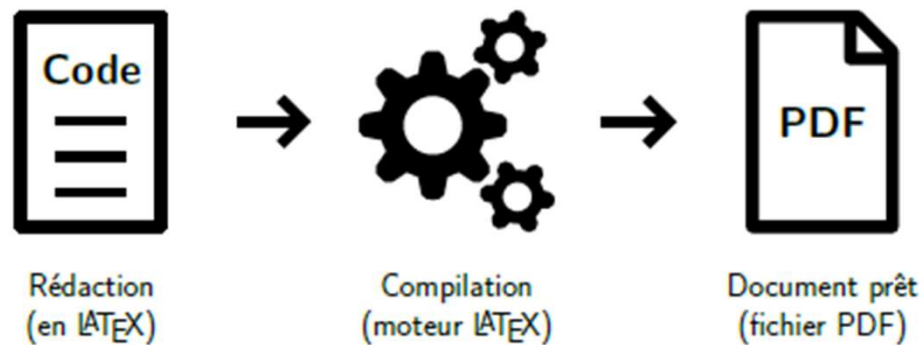
Les règles pour faire du LATEX

Ces règles sont officieuses:

- **Règle no 1 :** Tout est possible en LATEX.
- **Règle no 2 :** La règle no 1 est toujours vraie.
- **Règle no 3 :** LATEX implique d'écrire des commandes soit des lignes de code. Aérer et ordonner son code en facilite la relecture.
- **Règle no 4 :** La voie de la perfection en LATEX passe par une recherche régulière sur Internet.
- **Règle no 5 :** Si tu rencontres des difficultés, il ne faut pas hésiter à demander des conseils.

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.



Plusieurs moteurs de Tex

- Le moteur de TeX ne pouvait produire que des fichiers au format DVI et ne fonctionnait bien évidemment que pour TeX.
- Le moteur actuel de **LaTeX** est PdfTeX qui permet de générer des fichiers au format PDF.
- Cependant, plusieurs autres moteurs sont ensuite apparus.
- Après PdfTeX, il y a eu **XeTeX** pour utiliser des polices True Type ou Open Type, c'est-à-dire celles utilisables directement par le système d'exploitation et ses logiciels) qui ne peut générer que des fichiers PDF.
- Enfin, il y a eu **LuaTeX**, basé sur le langage **Lua**, qui permet de réellement programmer à l'intérieur du document pour toujours plus de contrôle sur celui-ci. Il n'est pas encore finalisé à ce jour, mais est utilisable.
- LuaTeX est destiné à devenir le successeur de PdfTeX.
- Un moteur peut avoir une commande pour créer un document PDF et une autre pour créer un document DVI. Cela donne lieu à plusieurs commandes pour créer un fichier.

Les fichiers LATEX

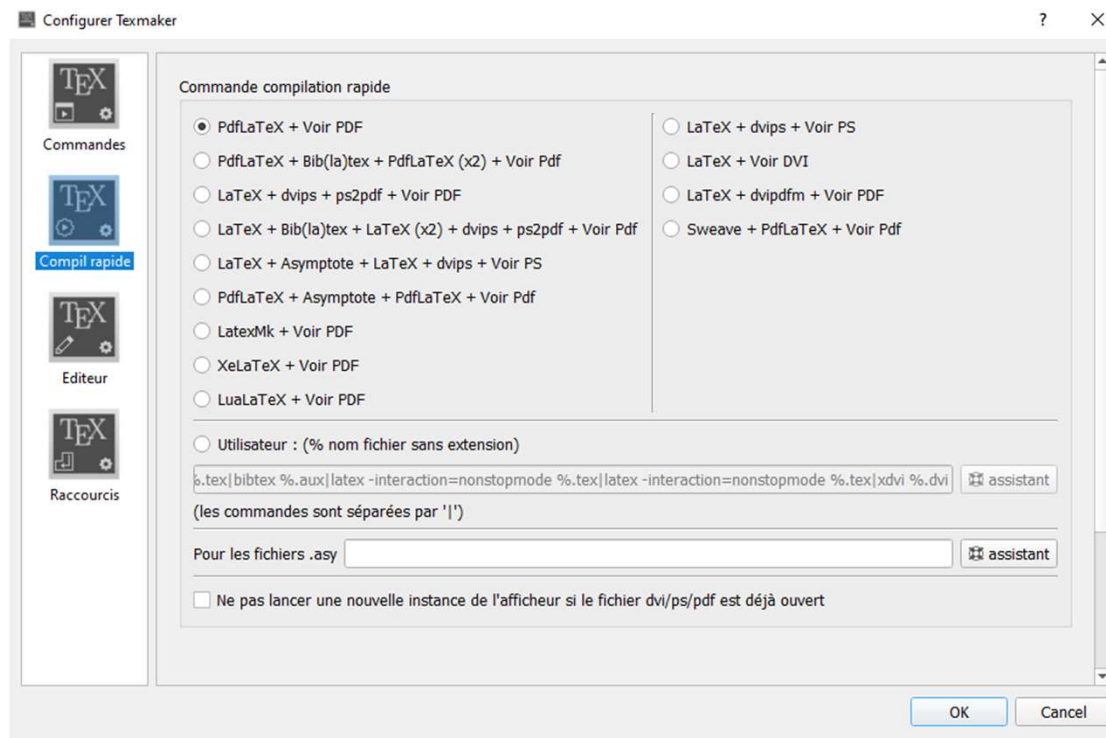
LATEX est un langage de programmation, qui génère plusieurs types de fichiers. On trouve des fichiers:

- **.tex**: Ce sont les fichiers contenant toutes les commandes que vous allez taper, i.e les fichiers sources
- **.dvi**: C'est le résultat de la compilation standard de vos commandes. On peut visualiser ces fichiers à l'aide du logiciel xdvi
- **.ps** ou **.pdf**: Il s'agit des fichiers destinés à la publication, après conversion depuis le .dvi
- **.bib** et **.bbl**: Ces fichiers servent à la gestion de la bibliographie
- **.aux**, **.toc**, **.idx**: Ces fichiers sont utilisés par LATEX pour gérer les références dans votre document
- **.log** contient les informations de compilation.

Compilation: conseils

- Peu importe le moteur utilisé pour la compilation, un fichier .tex entraîne toujours la génération d'autres fichiers (.log, .aux, .bib, .bbl...)
- C'est pourquoi je recommande toujours de travailler avec le fichier .tex placé dans un dossier pour éviter de submerger tes autres dossiers et te perdre parmi les fichiers.

Configuration



Principes de base

- Les commandes sont le cœur de LaTeX.
- Elles permettent par exemple d'indiquer que tel chose est un titre, qu'une autre est importante...
- **\commande**: Une commande commence par un antislash suivi de son nom.
- De plus, une commande peut prendre des arguments.
- Il y a deux titres d'arguments: **les arguments obligatoires et les arguments optionnels**.

Les 3 règles d'or en LATEX

- **Règle d'or no 1 :** Toute commande LATEX débute par un backslash
- **Règle d'or no 2 :** Tout texte concerné par une commande LATEX est délimité par des accolades “{” et “}”.
- **Règle d'or no 3 :** Toute commande LATEX qui comprend un begin finit par un end: ce genre de structure s'appelle un environnement.

Structure d'un document LATEX

- Un fichier source LATEX est toujours composé de deux parties.

Préambule

`\documentclass[11pt,french]{article}`

`\usepackage{babel}`

`\usepackage[autolanguage]{numprint}`

`\usepackage[utf8]{inputenc}`

`\usepackage[T1]{fontenc}`

corps

`\begin{document}`

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec quam nulla, bibendum vitae ipsum vel, fermentum pellentesque orci.

`\end{document}`

Différents types de documents

- **report** : petits documents (1 ou 2 pages)
 - **article** : rapports courts, articles de revues, ...
 - **book** : documents très longs (livres, thèses, ...)
 - **letter** : rédiger des lettres
 - **beamer** : pour faire des présentations avec des slides
- Le choix d'un style détermine certaines caractéristiques du document comme par exemple la taille des titres, les commandes de sectionnement autorisées ou encore les indentations. Chaque style admet un certain nombre d'options qui permettent de préciser par exemple la taille par défaut des caractères.

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 à 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 `\usepackage`.
 - `\usepackage[option]{package}`

Les packages, un système d'extensions

- Écrire en français demande de faire appel à des packages particuliers :
 - `\usepackage[T1]{fontenc}`
 - `\usepackage[utf8]{inputenc}`
 - `\usepackage[francais]{babel}`
- Les différences d'encodage s'estompent avec le temps, mais posent toujours des problèmes. Il vous faudra parfois revenir en latin1 au lieu de utf8, même sous Linux (si par exemple vous échangez des fichiers avec quelqu'un qui travaille sous Windows).
- Une règle générale pour les accents, et cédilles : vous éviterez les problèmes en tapant vos accents sous la forme suivante:
 - `\ + accent + lettre` : `\'e` ou `\'a` ou encore `\c{c}`

Les packages, un système d'extensions

- **babel** : traduction des mots-clés prédéfinis, typographie française, coupure de mots, document multilingue
- **inputenc** et **fontenc** : lettres accentuée
- **icomma** : virgule comme séparateur décimal
- **numprint** : espace comme séparateur des milliers dans le code source (pdfLATEX seulement)

Un premier essai : code source

```
\documentclass[12pt,a4paper]{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\begin{document}
\section{Premier essai}
Bonjour tout le monde. Je parle et \text{'}ecris fran\c{c}ais.
Je m'essaie au \LaTeX.\
Pour l'instant, tout va bien !
\subsection{La suite}
Passons \text{'}a la suite.
\end{document}
```

Sectionnement et Style

- On distingue par ordre décroissant :
 - `\part{titre}` `\chapter{titre}`
 - `\section{titre}` `\subsection{titre}` `\subsubsection{titre}`
 - `\Paragraph{titre}` `\subparagraph{titre}`
- La numérotation est automatique.
- Il est possible de l'enlever en rajoutant *:
 - `\section{Section avec numérotation}`
 - `\section*{Section sans numérotation}`
- ❖ Que se passe-t-il si on alterne les sections numérotées et non-numérotées?

Taille des caractères

<code>tiny</code>	<code>minuscule</code>
<code>scriptsize</code>	très petit
<code>footnotesize</code>	assez petit
<code>small</code>	petit
<code>normalsize</code>	normal
<code>large</code>	grand
<code>Large</code>	plus grand
<code>LARGE</code>	très grand
<code>huge</code>	énorme
<code>Huge</code>	géant

Style de caractères

- `\textbf{Gras}`: **Gras** → "Bold" ou Gras
- `\textit{italique}` ou `\emph{italique}`: *Italique* → "Italic" ou Italique
- `\underline{souligné}`: souligné → "Underlined" ou Souligné

Espacements sous LaTeX

1. Commandes pour les espaces horizontaux fixes:

- espace insécable : `\,`
- espace fine : `\,`
- espace moyenne : `\:`
- espace large : `\;`
- espace plus large (cadratin) : `\quad`
- espace encore plus large (double cadratin) : `\qquad`
- exemple d'espace horizontal de largeur donnée : `\hspace*{2cm}`

E spacements sous LaTeX

2. Commandes pour les espaces horizontaux "élastiques":

- `\hfill` : remplit l'espace restant sur la ligne courante
- `\dotfill` : remplit l'espace restant sur la ligne courante par des pointillés
- `\hrulefill` : remplit l'espace restant sur la ligne courante par un ?let horizontal

3. Commandes pour les espaces verticaux fixes :

- petit espace vertical : `\smallskip`
- espace vertical moyen : `\medskip`
- grand espace vertical : `\bigskip`
- exemple d'espace vertical de hauteur donnée : `\vspace*{2cm}`

Espacements sous LaTeX

4. Commandes pour les espaces verticaux "élastiques":

- `\vfill` : remplit l'espace vertical restant sur la page courante

5. Autres espacements :

- `\linebreak` : "suggère" à LaTeX de couper la ligne à l'endroit du document où la commande est utilisée (commande inverse : `\nolinebreak`)
- `\pagebreak` : "suggère" à LaTeX de changer de page à l'endroit du document où la commande est utilisée (commande inverse : `\nopagebreak`)
- `\newpage` : impose un changement de page à l'endroit du document où la commande est utilisée

Mettre de la couleur

- Appeler le package color en début de document :
 - `\usepackage{color}`

- Deux fonctions existent :

1. `\color{blue}`

- Tout le texte qui sera placé après cette balise sera de couleur bleue.

❖ **Ne pas oublier**

- `\color{black}`: pour revenir à la couleur par défaut.

2. `\textcolor{magenta}{le texte en magenta}`

- Seul le texte placé en deuxième argument (ici "le texte en magenta") sera de couleur magenta.

Faire un titre, auteur, date

- Instructions `\title`, `\author`, `\date` et `\abstract`.
- Puis instruction `\maketitle`.

Table des matières

- La commande `\tableofcontents` produit automatiquement la table des matières.
- Requiert plusieurs compilations
- ❖ Sections non numérotées pas incluses
- ❖ `\addcontentsline{toc}{chapter}{Conclusion et Perspectives}`: permet d'ajouter dans la table des matières Conclusion et Perspectives
- ❖ `\setcounter{secnumdepth}{3}` avec `\setcounter{tocdepth}{3}`: Include \subsubsection dans ToC

Liste des figures, des tableaux

- La commande `\listoffigures` produit automatiquement la liste des figures.
- La commande `\listoftables` produit automatiquement la liste des figures.
 - Requiert plusieurs compilations
 - Figures/tables non numérotées pas incluses

Haut et bas de page

- Ils sont gérés automatiquement par LATEX.
- Mais si vous voulez les modifier, vous pouvez utiliser le package fancyhdr, disponible depuis la page de style fancyhdr.sty.

```
\documentclass[11pt]{article}
```

```
\usepackage{fancyhdr}
```

```
\pagestyle{fancy}
```

```
\begin{document}
```

```
\lhead{haut de page gauche}
```

```
\chead{haut de page centre}
```

```
\rhead{haut de page droit}
```

Haut et bas de page

`\lfoot{Pied de page gauche}`

`\cfoot{Pied de page centre}`

`\rfoot{Pied de page droit}`

`\newpage`

Bla Bla Bla ...

`\end{document}`

- ❖ L'instruction `\thepage` incluse dans les pieds de page permet d'indiquer les numéros de page.

Environnements

- Délimités par:

`\begin{environnement}`

...

`\end{environnement}`

- ❖ Contenu de l'environnement traité différemment du reste du texte
- ❖ Changements s'appliquent uniquement à l'intérieur de l'environnement

Les environnements les plus utilisés

- Différents environnements permettent de structurer un document, nous allons détailler les principaux:
 - Les listes
 - Les tableaux
 - Les figures
 - La bibliographie
- ❖ Un environnement se commence toujours par `begin{environnement}` et se fini par `end{environnement}`

Présenter des cas : les listes

- Les listes se déclarent dans un environnement avec un `begin` et un `end` dont l'intitulé dépend du type de liste que vous voulez générer :
- `enumerate`, `itemize`, `description`.

```
\begin{enumerate}  
  \item liste 1  
  \item liste 2  
  \item liste 3  
\end{enumerate}
```

Présenter des cas : les listes

```
\begin{itemize}  
  \item liste  
  \item tableaux  
  \item liste numérotée  
\end{itemize}
```

```
\begin{description}  
  \item[cas 1] liste  
  \item[cas 2] tableaux  
  \item[cas 3] liste numérotée  
\end{description}
```

Les tableaux

- Environnements **table** et **tabular**.
- Pour obtenir des tableaux typographiquement corrects, il est préférable de charger l'extension array: `\usepackage{array}`

```
\begin{table}[pos]
\centering
... % Définition du tableau en soi
\caption{Tableau comparatif}
\label{tab:myTab}
\end{table}
```

Le tableau `\ref{tab:myTab}` montre...

Les tableaux

- L'option `pos` définit Les options de positionnement pour l'alignement du table.

Position	Signification
<code>h</code>	<i>here</i> : placement au mieux , là où l'élément apparaît dans le texte (éventuellement sur la page suivante s'il occupe un trop grand espace jusqu'à la fin de la page)
<code>H</code>	<i>HERE</i> : placement exactement là où apparaît la définition dans le texte
<code>t</code>	<i>top</i> : en haut d'une page de texte
<code>b</code>	<i>bottom</i> : en bas d'une page de texte
<code>p</code>	<i>page of float</i> : sur une page à part qui ne contient aucun texte

Les tableaux

- L'environnement **tabular** (ou `array` pour le mode mathématique) permet de créer des tableaux horizontaux ou verticaux et LATEX détermine automatiquement la taille des colonnes.
- La syntaxe est la suivante :

```
\begin{tabular}[pos]{cols}
```

```
ligne1 \ \
```

```
ligne2 \ \ ...
```

```
\end{tabular}
```

- L'option **pos** définit Les options de positionnement pour l'alignement du tableau similaires à celles de l'environnement `table`
- L'argument **cols** configure le format des colonnes. Chaque caractère correspond à une colonne ou une séparation de colonne :

Les tableaux

- **l** : une colonne où les éléments sont alignés à gauche
- **r** : une colonne où les éléments sont alignés à droite
- **c** : une colonne où les éléments sont centrés
- **|** : une ligne verticale entre deux colonnes

```
\begin{tabular}{l l l}
```

```
1.1 & 1.2 & 1.3 \\\
```

```
2.1 & 2.2 & 2.3 \\\
```

```
\end{tabular}
```

Les tableaux

```
\begin{table}[h!]  
  \begin{tabular}{|l|cc|}  
    OS & Plateforme & Part des serveurs http \ \\  
    \hline  
    Unix & Toutes & 32\% \ \\  
    Linux & Toutes & 26\% \ \\  
    Windows NT & Intel & 23\% \ \\  
  \end{tabular}  
  \caption{des serveurs occupés par chaque système d'exploitation.}  
  \label{tab_serveur}  
\end{table}
```

Les tableaux: Fusion des cellules

- Il est possible de fusionner plusieurs colonnes à l'aide de la commande:
`\multicolumn{num}{cols}{texte}`.
 - **num** précise alors le nombre de cellules à fusionner
 - **cols** le format de la colonne
 - **texte** correspond au texte à écrire dans la cellule.
- ❖ La fusion de plusieurs ligne nécessite le package **multirow** et la commande du même nom **`\multirow{num}{l}{texte}`** qui prend donc trois paramètres : le nombre de lignes à fusionner, la largeur de la cellule et enfin le texte à inclure.

Les tableaux: Fusion des cellules

```
\begin{tabular}{|c|c|c|c|}  
  \hline  
  \multirow{2}{1.5cm}{\centering Fruit} & \multicolumn{3}{c|}{Vente} \\ \cline{2-4} & 1995 & 1996 & 1997 \\ \hline  
  Poire & 6.5 & 7.8 & 7.2 \\ \hline  
  Pomme & 15 & 16 & 15.5 \\ \hline  
  Prune & 7.8 & 9 & 9.2 \\ \hline  
 \end{tabular}
```

Les figures

- On va le plus souvent utiliser l'environnement **figure**. On peut y insérer différents types d'images:
- 1. Les autres formats, dont PDF, sont employés avec la commande **\includegraphics** proposée par le package **graphics**

Les figures

```
\begin{figure}[h!]  
\begin{center}  
\includegraphics[height=7cm]{CHAMEAU.pdf} \\  
% [scale=0.3] : fixe l'échelle de l'image à 30%;  
% [width=5cm] : fixe la largeur de l'image à 5 cm;  
% [height=3cm] : fixe la hauteur à 3 cm  
\caption{{\emph{Le joli chameau !!!}}}  
\label{fig1}  
\end{center}  
\end{figure}
```

Ceci est un commentaire

➤ \includgrafics peut accepter n'importe quel type d'image: eps, jpg, png, pdf....

Les maths et LATEX

- C'est la vraie raison du succès de LATEX, qui permet une écriture aisée et rigoureuse de formules mathématiques.
- Il est conseillé d'utiliser les packages additionnels **amsmath**, **amsthm**, **amsfonts**, **amssymb** qui facilitent la vie.
- Pour écrire des formules et/ou des symboles mathématiques, il faut d'abord passer en mode mathématiques.
- On distingue principalement deux méthodes:
 1. `$...$` ou `\(...\)` : la formule fait partie d'un texte.
 2. `\[...\]` : le formule est mise en évidence : il y a un saut de ligne et la formule est centrée.

❖ Exemple :

- Soit $\epsilon > 0$ un seuil arbitraire :

```
\[  
  \forall \alpha \geq \epsilon,  
  f(\alpha) < \frac{1}{2}  
\]
```

Les maths et LATEX: Les équations

- Sur une seule ligne: On utilise l'environnement **equation** :

```
\begin{equation}
\label{eq:test}
f(x) + g(x) = \frac{1}{x-1}
\end{equation}
```

L'équation `\ref{eq:test}` montre.....

- Comme toujours, on peut supprimer la numérotation en utilisant :
`\begin{equation*}...\end{equation*}`

Les maths et LATEX: Les équations

- Sur plusieurs lignes, alignées: On utilise alors l'environnement `\equarray*` :

```
\begin{eqnarray*}
```

```
\cos 2 \theta &= & \cos^2 \theta - \sin^2 \theta \quad \backslash
```

```
&= & 2 \cos^2 \theta - 1.
```

```
\end{eqnarray*}
```

- Sans le caractère '*', chaque ligne sera numérotée.

La bibliographie - dans le .tex

- Environnement `\thebibliography`
- Commande `\bibitem` pour "labelliser" les références
- Commande `\cite` pour appeler les références dans le texte

Exemple

- On englobe toutes les entrées dans l'environnement `\thebibliography` (placé à la fin du document) qui reçoit en argument le nombre maximum d'entrées contenues.
- Ensuite, chaque entrée est introduite par la commande `\bibitem{clef}`
 - `\bibitem{clef} Auteur, TITRE, éditeur, année`
- Avec l'appel de la commande `\cite{clef}`, la référence est citée.

Exemple

```
\begin{thebibliography}{9}
\bibitem[STI96]{stinson96} Douglas Stinson, \
{\it Cryptographie, Theorie et pratique}, Vuibert, 1996
\end{thebibliography}
```

Les citations `\cite{stinson96}` peut être cité sous la forme simple présente ou sous une forme différente

La bibliographie - avec un .bib

- On définit un ou plusieurs fichiers file.bib qui contiennent les entrées bibliographiques.
- Ces fichiers sont appelés dans le document source par la commande `\bibliography{file1, file2,...}`.
- Le contenu d'un file.bib est un ensemble de déclarations par catégorie :

```
@CATEGORIE{clef,  
    author = {liste-noms-auteur},  
    title = {titre-article},  
    month = {mois-parution},  
    year = {annee-parution},  
    note = {note},  
}
```

La bibliographie - avec un .bib: styles bibliographiques

- Il existe plusieurs styles de présentation des bibliographies. Ils sont configurés par la commande **`\bibliographystyle{style}`**.
- Les principales valeurs du paramètre style sont :
 - **plain** : les entrées sont triées par ordre alphabétique et apparaissent numérotées
 - **unsrt** : les entrées ne sont pas triées et apparaissent numérotées dans l'ordre d'apparition
 - **abbrv** : comme plain mais les prénoms des auteurs, les noms des journaux et les mois sont abrégés
 - **alpha** : comme plain mais les clefs ne sont plus de simples numéros. Ils se bases sur les initiales des auteurs et l'année de publication.

La commandes `\input`

- Pour inclure un fichier dans un autre, nous pouvons utiliser la commande `\input` qui prend en paramètre le chemin du fichier à inclure.
- S'il s'agit de deux fichier latex, on n'ajoute pas l'extension du fichier à inclure.
- S'il s'agit d'un autre fichier à inclure (autre que latex), il faut préciser son extension

La commande `\include`

- Pour inclure un fichier dans un autre, nous pouvons utiliser la commande `\include` qui prend en paramètre le chemin du fichier à inclure
- Avec cette commande, il ne faut pas mettre l'extension du fichier
- ❖ **la différence entre `include` et `input`**
 - la commande `\include` commence une nouvelle page avant d'inclure le fichier et en commence une autre après avoir inclus le fichier (elle est idéale pour les chapitre de livres).
 - De plus, un document qui est inclus avec la commande `\include` ne peut pas utiliser lui-même une commande `\include`, ceci mènera à une erreur. On ne peut pas imbriquer les `\include`
- La commande `\include` semble quand même plutôt limitée comparée à `\input`