

# LaTeX et la rédaction de mémoire (EnC 2556)

(pdf réalisé avec pandoc à partir du fichier cours.md)

Enimie Rouquette (2024-2025)

## Contents

<b>Premiers pas</b>	<b>3</b>
La structure d'un document . . . . .	3
Taper en LaTeX . . . . .	5
Les packages . . . . .	5
Les commandes . . . . .	5
Structurer un document . . . . .	6
Faire apparaître le titre du document . . . . .	6
Commandes de section . . . . .	6
Les environnements . . . . .	7
<b>Mettre en sens, mettre en forme</b>	<b>7</b>
Mettre en sens . . . . .	7
Le paratexte . . . . .	7
Listes . . . . .	7
Les citations . . . . .	8
Mettre en forme: quelques commandes et environnements . . . . .	9
Commandes portant sur le style de caractères . . . . .	9
Commandes “à bascule” pour changer la taille des caractères .	10
Environnements modifiant l'alignement du texte . . . . .	10
Commandes modifiant ponctuellement la mise en page . . . .	11
Encadrer du texte en le mettant dans des boîtes . . . . .	11
<b>Créer ses commandes et environnements</b>	<b>12</b>
Syntaxe pour créer une commande . . . . .	12

Syntaxe pour créer un environnement . . . . .	12
<b>Gérer sa bibliographie: biblatex, biber et Zotero</b>	<b>13</b>
Principes . . . . .	13
La base de données bibliographiques . . . . .	15
Types d'entrées . . . . .	15
Quelques champs . . . . .	16
Les styles bibliographiques . . . . .	19
Imprimer sa bibliographie . . . . .	20
Zotero et LaTeX . . . . .	21
Importer une bibliographie BibTeX vers Zotero: . . . . .	21
Le module BetterBibTeX . . . . .	22
<b>Les éléments non textuels</b>	<b>23</b>
Les flottants . . . . .	23
Insérer une image . . . . .	24
Insérer tableau . . . . .	24
Faire un graphique avec TikZ . . . . .	24
Faire des histogrammes avec TikZ et le package pgfplots . . . . .	25
Importer des données d'un fichier . csv pour obtenir tableaux et graphes avec TikZ, pgfplots et le package pgfplotstable . . . . .	25
Étape préliminaire: stocker les données dans une variable . . . . .	26
Importer les données dans un tableau . . . . .	26
Importer les données dans un diagramme . . . . .	26
<b>Les packages reledmac et reledpar pour l'édition critique</b>	<b>27</b>
reledmac . . . . .	27
reledpar . . . . .	28
<b>Naviguer dans un long document</b>	<b>29</b>
Les renvois internes . . . . .	29
Les index . . . . .	29
Création d'un index simple . . . . .	29
Créer plusieurs index . . . . .	30
Les glossaires . . . . .	31
Modifier les césures automatiques . . . . .	32
Créer un compteur . . . . .	32

Utiliser un document maître	33
“Trucs” et ressources utiles	34
Citer du code . . . . .	34
Utiliser plusieurs polices de caractères . . . . .	34
Quelques <i>packages</i> . . . . .	35
Quelques messages d’erreurs courants . . . . .	37
(Très courte) bibliographie . . . . .	38
Autres ressources . . . . .	38
Trouver de l’aide . . . . .	38
Forums francophones . . . . .	38
Forums anglophones . . . . .	38



Figure 1: LaTeX

## Premiers pas

### La structure d’un document

- un préambule
- le corps du texte entre les balises `\begin{document}` et `\end{document}`

```

\documentclass[12pt,a4paper]{article} %appel de classe
%appel des packages:
\usepackage{fontspec} %package pour gérer les fontes
\usepackage{xunicode} %package pour gérer l'unicode

```

```
\usepackage{polyglossia} %package pour gérer les langues
\setmainlanguage{french}
```

*%d'autres packages peuvent être appelés ici*  
*%des commandes et des environnements peuvent être (re)définis ici*

```
\begin{document}
corps du texte
\end{document}
```

**Attention** - La dernière version du package `polyglossia` pose des problèmes avec le style bibliographique de l'ÉnC. Charger à sa place `babel`: remplacer les lignes

```
\usepackage{polyglossia} %package pour gérer les langues
\setmainlanguage{french}
```

par:

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

- Quelques classes possibles: `article` `book` `beamer` `memoir`
- Quelques options possibles de l'appel de classe:

option	effet
10pt	pour une police de base en 10 pt
11pt	pour une police de base en 11 pt
12pt	pour une police de base en 12 pt
onecolumn	pour un texte sur une seule colonne (par défaut pour les classes présentées)
twocolumn	pour un texte sur deux colonnes
oneside	pour une impression en recto seulement
twoside	pour une impression en recto-verso
notitlepage	pour ne pas avoir de page de titre spécifique (par défaut pour la classe <code>article</code> )
titlepage	pour avoir une page de titre spécifique (par défaut pour la classe <code>book</code> )

## Taper en LaTeX

- le nombre d'espaces ne change rien
- un retour à la ligne → un espace
- un tilde (~) → espace insécable
- deux retours à la ligne ou plus → nouveau paragraphe
- -- → demi-cadratin, --- → cadratin
- gestion des espaces automatique avant ponctuation
- pour taper des points de suspensions: `\dots`
- tout ce qui suit le caractère % est ignoré (commentaire)
- attention, certains caractères sont utilisés par LaTeX: `\ ~ ^ { } % _ & $ #`. Pour insérer ces caractères dans un texte, il faut taper:  
`\textbackslash \textasciitilde \textasciicircum \{ \} \% \_ \& \$ \#`

## Les packages

- *Package*: code permettant de définir un ensemble de commandes orienté vers utilisation spécifique. l'ensemble de ce code est contenu dans un fichier avec l'extension .sty
- Appeler un package: dans le préambule, `\usepackage{nom_du_package}`
- Obtenir la documentation: taper dans un terminal `texdoc nom_du_package`

## Les commandes

- *Commande*: bout de code qui va être interprété par le compilateur.
- *Argument* paramètre passé à une commande et qui en modifie le comportement ou qui indique ce sur quoi elle doit s'appliquer .
- Syntaxe d'une commande:
  1. backslash (\ + nom de la commande)
  2. entre crochets: arguments optionnels
  3. entre accolades: arguments obligatoires

## Structurer un document

### Faire apparaître le titre du document

- dans le préambule: `\title{titre_du_document} \author{auteur_du_document} \date{date_du_document}` (pour qu'un de ces éléments n'apparaisse pas, laisser l'argument vide. Par défaut, si elle n'est pas indiquée, la date est celle du jour)
- dans le corps du document `\maketitle`

### Commandes de section

Commande	Sens	Numéro de niveau
<code>\part{#1}</code>	Titre de partie	-1
<code>\chapter{#1}</code>	Titre de chapitre	0
<code>\section{#1}</code>	Titre de section	1
<code>\subsection{#1}</code>	Titre de sous-section	2
<code>\subsubsection{#1}</code>	Titre de sous-sous-section	3
<code>\paragraph{#1}</code>	Titre de paragraphe	4
<code>\subparagraph{#1}</code>	Titre de sous-paragraphe	5

- l'argument (`#1`) est le titre que l'on donne à la section.
- La commande `\chapter` n'existe que pour la classe `book`
- Le numéro des niveaux de titre sert lors de l'affichage de la table des matières pour définir sa profondeur (voir dernier cours)
- Les niveaux dont les numéros sont inférieurs à 1 provoquent un changement de page.
- Les niveaux dont les numéros sont supérieurs à 3 ne provoquent pas de changement de paragraphe.
- Faire apparaître la table des matières: `\tableofcontents`
- Les commandes de section peuvent avoir un argument optionnel: c'est celui qui sera utilisé pour la table des matières
- Les commandes de section étoilées (par exemple `\section*{#1}`) ne sont pas numérotées et n'apparaissent pas dans la table des matières

- Pour les faire apparaître: `\addcontentsline{toc}{#1}{#2}`. #1 = le niveau de titre (section, chapter, etc); #2 = le titre que l'on veut faire apparaître dans la table des matières

## Les environnements

- *environnement*: portion de document qui a une signification spécifique et subit un traitement spécifique. L'environnement accepte le saut de paragraphe. On peut utiliser des commandes au sein d'environnements.
- syntaxe d'un environnement: `\begin{nom_de_l_environnement}`  
`\end{nom_de_l_environnement}`. Tout ce qui se trouve entre ces deux balises est à l'intérieur de l'environnement
- exemple: l'environnement `abstract` de la classe `article`

## Mettre en sens, mettre en forme

### Mettre en sens

#### Le paratexte

Commande	Effet
<code>\footnote{#1}</code>	Note de bas de page
<code>\marginpar{#1}</code>	Note marginale

- L'apparence du numéro dans la note de bas de page peut être modifiée avec `polyglossia`. Mettre à la commande `\setmainlanguage{french}`, dans l'argument optionnel, l'option `frenchfootnote=true`
- Ceci n'est pas nécessaire si on utilise le package `babel`

### Listes

- Au sein des environnements de liste, chaque nouvelle entrée est appelée par la commande `\item`

Environnement	Effet
<code>itemize</code>	liste simple

Environnement	Effet
<code>enumerate</code>	liste numérotée
<code>description</code>	liste faisant correspondre deux valeurs ensemble. La première valeur est mis dans l'argument optionnel de <code>\item</code>

- Si l'on met un argument optionnel à `\item`, on peut modifier ponctuellement le label (l'élément précédant l'entrée dans la liste: tiret, point,...)
- Pour modifier sur l'ensemble du document le label des environnements `itemize`: mettre à la commande `\setmainlanguage{french}` l'option `[frenchitemlabels=true]{french}` pour obtenir un cadratin; pour choisir le label, ajouter ensuite comme option `itemlabels=` avec le label choisi.
- Exemple: `\setmainlanguage[frenchitemlabels=true, itemlabels=\textendash]{french}` pour obtenir un demi-cadratin.
- **nb** si l'on utilise le package `babel` plutôt que `polyglossia`, les items sont déjà des cadratins
- Pour modifier le label d'un environnement `enumerate` donné, il faut utiliser le package `enumerate`. Exemple: `\begin{enumerate}[(I)]` pour numéroté la liste en chiffres romains entre parenthèses; `\begin{enumerate}[exemple 1:]` pour faire précéder chaque item de `exemple` suivi du numéro et de deux points. Compteurs possibles: `a` (lettres de l'alphabet), `A` (lettres en majuscules), `i` (chiffres romains en minuscule), `I` (chiffres romains en majuscule), `1` (chiffres arabes, par défaut).

## Les citations

Commande ou environnement	Effet
<code>\enquote{#1}</code> (requiert le package <code>csquotes</code> )	Guillemets
<code>quote</code>	Citation sur un paragraphe
<code>quotation</code>	Citation sur plusieurs paragraphes



Commande ou environnement	Effet
<code>\textelp{}</code> (requiert le package <code>csquotes</code> )	Indiquer une citation tronquée. Si l'argument est vide: met simplement des points de suspension
<code>\textins{#1}</code> (requiert le package <code>csquotes</code> )	Indiquer une modification de la citation
<code>\url{#1}</code>	Créer un lien vers une URL. Utiliser le package <code>hyperref</code> pour avoir un lien cliquable (à charger EN DERNIER) pour éviter des problèmes de compatibilité avec d'autres package
<code>verse</code> (requiert le package <code>verse</code> )	Pour citer des vers. Voir le manuel

## Mettre en forme: quelques commandes et environnements

### Commandes portant sur le style de caractères

Commande	Effet
<code>\textit{#1}</code>	Italique
<code>\emph{#1}</code>	Emphase (à préférer à <code>textit</code> )
<code>\textbf{#1}</code>	Gras
<code>\textsc{#1}</code>	Petites capitales
<code>\underline{#1}</code>	Souligné
<code>\textsuperscript{#1}</code>	Exposant
<code>\textsubscript{#1}</code>	Indice
<code>\textcolor{#1}{#2}</code> (nécessite le package <code>xcolor</code> )	Mettre en couleur

**nb** - `#1` est le texte à mettre en forme, sauf pour `\textcolor` où: `#1`=nom de la couleur, `#2`=texte à mettre en couleur - La commande `\emph` n'accepte pas les sauts de ligne. Pour mettre en emphase un texte plus long, utiliser l'environnement `em`

## Commandes “à bascule” pour changer la taille des caractères

- Si l’on veut modifier sur une portion de texte donnée la taille des caractères, il faut utiliser une commande à bascule. Le comportement de ce type de commandes est différent: elles ne prennent pas d’argument, mais sont insérées entre accolades; le texte dont la taille change est entre les accolades, après la commande.
- Exemple: `{\large un texte plus gros}` donne {un texte plus gros}
- La taille obtenue est relative: elle dépend de la taille définie lors de l’appel de classe.

Liste des commandes à bascule, pour un résultat du plus petit au plus gros (`normalsize` correspond à la taille du corps du texte, `footnotesize` à la taille des notes de bas de page):

Commande
<code>{\tiny }</code>
<code>{\scriptsize }</code>
<code>{\footnotesize }</code>
<code>{\small }</code>
<code>{\normalsize }</code>
<code>{\large }</code>
<code>{\Large }</code>
<code>{\LARGE }</code>
<code>{\huge }</code>
<code>{\Huge }</code>

## Environnements modifiant l’alignement du texte

Environnement	Effet
<code>flushleft</code>	Justifié à gauche
<code>flushright</code>	Justifié à droite
<code>center</code>	Centré

## Commandes modifiant ponctuellement la mise en page

Commande	Effet
<code>\newline</code> ou <code>\\</code>	Saut de ligne; la ligne suivante n'est pas indentée
<code>\newpage</code> ou <code>\pagebreak</code>	Commencer une nouvelle page
<code>\noindent</code>	Pas d'indentation au début du paragraphe
<code>\par</code>	Nouveau paragraphe (à utiliser dans la définition de commandes ou d'environnements)

nb:

- `\newpage` va mettre tout l'espace disponible restant en bas de la page; `\pagebreak` va répartir cet espace sur la page. Pour cette raison, `\newpage` vous sera plus utile.
- `\newline` ou `\\`, de même que `\par`, sont à réserver pour la définition de commandes ou d'environnement, et ne doivent qu'exceptionnellement être utilisés dans le corps du texte. Ces commandes **ne doivent pas être utilisées** pour créer des paragraphes au cours de la phrase (il suffit de laisser une ligne blanche!) , car elles peuvent provoquer des problèmes de répartition de blanc dans la page.

## Encadrer du texte en le mettant dans des boîtes

Commande/environnement	Effet
<code>\fcolorbox{#1}{#2}{#3}</code> (requiert le package <code>xcolor</code> )	Mettre du texte ( <b>#3</b> ) dans une boîte de la couleur choisie ( <b>#1</b> pour le cadre, <b>#2</b> pour le fond)
<code>\fbox{#1}</code> (ou <code>\framebox{#1}</code> )	Texte court encadré (sans changement de ligne/de paragraphe)
<code>framed</code> (requiert le package <code>framed</code> )	Encadre des textes qui peuvent courir sur plusieurs lignes/paragraphes

Il existe aussi le package `fancybox` pour créer des boîtes encadrées

Sur la notion de boîtes dans LaTeX, voir LOZANO, Vincent, Tout ce que vous avez toujours voulu savoir sur LaTeX sans jamais oser le demander, p. 73 *sq.*

## Créer ses commandes et environnements

### Syntaxe pour créer une commande

- `\newcommand{#1}[#2]{#3}`. #1= nom de la commande; #2=nombre d'arguments de la nouvelle commande; #3= code de la nouvelle commande
- Les commandes doivent être créées dans le préambule.
- Syntaxe pour créer une commande avec argument optionnel, grâce au package `xargs`:
- `\newcommandx{#1}[#2][#3]{#4}` #1=nom de la commande; #2= nombre d'arguments; #3=liste des arguments optionnels (liste de numéros indiquant la position des arguments optionnels dans la commande). Une valeur par défaut peut être attribuée aux arguments optionnels, consulter le manuel de `xarg`; #4= le code
- Tester si un argument est vide grâce au packet `etoolbox`: utiliser la commande `\ifstrempy{#1}{#2}{#3}`. #1=la chaîne de caractères que l'on va tester (ça peut être un argument de notre nouvelle commande, par exemple #1); #2=ce qui se passe si la chaîne est vide; #3=ce qui se passe sinon.

### Syntaxe pour créer un environnement

`-\newenvironment{#1}[#2]{#3}{#4}`. #1= nom de l'environnement; #2=nombre d'arguments du nouvel environnement; #3= code appelé à l'ouverture de l'environnement; #4 = code appelé à la fermeture de l'environnement

- Pour créer un environnement avec arguments optionnels, utilisez le package `xargs`
- **nb** Il n'est pas possible d'appeler l'argument du nouvel environnement

dans la *dernière partie* de la commande `newenvironment` (celle qui indique le code qui sera appelé à la fermeture de l’environnement). Il faut:

- créer une “boite de sauvegarde” en mettant, avant la définition du nouvel environnement, `\newsavebox{\boite}` (on peut bien sûr donner un autre nom à la boite).
- Dans le code appelé à l’ouverture du nouvel environnement, mettre l’argument dans cette boite, par exemple: `\savebox{\boite}{#1}`
- Dans le code appelé à la fermeture de l’environnement, on peut utiliser cette boite (et donc le contenu de l’argument) en utilisant: `\usebox{\boite}`.
- Pour des exemples d’utilisation, voir l’ouvrage de LOZANO, Vincent, cité plus haut et mis dans la bibliographie.

## Gérer sa bibliographie: biblatex, biber et Zotero

### Principes

1. un fichier au format `.bib` contient toutes les références bibliographiques (“entrées”) (= fichier BibTex)
  - à chaque entrée est associée une clef (“key”) qui l’identifie
  - chaque entrée comporte un certain nombre d’éléments de description bibliographique (nature de la référence, titre, auteur, date, etc), chacun indiqué dans le champ correspondant.
  - exemple d’entrée:

```
@Book{maieul_rouquette_2012,
  author = {Maïeul Rouquette and Enimie Rouquette and Brendan Chabannes},
  title = {(Xe)\LaTeX{} appliqué aux sciences humaines},
  publisher = {Atramenta},
  date = {2012},
  address = {Tampere},
  url = {https://halshs.archives-ouvertes.fr/halshs-00924546}
}
```

Dans cet exemple, la référence est un livre (`@book`), sa clef est `maieul_rouquette_2012`, et les éléments indiqués sont les auteurs, le titre, l'éditeur, la date, le lieu de publication, et l'url pour la publication en ligne.

2. Dans le préambule, on charge le package `biblatex` et on indique le chemin du fichier `.bib` avec la commande `\addbibresource{#1}`. C'est `biblatex` qui affiche les citations selon un style choisi, en utilisant le moteur `biber`.

Exemple de préambule appelant `biblatex` et un fichier bibliographique:

```
\documentclass[a4paper]{book}
\usepackage{fontspec}
\usepackage{xunicode}
\usepackage{polyglossia}
\setmainlanguage{french}
\usepackage{biblatex}
\addbibresource{/home/user/Documents/memoire/bibliographie.bib}
```

**nb** il est possible d'appeler plusieurs fichiers bibliographiques.

3. Exemple de commandes `biblatex`:

- pour faire une citation dans le corps du texte et en note de bas de page: `\cite[#1][#2]{#3}` et `\footcite[#1][#2]{#3}`, où `#1`= une prénote (texte avant la référence), `#2`=postnote (numéro de page, texte après la référence), `#3`=la clef de la référence bibliographique
  - S'il n'y a qu'un seul argument optionnel, il est considéré comme l'argument `postnote`. Si l'on veut l'argument `prenote` sans `postnote`, il faut mettre l'argument `#2` vide
  - Si `#2` ne contient que le numéro de la page, le préfixe (p., pp., col.,) s'affiche automatiquement. (par défaut, on pagine en pages. Si l'on veut paginer en colonnes, il faut ajouter dans l'entrée bibliographique le champ `pagination` avec pour valeur `column` ; voir le manuel pour d'autres valeurs possibles)
  - Si la postnote contient aussi du texte, il faut faire précéder le numéro de pagination de la commande `\pno~`.

Exemples:

```
\footcite[Sur ce sujet, voir][99]{maieul_rouquette_2012}
```

`\footcite[Sur ce sujet, voir][maieul_rouquette_2012]`  
`\footcite[\pno~99 et suivantes]{maieul_rouquette_2012}`

- pour ne citer respectivement que l'auteur et le titre d'une référence:  
`\citeauthor{clef}` et `\citetitle{clef}`
- pour imprimer la bibliographie: `\printbibliography`

4. Trois compilations sont nécessaires:

- compilation avec XeLaTeX
- compilation avec Biber
- nouvelle compilation avec XeLaTeX
- Configurer TexStudio pour compiler avec biber: `options - configurer texstudio - production - moteur de bibliographie par défaut - biber`. Il faut parfois indiquer le chemin de biber dans l'onglet `compilation` des configurations: on le connaît en tapant `which biber` dans le terminal (pour linux)
- Pour compiler soi-même: dans un terminal, se placer là où est le fichier à compiler, et taper `xelatex nom_du_fichier`, puis `biber nom_du_fichier`.

**nb:** - la première compilation produit un fichier auxiliaire `.bbl`, c'est lui que l'on compile avec Biber (mais ce n'est pas la peine de mettre l'extension si l'on compile dans le terminal) - Si l'on compile manuellement dans le terminal, il est possible d'utiliser la commande `latexmk` qui fait automatiquement les compilations nécessaires. Voir sur ce sujet le blog *geekographie maïeulesque*

## La base de données bibliographiques

### Types d'entrées

Quelques types d'entrées possibles; pour plus d'entrées, voir le manuel de `biblatex` p.8sq, ainsi que ROUQUETTE, Maïeul, XeLaTeX appliqué aux sciences humaines, p.81-82 sq

Nom	type d'entrée
@article	Article de revue

Nom	type d'entrée
@book	Livre
@bookinbook	Livre dans un livre, par exemple différents écrits d'un auteur rassemblés en un seul volume
@collection	Ouvrage collectif mais avec des parties distinctes par auteur
@inbook	Partie d'un livre
@incollection	Partie individuelle (avec un auteur propre) dans un ouvrage collectif
@inproceedings	Contribution à un colloque
@inreference	Article de dictionnaire, d'encyclopédie, etc
@misc	Entrée générique, pour tout type d'entrée non catégorisable
@online	Ressource internet (sans équivalent physique)
@periodical	Numéro précis d'un périodique
@proceedings	Acte de colloque
@suppbook	Partie annexe d'un livre, comme par exemple la préface ou les appendices
@reference	Dictionnaire, encyclopédie, etc
@thesis	Thèse de doctorat, mémoire de maîtrise ou tout travail rédigé en vue de l'obtention d'un titre scolaire ou universitaire.
@unpublished	Ouvrage non publié

## Quelques champs

Pour plus de champs, voir le manuel de `biblatex` p.17sq, ainsi que ROUQUETTE, Maïeul, XeLaTeX appliqué aux sciences humaines, p.83 sq

### *Champs de personnes*

- Les valeurs des champs de personnes sont de type “Prénom Nom” ou “Nom, Prénom”. Pour considérer un ensemble de mot comme le “nom” (par exemple pour les auteurs antiques), le mettre entre accolades.



- Lorsqu'il y a plusieurs personnes dans un même champ, elles sont séparées par **and**

exemples: - `author = {Maïeul Rouquette and Enimie Rouquette and Brendan Chabannes}`

ou: `author = {Rouquette, Maïeul and Rouquette, Enimie and Chabannes, Brendan}` - `author = {{Eugène de Tolède}}`

Champ	Explication
<code>annotator</code>	Auteur(s) des annotations
<code>author</code>	Auteur(s) de l'œuvre
<code>commentator</code>	Auteur(s) des commentaires
<code>editor</code>	Éditeur(s) scientifique(s). On peut en préciser le rôle grâce au champ <b><code>editortype</code></b>
<code>foreword</code>	Auteur(s) de la préface
<code>introduction</code>	Auteur(s) de l'introduction
<code>translator</code>	Traducteur(s)

Lorsque des champs possèdent des valeurs égales, biblatex fusionne ces champs lors de l'affichage. Par exemple, si `translator` et `editor` sont identiques, on obtient "édité et traduit par..."

#### *Champs de titre*

Champ	Explication
<code>booktitle</code>	Titre du livre dans lequel l'entrée se situe
<code>chapter</code>	Chapitre d'un livre. Pour les entrées de type <code>@inbook</code>
<code>issuetitle</code>	Titre d'un numéro spécifique d'un périodique
<code>journaltitle</code>	Titre d'un périodique
<code>maintitle</code>	Titre d'une œuvre en plusieurs volumes
<code>maintitleaddon</code>	Ajout au titre d'une œuvre en plusieurs volumes
<code>subtitle</code>	Sous-titre de l'œuvre
<code>title</code>	Titre de l'œuvre
<code>titleaddon</code>	Ajout au titre de l'œuvre

### *Champs d'information sur la publication*

Pour plus de champs, voir le manuel et ROUQUETTE, Maïeul, XeLaTeX appliqué aux sciences humaines, p.88 sq

Champ	Explication
address	Lieu de publication.
date	Date de publication
edition	Numéro d'édition si plusieurs éditions existent
institution	Institution dans laquelle l'œuvre a été produite, pour les entrées de type @thesis
issue	Détail d'un numéro spécifique d'un périodique
number	Numéro d'un périodique ou numéro au sein d'une collection
origdate	Date de l'édition originale
origlanguage	Langue originelle
pagination	Le type de pagination (par défaut: <b>page</b> )
pages	Pages de l'article ou de la partie du livre étudiée
publisher	Éditeur commercial
type	Pour les entrées de type @thesis, précise le type de travail
url	Url (adresse électronique) d'une publication en ligne
urldate	Date à laquelle une publication électronique a été consultée
volume	Volume dans une œuvre en plusieurs volumes
volumes	Nombre de volumes dans une œuvre en plusieurs volumes

## Les styles bibliographiques

- Il est possible de charger différents styles pour les citations et pour la bibliographie, dans l'argument optionnel de l'appel du package:
  - L'option `style=nom_du_style` charge à la fois le style de citation et le style bibliographique
  - Les options `bibstyle=nom_du_style` et `citestyle=nom_du_style` chargent respectivement le style bibliographique et le style de citation

Exemples de style:

Style	Description
numeric	chaque entrée se voit attribuer un numéro, qui est appelé lorsque l'on renvoie à cette entrée.
authortitle	sont indiqués seulement l'auteur et le titre de l'œuvre
verbose	la description complète de l'entrée est donnée la première fois, une version abrégée est affichée ensuite.
verbose-ibid	Comme verbose, mais avec l'abréviation <i>ibidem</i>
verbose-trad1, verbose-trad2, verbose-trad3	Comme verbose-ibid, mais avec également les abréviations <i>op. cit.</i> , <i>ibidem</i> , etc.

Pour plus de précisions, voir le manuel p.74 sq

- Il existe un style de citations qui suit les normes de l'École des Chartes:
  - biblatex-enc
  - on peut l'installer dans son dossier `texmf local` (par exemple `/usr/local/texlive/texmf-local/tex/latex/biblatex-enc`) (voir le readme du style).
  - **nb:** pour en apprendre plus sur l'arborescence d'une distribution TeX et ce que signifie `texmf local`, voir: Guide pratique de Tex Live 2023 p.7, ou Daniel Flipo, Administration d'une installation TeX

- On le passe ensuite en option à `biblatex`: `\usepackage[style=enc]{biblatex}`

**Attention** La dernière version de `polyglossia` provoque des erreurs de compilation avec ce style. Deux possibilités: - télécharger et utiliser localement la version antérieure de `polyglossia` en attendant que ce problème soit résolu - utiliser le package `babel` à la place de `polyglossia` (voir dans la définition du préambule)

---

**RQ** - Pour que les œuvres anonymes soient classées en début de bibliographie, par titre, il faut utiliser le package `biblatex-anonymous`, et ajouter l'option `sorting` à l'appel du package `biblatex`:

```
\usepackage[style=enc,sorting=anonymous]{biblatex}
\usepackage{biblatex-anonymous}
```

- l'appel du package `biblatex-anonymous` doit se faire après l'appel du package `biblatex`

**Attention** `biblatex-anonymous` ne fonctionne pas avec `babel` mais simplement avec `polyglossia`.

## Imprimer sa bibliographie

- commande `\printbibliography`
- l'option `\nocite{#1}` permet d'imprimer dans la bibliographie des entrées non citées dans le corps du texte. L'argument peut-être soit un astérisque, pour indiquer que toutes les entrées du fichier `.bib` doivent être imprimées, soit une liste de clefs séparées par des virgules: `\nocite{*, \nocite{clef1, clef2, clef3, ...}`
- Il est possible de n'imprimer que des entrées ayant, ou n'ayant pas, un mot-clef: options `keyword=xx` et `notkeyword=xx`; on peut de cette façon imprimer plusieurs sous-parties de bibliographie.

Exemple:

```
\printbibliography[keyword=sources-primaires]
\printbibliography[notkeyword=sources-primaires]
```

D'autres choix sont possible (imprimer des bibliographies intermédiaires par exemple). Voir le manuel p. 90 et p. 140 - Pour modifier le titre de la

ou des bibliographies: + dans le préambule, `\defbibheading{#1}{#2}`, où #1=une clef attribuée au titre et #2 le titre + dans l’argument optionnel de la commande `\printbibliography`, `heading=clef`

Exemple:

```
\documentclass[a4paper]{book}
\usepackage{fontspec}
\usepackage{xunicode}
\usepackage{polyglossia}
\setmainlanguage{french}
\usepackage{biblatex}
\addbibresource{/home/user/Documents/memoire/bibliographie.bib}

\defbibheading{primaires}{\subsection*{Sources primaires}}
\defbibheading{secondaires}{\subsection*{Sources secondaires}}

\begin{document}
...
\section*{Bibliographie}
\printbibliography[heading=primaires, keyword=sources-primaires]
\printbibliography[heading=secondaires, notkeyword=sources-primaires]
\end{document}
```

## Zotero et LaTeX

### Importer une bibliographie BibTex vers Zotero:

- Fichier-Importer :
  - les clefs de citation restent telles quelles.
  - les *keywords* sont importées dans “marqueurs”
- Remarques:
  - certains champs bibtex n’existent pas dans zotero. Ex: *maintitle*, *volumes*, *gender*, *venue*. Ils sont importés dans *extra* sous la forme: `tex.maintitle: XXX`.
  - Le formatage au sein d’un champ (des italiques par exemple) est possible au moyen de balises html. Ainsi `\emph{xx}` devient

*xx*. Pour les autres possibilités de formatage, voir la Documentation Zotero.

-> on peut exploiter ces possibilités pour donner plus de souplesse à Zotero

### Le module BetterBibTex

- Le module BetterBibTex permet de répercuter dans un fichier BibTex les modifications apportées dans une collection (ou un ensemble de collections) gérée avec Zotero
- l'inverse n'est pas vrai : les modifications apportées dans le fichier BibTeX ne sont pas "remontées" dans Zotero.

Installation:

1. télécharger le XPI file et le sauvegarder. **attention: sous firefox, faire clic droit et enregistrer la cible du lien** (un simple clic sur le lien ne marchera pas)
2. Dans zotero: extension - ajouter depuis un fichier - installer - redémarrer zotero
3. Configurer Betterbibtex pour l'exportation: outils - betterbibtex - ouvrir les preference de betterbitex:
  - onglet **exportation**:
    - décocher "exporter les caractères unicodes"
    - choisir "ajouter les url à l'exportation dans le champs url"
    - onglet divers: décocher "appliquer la capitalisation aux titres"
  - onglet **clef de citation**: possibilité de choisir la façon dont les clefs seront générées
4. Mettre en place la synchronisation entre un fichier BibTex et une collection dans Zotero: faire un clic-droit sur la collection à exporter: exporter - choisir betterbibtex - cocher "garder à jour"

**RQ** Quand vous "aspirez" une référence bibliographique dans Zotero, pensez à vérifier que les informations sont bien entrées; vous pouvez utiliser le champs "extra" pour rajouter des champs propres à bibtex (voir supra)

# Les éléments non textuels

## Les flottants

- Lorsque l'on insère un tableau, une image ou un graphique, ils apparaissent exactement là où ils sont appelés, ce qui peut poser des problèmes de mise en page. De plus, ils ne comportent pas de légende.
- Il faut donc les transformer en flottant.
- *Flottant*: élément non textuel dont l'insertion dans la page va être calculée de façon à maintenir une mise en page correcte.
- Deux environnements pour faire des flottants: **figure** et **table**
- Au sein de ces environnements, la commande `\caption{#1}` permet d'insérer une légende (**nb**: on peut lui passer un argument optionnel: ce qui apparaîtra dans la liste des figures ou des tables)
- Insérer la liste des figures: `\listoffigures`; insérer la liste des tableaux: `\listoftable`
- Pour préciser où l'on souhaite insérer le flottant, on peut mettre un argument optionnel à ces environnements. Valeurs possibles:

---

valeur	effet
h	positionne le flottant à peu près à l'emplacement de son appel
t	positionne le flottant en haut d'une page
b	positionne le flottant en bas d'une page
p	positionne le flottant sur une page dédiée aux flottants

---

- Les valeurs peuvent être combinées. Je conseille d'utiliser **htb**
- **nb** Pour éviter que le flottant n'aille trop loin de l'endroit où on l'a appelé, utiliser la commande `\FloatBarrier` du package **placeins**: tous les flottants appelés avant la commande sont placés avant celle-ci.
- Le package **wrapfig** permet de mettre du texte autour du flottant (voir le manuel)

## Insérer une image

- Pour insérer une image, il faut utiliser la commande `\includegraphics[options]{chemin de l'image}` du package `graphicx`. Pour régler la taille de l'image, indiquer dans l'argument optionnel `scale=xx` où `xx` est une valeur numérale (au-dessus de 1: l'image est agrandie; en dessous de 1, elle est diminuée). Pour les autres options, voir le manuel.
- L'environnement `landscape` du package `lscape` permet de mettre l'image (ou tout autre élément: tableau, texte) en format paysage.
- le package `adjustbox` permet d'ajuster l'image (ou tout autre flottant) à la taille de la page; voir le manuel

## Insérer tableau

- Environnement `tabular` Utiliser l'assistant de Texstudio, très efficace. Syntaxe:
  - l'argument de `tabular` permet d'indiquer le nombre de colonnes et la position du texte (centré: `c`, aligné à gauche: `l`, aligné à droite `r`). Le signe `|` indique que les colonnes doivent être séparées par une bande verticale
  - `hline` permet de tracer des traits horizontaux entre les lignes
  - Au sein d'une ligne: on indique un changement de colonne par le signe `&`, et la fin de la ligne par `\\`
  - Il est possible également d'indiquer la taille des colonnes. Voir par exemple le tutoriel ici:  
`tabular`, ou utiliser simplement l'assistant de Texstudio.
- **nb** Il existe un package `longtable` permettant de faire des tableaux allant sur plusieurs pages et pouvant recevoir des notes de bas de page (`\footnote`).
- Le package `tabbing` permet d'aligner du texte en colonne sans réellement faire de tableau. Il est parfois plus adapté que l'environnement `tabular` ; vous trouverez un tutoriel ici: `tabbing`

## Faire un graphique avec TikZ

- Utiliser l'environnement `tikzpicture` du package `tikz`.
- `tikz` a une syntaxe spécifique. Notamment, chaque commande doit être terminée par un `;`.



- Le manuel en français de Jean-Pierre Franc, TikZ, dessiner avec LaTeX, est très clair

## Faire des histogrammes avec TikZ et le package pgfplots

- Un tutoriel ici: `pgfplots`. **Attention**, l’auteur donne à chaque fois également les commandes pour ceux qui utilisent ConTeXt: ne pas en tenir compte
- Dans l’environnement `tikzpicture`, utiliser l’environnement `axis` pour créer des abscisses et des ordonnées.
- Dans l’environnement `axis`, la commande `\addplot coordinates {#1}`; permet d’ajouter des points (en `#1`):
  - chaque point est indiqué entre parenthèses par : **abscisse-virgule-ordonnée**.
  - **Attention** pour les valeurs décimales, utiliser non une virgule mais un point (notation anglo-saxonne)
- Il peut y avoir plusieurs commandes `addplots` à la suite.
- Pour indiquer la légende: `\legend{#1}`. S’il y a plusieurs ensembles de données (plusieurs `\addplot`), il doit y avoir plusieurs éléments dans la légende, séparés par une virgule. Exemple: `\legend{légende pour le premier \addplot,légende pour le second \addplot}`
- Pour faire un diagramme en barres verticales ou horizontales: passer à l’environnement `axis` l’option `ybar` et `xbar`; `ybar stacked` pour que les barres soient les unes sur les autres
- l’option `xtick=data` permet de régler les “labels”: de ne faire apparaître sur la ligne des abscisses que les données utilisées.
- Pour que les nombres supérieurs à 999 ne soient pas notés avec une virgule (notation anglo-saxonne), insérer dans le préambule la commande `\pgfplotsset{/pgf/number format/1000 sep=}`

Pour des exemples, voir les exercices.

- **nb** Pour faire un diagramme camembert, utiliser le package `pgf-pie`.

## Importer des données d’un fichier . csv pour obtenir tableaux et graphes avec TikZ, pgfplots et le package pgfplotstable

- Pour des exemples, voir les exercices.

### Étape préliminaire: stocker les données dans une variable

- `\pgfplotstableread[col sep=XX]{#1}{#2}`, où `XX` peut prendre comme valeur `comma` ou `tab` selon le fichier csv que l'on importe (s'il y a des virgules dans les données importées, utiliser `tab`); `#1`=chemin du fichier (avec l'extension); `#2`=nom donné à une variable ainsi créée et qui va stocker les données du fichier.
- Exemple: `\pgfplotstableread[col sep=comma]{mon_fichier.csv}{\table}`
- Mettre dans le préambule `\pgfplotsset{/pgf/number format/read comma as period}` pour que les virgules dans les nombres décimaux soient bien comprises comme telles (notation française *vs* notation anglo-saxonne)

### Importer les données dans un tableau

- `\pgfplotstabletypesetfile{#1}`: lit le contenu de la variable précédemment créée et indiquée en `#1` et le transforme en tableau. Exemple: `\pgfplotstabletypesetfile{\table}`
- Pour modifier apparence des tableaux: passer des arguments optionnels à `\pgfplotstabletypesetfile`, ou régler l'apparence de tous les tableaux dans le préambule avec la commande `\pgfplotstableset`. Pour des exemples, voir les exercices et le manuel de `pgfplotstable`
- **Quand les données ne sont pas des données numériques mais des chaînes de caractères**: passer à `\pgfplotstabletypesetfile` l'option `string type`.

### Importer les données dans un diagramme

- reprendre l'environnement `axis` au sein de `tikzpicture`
- avec la commande `\addplot`, à la place de `coordinates`, mettre: `table[option]{#1}`, où `#1`=le nom de la variable où l'on a stocké les données.
- Pour indiquer à `pgfplotstable` quelles colonnes du fichier utiliser, indiquer en option: `x=nom de la colonne`, `y=nom de la colonne`.

- Quand les données ne sont pas des données numériques mais des chaînes de caractères:
  1. Passer comme option à `\addplot table: x expr=\coordindex`
  2. Indiquer que les données (non numériques) d'une colonne doivent apparaître comme label sur la ligne des abscisses: passer comme option à `axis: xticklabels from table={#1}{#2}`, où #1=le nom de la variable où l'on a stocké les données, et #2=le nom de la colonne utilisée.
  3. Pour faire tourner sur la ligne des abscisses les "labels" pour qu'ils apparaissent à la verticale, passer en option à `axis: x tick label style={rotate=90}`

## Les packages `reledmac` et `reledpar` pour l'édition critique

### `reledmac`

1. Pour numéroté les lignes:

```
\beginnumbering
\pstart
Le texte à numéroté
\pend
\endnumbering
```

- Pour que chaque paragraphe soit automatiquement mis entre les balises `\pstart` et `\pend`, il faut mettre `\autopar` au début du texte et terminer le texte par une ligne blanche ou la commande `\par`:

```
\beginnumbering
\autopar
premier paragraphe

deuxième paragraphe

etc

\endnumbering
```

- Pour configurer la numérotation, voir le manuel; idem pour la syntaxe des vers. Voir aussi le chapitre 20 de l'ouvrage XeLaTeX appliqué aux sciences humaines
- Dans le cas de vers, nous aurons:

```
\beginnumbering
\stanza
un vers.&
le dernier vers de la strophe\&
\endnumbering
```

## 2. Apparat critique:

- les “familiar notes”: `\footnoteA{#1}`, `\footnoteB{#1}`, etc (cinq niveaux, jusqu’) `\footnoteE`
- les “critical notes” (sans appel de note, mais avec, dans la note, le numéro de ligne ou de vers et un lemme, c’est-à-dire le morceau de texte pour lequel on indique des variantes): `:\Afootnote{#1}`, `\Bfootnote{#1}`, etc (cinq niveaux, jusqu’) `\Efootnote`
- les notes de fin (`\Aendnote{#1}`, etc, jusqu’à `\Eendnote`)
- Pour faire une note critique (avec lemme): `\edtext{#1}{#2}`, où `#1`= le mot qui va apparaître comme lemme, et `#2`= la “critical note”.

Pour des exemples, voir les exercices.

## reledpar

- à charger avec `reledmac`. Sert à mettre deux textes en vis-à-vis. Syntaxe:

```
\begin{pages} %Mettre en vis-à-vis sur deux pages
\begin{Leftside}
Texte de gauche
\end{Leftside}

\begin{Rightside}
Texte de droite
```

```

\end{Rightside}
\end{pages}
\Pages %imprimer le vis-à-vis

```

- Pour avoir un vis-à-vis en colonnes, remplacer l’environnement `pages` par `pairs` et la commande `\Pages` par `\Columns`
- Le package `reledpar` fait correspondre en vis-à-vis chaque “boîte” d’un texte à la boîte correspondante de l’autre texte : il faut donc qu’il y ait le même nombre de boîtes des deux côtés.
- Pour la poésie, chaque vers est une boîte
- en prose, les boîtes sont délimitées par `\pstart` et `\pend`.

## Naviguer dans un long document

### Les renvois internes

- Package `hyperref` (rappel: à charger après tous les packages, sauf exception, comme par exemple le package `glossaries`)
- Poser un label: `\label{nom_du_label}`
- Renvoyer au label au cours du texte (dans le corps du texte ou au sein d’un flottant):
  - `\pageref{nom_du_label}`: renvoie à la page où se trouve le label
  - `\ref{nom_du_label}`: renvoie au numéro de section/de flottant où se trouve le label
  - `\nameref{nom_du_label}`: renvoie au titre de la section où se trouve le label
- **Attention:** dans un flottant, le label doit être placé *après* `\caption`
- Rq: le package `hyperref` crée également automatiquement des signets dans le pdf, qui suivent la table des matières.

### Les index

#### Création d’un index simple

- Dans le préambule:

```
\usepackage{imakeidx}
\makeindex
```

- Indexer un mot dans le corps du texte: `\index{mot}`
- **Attention:** l’argument de la commande `\index{}` n’apparaît *pas* dans le texte. Il faut donc taper, par exemple, “Auguste”
- Imprimer l’index: `\printindex`. (il faut trois compilations)
- Les mots commençant par des caractères accentués sont mis en fin d’index. Pour résoudre les problèmes de tri avec les accents, indexer de cette façon: `\index{mot utilisé pour le tri dans l'index @ mot qui apparaîtra dans l'index}`
- Exemple: `\index{Elisabeth@Élisabeth}` permettra de faire apparaître “Élisabeth” dans l’index à la lettre E
- Il est possible de faire des subdivisions d’entrées: `\index{Empereur!Auguste}`, `\index{Empereur!Caligula}`
- Faire des références croisées (qui apparaîtront dans l’index sous la forme: “xx, voir yy”): `Bonaparte\index{Bonaparte|see{Napoléon}}\index{Napoléon}`.
- Faire apparaître l’index dans la table des matières: mettre l’option `intoc` à la commande `\makeindex`: `\makeindex[intoc]`
- **rq:** la création d’index génère des fichiers auxiliaires `.idx`, `.ilg` et `.ind`

### Créer plusieurs index

- Pour chaque index que l’on veut créer, mettre en préambule une commande `\makeindex`, avec dans l’argument optionnel `title` “nom de l’index” et `name` “clef de l’index”
- Exemple:

```
\usepackage{imakeidx}
\makeindex[intoc, title=Index général]
\makeindex[intoc, name=aut, title=Index des auteurs]
```

- Pour indexer une entrée dans un index en particulier, il faut mettre en argument optionnel à l’entrée `\index` la clef correspondante. Exemple: `\index{Entrée}` sera indexé dans l’index général, tandis que

`\index[aut]{entrée}` sera indexé dans l'index des auteurs.

- pour imprimer un index précis, ajouter en argument optionnel à `\printindex` la clef correspondante. Exemple: `\printindex[aut]`
- Pour indexer des auteurs cités avec biblatex, voir XeLaTeX appliqué aux sciences humaines p.149

## Les glossaires

```
\usepackage[toc=true]{glossaries}%doit être appelé après hyperref
%(exception à la règle)
\makeglossaries
```

- Définir les entrées de glossaire dans le préambule ou dans un fichier à part, avec la commande `\newglossaryentry{#1}{#2}` où #1 = la clef de l'entrée, et #2 contient le mot et sa définition. Exemples:

```
\newglossaryentry{ex}{%
  name={exemple},%
  description={Illustration%
    d'un concept}}
```

```
\newglossaryentry{co}{%
  name={concept},%
  description={Idée générale, représentation mentale
et abstraite que l'on a d'un objet}}
```

- on appelle dans le corps du texte les mots définis dans le glossaire au moyen de la commande `\gls{entree_du_glossaire}`. Exemple: `\gls{ex}`
- pour compiler le glossaire, si l'on a en plus un index, il faut lancer de le terminal la commande `makeglossaries nomdufichier`. Texstudio le fait avec la commande: `outil-glossaire (F9)`
- Pour imprimer le glossaire: `\printglossary`. On peut modifier le titre en argument optionnel: `\printglossary[title=le nouveau titre]`
- Pour faire apparaître le glossaire dans la table des matières, on passe une option à l'appel de package `\usepackage[toc=true]{glossaries}`
- on peut créer un glossaire dans un fichier à part et l'appeler dans le préambule par `\loadglsentries{fichier.tex}`

## Modifier les césures automatiques

- Il est possible d'indiquer des règles d'hyphénation (de césure) pour des mots précis, par langue:

```
\begin{hyphenrules}{french}  
  \hyphenation{  
\end{hyphenrules}
```

- Dans la commande `\hyphenation{#1}`, l'argument comprend une liste de mots séparés par un espace. Chaque mot comporte un ou plusieurs tirets indiquant les endroits possibles pour une césure. S'il n'y a pas de tiret, le mot est insécable.
- Exemple: `\hyphenation{Orléans inter-li-né-aire}`

## Créer un compteur

- Créer un nouveau compteur: `\newcounter{#1}` où `#1`= le nom du compteur. Exemple `\newcounter{ex}`. Par défaut, ce compteur est mis à zéro
- Mettre une autre valeur au compteur: `\setcounter{#1}{#2}` où `#1`= le nom du compteur et `#2`=une valeur. Ex: `\setcounter{ex}{5}`: le compteur "ex" est mis sur 5
- incrémenter le compteur: `\addtocounter{#1}{#2}` où `#1`= le nom du compteur et `#2`=une valeur. Exemple; `\addtocounter{ex}{1}`: on ajoute 1 au compteur "ex".
- Afficher la valeur actuelle du compteur: `\theNom_du_compteur`. Exemple; `\theex` affichera la valeur du compteur `ex`
- autres façon d'afficher la valeur du compteur:
  - `\roman{ex}` affichera la valeur du compteur "ex" en nombre romains minuscules
  - `\Roman{ex}` affichera la valeur du compteur "ex" en nombre romains majuscules
  - `\alph{ex}` affichera la valeur du compteur "ex" en lettres minuscules
  - `\Alph{ex}` affichera la valeur du compteur "ex" en lettres majuscules
- Exemple d'utilisation de compteur dans un nouvel environnement:

```
\newcounter{ex}%création du compteur "ex"
```



```
\newenvironment{exemple}%
{\addtocounter{ex}{1}\textbf{Exemple \theex}}%à l'ouverture de l'environnement,
%on ajoute 1 au compteur "ex",
% et on fait apparaître en gras le numéro de l'exemple avec \theex
{}}%il ne se passe rien à la fermeture de l'environnement
```

ce qui donnera l'utilisation suivante, où les exemples seront numérotés:

```
\begin{exemple}
Un premier exemple
\end{exemple}
```

```
\begin{exemple}
Un second exemple
\end{exemple}
```

## Utiliser un document maître

- Créer un document (ou utiliser celui fourni) contenant le préambule, les balises `\begin{document}` et `\end{document}`, et au sein de ces balises les grandes divisions (`\frontmatter` et `\appendix` si besoin, `\mainmatter` pour le corps du texte et `\backmatter` pour la bibliographie, le(s) index et table(s) des figures ou des tableaux, la table des matières
- Créer un document `.tex` par structure logique (chapitre ou section, selon la taille du travail).
  - ces documents “enfants” ne **doivent pas** avoir de préambule, puisque le préambule est contenu dans le document maître
  - appeler dans le document “maître” ces documents “enfants” avec la commande `\input{}` en indiquant le chemin relatif (par exemple: `\input{./frontmatter/section1.tex}` où le point indique l'emplacement du document maître et chaque / le passage à un autre dossier-enfant)
- Compiler le document maître. **nb** Avec TexStudio, il est possible de déclarer un document comme document maître: Option - Document maître - déclarer le document en cours comme document maître explicite. Cela permet ensuite de compiler directement depuis les fichiers enfants (du moment qu'ils sont bien appelés dans le document maître)

- Autre possibilité: garder l’option “détecter automatiquement le document maître”. On pourra compiler depuis n’importe quel fichier appelé dans le document maître, du moment que celui-ci a aussi été ouvert une première fois

## “Trucs” et ressources utiles

### Citer du code

- package `minted`
- prérequis: avoir installé Python ainsi que Pygments
- Attention: la compilation doit se faire avec l’option `-shell-escape` passée à la commande `xelatex`: il est possible de configurer TexStudio pour cela.
- utilisation: `\begin{minted}{nom du langage} ... \end{minted}`
- pour connaître la liste des langages possibles, taper dans le terminal `pygmentize -L lexers`
- attention, les tabulation ou les espaces surnuméraires apparaissent dans la coloration, penser à les enlever
- il est possible en argument à l’environnement d’ajouter les options `linenos` (numéroter les lignes) et `firstnumber=nombre` pour faire commencer la numérotation au nombre indiqué.

Exemple:

```
\begin{minted}[linenos, firstnumber=3]{python}
def f(x):
    return x**2
\end{minted}
```

### Utiliser plusieurs polices de caractères

- La package `fontspec` permet d’utiliser plusieurs police de caractères unicode.
- Pour indiquer la police de caractères principale: `\setmainfont{nom de la police}`. Exemple: `\setmainfont{LibertinusSerif-Regular}`
- il est possible d’indiquer une police pour les caractères sans empattement avec la commande `\setsansfont`, etc.

- On peut définir une nouvelle “famille” de caractère qui sera utilisée pour une destination précise (par exemple: pour le grec, pour l’araméen, etc), avec la commande `\newfontfamily\nomdelafamille{nom de la police}`. Il faut ensuite créer une commande ou un environnement qui utilisera cette police. Par exemple:

```
\newfontfamily\Smielyfnt{Humeur}
\newcommand{\smiley}[1]{\Smielyfnt #1}
```

Ceci permettra d’utiliser la police `Humeur` dans la commande `smiley`.

**nb:** l’appel à une famille de caractères (dans la définition de notre nouvelle commande) se fait au moyen d’une “commande à bascule”.

## Quelques *packages*

Vous pouvez chercher des packages adaptés à vos besoins sur le site du CTAN (The Comprehensive TeX Archive Network). Je vous propose ici une liste (non-exhaustive) de quelques packages qui pourraient vous servir.

Package	Usage
<code>adjustbox</code>	Adapter au mieux la taille d’un flottant
<code>array</code>	améliorer les tableaux
<code>biblatex</code>	Bibliographie
<code>biblatex-manuscripts-philology</code>	Décrire dans une base bibliographique des manuscrits
<code>biblatex-multiple-dm</code>	Pouvoir utiliser plusieurs modèles de données avec <code>biblatex</code>
<code>bibleref-french</code>	Références bibliques (français)
<code>csquotes</code>	Gérer les guillemets
<code>endnote</code>	Créer des notes de fin
<code>enumitem</code>	Modifier l’apparence des listes
<code>etoolbox</code>	Commandes LaTeX remplaçant des commandes TeX (difficulté +++ ; pour ceux qui aiment programmer!). Comporte des commandes pour des instructions de type <code>if... then.. else</code>
<code>fancyhdr</code>	Personnaliser la mise en page

Package	Usage
<code>framed</code>	Encadrer, surligner,... du texte
<code>geometry</code>	Modifier la géométrie de la page (marges)
<code>graphicx</code>	Insérer des images, manipuler des boîtes (rotation,...)
<code>hyperref</code>	Hyperliens. (Appel du package à mettre à la fin; amélioré si utilisé avec <code>bookmark</code> )
<code>ifthen</code>	Commandes pour écrire une instruction if then else. Moins pratique que les commandes du package <code>etoolbox</code> , mais plus simple à comprendre
<code>indextools</code>	Faire un ou plusieurs index (fork de <code>imakeidx</code> )
<code>lettrines</code>	Insérer des lettrines
<code>lscape</code>	Pour inclure des pages en format paysage
<code>pdflscape</code>	Inclure des pdf en format paysage
<code>perpage</code>	Faire redémarrer un compteur à chaque page
<code>pgf-pie</code>	Faire un diagramme “camembert”
<code>minted</code>	Colorer des citations de code informatique (attention, nécessite une compilation en ligne de commande avec une option: voir le manuel, section 3.1)
<code>multicol</code>	Mettre un texte sur plusieurs colonnes
<code>pict2e</code>	Tracer des dessins
<code>placeins</code>	Mettre une barrière aux flottants
<code>reledmac</code>	Éditions critiques
<code>reledpar</code>	Vis-à-vis (avec <code>reledmac</code> )
<code>setspace</code>	Modifier l’interligne
<code>tikz</code>	Faire des schémas
<code>verse</code>	Citer des vers

Package	Usage
<code>wrapfig</code>	Mettre du texte autour de flottants
<code>xargs</code>	Créer commandes et environnements avec plusieurs arguments optionnels
<code>xcolor</code>	Mettre de la couleur
<code>xspace</code>	Insérer un espace sauf avant certains signes de ponctuation

## Quelques messages d'erreurs courants

- `!Emergency stop` suivi de `(job aborted, no legal \end found:` il manque `\end{document}`
- `! LaTeX Error: Missing \begin{document}:` il y a du texte dans le préambule (en dehors des définitions et des appels de package)
- `!Undefined control sequence:` la commande n'existe pas - son nom est mal tapé, ou le package qui la définit n'a pas été chargé
- `LaTeX Error: \begin{<nom>} on input line 8 ended by \end{document}:` l'environnement n'a pas été fermé
- `! LaTeX Error: \begin{<nom1>} on input line 9 ended by \end{<nom2>}:` deux environnements se superposent, ou un environnement est fermé par la balise d'un autre environnement
- `! File ended while scanning use of \<nom de la commande>:` on a oublié l'accolade fermant la commande .
- `! LaTeX Error: Command \<nom> already defined:` on a essayé de définir une nouvelle commande ou un nouvel environnement en lui donnant le nom d'une commande ou d'un environnement existants.
- `! File ended while scanning use of \@argdef:` une accolade n'est pas fermée dans la définition d'une nouvelle commande
- `! File ended while scanning use of \@newenv:` une accolade n'est pas fermée dans la définition d'un nouvel environnement
- `! Missing number, treated as zero:` on a appelé un argument dans la définition d'une nouvelle commande ou d'un nouvel environnement, sans indiquer en option qu'il y aurait un argument
- `! Illegal parameter number in definition of \<nom>:` on appelle, dans la définition d'une nouvelle commande ou d'un nouvel environnement, un argument d'un nombre plus élevé que le nombre d'argument indiqué en option

- **Missing \$ inserted:** on a tapé une commande mathématique sans la mettre entre les symboles \$.
- **Nested note:** une `footnote` dans une autre.

## **(Très courte) bibliographie**

ROUQUETTE, Maïeul, XeLaTeX appliqué aux sciences humaines

FRANC, Jean-Pierre, TikZ, dessiner avec LaTeX,

KARNAJ, Zeste de savoir. Présenter du code source dans un document LaTeX

LOZANO, Vincent, Tout ce que vous avez toujours voulu savoir sur LaTeX sans jamais oser le demander

OETIKER, Tobias, Une courte (?) introduction à LaTeX

TISSEAU, Gérard et DUMAS, Jacque, TikZ pour l' impatient

## **Autres ressources**

Geekographie maïeulesque

Les fiches à Bébert

## **Trouver de l'aide**

### **Forums francophones**

Developpez.com

TeXnique

### **Forums anglophones**

Latex Community

StackExchange