

Contents

1	Introduction	2
1.1	Qu'est-ce que \LaTeX ?	2
1.2	Pourquoi utiliser \LuaTeX ?	2
1.3	Comment utiliser \LaTeX ?	2
2	Installation des outils	3
2.1	Installation de \LaTeX	3
2.1.1	Sous Windows	3
2.1.2	Sous macOS	3
2.1.3	Sous Linux	3
2.2	Installation d'un éditeur de texte	4
2.3	Installation de git	4
2.4	Installation de \LuaTeX	4
3	Utilisation de \LuaTeX	5
3.1	Création d'un document	5
3.1.1	Explication de la structure du document	6
3.2	Compilation du document	7

1 Introduction

1.1 Qu'est-ce que \LaTeX ?

\LaTeX est un système de composition de documents créé par Leslie Lamport en 1983, basé sur le logiciel \TeX créé par Donald Knuth en 1978. \LaTeX est un logiciel libre disponible sur la plupart des systèmes d'exploitation.

Il est particulièrement adapté à la rédaction de documents scientifiques, techniques, ou académiques, mais peut être utilisé pour tout type de document. Il est particulièrement apprécié pour la qualité typographique de ses documents et pour sa capacité à gérer de très longs documents.

1.2 Pourquoi utiliser \LuaTeX ?

\LuaTeX est une version de \LaTeX qui permet d'utiliser le langage de programmation Lua pour étendre les fonctionnalités de \LaTeX . \LuaTeX est particulièrement adapté pour la création de documents multilingues ou pour la création de documents nécessitant des calculs ou des traitements de données.

1.3 Comment utiliser \LaTeX ?

Si vous voulez éviter de prendre du temps pour installer les outils nécessaires à l'utilisation de \LaTeX , vous pouvez utiliser un éditeur en ligne. Il en existe plusieurs, mais Overleaf est l'un des plus populaires. Overleaf est un éditeur en ligne qui permet de créer des documents \LaTeX sans avoir à installer quoi que ce soit sur votre ordinateur. Il est particulièrement adapté pour les débutants car il propose de nombreux modèles de documents et permet de compiler les documents en temps réel.

Dans ce document, nous allons installer ensemble les outils nécessaires directement sur votre ordinateur.

2 Installation des outils

2.1 Installation de \LaTeX

L'installation de \LaTeX est relativement simple, mais peut prendre un peu de temps et dépend de votre système d'exploitation.

2.1.1 Sous Windows

Sous Windows, le moyen le plus simple d'installer \LaTeX est d'utiliser MikTeX. MikTeX est une distribution \LaTeX pour Windows, qui inclut un gestionnaire de paquets pour installer facilement des extensions.

Pour installer MikTeX, rendez-vous sur le site officiel de MikTeX à l'adresse suivante : <https://miktex.org/download>.

Téléchargez le programme d'installation, et exécutez-le. Suivez les instructions à l'écran pour installer MikTeX.

2.1.2 Sous macOS

Sous macOS, le moyen le plus simple d'installer \LaTeX est d'utiliser MacTeX. MacTeX est une distribution \LaTeX pour macOS, qui inclut un gestionnaire de paquets pour installer facilement des extensions.

Pour installer MacTeX, rendez-vous sur le site officiel de MacTeX à l'adresse suivante : <https://www.tug.org/mactex/>.

Téléchargez le programme d'installation, et exécutez-le. Suivez les instructions à l'écran pour installer MacTeX.

2.1.3 Sous Linux

Sous Linux, le moyen le plus simple d'installer \LaTeX est d'utiliser TeX Live. TeX Live est une distribution \LaTeX pour Linux, qui inclut un gestionnaire de paquets pour installer facilement des extensions.

Pour installer TeX Live, ouvrez un terminal et exécutez la commande suivante :

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

pour les distributions basées sur Debian, ou

```
sudo dnf install texlive-scheme-full
```

pour les distributions basées sur Fedora.

2.2 Installation d'un éditeur de texte

Pour écrire des documents \LaTeX , vous aurez besoin d'un éditeur de texte. Il existe de nombreux éditeurs de texte adaptés à \LaTeX , nous utiliserons dans ce document Visual Studio Code.

Visual Studio Code est un éditeur de texte gratuit, open-source, et multi-plateforme, développé par Microsoft.

Pour installer Visual Studio Code, rendez-vous sur le site officiel de Visual Studio Code à l'adresse suivante : <https://code.visualstudio.com/>. Téléchargez le programme d'installation, et exécutez-le. Suivez les instructions à l'écran pour installer Visual Studio Code.

2.3 Installation de git

Pour pouvoir enregistrer vos documents \LaTeX , je vous recommande d'utiliser git. Git est un logiciel de gestion de versions, qui permet de suivre l'évolution de vos documents, et de les partager avec d'autres personnes, ce qui est particulièrement utile pour travailler en équipe.

Pour ma part, j'utilise git à travers le module git intégré à Visual Studio Code. Puisque cela ne nécessite pas d'installation supplémentaire, je ne détaillerai pas ici l'installation de git, ni son utilisation. Si vous souhaitez en savoir plus sur git, je vous recommande de consulter la documentation officielle de git à l'adresse suivante : <https://git-scm.com/>.

2.4 Installation de Lua \LaTeX

Pour installer Lua \LaTeX , vous n'avez rien à faire de particulier, car Lua \LaTeX est inclus dans les distributions \LaTeX modernes, telles que Mik \TeX , Mac \TeX , ou TeX Live.

3 Utilisation de Lua \LaTeX

3.1 Création d'un document

Pour créer un document avec Lua \LaTeX , ouvrez un dossier vide dans votre éditeur de texte. La structure de votre dossier est à votre convenance, mais je vous recommande d'adopter une structure de dossier claire et organisée, comme celle-ci :

- `content/`
 - `chapter1.tex`
 - `chapter2.tex`
 - `chapter3.tex`
 - `main.tex`
- `img/`
 - `image1.png`
 - `image2.png`
 - `image3.png`
- `utils/`
 - `preamble.tex`

Commencez par créer un fichier `main.tex` dans le dossier racine de votre projet, et copiez le contenu suivant dans ce fichier :

```
\documentclass{scrreprt}

\input{./utils/preamble}

\begin{document}

\subfile{content/chapter1}

\end{document}
```

Ensuite, créez un fichier `preamble.tex` dans le dossier `utils/`, et copiez le contenu suivant dans ce fichier :

```

%%% Preamble

%%% Packages

%% Couleurs
\usepackage{xcolor}
% Définition de la couleur bleu foncé
\definecolor{myblue}{RGB}{0, 0, 139}

%% Liens et références
\usepackage{hyperref}
% Configuration des liens hypertexte
\hypersetup{
    colorlinks=true,
    linkcolor=myblue,
    urlcolor=myblue,
}

\usepackage{subfiles} % Best loaded last in the preamble

```

Enfin, créez un fichier `chapter1.tex` dans le dossier `content/`, et copiez le contenu suivant dans ce fichier :

```

\documentclass[../main.tex]{subfiles}

\begin{document}

\chapter{Chapitre 1}

Ceci est le premier chapitre de votre document.

\end{document}

```

3.1.1 Explication de la structure du document

Le fichier `main.tex` est le fichier principal de votre document. Il contient les commandes de base pour définir le type de document, charger les packages nécessaires, et inclure les différents chapitres de votre document. La première ligne de ce fichier est la commande `\documentclass{scrreprt}`, qui définit le type de document comme un rapport. Cela correspond à un document structuré en chapitres, ce qui est adapté pour la plupart des documents académiques ou techniques.

La deuxième ligne de ce fichier est la commande `\input{../utils/preamble}`, qui charge les packages nécessaires à la compilation de votre document. Par souci de clarté et d'organisation, j'ai choisi de stocker les différents packages nécessaires à la compilation

de votre document dans un fichier `preamble.tex`, que j'ai placé dans un dossier `utils/`. Vous pouvez bien sûr stocker ces packages directement dans le fichier `main.tex`, mais je vous recommande de les stocker dans un fichier séparé pour plus de clarté.

La troisième ligne de ce fichier est la commande `\begin{document}`, qui indique le début du contenu de votre document. La quatrième ligne de ce fichier est la commande `\subfile{content/chapter1}`, qui inclut le contenu du fichier `chapter1.tex` dans le document. La cinquième ligne de ce fichier est la commande `\end{document}`, qui indique la fin du contenu de votre document.

Le fichier `preamble.tex` contient les packages nécessaires à la compilation de votre document. Dans cet exemple, j'ai choisi de charger deux packages : le package `xcolor`, qui permet de définir des couleurs, et le package `hyperref`, qui permet de configurer les liens hypertexte. J'ai également chargé le package `subfiles`, qui permet d'inclure des fichiers `.tex` dans d'autres fichiers `.tex`.

Le fichier `chapter1.tex` est le premier chapitre de votre document. Il contient les commandes de base pour définir le type de document, charger les packages nécessaires, et inclure le contenu de ce chapitre dans le document. La première ligne de ce fichier est la commande `\documentclass[../main.tex]{subfiles}`, qui indique que ce fichier est un sous-fichier du fichier `main.tex`. La deuxième ligne de ce fichier est la commande `\begin{document}`, qui indique le début du contenu de ce chapitre. La troisième ligne de ce fichier est la commande `\chapter{Chapitre 1}`, qui indique le début du premier chapitre de votre document. La quatrième ligne de ce fichier est le contenu de ce chapitre. La cinquième ligne de ce fichier est la commande `\end{document}`, qui indique la fin du contenu de ce chapitre.

Note : Ceci n'est qu'un exemple minimaliste pour vous aider à démarrer. Vous pouvez bien sûr ajouter d'autres fichiers `.tex` pour créer un document plus complexe, et vous pouvez également ajouter d'autres packages dans le fichier `preamble.tex` pour étendre les fonctionnalités de votre document. Je vous recommande de consulter la documentation officielle de L^AT_EX et également à faire vos propres recherches pour en apprendre plus sur les différentes fonctionnalités de L^AT_EX. Vous pouvez également consulter l'intégralité de ce document sur <https://github.com/GuyGnole/lualatex-tuto-template>.

3.2 Compilation du document

Pour compiler votre document, ouvrez un terminal, et placez-vous dans le dossier racine de votre projet. Ensuite, exécutez la commande suivante :

```
lualatex ./content/main.tex
```

Ou utiliser le bouton de compilation de votre éditeur de texte si celui-ci le permet.