

# Préparer un rapport simplement

*md2pdf -r*

Février 2021

**Bruno BEAUFILS**  
John MAC FARLANE  
Leslie LAMPORT



Licence Professionnelle

**da2i**

Licence Professionnelle

**CGIR**



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Fonctionnement</b>	<b>7</b>
2.1	Guide rapide . . . . .	7
2.2	Outils de transformation . . . . .	7
2.3	Utiliser md2pdf pour faire un rapport . . . . .	7
<b>3</b>	<b>Une courte référence</b>	<b>9</b>
3.1	Markdown . . . . .	9
3.2	Extensions Pandoc à Markdown . . . . .	9
3.2.1	Images . . . . .	10
3.2.2	Code source . . . . .	10
3.2.3	Tableaux . . . . .	11
3.2.4	Fichiers séparés . . . . .	11
3.3	Modèles pandoc . . . . .	11
3.3.1	Titre . . . . .	12
3.3.2	Sommaire . . . . .	12
3.3.3	Annexes . . . . .	12
3.3.4	Code $\LaTeX$ . . . . .	12
3.4	Autres détails . . . . .	12
	<b>Annexes</b>	<b>15</b>
<b>A</b>	<b>Réglages</b>	<b>15</b>
A.1	makefile-pandoc . . . . .	15
<b>B</b>	<b>Templates</b>	<b>17</b>
B.1	pandoc-beamer.tex . . . . .	17
B.2	pandoc-report.tex . . . . .	19



# Chapitre 1

## Introduction

Ce court document permet de comprendre comment produire rapidement un rapport propre au format PDF à partir de fichiers au format Markdown. Son contenu est donc volontairement simple. Pour le rendre *crédible* en terme de volume, dans la fin de cette introduction pour les 2 paragraphes suivants, on fait appel à du [faux texte](#).

---

Nullam eu ante vel est convallis dignissim. Fusce suscipit, wisi nec facilisis facilisis, est dui fermentum leo, quis tempor ligula erat quis odio. Nunc porta vulputate tellus. Nunc rutrum turpis sed pede. Sed bibendum. Aliquam posuere. Nunc aliquet, augue nec adipiscing interdum, lacus tellus malesuada massa, quis varius mi purus non odio. Pellentesque condimentum, magna ut suscipit hendrerit, ipsum augue ornare nulla, non luctus diam neque sit amet urna. Curabitur vulputate vestibulum lorem. Fusce sagittis, libero non molestie mollis, magna orci ultrices dolor, at vulputate neque nulla lacinia eros. Sed id ligula quis est convallis tempor. Curabitur lacinia pulvinar nibh. Nam a sapien.

Aliquam erat volutpat. Nunc eleifend leo vitae magna. In id erat non orci commodo lobortis. Proin neque massa, cursus ut, gravida ut, lobortis eget, lacus. Sed diam. Praesent fermentum tempor tellus. Nullam tempus. Mauris ac felis vel velit tristique imperdiet. Donec at pede. Etiam vel neque nec dui dignissim bibendum. Vivamus id enim. Phasellus neque orci, porta a, aliquet quis, semper a, massa. Phasellus purus. Pellentesque tristique imperdiet tortor. Nam euismod tellus id erat.



# Chapitre 2

## Fonctionnement

### 2.1 Guide rapide

Ce rapport est produit à partir de `modele-rapport.md`. Il est conçu pour faire un rapport simple d'un projet ou d'un stage.

Il peut être utilisé comme modèle pour faire un rapport. Dans ce cas il faut :

1. créer un nouveau dossier
2. y copier, renommer puis **modifier** le fichier `modele-rapport.md`
  - copier également les images (dossier `img`) si nécessaire
3. créer la présentation via la commande

```
md2pdf -r nom-du-fichier.md
```

### 2.2 Outils de transformation

Pour produire un rapport au format PDF on utilise différents format et outils :

- [Markdown](#) est un langage de **balisage simple**. Il est conçu pour être facile à lire et écrire par les humains et les programmes. Il permet de transformer un fichier balisé vers HTML ou PDF *automatiquement*. Il a été initialement construit uniquement pour la conversion vers HTML. Il en existe aujourd'hui plusieurs [variantes](#).
- [L<sup>A</sup>T<sub>E</sub>X](#) est un outil de composition de texte de haute qualité typographique.
- [Pandoc](#) est un logiciel de conversion de documents. Il permet de convertir des fichiers d'un format de balisage à un autre. Pour la conversion vers PDF il utilise notamment [L<sup>A</sup>T<sub>E</sub>X](#).
- `md2pdf` est un simple script shell qui *enveloppe* la conversion d'un fichier Markdown vers un un fichier PDF (présentation ou rapport). Il paramètre `pandoc` pour personnaliser et faciliter la conversion.

### 2.3 Utiliser md2pdf pour faire un rapport

Comme pour tout document il faut d'abord s'atteler à **construire la structure (plan)** du rapport en plaçant dans un document les titres des différentes parties, chapitres et sections. Dans cette étape il faut penser à utiliser les six différents niveaux de titre offerts par Markdown.

Ensuite il faut remplir le contenu de chacune des sections. Il est possible de déporter le contenu dans des fichiers séparés. On peut par exemple créer un fichier par chapitre. Cela permet par exemple d'augmenter la lisibilité ou distribuer la rédaction entre plusieurs auteurs.

Dans cette étape on ne s'occupe pas de la mise en forme (`md2pdf` s'en occupe).

Dans toutes ces étapes il faut avoir un peu de discipline sur l'édition. Il faut par exemple :

- être attentif aux **indentations** des lignes
- espacer les sections (avec des lignes vides par exemple)
- préférer les images vectorielles (SVG) aux matricielles
- préférer le format PNG pour les images matricielles

Parmi les avantages on peut citer la construction **très rapide** de rapport, la réutilisation de documents déjà rédigés et le fait de ne pas avoir à se préoccuper de la forme qui est déporté vers un *thème*  $\text{\LaTeX}$ .

Parmi les inconvénients on peut notamment citer le fait que les constructions typographiques trop complexes sont parfois difficilement accessibles. Il faut s'en remettre à  $\text{\LaTeX}$  directement pour résoudre directement ce genre de problèmes. Il est également souvent difficile de déboguer la production sans maîtriser la chaîne `pandoc-pdflatex`. Dans un tel cas, transformer le fichier Markdown en  $\text{\LaTeX}$  peut aider. L'option `-L` de `md2pdf` permet cette conversion.



# Chapitre 3

## Une courte référence

Pour bien comprendre les transformation il est intéressant de lire le contenu du fichier source, `rapport-reference.md` et de regarder **en même temps** le résultat en PDF.

### 3.1 Markdown

Markdown est un format texte permettant de définir des documents. La syntaxe de Markdown permet

- de donner une **structure** au document
  - **paragraphes** : séparation par des lignes vides
  - **titres de différents niveaux** : lignes débutant par des #
  - **listes libres** : indentation de paragraphe avec lignes débutant par des – ou des \*
  - **listes ordonnées** : indentation avec lignes débutant par des 1 .
  - **blocs de citations** : paragraphes préfixés par “>”
  - **blocs de codes** : groupes de paragraphes entourés de lignes vides **et**
    - **indentés** par 4 espaces
    - ou **encadrés** par des lignes avec 3 accent graves (*backquotes*) ````` ou tildes `~~~`
- d’inclure des éléments de mise en forme logique  
*emphase*, **accentuation forte** et code
- d’inclure des liens vers des URL
  - <https://daringfireball.net/projects/markdown>
  - Pandoc Markdown
  - CommonMark
  - Markdown à la sauce GitLab
  - Markdown à la sauce Github
- d’inclure des images via des liens spécifiques : `![Texte alternatif](img/tex-friendly-zone.svg)`

Il existe beaucoup d’autres détails (cf liens présent dans la page) et de nombreuses **variantes** et extensions du format de base.

### 3.2 Extensions Pandoc à Markdown

**Pandoc** est un outil permettant de convertir un grand nombre de format de documents les uns vers les autres.

Par ailleurs il existe une extension **Markdown spécifique à Pandoc**. Sa philosophie est un peu étendue par rapport à celle de la version initiale de Markdown puisqu’elle ne se limite pas à convertir vers HTML. Elle est particulièrement adaptée à l’outil et aux différentes conversions qu’il est capable de faire.

Toutes les fonctionnalités standards de cette extension sont accessibles lors de la création de document via la commande `md2pdf` et peuvent donc être incluses dans les fichiers Markdown utilisés.

Un changement important est la possibilité de préciser la nature des éléments (titres, liens, images, blocs, etc.) par l’utilisation d’une zone ajoutée juste après un élément. Cette zone est entourée d’accolages (`{ . . . }`). Elle peut contenir

- la fixation de valeurs pour des attributs : `attribut=valeur`
- le rattachement à une classe : `.classe`

On trouve tous les détails dans la documentation de la **variante Pandoc de Markdown**.

En voici quelques autres exemples.

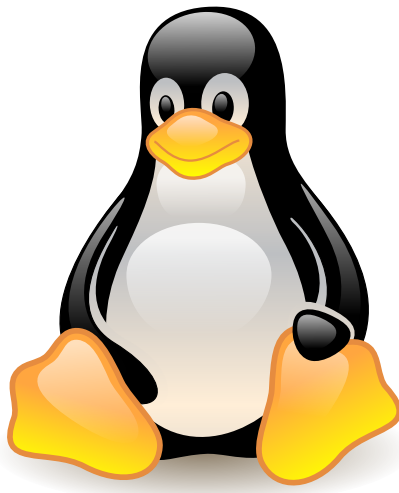


FIGURE 3.1 – Un joli pingouin

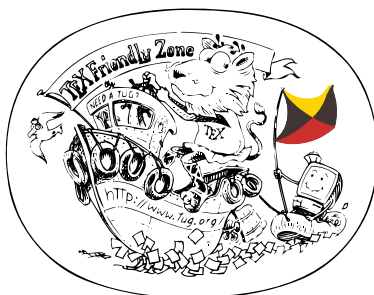
### 3.2.1 Images

On peut insérer des images dans le rapport.

Si elles sont seules dans leur paragraphe, elles seront traitées comme des figures et placées automatiquement à l'endroit le plus adéquat. Dans ce cas la légende de la figure est spécifié dans la zone texte du lien. C'est le cas du *joli pingouin* que l'on peut voir quelque part dans cette page ou la suivante.

Quand elle ne sont pas seules dans leur paragraphe (par exemple en ajoutant un blanc protégé juste avant ou juste après elles), elles sont placées exactement à l'endroit choisi sans être considérées comme des figures. C'est le cas de la *TeX Friendly Zone* ci dessous.

Grâce au filtre `center-image.lua` (activé par défaut par `md2pdf`), si elles ont la classe `.center`, elles sont, en plus, centrées sur la largeur de la page.



Par ailleurs, grâce au filtre `svg-image-to-pdf.lua` (activé par défaut par `md2pdf`), quand on inclut une image dont le nom se termine par `.svg`, `pandoc` remplace cette extension par `.pdf` de sorte que `pdflatex` puisse l'inclure correctement. Le fichier pdf doit donc exister, i.e. `md2pdf` ne se charge pas de la transformation du fichier lui même.

### 3.2.2 Code source

On peut insérer du code source en spécifiant le langage utilisé de façon à ce qu'il soit présenté avec une colorisation syntaxique adaptée.

```
#include <stdio.h>
int main(int argc, char ** argv) {
    printf("Hello world !\n");
}
```

Grâce au filtre `include-code-files` (activé par défaut par `md2pdf`), on peut insérer du code directement depuis un fichier : il suffit d'utiliser un bloc de code qui fixe l'attribut `include` au nom du fichier que l'on veut inclure, ajouter une classe correspondant au langage à utiliser plus d'autres classes optionnelles si on veut.

Par exemple le code suivant inclut le fichier `rapport-exemple-code` qui contient du code bash en numérotant les lignes.

```
```{include="md2pdf-exemple-code" .bash .numberLines}
```

Ce qui donne le résultat suivant.

```
1 #!/bin/bash
2 for a in "$@"
3 do
4     echo $a
5 done
```

### 3.2.3 Tableaux

On peut insérer des tableaux de plusieurs manières. Les tableaux sont considérés comme des tables au sens  $\LaTeX$ . Ils sont donc flottant et leur position final dans le PDF peut surprendre.

Outil	Utilité
pandoc	conversion de formats
pdflatex	composition de $\LaTeX$ en PDF
beamer	jeux de balisage $\LaTeX$ pour la composition de présentation

La section [tables du manuel de pandoc](#) liste et illustre les différentes manières de saisir des tableaux.

### 3.2.4 Fichiers séparés

Grâce au filtre [include-files](#) (activé par défaut par md2pdf), on peut insérer le contenu d'un fichier Markdown dans le fichier principal. Pour inclure le contenu d'un fichier il suffit d'utiliser un bloc de code qui a la classe `.include`. Chaque ligne du bloc de code contient alors un fichier à inclure.

```
```{.include}
FICHIER
```
```

FICHIER doit être un chemin vers le fichier à inclure.

Les entêtes du fichier inclus peuvent être décalées pour s'apparier correctement à la structure du document. On peut spécifier ce décalage en fixant une valeur au paramètre `shift-heading-level-by` entre accolades dans les attributs du bloc de code. On peut aussi laisser pandoc gérer automatiquement ce décalage en fixant `include-auto` à `true` dans un bloc YAML.

## 3.3 Modèles pandoc

md2pdf utilise un modèle pandoc ad-hoc ([template](#)) pour la transformation en  $\LaTeX$ . Pour les présentations c'est le fichier `pandoc-beamer.tex`, pour les rapports c'est le fichier `pandoc-report.tex`.

Ces modèles gèrent par exemple des variables pour personnaliser la page de titre, le sommaires ou des commandes  $\LaTeX$  pour gérer les annexes. Ces variables pandoc sont ajustables de plusieurs manières différentes.

La plus simple est d'utiliser un bloc [YAML](#) dans le source du document source lui même. Généralement on le place en début de fichier principal. Le bloc est en YAML et est donc encadré par des lignes ne contenant que 3 tirets (`---`) chacune.

Les autres possibilités sont de transmettre à pandoc les options fixant les variables via sa ligne de commande :

- via l'option `-P` passée de md2pdf qui lui passe son argument tel quel,
- via la variable d'environnement `$EXTRAS` dont le contenu est ajouté à la ligne d'appel de pandoc.

Par exemple pour fixer la variable `toc` à `false` on peut inclure le bloc suivant dans le fichier source.

```
---
toc: false
---
```

De la même manière on peut passer l'option `-P "-V toc=false"` (les guillemets sont importantes) à `md2pdf` ou bien fixer comme valeur `-V toc=false` à la variable d'environnement `$EXTRAS`.

### 3.3.1 Titre

Les paramètres suivants sont utilisés pour générer la page de titre.

- `title, subtitle, author, date`
- `titlegraphics` : images (type logos) sur la page de titre. Chaque logo doit contenir les variables
  - `file` : le nom du fichier sans extension
  - `width` : largeur en pourcentage (<1) de largeur diapo (0.1 par défaut)
  - `nl` : passage à la ligne après le logo ou pas
- `titlebackground` : nom du fichier image supplémentaire sur la page de titre (fond de page)

### 3.3.2 Sommaire

Les paramètres suivants sont pris en compte pour la génération des sommaires :

- `toc, toc-title` : générer ou pas la table des matières, changer le nom de la table des matières
- `toc-depth` : la profondeur de la table des matières
- `lot, lof` : générer ou pas la liste de tables, la liste des figures

### 3.3.3 Annexes

Pour démarrer une partie du document contenant les annexes il suffit de placer la commande `\appendixes` sur une ligne avant les sections qui contiennent chacune des annexes. Une page précisant le début des annexes est alors créée et la numérotation des sections suivantes est adaptée.

### 3.3.4 Code $\LaTeX$

On peut personnaliser le code  $\LaTeX$  créé par la transformation en utilisant les variables `include-before`, `include-after`. Elles peuvent contenir du code  $\LaTeX$  à ajouter en début ou fin de document.

## 3.4 Autres détails

`md2pdf` crée un répertoire `etc` lors de son exécution. On y retrouve les réglages qu'il utilise pour son appel à `pandoc`. Par exemple `etc/makefile-pandoc` contient les règles et paramétrages d'appel de `pandoc`, `etc/pandoc-report.tex` est le *template* utilisé pour la transformation, etc. Le nom de ce répertoire est modifiable via l'option `-C` de `md2pdf` ou la variable `$CONFDIR`. Si les fichiers existent déjà ils ne sont remplacés que si l'option `-O` est utilisée.

`md2pdf` ajoute les paramètres qu'il trouve dans la variable d'environnement `$EXTRAS` et ceux qu'il reçoit par le paramètre `-P` à la ligne de commande de `pandoc` qu'il utilise.

La documentation de `md2pdf` est accessible via l'option `-h` (version courte) ou l'option `--man` (version complète).

La documentation de `pandoc` est accessible dans [pandoc\(1\)](#) et se complète par le site web de [Pandoc](#).

## **Annexes**



# Annexe A

## Réglages

### A.1 makefile-pandoc

```
.PHONY: clean reset

CONFDIR ?= $(dir $(MAKEFILE_LIST))etc

beamer: $(FILE)
beamer: TEMPLATE ?= $(CONFDIR)/pandoc-beamer.tex
beamer: THEME ?= ulille
beamer: SLIDELEVEL ?= 2
beamer: ARGS += -t beamer
beamer: ARGS += --slide-level=$(SLIDELEVEL)

report: $(FILE)
report: TEMPLATE ?= $(CONFDIR)/pandoc-report.tex
report: THEME ?= ulille
report: ARGS += -t latex
report: ARGS += --top-level-division=chapter
report: ARGS += -N
report: ARGS += --toc

ARGS = --standalone
ARGS += --template=$(TEMPLATE)
ARGS += --pdf-engine-opt=-shell-escape
ARGS += -L $(CONFDIR)/include-files.lua
ARGS += -L $(CONFDIR)/include-code-files.lua
ARGS += -L $(CONFDIR)/center-image.lua
ARGS += -L $(CONFDIR)/svg-image-to-pdf.lua
ARGS += -V theme=$(THEME)
ARGS += -V themeoptions=$(THEMEOPTIONS)
ARGS += --highlight-style=tango

%.pdf %.tex: %.md $(MAKEFILE_LIST) $(wildcard $(CONFDIR)/*)
    TEXINPUTS=./:$(CONFDIR): pandoc $(ARGS) $(EXTRAS) -o $@ -- $<

%.pdf: %.svg
    inkscape -d 2400 -o $@ -T $<

clean:
    -rm -rf *~ _minted*

reset: clean
    -test -z $(FILE) || rm $(FILE)
```





# Annexe B

## Templates

### B.1 pandoc-beamer.tex

```
\documentclass[10pt,t]{beamer}

% Paquets LaTeX %%%%%%%%%%%%%%

%% Une gestion correcte du français (en entrée et en sortie)
\usepackage[french]{babel}
\usepackage{typelec} % devant fontenc (cf typelec.sty)
\usepackage[T1]{fontenc} % devant inputenc (utf8 choisi en fonction de ça)
\usepackage[utf8]{inputenc}
\DeclareUnicodeCharacter{20AC}{\euro} % pour la saisie du caractère euro

%% Des "jolies" polices de caractères
\usepackage{lmodern} % pour sf et tt
\usepackage{fourier} % pour rm
\usepackage{bbm} % pour les mathbbm

%% Plein de symboles
\usepackage{amssymb} % Les symboles mathématiques de l'AMS
\usepackage{latexsym} % Quelques symboles manquants dans LaTeX 2e
\usepackage{marvosym} % Quelques symboles en vrac par Martin Vogel
\usepackage{wasysym} % Quelques symboles en vrac par Roland Waldi
\usepackage{pifont} % Les symboles Dingbats
\usepackage{textcomp} % \textcopyright
\usepackage[copyright]{ccicons} % Les (c) comme dans Creative Commons
\usepackage[official,right]{eurosym} % L'euro

%% Quelques paquets utiles
\usepackage{array} % pour faciliter les styles de tableaux
\usepackage{longtable,booktabs} % pour les longues tables générées par pandoc
\usepackage{relsize} % pour faciliter le changement de taille des polices
\usepackage[normalem]{ulem} % pour avoir des soulignements funky
\usepackage{tikz} % pour les dessins portables
\usepackage{pgfpages} % pour les présentations en double-écran
\usepackage{fixltx2e} % provides \textsubscript
\usepackage{graphicx,grffile} % pour les images
\usepackage{fancyvrb}
\usepackage{minted} % pour les programmes

% Configuration pandoc %%%%%%%%%%%%%%

%% Images management
% Scale images if necessary, so that they will not overflow the page
% margins by default, and it is still possible to overwrite the defaults
```

```

% using explicit options in \includegraphics[width, height, ...]{}
$if(graphics)$
\makeatletter
\def\maxwidth{\ifdim\Gin@nat@width>\linewidth\linewidth\else\Gin@nat@width\fi}
\def\maxheight{\ifdim\Gin@nat@height>\textheight0.8\textheight\else\Gin@nat@height\fi}
\makeatother
\setkeys{Gin}{width=\maxwidth,height=\maxheight,keepaspectratio}
$endif$

%% Syntax highlighting
$if(highlighting-macros)$
$highlighting-macros$
$endif$

%% Generated lists
\providecommand{\tightlist}{%
  \setlength{\itemsep}{0pt}\setlength{\parskip}{0pt}}

%% Prevent overfull lines
\setlength{\emergencystretch}{3em}

% Paramétrages Beamer %%%%%%%%%%%%%%%

$if(theme)$\usetheme[$for(themeoptions)$$themeoptions$$$sep$, $endfor$]{$theme$}$endif$
$if(titlebackground)$
\makeatletter
\@ifundefined{titlebackground}
{
  \newcommand{\addtitlebackground}{%
    \begin{tikzpicture}[remember picture,overlay]
      \node[opacity=0.15,inner sep=0pt] at (current page.center){%
        \includegraphics[width=\paperwidth,height=\paperheight]{\inserttitlebackground}%
      };
    \end{tikzpicture}
  }
}
{
  \newcommand{\addtitlebackground}{}
}
\providecommand{\titlebackground}[1]{\def\inserttitlebackground{#1}}
\makeatother
$endif$

% Méta-données du document %%%%%%%%%%%%%%%

$if(title)$
\title{$title$}
$endif$
$if(subtitle)$
\subtitle{$subtitle$}
$endif$
$if(author)$
\author{$for(author)$$author$$$sep$ \and $endfor$}
$endif$
$if(titlegraphic)$
\titlegraphic{%
  \rule{2em}{0cm}%
}
$for(titlegraphic)$
$if(titlegraphic.width)$
\includegraphics[width=$titlegraphic.width$\columnwidth]{$titlegraphic.file$}%
$else$
\includegraphics[width=.1\columnwidth]{$titlegraphic.file$}%
$endif$
$endif$

```

```

$endif$
\rule{2em}{0cm}$if(titlegraphic.nl)$\par\medskip\rule{2em}{0cm}$endif$%
$endfor$
}
$endif$
$if(titlebackground)$
\titlebackground{$titlebackground$}
$endif$
\date{$date$}

%%

\begin{document}

\begin{frame}[plain]
  $if(titlebackground)$\addtitlebackground
  $endif$\titlepage
\end{frame}

$if(toc)$$if(toc-title)$\renewcommand*{\contentsname}{$toc-title}$endif$
\begin{frame}[allowframebreaks]{$if(toc-title)$$toc-title$$else$Plan$endif$}
  \tableofcontents % il faut compiler deux fois pour mettre à jour la TDM
\end{frame}
$endif$

$body$

\end{document}

```

## B.2 pandoc-report.tex

```

\documentclass[10pt,twoside,openright]{report}

% Paquets LaTeX %%%%%%%%%%

% Une gestion correcte du français (en entrée et en sortie)
\usepackage[french]{babel}
\usepackage{typelec} % devant fontenc (cf typelec.sty)
\usepackage[T1]{fontenc} % devant inputenc (utf8 choisi en fonction de ça)
\usepackage[utf8]{inputenc}
\DeclareUnicodeCharacter{20AC}{\euro} % pour la saisie du caractère euro

% Des "jolies" polices de caractères
\usepackage{lmodern} % pour sf et tt
\usepackage{fourier} % pour rm
\usepackage{bbm} % pour les mathbbm

% Plein de symboles
\usepackage{amssymb,amsmath} % Les symboles mathématiques de l'AMS
\usepackage{latexsym} % Quelques symboles manquants dans LaTeX 2e
\usepackage{marvosym} % Quelques symboles en vrac par Martin Vogel
\usepackage{wasysym} % Quelques symboles en vrac par Roland Waldi
\usepackage{pifont} % Les symboles Dingbats
\usepackage{textcomp} % \textcopyright
\usepackage[copyright]{ccicons} % Les (c) comme dans Creative Commons
\usepackage[official,right]{eurosym} % L'euro

% Mise en page
\usepackage{geometry}

```

```

%% Quelques paquets utiles
\usepackage{appendix}           % pour la gestion des annexes
\usepackage{array}              % pour faciliter les styles de tableaux
\usepackage{longtable,booktabs} % pour les longues tables générées par pandoc
\usepackage{relsize}            % pour le changement de taille des polices
\usepackage[normalem]{ulem}     % pour avoir des soulignements funky
\usepackage{tikz}               % pour les dessins portables
\usepackage{fixltx2e}           % provides \textsubscript
\usepackage{graphicx,grffile}   % pour les images
\usepackage{fancyvrb}           % pour le code en verbatim
\usepackage{minted}             % pour les programmes
\usepackage{hyperref}           % le plus tard possible d'après la doc

%% Configuration pandoc %%

%% Images management
% Scale images if necessary so that they will not overflow the page margins by
% default, and it is still possible to overwrite the defaults using explicit
% options in \includegraphics[width, height, ...]{f}
$if(graphics)$
\makeatletter
\def\maxwidth{\ifdim\Gin@nat@width>\linewidth\linewidth\else\Gin@nat@width\fi}
\def\maxheight{\ifdim\Gin@nat@height>\textheight0.8\textheight\else\Gin@nat@height\fi}
\makeatother
\setkeys{Gin}{width=\maxwidth,height=\maxheight,keepaspectratio}
$endif$

%% Syntax highlighting
$if(highlighting-macros)$
$highlighting-macros$
$endif$

%% Generated lists
\providecommand{\tightlist}{%
  \setlength{\itemsep}{0pt}\setlength{\parskip}{0pt}}

%% Redefines (sub)paragraphs to behave more like sections
\ifx\paragraph\undefined\else
\let\oldparagraph\paragraph
\renewcommand{\paragraph}[1]{\oldparagraph{#1}\mbox{}}
\fi
\ifx\subparagraph\undefined\else
\let\oldsubparagraph\subparagraph
\renewcommand{\subparagraph}[1]{\oldsubparagraph{#1}\mbox{}}
\fi

%% Prevent overfull lines
\setlength{\emergencystretch}{3em}

% Adaptation LaTeX %%

\makeatletter

%% pour séparer les auteurs
\renewcommand{\and}{\endtabular\par\tabular[t]{c}}

%% sous-titre
\newcommand{\subtitle}[1]{\gdef\insertsubtitle{#1}}

%% images (logos) sur la page de titre
\newif\iftitlegraphic\titlegraphicfalse
\newcommand{\titlegraphic}[1]{\titlegraphictrue\def\inserttitlegraphic{#1}}

```

```

%% fond de la page de titre
\newif\iftitlebackground\titlebackgroundfalse
\newcommand{\titlebackground}[1]{\titlebackgroundtrue\def\inserttitlebackground{#1}}

%% la page de titre
\renewcommand{\maketitle}{%
  \begin{titlepage}%
    % Fond de page
    \iftitlebackground
    \begin{tikzpicture}[remember picture,overlay]
      \node[opacity=0.15,inner sep=0pt] at (current page.center){%
        \includegraphics[width=\paperwidth,height=\paperheight]{\inserttitlebackground}%
      };
    \end{tikzpicture}
    \fi
    % Titre & Date
    \begin{tikzpicture}[remember picture,overlay]
      \node[yshift=-6cm,inner sep=0pt] at (current page.north){%
        \begin{minipage}{.99\columnwidth}
          \centering%
          {\larger[4]\bfseries \@title}\par%
          \@ifundefined{insertsubtitle}{\par%
            \bigskip%
            {\larger[3]\color{gray}{.3}\slshape \insertsubtitle}\par}%
          \@ifundefined{@date}{\par%
            \bigskip%
            {\larger[3] \@date \par}}}%
        \end{minipage}
      };%
    \end{tikzpicture}
    % Auteurs
    \begin{tikzpicture}[remember picture,overlay]
      \node[yshift=-6cm,inner sep=0pt] at (current page.center){%
        \begin{minipage}{.9\columnwidth}
          \centering\larger[2]
          \begin{tabular}[t]{c}%
            \@author
          \end{tabular}%
        \end{minipage}
      };%
    \end{tikzpicture}
    % Logos
    \iftitlegraphic
    \begin{tikzpicture}[remember picture,overlay]
      \node[yshift=3cm,inner sep=0pt] at (current page.south){%
        \begin{minipage}{.99\columnwidth}
          \centering\larger[2]\inserttitlegraphic
        \end{minipage}
      };%
    \end{tikzpicture}
    \fi
  \end{titlepage}
}

%% pour débiter les annexes
\newcommand{\appendixes}{\appendix\appendixpage\addappheadtotoc}
\renewcommand{\appendixname}{Annexes}
\renewcommand{\appendixtocname}{Annexes}
\renewcommand{\appendixpagenam}{Annexes}

\makeatother

```

*% Configuration document %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*

```
\frenchbsetup{%
  ItemLabels=\textendash,%
  og=«, %
  fg=»}
\geometry{%
  a4paper,
  top=1cm,bottom=2cm,
  left=1.5cm,right=1cm}
\hypersetup{%
  $if(title-meta)$
  pdftitle={${title-meta$}},
  $endif$
  $if(author-meta)$
  pdfauthor={${author-meta$}},
  $endif$
  $if(keywords)$
  pdfkeywords={${for(keywords)}${keywords}$sep$; $endfor$},
  $endif$
  hyperfootnotes=false,
  colorlinks,
  urlcolor=blue,
  linkcolor=,
  pdfstartview=Fit}
\setlength{\parindent}{0pt}
\setlength{\parskip}{6pt plus 2pt minus 1pt}
$if(theme)$\usepackage{reporttheme$theme$}$endif$
```

```
$for(header-includes)$
$header-includes$
$endfor$
```

*% Méta-données du document %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*

```
$if(title)$
\title{${title$}}
$endif$
$if(subtitle)$
\subtitle{${subtitle$}}
$endif$
$if(author)$
\author{${for(author)}${author}$sep$ \and $endfor$}
$endif$
$if(titlegraphic)$
\titlegraphic{%
  \rule{4em}{0cm}%
$for(titlegraphic)$
$if(titlegraphic.width)$
  \begin{minipage}[c]{${titlegraphic.width$\textwidth}}
$else$
  \begin{minipage}[c]{.1\textwidth}
$endif$
  \includegraphics[width=\textwidth]{${titlegraphic.file$}}%
  \end{minipage}
  \rule{4em}{0cm}$if(titlegraphic.nl)$\par\bigskip\rule{4em}{0cm}$endif$%
$endfor$
}
$endif$
$if(titlebackground)$
\titlebackground{${titlebackground$}}
$endif$
```

```

\date{$date$}

%%

\begin{document}

$for(include-before)$
$include-before$
$endfor$

\maketitle

$if(toc)$ $if(toc-title)$ \renewcommand*{\contentsname}{$toc-title$}
$endif$ $if(toc-depth)$ \setcounter{tocdepth}{$toc-depth$}
$endif$ \tableofcontents$endif$
$if(lot)$ \listoftables$endif$
$if(lof)$ \listoffigures$endif$

$body$

$for(include-after)$
$include-after$
$endfor$

\end{document}

```