Site compagnon de Markdown & vous

Bernard Pochet

2024-11-02

Table des matières

Préface **4**

1 Introduction **5**
1.1 Commandes de base . . . 5
1.2 *LaTeX* . . . 6
1.2.1 Classes . . . 6
1.2.2 Paquets . . . 6
1.2.3 Format . . . 7
1.2.4 Métadonnées . . . 7
1.2.5 Templates . . . 7

2 Stucture du document **8**

3 La mise en page **10**
3.1 *Title page* . . . 10
3.2 Options de mise en page . . . 10
3.3 En-têtes et pieds de page . . . 11
3.4 Numérotation des pages . . . 13
3.5 Positionnement du texte sur la page . . . 14
3.5.1 Horizontalement . . . 14
3.5.2 Verticalement . . . 14
3.6 Texte (notes) dans les marges . . . 15
3.7 Préparation à l'impression . . . 15

4 Les caractères **16**
4.1 Police de caractères . . . 16
4.2 Tailles des caractères . . . 17
4.3 Lettrine . . . 17
4.4 Couleurs . . . 17
4.5 Petites capitales . . . 18
4.6 Boites de texte . . . 19
4.6.1 Cadre . . . 19
4.6.2 Bloc de texte . . . 20
4.7 Commentaires . . . 20
4.8 Autres commandes utiles . . . 20
4.8.1 Logo ORCID . . . 20
4.8.2 Filets (lignes) . . . 21
4.8.3 Caractères spéciaux . . . 21
4.8.4 Cercle . . . 22
4.8.5 Angles . . . 22

5 Les illustrations **23**
5.1 Tableaux et les figures 23
5.2 Positions 23
5.3 Wallpaper 24
5.4 Graphiques 24

6 Les index **26**
6.1 Table des matières 26
6.2 Table des figures et table des tableaux 26
6.3 Index de mots-clés 27

7 Rédiger un article **28**
7.1 Format 28
7.2 Bibliographie 29
7.3 Modèle 30
7.4 Épreuves 31

8 Réaliser un curriculum **32**
8.1 Principes 32
8.2 Modèle 32

9 Faire un diaporama **35**
9.1 Beamer 35
9.1.1 Modèle 35
9.1.2 Les commandes spécifiques les plus utiles sont : 36
9.2 Remark.js 37
9.2.1 Modèle 37
9.2.2 Utilisation 39

Préface

Ce site[1] complète l'ouvrage "Markdown & vous" publié en 2023 sur e-publish, la plateforme d'édition numérique d'ULiège Library.

L'ouvrage "Markdow & vous" présente les bases de l'utilisation de *Markdown* et justifie l'utilisation du mode texte comme outil d'écriture académique.

Ce complément présente une utilisation plus avancée pour la création de documents rédigés avec *Markdown* et transformés en fichiers pdf avec l'insertion de commandes spécifiques à *LaTeX*[2] utilisé par *Pandoc* pour créer ces document au format pdf.

D'une manière générale, il y a plusieurs méthodes pour arriver à un même résultat. Quelques-unes des méthodes liées à l'écriture de documents scientifiques avec *Markdown* sont présentées sont présentées dans les pages qui suivent[3].

Les promesses de *Markdown*, format texte, pour rédiger et produire des documents avec *Pandoc* et *LaTeX* sont nombreuses. La durabilité[4], la simplicité et la légèreté mais surtout la possibilité de reprendre possession de notre utilisation du numérique et la maîtrise d'outils non privateurs de libertés. Il ne faut pas pour autant que ce soit au prix d'une trop grande complexité.

Les pages qui suivent proposent des solutions pragmatiques à des problèmes que l'on peut rencontrer dans l'utilisation de *Markdown*.

Il ne faut pas que ce choix, principalement éthique, de sobriété et d'hygiène numérique, devienne une complication supplémentaire…

Trois modèles viennent compléter le site :

- Chapitre 7
- Chapitre 8
- Chapitre 9

[à propos de l'auteur]

[1] Pages rédigées en *Markdown* (sources) et générées avec *Quarto-book*.

[2] La source principale vient de huit années de pratique de cette technologie mais également la consultation approfondie de Desgraupes, B. (2019). *LaTeX : Apprentissage, guide et référence*. De Boeck Superieur.

[3] Toutes les commandes ont été testées avec *Pandoc 3.1.3* et *TeX Live 2023/Debian*.

[4] voir https://infolit.be/site/dd/.

1 Introduction

Markdown a évolué dans le temps :

- le *CommonMark*, format original, a été créé en 2004 pour produire des pages web[1] ;
- le *Markdown Extra* ajoute, par rapport au *CommonMark*, des tableaux, des abréviations, des notes en bas de page et des définitions. Il est par exemple utilisé pour les Wiki ou les CMS[2] comme Drupal ;
- le *Github Markdown* a aussi permis l'ajout de tableaux, des méthodes pour ajouter du code et l'utilisation de couleurs. C'est le format de tous les fichiers *readme* sur *Github* ;
- le *Vanilla Flavored Markdown* a amélioré les liens, permis l'inclusion de codes html et amélioré les listes (numérotation et indentation) ;
- enfin, le *MultiMarkdown* (assez proche du *Pandoc Markdown* utilisé dans le manuel) est plus orienté travail universitaire avec par exemple l'intégration et/ou l'amélioration des tables, des notes en bas de page, des références croisées ou des équations mathématiques.

En mars 2016, dans le but de standardiser le langage, deux RFC (appel à commentaires) ont été publiés :

- RFC 776314, qui introduit le type MIME text/markdown à partir de la variante originale de Markdown.
- RFC 776415, qui répertorie des variantes MultiMarkdown, GitHub Flavored Markdown (GFM), Pandoc, CommonMark, Markdown Extra et autres (source Wikipedia).

... Malgré son intérêt indéniable, il n'y a néanmoins toujours pas de norme associée à *Markdown*. C'est une critique régulièrement entendue.

1.1 Commandes de base

Les commandes de base du *Pandoc Markdown* présentées dans le manuel "Markdown & vous" sont communes à tous les formats générés. Ce sont les commandes relatives :

- à la structure du document (titres, paragraphes et listes) ;

[1] Pour transformer un fichier .md en fichier .html avec la méthode historique, on va utiliser **markdown.pl** (une application PERL créé par J. Gruber). On peut retrouver la source ici : https://github.com/lauriro/markdown/tree/master.

[2] *Content Management System*.

- à la mise en forme du texte (gras, italique, indice et exposant…) ;
- aux compléments du texte (codes, notes, liens, figures, tableaux, équations, citations et bibliographie).

L'utilisation, par *Pandoc* de *LaTeX* pour produire des documents pdf permet d'ajouter de nombreuses fonctions proposées par *LaTeX*. C'est une manière d'utiliser *LaTeX*, avec la qualité esthétique des documents qu'il produit, via *Markdown* sans passer par l'édition d'un fichier *LaTeX*, pas toujours simple à maîtriser.

Ainsi, avec les fonctionnalités *Markdown*, les commandes *LaTeX* ajoutées, le tout converti avec *Pandoc*, rédiger des documents scientifiques de qualité sur la base de solutions ouvertes et *low-tech* devient possible (et accessible) pour le plus grand nombre.

1.2 *LaTeX*

1.2.1 Classes

Lorsqu'on travaille avec LaTeX, on doit d'abord déclarer une *classe* de document. Avec *Markdown*, pour la production de documents pdf, on va faire cette déclaration dans l'en-tête YAML (voir dans le manuel) avec la commande :

"**documentclass: *classe***".

Les principales *classes* possibles (sans les variantes) sont :

- *article* (la classe par défaut)
- *report*
- *book*
- *letter* (non décrit dans le manuel)
- *beamer* (créée à partir du module *beamer* et décrit dans le chapitre Beamer).

1.2.2 Paquets

Après avoir déclaré une classe, on peut modifier les fonctionnalités de *LaTeX* en ajoutant un ou plusieurs **packages** (ou "paquets").

Avec *Markdow*, toujours dans l'en-tête YAML, on appelle ces **paquets** avec la commande :

\header-includes:

et, par exemple, pour modifier la langue du document :

\usepackage[french]{babel}

Les paquets peuvent :

- modifier la façon dont certaines parties de *LaTeX* fonctionnent ;
- changer l'apparence du document ;
- ajouter de nouvelles commandes qui seront utilisables par la suite dans la rédaction.

De nombreux exemples d'utilisation des paquets sont présentés dans les pages suivantes.

1.2.3 Format

Dans l'en-tête d'un document *LaTeX* (l'en-tête YAML pour un fichier *Markdown*), on va également définir certaines caractéristiques comme le format de la page (**geometry:**), la taille des caractères (**fontsize:**), la présence ou non d'une table des matières (**toc:**) ou la numérotation de titres (**numbersections:**).

1.2.4 Métadonnées

Dans l'en-tête YAML on va enfin ajouter les métadonnées (title, autor, keywords...).

Certaines de ces métadonnées apparaissent en en-tête du document pdf produit (voir Chapitre 2). Elles sont également utilisées pour les autres types de documents générés avec *Pandoc* (ePub, html...).

1.2.5 Templates

Il est aussi possible de se servir d'un template (modèle) qui sera appelé lors de la transformation avec *Pandoc* (–template=montemplate.tex).

Quelques références sont présentées dans le manuel.

2 Stucture du document

La structuration d'un document est un point essentiel dans sa qualité.

Pour rappel, avec *Markdown*, cette structure est particulièrement simple à mettre en œuvre avec l'utilisation du caractère # (**# titre de niveau 1**, **## titre de niveau 2**...).

Markdown permet six niveaux de titre mais il est rare de dépasser quatre niveaux dans un document déjà complexe.

```
# Titre de niveau 1
## Titre de niveau 2
### Titre de niveau 3
#### Titre de niveau 4
##### Titre de niveau 5
###### Titre de niveau 6
```

Ces niveaux seront différents en fonction de la *classe* du document.

Pour un **livre** on aura des **parties** (qui commenceront par défaut sur une page paire, à droite, donc avec l'ajout éventuel d'une page blanche), des **chapitres**, des **sections** et des **sous-sections**. Pour un **rapport**, ce sera la même structure mais il n'y a pas de notion de pages paires et impaires. Pour un **article**, le niveau 1 (# titre) sera un **chapitre**.

Table 2.1: Synthèse de la structure d'un document créé avec *Markdown* et *LaTeX* :

classe	page titre	niveau 1	niveau 2	résumé	entêtes
article	non (page 1)	Chapitre	Section	oui	non[1]
report	oui	Partie	Chapitre	oui, page 2	non
book	oui	Partie (page paire)	Chapitre	non	oui

Il va sans dire qu'un même niveau ne peut pas apparaître seul. Si un chapitre ne contient qu'une seule section, le titre de cette section doit être supprimé ou faire partie du titre du chapitre. Un document structuré se construit comme un arbre et on ne peut avoir une branche contenant seulement une sous-branche.

[1] Il est bien évidemment possible d'en ajouter avec le paquet *fancy*.

Pour un document html, contrairement à un document pdf où le titre du document se trouve dans l'entête YAML, le titre de niveau 1 (**# titre 1**) est le titre de la page. Il ne doit donc apparaître qu'une fois.

Le deuxième outil de structuration qui est utilisé est le paragraphe (pour rappel, deux sauts de ligne en *Markdown*). Il faut être attentif à rendre le texte lisible et donc éviter de mettre plusieurs idées dans un même paragraphe.

Un troisième outil, dernier niveau dans la granularité, est l'outil **listes** déjà décrit dans le manuel.

Le caractère **#**, les paragraphes et les listes (simples ou numériques) sont donc les trois éléments à utiliser pour structurer un document avec *Markdown* et *LaTeX*.

3 La mise en page

3.1 *Title page*

Nous l'avons vu, il existe plusieurs classes de documents (*article*, *report*...).

Les métadonnées qui apparaissent sur le haut de la première page (pour la classe *article*) et sur la première page (pour les classes *report* et *book*) sont déclarées dans l'en-tête YAML.

Table 3.1: Affichage des métadonnées YAML :

métadonnées	article	report	book
title:	oui	oui	oui
subtitle:	oui	oui	oui
author:	oui	oui	oui
date:	oui	oui	oui
abstract:	oui	oui[1]	non
thanks[2]:	oui	oui	oui

Les autres métadonnées (essentiellement *Keywords*, *DOI* et *Right*) sont enregistrées dans le document mais n'apparaissent pas sur la première page[3].

3.2 Options de mise en page

Dans l'en-tête YAML, l'instruction **geometry:** permet de préciser le format de la page (**A4paper** par exemple) mais aussi les quatre marges (*left*, *right*, *top* et *bottom* : avec **left=2.5cm** par exemple) et la taille des notes dans la marge (**marginpar=1.8cm** par exemple).

Les autres options sont :

- **landscape** (**portrait** par défaut) ;
- **onecolumn** (par défaut) ;
- **twocolumn** (réglage de la gouttière avec la macro **\setlength{columnsep}{5cm}** et de la ligne de séparation avec la macro **\setlength{columnseprule}{1pt}**)[4]

[1] Apparaît sur la deuxième page.
[2] Note sur le titre.
[3] Pour un document *html*, l'option **-s** dans la commande *Pandoc* va les inclure dansle *header* de la page.
[4] Pour rappel, il existe une autre méthode pour utiliser les colonnes, voir dans le manuel.

;
- **oneside** (pas de page gauche et droite, par défaut pour les classes *article* et *report*)
- **twoside** (par défaut pour la classe *book*) ;
- **openany** (uniquement pour la classe *book*, empêche l'ajout d'une page blanche pour commencer un chapitre sur une page de droite).

3.3 En-têtes et pieds de page

L'extension qui gère les en-têtes et pieds de page se charge en incluant, dans l'en-tête YAML, les commandes :

\usepackage{fancyhdr}

et

\pagestyle{fancy}

Le paramétrage des en-têtes se fait avec la commande (voir un exemple plus bas) :

\fancyhead[zone]{contenu}

et le paramétrage des pieds de page avec la commande

\fancyfoot[zone]{contenu}

Pour définir la zone, on utilise le codage suivant :

- L : champ gauche pour toutes les pages
- LE : champ gauche pour les pages paires
- LO : champ gauche pour les pages impaires
- C : champ central pour toutes les pages
- CE : champ central pour les pages paires
- CO : champ central pour les pages impaires
- R : champ droit pour toutes les pages
- RE : champ droit pour les pages paires
- RO : champ droit pour les pages impaires

Pour le contenu, on peut ajouter du texte mais on dispose également de commandes de formatage de *LaTeX*, par exemple des instructions suivantes :

\thepage

imprime le numéro de la page courante

\thesection

imprime le numéro de la section courante

\thechapter

avec un document du type books ou report : imprime le numéro du chapitre courant

\leftmark

- avec un document du type *article*, imprime le nom de la section courante
- avec un document du type *books* ou *report*, imprime le nom du chapitre courant

\rightmark

- avec un document du type *article*, imprime le nom de la sous-section courante
- avec un document du type *books* ou *report*, imprime le nom de la section courante

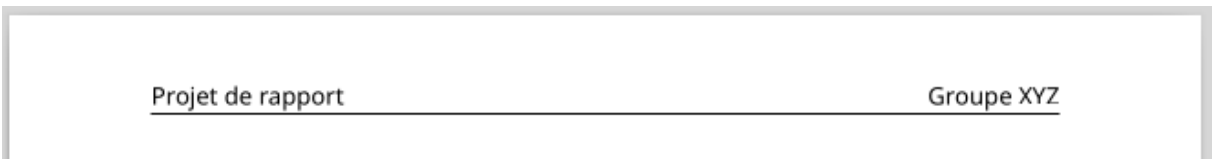
Remarques :

- pour forcer une zone à rester vide, il suffit de laisser vide la zone entre { et } ;
- pour forcer l'affichage d'une ligne horizontale, il suffit d'utiliser la commande : **\renewcommand{\headrulewidth}{1pt}** pour les en-têtes et **\renewcommand{\footrulewidth}{1pt}** pour les pieds de page ;
- pour ne pas afficher de ligne horizontale, il suffit d'utiliser la commande **\renewcommand{\headrulewidth}{0pt}** pour les en-têtes et **\renewcommand{\footrulewidth}{0pt}** pour les pieds de page ;
- pour empêcher l'affichage de l'en-tête et du pied de page sur la première page, il faut ajouter **\thispagestyle{empty}** juste en dessous de l'en-tête YAML.

En résumé, dans l'en-tête YAML, on pourrait avoir :

```
- \usepackage{lastpage}
- \usepackage{fancyhdr}
- \pagestyle{fancy}
- \fancyhead[L]{Projet de rapport}
- \fancyhead[C]{}
- \fancyhead[R]{Groupe XYZ}
- \fancyfoot[L]{\today}
- \fancyfoot[C]{}
- \fancyfoot[R]{\thepage/\pageref{LastPage}}
- \renewcommand{\headrulewidth}{0.1pt}
```

qui donnera en en-tête :



et en pied de page :

3.4 Numérotation des pages

Il est possible à tout moment de modifier la numérotation des pages.

\pagenumbering{*style*}

Va modifier le style de numérotation. Les possibilités pour *style* sont :

- **arabic** pour une numérotation en chiffres arabes (par défaut)
- **roman** pour une numérotation en chiffres romains
- **Roman** pour une numérotation en chiffres romains en capitales
- **alpha** pour une numérotation en lettres
- **Alpha** pour une numérotation en lettres capitales

Le numéro de la page en cours peut aussi être modifié :

\setcounter{page}{10}

va forcer la numérotation à 10 pour la page courante. Les deux commandes peuvent être combinées. Par exemple :

\pagenumbering{roman}

pour que la numérotation du début du document (préface par exemple) se fasse en chiffres romains, puis :

\pagenumbering{arabic}
\setcounter{page}{1}

pour recommencer la numérotation à 1, en chiffres arabes, pour la suite du document.

S'il y a beaucoup de notes en bas de page, une commande similaire permet de remettre à zéro la numérotation des notes en bas de page :

\setcounter{footnote}{0}

3.5 Positionnement du texte sur la page

3.5.1 Horizontalement

Avec la commande **\begin{center}** on va centrer le texte (ou tout autre contenu) qui suit. Il faut entrer **\end{center}** pour annuler le centrage. On peut aussi utiliser la commande **\centering** mais alors tout ce qui suit, jusqu'à la fin du document, sera centré.

La commande **\begin{flushright}** (et **\end{flushright}**) va pousser le texte à droite. On peut aussi utiliser **\raggedright** mais, comme pour le centrage avec **\centering**, tout ce qui suit sera aligné à droite.

Par défaut, une page va être justifiée (alignement à gauche et à droite). Pour améliorer la lecture, il est possible de forcer l'alignement à gauche (en "drapeau") avec **\raggedleft**, jusqu'à la fin du document.

3.5.2 Verticalement

La commande **\vspace{x}** va ajouter des lignes vides. **\vspace{5cm}** va par exemple "pousser" la suite 5 cm plus bas.

Il est aussi possible de sauter une ligne en insérant un paragraphe vide (ne contenant qu'une espace avec ** **).

La commande **\newline** (avec une espace avant la suite du texte ou **** suivi d'un retour ligne, sans espace) va provoquer un saut de ligne à l'intérieur d'un paragraphe.

La commande **\newpage** va provoquer un saut de page.

Pour rappel, afin d'augmenter la lisibilité d'un texte, on peut augmenter l'interligne (espace entre deux lignes) en mettant **linestretch: 1.5** (**1** par défaut) dans l'en-tête YAML.

On peut aussi jouer localement sur l'interligne avec le paquet **setspace** et la commande **spacing**.

On ajoute :

\usepackage{setspace}

dans l'en-tête YAML et **\begin{spacing}{2}** (ici pour doubler l'interligne) et **\end\{spacing}** (pour revenir à la taille normale de cet interligne) dans le texte.

Par exemple, pour coller (interligne = 0) un filet (avec **rule**, voir Section 4.8.2) à la ligne précédente, on aura :

\begin{spacing}{0}\rule{16cm}{0.02cm}\end{spacing}

3.6 Texte (notes) dans les marges

Texte texte \marginpar{ceci est une note courte} texte texte[5] va insérer, sur la ligne en cours, dans la marge extérieure (à droite pour la classe “article”), le bout de texte (note ou commentaire) qui se trouve entre { et }.

Pour avoir un texte dans un caractère plus petit, on peut ajouter **\scriptsize** (voir Section 4.2) au début de la note :

> **texte texte \marginpar{\scriptsize ceci est une note courte} texte texte.**

On peut par ailleurs préciser la largeur du texte dans la marge dans les instructions **geometry:** de l’en-tête : **marginpar=1.8cm**.

3.7 Préparation à l’impression

Pour aller jusqu’au bout d’un projet numérique pour un livre (*classe* “book”) on peut en faire une version imprimée (en général à la demande).

Pour simplifier le travail de l’imprimeur, on peut lui fournir un document pdf avec des traits de coupe qui indiquent avec précision l’endroit où il faut rogner le livre après impression et avant la reliure

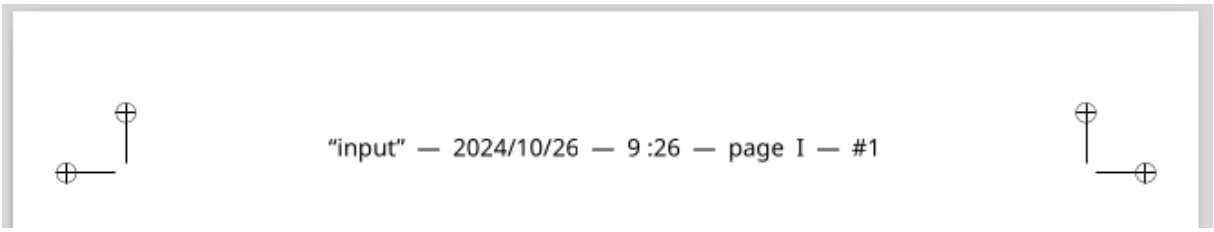
Pour un ouvrage au format classique 170 x 240 mm on va, sur une page A4 (210 x 297 mm), déclarer le format fini (y compris les marges) dans **geometry:**

> **geometry:**
> **- paperwidth=17.0cm**
> **- paperheight=24.0cm**
> **- left=2.5cm**
> **- right=2.5cm**
> **- top=2cm**
> **- bottom=2cm**

et la taille de la feuille dans la commande **crop**.

> **\usepackage[cam, a4, center]{crop}**

cam va ajouter le nom du fichier, la date et l’heure d’impression et le numéro de la page et **center** va centrer le tout sur la page A4.



[5] Ne fonctionne pas si le texte est en colonne via **\usepackage{multicol}** mais bien via **classoption: twocolumn**.

4 Les caractères

> “La typographie – du caractère jusqu’à la mise en page – relève du soin. Pour bien mettre en page un texte, il faut beaucoup d’empathie, il faut avoir envie de faire du bien au lecteur en lui rendant agréable sa rencontre avec le texte (Tafin, 2023)”.

4.1 Police de caractères

Même si la police de caractères par défaut (*lmodern*) est agréable, il est possible de changer de police pour la création de documents pdf.

Il faut d’abord faire la différence entre :

- la famille (monospace, avec empâtement (*serif*) ou sans empâtement (*sans serif*))
- le style (gras, italique…)
- la police.

Ici encore, le changement de police se fait via une instruction dans l’en-tête YAML. Voici les polices déjà proposées dans le manuel (que l’on appelle en ajoutant **\usepackage{nom_de_la_police}** dans la section **header-includes:** de l’en-tête YAML) :

- times
- palatino
- bookman
- newcent
- helvet
- avant

Pour les polices *helvet* et *avant*, qui sont des polices sans empâtement (*sans serif*), il faut ajouter la ligne **\renewcommand{\familydefault}{\sfdefault}** dans la section **header-includes:** de l’en-tête YAML.

Pour la police “Noto sans”, de plus en plus utilisée parce que particulièrement lisible, il faut placer **\usepackage[sfdefault]{noto}** dans la section **header-includes:** de l’en-tête YAML.

Pour retrouver l’ensemble des polices possibles, par famille, il faut consulter le catalogue *LaTeX*. Pour chacune des polices, la commande à ajouter dans l’en-tête (YAML) est précisée.

4.2 Tailles des caractères

Dans un document, il est possible de modifier la taille des caractères en déclarant cette taille (10 possibilités, de 5 à 25 pt). Pour la plus grande taille, on va utilise **\Huge** :

\Huge texte \normalsize

(ne pas oublier de revenir à la taille normale avec **\normalsize**).

Le tableau ci-dessous reprend, pour les trois tailles standard (10, 11 et 12 pt), la taille résultante.

Table 4.1: Taille des caractères

Taille	10 pt	11pt	12 pt
\Huge	25pt	25pt	25pt
\huge	20pt	20pt	25pt
\LARGE	18pt	18pt	20pt
\Large	14pt	14pt	18pt
\large	12pt	12pt	14pt
\normalsize	10pt	11pt	12pt
\small	9pt	10pt	11pt
\footnotesize	8pt	9pt	10pt
\scriptsize	7pt	8pt	8pt
\tiny	5pt	6pt	6pt

4.3 Lettrine

L'appel du paquet **\usepackage{lettrine}** dans l'en-tête YAML permet d'ajouter une lettrine en début de paragraphe. Pour l'activer, il faut entrer **\lettrine** au début du paragraphe et laisser une espace.

\lettrine Texte du paragraphe

Va faire apparaître le "T" de "Texte" sur deux lignes.

4.4 Couleurs

Par défaut, *LaTeX* connaît les 19 couleurs suivantes : black, blue, brown, cyan, dark-gray, gray, green, lightgray, lime, magenta, olive, orange, pink, purple, red, teal, violet, white et yellow.

Pour utiliser les 64 couleurs CMYK de base, il faut ajouter : **\usepackage{xcolor}** dans la section **header-includes:** de l'en-tête YAML.

Les 64 couleurs possibles :

Figure 4.1: Source : Overleaf

(Attention à la première lettre en majuscules)

Quelques exemples :

> **\color{cyan}écrit en cyan \color{black}**
>
> **\color{Tuquoise}écrit en tuquoise \color{black}**
>
> **\colorbox{BurntOrange}{ajoute un fond orange dans un box de texte}**

(voir d'autres exemples plus bas)

On peut aussi utiliser ces couleurs pour définir, dans l'en-tête YAML, la couleur des liens (internes et externes) :

> **linkcolor: YellowGreen**
>
> **urlcolor: PineGreen**

4.5 Petites capitales

Pour avoir du texte en petites capitales, on a deux possibilités :

> **\textsc{Texte texte texte}**

ou

[Texte texte texte]{.smallcaps}

4.6 Boites de texte

4.6.1 Cadre

Les cadres de texte peuvent être en noir & blanc avec un filet sur les quatre côtés ou en couleur avec la possibilité de choisir la couleur du fond et celle des bords. Dans tous les cas, ces cadres sont limités à une seule ligne de texte.

Pour un cadre simple :

\fbox{boite de texte}

Pour un cadre avec de la couleur (voir ci-dessus pour la liste des couleurs) :

\colorbox{color1}{boite de texte}

Avec seulement une couleur pour le fond.

\fcolorbox{color1}{color2}{boite de texte}

Avec color1 qui est la couleur du cadre et color2 la couleur du fond. Par exemple avec **Fuchsia** et **Apricot** :

boite de texte

Pour un cadre qui dépasse une ligne (un paragraphe), il faut passer par un **mini-page** :

\fbox{\begin{minipage}{1\textwidth}Un texte de plusieurs lignes à encadrer\end{minipage}}

Ici “1” est la taille maximale du cadre. En mettant 0.75, on a un cadre qui occupe 75 % de la largeur entre les marges.

Il est par ailleurs possible de combiner ces instructions avec une modification de la position (centré ou aligné à droite) et avec une modification de la taille des caractères (voir plus haut).

4.6.2 Bloc de texte

\parbox crée un bloc de texte dont la largeur est définie (en pouces ou en cm). Contrairement aux boîtes de texte, le bloc de texte peut contenir plus qu'une ligne.

\parbox{10cm}{\scriptsize Bloc de texte.}

Ce bloc a une largeur de 10 cm (et apparaît en **scriptsize**). Par défaut il est aligné à gauche mais avec la commande **\begin{center}** il sera centré entre les deux marges.

Le bloc de texte n'est pas entouré d'un filet (lignes).

4.7 Commentaires

On peut, en cours de rédaction, insérer un commentaire dans le document source avec :

[comment]: ceci est un commentaire

ou avec :

[//]: ceci est un commentaire

Cette commande insère un commentaire qui ne se trouvera pas dans le document pdf produit[1] mais permet de documenter le texte en cours de rédaction et de cacher une partie du texte qui pourra par la suite être "dé-commentée".

4.8 Autres commandes utiles

4.8.1 Logo ORCID

On peut, dans un document, placer le logo (et le lien vers) ORCID. Il faut pour cela, dans l'en-tête YAML ajouter le paquet **orcidlink** et ajouter le lien derrière le nom de l'auteur.

\usepackage{orcidlink}

pour le paquet

author: Nom, Prénom\orcidlink{0000-0002-1873-1237}

[1] Fonctionne aussi avec les autres formats comme le html.

pour compléter les métadonnées

Cette commande fonctionne pour la création de documents pdf mais pas lors de l'exportation (avec *Pandoc*) dans un autre format comme .docx (par exemple pour la soumission d'un article, voir Chapitre 7).

4.8.2 Filets (lignes)

La commande **---** (trois tirets) déjà présentée dans le manuel va insérer une ligne horizontale, au centre de la page. Avec **rule** il est possible de "dessiner" des lignes de longueurs variables en précisant la largeur et la hauteur de celles-ci avec **\rule{largeur}{hauteur}**.

\rule{5cm}{0.02cm}

dessine une fine ligne de 5 cm.

\rule{0,5cm}{0,5cm}

va dessiner un carré de 0,5 cm de côté.

4.8.3 Caractères spéciaux

En ajoutant le paquet **dingbat**, il est possible d'insérer des caractères spéciaux dans un document.

Pour le caractère *Checkmark*, il faut donc ajouter

\usepackage{dingbat}

dans l'en-tête YAML et, dans le texte, appeler

\checkmark

Avec la police Time, de nombreux caractères spéciaux sont déjà inclus au départ mais par contre l'ajout de caractères spéciaux ne fonctionne pas avec certaines polices dont *Noto*.

En cas de problème, le paquet **bbding** peut être une alternative.

4.8.4 Cercle

On peut aussi insérer un cercle dans un document. La commande fonctionne sans appeler de paquet particulier.

\circle{15}

va insérer un cercle de 15 points de diamètre.

4.8.5 Angles

Si on souhaite modifier l'angle d'apparition du texte (oblique, vertical...), on va utiliser la commande **turn** après avoir appelé le paquet **rotating**.

\usepackage{rotating}

pour appeler le paquet (dans l'en-tête YAML)

\begin{turn}{45}texte texte texte\end{turn}

pour orienter le texte à 45° plutôt qu'à l'horizontale.

5 Les illustrations

5.1 Tableaux et les figures

Pour rappel, voir les commandes de base dans le manuel.

Plus spécifiquement, pour les tableaux, la construction est parfois complexe. Deux fonctions *LaTeX* sont souvent utiles.

Pour aérer un tableau, on peut ajouter, dans l'en-tête YAML ou juste avant le tableau concerné, **\renewcommand{1}{1.2}**[1] pour ajuster l'espacement vertical entre les lignes d'un tableau. Ici **1,2** au lieu de 1 signifie qu'on augmente l'espace de 20%.

Par contre, si la place manque, il est parfois utile de réduire la taille des caractères avec les commandes présentées Section 4.2.

5.2 Positions

Par défaut, *LaTeX* va placer les figures et les tables là où il y a de la place, c'est la fonction *floating*. C'est cette fonction qui a participé à la bonne réputation de *LaTeX*. Sans cette fonction, une table ou une illustration trop grande peut provoquer un saut de page et dès lors laisser un certain nombre de lignes vides, qui rendent le document inesthétique.

Parfois, les illustrations sont rassemblées à la fin. Les décisions de *LaTeX* ne sont pas toujours simples à comprendre.

S'il y a trop de figures et/ou de tables, il est possible qu'il y ait "encombrement" en fin de document. Il est cependant possible de "forcer" *LaTeX* à les insérer à un endroit de notre choix avec la commande **\clearpage**. Cette instruction va forcer *LaTeX* à ajouter un saut de page et insérer les tableaux et les figures en attente. Il faut néanmoins veiller à ce que ce saut de page n'intervienne pas en début de page.

[1] \renewcommand{...} permet de redéfinir une commande existante.

5.3 Wallpaper

Il est aussi possible d'ajouter une image avec le paquet **WallPaper**.

- \usepackage{wallpaper}

Prévu pour gérer les fonds de page (que l'on peut aussi gérer avec le paquet **background**), ce paquet permet de positionner une image dans un des quatre coins de la page.

Avec la commande **\ThisURCornerWallPaper{0.2}{image.jpg}**, l'image sera placée en haut, à droite, et aura une taille de 0,2 point (voir l'exemple dans le Chapitre 8). On utilise :

- L et U pour *Lower* et *Uper*
- L et R pour *Left* et *Right*

5.4 Graphiques

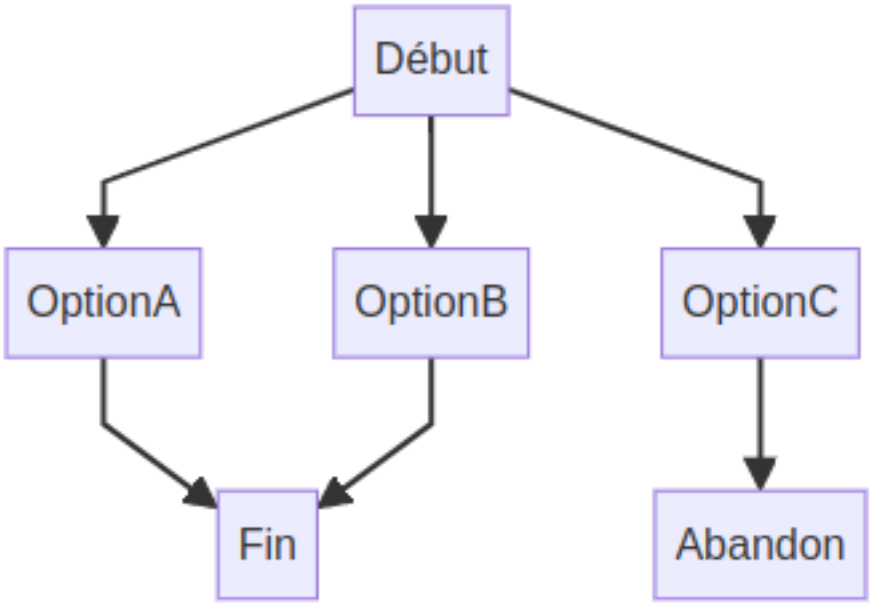
Markdown permet de créer des graphiques professionnels avec Mermaid. Ces graphiques peuvent être des :

- organigrammes
- diagrammes de séquence
- diagrammes de Gantt
- diagrammes de classes
- diagrammes d'état
- diagrammes circulaires
- graphiques à barres
- ligne du temps
- ...

Par exemple :

```
graph TD;
    Début-->OptionA;
    Début-->OptionB;
    Début-->OptionC;
    OptionC-->Abandon;
    OptionA-->Fin;
    OptionB-->Fin;
```

donnera :



Pour débuter, il existe des éditeurs en ligne qui permettent de faciliter la création de graphiques *Mermaid* :

- mermaid.d.foundation
- mermaid.live

Dans la commande *Pandoc*, il faut appeler le filtre **mermaid** :

pandoc -F mermaid-filter document.md -o document.pdf

6 Les index

6.1 Table des matières

toc: true dans l'en-tête YAML va insérer la table des matières au début du document.

La profondeur de la table des matières est déclarée dans l'en-tête YAML avec **toc_depth: x** (x étant le nombre de niveaux).

Pour placer cette table des matières ailleurs qu'au début, il faut l'appeler avec **\tableofcontent**.

On peut également modifier :

- son titre avec **\renewcommand{\contentsname}{Sommaire}** (ici "Sommaire" au lieu de "Table des matières") ;
- ou la couleur des liens avec **\hypersetup{colorlinks=true, linkcolor=black}**.

Ces deux commandes doivent apparaître avant **\tableofcontents**

6.2 Table des figures et table des tableaux

Les tables des figures et des tableaux sont automatiquement générées en ajoutant :

- **tof: true**
- **tot: true**

dans l'en-tête YAML.

Elles apparaissent en fin de document.

6.3 Index de mots-clés

Avec *Markdown* et *Pandoc*, il n'est pas possible d'activer la fonction intégrée dans *LaTeX*[1]. Il faut donc détourner le problème en :

- identifiant les termes à mettre dans l'index avec **\label{terme}terme** ;
- en fin de document (après un titre **# index** par exemple), ajouter les termes avec **terme\dotfill\pageref{mot}** (liste à trier alphabétiquement).

\dotfill va aligner le numéro de page à droite avec une ligne de points.

\pageref{mot} va insérer le numéro de la page où le terme a été "taggé" (avec **\label{terme}terme**).

Index

terme1 4
terme2 2
terme3 10

[1] Cet index est généré avec plusieurs compilations successives.

7 Rédiger un article

Le manuel “Comprendre et maîtriser la lecture scientifique” présente deux chapitres à consulter avant de se lancer dans la rédaction d’un article :

- La rédaction d’un document scientifique
- La rédaction d’un article scientifique

(voir aussi le support de cours)

Markdown est un excellent format pour cette rédaction. Il permet d’aller à l’essentiel (écrire) sans distractions inutiles.

Pour rédiger un article, on commence par ouvrir un éditeur de texte (à moins de choisir une autre méthode, voir dans le manuel le chapitre “outils”). On enregistre le document, en indiquant **.md** comme extension (typiquement **article.md**).

7.1 Format

Le principe de base, est de rédiger cet article en suivant le guide des auteurs de la revue à laquelle l’article va être soumis. Il est en effet essentiel, dès le départ, de :

- choisir la revue (pour rédiger directement avec les bonnes instructions) ;
- décider de la liste des auteurs (pour éviter de mauvaises surprises par la suite).

Dans la majorité des cas, les instructions demanderont :

- de fournir un document au format **.docx** ;
- de fournir un titre (et éventuellement un titre court pour l’en-tête), en français et en anglais ;
- de rédiger un résumé (le plus souvent structuré) en français et en anglais ;
- de fournir des mots-clés et des keywords ;
- de structurer l’article (par exemple, pour un article STM, au format IMReD[1]) ;
- de préciser, pour chaque autrice et auteur, un identifiant ORCiD et l’apport individuel en utilisant éventuellement la taxonomie CReDiT.

[1] Introduction, Matériel et méthode, Résultats et Discussion.

Comme il faut fournir un fichier **.docx** (ou éventuellement un fichier **.odt**), il faut éviter les instructions spécifiques à la création des fichiers **.pdf** via *LaTeX* et se "limiter" aux instructions *Markdown* de base qui sont de toutes façons largement suffisantes (tout ce dont on a besoin s'y trouve) .

Il est totalement inutile d'essayer de soumettre un article avec un *look* d'article fini, comme *LaTeX* le fait parfaitement. C'est l'éditeur qui fait ce travail en tenant compte de ses outils, de ses choix éditoriaux et de sa maquette. Tous les choix esthétiques que vous feriez seront effacés.

7.2 Bibliographie

La liste bibliographique et les citations dans le texte constituent des éléments essentiels d'un article scientifique. L'utilisation du format *Pandoc Markdown* est particulièrement puissant à ce propos.

Pour rappel :

- il faut disposer de deux fichiers bien spécifiques :
  - un fichier **BibTex** créé avec un gestionnaire bibliographique. Avec *Zotero*, il faut utiliser la fonction "exporter les documents" (sélectionnés) et choisir le format *BibTex* pour créer un fichier **.bib** ;
  - copier ce fichier dans le même dossier que l'article ;
  - copier le fichier **csl** (*Citation Stype Language*) correspondant au style bibliographique que vous souhaitez utiliser (l'éditeur vous demandera peut-être un style spécifique mais vous fournira généralement le fichier **csl** correspondant) ;
  - que ces deux fichiers soient renseignés dans l'en-tête YAML (voir le modèle ci-dessous) ;

Pour insérer les citations :

- soit avec l'appel [@auteur0000][2] pour une citation avec des parenthèses : "texte texte texte (Auteur, 0000)" ;
- soit avec l'appel @auteur0000 (sans [et]) pour une citation directe : "texte texte texte Auteur (0000)" ;

Ajouter :

- le titre "Bibliographie" en fin de document ;
- "–citeproc" dans la commande *Pandoc* : **pandoc --citeproc article.md -o article.docx**

[2] @auteur0000 est la clé de citation enregistrée dans le gestionnaire bibliographique

7.3 Modèle

Ci-dessous un modèle basique que vous pouvez copier-coller dans votre éditeur de texte pour débuter la rédaction. Dans ce modèle :

- on choisit la classe *article* ;
- pour améliorer la lisibilité, on veille à choisir un interligne supérieur à la valeur par défaut (ici 1,5 au lieu de 1) et on choisit un caractère de 12 ;
- on n'utilise pas le paquet **orcidlink** (voir Section 4.8.1) parce qu'il ne fonctionne pas pour les exports **.docx** (on va néanmoins introduire le n° ORCiD au début de l'affiliation qui apparaît en note pour chaque auteur) ;
- on va mettre le résumé dans les deux langues et les mots-clés dans les deux langues en dehors de l'en-tête YAML (avec les métadonnées YAML, on ne gère qu'une seule langue à la fois), au début du texte.

Pour l'interligne et la taille des caractères, il faudra de toutes façons se conformer aux instructions de l'éditeur.

```
---
documentclass: article
header-includes:
    - \usepackage[french]{babel}
    - \usepackage[sfdefault]{noto}
title: Titre
subtitle: Sous-titre
author:
- Auteur1^[Orcid1, université 1]
- Auteur2^[Orcid2, université 2]
date: 2024
right: CC-BY
geometry:
- left=2cm
- right=2cm
- top=2cm
- bottom=2cm
- a4paper
linestretch: 1.5
fontsize: 12pt
bibliography: library.bib
csl: apa.csl
---

## Résumé

## Abstract
```

```
## mots-clés

## Keywords

# Introduction

# Matériel et méthode

# Résultats

# Discussion/conclusion

# remerciements

# Bibliographie
```

7.4 Épreuves

Une fois le travail de rédaction terminé (ou à tout moment d'ailleurs) on peut, pour une épreuve, exporter un fichier **.pdf**, ou pour la soumission, faire une exportation au format **.docx**.

> **pandoc –citeproc article.md -o article.docx**

ou

> **pandoc –citeproc article.md -o article.pdf**

Voici ce que donne ce modèle et voici un exemple plus complet.

Lors du processus d'édition, le fichier **.docx** fera l'objet d'annotations ou de suggestions de modifications. Il est parfois pénible d'identifier celles-ci dans le document reçu.

Avec *LibreOffice*, la "comparaison de documents" (**Édition** -> **Suivit des modifications** -> **Comparer le document**) permet d'obtenir une liste des changements entre le document soumis (ou une épreuve intermédiaire) et le document transmis par l'éditeur.

On peut aussi décider de poursuivre avec un traitement de texte mais alors on perd une partie de la souplesse du travail avec un format texte…

8 Réaliser un curriculum

8.1 Principes

Réaliser un curriculum avec *Markdown* est relativement simple et souple. Comme il s'agit d'un document en une ou deux pages (ce qui est demandé par les employeurs), on va choisir la classe *article*. On va également choisir un rendu sobre.

On commence par ouvrir un éditeur de texte (à moins de choisir une autre méthode, voir dans le manuel le chapitre "outils"), et on enregistre le document, en indiquant **.md** comme extension (typiquement **curriculum.md**).

8.2 Modèle

Ci-dessous un modèle basique que vous pouvez copier-coller dans votre éditeur de texte pour débuter la rédaction.

Dans ce modèle, rien de particulier si ce n'est l'utilisation :

- du paquet **orcidlink** (voir Section 4.8.1)
- l'ajout d'une photographie (en haut, à droite) avec le paquet **wallpaper** (voir Section 5.3)
- le choix de la police de caractères **Noto** (voir Section 4.1) qui est plus élégante que la police par défaut.

```
---
documentclass: article
header-includes:
    - \usepackage[french]{babel}
    - \usepackage{wallpaper}
    - \usepackage[sfdefault]{noto}
    - \usepackage{orcidlink}
geometry:
- a4paper
- top=2.5cm
- bottom=2cm
- left=2.5cm
- right=2.5cm
```

```
right: CC-BY
fontsize: 12pt
linkcolor: blue
---
\ThisURCornerWallPaper{0.2}{personne.jpg}

# Curriculum Vitae -- Nom Prénom \orcidlink{0000-0001-0002-0003}

né le 1 janvier 1980 à Pétaouchnok\
nationalité : européen\
état civil : pas marié, pas d'enfant\
courriel : non.prenom@mailo.fr\
Mastodon : https://mastodon.zaclys.com/\@nomp\
LinkedIn : https://be.linkedin.com/in/nomp\
Blog : https://monblog.org

---

## 1. Diplômes universitaires

- Master &hellip; ;
- Doctorat en &hellip;.

## 2. Autres diplômes

- &hellip;

## 3. Emplois

- &hellip;

### Autres mandats :

- &hellip; (dates)

### Anciens mandats ou activités significatives :

- &hellip; (dates)

## 4. Autres activités et hobbies

- &hellip;

## 5. Publications (et activités de recherche)

- &hellip;

## 6. Compétences linguistiques
```

```
Langue maternelle : français

autres langues :

-   Anglais : ABCD
-   Néerlandais : ABCD
-   Espagnol : ABCD

### Résumé des domaines d'expertise

- &hellip;

### Coordonnées privées :

4, rue du Machin\
01235140 Houtsiplou (Europe)\
tél. : +12(0)34 56 78 90\
Courriel : [non.prenom@mailo.fr](mailto:non.prenom@mailo.fr)

### Coordonnées professionnelles :

Université Machin-chose\
&hellip;\
&hellip;\
tél. : &hellip;\
Courriel : [non.prenom@umchosec.eu](non.prenom@umchosec.eu)
```

Pour créer le fichier **pdf**, on va utiliser la commande habituelle.

pandoc curriculum.md -o curriculum.pdf

Voici ce que donne ce modèle. On ne va pas fournir d'autres formats (**.docx** par exemple) parce que le document ne doit être modifié que par l'auteur.

9 Faire un diaporama

Cette page complète la page du manuel consacrée à *Beamer*. Vous trouverez ci-dessous deux exemples complets avec *Beamer* (une classe *LaTeX*) et *Remark.js* (une page web avec du code *javascript*).

L'objectif d'utiliser *Markdown* pour produire des diapositives permet de travailler de manière[1] :

- simple et rapide : aucune sophistication
- propre : pas de codes cachés
- ouverte : source lisible avec un simple éditeur
- durable : le fichier source est un fichier texte
- structurée : grâce à la simplicité de *Markdown*
- esthétique : grâce aux nombreuses possibilités offertes

Les deux exemples proposent le même contenu mais avec des technologies différentes. Avec *Remark.js*, il n'est pas nécessaire d'installer *Pandoc*, un simple éditeur de texte (pour éditer un fichier *html*) suffit.

9.1 Beamer

La création de documents avec *Beamer* va se faire comme avec les documents présentés dans les deux modèles précédents. Voici un exemple.

9.1.1 Modèle

Ci-dessous un modèle basique que vous pouvez copier-coller dans votre éditeur de texte pour débuter la création de diapositives. Dans ce modèle :

- on choisit la classe (**output: beamer_presentation**) ;
- on choisit un **theme**, ici *Madrid* et un **colortheme**, ici *seahorse* ;
- on précise le format, ici **16*9** pour correspondre aux écrans actuels ;
- on précise le logo qui apparaîtra sur la première diapositive (**titlegraphic: logo.png**) et celui qui servira de fonds d'écran sur toutes les diapositives (**logo: fond.png**).

[1] Ce qui est diamétralement différent ce que propose *PowerPoint*.

Les métadonnées (title, subtitle, author et date), avec le thème choisi, apparaîtront sur la première diapositive et en pied de page pour toutes les diapositives (sauf subtitle)[2].

```
---
title: Titre
subtitle: Sous-titre
author: Nom, Prénom
date: 2021 (cc-by)
output: beamer_presentation
theme: Madrid
colortheme: seahorse
fontsize: 10pt
aspectratio: 169
titlegraphic: logo.png
logo: fond.png
---

# Diapositive 1

Liste :

- un
- deux
- trois

![Une image](image.jpg)

# Diapositive 2

## Premier cadre

texte

## Deuxième cadre

texte
```

9.1.2 Les commandes spécifiques les plus utiles sont :

- “#” pour créer une nouvelle page avec un titre courant (qui suit le signe “#” dans la source)

[2] Il faut veiller à ne pas choisir un titre trop long.

- “---” (trois tirets) pour créer une nouvelle page sans titre
- “##” pour créer un bloc (avec cadre pour certains thèmes) dans une diapositive
- “. . .” (trois points séparés par une espace) pour créer une pause dans la présentation. *Beamer* va créer autant de pages qu’il y a de pauses (un “clic” pour afficher la suite de la page). C’est la seule animation possible).

Le texte est aligné à gauche.

Les images sont alignées à gauche et automatiquement redimensionnées si elles dépassent le cadre. S’il y a une légende (**![Titre](image.png)**), l’image est alors centrée. Il est possible de réduire la taille des images avec la commande **{ width=50% }**.

Les tableaux sont bien reproduits. Les formules également.

Pour produire le fichier pdf, on va utiliser la commande :

pandoc -t beamer votrefichier.md -o votrefichier.pdf

9.2 Remark.js

Avec Remark.js, la création de diaporamas est encore plus légère et simple. Basé sur le format *Github flavor Markdown*, le fichier est directement interprété par le navigateur. Voici un exemple (pressez CTRL-u pour voir le contenu du fichier).

9.2.1 Modèle

Il n’est pas obligatoirement destiné aux personnes connaissant les codes HTML et CSS. Il ne nécessite qu’un simple éditeur.

Vous pouvez copier le code ci-dessous dans un fichier texte et le sauver comme une page web (typiquement : **diapo.html**).

Attention à ne pas toucher au début, ni à la fin du code. Vous écrivez entre **<textarea id=“source”>** et **<\textarea>**. Vous pouvez néanmoins changer le nom de la “page” html (entre **<title>** et **</title>**).

On peut introduire des images (****), des fond d’écran (**background-image: url(image.jpg)**), des liens, des listes ou des tableaux.

L’ensemble des commandes possibles est décrit sur le wiki du projet.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titre</title>
    <meta charset="utf-8">
    <style>
    @import url(https://fonts.googleapis.com/css?family=Yanone+Kaffeesatz)
    @import url(https://fonts.googleapis.com/css?family=Droid+Serif:400,70
    @import url(https://fonts.googleapis.com/css?family=Ubuntu+Mono:400,70

      body { font-family: 'Droid Serif'; }
      h1, h2, h3 {
        font-family: 'Yanone Kaffeesatz';
        font-weight: normal;
      }
    .remark-code, .remark-inline-code { font-family: 'Ubuntu Mono'; }
    </style>
  </head>
  <body>
    <textarea id="source">

class: center, middle

# Titre
## Sous-titre

date

---

# Diapositive 1

Liste :

- un
- deux
- trois

![Une image](image.jpg)

# Diapositive 2

## Premier sous-titre

texte

## Deuxième sous-titre
```

```
texte

    </textarea>
   <script src=”https://remarkjs.com/downloads/remark-latest.min.js”>
    </script>
    <script>
      var slideshow = remark.create({ratio: '16:9'});
    </script>
  </body>
</html>
```

9.2.2 Utilisation

Une fois le fichier **html** créé, complété et sauvé, il est ouvert avec un navigateur web (*Firefox* ou autre).

Si on dispose d’un espace web, on peut l’y copier (attention à joindre les images utilisées).