

L'écriture académique au format texte avec Markdown et Pandoc

L'écriture académique au format texte avec Markdown et Pandoc

BERNARD POCHE

ULIÈGE LIBRARY / PRESSES UNIVERSITAIRES DE LIÈGE
LIÈGE



L'écriture académique au format texte avec Markdown et Pandoc Droit d'auteur © 2025 par Bernard Pochet est sous licence [Licence Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/), sauf indication contraire.

Vous êtes libre de reproduire, de modifier, de distribuer et de communiquer cette création au public à la condition de citer le nom de l'auteur original de la manière indiquée par l'auteur de l'œuvre ou le titulaire des droits qui vous confère cette autorisation (mais pas d'une manière qui suggérerait qu'ils vous soutiennent ou approuvent votre utilisation de l'œuvre).

Le texte de la licence peut être consulté à l'adresse : <http://creativecommons.org/licenses/by/4.0/deed.fr>

Rien dans cette licence ne diminue ou ne restreint le droit moral de l'auteur.

Ce livre a été créé avec Pressbooks (<https://pressbooks.com>) et converti par Prince.

Table des matières

Introduction	9
<u>1. Reprendre le contrôle du numérique</u>	13
Le numérique	17
Le traitement de texte	21
Les formats	26
Historique	35
Limites, normes et standards	38
<u>2. Le Pandoc-Markdown</u>	43
Principe	45
La structure du document	46
La mise en forme du texte	50
Les compléments du texte	51
Les codes et textes non formatés	59
Autres codes utiles pour la mise en forme	60
L'en-tête YAML	62
Les outils	68

<u>3. LaTeX</u>	81
Principes de base	83
Les caractères	85
La mise en page	95
Les illustrations	105
Les index	110
Créer un diaporama	112
<u>4. Exemples et modèles</u>	119
Un article	121
Un diaporama	127
Un curriculum	132
<u>Compléments</u>	137
Sites de référence	139
Exercices	140
Aide mémoire	142
Pour aller plus loin... ..	144
Conclusion	147
Bibliographie	149
À propos de l'auteur	152

Introduction

Lorsque je pose, à mon entourage, la question :

- quel logiciel utilises-tu pour rédiger tes textes ?

La réponse est, quasi invariablement :

- Word.

Si j'ajoute :

- et pour prendre des notes ?

La réponse la plus fréquente est :

- aussi, pourquoi ?

Pour reprendre la main face aux impératifs de l'industrie informatique, il est indispensable de revoir nos méthodes, outils et formats.

Cet ouvrage est une nouvelle édition, entièrement revue et augmentée, du manuel « Markdown & vous » publié en 2023 sur le site e-publish¹ de l'Université de Liège. Il propose une réflexion sur une approche durable, ouverte et libre qui redonne la pleine maîtrise dans les activités de rédaction.

Le format Markdown est un langage sémantique simple, lisible par les humains et pérenne grâce à sa simplicité syntaxique. Il permet de revenir à des méthodes qui vont à l'essentiel en faisant les choses dans l'ordre, écrire, illustrer, éditer puis publier, sans perte de temps et d'énergie inutiles.

Le choix de Markdown supprime la dépendance à un logiciel spécifique, comme c'est le cas avec un traitement de texte. Ce choix est aussi une démarche liée à la science ouverte. En travaillant avec Markdown et *Pan-*

1. <https://e-publish.uliege.be/md>

doc on va ouvrir l'ensemble de nos textes (*Open Access*), dans des formats ouverts (*Open Data*) avec des logiciels ouverts (*Open Source*).

Markdown est aussi une solution *Single Source Publishing* (édition à partir d'une source unique), c'est-à-dire qu'avec un (ou plusieurs) même(s) fichier(s) source(s) on peut produire des documents dans différents formats (essentiellement docx, odt, HTML, ePub et pdf).

Plusieurs thèses soutenues récemment, à Bordeaux ([Perret, 2022](#)), à Rouen ([Dehut, 2023](#)) et à Montréal ([Fauchié, 2024](#)), font une large part à Markdown. Ces travaux sont le signe que le monde académique commence à s'intéresser sérieusement à Markdown comme format de rédaction.

L'ouvrage est divisé en quatre parties.

La [première](#) fait le point sur les évolutions du numérique, de la science ouverte, du développement durable et des outils actuels. L'objectif de cette première partie est de présenter la situation avec conviction pour susciter la réflexion et éventuellement induire des changements.

La [deuxième](#) partie présente le Pandoc-Markdown², les choix techniques et les bases de l'écriture avec ce langage.

La [troisième](#) partie présente l'utilisation de Markdown pour transformer la puissance et la complexité de *LaTeX* (un éditeur de textes scientifiques) en un processus accessible au plus grand nombre.

La [quatrième](#) partie propose des exemples et des modèles.

L'ouvrage est complété par des [exercices](#), un [aide mémoire](#) et des [sources complémentaires](#) pour aller plus loin.

Ce manuel rassemble les fonctions de base présentes dans tous les bons manuels comme le manuel de Mailund (2019), « *Introducing Markdown and Pandoc: using markup language and document converter* » ou des fonctions plus « pratiques » glanées au fil de lectures et de recherches.

2. Dans ce manuel, nous utiliserons l'expression "Pandoc-Markdown" pour désigner la variante *Pandoc* de Markdown.

Apprendre Markdown (ou un autre format) ce n'est pas juste apprendre un nouveau langage mais c'est avant tout adhérer à de nouvelles méthodes et à de nouveaux outils, c'est décider de reprendre le contrôle et tourner définitivement le dos à l'impératif fonctionnel si bien décrit par Vitali-Rosati ([2024a](#)).

« La technologie nous promet de nous rendre libres en nous délivrant de toutes les tâches ennuyeuses, triviales, répétitives, purement matérielles, en nous fournissant des 'solutions' de plus en plus performantes, simples, intuitives, rapides et efficaces. Les GAFAM³ nous font rêver d'un monde qui fonctionne tout seul, d'un monde où nous sommes pris-es en charge jusqu'à nos identités par un petit nombre de compagnies privées. Mais cette délivrance se fait au prix d'une perte totale d'autonomie, avec une dépendance complète à ces technologies et à ces entreprises » ([Vitali-Rosati, 2024a](#)).

Adopter Markdown est un choix qui ne va pas nécessairement de soi. Il y a, dans un premier temps, des tâtonnements, des pertes de temps pour résoudre un problème particulier, des renoncements mais, à l'arrivée, le gain de temps et de liberté est évident.

« Il faut reconnaître que la production des documents à partir du Markdown en conjonction de *Pandoc* demande dans un premier temps un certain investissement, en tout cas une motivation quotidienne, et peut-être quelque chose d'une résolution indéfectible. Cet investissement nous semble pourtant la contrepartie indispensable dont on doit s'acquitter dans le dessein de s'approprier, au sens de faire sien, ce qu'est devenu l'écrit dans notre société aujourd'hui » ([Dehut, 2018](#)).

3. Pour *Google, Amazon, Facebook, Apple et Microsoft*.

1. Reprendre le contrôle du numérique

Indépendamment de ce que le traitement de texte nous impose (voir plus loin), basé sur un impératif fonctionnel ([Vitali-Rosati, 2024a](#)) et un capitalisme de surveillance ([Zuboff et al., 2020](#)), liberticides et privateurs, l'industrie informatique, pour exister, nous impose des solutions hi-tech de plus en plus performantes mais aussi de plus en plus chères, gourmandes en ressources énergétiques et en matériel, peu durables et fermées. On les utilise parce que ça marche (en général bien) mais de moins en moins de gens les maîtrisent vraiment...

Les choix technologiques de cette industrie ne sont pas neutres. « Tout programme incorpore des connaissances. Si le programme est privé, il cache, il refuse ces connaissances aux étudiants, et cela va à l'encontre de l'esprit de l'éducation. [...] Le logiciel libre respecte l'esprit de l'éducation en mettant les connaissances à disposition de tous » ([Stallman et al., 2017](#)).

Avec ce numérique et ce qu'il peut nous offrir mais aussi ce qu'il a tendance à nous imposer, il devient nécessaire de s'arrêter un moment et voir comment reprendre le contrôle.

Écologie numérique et développement durable

Il persiste beaucoup de contradictions sur les chiffres liés au numérique. Ce dont nous sommes certains, c'est que près de 80 % de l'empreinte écologique du numérique (eau, énergie, métaux précieux, autres matériaux...) est liée à la fabrication du matériel⁴ et que 80 % de la bande passante sur Internet est utilisée pour des contenus vidéo⁵ ([Petit, 2023](#)).

4. Voir données Ademe : <https://infos.ademe.fr/magazine-avril-2022/faits-et-chiffres/numerique-quel-impact-environnemental/>

5. Essentiellement le streaming. 10 heures de film en haute définition (HD) correspondent à la totalité des contenus textuels de Wikipédia. Une heure de streaming en HD va générer 28 kg de CO₂, soit l'équivalent de 28 km en voiture.

Pour réagir, il faut repenser la matérialité du numérique ([Bordage & Morelle, 2021](#)) et fabriquer moins d'équipements, fabriquer des équipements plus durables (obsolescence programmée⁶) et repenser l'usage de ces équipements (ce qui est nécessaire).

Il est néanmoins indispensable de penser un numérique acceptable ([Derrac, n.d.](#)) qui soit émancipateur et non aliénant, choisi et non subi et soutenable socialement et écologiquement.

Lire à ce propos, le Framabook « Libertés numériques. Guide des bonnes pratiques à l'usage des DuMo⁷ » .

Une certaine sobriété

Nous n'avons pas la main sur l'industrie (matériel et logiciels) mais nous sommes leurs clients.

Un développement plus durable intègre l'utilisateur dans une communauté sans attendre le bon vouloir des propriétaires des produits et des services. Il passe par l'utilisation de logiciels libres et ouverts⁸.

L'utilisation de logiciels libres réduit la menace de l'obsolescence prévisible par défaut de support sur le moyen terme et surtout l'utilisation de technologies, nécessitant toujours plus de puissance, qui raccourcissent la durée de vie du matériel qui reste lui aux mains de l'industrie ([European Union, 2021](#)).

Le choix de logiciels (système d'exploitation Linux et logiciel libres, en général plus légers) permet par ailleurs de conserver plus longtemps un même ordinateur voire de reconditionner un ordinateur plus ancien⁹.

6. <https://www.nobsolete.fr/>

7. <https://docs.framasoft.org/fr/manueldumo/>

8. <https://www.gnu.org/philosophy/free-sw.html>

9. *Microsoft* a annoncé l'arrêt du support de Windows 10 au 14 octobre 2025 et le passage forcé à Windows 11. Windows 11 nécessite une configuration relativement récente (avec un processeur particulier) et les ordinateurs trop anciens (on parle d'ordinateurs construits avant 2021) ne sont pas compatibles.

Il existe de nombreuses distributions Linux¹⁰ (libres et gratuites) qui peuvent être installées sur des ordinateurs plus anciens sans pour autant perdre en confort et en efficacité. Ces distributions contiennent par défaut tout ce dont on a besoin (navigateur, traitement de texte, tableur, éditeur, gestionnaire de courriels ou de fichiers...).

Le principe de l'utilisation de Markdown (un [format texte](#)) est de se passer de logiciels complexes et énergivores¹¹.

Utiliser Markdown, c'est aussi revenir aux principes de *low-tech* (utile, accessible et durable) et de la sobriété numérique¹²

Particularités du Web

Le web (on parle bien de « Web » et pas d' « Internet », inventé 20 ans plus tôt) est lui aussi en grande partie devenu dépendant de quelques grandes entreprises (essentiellement les GAFAM, mais pas que) qui en concentrent l'essentiel et exercent par conséquent un pouvoir démesuré. Il faut désencombrer le web et le rendre plus durable, plus résilient et plus lisible. Il faut arrêter de le surcharger avec des contenus non nécessaires, soumis aux effets de mode et aux aléas des technologies disponibles ([Masutti, 2023](#)).

Avec les développements techniques, les logiciels CMS (Système de gestion de contenus) sont devenus de plus en plus performants mais aussi de plus en plus complexes. Ils sont aussi énergivores, sur les serveurs qui les hébergent et en utilisation du trafic, entre le client (vous) et ces serveurs. Voir par exemple le plaidoyer pour un *small web* ([Balkan, 2020](#)).

-
10. <https://www.sitegeek.fr/article-technologie/7-distributions-linux-legeres-pour-vieux-ordinateurs-en-2022/>
 11. Les traitements de texte utilisent, en permanence, des fonctionnalités de mise en forme complexe, d'insertion d'images et d'autres éléments multimédias qui nécessitent un traitement intensif des ressources informatiques ([Kern et al., 2011](#)).
 12. Voir l'article en ligne "Sobriété numérique (GreenIT)" : <https://www.hygiene-numerique.com/dossiers/sobriete-numerique-greenit-vos-leviers-daction-pour-reduire-votre-empreinte-environnementale/>

La mouvance des sites statiques ([Taillandier, 2016](#)) est un début de solution plus durable et plus légère¹³. Ces sites sont le plus souvent construits, et c'est pourquoi cette question est abordée ici, à partir de fichiers Markdown.

Dans la même veine, mais on dépasse ici le propos de l'ouvrage, pour le Web, l'affichage d'images trop lourdes est inutile. Cela va non seulement ralentir l'affichage mais va aussi générer plus de trafic. La taille idéale d'une image doit correspondre à la résolution d'un écran de PC de bureau (en général 1920 x 1080 pixels) avec une résolution de 72 dpi (points par pouce).

Il est bien évident que chaque lectrice et lecteur de ce livre ne va pas sauver la planète en reprenant le contrôle de son numérique. C'est la conjonction, la conviction, le partage d'expérience, et la collaboration qui vont nous faire avancer vers un futur plus durable.

13. Et donc avec un affichage des pages plus rapide. Durabilité ne rime pas avec lenteur.

Le numérique

Le livre imprimé existe depuis le milieu du XV^e siècle. Le livre numérique a déjà plus d'un demi-siècle. Le premier livre numérique, la première réalisation du projet Gutenberg (le « eText #1 ») de création de versions électroniques gratuites d'œuvres littéraires, a été produit en 1971.

« Au XVI^e siècle, Gutenberg a permis à chacun d'avoir quelques livres. Au XXI^e siècle, le Projet Gutenberg permettrait à chacun de disposer d'une bibliothèque gratuite. L'arrivée d'Internet en 1974, puis du World Wide Web en 1990, a fait de ce projet une réalité » ([Lebert, 2016](#)).

De quoi parle-t-on ?

Alors que l'édition traditionnelle se concentre sur la qualité esthétique du document et de la page imprimée, l'édition numérique va plutôt se concentrer sur la qualité des fichiers utilisés pour la publication dans ses différents formats. Avec l'édition numérique, le travail sur le fond et le travail sur la forme sont clairement distincts.

Pour s'adapter aux nouveaux modes de consultation (à l'écran essentiellement), l'édition numérique doit produire des fichiers de type html (des pages web) et ePub¹⁴ (créés spécifiquement pour les liseuses électroniques).

Néanmoins, alors que les formats html et ePub s'adaptent aux dimensions des appareils, le format pdf¹⁵, paginé¹⁶, est peu adapté à la lecture à l'écran mais reste néanmoins le format privilégié pour le monde scientifique. Il est surtout réservé à l'impression (impression individuelle ou impression de livres et de revues).

14. Un fichier ePub est constitué d'un ensemble de pages html regroupées dans un fichier compressé.

15. Le format pdf (Portable Document Format) est un langage de description de page créé par Adobe en 1992. En 2008, c'est devenu un format ouvert (norme ISO 32000-1:2008).

16. Pour un document au format pdf, le découpage en pages est figé alors que pour un document numérique, au format ePub, le découpage varie en fonction des options d'affichage.

Ce changement de point focal (qualité des fichiers) permet de proposer des processus plus ouverts. Ils sont basés sur les principes de l'édition à source unique (*Single Source Publishing*¹⁷), c'est-à-dire l'utilisation d'un fichier unique (ou un ensemble de fichiers) pour produire un même contenu dans différents formats. Markdown, utilisé avec *Pandoc* et *LaTeX*, est particulièrement approprié pour l'édition à source unique.

La démarche de la publication numérique consiste dès lors à remplacer une pseudo-structuration esthétique par manipulation des attributs typographiques (majuscules, tailles, gras, italiques, soulignés ou styles aléatoires des traitements de texte pour signifier la structure d'un texte) par une structuration sémantique stricte permettant aux logiciels utilisés de « comprendre » la structure d'un texte et de la répercuter esthétiquement et fonctionnellement.

Produire un document pdf avec un traitement de texte (*Word*) ou avec un logiciel de mise en page (*InDesign*) n'est donc pas de l'édition numérique.

Il faut par ailleurs considérer que l'édition numérique et le web sont liés. D'un point de vue technique, leurs langages de programmation sont identiques, et au niveau de leurs procédés de conception, leurs workflows sont comparables ([Usclat, 2017](#)).

Parce que l'édition numérique a remodelé la chaîne éditoriale traditionnelle, permettant une structuration des contenus pour des formes de diffusion multiples, et non plus simplement pour un livre imprimé, l'édition numérique a transformé le rôle de l'éditeur.

Les éditeurs, qui restent les garants de la validité des contenus et de la qualité esthétique des documents produits, doivent progressivement envisager tous les aspects techniques de la publication (indexation, liens hypertextes, espaces de communication et multimédia) et organiser leur production et leur diffusion pour plusieurs formats.

Ils sont par ailleurs amenés à travailler plus étroitement avec les auteurs pour les aider à comprendre les possibilités offertes par la publication numérique et les intégrer dans le travail de création. Ils doivent dès lors

17. <https://ssp.digitaltextualities.ca/>

apprendre à mobiliser de nouvelles compétences qui ne faisaient pas partie de leurs savoir-faire traditionnels.

Cette évolution impose une réflexion sur la mission et la place de l'éditeur dans le panorama numérique ([Boismenu & Beaudry, 2002](#) ; [Dufrasne, 2024](#) ; [Epron & Vitali Rosati, 2018](#)).

Le plus souvent, les éditeurs s'associent à d'autres prestataires (dont les bibliothèques et les spécialistes du web) pour faire le grand saut du numérique.

L'édition numérique ouverte dans la démarche académique

L'édition numérique est un des fondements sur lequel repose le modèle du libre accès. Elle facilite une diffusion plus large et plus rapide des connaissances scientifiques tout en respectant les droits d'auteur. Ce rapport entre les deux est essentiel pour promouvoir une science ouverte et accessible à tous.

L'édition scientifique est soumise à plusieurs contraintes. Fauchier et al. ([2022](#)) ont listé les éléments indispensables à retenir :

1. l'appareil critique : notes de bas de page, citations, bibliographies, figures, encarts, etc. ;
2. les métadonnées : données sur les données, comme le titre, le sous-titre, les auteurs, les identifiants standards (ORCID, Wikidata, etc.) ;
3. la structuration du texte : la structuration sémantique des textes est une nécessité pour la diffusion numérique ;
4. les données bibliographiques : avec des références structurées ;
5. des formats spécifiques : articles, livres, actes de conférence, etc. ;
6. l'évaluation par les pairs : une circulation et une validation complexes des textes ;
7. le rejet du modèle *publish or perish* : publier beaucoup mais lentement.

L'utilisation de fichiers au format « texte », balisés avec Markdown et transformés avec *Pandoc*, repose sur le principe de l'édition numérique ouverte et va rencontrer ces contraintes.

Le traitement de texte

Écrire

Sauf exceptions, l'écriture est le plus souvent numérique. Elle suppose des compétences nouvelles ([Fauchié, 2020](#) ; [Ruiz, 2020](#)).

Écrire, rédiger, c'est bien évidemment respecter un certain nombre de règles syntaxiques, orthographiques, lexicales mais c'est aussi organiser et structurer ses idées, prendre le temps qu'il faut et l'organiser.

Écrire est une démarche complexe qui nécessite de se poser la question du format et des outils.

Traditionnellement et à cause du monopole de son éditeur, c'est le logiciel *Word* qui est majoritairement utilisé pour la rédaction de textes.

Word est techniquement orienté vers le formatage et la mise en page de textes. Alors que les journalistes, les enseignants et même les poètes ne se soucient pas des polices de caractères, des guillemets et des marges. Ils ont été contraints de l'utiliser non pas parce que c'est la meilleure application pour écrire mais parce que les universités, les rédacteurs en chef, les éditeurs proposent et attendent le format .docx. Pour l'essentiel, les allers-retours entre les auteurs et les éditeurs se font par le biais de la fonction « Suivi des modifications » de *Word* ([Reichenstein, 2025](#)).

Du côté des traitements de texte, il y a plusieurs alternatives libres comme la suite bureautique *LibreOffice*¹⁸, une alternative libre. Cette suite propose des outils comparables, fonctionnant sur toutes les plateformes (*Windows, Mac ou Linux*).

Les traitements de texte sont cependant des programmes complexes avec de multiples fonctions dont certaines sont mal, rarement, voire jamais utilisées.

À l'origine, le traitement de texte avait pour objectif de remplacer la machine à écrire pour produire des pages imprimées. Ce n'est pas un

18. Attention à ne plus utiliser *OpenOffice* qui n'est plus mis à jour depuis 2015.

outil sémantique, « la qualification des différents types de texte n'est pas mise en avant, comme les niveaux de titre (titre principal, sous-titres, etc.). C'est le rendu graphique de ces éléments qui prime » ([Fauchié, 2018](#)).

Pour le traitement de texte, la dimension de l'architecture du texte d'un document, sauf à utiliser correctement les « styles » proposés par ces outils, est accessoire. C'est ce que l'on voit qui compte. Ce sont les attributs typographiques (Titre en gras, en italique, dans une taille de caractère plus grande...) qui structurent le texte.

Par ailleurs, l'orientation *WYSIWYG*¹⁹, devenue la norme pour les traitements de texte, est gourmande en ressources informatiques et souvent source de lenteur parce qu'elle ajuste en permanence l'affichage. Elle donne un sentiment d'autonomie alors que le système est totalement opaque et que l'accès au code a progressivement disparu²⁰ ([Vitali-Rosati, 2024b](#)).

Il est souvent ardu de comprendre et modifier certaines options de mise en page imposées par le logiciel. De plus, les documents qu'ils produisent maintenant sont lourds et ne sont plus adaptés aux consultations les plus courantes, à l'écran ([Dehut, 2018](#)).

Il n'est pas possible d'imaginer que la « numérisation » de l'écriture pour rédiger un courrier, écrire un texte ou préparer un diaporama ne transforme pas profondément les pratiques d'écriture ([Crouzet, 2016](#)).

Logiciels propriétaires

Le format propriétaire des traitements de texte (comme *Word* de *Microsoft* ou *Pages* d'*Apple*), basé sur la stratégie du profit²¹, est aussi un problème.

Nous n'avons aucune garantie de pouvoir ouvrir un document sans la licence d'utilisation ou avec une version plus récente (tout particu-

19. *What You See Is What You Get*, autrement dit, ce que vous voyez à l'écran est ce que vous aurez comme résultat imprimé.

20. Les traitements de texte comme *WordStar* ou *WordPerfect* permettaient d'accéder au code et de contrôler la mise en page.

21. Voir : <https://ploum.net/histoire-du-logiciel-entre-collaboration-et-confiscation-des-libertes/>

lièrement pour *Word*, même avec le nouveau format *docx* qui donne théoriquement la possibilité d'ouvrir les fichiers xml qui constituent le document).

Plus spécifiquement pour *Microsoft*, en lisant attentivement les conditions générales (*Microsoft Privacy Statement*), on découvre que le capitalisme de surveillance ([Zuboff et al. 2020](#)) est omniprésent :

- les clés de chiffrement (*BitLocker*) sont envoyées à *Microsoft* ;
- un document *docx/pptx* ouvert envoie son contenu aux serveurs *Microsoft* ;
- avec le nouvel *Outlook*, *Microsoft* a tous vos mails et tous vos mots de passe ;
- *Edge* envoie les URLs visitées à *Microsoft* ;
- *OneDrive* récupère par défaut tous vos fichiers ;
- *Recall (Windows 11)* enregistre l'écran (plusieurs fois par minute) pour alimenter l'IA de *Microsoft*.

Microsoft propose un écosystème très complet (*Word, Outlook, Teams, PowerPoint, OneDrive...*) qui explique en partie son succès mais aussi les difficultés à s'en passer.

Vitali-Rosati ([2024b](#)) parle de « La glu Microsoft » !

Google propose aussi un écosystème bureautique complet, gratuit²² et qui crée également une dépendance. Disponible via *Google Drive*, avec un logiciel de traitement de texte (*Docs*), un tableur (*Sheets*), un logiciel de présentation (*Slides*) et un formulaire (*Forms*), cet ensemble permet de travailler à plusieurs sur un même document. Le document est enregistré en ligne et est accessible à partir de n'importe quel ordinateur équipé d'un navigateur Web.

On ignore cependant l'utilisation qui est faite des données que nous confions à ces outils.

22. L'adage habituel est "si c'est gratuit, c'est toi le produit", il faut néanmoins le nuancer et plutôt dire : "c'est toi qui produis" voir : <https://www.librealire.org/si-c-est-gratuit-c-est-toi-qui-produis>

Google Docs (le traitement de texte), après avoir intégré *Zotero* (logiciel *Open Source*) pour la gestion bibliographique, intègre maintenant Markdown dans ses formats, ce qui suppose une tentative de réappropriation de Markdown par les grandes entreprises du numérique.

Dans le même registre, *Microsoft* est, depuis 2018, propriétaire de *GitHub*²³, l'outil utilisé par la majorité des développeurs de logiciel et qui utilise Markdown pour tous ses fichiers texte.

Écriture numérique

L'écriture numérique doit donc être structurée. Cette structure doit à la fois organiser la pensée et permettre d'exprimer cette organisation. Elle doit aussi respecter les standards ouverts pour la diffusion et le partage de cette écriture.

Le principe de base est de séparer l'écriture et la mise en forme pour ensuite exporter son texte dans n'importe quel format, pouvoir l'éditer avec n'importe quel outil et sur n'importe quelle plateforme (*Windows, OSX, Linux...*), avec une garantie de pérennité.

Avec les nouveaux modes de consultation (à l'écran), le travail doit reposer sur les principes de l'édition numérique (voir plus haut) pour produire des documents numériques. Le balisage graphique (pseudo-structuration esthétique) produit par un traitement de texte ne convient donc pas pour la production de textes numériques structurés. Ces formats nécessitent une structure (sémantique) stricte pour un résultat de qualité.

Ces documents numériques doivent aussi contenir la description du document. On parle de métadonnées. La structure de ces métadonnées fait l'objet de normes et permet le partage d'informations entre les outils et l'alimentation automatique des bases de données.

Il faut revenir aux fondamentaux et donc ne pas nécessairement essayer de reproduire coûte que coûte tout ce qui est possible avec un traitement de texte classique (dont on peut souvent se passer).

Les traitements de texte et les modes WYSIWYG masquent les solutions possibles pour l'écriture numérique. Pour l'écriture numérique, il faut

23. <https://fr.wikipedia.org/wiki/GitHub>

choisir des solutions simples, en termes d'usage. Des solutions intégrant une véritable structuration sémantique visible par l'utilisateur ([Fauchié, 2018](#)).

Les formats

Les fichiers

Pour continuer à convaincre les lectrices et les lecteurs de ce manuel, si c'est encore nécessaire, il faut se pencher sur les formats d'enregistrement des différents logiciels.

Les traitements de texte, posent de nombreux problèmes à la fois de lisibilité mais aussi de durabilité.

Le petit texte ci-dessous est un exemple créé avec *LibreOffice* (avec deux niveaux de titres, de l'italique, du gras et une liste simple). Les différentes « versions » présentées ci-dessous parlent d'elles-mêmes.

Titre niveau 1

Titre niveau 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi est urna, sodales vitae maximus et, feugiat vel ipsum.

Proin ut pulvinar turpis. Aliquam nec purus arcu. Cras accumsan imperdiet finibus.

Etiam at velit bibendum, suscipit quam in, porttitor ligula.

Aliquam nunc nunc, commodo sed sapien non, ornare volutpat ex. Mauris sed dui et augue tincidunt tempor ac pulvinar massa. Cras vulputate elit a molestie tincidunt. Sed vitae fringilla elit, sed facilisis ipsum. Duis semper sapien quam, a interdum purus posuere eget. Quisque hendrerit, massa sit amet interdum porttitor, elit lacus tempor sapien, ut sagittis sapien eros nec sem. Curabitur convallis interdum turpis.

Titre niveau 2

Aliquam quis auctor nisi. In consequat mattis erat, ut luctus felis. Sed convallis turpis lorem, in placerat eros mattis at. Vestibulum varius commodo turpis in placerat. Cras id odio lectus. Donec facilisis magna ac lacus placerat, eu posuere neque aliquet. In ut auctor tortor, sit amet placerat magna. Liste :

- Pellentesque cursus urna id risus faucibus volutpat ;
- Sed quis tellus turpis. Nullam tempor, nibh id venenatis bibendum, ante ipsum ultrices nunc, ac lobortis libero leo consectetur libero ;
- Curabitur ut dictum ipsum.

exemple « fini » au format pdf

RTF

Dans le format *rtf* (*Rich Text Format*, le format d'échange entre logiciels de traitement de texte) exporté depuis le traitement le texte (ici *LibreOffice*), ouvert avec un simple éditeur, voici ce que contient le fichier.

NB : pour éviter de présenter une page trop longue, il n'y a ici qu'un extrait du fichier (qui fait 75 lignes).

```
{\rtf1\ansi\deff4\adeflang1025
{\fonttbl{\f0\froman\prq2\charset0 Times New Roman;}{\f1\froman
\prq2\charset2 Symbol;}{\f2\fswiss\prq2\charset0 Arial;}{\f3
\froman\prq2\charset0 Liberation Serif{\*\falt Times New Roman}
;}{\f4\froman\prq2\charset0 Times New Roman;}{\f5\fswiss\prq2
\charset0 Arial;}{\f6\fnil\prq0\charset0 StarSymbol{\*\falt
Arial Unicode MS;}{\f7\modern\prq1\charset0 Courier New;}{
\f8\fnil\prq2\charset0 Lucida Sans Unicode;}{\f9\
.....
Aliquam quis auctor nisi. In consequat mattis erat, ut luctus felis. Sed
convallis turpis lorem, in placerat eros mattis at. Vestibulum varius
commodo turpis in placerat. Cras id odio lectus. Donec facilisis magna
ac lacus placerat, eu posuere neque aliquet. In ut auctor tortor, sit
amet placerat magna. Liste\~;}\par \pard\plain \s33\loch\sb86\sa86
{\listtext\pard\plain \rtlch\af6\afs18 \ltrch\hich\af6\loch\af6\fs18
\dbch\af6 \u8226'95\tab}\ilvl0\ls2 \li720\ri0\lin720\ri0\fi-360\ql\
sb0\sa0\ltrpar{\loch Pellentesque cursus urna id risus faucibus
volutpat\~;}\par \pard\plain \s33\loch\sb86\sa86{\listtext\pard\plain
\rtlch\af6\afs18 \ltrch\hich\af6\loch\af6\fs18\dbch\af6 \u8226'95\tab}
\ilvl0\ls2 \li720\ri0\lin720\ri0\fi-360\ql\sb0\sa0\ltrpar{\loch Sed
quis tellus turpis. Nullam tempor, nibh id venenatis bibendum, ante ipsum
ultrices nunc, ac lobortis libero leo consectetur libero\~;}\par \pard
\plain \s33\loch\sb86\sa86{\listtext\pard\plain \rtlch\af6\afs18 \ltrch
\hich\af6\loch\af6\fs18\dbch\af6 \u8226'95\tab}\ilvl0\ls2 \li720\ri0
\lin720\ri0\fi-360\ql\sb0\sa0\ltrpar{\loch Curabitur ut dictum ipsum.}
\par }
```

HTML

Avec le format HTML, on a un format « texte » (balisé). En important le fichier dans un simple éditeur, on voit que le contenu est plus lisible mais il n'y a pas de traitement sémantique de la structure. Le titre de niveau 1 est mis en évidence par des codes de mise en page (**span style='font-size: x-large;' et strong**). Ce code aurait du être : **<h1>Titre niveau 1</h1>**.

Cette structuration est seulement visuelle et aucun système informatique ne sera capable d'identifier ces titres comme des titres.

Dans un traitement de texte il est bien évidemment possible d'utiliser des styles liés aux titres et à leur niveau cependant, non seulement peu d'auteurs utilisent cette possibilité, mais également, ces styles sont propres aux logiciels utilisés.

```
<html><head>
<meta charset='UTF-8'><meta name='viewport' content='width=device
-width initial-scale=1'>
<title>lorem</title>
</head>
<body>
<span style="font-size: x-large;"><strong> Titre niveau 1</strong>
</span><span style="font-size: large;"> Titre niveau 2</span><p>
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi est
urna, sodales vitae maximus et, feugiat vel ipsum.</p>
<p><i>Proin ut pulvinar turpis. Aliquam nec purus arcu. Cras
accumsan imperdiet finibus.</i></p>
<p><strong>Etiam at velit bibendum, suscipit quam in, porttitor
ligula. </strong></p>
<p>Aliquam nunc nunc, commodo sed sapien non, ornare volutpat ex.
Mauris sed dui et augue tincidunt tempor ac pulvinar massa. Cras
vulputate elit a molestie tincidunt. Sed vitae fringilla elit, sed
facilisis ipsum. Duis semper sapien quam, a interdum purus posuere
egit. Quisque hendrerit, massa sit amet interdum porttitor, elit
lacus tempor sapien, ut sagittis sapien eros nec sem. Curabitur
convallis interdum turpis.</p>
.....
```

Markdown

Avec un fichier « texte » balisé avec Markdown, les différents éléments du texte sont identifiables et un système informatique sera capable d'identifier les titres, les éléments de mise en forme ou les listes.

Non seulement ce texte est lisible par un système informatique, mais il est aussi lisible et compréhensible par tout le monde. Le code (standardisé) est léger et facile à partager.

Titre niveau 1

Titre niveau 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi est urna, sodales vitae maximus et, feugiat vel ipsum.

Proin ut pulvinar turpis. Aliquam nec purus arcu. Cras accumsan imperdiet finibus.

Etiam at velit bibendum, suscipit quam in, porttitor ligula.

Aliquam nunc nunc, commodo sed sapien non, ornare volutpat ex. Mauris sed dui et augue tincidunt tempor ac pulvinar massa. Cras vulputate elit a molestie tincidunt. Sed vitae fringilla elit, sed facilisis ipsum. Duis semper sapien quam, a interdum purus posuere eget.

Quisque hendrerit, massa sit amet interdum porttitor, elit lacus tempor sapien, ut sagittis sapien eros nec sem. Curabitur convallis interdum turpis.

Titre niveau 2

Aliquam quis auctor nisi. In consequat mattis erat, ut luctus felis. Sed convallis turpis lorem, in placerat eros mattis at. Vestibulum varius commodo turpis in placerat. Cras id odio lectus. Donec facilisis magna ac lacus placerat, eu posuere neque aliquet. In ut
.....

Taille des fichiers

La taille des fichiers est aussi un aspect intéressant à observer. Pour le court extrait présenté ci-dessus, voici ce que cela donne :

 lorem.md	1,3 ko
 lorem.html	1,6 ko
 lorem.odt	10,2 ko
 lorem.docx	10,7 ko
 lorem.rtf	11,3 ko
 lorem.pdf	97,9 ko

Taille des différents fichiers

Les fichiers Markdown et html sont des fichiers au format texte avec balisage simple. Ce sont les plus légers.

Les fichiers bureautiques (.odt, .docx et .rtf) sont eux presque dix fois plus « lourds ».

Le document final au format pdf est particulièrement volumineux.

Dans le cas de documents plus longs, indépendamment de leur complexité, cette différence est loin d'être négligeable en termes de stockage et de transmission.

Le format texte

Lorsque l'on parle de format texte, on parle traditionnellement de fichiers avec l'extension `.txt` (voir le site *Plain Text Project*²⁴ à propos de ce format). Ils contiennent une suite de caractères alphanumériques et de signes typographiques et sont lisibles avec un simple éditeur.

Balisages

Aucun balisage

Le format sans balisage est essentiellement utilisé pour prendre des notes.

Balisage léger

Un fichier texte peut contenir un balisage léger qui ne perturbe que peu la lecture. Ci-dessous, les trois principaux langages de balisage léger. La troisième colonne montre l'exemple du balisage utilisé pour mettre du *texte en italique*.

langage	création	exemple	formats des documents créés
<i>LaTeX</i>	1982	<code>\emph{texte}</code>	pdf
HTML	1993	<code>texte</code>	pages Web
Markdown	2004	<code>*texte*</code>	tous types

24. <https://plaintextproject.online/index.html>. Ce site n'est plus mis à jour depuis 2022 mais reste une référence sur le sujet.

Le format texte, avec balisage léger²⁵, en plus d'être facilement lisible, permet :

- de séparer l'écriture et la mise en forme ;
- d'exporter son texte (dans n'importe quel format avec *Pandoc*) ;
- de faciliter le partage ;
- de l'éditer avec n'importe quel outil et sur n'importe quelle plateforme (*Windows, Mac, Linux...*) ;
- d'avoir une garantie de pérennité ;
- de travailler hors ligne.

Avec le format texte on peut aussi envisager une orientation *WYSIWYM* (*What You See Is What You Mind*²⁶)

Balisage plus lourd

Pour mémoire, le langage de balisage XML, créé en 1999, structure les fichiers *rss, xsl, docbook, tei, json, jats, bits...* dont certains sont utilisés quotidiennement.

Bien que lisibles avec un simple éditeur, le balisage des fichiers XML est plus complexe, avec une syntaxe très stricte.

Notez que le code source des logiciels (s'ils sont libres, *Open Source*, ils sont accessibles) repose aussi sur du format texte. Il contient une série d'instructions, de commentaires, de variables, qui sont lisibles avec un éditeur et compréhensibles par un logiciel qui va interpréter son contenu pour le transformer en programme.

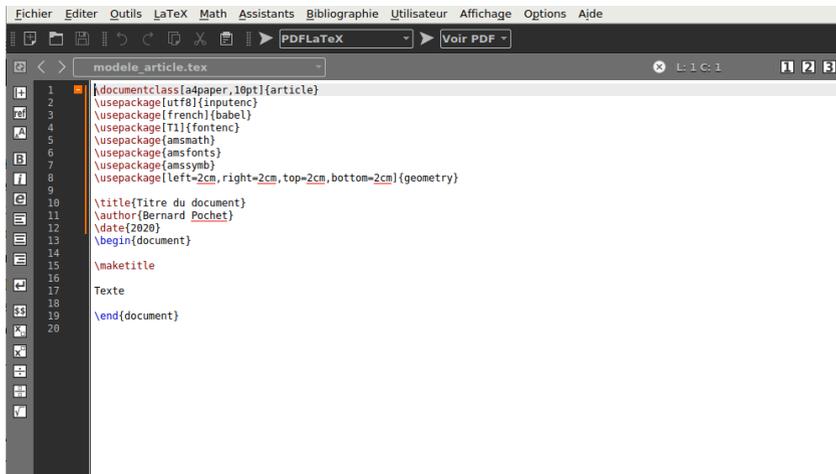
25. Il existe d'autres formats avec balisage léger. Les plus régulièrement cités sont : *asciidoc, Setext, atx, Textile, reStructuredText, Grutatext* et *EtText*.

26. Voir : <https://economicsfromthetopdown.com/2020/12/10/why-and-how-i-write-scientific-documents-in-plain-text/>

LaTeX, séparer fond et forme

LaTeX, le langage de balisage léger créé il y a plus de 40 ans, résout le problème du codage et de la structure. Plusieurs logiciels libres permettent d'éditer des fichiers LaTeX (le plus connu et le plus simple d'utilisation est *TeXEdit*). Il sépare le fond et la forme et donne le contrôle total sur le texte.

C'est le logiciel qui se charge de la mise en page finale en fonction de vos instructions ([Pierre, 2017](#)). Il est cependant difficile à maîtriser. Il est surtout adopté par certaines professions (ingénieurs, mathématiciens ou informaticiens) mais pas uniquement, voir par exemple le guide de Rouquette ([2012](#)) pour les Sciences humaines.



```

Fichier  Editer  Outils  LaTeX  Math  Assistants  Bibliographie  Utilisateur  Affichage  Options  Aide
modele_article.tex
L: 1 C: 1
1  \documentclass[a4paper,10pt]{article}
2
3  \usepackage[utf8]{inputenc}
4  \usepackage[french]{babel}
5  \usepackage[T1]{fontenc}
6  \usepackage{amsmath}
7  \usepackage{amsfonts}
8  \usepackage{amssymb}
9  \usepackage[left=2cm,right=2cm,top=2cm,bottom=2cm]{geometry}
10
11  \title{Titre du document}
12  \author{Bernard Pochet}
13  \date{2020}
14  \begin{document}
15
16  \maketitle
17
18  Texte
19
20  \end{document}

```

Un écran TexMaker

Écriture sans distraction

Pour se concentrer sur la rédaction, il faut nécessairement faire référence au point de départ, à l'écriture avec une feuille papier et un crayon,

sans distraction. *LaTeX*, avec son codage particulier, n'est pas un modèle d'écriture sans distraction.

Une solution moins complexe est assurément l'utilisation du format Markdown (qui fait l'objet de ce manuel et qui a par ailleurs été utilisé pour le créer). C'est un langage de balisage particulièrement léger, simple à lire, à écrire et à mémoriser.

L'utilisation de Markdown avec n'importe quel logiciel d'édition (spécialisé ou non) permet aux auteurs de se concentrer sur la rédaction sans se préoccuper de la mise en page finale du document ([Dehut, 2018](#)).



Le choix du bon outil ?

Avec ses évolutions, en utilisant les extensions *Pandoc* ([Bonjour, 2014](#)), Markdown n'a plus grand-chose à envier à *LaTeX* que *Pandoc* va utiliser pour transformer les fichiers Markdown en documents pdf paginés.

Historique

Langage

Markdown est donc un langage de balisage léger. Il a été créé par Aaron Swartz²⁷ et John Gruber en 2004 (voir le site de référence maintenu par John Gruber²⁸) afin de générer facilement du code HTML pour le web.

Le nom « Markdown » est un clin d'œil de ses créateurs qui détournent l'expression « *markup language* », qui se traduit par « langage de balisage » en remplaçant « *up* » par « *down* ». Ce néologisme souligne le caractère particulièrement léger et simplifié de Markdown.

La syntaxe de Markdown a été inspirée par plusieurs autres langages utilisés pour la génération de documents HTML dont *Setext* (1992), *Textile* (2002) et *reStructuredText* (2002) mais surtout le format du texte brut des courriels.

La syntaxe originale de Markdown est uniquement composée de caractères typographiques moins fréquents (par exemple le choix de # pour les titres). Elle est prévue pour être facile à lire et à écrire et ne correspond qu'à un petit nombre de balises HTML.

Markdown n'a pas été inventé pour remplacer le langage HTML, ni même pour s'en rapprocher, mais pour faire gagner du temps aux rédacteurs de pages Web.

Pour transformer un fichier Markdown en fichier HTML avec la méthode historique, on peut utiliser *markdown.pl* (une application *PERL*²⁹ créé par J. Gruber). On peut retrouver la source ici : <https://github.com/lau->

27. Aaron Swartz a marqué l'histoire d'Internet. Informaticien, écrivain, militant politique et hacktivist américain, il plaidait, entre autres, pour un accès libre à la littérature scientifique. Il se suicide le 11 janvier 2013 à l'âge de 26 ans, un mois avant la tenue d'un procès fédéral pour fraude électronique. Il risquait 35 ans de prison pour avoir téléchargé 4,8 millions d'articles scientifiques pour les mettre en libre accès sur l'*Open Library*.

28. <https://daringfireball.net/projects/markdown/>

29. Cette opération est réservée aux geeks ou aux personnes qui maîtrisent le mode "Terminal".

[riro/markdown/tree/master](#). Il existe aussi une version en ligne³⁰ de l'outil de transformation.

Évolution

Depuis sa création, Markdown a été étendu avec, entre autres, le **Markdown Extra** qui ajoute des tableaux, des abréviations, des notes en bas de page et des définitions au **CommonMark**³¹ (le nom en définitive donné au format original). Le *Markdown extra* est par exemple utilisé pour les Wikis (comme *wikipedia*) ou les CMS³² (comme *Drupal*).

Le **GitHub Markdown** a permis l'ajout de 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 ce manuel) est plus adapté pour le 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.

Pandoc

Avec l'apparition de *Pandoc*³³ en 2006, Markdown peut être utilisé pour produire d'autres types de documents (que le HTML), dont le pdf³⁴. Grâce à l'utilisation de *Pandoc* qui fait appel à *LaTeX*, il est possible de produire des documents avec une qualité identique à celle des documents produits directement avec *LaTeX* ([Pierre, 2017](#)).

30. <https://daringfireball.net/projects/markdown/dingus>

31. <https://commonmark.org/>

32. Content Managment Systems.

33. <https://pandoc.org>

34. C'est depuis la version de 0.3 de mai 2007 que *Pandoc* permet la transformation de Markdown en fichiers pdf paginés. *Pandoc* a été créé et est maintenu par John MacFarlane.

Utiliser Markdown ne doit pas être considéré comme une complication supplémentaire mais fait au contraire partie d'un mouvement *Low-Tech* (utile, accessible et durable).

Pour rappel, il fait également partie de ces solutions *Single Source Publishing*, c'est-à-dire qu'avec un (ou plusieurs) même(s) fichier(s) source(s) on peut produire des documents dans différents formats (pdf, HTML, ePub...).

Limites, normes et standards

Depuis sa création, et encore actuellement, Markdown fait face à de nombreuses critiques dont essentiellement la multiplication des variantes, l'absence de normes liées à Markdown et sa légèreté, voire sa simplicité.

« Si l'argument de la simplicité revient dans toutes les discussions sur les avantages du Markdown, à l'opposé, toutes les critiques convergent vers les difficultés inhérentes à sa non-standardisation. Avec l'évolution exponentielle de la publication numérique, en ligne ou non, le Markdown devient populaire comme langage d'écriture et de publication, quelle que soit la spécialisation du rédacteur : les différentes variantes du Markdown, souvent nommées *flavor* (parfum, ou saveur), de manière assez suggestive, entérinent cette évolution. En l'absence d'une direction donnée par son fondateur, le Markdown ouvre la porte à une multitude de variantes qui l'éloignent autant de l'interopérabilité, chacune ajoutant ses spécificités à la norme déjà bancale de l'original. L'un des développeurs les plus avancés dans l'évolution du Markdown a même mis au point, en guise de test, une application permettant de mesurer les différences entre les 22 versions connues du langage, qu'il nomme *babelmark*, en référence à l'épisode biblique » ([Mpondo-Dicka, 2020](#)).

Variantes

En matière de multiplication des variantes, en effet, en 2013, Martin Fenner³⁵ parlait déjà de 31 *flavors*, saveurs, pour Markdown (« *31 flavors is great for ice cream but not markdown* »). Mpondo-Dicka (2020) de son côté, cite 22 variantes. Depuis, BabelMark³⁶ décline jusqu'à 35 variantes.

Les différences sont parfois minimales d'une variante à l'autre. Elles sont créées par des communautés afin d'adapter Markdown à leurs besoins et

35. Sur son blog : [https://blog.front-matter.io/posts/](https://blog.front-matter.io/posts/what-flavor-is-scholarly-markdown/)

36. <https://babelmark.github.io/?text=%23+titre%0A>

à leurs outils. Les quatre déclinaisons du *CommonMark*³⁷ présentées plus haut restent les principales variantes.

La syntaxe de base qui s'applique aux titres, aux listes, aux formatages simples (gras, italique...), aux liens et aux images reste toujours la même dans toutes les variantes.

Normalisation

En matière de standardisation, malgré l'intérêt indéniable de Markdown, il n'y a toujours pas de norme qui y est associée. Néanmoins, pour Fauché (2024) « Markdown s'est imposé comme une forme de standard de fait ».

Le *CommonMark*³⁸, dont la rédaction des spécifications techniques : <https://spec.commonmark.org/>³⁹ a débuté en 2014 (dernière mise à jour en janvier 2024), est loin d'être généralisé et ce n'est d'ailleurs toujours pas une norme.

L'idée est aussi que Markdown puisse être uniquement considéré comme un outil, orienté structuration sémantique de contenus, pour la création de pages web et de documents dans des formats divers et variés.

C'est donc un format « technique » qui permet le développement de technologies. Dès lors, considérant que HTML, pdf, ePub, docx, odt et autres font l'objet de normes et standards, pour d'aucun, le problème ne se pose pas vraiment.

Néanmoins, en mars 2016, dans le but de standardiser le langage, deux RFC (appel à commentaires) ont été publiés (source *Wikipedia*⁴⁰) :

- RFC 7763, qui introduit le type MIME text/markdown à partir de la

37. *Markdown Extra*, *GitHub Markdown*, *Vanilla Flavored Markdown* et *MultiMarkdown*, le *Pandoc-Markdown* étant une déclinaison de ce dernier.

38. <https://commonmark.org/>

39. Ce travail destiné à corriger les erreurs et imprécisions du Markdown d'origine est réalisé par une petite équipe, soutenue par GitHub, Stack Overflow, Stack Exchange, Reddit, Discourse et LinuxFr.org et pilotée par John MacFarlane, le créateur de Pandoc. John Gruber n'y est pas associé.

40. <https://fr.wikipedia.org/wiki/Markdown>

variante originale de Markdown ;

- RFC 7764, qui répertorie des variantes *MultiMarkdown*, *GitHub Flavored Markdown* (GFM), *Pandoc*, *CommonMark*, *Markdown Extra* et autres.

Ici encore, le travail est en cours.

Simplicité

Enfin, à propos de la légèreté et de la simplicité de Markdown, il faut revenir au point de départ et relire la présentation de John Gruber⁴¹ en 2004 :

« *Markdown is a text-to-HTML conversion tool for web writers. Markdown allows you to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML (or HTML)* » .

Pour Mpondo-Dicka (2020) :

« Le Markdown correspond donc à un certain nombre de besoins d'écriture que son développeur a implémenté dans un script : écrire, relire et mettre en forme sans effort des articles publiés en ligne au format HTML ; c'est pourquoi Gruber néglige certains aspects de l'écriture qui ne lui sont pas utiles (notes, table des matières...). Plus encore, il n'a que faire de la standardisation de son langage : le HTML est le standard, le Markdown n'est qu'un outil pour en faciliter l'écriture » .

Cette simplicité était donc bien un objectif en soi. Si la « critique » concerne le *CommonMark*⁴², pourquoi pas mais depuis lors et particulièrement quand *Pandoc*, associé à *LaTeX*, est apparu, cette critique a perdu tout son sens et les documents produits avec ces outils n'ont plus rien à envier à ceux des traitements de texte, lorsqu'ils sont bien utilisés.

41. <https://daringfireball.net/projects/markdown/>

42. <https://commonmark.org/>

Pour voir ce qui est possible, consultez par exemple le rapport du projet AcOBE⁴³, dont le sujet est proche de celui du présent manuel et dont le processus d'édition et de publication est basé sur Markdown avec les logiciels *Pandoc* et *LaTeX*.

43. https://orbi.uliege.be/bitstream/2268/312724/1/rapport_final_AcOBE.pdf

2. Le Pandoc-Markdown

Toutes les fonctions présentées dans cette partie illustrent l'utilisation de la variante *Pandoc* de Markdown⁴⁴. Cette partie reprend les « commandes » (ou « balises ») de base.

La [partie suivante](#) décrit comment tirer le meilleur parti de *LaTeX* pour produire des documents plus complexes.

44. <https://pandoc.org/MANUAL.html#pandocs-markdown>

Principe

Avec Markdown, il y a de nombreuses fonctions proposées dans les traitements de texte impossibles à reproduire. Le fait qu'elles existent n'en fait d'ailleurs pas des fonctions indispensables. L'objectif est bien de proposer une méthode simple, ouverte et durable pour rédiger et produire de beaux documents numériques ([Audet et al., 2018](#)).

On l'a vu plus haut, le principe de base est de séparer le fond de la forme. Le fond, c'est le texte, une suite de caractères, qui se lit en dehors de toute mise en forme. La forme, c'est la traduction graphique, enrichie, du texte. Elle peut être différente d'un format à l'autre.

Pratiquement, il faut créer un fichier texte avec n'importe quel éditeur (voir les outils) et l'enregistrer avec l'extension « .md » (voir un modèle de document à télécharger⁴⁵).

Autour du fond et du texte, il y a cinq éléments à prendre en compte :

1. la structure du document (parties, chapitres, titres, paragraphes et listes) ;
2. la mise en forme du texte (gras, italique, exposant, indice...) ;
3. les compléments du texte (notes, liens, figures, tableaux, citations et bibliographie) ;
4. les éléments de mise en forme du document ;
5. la description du document (les métadonnées).

Les instructions de mise en forme du document (langue du document, taille du papier, taille des caractères, présence ou non d'une table des matières...) et la description du document, ses métadonnées, sont intégrées en en-tête du document, dans une partie appelée YAML.

45. <https://opensolution.be/docs/exemple.zip>

La structure du document

La structuration d'un document est un point essentiel dans sa qualité. Le caractère #, les paragraphes et les listes (simples ou numériques) sont donc les trois éléments à utiliser pour structurer un document avec Markdown.

Les titres

Pour être correctement structuré, un texte doit être divisé en parties, chapitres, sous-chapitres, etc. 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 (voir tableau ci-dessous).

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**.

En principe, 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. Il ne peut y avoir une branche contenant seulement une sous-branche.

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

classe	page titre	niveau 1	niveau 2
article	non (page 1)	Chapitre	Section
report	oui	Partie	Chapitre
book	oui	Partie ⁴⁶ (page paire)	Chapitre

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

Les titres peuvent être numérotés automatiquement via une instruction présente dans l'en-tête YAML (**numbersections: true**). Pour empêcher la numérotation d'un titre en particulier, il faut ajouter « `{-}` » à la suite du titre (sans espace).

Les paragraphes

Dans le texte, la séparation en paragraphes est réalisée par un double saut de ligne.

Il est cependant possible, en ajoutant « `\` » (ou, avec les éditeurs qui l'acceptent, trois espaces consécutifs) à la fin de la ligne, de provoquer un saut de ligne sans créer un nouveau paragraphe.

Les listes

Markdown permet de créer des listes simples et des listes numérotées.

46. Pour introduire le livre (introduction et/ou préface) ou pour ajouter des annexes ou une bibliographie, il faut ajouter les attributs `\frontmatter`, `\mainmatter` et `\backmatter` au début des parties. Pour la classe **book**, si on ne fait pas précéder l'introduction de `\frontmatter`, l'introduction sera numérotée comme une partie.

Pour les listes simples, il faut utiliser le tiret (ou le +) suivi d'une espace⁴⁷.

Ceci :

```
- item 1
- item 2
+ item 3
```

Donnera :

- item 1
- item 2
- item 2

Notez au passage⁴⁸ que les bullets sont ici remplacés par des tirets longs (cadratin)⁴⁹.

Pour les listes numérotées, il faut mettre un chiffre (peu importe le chiffre, mais le premier indique le début de la numérotation), suivi d'un point et d'une espace.

Ceci :

```
1. item 1
1. item 2
1. item 3
```

Donnera :

1. item 1
2. item 2

47. En typographie, le mot espace est au féminin.

48. Spécifique pour les documents déclarés en français dans l'en-tête YAML.

49. Le cadratin est une grandeur typographique qui équivaut à la largeur de la lettre la plus large, qui peut être différente d'une police à l'autre. C'est approximativement l'espace occupé par "00" (deux zéro).

3. item 3

Les listes doivent être précédées d'un saut de ligne. Il est possible de combiner les deux types de listes.

Peut-être moins utile, il est aussi possible de créer des cases à cocher.

Ceci :

- [x] item 1, coché
- [] item 2, non-coché
- [x] item 2.1, coché
- [] item 2.2, non-coché
- [] item 3, non-coché
- [X] item 4, coché

(attention aux espaces)

Donnera :

- item 1, coché
- item 2, non-coché
 - item 2.1, coché
 - item 2.2, non-coché
- item 3, non-coché
- item 4, coché

La mise en forme du texte

Les styles

Les fonctions de base sont l'*italique* et le **gras**. Il est aussi possible de barrer des caractères.

Ce passage :

```
*Ceci est un texte italique*, **Ceci est un texte gras**  
et ~~Ceci est un texte barré~~
```

Donnera :

Ceci est un texte italique, **Ceci est un texte gras** et ~~Ceci est un texte barré~~
(il est possible de les combiner)

Les indices et les exposants sont réalisés avec ~ et ^ donc :

```
H~2~O et X^2^
```

pour : H₂O et X²

Les tirets longs (cadratin) et demi-longs (demi-cadratin) sont créés avec trois et deux tirets (le signe moins du clavier) : — et –

Les compléments du texte

Appelés également « appareil critique », ces compléments sont la base des textes scientifiques.

Les notes de bas de page

Les notes sont créées avec l'ajout d'un « ^ » et le texte de la note est encadré de « [» et «] » :

```
ceci est le texte^[Et ceci est la note].
```

Les notes de bas de page sont numérotées automatiquement. Cette méthode est spécifique à la création de documents pdf avec Pandoc-Markdown. On peut aussi utiliser cette méthode pour ajouter une note dans les éléments de l'en-tête (pour le titre, le résumé ou un auteur).

Il y a une méthode un peu plus complexe (et compatible pour les exportations ePub, HTML, odt et docx) qui se fait en deux temps, avec l'appel de note :

```
ceci est le texte[^numéro de la note].
```

et avec toutes les notes regroupées à la fin du document .md :

```
[^numéro de la note]: Et ceci est la note.
```

Les liens

Pour créer un lien vers une page Web, il faut utiliser la séquence [](). Entre les crochets, on va placer le texte à afficher et, entre les parenthèses, l'URL cible :

```
[texte à afficher](https://url.cible.org)
```

On peut aussi, pour une écriture plus rapide, placer l'URL entre « < » et « > ».

```
<https://URL.be>
```

mais alors, le texte du lien sera l'URL elle-même.

Si une même URL doit être utilisée plusieurs fois dans le même document, on peut regrouper les liens sur une même adresse. À la fin du document, on ajoute :

```
[1]: <https://opensolution.be> "site de référence"
```

et dans le texte on fait le lien à cette URL en l'appelant :

```
texte texte vers ce [site][1]
```

(Attention, le lien est entre crochets et non entre parenthèses.)

Il est aussi possible de créer des liens à l'intérieur d'un document⁵⁰. Pour cela :

1. il faut ajouter une ancre (ici, après un titre) :

```
# chapitre 10 {#chap10}
```

50. Ne fonctionne pas si les liens sont transformés en notes en bas de page avec l'instruction **links-as-notes: true** dans l'en-tête YAML.

2. créer le lien avec la séquence :

```
texte texte texte, voir [chapitre 10](#chap10)
```

Les figures

La syntaxe pour insérer une figure est comparable à celle de la création d'un lien. Il faut seulement faire précéder l'instruction d'un « ! ». L'image qui sera insérée peut être un fichier présent sur l'ordinateur (attention à appeler le fichier au bon endroit sur le disque dur) ou une image quelque part sur le web (il faut alors entrer une URL correcte).

S'il y a une légende (texte entre le [et]), les images sont centrées et automatiquement numérotées avec « Figure x : » + la légende. *Pandoc* va éventuellement déplacer le texte avant et après les figures pour optimiser le remplissage des pages.

Ceci :

```
![Le logo de *Markdown*](markdown-large.png)
```

Donnera :



FIGURE 3 – Le logo de *Markdown*

S'il n'y a pas de légende (texte entre le [et]), la figure sera alignée à gauche. La mise en forme sera peut-être moins optimale.

Il est possible de modifier la taille de l'image en ajoutant { **width=50%** } derrière l'appel d'image (ici, réduction de la taille de 50%).

Les tableaux

Markdown gère plutôt bien les tableaux. Il faut utiliser le « | » pour séparer les colonnes et le saut de ligne pour séparer les lignes.

Pour aligner à gauche (les colonnes sont alignées à gauche par défaut), centrer ou aligner à droite le contenu d'une colonne, il faut le déclarer dans la deuxième ligne en ajoutant « : » à gauche, « : » des deux côtés ou « : » à droite des trois « - ». Le nombre de tirets (« - ») va influencer la largeur de la colonne.

La séquence suivante :

```
| gauche | centre | droite |
|:---|:---:|---:|
| cellule 1.1 | cellule 2.1 | cellule 3.1 |
| cellule 1.2 | cellule 2.2 | cellule 3.2 |
| cellule 1.3 | cellule 2.3 | cellule 3.3 |
: Titre du tableau
```

Donnera :

TABLE 1: Titre du tableau

gauche	centre	droite
cellule 1.1	cellule 2.1	cellule 3.1
cellule 1.2	cellule 2.2	cellule 3.2
cellule 1.3	cellule 2.3	cellule 3.3

Les enrichissements (gras, italique...) sont autorisés pour toutes les cellules du tableau.

Le titre du tableau apparaît au-dessus, numéroté, comme pour les figures (où il apparaît en dessous).

Avec cette syntaxe, il n'est pas possible, avec Pandoc-Markdown, de fusionner des cellules. Il faut alors utiliser une autre syntaxe un peu plus complexe (ici, pour fusionner les cellules A2 et A3) :

```

+-----+:-----:-----:--+
| gauche|centre| droite|
+-----+-----+-----+
| cellules fus. |cellule 2.1|cellule 3.1 |
|           |cellule 2.2|cellule 3.2|
+-----+-----+-----+
| cellule 1.3  |cellule 2.3|cellule 3.3|
+-----+-----+-----+
: Titre du tableau

```

pour avoir :

TABLE 2.3: Titre du tableau

gauche	centre	droite
cellules fus.	cellule 2.1	cellule 3.1
	cellule 2.2	cellule 3.2
cellule 1.3	cellule 2.3	cellule 3.3

L'utilisation de la syntaxe html est aussi une solution.

La création de tableaux avec Markdown peut paraître assez complexe. Plusieurs outils d'édition prévoient des aides. Il existe aussi sur le Web un générateur de tables⁵¹ Markdown qui facilite leur création (y compris la transformation de fichiers .csv provenant d'un tableur).

Les équations mathématiques

Il est aussi possible d'intégrer des équations mathématiques avec la syntaxe *TeX Math* de *LaTeX*. Dans un fichier Markdown, ces équations sont encadrées par « `$$` ».

La séquence suivante :

```
$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$
```

donnera :

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Encore une fois, l'objectif reste juste de montrer ce qui est possible et non de détailler cette syntaxe. Il y a un tutoriel très complet sur *WikiBooks*⁵². On peut aussi utiliser un éditeur en ligne⁵³ pour écrire les équations.

51. https://www.tablesgenerator.com/markdown*tables

52. <https://en.wikibooks.org/wiki/LaTeX/Mathematics>

53. <http://www.hostmath.com/>

Les citations

Pour citer et mettre en évidence un passage de texte, il faut utiliser le caractère « > » en début de paragraphe.

Ce passage :

> "But here's the most important bit, saved for last: none of this matters if you want to write a book. Quite a few people have told me that they want to write a book, but they're not sure about which tools to use. My advice: all you need to write a book is a program that allows you to write text into a file" [Ball2018].

Donnera :

« But here's the most important bit, saved for last: none of this matters if you want to write a book. Quite a few people have told me that they want to write a book, but they're not sure about which tools to use. My advice: all you need to write a book is a program that allows you to write text into a file » (Ball, 2018).

Le paragraphe est décalé à droite avec une justification gauche et droite (comme le reste du texte). Avec deux « > », le retrait sera plus important.

Notez au passage l'ajout de la citation à l'auteur du texte qui va reproduire le renvoi à la bibliographie suivant la norme sélectionnée dans l'en-tête YAML et ajouter la référence dans la bibliographie en fin de document (voir ci-dessous pour plus d'explications à propos des possibilités en matière de bibliographie).

Les appels bibliographiques et la bibliographie

Un élément puissant de l'utilisation de Markdown avec *Pandoc* est la facilité d'insertion de citations bibliographiques dans le texte et de génération d'une bibliographie à la fin du document (pdf mais aussi HTML, ePub, odt, docx...).

Pour que cela fonctionne :

1. il faut que deux fichiers soient présents dans le même répertoire que le document .md :
 - un fichier BibTeX créé avec un gestionnaire bibliographique. Avec Zotero⁵⁴, il faut utiliser la fonction « exporter les documents » et choisir le format BibTeX pour créer un fichier .bib ;
 - le fichier .csl (*Citation Style Language*) correspondant au style bibliographique que vous souhaitez utiliser ;
2. que ces deux fichiers soient renseignés dans l'en-tête YAML (voir plus loin) ;
3. insérer les citations :
 - soit avec l'appel [*@auteur0000*] pour une citation avec des parenthèses : « texte texte texte (*Auteur, 0000*) » ;⁵⁵
 - soit avec l'appel *@auteur0000* pour une citation directe : « texte texte texte *Auteur (0000)* » ;
4. ajouter le titre « Bibliographie » en fin de document ;
5. ajouter « *-citeproc* » dans la commande *pandoc* :

```
pandoc --citeproc source.md -o destination.pdf
```

Il existe des explications plus complètes sur le blog « Zotero francophone »⁵⁶.

54. <https://www.zotero.org/>

55. *@auteur0000* est la clé de citation enregistrée dans le gestionnaire bibliographique.

56. <https://zotero.hypotheses.org/2258>

Les codes et textes non formatés

Pour afficher du code sans formatage (comme fait à plusieurs reprises plus haut), il faut :

- soit encadrer le texte (ligne précédente et ligne suivante) avec trois accents graves consécutifs ;
- soit, pour une seule ligne par exemple, faire précéder le texte par une tabulation ou au moins trois espaces.

Autres codes utiles pour la mise en forme

Quelques petites commandes peuvent encore se révéler utiles :

- utilisation de commandes *LaTeX* pour :
 - insérer un saut de page : `\newpage`,
 - encadrer du texte : `\fbox{texte texte texte}`⁵⁷,
 - mettre du texte en couleur : `\textcolor{red}{texte en rouge}`,
 - réduire la taille des caractères : `\small` (`\normalise` pour revenir à la taille normale),
 - augmenter la taille des caractères : `\large` (`\normalise` pour revenir à la taille normale)⁵⁸,
 - mettre du texte en petites capitales : `[Texte texte texte]{.small-caps}`,
- insérer une ligne horizontale : `***` ou `---` ;
- insérer la date du jour (dans un titre ou dans le texte) avec la variable `\today` ;
- insérer des espaces insécables (pour insérer une ligne vide ou empêcher le positionnement en début de ligne de « ? », « : » ou « ; ») avec ` ` ;
- comme Markdown utilise des caractères pour le formatage, si on souhaite les utiliser comme tels, il faut empêcher Pandoc de les interpréter comme des instructions et faire précéder ce caractère par le caractère d'échappement « \ ».

Voici les caractères concernés :

- barre oblique inversée : `\`
- astérisque : `*`

57. Limité à une ligne de texte.

58. Voir la troisième partie pour voir toutes les possibilités.

- trait d’union : –
- souligné : `_`⁵⁹
- crochets : `[]`⁶⁰
- accolades : `{ }`

et en début de ligne :

- point d’exclamation (en début de ligne) : `!`
- croisillon (en début de ligne) : `#`
- triple accent grave : `````

Une série de commandes *LaTeX* supplémentaires sont proposées dans la troisième partie du manuel.

59. Il est équivalent à `*` pour mettre en italique.

60. L’échappement doit impérativement précéder chacun de ces caractères. Cela concerne donc autant les crochets ouvrants que les crochets fermants.

L'en-tête YAML

Pandoc prend en charge les en-têtes YAML⁶¹ pour les fichiers Markdown.

L'en-tête YAML n'est pas indispensable pour produire un document, mais il permet de modifier les valeurs par défaut et d'améliorer la qualité du document (mise en page et métadonnées du document).

Il n'est pas indispensable de retenir tout ce que contient l'en-tête YAML. Le plus simple est de copier un modèle dans un fichier et de le réutiliser pour chaque projet.

L'en-tête YAML va se placer au début du document Markdown (il peut également se trouver dans un fichier à part) et est encadré de trois tirets au début et à la fin.

Ci-dessous un exemple d'un en-tête YAML qui contient quelques instructions de **mise en forme** et les **métadonnées** du document produit (ici un pdf).

```
---  
# format du document  
  
documentclass: article  
lang: fr  
geometry:  
- left=2.5cm  
- right=2.5cm  
- top=2.5cm  
- bottom=2.5cm  
- a4paper  
linestretch: 1.5  
fontsize: 11pt  
toc: false  
toc depth: 2
```

61. <https://geekflare.com/fr/what-is-yaml/>

```

numbersections: false

# métadonnées

title: Titre du document
subtitle: Sous-titre du document
author: Nom, Prénom
affiliation: ULiège
date: 2021 (cc-by)
abstract: Ce court document est destiné à démontrer ...
keywords: Markdown, rédiger
subject: écriture numérique ouverte
Modified: 01/03/25 09:12:20
right: CC-BY 0.4

# liens

links-as-notes: true
linkcolor: green
urlcolor: blue

# bibliographie

bibliography: library.bib
csl: apa.csl
---
```

Pour le rendre plus lisible, il est possible d'insérer (comme ci-dessus) des commentaires.

Toutes les lignes de l'en-tête font directement référence à des variables que l'on modifie directement (**variable: valeur**).

Il est possible d'ajouter de nombreuses options avec *header-includes* (contenu dans le préambule⁶² *LaTeX*) via des packages (voir « [Troisième partie](#) »).

62. Le préambule est la première partie du fichier source *LaTeX*. Il contient les instructions et les variables qui vont affecter le document.

Format du document

documentclass:

avec **book** ou **report** ou **article**. La mise en forme finale va varier en fonction de la *classe*.

lang:

Pour la langue (qui par défaut est l'anglais), il faut utiliser **fr** pour franciser la mise en forme (par exemple : « Table des matières » au lieu de « Contents » ou des tirets longs au lieu de puces dans les listes).

geometry:

va formater la page. On peut ajouter **landscape** à la suite des quatre marges pour produire un document au format « paysage ».

On peut choisir de n'indiquer qu'une valeur pour les quatre marges avec **margin=xcm**.

linestretch:

permet, si nécessaire, d'augmenter ou de diminuer l'interligne (**1** est la valeur par défaut).

fontsize:

la taille des caractères peut être **10**, **11** ou **12** pt. Il est possible d'avoir d'autres tailles (8, 9, 14, 17 ou 20 pt), après avoir remplacé **article** par **extarticle** dans « *documentclass* ».

toc:

true/false (créer une table des matières en début du document). On peut aussi utiliser **lof:** et **lot:** pour les tables de figures et de tableaux.

toc depth:

de **1** à **x** (précise la profondeur de cette table des matières).

numbersections:

true/false (pour numéroter automatiquement les titres). Pour empêcher la numérotation d'un titre en particulier, il faut ajouter « {-} » à la suite du titre, sans espace.

Liens

linkcolor:

par exemple **green** (pour définir la couleur des lien interne en vert).

urlcolor:

par exemple **blue** (pour définir la couleur des liens externes en bleu).

links-as-notes:

true/false (pour transformer tous les liens en notes de bas de page).

Bibliographie

Deux variables sont indispensables pour insérer automatiquement des citations et une liste bibliographique :

bibliography:

library.bib (le nom du fichier *bibtex* créé, par exemple, avec *Zotero*).

csf:

apa.csl (le style bibliographique).

Métadonnées

Pour faciliter le partage et la diffusion d'un document numérique, celui-ci doit contenir sa propre description. On parle de métadonnées⁶³. La structure de ces métadonnées fait l'objet de normes.

63. <https://doranum.fr/metadonnees-standards-formats/cours-introductif-sur-les-metadonnees/>

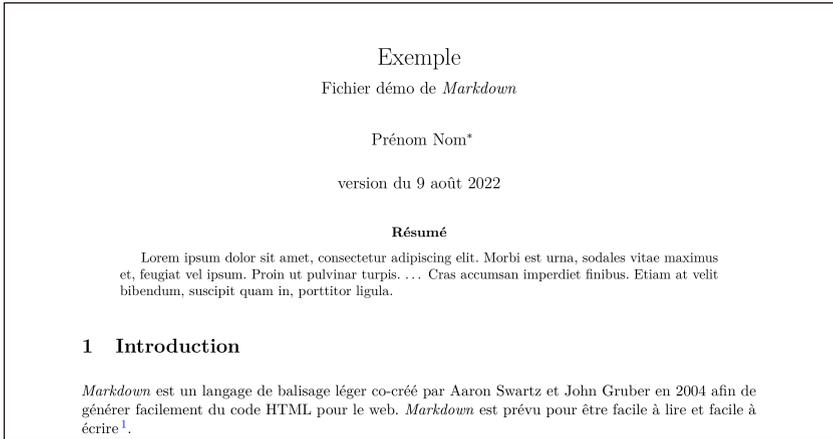
À l'heure actuelle, il n'est pas encore possible d'intégrer les 15 champs de description *Dublin Core*⁶⁴ dans un document Markdown, mais l'utilisation de l'en-tête YAML permet néanmoins le partage des principales informations entre outils ainsi que l'alimentation automatique des bases de données.

Les éléments utilisés ici sont :

- title
- subtitle
- author
- date
- abstract
- affiliation
- keywords
- subject
- project
- modified
- right

Les cinq premiers items servent aussi pour la mise en forme. Par exemple, avec le type *article*, ces informations apparaissent au début du document.

64. <https://www.bnf.fr/fr/dublin-core>



Exemple de début d'un document de type « article » avec affichage des cinq premières métadonnées

Pour le champs « date: », on peut indiquer « `\today` » pour avoir la date du jour et « `\hhmm` » pour l'heure.

En utilisant *LUA*⁶⁵, il est possible d'identifier et de provoquer l'affichage d'une affiliation différente pour chaque auteur, mais il est déjà possible de renseigner plusieurs auteurs avec des affiliations différentes (qui apparaîtront en notes en bas de page) en utilisant les notes :



Pour aller plus loin, on peut consulter la liste complète⁶⁶ des variables possibles à insérer dans l'en-tête YAML pour produire des pdf.

65. <https://github.com/mpeterv/markdown>

66. https://yamlthis.r-lib.org/reference/yml_LaTeX_opts.html

Les outils

Les outils en ligne

On peut trouver nombre d'outils en ligne pour éditer au format Markdown. Les transformations sont variables d'un site à l'autre, du *CommonMark*⁶⁷ à une gamme très complète de commandes interprétées et de formats produits.

Ces sites Web permettent non seulement de se familiariser avec Markdown mais sont aussi des outils d'écriture collaborative en substitution d'outils (GAFAM) comme *Google Docs*.

Les outils les plus courants sont :

- *HedgeDoc*⁶⁸
- *markup.rocks*⁶⁹
- *Dillinger*⁷⁰
- *StackEdit*⁷¹
- *Preview Markdown*⁷²
- *Markdit*⁷³
- *HackMD*⁷⁴
- *Stylo*⁷⁵

67. Donc en général sans possibilité d'introduire des équations mathématiques, des notes en base de page ou des citations bibliographiques.

68. <https://demo.hedgedoc.org/>

69. <https://markup.rocks/>

70. <https://dillinger.io/>

71. https://stackedit.io

72. <https://harshkapadia2.github.io/preview-markdown/>

73. <https://markdit.com/>

74. <https://hackmd.io/>

75. <https://stylo.huma-num.fr/>

On les utilise via le navigateur, sans installation. Les deux derniers nécessitent une inscription, gratuite. Avec *HackMD*, on peut produire une page web qui peut être partagée. *Stylo*⁷⁶ pousse très loin tous les aspects de l'écriture académique (voir l'aide⁷⁷, très complète).

Du côté des outils en ligne, il y a aussi les outils spécifiquement développés pour la prise de note collaborative comme *Framapad*⁷⁸, *Etherpad*⁷⁹ ou le Pad « domaine public »⁸⁰, basé sur le même logiciel.

Pour travailler en mode local avec toutes les fonctionnalités liées au Pandoc-Markdown, il faut nécessairement installer *Pandoc* et *LaTeX*.

Les outils pour éditer

Un simple éditeur de texte (comme *Notepad* pour *Windows*, *Simplenote* pour *macOS* ou *Xed* pour *Linux*) peut être utilisé pour créer et gérer les fichiers Markdown. Des éditeurs comme *Atom*⁸¹ ou *Komodo Edit*⁸², particulièrement appréciés par les programmeurs, intègrent le format Markdown⁸³.

Pour débiter, il y a également *Typora*⁸⁴ qui permet de se familiariser avec *Markdown*. Il guide l'utilisateur dans l'installation de *Pandoc* et de *LaTeX*. *MarkText*⁸⁵ est une alternative libre à *Typora* mais il propose moins de fonctions (en exportation par exemple).

*Zettlr*⁸⁶ est un autre logiciel libre qui commence à avoir beaucoup de succès. Il propose une aide très complète. Il est néanmoins un peu moins aisé à prendre en main.

76. Créé par la Chaire de recherche du Canada sur les écritures numériques et est hébergé sur le site français huma-num.fr.

77. <https://stylo-doc.ecrituresnumeriques.ca/fr/about/>

78. <https://framapad.org/abc/fr/>

79. <https://etherpad.org/>

80. <https://pads.domainepublic.net/>

81. <https://atom.io>

82. <https://docs.activestate.com/komodo/12/>

83. Ils proposent une coloration syntaxique qui permet d'identifier rapidement les balises utilisées.

84. <https://typora.io>. Non libre et payant (14,99 \$) depuis la version 1.0 – 23/11/2021

85. <https://github.com/marktext/marktext/releases>

86. <https://www.zettlr.com/>

Il y a également *Ghostwriter*⁸⁷ qui intègre plusieurs modes d'édition, de visualisation et d'exportation.

Cette courte liste n'est pas exhaustive, il existe bien d'autres logiciels qui permettent d'éditer des textes Markdown. Ces logiciels sont soit spécifiques à un système d'exploitation (*Windows*, *Mac* ou *Linux*) soit utilisables sur les trois. Une recherche rapide avec un moteur de recherche va permettre de découvrir des listes des logiciels les plus récents et/ou les plus appréciés.

Ces logiciels poussent parfois très loin le principe de l'écriture sans distraction avec des écrans sans menus, uniquement le texte en cours de rédaction.

Typora et *MarkText* vont automatiquement intégrer les balises Markdown dans l'affichage. Il est toutefois possible, par une simple combinaison de touches de les faire apparaître. *Ghostwriter* va par contre permettre la prévisualisation avec à gauche le texte brut, balisé, et à droite le rendu après interprétation. La majorité des outils qui proposent la prévisualisation (soit via un éditeur soit en ligne), vont proposer une transformation HTML avec la syntaxe *CommonMark*. Avec *Ghostwriter*, il est possible de choisir de manière plus fine le mode de prévisualisation (*Pandoc-Markdown* par exemple).

Malgré la « simplicité » de Markdown, il faut consacrer un certain temps à le maîtriser surtout si on est habitué aux traitements de texte tels que *Word* voire *LibreOffice*. Les éditeurs Markdown permettent de reproduire en partie ce que les traitements de texte proposent au moyen d'extensions (vérifier la grammaire, l'orthographe, l'affichage de l'historique des modifications ou le code).

Une tâche apparemment simple comme l'insertion d'une image ou d'un tableau peut être parfois difficile avec Markdown. Tous ces logiciels sont différents. Il ne faut donc pas hésiter à passer du temps à les essayer avant d'en adopter un (ou plusieurs en fonction de leurs spécificités).

87. <https://ghostwriter.kde.org/fr/>



Ghostwriter en mode prévisualisation



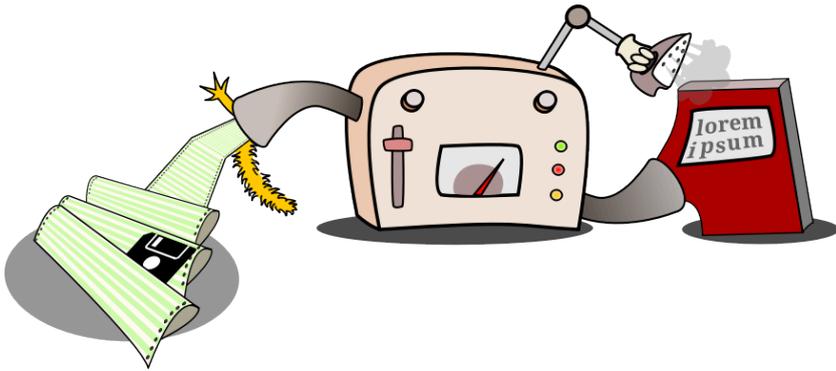
Ghostwriter en mode sans distraction

Les outils de transformation

Pandoc

Pandoc a été créé en 2006, et est toujours maintenu, par John MacFarlane⁸⁸. C'est un logiciel libre (publié sous licence GPL) considéré comme le couteau suisse de l'édition. Il permet de passer facilement d'un format de fichier à un autre.

Pandoc gère plus de 50 formats différents⁸⁹ !



L'illustration proposée en page d'accueil pandoc.org

Pour installer *Pandoc* il faut se rendre sur la page <https://pandoc.org/installing.html> et télécharger le fichier qui correspond au système d'exploitation que l'on utilise.

LaTeX

LaTeX est un système de composition de documents, idéal pour les textes scientifiques et techniques. Il permet une séparation claire entre le

88. <https://johnmacfarlane.net/>

89. <https://pandoc.org>

contenu et la mise en forme, offrant un contrôle précis sur la typographie et les équations mathématiques.

LaTeX est une extension de *TeX*, un langage de programmation de mise en page développé en 1977. *LaTeX*, existe depuis 1982 et n'a pas cessé d'être amélioré. Il est actuellement maintenu par une équipe de bénévoles au sein du projet *LaTeX3*, assurant sa pérennité et son évolution.

LaTeX produit des documents au rendu professionnel, avec une mise en page soignée. Il excelle dans la composition de formules mathématiques complexes, la création de bibliographies, tables des matières et de renvois. Avec *LaTeX*, Les utilisateurs se concentrent sur le contenu, tandis que *LaTeX* gère la mise en forme automatiquement. *LaTeX* est extensible avec de nombreux packages (voir dans la troisième partie) pour répondre à des besoins spécifiques.

Pour installer *LaTeX*, le logiciel à installer dépend du système d'exploitation :

- pour *Linux*, on va préférer *Texlive*. Pour l'installer, dans un terminal, on va entrer la commande :

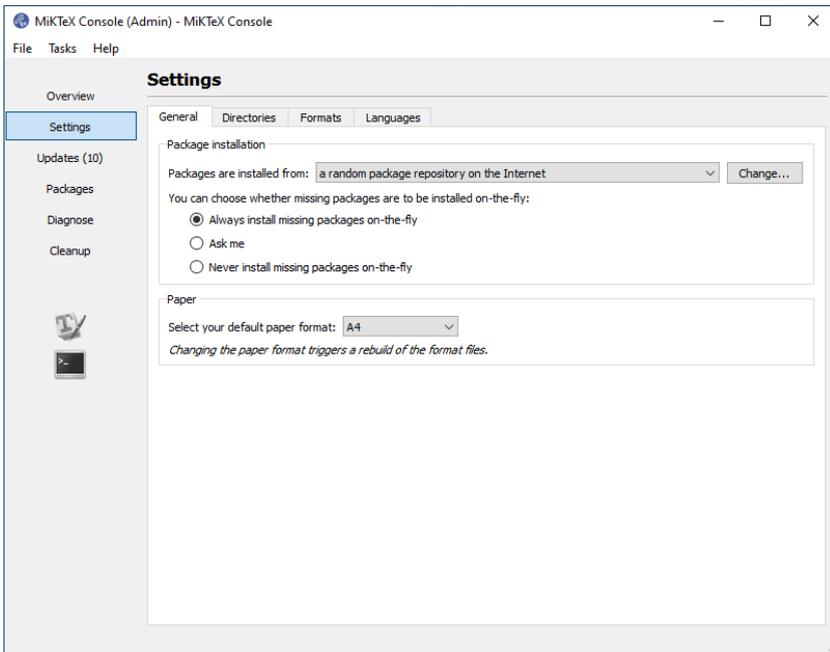
```
sudo apt-get install texlive texlive-xetex
texlive-latex-extra texlive-lang-french
```

- pour *macOS*, on va utiliser *MacTeX* qui est téléchargeable à l'adresse : <https://tug.org/mactex/mactex-download.html>. Ce programme s'installe comme tous les autres programmes sur un Mac (double clic sur un fichier avec extension « .dmg ») ;
- pour *Windows*, on va préférer *MiKTeX*. Le lien de téléchargement est : <https://miktex.org/download>.

Avec *MiKTeX* (qui peut aussi être installé sur *Linux* ou *macOS*), il faut bien lire les instructions sur la page de téléchargement. L'installation doit être complétée via « MiKTeX Console » (que l'on va appeler via le menu « Démarrer » de *Windows*, à gauche).

Il faut passer en mode administrateur *MiKTeX*. À partir de là accéder à la page des mises à jour.

Il faut alors demander à *MiKTeX* d'installer automatiquement les paquets manquants et via la page « Settings » (paramètres), cliquer sur le bouton radio : *Always install missing packages on-the-fly* (pour éviter qu'il demande de confirmer l'installation des paquets manquants, un à la fois).



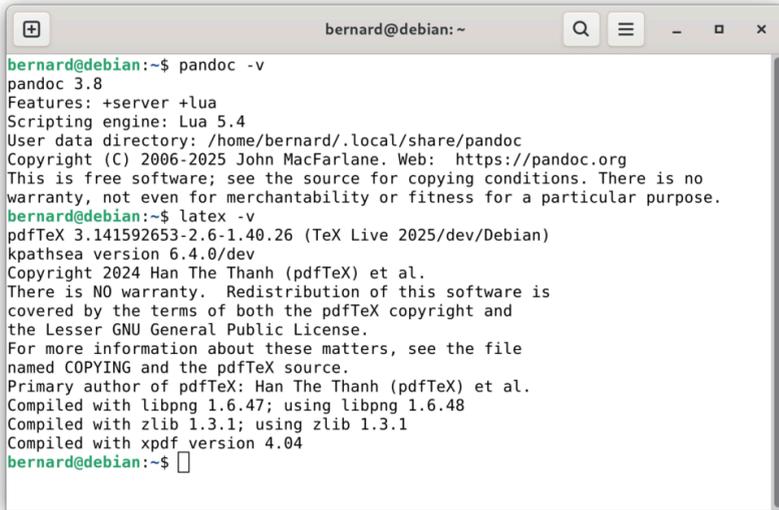
Mode administrateur *MiKTeX*.

Pour vérifier l'installation de *Pandoc* et de *LaTeX*, il faut exécuter, dans le terminal :

```
pandoc -v
```

et

```
latex -v
```



```

bernard@debian:~$ pandoc -v
pandoc 3.8
Features: +server +lua
Scripting engine: Lua 5.4
User data directory: /home/bernard/.local/share/pandoc
Copyright (C) 2006-2025 John MacFarlane. Web: https://pandoc.org
This is free software; see the source for copying conditions. There is no
warranty, not even for merchantability or fitness for a particular purpose.
bernard@debian:~$ latex -v
pdfTeX 3.141592653-2.6-1.40.26 (TeX Live 2025/dev/Debian)
kpathsea version 6.4.0/dev
Copyright 2024 Han The Thanh (pdfTeX) et al.
There is NO warranty. Redistribution of this software is
covered by the terms of both the pdfTeX copyright and
the Lesser GNU General Public License.
For more information about these matters, see the file
named COPYING and the pdfTeX source.
Primary author of pdfTeX: Han The Thanh (pdfTeX) et al.
Compiled with libpng 1.6.47; using libpng 1.6.48
Compiled with zlib 1.3.1; using zlib 1.3.1
Compiled with xpdf version 4.04
bernard@debian:~$ █

```

Vérification de l'installation dans un terminal.

Si l'installation n'est pas conforme, la commande renverra une erreur.

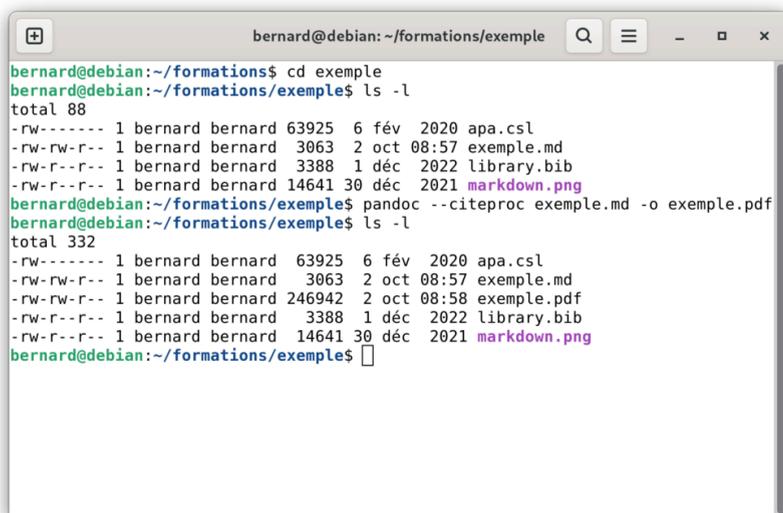
La transformation

Par défaut, *Pandoc* s'utilise en mode terminal (ou ligne de commande).

Pour accéder au mode terminal, sur une machine *Linux* ou *Mac*, il faut chercher « terminal ». Pour *Windows* il faut entrer « cmd » dans la zone de recherche en bas, à gauche.

Le terminal apparaît en général (à moins d'en avoir modifié les paramètres) sous la forme d'un petit rectangle noir avec le texte en blanc.

Le mode terminal permet d'entrer directement des commandes au système sans passer par une interface graphique. Pour l'utilisation de *Pandoc*, le nombre de commandes à retenir est réduit.



```
bernard@debian: ~/formations/exemple
bernard@debian:~/formations$ cd exemple
bernard@debian:~/formations/exemple$ ls -l
total 88
-rw----- 1 bernard bernard 63925 6 fév 2020 apa.csl
-rw-rw-r-- 1 bernard bernard 3063 2 oct 08:57 exemple.md
-rw-r--r-- 1 bernard bernard 3388 1 déc 2022 library.bib
-rw-r--r-- 1 bernard bernard 14641 30 déc 2021 markdown.png
bernard@debian:~/formations/exemple$ pandoc --citeproc exemple.md -o exemple.pdf
bernard@debian:~/formations/exemple$ ls -l
total 332
-rw----- 1 bernard bernard 63925 6 fév 2020 apa.csl
-rw-rw-r-- 1 bernard bernard 3063 2 oct 08:57 exemple.md
-rw-rw-r-- 1 bernard bernard 246942 2 oct 08:58 exemple.pdf
-rw-r--r-- 1 bernard bernard 3388 1 déc 2022 library.bib
-rw-r--r-- 1 bernard bernard 14641 30 déc 2021 markdown.png
bernard@debian:~/formations/exemple$
```

Quatre commandes dans un terminal

Il n'est pas nécessaire de maîtriser le mode terminal pour effectuer cette manipulation et les commandes listées ci-dessous sont sans danger pour l'ordinateur.

Pour que la commande fonctionne, il faut néanmoins préciser l'endroit où se trouve(nt) le(s) fichier(s) .md dans l'arborescence (« dir » sous *Windows* et « ls -l » sous *Linux* ou *macOS*) ou se placer dans le bon répertoire (avec la commande « cd » pour *change directory*).

La commande *Pandoc* de base est la suivante :

```
pandoc monfichier.md -o monfichier.pdf
```

Le « -o »⁹⁰ (pour *output*) doit précéder le nom du fichier destination. La destination peut aussi être un fichier .html, un fichier .epub voire .docx ou .odt (pour rappel *Pandoc* gère plus de 50 formats).

Il est possible d'utiliser plusieurs fichiers « sources » (parties, chapitres...) simultanément dans la commande. Ici, l'en-tête YAML se trouve dans le premier fichier, il ne doit pas être répété.

```
pandoc yaml.md intro.md chap1.md chap2.md
chap3.md chap4.md -o monlivre.pdf
```

Si les fichiers sont numérotés (pour qu'ils soient traités dans le bon ordre, 01yaml.md 02intro.md, 03chap1.md, 04chap2.md...), on peut simplifier la commande :

```
pandoc *.md -o monlivre.pdf
```

Par défaut, *Pandoc* utilise *pdflatex* pour la transformation en pdf. Si le document contient des caractères spéciaux ou des polices spécifiques (message d'erreur lors de la transformation : *Error producing PDF*), on peut utiliser *xelatex* ou *lualatex* via l'option **--pdf-engine**. Dans la commande *Pandoc* :

```
pandoc --pdf-engine=xelatex monfichier.md -o monfichier.pdf
```

Pour rappel, pour insérer une bibliographie, il faut ajouter **--citeproc** dans la commande de transformation :

90. Pour les options longues (--pdf-engine=xelatex ou --citeproc) on utilise deux tirets, pour les courtes (-s qui équivaut à --standalone ou -o pour --output) ce sera un seul. --citeproc n'a pas de version courte.

```
pandoc --citeproc monfichier.md -o monfichier.pdf
```

Avec *Typora*, *MarkText* et plusieurs autres outils, il est possible d'exporter directement un fichier Markdown en fichier pdf sans quitter l'interface graphique.

Avec *Typora* le fichier créé n'aura cependant pas une forme comparable à un fichier pdf produit via *LaTeX* (mise en page stricte et professionnelle avec utilisation des données YAML, numérotation des pages, notes de bas de page, équations mathématiques, citations et bibliographie...). *Typora* n'utilise que *TeX Math* de *LaTeX*.

Zettlr ou *Ghostwriter* vont eux utiliser directement *Pandoc* et *LaTeX* pour produire les documents pdf. *Zettlr* propose d'ailleurs des réglages très fins via la « gestion des assets » (commande Alt+Ctrl+, sous *Linux*). Avec *Ghostwriter* les principaux réglages sont réalisés via l'en-tête YAML.

Les modèles

Il est possible d'utiliser des modèles (des *templates*)⁹¹ pour l'exportation.

Dans la dernière partie du manuel se trouve une liste de liens vers des *templates LaTeX* qui remplacent le *template* par défaut. Pour appeler un modèle, il faut ajouter son nom dans la ligne de commande.

```
pandoc --template=montemplate.tex monfichier.md -o monfichier.pdf
```

Cette possibilité existe aussi pour les exportations .docx ou .odt. Après avoir créé un modèle (mise en page, styles, en-tête et pieds de page...) avec le traitement de texte, on fait appel à ce modèle dans la commande Pandoc.

91. Voir : <https://github.com/jgm/pandoc-templates>.

```
pandoc --reference-doc=monmodele.odt monfichier.md -o monfichier.odt
```

À propos de la maîtrise de l'ordinateur

Le mode terminal fait généralement peur. Les personnes qui ont commencé à utiliser un ordinateur avec le DOS de *Microsoft*, de la fin des années 1980 et jusqu'en 1995⁹², ou utilisatrices de *Linux* sont un peu familiarisées avec la ligne de commande.

Avec le phénomène de l' « impératif fonctionnel » ([Vitali-Rosati, 2024a](#)) et avec la généralisation de l'utilisation des tablettes et des smartphones, la maîtrise de l'ordinateur est de moins en moins évidente⁹³.

La connaissance des notions de dossiers, fichiers, programmes n'est plus nécessaire. Pourtant, pour reprendre le contrôle de son ordinateur, il est indispensable de :

- comprendre l'organisation de son disque dur (programmes, dossiers et fichiers) ;
- savoir où se trouvent les documents (fichiers) que l'on utilise ;
- savoir renommer et déplacer des fichiers et des dossiers ;
- savoir installer un programme et le lancer ;
- savoir comment télécharger des fichiers, les compresser et les décompresser.

92. Apparition de Windows 95.

93. Voir <https://www.mac4ever.com/societe/187334-pourquoi-les-jeunes-sont-devenus-si-nuls-en-informatique>

3. *LaTeX*

L*aTeX* est donc un langage sémantique et un procédé de génération de pdf, principalement développé pour l'impression de documents scientifiques comme des articles et des livres. Sa syntaxe est riche et standardisée, mais compliquée à lire et à utiliser.

Cette partie 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*. La source principale de cette partie vient de huit années de pratique de cette technologie mais également d'une consultation approfondie de sites web traitant de *LaTeX* et d'ouvrages spécialisés (par exemple ([Desgraupes, 2019](#)) ou le manuel en ligne d'Éric Guichard : *LaTeX pour littéraires, éditeurs et designers*⁹⁴).

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.

Les promesses de Markdown, format texte, pour rédiger et produire des documents avec *Pandoc* et *LaTeX* sont nombreuses. La durabilité, la simplicité et la légèreté mais surtout la possibilité de reprendre possession de notre utilisation du numérique et la maîtrise d'outils non propriétaires 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.

94. <https://www.ericguichard.fr/LaTeX-pour-litteraires-sans-hir.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 dans les pages qui suivent⁹⁵.

95. Toutes les commandes ont été testées avec Pandoc 3.1.3 et TeX Live 2023/Debian.

Principes de base

Les 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 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** (pour créer des diaporamas, voir plus loin).

Les paquets

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

Avec Markdown, 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 instructions qui seront utilisables par la suite dans la rédaction.

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

Les formats

Dans le préambule 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:*).

Les 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. Elles sont également utilisées pour les autres types de documents générés avec *Pandoc* (ePub, HTML...).

Les templates

Il est aussi possible de se servir d'un template⁹⁶ (modèle) qui sera appelé lors de la transformation avec *Pandoc* (`--template=montemplate.tex`).

96. <https://www.overleaf.com/latex/templates>

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 » ([Taffin & Melot, 2023](#)).

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 empattement (*serif*) ou sans empattement (*sans serif*) ;
- le style (gras, italique...);
- la police.

Le changement de police se fait via une instruction dans l’en-tête YAML. On appelle les polices en ajoutant `\usepackage{nom_de_la_police}` dans la section *header-includes*: de l’en-tête YAML. Voici quelques polices qui fonctionnent bien :

- **times** ;
- **palatino** ;
- **bookman** ;
- **newcent** ;
- **helvet** ;
- **avant**.

Pour les polices **helvet** et **avant**, qui sont des polices sans empattement (*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 familles, il faut consulter le catalogue *LaTeX*⁹⁷. Pour chacune des polices, la commande à ajouter dans l'en-tête (YAML) est précisée.

Tailles des caractères

Dans un document, il est possible de déclarer 3 tailles différentes (10, 11 ou 12 pt) via l'en-tête YAML.

On peut aussi modifier cette taille dans le texte avec 10 possibilités qui modifient la taille de 5 à 25 pt. Par exemple, pour la plus grande taille, on va utiliser `\Huge` :

```
\Huge texte \normalsize
```

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

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

97. <https://tug.org/FontCatalogue/>

Taille des caractères

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

Pour réduire, pour tout le document, la taille des caractères dans toutes les notes on peut insérer la commande :

```
\renewcommand{\footnotesize}{\scriptsize}
```

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.

Couleurs

Par défaut, *LaTeX* connaît les 19 couleurs suivantes : black, blue, brown, cyan, darkgray, 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.

 Apricot	 Aquamarine	 Bittersweet	 Black
 Blue	 BlueGreen	 BlueViolet	 BrickRed
 Brown	 BurntOrange	 CadetBlue	 CarnationPink
 Cerulean	 CornflowerBlue	 Cyan	 Dandelion
 DarkOrchid	 Emerald	 ForestGreen	 Fuchsia
 Goldenrod	 Gray	 Green	 GreenYellow
 JungleGreen	 Lavender	 LimeGreen	 Magenta
 Mahogany	 Maroon	 Melon	 MidnightBlue
 Mulberry	 NavyBlue	 OliveGreen	 Orange
 OrangeRed	 Orchid	 Peach	 Periwinkle
 PineGreen	 Plum	 ProcessBlue	 Purple
 RawSienna	 Red	 RedOrange	 RedViolet
 Rhodamine	 RoyalBlue	 RoyalPurple	 RubineRed
 Salmon	 SeaGreen	 Sepia	 SkyBlue
 SpringGreen	 Tan	 TealBlue	 Thistle
 Turquoise	 Violet	 VioletRed	 White
 WildStrawberry	 Yellow	 YellowGreen	 YellowOrange

Les 64 couleurs *xcolor* possibles Source : Overleaf

Quelques exemples :

```
\color{cyan}écrit en cyan \color{black}
\color{Tuquoise}écrit en turquoise \color{black}
\colorbox{BurntOrange}{avec un fond orange dans un box de texte}
```

(attention à la majuscule)

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

```
linkcolor: YellowGreen urlcolor: PineGreen
```

Petites capitales et autres attributs

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

```
\textsc{Texte texte texte}
```

ou

```
[Texte texte texte]{.smallcaps}
```

En suivant ce même modèle (utilisation de l'attribut **smallcaps**), pour mettre du texte en surbrillance (ici en jaune), on va utiliser l'attribut **mark** :

```
[texte]{.mark}
```

Pour souligner du texte, on va utiliser l'attribut **underline** :

```
[texte]{.underline}
```

Boîtes de texte

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 **minipage**.

```
\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).

Bloc de texte

`\parbox` crée un bloc de texte dont la largeur est définie (en pouces ou en cm).

Le bloc de texte n'est pas entouré d'un filet (lignes) et peut contenir plus d'une ligne de texte.

```
\begin{centre}\parbox{10cm}{\scriptsize Bloc
de texte.}\normalsize\end{centre}
```

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

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 produit⁹⁸ mais permet de documenter le texte en cours de rédaction ou de cacher une partie du texte qui pourra par la suite être « dé-commentée ».

Autres commandes utiles

Logo ORCID

On peut, dans un document, placer le logo (et le lien vers) ORCID⁹⁹. 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}
```

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).

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}**.

98. Fonctionne aussi avec les autres formats comme le html.

99. Pour *Open Researcher and Contributor ID* : <https://orcid.org/>

```
\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.



Caractères spéciaux

En ajoutant le paquet ***dingbat***, il est possible d'insérer des caractères spéciaux¹⁰⁰ 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 **Times**, 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.

100. <https://freefontsvault.com/itc-zapf-dingbats/>

LaTeX propose aussi sa série de caractères spéciaux¹⁰¹. Il faut s'assurer, pour pouvoir les utiliser, d'avoir ajouté **fontenc** dans les packages :

```
\usepackage[T1]{fontenc}
```

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.

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.

101. Voir par exemple ici : https://latex-programming.fandom.com/wiki/List_of_LaTeX_symbols

La mise en page

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.

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 ¹⁰²	non ¹⁰³
thanks ¹⁰⁴ :	oui	oui	oui

Les autres métadonnées (essentiellement *Keywords*, *DOI* et *Right*) sont enregistrées dans le document mais n'apparaissent pas sur la première page¹⁰⁵.

Options de mise en page

Pour rappel, dans l'en-tête YAML, l'instruction *geometry*: permet de préciser le format de la page (**a4paper** par exemple) mais aussi les quatre

102. Apparaît sur la deuxième page.

103. La présence d'un résumé dans l'entête YAML va provoquer une erreur.

104. Note sur le titre.

105. Pour un document HTML, l'option `-s` dans la commande *Pandoc* va les inclure dans le *header* de la page.

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}`) ;
- **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).

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
```

pour imprimer le numéro de la page courante ;

```
\thesection
```

pour imprimer le numéro de la section courante ;

```
\thechapter
```

pour imprimer le numéro du chapitre courant avec un document du type **books** ou **report** ;

```
\leftmark
```

- avec un document du type **article**, pour imprimer le nom de la section courante ;
- avec un document du type **books** ou **report**, pour imprimer le nom du chapitre courant ;
- avec un document du type **article**, pour imprimer le nom de la sous-section courante ;
- avec un document du type **books** ou **report**, pour imprimer 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 :



Remplissage des pages

Pour améliorer la mise en page finale, pour que la dernière ligne du texte se trouve sur la dernière ligne de la page, on va utiliser `\vfill` (à placer au début du document).

Cette instruction va provoquer l'insertion d'espaces blancs pour ajuster automatiquement le remplissage de la page.

Numérotation des pages

À la différence d'un fichier HTML (des pages web) ou d'un fichier ePub (le format standard des livres numériques), un fichier pdf est paginé,

c'est-à-dire découpé en pages. Il est possible, à tout moment, de modifier la présentation de 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 document¹⁰⁶.

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}
```

Indentation

Par défaut, il n'y a pas de retrait en début de paragraphes. Pour ajouter une indentation des paragraphes, il faut placer, au début du document, après l'en-tête YAML :

```
\setlength{\parindent}{20pt}
```

Cette commande va provoquer un retrait systématique de vingt points au début de chaque paragraphe.

Positionnement du texte sur la page

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.

106. C'est le comportement par défaut pour la classe **Book**.

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.

Verticalement

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

La commande `\vspace{x}` va ajouter des lignes vides. `\vspace{1.5cm}` va par exemple « pousser » la suite 1,5 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.

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

```
\usepackage{setspace}
```

ajouté 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`) à la ligne précédente, on aura :

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

Texte (notes) dans les marges

```
Texte texte \marginpar{ceci est une note courte} texte texte
```

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 }¹⁰⁷.

Pour avoir un texte dans un caractère plus petit, on peut ajouter `\scriptsize` au début de la note :

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

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

Néanmoins, contrairement à la fonction « notes », il n'y a pas de numérotation automatique des notes ajoutées dans les marges.

On peut également créer des notes dans les marges avec **marginnote**¹⁰⁸.

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.

107. Ne fonctionne pas si le texte est en colonnes via `\usepackage{multicol}` mais bien via **classoption: twocolumn**.

108. Il faut l'activer dans l'entête avec `"\usepackage{marginnote}"` et remplacer `"\marginpar"` par `"\marginnote"`.

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  
- margin=2.5cm
```

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.



```
"input" — 2024/10/26 — 9:26 — page I — #1
```

Les illustrations

Tableaux

Pour les tableaux dont 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{\1.2}`¹⁰⁹ pour ajuster l'espace 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 (voir plus haut [« taille des caractères »](#)).

Numérotation des équations

Si une légende est ajoutée à l'image (entre les crochets) ou un titre au tableau (: suivi du titre, dans le bas du tableau), la numérotation est automatique. Il n'en est pas de même pour les équations.

Pour assigner (manuellement) un numéro aux équations, il faut utiliser la commande `\tag{number}`.

```

$$
\tag{1.1}
V_{\text{sphere}} = \frac{4}{3}\pi r^3
$$

```

Cette commande va ajouter le numéro **1.1** à l'équation.

109. `\renewcommand{...}` permet de redéfinir une commande existante.

Liens

Pour appeler les tableaux et figures dans le texte, il faut faire appel à *pandoc-crossref*¹¹⁰.

Pour appeler une figure dans le texte, il faut ajouter une ancre après le titre de la figure, sans espace :

```
![Légende le figure](figure.png){fig:logo}
```

et dans le texte :

```
texte texte ([voir figure x]{#fig:logo})...
```

Pour les tableaux, ce sera la même méthode.

Positions

Par défaut, *LaTeX* va placer les figures et les tableaux 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 tableaux, 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.

110. <https://github.com/lierdakil/pandoc-crossref>, à installer.

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. On utilise :

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

On peut aussi insérer, via l'en-tête YAML, une image au-dessus (ou en dessous) du titre en ajoutant :

```
title: |
  
  Titre du document
```

Cette image peut avoir la taille de la page et dès lors constituer la couverture du document.

Graphiques

Markdown permet de créer des graphiques professionnels avec **Mermaid**¹¹¹. Ces graphiques peuvent être des :

- organigrammes ;

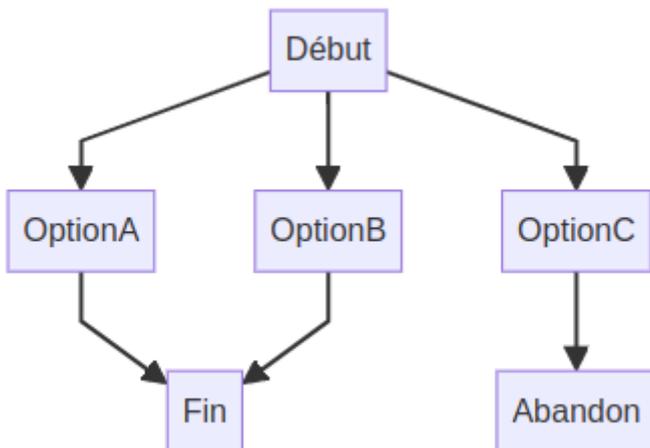
¹¹¹. <https://github.com/mermaid-js/mermaid>

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

Par exemple :

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

donnera :



*Un exemple de graphique créé avec **Mermaid***

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

- mermaid.d.foundation¹¹² ;
- mermaid.live¹¹³.

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

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

112. <https://mermaid.d.foundation/>

113. <https://mermaid.live>

Les index

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**.

Table des figures et table des tableaux

Les tables (t et f) sont automatiquement générées en ajoutant :

```
- tof: true  
- tot: true
```

dans l'en-tête YAML.

Elles apparaissent en fin de document.

Index de mots-clés

Avec Markdown et *Pandoc*, il n'est pas possible d'activer la fonction intégrée dans *LaTeX*¹¹⁴. Il faut donc détourner le problème en :

- identifiant les termes à mettre dans l'index avec `\label{terme}terme` ;
- ajouter, en fin de document (après un titre **# index** par exemple), les termes avec :

```
terme\dotfill\pageref{terme}
```

`\dotfill` va aligner le numéro de page à droite avec une ligne de points et `\pageref{terme}` va insérer le numéro de la page où le terme a été « tagué » (avec `\label{terme}terme`).

(liste à trier alphabétiquement).

Index

terme1	4
terme2	2
terme3	10

Un exemple d'index

114. Cet index est généré avec plusieurs compilations successives.

Créer un diaporama

Les formats pdf, HTML (des pages Web) et ePub (un ensemble de pages HTML rassemblées dans un fichier compressé et destiné aux liseuses) sont souvent utilisés pour exporter des documents Markdown. C'est cependant l'utilisation, plus rarement évoquée, de Markdown pour créer des présentations qui doit aussi retenir notre attention.

Depuis l'apparition des projecteurs d'écrans, nous sommes habitués à créer des présentations, des *slides*, avec une suite bureautique, *Microsoft Office* ou, mieux, *LibreOffice*.

La question « avez-vous une présentation *PowerPoint* ? » fait d'ailleurs partie du langage courant et peu de personnes se posent la question d'une éventuelle alternative.

Markdown et *Pandoc*, associés à **Beamer** (une classe particulière de *LaTeX* pour la création de diapositives), sont justement une alternative non seulement libre (ce que *LibreOffice* fait déjà) mais également plus :

- 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 de **Beamer**.

Beamer

Pour pouvoir produire une présentation sous forme de diapositives, on doit activer **Beamer**¹¹⁵. **Beamer** propose une série de **thèmes**¹¹⁶ qui modifient la couleur, les styles et la disposition des informations sur la

115. <https://fr.wikipedia.org/wiki/Beamer>

116. <https://hartwork.org/beamer-theme-matrix/>

page. Les deux exemples ci-dessous utilisent le *theme* **Madrid** et le *color-theme* **seahorse**.



Quelques exemples de thèmes Beamer

Il y a de nombreuses autres possibilités.

How to search and organize documents?

Bernard Pochet, PhD (ULiège Library)

2021 (cc-by)



Bernard Pochet, PhD (ULiège Library)

How to search and organize documents?

2021 (cc-by) 1 / 98

La première diapositive qui reprend les métadonnées de l'en-tête YAML

Le code suivant :

```
# What is the best way to ask a question?
```

```
Doing a literature search is not « looking for information about. »
```

You must :

- start with a question (a sentence with a « ? »)
- to obtain an answer(s) to this question

Therefore :

- write a « real question »
- identify the concepts present in this question
- search for the keywords related to these concepts
- write the documentary question

donnera la diapositive ci-dessous (avec deux cadres).

What is the best way to ask a question?

Doing a literature search is not "looking for information about."

You must :

- start with a question (a sentence with a "?")
- to obtain an answer(s) to this question

Therefore :

- write a "real" question
- identify the concepts present in this question
- search for the keywords related to these concepts
- write the documentary question

Bernard Pochet, PhD (ULiège Library) How to search and organize documents? 2021 (cc-by) 7 / 98

Une diapositive avec deux cadres

Les commandes de base

La majorité des commandes Markdown sont utilisables pour produire une présentation **beamer**.

Les commandes spécifiques les plus utiles sont :

- « # » pour créer une nouvelle page avec un titre courant (qui suit le signe « # » dans la source) ;
- « --- » (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 (![légende](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 équations également.

L'en-tête YAML

Pour entrer les métadonnées de votre présentation (titre, sous-titre, auteur et date) et quelques instructions (le format, le thème à utiliser ou la taille des caractères), vous devez créer un en-tête YAML spécifique.

Voici un exemple d'en-tête utilisée pour une présentation :

```
---
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
linkcolor: blue
---
```

Avec le thème **Madrid**, l'auteur, le titre et la date apparaissent en pied de page de chacune des diapositives (voir exemples ci-dessus).

« *titlegraphic* » ajoute un logo sur la première page. « *logo:* » insère une image en arrière plan sur toutes les diapositives.

La création du fichier pdf

Pour exporter la présentation, qui est au format Markdown, dans le format pdf qui sera projeté (mode « présentation »), la commande *Pandoc* sera la suivante :

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

Il est bien évident que les présentations créées avec cette méthode, comme les textes vus plus haut, sont moins « *funs* » que celles créées avec *PowerPoint* ou *Presentation* de *LibreOffice* ou directement avec *LaTeX*. Pour rappel, l'objectif reste toujours de simplifier et d'ouvrir la tâche.

Beaucoup de fonctions (animations, mises en page compliquées...) ne sont pas accessibles avec cette solution et c'est donc un choix de rapidité, d'efficacité, de propreté, de durabilité et d'ouverture que l'on fait en choisissant de créer des présentations avec Markdown.

Autres pistes

Avec *Remarkjs*, la création de diaporamas est encore plus légère et simple. Basé sur le format *GitHub flavor Markdown*, le fichier html est directement interprété par le navigateur et les diapositives sont visualisées via le navigateur web.

Il n'est pas obligatoirement destiné aux personnes connaissant les codes HTML et CSS. Il ne nécessite qu'un simple éditeur. Voir cet exemple : https://opensolution.be/slides/Slides_rem.html (enregistrer le fichier ou presser CTRL+u (avec *Firefox*) pour voir le code de la présentation que l'on peut copier et sauver dans un fichier HTML puis modifier).

Une autre solution simple est proposée par l'application *Marmota*¹¹⁷.

Il est également possible de créer des présentations au format web avec *s5*, *slidy*, *slideous*, *dzslides* ou *revealjs*. La transformation se fait également avec *Pandoc*. Il faut indiquer « **-t FORMAT** » (un des cinq formats) dans la commande *Pandoc* et utiliser l'extension *.html* en format de sortie.

```
pandoc -t revealjs votrefichier.md -o votrefichier.html
```

reveal.js est un peu plus complexe que *remark.js* qui ne nécessite qu'un seul fichier HTML. Il y a plusieurs façons de travailler avec *reveal.js* :

- en utilisant *Pandoc* comme ci-dessus (voir les instructions : <https://gist.github.com/jsoma/629b9564af5b1e7fa62d0a3a0a47c296>) ;
- en téléchargeant l'ensemble des fichiers nécessaires : <https://github.com/hakimel/reveal.js/archive/master.zip>. Ces fichiers, pour que les diapositives soient en ligne, doivent être téléchargés sur un serveur. C'est le fichier *index.html* qui contient toutes les diapositives et qui doit être modifié (voir exemple : <https://opensolution.be/revslides/>) ;

117. À installer, voir : <https://marmota.app/>.

- en installant l'application. Toutes les instructions sont présentées ici : <https://revealjs.com/>.

4. Exemples et modèles

Les pages qui suivent présentent des exemples et modèles d'utilisation de Markdown (associé à *Pandoc* et *LaTeX*) pour la création de documents professionnels.

Chacun des modèles peut être copié-collé ou téléchargé¹¹⁸ pour démarrer un nouveau projet.

118. <https://opensolution.be/docs/modelesmd.zip>

Un article

Le manuel « Comprendre et maîtriser la lecture scientifique¹¹⁹ » présente deux chapitres à consulter avant de se lancer dans la rédaction d'un article :

- La rédaction d'un document scientifique¹²⁰ ;
- La rédaction d'un article scientifique¹²¹.

(voir aussi le support de cours¹²²)

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 chapitre outils de la deuxième partie). On enregistre le document, en indiquant **.md** comme extension (typiquement **article.md**).

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 :

119. <https://opensolution.be/CoMLiS>

120. <https://opensolution.be/CoMLiS/ch14.html>

121. <https://opensolution.be/CoMLiS/ch15.html>

122. <https://opensolution.be/fowrite>

- 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 IMReD¹²³) ;
- de préciser, pour chaque autrice et auteur, un identifiant ORCID¹²⁴ et l'apport individuel en utilisant éventuellement la taxonomie CRe-diT¹²⁵.

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.

Si l'éditeur fournit un document modèle aux formats **.odt** ou **.docx**, on peut l'utiliser directement pour produire le manuscrit à transmettre :

```
pandoc --template=modele.docx article.md -o article.docx
```

123. Introduction, Matériel et méthode, Résultats et Discussion.

124. <https://orcid.org/>

125. <https://credit.niso.org/>

Bibliographie

La liste bibliographique et les citations dans le texte sont des éléments essentiels pour 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*¹²⁶, 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 **cs1** (*Citation Style Language*) correspondant au style bibliographique que vous souhaitez utiliser (l'éditeur vous demandera peut-être un style spécifique mais vous fournira généralement le fichier **cs1** 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*]¹²⁷ 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* :

126. <https://zotero.org>

127. *@auteur0000* est la clé de citation enregistrée dans le gestionnaire bibliographique.

```
pandoc --citeproc article.md -o article.docx
```

Modèle

Ci-dessous un modèle élémentaire que vous pouvez copier-coller dans votre éditeur de texte pour débiter 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** parce qu'il ne fonctionne pas pour les exports **docx** (on va néanmoins introduire le n^o 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
lang: fr
header-includes:
  - \usepackage[sfdefault]{noto}
title: Titre
subtitle: Sous-titre
author:
  - Auteur1^[Orcid1, université 1]
  - Auteur2^[Orcid2, université 2]
date: 2024
right: CC-BY
```

```
geometry:  
  - margin=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
```

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
```

Voir ce que donne ce modèle : <https://opensolution.be/docs/article.pdf>¹²⁸

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 -> Suivi 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...

128. Et un exemple plus complet : <https://opensolution.be/docs/5PMIS2020.pdf>

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*).

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.

Beamer

La création de documents avec **Beamer** va se faire comme avec les documents présentés dans les deux modèles précédents. Voici un exemple : <https://opensolution.be/docs/Friday.pdf>.

Modèle

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

- on choisit la *classe* (**output: beamer_presentation**) ;
- on choisit un *theme*¹²⁹, 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 (**title-graphic: logo.png**) et celui qui servira de fond d'écran sur toutes les diapositives (**logo: fond.png**).

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)¹³⁰.

129. <https://hartwork.org/beamer-theme-matrix/>

130. Il faut veiller à ne pas choisir un titre trop long.

```
---  
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
```

Création du fichier pdf

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

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

Remark.js

Avec Remark.js¹³¹, 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 : <https://opensolution.be/fooa/> (poussez CTRL-u pour voir le contenu du fichier).

Modèle

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 fonds 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 projet¹³².

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titre</title>
    <meta charset="utf-8">
    <style>
      body { font-family: "Noto Sans", "Arial", "Helvetica", sans-serif; }
    </style>
  </head>
  <body>
```

131. <https://remarkjs.com/>

132. <https://github.com/gnab/remark/wiki>

```
<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>
```

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).

Un curriculum

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 chapitre « outils » de la deuxième partie), et on enregistre le document, en indiquant **.md** comme extension (typiquement **curriculum.md**).

Modèle

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

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

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

```
---
documentclass: article
lang: fr
header-includes:
- \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\
nationalité : européen\
état civil : pas marié, pas d'enfant\
courriel : nom.prenom@mailo.fr\
Mastodon : https://mastodon.zaclys.com/\@nomp\
LinkedIn : https://be.linkedin.com/in/nomp\
Blog : https://monblog.org

---

## 1. Diplômes universitaires

- Master ... ;
- Doctorat en ....

## 2. Autres diplômes

- ...

## 3. Emplois

- ...

### Autres mandats :
```

- ... (dates)

Anciens mandats ou activités significatives :

- ... (dates)

4. Autres activités et hobbies

- ...

5. Publications (et activités de recherche)

- ...

6. Compétences linguistiques

Langue maternelle : français

autres langues :

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

Résumé des domaines d'expertise

- ...

Coordonnées privées :

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

Coordonnées professionnelles :

Université Machin-chose\
tél. : ...\
Courriel : 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 : <https://opensolution.be/docs/curriculum.pdf>

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

Compléments

Sites de référence

- Élaboration et conversion de documents avec Markdown et Pandoc (EPFL) : <https://www.jdbonjour.ch/cours/markdown-pandoc/>
- Rédaction durable avec Pandoc et Markdown (Programming Historian) : <https://programminghistorian.org/fr/lecons/redaction-durable-avec-pandoc-et-markdown>
- Tutoriel et aide mémoire Markdown (Arthur Perret) : <https://www.arthurperret.fr/tutomd/>
- Catalogue Markdown : <https://markdown.forge.apps.education.fr/>
- Markdown Guide (Matt Cone) : <https://www.markdownguide.org/>
- [Pandoc MANUAL](#) (il existe une version française¹³³ du manuel mais elle n'est pas tout à fait à jour).

et deux blogs incontournables de passionnés, respectivement à Lyon et à Grenoble :

- Celui d'Arthur Perret : <https://www.arthurperret.fr/>
- Celui d'Antoine Fauché : <https://www.quaternum.net/>

Voir aussi les **supports de formation** (format *Remak.js*) associés à ce manuel : <https://opensolution.be/fomd/> et <https://opensolution.be/fomd2>

133. <https://github.com/jgm/pandoc/wiki/Pandoc-User's-Manual-in-French>

Exercices

Comment ne pas envisager de compléter ce manuel avec deux exercices qui permettent de débiter et de se familiariser avec la solution Markdown + *Pandoc* + *LaTeX*.

Exercice pratique 1

1. Avec votre navigateur, ouvrez : <https://markup.rocks/> et videz l'écran de gauche
2. commencez à écrire et créez :
 - un titre de niveau 1 et un titre de niveau 2
 - une liste normale
 - une liste numérotée
 - des mots en italique
 - des mots en gras
 - un texte sous la forme de citation
 - un tableau avec 3 lignes et 3 colonnes

Exercice pratique 2

(instructions basées sur l'utilisation d'une machine *Windows*, à adapter en fonction de votre poste de travail)

1. **créez un répertoire** « md » sur votre poste de travail (à la racine du disque dur)
2. **téléchargez** le fichier <https://opensolution.be/docs/exemple.zip> dans ce répertoire
3. **décompressez** le fichier (vous devriez avoir trois fichiers)
4. ouvrez le fichier « exemple.md » avec un éditeur de texte (*Note-pad*, ...) pour voir le contenu et éventuellement le modifier

5. ouvrez le **mode terminal** sur votre poste (sous *Windows*, entrer « cmd » dans la zone de recherche)
6. **déplacez-vous** dans le répertoire « md » où se trouve le fichier que vous avez téléchargé/modifié avec la commande « cd » (pour *change directory*)
7. entrez la **commande**¹³⁴

```
pandoc --citeproc exemple.md -o exemple.pdf
```

Ensuite, testez la même commande en remplaçant **pdf** par **docx** et **odt** et retournez en mode *Windows* et ouvrez les fichiers créés avec *Pandoc*.

134. *Pandoc* et *LaTeX* doivent être installés.

Aide mémoire

titres	# Titre de niveau 1 ## Titre de niveau 2 ### Titre de niveau 3 #### Titre de niveau 4 ##### Titre de niveau 5 ##### Titre de niveau 6
liste simple	– item 1 – item 2 + item 3
liste numérotée	1. item 1 1. item 2 1. item 3
italique	*Ceci est un texte italique*
gras	**Ceci est un texte gras**
barré	~~Ceci est un texte barré~~
indice	H~2~O
exposant	X^2^
mise en évidence/ citation	">" en début de paragraphe
codes et texte non formaté	encadrer le texte (ligne précédente et ligne suivante) avec trois accents graves consécutifs <i>ou</i> pour une seule ligne, faire précéder le texte par une tabulation ou au moins trois espaces
lien	[texte à afficher](https://url_cible.org) <i>ou</i> [texte à afficher](#chap10)
ancres	# chapitre 10{#chap10}
image	![légende](image.png)

tableau	<pre> gauche centre droite :— :— :— cellule 1.1 cellule 2.1 cellule 3.1 cellule 1.2 cellule 2.2 cellule 3.2 cellule 1.3 cellule 2.3 cellule 3.3 : Titre du tableau </pre>
équation mathématique	$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
note de bas de page	<pre> ceci est le texte^[Et ceci est la note.] ou ceci est le texte[^numéro de la note] et, en fin de document [^numéro de la note]: Et ceci est la note. </pre>
citation dans le texte	texte texte texte [@auteur1234]
tiret long (cadratin)	---
tiret semi-long (demi-cadratin)	--
espace insécable	
insérer un saut de page	\newpage
insérer la date du jour	\today
insérer une ligne horizontale	***
empêcher un saut de paragraphe	\ à la fin de la ligne
cadre de texte	\fbox{texte texte texte}
texte en couleur	\textcolor{red}{texte en rouge}
texte en petites capitales	[Texte texte texte]{.smallcaps}

Pour aller plus loin...

- *MarkLab* (Une série d'applications qui utilisent Markdown – Cédric Eyssette) : <https://marklab.forge.apps.education.fr/#les-applications>
- Vous trouverez sur le site [Pandoc.org](https://pandoc.org) une grande quantité d'informations utiles (dérouler l'onglet « documentation »)
- *Markdown tools* pour manipuler des fichiers Markdown (html, csv, json...) : <https://www.markdowntools.com/>
- *Awesome Markdown Editors & (Pre)viewers* : <https://github.com/mundimark/awesome-markdown-editors>
- Convertir en Markdown des fichiers Word, Excel, PowerPoint, PDF : <https://msftmd.replit.app/>
- Quelques conseils/astuces supplémentaires, jdha0's digital space : <https://jdha0.github.io/tags/markdown/>
- Une référence HTML5 (pour rappel, il est possible d'intégrer des codes html dans des pages *Markdown*) : <https://www.w3schools.com/html/default.asp>
- Quelques conseils d'intégration d'éléments *LaTeX* dans un document *Markdown* : https://www.fabriziomusacchio.com/blog/2021-08-10-How_to_use_LaTeX_in_Markdown/
- Une référence à propos de *LaTeX* : <https://texdoc.org/serve/latex2e.pdf/0>
- Intégration de diagrammes avec Mermaid : <https://mermaid-js.github.io/mermaid/#/>
- Éditeur en ligne de diagrammes Mermaid : <https://mermaid-js.github.io/mermaid-live-editor/>
- *Convert Markdown to HTML using a custom template* : <https://vimalkvn.com/pandoc-markdown-to-html/>
- *Pandoc Templates* : <https://pandoc-templates.org/>
- Quelques templates *LaTeX* (Overleaf) : <https://www.overleaf.com/latex/templates>

- Le gabarit Abrüpt (« Une boîte à outils pour créer facilement un livre numérique ou papier ») : <https://gitlab.com/cestabrupt/gabarit-abrupt>
- Le template *Eisvogel* : <https://github.com/Wandmalfarbe/pandoc-latex-template>
- Le template *Typademic* : <https://github.com/maehr/academic-pandoc-template/archive/master.zip>
- L'utilisation de *LUA* : <https://github.com/mpeterv/markdown>
- [Stackoverflow.com](https://stackoverflow.com) ... un site où vous trouverez certainement réponse(s) à vos nombreuses questions.

Conclusion

Les deux piliers sur lesquels repose l'écriture académique au format texte avec Markdown et *Pandoc* sont la science ouverte (au minimum le libre accès aux publications, des logiciels libres et des données ouvertes) et le *low-tech* (est-ce que c'est utile, facilement accessible et durable ?). Ils nous amènent à nous interroger sur l'adéquation de nos méthodes, nos formats et nos outils.

Celles et ceux qui auront, à la lecture de ce manuel, testé Markdown via un site en ligne, voire installé et testé quelques outils, se sont probablement rendu compte que cette pratique d'écriture permet de reprendre la main sur toutes les étapes de la rédaction, de l'écrit initial au partage final.

Avec Markdown, contrairement à ce que l'impératif fonctionnel nous impose (« nous nous occupons de tout, vous n'avez pas besoin de savoir comment »), nous avons toujours le choix des formats, des méthodes et des outils. Il faut revenir à des méthodes plus simples, faire les choses dans l'ordre : écrire, illustrer, éditer puis diffuser.

Changer ses habitudes, se familiariser à de nouveaux outils, à de nouveaux formats et à de nouvelles méthodes n'est bien évidemment pas simple. Il faut y aller progressivement. Partir d'un texte simple, un rapport de quelques pages, une courte présentation et tout devient progressivement possible. Cependant, comme lorsque l'on décide de faire le choix du « libre » en remplaçant *Word* par *LibreOffice* ou de remplacer *Windows* par une distribution *Linux*, le coût du changement (des habitudes, mais pas seulement) est souvent élevé. C'est un choix que l'on fait. Il ne peut pas être imposé. C'est une décision personnelle.

Contrairement à une première impression que l'on pourrait avoir, il ne faut pas être un informaticien chevronné pour maîtriser Markdown. Personnellement, je m'y suis intéressé, il y a maintenant dix ans, parce que je cherchais un outil de rédaction sans distraction. Le premier outil proposé à partir d'une simple recherche a été Typora... basé sur Markdown. Le reste est venu avec des changements progressifs d'habitudes.

Pour d'aucuns, reprendre la main, c'est aussi s'intéresser à l'outil « ordinateur ». Bien intégrer les notions de fichiers, dossiers et programmes. Savoir où ils sont. Savoir les nommer, renommer, déplacer et effacer sans attendre qu'une application décide de le faire à votre place pour « vous faciliter la vie... ». Et ça aussi c'est important.

Bibliographie

- Audet, R., Blanc, J., Bourassa, R., Brassard, L.-O., Casenave, J., Hourez, J., Miranda, X., Mellet, M., Monjour, S., Paquin, M.-O., Pons, L., Soubret, J.-L., Sauret, N., & Youssef, E. (2018). *Version 0. Notes sur le livre numérique*. CRC en écritures numériques. <https://hal.archives-ouvertes.fr/hal-01922682>
- Balkan, A. (2020, August 7). *What is the Small web?* <https://ar.al/2020/08/07/what-is-the-small-web/>
- Boismenu, G., & Beaudry, G. (2002). 2. L'édition à l'ère du numérique. In *Le nouveau monde numérique : Le cas des revues universitaires* (pp. 51–70). Presses de l'Université de Montréal. <https://doi.org/10.4000/books.pum.9264>
- Bonjour, J.-D. (2014). *Élaboration et conversion de documents avec markdown et pandoc*. <https://www.jdbonjour.ch/cours/markdown-pandoc/>
- Bordage, F., & Morelle, M. (2021). *Tendre vers la sobriété numérique*. Actes Sud.
- Crouzet, T. (2016). *La mécanique du texte*. Thaulk.
- Dehut, J. (2018). En finir avec Word ! Pour une analyse des enjeux relatifs aux traitements de texte et à leur utilisation. *L'Atelier des Savoirs*, 23, 1–17. <https://eriac.hypotheses.org/80>
- Dehut, J. (2023). *Rhétorique et informatique : à la source des humanités numériques*. [Thèse de doctorat, Université de Rouen Normandie]. <https://theses.hal.science/tel-04452718>
- Derrac, L. (n.d.). *Altnumérisme radical*. Récupéré le 12 février 2025, from <https://louisderrac.com/numerique-acceptable/>
- Desgraupes, B. (2019). *LaTeX : Apprentissage, guide et référence* (3e éd). De Boeck supérieur.
- Dufrasne, J. (2024). *Rapport final du projet interuniversitaire AcOBE*. Université de Liège. <https://orbi.uliege.be/handle/2268/312724>
- European Union. (2021). *Study about the impact of open source software and hardware on technological independence, competitiveness*

- and innovation in the EU economy shaping europe's digital future.* <https://digital-strategy.ec.europa.eu/en/library/study-about-impact-open-source-software-and-hardware-technological-independence>
- Epron, B., & Vitali-Rosati, M. (2018). *L'édition à l'ère numérique*. La Découverte.
- Fauchié, A. (2018). Markdown comme condition d'une norme de l'écriture numérique. *Réel Virtuel*, 6. <http://www.reel-virtuel.com/numeros/numero6/sentinelles/markdown-condition-ecriture-numerique>
- Fauchié, A. (2020). Les technologies d'édition numérique sont-elles des documents comme les autres ? *Balisages*, 1. <https://doi.org/10.35562/balisages.321>
- Fauchié, A. (2024). *Fabriquer des éditions, éditer des fabriques : reconfiguration des processus techniques éditoriaux et nouveaux modèles épistémologiques* [Thèse de doctorat, Université de Montréal]. <https://these.quatenum.net/>
- Fauchié, A., Mellet, M., & Vitali-Rosati, M. (2022, May 18). *The importance of single source publishing in scientific publishing* [Présentation]. <https://ssp.digitaltextualities.ca/>
- Kern, E., Dick, M., Johann, T., Naumann, S., Marx-Gómez, J., Fertsch, M., & Golinska, P. (2011). Green Software and Green IT: An End Users Perspective. In *Information Technologies in Environmental Engineering* (pp. 199–211). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-19536-5_16
- Lebert, M. (2016). *Le livre 010101 (1971-2015)*. *Le livre 010101 (1971-2015)*. <https://www.010101book.net/fr/>
- Mailund, T. (2019). *Introducing Markdown and Pandoc: Using Markup Language and Document Converter*. Apress. <https://public.ebookcentral.proquest.com/choice/publicfullrecord.aspx?p=5852233>
- Masutti, C. (2023). *Pour un web désencombré*. *Framatophe Blog*. Retrieved February 12, 2025, from <https://golb.stadium.link/post/20230307-webdesencombre/>
- Mpondo-Dicka, P. (2020). Le Markdown, une praxis énonciative du numérique. *Interfaces numériques*, 8, 1.1 Mo. <https://doi.org/10.25965/INTERFACES-NUMERIQUES.3915>
- Perret, A. (2022). *De l'héritage épistémologique de Paul Otlet à une*

- théorie relationnelle de l'organisation des connaissances* [Thèse de doctorat, Université Bordeaux Montaigne]. <https://these.arthurperret.fr/>
- Petit, H. (2023). *La sobriété numérique: Comment rester connecté en conscience*. Ulmer.
- Pierre, A. (2017). *De l'avantage de LaTeX sur les traitements de texte modernes*. <https://aurelienpierre.com/de-lavantage-de-latex-sur-les-traitements-de-texte-modernes/>
- Reichenstein, O. (2025). *Markdown and the slow fade of the formatting fetish*. <https://ia.net/topics/markdown-and-the-slow-fade-of-the-formatting-fetish>
- Rouquette, M. (2012). *(Xe)LaTeX appliqué aux sciences humaines* (p. 268). Atramenta. <https://shs.hal.science/halshs-00924546>
- Ruiz, É. (2020, April 18). *Apprendre à écrire ? La boîte à outils des historien-ne-s*. <https://boiteaoutils.info/2020/04/apprendre-ecrire/>
- Stallman, R. M., Liquète, V., & Lehmans, A. (2017). Le libre comme alternative pour l'école contemporaine? *Hermès*, n° 78(2), 104. <https://doi.org/10.3917/herm.078.0104>
- Taffin, N., & Melot, M. (2023). *Typothérapie : Fragments d'une amitié typographique*. C&F éditions.
- Taillandier, F. (2016). *La mouvance statique*. Frank Taillandier. <https://frank.taillandier.me/2016/03/08/les-gestionnaires-de-contenu-statique/>
- Usclat, E. (2017). *Web et édition numérique* [Diplôme national de master]. Enssib. <https://www.enssib.fr/bibliotheque-numerique/notices/67717-web-et-edition-numerique>
- Vitali-Rosati, M. (2024a). *Éloge du bug*. Zones. <https://www.editions-zones.fr/livres/eloge-du-bug/>
- Vitali-Rosati, M. (2024b). Comment pensent docx, TEI et ekdosis ? Habiter l'espace numérique. *Les temps qui restent*, 3. <https://les-tempsquirestent.org/fr/numeros/numero-3/habiter-les-formats-d-ecriture-ce-que-pensent-docx-tei-et-ekdosis>
- Zuboff, S., Formentelli, B., & Homassel, A.-S. (2020). *L'âge du capitalisme de surveillance : le combat pour un avenir humain face aux nouvelles frontières du pouvoir*. Éditions Zulma.

À propos de l'auteur

Auteur



Bernard Pochet

UNIVERSITÉ DE LIÈGE

<https://orcid.org/0000-0002-1873-1237>

Bernard Pochet, retraité depuis juin 2024, est titulaire d'un diplôme en sciences psycho-pédagogique (université de Mons), d'un diplôme en sciences de l'information (Université libre de Bruxelles) et d'un doctorat (Université de Liège).

Il a été directeur de la bibliothèque des sciences agronomiques de l'Université de Liège de 1995 à 2020. Maître de conférence en littérature scientifique, il a également été directeur des presses universitaires de Gembloux pendant 25 ans et directeur de publication de plusieurs revues scientifiques. Il a terminé sa carrière comme responsable de la cellule d'édition numérique ouverte de la bibliothèque de l'Université de Liège et responsable de son pôle OpenScience. Il anime actuellement des ateliers de formations à Markdown pour les doctorants.