

Baliser les sciences sociales : langages et pratiques

Question

Lorsque vous lisez une page Web, un article scientifique ou un curriculum vitae professionnel, vous vous doutez peut-être que le texte n'est pas toujours produit à l'aide d'un simple logiciel de traitement de texte comme Microsoft Word, Apple Pages ou LibreOffice Writer. La mise en page réglée au millimètre près, la qualité des figures et graphiques, le style des références, la présence d'éléments interactifs et la cohérence hiérarchique du texte sont difficiles à reproduire à l'aide d'un logiciel de traitement de texte régulier, entre autres. L'insertion de tableaux de régression, de figures et d'extraits de code de haute qualité graphique ainsi que leur personnalisation nécessitent une interface particulière.

Pour ces raisons et plusieurs autres, les chercheurs en sciences sociales font souvent appel aux langages de balisage, ou *markup languages*. Ceux-ci permettent de produire des documents et pages Web sans les limitations des logiciels de traitement de texte. Le présent livre, par exemple, est écrit à l'aide du langage de balisage Markdown et de la plateforme de publication Quarto. D'entrée de jeu, vous vous demandez peut-être quelle est l'utilité d'apprendre ces langages alors que les logiciels de traitement de texte sont nombreux, simples d'approche et en amélioration constante. Ce chapitre tentera donc de répondre aux questions suivantes : « Pourquoi apprendre à utiliser des langages de balisage? Dans quels contextes sont-ils plus utiles que les logiciels de traitement de texte? Comment les utiliser? » L'accent sera mis sur Quarto, L^AT_EX, BibTeX et HTML.

Le premier langage de balisage, le Generalized Markup Language (GML), a été inventé en 1969 par les chercheurs Charles F. Goldfarb, Ed Mosher et Ray Lorie pour la compagnie IBM. Goldfarb et ses collègues devaient intégrer trois applications créées avec des langages différents et avec une logique différente pour les besoins d'un bureau de droit. Même après avoir créé un programme qui permettait aux trois applications d'interagir, ces langages demeuraient différents et avaient chacun leur propre fonctionnement. Le développement de GML a permis de résoudre ce problème en standardisant et en structurant le langage : les mêmes commandes étaient utilisées pour accomplir les mêmes tâches dans chaque programme [goldfarb96]. GML a été amélioré durant les décennies suivantes et a été suivi par d'autres langages de balisage, dont L^AT_EX(1984), HTML (1993), XML (1998) et Markdown (2004).

Un langage de balisage constitue un ensemble de commandes qui peuvent être entremêlées à du texte afin de produire une action informatique. Chaque langage contient son ensemble de commandes cohérentes et complémentaires. De manière plus formelle, ces commandes sont nommées *balises* (*tags* en anglais) et inscrites par le chercheur lui-même au travers du texte. Les balises constituent une manière de communiquer avec le logiciel que vous utilisez dans un langage qu'il peut comprendre, par exemple pour lui indiquer que vous désirez qu'une section du texte soit écrite en caractères gras, en italique, à double interligne ou encore que vous souhaitez positionner une image d'une certaine manière au travers du texte. Cette interaction est rendue possible par la standardisation des langages de balisage : chaque balise correspond

à une action précise, peu importe le logiciel utilisé, la langue dans laquelle le texte est rédigé, le type d'ordinateur utilisé, etc. Dans votre document source, les balises sont entremêlées au contenu de votre document, puis au moment de compiler ce dernier, les balises disparaissent, produisent les actions informatisées qu'elles commandent et ne laissent comme document final que son contenu mis en page tel que vous l'avez défini via les balises utilisées.

Plusieurs langages de balisage existent et permettent d'effectuer différentes tâches. Le plus répandu est le langage HTML, qui permet de formater des sites web. Le langage XML, lui aussi très utilisé, permet de structurer de larges volumes de données. \LaTeX permet pour sa part de formater du texte et de créer des documents en format PDF. Markdown permet également de créer des documents de format PDF, mais aussi HTML et DOCX. Depuis 2014, le *package* R Markdown permet d'ajouter des extraits de code R à un fichier en langage Markdown. Depuis 2022, le système de publication Quarto permet d'intégrer des extraits de code R, Python ou \LaTeX à un fichier en langage Markdown. \LaTeX , Quarto et Markdown permettent aussi d'intégrer les références bibliographiques du système de traitement de références BibTeX, créé en 1985, qui constitue également un langage de balisage. Le explique la manière de citer les références en utilisant BibTeX par le biais de Zotero et Better BibTeX.

Les balises constituent une manière de donner manuellement des commandes au logiciel que vous utilisez. Par exemple, si vous utilisez Microsoft Word, vous avez accès à une panoplie de boutons qui vous permettent de formater votre texte. Les balises exercent les mêmes fonctions, mais de manière manuelle. Lorsque vous appuyez sur un bouton dans Word, celui-ci ajoute des balises au travers de votre texte, mais rend celles-ci invisibles dans l'interface que vous utilisez. Cela permet d'avoir un texte élégant et facile à lire, mais comporte aussi plusieurs inconvénients. Le principal inconvénient est que vous êtes condamné à avoir un pouvoir limité sur le formatage de votre texte. En effet, si les boutons à votre disposition ne vous permettent pas de réaliser une opération, celle-ci sera éternellement impossible à réaliser pour vous. A contrario, les langages de balisage permettent un contrôle presque infini sur les opérations que vous souhaitez réaliser. Incidemment, dans la mesure où vous utilisez le langage approprié pour la tâche que vous souhaitez accomplir, vous devriez être capable de donner exactement la commande nécessaire à votre logiciel. Les langages de balisage, bien qu'ils aient un coût d'apprentissage qui peut s'avérer important et qu'ils soient moins élégants qu'un simple document Word, vous offrent une plus grande flexibilité.

Afin d'utiliser un langage de balisage, il est impératif que le logiciel que vous utilisez puisse prendre en compte ce langage. Un logiciel permet rarement d'utiliser n'importe quel langage. Il est aussi impératif de bien utiliser le langage de balisage. En effet, comme pour les langages de programmation, les langages de balisage ne peuvent pas déduire ce que vous souhaitez leur faire comprendre. Si vous souhaitez mettre du texte en gras, vous devez utiliser les bonnes balises. La moindre erreur est fatale, puisqu'une erreur dans la balise que vous utilisez produira un message d'erreur, le logiciel ne réussissant pas à associer votre balise mal inscrite à une action informatisée. Conséquemment, il est impératif de bien vérifier les balises utilisées afin d'éviter toute erreur qui empêcherait votre document d'être compilé, c'est-à-dire d'être traduit

dans son format final.¹ Chaque caractère dans une balise est important et il y a rarement plus d’une seule manière de commander une action. Le positionnement des balises est lui aussi critique : il délimite la portion de texte à laquelle doit être appliquée l’action commandée par la balise.

Il est important de distinguer les langages de balisage des langages de programmation. En effet, ceux-ci sont similaires à certains égards, mais ont des vocations différentes. Les deux s’appuient sur un langage informatisé, mais les langages et leurs objectifs diffèrent. Un langage de programmation définit des processus informatisés alors qu’un langage de balisage permet d’encoder du contenu de manière à ce que celui-ci soit lisible tant pour l’humain que pour son ordinateur.

Dans le contexte de la recherche en sciences sociales, la programmation est généralement utilisée afin de récolter, d’analyser et de présenter visuellement des données. Une fois cartes, tableaux et graphiques produits, ceux-ci peuvent être enregistrés – par exemple en format PDF ou PNG – et inclus au sein d’un document qui sera formaté en utilisant un langage de balisage. De manière simple, le langage de programmation contribue à l’analyse alors que le langage de balisage est essentiellement utile afin de présenter les travaux de recherche, que ce soit dans un document écrit ou sur un site web. C’est principalement de cette manière que sont utilisés les langages de programmation et de balisage dans le cadre de la recherche en sciences sociales.

Réflexion théorique

La plupart des langages de balisage permettent de remplir l’une des deux fonctions suivantes, qui sont particulièrement importantes dans le contexte de la recherche en sciences sociales : produire des documents écrits et gérer des pages Web. Dans les deux cas, cependant, certains sites Web et applications permettent également de remplir ces fonctions, mais avec des limites importantes.

Pour l’écriture de documents très simples comme une liste d’épicerie ou des notes rapides pendant une conférence, les logiciels de traitement de texte sont tout-à-fait convenables : ils sont simples et rapides à utiliser, un formatage professionnel du document n’est pas de mise. Utiliser un langage de balisage pour des tâches de base n’est en effet pas nécessaire. Par contre, plus la complexité d’un document augmente, plus il devient difficile d’obtenir un résultat satisfaisant en utilisant un logiciel de traitement de texte tel que Word, Pages ou Writer. A contrario, \LaTeX permet de produire des documents de tous les niveaux de complexité, tel que démontré sur la Figure ?? . Quant à Markdown, sa courbe se situerait logiquement entre celles de \LaTeX et de Word. Plus généralement, utiliser un langage de balisage comme

¹Les logiciels permettent plus ou moins efficacement d’identifier les balises problématiques. Certains ne produisent qu’un message d’erreur sans donner d’indication sur la source du problème, alors que d’autres ciblent très spécifiquement la ligne de syntaxe où se situe la balise problématique.

L^AT_EXou Markdown² comporte plusieurs avantages par rapport aux logiciels de traitement de texte traditionnels. Ces avantages peuvent se résumer en quatre concepts : automatisation, personnalisation, flexibilité et qualité graphique.

```
\begin{figure} \begin{center} \caption{Utilité relative de Word et \LaTeX\ selon la complexité et la taille du document \label{latex-vs-word}} \includegraphics[width=0.75\textwidth]{../_SharedFolder_Latex/ssn/Graphiques/Chapitre6/word-vs-latex.png} \end{center} \footnotesize{Source~: Yannick Dufresne (2015).} \end{figure}
```

Premièrement, L^AT_EXet Markdown permettent d'intégrer une bibliographie *automatique* et professionnelle en utilisant BibTeX. Cette bibliographie peut être adaptée très facilement en différents styles bibliographiques reconnus ou en un style bibliographique personnalisé. Avec BibTeX, plus besoin de vérifier si le titre de l'article est toujours en italique, si le numéro de volume est toujours entre parenthèses ou si le nom de famille des deuxièmes auteurs est toujours avant ou après le prénom puisque tout ceci est fait de manière *automatique*. BibTeX comprend également les différences entre les types de sources (articles scientifiques, livres, sites Internet, etc.) et ajuste leur présentation en conséquence. De plus, si une des sources que vous citez n'est pas incluse dans la bibliographie, une erreur s'affiche, vous permettant d'identifier le problème plutôt que de vous retrouver avec une référence manquante. À l'inverse, si une source est retirée du texte, elle disparaît *automatiquement* de la bibliographie dans le document final mais demeure présente dans le fichier où se trouvent les références bibliographiques. Cela évite les aller-retour pour vérifier que chaque source de la bibliographie se trouve au moins une fois dans le texte et que chaque source dans le texte est citée en bibliographie. Grâce aux balises, en cliquant sur les références incluses dans le document, celui-ci change de page pour se retrouver automatiquement à l'entrée bibliographique associée. Les références BibTeX pour articles scientifiques peuvent être copiées-collées à partir de Google Scholar. BibTeX rend donc extrêmement simple et efficace l'utilisation des références bibliographiques grâce à sa capacité à *personnaliser* et *automatiser* leur présentation.

L'intégration de figures et tableaux dans le texte est aussi rendue très simple et professionnelle grâce à L^AT_EXet Markdown. La taille de la figure ou du tableau, son positionnement et son intégration par rapport au texte environnant peuvent être réglés de telle sorte que l'ajout de texte avant ou après la figure ou le tableau ne produira pas des résultats inattendus. Au contraire, en définissant des paramètres pour l'ensemble du texte, le chercheur pourra *personnaliser* entièrement la présentation des figures et tableaux. De plus, la qualité des figures et tableaux ne diminue pas lors de leur intégration : les figures restent aussi belles qu'elles l'étaient originalement, ce qui n'est pas toujours le cas avec certains logiciels de traitement de texte. Les numéros des figures et tableaux sont aussi mis-à-jour *automatiquement*, ce qui veut dire que vous n'aurez jamais à vous préoccuper de modifier leur numéro lorsque vous rajoutez une figure ou un tableau dans le texte. Grâce aux balises, en cliquant sur le numéro associé à

²Les avantages et désavantages de Markdown cités dans la prochaine section s'appliquent eux aussi à Quarto. Les avantages ou inconvénients ne s'appliquant qu'à Quarto et non à Markdown tout court sont présentés comme tels.

la figure ou au tableau dans le texte, le document se retrouve automatiquement à l'endroit où se trouve le graphique ou tableau.

Markdown et \LaTeX permettent aussi la gestion *automatisée* de la table des matières, et les références aux pages appropriées se mettent à jour en continu. La table des matières prend en compte l'architecture du texte choisie manuellement par le chercheur, qui est définie par des balises définissant différents niveaux hiérarchiques de sections, sous-sections ou chapitres. Des manières *automatiques* de référencer les figures et les tableaux dans des sections distinctes de la table des matières sont également offertes, encore une fois *personnalisables* au goût du chercheur.

Bien que la mise en page de documents produits via Markdown et \LaTeX puisse être définie entièrement manuellement par un utilisateur expérimenté, les débutants apprécieront les nombreux gabarits (*templates*) disponibles en libre permettent de gérer *automatiquement* la mise en page de manière clé-en-main. Ceux-ci permettent de rendre l'apparence d'un document plus esthétique et uniforme et peuvent être utilisés tels quels ou peuvent servir de point de départ pour un chercheur souhaitant y apporter certaines modifications sans toutefois partir d'une feuille blanche. La majorité des utilisateurs, même les plus expérimentés, utilisent ces gabarits comme base lorsqu'ils rédigent un document. Ceux-ci constituent une mine d'or puisqu'ils rendent accessible le code Markdown et \LaTeX ayant servi à la conception du gabarit, permettant au chercheur de comprendre comment est obtenu le résultat que lui offre le gabarit. Incidemment, le chercheur peut identifier les sections de code produisant certains éléments de mise en page (ex : positionnement des numéros de page, positionnement du nom des auteurs, etc.) et les modifier ou s'en inspirer afin de modifier d'autres gabarits. L'utilisation de ces gabarits peut s'avérer complexe pour les non-initiés, mais il s'agit d'une complexité qui s'avère ultimement extrêmement productive puisqu'elle permet au chercheur de devenir autonome et d'ajuster les gabarits à sa convenance afin de produire exactement le résultat désiré en terme de mise-en-page. La liste des gabarits disponibles est extrêmement large et ceux-ci peuvent servir une variété de fonctions. En effet, une variété de gabarits professionnels et de haute *qualité graphique* sont offerts gratuitement en ligne pour des articles, des livres, des rapports, des *curriculum vitæ* () ou encore des feuilles de temps pour des contrats rémunérés ().

Photos de CVs et feuilles de temps professionnels

Un autre avantage non-négligeable de Markdown – qui le distingue à cet égard de \LaTeX – est la *flexibilité* qu'il offre à ses utilisateurs. En effet, en utilisant Pandoc Markdown, qui est une extension du langage Markdown de base permettant de combiner plusieurs langages de balisage différents en un seul document, il est possible d'intégrer dans un seul document plusieurs langages de balisage différents tels que \LaTeX , HTML, CSS ou JavaScript ainsi que du code R en utilisant Quarto. Quarto est également habilité à travailler avec des fichiers Python dans des environnements de type Jupyter Notebook. Ceci permet donc à l'utilisateur de bénéficier des fonctionnalités de différents langages dans un seul document, rendant ainsi possible une variété de *personnalisations* qui ne seraient pas possible autrement. Qui plus est, il est aussi important de noter que Markdown permet de créer des fichiers Word réguliers, PDF professionnels et HTML à partir d'un même document. L'utilisateur peut donc choisir

à sa convenance et à tout moment de quelle manière sera compilée le document rédigé. Cette fonctionnalité est particulièrement pratique dans le cadre de collaboration avec des chercheurs n'utilisant pas les langages de balisage ainsi que lors de l'envoi de manuscrits à des revues scientifiques puisque certaines d'entre elles exigent de recevoir ceux-ci sous forme de document Word.

La facilité avec laquelle peuvent être intégrés et gérés les figures et tableaux dans des documents \LaTeX et Markdown a déjà été abordée, mais il est important de souligner que l'utilisation de l'extension Quarto permet d'ajouter une couche supplémentaire d'intégration. En effet, Quarto permet de créer une figure grâce à du code R, ainsi que d'intégrer celle-ci au texte et la formater en un seul document. Cela se fait grâce à l'intégration de R *code chunks* dans le document. Le code est produit dans le *chunk* et la figure ou le tableau qui en résulte apparaît dans le document Quarto *et* sur le document fini. La différence entre Quarto et \LaTeX est que ce dernier ne peut pas prendre en compte le code R et les figures et tableaux doivent donc être créées dans un document séparé avant d'être intégrées dans le document \LaTeX .

Bien que l'apprentissage de \LaTeX et de Markdown puisse être parsemé de nombreuses embûches, ces deux langages bénéficient d'une communauté d'utilisateurs en ligne sur laquelle il est possible de s'appuyer afin de résoudre tout problème rencontré. Les utilisateurs – particulièrement les plus expérimentés – sont nombreux à partager leur expérience à leurs collègues rencontrant des problèmes afin de contribuer à régler ceux-ci. Cette communauté est présente sur une multitude de sites Web, bien que le point de rencontre principal soit le forum @stackoverflow23, qui est également utilisé pour régler des problèmes de programmation en R. Une simple recherche sur Google d'un problème rencontré avec \LaTeX ou Markdown offrira à l'utilisateur des centaines de résultats pertinents afin de l'aider, la plupart de ces résultats étant probablement des échanges sur Stack Overflow. L'utilisateur pourra donc filtrer les résultats et observer les nombreuses solutions envisageables à son problème afin de définir laquelle est la plus appropriée dans sa situation. Il est important de noter, toutefois, que cette communauté est nettement plus développée pour les utilisateurs de \LaTeX que de Markdown, puisque ce dernier langage est moins répandu que le premier.

Autre preuve de leur grande *flexibilité* et capacité de *personnalisation*, certaines manières plutôt spécifiques de formater le texte sont présentement uniquement disponibles avec \LaTeX ou Markdown. C'est le cas de la possibilité de séparer automatiquement un mot en deux en fin de ligne à l'aide d'un tiret s'il est suffisamment long, ou encore de la possibilité de permettre que la dernière ligne d'un paragraphe apparaisse seule en haut d'une page – ou, à l'inverse, que la première ligne d'un paragraphe apparaisse seule en bas d'une page. Bien qu'il soit rare que nous ayons absolument besoin de personnaliser le texte de cette manière, ces possibilités peuvent s'avérer utile lorsque vous rédiger un texte qui doit se conformer en tout point à un gabarit spécifique. En effet, certaines revues scientifiques, maisons d'édition ou universités (dans le cadre de la rédaction de mémoires et thèses) imposent ce type de gabarit inflexible et parfois plutôt capricieux. C'est dans ce type de contexte que la *flexibilité* incomparable de \LaTeX et Markdown peut s'avérer utile.

Il est aussi important de noter que plusieurs revues scientifiques recommandent fortement, voire imposent, de leur soumettre des manuscrits en format \LaTeX ou Markdown afin de rendre plus facile pour leurs éditeurs d'adapter ceux-ci à la mise en page de leur revue.

Finalement, il est important de mentionner en terminant que Markdown et \LaTeX sont entièrement gratuits et accessibles aux utilisateurs de tous les systèmes d'exploitation.

Ceci dit, \LaTeX et Markdown comportent eux aussi leurs désavantages.

Premièrement, \LaTeX est difficile à apprendre. Certaines tâches qui peuvent sembler simples comme l'ajout d'un tableau peuvent nécessiter de nombreuses lignes de code. De plus, à la moindre erreur de frappe dans l'utilisation d'une balise, le code risque de planter et de ne pas produire le document PDF souhaité. C'est ce qu'on appelle une erreur de compilation. La compilation est le processus par lequel un document écrit en langage de balisage est transformé en fichier textuel, en format PDF dans le cas de \LaTeX . Markdown est un langage plus simple à apprendre, avec des balises plus courtes et intuitives. Il occasionne donc moins d'erreurs de compilation.

Deuxièmement, \LaTeX est incompatible avec Word, Pages ou Writer. Pour transférer un fichier de traitement de texte vers \LaTeX , les balises doivent être ajoutées manuellement une par une. À l'inverse, pour transférer un document \LaTeX vers un fichier de traitement de texte, les balises doivent être retirées une par une. Il est aussi possible de copier le texte directement à partir du fichier PDF produit par \LaTeX , mais les fins de ligne sont interprétées par Word, Pages ou Writer comme des retours plutôt que des espaces, et les accents sont souvent mal copiés et doivent être réécrits manuellement. Encore une fois, Markdown évite ce problème en permettant d'écrire un fichier DOCX à partir du langage de balisage. Le formatage du fichier DOCX demeure un peu compliqué cependant et doit être fait à partir du modèle d'un autre document DOCX formaté tel que souhaité. De plus, les fichiers DOCX ne peuvent pas être transformés en format Markdown.

Troisièmement, bien que plusieurs fonctions \LaTeX puissent être utilisées dans des fichiers Markdown, certaines demeurent incompatibles, ce qui rend certaines tâches possibles à faire uniquement par \LaTeX .

Quatrièmement, il n'y a aucun suivi des modifications en Markdown ou en \LaTeX . Pour réviser un travail fait dans l'un de ces deux formats, des commentaires peuvent être ajoutés sur le PDF avec des logiciels comme Adobe Reader, mais des commentaires peuvent aussi être faits directement dans le document \LaTeX ou Markdown. Ces commentaires sont identifiés à l'aide de balises et n'apparaissent pas dans le PDF résultant.

Cinquièmement, certaines revues scientifiques refusent les fichiers PDF et demandent que les soumissions soient faites en format DOCX, justement pour des raisons de suivi des modifications.

Sixièmement, R Markdown permet de visualiser les résultats d'un code R directement dans le document, mais dans certains cas cela occasionne des erreurs.

Septièmement, avec les logiciels de bureau qui permettent d'utiliser \LaTeX et Markdown, il est impossible de visualiser le résultat final en temps réel. La compilation est nécessaire au préalable.

Huitièmement, il n'y a pas de compteur de mots ou de caractères.

Neuvièmement, Markdown et \LaTeX impliquent la création d'un fichier `.TEX` ou `.MD` en plus d'un `.PDF`, un `.BIB` et plusieurs autres. Word ne nécessite qu'un document `.DOCX`.

Finalement, les langages de balisage permettent également de créer des pages Web. Bien que les pages Web puissent être créées à partir de sites Web comme WordPress, les langages de balisage permettent de produire des résultats plus personnalisables, plus automatisables et avec une plus grande qualité graphique également. Ainsi, l'utilisation de fichiers HTML, JavaScript et CSS, dont le code peut également être intégré dans un fichier Markdown, tout comme c'est le cas avec les fichiers \LaTeX , afin de produire des pages Web. HTML peut être appris via des cours sur Code Academy.

Somme toute, Word n'est pas à antagoniser. Il a ses qualités. Mais

Réflexion méthodologique

En pratique, comment utiliser Markdown, \LaTeX et BibTeX?

\LaTeX a une syntaxe particulière qui demande un certain temps d'adaptation. Pour écrire une phrase simple comme celle-ci, la phrase peut être écrite telle quelle. Par contre, pour mettre un **mot** en caractères gras, il faut utiliser la balise suivante: `\textbf{mot}`. Pour mettre le **mot** en rouge, la balise est `\textcolor{red}{mot}`. Pour le mettre en italique et en note de bas de page³, les balises `\footnote{\emph{mot}}` peuvent être utilisées. Ainsi, des balises peuvent contenir d'autres balises. En langage \LaTeX , une balise commence toujours par une barre oblique inversée. Par la suite, le nom de la fonction (*emph*, *textbf*, *textcolor*, etc.) est appelé. Enfin, généralement, le mot à formater est placé entre accolades (`{}`).

Chaque document \LaTeX commence par un préambule. Celui-ci présente des informations telles que la taille des caractères, le type d'article, le format de mise en page, la police de caractères, l'utilisation d'en-têtes et de pieds de page, ainsi que l'utilisation de *packages* \LaTeX permettant différentes fonctionnalités de personnalisation du document.

Il n'est pas nécessaire ni souhaitable d'apprendre l'ensemble des fonctions et des *packages* \LaTeX qui existent. Au contraire, il est souvent mieux de commencer par un gabarit de document qui convient au type de document que vous voulez créer et ensuite de rechercher en anglais sur Stack Overflow la manière d'ajouter des éléments de formatage que vous ne connaissez pas (par exemple, *highlight latex text*). Des gabarits sont disponibles sur Overleaf, au <https://fr.overleaf.com/latex/templates>.

³*mot*

Markdown fonctionne de manière similaire à \LaTeX , mais se démarque par sa plus grande flexibilité et sa syntaxe beaucoup plus légère. Par contre, Markdown est moins *user-friendly*, c'est-à-dire qu'il y est plus difficile de modifier l'aspect visuel d'un document. Tout document Markdown débute avec un court bloc de syntaxe **YAML** (acronyme de **Yet Another Markup Language**) qui définit les paramètres généraux du document. Voici un bloc **YAML** typique:

```
---
title: "Les langages de balisage"
subtitle: "Ça change pas le monde, sauf que..."
author:
  - Alexandre Fortier-Chouinard^[University of Toronto]
  - Maxime Blanchard^[McGill University]
output: pdf_document
documentclass: article
bibliography: references.bib
---
```

Outre le titre, le sous-titre et le nom des auteurs, on y trouve aussi le gabarit servant à construire l'aspect visuel du chapitre, la manière dans laquelle il est compilé – dans ce cas-ci, PDF – ainsi que le chemin d'arborescence afin d'accéder au document BibTeX où sont enregistrées les références utilisées. Il est aussi possible d'y définir la taille de la police ou encore le gabarit servant à définir le type de bibliographie qui sera utilisé. De manière particulièrement importante, c'est l'endroit où sont chargés les *packages* \LaTeX qui seront utilisés. En effet, la quasi-totalité des *packages* et fonctions \LaTeX sont utilisables dans Markdown, alors que l'inverse n'est pas vrai. Il est donc possible de personnaliser un document Markdown en utilisant des *packages* ayant été créés pour \LaTeX .

La syntaxe à utiliser au travers du texte est somme toute plutôt simple. Pour mettre un ou plusieurs **mots en gras**, il suffit de les entourer de deux astérisques (****mots en gras****); pour les mettre *en italique*, il faut les encadrer d'une seule astérisque (**en italique**). Pour définir un titre de section ou de sous-section, il suffit de mettre des # devant le titre en question. Plus vous ajoutez de #, plus le titre sera petit et plus il sera considéré à un niveau hiérarchique inférieur dans la structure du texte. La syntaxe Markdown est donc plus légère que celle de \LaTeX , dans le but d'en rendre la lecture plus simple pour son utilisateur.

Bien que des gabarits Markdown soient disponibles, ceux-ci sont plus rares. Ils se trouvent pour la plupart sur GitHub, rendus disponibles par leur créateur. Cela étant dit, leur personnalisation peut s'avérer plutôt complexe. En somme, Markdown est particulièrement pratique pour les documents ne nécessitant pas de respecter un gabarit précis et réquérant simplement un document d'allure simple et professionnelle.

Pour sa part, BibTeX a une syntaxe relativement simple. D'emblée, les références BibTeX pour des articles et ouvrages scientifiques sont disponibles sur Google Scholar. Toutefois, pour

citer des sites Web ou des articles de médias, la référence doit être écrite à la main selon un format précis. Une bibliographie sur BibTeX peut ressembler à ceci :

```
@book{darwin03,
  address = {London},
  author = {Darwin, Charles},
  publisher = {John Murray},
  title = {{On the Origin of Species by Means of Natural Selection
or the Preservation of Favoured Races in the Struggle for Life}},
  year = {1859}
}

@article{goldfarb96,
  title={The Roots of SGML: A Personal Recollection},
  author={Goldfarb, Charles F},
  journal={Technical communication},
  volume={46},
  number={1},
  pages={75},
  year={1999},
  publisher={Society for Technical Communication}
}
```

Un fichier BibTeX ne contient rien de plus qu'une série de publications commençant chacune par la balise @ suivie du type d'article – *article*, *book*, *incollection* pour un chapitre de livre, *inproceedings* pour une présentation dans une conférence, *unpublished* pour un article non publié et *online* pour un site Web sont parmi les plus connus – et des informations sur la publication mises entre accolades. La première information entre accolades est le code de la référence, par exemple `goldfarb96`. Dans le fichier L^AT_EX, l'auteur doit écrire `\cite{goldfarb96}` pour voir dans le document PDF compilé @goldfarb96; le lien est automatiquement cliquable et renvoie à la notice bibliographique correspondante. L'ordre des publications dans le document BibTeX a peu d'importance, puisque L^AT_EX réordonne par défaut la bibliographie en ordre alphabétique.

Trucs et astuces

Où puis-je utiliser ces langages de balisage? Contrairement à Microsoft Word et Apple Pages, plusieurs options

Logiciels de bureau

MacTeX, MikTeX et autres distributions \LaTeX

RStudio (exemples précédents), Visual Studio et autres logiciels du chapitre 5 (tous?)

Logiciels en ligne

Overleaf

VS code with LiveShare

Markdown en Overleaf

Références