

Rapport technique

Avec Markdown et l'aide de pandoc

Yoan Blanc*

23 novembre 2017

*yoan@dosimple.ch

This document aims at presenting how to build a technical report respecting the guidelines set by the Haute-École Arc using open source softwares and the Markdown format.

Keywords : *pandoc, markdown.*

Ce document présente sommairement comment construire un document respectant les critères de la Haute-École Arc à l'aide de logiciels libres et du format *Markdown*.

Mots-clés : pandoc, markdown.

Table des matières

1	Introduction	7
1.1	Rapport technique	7
1.2	Markdown	8
1.3	Pandoc	10
1.4	Installation	10
2	Réalisation	13
2.1	Le B.a.-ba	13
2.2	Bibliographie avec <i>pandoc-citeproc</i>	15
2.3	Références avec <i>pandoc-crossref</i>	16
2.4	Étendre <i>pandoc</i>	17
3	Conclusion	20
	Annexes	21
	Glossaire	21
	Sources	21

Liste des tableaux

1.1 formats d'entrée et de sortie. 10

Table des figures

2.1	Le trombone de Microsoft Word “Clippy” tout triste	17
-----	--	----

Liste des extraits de code

2.1	<code>bibliographie.bib</code>	16
2.2	<code>english.lua</code>	19

1 Introduction

Traditionnellement, les outils d'édition de rapport privilégiés des étudiants sont *Microsoft Word* ou *LibreOffice Writer*. D'autres, plus courageux, se lancent dans \LaTeX à l'aide d'un éditeur riche comme *LyX*¹ ou, l'éditeur en ligne qu'est *ShareLaTeX*².

L'approche présentée ici se place entre ces deux univers. *Markdown* est un format très facile à prendre en main, partagés, et *pandoc*³ offre une conversion vers PDF qui fonctionne. Les limitations de *Markdown* impliquent de recourir à d'autres outils pour gérer la bibliographie ou les références.

Cette approche a été utilisée pour le support du cours de troisième année Développement Web⁴, ainsi que par d'anciens étudiants fraîchement diplômés.

1.1 Rapport technique

Selon les directives de l'école (Perret, Raboud, Babey 2017), les difficultés d'un rapport technique au niveau de sa structure sont de réaliser correctement les éléments suivants :

- page de titre ;
- table des matières ;
- liste des figures, tableaux et/ou équations ;
- *abstract*, introduction et conclusion ;
- le fameux SPRI (Timbal-Duclaux 1982), ou IMMRID (voir le [glossaire](#)) ;
- et la bibliographie.

Parmi ceux-ci, les gros morceaux requièrent un gros investissement en terme de rédaction. Mais également, un travail d'organisation et de documentation très fin. Si constituer une bibliographie via Zotero⁵, BibDesk⁶, ou autre, est relativement aisé ; gérer les citations, références et formater le tout correctement peut être fastidieux. Tout comme documenter et référencer chaque image, équation ou table de données. Et c'est sans parler que c'est un travail de groupe dont il faut également coordonner les diverses contributions.

¹<https://www.lyx.org/>

²<https://www.sharelatex.com/>

³<https://pandoc.org/>

⁴<https://he-arc.github.io/slides-devweb/>

⁵<https://zotero.org/>

⁶<http://bibdesk.sourceforge.net/>

1.2 Markdown

L'idée est de reposer sur un format de fichier textuel, similaire à HTML mais avec une syntaxe plus légère. *Markdown* est inspiré d'un formalisme apparu *de facto* dans les listes de diffusion d'e-mails.

Ce format semi-structuré est devenu au fil des ans le favori des *geeks*. HTML étant trop fastidieux, ils n'ont eu de cesse d'imaginer des alternatives pour permettre l'édition de pages web au plus grand nombre, notamment au travers des *wikiwikiwebs* (ou simplement *wikis*), tel que *Wikipedia*.

“A Markdown-formatted document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions.”

– John Gruber

```
# h1 (titre 1)
## h2
### h3
#### h4
##### h5
##### h6
```

```
*emphase* ou _emphase_ (affiché en italique)
```

```
**forte emphase** ou __forte emphase__ (affiché en gras)
```

```
- liste
- non
- ordonnée
```

```
1. liste
2. ordonnée
```

```
liste
```

```
: de définition
```

```
[lien hypertext](https://example.org)
```

```
![texte alternatif](image.png)
```


> `citation`

`texte non-formaté`

L'avantage à utiliser du texte est qu'il est aisé à versionner (`svn`, `git`, etc.) et qu'il sépare les tâches de rédaction et de mise en forme. Notre cerveau, en mode écriture, est ainsi libre dans sa créativité et ne s'occupe pas de la mise en page.

GitHub est un grand utilisateur du format *Markdown* et propose un guide utile⁷ à son apprentissage. En matière d'édition collaborative, *HackMD.io*⁸ propose un système proche de *Google Docs*.

1.2.1 Formules mathématiques

Il est même possible de faire des tables HTML et même des formules mathématiques grâce à *MathJax*⁹.

$$-1 = e^{j\pi} \quad (1.1)$$

Le format utilisé est celui de *LaTeX*. Avec un signe dollar (\$) pour une formule en ligne, par exemple $e^{j\pi}$. Et deux \$\$ pour une formule centrée sur sa propre ligne (telle que eq. 1.1).

```
$$-1 = e^{j \pi}$$
```

1.2.2 Méta-données

Ce qu'il manque à Markdown est d'être capable de renseigner des informations à propos du contenu : des *métadonnées*. Le moyen de faire ceci est d'utiliser un prologue dans autre format, nommé *YAML*. Le prologue *YAML* est enfermé dans deux triple-tirets ---.

```
---
title: Mon titre
lang: fr
---
```

... mon rapport technique ...

Ici, nous avons renseigné des informations permettant de construire la page de titre du rapport, ainsi que d'appliquer les règles de formatage typographiques propre au français (*fr*).

⁷<https://guides.github.com/features/mastering-markdown/>

⁸<https://hackmd.io/new>

⁹<https://www.mathjax.org/>

1.3 Pandoc

Un logiciel très puissant issu du temps libre d'un certain John MacFarlane, professeur de psychologie à l'université de Berkeley. Supportant pléthore de formats différents, il est capable de les convertir de l'un à l'autre selon la table ci-dessous (tab 1.1).

TAB. 1.1 : formats d'entrée et de sortie.

entrée	sortie
docbook,	idem +
docx,	beamer,
epub, html,	dzslides,
ΛT _E X,	epub3,
mark-	html5, pdf,
down,	revealjs,
mediawiki,	rtf, s5,
odt, rst,	slideous,
textile,	slidy, ...
twiki, ...	

La liste complète des options à disposition se trouve dans l'aide de la commande.

\$ `man pandoc`

1.3.1 Alternatives et variantes

S'il est possible de tout faire avec *Pandoc*, un usage intermédiaire est envisageable où le squelette du rapport est réalisé avec ΛT_EX et le contenu des différentes sections écrit en Markdown pour des raisons de simplicité. Un template pour la HES-SO a été créé par Marc Demierre¹⁰ et peut servir de base.

Sphinx¹¹ est une bonne alternative si le but est de produire une documentation technique en lien avec un code source.

1.4 Installation

Se lancer dans la production de document PDF avec ΛT_EX demande de télécharger la moitié d'Internet et peut être déroutant.

¹⁰<https://github.com/mdemierre/hesso-latextemplate-thesis>

¹¹<https://sphinx.org/>

La suite des opérations est prévue pour fonctionner sous Ubuntu ainsi que Ubuntu sous Windows (aussi nommé WSL (*Windows Subsystem for Linux*) disponible sous Windows 10 via le *Store*). Elle a été testée dans l'environnement suivant :

```
$ lsb_release -a
Distributor ID: Ubuntu
Description:    Ubuntu 16.04.3 LTS
Release:        16.04
Codename:       xenial
```

Depuis un terminal, installez quelques dépendances à l'aide d'`apt`.

```
$ sudo apt install \
    pandoc \
    pandoc-citeproc \
    texlive-fonts-extra \
    texlive-lang-english \
    texlive-lang-french \
    texlive-math-extra \
    texlive-xetex \
    ttf-mscorefonts-installer
```

Quoi est quoi ?

- `\` évite de tout avoir sur seule ligne, c'est une astuce pour mettre de longues commandes dans un fichier imprimable tel que celui-ci ;
- `pandoc` convertit des formats textes en d'autres, ici de *Markdown* vers \LaTeX ;
- `pandoc-citeproc` gère les citations et la bibliographie ;
- `texlive-fonts-extra` plein de polices de caractères ;
- `texlive-lang-english` les bouts de code source seront en anglais ;
- `texlive-lang-french` notre contenu sera en français ;
- `texlive-math-extra` affichage des formules mathématiques ;
- `texlive-xetex` un meilleur support des polices et de l'Unicode ;
- `ttf-mscorefonts-installer` pour ceux qui veulent utiliser les polices Windows (Comic Sans, Impact, ...).

1.4.1 Pandoc 2

Les versions fournies avec Ubuntu Xenial (16.04) sont un peu vieilles. Une alternative est d'installer `pandoc` (et `pandoc-citeproc`) directement depuis GitHub¹².

```
$ wget https://github.com/jgm/pandoc/releases\
> /download/2.0.3/pandoc-2.0.3-1-amd64.deb
```

¹²<https://github.com/jgm/pandoc/releases>

```
$ sudo dpkg -i pandoc-2.0.3-1-amd64.deb
```

```
$ pandoc -v  
pandoc 2.0.3  
...
```

NB : les exemples suivants partent du principe que vous avez installés *Pandoc* version 2.

2 Réalisation

Une fois le langage de description *Markdown* apprivoisé, il va s'agir de réussir la génération d'un document PDF. Puis nous y incluerons les éléments importants du rapport technique tels que la bibliographie, table des matières, les références, et la page de titre.

2.1 Le B.a.-ba

À partir d'un document Markdown, ici nommé `rapport.md`, nous pouvons débiter.

```
$ pandoc rapport.md --to latex
```

Le contenu est transformé selon le formalisme \LaTeX . Et seulement, le contenu. Le drapeau `--standalone` permet d'en faire un document complet.

```
$ pandoc rapport.md \
  --standalone \
  --to latex
```

La commande suivante le transformera, avec un peu de chance, en document PDF. Le moteur de rendu est à spécifier, ici \XeLaTeX .

```
$ pandoc rapport.md \
  --standalone \
  --to latex \
  --pdf-engine xelatex \
  --output rapport.pdf
```

Le détail de chaque drapeau :

- `--standalone` ajoute les entête et pied au document. Sinon, seul le contenu est produit ;
- `--to latex` décrit que le format de sortie désiré est \LaTeX ;
- `--pdf-engine xelatex` pour un meilleur support des polices (OTF, TTF, etc.) et de l'Unicode ;
- `--output rapport.pdf` le nom du fichier de sortie.

2.1.1 Page de titre

Markdown ne peut s'occuper que du contenu. Donc il est nécessaire de passer par le prologue *YAML* du fichier.

```
---
title: My awesome project
lang: en
author:
- Yoan Blanc <yoan@dosimple.ch>
- Marc Schaefer <marc.schaefer@he-arc.ch>
date: \today
pagesize: A4
---
```

2.1.2 Table des matières

Le format \LaTeX contient des macros permettant de produire une superbe table des matières. Il suffit de l'activer dans le prologue.

```
---
toc: true
toc-depth: 5
---
```

2.1.3 Mise en forme

Avec quelques options de base, il est déjà possible d'obtenir un résultat intéressant. Pour aller plus loin, s'intéresser au fonctionnement des templates \LaTeX devient indispensable.

```
---
fontsize: 14pt
documentclass: scrreprt
documentoptions: twoside
numbersections: true
---
```

Les styles de base tels que `article`, `report` et `book` ont leur pendant dans la bibliothèque `Koma-Script`¹ avec `scrartcl`, `scrreprt` et `scrbook`. `Koma` est assez populaire car son rendu est de bonne facture.

¹<https://www.ctan.org/pkg/koma-script>

2.1.4 Polices de caractères et couleurs

Les polices de caractères Microsoft (ttf-mscorefonts-installer) sont installées. Il est possible de s'en servir et de pouvoir ressembler à un document produit avec Microsoft Word.

```
---
colorlinks: blue
mainfont: Times New Roman
sansfont: Arial Black
monofont: Andale Mono
monofontoptions: Scale=0.8 # Pour ceux qui écrivent en Java
                        # c'est peut-être encore trop grand.
---
```

Sous GNU/Linux, les polices sont généralement installées dans `/usr/share/fonts`. Et `apt` permet d'en installer tout un tas d'autres de bonne facture via `texlive-fonts-extra`. La famille Linux Libertine² est utilisée dans ce document.

2.2 Bibliographie avec *pandoc-citeproc*

Le format \LaTeX possède un outil, ou de nombreux outils, permettant de gérer les bibliographies. *pandoc-citeproc* en supporte plusieurs dont Bib \LaTeX , le format généralement utilisé avec \LaTeX .

```
---
bibliography: bibliographie.bib
---
```

Création de la référence [arc2017].

NB : Pour qu'un élément apparaisse dans la bibliographie, il est nécessaire qu'il soit cité dans le corps du document.

```
$ pandoc rapport.md \
  --filter pandoc-citeproc \
  ...
```

Un moyen simple de créer son fichier `.bib` (par exemple lst. 2.1) est d'utiliser un logiciel permettant de collecter les références bibliographiques tel que *Zotero* ou de les exporter via *Google Scholar*³.

²<http://www.linuxlibertine.org/>

³<https://scholar.google.fr/intl/fr/scholar/help.html#export>

Listing 2.1 bibliographie.bib

```
@article{arc2017,  
  title={Communication I : le rapport technique},  
  author={Perret, Michael and Raboud, Stéphane and Babey, Magali},  
  year={2017}  
}
```

2.2.1 Style de la bibliographie

Pandoc utilise le style dit *Chicago* (Webster 2017) car très populaire dans les sciences humaines. Il ne l'est pas forcément au sein des sciences de l'ingénierie.

Les styles de bibliographie sont nombreux et donc interchangeables d'une plateforme de publication à l'autre. *CitationStyles*⁴ permet de décrire le format de la bibliographie et des citations. Vous trouverez des styles à choix sur le *Zotero Style Repository*⁵. Le format francophone et le style *ISO 690* sont recommandés par l'école.

```
---  
bibliography: README.bib  
csl: iso690-author-date-fr.csl  
---
```

2.3 Références avec *pandoc-crossref*

Dans un document scientifique, il est recommandé de numéroté et référencer toutes les figures, et tables de données. Une tâche qui est un peu fastidieuse. Elle devient presque aisée à l'aide d'un outil comme *pandoc-crossref*⁶.

```
$ wget https://github.com/lierdakil/pandoc-crossref/releases\  
> /download/v0.3.0-beta3/linux-ghc8-pandoc-2-0.tar.gz
```

```
$ tar xf linux-ghc8-pandoc-2-0.tar.gz  
$ sudo mv pandoc-crossref /usr/local/bin
```

Il est important d'exécuter le filtre des références avant celui des citations.

```
$ pandoc rapport.md \  
  --filter pandoc-crossref \  
  --filter pandoc-citeproc \  
  ...
```

⁴<https://www.citationstyles.org>

⁵<https://www.zotero.org/styles/>

⁶<https://github.com/lierdakil/pandoc-crossref>



FIG. 2.1 : Le trombone de Microsoft Word “Clippy” tout triste

Ensuite, référencer une image telle que le *Clippy* triste (fig. 2.1, © 2016 Joana Afonso⁷) se fait ainsi :

Un paragraphe décrivant l’illustration (@fig:id).

```
![[légende](fichier.jpg){#fig:id}
```

En \LaTeX , afficher la table des figures se fait via `\listoffigures`, qui existe au sein du prologue avec `lof` (et `lot` pour `\listoftables`).

```
---  
lof: true  # table des figures  
lot: true  # liste des tableaux  
---
```

ou, en \LaTeX directement pour contrôler leur place et/ou leur ordre dans le document.

```
\listoffigures  
\listoftables  
\listoflistings    # ‘lol’ n’existe pas dans pandoc.
```

Les options sont expliquées dans la page manuelle de la commande elle-même.

```
$ pandoc-crossref --man
```

2.4 Étendre *pandoc*

Il est possible de modifier le format produit par *pandoc* à l’aide de filtres en Haskell (le langage de *Pandoc*), Lua (*Pandoc 2*) ou Python (*pandoc-filters*).

⁷<http://joanaafonso.com>

2.4.1 Exemple : les problèmes liés au français

La bibliothèque gérant les langues va nous fournir des corrections adaptées à la langue choisie. Comme nous écrivons en français, elle va, entre autres, ajouter un espace insécable devant certains éléments de ponctuation tels que le deux-points (:), le point-virgule (;). Et elle s'occupe également de modifier l'apostrophe, et les “guillemets”. Choses qui ne sont pas désirées lorsqu'il s'agit de code source.

Le filtre (lst. 2.2) écrit en Lua change la langue utilisée dans les blocs de code. S'en servir est aussi simple qu'avec les autres filtres.

```
$ pandoc rapport.md \  
    --lua-filter english.lua \  
    --filter pandoc-crossref \  
    --filter pandoc-citeproc \  
    ...
```

Sans ceci, tous les exemples contenant ces éléments de ponctuation seraient cassés. C'est un usage très avancé, que les plus informaticiens d'entre vous sauront dompter.

Listing 2.2 english.lua

```
1 local lang = 'english'
2
3 local before = pandoc.RawInline(
4   'tex', '\\begin{otherlanguage}{ ' .. lang .. ' }')
5 local after = pandoc.RawInline(
6   'tex', '\\end{otherlanguage}')
7
8 return {
9   {
10    Code = function(code)
11      return {before, code, after}
12    end,
13
14    CodeBlock = function(code_block)
15      return {
16        pandoc.Para({before}),
17        code_block,
18        pandoc.Para({after})
19      }
20    end,
21
22    RawBlock = function(raw_block)
23      return {
24        pandoc.Para({before}),
25        raw_block,
26        pandoc.Para({after})
27      }
28    end,
29  }
30 }
```

3 Conclusion

Réaliser le rapport technique d'un projet de semestre, de diplôme, seul ou en groupe est un gros investissement en énergie. C'est parfois l'élément du projet qui est sous-estimé par les étudiants. *Markdown* est un format suffisamment simple pour permettre la prise de note tout au long du projet. Il est possible d'en faire l'édition en ligne aisément voire même de collaborer en même temps sur un document.

Pandoc est ensuite la baguette magique transformant tout ceci en un document de bonne facture. Avec l'avantage d'être progressif. Un document \LaTeX demande un investissement relativement important d'apprentissage, mais permet beaucoup plus d'options. Et à l'inverse Microsoft Word est trop laxiste, n'imposant pas une structure forte dès le départ rendant les modifications futures complexes.

Le but de ce document est de servir de base pour correspondre aux attentes de la Haute-École Arc. Il est améliorable sous de nombreux aspects et reste suffisamment ouvert afin de permettre un peu de créativité.

La source de ce document se trouve sur <https://github.com/HE-Arc/rapport-technique>. Faites-en bon usage !

Annexes

Glossaire

IMMRID Introduction, Matériel, Méthode, Résultats, Interprétation et Discussion. Méthode de structuration adaptée à un projet de recherche.

L^AT_EX Un langage de description destiné à la production de documents techniques et scientifiques. <https://www.latex-project.org/>

Lua Un langage de programmation léger et généralement embarqué pour étendre une application. <https://www.lua.org/>

Python Un langage de programmation impératif facile, ludique et populaire. <https://python.org>

SPRI Situation Problème Résolution et Informations. Une méthode permettant structurer un plan de présentation et introduite par Louis Timbal-Duclaux.

WSL *Windows Subsystem for Linux* une plateforme permettant de faire fonctionner des logiciels GNU/Linux sur le système Windows sans passer par une machine virtuelle. <https://msdn.microsoft.com/en-us/commandline/wsl/about>

YAML *YAML Ain't a Markup Language* un format de description structuré n'étant pas XML. <https://yaml.org>

Sources

PERRET, Michael, RABOUD, Stéphane et BABEY, Magali, 2017. Communication I : le rapport technique.. 2017.

TIMBAL-DUCLAUX, L, 1982. Communication. Comment exposer efficacement des idées complexes : la méthode SPRI. *Revue Générale Nucléaire*. 1982. N° 5, pp. 459-462.

WEBSTER, Gretchen, 2017. *The Chicago Manual of Style*. 2017. Springer.