Academic Markdown

A Guide to Writing Academic Works from Markdown

Zohar Cochavi

Contents

Preface **1**

Introduction **3**

Conventions - Using Markdown **5**
A Brief History 5
Our Convention 5
Figures and Internal References 5
External References or Citations 6

Conclusion **7**

References **9**

Appendix **11**
Syntax Overview 11
Document Sections 11
Text formating 12
Inter- and Extra-document Links 13

Preface

A long time ago, there was a boy, who, like all others in the land of academics whished to create nicely formatted, accessible documents. As most people in the land, he was forced to speak in LaTeX. And although he was first confused, he eventually learned to love it.

The joy of simply reading through a document that looked professional, sleek, dare one say *like a work of art*. "Who cares about content?", the boy thought. And so he spent his days formatting, formatting, and formatting. And oh how he formatted.

> *To the sound of 'my favorite things'*
>
> From tables, to templates, and tensors, to tokens.
> And sections and headings, and figures, he opens.
> Each one of these looks as best as can be.
> Those are some of his favorite things.

Until one day, his deadline was due, he got into trouble. You see, as was *so* meticulously formatting, and formatting, and formatting, he forgot to write! Shock and terror overtook the boy, as it slowly dawned on him: "Who cares about formatting?!".

That's right, who cares about formatting? Well, I won't lie, I do. But as I have learned, it should not distract from the content. You might be surprised to hear that the boy from the parable is me, and you might be even more surprised to hear that no-one should care about formatting. "What?" Yes, wait, please bear with me.

Let me rephrase: you should not care about formatting. As a writer, I have to care about formatting, but it should never be your focus. Formatting exists to support the message, but for the formatting to support your message, you should have a good message in the first place. The reader, you, should not care about formatting. The message should be the focus.

As writing the message should be the focus, why would we use tools riddled with markup language when writing? Honestly, I don't know. There are many

different possibilities with regard to writing workflows, and I don't claim to know them all. But I do have the feeling that not many of them involve raw text in a distraction-free environment.

That is not to say I don't think they're useful. Languages such as the TeX[1] provide incredibly detailed typesetting configurations. Microsoft Word, regardless of its many shortcomings, provides a more intuitive way for people to format their documents. The former has allows is more malleable, while the latter is generally easier to use. But these are not necessary for *writing*, rather they are necessary for typesetting.

My personal belief, however, is that your message should stand as independent from your formatting as possible. Notwithstanding the importance of accessibility, it should not matter whether your 'document' is a collection of sheep with glyphs shaved in their wool, or a carefully type-set LaTeX document. Although, the latter will probably smell significantly less.

I therefore want to urge anyone who is reading the following: Start writing in plaintext. Literal *plain text*. Not because it is easy to distribute, or because it does not require any tools besides a text editor, or even because of it is small size. Write in plain text because you will not be able to fool yourself with anything but your own writing. Tools are a windows into the world; when you are holding a hammer, everything suddenly looks 'hammerable'.

This starts to sound more like a manifesto and less like a manual or user-guide, so I will slack on the opinions. I nevertheless hope I have at least motivated my choice, and I hope to have at least made you think about your writing and its efficacy. Which brings me, and you for that matter, to the part where I will explain how *I* practice what I preach, and how you can copy said practice.

[1] It's actually a fully Turing-complete language. LaTeX is only a set of macros (i.e. functions) for the TeX language.

Introduction

Brief about why you should be interested in this document

For this document, the most important thing to know is that I am not an expert by any means. This is simply *my* view on how Markdown should be used and a proposal for the infrastructure, or method, for writing. That is, the workflow.

My workflow, or the conjunction of *convention* and *infrastructure*, works for me, but I have tried to make it as accessible as possible while taking into account different people and environments. I hope this widens the audience and reduces the barrier for entry. Thereby showing as many people as possible how cool and useful Markdown is.

This guide is split according to the workflow as defined above, meaning into *convention* and *infrastructure*. First, we will discuss how I think Markdown should be used and why. Then, we will discuss the different tools to use Markdown, enforce those standards, and export Markdown to the different necessary formats.

The first section should be platform-agnostic, while the latter will be split into a platform-agnostic and a *VSCode*-specific part. This is to accommodate both the people that are particular about their environments and are only looking for supplementary tools, as well as the people that are looking for a complete writing system.

To be clear, I am a programmer by trade and while I have an interest in accessibility and communication, which is hopefully reflected in the user experience of this workflow, this workflow is still made for me, i.e. a programmer. I think the only 'nice-to-have' is knowledge of a language like LaTeX. This is only because it reflects a certain experience, not because it is necessary for this workflow in any way.

Nevertheless, I think anyone might find parts this text interesting. Whether it is only to learn about niche Markdown features or join me in exploring the vast depth that is `pandoc`.

Conventions - Using Markdown

The goal of writing in Markdown is to use a minimal set of markup options, thereby maintaining focus on writing instead of formatting. Of course, this limits the number of options when wanting to format your document, but the first step should really be to write.

The problem is that quite a few Markdown definitions exist. Originally, it was meant for converting to HTML, but many have adopted it to work for a range of formats to export to.

A Brief History

Our Convention

Considering the context in which the original markdown spec was created, we have some special requirements that should be accommodated with extended syntax. In an academic context, large complex documents are the norm and special attention should therefore be given to the use of (i) figures and internal references, and (ii) external references.

Figures and Internal References

Firstly, figures and internal references provide structure and clarity within a document. There should not be any contention concerning the figure, table, or resource that is referenced. When writing in LaTeX, for example, a figure is simply denoted by `\begin{figure}` and the corresponding caption number is automatically generated, and links are created by internal `\label` and `\ref` commands. This avoids the need for constant updating of figure numbers, markup we would not want to bother ourselves with when writing.

As the original Markdown spec required a minimal markup syntax, this was omitted. Such functionality could still be accessed through the use of anchor

`<a>` tags. But since we would like complete interoperability between LaTeX and Markdown, we need something more native.

Thus, to accommodate the use of figures, and especially figure captions and references, we use `pandoc-crossref`. This `pandoc` filter[2] extends the Markdown syntax to include figure markup and automatically generates link identifiers for headers/sections too (see fig. 1 and lst. 0.1). This one of the great things about `pandoc`, as it is very configurable it can be molded to your specific needs. But this of course has a downside, namely that it can be quite complex.

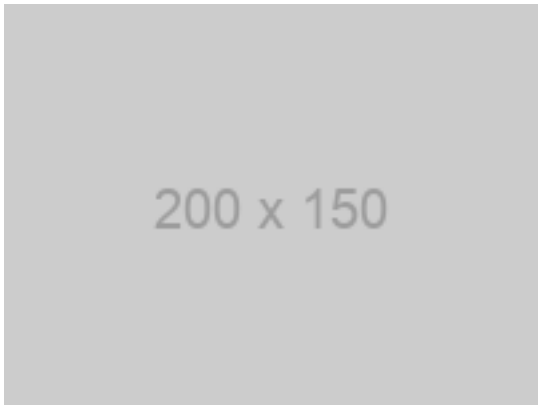

Figure 1: An example image

Listing 0.1 The corresponding code for an example image

```
![An example image](https://via.placeholder.com/200x150){#fig:example_image}
```

Since we are working in `pandoc` and it allows for some extensions which are not native to Markdown, it is important to note that we work in *pandoc-flavored* Markdown. Including this style of Markdown in for example Github READMEs, might give mixed results depending on the features used. This is not necessarily a problem as you will simply realize that GitHub does not support a particular feature (figures are one such example), but it is important to realize Markdown is not fully standardized and be aware of such issues. Therefore, we export from our pandoc-flavored Markdown to *GitHub-flavored* Markdown with `pandoc`[3].

External References or Citations

[2] A `pandoc` filter is a program that is run in series with the parser (in this case the Markdown parser). In this case, the filter is used to extend the Markdown syntax, but it can be used for various other purposes as well.

[3] In `pandoc`, GitHub-flavored Markdown is denoted as `gfm`. When exporting to Markdown with `academic-markdown.py`, this is the default setting.

Conclusion

The conclusion to a very serious and important article.

References

Appendix

Syntax Overview

This section is supposed to serve as a reference to the *recommended* syntax. In many cases, variations on the syntax exist. I have often chosen not to include these to ensure clarity and consistency. Especially in the case of [Figures], the syntax I show is chosen because it operates more easily with different tools and formats.

Document Sections

In these examples a multi-document setup is denoted as *MD*, and a single-document setup as *SD*.

Title (MD)

Listing 0.2 In a single document, including this in the frontmatter is *not* the preferred way of denoting the title, simply for compatibility with other markdown flavours. In a multi-document setup, this should be denoted in the `metadata.yaml` file.

```
title: Title or Section
```

Title (SD) or Section (MD)

Listing 0.3 In single documents, this will be the title. In multi-document formats, this will denote a level-1 heading or a section.

```
# Title or Section
```

Section (SD) or Subsection (MD)

Subsection (SD) or Subsubsection (MD)

```
### Subsection (SD) or Subsubsection (MD)
```

Listing 0.4 Section in a single-document setup, or subsection in a multi-document one.

```
## Section (SD) of Subsection (MD)
```

Subsubsection (SD) or Subsubsubsection (MD)

Listing 0.5 Subsubsubsubsubsubsubsubsub...

```
#### Subsubsection (SD) or Subsubsubsection (MD)
```

Text formating

Common text is just written as normal text. There are some tricks included in that might not immediately obvious. To force a newline you can end the current line with two spaces.
Line this. A separate paragraph, however, is made by a blank line between two blocks of text, i.e. two newlines.

The next code blocks show how you can format text in different ways. Like many other things in markdown, there are multiple syntaxes. These are, however, the ones that are most commonly used, or simply the type that I consider 'better' either clarity.

`Common text`

Emphasized text

`_Emphasized text_ or *Emphasized text*`

~~Strikethrough text~~

`~~Strikethrough text~~`

Strong text

`__Strong text__ or **Strong text**`

Strong emphasized text

Listing 0.6 An alternative is to used the following `***Strong emphasized text***`, but this is hard to distinguish from normal `**Strong text**` (similarly for emphasized text).

```
___Strong emphasized text___
or
***Strong emphasized text***
or
_**Strong emphasized text**_
```

- Bullet list
  - Nested bullet

        - Sub-nested bullet etc
- Bullet list item 2

```
- Bullet list
    - Nested bullet
        - Sub-nested bullet etc
- Bullet list item 2
```

-OR-

```
* Bullet list
    * Nested bullet
        * Sub-nested bullet etc
* Bullet list item 2
```

1. A numbered list
    1. A nested numbered list
    2. Which is numbered
2. Which is numbered

```
1. A numbered list
    1. A nested numbered list
    2. Which is numbered
2. Which is numbered
```

- ☐ An uncompleted task
- ☒ A completed task

```
- [ ] An uncompleted task
- [x] A completed task
```

- ☐ An uncompleted task
    - ☐ A subtask

```
- [ ] An uncompleted task
    - [ ] A subtask
```

> Blockquote
>> Nested Blockquote

```
> Blockquote
    >> Nested Blockquote
```

Inter- and Extra-document Links

Website Links

Named Link or http://example.com/

```
[Named Link](https://www.example.com/) or <http://example.com/>
```

Link to Title (MD)

```
Link to [Title (MD)]
```

Table, like this one :

First Header	Second Header
Content Cell	Content Cell
Content Cell	Content Cell

```
First Header | Second Header
-------------|--------------
Content Cell | Content Cell
Content Cell | Content Cell
```

Adding a pipe | in a cell :

First Header	Second Header
Content Cell	Content Cell
Content Cell	|

```
First Header | Second Header
-------------|--------------
Content Cell | Content Cell
Content Cell | \|
```

Left, right and center aligned table

Left aligned Header	Right aligned Header	Center aligned Header
Content Cell	Content Cell	Content Cell
Content Cell	Content Cell	Content Cell

```
Left aligned Header | Right aligned Header | Center aligned Header
:-------------------|---------------------:|:--------------------:
Content Cell        |         Content Cell |     Content Cell
Content Cell        |         Content Cell |     Content Cell
```

```
code()
```

```
`code()`
```

Horizontal line : - - - -

```
- - - -
```

Image with alt :

```
![picture alt](http://via.placeholder.com/200x150 "Title is optional")
```

Inline LaTeX

Listing 0.7 An alternative to the backticks is to use three sguiggly lines `~~~`. This is useful if you want to show a markdown code block in a code block. Not really useful unless you want tot present markdown notation.

```
var specificLanguage_code =
{
    "data": {
        "lookedUpPlatform": 1,
        "query": "Kasabian+Test+Transmission",
        "lookedUpItem": {
            "name": "Test Transmission",
            "artist": "Kasabian",
            "album": "Kasabian",
            "picture": null,
            "link": "http://open.spotify.com/track/5jhJur5n4fasblLSCOcrTp"
        }
    }
}
```

```
$\LaTeX$
```

Block equation:

$$E = mc^2$$

```
$$
E = mc^2
$$
```

Footnotes:

Something to read for later[4]

```
Something to read for later[^1]

[^1]: [A Brief History of Time](https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvb
```

[4] A Brief History of Time

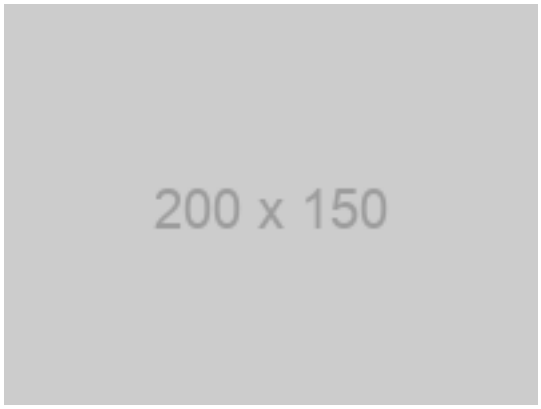

Figure 2: picture alt