

Markdown

Tools for writing science

Andrés Aravena, PhD

October 13, 2022

Ancient art of magic

In my view, the “old ways” that are powerful magic include:

- Command line
- Text files
- Editing text in a terminal
- Combining all of them

Text files are for humans and computers

- Binary files are hard to read
 - unless you have the correct program
- Text files can be read by humans
 - Each byte is a letter
- Text files can be read by computers
 - Data must be **recyclable**
 - The output of one program may be the input of another program

Text editors instead of Word processors

The easiest way to handle *text files* is to use a **text editor**

These are programs to view and edit text files

They use a monospaced font, like Courier

Each letter has the same width

Text editor have syntax coloring

Since each letter has the same size, text editor use color

The color depends on the **role** of each text

For example, *headings* can be in red color

The color is not in the file. The editor puts colors

Text editors handling Markdown

These work with Markdown and other formats

- Visual Studio **Code**: code.visualstudio.com
- Atom: atom.io
- RStudio:
rstudio.com/products/rstudio/download/

All are good. We use *VSCode*

Markdown Text editors

- Typora: typora.io
- MarkdownPad: markdownpad.com
- Markdown Monster:
markdownmonster.west-wind.com

Online Markdown editors

- StackEdit: `stackedit.io`

- Dillinger: `dillinger.io`

- Draft: `draftin.com`

- Markdown Editor:

`jbt.github.io/markdown-editor/`

Text files are for ever

Free

- nothing to pay
- you can do whatever you want

Never get obsolete

**But they do not have
structure**

Structured Documents

We want to identify the *meaning*, not the *shapes*

- Title
- Sections
 - Subsections
 - Lists
 - Figures
 - Tables
- References to other works

Separation of concerns

The key idea is to describe *what things are*, not *how they look*

Describe the role of text, not the “looks”

Separate style from structure

Text files with structure

There are several markup languages that encode the structure of a *text* document

- LaTeX
- ReStructured Text
- MediaWiki
- HTML
- Markdown
- Textile
- AsciiDoc

Markdown

Markdown is a widely used markup language

- Same philosophy as LaTeX, but simpler
- The text file can be read and understood easily
- It can be transformed into other formats
 - PDF, Word, Webpage (HTML)
- Used in R, Python, Julia (Jupyter), in GitHub, and many other modern platforms

Markdown's author says:

“The overriding design goal for Markdown’s formatting syntax is to make it as readable as possible.

“The idea is that 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 <https://daringfireball.net/projects/markdown/>

Flavors of Markdown

Compiling is transforming from Markdown to other format

There are many different Markdown compilers

Many people make their own compiler, and they expand the original idea

Unfortunately, they are not always 100% compatible

There is not yet an official standard

Recommendation: *pandoc*

(if you have RStudio, you have *Pandoc*)

Pandoc

If you need to convert files from one markup format into another, pandoc is your swiss-army knife

John MacFarlane, developer of Pandoc

Pandoc can convert between many formats, including

- Markdown
- Microsoft Word/Powerpoint
- LaTeX
- Jupyter notebook

John MacFarlane

Professor of Philosophy, University of California,
Berkeley

Author of books

- Philosophical Logic: A Contemporary Introduction
- Assessment Sensitivity: Relative Truth and Its Applications

Recent papers:

- “Lecture I: Vagueness and Communication”
- “Lecture II: Seeing Through the Clouds”
- “Lecture III: Indeterminacy as Indecision”
- “On Probabilistic Knowledge”



Pandoc advantages

- Text files
- It is easy to write tables in Markdown
- It is easy to write lists
- Can be used for slides
 - Several web platforms (like this document)
 - Microsoft Powerpoint
- Handles BiBTeX references

Using Pandoc

It is a command line command, and can be used inside VSCode

There is even a plugin

In the command line we write

```
pandoc document.md --output document.pdf
```

and there are many options. See <https://pandoc.org>

Paragraphs

- Consecutive lines of text are one paragraph.
- They are separated by an empty line

The first paragraph.

Another paragraph

The first paragraph.

Another paragraph

Headers

```
# Header 1  
## Header 2  
### Header 3  
#### Header 4
```

Header 1

Header 2

Header 3

Header 4

Unordered Lists

```
+ Item 1  
+ Item 2  
    + Item 2a  
    + Item 2b
```

- *Item 1*
- *Item 2*
 - *Item 2a*
 - *Item 2b*

Sub-lists are indented by 4 spaces

Ordered Lists

```
1. Item 1  
1. Item 2  
1. Item 3  
    1. Item 3a  
    1. Item 3b
```

1. Item 1
2. Item 2
3. Item 3
 1. Item 3a
 2. Item 3b

Images

You have to indicate the web address of the image

```
![optional text](http://example.com/logo.png)
```

or the name of a file in the same directory

```
![optional text](images/logo.png)
```



Optional text is shown when the image is not found

```
![optional text](images/logo.pn)
```



optional text

Figures with Captions

This is a *pandoc* extension, not standar Markdown

If the figure is a paragraph (has empty lines before and after_ then the_optional text_ becomes the caption,

```
![This is the caption of the figure.](images/logo.png)
```



This is the caption of the figure.

Tables

There are several formats. The easiest one is this

	sample	dose	time	agent
1	GSM91440	low	5 min	caffeine
2	GSM91893	low	5 min	caffeine
3	GSM91428	low	5 min	calcofluor white
4	GSM91881	low	5 min	calcofluor white

	sample	dose	time	agent
1	GSM91440	low	5 min	caffeine
2	GSM91893	low	5 min	caffeine
3	GSM91428	low	5 min	calcofluor white
4	GSM91881	low	5 min	calcofluor white

Tables with captions (*pandoc* extension)

Write `Table:` and the caption just after the table

	sample	dose	time	agent
1	GSM91440	low	5 min	caffeine
2	GSM91893	low	5 min	caffeine
3	GSM91428	low	5 min	calcofluor white
4	GSM91881	low	5 min	calcofluor white

Table: This is the table caption

This is the table caption

	sample	dose	time	agent
1	GSM91440	low	5 min	caffeine
2	GSM91893	low	5 min	caffeine
3	GSM91428	low	5 min	calcofluor white
4	GSM91881	low	5 min	calcofluor white

Making tables

There are some VSCode plug-ins that can make tables for you

Or you can make them in R using `knitr` or `pander` libraries

A good alternative is this website:

https://www.tablesgenerator.com/markdown_tables

Computer code

Programs are usually written in a *monospaced* font. That is, all letters have the same width.

```
```\nthis <- is.computer(code) {\n  # comment\n}\n```
```

```
this <- is.computer(code) {\n # comment\n}
```

# Nicer computer code

You can indicate the language, and get colors

```
```r  
this <- is.computer(code) {  
  # comment  
}  
```
```

```
this <- is.computer(code) {
 # comment
}
```



# Links

To write a reference for a web page

```
This is a [clickable text](https://dry-lab.org/).
```

*This is a [clickable text](https://dry-lab.org/).*

# Nicer Links

It can also be written in [a cleaner format].

[a cleaner format]: <http://dry-lab.org>

*It can also be written in [a cleaner format](http://dry-lab.org).*

The link address can given in a separate line

# Footnotes

```
Here is a footnote reference,[^1] and another.[^longnote]
```

```
[^1]: Here is the footnote.
```

```
[^longnote]: Here's one with multiple blocks.
```

```
 Subsequent paragraphs are indented to show that they
 belong to the previous footnote.
```

```
This paragraph won't be part of the note, because it
isn't indented.
```

Here is a footnote reference,<sup>1</sup> and another.<sup>2</sup>

This paragraph won't be part of the note, because it isn't indented.

# Inline code

We can compare ``x`` and ``data``

*We can compare `x` and `data`*



# Emphasis

Use it **only** when strictly necessary

```
Inside the paragraph we can have *italics*
and **bold** text
```

Inside the paragraph we can have *italics* and **bold** text

# Comments in Pandoc

Pandoc can understand some HTML

If we write an HTML comment, it will not show in the output

```
<!-- this part does not show -->
```

(Must use pandoc option `--strip-comments`)

# Online resources

For your weekend

- <https://www.thecloudtutorial.com/using-markdown-for-documentation/>
- <https://www.markdownguide.org/getting-started>
- <https://pandoc.org/try/>

---

1. Here is the footnote.↩

2. Here's one with multiple blocks.

Subsequent paragraphs are indented to show that they belong to the previous footnote.↩

