

A poor man guide to write and publish using CommonMark

Daniele Salvini

March 16, 2016

Contents

Introduction	3
Marking the text	3
An example of marked writing	3
Markdown and CommonMark	3
Using a text editor	3
Syntax	4
Chapters	4
Paragraphs	4
Text attributions	4
Italic	4
Bold	5
Strikethrough	5
Links	5
Automatic link	5
Inline link	5
Reference link	5
Internal link	6
Image link	6
Using links	6

Representing code	7
Blockquotes	7
Lists	7
Footnotes	8
Publishing	8
Layout	8
Bibliographic information	8
Page break	8
Conversion	8
Citations	9
Insert a citation	9
Create a LaTeX of PDF document with citations	9
Using an advanced text editor	11
Emacs	11
Using Emacs to write Markdown	11
Vim	12
Using Vim to write Markdown	12
Outline navigation	13
Summary	14
Bibliography	14

Introduction

Marking the text

A markup is a conventional sign to indicate formatting.

Lightweight markup languages is composed by codified expressions that are simple enough to be used during creative writing.

In other words a language with simple rules for describing the representation of text where from is possible to export structured documents for various devices.

An example of marked writing

```
# First Chapter
  blah blah blah

# Second Chapter
  blah blah blah
```

Is important to note that reading the raw text is easy.

Markdown and CommonMark

Markdown is the markup language created by Aaron Swartz e John Gruber. Created to publish HTML, it has some limitations when it comes to other formats. CommonMark is a dialect aiming to overcome those limitations and to standardize the conventions. We call Markdown the strict format invented in 2004 and CommonMark his extended form with the aid of the documents converter Pandoc. A text written in CommonMark can be easily converted in a publishing format as HTML, PDF, EPUB or LaTeX.

This document has been written in extended Markdown and converted with Pandoc. It may be useful to read and compare the [raw CommonMark format](#) with the [converted HTML page](#) or the [PDF](#).

Using a text editor

To write and freely insert marks we may want to use a text editor, good for writing, saving, rewriting and print and who ignore formatting¹, for this is a task for the writer, using marks.

¹Documents written in a markup language like HTML are usually viewed with the formatting rendered by a browser, but still they are plain text documents.

Every text editor will do nicely, providing that writes and saves in plain text format (TXT). For instance [Gedit](#) is also capable to assist in the logic structure of the text, highlighting the marks.

When we assign to the document a desinence that declares the format, the text editor can recognize it automagically: .md stands for Markdown.

For instance: this file is named: commonmark-good.md

Syntax

Chapters

A chapter (header) get represented by an hash at beginning of line.

We can go deep until six headers using crunches : #, ##, ###, and so on².

Example:

```
# Title
## Chapter
### Sub chapter
```

Paragraphs

To represents a line break, we use two spaces at the end of the line.

```
The question is: What's a ManaMana?
The question is: Who's care?
Statler & Waldorf, A view from the Balcony.
```

Paragraphs are separated by a blank line.

Text attributions

To create attributions we often use the asterisk.

Italic

A word between asterisks to be *italic*

```
*italic*
```

²This is not the only way to create headers, we can use the style we like, but is better do not mix them.

Bold

A word (or a sentence) between two asterisks to be **bold**

****bold****

Strikethrough

A word between tilde to be ~~strikethrough~~³

~~~~strikethrough~~~~

## **Links**

Let's consider four types of links: automatic, inline, reference and page internal.

### **Automatic link**

A web address (URL) between angle brackets get automatically represented.

<URL>

<https://stackoverflow.com/editing-help>

### **Inline link**

The syntax of an inline link is: name between square brackets and URL between round brackets.

[name] (url)

Example: [markdown](#) [announce](#) [page](#)

### **Reference link**

A reference link takes two parts: one refers to the other.

---

<sup>3</sup>We can also have other attributes: superscript, tag and spoiler, tables and math functions that will not be described here.

**Explicit reference link** An explicit reference link uses only square brackets and refers to a name that get expressed later on, for instance at the end of the page.

```
[reference] [name]
```

and

```
[name] :url
```

Example: [12121](#) Markdown page

**Implicit reference link** An implicit reference link leave out the name.

```
[reference] []
```

and

```
[reference] :url
```

### Internal link

To represent a link to a document's section, we can point it out to a chapter.

go to `[Syntax](#syntax)` section

go to [Syntax](#) section

### Image link

Insert an image uses the same syntax of a link, leaded by a bang (exclamation point).

```
![name](image.png)
```

### Using links

We can put a link description between quotes following the URL.

```
[name](url "description")
```

An inline link is faster to insert during writing, but makes harder reading the raw text, where a reference link reads better.

Here an example of an inline link with description to the [word Markdown](#) on the free encyclopedia. While is good for web publishing, reading the raw text is not easy.

Viceversa that implicit reference link with descriptions to [the word Markdown](#) on the free encyclopedia reads well. The reference is at the bottom of the paragraph.

## Representing code

A block of text indented with four spaces (or a Tab) get represented as code.

```
a block of text
represented as code
```

We can also put a word or a sentence between backticks.

```
'code'
```

Example: `that sentence get represented as code`

## Blockquotes

To represent a block of text separated from the rest we can use blockquotes email-style, starting the line with an angle bracket >.

Example:

```
> a block of text separated from the rest
```

to be

```
    a block of text separated from the rest
```

## Lists

To represent an unsorted list we use asterisks.

Example:

```
* milk
* coffee
* honey
```

to be

- milk
- coffee
- honey

To represent a numbered list we use numbers:

1. apple
2. orange
3. kiwi

It is not mandatory that the numeration is progressive, but is good advice.

## Footnotes

Footnotes takes two parts using a caret ^ between square brackets.

```
[^note]
```

e

```
[^note]: reference
```

Here's a note:<sup>4</sup>

## Publishing

### Layout

#### Bibliographic information

Title, author and data can be represented with that syntax at the beginning of the document.

```
% Title  
% Author  
% Data
```

#### Page break

To represent a page break<sup>5</sup> we can use:

```
\newpage
```

## Conversion

Once written, the document can be converted in a publishing format, here we have some examples<sup>6</sup> using the free and open source document converter Pandoc:

Install Pandoc e Citeproc:

```
apt-get install pandoc pandoc-citeproc
```

Convert from Markdown to HTML:

---

<sup>4</sup>The footnote can be placed everywhere and he will show up at the bottom of the page.

The footnote may continue in a block of text.

<sup>5</sup>the page interruption will not be rendered in HTML

<sup>6</sup>Using BunsenLabs, Linux distro based on Debian 8.



```
pandoc -s post.md -o post.html
```

Convert from Markdown to PDF:

```
pandoc paper.md -o paper.pdf
```

Convert from Markdown to EPUB:

```
pandoc -f markdown book.md -t epub -o book.epub
```

Convert from Markdown to LaTeX:

```
pandoc -f markdown paper.md -t latex -o paper.tex
```

## Citations

### Insert a citation

It is always a good idea to have inscribed somewhere, in large friendly letters, the words: DON'T PANIC (see Douglas 2005, 1:5).

The syntax to insert a citation from an external bibliographic file: `[@id]`

Example citation:

```
[see @dougadams, pp. 21-22]
```

Example double citation:

```
[consider @dougadams79, pp. 21-22 ; but also @tumembete85, pp. 42]
```

We can also use inline citations:

```
@manzoni wrote that
```

### Create a LaTeX of PDF document with citations

Once the text is complete with citations it can be converted from Markdown in the desired format using Pandoc and the `pandoc-citeproc` extension.

For instance a BibTeX file `biblio.bib` contains the following:

```
@book{dougadams:hg2g,  
  author = {Douglas, Adams},  
  title = {The Hitchhikers Guide to the Galaxy},  
  volume = {1},  
  publisher = {Pan Books},  
  address = {London},  
  year = {2005}  
}
```

Conversion from Markdown to LaTeX pointing the external bibliographic resource:

```
pandoc --bibliography=biblio.bib paper.md -o paper.tex
```

Conversion from Markdown to PDF generating an index:

```
pandoc --bibliography=biblio.bib --toc -s saggio.md -o saggio.pdf
```

Using the same command with the desinence .epub will produce the desired effect (to generate an EPUB).

The standalone option `-s` produces an independet document;

The `--toc` option generates a table of contents;

The `-N` option generates numbered chapters;

The `-H` option, for instance: `-H head.css`, may be used to link a css file in the HTML header.

When the `--cs1` option is not specified, Pandoc defaults to the Chicago Manual of Style.

Please read the [Pandoc](#) manual for further options. Pandoc uses UTF-8 character encoding.

Citations get listed at the end of the document, thus the last chapter be better named: Bibliography.

```
# Bibliography
```

# Using an advanced text editor

## Emacs

### Using Emacs to write Markdown

GNU Emacs is a free text editor that uses macros and different *modes* to offer different behaviors<sup>7</sup> seconding the style of the writing. Here we used GNU Emacs 24.4.1 and markdown-mode.

**Markdown mode** The *mode* that Emacs uses to write Markdown is named [markdown-mode.el](#) and ports few conventions from org-mode to Markdown writing.

Once downloaded [markdown-mode](#) in the Emacs path, usually `~/.emacs.d/lisp/`, it may be handy to have Emacs to open every file that end with `.md` directly in markdown-mode: adding those lines to our `~/.emacs`

```
; activate path
  (setq load-path (cons "~/.emacs.d/lisp/" load-path))

; activate markdown-mode for file.md
  (autoload 'markdown-mode "markdown-mode" "Major mode for editing Markdown files" t)
  (add-to-list 'auto-mode-alist '("\\.md\\\\" . markdown-mode))
```

**Integration of markdown-mode.el with Pandoc** Let's define a command to convert from Markdown in another format using Pandoc

```
(defun convert-markdown-to (newtype)
  (interactive "sOutput [html|html5|rtf|pdf|mediawiki|latex|..]: ")
  (let ((current-document (buffer-file-name))
        (temp-filename (concat "./output." newtype)))
    (with-temp-file temp-filename
      (call-process-shell-command (concat "pandoc -s -f markdown -t " newtype)
        nil t nil current-document))
```

Now the command `convert-markdown-to` will ask which format to convert to. For instance: HTML, HTML5, PDF, MediaWiki, EPUB, LaTeX. Converted file is named `output.[xxx]` and will be saved in the same directory.

We can also define a keybind to the command `convert-markdown-to`

```
(browse-url temp-filename)))
  (eval-after-load "markdown-mode"
    '(define-key markdown-mode-map (kbd "C-c C-c c") 'convert-markdown-to))
```

---

<sup>7</sup>Because there is a difference between coding in Python and hammering a screenplay

## Vim<sup>8</sup>

### Using Vim to write Markdown

Vim (or Vi IMproved) is a free advanced text editor commonly found on operative systems Unix-like, such as Linux, MacOSX e BSD. In that context we used Vim 7.4 or later.

Integration with Pandoc works perfectly with the [vim-pandoc](#) *plugin*.

**Installation of vim-pandoc** Vim-pandoc plugin installation can be either achieved using one of the following plugin-manager: Pathogen, Vundle or NeoBundle. We suggest using [Vundle](#) that requires adding the following lines in Vim configuration file, usually: `~/.vimrc`

```
Plugin 'vim-pandoc/vim-pandoc'  
Plugin 'vim-pandoc/vim-pandoc-syntax'
```

From within Vim, the following command will download and install the plugin:

```
:PluginInstall
```

**Using vim-pandoc** Vim uses that plugin to support Markdown writing in several ways, including the automatic creation of HTML pages, PDF and Tex documents as every other format supported by Pandoc.

To convert in HTML a document that is being edited, we can use the following command:

```
:Pandoc
```

Please read the plugin documentation for more details.

---

<sup>8</sup>Contribution by Andrea Marchesini

## Outline navigation

Using a text editor we can copy and paste sections of the text applying non-linear editing to our writing. Using an advanced text editor like Emacs or Vim we also acquire a feature called outline navigation. Using outline navigation we augment the non-linear editing capacity from micro to macro: previewing a map of the editing text and being able to use the copy-paste technique directly on chapters.

Outline navigation is a functionality found in the best structural text editors either folding or side panel style. Can induce dadaism as a collateral.

Example:

```
# Introduction
## Marking the text
A markup is a conventional sign to indicate formatting.
*Lightweight* markup languages is composed by codified expressions that are
simple enough to be used during creative writing.
In other words a language with simple rules for describing the representation
of text where from is possible to export structured documents for various devices.
## An example of marked writing
# First Chapter
  blah blah blah
```

The first image shows this text during writing

```
# Introduction...
## Marking the text...
## An example of marked writing...
## Markdown and CommonMark...
## Using a text editor...
# Syntax...
## Chapters...
## Paragraphs...
## Text attributions...
### Italic...
### Bold...
### Strikethrough...
## Links...
### Automatic link ...
### Inline link...
### Reference link...
#### Explicit reference link...
```

the second image shows the same text collapsed in outline folding navigation.

## Summary

```
# header      *italic*      **bold**      `code`
list:         * * * *      1. 2. 3. 4.
link:         [name](url)
link ref:     [ref] []      [ref]:url
note:        [^note] []    [^note]:what
image:       ![name](image.png)
citation:     [see @eco, pp. 42]
breakpage:   \newpage
```

Emacs markdown-mode keybinds:

Folding outline navigation: **Shift-Tab**

Go to previous/next chapter: **C-c C-p**, **C-c C-n**

Convert and preview HTML in browser via Pandoc: **C-c C-c c**

List available keys: **C-c C-h**

[Syntax Markdown page](#)

[Pandoc user guide](#)

[Pandoc citerproc homepage](#)

[Emacs tutorial](#)

[Vim tutorial](#)

[GNU/Linux Debian](#)

[BunsenLabs Linux](#)

## Bibliography

Douglas, Adams. 2005. *The Hitchhikers Guide to the Galaxy*. Vol. 1. London: Pan Books.