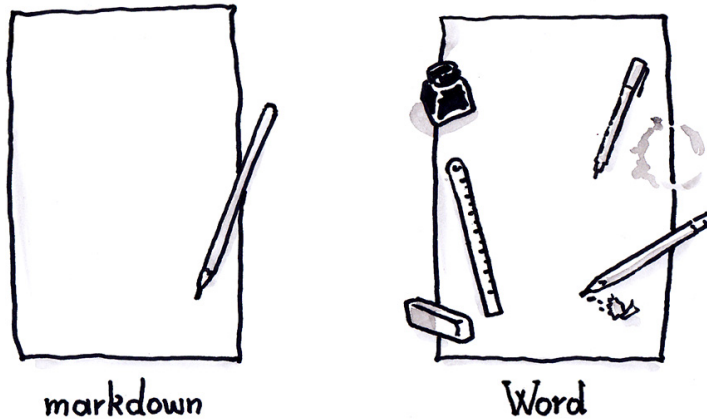


In the beginning, there is Markdown

Markdown is a plain-text markup language. What this means is that it doesn't require specialized software and that you can write without worrying about the looks (presentation), but just cater to the hierarchy (structure) of content and the content itself. This forced austerity is what makes it ideal for creative writers.

No need to sacrifice the appearance though: download `markdown.txt` to see what this text here has been converted from.



Markdown has a pared down syntax, but also comes in various *flavors* that enhance its capabilities. My flavor of choice is MultiMarkdown, simply because it includes footnotes and references. The best quickstart resource on the syntax, is iA's writer guide.

Some example marked-down text looks like this:

```
# This is a heading, a title
```

```
However this is some body content, including words in italics and bold.
It may also include [hyperlinks](https://youtu.be/2Z4m4lnjxkY)
and sometimes, lists of things:
```

- an apple
- an orange
- a zebra

Disadvantages

- **Unfamiliarity**; the greatest barrier. In a few occasions, one's workflow (see professional writers) may be too confined because of predetermined strict standards in collaborating with others. Most often however, the uninitiated are just too faint to try anything new. "Why bother?"

- Requires a certain **discipline** in mentally abstracting content and structure from presentation. Unfortunately most *Microsoft Word* users ignore proper structures jump right into changing typefaces and font sizes. If you've read this far, do everybody you work with a favor and learn about Style basics in Word.
- Requires a **customized pipeline**, or workflow to produce a presentable document. Then again, so do DTP applications.

Advantages

- **Consistency.** No bells and whistles, just content. You stay focused on ideas.
- **Portability.** Your documents are platform (Windows, Linux, MacOS, iOS, and Android) agnostic and software (MS Word, Libreoffice, Notepad) independent, which makes it easier to sync your files or use them on the web (WordPress etc) without distorted looks.
- **Ease of conversion.** A simple `.txt` formatted with Markdown, can be the source document of any number of Word documents, decent-looking PDFs, web pages and eBooks.

For readers familiar with LaTeX, all this may seem too simplistic, yet that is exactly the concept behind Markdown. The first was designed for PDF and other set-in-stone formats yet the latter arose from the continuous text flow of the web. Neither needs to replace the other, for their purpose is different.

Quick History

Markdown was created in 2004 by John Gruber as a quick way to generate HTML code without coding, and quickly gained popularity among web designers. It became further popular with GitHub and the Stack Exchange Q&A network adopting it for all descriptions and comments. WordPress, the most popular web publishing platform, supports it as well.

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

A promising group project to keep an eye on, is CommonMark, which works towards the standardization of the language's syntax and specs.

A proposed workflow

In most scenarios, the purpose is to have a source Markdown text and convert to a presentation-ready PDF. Publishing to the web is much easier and won't be covered herein. I have only sparingly used Markdown when

writing, but after a lot of research, I've come up with this workflow: *Sublime Text 3*; *Mendeley*; the *tufte-latex* template; *pandoc*.

Choosing an editor

Although you can simply use the humble *Notepad*, there are a number of enhanced Markdown writers which focus on simplicity and clean interfaces; zenware if you prefer. Most of them offer features such as auto-completion, spell-checking and live HTML preview, while some more advanced include document synchronization across platforms and multiple export options. Markdown files can have a *.txt*, or *.md*, or *.markdown* extension, but it doesn't really matter, because you can open them with any text editor.

Here are listed a few, several Open Source, in no specific order:

- Remarkable (Windows, Linux) Open Source, Free
- MacDown (MacOS) Open Source, Free
- WriteMonkey (Windows) Proprietary, Free
- Haroopad (Windows, Linux, MacOS) Proprietary, Free
- iA writer (iOS, MacOS, Android) Proprietary, Paid
- JotterPad (Android) Proprietary, Free
- Dillinger (online) Open Source, Free
- StackEdit (online) Proprietary, Freemium

For the technically oriented, many code editors have nice plugins for Markdown, notably:

- Atom (Windows, Linux, MacOS) Open Source, Free
 - markdown-preview (included)
 - markdown-pdf
 - markdown-scroll-sync
 - markdown-writer
- Sublime Text (Windows, Linux, MacOS) Proprietary, Shareware
 - Markdown Preview
 - MarkdownEditing

The extra reading section includes some guides on how to set up these editors. And for those times in a pinch, there's a handy online converter from Markdown to PDF.

Sublime Text

Although *pandoc* is indispensable, *Sublime Text* could be easily swapped with the open source *Atom* editor, but as of yet it lacks the polish and — most importantly— responsiveness of *Sublime Text*. My setup consists of the previously mentioned plugins, and this build script¹ which allows to run *pandoc* from within the editor:

¹ Tools > Build Systems > New Build System...

```
{
  "selector": "text.html.markdown",
  "shell": true,
  "cmd": [
    "C:/Users/Heracles/AppData/Local/Pandoc/pandoc.exe",
    "--latex-engine", "C:/Program Files/MiKTeX 2.9/miktex/bin/x64/xelatex.exe",
    "-S", "-s",
    "-f", "markdown_mmd+yaml_metadata_block",
    "--template", "tufte.tex",
    "-o", "$file_base_name.pdf", "$file_name",
    // "--toc",
    // "-bibliography", "citations.bib",
    "&", // *nix users should replace the & with a ;
    "C:/Program Files (x86)/SumatraPDF/SumatraPDF.exe", "$file_base_name.pdf",
  ]
}
```

Your mileage may vary depending on where you're going to install *pandoc* and *LaTeX*, so adjust the above accordingly. You may run this build system with a shortcut (F7 by default) and also open automatically the final document in your PDF reader of choice. Without any doubt, SumatraPDF is the lightest and fastest reader for Windows, plus it's free and Open Source.

Installing Mendeley (optional)

<https://www.mendeley.com/Mendeley> (Windows, Linux, MacOS, iOS, Android) is a freemium bibliography and reference manager. Zotero (Windows, Linux, MacOS) is an excellent FOSS alternative. They both offer great plugins for *MS Word* and *LibreOffice*, but what we need them for in this pipeline, is their ability to generate .bib files and style the references we use with a plethora of available reference styles.

Installing pandoc

Pandoc is a little command-line program for converting from one markup format to another. It supports many common and obscure text document formats.

You can install the command-line tool from pandoc.org in a very straight-forward way. For quick shortcuts to *pandoc* conversions in your right click menu, see the following:

- MarkdownMenu (Windows)
- Pandoc OSX Services (MacOS)

Otherwise, if you'd prefer a GUI then you may download PanConvert (Windows, Linux, MacOS) from its download page. For some mysterious

reason, many FOSS programs, like to say “download the newest Binaries” as if people understand what binaries are. It seems nice, but I haven’t used it.

Installing LaTeX

In order to make PDF files with *pandoc* You will also need a LATEX² engine. Long story short, this is a nice system for beautifully typeset documents. It has nothing to do with WYSIWYG word processors such as *LibreOffice Writer* and —oh dear— *Microsoft Word*, or DTP software such as *Abode InDesign*; each has a different purpose.

Pandoc currently suggests *MikTeX* for Windows users, and you can install it from miktex.org. Their suggested download is a basic set of the *TeX/LaTeX* system, and offers a nice on-the-fly download system for pissing packages, but from my experience, it only evokes a hellstorm of UAC windows and build timeouts, meaning you have to click accept too many times and probably re-run the *pandoc* commands the first time you use it.

Consider yourself warned if you go down this path, either way you choose, this is the most time consuming step of the whole process.

Unless you’re worried about disk space, I suggest selecting *Other Downloads* and picking the *OS Net Installer*; if you have a newer computer and know what you’re doing, go ahead and pick the *64-bit* one. Once you run the installer and agree with their license, select *Download MikTeX* and then *Complete MikTeX*. Then, pick a download source near to where you live, pour yourself a drink and wait for the download to complete. After downloading, you may have to run the installer again, in order to install the downloaded files.

Templates (optional)

Templates may be used to allow for different layouts when converting with *pandoc*. Under Windows they live in `%appdata%\Pandoc\templates` but if the folder doesn’t exist, you have to create it. Go ahead and Copy & Paste `%appdata%` in Explorer’s address bar.

I’m really fond of the Tufte-LaTeX handout style, which recreates the feel of the beautiful books by Edward Tufte, but there are many others available. The template we need is `handout.tex`, to which I’ve added support for Greek fonts and a couple of fixes for issues I’ve run into when converting from Markdown. The font used is *Gentium Plus* but a great alternative would be *Cardo* which is closer to *Bembo*, the main font Tufte’s books use. The monospaced font, is *Ubuntu Mono*. See the template’s source code about changing the fonts.

Of course it’s all a matter of personal preference; different use cases may call for different templates.

² The characters T, E, X in the name come from the Greek: τέχνη (skill, art, technique).

Making PDFs with pandoc

If you are a MacOS or Linux user, then you already know how to open a Terminal. If you're using Windows, go to the folder where you keep your Markdown files, right click somewhere in an empty area holding Shift and select *Open command window here*. Then paste the following, adjusting accordingly:

```
pandoc -s -S -from markdown_mmd+yaml_metadata_block -o "yourfile.pdf" "yourfile.md" --latex-engine=xelatex
```

The `+yaml_metadata_block` is particularly important if you include metadata in your text. Unfortunately *MultiMarkdown* uses a different style for metadata from *pandoc*, but here's hoping this is going to change soon. Append `--template=handout.tex` for the Tufte-style template and `--bibliography citations.bib` to include your bibliography file.

Changing the extension, allows for converting to a variety of formats. Simple as a cake.

How to draw a cat



1- Draw some circles.



2- Fill in the details.

Extra reading

Here follow a few links towards similar guides or with more to read on the tools suggested:

- [How to ditch Word](#)
- [GitHub Flavored Markdown](#)
- [Atom.io for Markdown editing](#)
- [How To Set Up Sublime Text for Markdown Editing](#)
- [Using markdown + pandoc to write my biology PhD thesis](#)
- [Instant PDFs via Right Click](#)
- [Pandoc User's Guide](#)
- [Markdown or LaTeX?](#)
- [How We Automated Our Ebook Builds With Pandoc and KindleGen](#)
- [Please don't send me Microsoft Word documents](#)

Congratulations are due if you've reached this far. I'd be glad to hear your feedback or help you out in your first steps, so don't shy away from contacting me.

*[HTML]: HyperText Markup Language

*[Q&A]: Question & Answer

*[GUI]: Graphical User Interface

*[FOSS]: Free and Open Source Software

*[UAC]: User Account Control

*[WYSIWYG]: What You See Is What You Get

*[DTP]: Desktop Publishing