# Introduction to Markdown and Pandoc

This section of the workshop introduces Markdown (a simple language designed for easy formatting of documents) and Pandoc (a program that converts between document types). You can use Markdown and Pandoc to collaboratively write documents and track text modifications. Markdown is also excellent for noting down ideas or writing guides for experiments and analyses.

## Introduction to Markdown

Markdown is a language with simple syntax that is useful for writing clear and well-formatted documents. You can do things like insert headings, make lists, and convert words to **bold** and *italic* by inserting symbols such as \* or #.

Markdown was designed to use simple syntax that does not interfere with readability of the text. The code shows your writing more clearly than to other languages like HTML or LaTeX. For example:

We can write \*\*bold text\*\*, \_italic text\_, or even  
  
1. make  
2. a  
3. list  
  
## We can also create headings  
### Smaller headings  
#### \_Or even smaller headings in italics\_

We can write **bold text**, *italic text*, or even

1. make
2. a
3. list

## We can also create headings

### Smaller headings

#### *Or even smaller headings in italics*

These workshop documents were written in in Markdown. If you are viewing this on Github, click 'Raw' to see the Markdown code that produced this text.

Markdown is also nice because it is interpreted similarly across computers. With the very simple formatting style you do not get strange/unexplained paragraph or line spacing changes that sometimes occur with word documents (for example when MS Word documents written in Windows are opened on a Mac). Markdown formatted text can also be generated automatically using other programming languages. For example, advanced users could write a script which generates the text for a manuscript and automatically inserts the most recent versions of figure image files.

There are Markdown editors which show code and formatted text simultaneously for [Mac](https://macdown.uranusjr.com/) and [Windows](http://markdownpad.com/). You can also edit Markdown documents [in your browser](https://stackedit.io/).

Here is a great [tutorial](www.markdowntutorial.com) for learning markdown, as well as a [cheat sheet](https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet) in case you forget how to format something.

## Collaborative writing

Markdown is code, which means that you can use repositories to keep track of your updates and changes. You can use all of the features described in the [Introduction to Git](https://github.com/Decision-Neuroscience-Lab/coding-workshop-material/blob/master/Writing-Clear-Code.md) section to manage your written text. Had a great paragraph that you deleted in an earlier edit? You can retrieve it by making a new branch off an earlier commit. Making edits to the methods section of a collaborator's manuscript draft? Create a new branch, make the edits, and then lodge a pull request to propose, describe and discuss your suggested changes.

Multiple people can work on the same document at once, and Git will keep track of everyone's input. For this reason Wiki pages for projects or toolboxes (for example the [documentation for DDTBOX](https://github.com/DDTBOX/tutorials/blob/master/DDTBox%20Documentation.md)) are commonly written in Markdown.

Here are some links describing authors' experiences of writing with Markdown:

[Collaborative writing on github with Markdown](https://oleb.net/blog/2016/02/collaborative-writing-on-github/)

[Writing technical papers with Markdown](http://blog.kdheepak.com/writing-papers-with-markdown.html)

[Writing scientific papers using Markdown](https://danieljhocking.wordpress.com/2014/12/09/writing-scientific-papers-using-markdown/)

[Using Markdown with R Studio](https://danieljhocking.wordpress.com/2013/09/25/knitting-beautiful-documents-in-rstudio/)

## Converting across document formats (introduction to Pandoc)

Pandoc is a tool for converting documents across different file types. For example, Pandoc lets you convert your markdown document into a Microsoft Word .docx file, a LaTeX document, a pdf, or something else. There is an [impressive list](http://pandoc.org/index.html) of file types that pandoc can handle. Pandoc also has capabilities for working with a range of citation formats, for example Endnote.

You can download Pandoc [here](https://github.com/jgm/pandoc/releases/tag/1.19.2.1), and installation instructions can be found [here](http://pandoc.org/installing.html). You can even use Pandoc [in your browser](http://pandoc.org/try/) without installing it on your computer, although the browser version only converts between a limited number of file types.

Pandoc uses the command line, which means that it does not have a graphical interface. A guide to using pandoc can be found [here](http://pandoc.org/getting-started.html). Most conversions can be done with a single command, as shown in this [list of examples](http://pandoc.org/demos.html). This list contains links to the original and converted files, so you can see what your converted files might look like.

As an example, you can convert this markdown file into a Microsoft Word .docx document by doing the following:

1. Open the command line (or terminal for Macs)
2. Navigate to the directory which contains the file that you want to convert.
3. Enter the following into the command line:

pandoc Intro-to-Markdown-and-Pandoc.md -f markdown -t docx -s -o test-pandoc-file.docx

The -f and -t parts tell pandoc which file types to convert between. The -o part goes before the output file name. An example of the word doc created from this markdown file is [included](https://github.com/Decision-Neuroscience-Lab/coding-workshop-material/blob/master/test-pandoc-file.docx) in the workshop repository (test-pandoc-file.docx).

A handy list of commands is available in the [user guide](http://pandoc.org/MANUAL.html) or by entering pandoc --help into the command line.

By using Markdown with Pandoc you can quickly and easily format your writing, keep track of your edits like you would with code, and convert your text into a manuscript in .docx or LaTeX format for your collaborators. By using Markdown with Git you also have a complete history of your writing process, and an easy way to backup your documents to a private repository in the cloud.

<<< [Code Peer-Review](https://github.com/Decision-Neuroscience-Lab/coding-workshop-material/blob/master/Code-Peer-Review.md)

[Back to workshop overview](https://github.com/Decision-Neuroscience-Lab/coding-workshop-material/blob/master/Coding%20Workshop%20DNLab.md)