

# Markdown #Citymash handout

## What this session will cover

- A discussion of some of the problems with proprietary file formats.
- A look at some of the benefits of using plain text files alongside some of the limitations of text files.
- a chance to play around with markdown
- a discussion of potential relevance to LIS?

## Problems with proprietary file formats

- Proprietary file formats encode text in binary format.
- There are different ways of accessing propitiatory formats:
  - We can use emulators such as to access old file types [A Wordstar emulator](#).
  - We can use OpenOffice or Libre Office to open most Doc and Docx files, but we cannot always <sup>1</sup> use word to open OpenOffice (odt) files.
  - it may be possible to access corrupted files using specialist software but this is very time consuming.

## Text files

‘Refers to textual data in ASCII format. Plain text is the most portable format because it is supported by nearly every application on every machine. It is quite limited, however, because it cannot contain any formatting commands.’  
<https://archive.org/details/wordstar2.26osborne1981micropro>

Because text files use minimal encoding understood by most computers they are very stable file formats that can easily move across different devices without compatibility issues.

---

<sup>1</sup>It is not always clear whether it is possible or not. It can depend on the version of the Word Processor being used at each end, the operating system and the type of files embedded in the document.

To demonstrate this we can try opening [old plain text files](#). This site collects text files primarily from the years 1980-95 all of which can still be opened both in your browser and with software on your computer. If we compare this to files produced by word processors we can have compatibility and formatting issues from files produced by an earlier version of the same software. Working with an individual file this may not be such a problem but if we are working with a whole collection of different documents we can see potential difficulties emerging.

### Some potential problems with plain text

Whilst plain text files are great in many ways, one of the potential problems with them is that they are a bit plain. Often we need to structure documents in some ways. Some text serves as headings, other points we might want to emphasise, we may want to separate quoted text from the rest of the text. This isn't about formatting the text yet but purely about the logical structure of the text. We can 'mark' these aspects of text in different ways. One solution would be to use HTML, however a problem with using HTML is that it is already more concerned with formatting how text will be displayed. Another potential option is to use LaTeX, which is a method designed for producing well typeset documents, however the syntax (language) has a bit of a learning curve and LaTeX documents are not primarily designed for the web.

### LaTeX example <sup>2</sup>

```
\documentclass[12pt]{article}

\setlength{\parskip}{1.2ex}    % space between paragraphs
\setlength{\parindent}{2em}    % amount of indentation
\setlength{\textwidth}{175cm}  % default = 6.5"
\setlength{\oddsidemargin}{-5mm} % default = 0"
\setlength{\textheight}{225mm} % default = 9"
\setlength{\topmargin}{-12mm}  % default = 0"

% --- define your own commands!! ---
\newcommand{\iii}{\indent \indent \indent} % triple-depth indent

\title{A Sample \LaTeX\ Document}
\author{Schl\oe ff\d{o}nffl\t{oo}\ae g\{"{e}n, \L\{"{a}rs}
\date{July 14, 1992}

\begin{document}

\maketitle
```

---

<sup>2</sup><http://spot.colorado.edu/~sitelic/samplecode/latex/sample.html>

```
\subsection{Spacing in the source text}
The ends of words and sentences are marked
  by spaces. It doesn't matter how many
spaces
you
type; one is as good as 100.
The end of a line counts as a space.
```

One or more blank lines denote the end of a paragraph.

Since any number of consecutive spaces are treated like a single one, the formatting of the input file makes no difference to `\TeX`, but it makes a difference to you. When you use `\LaTeX`, making your input file as easy to read as possible will be a great help as you write your document and when you change it. Keep typed lines `{\bf short}` in length, and use `\verb9 % comments9`. % like this!

Because printing is different from typewriting, there are a number of things that you have to do differently when preparing an input file than if you were just typing the document directly. Quotation marks like ```this''` have to be handled specially.

```
\end{document}
```

Although `LaTeX` is perfect for typesetting documents that will be printed, something we will return to later, it can be to cumbersome for short documents which may not necessarily be intended primarily for printing.

## What sort of documents do we need?

We may not always have a choice in the type of documents we are going to be working with, however, when a choice is available it makes sense to consider what characteristics we want our documents to have. The following is a non-comprehensive list of properties we want our documents to have:

- **Sustainable** - we want to use a document format which is sustainable and will be usable in both the short and long run.
- **Portable** - we should be able to move our documents around easily. Although memory isn't as big a problem it once was, it is still worth consideration. Portability not only refers to the size of documents but how easily it is opened across different devices and with different software.
- **'Translatable'** - although we may produce documents with an initial purpose, we may later want to use the document, or part of it, for a different purpose.

- **Version Control** - we want to write in a format which allows version control
- **Human readable** - This ties to the issue of sustainability, it should be possible for a human to read the raw text without compiling the document.

## What is Markdown?

Markdown is designed to address the limitations associated with plain text whilst retaining the benefits. It is markup which attempts to limit the amount of markup required; hence the ‘down’ in the name. Markdown is a syntax for setting out the structure of the document in a way which is human readable and compilable by an interpreter. This means that markdown files can easily be converted into different formats with consistent rules.

There are a number of different ‘flavours’ of markdown but they all have similar conventions so it doesn’t involve much effort to switch between them. Which you will use will largely depend on where you will be using your markdown files.

## How to write in Markdown

We don’t need any special software to write in Markdown since it is a plain text file. We can use Textedit on Mac or Notepad on Windows or use a more advanced text editor (not Word Processors). We just need to learn the syntax for markdown which doesn’t take long, and can be supplemented with Google searches when you forget how to do something. There are a variety of on-line markdown editors we can use to see how markdown works.

### Exercise 1 - Try an online markdown editor

1. Choose a cheat sheet which makes sense to you, the following are all good ([Markdown Cheatsheet](#), [Mastering Markdown Pandoc Markdown](#)) but there are plenty of others available online.
2. Choose one of the following online markdown editors or find another one if you don’t like any of these
  - [Dillinger](#)
  - [Stackedit](#)
  - [Markable](#)
3. Try writing some text using some common markdown syntax such as headings, lists, bold, italics etc. It might be helpful to try thinking about how you would structure different types of text i.e. a blog post, a web page or an academic piece of writing. Some examples are included in the github repository for this lesson.

4. These online editors are useful for picking up markdown because they usually include a live preview of what you have written which is useful when you aren't yet familiar with the markdown syntax.
5. Once you have got something written in markdown we can try doing a number of things with it.

### Some things to try with Markdown

These are a few different places we can play around with converting our markdown file into other formats.

**Markdown to HTML** There are a number of different ways that we can convert markdown files to HTML. Some websites 'natively' deal with markdown files. These include Wordpress and Github amongst others. We can also use tools to convert our markdown into HTML. There are number of ways we can do this. We can install software to do this on our computers but before we do that we may want to try playing around with some online tools.

1. Using the markdown file we just wrote try converting to HTML using one of these sites.
  - <http://daringfireball.net/projects/markdown/dingus>
  - Markable, one of the editors above allows us to convert to HTML.

**Optional - HTML to Markdown** It can be helpful to be able to convert from HTML to markdown. We may have existing work in HTML or want to use existing documents in HTML. If you have an existing HTML document you can try converting it to markdown. There is also an example HTML file in the github folder for this workshop you can use.

- <https://domchristie.github.io/to-markdown/>

### Markdown to PDF

We may originally write something in markdown which we later want to print. We can convert to PDF using an online converter.

- <http://www.markdowntopdf.com/>

The output of these files are not always perfect but it gives a demonstration of the benefits of markdown. We can use more advanced tools later to have more control over the process.

## Pandoc and LaTeX (final steps requires you to install software)

Pandoc is a powerful piece of software that allows conversion between a large number of document types. It is particularly useful for converting Markdown to a range of other formats including; HTML, HTML5, PDF, Docx, EPUB but also many more specialist document types. It is a potentially very useful tool for a number of applications.

**Trying Pandoc** Although it is not possible to use all of the features of pandoc without installing the software we can still try a number out some of the features of Pandoc using an online test version.

- <http://pandoc.org/try/> Pandoc allows you to convert between a large number of different formats

## Using Pandoc for academic writing

Pandoc extends some of the functionality of Markdown and in particular makes it more suitable for academic writing. If you have time you can try following a [tutorial](#) on the Programming Historian site. It is probably ambitious to try to cover all of the things that can be done with Pandoc in a short session but it might be possible to install the software (if you want) and to start exploring Pandoc.

**Disclaimer: although it is unlikely, unexpected things can happen to computers when you install new software. Pandoc carries no warranty of any kind.**

To use pandoc we will have to:

- install Pandoc on our computer. Instructions for different installation methods can be found [here](#).
- have a plain text editor. For now notepad or text edit will be fine though you may want to try out some more advanced text editors in the future.
- if we want to produce PDFs we will also need to install LaTeX. There are a variety of different installations of LaTeX available and it is probably worth setting aside some more time to install this.
- have some citations in bib format. There are some included in the folder for this lesson that you can use.

## Other uses for Markdown

There are other potential ways of using markdown: \* we can use markdown as a first writing stage before converting to LaTeX to play

around more closely with fancy typography \* we can use markdown and Jekyll to build a blog. The site for this conference can be found here: <https://github.com/citymash/citymash.github.io>. It uses Markdown, Jekyll and Github pages, though you can find more handsome examples of jekyll and markdown in action. - a tutorial on using markdown and Jekyll can be found [here](#) - the programming historian site gives an explanation of how they use Markdown and Jekyll to build their site. \* we can use markdown for making slides <http://remarkjs.com/#1>, <http://ogom.github.io/sublimetext-markdown-slideshow/> \* You will probably find more examples of using Markdown online or you may know of some yourself.

## Discussion?

We could potentially have a discussion about whether we think markdown is useful for librarians or not?

---