

# CITS4407 Open Source Tools and Scripting

## Writing in Markdown

Unit coordinator: Arran Stewart

# Overview

This week:

- Creating documents using Markdown

## CITS4407 documents and web pages

All the documents created for this course – web pages, lecture slides, “README” files, lab worksheets, and assignment specs – weren’t written using a word processor or any specialised tool, but just in the `vim` text editor, in a format called Markdown.

# What is Markdown?

Markdown specifies a way of writing text files so that you can indicate whether text is intended to be bold, italic, a heading, a link to a webpage, and so on.

# What is Markdown?

Markdown specifies a way of writing text files so that you can indicate whether text is intended to be bold, italic, a heading, a link to a webpage, and so on.

In a word processor like Microsoft Word, you can format paragraphs or portions of text using menus, buttons and shortcut keys. Once you do that, the formatting you've applied is immediately visible on-screen.

This is called WYSIWYG (“what you see is what you get”) formatting.

But Markdown works differently to that.

# What is Markdown?

When writing a document in Markdown format, you use special syntax to indicate how words and phrases should look when formatted.

The syntax is intended to be “intuitive” – reasonably similar to the sorts of informal “formatting” conventions people use when writing plain text files and messages. (For instance, you may see people use this in Help4407 messages.)

As an example – to indicate text should be bold, in Markdown, you put it between double asterisks:

```
**some bold text**
```

# Markdown syntax

Some other sorts of formatting you can indicate in Markdown include:

- italic - surrounded by single asterisks

You should *\*always\** remember to floss

- top-level heading - prefixed with a hash character

# My summer holiday

# Markdown syntax

- Bullet points - each item prefixed with an asterisk or hyphen

```
- one fish  
- two fish  
- red fish  
- blue fish
```



# Markdown syntax

- Numbered lists - first item prefixed with a number, other items with a number or hash

I sampled the defendant's wares:

1. on a boat
- #. with a goat

# Markdown syntax

This special syntax is called *markup*, and Markdown is called a *markup language*.

The term “markup” comes from the practice of editors “marking up” author’s typewritten manuscripts with a red or blue pencil to indicate to typesetters how the text should be formatted.

# Markup languages

Other sorts of Markup language exist. For instance, HTML is the Markup language used in web pages sent from a webserver to your browser.

In most browsers, there should be an option for “viewing code” of a web page. Doing that for Wikipedia’s page on “Markup languages” shows that the web page starts with content like this:

```
<!DOCTYPE html>
<html class="client-nojs" lang="en" dir="ltr">
<head>
<meta charset="UTF-8"/>
<title>Markup language - Wikipedia</title>
<script>document.documentElement.className="client-js";RLCONF={
"wgBreakFrames":!1,"wgSeparatorTransformTable":["",""],
"wgDigitTransformTable":["",""],"wgDefaultDateFormat":"dmy","wgMonthNames":
```

# Markup languages

In HTML, one can indicate a portion of text should be displayed in bold as follows:

```
<b>some bold text</b>
```

# Markup languages

Another markup language is Rich Text Format (RTF), devised by Microsoft for use in MS Word.

Text intended to be bold looks something like this:

```
{\pard \ql \f0 \sa180 \li0 \fi0 {\b some bold text}\par}
```

Markdown, however, tries to keep its syntax as readable as possible, and as close as possible to informal conventions for writing plain text.

# Processing Markdown

So how do we turn Markdown into web pages, or lecture slides, or PDF documents?

We need to use what's called a Markdown *processor*, which turns a text file written in Markdown into some other format.

# Processing Markdown

Sometimes, the processing is done for us “behind the scenes”.

For instance, if you write comments on the social media platform Reddit, the Reddit servers handle the task of turning your comments into HTML that can be displayed on the website.

all 5 comments

sorted by: best ▼

Hi, I'm having the same problems with a Bash script - I've tried  
\*everything\*!

save

[content policy](#) [formatting help](#)

# Processing Markdown

Likewise, if we create a website using [GitHub Pages](#) (which is how the website for CITS4407 is created), GitHub's server will turn our Markdown into HTML:

```
---  
title: "Resources"  
tags: ['toppage']  
layout: page  
---
```

Lecture slides and lab exercises will appear here as the semester progresses.

All material used in the unit is available online. Readings for each week are given in the  
[\*\*schedule\*\*]({{ "/schedule" | relative\_url }}){: class="hi-pri" :}.



# Processing Markdown

And some online Markdown editors like <https://dillinger.io> do something similar:

The screenshot shows the Dillinger online Markdown editor. The interface is split into two main sections: a left pane for editing Markdown and a right pane for previewing the rendered HTML. The top navigation bar includes a hamburger menu, the 'DILLINGER' logo, and buttons for 'PREVIEW AS', 'EXPORT AS', 'SAVE TO', and 'IMPORT FROM', along with a settings gear icon. Below the navigation bar, the document name is 'Untitled Document.nd'. On the right side of this bar, it shows 'READING TIME: 3 MIN READ' and 'WORDS: 549'. The left pane, labeled 'MARKDOWN', contains a list of 13 items, each with a number and a bullet point. The right pane, labeled 'PREVIEW', shows the rendered HTML output of the Markdown in the left pane. The rendered HTML includes a title 'Dillinger', a subtitle 'The Last Markdown Editor, Ever', a 'POWERED BY' section with the Solid logo, a 'build passing' status badge, and a list of features.

1 - Dillinger  
2 - The Last Markdown Editor, Ever\_  
3  
4 `[[!N|Solid|](https://cloud.com/dTspP1910f.thumb.png)](https://nodesource.co`  
5 `n/products/nsolid)`  
6 `[[!Build Status|](https://travis-ci.org/joemccann/dillinger.svg?branch=mast`  
7 `er)](https://travis-ci.org/joemccann/dillinger)`  
8 Dillinger is a cloud-enabled, mobile-ready, offline-storage compatible,  
9 AngularJS-powered HTML5 Markdown editor.  
10  
11 - Type some Markdown on the left  
12 - See HTML in the right  
13 - Magic

**Dillinger**  
*The Last Markdown Editor, Ever*  
POWERED BY Solid  
build passing  
Dillinger is a cloud-enabled, mobile-ready, offline-storage compatible,  
AngularJS-powered HTML5 Markdown editor.  
• Type some Markdown on the left  
• See HTML in the right  
• Magic

# Processing Markdown

But for the most control, you can download and run a Markdown processor yourself – the one we will use in lab/workshops is called *Pandoc* (<https://pandoc.org>).

## About pandoc

---

If you need to convert files from one markup format into another, pandoc is your swiss-army knife. Pandoc can convert between the following formats:

(← = conversion from; → = conversion to; ↔ = conversion from and to)

### Lightweight markup formats

- ↔ Markdown (including CommonMark and GitHub-flavored Markdown)
- ↔ reStructuredText
- AsciiDoc
- ↔ Emacs Org-Mode
- ↔ Emacs Muse
- ↔ Textile
- ← txt2tags

### HTML formats

- ↔ (X)HTML 4
- ↔ HTML5

### Ebooks

- ↔ EPUB version 2 or 3
- ↔ FictionBook2

### Word processor formats

- ↔ Microsoft Word docx
- ↔ OpenOffice/LibreOffice ODT
- OpenDocument XML
- Microsoft PowerPoint

### Interactive notebook formats

- ↔ Jupyter notebook (ipynb)

### Page layout formats

- InDesign ICML

### Wiki markup formats

- ↔ MediaWiki markup
- ↔ DokuWiki markup
- ← TikiWiki markup
- ← TWiki markup
- ← Vimwiki markup

# Processing Markdown

We'll see in labs how you can create your own webpages, PDF documents, or even e-books using Pandoc.

In the meantime, you can find out more about what Markdown looks like by looking at how it's used for the assignment README – <https://raw.githubusercontent.com/cits4407/assignment1/master/README.md>

```
# CITS4407 Assignment - Semester 1, 2021
```

```
## Details
```

```
Version: 0.3
```

```
Date: 2021-04-21
```

```
Changelog:
```

- 2021-04-07 - initial version
- 2021-04-19 - minor corrections and clarifications
- 2021-04-21 - submission procedure

```
Please check the CITS4407 website or the Git repository
```