



MARKDOWN

Markdown

- A lightweight language for creating documents.
- Relatively readable in raw form.
- GitHub will automatically render and display a README in markdown.
- Use the file extension .md

Markdown

- The syntax of markdown is not uniquely specified by the original design documentation (Gruber 2004) so there are variants.
- CommonMark is a newer specification that pins down the ambiguities and has been adopted as a base for many markdown flavors
- In practice, it is fairly transportable especially if you follow the best practices.

Text Size

- You have a limited ability to control the size of text using headings.
- The level/size of the heading is controlled by the number of # you put in front of it.
- Some markdown versions require a space after #

```
# Heading level 1 is largest
```

```
##### Heading level 6 is smallest
```

Text style

- Use * or _ pairs for italics
- Use ** or __ pairs for bold
- Use *** or ___ pairs for bold italics.
- Don't use underscore in the middle of words.

For **bold text** you can `__use__` * or _.
You can bold inside a word Un**it**e using
*, but _ may be unpredictable.
This is italics.
*****Pay attention.*****

Paragraphs

- The easiest way to create a paragraph is to use a blank line.
- If you need a line break, you can use a `
` at the end of the line.

Paragraph 1 is not very long.
But it does have a second sentence

Paragraph 2 is also not very long.

But it uses a line break.

Block Quotes

- We can use > at the front of paragraphs to quote the paragraph.
- Multiple > allow nesting

```
> Quoting the first paragraph.
```

```
>
```

```
>Includes the next paragraph if we include  
the blank line separator.
```

```
And we can nest.
```

```
>> Nested!
```

```
And the last line of the second paragraph  
is still quoted.
```

Code

- We use back ticks to indicate code.
- Inside a paragraph, Use a pair of back ticks to indicate code.
- To create a code block start and end the code with a line of three back ticks. (extended)

The function ``sumIt`` has the definition

```
```\n\ndef sumIt (n):\n    x = 0\n    for x in range(0,n):\n        sum += x\n    return sum\n```
```



# Unordered Lists

- Items are marked with a bullet
- Use \* or – (dash) to indicate an item. (Don't mix them.)
- Use 4 spaces to create a sublist

```
* Go to the store
* Buy Food
 * Rice
 * Beans
* Come home
```

# Ordered Lists

- Items are marked with a number and a period
- Use 4 spaces to indicate a sublist.
- Items go up by one each time, starting at the first value. There are places where the actual number for the item will be used to restart the list from that value. In general start with 1.

```
1. Go to the store
2. Buy Food
 1. Rice (4 space.. start at i)
 6. Beans (get ii)
1. Come home (get 3)
2. After coming home
3. And this should be the final line.
```

# Horizontal lines

- Three or more \* or – will give a horizontal rule.
- Use an empty line before and after. (Dropping the blank line generates a heading.)

Before the rule we want a blank line

---

And after as well.

# Links

- Use `[text](link)`.
- The link can either be relative or absolute.
- You can use styles on the links.

Lets do a search

[Google] (<https://www.google.com>)

Or go to another markdown

\*\*[page](second.md)\*\* local to the repo.

# Quick Links

- Use angle “brackets” < and > with a URL or email address.
- Use %20 instead of space.

Lets go to amazon <<https://www.amazon.com>>

# Images

- Use `![text](location)`.
- Typical to put any images together in a folder to avoid clutter.

We want an image here and  
`![Delete](images/delete.gif)` is a middle  
sized animated gif.

# Escape Characters

- If you want to use characters that indicate formatting as text, escape them with a backslash.

```
I want to ride the merry_go_round is
different to the merry_go_round.
```



# GITHUB FLAVORED MARKDOWN





# Extensions to Markdown

- The original markdown specification was too simple for some applications, so extensions exist.
- We will explore the additions available in GitHub Flavored Markdown which is a strict superset of CommonMark.

# Code Fencing

- This is not an extension, but part of the pinning down of the syntax. A code block is surrounded by at least 3 backticks or tildes.
- Can not mix.
- Closed by a matching against the opening with at least as many of the same fencing character as the open.

# HTML Block Specification

- Raw HTML code blocks can be added in appropriate ways.
- I advise sparing use as it may affect the readability of the markdown.

# Tables (Extension)

- Use bars (pipes) to separate items.
- Use dashes to separate headers from data.
- Use : to show alignment on dashes (:left/:center:/right:)
- Can add spaces to help readability

This is a truth table

	P		Q		P and Q	
	---		---		---	:
	T		T		T	
	T		F		F	
	F		T		F	
	F		F		F	

# CheckList (Extension)

- Like a list, but we have an additional [ ] or [x] indicating an unchecked or checked item.
- Need a space for the unchecked box.
  - [ ] item 1
  - [x] item 2 (it's done!)
  - [ ] item 3

# Strike through (Extension)

- A chunk of text wrapped in tilde pairs

`~~Oops, didn't mean to do this.~~`

# Math

- As of May 2022 supports rendering of mathematics via the use of embedded Tex/LaTeX.
- Use pairs of \$ to surround math inline or use \$\$ pairs to mark a block.


The function is  $f(x) = x^2$ .

\$\$


$$g(x) = x+2$$

+  $\pi$

\$\$



# JUPYTER NOTEBOOK MARKDOWN





# Jupyter Notebooks

- Make extensive use of markdown based on CommonMark
- Supports tables as an extension
- Supports rendering equations using  $pairs for inline and  $pairs for a block of lines.$$

The function is  $f(x) = x^2$ .

# References

- [MarkDown Syntax](#)
- [Markdown Cheat Sheet](#)
- [CommonMark Specifications](#)
- [GitHub Flavored Markdown](#)
- [GitHub example for markdown](#)