

CODE BEHOLD (HTTP://CODEBEHOLD.COM/)

Behold, the Code.

Markdown Cheatsheet

Markdown Cheat Sheet

Credit: Part of this cheatsheet is the original work of: revolunet (ongithub) (<https://github.com/revolunet/sublimetext-markdown-preview>), with additions/modifications I gathered from other sources (cse).

If you use the fabulous Sublime Text 2/3 editor (<http://sublimetext.com>) along with the Markdown Preview plugin (<https://github.com/revolunet/sublimetext-markdown-preview>), open your ST2 Palette with `CMD+⇧+P` then choose Markdown Preview in browser to see the result in your browser.

Text basics

this is **italic** and this is ****bold**** . another _italic_ and another __bold__

this is *italic* and this is **bold** . another *italic* and another **bold**

Important Text: Surround with Grave symbol (backtick): `` some text ``

this is `important text.` and percentage signs : % and %

FOOTNOTES

FOOTNOTE: [^{^footnote-1}] [^{^footnote-2}] [^{^cnn}] ...

Later, your footnotes will appear at the bottom of the document regardless of where you place their definition

This is a paragraph with a footnote (builtin parser only). [^{^footnote-1}] This is another footnote[^{^footnote-2}]. Footnotes, they should be on the news[^{^cnn}], but not the weather^{^weather} ([TWC]).

```
FOOTNOTE DEFINED HERE ... placed at 'foot' of document
[^footnote-1]: This is the text of the first footnote.
[^footnote-2]: This is the text of the second footnote.
[^cnn]: [CNN](http://cnn.com) - Cable News Network
[^weather]: [TWC]

[TWC]: http://weather.com
```

[^{^footnote-1}]: This is the text of the first footnote.

[^{^footnote-2}]: This is the text of the second footnote.

[^{^cnn}]: CNN (http://cnn.com) – Cable News Network

Insert [TOC] without spaces to generate a table of contents (builtin parsers only).

Superscript & Subscript

Surround with or

```
<sup>superscript</sup>
<sub>subscript</sub>
```

Example

This is a phrase with superscript^{cool} and with subscript_{sick}

Indentation

```
> Here is some indented text
>> even more indented
```

Here is some indented text

even more indented

Titles

```
# Big title (h1)
```

```
## Middle title (h2)
```

```
### Smaller title (h3)
```

```
#### and so on (hX)
```

```
##### and so on (hX)
```

```
##### and so on (hX)
```

Big title (h1)

Middle title (h2)

Smaller title (h3)

and so on (hX)

and so on (hX)

and so on (hX)

Example lists (1)

- bullets can be ``-``, ``+``, or ``*``
- bullet list 1
- bullet list 2
 - sub item 1
 - sub item 2

with indented text inside

- bullet list 3
- + bullet list 4
- * bullet list 5

- bullets can be `-`, `+`, or `*`
- bullet list 1
- bullet list 2
 - sub item 1
 - sub item 2

with indented text inside

- bullet list 3
- bullet list 4
- bullet list 5

Links

This is an [example inline link (google)](<http://google.com/>) and [another one with a title](<http://google.com/> "Hello, world").

Links can also be reference based : [reference 1 (google.com)][ref1] or [reference 2 with title (google.com)][ref2].

This is an example inline link (google) (<http://google.com/>) and another one with a title (<http://google.com/>).

Links can also be reference based : reference 1 (google.com) (<http://google.com/>) or reference 2 with title (google.com) (<http://google.com/>).

References are usually placed at the bottom of the document like this:

```
[ref1]: http://google.com/
```

Images

A sample image (quantumeon):

```
![Lena - Public Domain](http://quantumeon.com/info/cheatSheets/lena-public-domain.jpg "Hover Title: Lena - Public Domain")
```



Lena - Public Domain

Local relative path

```

```

As links, images can also use references instead of inline links :

```
![Lena][] ... Later in document, usually bottom ... [Lena]: img/lena.jpg
```

or as others shown above

```
[Lena]: http://quantumeon.com/info/cheatSheets/lena-public-domain.jpg "Lena - Public Domain"
```



Lena

Embed base64 images within the markdown!

Ensure the base 64 image is all on a single line — **no spaces!**

Embed Format

```
//ref in doc
![tag_txt][]

//link (at bottom of md file)
[tag_txt]:...
```

Example

```
// inline
![Redragon M901 PERDITION]( ... BnABNEAAAAAAA)
```

Better for the readability of your markdown file to include/embed base64 images as references

Redragon M901 PERDITION 16400 DPI High-Precision Programmable Laser Gaming Mouse for PC, MMO, 18 Programmable Buttons, Weight Tuning Cartridge, 12 Side Buttons, 5 programmable user profiles, Omron Micro Switches (Black)

```
// linked (better)
![Redragon M901 PERDITION][]
... then, at the bottom of your markdown file, place the reference
[Redragon M901 PERDITION]: ... ABNEAAAAAAA
```



Code

It's quite easy to show code in markdown files.

Backticks can be used to highlight some words.

Also, any indented block is considered a code block. If `enable_highlight` is `true`, syntax highlighting will be included (for the builtin parser – the github parser does this automatically).

```
<script>
  document.location = 'http://lmgty.com/?q=markdown+cheat+sheet';
</script>
```

Comments

Method 1

```
<!--  
    comment  
-->
```

Method 2: Doesn't show up in the converted source code

```
[//]: # (This may be the most platform independent comment)
```

Math

When `enable_mathjax` is `true`, inline math can be included $\left(\frac{\pi}{2}\right) \pi$

Alternatively, math can be written on its own line:

$$F(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

$$\int_0^1 f(t) dt$$

$$\sum_j \gamma_j^2 / d_j$$

GitHub Flavored Markdown

If you use the Github parser, you can use some of Github Flavored Markdown (<http://github.github.com/github-flavored-markdown/>) syntax :

- User/Project@SHA: `revolunet/sublimetext-markdown-preview@7da61baded468b5019869d11000307e07e07401`
- User/Project#Issue: `revolunet/sublimetext-markdown-preview#1`
- User : `@revolunet`

Some Python code :


```
import random

class CardGame(object):
    """ a sample python class """
    NB_CARDS = 32
    def __init__(self, cards=5):
        self.cards = random.sample(range(self.NB_CARDS), 5)
        print 'ready to play'
```

Some Javascript code :

```
var config = {
    duration: 5,
    comment: 'WTF'
}
// callbacks beauty un action
async_call('/path/to/api', function(json) {
    another_call(json, function(result2) {
        another_another_call(result2, function(result3) {
            another_another_another_call(result3, function(result4) {
                alert('And if all went well, i got my result :)');
            });
        });
    });
});
```

The Github Markdown also brings some nice Emoji support (<http://www.emoji-cheat-sheet.com/>) : +1: :heart: :beer:

Parsers and Extensions

Markdown Preview comes with **Python-Markdown** preloaded.

Python-Markdown

The Python-Markdown Parser (<https://github.com/waylan/Python-Markdown>) provides support for several extensions.

Extra Extensions

- `abbr` — Abbreviations (<http://pythonhosted.org/Markdown/extensions/abbreviations.html>)
- `attr_list` — Attribute Lists (http://pythonhosted.org/Markdown/extensions/attr_list.html)
- `def_list` — Definition Lists (http://pythonhosted.org/Markdown/extensions/definition_lists.html)
- `fenced_code` — Fenced Code Blocks (http://pythonhosted.org/Markdown/extensions/fenced_code_blocks.html)
- `footnotes` — Footnotes (<http://pythonhosted.org/Markdown/extensions/footnotes.html>)
- `tables` — Tables (<http://pythonhosted.org/Markdown/extensions/tables.html>)
- `smart_strong` — Smart Strong (http://pythonhosted.org/Markdown/extensions/smart_strong.html)

You can enable them all at once using the `extra` keyword.

```
extensions: [ 'extra' ]
```

If you want all the extras plus the `toc` extension,
your settings would look like this:

```
{
    ...
    parser: 'markdown',
    extensions: ['extra', 'toc'],
    ...
}
```

Other Extensions

There are also some extensions that are not included in Markdown Extra
but come in the standard Python-Markdown library.

- `code-hilite` — CodeHilite (http://pythonhosted.org/Markdown/extensions/code_hilite.html)
- `html-tidy` — HTML Tidy (http://pythonhosted.org/Markdown/extensions/html_tidy.html)
- `header-id` — HeaderId (http://pythonhosted.org/Markdown/extensions/header_id.html)

- `meta_data` — Meta-Data (http://pythonhosted.org/Markdown/extensions/meta_data.html)
- `nl2br` — New Line to Break (<http://pythonhosted.org/Markdown/extensions/nl2br.html>)
- `rss` — RSS (<http://pythonhosted.org/Markdown/extensions/rss.html>)
- `sane_lists` — Sane Lists (http://pythonhosted.org/Markdown/extensions/sane_lists.html)
- `smarty` — Smarty (<https://pythonhosted.org/Markdown/extensions/smarty.html>)
- `toc` — Table of Contents (<http://pythonhosted.org/Markdown/extensions/toc.html>)
- `wikilinks` — WikiLinks (<http://pythonhosted.org/Markdown/extensions/wikilinks.html>)

3rd Party Extensions

Python-Markdown is designed to be extended.

Some included ones are:

- `delete` — github style delete support via `~~word~~`
- `githubemoji` — github emoji support
- `tasklist` — github style tasklists
- `magiclink` — github style auto link conversion of http|ftp links
- `headeranchor` — github style header anchor links
- `github` — Adds the above extensions in one shot
- `b64` — convert and embed local images to base64. Setup by adding this `b64(base_path=${BASE_PATH})`

There are also a number of others available:

Just fork this repo and add your extensions inside the `.../Packages/Markdown Preview/markdown/extensions/` folder.

Check out the list of 3rd Party extensions (<https://github.com/waylan/Python-Markdown/wiki/Third-Party-Extensions>).

Default Extensions

The default extensions are:

- `footnotes` — Footnotes (<http://pythonhosted.org/Markdown/extensions/footnotes.html>)
- `toc` — Table of Contents (<http://pythonhosted.org/Markdown/extensions/toc.html>)
- `fenced_code` — Fenced Code Blocks (http://pythonhosted.org/Markdown/extensions/fenced_code_blocks.html)
- `tables` — Tables (<http://pythonhosted.org/Markdown/extensions/tables.html>)

Use the `default` keyword, to select them all.

If you want all the defaults plus the `definition_lists` extension, your settings would look like this:

```
{
    ...
    parser: 'markdown',
    extensions: ['default', 'definition_lists'],
    ...
}
```

Examples

Tables

The `tables` extension of the *Python-Markdown* parser is activated by default, but is currently **not** available in *Markdown2*.

The syntax was adopted from the php markdown project (<http://michelf.ca/projects/php-markdown/extra/#table>), and is also used in github flavoured markdown.

```
| Year | Temp °C (low) | Temp °C (high) |
| --- | --- | --- | ← notice only three dashes are needed for each column
| 1900 |          -10 |          25 |
| 1910 |          -15 |          30 |
| 1920 |          -10 |          32 |
```

Yields:

Year **Temp °C (low)** **Temp °C (high)**

1900-10 25

1910-15 30

1920-10 32

COLON – COLUMN ALIGNMENT

| ----: | ← Right Align

| :---: | ← Center Align

| :--- | ← Left Align (default if no colons are used)

Alignment shown below: Right, Center, Center

DirectionalFocus Win +Move Win + Shift +

Left	J	J
Up	K	K
Down	L	L
Right	;	;

Wiki Tables

If you are using *Markdown2* with the `wiki-tables` extra activated you should see a table below:

```
|| Year || Temperature (low) || Temperature (high) ||
|| 1900 || -10 || 25 ||
|| 1910 || -15 || 30 ||
|| 1920 || -10 || 32 ||
```

Definition Lists

This example requires *Python Markdown*'s `def_list` extension.

Apple

: Pomaceous fruit of plants of the genus *Malus* in the family Rosaceae.

Orange

: The fruit of an evergreen tree of the genus *Citrus*.