



Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

[Read the guide](#)

Markdown Cheatsheet

Adam Pritchard edited this page on May 29, 2017 · 96 revisions

This is intended as a quick reference and showcase. For more complete info, see [John Gruber's original spec](#) and the [Github-flavored Markdown info page](#).

Note that there is also a [Cheatsheet specific to Markdown Here](#) if that's what you're looking for. You can also check out [more Markdown tools](#).

Table of Contents

- [Headers](#)
- [Emphasis](#)
- [Lists](#)
- [Links](#)
- [Images](#)
- [Code and Syntax Highlighting](#)
- [Tables](#)
- [Blockquotes](#)
- [Inline HTML](#)
- [Horizontal Rule](#)
- [Line Breaks](#)
- [YouTube Videos](#)

Headers

```
# H1
## H2
### H3
#### H4
##### H5
##### H6
```

▼ Pages 10

[Home](#)

[Compatibility](#)

[Development Notes](#)

[Markdown Cheatsheet](#)

[Markdown Here Cheatsheet](#)

[Other Markdown Tools](#)

[Press, Posts, Reviews, Etc.](#)

[Reviews](#)

[Tips and Tricks](#)

[Troubleshooting](#)

Clone this wiki locally

<https://github.com/adam-p/m>



Alternatively, for H1 and H2, an underline-ish style:

```
Alt-H1
=====

Alt-H2
-----
```

H1

H2

H3

H4

H5

H6

Alternatively, for H1 and H2, an underline-ish style:

Alt-H1

Alt-H2

Emphasis

Emphasis, aka italics, with *asterisks* or *underscores*.

Strong emphasis, aka bold, with **asterisks** or **underscores**.

Combined emphasis with **asterisks and underscores**.

Strikethrough uses two tildes. ~~Scratch this.~~

Emphasis, aka italics, with *asterisks* or *underscores*.

Strong emphasis, aka bold, with **asterisks** or **underscores**.

Combined emphasis with **asterisks and underscores**.

Strikethrough uses two tildes. ~~Scratch this.~~

Lists

(In this example, leading and trailing spaces are shown with with dots: ·)

```
1. First ordered list item
2. Another item
..* Unordered sub-list.
1. Actual numbers don't matter, just that it's a number
..1. Ordered sub-list
4. And another item.

...You can have properly indented paragraphs within list items. Notice the blank line
above, and the leading spaces (at least one, but we'll use three here to also align
the raw Markdown).

...To have a line break without a paragraph, you will need to use two trailing
spaces...
...Note that this line is separate, but within the same paragraph...
...(This is contrary to the typical GFM line break behaviour, where trailing spaces
are not required.)

* Unordered list can use asterisks
- Or minuses
+ Or pluses
```

1. First ordered list item
2. Another item

- Unordered sub-list.

1. Actual numbers don't matter, just that it's a number

2. Ordered sub-list

3. And another item.

You can have properly indented paragraphs within list items. Notice the blank line above, and the leading spaces (at least one, but we'll use three here to also align the raw Markdown).

To have a line break without a paragraph, you will need to use two trailing spaces.

Note that this line is separate, but within the same paragraph.

(This is contrary to the typical GFM line break behaviour, where trailing spaces are not required.)

- Unordered list can use asterisks

- Or minuses

- Or pluses

Links

There are two ways to create links.

```
[I'm an inline-style link](https://www.google.com)

[I'm an inline-style link with title](https://www.google.com "Google's Homepage")

[I'm a reference-style link][Arbitrary case-insensitive reference text]

[I'm a relative reference to a repository file](../blob/master/LICENSE)

[You can use numbers for reference-style link definitions][1]

Or leave it empty and use the [link text itself].

URLs and URLs in angle brackets will automatically get turned into links.
http://www.example.com or <http://www.example.com> and sometimes
example.com (but not on Github, for example).

Some text to show that the reference links can follow later.

[arbitrary case-insensitive reference text]: https://www.mozilla.org
[1]: http://slashdot.org
[link text itself]: http://www.reddit.com
```

[I'm an inline-style link](https://www.google.com)

[I'm an inline-style link with title](https://www.google.com "Google's Homepage")

[I'm a reference-style link](#)

[I'm a relative reference to a repository file](#)

[You can use numbers for reference-style link definitions](#)

Or leave it empty and use the [link text itself](#).

URLs and URLs in angle brackets will automatically get turned into links.
<http://www.example.com> or <http://www.example.com> and sometimes example.com (but not on Github, for example).

Some text to show that the reference links can follow later.

Images

Here's our logo (hover to see the title text):

Inline-style:
![alt text](https://github.com/adam-p/markdown-
here/raw/master/src/common/images/icon48.png "Logo Title Text 1")

Reference-style:
![alt text][logo]

[logo]: https://github.com/adam-p/markdown-
here/raw/master/src/common/images/icon48.png "Logo Title Text 2"

Here's our logo (hover to see the title text):



Inline-style:



Reference-style:

Code and Syntax Highlighting

Code blocks are part of the Markdown spec, but syntax highlighting isn't. However, many renderers -- like Github's and *Markdown Here* -- support syntax highlighting. Which languages are supported and how those language names should be written will vary from renderer to renderer. *Markdown Here* supports highlighting for dozens of languages (and not-really-languages, like diffs and HTTP headers); to see the complete list, and how to write the language names, see the [highlight.js demo page](#).

```
Inline `code` has `back-ticks` around it.
```

Inline `code` has `back-ticks` around it.

Blocks of code are either fenced by lines with three back-ticks `````, or are indented with four spaces. I recommend only using the fenced code blocks -- they're easier and only they support syntax highlighting.

```
```javascript
var s = "JavaScript syntax highlighting";
alert(s);
```

```python
s = "Python syntax highlighting"
print s
```

```
No language indicated, so no syntax highlighting.
But let's throw in a tag.
```
```

```
var s = "JavaScript syntax highlighting";
alert(s);
```

```
s = "Python syntax highlighting"
print s
```

No language indicated, so no syntax highlighting in Markdown Here (varies on Github).
But let's throw in a `tag`.

Tables

Tables aren't part of the core Markdown spec, but they are part of GFM and *Markdown Here* supports them. They are an easy way of adding tables to your email -- a task that would otherwise require copy-pasting from another application.

Colons can be used to align columns.

```
Tables	Are	Cool
col 3 is	right-aligned	$1600
col 2 is	centered	$12
zebra stripes	are neat	$1
```

There must be at least 3 dashes separating each header cell.
The outer pipes (|) are optional, and you don't need to make the raw Markdown line up prettily. You can also use inline Markdown.

```
Markdown	Less	Pretty
*Still* | `renders` | **nicely**
1 | 2 | 3
```

Colons can be used to align columns.

| Tables | Are | Cool |
|---------------|---------------|--------|
| col 3 is | right-aligned | \$1600 |
| col 2 is | centered | \$12 |
| zebra stripes | are neat | \$1 |

There must be at least 3 dashes separating each header cell. The outer pipes (|) are optional, and you don't need to make the raw Markdown line up prettily. You can also use inline Markdown.

| Markdown | Less | Pretty |
|--------------|----------------------|---------------|
| <i>Still</i> | <code>renders</code> | nicely |
| 1 | 2 | 3 |

Blockquotes

```
> Blockquotes are very handy in email to emulate reply text.
> This line is part of the same quote.
```

Quote break.

```
> This is a very long line that will still be quoted properly when it wraps. Oh boy let's keep writing to make sure this is long enough to actually wrap for everyone. Oh, you can *put* Markdown into a blockquote.
```

Blockquotes are very handy in email to emulate reply text. This line is part of the same quote.

Quote break.

This is a very long line that will still be quoted properly when it wraps. Oh boy let's keep writing to make sure this is long enough to actually wrap for everyone. Oh, you can *put* **Markdown** into a blockquote.

Inline HTML

You can also use raw HTML in your Markdown, and it'll mostly work pretty well.

```
<dl>
  <dt>Definition list</dt>
  <dd>Is something people use sometimes.</dd>

  <dt>Markdown in HTML</dt>
  <dd>Does *not* work **very** well. Use HTML <em>tags</em>.</dd>
</dl>
```

Definition list

Is something people use sometimes.

Markdown in HTML

Does **not** work ****very**** well. Use HTML *tags*.

Horizontal Rule

Three or more...

Hyphens

Asterisks

—

Underscores

Three or more...

Hyphens

Asterisks

Underscores

Line Breaks

My basic recommendation for learning how line breaks work is to experiment and discover -- hit <Enter> once (i.e., insert one newline), then hit it twice (i.e., insert two newlines), see what happens. You'll soon learn to get what you want. "Markdown Toggle" is your friend.

Here are some things to try out:

```
Here's a line for us to start with.
```

```
This line is separated from the one above by two newlines, so it will be a *separate paragraph*.
```

```
This line is also a separate paragraph, but...
```

```
This line is only separated by a single newline, so it's a separate line in the *same paragraph*.
```

Here's a line for us to start with.

This line is separated from the one above by two newlines, so it will be a *separate paragraph*.

This line is also begins a separate paragraph, but...

This line is only separated by a single newline, so it's a separate line in the *same paragraph*.

(Technical note: *Markdown Here* uses GFM line breaks, so there's no need to use MD's two-space line breaks.)

YouTube Videos

They can't be added directly but you can add an image with a link to the video like this:

```
<a href="http://www.youtube.com/watch?feature=player_embedded&v=YOUTUBE_VIDEO_ID_HERE
```



```
" target="_blank"></a>
```

Or, in pure Markdown, but losing the image sizing and border:

```
[![IMAGE ALT TEXT HERE](http://img.youtube.com/vi/YOUTUBE_VIDEO_ID_HERE/0.jpg)]
(http://www.youtube.com/watch?v=YOUTUBE_VIDEO_ID_HERE)
```

Referencing a bug by #bugID in your git commit links it to the slip. For example #1.

License: 