

Github Markdown Flavored document

Here is a sample **Python** *listing*:

```
def f(x):
    s = ("Test", 2 + 3, {'a': 'b'}, x) # Comment
    if s:
        print(s[0].lower())

class Foo:
    def __init__(self):
        byte_string = 'newline:\n also newline:\x0a'
        text_string = "Cyrillic Я is \u042f."
        self.makeSense(whatever=1)

    def make_sense(self, whatever):
        self.sense = whatever

x = len('abc')
print(f.__doc__)
```

And a sample **Java** *listing*:

```
public class Application {
    public static void main(String[] args) {
        System.out.println("An awesome text.");
    }
}
```

Headers

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

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

```
Alt-H1
=====
```

```
Alt-H2
```

H1

H2

H3

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

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
```

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. es

```
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 <b>tag</b>.
'''

```

```

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

```

```

s = "Python syntax highlighting"
print(s)

```

```

No language indicated, so no syntax highlighting.
But let's throw in a <b>tag</b>.

```

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**

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>	renders	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.