Published in Towards Data Science

Mahbub Zaman Follow
Apr 9, 2021 · 12 min read · ✦ · Listen

Save

The Ultimate Markdown Cheat Sheet

Write an awesome README

Photo by Christopher Gower on Unsplash

What is Markdown? Markdown is a way of writing rich-text (formatted text) content using plain text formatting syntax. From this post, you'll learn all the Markdown's major commands that will help you create an awesome GitHub README. I'll talk about the 11 most commonly used markdown topics. I've also mentioned different ways, such as using HTML tags to style your README. If you're interested, you can have a look at the HTML whitelists on GitHub.

. . .

You can download everything I've discussed in this post from my GitHub repository. Below I have a table of content so that you can easily navigate this post.

```
• Headings
• Text styles
• Syntax Highlighting
• Alignments
• Tables
• Links
• Images
• Lists
• Buttons (6 January 2023)
• Horizontal Rule
• Diagrams (19 July 2022)
• Mathematical expressions (19 July 2022)
• Miscellaneous
• Bitbucket
• Azure DevOps Project wiki
• Tools
```

Headings

There are a few options for creating headings. We can use Markdown, HTML, or an alternative syntax to create our desired headings.

First, let's talk about the markdown syntax.

```
# Heading 1
## Heading 2
### Heading 3
#### Heading 4
##### Heading 5
```

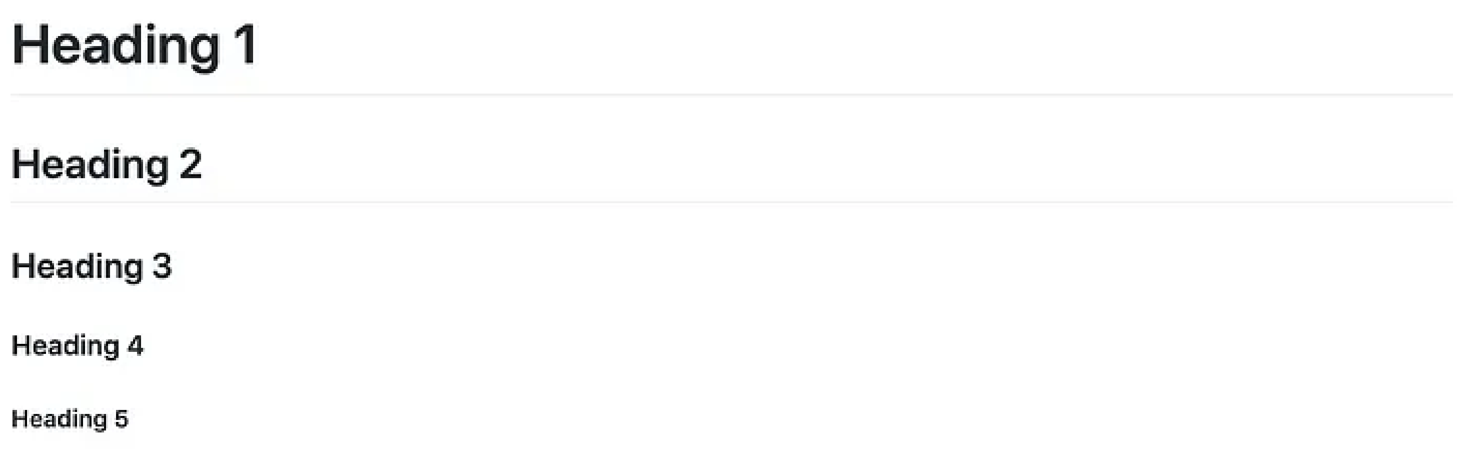
The second option uses the HTML syntax.

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
```

Finally, we can use an alternate syntax. This method only works for heading 1 and heading 2. Add any number of `=` or `-` below the text for heading 1 or heading 2.

```
Heading 1
=
Heading 2
-
```

Now, let’s see how it looks on GitHub.



Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Headings

Text styles

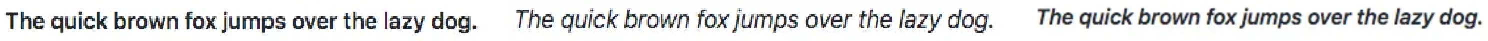
Using markdown syntax, we can change texts’ styles, including bold, italic, blockquotes, monospaced, underlined, strike-through, boxed, subscript, and superscript.

We can use two asterisks (`**`), underscores (`__`), or an HTML tag `<strong>` to make the text bold. To make the text italic, we can use one asterisk (`*`), underscore (`_`), or an HTML tag `<em>`. Also, we can make the text bold and italic at the same time.

```
Bold
**The quick brown fox jumps over the lazy dog.**
__The quick brown fox jumps over the lazy dog.__
```

```
<strong>The quick brown fox jumps over the lazy dog.</strong>
Italic
*The quick brown fox jumps over the lazy dog.*
_The quick brown fox jumps over the lazy dog._
<em>The quick brown fox jumps over the lazy dog.</em>
Bold and Italic
**_The quick brown fox jumps over the lazy dog._**
<strong><em>The quick brown fox jumps over the lazy dog.</em></strong>
```

Now, let’s see how it looks on GitHub.



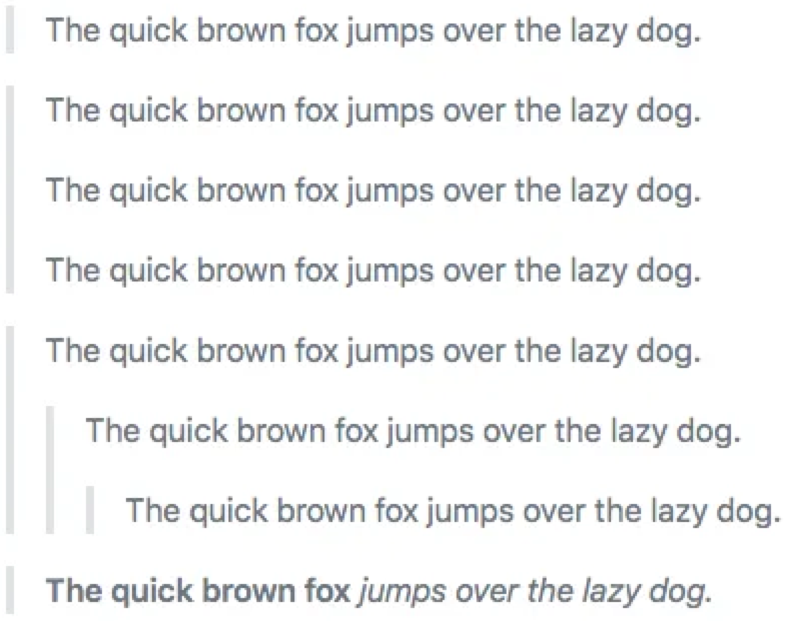
The quick brown fox jumps over the lazy dog. *The quick brown fox jumps over the lazy dog.* ***The quick brown fox jumps over the lazy dog.***

Bold and Italic text styles

To create a blockquote, we can use the greater than sign `>`. We can create a single-line or multi-line blockquote. Also, blockquote inside a blockquote. We can add other text styles inside a blockquote, such as bold or italic text styles.

```
Blockquotes
> The quick brown fox jumps over the lazy dog.

> The quick brown fox jumps over the lazy dog.
>
> The quick brown fox jumps over the lazy dog.
>
> The quick brown fox jumps over the lazy dog.

> The quick brown fox jumps over the lazy dog.
>> The quick brown fox jumps over the lazy dog.
>>> The quick brown fox jumps over the lazy dog.

> **The quick brown fox** *jumps over the lazy dog.*
```

Now, let’s see how it looks on GitHub.



> The quick brown fox jumps over the lazy dog.

> The quick brown fox jumps over the lazy dog.
>
> The quick brown fox jumps over the lazy dog.
>
> The quick brown fox jumps over the lazy dog.

> The quick brown fox jumps over the lazy dog.
>
> > The quick brown fox jumps over the lazy dog.
> >
> > > The quick brown fox jumps over the lazy dog.

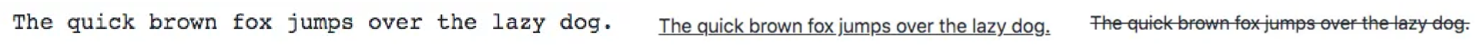
> **The quick brown fox** *jumps over the lazy dog.*

Blockquotes text styles

We can achieve monospaced and underlined styles using HTML tags `<samp>` and `<ins>`. For a strike-through style, we can use two tilda sign `~~`.

```
Monospaced
<samp>The quick brown fox jumps over the lazy dog.</samp>

Underlined
<ins>The quick brown fox jumps over the lazy dog.</ins>

Strike-through
~~The quick brown fox jumps over the lazy dog.~~
```

Now, let’s see how it looks on GitHub.



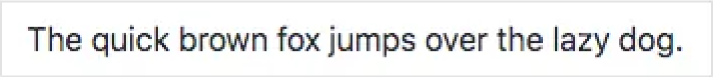
`The quick brown fox jumps over the lazy dog.` <ins>The quick brown fox jumps over the lazy dog.</ins> ~~The quick brown fox jumps over the lazy dog.~~

Monospaced, Underlined, and Strike-through text styles

We can use an HTML `<table>` tag to create a box.

```
Boxed
<table><tr><td>The quick brown fox jumps over the lazy dog.</td></tr></table>
```

Now, let’s see how it looks on GitHub.

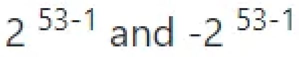

The quick brown fox jumps over the lazy dog.

Boxed text style

We can achieve subscript and superscript styles using HTML tags `<sub>` and `<sup>`. It is useful when you're writing a mathematical formula.

```
2 <sup>53-1</sup> and -2 <sup>53-1</sup>
```

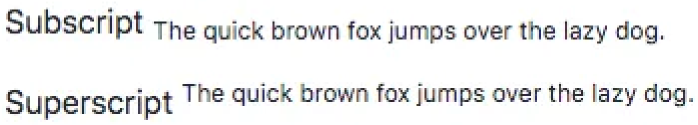
Now, let's see how it looks on GitHub.


2^{53-1} and -2^{53-1}

Subscript and Superscript text styles

```
Subscript <sub>The quick brown fox jumps over the lazy dog.</sub>
Superscript <sup>The quick brown fox jumps over the lazy dog.</sup>
```

Now, let's see how it looks on GitHub.


Subscript $_{\text{The quick brown fox jumps over the lazy dog.}}$

Superscript $^{\text{The quick brown fox jumps over the lazy dog.}}$

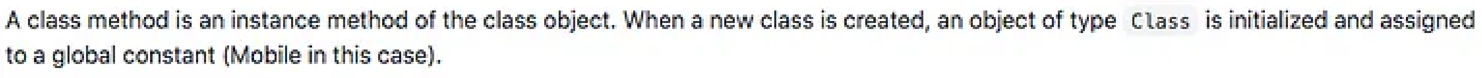
Subscript and Superscript text styles

Syntax Highlighting

We can use a single backtick `` ` `` before and after the code block to create the following view.

```
A class method is an instance method of the class object. When a new class is created, a
```

As you can see, the word **Class** is highlighted.

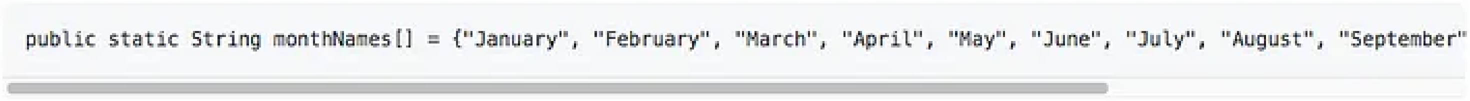


A class method is an instance method of the class object. When a new class is created, an object of type `Class` is initialized and assigned to a global constant (Mobile in this case).

Code Highlighting

We can also use triple backticks ``` before and after the code block to create the following view.

~~~
```
public static String monthNames[] = {"January", "February", "March", "April", "May", "Ju
```
~~~



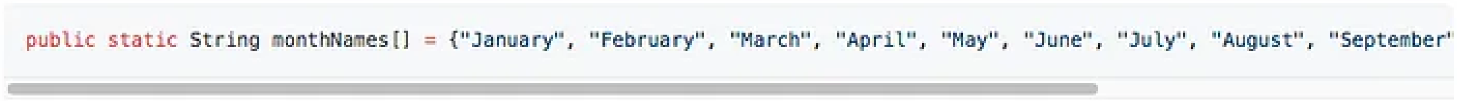
```
public static String monthNames[] = {"January", "February", "March", "April", "May", "June", "July", "August", "September"
```

Code Highlighting

We can add an optional language identifier to enable syntax highlighting. Refer to this and this GitHub document to find all the valid keywords.

~~~
```java
public static String monthNames[] = {"January", "February", "March", "April", "May", "Ju
```
~~~

Now, let’s see how it looks on GitHub.



```java
public static String monthNames[] = {"January", "February", "March", "April", "May", "June", "July", "August", "September"
```

Syntax Highlighting

Alignments

By using HTML tags, we can align README contents.

```
<p align="left">
<img src="https://images.unsplash.com/photo-1415604934674-561df9abf539?ixlib=rb-1.2.1&ix
</p>
```

Left align

```
<p align="center">
<img src="https://images.unsplash.com/photo-1415604934674-561df9abf539?ixlib=rb-1.2.1&ix
</p>
```

Center align

```
<p align="right">
<img src="https://images.unsplash.com/photo-1415604934674-561df9abf539?ixlib=rb-1.2.1&ix
</p>
```

Right align

Now, let's align a text.

```
<h3 align="center"> My latest Medium posts </h3>
```

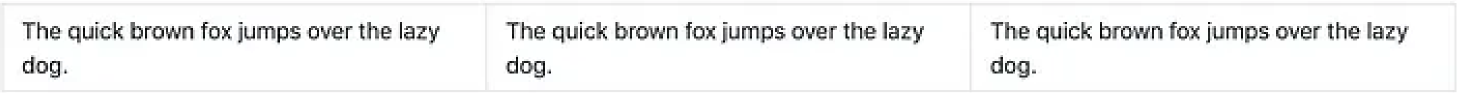
My latest Medium posts

Center align

Tables

Let’s create a table without headers.

```
<table>
<tr>
<td width="33%">
The quick brown fox jumps over the lazy dog.
</td>
<td width="33%">
The quick brown fox jumps over the lazy dog.
</td>
<td width="33%">
The quick brown fox jumps over the lazy dog.
</td>
</tr>
</table>
```



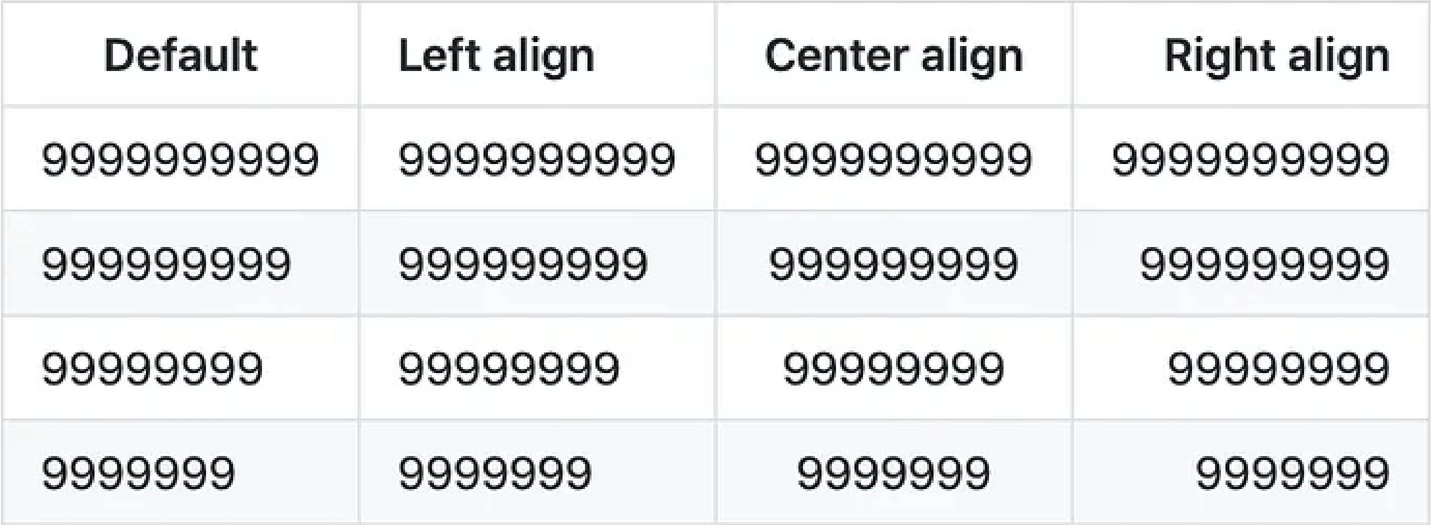
The quick brown fox jumps over the lazy dog.	The quick brown fox jumps over the lazy dog.	The quick brown fox jumps over the lazy dog.

Table without header

To create a table with headers, we need to use dashes to separate each header cell and use pipes to separate columns. The outer pipes are optional. We can use any number of dashes and spaces to increase readability. We can use colons to align columns. For left-align text, use a colon to the left of the dashes. For center-align text, use a colon on both sides of the dashes. For right-align text, use a colon to the right of the dashes. By default Left alignment is used.

```
| Default | Left align | Center align | Right align |
| - | :- | :-: | -: |
| 9999999999 | 9999999999 | 9999999999 | 9999999999 |
| 999999999 | 999999999 | 999999999 | 999999999 |
| 99999999 | 99999999 | 99999999 | 99999999 |
| 9999999 | 9999999 | 9999999 | 9999999 |
| Default    | Left align | Center align | Right align |
| ---------- | :--------- | :----------: | ----------: |
| 9999999999 | 9999999999 | 9999999999   | 9999999999  |
| 999999999  | 999999999  | 999999999    | 999999999   |
| 99999999   | 99999999   | 99999999     | 99999999    |
| 9999999    | 9999999    | 9999999      | 9999999     |
Default    | Left align | Center align | Right align
---------- | :--------- | :----------: | ----------:
```

```
9999999999 | 9999999999 | 9999999999    | 9999999999
999999999  | 999999999  | 999999999     | 999999999
99999999   | 99999999   | 99999999      | 99999999
9999999    | 9999999    | 9999999       | 9999999
```

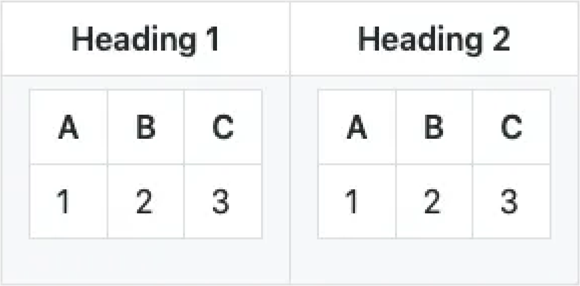


Default	Left align	Center align	Right align
9999999999	9999999999	9999999999	9999999999
999999999	999999999	999999999	999999999
99999999	99999999	99999999	99999999
9999999	9999999	9999999	9999999

Table with different alignments

Now display two tables side by side.

```
<table>
<tr>
<th>Heading 1</th>
<th>Heading 2</th>
</tr>
<tr>

<td>

| A | B | C |
|--|--|--|
| 1 | 2 | 3 |

</td><td>

| A | B | C |
|--|--|--|
| 1 | 2 | 3 |

</td></tr> </table>
```

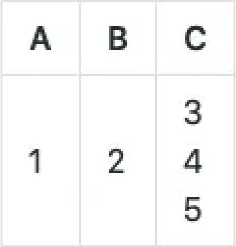


<table>
<tr><th>Heading 1</th><th>Heading 2</th></tr>
<tr>
<td>
<table>
<tr><th>A</th><th>B</th><th>C</th></tr>
<tr><td>1</td><td>2</td><td>3</td></tr>
</table>
</td>
<td>
<table>
<tr><th>A</th><th>B</th><th>C</th></tr>
<tr><td>1</td><td>2</td><td>3</td></tr>
</table>
</td>
</tr>
</table>

Display two tables side by side

Let’s create a table with multiple lines using the HTML `<br/>` tag.

```
| A | B | C |
|---|---|---|
| 1 | 2 | 3 <br/> 4 <br/> 5 |
```



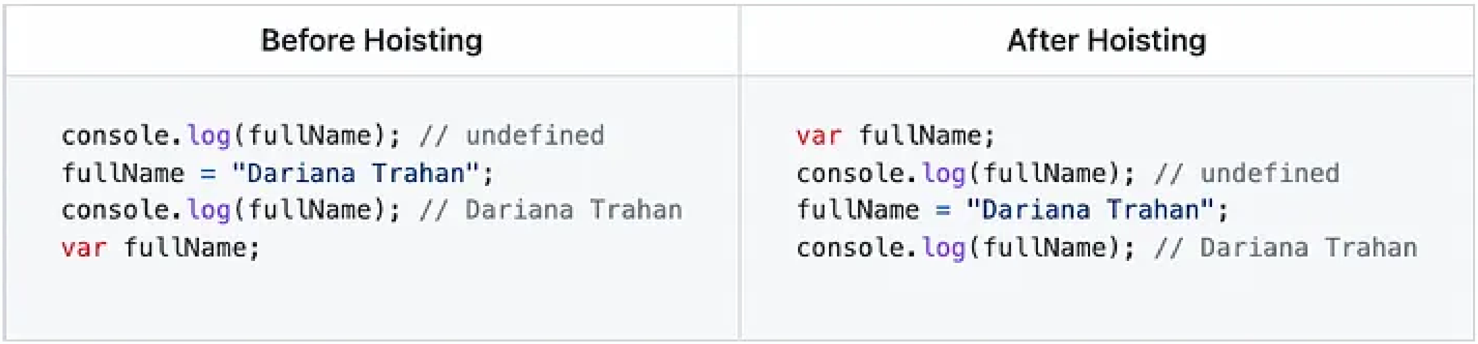
A	B	C
1	2	3 4 5

A table with multiple lines

Another example of two tables side by side.

```
<table>
<tr>
<th>Before Hoisting</th>
<th>After Hoisting</th>
</tr>
<tr>
<td>
<pre lang="js">
console.log(fullName); // undefined
fullName = "Dariana Trahan";
console.log(fullName); // Dariana Trahan
var fullName;
</pre>
</td>
<td>
<pre lang="js">
var fullName;
console.log(fullName); // undefined
fullName = "Dariana Trahan";
```

```
console.log(fullName); // Dariana Trahan
</pre>
</td>
</tr>
</table>
```



Before Hoisting	After Hoisting

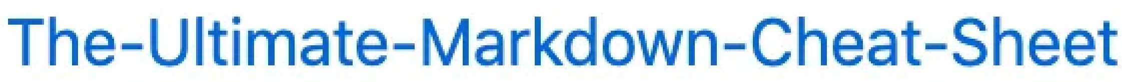
console.log(fullName); // undefined
fullName = "Dariana Trahan";
console.log(fullName); // Dariana Trahan
var fullName;
	var fullName;
	console.log(fullName); // undefined
	fullName = "Dariana Trahan";
	console.log(fullName); // Dariana Trahan

Display two tables side by side

Links

We can create a link in four ways. The first one is by using an inline style. The second one uses reference style, the third one uses relative links, and finally auto links.

```
[The-Ultimate-Markdown-Cheat-Sheet](https://github.com/lifeparticle/Markdown-Cheatsheet)
```

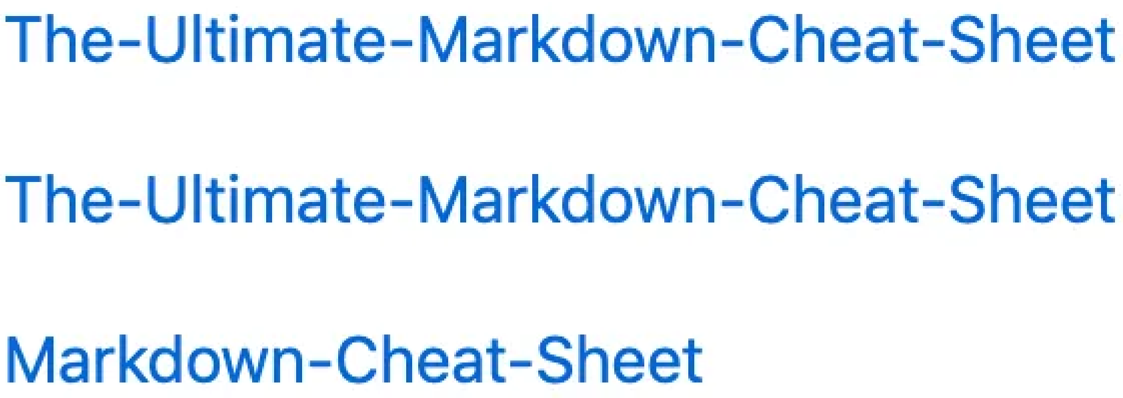


The-Ultimate-Markdown-Cheat-Sheet

Inline-style

If you’re using the same link more than once, then using the reference style would be beneficial since you don’t have to write the link every time, and also, it’s easy to update the link. Moreover, you can use numbers for the reference text. Also, you can use the reference text as the link text.

```
[The-Ultimate-Markdown-Cheat-Sheet][reference text]
[The-Ultimate-Markdown-Cheat-Sheet][1]
[Markdown-Cheat-Sheet]

[reference text]: https://github.com/lifeparticle/Markdown-Cheatsheet
[1]: https://github.com/lifeparticle/Markdown-Cheatsheet
[Markdown-Cheat-Sheet]: https://github.com/lifeparticle/Markdown-Cheatsheet
```



The-Ultimate-Markdown-Cheat-Sheet

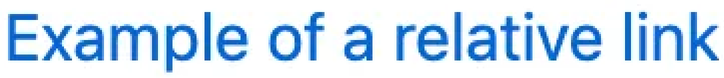
The-Ultimate-Markdown-Cheat-Sheet

Markdown-Cheat-Sheet

Three variations of reference-style

We can also create relative links with all relative link operands, such as `./` and `../`.

```
[Example of a relative link](rl.md)
```

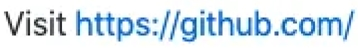


Example of a relative link

Example of a relative link

GitHub can automatically create links from standard URLs.

```
Visit https://github.com/
```



Visit https://github.com/

Auto link

Images

We can add images using similar techniques we used for links.

```
![alt text](https://images.unsplash.com/photo-1415604934674-561df9abf539?ixlib=rb-1.2.1&
```

Inline-style

```
![alt text][image]
[image]: https://images.unsplash.com/photo-1415604934674-561df9abf539?ixlib=rb-1.2.1&ixi
```

Reference-style

Also, we can use the HTML `img` tag to add an image.

```
<img src="https://images.unsplash.com/photo-1415604934674-561df9abf539?ixlib=rb-1.2.1&ix
```

img tag

We can also embed GIFs and SVG.

```
<img src="https://media.giphy.com/media/qLHzYjlA2FW8g/giphy.gif" />
```

GIF

```
<img src="https://img.shields.io/badge/theultimatemarkdowncheatsheet-brightgreen.svg" />
```

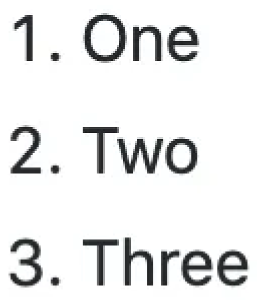
theultimatemarkdowncheatsheet

SVG

Lists

For lists, we can have ordered and unordered lists.

```
1. One
2. Two
3. Three
```

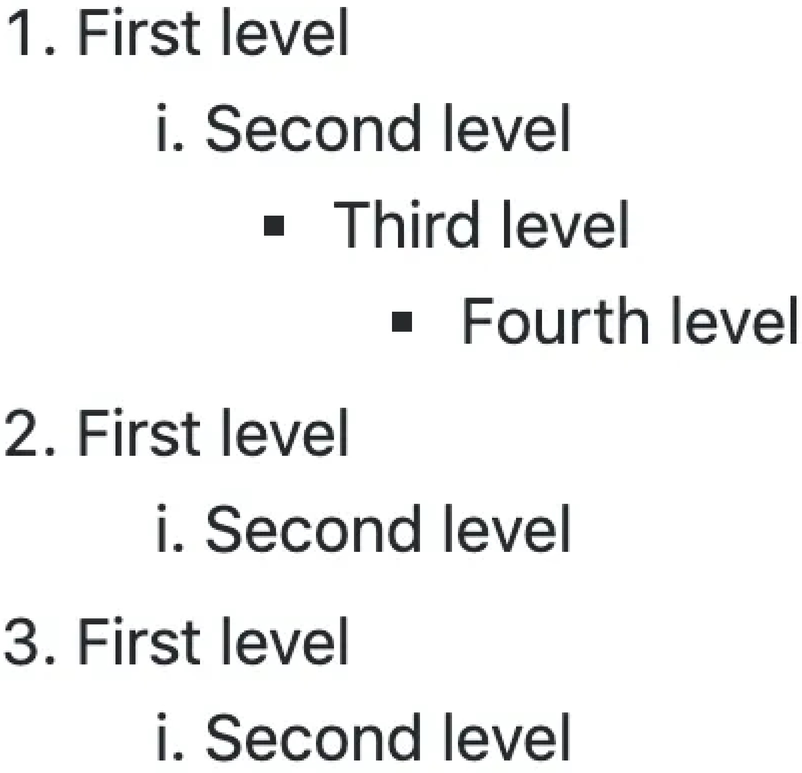


1. One
2. Two
3. Three

An ordered list

Now let’s create an ordered list with sub-items.

```
1. First level
    1. Second level
        - Third level
            - Fourth level
2. First level
    1. Second level
3. First level
    1. Second level
```

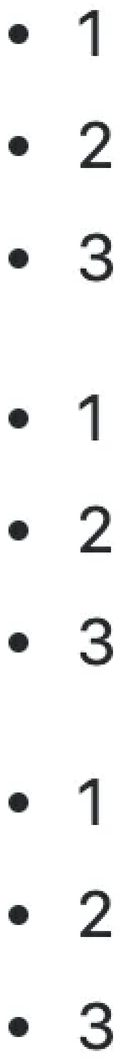


1. First level
    - i. Second level
        - ▪ Third level
            - ▪ Fourth level
2. First level
    - i. Second level
3. First level
    - i. Second level

An ordered list with sub-items

To create an unordered list, we can asterisk, plus, or minus sign.

```
* 1
* 2
* 3

+ 1
+ 2
+ 3

- 1
- 2
- 3
```



- 1
- 2
- 3

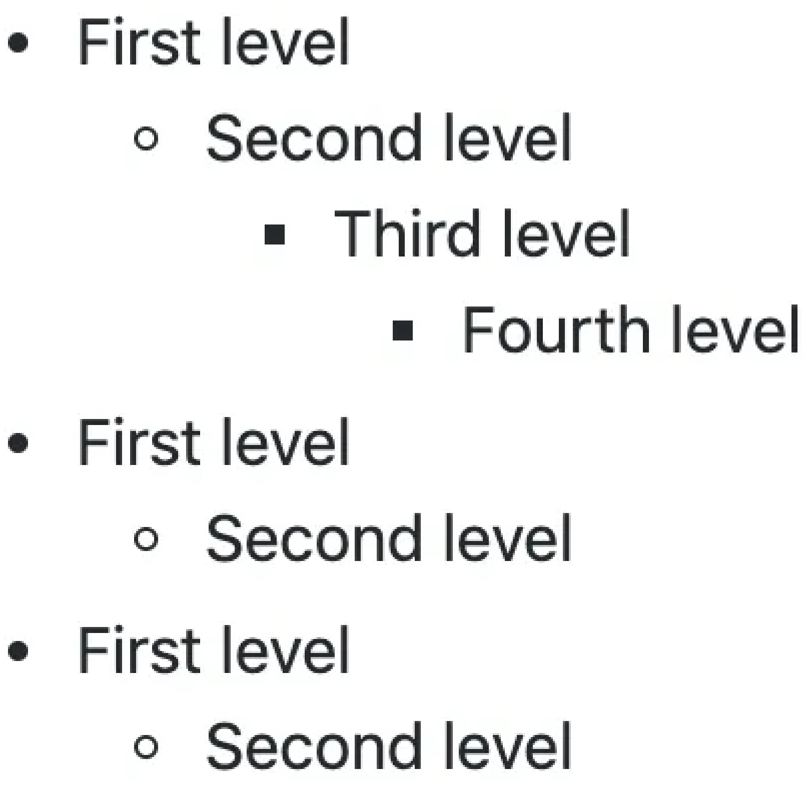
- 1
- 2
- 3

- 1
- 2
- 3

Unordered lists

Now let’s create an unordered list with sub-items.

```
- First level
    - Second level
        - Third level
            - Fourth level
- First level
    - Second level
- First level
    - Second level
```

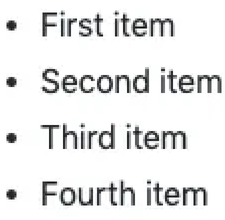


- First level
    - Second level
        - Third level
            - Fourth level
- First level
    - Second level
- First level
    - Second level

An unordered list with sub-items

We can also use HTML to create a list.

```
<ul>
<li>First item</li>
<li>Second item</li>
<li>Third item</li>
<li>Fourth item</li>
</ul>
```

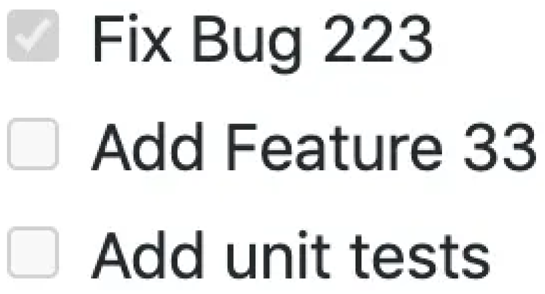


- First item
- Second item
- Third item
- Fourth item

A list using HTML

Now let’s create a task list. We can create a task list using a hyphen followed by `[ ]`, and to mark a task complete, put an `x` inside the brackets.

```
- [x] Fix Bug 223
- [ ] Add Feature 33
- [ ] Add unit tests
```

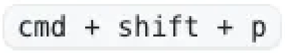

☑ Fix Bug 223
□ Add Feature 33
□ Add unit tests

Tasklist

Buttons (6 January 2023)

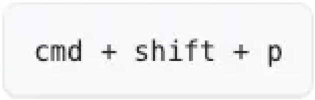
We can add buttons to describe keyboard shortcuts.

```
<kbd>cmd + shift + p</kbd>
```


cmd + shift + p

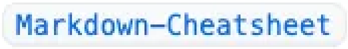
Button

```
<kbd> <br> cmd + shift + p <br> </kbd>
```


cmd + shift + p

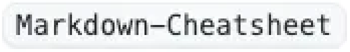
Button with space

```
<kbd>[Markdown-Cheatsheet](https://github.com/lifeparticle/Markdown-Cheatsheet)</kbd>
```


Markdown-Cheatsheet

Button with link

```
[<kbd>Markdown-Cheatsheet</kbd>](https://github.com/lifeparticle/Markdown-Cheatsheet)
```


Markdown-Cheatsheet

Button with link

Horizontal Rule

We can use three hyphens, asterisks, or underscores to create a horizontal line.

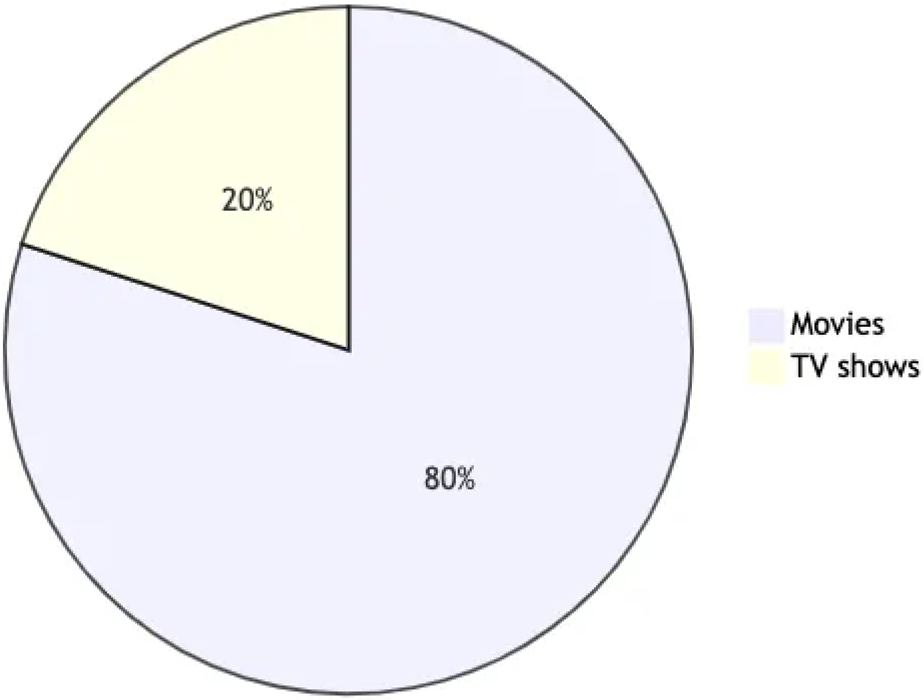
```
---
***
___
```

Horizontal Rule

Diagrams (19 July 2022)

You can now include diagrams using Mermaid. It supports diagrams like flowcharts, Sequence diagrams, Gantt charts, Class diagrams, State diagrams, Pie charts, User Journey diagrams, etc. Here is an example of a pie chart. Read more on GitHub.

~~~
```mermaid
pie
"Movies" : 80
"TV shows" : 20
```
~~~

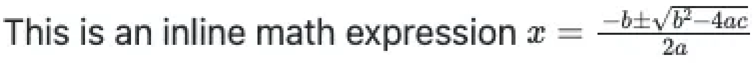


Pie chart

Mathematical expressions (19 July 2022)

You can now include diagrams using mathjax. You can read more on GitHub. We can use `$` symbol to show inline and `$$` symbols to show math expressions as a block. Here is an example of an inline math expression:

```
This is an inline math expression $x = {-b \pm \sqrt{b^2-4ac} \over 2a}$
```

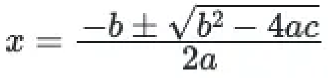


This is an inline math expression $x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$

Inline math expression

Here is an example of math expression as a block:

```
$$
x = {-b \pm \sqrt{b^2-4ac} \over 2a}
$$
```



$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Math expression as a block

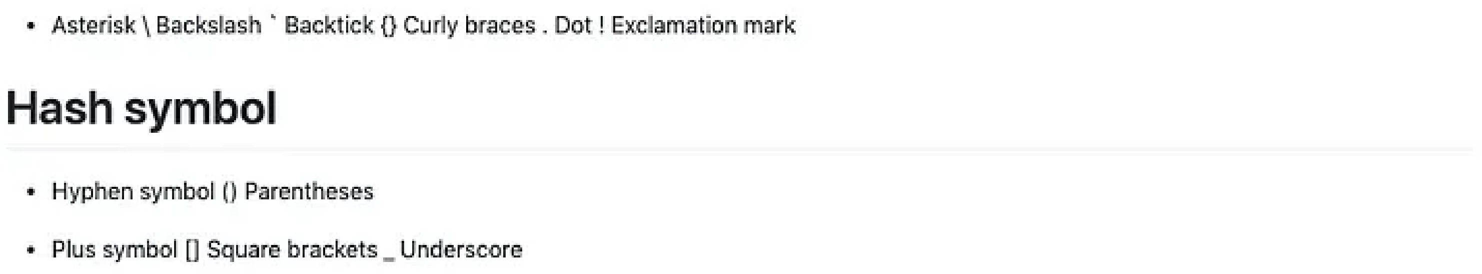
Miscellaneous

We can include comments inside a `.md` file.

```
<!--
Lorem ipsum dolor sit amet
-->
```

We can use a backslash to escape literal characters. Before escaping.

```
*   Asterisk
\   Backslash
`   Backtick
{}  Curly braces
.   Dot
!   Exclamation mark
#   Hash symbol
-   Hyphen symbol
()  Parentheses
+   Plus symbol
[]  Square brackets
_   Underscore
```



- Asterisk \ Backslash ` Backtick {} Curly braces . Dot ! Exclamation mark

Hash symbol

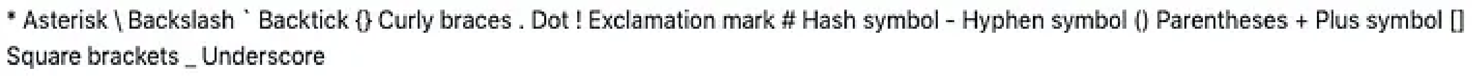
- Hyphen symbol () Parentheses
- Plus symbol [] Square brackets _ Underscore

Before escaping

After escaping.

```
\*   Asterisk
\\   Backslash
```

```
    \`   Backtick
    \{}  Curly braces
    \.   Dot
    \!   Exclamation mark
    \#   Hash symbol
    \-   Hyphen symbol
    \()  Parentheses
    \+   Plus symbol
    \[]  Square brackets
    \_   Underscore
```



* Asterisk \ Backslash ` Backtick {} Curly braces . Dot ! Exclamation mark # Hash symbol - Hyphen symbol () Parentheses + Plus symbol [] Square brackets _ Underscore

After escaping

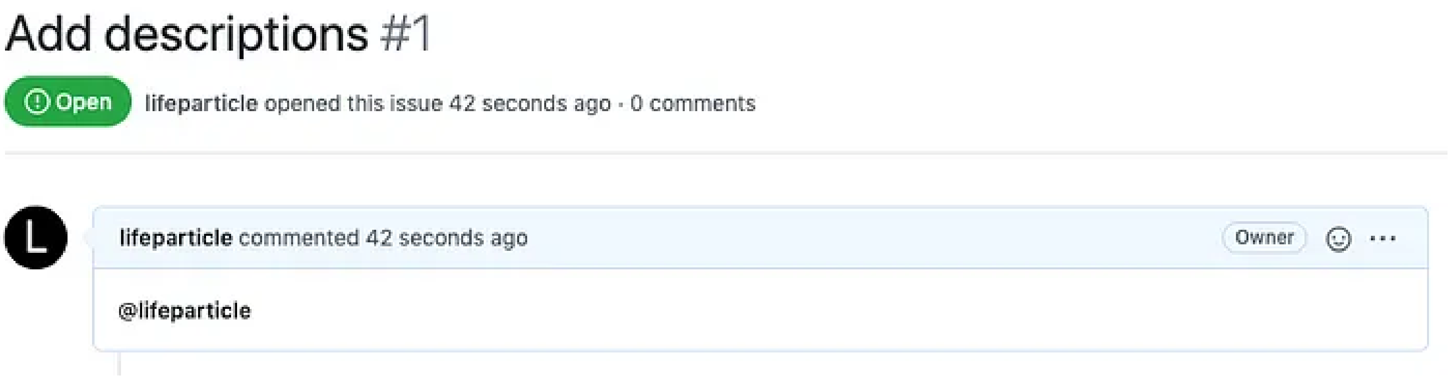
We can also include emojis in our `.md` file.

```
:octocat:
```

An octocat emoji

We can mention a person or team by typing `@` with their username or team name.

```
@lifeparticle
```



Add descriptions #1

lifeparticle opened this issue 42 seconds ago · 0 comments

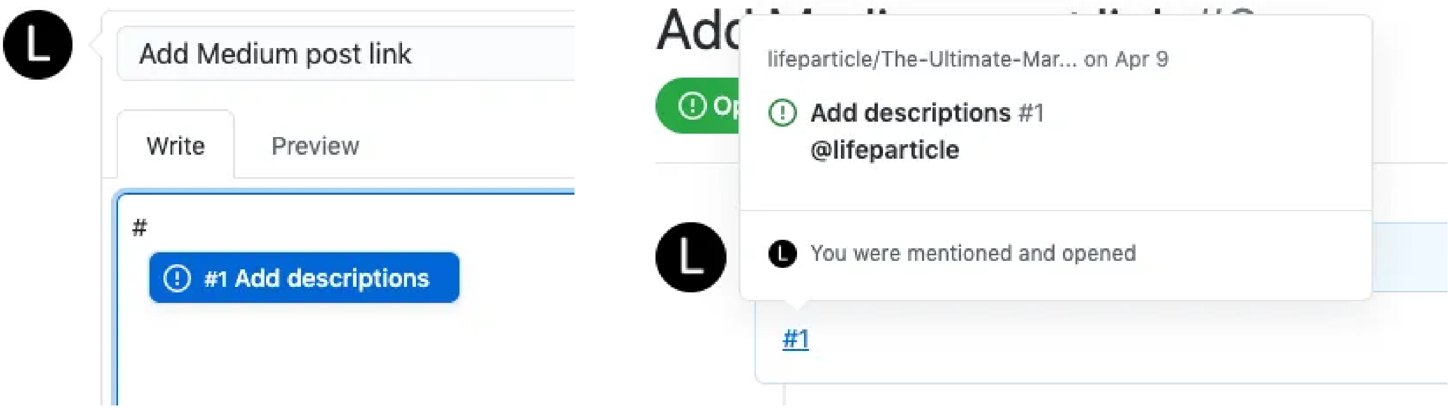
lifeparticle commented 42 seconds ago

@lifeparticle

Mention a person

We can also bring up a list of suggested issues and pull requests within the repository by typing `#`.

```
#
```



Add Medium post link

#

#1 Add descriptions

lifeparticle/The-Ultimate-Mar... on Apr 9

Add descriptions #1

@lifeparticle

You were mentioned and opened

#1

Referencing issues and pull requests

Bitbucket

Bitbucket Supported Markdown for READMEs. Also, create a table of contents.

Azure DevOps Project wiki

Azure DevOps Supported Markdown for Project wiki.

Tools

2. Create an empty Markdown table — Tablesgenerator

3. Convert Excel to Markdown table — Tableconvert

4. Markdown preview for Sublime Text 3 — Packagecontrol

5. Markdown preview Visual Studio Code — Marketplace

• • •

Congratulation! Now you know how to create an awesome README for your next project. I hope you learned something new. Now have a look at the official documentation on GitHub.

Happy coding!

Programming Software Engineering Technology Markdown Data Science

179