

Markdown Basics - MayADevBe Blog

 mayadevbe.me/posts/blog/markdown_basics

April 19, 2021

Markdown Basics

Published - 2021-04-19 | ⌚ 6min

In one of my previous posts: [Why you should know Markdown article](#), I explained what Markdown is and why it is good to know it. In this article, I want to give an overview of the most important syntax.

Here is a short definition of Markdown:

Markdown is a text-to-HTML conversion tool for web writers. Markdown allows you to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML (or HTML). ¹

I will explain the Markdown basics. I separated them into structuring, text formatting, linking and HTML integration. I will also give a hint of what you should be aware of in regards to special characters and different Markdown Flavors.

Different Markdown Flavors

Before I start explaining the basic syntax, I wanted to mention, that there are different Markdown 'flavours'. This means, depending on the application/service there might be slight changes or additions, that you have to learn. But the basic syntax should be consistent.

The Basics

1. Structure

Lets first take a look at how to structure your text documents.

Headings

Headings are an important part of every text document. In Markdown, like in HTML, there are six levels of headings.

To create a heading you start the line with a one to six **number signs (#)**. One number sign indicates a level-1 heading, two number signs indicate a level-2 header and so on. Make sure to keep a space between the number sign and your text. *Example: ###*

Level-3 Heading

Markdown also offers an alternative for the level-1 and level-2 header:

```
Header      Level-1
           ===
Header      Level-2
           ---
```

Paragraphs

Paragraphs are very simple. You only have to leave an **empty line** between the blocks.

```
      This is the first paragraph.

      This is the second paragraph,
first.  separated by one empty line between the
```

You can also fall back on the HTML version a `
` tag, to create a new line without necessarily creating a new paragraph.

Line Break

A line break is done by pressing the enter/return key. However, depending on the application you might want to make sure to make two spaces before you press the return key.

Horizontal Rule

To better separate content visually, it is possible to create horizontal rules. This can be done with three or more **dashes**(-), **underscores**(_) or **asterisks**(*). *Example:*

or:

or:

—

Lists

Markdown offers a syntax for ordered and unordered lists.

```
1. first item in the ordered
list
2. second item in the ordered
list

- first item in the unordered
list
- second item in the unordered
list
```

Some things to note:

- Ordered lists do not have to start with the number one
- Unordered lists could also use the **plus sign**(+) or **asterisk**(*)
- Both lists need to have a **space** before the text

Blockquotes

If you want to include quotes and definitions into your text, then you can use blockquotes. Simply start the line with a **greater-than** symbol (>). *Example:*

```
| This is a blockquote.
```

```
> This is a
blockquote.
```

Codeblock

And for all the coders out there, it is possible to create a code block. You need to frame your code with three **grave accents** (`) at the beginning and the end. *Example:*

```
    ```\n        This is the syntax for a code block in a code\nblock.    ```\n
```

It is also possible to define the programming language of your code to get syntax highlighting (depending on the application). This is done by writing the language (or a short form) right after the first three grave accents (without space). *Example:*

```
    ```py\n        print("Hello\nWorld!")\n    ```\n
```

(with `py` being the short form for 'Python')

2. Text Formatting

Now, let's take a look at some ways to make text stand out.

Bold

You can write part of your text in **bold**, by framing the text with two **asterisks**(*) or **underscores**(). *Example:* `**bold text**` or `__bold text__`

Italic

You can write part of your text in *italic*, by framing the text with one **asterisk**(*) or **underscore**(). *Example:* `*italic text*` or `_italic text_`

Bold and Italic

You can write part of your text in ***bold and italic***, by framing the text with three **asterisk**(*) or **underscore**().

Strikethrough

You can ~~strike through text~~ by framing the text with two **tilde**(~).

Inline Code

Finally, if you do not want to write code in a block, or only want to reference a `variable`, you just need to frame the code in one **grave accent** (`). This could also be used to emphasize a phrase, next to using bold or italics. It is not possible to specify the programming language though.

3. Linking

External Link

A lot of applications automatically recognize a link or email. *Example:* <https://mayadevbe.me/> However, it is also possible to connect a link to words. In markdown it would look like this: `[connect a link to words]`
(<https://mayadevbe.me/>)

Internal Link

It is also possible to use the link structure from external links to link internal documents or parts of a document. Just switch the link with the path to the file to link another document. Other than a document you can link headings inside a document to refer to another part. In Markdown, it would look like this: `[headings inside a document]`
([Markdown%20Basics%20Article#The%20Basics](#)) Notice two things:

- The Heading is indicated by a **number sign** (#).
- The **spaces** are replaced by `%20`.

Link Images

Linking images is very similar to the already introduced links: `![Alt Text]`
(<https://upload.wikimedia.org/wikipedia/commons/thumb/4/48/Markdown-mark.svg/1200px-Markdown-mark.svg.png> "Hover Info") Notice three things:

- The link starts with an **exclamation sign** (!).
- The text in the **square brackets**([]) is the alt text (the text that is shown if the image can not be loaded).
- After the address/local path a text in **quotation marks**(") can be added. This text will be shown when you hover over the image.

Footnotes

It is possible to create footnotes², that link to a note at the end of the file. The footnote in the text will look like this: `[^1]`, with 1 being a unique identifier (it could also be a letter, word, etc.). The content of the footnote is placed at the end of the file. It looks like the following in Markdown:

`[^1]: Example of a footnote`

4. HTML

Since Markdown was designed to be converted into HTML, it also allows the inclusion of HTML snippets. How good this functionality works, might also depend on your application.

Escape Special Characters

In the syntax explanation, you can see a few characters with a special meaning in Markdown. For example #, *, [,], < and more. If you want to display these symbols as plain text you need to 'escape' them. Meaning, you need to put a backslash(\) before the characters. *Example:* \#, *, \[, \], \< and more If your application should ever show weird test or even miss some, it might be, because you did not escape a special character.

Conclusion

I have explained the basic syntax of Markdown so that you should be able to write Markdown. This is, however, not everything you can learn about Markdown. If you want to learn more about the syntax of Markdown, then check out [The Markdown Guide](#).

1. Markdown <https://daringfireball.net/projects/markdown/> ↵

2. Example of a footnote ↵

Comments:

© 2021-24 MayADevBe