InterviewBit
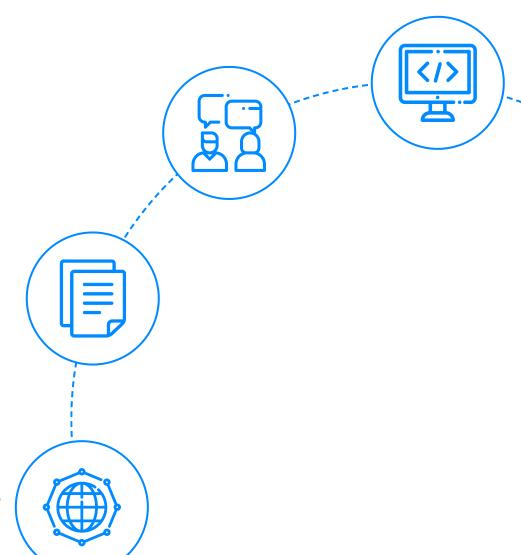
Markdown Cheat Sheet

To view the live version of the page, click here.

Contents

Markdown Tutorial: Basics to Advanced Concepts

Let's get Started

What is Markdown?

Markdown is a way of writing and formatting rich text using plain text formatting syntax. Basically, it is a lightweight markup language, using which we can create richly formatted text. The main intention behind its invention was to provide users with a markup language, that would appeal to its readers, even in its source code form, and thus make text formatting easier.

In this cheat sheet, we will discuss some of the major features of markdown along with the code syntaxes and examples which will allow the reader to make richly-formatted text articles by themselves.

Markdown Tutorial: Basics to Advanced Concepts

1. Markdown Files

A text file created with several syntaxes of a Markdown language is called a Markdown File. It makes use of plain text formatting consisting of inline text symbols for specifying how a text can be formatted.

How to open a markdown file offline?

Markdown files are of the extension "*.md" where * represents the file name. They can be opened offline using certain Chrome plugins like the "Markdown Preview Plus". Visual Studio Code by Microsoft also allows the user to toggle between Markdown code and its preview. Or you can simply open it using Notepad/Notepad++ or WordPad software.

Online Markdown Editors:

There are many online Markdown editors in the market currently, which have soared in popularity due to their easy use case and a wide variety of features available. Some of the most popular ones are HackMD and StackEdit, which provide many features along with basic markdown, to render richly formatted content.

Advantages of Markdown Files:

Some of the advantages of Markdown are stated below:

- Richly Formatted content can be created relatively quickly.
- It has become the go-to language for creating documentation, because of its lightweight and platform-transferrable nature. Even Github readme files are created using Markdown, allowing them to be attractive to read(due to formatting) as well as lightweight in nature.

2. Markdown Headings

Headings tell us about the title or subject of a piece of work. In markdown, there are 2 main ways to write headings both of which are discussed below:

- **Headings using #:**

We can write headings in markdown using the # symbol, followed by the heading that we want to write. The number of '#' symbols indicate the size of the header. More the number of '#' symbols, smaller the font size of the header. There are a total of 6 types of headers available in markdown.

Example:

```
# Heading1
## Heading2
### Heading3
#### Heading4
##### Heading5
###### Heading6
```

Output:

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

- **Using HTML syntax:**

We can also write headers using HTML formats using <hi></hi> (where i varies from 1 to 6, and it represents the size of the headers, from large to small) tags.

Example:

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```

Output:

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

3. Text Styles

We can use markdown to change the styles of selected text to different styles like bold, italics, blockquotes, underline, strike-through, etc.

- **Markdown Bold:**
  - We can make a text bold by enclosing the text between 2 * symbols as -> **text**.
  - We can also make it bold using the HTML "strong" tags.
  - For example:

```
Method 1:
*Hello*
Method 2:
<em>Hello</em>
```

Output:

Hello
Hello

- **Markdown Italics:**
  - We can make a text Italics by enclosing the text in single * symbol as -> *text*.
  - We can also make it in Italics using the HTML "em" tags.
  - For example:

```
Method 1:
*Hello*
Method 2:
<em>Hello</em>
```

Output:

Hello
Hello

- **Markdown Blockquotes:**
  - Blockquotes are used to indicate that the enclosed text is an extended quotation. Blockquotes are written in markdown using > symbol. Multiple > can stacked together to get nested blockquote.
  - Example:

```
> This is a blockquote.
>> Nested blockquotes.
>>> More nesting.
```

Output:

> This is a blockquote.
>> Nested blockquotes.
>>> More nesting.

- **Markdown Underlined:**
  - We can underline a piece of text in Markdown using the HTML "ins" tag.
  - Example:

```
<ins>This is an underlined text</ins>
```

Output:

This is an underlined text

- **Strike-through:**
  - We can strike-through a piece of text in Markdown by enclosing the text inside double ~ symbol as -> ~~text~~.
  - Example:

```
~~The text is struck~~
```

Output:

~~The text is struck~~

- **Markdown Boxed:**
  - "Boxed" allows us to form a box around our given piece of text to highlight its importance. It can be formed with using the "table", "tr", and "td" tags in conjunction, similar to using HTML tables.
  - Example:

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

Output:

The quick brown fox jumps over the lazy dog.

- **Markdown Superscript and Subscript:**
  - Superscript or Subscript are characters that are set slighly above or below the normal line of text. We can set them using the "sup" or the "sub" HTML tags respectively.
  - Example:

```
Subscript <sub>Example.</sub>
Superscript <sup>Example.</sup>
```

Output:

Subscript $_{Example.}$

Superscript $^{Example.}$

4. Syntax Highlighting

We can highlight code syntaxes using markdown by using a backtick(` `` `) symbol before and after the code block. For multiline code formatting, we can also use 3 backticks for the purpose as illustrated in the example below.

- **Example for single line code:**

This is an example of a `code` block.

Output:

This is an example of a `code` block.

- **Example for multiline code:**
```

This is an
example of
multiline code block.
```

Output:

```
This is an
example of
multiline code block.
```

Code Highlighting:

- We can highlight code snippets using the 3 backticks:
- Syntax:

```
int x = 10;
cout << x << endl;
```

Output:

```
int x = 10;
cout << x << endl;
```

Code Highlighting with Language specified:

- We can highlight code snippets for a specific language too, along with syntax highlighting by specifying the language name along with the 3 backticks. By doing this, the data types and variables will be highlighted in different colours for differentiating between them and for better visibility.
- Example Syntax for C++:

```C++
int x = 10;
cout << x << endl;
```

Output:

```
int x = 10;
cout << x << endl;
```

5. Markdown Tables

- We can create a markdown table with the HTML <table></table> tags without headers.
  - **Example**

```
<table>
    <tr>
        <td>
            First Entry
        </td>
        <td>
            Second Entry
        </td>
        <td>
            Third Entry
        </td>
    </tr>
</table>
```

Output:

| First Entry | Second Entry | Third Entry |

- We can align the values of the columns of the table left, right and centre using :--: symbols.
  - **Example:**

In the example, the text in the columns will get aligned to the side where we put the : symbol. If the : is put on both sides, the text will be centre aligned. No alignment appears the same as left align.

```
| No Align | Right Align | Left Align | Center Align |
| -------- | ----------:|:--------- |:------------:|
| 1        |           1 | 1         |      1       |
| 11       |          11 | 11        |      11      |
| 111      |         111 | 111       |     111      |
| 1111     |        1111 | 1111      |     1111     |
```

Output:

No Align	Right Align	Left Align	Center Align
1	1	1	1
11	11	11	11
111	111	111	111
1111	1111	1111	1111

- We can also build a table with multiple lines using the HTML <br> tag. It is to be noted that the <br> tag can also be used outside of tables for moving to a new line.
    - **Example:**

In this example, the use of the <br> tag is shown to build a table with multiple lines in a column.

```
| Column 1 | Column 2 |   Column 3      |
|:--------:|:--------:|:---------------:|
|    A     |    B     | C <br> D <br> E |
```

Output:

Column 1	Column 2	Column 3
A	B	C D E

6. Links

There are 4 ways of creating a link in markdown:

- **Inline Link:**
  - Inline Links are used to link users to another page, displayed in the form of blue hyperlinked words.
  - For example: Here the Link1 link points to our practice page of InterviewBit.

```
[Link1] (https://www.interviewbit.com/practice/)
```

Output:

[Link1](#)

- **Reference Link:**
  - Reference Links are used to link one information object to another, by providing some references to that new page.
  - The main difference between Inline and Reference Links is that in Inline links, the links appear directly at the point where they are placed, whereas in reference links, they used to cite references to other webpages from the current page.
  - For example:

```
[Link1][reference text]
[reference text]: https://www.interviewbit.com/practice/
```

Output:

Link1

- **Relative Link:**
    - Relative Links are links that show the relationship between the current page's URL and the linked page's URL.

```
[A relative Link](rl.md)
```

Output:

A relative link

- **Auto Link:**
    - Many popular markdown processors directly turn URLs pasted into the editor into links. Such links are called auto links.
    - For example:

```
Visit https://www.interviewbit.com/practice/
```

Output:

Visit https://www.interviewbit.com/practice/

7. Images and GIFs

Images can also be added in markdown using methods similar to what we used for adding links.

- **Inline Style:**
    - Inline Images are used to link images directly onto the current page to be displayed.
    - Syntax:

```
![alt text](https://d3n0h9tb65y8q.cloudfront.net/public_assets/assets/000/002/559/origi
```

Output:

- **Reference Style:**
  - The reference style of image insertion is used to link an image to the current page, similar to a reference link.
  - Syntax:

```
![alt text][image]
[image]: https://d3n0h9tb65y8q.cloudfront.net/public_assets/assets/000/002/561/original
```

Output:

- **HTML `<img>` Style:**
  - The HTML <img> tag can also be used to add images in Markdown.
  - Syntax:

```
<img src="https://d3n0h9tb65y8q.cloudfront.net/public_assets/assets/000/002/562/origina
```

Output:

- **Adding GIFs:**
  - GIFs can also be added using Markdown similar to the way we add images in markdown using the HTML <img> tag.
  - Syntax:

```
<img src="https://d3n0h9tb65y8q.cloudfront.net/public_assets/assets/000/002/564/origina
```

Output:

8. Lists

There are 2 types of lists we can add in markdown,

1. Ordered Lists:

- An ordered list defines a list of items in which the order of the items matter.
- Syntax:

```
1. Element 1
2. Element 2
3. Element 3
```

Output:

1. Element 1
2. Element 2
3. Element 3

- **Ordered List with Sublists:**
  - Lists can be nested into sublists to include items that are a part of a longer list. The example below shows how to implement nested unordered lists in markdown.
  - Syntax:

```
1. Level 1
    1. Level 2
      - Level 3
          - Level 4
2. Level 1
    1. Level 2
3. Level 1
    1. Level 2
```

Output:

1. Level 1
    1. Level 2
        - Level 3
        - Level 4
2. Level 1
    1. Level 2
3. Level 1
    1. Level 2

2. Unordered Lists:

We can create unordered lists with an asterisk(*), (+), or (-) signs as shown below:

- **With asterisk(*):**

```
* Element 1
* Element 2
* Element 3
```

Output:

- Element 1
- Element 2
- Element 3

- **With plus(+):**

```
+ Element 1
+ Element 2
+ Element 3
```

Output:

- Element 1
- Element 2
- Element 3

- **With minus(-):**

```
- Element 1
- Element 2
- Element 3
```

Output:

- Element 1
- Element 2
- Element 3

3. Nested Unordered Lists:

Unordered Lists can also be nested into multiple sublists beneath them similar to ordered lists. The example below shows how to implement nested unordered lists in markdown.

Syntax:

```
- Level 1
    - Level 2
        - Level 3
            - Level 4
- Level 1
    - Level 2
- Level 1
    - Level 2
```

Output:

- Level 1
    - Level 2
        - Level 3
            - Level 4
- Level 1
    - Level 2
- Level 1
    - Level 2

List using HTML:

We can create lists in the following way by using the HTML tags as shown below:

- **Ordered List:**
    - We can also create ordered lists in markdown using the HTML `<ol></ol>` tags. The `<ol></ol>` tags denote the start and end of the lists. The items in the list are added using the `<li></li>` tags.
    - **Syntax:**

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

Output:

1. First item
2. Second item
3. Third item
4. Fourth item

- **Unordered List:**
    - We can also create unordered lists in markdown using the HTML tags. The tags denote the start and end of the lists. The items in the list are added using the tags.
    - **Syntax:**

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

Output:

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

Creating Task Lists:

We can also create task lists in markdown using a hyphen(-) followed by [ ] braces. An x is put inside the braces to indicate the task is complete.

Example:

```
- [x] Buy Eggs.
- [ ] Buy Milk.
- [ ] Wash Clothes.
```

☑ Buy Eggs
☐ Buy Milk
☐ Buy Books

9. Horizontal Rule

- A horizontal rule (<hr>) is a line that goes across a webpage. We can create a horizontal rule using three hyphens(-), an asterisk(*) or underscores.
- Example:

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

Output:

10. Comments

We can add markdown comments by enclosing them within <!-- --> symbols.

Example:

<!-- This is a comment. -->

Output:

```
<!-- This is a comment. -->
```

11. Escape Characters

We can use a backslash(\) to escape literal characters.

Before escaping they will look like,

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

Output:

This is the output that will be shown, if we just type the special characters directly, without using their escape sequences, we get the below output:

After escaping by using the \ symbol as:

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

Output:

This is the output that will be shown if we type the special characters along with their escape sequences.

* Asterisk

\ Backslash

` Backtick

{} Curly braces

. Dot

! Exclamation mark

Hash symbol

- Hyphen symbol

() Parentheses

+ Plus symbol

[] Square brackets

_ Underscore

InterviewBit

Conclusion:

From the above cheat sheet, we can observe that using markdown we can format even a simple text document into rich, highly engaging content. Markdown is used in a lot of domains including ed-tech and education fields to create highly engaging and high-quality content for the consumers at a fast pace. With its rich list of available features, it can be used innovatively in a wide variety of ways and purposes.

Additional Useful Resources:

- [Bitbucket vs GitHub](#)
- [HTML Interview Questions](#)
- [Git Interview Questions](#)

Links to More Interview Questions

C Interview Questions

Php Interview Questions

C Sharp Interview Questions

Web Api Interview Questions

Hibernate Interview Questions

Node Js Interview Questions

Cpp Interview Questions

Oops Interview Questions

Devops Interview Questions

Machine Learning Interview Questions

Docker Interview Questions

Mysql Interview Questions

Css Interview Questions

Laravel Interview Questions

Asp Net Interview Questions

Django Interview Questions

Dot Net Interview Questions

Kubernetes Interview Questions

Operating System Interview Questions

React Native Interview Questions

Aws Interview Questions

Git Interview Questions

Java 8 Interview Questions

Mongodb Interview Questions

Dbms Interview Questions

Spring Boot Interview Questions

Power Bi Interview Questions

Pl Sql Interview Questions

Tableau Interview Questions

Linux Interview Questions

Ansible Interview Questions

Java Interview Questions

Jenkins Interview Questions