Opensource.com: Markdown Cheat Sheet BY MATT BROBERG

Markdown is the most widely adopted way to format documents using plain text syntax. Here are the essentials.

STYLIZE TEXT

italic	*italic*
bold	**bold**
~~strikethrough~~	~~strikethrough~~
code in-line	`code in-line`
code block	```code block```
> text	blockquote

HEADERS AND BREAKS

# Text	H1
## Text	H2
### Text	H3
---	Horizontal Rule

LISTS

Lists can be unordered or ordered, which is determined by the symbol used to list them. Unordered lists have either `-`, `+`, or `*`, while ordered lists require any integer followed by a `.` or `)` character.

An ordered list:

```
1. First item
2. Second item
3. Third item
```

Unordered List:

```
- First item
- Second item
- Third item
```

Combining the two:

```
1. First ordered list item
1. Second ordered list item (which
   shows as 2.)
1. (Markdown ignores the written
   number in favor of enumerating
   by integer)
    - Indented lists require four
      spaces in most specifications
    - Sublists are implementation
      specific
    - Try it out and render to be
      sure it works
1. Continuing the list after an
   indentation
```

TABLES

Tables are a common extension of the official markdown definition (CommonMark). They can be formatted in quite a lot of ways:

```
| Column Title       | Another Column                  | One More |
|:-------------------|:----------|:---------|
| **Bolded text**    | Columns don't need to be aligned  |          |
| Or even filled in |                                  |   code   |
```

Renders as:

Column Title	Another Column	One More
bolded text	Columns don't need to be aligned	
Or even filled in		`code`

That can be painful to look at. Here's a prettier example with alignment based on the placement of the `:` in the table structure:

```
| Tables   |      Are      | Cool |
|:---------|:-------------:|-----:|
| col 1 is | left-aligned  |   $1 |
| col 2 is |   centered    |   $2 |
| col 3 is | right-aligned |   $3 |
```

Tables	Are	Cool
col 1 is	left-aligned	$1
col 2 is	centered	$2
col 3 is	right-aligned	$3

Tables are always a little tricky to remember. Use available online tools to help build them, like: https://csvtomd.com

LINKS

There are two ways to create links. The most commonly used format includes brackets followed by parentheses: `[]()`. To render a link to an image, begin with a `!`.

Examples.

```
[Link inline this way](https://opensource.com)
[Or add a title for the link](https://opensource.com "Google's Homepage")
[Here is a relative link within a repository](../blob/master/LICENSE)
![I link to an image](path/to/image.png)
```

The second format involves brackets, `[][]` or `[]`, followed by a reference formatted with a bracket and colon, `[]:`, anywhere else in the document.

Examples.

```
[Reference-style link][Case-insensitive Reference Text]
[Numbers are commonly used][1]
[Text can link on its own as a reference]
[case-insensitive reference text]: https://opensource.com
[1]: https://opensource.com
[text can link on its own as a reference]: https://opensource.com
```

URLs, in and out of angle brackets, will automatically get turned into links with most implementations.

Example.

```
Both https://opensource.com and <https://opensource.com> will render as
links, as well as opensource.com on some renderings.
```


BEST OF GITHUB FLAVORED MARKDOWN

Task lists are fantastic usage of the GitHub-specific implementation:

```
Task List
- [x] Step one is complete
- [ ] Step two in this unordered list is
- [x] Step three is done as well
```

Renders as:

Task List

- [x] Step one is complete
- [ ] Step two in this unordered list is
- [x] Step three is done as well

Drop-downs are an incredible feature to tidy up files:

```
<details>
  <summary>Q1: What is the best website in
     the world? </summary>
   A1: Opensource.com
</details>
```

Renders as a clickable drop-down menu. See the example at github.com/opensourceway/markdown-example

You can also have language-specific syntax highlighting. Instead of having a code block of black-and-white text, append the language to the first set of backticks to have highlighting enabled:

```
<html>
<head>
  <meta content="text/html;charset=utf-8"
   http-equiv="Content-Type" />
</head>
<body>
  <script src='./pkg/my_wasm_library.js'></script>
  <script>
    window.addEventListener('load', async () => {
      // Load the wasm file
      await wasm_bindgen('./pkg/my_wasm_library_bg.wasm');
      // Once it's loaded the `wasm_bindgen` object is
      // populated with the functions defined in our Rust code
      const greeting = wasm_bindgen.excited_greeting("Matt")
      console.log(greeting)
    });
  </script>
</body>
</html>
```

Nearly all programming languages are supported using this syntax (python, ruby, go, rust, javascript, and java to name a few). See GitHub's documentation for the full list: https://help.github.com/en/articles/creating-and-highlighting-code-blocks#syntax-highlighting

GITLAB SPECIFIC REFERENCES

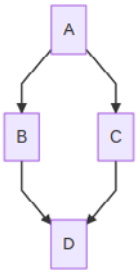
GitLab, the second largest Git-based repository on the Internet, has unique global references designed for teamwork.

@user_name	specific user
@group_name	specific group
@all	entire team
#123	issue
!123	merge request
$123	snippet
~123	label by ID
~bug	one-word label by name
9ba12248	specific commit
9ba12248...b19a04f5	commit range comparison
`[README](doc/Readme)`	repository file references
`/tableflip <comment>`	Quick reaction that includes (ノ°□°)ノ ⌒ ┻┻

You can also design flow diagrams:

```
mermaid
graph TD;
  A-->B;
  A-->C;
  B-->D;
  C-->D;
```

Becomes:



THE BEST OF BOTH GITHUB AND GITLAB

Emojis bring both formats together. Use everything from `:abc:` to `:zap:` to add emojis to your markdown 🎉.

A searchable list of emoji icons is available at https://www.webfx.com/tools/emoji-cheat-sheet/

REFERENCES

https://commonmark.org/

https://spec.commonmark.org/0.28/

https://github.github.com/gfm/

https://docs.gitlab.com/ee/user/markdown.html