Markdown Cheatsheet

Markdown Cheatsheet

A quick reference for Markdown and GitHub Flavored Markdown (GFM).

Text Formatting

```
**bold** or __bold__
*italic* or _italic_
~~strikethrough~~
`inline code`
***bold and italic***
```

bold or **bold** *italic* or *italic* ~~strikethrough~~ `inline code` ***bold and italic***

Headings

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

Links & Images

```
[link text](https://example.com)
[link with title](https://example.com "Title")
[relative link](../other-file.md)

![alt text](image.png)
![alt text](image.png "Image title")
![linked image](https://example.com/image.png)
```

link text 

Blockquotes

```
> Single line quote
>
> > Nested quote
>
> **Formatted** text inside a quote
```

> Single line quote
>
> > Nested quote
>
> **Formatted** text inside a quote

Lists

Unordered

```
- Item
- Item
  - Nested item
  - Nested item
    - Deeply nested

* Also works with asterisks
+ And plus signs
```

- Item
- Item
  - Nested item
  - Nested item
    - Deeply nested

Ordered

```
1. First
2. Second
3. Third
   1. Nested (indented)
   2. Still nested
```

1. First
2. Second
3. Third
    1. Nested (indented)
    2. Still nested

Mixed

```
1. Step one
   - Sub-point
   - Another sub-point
2. Step two
   > A note about step two
3. Step three
```

1. Step one
    - Sub-point
    - Another sub-point
2. Step two

    > A note about step two

3. Step three

Task Lists (GFM)

```
- [x] Completed task
- [x] Another completed task
- [ ] Pending task
- [ ] Blocked task
```

- [x] Completed task
- [x] Another completed task
- [ ] Pending task
- [ ] Blocked task

Code Blocks

Fenced with Syntax Highlighting

````
```python
def greet(name: str) -> str:
    return f"Hello, {name}!"

print(greet("World"))
```
````

```python
def greet(name: str) -> str:
    return f"Hello, {name}!"

print(greet("World"))
```

Inline Code

```
Use `npm install` to install dependencies.
```

Use `npm install` to install dependencies.

Common Language Identifiers

Language	Identifier
JavaScript	`js` or `javascript`
TypeScript	`ts` or `typescript`
Python	`python` or `py`
Bash	`bash` or `sh`
YAML	`yaml` or `yml`
JSON	`json`
SQL	`sql`
CSS	`css`
HTML	`html`
Dockerfile	`dockerfile`

Tables (GFM)

```
| Header 1 | Header 2 | Header 3 |
| ---------- | ---------- | ---------- |
| Cell 1 | Cell 2 | Cell 3 |
| Cell 4 | Cell 5 | Cell 6 |

| Left-aligned | Center-aligned | Right-aligned |
| :------------ | :--------------: | --------------: |
| Left | Center | Right |
| L | C | R |
```

Header 1	Header 2	Header 3
Cell 1	Cell 2	Cell 3
Cell 4	Cell 5	Cell 6

Left-aligned	Center-aligned	Right-aligned
Left	Center	Right
L	C	R

Horizontal Rule

```
---

***


___
```

Escaping

```
\*not italic\*
\# not a heading
\[not a link\]
\`not code\`
```

not italic

Keyboard Keys (GFM)

```
Press Ctrl + C to copy.
Press Alt + F4 to close.
```

Press Ctrl + C to copy. Press Alt + F4 to close.

URLs & Autolinks

```
<https://example.com>
<user@example.com>
```

Emojis (GFM)

```
:thumbsup: :rocket: :warning: :check_mark:
```

:thumbsup: :rocket: :warning: :check_mark:

Math (where supported)

```
Inline: $E = mc^2$

Block:
$$
\sum_{i=1}^{n} i = \frac{n(n+1)}{2}
$$
```

Inline: $E = mc^2$

Block: $$ \sum_{i=1}^{n} i = \frac{n(n+1)}{2} $$

Mermaid Diagrams (where supported)

```
graph LR
    A[Start] --> B{Decision}
B --> | Yes | C[Action]
B --> | No | D[End]
    C --> D
```

HTML in Markdown

Most Markdown processors allow inline HTML. Details, superscript, subscript, line breaks, and centered divs work in compatible processors.