

---

description: "Markdown style guide" applyTo: "\*\*/\*.md"

# Markdown Style Guide

## Introduction

This consolidated Markdown style guide combines our existing rules with widely-used best practices and examples from the Markdown reference material. It covers basic syntax, extended features, compatibility notes, and a few safe "hacks" when HTML support is available.

Use this guide when authoring documentation, READMEs, and other Markdown content in the repository. When in doubt, prefer CommonMark/GitHub Flavored Markdown (GFM) compatible constructs for the best cross-tool behavior.

## Headings

Use ATX-style headings (hash marks) and put a single space after the hashes. Start documents with `#` for the main title and do not skip levels (e.g., don't jump from `##` to `####`). Add a blank line before and after headings for better compatibility with Markdown processors.

Rules:

- Use one space after `#` (e.g., `## Section title`).
- Use sentence case for headings.
- Keep heading depth meaningful and avoid skipping levels.

Good:

```
# Project Phoenix
## Overview
### Requirements

```

Avoid:

```
-#MissingSpace

```

## Paragraphs and Line Breaks

Paragraphs are separated by one blank line. Avoid indenting normal paragraphs with spaces or tabs (unless intentionally creating a code block).

To create a line break (soft break), prefer an explicit `<br>` tag or use two trailing spaces at the end of a line for compatibility; note that trailing spaces are easy to miss in source.

Rules:

- Use a blank line to separate paragraphs.
- Avoid leading spaces/tabs on paragraph lines.
- For visible new lines inside a paragraph: use two trailing spaces + Enter or `<br>` when supported.

## Emphasis (Bold, Italic)

Prefer asterisks for intra-word emphasis to avoid processor differences with underscores.

Rules:

- Italic: `*text*`
- Bold: `**text**`
- Bold + Italic: `***text***`

Do not rely on underscores for mid-word emphasis (e.g., use `Love**is**bold` not `Love__is__bold`).

## Lists

Use hyphens (-) for unordered lists for consistency. For ordered lists use `1.` (the renderer will number list items correctly). Indent nested list content with four spaces.

Rules:

- Unordered lists: `- item`
- Ordered lists: `1. item`
- Indent nested lists with 4 spaces.
- Don't mix list delimiters within the same list.

To keep lists readable, put a blank line before the list and between list blocks when appropriate.

## Links and Images

Use inline link syntax: `[text](https://example.com)` and provide descriptive link text. For images use the same pattern prefixed with `!` and always include alt text.

Rules:

- Links: `[label](https://example.com)`
- Images: `![alt text](path/to/image.png)`
- Avoid `click here` as link text; be descriptive instead.

If you need links to open in a new tab or to add attributes, use HTML anchors when supported (e.g., `<a href="..." target="_blank">`).

## Code

Inline code: use single backticks: ``code``. Use fenced code blocks (three backticks) for longer snippets. Specify the language for syntax highlighting when supported (e.g., `json` or `python`).

Rules:

- Inline: ``variable``
- Block: `````

```
def fn():  
    return True
```

```
- Add a language after the opening fence for highlighting: ```python ```
```

When you need backticks inside a fenced code block, you can fence with a larger number of backticks.

```
## Blockquotes
```

Use `>` to create blockquotes. Put blank lines before and after blockquotes for better compatibility. Blockquotes can contain headings, lists, and other block elements – but remember not every processor supports every combination.

```
## Horizontal Rules
```

Use three or more hyphens (`---`) for a visual section break.

```
## Extended Syntax (Tables, Footnotes, IDs, etc.)
```

Note: Extended features vary by processor. Prefer CommonMark/GFM-compatible constructs and check your target renderer.

Tables:

- Use pipes (``|``) and hyphens (``---``) to build tables. Add pipes at the ends of rows for readability.
- Align columns with colons in header separators: ``:---``, ``:---:`, ``---:``.
- Avoid complex block-level content inside table cells; if needed, use HTML.

Example:

```
```markdown
| Name | Role |
| --- | --- |
| Alice | Developer |
```
```

Fenced code blocks and syntax highlighting:

- Use triple backticks and specify a language for highlighting (````json, ````bash, etc.).

Footnotes:

- Use footnote references like ``[^1]`` and define them ``[^1]: note text`` anywhere in the document (not inside lists/tables). Footnotes are numbered in output.

Heading IDs and anchor links:

- Many processors support ``#{custom-id}`` after a heading. Use these for internal linking and ToC generation.

Definition lists, strikethrough, task lists, emoji, highlights, subscript, superscript:

- These are supported in various extended syntaxes (GFM/MultiMarkdown). Use them when your renderer supports them:

- Definition lists: ``Term\n: Definition``
- Strikethrough: ``~~text~~``
- Task lists: ``- [ ]`` and ``- [x]``
- Emoji shortcodes: ``:joy:`` (renderer-dependent)
- Highlight: ``==text==`` (not widely supported)
- Subscript: ``H~2~`` (renderer-dependent)
- Superscript: ``X^2^`` (renderer-dependent)

## Automatic URL linking

Many renderers auto-link bare URLs (e.g., ``http://example.com``). To prevent linking, mark a URL as code: `` `http://example.com` ``.

## Hacks and HTML fallbacks (Use sparingly)

If your Markdown processor allows raw HTML, some layout or styling needs can be solved with HTML. Use these hacks only when necessary and document the dependency on HTML support.

Common fallbacks:

- Centering: `<p style="text-align:center">Text</p>` or deprecated `<center>` tag.
- Color: `<span style="color:blue">text</span>` (avoid unless necessary).
- Image sizing / captions: use `` or `<figure><figcaption>` when supported.
- Comments (hidden in output): `[comment]: # (hidden note)` or `[This is a comment]: #`` (processor-dependent but widely used).
- Table cell line breaks and lists: use `<br>` or HTML lists inside table cells.

Warning: HTML tags like `<font>` and `<center>` are deprecated; prefer CSS when available.

## ## Accessibility and best practices

- Always provide `alt` text for images.
- Use meaningful link text for screen reader users.
- Keep tables simple and avoid using them for layout.
- For long documents, consider adding a Table of Contents with heading links.

## ## Quick Cheat Sheet (Common patterns)

- Heading: `# H1` / `## H2`
- Bold / Italic: `***bold**` / `*italic*` / `***bold italic***`
- Code inline: ``code``
- Code block: ````python\nprint()\n````
- Link: `[label](https://example.com)`
- Image: `![alt](image.png)`
- Table: `| col | col |\n| --- | --- |\n| a | b |``
- Task list: `- [ ] todo` / - [x] done``
- Strikethrough: `~~no longer~~`

## ## Notes on Compatibility

When in doubt follow CommonMark/GFM (GitHub Flavored Markdown) conventions. Always test documents in the target renderer (GitHub, MkDocs, VS Code preview, etc.) before publishing.

## ## References

- CommonMark: <https://commonmark.org>
- GitHub Flavored Markdown: <https://github.github.com/gfm/>
- The Markdown Guide: <https://www.markdownguide.org>

---

This file was updated by consolidating the project's Markdown training materials (basic, extended, hacks, and cheat sheet) into a single, practical style guide.

## # Markdown Style Guide

### ## Introduction

Markdown is a lightweight markup language that allows you to format plain text documents. While it's relatively simple to learn, a consistent style guide can improve readability and collaboration, especially in a team setting. This guide outlines best practices and conventions for writing clean, consistent, and maintainable Markdown.

### ### 1\. Headings

Headings are created using the hash symbol (`#`).

- \* **Rule:** Use one space between the hash and the heading text.
- \* **Rule:** Use the "atx" style (hash marks at the beginning of the line). Avoid the "Setext" style (underlined with `===` or `---`) as it's less flexible.
- \* **Rule:** Start with a Level 1 heading (`#`) for the main title of the document.
- \* **Rule:** Do not skip heading levels. For example, don't jump from an `##` heading to a `####` heading.
- \* **Recommendation:** Use sentence case for headings.

**Good:**

```
```markdown
# Main Title
## Subheading
### Another Sub-subheading
```
```

**Bad:**

```
```markdown
Main Title
=====

##Subheading
### Another Sub-subheading
```
```

### ### 2\. Paragraphs

Paragraphs are one or more lines of text followed by one or more blank lines.

- \* **Rule:** Use a single blank line to separate paragraphs.
- \* **Rule:** Avoid trailing spaces at the end of a line.

**Good:**

```markdown

This is the first paragraph. It is followed by a blank line.

This is the second paragraph.

```

**\*\*Bad:\*\***

```markdown

This is the first paragraph.

This is the second paragraph.

```

### ### 3\. Emphasis

Markdown uses asterisks (`\*`) or underscores (`\_`) for emphasis.

- \* **\*\*Rule:\*\*** Use asterisks for bold and italic text. This is a common convention and avoids confusion with filenames or code variables that might use underscores.

- \* **\*\*Rule:\*\*** Use single asterisks for italic (*\*italic\**) and double asterisks for bold (**\*\*\*bold\*\***).

- \* **\*\*Recommendation:\*\*** Place a space between the emphasis and any surrounding text.

**\*\*Good:\*\***

```markdown

This is *important* and this is **very important**.

```

**\*\*Bad:\*\***

```markdown

This is important and this is very important.

This is *important* and this is **very important**.

```

### ### 4\. Lists

Markdown supports ordered and unordered lists.

- \* **\*\*Rule (Unordered Lists):\*\*** Use a hyphen (`-`) for list items. Hyphens are often preferred over asterisks or plus signs as they are less likely to be mistaken for other Markdown syntax.

- \* **\*\*Rule (Ordered Lists):\*\*** Use `1.` for all items in an ordered list. The Markdown parser will correctly number the list. This makes reordering items easier.

- \* **\*\*Rule:\*\*** Indent nested lists with four spaces.

\* **Recommendation:** Place a blank line before and after a list for readability.

**Good:**

```
```markdown
Here are some items:
```

```
* Item one
* Item two
    * Nested item A
    * Nested item B
* Item three
```

Here are some steps:

```
1. Step one
2. Step two
3. Step three
```
```

**Bad:**

```
```markdown
- Item one
- Item two
* Nested item A
- Item three
```
```

### 5\. Links

Links are created with square brackets for the text and parentheses for the URL.

\* **Rule:** Use the "inline" style for links.  
\* **Rule:** Use clear and descriptive link text. Avoid generic phrases like "click here."

**Good:**

```
```markdown
For more information, visit the [Markdown Guide](https://www.markdownguide.org/).
```
```

**Bad:**

```
```markdown
For more information, [click here](https://www.markdownguide.org/).
```
```



...

### ### 6\. Images

Images are similar to links but with an exclamation mark at the beginning.

- \* **Rule:** Use the "inline" style.
- \* **Rule:** Provide alternative text in the square brackets. This is important for accessibility and when the image cannot be displayed.

**Good:**

```
```markdown
![A small cat](images/cat.jpg)
```
```

### ### 7\. Code

Markdown supports inline code and code blocks.

- \* **Rule (Inline Code):** Use a single backtick (`) to wrap inline code.
- \* **Rule (Code Blocks):** Use triple backticks (```) to create a code block.
- \* **Rule (Code Blocks):** Specify the language after the opening backticks for syntax highlighting (e.g., ```python```).

**Good:**

```
```markdown
The `print()` function is used to output text.
```

```
```javascript
const name = "John";
console.log(name);
```

□

□

### ### 8. Blockquotes

Blockquotes are used to quote text from another source.

- \* **Rule:** Use a single `>` character at the beginning of each line of the quote.
- \* **Rule:** A blank line separates a blockquote from the surrounding text.

**Good:**

```
```markdown
This is a paragraph.
```

```
> This is a quote from someone. It can span multiple lines.  
> Like this.
```

```
This is another paragraph.
```

## □ 9. Horizontal Rules

Horizontal rules are used to create a thematic break.

- **Rule:** Use three or more hyphens (---) or asterisks (\*\*\*) on a line.
- **Recommendation:** Use hyphens as they are visually cleaner.

**Good:**

```
□ ---
```

## □ 10. General Conventions

- **Line Length:** Keep lines to a reasonable length, such as 80 characters, to make the document easier to read and review in a diff tool.
- **File Extension:** Use the .md or .markdown file extension.
- **Blank Lines:** Use blank lines to separate major elements (headings, paragraphs, lists) to improve readability.
- **Punctuation:** Ensure there is no space between the closing parenthesis of a link and the punctuation that follows it. For example: [text](https://example.com).