Markdown indentation

✏️ **Edit**   🖨️ **Print**

Indentation (along with line breaks) is used to make your doc source code more readable, but it can also affect how it renders in HTML when you publish.

We recommend *not* using *hard tabs*, which use **the tab key** ↗ , and instead use *soft tabs*, which use the space key. This is because different text editors treat hard tabs differently. Depending on your editor and personal configuration, your tab key might produce anywhere from 1 space to 8 or more. When you have multiple people editing the same file, you can run into problems.

While Markdown is adaptive and inconsistent indentation is sometimes not a problem, it can often break the code.

> 💡 **GITHUB FLAVORED MARKDOWN**
>
> For a comprehensive guide on the Markdown flavor used on EngHub, see **GitHub Flavored Markdown Spec** ↗ .

> 💡 **TURN ON VS CODE DETECT INDENTATION**
>
> If you're using VS Code to edit documentation, go to **File > Preferences > Settings > Text Editor** and turn on **Detect Indentation** to have VS Code assist you by automatically detecting required indentation based on the file contents.

> 🔥 **PREVIEW YOUR CONTENT**
>
> Make sure you **Preview your RTU content** to ensure your admonitions and lists are rendering properly. Admonitions won't render in the standard GitLab file preview.

Lists

In general, we use 2-space tabs for bulleted lists and 4-space tabs for numbered lists. Some scenarios might require different levels of indentation, like when mixing different types of lists.

If you include a bulleted list nested in a numbered list, the bulleted list should be indented under the numbered list with a 4-space tab as an item of a numbered list, but any further nested bullets should return to 2-space tabs.

Standard list example

> ℹ️ **NOTE**
>
> Line breaks between the bullets or numbered steps are only used to improve readability—the lists will render the same. See **Lists** for more info.

This is a bulleted list:

```
- This is a bullet.
  - This is a sub-bullet.
  - This is a sub-bullet.
- This is a bullet.
  - This is a sub-bullet.
```

Numbered list example

Numbered list example

Here's a numbered list:

```
1. This is the first step.

    1. This is a sub-numbered step.
    2. This is a sub-numbered step.

2. This is the second step.

    1. This is a sub-numbered step.

3. This is the third step.
```

Mixed list example

Here's an example for a list with a bulleted list nested within a numbered list:

```
1. This is the first step.
    - This is the first bullet of the first step.
    - This is the second bullet of the first step.
2. This is the second step.
    - This is the fist bullet of the second step .
    - This is the second bullet of the second step.
        - This is the first sub-bullet of the second bullet of the second step.
3. This is the third step.
```

> **ⓘ NOTE**
>
> - Avoid nesting numbered lists within bulleted lists.
>
> - Avoid nesting beyond three levels.

Admonitions

In a normal context, admonitions need to be aligned with the margin to render properly. When included in a bulleted or numbered list, they need to be indented within the list like any other content in the list. For most cases, this means 2-space tabs for bulleted lists and 4-space tabs for numbered lists. Some scenarios might require different levels of indentation, especially when mixing different types of lists.

Admonition examples

If your admonition is not within any lists, it should be aligned to the margin:

```
This is a paragraph with some sentences.

:::note
This is a stand-alone admonition.
:::
```

If your admonition falls within a numbered list:

```
1. This is a step.
2. This is a step.

    :::note
    This is an admonition under step 2.
    :::

3. This is a step.
```

If your admonition falls within a bulleted list:
- This is a bullet.

```
- This is a bullet.
- This is a bullet.


  :::note
  This is an admonition under the 2nd bullet.
  :::

- This is a bullet.
```

> 💡 **TIP**
>
> There are no strict rules regarding blank lines before or after Markdown elements, but we use them to increase Markdown's readability. For example, we use blank lines before and after headings of any level (#), admonitions, and images. **Reach out to the EngHub writers team** 🔗 if you have any questions.

YAML files

YAML 🔗 is a markup language / data serialization language used for configuration files. On EngHub, we use the `mkdocs.yml` file (extension `.yml` or `.yaml`) to configure the ToC (table of contents). The indentation in the YAML file determines the structure (or hierarchy) of your ToC.

Use spaces to create indentation, *not* tabs. Tabs can cause parsing errors and different systems treat tabs differently. Note that modern editors can be configured to use tab to insert the appropriate number of spaces (manually or with **Detect Indentation** turned on), but using the space bar is the safest way.

While the latest YAML spec 🔗 doesn't specify an exact amount of spaces, *we recommend 2 spaces* (also referred to as 2-space tabs). This way, the structure remains uniform and readable but also compact to reduce any scrolling necessary when creating additional levels to your hierarchy.

YAML example

> ⚠ **IMPORTANT**
>
> A "bucket", or a level that contains sub-topics, can't also be a topic itself. For example, if you try adding a link on the same line as the "`Getting started:`" section below, you'll get an error.

Here's part of the `mkdocs.yml` for this guide:

```
nav:
  - Overview: intro.md
  - Getting started:
    - Getting started with EngHub: getting-started/getting-started.md
    - step 1 - Get ready to write: getting-started/step-1-get-ready-to-write.md
    - step 2 - Create your product overview: getting-started/step-2-create-overview.md
    - step 3 - Prepare an MVP doc set: getting-started/step-3-prepare-mvp.md
  - Tutorials:
    - Set up your environment: tutorials/set-up-your-environment.md
    - Perform a content audit: tutorials/perform-content-audit.md
    - Real-time updates:
      - RTU migration overview: tutorials/real-time-updates-overview.md
      - RTU preview your content: tutorials/rtu-preview-content.md
      - RTU publish your content: tutorials/rtu-publish-content.md
```

See our EngHub docs bootcamp for more info on creating the `mkdocs.yml` ToC file.

More info

- **Admonitions**

- **GitHub Flavored Markdown Spec** 🔗

- **Markdown: Basics** 🔗

- **Markdown: Basics** ⧉

💬 Feedback

Give feedback ⧉ to help improve our content to serve you better. Your feedback will be shared with the product and documentation teams.