

What is Markdown?

 docs.microsoft.com/en-us/learn/modules/communicate-using-markdown/2-what-is-markdown

Here, we discuss the structure and syntax of Markdown. We also cover features of GitHub-Flavored Markdown (GFM), which are syntax extensions that enable you to integrate GitHub features into content.

Note

This unit is intended to give you a taste of what Markdown is about. For a more in-depth review, see [Markdown syntax description](#) or [GitHub-Flavored Markdown Spec](#).

Markdown exists to shield content creators from the overhead of HTML. While HTML is great for rendering content exactly how it was intended, it takes up a lot of space and can be unwieldy to work with, even in small doses. The invention of Markdown offered a great compromise between the power of HTML for content description and the ease of plain text for editing.

Emphasizing text

The most important part of any communication on GitHub is usually the text itself. But how do you show that some parts of the text are more important than others?

Using italics in text is as easy as surrounding the target text with single asterisks (`*`) or single underscores (`_`). Just be sure to close an emphasis with the same character you used to open with. Be observant how you combine the use of asterisks and underscores. Here are several examples:

markdown

```
This is italic text.  
This is also italic text.
```

| This is *italic* text. This is also *italic* text.

Create bold text by using two asterisks (`**`) or two underscores (`__`).

markdown

```
This is bold text.  
This is also bold text.
```

| This is **bold** text. This is also **bold** text.

You can also mix different emphases.

markdown

```
_This is italic and bold text_ using a single underscore for italic and double asterisks for bold.  
__This is bold and italic text__ using double underscores for bold and single asterisks for italic.
```

| *This is **italic and bold** text* using a single underscore for italic and double asterisks for bold. **This is bold and *italic* text** using double underscores for bold and single asterisks for italic.

To use a literal asterisk, precede it with an escape character--in GFM that is a backslash (`\`). This example results in the underscores and asterisks being shown in the output.

markdown

```
\_This is all \*\*plain\*\* text\_.
```

| _This is all **plain** text_.

Declaring headings

HTML provides content headings, such as the `<h1>` tag. In Markdown, this is supported via the `#` symbol. Just use one `#` for each heading level from 1-6.

markdown

```
##### This is H6 text
```

| This is H6 text

Linking to images and sites

Image and site links use a very similar syntax.

markdown

```
![Link an image.](/learn/azure-devops/shared/media/mara.png)
```

markdown

[Link to Microsoft Learn](/learn)

| [Link to Microsoft Learn](#)



Making lists

You can define ordered or unordered lists. You can also define nested items through indentation.

- Ordered lists start with numbers.
- Unordered lists can use asterisks or dashes (-).

Here is the markdown for an ordered list.

markdown

```
1. First
1. Second
1. Third
```

Result:

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

markdown

```
- First
- Nested
- Second
- Third
```

Here is the markdown for an unordered list.

```
• First
  Nested
• Second
• Third
```

Building tables

Tables can be constructed using a combination of pipes (|) for column breaks and dashes (-) to designate the previous row as a header.

```
First|Second
-|-
1|2
3|4
```

First	Second
1	2
3	4

Table 1

Quoting text

Blockquotes can be created using the greater than (>) character.

> This is quoted text.

| | This is quoted text.

Filling the gaps with inline HTML

If you come across an HTML scenario not supported by Markdown, you can simply use that HTML inline.

markdown

Here is a
line break

```
Here is a
line break
```

Working with code

Markdown provides default behavior for working with inline code blocks delimited by the backtick (`) character. When decorating text with this character, it is rendered as code.

This is ``code``.

| This is `code` .

If you have a code segment spanning multiple lines, you can use three backticks (`````) before and after to create a fenced code block.

```
```  
var first = 1;
var second = 2;
var sum = first + second;
```
```

| `var first = 1;
var second = 2;
var sum = first + second;`

GFM extends this support with syntax highlighting for popular languages. Just specify the language as part of the first tick sequence.

```
```javascript  
var first = 1;
var second = 2;
var sum = first + second;
```
```

| JavaScript
`var first = 1;
var second = 2;
var sum = first + second;`

Cross-linking issues and pull requests

GFM supports a variety of shortcode formats to make it easy to link to issues and pull requests. The easiest way to do this is to use the format `#ID` , such as `#3602` . GitHub will automatically adjust longer links to this format if you paste them in. There are also additional conventions you can follow, such as if you are working with other tools or want to specify other projects/branches.

| Reference type | Raw reference | Short link |
|--|---|-----------------------------------|
| Issue or pull request URL | <code>https://github.com/desktop/desktop/pull/3602</code> | <code>#3602</code> |
| <code>#</code> and issue or pull request number | <code>#3602</code> | <code>#3602</code> |
| <code>GH-</code> and issue or pull request number | <code>GH-3602</code> | <code>GH-3602</code> |
| <code>Username/Repository#</code> and issue or pull request number | <code>desktop/desktop#3602</code> | <code>desktop/desktop#3602</code> |

Cross-linking issues and pull requests

For more information, see [Autolinked references and URLs](#).

Linking specific commits

You can link to a commit by pasting in its ID, or, simply using its secure hash algorithm (SHA).

| Reference type | Raw reference | Short link |
|-------------------------|---|--------------------------------------|
| Commit URL | <code>https://github.com/desktop/desktop/commit/8304e9c271a5e5ab4fda797304cd7bcca7158c87</code> | <code>8304e9c</code> |
| SHA | <code>8304e9c271a5e5ab4fda797304cd7bcca7158c87</code> | <code>8304e9c</code> |
| User@SHA | <code>desktop@8304e9c271a5e5ab4fda797304cd7bcca7158c87</code> | <code>desktop@8304e9c</code> |
| Username/Repository@SHA | <code>desktop/desktop@8304e9c271a5e5ab4fda797304cd7bcca7158c87</code> | <code>desktop/desktop@8304e9c</code> |

Linking specific commits

Mentioning users and teams

Typing an `@` symbol, followed by a GitHub username, will send a notification to that person about the comment. This is called an "`@mention`", because you're mentioning the individual. You can also `@mention` teams within an organization.

`@githubteacher`

`@githubteacher`

For more information, see [Receiving notifications about activity on GitHub](#).

Tracking task lists

You can create task lists within issues or pull requests using the syntax illustrated below. These can be helpful to track progress when used in the body of an issue or pull request.

- [x] First task
- [x] Second task
- [] Third task

- ☒ First task
☒ Second task
☐ Third task

Next unit: Exercise - Communicating using Markdown

[Continue](#)

Need help? See our [troubleshooting guide](#) or provide specific feedback by [reporting an issue](#).