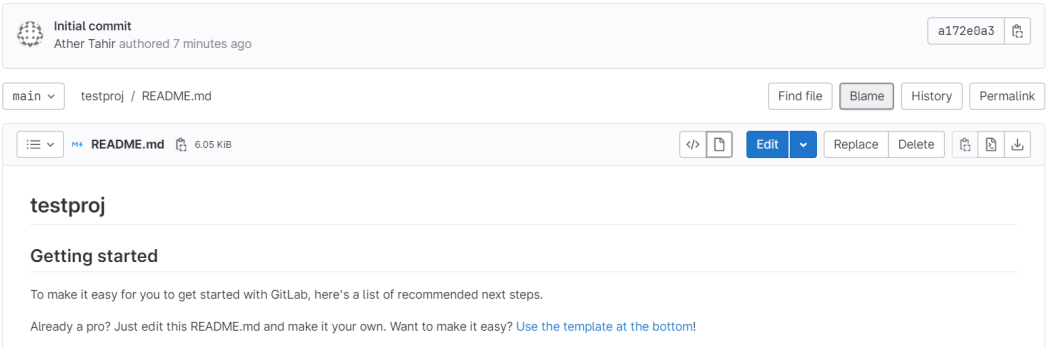
Lab: Gitlab Markdown

In this lab, we will cover following topics:

- Adding a table
- Adding a collapsed section
- Adding a quote
- Adding a comment
- Saving your work

Creating or editing your README

1. Create a repository with the, initializing the repository with a `README.md` file.
2. Edit the `README.md` file and delete the template text that is automatically added when you create the file.
3. Click `README.md` and then click `Edit` to start editing the file.



Adding a table

You can use Markdown tables to organize information. Here, you'll use a table to introduce yourself by ranking something, such as your most-used programming languages or frameworks, the things you're spending your time learning, or your favorite hobbies. When a table column contains numbers, it's useful to right-align the column by using the syntax `--:` below the header row.

1. Return to the **Edit file** tab.
2. To introduce yourself, two lines below the `</picture>` tag, add an `## About me` header and a short paragraph about yourself, like the following.

```
## About me

Hi, I'm Mona.
```

3. Two lines below this paragraph, insert a table by copying and pasting the following markup.

```
| Rank | THING-TO-RANK |
|-----:|---------------|
|    1|               |
|    2|               |
|    3|               |
```

4. In the column on the right, replace `THING-TO-RANK` with "Languages," "Hobbies," or anything else, and fill in the column with your list of things.
5. To check the table has rendered correctly, click the **Preview** tab.

Example

```
## About me

Hi, I'm Mona.

| Rank | Languages |
|-----:|-----------|
|     1| Javascript|
|     2| Python    |
|     3| SQL       |
```

How it looks

About me

Hi, I'm Mona.

Rank	Languages
1	Javascript
2	Python
3	SQL

Adding a collapsed section

To keep your content tidy, you can use the `<details>` tag to create an expandible collapsed section.

1. To create a collapsed section for the table you created, wrap your table in `<details>` tags like in the following example.

   HTML

```
<details>
<summary>My top THINGS-TO-RANK</summary>

YOUR TABLE

</details>
```

2. Between the `<summary>` tags, replace `THINGS-TO-RANK` with whatever you ranked in your table.
3. Optionally, to make the section display as open by default, add the `open` attribute to the `<details>` tag.

   ```
   <details open>
   ```

4. To check the collapsed section has rendered correctly, click the **Preview** tab.

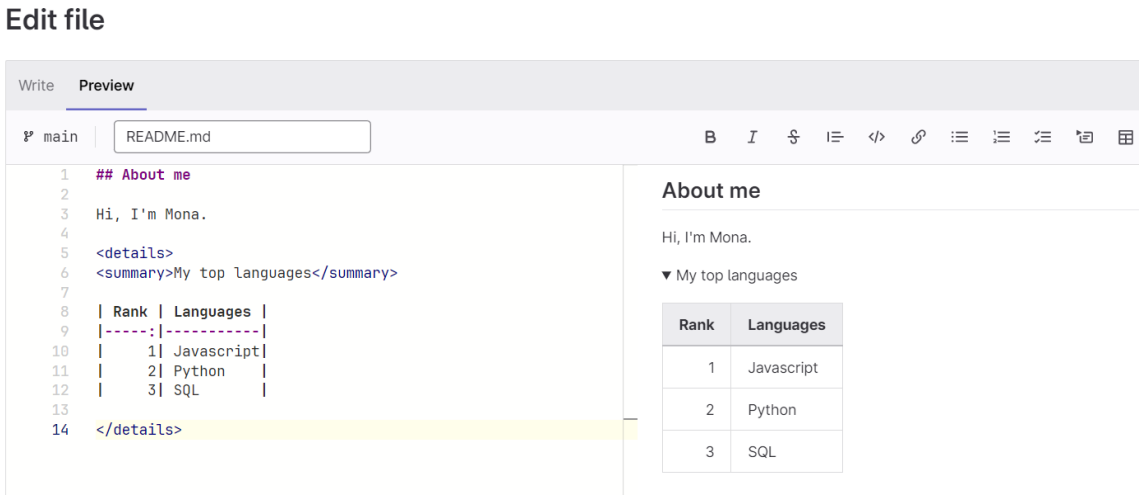
Example

```
<details>
<summary>My top languages</summary>

| Rank | Languages |
|-----:|-----------|
|    1| Javascript|
|    2| Python    |
|    3| SQL       |

</details>
```

How it looks



Adding a quote

Markdown has many other options for formatting your content. Here, you'll add a horizontal rule to divide your page and a blockquote to format your favorite quote.

1. At the bottom of your file, two lines below the `</details>` tag, add a horizontal rule by typing three or more dashes.

   ```
   ---
   ```

2. Below the `---` line, add a quote by typing markup like the following.

   ```
   > QUOTE
   ```

   Replace `QUOTE` with a quote of your choice. Alternatively, copy the quote from our example below.

3. To check everything has rendered correctly, click the **Preview** tab.

Example

```
---
> If we pull together and commit ourselves, then we can push through anything.

— Mona
```

How it looks

> If we pull together and commit ourselves, then we can push through anything.
>
> — Mona the Octocat

Adding a comment

You can use HTML comment syntax to add a comment that will be hidden in the output. Here, you'll add a comment to remind yourself to update your README later.

1. Two lines below the `## About me` header, insert a comment by using the following markup.

   ```
   <!-- COMMENT -->
   ```

   Replace `COMMENT` with a "to-do" item you remind yourself to do something later (for example, to add more items to the table).

2. To check your comment is hidden in the output, click the **Preview** tab.

Example

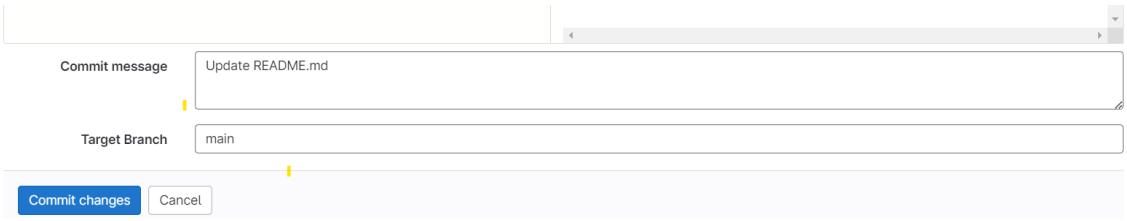
```
## About me

<!-- TO DO: add more details about me later -->
```

Saving your work

When you're happy with your changes, save your profile README by clicking **Commit changes**.

Committing directly to the `main` branch will make your changes visible to any visitor on your profile. If you want to save your work but aren't ready to make it visible on your profile, you can select **Create a new branch for this commit and start a pull request**.



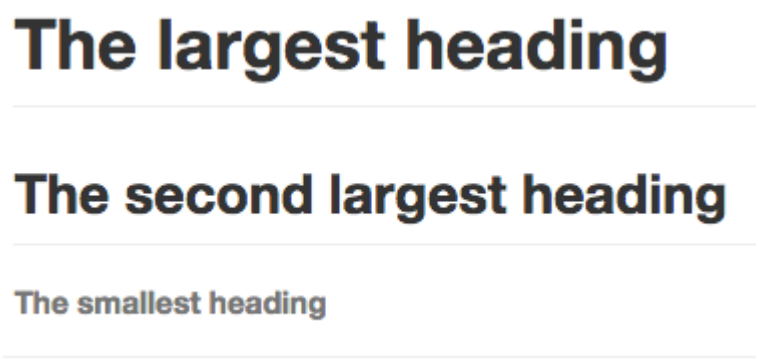
Basic writing and formatting syntax

Create sophisticated formatting for your prose and code on Gitlab with simple syntax.

Headings

To create a heading, add one to six [#] symbols before your heading text. The number of [#] you use will determine the size of the heading.

```
# The largest heading
## The second largest heading
###### The smallest heading
```



Quoting text

You can quote text with a [>].

```
Text that is not a quote

> Text that is a quote
```

Text that is not a quote

> Text that is a quote

Quoting code

You can call out code or a command within a sentence with single backticks. The text within the backticks will not be formatted. You can also press the [Command]+[E] (Mac) or [Ctrl]+[E] (Windows/Linux) keyboard shortcut to insert the backticks for a code block within a line of Markdown.

```
Use `git status` to list all new or modified files that haven't yet been committed.
```

Use `git status` to list all new or modified files that haven't yet been committed.

To format code or text into its own distinct block, use triple backticks.

```
Some basic Git commands are:
```
git status
git add
git commit
```
```

Some basic Git commands are:

```
git status
git add
git commit
```

If you are frequently editing code snippets and tables, you may benefit from enabling a fixed-width font in all comment fields on Gitlab.

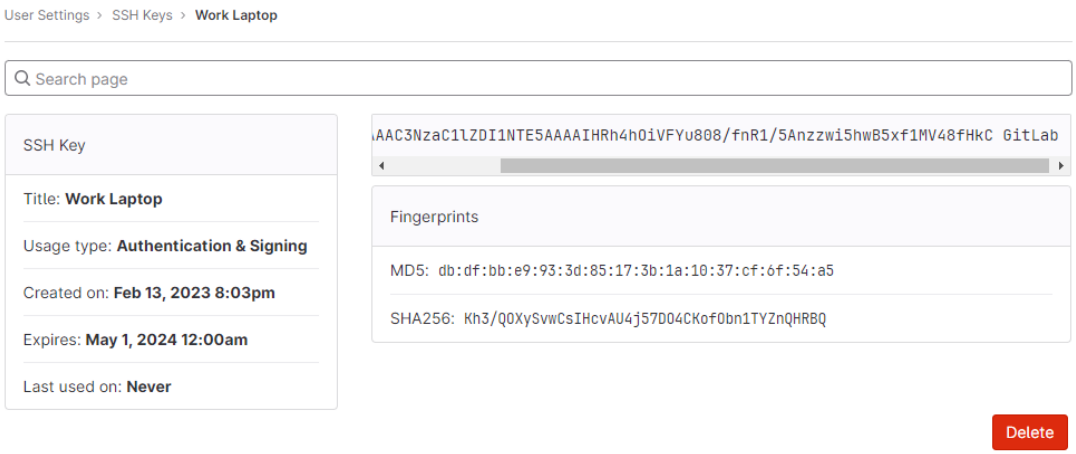
Supported color models

In issues, pull requests, and discussions, you can call out colors within a sentence by using backticks. A supported color model within backticks will display a visualization of the color.

```
The background color should be `#ffffff` for light mode and `#0d1117` for dark mode.
```

The background color should be `#ffffff` ○ for light mode and `#0d1117` ● for dark mode.

Here are the currently supported color models.



Notes:

- A supported color model cannot have any leading or trailing spaces within the backticks.
- The visualization of the color is only supported in issues, pull requests, and discussions.

Lists

You can make an unordered list by preceding one or more lines of text with [-], [*], or [+].

```
- George Washington
* John Adams
+ Thomas Jefferson
```

- George Washington
- John Adams
- Thomas Jefferson

To order your list, precede each line with a number.

```
1. James Madison
2. James Monroe
3. John Quincy Adams
```

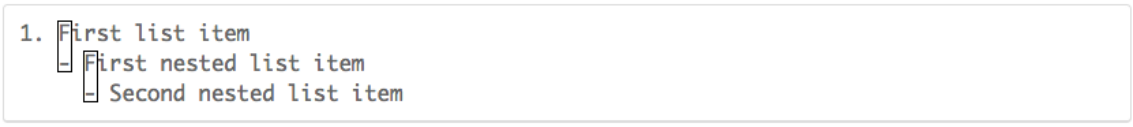
1. James Madison
2. James Monroe
3. John Quincy Adams

Nested Lists

You can create a nested list by indenting one or more list items below another item.

To create a nested list using the web editor on Gitlab or a text editor that uses a monospaced font, like `Visual Studio Code`, you can align your list visually. Type space characters in front of your nested list item, until the list marker character ([-] or [*]) lies directly below the first character of the text in the item above it.

```
1. First list item
   - First nested list item
     - Second nested list item
```



1. First list item
    - First nested list item
        - Second nested list item

To create a nested list in the comment editor on Gitlab, which doesn't use a monospaced font, you can look at the list item immediately above the nested list and count the number of characters that appear before the content of the item. Then type that number of space characters in front of the nested list item.

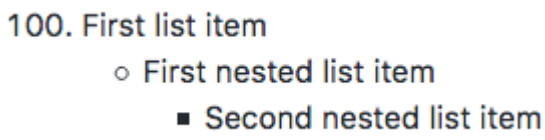
In this example, you could add a nested list item under the list item `100. First list item` by indenting the nested list item a minimum of five spaces, since there are five characters (`100. `) before `First list item`.

```
100. First list item
     - First nested list item
```

100. First list item
    - First nested list item

You can create multiple levels of nested lists using the same method. For example, because the first nested list item has seven characters (`␣␣␣␣-␣`) before the nested list content `First nested list item`, you would need to indent the second nested list item by seven spaces.

```
100. First list item
     - First nested list item
       - Second nested list item
```



Task lists

To create a task list, preface list items with a hyphen and space followed by `[ ]`. To mark a task as complete, use `[x]`.

```
- [x] Complete PR Review
- [ ] Add delight to the experience when all tasks are complete :tada:
```

Paragraphs

You can create a new paragraph by leaving a blank line between lines of text.

Footnotes

You can add footnotes to your content by using this bracket syntax:

```
Here is a simple footnote[^1].

A footnote can also have multiple lines[^2].

You can also use words, to fit your writing style more closely[^note].

[^1]: My reference.
[^2]: Every new line should be prefixed with 2 spaces.
  This allows you to have a footnote with multiple lines.
[^note]:
    Named footnotes will still render with numbers instead of the text but allow
easier identification and linking.
    This footnote also has been made with a different syntax using 4 spaces for new
lines.
```

The footnote will render like this:

Here is a simple footnote[1].

A footnote can also have multiple lines[2].

You can also use words, to fit your writing style more closely[3].

1. My reference.

2. Every new line should be prefixed with 2 spaces.
   This allows you to have a footnote with multiple lines.

3. Named footnotes will still render with numbers instead of the text but allow easier identification and linking.
   This footnote also has been made with a different syntax using 4 spaces for new lines.

Note: The position of a footnote in your Markdown does not influence where the footnote will be rendered. You can write a footnote right after your reference to the footnote, and the footnote will still render at the bottom of the Markdown.

Footnotes are not supported in wikis.

Hiding content with comments

You can tell Gitlab to hide content from the rendered Markdown by placing the content in an HTML comment.

```
<!-- This content will not appear in the rendered Markdown -->
```

Ignoring Markdown formatting

You can tell Gitlab to ignore (or escape) Markdown formatting by using [\] before the Markdown character.

```
Let's rename \*our-new-project\* to \*our-old-project\*.
```

Let's rename *our-new-project* to *our-old-project*.