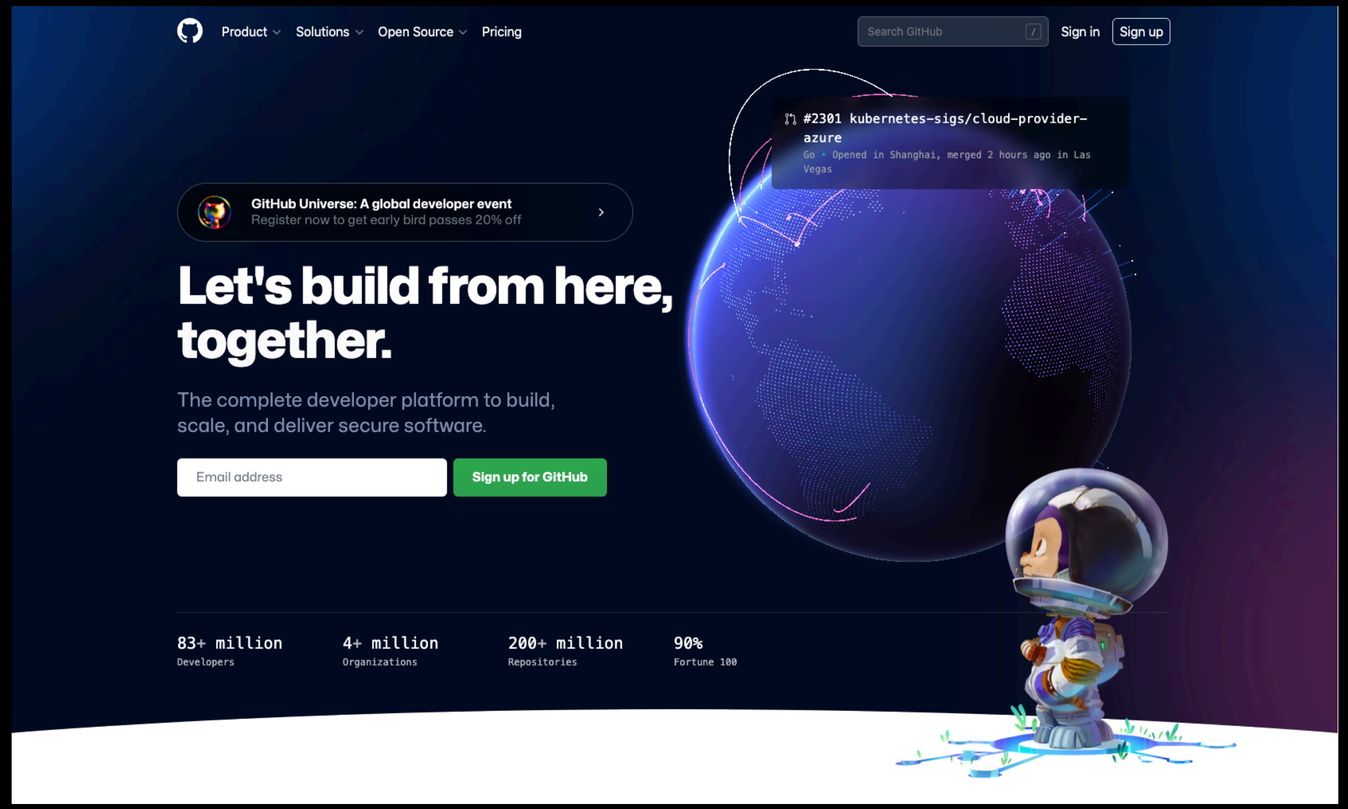
Bioinformatics of Sequence Variation

Ziying Yang

Introduction to Github

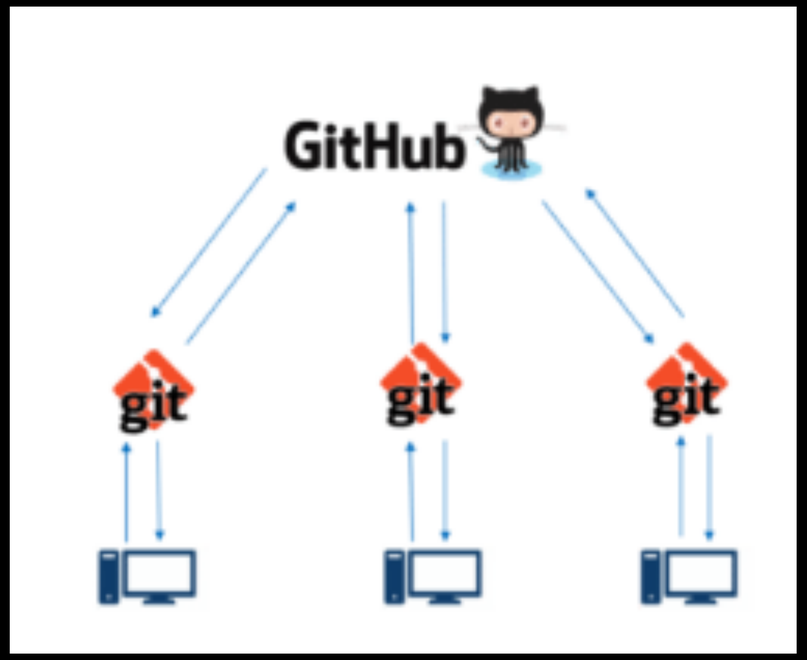
What is Github?

https://github.com/



- To be very crisp about what exactly is GitHub, it is a file or code-sharing service to collaborate with different people.

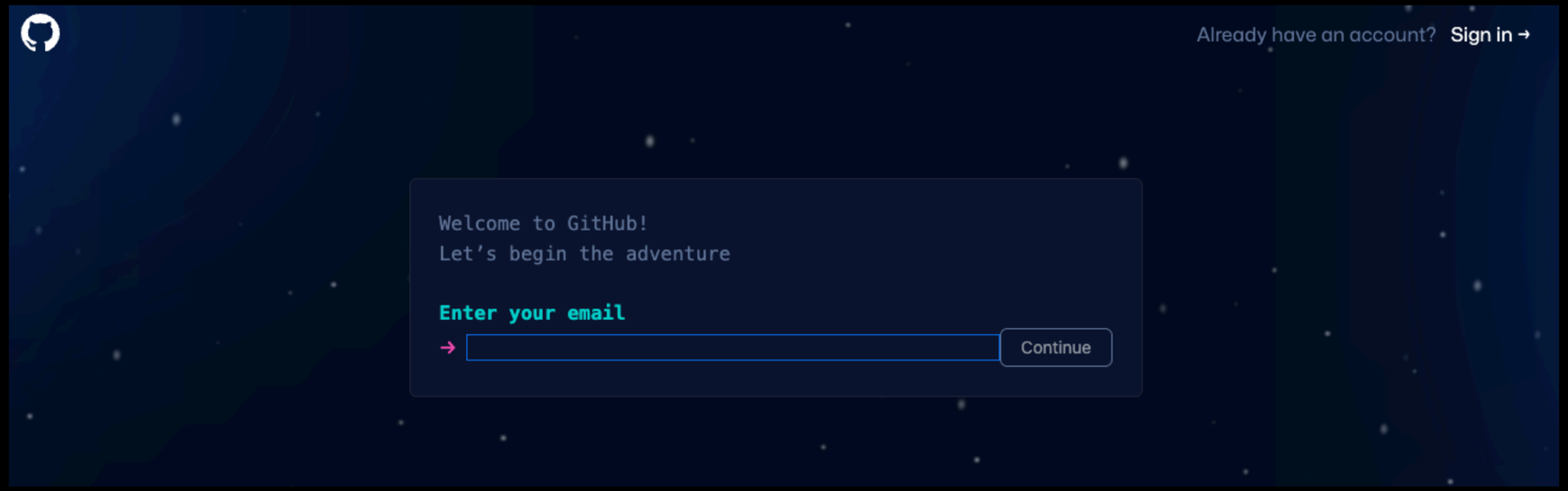
Why GitHub?



GitHub is a central repository and Git is a tool which allows you to create a local repository.

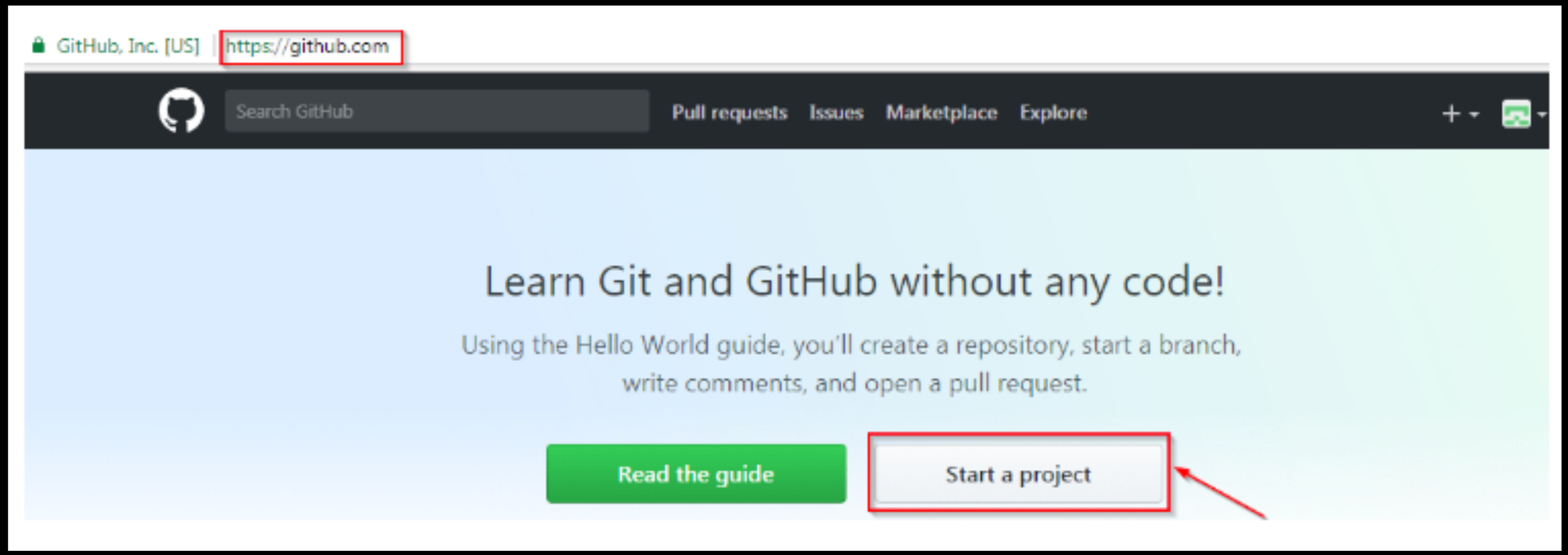
Create a GitHub account

- To be able to use GitHub, you will have to create an account first. You can do that on their website.



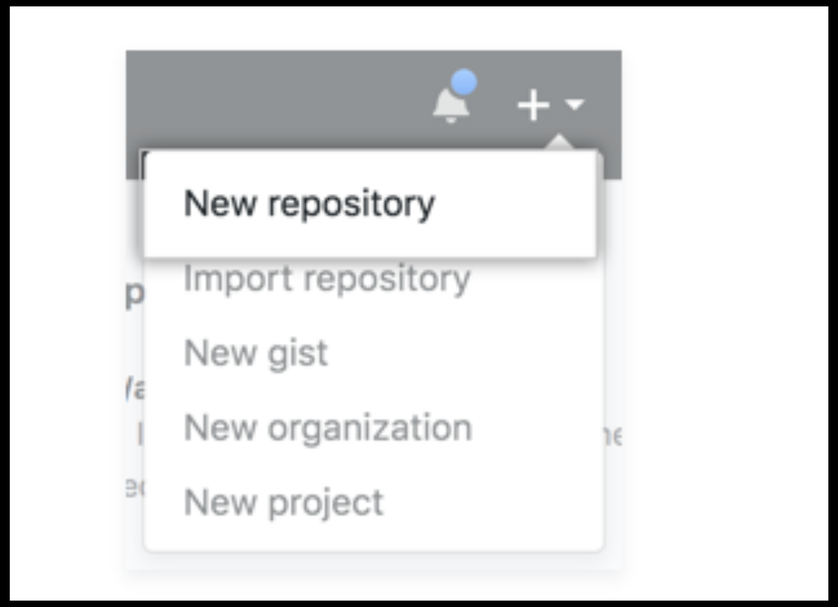
Create a GitHub Repository

- Click on “Start a new project”.

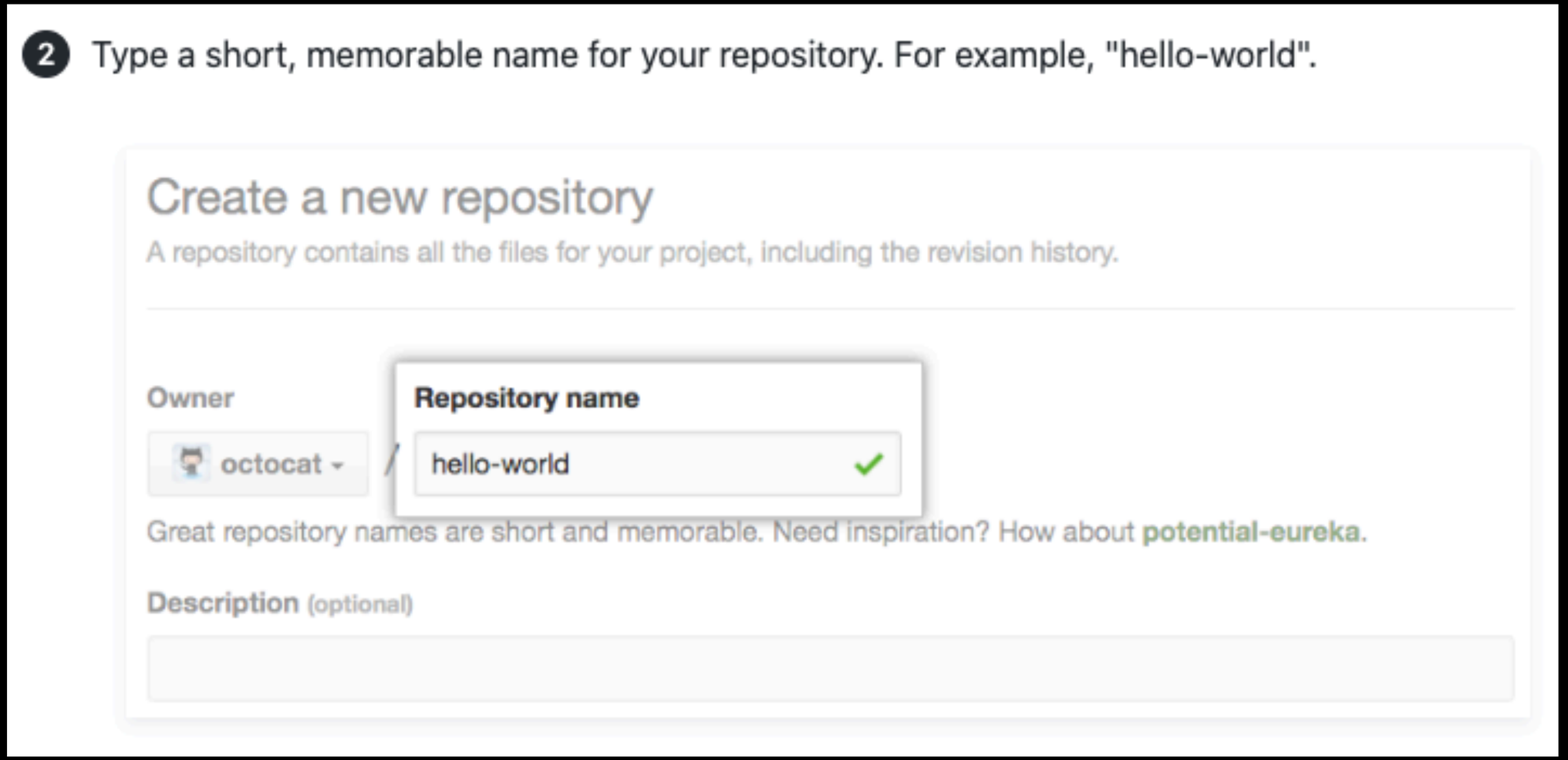


Create a GitHub Repository

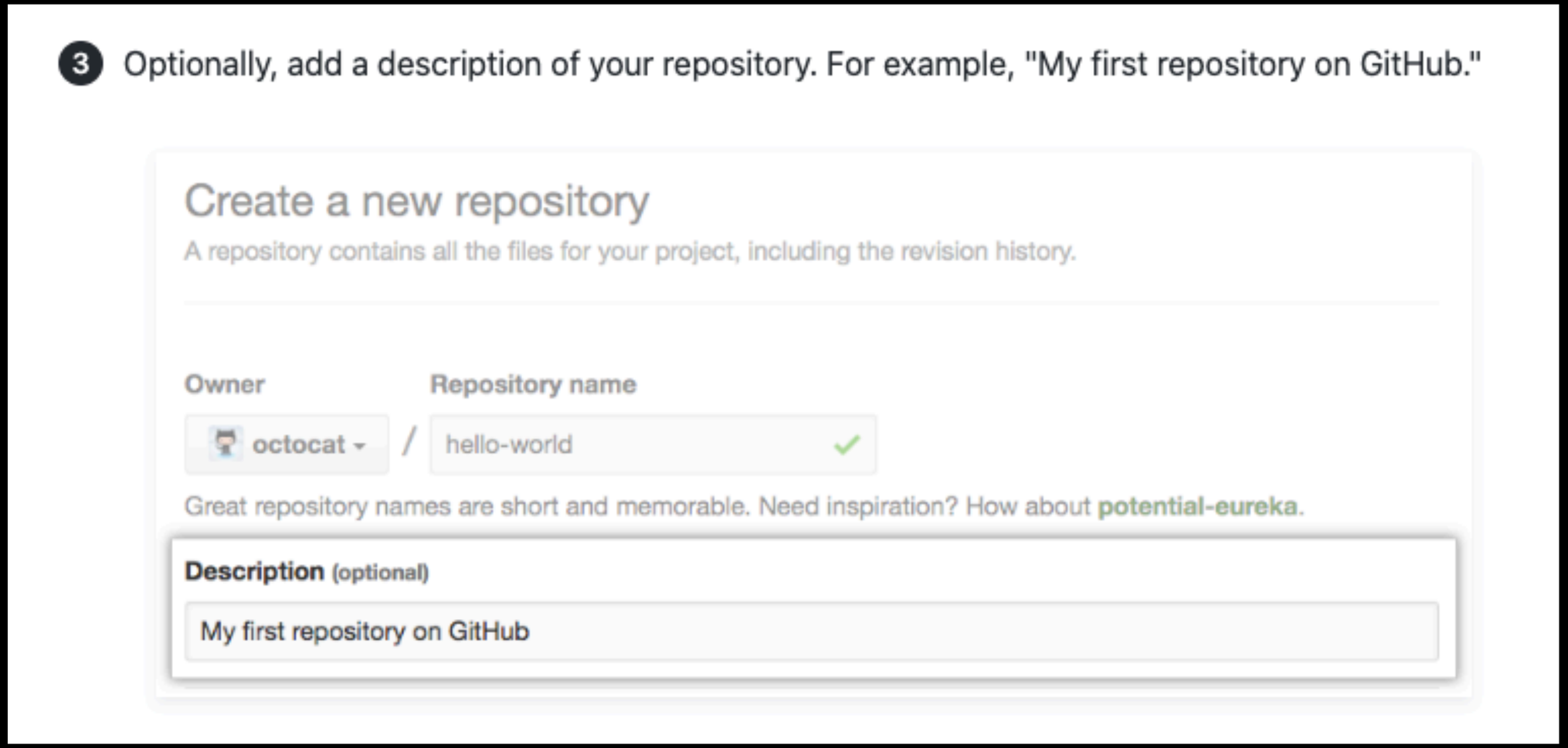
- In the upper-right corner of any page, use the drop-down menu, and select **New repository**.



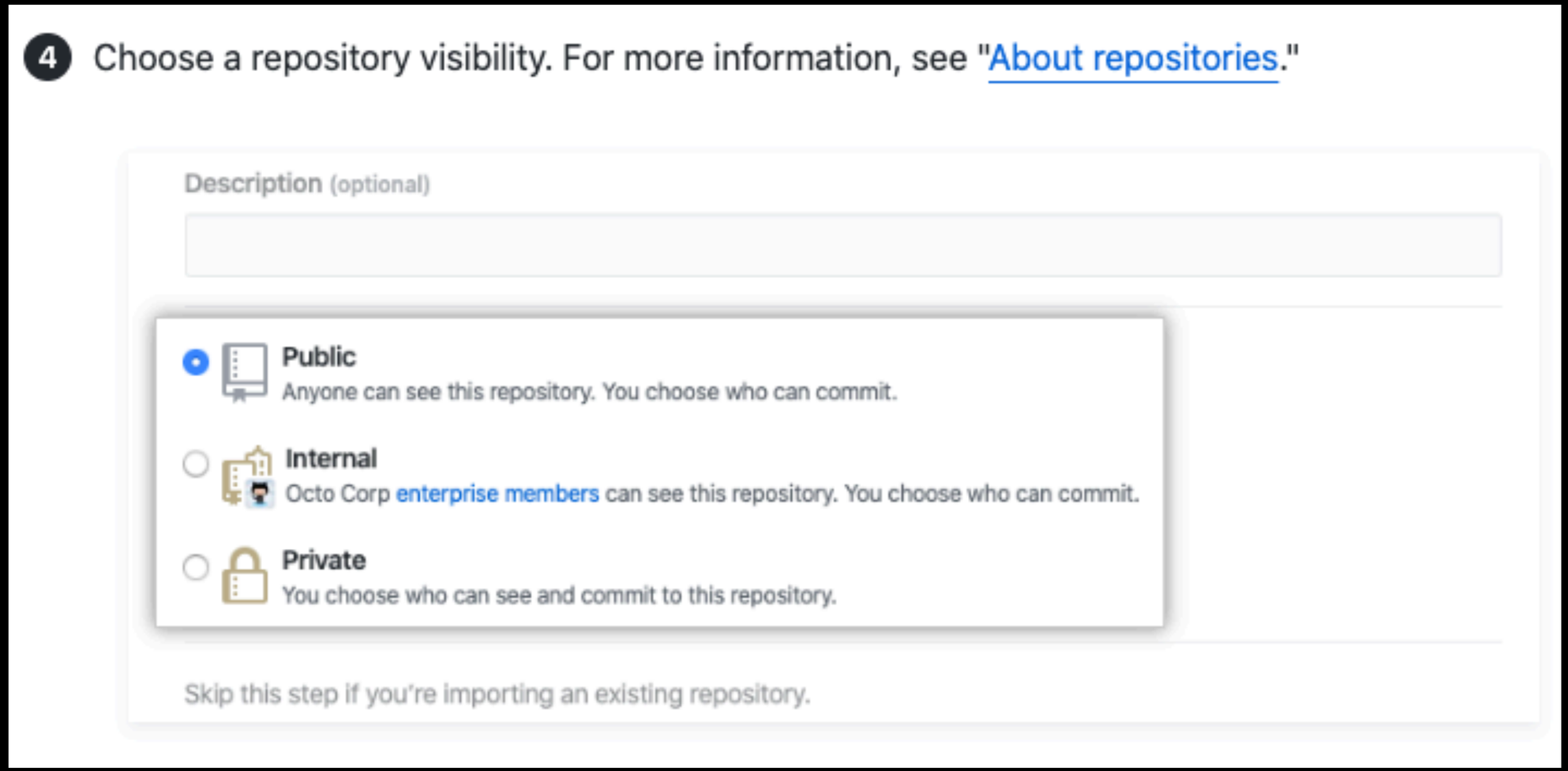
Create a GitHub Repository



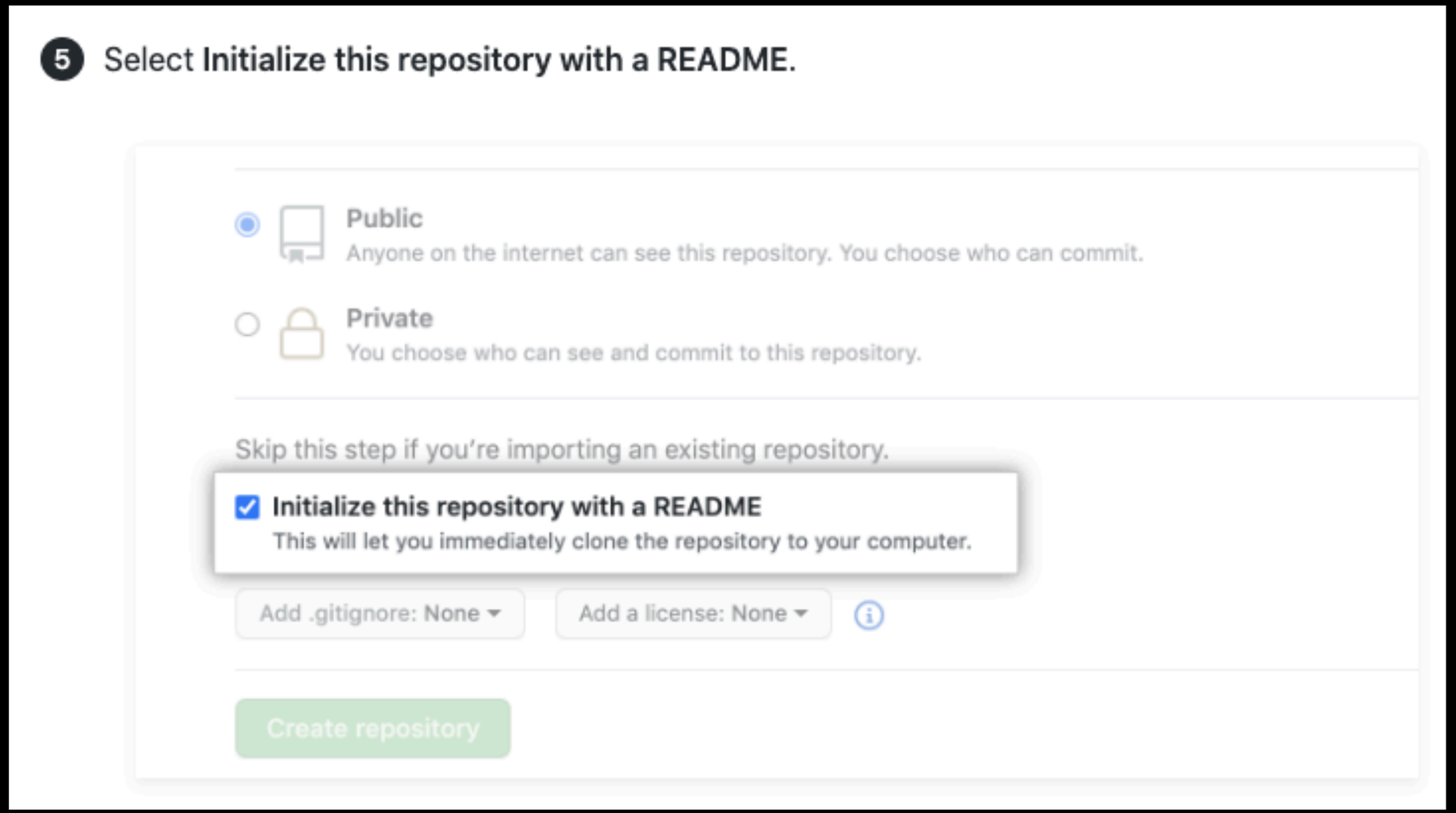
Create a GitHub Repository



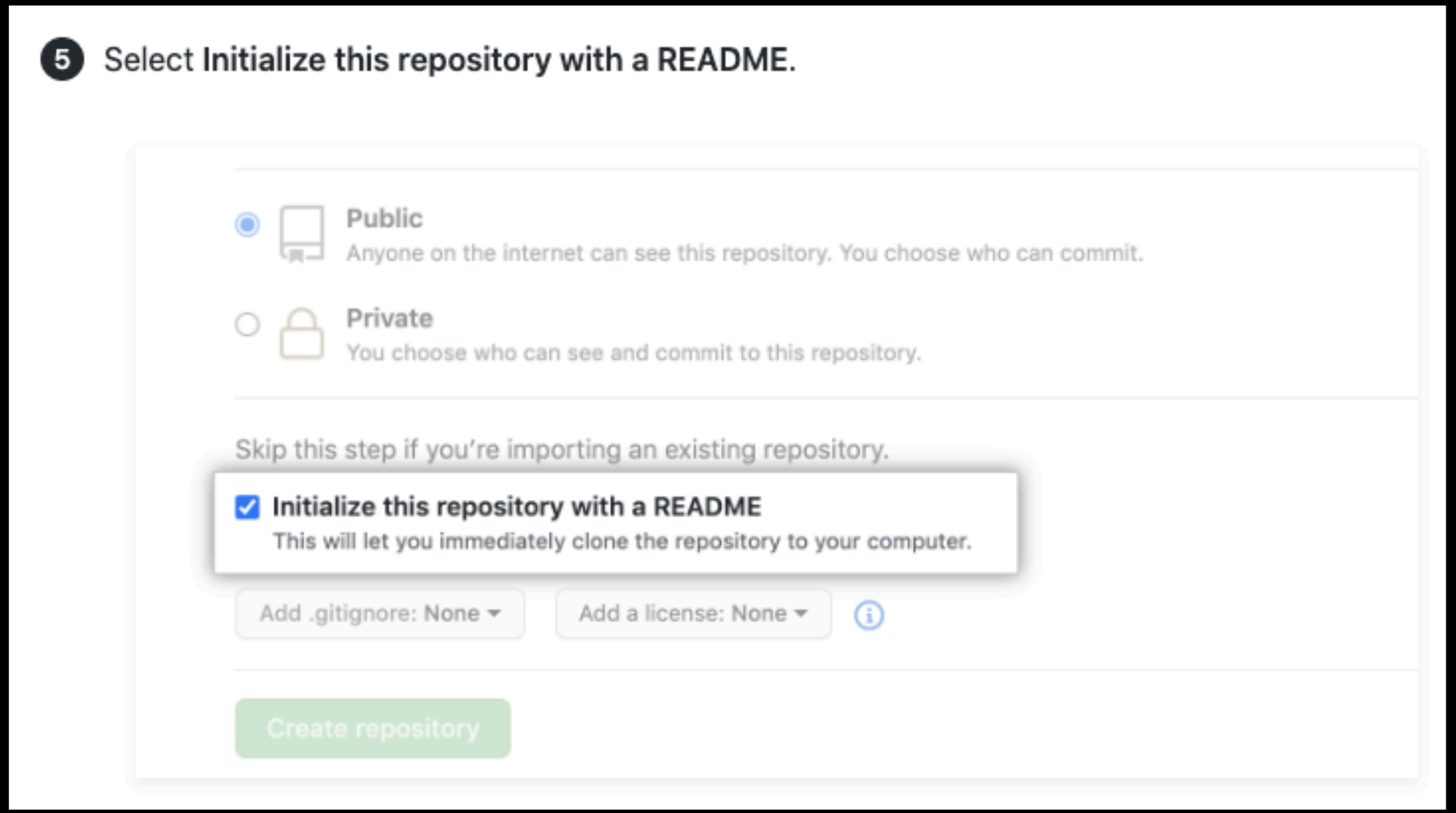
Create a GitHub Repository



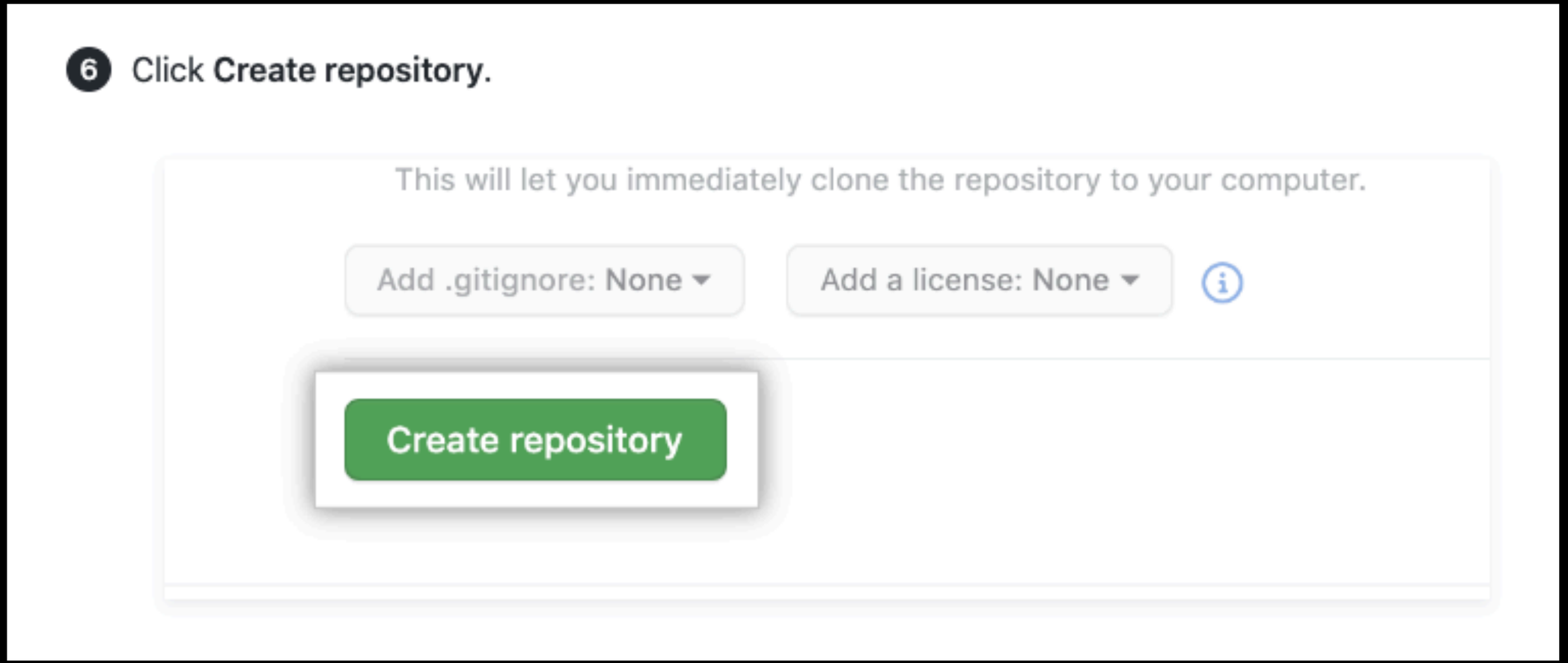
Create a GitHub Repository



Create a GitHub Repository



Create a GitHub Repository



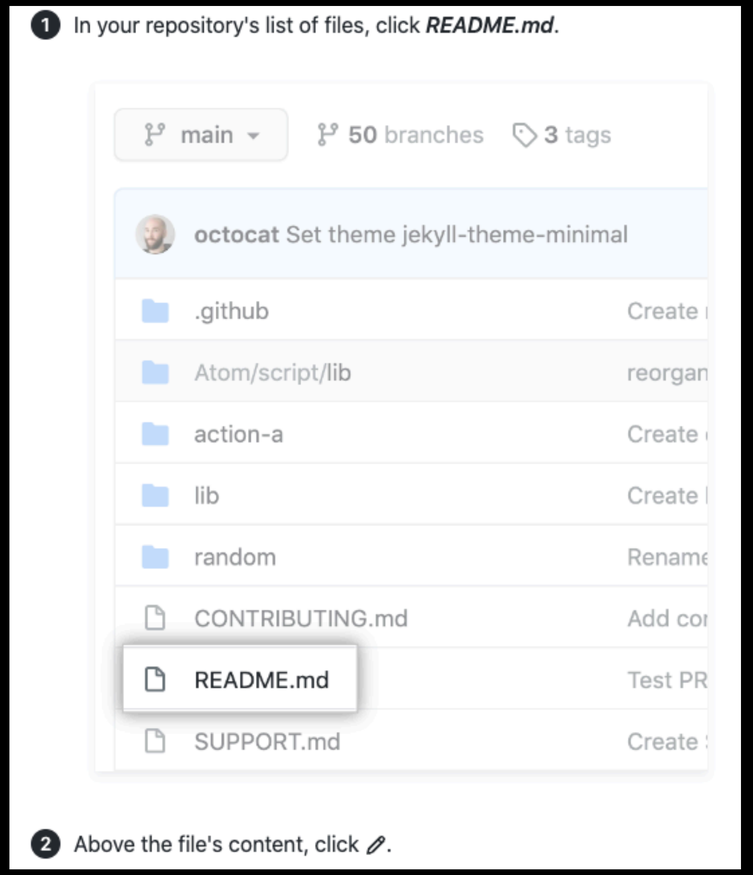
Exercise

Try to create your own repository!

Commit your first change

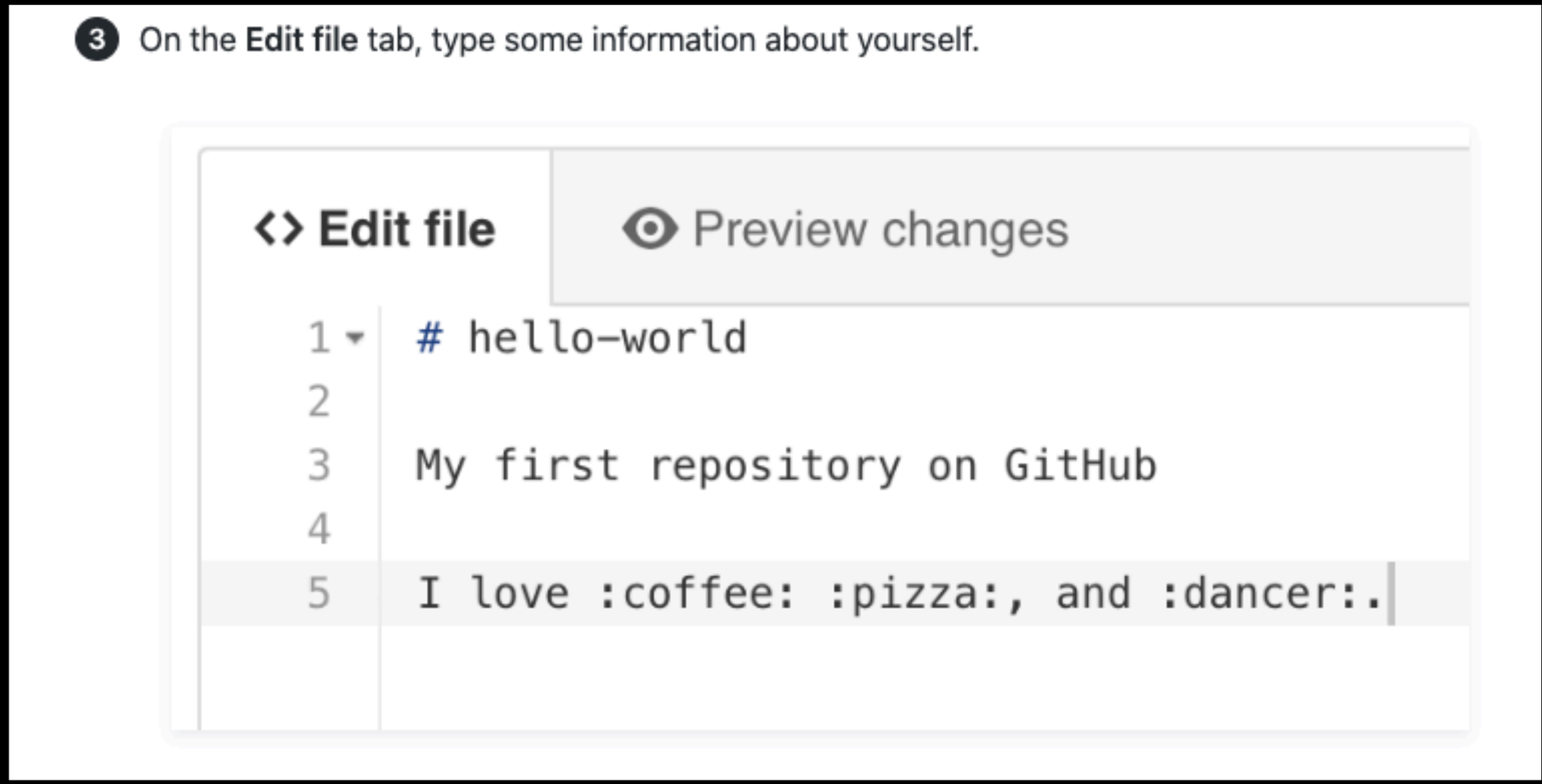
A *commit* is like a snapshot of all the files in your project at a particular point in time.

- Let's commit a change to the *README* file.



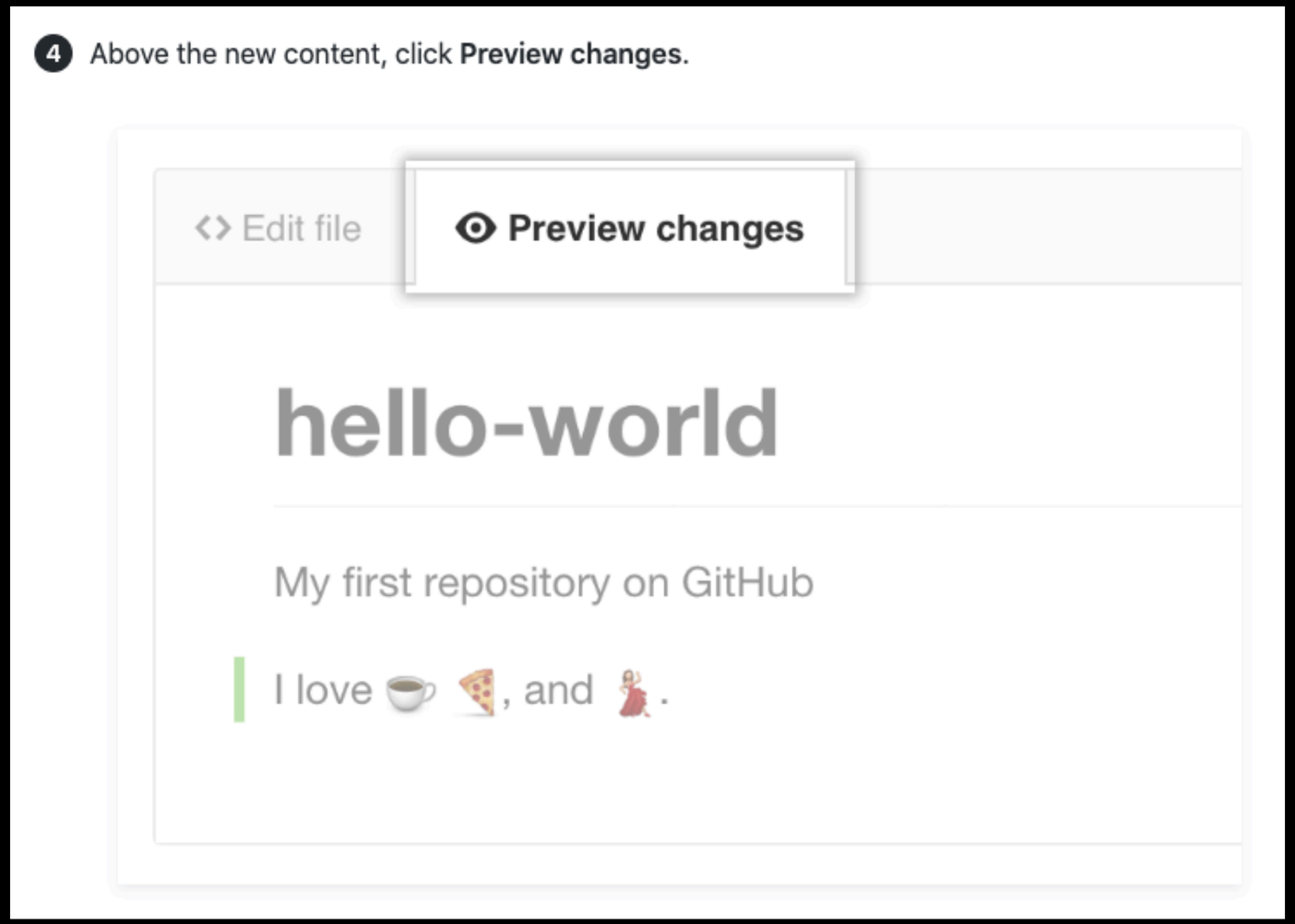
Commit your first change

A *commit* is like a snapshot of all the files in your project at a particular point in time.



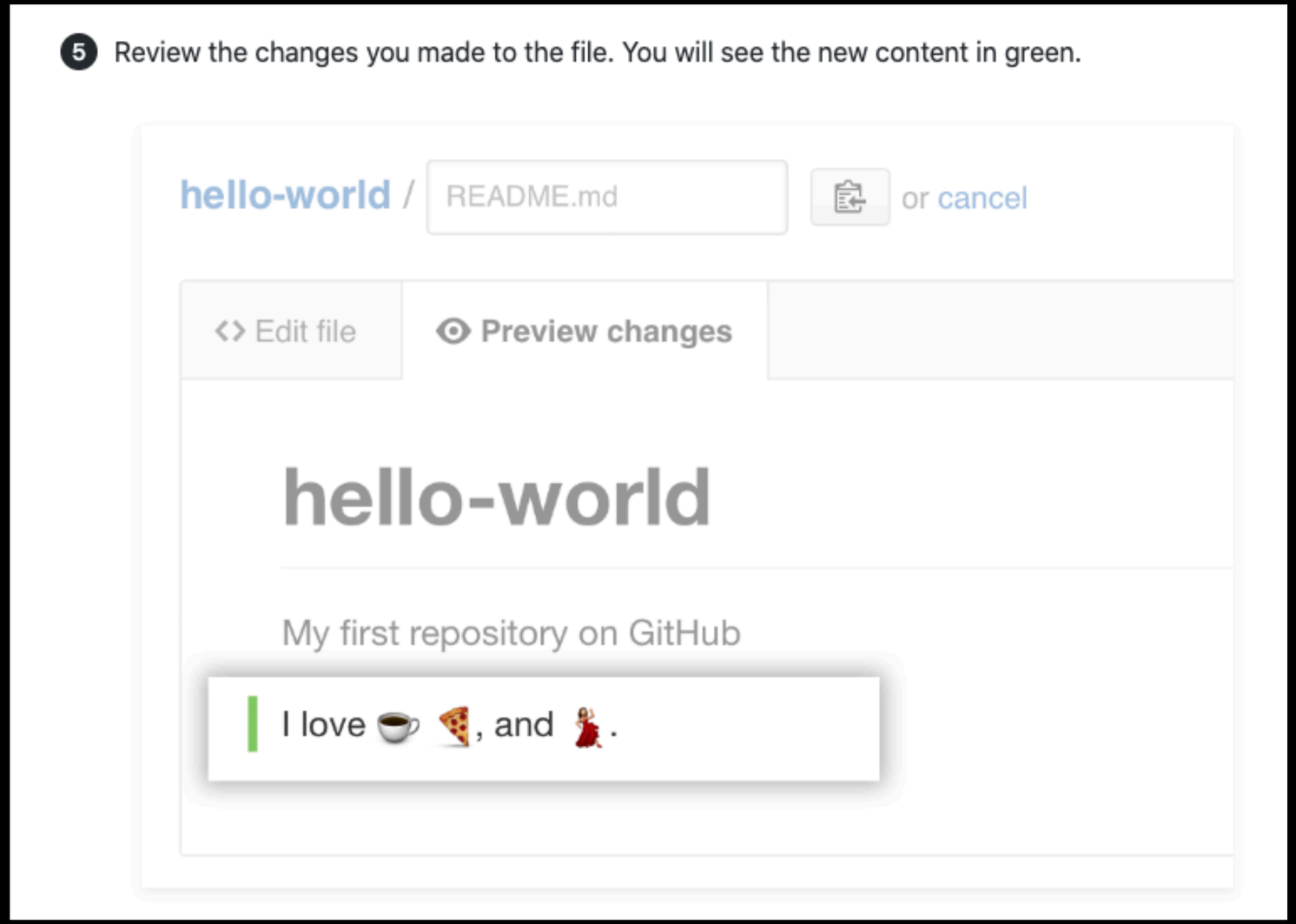
Commit your first change

A *commit* is like a snapshot of all the files in your project at a particular point in time.



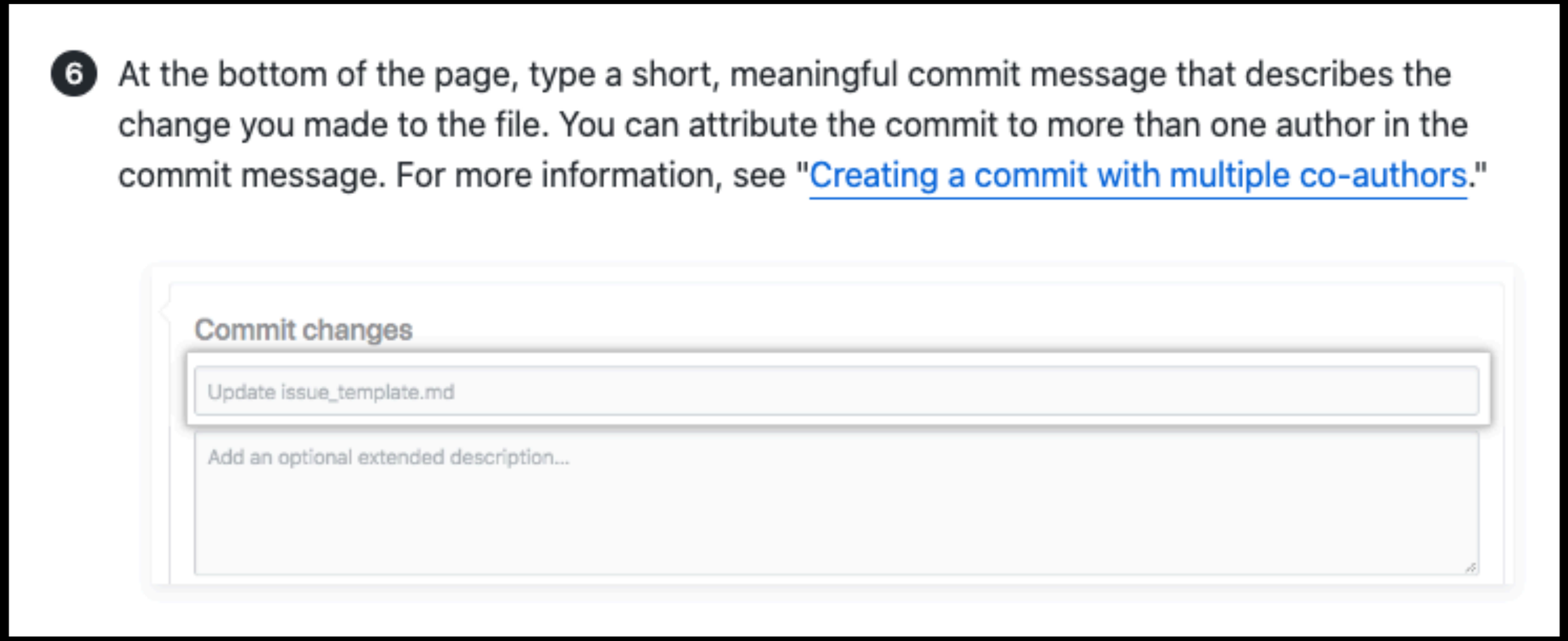
Commit your first change

A *commit* is like a snapshot of all the files in your project at a particular point in time.



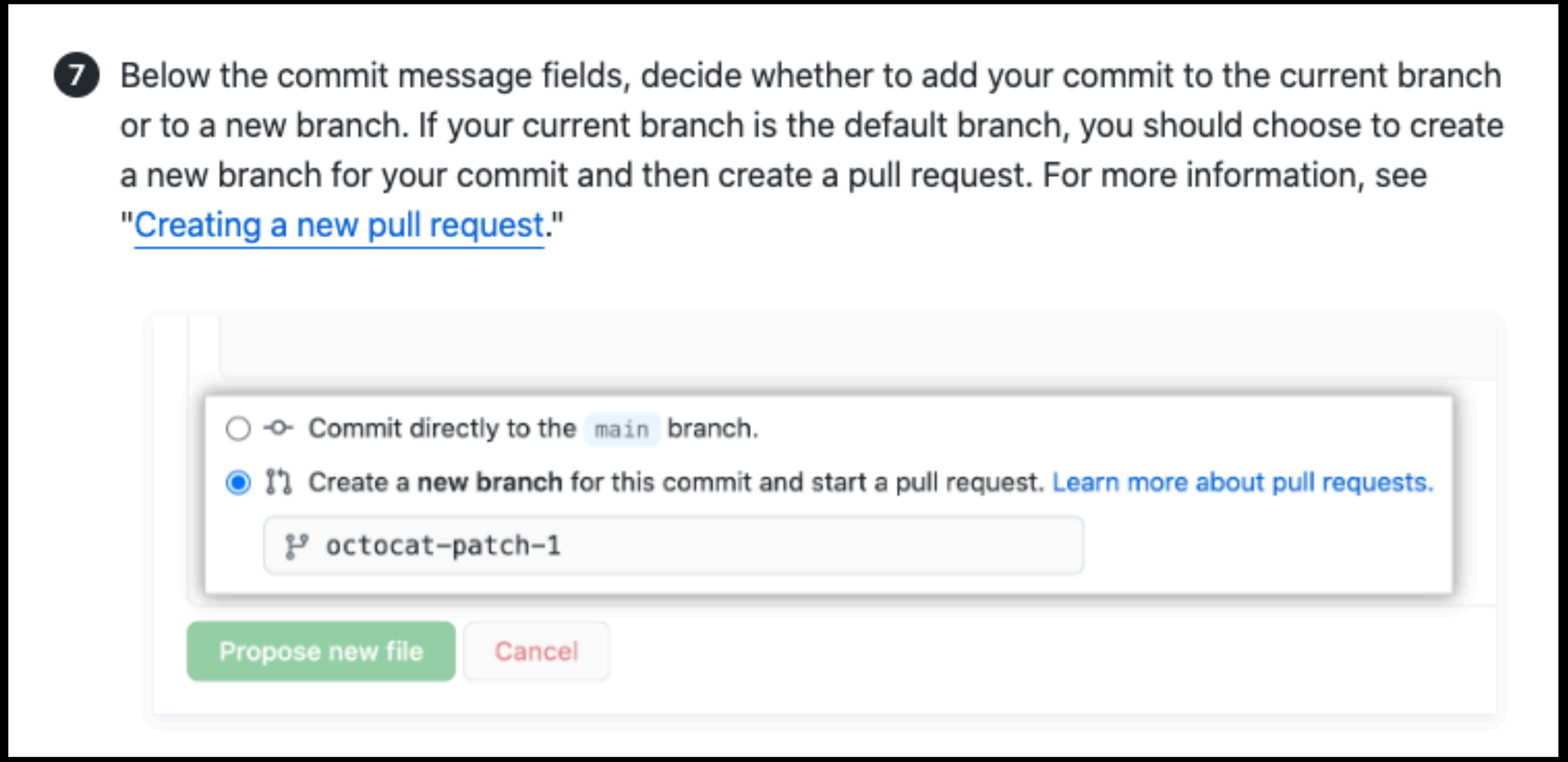
Commit your first change

A *commit* is like a snapshot of all the files in your project at a particular point in time.



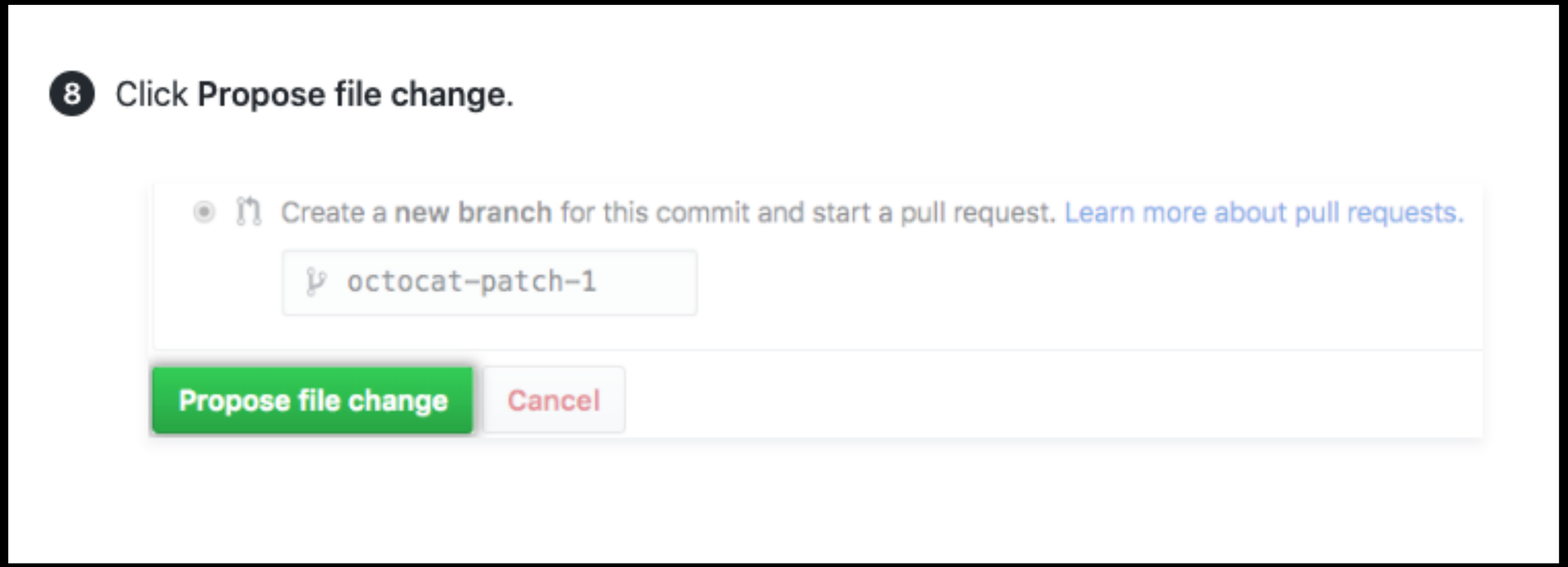
Commit your first change

A *commit* is like a snapshot of all the files in your project at a particular point in time.



Commit your first change

A *commit* is like a snapshot of all the files in your project at a particular point in time.



Next Step

- You can now clone a GitHub repository to create a local copy on your computer. From your local repository you can commit, and create a pull request to update the changes in the upstream repository. For more information, see "Cloning a repository" and "Set up Git."

- You can find interesting projects and repositories on GitHub and make changes to them by creating a fork of the repository. Forking a repository will allow you to make changes to another repository without affecting the original. For more information, see "Fork a repository."

- Each repository on GitHub is owned by a person or an organization. You can interact with the people, repositories, and organizations by connecting and following them on GitHub. For more information, see "Be social."

- GitHub has a great support community where you can ask for help and talk to people from around the world. Join the conversation on GitHub Community.

About writing and formatting on GitHub

- GitHub combines a syntax for formatting text called GitHub Flavored Markdown with a few unique writing features.

- Markdown is an easy-to-read, easy-to-write syntax for formatting plain text.

- GitHub have added some custom functionality to create GitHub Flavored Markdown, used to format prose and code across our site.

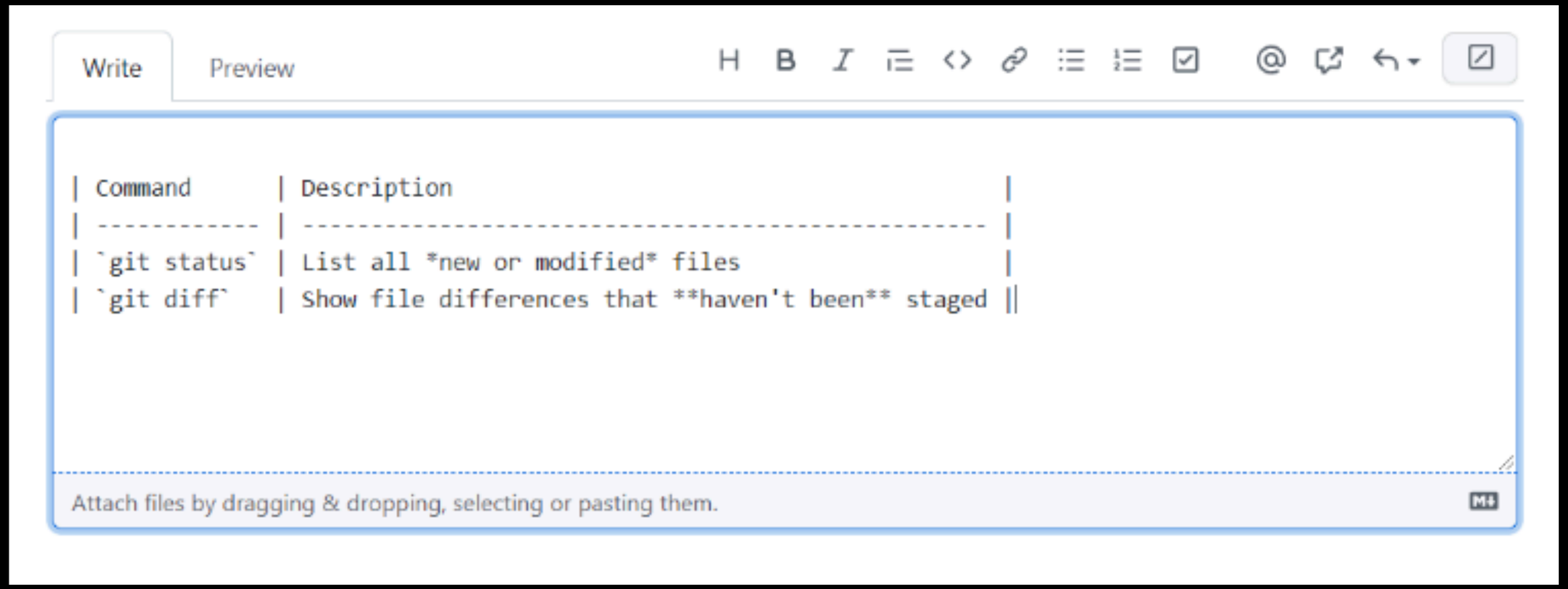
- You can also interact with other users in pull requests and issues using features like @mentions, issue and PR references, and emoji.

Text formatting toolbar

Every comment field on GitHub contains a text formatting toolbar, allowing you to format your text without learning Markdown syntax. In addition to Markdown formatting like bold and italic styles and creating headers, links, and lists, the toolbar includes GitHub-specific features such as @mentions, task lists, and links to issues and pull requests.

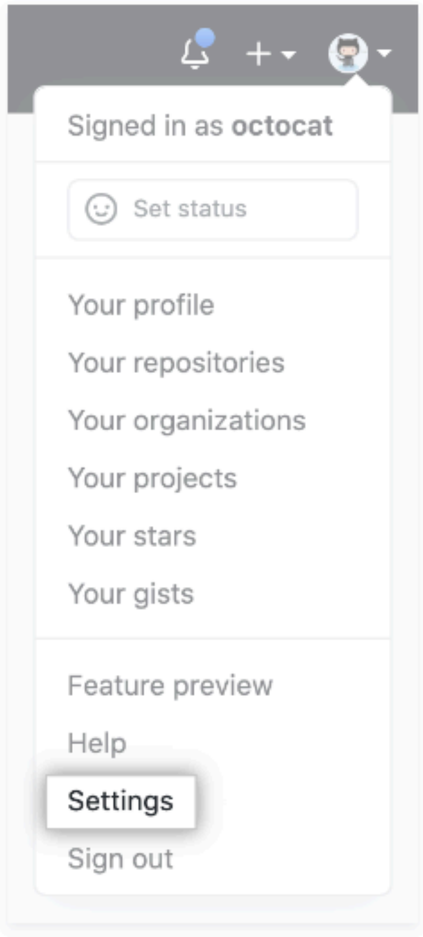
Enabling fixed-width fonts in the editor

You can enable a fixed-width font in every comment field on GitHub. Each character in a fixed-width, or monospace, font occupies the same horizontal space which can make it easier to edit advanced Markdown structures such as tables and code snippets.



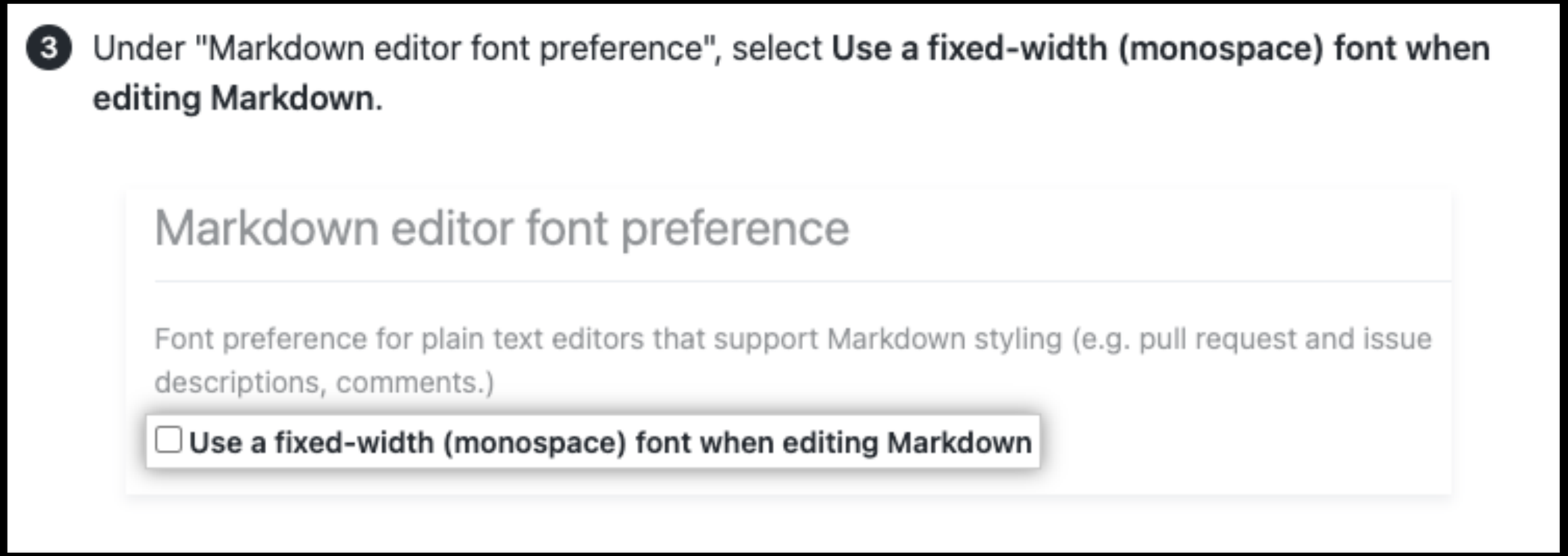
Enabling fixed-width fonts in the editor

1. In the upper-right corner of any page, click your profile photo, then click **Settings**.



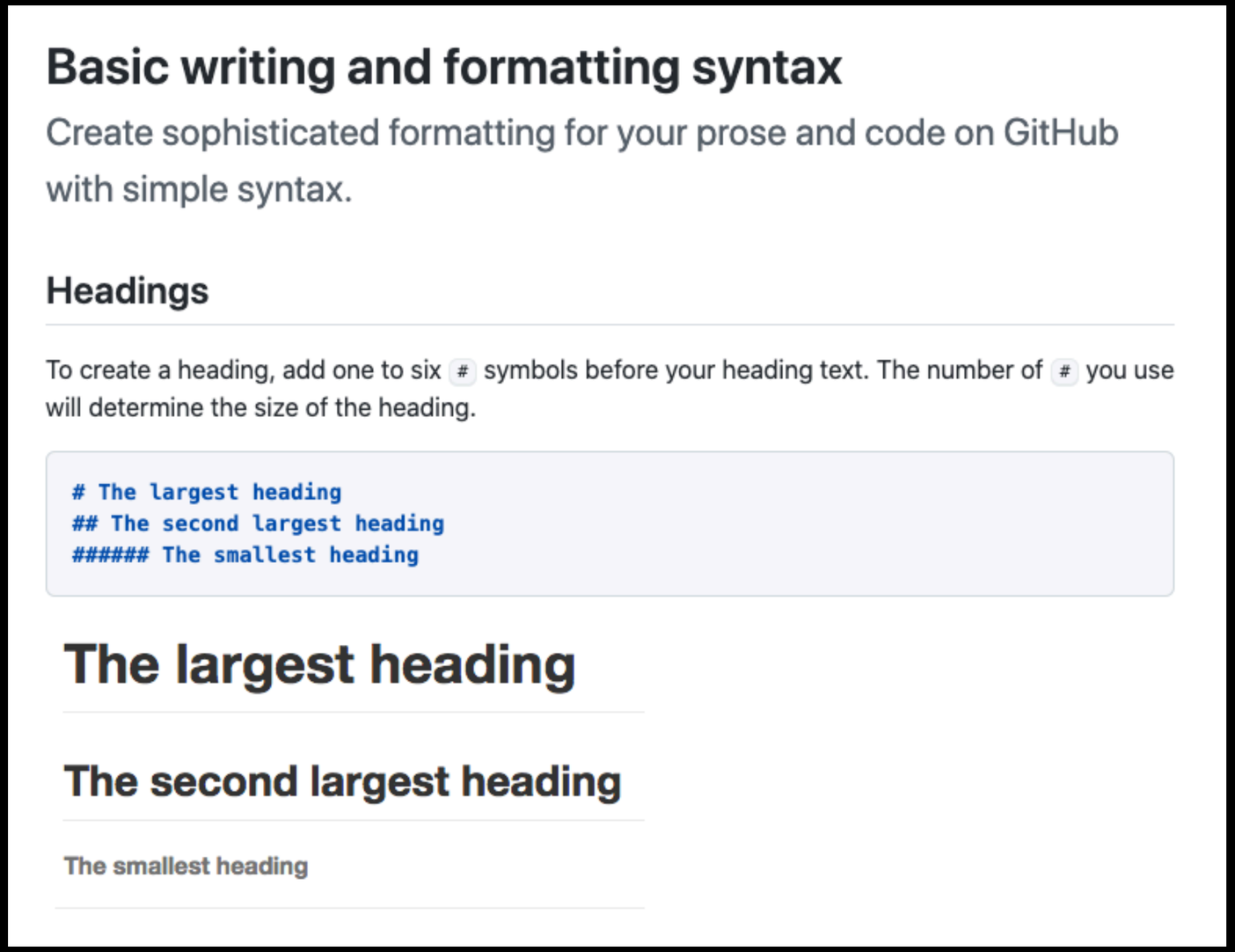
2. In the left sidebar, click **Appearance**.

Enabling fixed-width fonts in the editor



Basic writing and formatting syntax

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



Basic writing and formatting syntax

Create sophisticated formatting for your prose and code on GitHub 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
```

The largest heading

The second largest heading

The smallest heading

Basic writing and formatting syntax

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

Exercise: Use the link below and try all the different writing and formatting syntax in this article in your own readme.md file!

https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax

Diffrent interfaces

Atom

Text editors, also called code editors, are applications used by developers to write code. They highlight and format your code so that it’s easier to read and understand. If you’ve used Codecademy, you’re already familiar with a text editor! It’s the area you write your code in.

Text editors provide a number of advantages to web developers:

- Language-specific syntax highlighting
- Automatic code indentation
- Color schemes to suit your preferences and optimize code readability
- Plug-ins to catch errors in the code
- A tree view of your project’s folders and files, so you can conveniently navigate your project
- Key shortcuts for faster development

Atom

There are a number of text editors to choose from. Atom and Sublime Text are two of the most popular text editors used by developers.

Sublime Text has been the text editor of choice for many years. It is stable and reliable.

Atom was released by GitHub after Sublime Text. It’s a fully customizable text editor. Since Atom is written in HTML, CSS, and JavaScript, you can customize it yourself once you’ve learned those languages.

Either text editor is great for development, so you can’t make a bad decision here. When you are further along in your coding career, try another code editor to see what features work well with your workflow.

Atom

Create a dev folder

Below are the steps you need to follow to create a new folder for all of your programming projects. You will also learn how to load a new project folder into Atom. For steps 1 and 2, navigate to a folder using Finder (Mac users) or My Computer (PC users).

Navigate to a folder you visit regularly and create a new folder called projects. On Mac, this may be your User account. On PC, you may want to save this on your C drive.

Inside the projects directory, create a new folder called HelloWorld. Everything you add to this folder will be part of your HelloWorld project.

Open Atom on your computer.

Atom provides a tree view of your project, so you can conveniently navigate to different folders and files. In the Atom menu bar, choose File > Add Project Folder. This will launch your file manager. Navigate to the HelloWorld folder and select Open. The folder will open in Atom's side pane. At this point, there should not be any contents in the folder. We'll add a file in the next exercise.

Atom

Add a file

Once Atom loads a project folder, you can add files. The steps below describe how to add files.In Atom’s top bar, choose File > New File. An untitled, blank file will appear.

1. In Atom’s top bar, select File > New File. An untitled, blank file will appear.
2. In Atom’s top bar, choose File > Save or Save As. Name the file with its appropriate file extension (.html, .css, .csv). It is critical that you include the correct file extension, so programs know how to interpret its contents.
3. Begin coding! Save your file often. This will decrease the chances of losing unsaved work.

Atom

Add a file

Exercise III: Add a file.

In this exercise, you will create an index.html file in your Hello World project.

In Atom’s top bar, choose File > New File. An untitled, blank file will appear.

Before you save the file, copy and paste the following boilerplate HTML code:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <h1>Hello World</h1>
  </body>
</html>
```

.

Notice: All of the text in your file is the same color. This will change after you save the file as .html.

3. In Atom’s top bar, choose File > Save or Save As.

4. Name the file index.html. It’s crucial that you use the file extension .html so the text editor and web browser know how to interpret your code.

Atom

File Extensions and Syntax Highlighting

Exercise IV: Open your HTML File in a web browser

At this point, your file is ready to be viewed in a web browser. The following steps should be taken outside of Atom:

1. Back in your file system, navigate to the index.html file in your Hello World folder.
2. Double click index.html. the page should open in your default web browser.

Pycharm

Pycharm

- PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.
- Exercise: please follow the instructions of https://www.jetbrains.com/help/pycharm/run-for-the-first-time.html to run your own pycharm project!

Pycharm Exercise: write your script of transfering the pgxseg file to Dataframe, upload your python file to GitHub and describe your script in readme.md using Markdown

Paper reading: The Progenetix oncogenomic resource in 2021

Pycharm

- What is CNV/CNA?
- How will you describe or introduce progenetix (scale, data source, cancer types and so on)?
- Describe NCIt, ICOD, UBERON codes, and their relationships.
- What are CNV segmentations and CNV frequencies, and how to use them?
- What are APIs and how to use APIs in progenetix?
- How does progenetix visualise CNA profiles?
- What do you think should be improved in progenetix?