

You need internet access for this activity

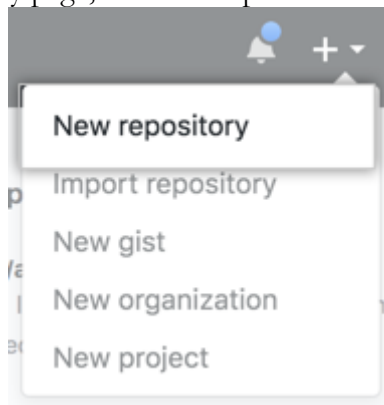
GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.

Work is saved in repositories

A repository is usually used to organize a single project. Repositories can contain folders and files, images, videos, spreadsheets, and data sets -- anything your project needs. Often, repositories include a *README* file, a file with information about your project. *README* files are written in the plain text Markdown language. GitHub lets you add a *README* file at the same time you create your new repository. GitHub also offers other common options such as a license file, but you do not have to select any of them now.

Your hello-world repository can be a place where you store ideas, resources, or even share and discuss things with others.

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



In the **Repository name** box, enter your repository

In the **Description** box, write a short description.


Select **Add a README file (optional)**.

Select whether your repository will be **Public** or **Private**.

Click **Create repository**.

Owner *

Repository name *

 octocat ▾

 /


hello-world ✓

Great repository names are short and memorable. Need inspiration? How about **ubiquitous-system**?


Description (optional)

My first repository

☐

 **Public**
Anyone on the internet can see this repository. You choose who can commit.

☒


 **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set  **main** as the default branch. Change the default name in your [settings](#).

Create repository

Creating a branch

Branching lets you have different versions of a repository at one time.

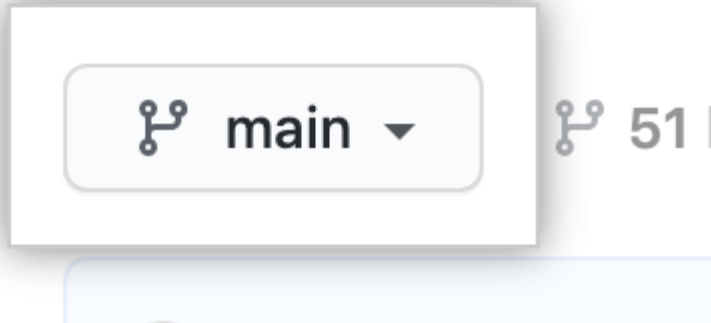
By default, your repository has one branch named **main** that is considered to be the definitive branch. You can create additional branches off of **main** in your repository. You can use branches to have different versions of a project at one time. This is helpful when you want to add new features to a project without changing the main source of code. The work done on different branches will not show up on the main branch until you merge it, which we will cover later in this guide. You can use branches to experiment and make edits before committing them to **main**.

When you create a branch off the **main** branch, you're making a copy, or snapshot, of **main** as it was at that point in time. If someone else made changes to the **main** branch while you were working on your branch, you could pull in those updates.

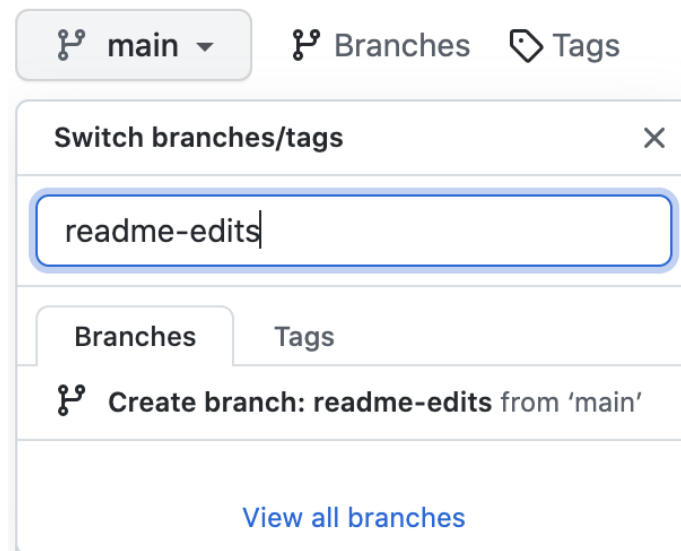
Here at GitHub, our developers, writers, and designers use branches for keeping bug fixes and feature work separate from our **main** (production) branch. When a change is ready, they merge their branch into **main**.

Create a branch

1. Click the **Code** tab of your hello-world repository.
2. Click the drop down at the top of the file list that says **main**.



3. Type a branch name, `readme-edits`, into the text box.
4. Click **Create branch: readme-edits from main**.



Now you have two branches, `main` and `readme-edits`. Right now, they look exactly the same. Next you'll add changes to the new branch.

Course work 1

Required

1. Create a GitHub account if you don't have one
2. Attempt Lab Activity 6 and upload the code to your main branch
3. Add **glubanga** as a collaborator
4. Merge the pull requests and upload the final copy of your work to the elearning platform