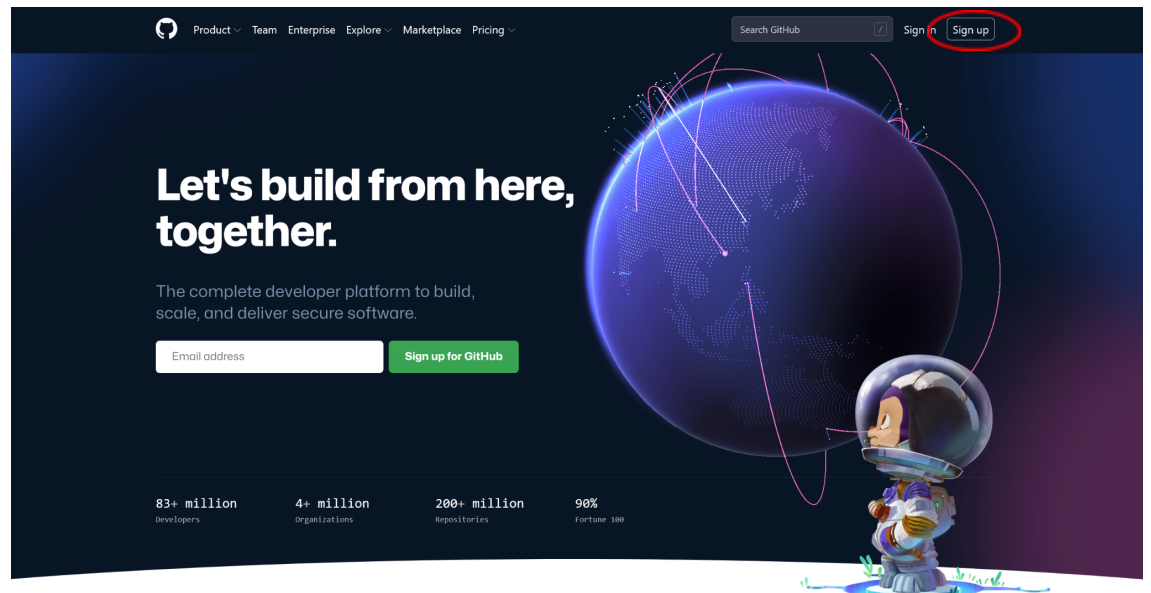


Git is a version control system created by Linus Torvalds in 2005. Using Git, developers can collaborate easily through branching and merging. **GitHub** offers Internet hosting service for software development and version control using Git. Getting your academic/personal projects on github is a way to build your own portfolio and it makes the recruiters see your accomplishments more directly!

Step 1: sign up & set up repository

1. Sign up or sign in:
 - a. Go to github's official website and click sign up on the top right if you don't already have an account.

<https://github.com/>



- b. Enter email, passwork, etc to create the account:

```
Welcome to GitHub!
Let's begin the adventure

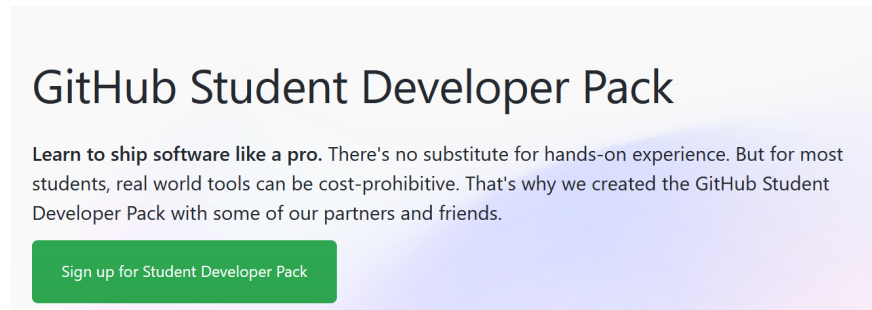
Enter your email
✓ abczyr@gmail.com

Create a password
✓ ●●●●●●●●●●

Enter a username
✓ samanthazhangxy

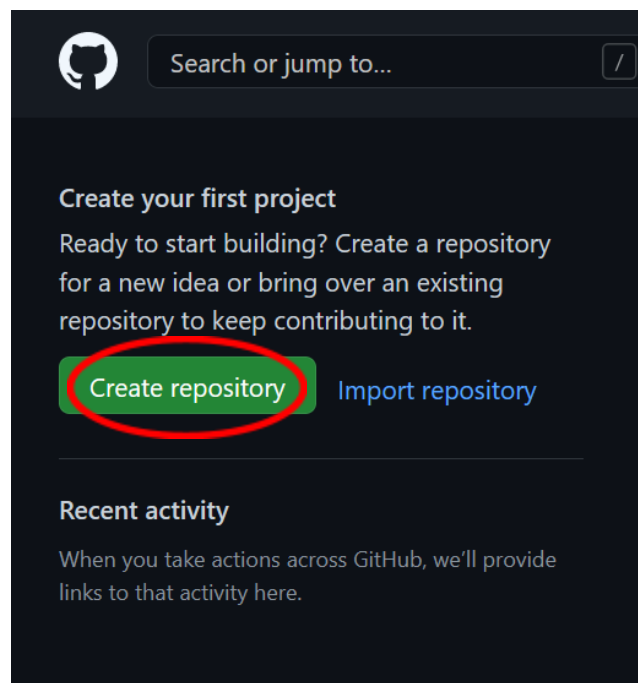
Would you like to receive product updates and announcements via email?
Type "y" for yes or "n" for no
→ n Continue
```

- c. As UC Berkeley students, we are eligible to get the student developer pack which includes free [GitHub Pro](#) while you are a student, many [GitHub Student Developer Pack](#) partner offers, and [GitHub Campus Expert training](#) for qualified applicants.



Apply with your berkeley email address.

2. At the landing page after sign in/sign up, the left side bar shows your recent github repositories. These can be the ones that you created or it can be the ones that other people added you to as a collaborator.
- a. Now we will create a new repository by clicking on the create repository/new button.



- b. It's always good to have something in your repository, so we can check the Add a README file, which basically acts as an abstract for your project. Also, when we create a new repository, we can always link our local files to the repository and the following tutorial will show you how to achieve that.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner ^{*} samanthazhangxy / Repository name ^{*} githubtest ✓

Great repository names are short and memorable. Need inspiration? How about [special-spork?](#)

Description (optional)

☒ **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.](#)

.gitignore template: None

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

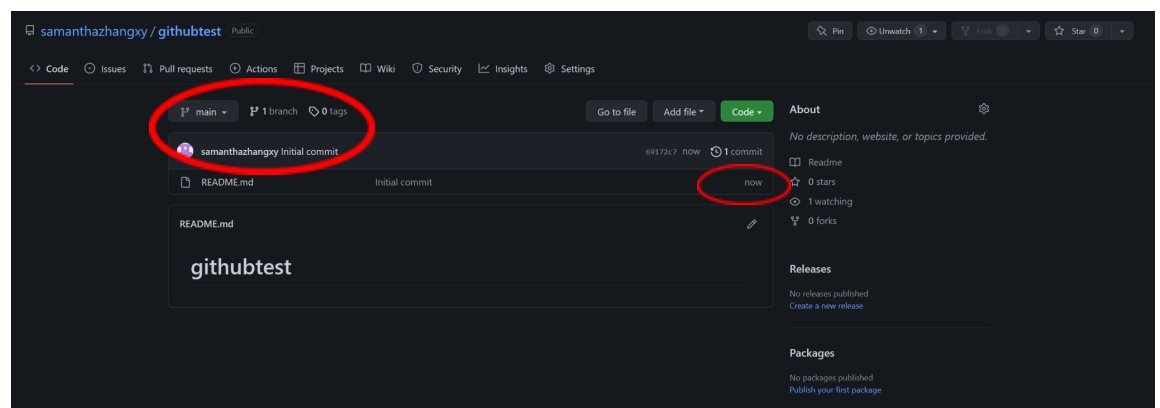
License: None

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

ⁱ You are creating a public repository in your personal account.

[Create repository](#)

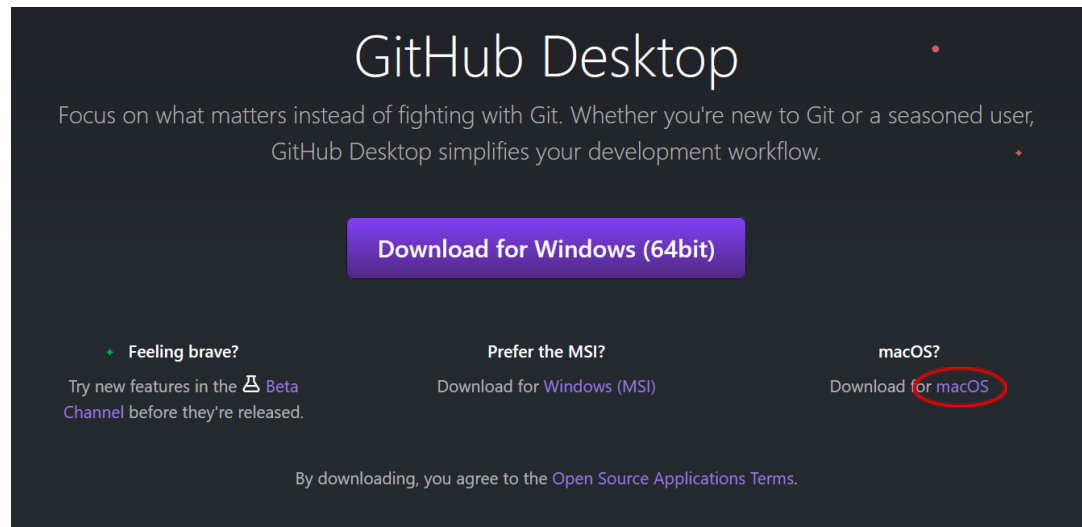
- c. Now, we can see that we have a new github test repo set up. This is the central portal of the repo where we can manage our branches, files, commits!



Step 2: Download Github Desktop:

1. To manage our local file in github, we can download github desktop for easier access.

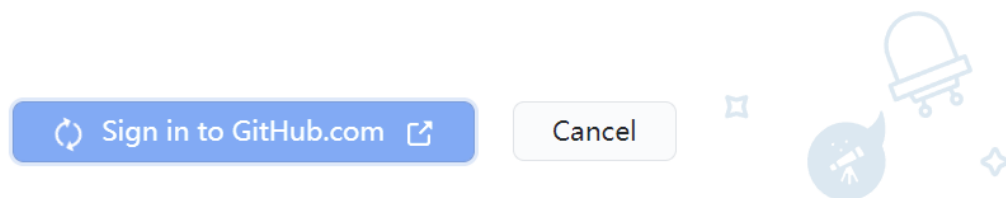
- a. Download at <https://desktop.github.com/> for both Windows and MacOS:



2. After installing github desktop, click sign in and authorize the access of your github account through desktop.

Welcome to GitHub Desktop

Your browser will redirect you back to GitHub Desktop once you've signed in. If your browser asks for your permission to launch GitHub Desktop please allow it to.



3. Going back to the Github Desktop, we are asked to do some configuration. GitHub Desktop allows you to set the name and email address you would like associated with the commits you make in your repositories. If your name and email address have already been set in the global Git configuration for your computer, GitHub Desktop will detect and use those values. GitHub Desktop also allows you to set a different name and email address for an individual repository. This is useful when you need to use a separate work email address for a specific repository.
- In the **Name** field, type the name you'd like to use for your Git configuration.
 - Select the **Email** dropdown and click the email address you would like to use for your commits.

Configure Git

This is used to identify the commits you create. Anyone will be able to see this information if you publish commits.

- ☒ Use my GitHub account name and email address
☐ Configure manually

Name

samanthazhangxy

Email

abczxr@gmail.com

Finish

Cancel

Example commit

Fix all the things

samanthazhangxy • 30 minutes ago

Step 3: Clone a github repository to local:

1. After we successfully set up github desktop, we can clone our test repo in github to our local computer.

Let's get started!

Add a repository to GitHub Desktop to start collaborating

Create a tutorial repository...

Clone a repository from the Internet...

Create a New Repository on your hard drive...

Add an Existing Repository from your hard drive...

Filter your repositories

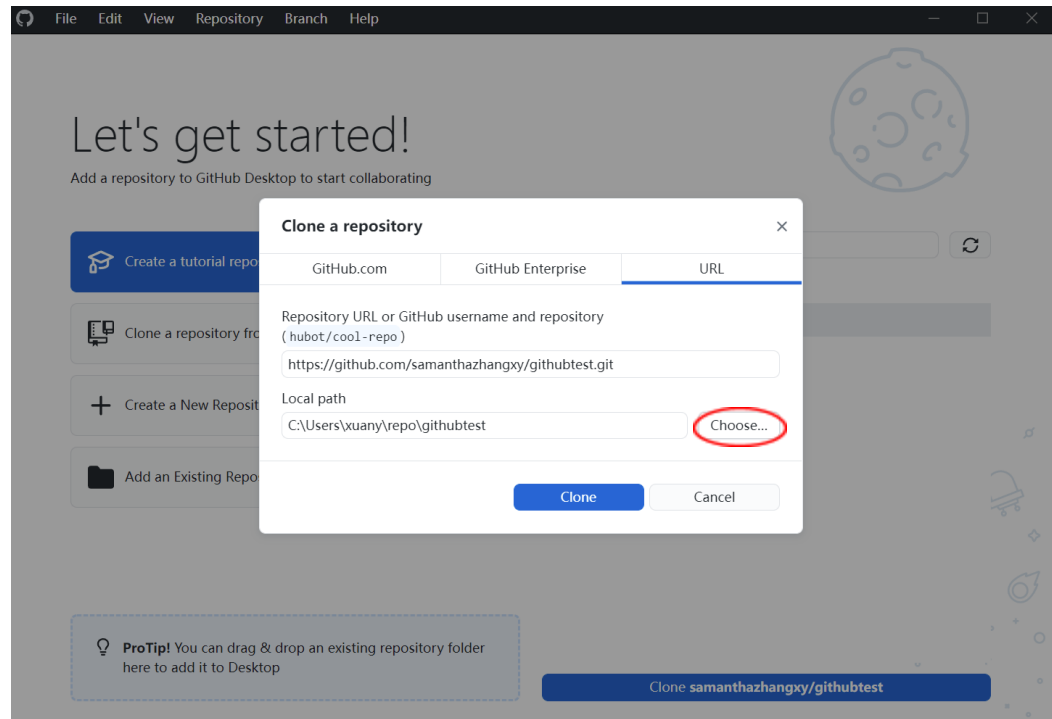
Your repositories

samanthazhangxy/githubtest

ProTip! You can drag & drop an existing repository folder here to add it to Desktop

Clone samanthazhangxy/githubtest

- a. We can choose to create a new local repository or if you already have a local project repo, you can place it in the local repo instead.



Step 4: Making changes in the local repo and commit to github:

1. Now that we have our test repo cloned to local, we can start working on our projects in the local test repo clone. A great practice is to always have your class projects published online, so in this tutorial, we are going to show that.
- a. We can first create folders for hw, labs, and projects for the class that we want to have on github. We can create those folders locally or through Git Bash for Windows or Terminal for MacOS.

hw	2022/8/16 11:20
labs	2022/8/16 11:17
projects	2022/8/16 11:16
README.md	2022/8/16 11:13

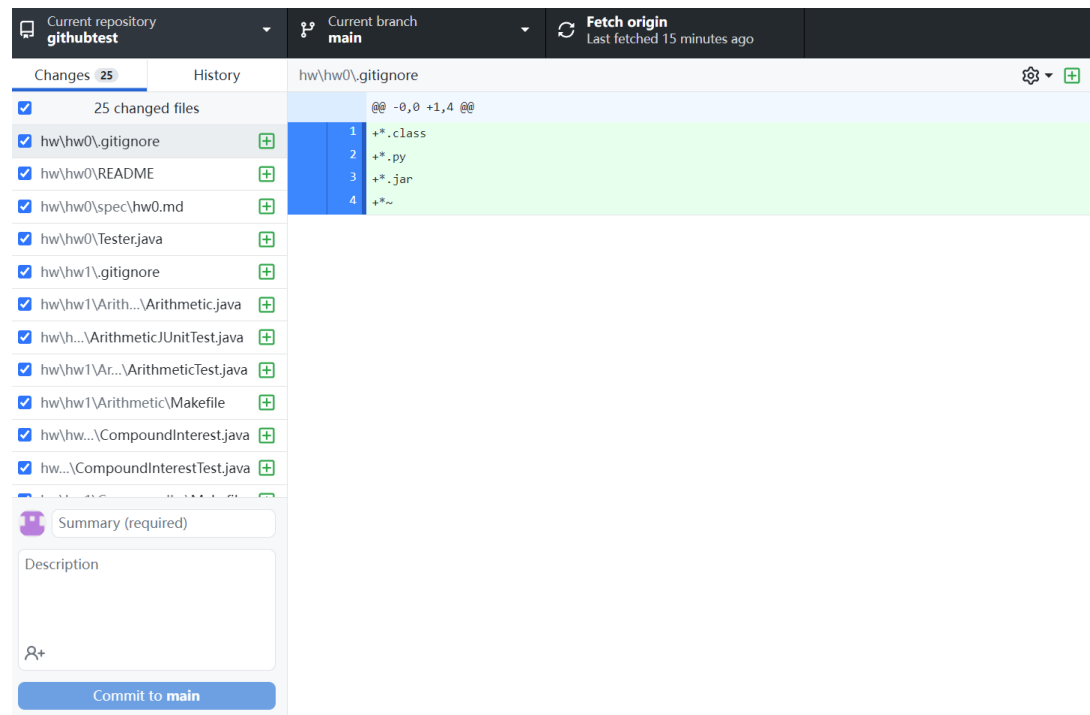
- b. After we create the folders, we can copy and paste our past work into these folders accordingly. Here, I copied two of my 61b hws into the hw folder.

hw0	2022/8/16 11:20
hw1	2022/8/16 11:20

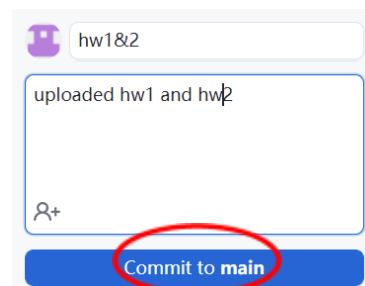
- c. Now go back to github desktop, we can see all of the changes we made. Git doesn't record empty folders created, so the labs and projects folders we created are not shown here.

Any additions we have in our code are shown in green, and deletions are shown in red.

The changes bar functions as status in Git, where all of the changes are shown in the staging area but not yet committed or pushed to our cloud github repository.



- d. As we know from using Git, it's always good to make commits in time whenever we make changes to our project files so that we have freedom to go back and forth between versions and won't risk losing our updates. So in order to commit, we need to give it a name in the Summary box, and we can add a description too, if we want. Once we are ready, hit the commit to the main button.



- e. After we have done all the changes we want in our local repo and committed all of the changes, we can push them to the origin remote. These commits will now be reflected in our githubtest remote repository.

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.



Push commits to the origin remote

You have 1 local commit waiting to be pushed to GitHub.

Always available in the toolbar when there are local commits waiting to be pushed or **Ctrl** **P**

Push origin

- f. Go back to our github online repo. We can see the commit we just made and track down which file was changed through the commit as well.

