Cubstart Web Lab 4

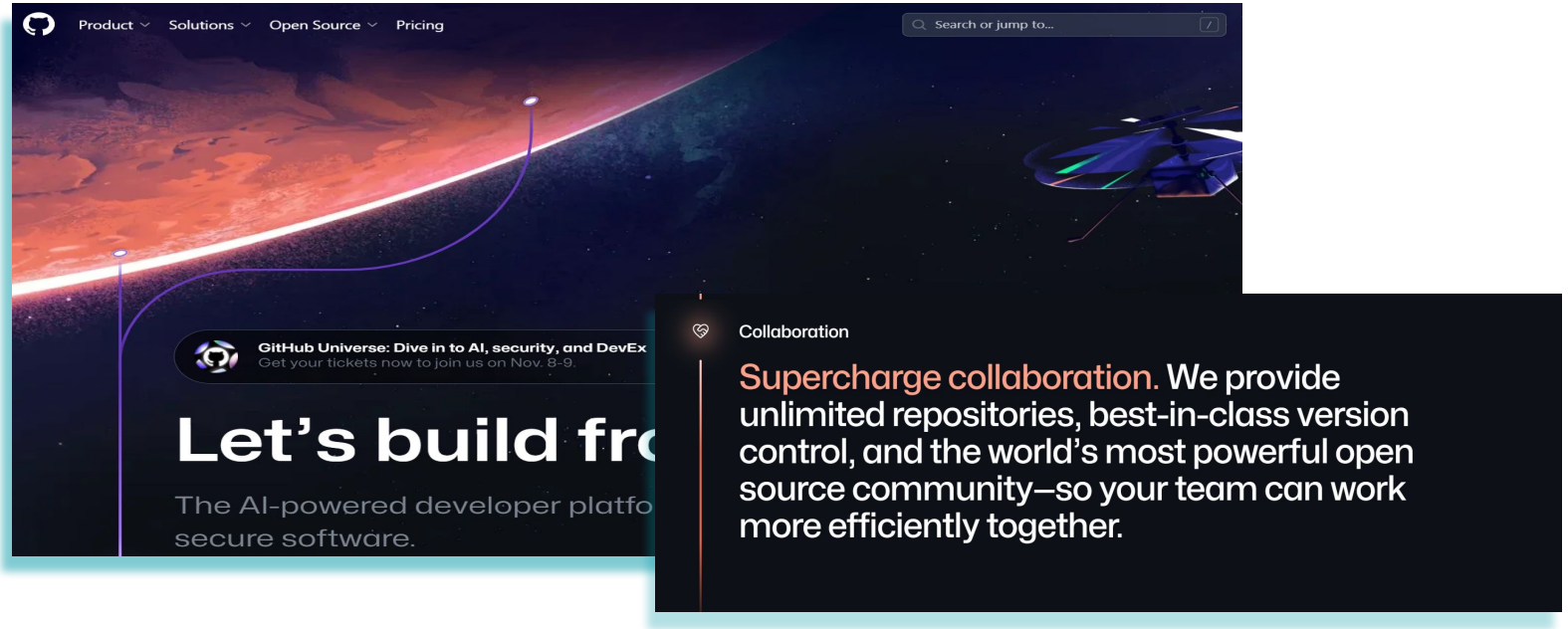
Administrivia

1. HW 3: JavaScript Playground due tonight
2. HW 4 is a Gradescope assignment, due in one week
3. These slides (Lab 4) are on cubstart.com so you can follow along!

This lab we will be learning how to use **Git**, **GitHub**, and deploying a website to **GitHub Pages.** For those of you who have used Git and GitHub before, the first half might be a little slow but please follow along and hang tight until the deployment section!

GitHub

GitHub is a powerful platform that simplifies and enhances collaboration, version control, and project management for software development



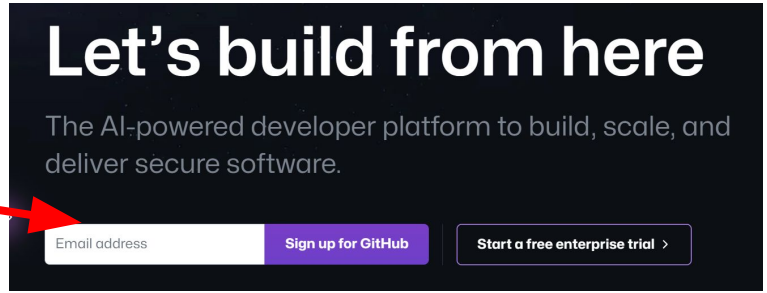
Create an Account

Step 1: Go to **https://github.com/**

Step 2: Type in your email address and click **Sign Up**

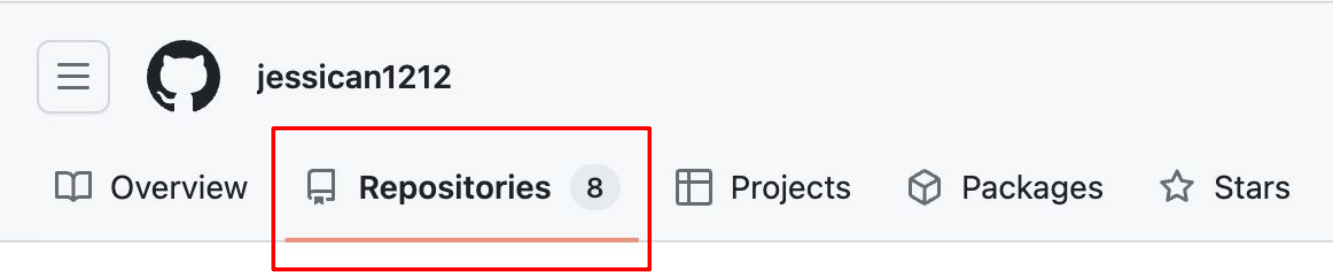
Step 3: Click **Continue & type in** your **password & username**

Step 4: Follow the instructions on GitHub to set up your account



Repositories

Repository: central location that **stores a collection of files** and their revision history

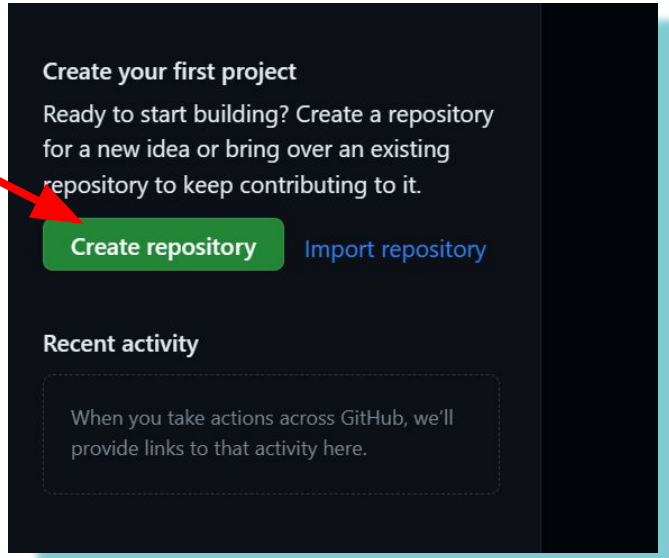


You typically store each of your projects (personal website, hackathon web application, etc.) in their own repository.

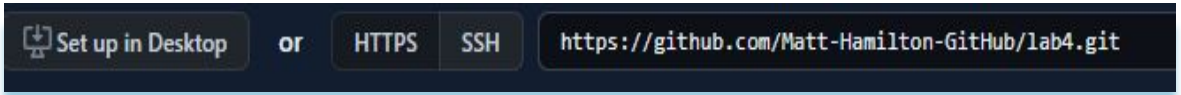
Create Repository on GitHub

Step 1: Click **Create Repository** (on dashboard)

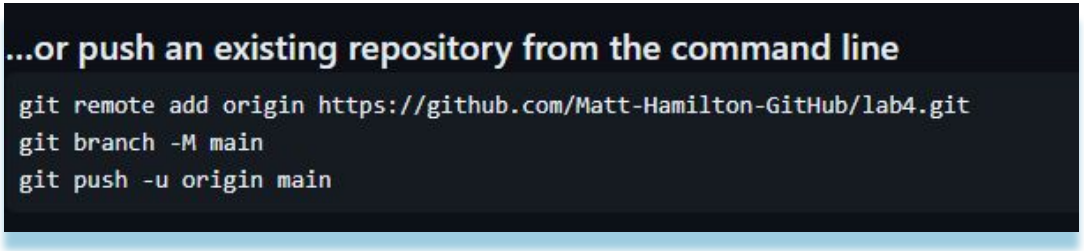
Step 2: Write **repository name** & choose type (Private or Public) & click **Create Repository**



Now you are able to see your repository URL



And some suggested git commands for your terminal to push your local files to the repository



Terminal

Command Terminals

MacOS Terminal:

- Navigate to the terminal using spotlight search

Git Bash Terminal (Windows):

- Git Bash comes pre-installed with Git
- Git Bash is highly customizable
- Easier to use
- User friendly interface

Installing Git on Mac

https://github.com/git-guides/install-git

MacOS: Most versions of MacOS will already have Git installed, and you can activate it through the terminal by typing in **git version**

***https://desktop.github.com/ If that doesn't work, installing **GitHub Desktop** will automatically install the latest version of Git if you don't already have it

Installing Git Bash on Windows

Step 1: Go to https://git-scm.com/downloads

Step 2: Choose Windows OS

Step 3: Choose **32bit** or **64bit** depending on you system (most of the time it's **64bit**)

Step 4: Install with the **Recommended** settings

Basic Command:

Navigation:

- Move into a folder: **cd folderName**
- Go back to previous: **cd ..**
- Create folder: **mkdir folderName**
- Create file: **touch index.html**

Other commands:

- View files at the current path**: ls**
- Clear terminal: **clear**

Create Lab Folders & Files

Create lab folder: **mkdir lab4**
Navigate to lab4: **cd lab4**

Create css and html files:

-> touch index.html

-> touch styles.css

OR we can write in one line by separating file names with a space:

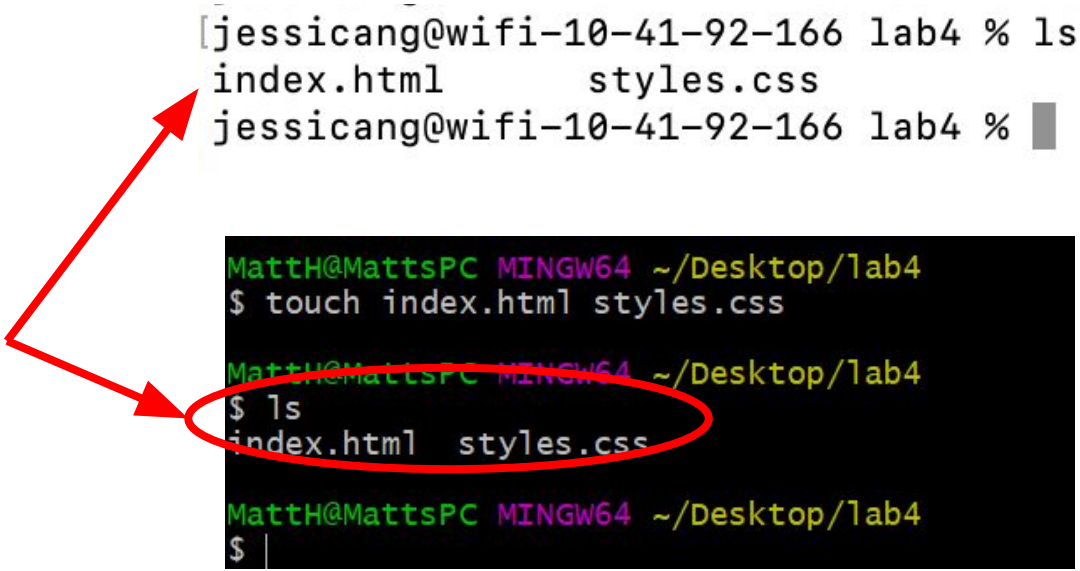
-> touch index.html styles.css

```
jessicang@wifi-10-41-92-166 ~ % cd Desktop
jessicang@wifi-10-41-92-166 Desktop % mkdir lab4
jessicang@wifi-10-41-92-166 Desktop % cd lab4
jessicang@wifi-10-41-92-166 lab4 % touch index.html
jessicang@wifi-10-41-92-166 lab4 % touch styles.css
```

Create Lab Folder & Files

Check your desktop (or wherever you created your html and css files)... they should be there!

Another way to check is in terminal. You can see the files in the folder by running **ls**



Push Files to the Repository

-> **git init**
(initializes a new Git repository in the current directory)

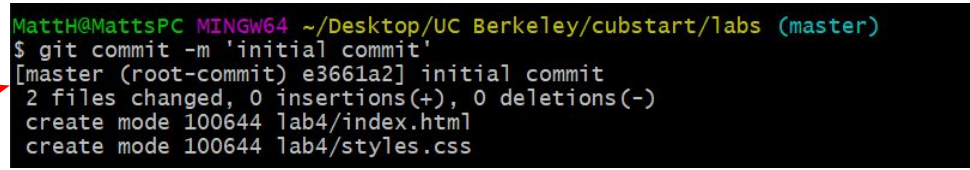
-> **git remote add origin** https://github.com/GitHub/cubstart-labs.git
(connects your respostry to your local folder)

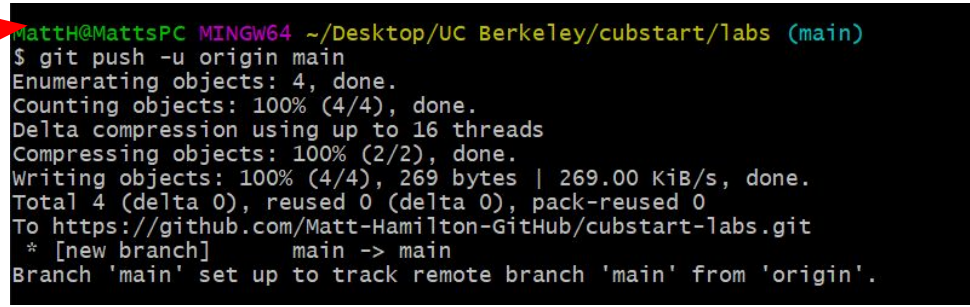
-> **git add .**
(Stages all changes in the current directory)

-> **git commit -m 'message'**
(Records a snapshot of the changes that have been staged)

-> git **branch** -M main
(Renames the default branch from its original name to main)

-> **git push -u origin main**
(updates the repository with all the changes committed)





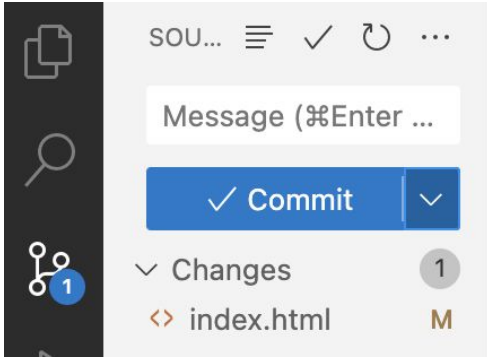
Modifying Files

Terminal...

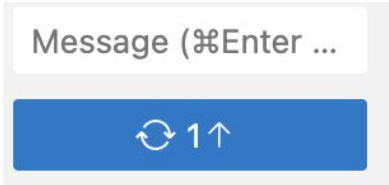
1. Make changes to your files in VS Code
2. Open the terminal and navigate to your project directory
3. -> **git add .**
4. -> **git commit -m “MESSAGE”**
5. -> **git push**

... or use VS Code:

1. Go to **Source Control** on left bar
2. Type in a message



3. Click **Commit**
4. Click this button to push changes to your repo

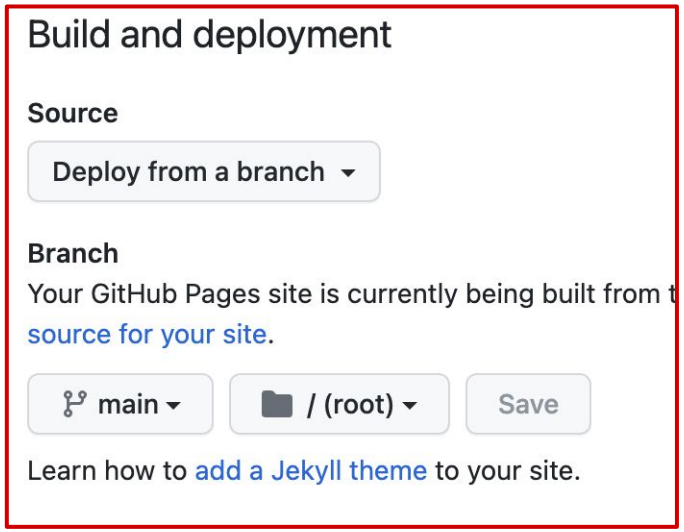


Publish/Deploy your site using GitHub Pages!

Configure a Publishing Source

You can configure your GitHub Pages site to publish when changes are pushed to a specific branch!

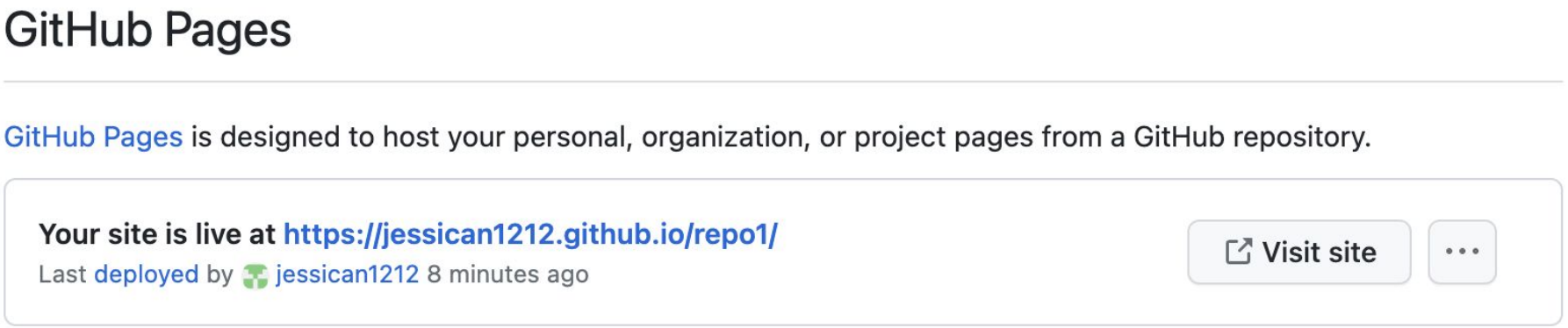
1. On GitHub, go to your site's repository
2. Go to **Settings**, then go to **Pages** on the left bar
3. Make your **Build and Deployment** section looks like this:



4. Click **Save**. This has kicked off the pages-build-deployment workflow
5. Go to **Actions** to view the progress of your pages-build-deployment process

Viewing the site

After a few seconds, this will appear at the top of **Settings** -> **Pages**



You can now send this link to ***anyone*** you want so they can view your site!
(previously, you could only view your site locally on your local computer)

Making changes

Anytime you add, commit, and push changes to your GitHub repo, GitHub Pages will **automatically re-publish** your site to the same link!