

# Cubstart Web

## Lab 4

# Administrivia

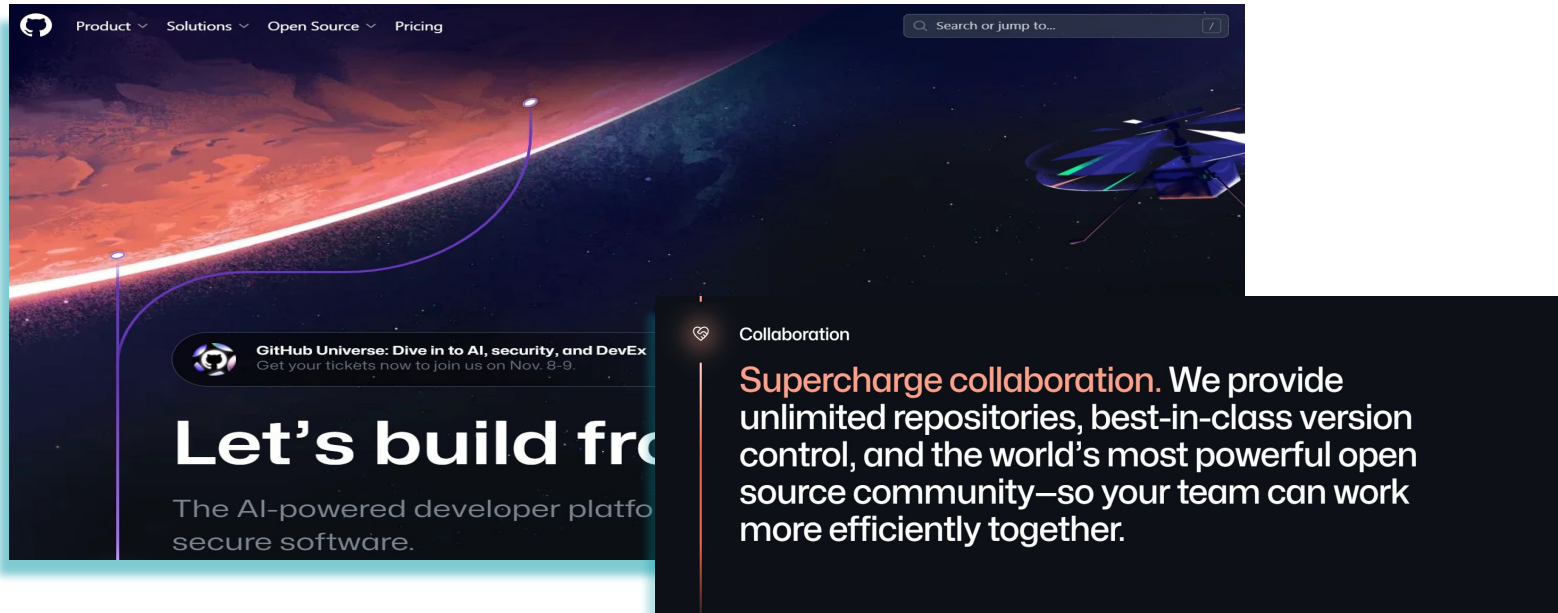
1. HW 3: JavaScript Playground due tonight
2. HW 4 is a Gradescope assignment, due in one week
3. These slides (Lab 4) are on [cubstart.com](https://cubstart.com) so you can follow along!

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



The screenshot shows the GitHub homepage with a dark, space-themed background. At the top, there is a navigation bar with links for Product, Solutions, Open Source, and Pricing, along with a search bar. The main content area features a large banner with the text "Let's build for" and "The AI-powered developer platform to build secure software." Below this, there is a section titled "Collaboration" with the text "Supercharge collaboration. We provide unlimited repositories, best-in-class version control, and the world's most powerful open source community—so your team can work more efficiently together."

Product Solutions Open Source Pricing

Search or jump to...

GitHub Universe: Dive in to AI, security, and DevEx  
Get your tickets now to join us on Nov. 8-9.

**Let's build for**

The AI-powered developer platform to build secure software.

Collaboration

**Supercharge collaboration.** We provide unlimited repositories, best-in-class version control, and the world's most powerful open source community—so your team can work more efficiently together.



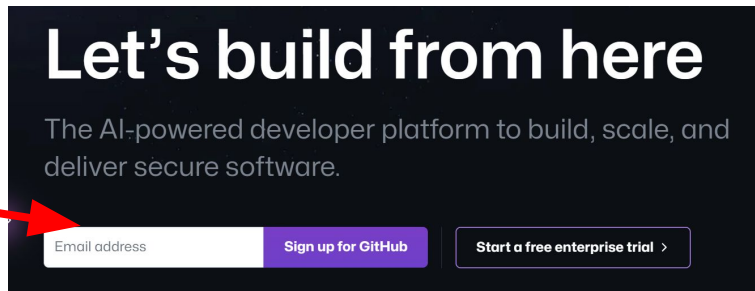
# Create an Account

**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

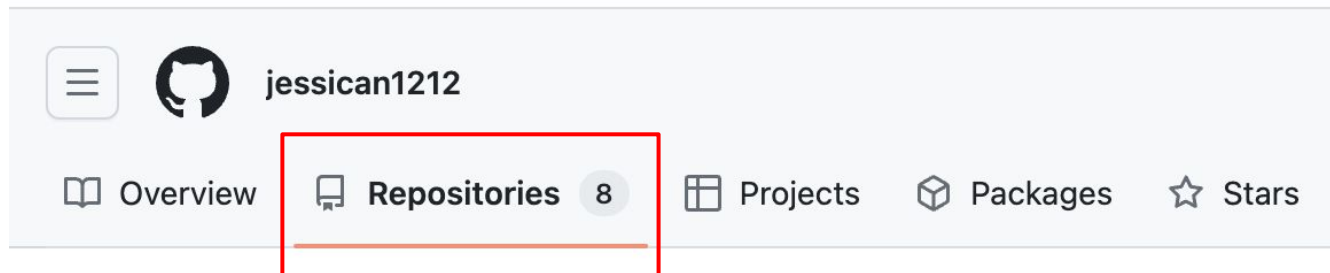


The screenshot shows the GitHub sign-up interface. At the top, it says "Let's build from here" in white text on a dark background. Below that, a subtitle reads "The AI-powered developer platform to build, scale, and deliver secure software." The main form area contains an input field labeled "Email address" with a red arrow pointing to it from the text "Type in your email address" in Step 2. To the right of the input field is a purple button labeled "Sign up for GitHub". Further right is a button labeled "Start a free enterprise trial >".



# 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.

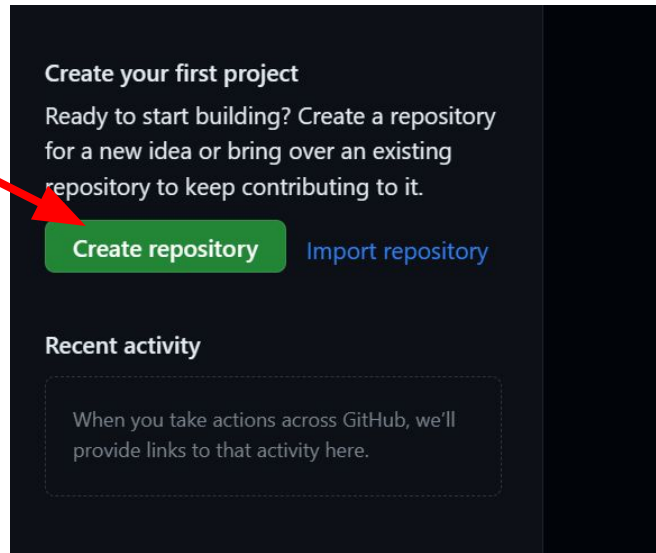


# Create Repository on GitHub

**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

...or push an existing repository from the command line

```
git remote add origin https://github.com/Matt-Hamilton-GitHub/lab4.git
git branch -M main
git push -u origin main
```



# 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 your system (most of the time it's **64bit**)

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



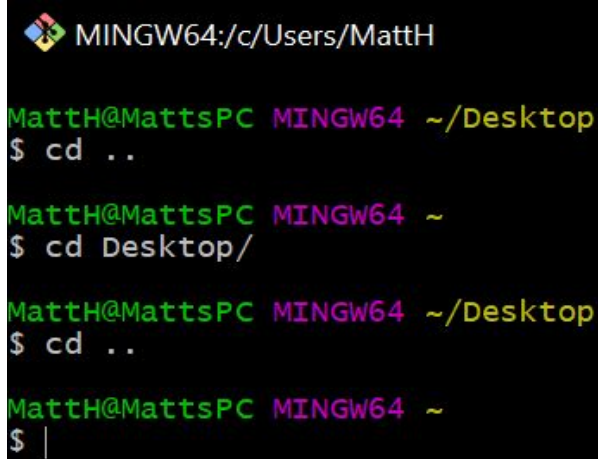
# Basic Command:

## 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**



```
MINGW64:/c/Users/MattH  
  
MattH@MattsPC MINGW64 ~/Desktop  
$ cd ..  
  
MattH@MattsPC MINGW64 ~  
$ cd Desktop/  
  
MattH@MattsPC MINGW64 ~/Desktop  
$ cd ..  
  
MattH@MattsPC MINGW64 ~  
$ |
```

# Create Lab Folders & Files

Create lab folder: **mkdir lab4**

Navigate to lab4: **cd lab4**



```
Matth@MattsPC MINGW64 ~/Desktop/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**

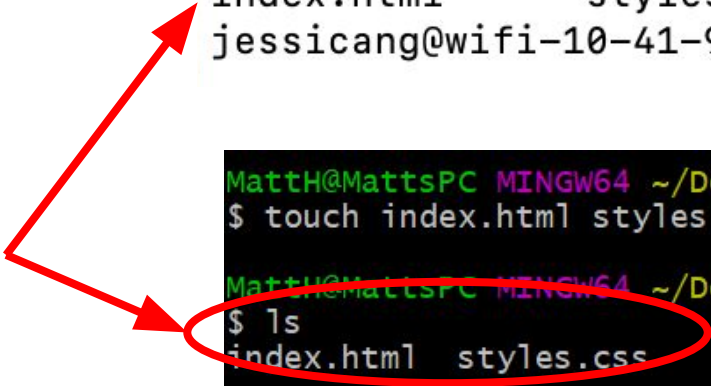
```
[jessicang@wifi-10-41-92-166 ~ % cd Desktop  
[jessicang@wifi-10-41-92-166 Desktop % mkdir lab4  
[jessicang@wifi-10-41-92-166 Desktop % cd lab4  
[jessicang@wifi-10-41-92-166 lab4 % touch index.html  
[jessicang@wifi-10-41-92-166 lab4 % touch styles.css
```

# Create Lab Folder & Files

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

```
[jessicang@wifi-10-41-92-166 lab4 % ls  
index.html      styles.css  
jessicang@wifi-10-41-92-166 lab4 %
```

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



```
Matth@MattsPC MINGW64 ~/Desktop/lab4  
$ touch index.html styles.css  
Matth@MattsPC MINGW64 ~/Desktop/lab4  
$ ls  
index.html  styles.css  
Matth@MattsPC MINGW64 ~/Desktop/lab4  
$
```



# 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 repository to your local folder)

-> **git add .**

(Stages all changes in the current directory)

-> **git commit -m 'message'**

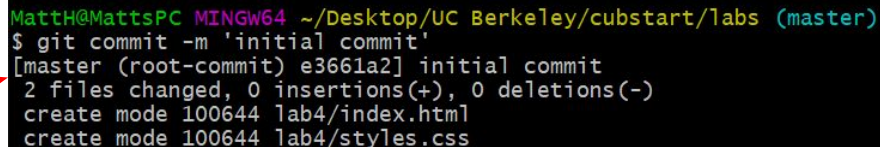
(Records a snapshot of the changes that have been staged)

-> **git branch -M main**

(Renames the default branch from its original name to main)

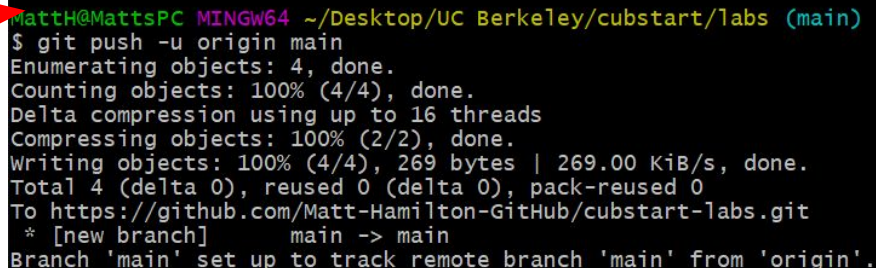
-> **git push -u origin main**

(updates the repository with all the changes committed)



Matth@MattsPC MINGW64 ~/Desktop/UC Berkeley/cubstart/labs (master)  
\$ git commit -m 'initial commit'  
[master (root-commit) e3661a2] initial commit  
2 files changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 lab4/index.html  
create mode 100644 lab4/styles.css

A red arrow points from the text "(Records a snapshot of the changes that have been staged)" to this terminal screenshot.



Matth@MattsPC MINGW64 ~/Desktop/UC Berkeley/cubstart/labs (main)  
\$ git push -u origin main  
Enumerating objects: 4, done.  
Counting objects: 100% (4/4), done.  
Delta compression using up to 16 threads  
Compressing objects: 100% (2/2), done.  
Writing objects: 100% (4/4), 269 bytes | 269.00 KiB/s, done.  
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0  
To https://github.com/Matt-Hamilton-GitHub/cubstart-labs.git  
\* [new branch] main -> main  
Branch 'main' set up to track remote branch 'main' from 'origin'.

A red arrow points from the text "(updates the repository with all the changes committed)" to this terminal screenshot.

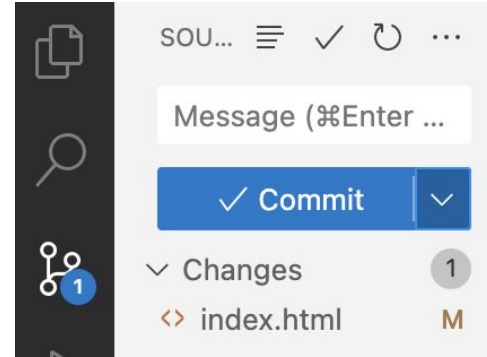
# Modifying Files

## 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!**

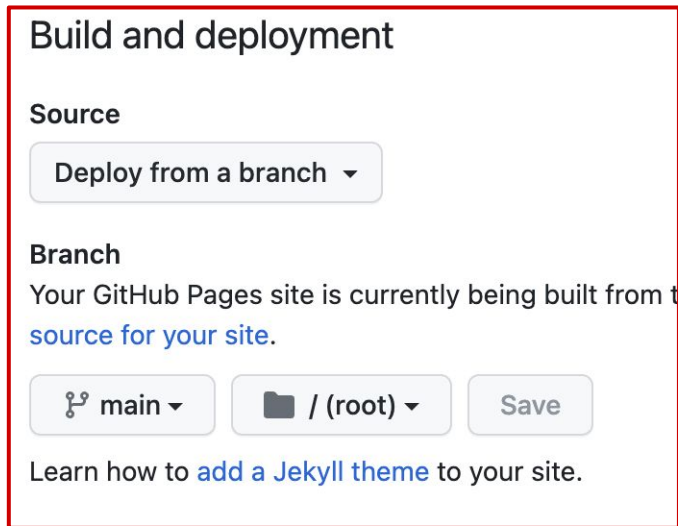




# Configure a Publishing Source

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:





Build and deployment

Source

Deploy from a branch ▾

Branch

Your GitHub Pages site is currently being built from the [source for your site](#).

 main ▾  / (root) ▾ Save

Learn how to [add a Jekyll theme](#) to your site.

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**

## GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at <https://jessican1212.github.io/repo1/>

Last deployed by  jessican1212 8 minutes ago

 Visit site

...

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!

