

# J220

## Coding for Journalists

LECTURER  
Soo Oh

### PROMPTS

Open <https://pollev.com/soooh>  
in a new window.

start Zoom recording + captions

# Agenda

Announcements

SAJA members on democracy event

Homework review + how much time

GitHub

**BREAK**

Study Hall

# Announcements

**April 24:** Career panel. Add questions in this [Google doc](#)!

- **Nancy Deville**, Editorial Project Manager at The Atlantic
- **Gabe Hongsdusit**, Visual Designer at The Markup
- **Evan Wagstaff**, Director of Newsroom Engineering at Hearst Newspapers

**May 1:** Final class: Presentations. You'll also be giving peer feedback.

**May 11: Final project due**

# Announcements

We are giving a **three day extension**. Review rubric, we need to see all of these in your wireframe.

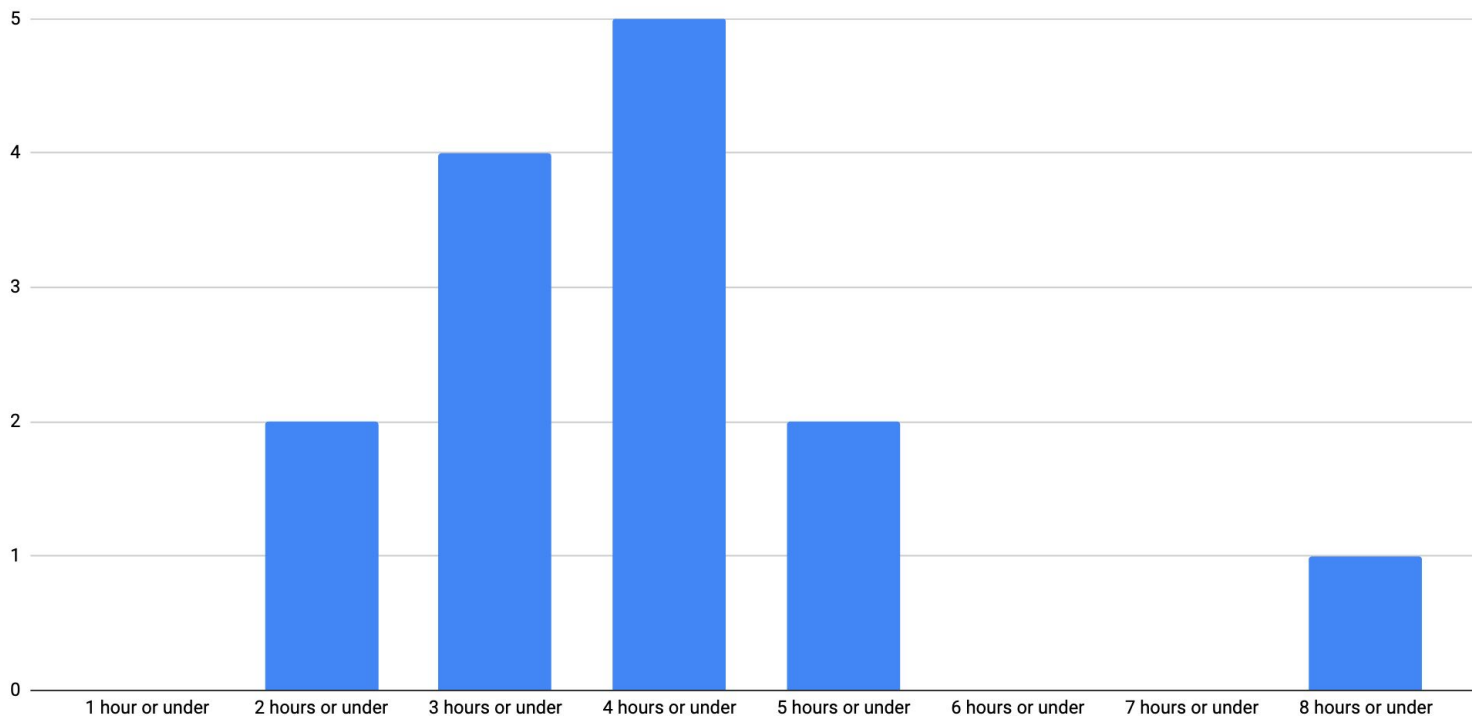
## Wireframe Specifications

Wireframe Specifications			
Criteria	Ratings		Pts
Accessibility - Color Annotate wireframe with the hex color for background, fonts, etc. that have been tested for multiple accessibility criteria	10 pts Full Marks	0 pts No Marks	10 pts
Accessibility - Text Annotate wireframe with font sizes and font family for titles, descriptions, section headers, text. Specify if using a header tag (h1, h2, h3, etc). The user must be able to scan the page visually and figure out the hierarchy of content.	10 pts Full Marks	0 pts No Marks	10 pts
Mobile Wireframe must include a mobile design	10 pts Full Marks	0 pts No Marks	10 pts
Desktop Wireframe must include a desktop design	10 pts Full Marks	0 pts No Marks	10 pts
Inspiration/Research Attach a list of at least three websites, books, photos, tv shows (anything!) that inspired part of your design. Include a 1-2 sentence description of what inspiration you drew from each piece.	10 pts Full Marks	0 pts No Marks	10 pts
Total Points: 50			

# SAJA event recap

# How much time spent on J220 last week

Week 04-10: Number of students grouped by hours spent outside of lecture and office hours



# Homework Review

[Short answer key](#)



# What questions do you have?

**This is going to  
be part of your  
homework!**

# GitHub Desktop

Installation + Publishing your portfolio online

# What is git and GitHub?

**Git** is a type of tool that allows for **version control** on your code or software. It's **revision history** for your code, kind of like [Google Docs' Version history](#), except Git doesn't save your code automatically at intervals.

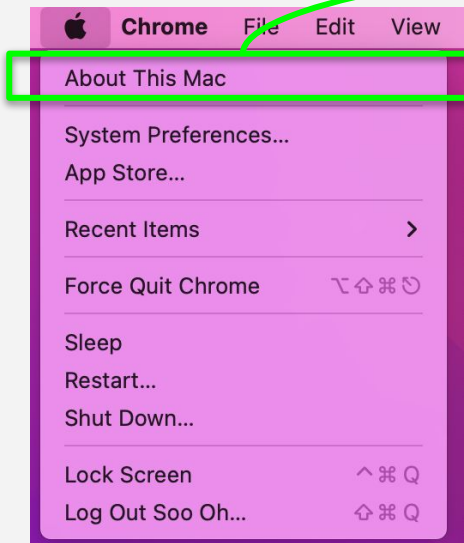
Instead, you have to **commit** (save) your code, which only happens on your computer.

Then you **push** or **publish** changes, which moves your code to a central platform that your collaborators also have access to.

There are similar kinds of version control tools for code and software, but Git is the most popular.

[GitHub.com](#) is a web-based platform that offers Git version control as a service. There are other platforms that use Git. You can even install your own version of Git onto private servers so no one outside of your organization can see them.

# Determine what kind of chip you have



1. Go to the Apple icon in the upper-left of your screen, and click on **About This Mac**.
2. Find your **Chip** or **Processor** here:



# Determine what kind of chip you have

## Apple Silicon

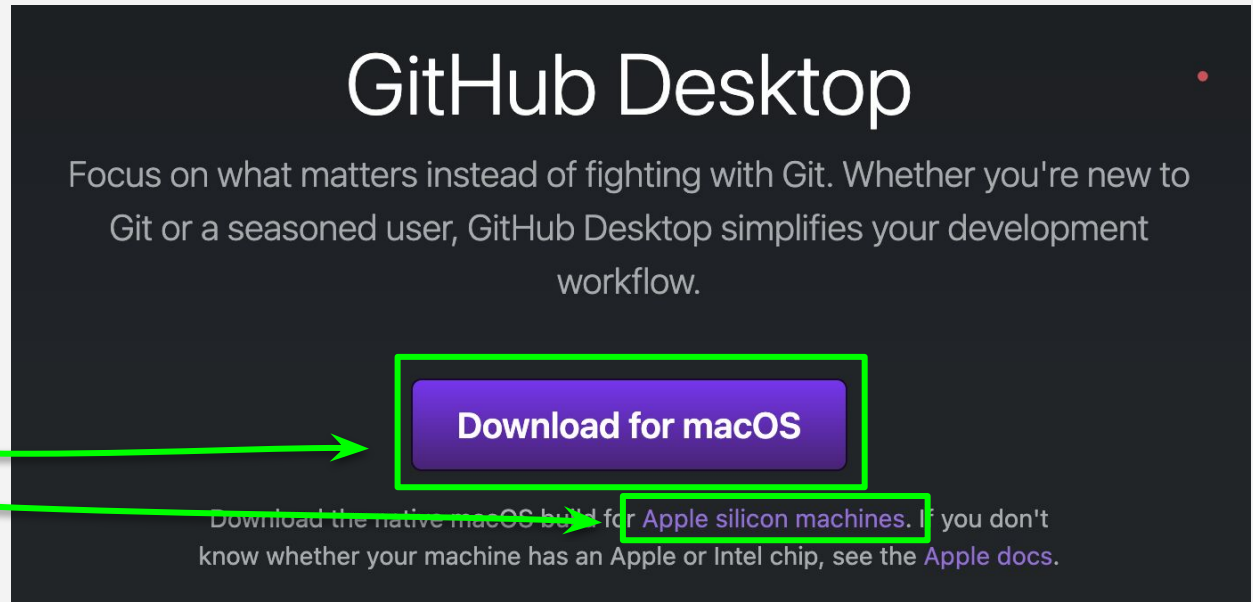


## Intel

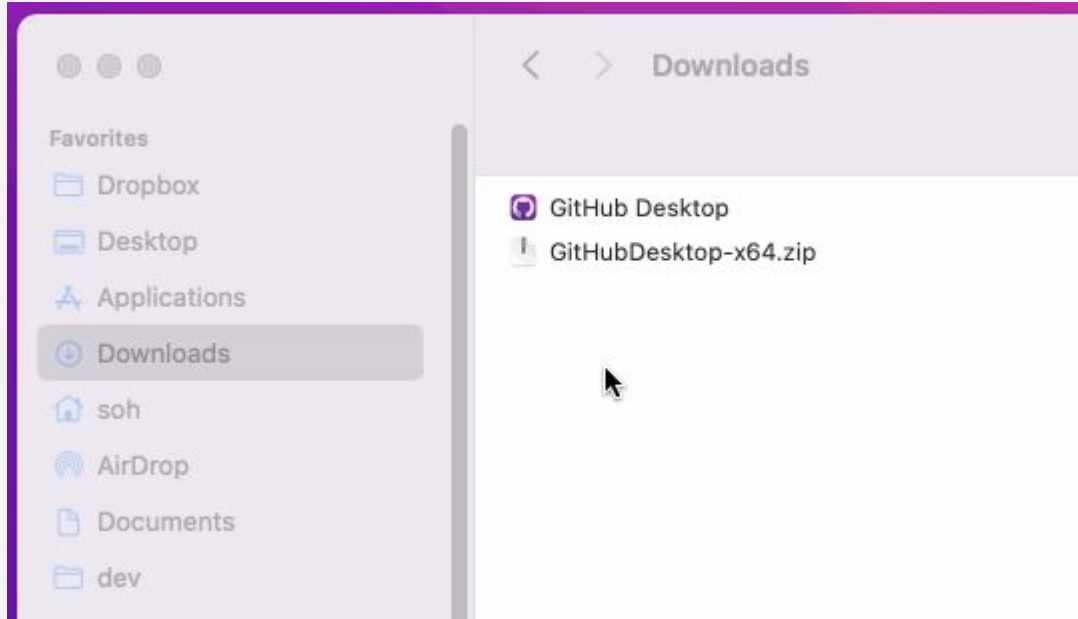


# Download GitHub Desktop

Go to <https://desktop.github.com> and download the correct version for your Macbook.

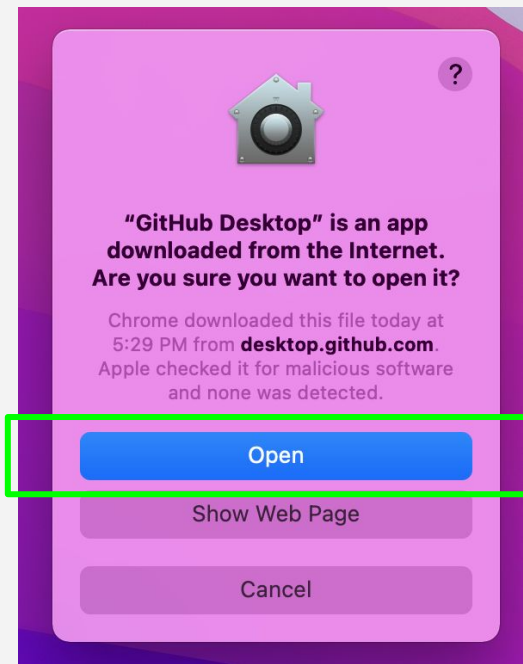


# Unzip the downloaded file...



... then drag the app to your Applications folder (don't forget this!)

# Go to your Applications folder and open the GitHub Desktop app



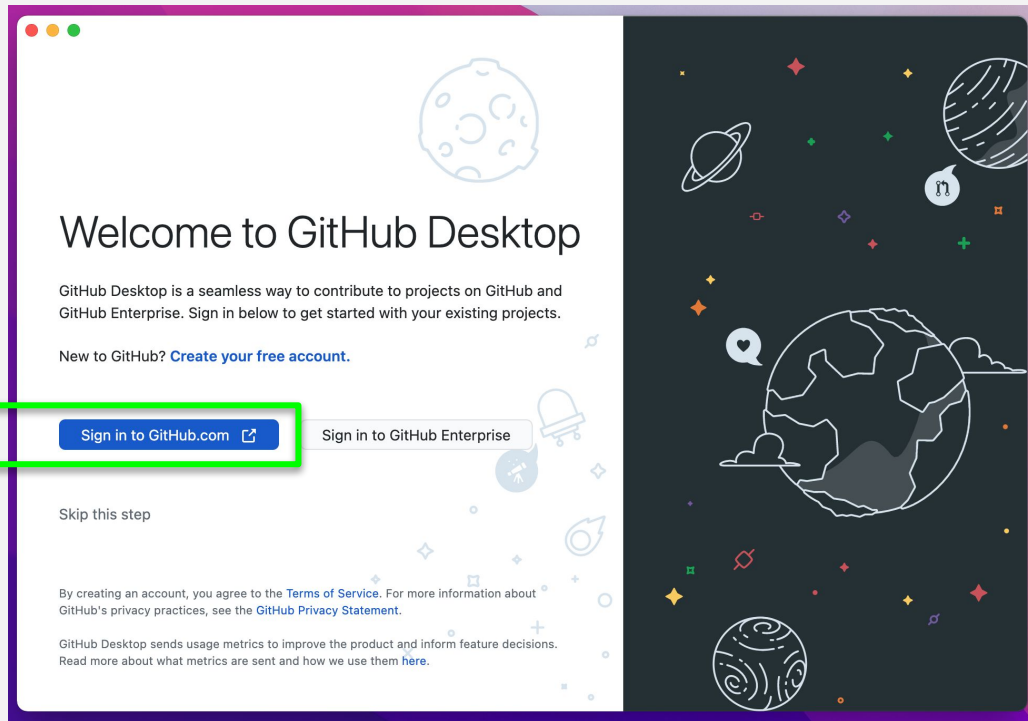
If you get this message, click on **Open**.



# Sign into GitHub.com

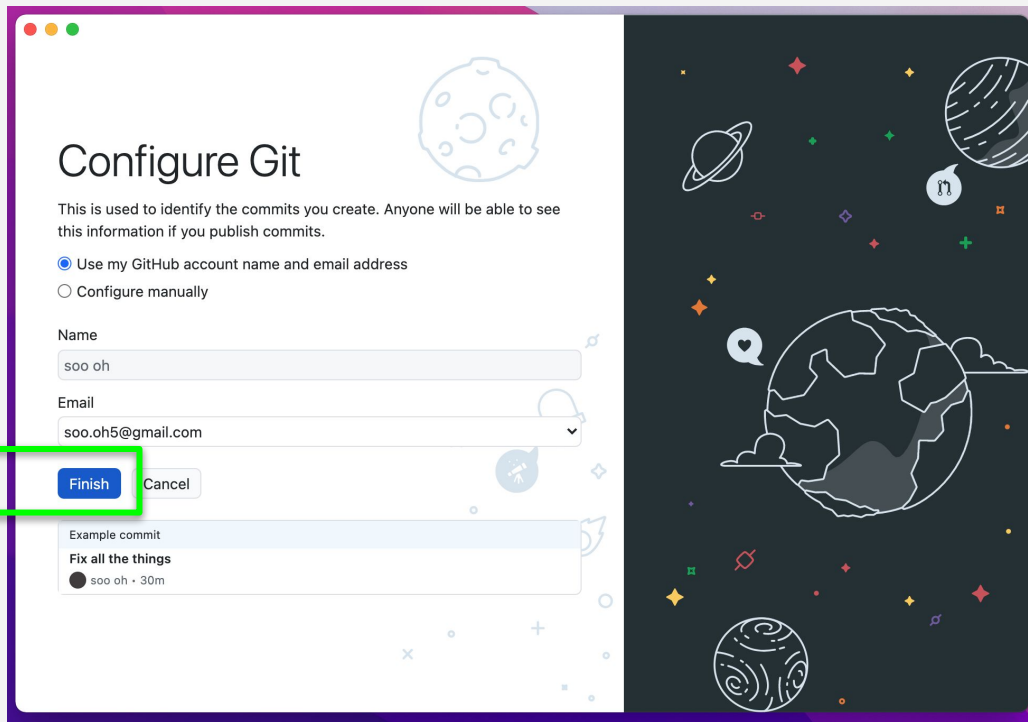
Click on the blue button.

That will open your default browser, and you'll log into GitHub.com.



# Configure Git

The default settings are fine. Click **Finish**.



The screenshot shows the 'Configure Git' window. It has a title bar with three colored dots (red, yellow, green). The window title is 'Configure Git'. Below the title is a moon icon. The main text says: 'This is used to identify the commits you create. Anyone will be able to see this information if you publish commits.' There are two radio buttons: 'Use my GitHub account name and email address' (selected) and 'Configure manually'. Below these are input fields for 'Name' (soo oh) and 'Email' (soo.oh5@gmail.com). At the bottom left, there is a blue 'Finish' button and a grey 'Cancel' button. A green box highlights the 'Finish' button. Below the buttons is an 'Example commit' section showing a commit message 'Fix all the things' by 'soo oh' 30m ago. The right side of the window has a dark background with a space-themed illustration of planets, stars, and a globe.

# Create a new repository

Click the button that says “Create a new repository on your computer.”

# Let's get started!

Add a repository to GitHub Desktop to



Create a Tutorial Repository...



Clone a Repository from the Internet



Create a New Repository on your local machine



Add an Existing Repository from your computer



**ProTip!** You can drag & drop a repository here to add it to Desktop

## Create a New Repository



Name

**username.github.io**

Description

Students: `Name` should be username.github.io

Local Path

/Users/soh/Documents/GitHub

Choose...

☒ Initialize this repository with a README

Git Ignore

None

License

MIT License

Cancel

Create Repository

To publish your site the easy way, you need to follow this exact format:

**username.github.io**

Replace *username* with your actual GitHub username!

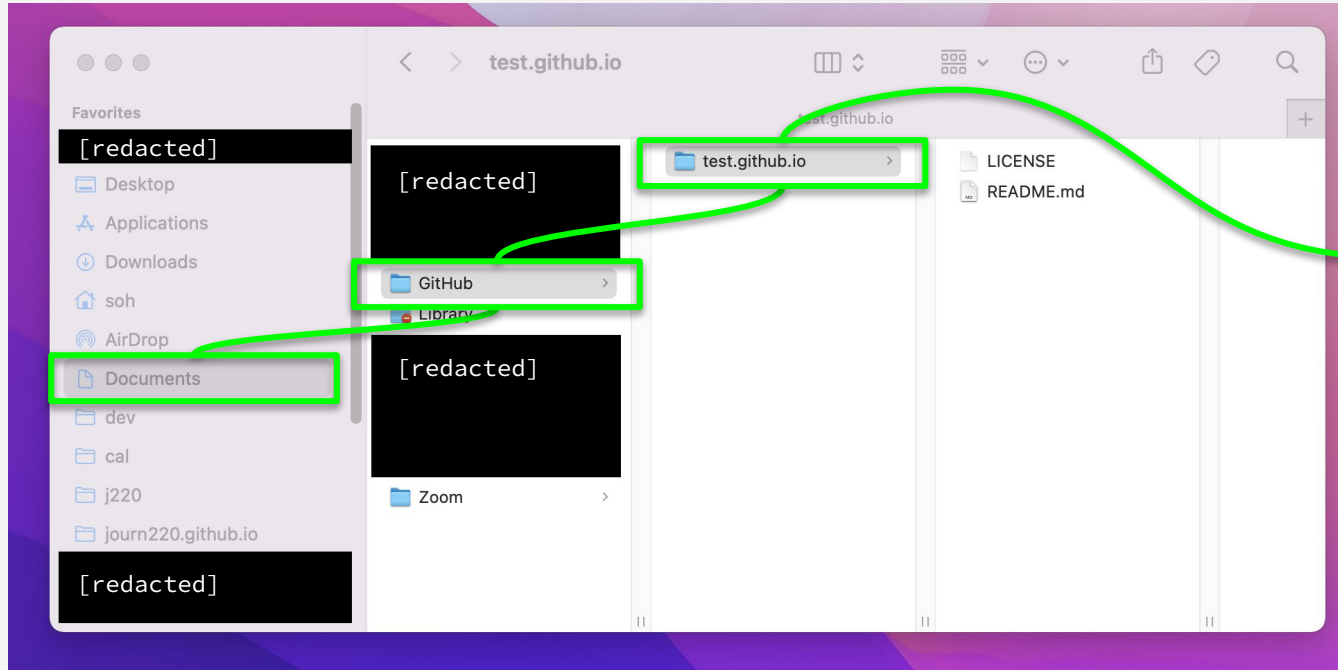
# What is your GitHub.io address?

Powered by  Poll Everywhere

Start the presentation to see live content. For screen share software, share the entire screen. Get help at [pollev.com/app](https://pollev.com/app)

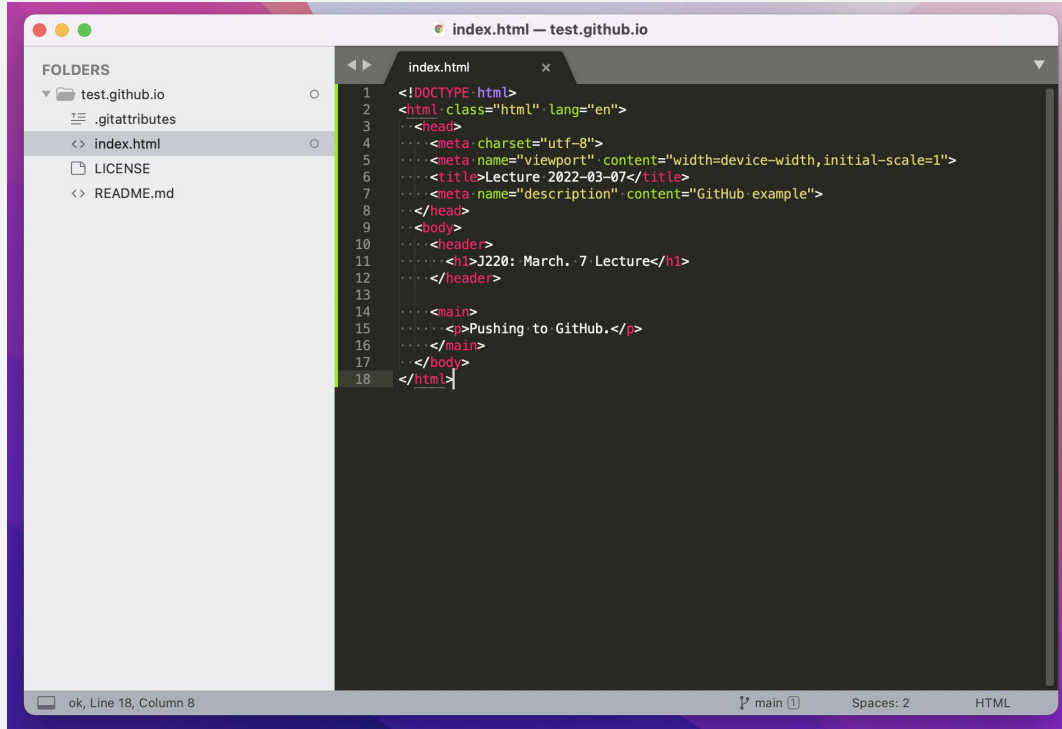


Navigate to your repo. It's in  
**Documents > GitHub > username.github.io**



Open the folder in your text editor by dragging it to **Sublime Text** (or other editor).

# Create **index.html** within your folder



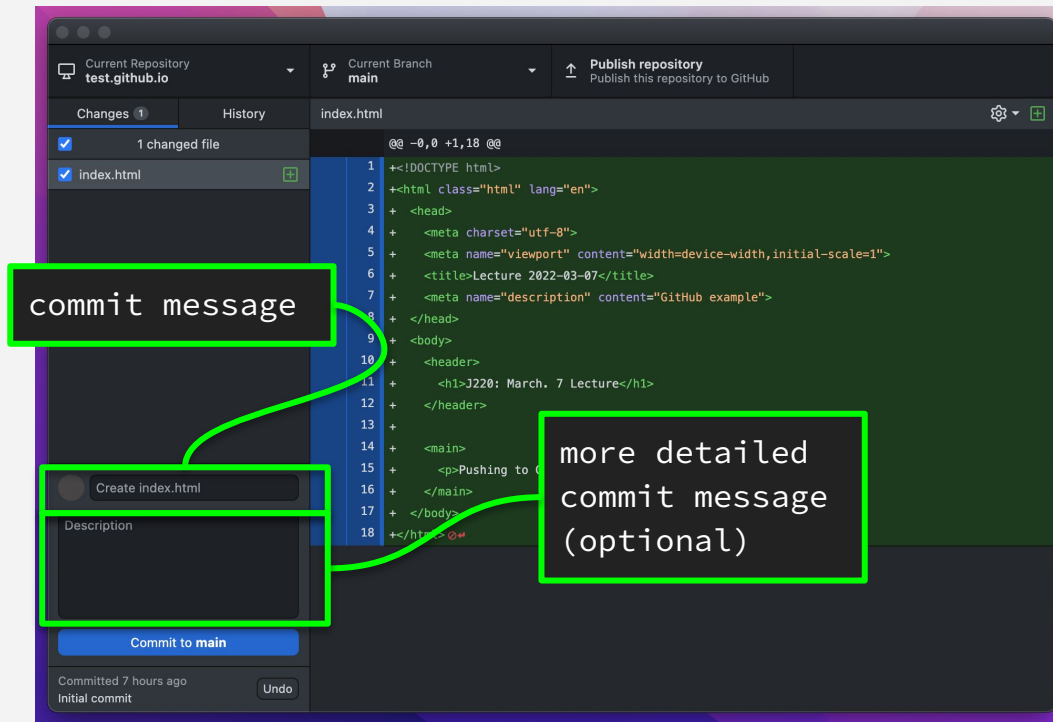
You can copy your portfolio HTML here.

# Return to GitHub Desktop

You should now see **index.html** here.

You now need to write a summary message, which is often called a “commit.”

“Committing” a file just means you’re saving a version of it.












# About commit messages

Every time you want to save a version of your code, you need to come up with a commit message.

A good commit message tells you what you did in case you want to go back and revert changes.

# Writing good commit messages matters

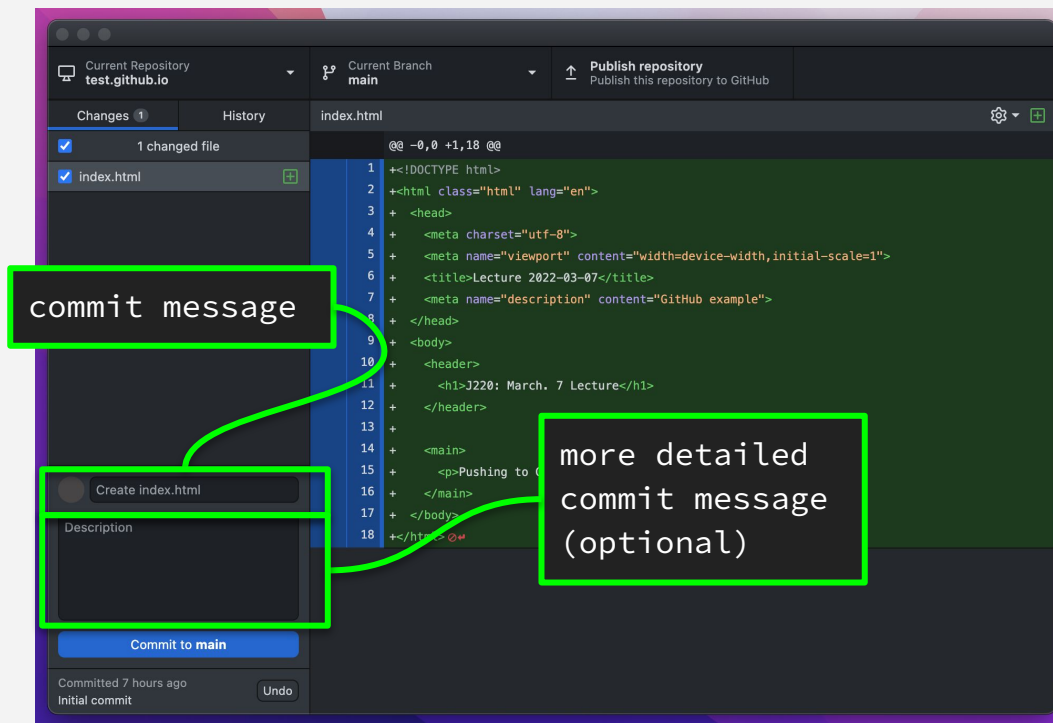
<p>Commits on Feb 15, 2022</p> <p><b>adding slides and recordings</b>  soooh committed 19 days ago</p>	<p>Commits on Feb 7, 2022</p> <p><b>adding feb 7 materials</b>  soooh committed 27 days ago</p>
<p>Commits on Feb 14, 2022</p> <p><b>adding feb 14 resources</b>  yolimartinez committed 20 days ago</p>	<p>Commits on Feb 1, 2022</p> <p><b>adding jan 31 slides and recording</b>  yolimartinez committed on Feb 1</p>
<p>Commits on Feb 8, 2022</p> <p><b>adding feb 7 slides and video</b>  soooh committed 26 days ago</p>	<p>Commits on Jan 31, 2022</p> <p><b>fixed assignment link</b>  yolimartinez committed on Jan 31</p> <p><b>added homework and recommended reading for Jan 31</b>  yolimartinez committed on Jan 31</p>

These are the commit messages for the J220 website.

How do you think we could have improved our commit messages?

# Write your first commit message

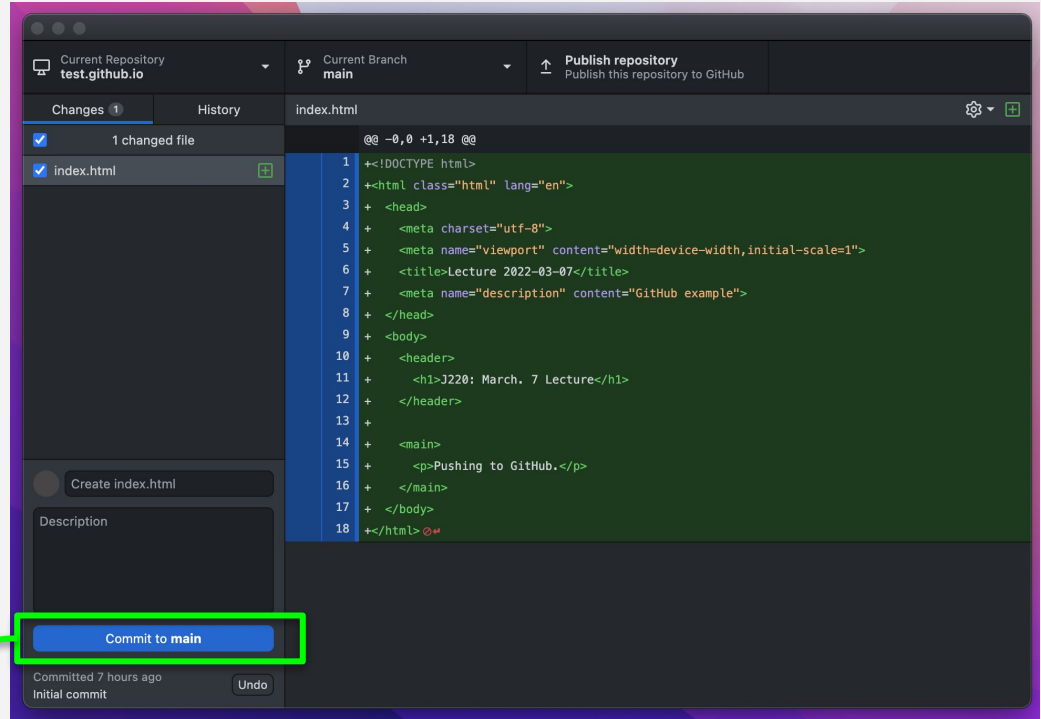
Write your first commit message. (Actually, the default one is pretty good, for now! But later you might want to write more specific ones.)



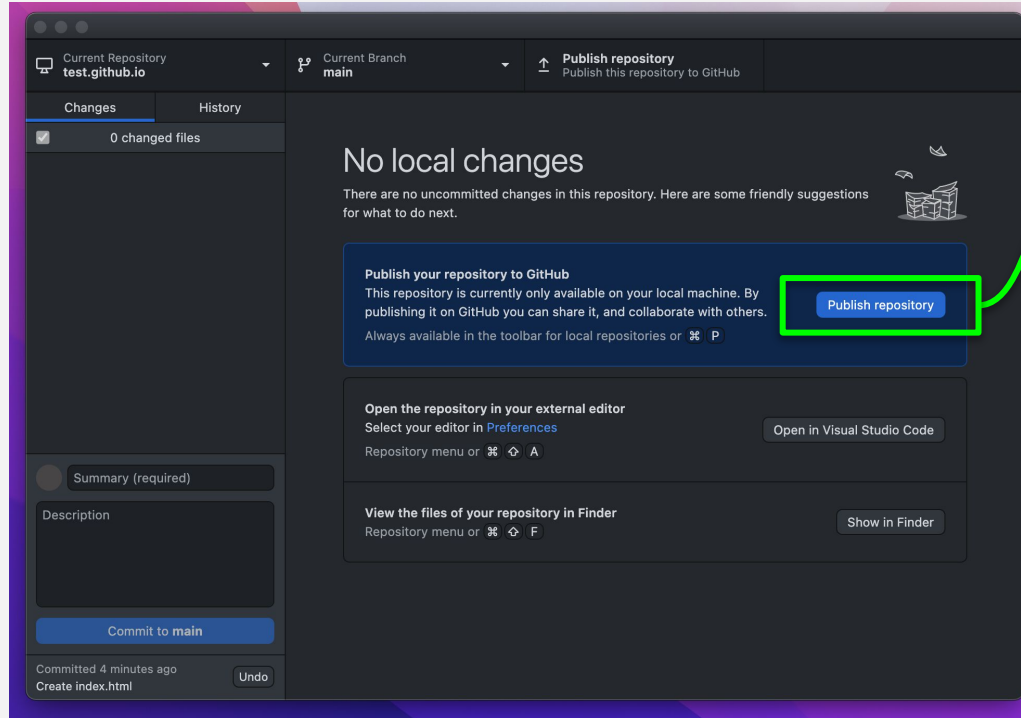
# Press the blue button to commit

After you've written a good commit message, you can press the blue button that says "Commit to **main**." (**main** is the name of the current branch, but you don't need to know this.)

Press this button  
to commit.



# Publish your repo to GitHub



Then you'll click on the button that says "**Publish repository.**"

# You'll get another popup...

**Publish Repository**

GitHub.com | GitHub Enterprise

Name  
**username.github.io**

Description  
Students: `Name` should be username.github.io

☒ Keep this code private

Organization  
None

Cancel | **Publish Repository**

Push this button!

It doesn't matter if you check  
off this box right now.

# Navigate to GitHub.com in your browser

Go to **GitHub.com** > **Settings** > **Pages**

Or, you can type the following in your browser (make sure to replace username with your actual username):

<https://github.com/username/username.github.io/settings/pages>

# That's it!

## GitHub Pages

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

[redacted]

### Source

Your GitHub Pages site is currently being built from the /docs folder in the main branch. [Learn more.](#)

Branch: main

/ (root)

Save

### Theme Chooser

Select a theme to publish your site with a Jekyll theme. [Learn more.](#)

Choose a theme

### Custom domain

Custom domains allow you to serve your site from a domain other than `journ220.github.io`. [Learn more.](#)

Save

Remove

### ☒ Enforce HTTPS

— Required for your site because you are using the default domain (`journ220.github.io`)

HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site. When HTTPS is enforced, your site will only be served over HTTPS. [Learn more.](#)

You'll get to this page. You don't have to change anything. GitHub.com will automatically start building your site at `username.github.io`.

Your site will be ready in about 5 to 10 minutes (or less).



# Making changes

You're still going to be using your **text editor** (like Sublime Text) to make your web pages.

You won't write any code in **GitHub Desktop**.

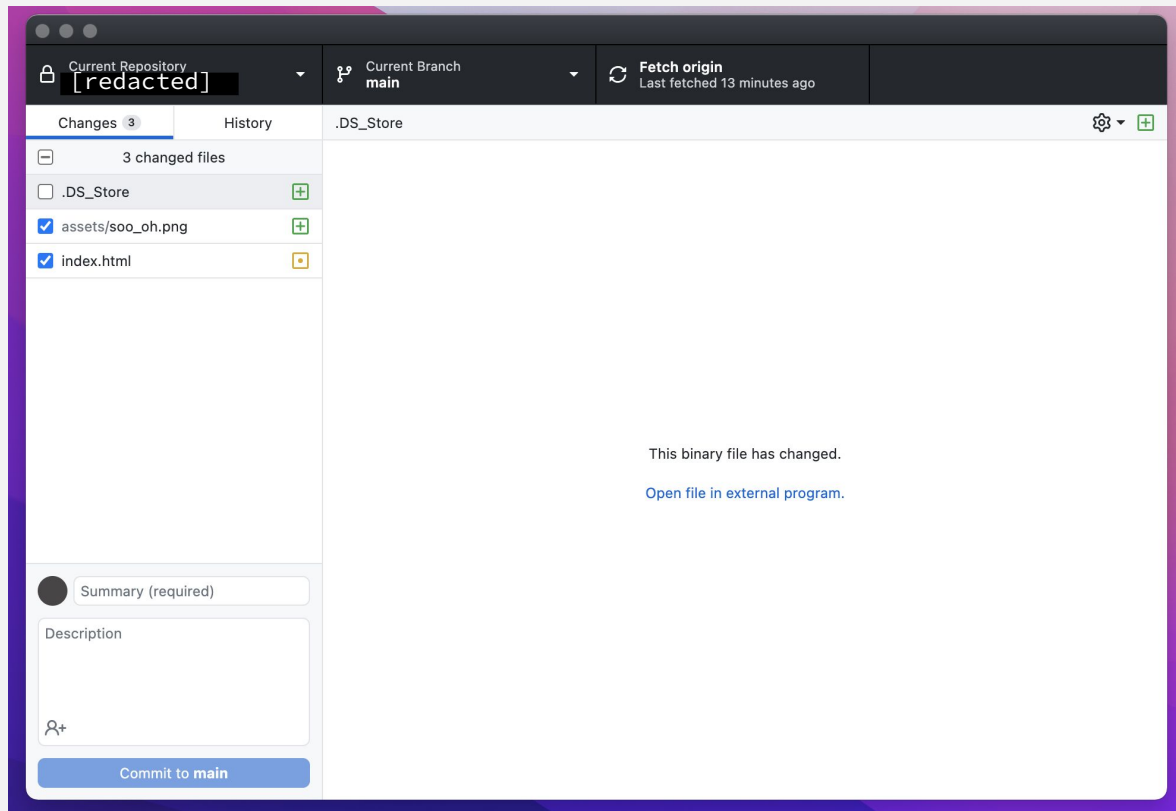
Anytime you want to make changes to your website, you'll go back to **GitHub Desktop**, write a **commit** message, then **publish** your repository.

Let's see what that looks like...

# Making changes

- I created an assets folder.
- I placed an image in my assets folder.
- I also edited index.html with my portfolio text.

Then I went back to GitHub Desktop.



# Making changes

- `.DS_Store` is a Mac file. Uncheck this box.
- I placed an image in my assets folder.
- I also edited `index.html` with my portfolio text.

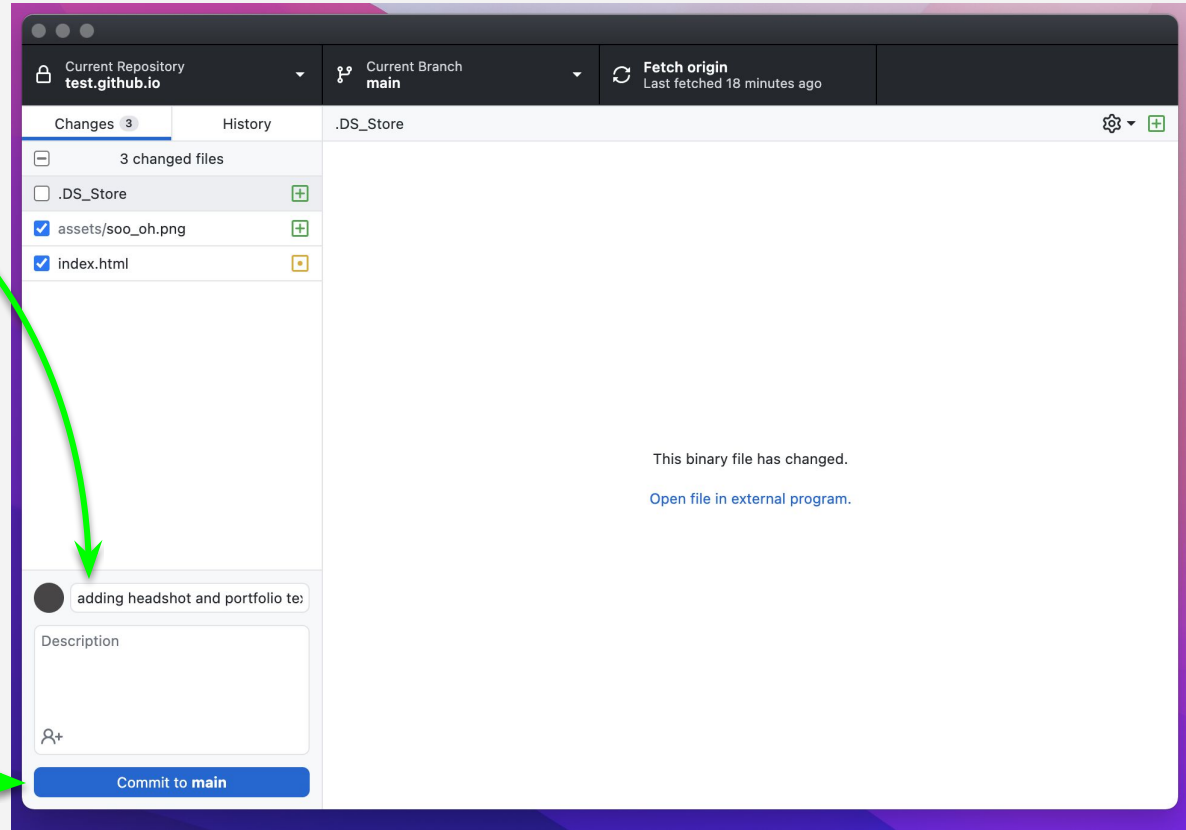
What is `.DS_Store`?  
Uncheck this box if you see it.

green indicates new file

yellow indicates change to existing file

# Commit message

Write your commit message, push that blue button, and publish your repository.



# GitHub extras

You can make a separate repo for your final project (if it's not a portfolio).

Then, you can set up your repo to display as

`https://username.github.io/repo-name` in your settings.

So, your final project URL could look like:

<https://soooh.github.io/timer-app>, where **soooh** is the username and **timer-app** is the name of the repo.

Come to office hours if you'd like to learn how!

# Homework (and other deadlines to plan for)

You don't really have anything to work on except your final project for the next few weeks.

**Assignment 04-17:** Due April 22. Link to portfolio on Github.io + extra credit Q

**Assignment 04-24:** Due April 29. Screenshot of any updates to your wireframe/final project

**May 1 Presentations (final lecture):** Prepare for in-class presentation. Peer critiques are due two days later on May 3.

**Final project: May 11!**