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!

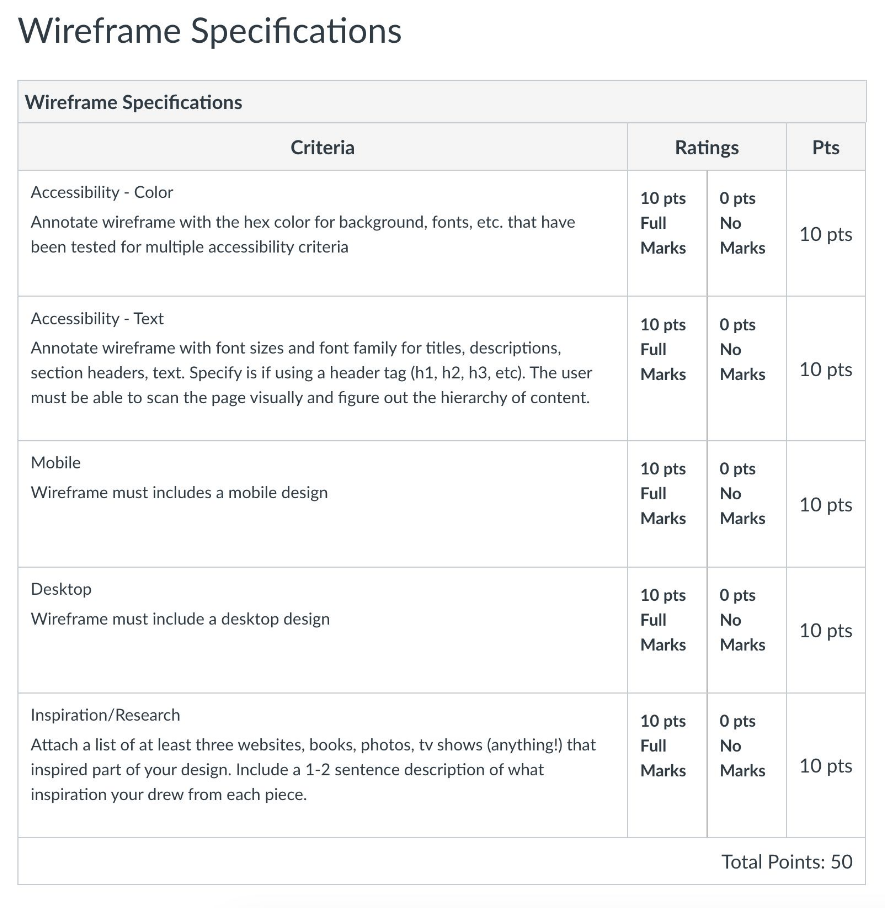
- **Nancy Deville**, Editorial Project Manager at The Atlantic
- **Gabe Hongsdusit**, Visual Designer at The Markup
- **Evan Wagstaff**, Director of Newsroom Engineering at Hearst Newspapers

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 to the Wireframe assignment**. Review rubric, we need to see all of these in your wireframe.



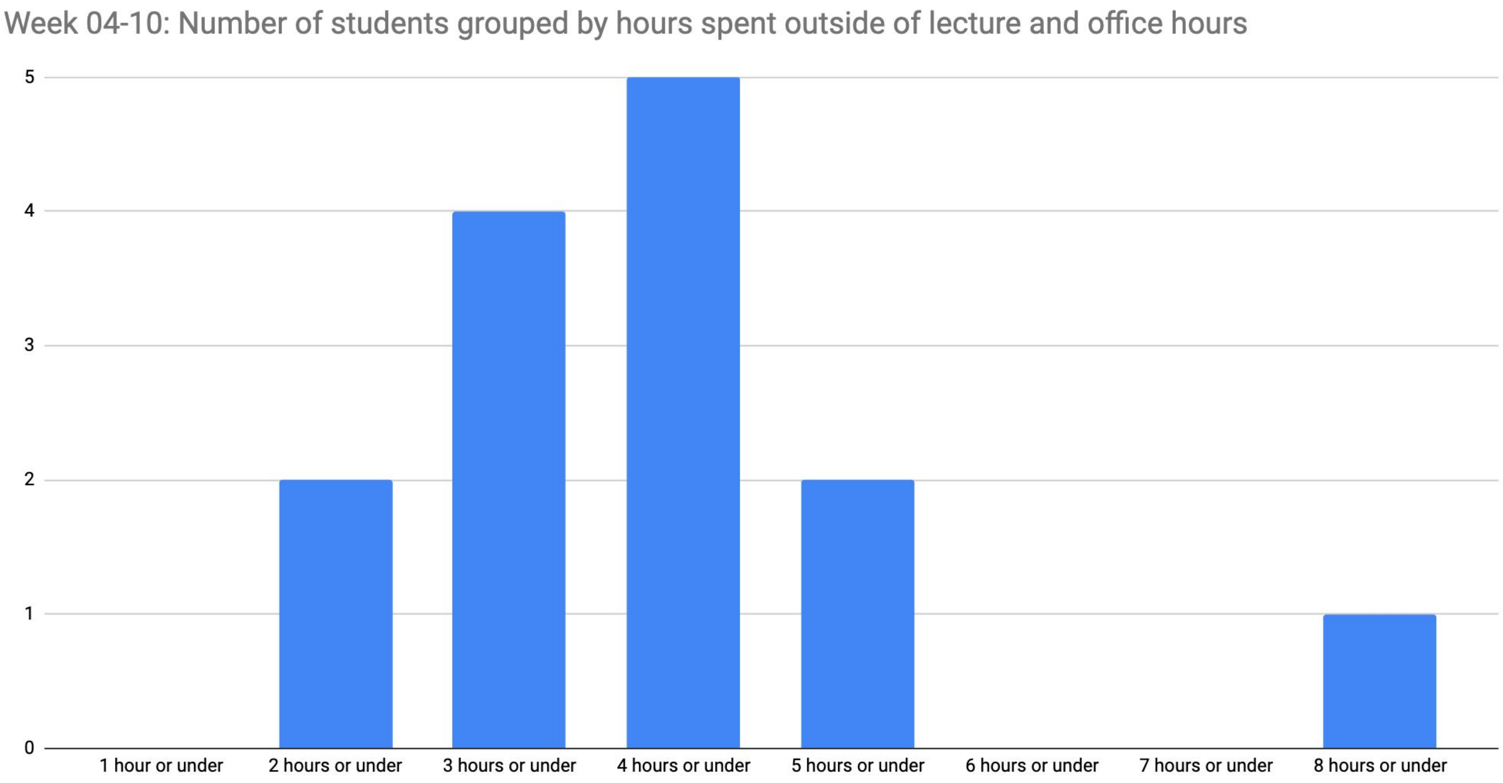
Wireframe Specifications

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 is 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 includes 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 your drew from each piece.	**10 pts** **Full Marks**	**0 pts** **No Marks**	10 pts
			Total Points: 50

SAJA event recap

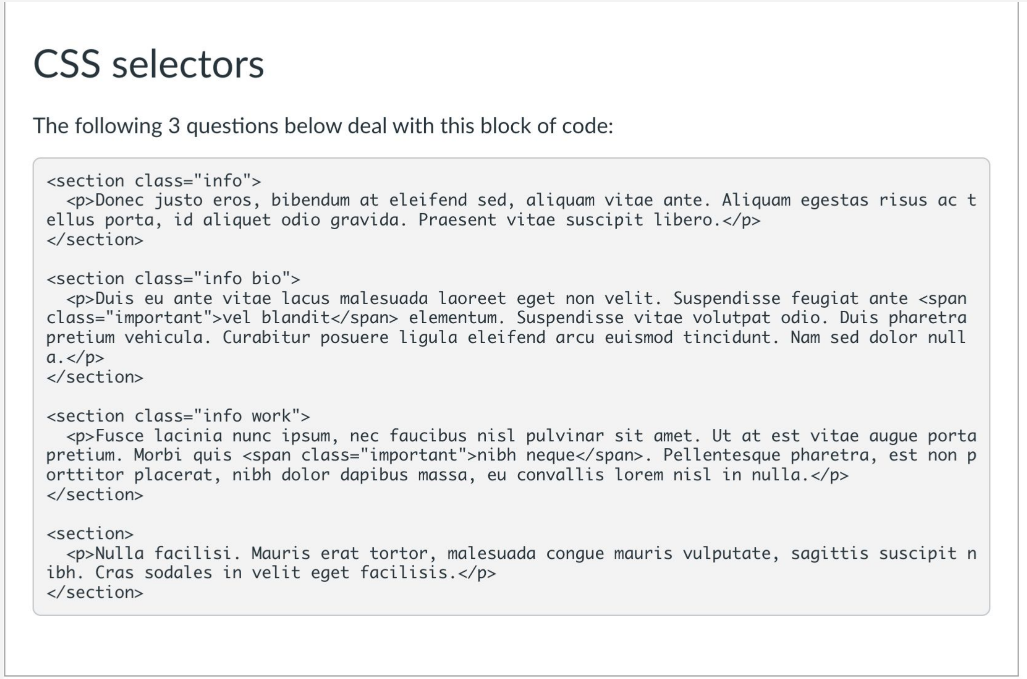
How much time spent on J220 last week



Homework Review

Short answer key

Homework Review



CSS selectors

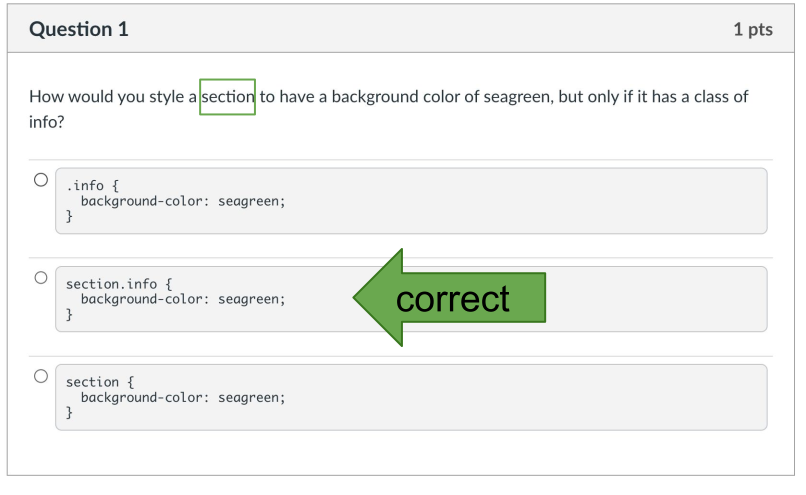
The following 3 questions below deal with this block of code:

```
<section class="info">
  <p>Donec justo eros, bibendum at eleifend sed, aliquam vitae ante. Aliquam egestas risus ac t
ellus porta, id aliquet odio gravida. Praesent vitae suscipit libero.</p>
</section>

<section class="info bio">
  <p>Duis eu ante vitae lacus malesuada laoreet eget non velit. Suspendisse feugiat ante <span
class="important">vel blandit</span> elementum. Suspendisse vitae volutpat odio. Duis pharetra
pretium vehicula. Curabitur posuere ligula eleifend arcu euismod tincidunt. Nam sed dolor null
a.</p>
</section>

<section class="info work">
  <p>Fusce lacinia nunc ipsum, nec faucibus nisl pulvinar sit amet. Ut at est vitae augue porta
pretium. Morbi quis <span class="important">nibh neque</span>. Pellentesque pharetra, est non p
orttitor placerat, nibh dolor dapibus massa, eu convallis lorem nisl in nulla.</p>
</section>

<section>
  <p>Nulla facilisi. Mauris erat tortor, malesuada congue mauris vulputate, sagittis suscipit n
ibh. Cras sodales in velit eget facilisis.</p>
</section>
```



Question 1 1 pts

How would you style a section to have a background color of seagreen, but only if it has a class of info?

○
```
.info {
  background-color: seagreen;
}
```

○
```
section.info {
  background-color: seagreen;
}
```

correct

○
```
section {
  background-color: seagreen;
}
```

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.

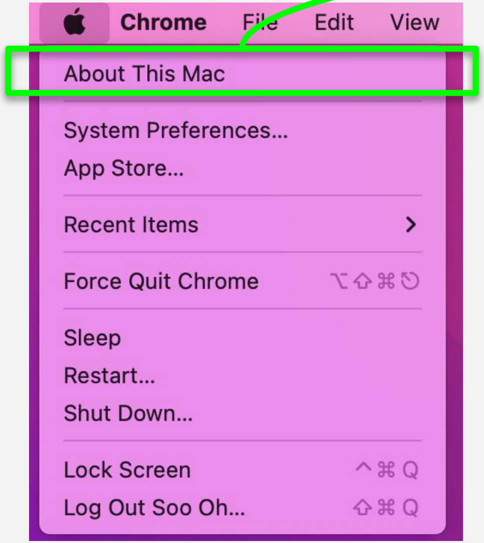
Instead, you have to **commit** (save) your code, which only happens on your computer.

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

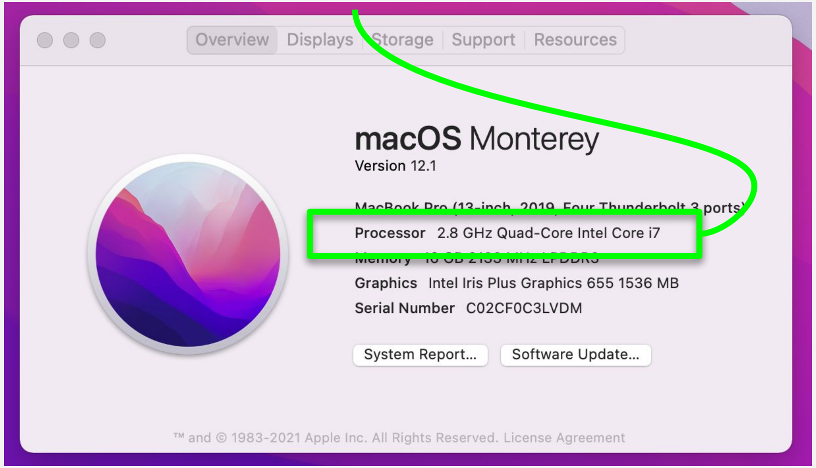
There are similar kinds of version control tools for code and software, but Git is the most popular.

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:



macOS Monterey

Version 12.1

MacBook Pro (13-inch, 2019, Four Thunderbolt 3 ports)

Processor 2.8 GHz Quad-Core Intel Core i7

Memory 16 GB 2133 MHz LPDDR3

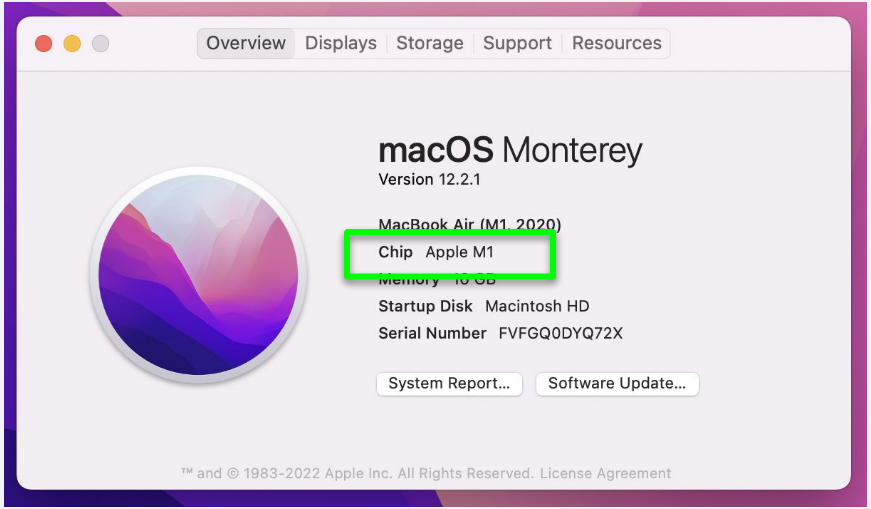
Graphics Intel Iris Plus Graphics 655 1536 MB

Serial Number C02CF0C3LVDM

™ and © 1983-2021 Apple Inc. All Rights Reserved. License Agreement

Determine what kind of chip you have

Apple Silicon

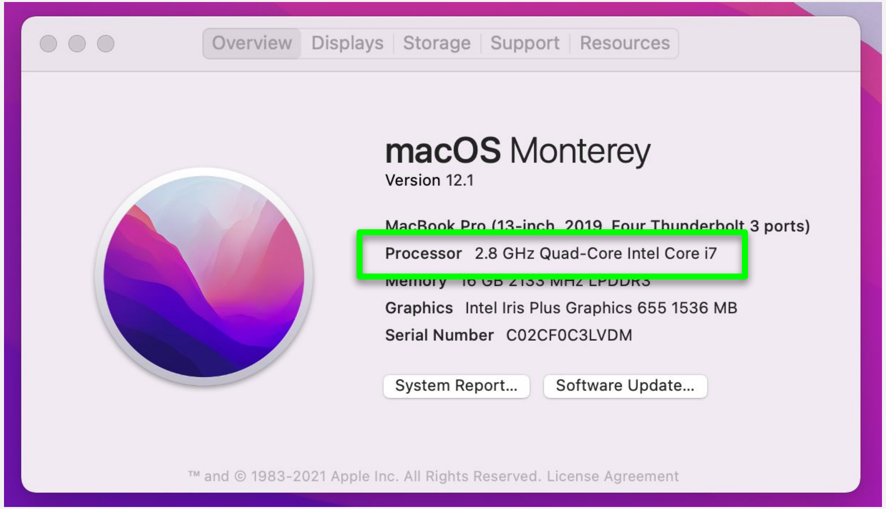


macOS Monterey
Version 12.2.1

MacBook Air (M1, 2020)
Chip Apple M1
Memory 16 GB
Startup Disk Macintosh HD
Serial Number FVFGQ0DYQ72X

™ and © 1983-2022 Apple Inc. All Rights Reserved. License Agreement

Intel



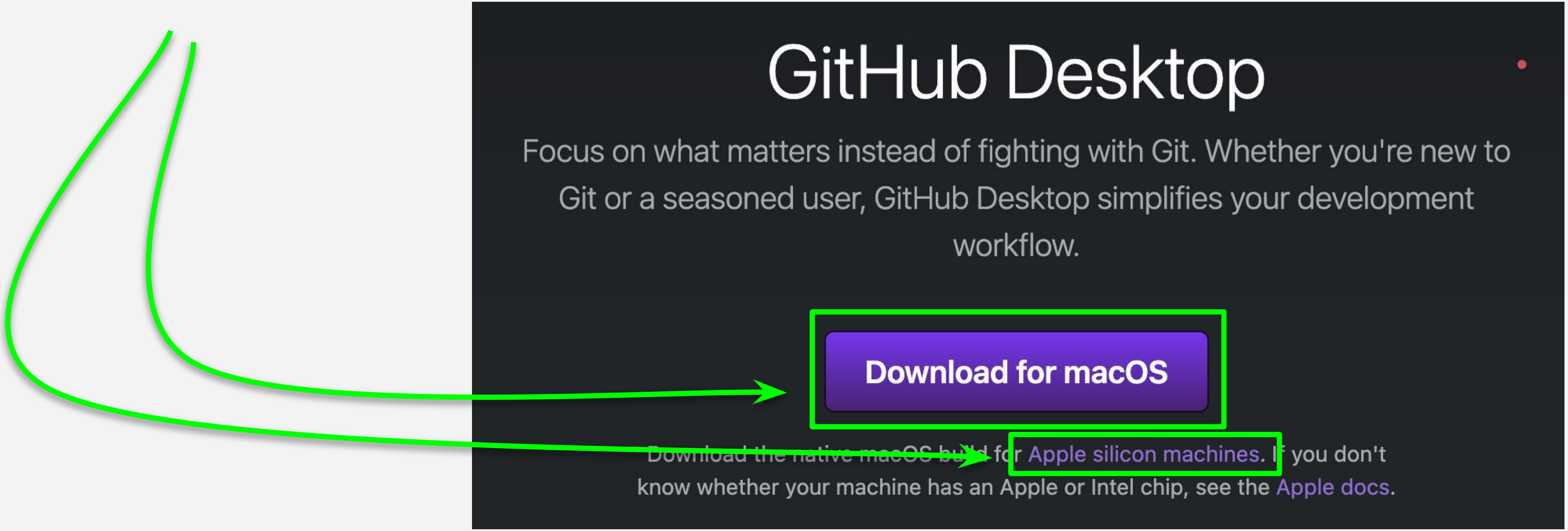
macOS Monterey
Version 12.1

MacBook Pro (13-inch, 2019, Four Thunderbolt 3 ports)
Processor 2.8 GHz Quad-Core Intel Core i7
Memory 16 GB 2133 MHz LPDDR3
Graphics Intel Iris Plus Graphics 655 1536 MB
Serial Number C02CF0C3LVDM

™ and © 1983-2021 Apple Inc. All Rights Reserved. License Agreement

Download GitHub Desktop

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

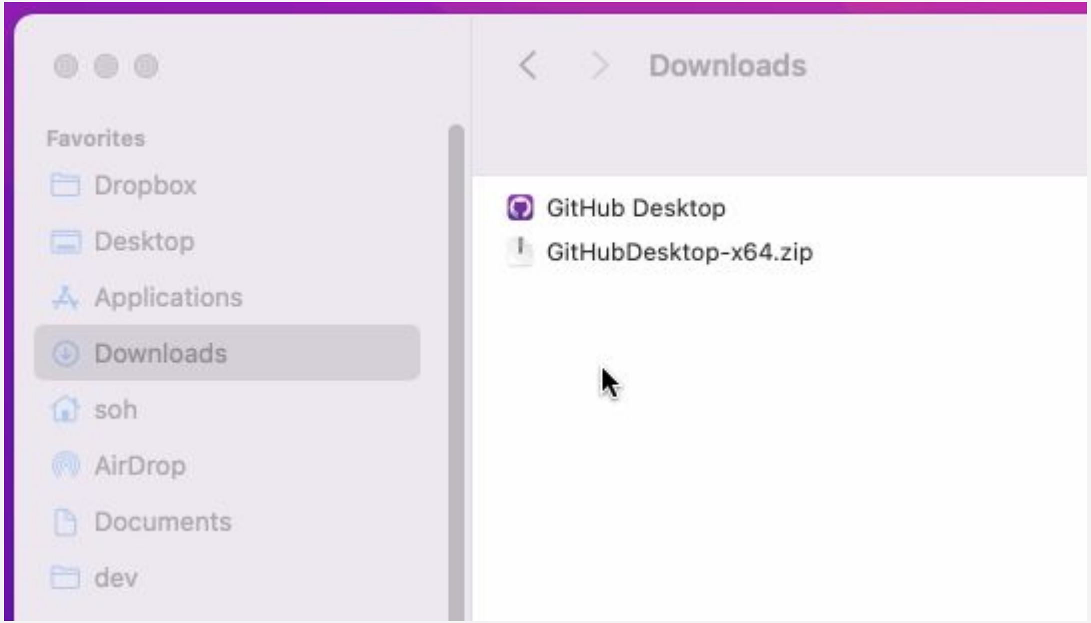


GitHub Desktop

Focus on what matters instead of fighting with Git. Whether you're new to Git or a seasoned user, GitHub Desktop simplifies your development workflow.

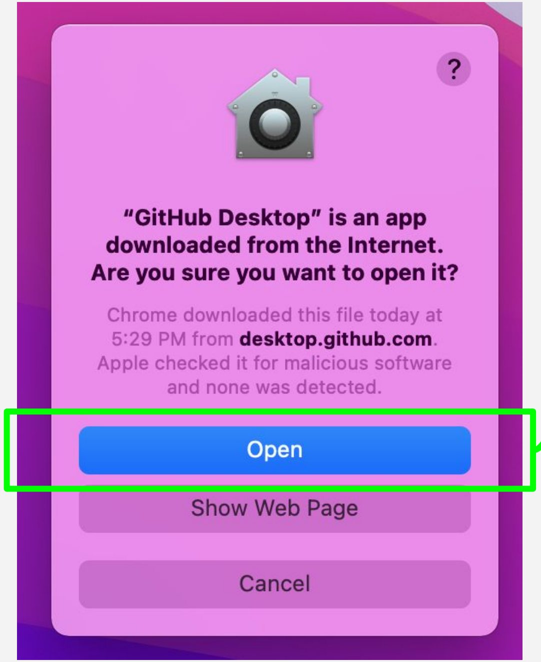
Download the native macOS build for Apple silicon machines. If you don't know whether your machine has an Apple or Intel chip, see the Apple docs.

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



"GitHub Desktop" is an app downloaded from the Internet. Are you sure you want to open it?

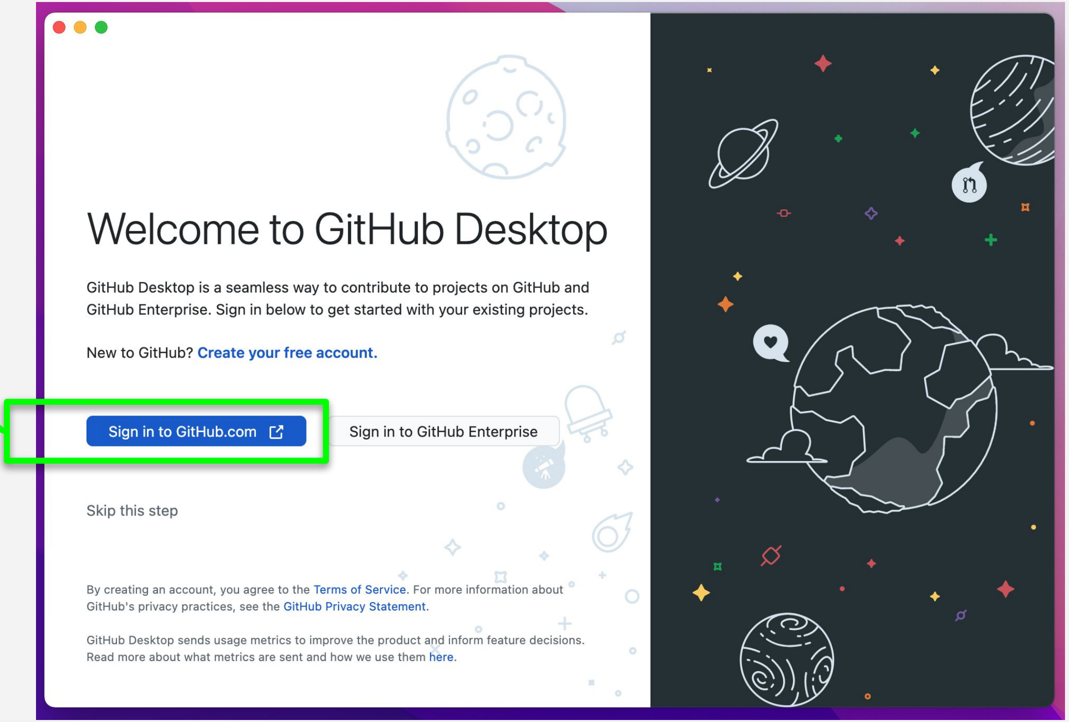
Chrome downloaded this file today at 5:29 PM from **desktop.github.com**. Apple checked it for malicious software and none was detected.

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.



Welcome to GitHub Desktop

GitHub Desktop is a seamless way to contribute to projects on GitHub and GitHub Enterprise. Sign in below to get started with your existing projects.

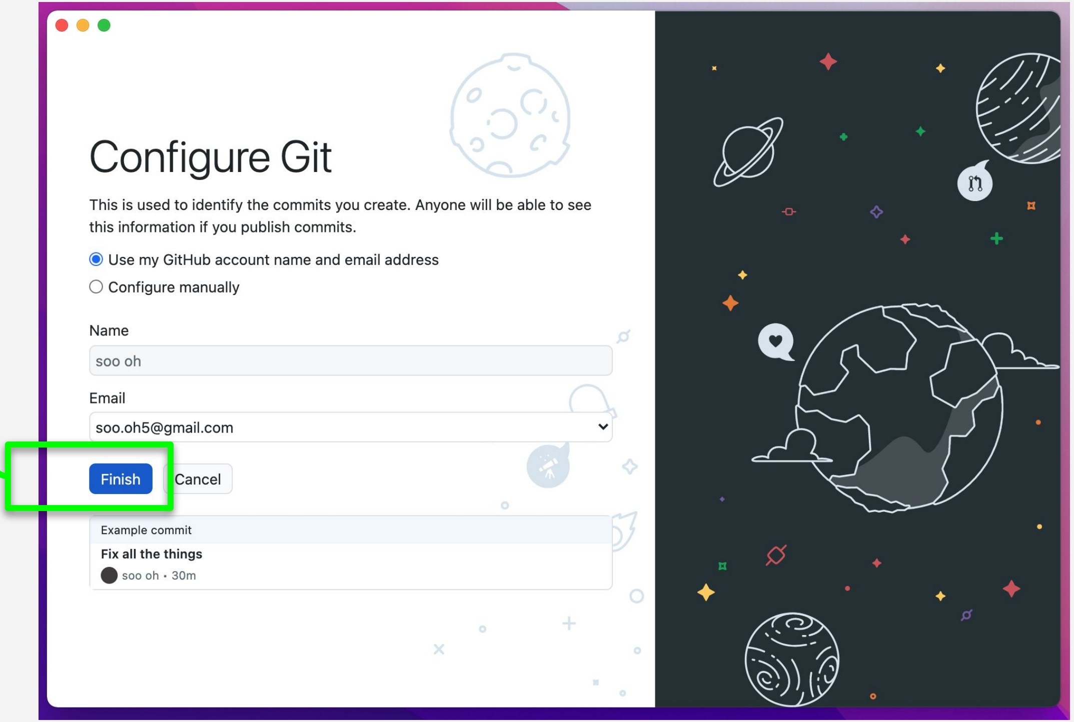
New to GitHub? Create your free account.

By creating an account, you agree to the Terms of Service. For more information about GitHub's privacy practices, see the GitHub Privacy Statement.

GitHub Desktop sends usage metrics to improve the product and inform feature decisions. Read more about what metrics are sent and how we use them here.

Configure Git

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



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: soo oh

Email: soo.oh5@gmail.com

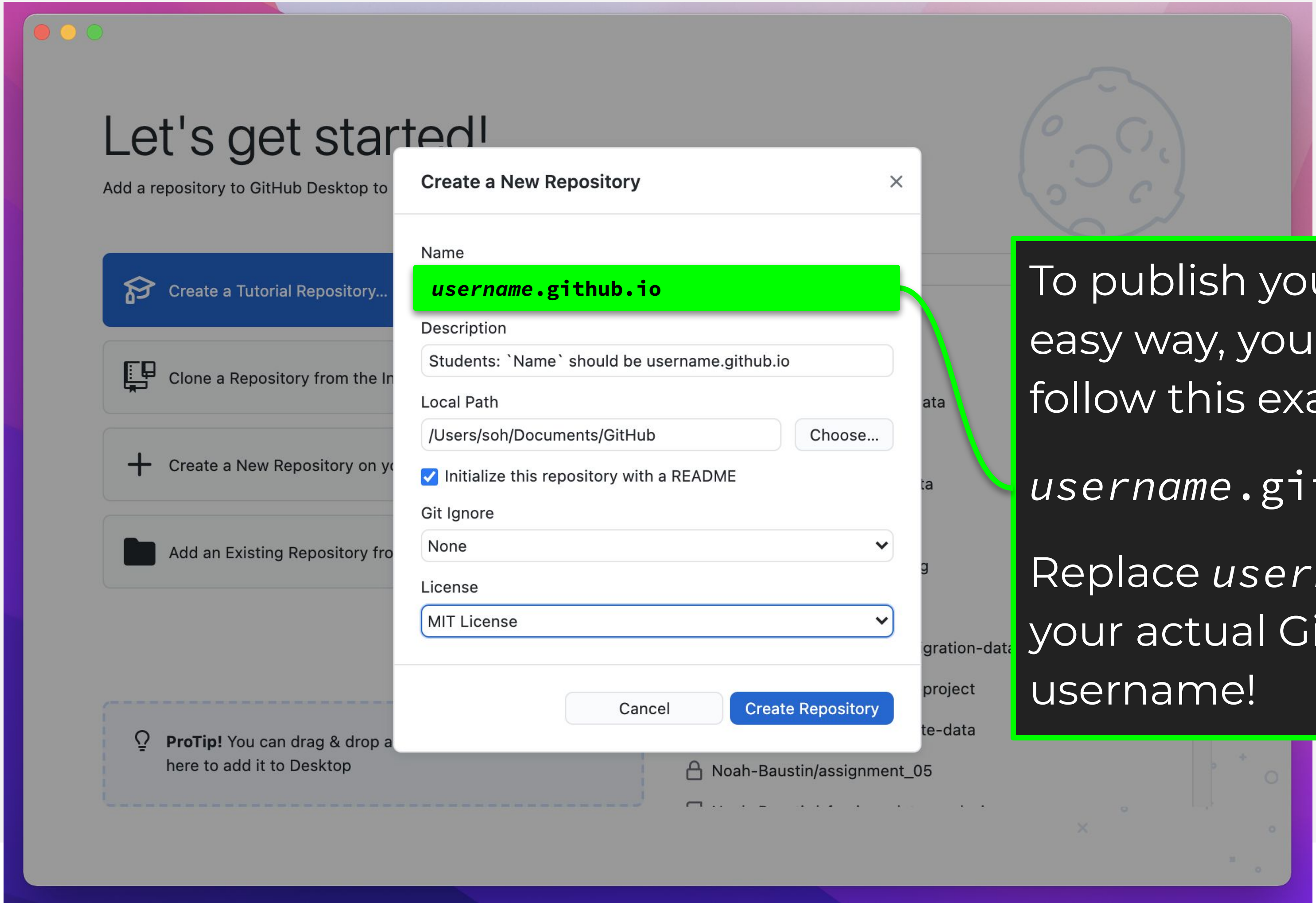
Example commit

Fix all the things

soo oh • 30m

Create a new repository

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


Let's get started!
Create a Tutorial Repository...
Clone a Repository from the In
Create a New Repository on yo
Add an Existing Repository fro
Noah-Baustin/assignment_05
Create Repository
Cancel
Choose...

Add a repository to GitHub Desktop to

ProTip! You can drag & drop a
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

☑ Initialize this repository with a README

Git Ignore

None

License

MIT License

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

username.github.io

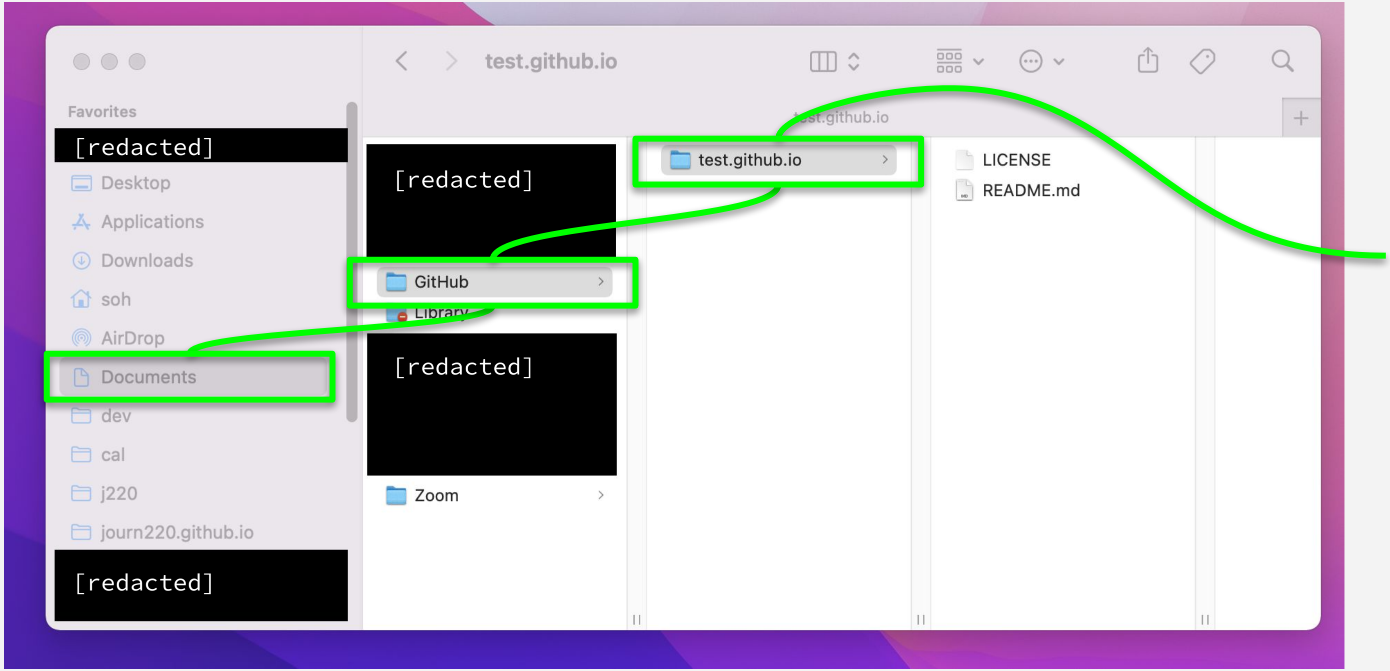
Replace *username* with your actual GitHub username!

What is your GitHub.io address?

Powered by Poll Everywhere

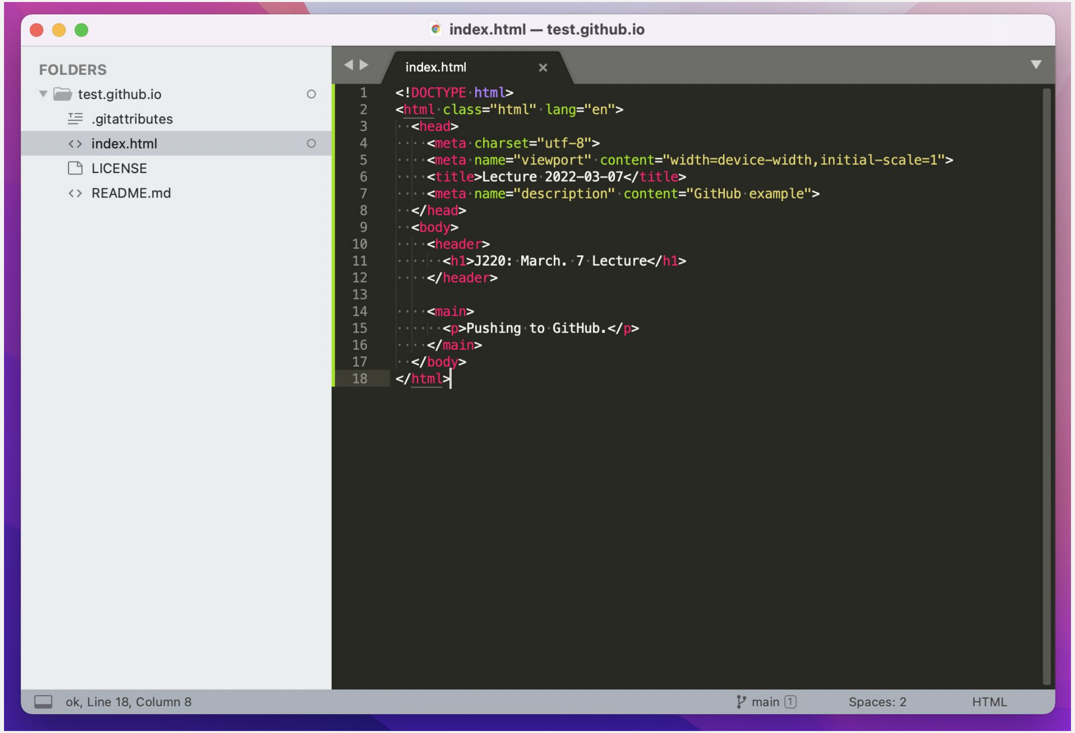
Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

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



```html
<!DOCTYPE html>
<html class="html" lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=1">
    <title>Lecture 2022-03-07</title>
    <meta name="description" content="GitHub example">
  </head>
  <body>
    <header>
      <h1>J220: March. 7 Lecture</h1>
    </header>

    <main>
      <p>Pushing to GitHub.</p>
    </main>
  </body>
</html>
```

ok, Line 18, Column 8

Spaces: 2

HTML

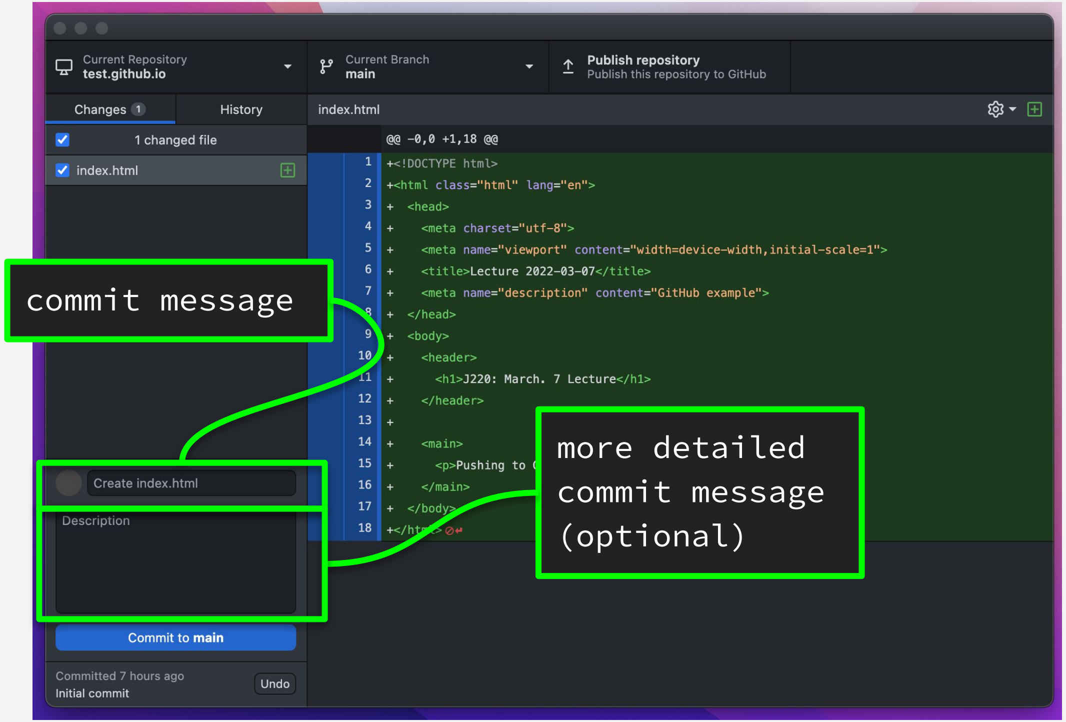
You can copy your portfolio HTML here.

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.



```
@@ -0,0 +1,18 @@
+<!DOCTYPE html>
+<html class="html" lang="en">
+  <head>
+    <meta charset="utf-8">
+    <meta name="viewport" content="width=device-width,initial-scale=1">
+    <title>Lecture 2022-03-07</title>
+    <meta name="description" content="GitHub example">
+  </head>
+  <body>
+    <header>
+      <h1>J220: March. 7 Lecture</h1>
+    </header>
+
+    <main>
+      <p>Pushing to (
+    </main>
+  </body>
+</ht
```

Create index.html

commit message

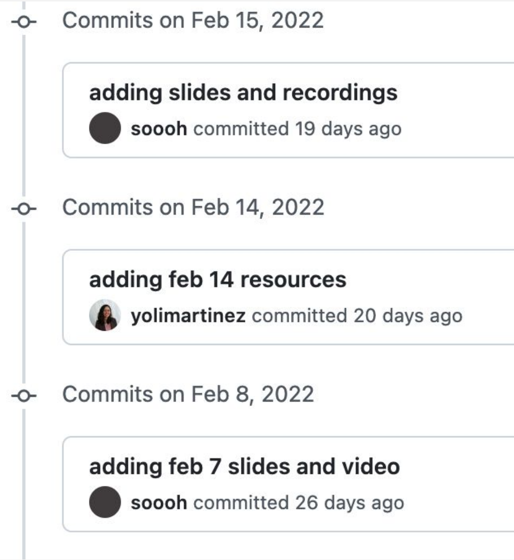
more detailed commit message (optional)

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



Commits on Feb 15, 2022

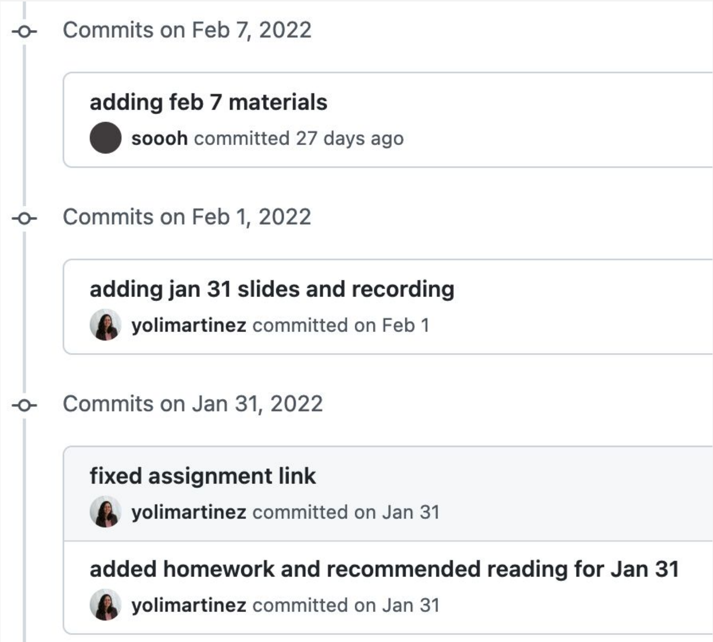
adding slides and recordings
soooh committed 19 days ago

Commits on Feb 14, 2022

adding feb 14 resources
yolimartinez committed 20 days ago

Commits on Feb 8, 2022

adding feb 7 slides and video
soooh committed 26 days ago



Commits on Feb 7, 2022

adding feb 7 materials
soooh committed 27 days ago

Commits on Feb 1, 2022

adding jan 31 slides and recording
yolimartinez committed on Feb 1

Commits on Jan 31, 2022

fixed assignment link
yolimartinez committed on Jan 31

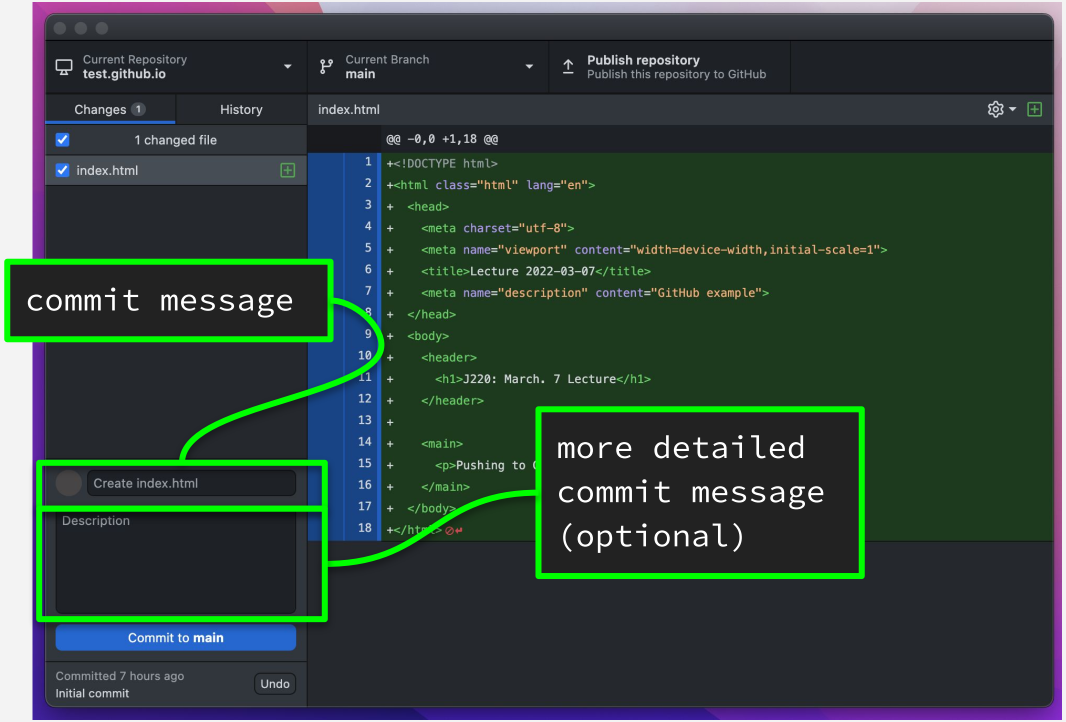
added homework and recommended reading for Jan 31
yolimartinez committed on Jan 31

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.)



commit message

more detailed commit message (optional)

Create index.html

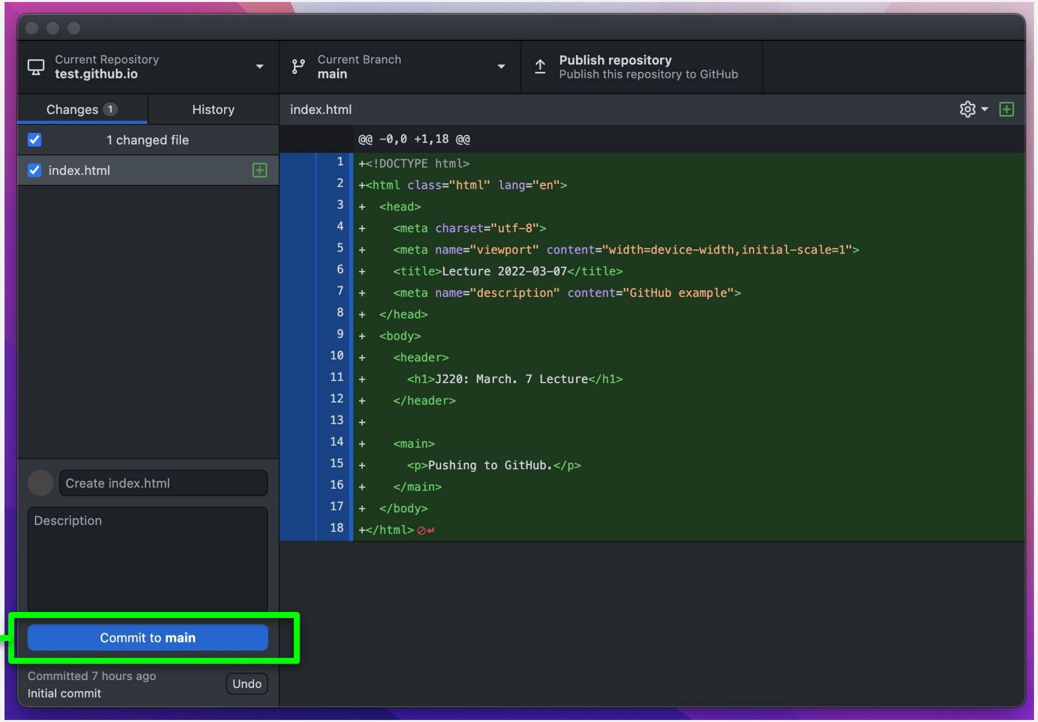
Description

Committed 7 hours ago
Initial commit

```
@@ -0,0 +1,18 @@
+<!DOCTYPE html>
+<html class="html" lang="en">
+  <head>
+    <meta charset="utf-8">
+    <meta name="viewport" content="width=device-width,initial-scale=1">
+    <title>Lecture 2022-03-07</title>
+    <meta name="description" content="GitHub example">
+  </head>
+  <body>
+    <header>
+      <h1>J220: March. 7 Lecture</h1>
+    </header>
+
+    <main>
+      <p>Pushing to (
+    </main>
+  </body>
+</html>
```

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.)

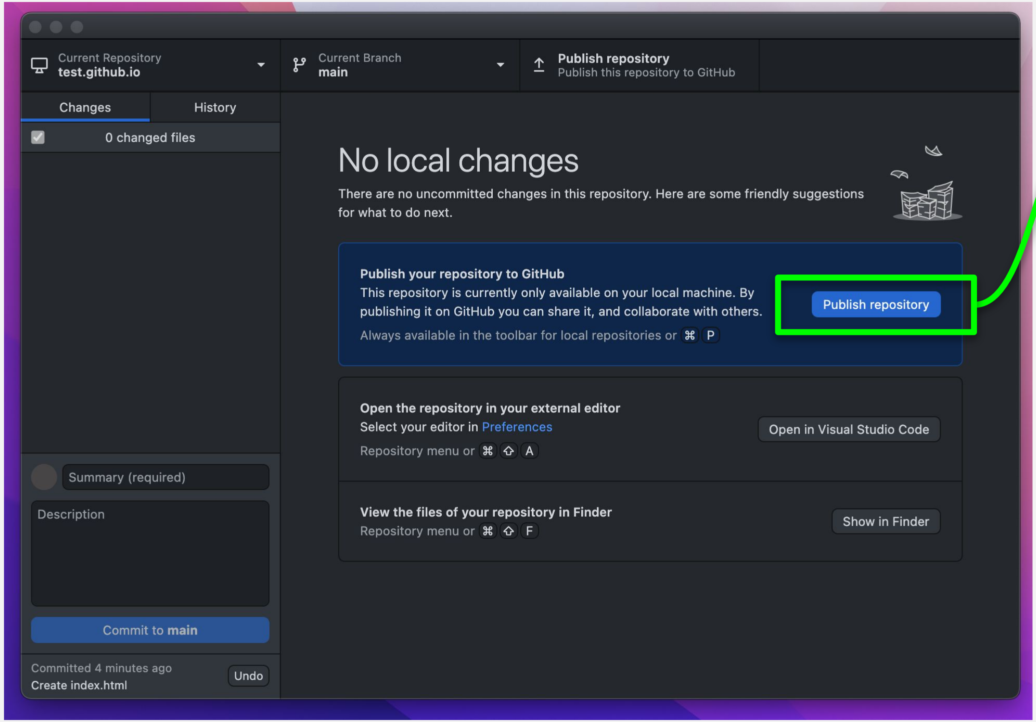


Create index.html

```
@@ -0,0 +1,18 @@
+<!DOCTYPE html>
+<html class="html" lang="en">
+  <head>
+    <meta charset="utf-8">
+    <meta name="viewport" content="width=device-width,initial-scale=1">
+    <title>Lecture 2022-03-07</title>
+    <meta name="description" content="GitHub example">
+  </head>
+  <body>
+    <header>
+      <h1>J220: March. 7 Lecture</h1>
+    </header>
+
+    <main>
+      <p>Pushing to GitHub.</p>
+    </main>
+  </body>
+</html>
```

Press this button to commit.

Publish your repo to GitHub



Current Repository
test.github.io

Current Branch
main

Publish repository
Publish this repository to GitHub

Changes | History

☑ 0 changed files

No local changes

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

Publish your repository to GitHub
This repository is currently only available on your local machine. By publishing it on GitHub you can share it, and collaborate with others.
Always available in the toolbar for local repositories or ⌘ P

Publish repository

Open the repository in your external editor
Select your editor in Preferences
Repository menu or ⌘ ⇧ A

Open in Visual Studio Code

View the files of your repository in Finder
Repository menu or ⌘ ⇧ F

Show in Finder

Summary (required)

Description

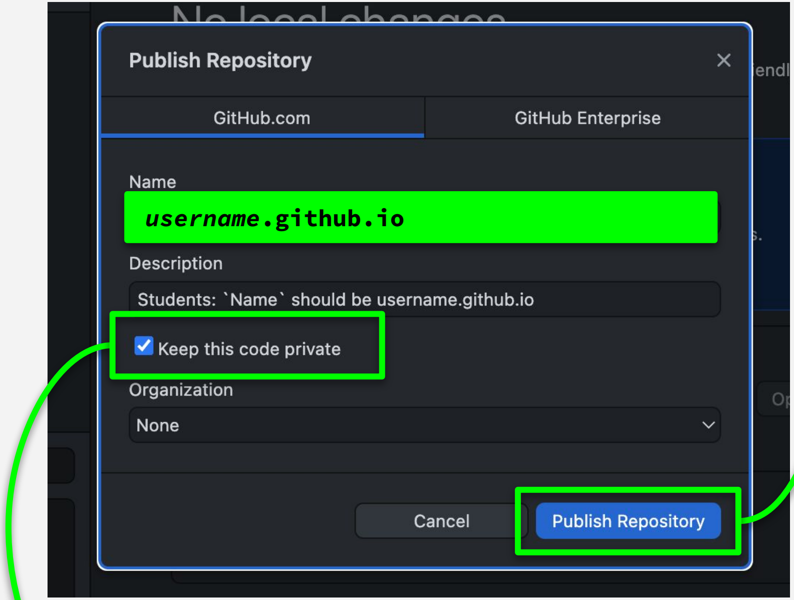
Commit to **main**

Committed 4 minutes ago
Create index.html

Undo

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

You’ll get another popup...



Publish Repository

Name: ***username*.github.io**

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

☑ Keep this code private

Organization: None

Push this button!

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

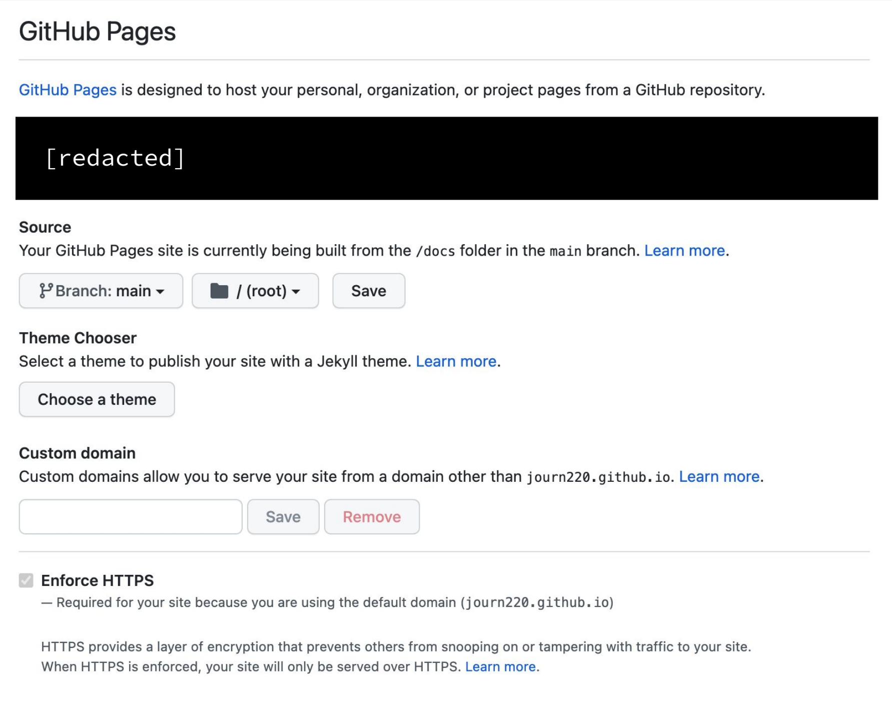
Navigate to GitHub.com in your browser

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) ▾

Theme Chooser

Select a theme to publish your site with a Jekyll theme. Learn more.

Custom domain

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

☑ **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**.

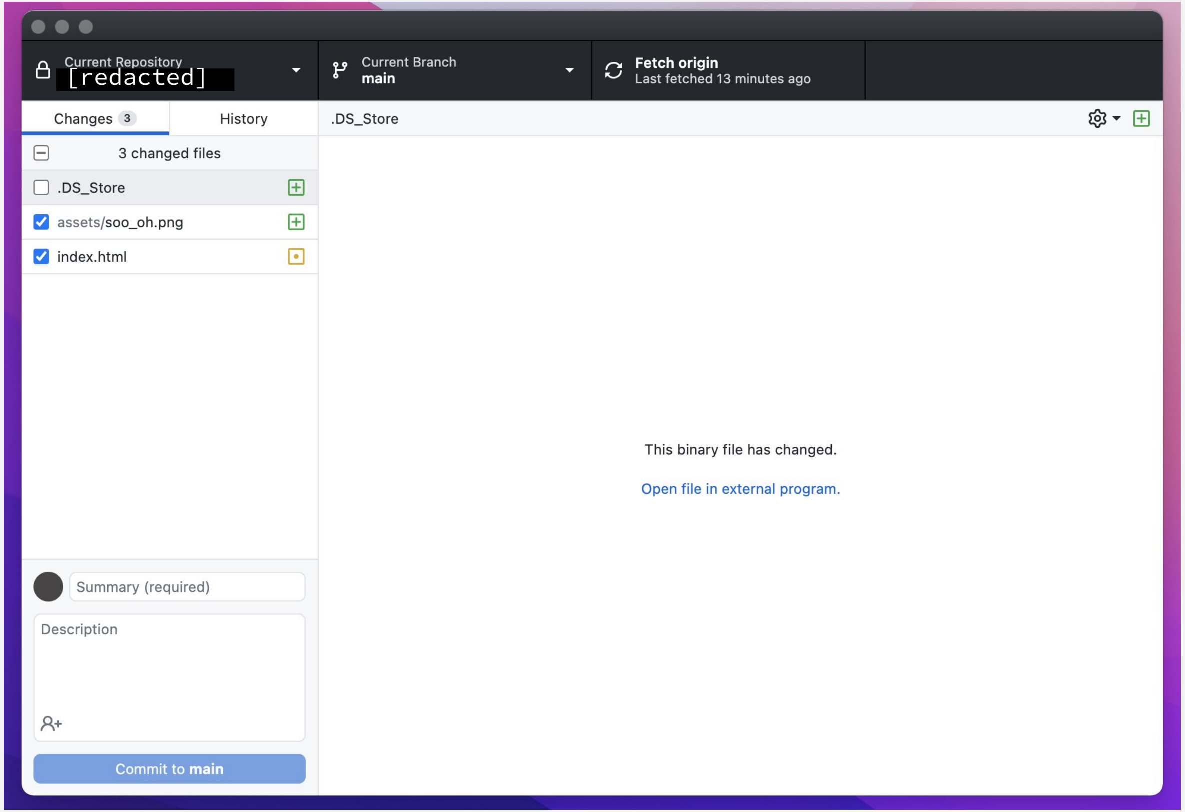
Anytime you want to make changes to your website, you'll go back to **GitHub Desktop**, write a **commit** message, then **publish** your repository.

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.



Current Repository [redacted]

Current Branch main

Fetch origin
Last fetched 13 minutes ago

Changes 3 | History

.DS_Store

3 changed files

☐ .DS_Store
☑ assets/soo_oh.png
☑ index.html

This binary file has changed.

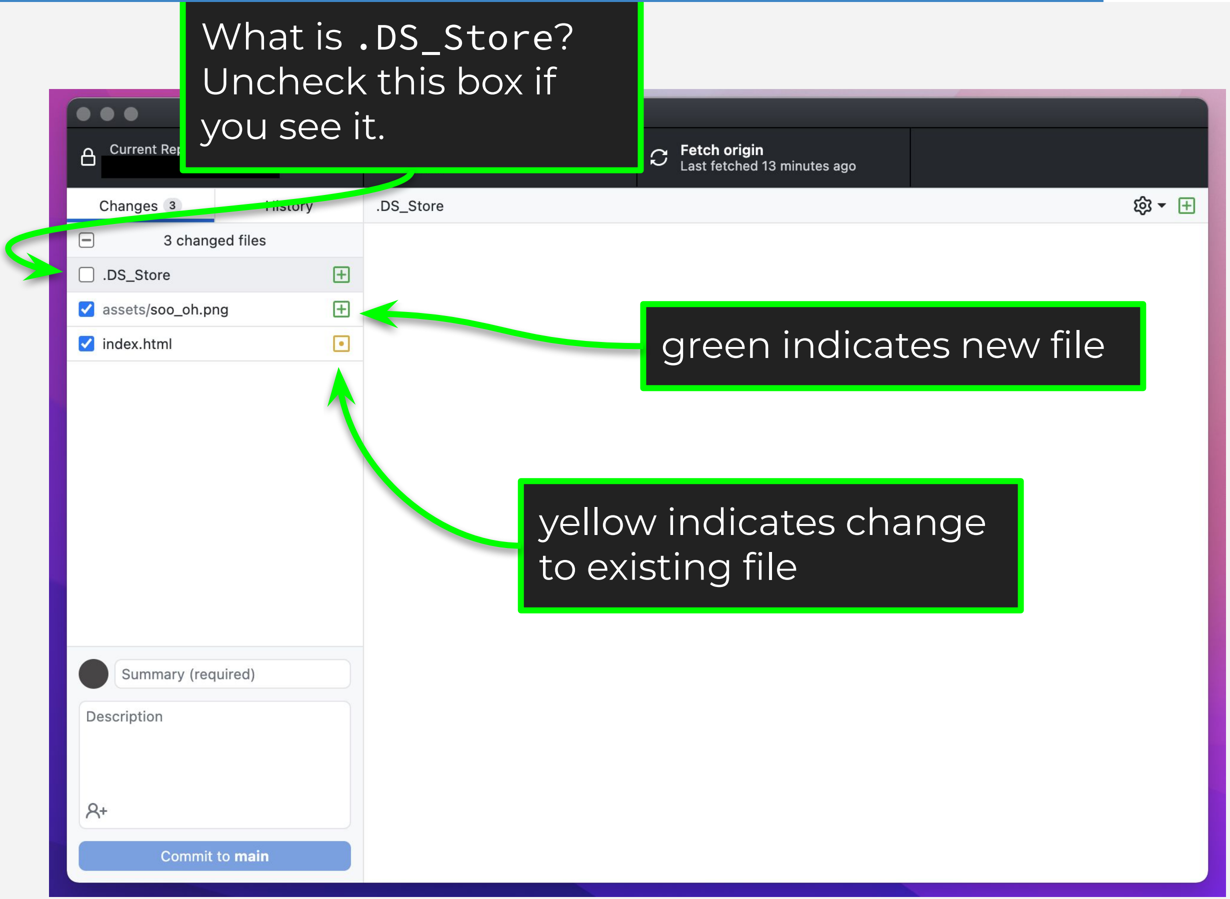
Open file in external program.

Summary (required)

Description

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.



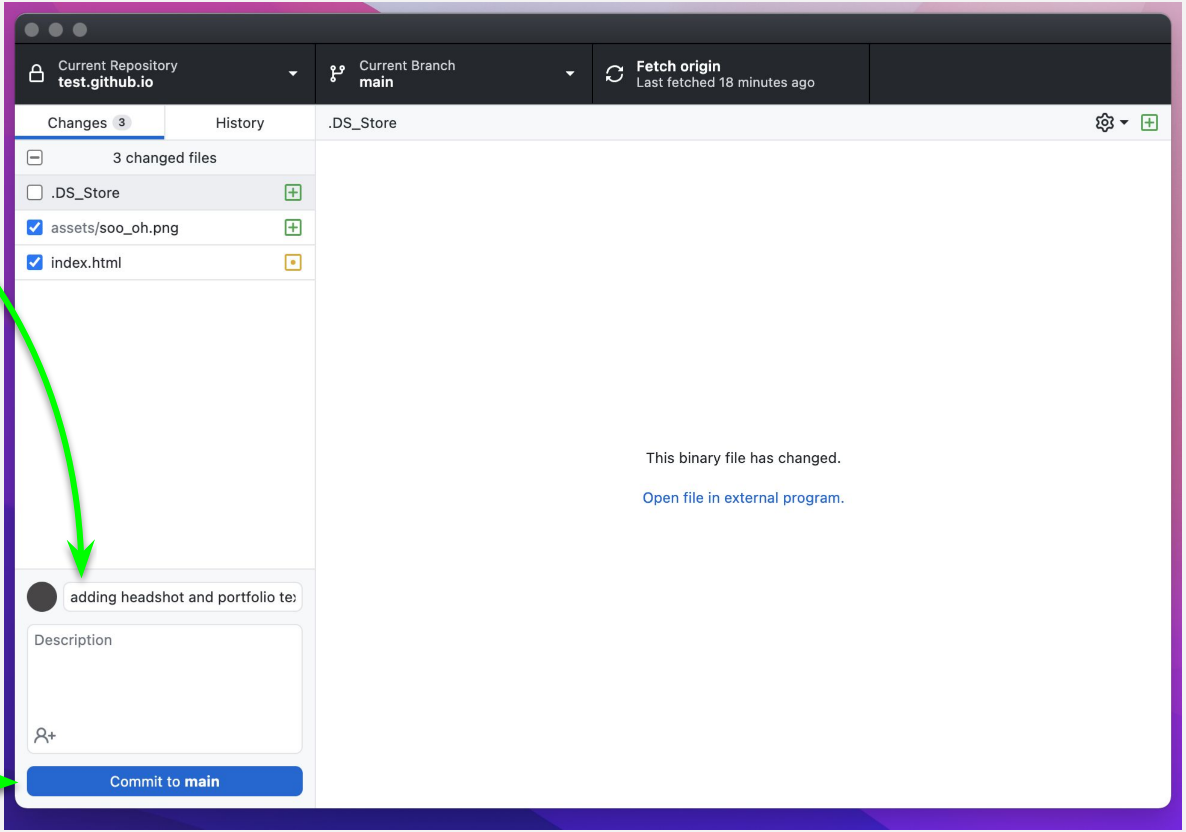
What is `.DS_Store`? Uncheck this box if you see it.

green indicates new file

yellow indicates change to existing file

Commit message

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



☐ .DS_Store

☑ assets/soo_oh.png

☑ index.html

adding headshot and portfolio te:

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 12!