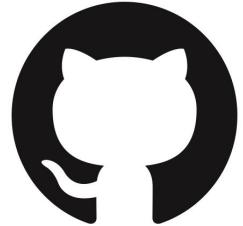
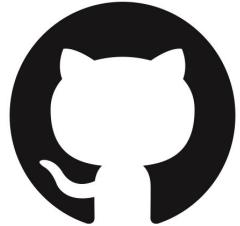


Introduction To GitHub



About Me (TA)

- My name is David Ferrufino
- I am a PhD Student here at VCU
- My TA Hours are in East Hall E 4222 are from:
 - Tues: 1:00 - 2:00
 - Wed: 3:15 - 4:15
- My Email is:
 - ferrufinoda2@vcu.edu



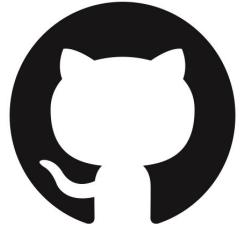
Today's learning objectives

- We will learn about git and how to use git
- We will learn about GitHub and hosting our coding projects on it
- You will learn to build and host your own website with CI/CD (Continuous Integration/Continuous Delivery)

What is Git

- Git is a version control tool used to track changes in your coding projects
- Suppose you write code that breaks your coding project, you can revert back to the latest working version
- Git is the standard version control system across most major software companies





What is Github

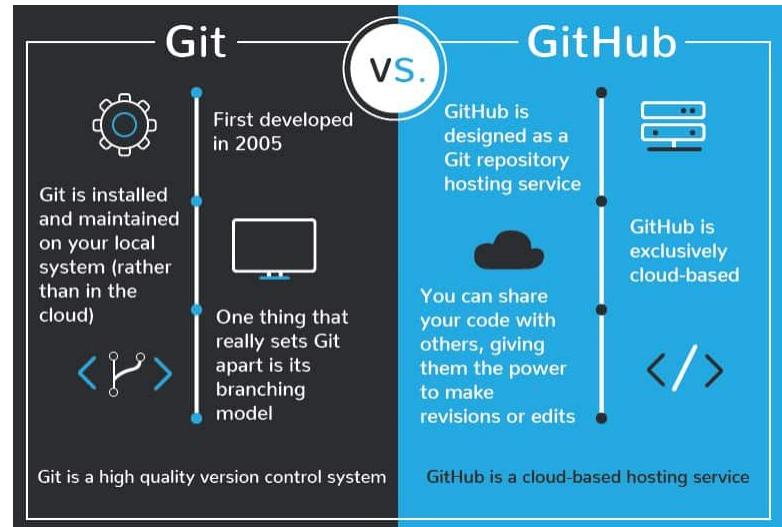
- GitHub is a website that hosts coding repositories (coding project)
- It is used industry wide!
- It handles the version control logistics (where to host & how to handle code changes)

Why use GitHub?

- Prevent Data Loss of your Projects!
- Allows for much easier collaboration
- Lets us structure the coding process
- It's a great place to show off your coding projects
 - As well as demonstrate a portfolio to employers

What is the Difference between GitHub and Git?

- Git is the version control system (it does the pushing, pulling, fetching)
- GitHub is where you can host your projects, and you use Git to manage the version control



How is this better than just saving my files?

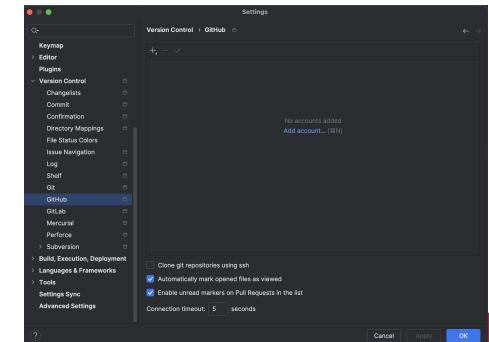
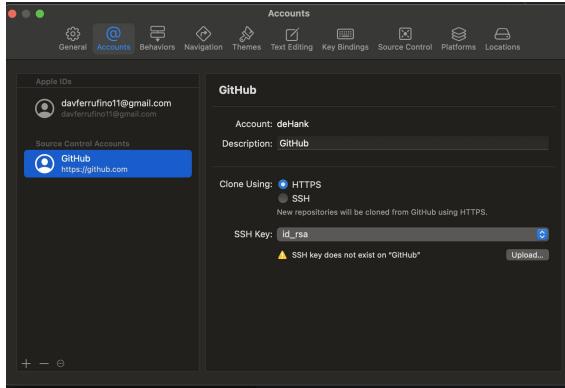
- Github lets multiple people work on different parts of a coding project
- It also allows for storing way more versions than if you just manually saved
- It's also a great way to demonstrate your code to employers!

A little bit of terminology to know...

- Repository: The actual coding project (think of a folder holding all your code)
- Commit: A commit is the changes you have made to a project
 - It's the code you write + a signature from you
 - Note: This doesn't mean that it's submitted to your repository. You can have commits on your PC ready to be sent, but not pushed
- Branch: Branches are copies of your codeline
 - This might be confusing... I'll explain!
- Push: Submission of your list of commits
- Pull: The process of receiving new changes
- Merge: Combining Changes

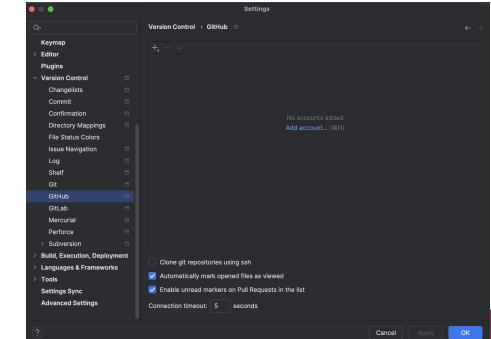
GitHub Clients?

- There are many GitHub Clients!
- You may use the Official GitHub client
- Many IDE's also have a GitHub client built in already
- There is also a Command Line version



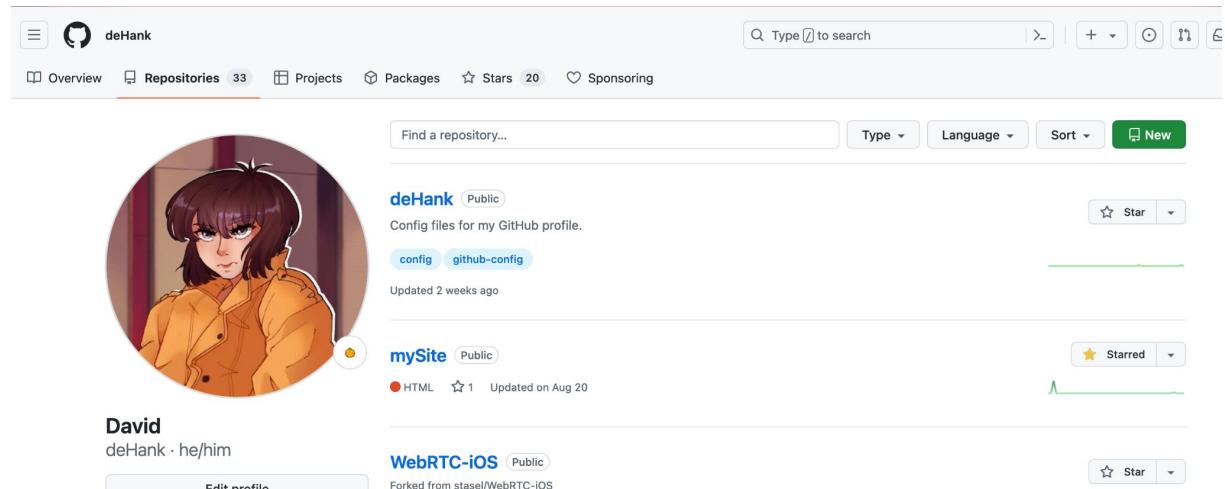
Make sure you have Git Installed (Required)

- GitHub requires Git (to handle the version control)
- If you have a mac install Xtools
 - Open terminal and type: xcode-select --install
- If you have windows:
 - Download the 64-bit Git for Windows Setup and run it
 - [Link here](#)
- If you have Linux:
 - Use your package manager

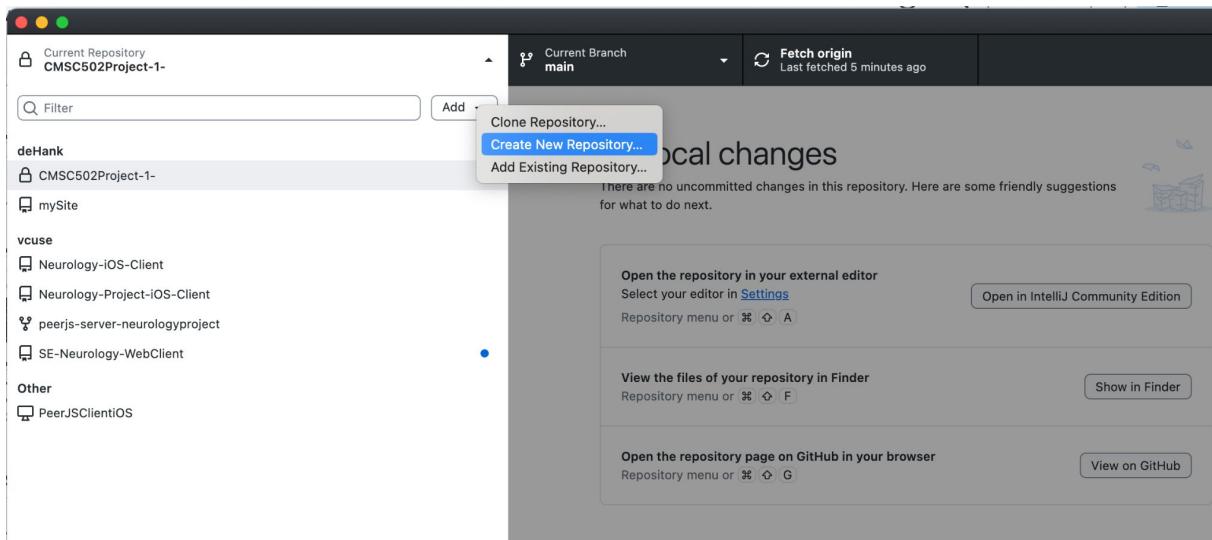


How do I Make A Repository?

- Press the green button that says new



Or Make a new repository in Github Desktop

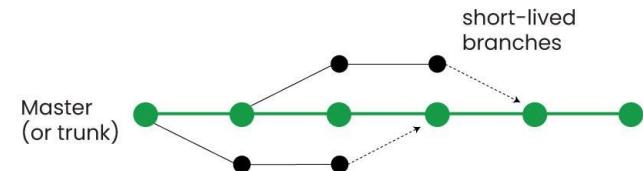


Let's just focus on GitHub Desktop

- Download link here: <https://desktop.github.com/download/>

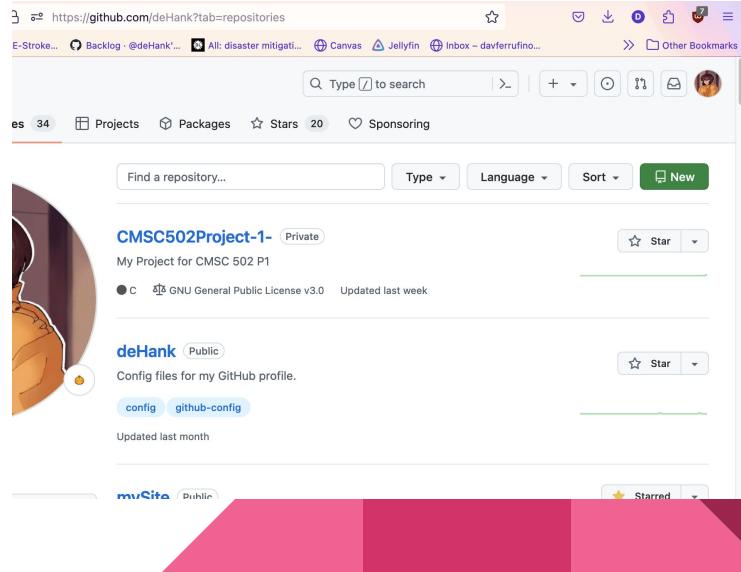
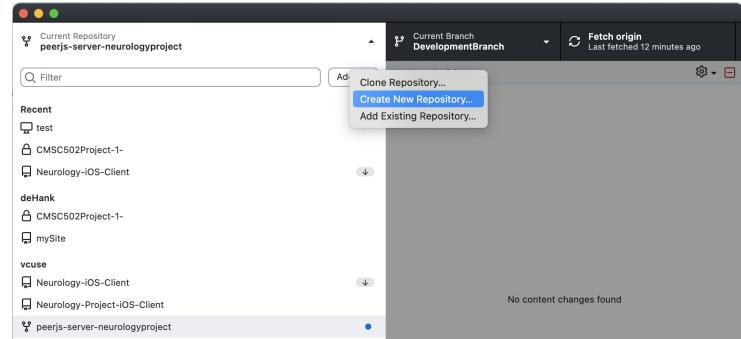
What is the general workflow of GitHub?

- You have many options for how to organize your workflow
- I will teach you Trunk Based Development
 - This is one of the most common ways & a great way to learn version control
- Trunk Based Development involves having a Main Branch with smaller branches for each features/change/bugfix
 - And for each feature or fix, you will make another branch off of that main
 - After you have completed that feature, you will merge those changes from your new branch back into main
 - Do not worry if you are confused about this, I will show you a demonstration



First I'll make a new Repo!

- In GitHub Desktop go to add -> Create New Repository
- Or if using the Website go to your profile → repositories, and click the green new button



Fill in this Information

- Give your repo a name, a description
- Optionally, you may include a README file
 - This is the description that you see when you open up a project

Create a new repository

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

Required fields are marked with an asterisk (*).

Owner * Repository name *

 deHank / Github-Demo

Github-Demo is available.

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

Description (optional)

This is a demonstration of GitHub

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:

Add a README file

This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about Ignoring files](#).

Choose a license

License: GNU General Public License v3.0

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

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

 You are creating a public repository in your personal account.

Create repository

Your repository should look similar to this one!

The screenshot shows a GitHub repository page for a public repository named "Github-Demo".

Repository Header: deHank / Github-Demo

Navigation Bar: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, Settings

Repository Title: Github-Demo (Public)

Branches and Tags: main (selected), 1 Branch, 0 Tags

Search and Add File: Go to file, Add file

Code Tab: Code (selected)

Commit History:

- deHank Initial commit (3f49600 · now) - 1 Commit
- LICENSE (Initial commit, now)
- README.md (Initial commit, now)

File Content Preview:

README: This is a demonstration of GitHub

About: This is a demonstration of GitHub

Activity: Readme, Activity, 0 stars, 1 watching, 0 forks

Releases: No releases published, Create a new release

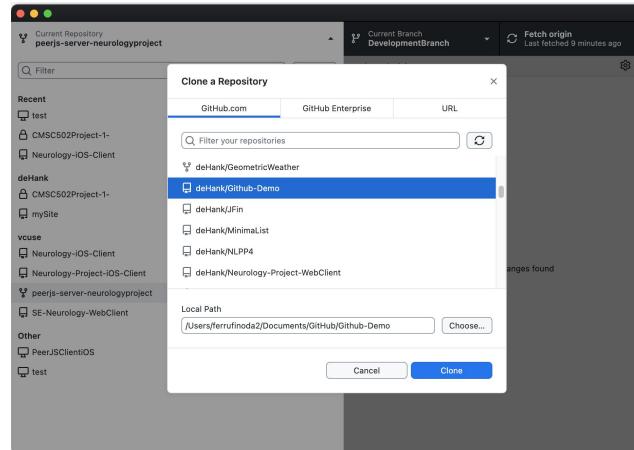
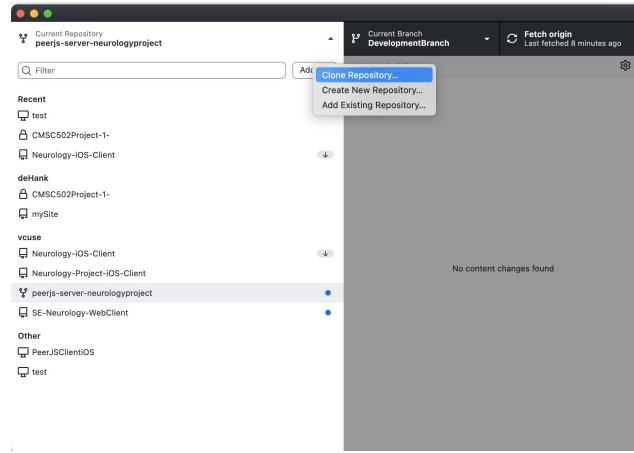
Packages: No packages published, Publish your first package



© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

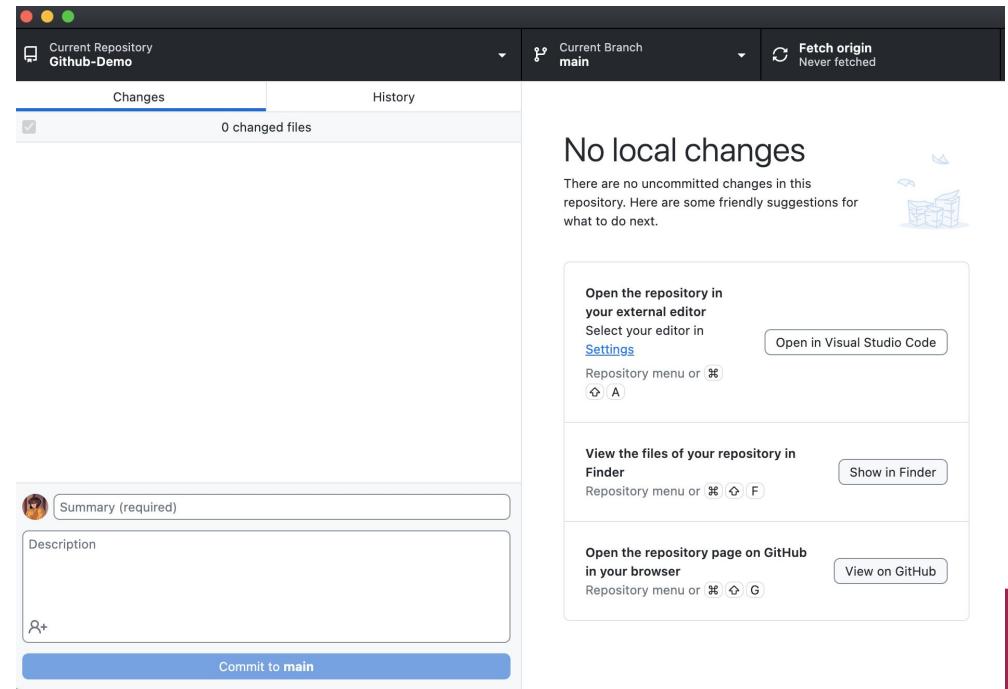
Let's add some code to the repo

- We'll clone our newly created repo
 - This makes a copy, and also automatically sets up git version control for the project

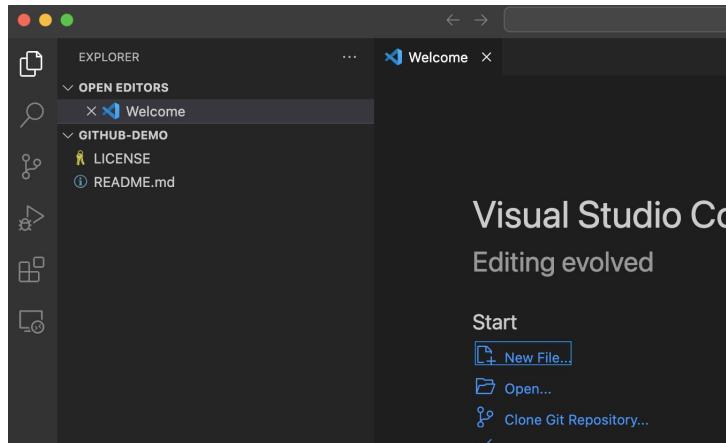


Now we can open it up in our IDE of choice

- I set mine to visual studio code, but you can use literally any editor



In Visual Studio, now we see the files from our GitHub Repository

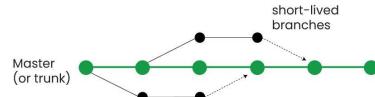


A screenshot of a GitHub repository page for "Github-Demo". The repository is public. It shows one branch ("main"), zero tags, and one commit by "deHank" (Initial commit). The commit was made 3 days ago. There are three files listed: "LICENSE" and "README.md", both of which are initial commits from 3 days ago. Below the files, there is a "README" section with the text "This is a demonstration of GitHub".

File	Commit	Time
LICENSE	Initial commit	3 days ago
README.md	Initial commit	3 days ago

Now let's write some code

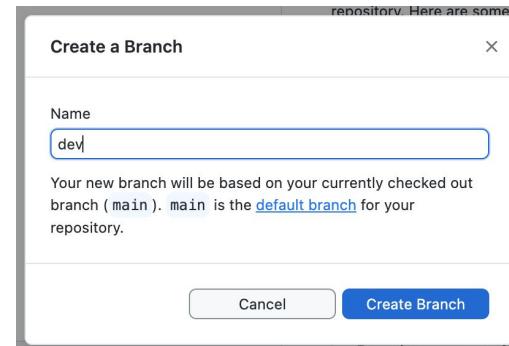
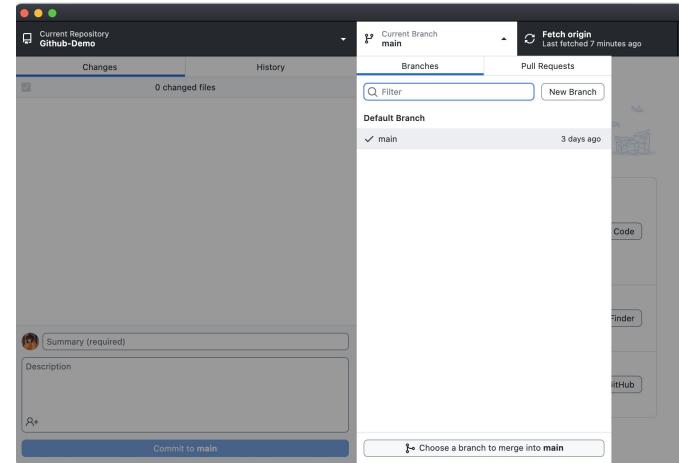
- This is where we use our Trunk-Based workflow
- But first we have to understand our branches
 - We currently only have one main branch →
- The way that we do Trunk-Based development is:
 - For each change/feature we create a new branch
 - Then those branches are merged into main
 - So we must create a new branch!



The screenshot shows a GitHub repository interface for "Github-Demo" (Public). At the top, there are buttons for "main" (selected), "1 Branch", and "0 Tags". Below this is a search bar with the placeholder "Find or create a branch...". Underneath, there are tabs for "Branches" (selected) and "Tags". A single branch named "main" is listed, with a "default" button next to it. Below the branches section, there is a link "View all branches". The repository name "Github-Demo" is displayed prominently, followed by the text "This is a demonstration of GitHub".

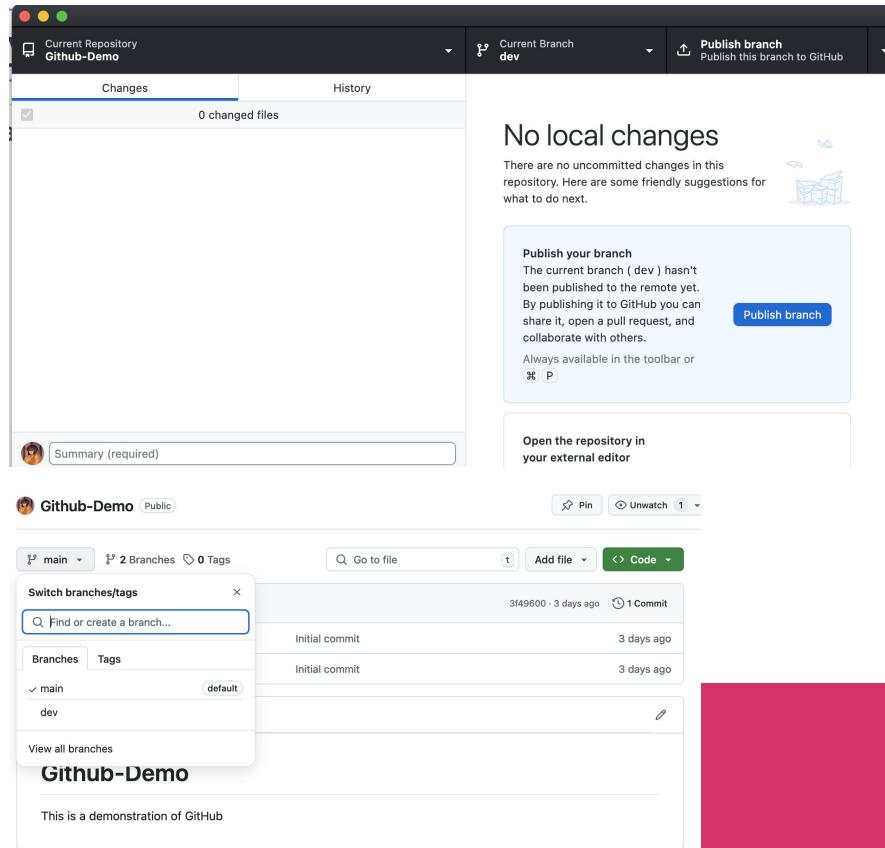
Making a new branch for our code

- In GitHub Desktop, make a new branch by clicking the tab that says “Current Branch” and clicking New Branch
 - Give it a good name!
 - This new branch will be a copy of main, that we will make changes to



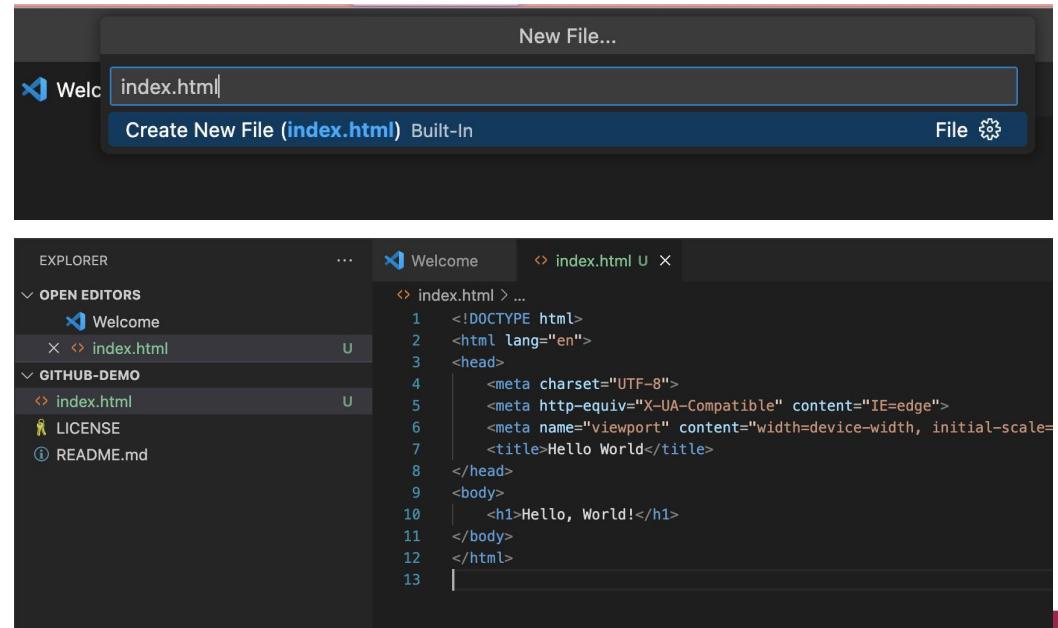
Once you have the new branch click publish branch

- You should then see the new branch in your repository!
 - Note: If you have an issue where you don't see it, make sure you published it



Now we'll add some code to our project

- We go back to our IDE and make a new file and write some code in it



The screenshot shows a dark-themed IDE interface. At the top, there is a search bar with the text "index.html" and a button labeled "Create New File (index.html) Built-In". Below the search bar is a navigation bar with a "File" option. On the left, the "EXPLORER" sidebar shows a tree structure with "OPEN EDITORS" containing "Welcome" and "index.html", and a "GITHUB-DEMO" folder containing "index.html", "LICENSE", and "README.md". The main workspace shows the "index.html" file open, displaying the following code:

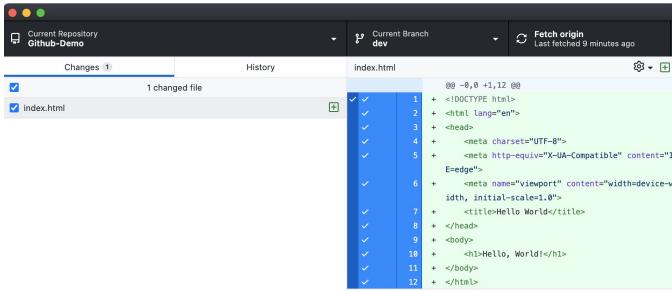
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello World</title>
</head>
<body>
    <h1>Hello, World!</h1>
</body>
</html>
```

But how do we get the code we wrote onto GitHub?

- Once you want to save it, you can commit it (make a list of those changes)
- Then you must PUSH those changes
 - This is the actual part that uploads your code to GitHub

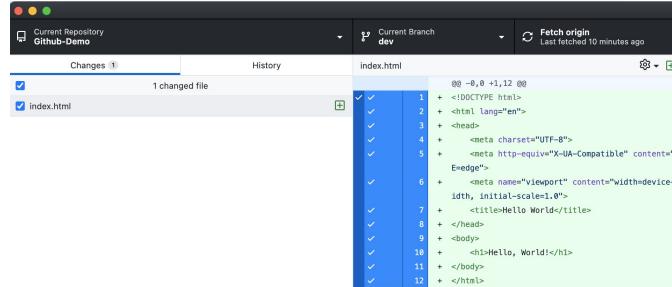
Let's make the commit

- Go back to GitHub Desktop and you will see your code changes with a green checkmark
 - This means they will be staged (included in the commit) for the commit
 - Give this list of changes a good title and description
 - Then just hit commit to dev!



A screenshot of the GitHub Desktop application interface. The top bar shows "Current Repository: Github-Demo", "Current Branch: dev", and "Fetch origin Last fetched 9 minutes ago". The main area is titled "Changes 1" and shows "1 changed file: index.html". The code editor displays the following content:

```
00 -0,0 +1,12 @@  
+ <!DOCTYPE html>  
+ <html lang="en">  
+ <head>  
+   <meta charset="UTF-8">  
+   <meta http-equiv="X-UA-Compatible" content="IE=edge">  
+   <meta name="viewport" content="width=device-width, initial-scale=1.0">  
+   <title>Hello World</title>  
</head>  
<body>  
+   <h1>Hello, World!</h1>  
</body>  
</html>
```



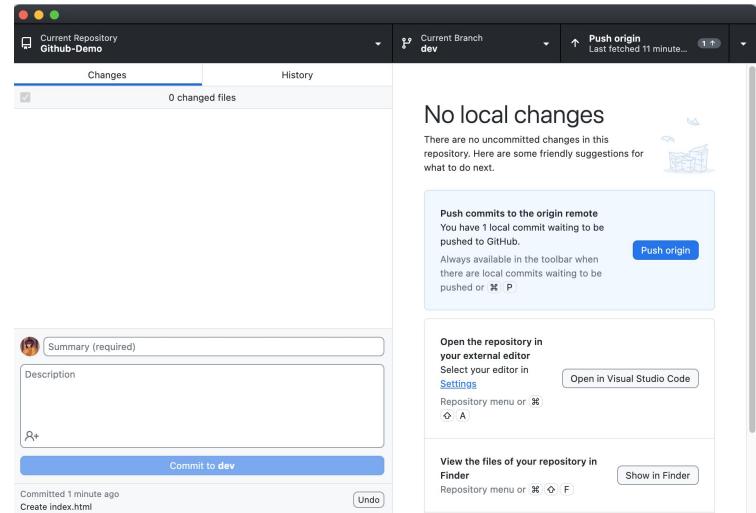
A screenshot of the GitHub Desktop application interface, identical to the first one but with a commit message applied. The top bar shows "Current Repository: Github-Demo", "Current Branch: dev", and "Fetch origin Last fetched 10 minutes ago". The main area is titled "Changes 1" and shows "1 changed file: index.html". The code editor displays the same content as before, with the addition of a commit message at the top:

```
00 -0,0 +1,12 @@  
+ <!DOCTYPE html>  
+ <html lang="en">  
+ <head>  
+   <meta charset="UTF-8">  
+   <meta http-equiv="X-UA-Compatible" content="IE=edge">  
+   <meta name="viewport" content="width=device-width, initial-scale=1.0">  
+   <title>Hello World</title>  
</head>  
<body>  
+   <h1>Hello, World!</h1>  
</body>  
</html>
```



But we need to still push those changes

- After we make our commit, we click push to upload those changes to our dev GitHub branch
- Click push origin & go check GitHub!!



Where are our changes in GitHub? I don't see them...

- Notice that we do not see our new index.html file... what gives?
- We are in the main branch (which we were not editing)
- Our changes were made to the dev branch

The screenshot shows a GitHub repository interface. At the top, a yellow banner indicates "dev had recent pushes 16 seconds ago". Below the banner, the repository navigation bar shows "main" (selected), "2 Branches", "0 Tags", and search/filter options. A green "Compare & pull request" button is also present. The main content area displays three recent commits from user "deHank":

- "Initial commit" (3f49600 · 3 days ago) - Associated with the "LICENSE" file.
- "Initial commit" (3 days ago) - Associated with the "README.md" file.
- "Initial commit" (3 days ago) - Associated with the "README" file, which has a "GPL-3.0 license" link.

A large text box below the commits contains the content of the README file:

Github-Demo
This is a demonstration of GitHub

Look at the dev branch

- Now we see our index.html
- We just need to get this file onto our main branch
- We must open a **Pull Request**

The screenshot shows a GitHub repository interface. At the top, a yellow banner indicates "dev had recent pushes 1 minute ago" and features a "Compare & pull request" button. Below the banner, the repository navigation bar shows "dev" selected, "2 Branches", and "0 Tags". A search bar with "Go to file" and a "t" icon is followed by "Add file" and "Code" buttons. A message states "This branch is 1 commit ahead of main." with a "Contribute" button. The main content area displays four commits from user "deHank":

File	Commit Message	Time
LICENSE	Initial commit	3 days ago
README.md	Initial commit	3 days ago
index.html	Create index.html	3 minutes ago

At the bottom, there are links for "README" and "License" with edit icons.

Github-Demo

What is a pull request?

- A pull request is a list of changes that someone wants to put on another branch
 - It is a request and must be approved
- Notice in this example, lauren wants to merge a change into **main** from a **different branch**

NIH form views #24

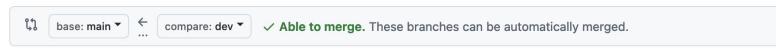
The screenshot shows a GitHub pull request page. At the top, it says "NIH form views #24" with an "Open" button. Below that, there are tabs for "Conversation" (0), "Commits" (1), "Checks" (0), and "Files changed" (12). A comment from "laureniado" is shown, dated 4 days ago, stating: "this has the form UI button inside calls, im working on getting them to save and populate in a different page using core data, theres some of that in this merge but it isn't functioning just yet." Below the comment, there's a reply from "deHank" requesting a review. The pull request summary at the bottom includes sections for "Review requested" (1 pending reviewer), "Continuous integration has not been set up" (GitHub Actions and several other apps can be used to automatically catch bugs and enforce style), and "This branch has no conflicts with the base branch" (Merging can be performed automatically). A green "Merge pull request" button is at the bottom.

So let's make a pull request for our dev changes → main

- Click compare & pull request
- Then click create pull request

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#). Learn more about



Add a title

Add a description

Write Preview

Creating a homepage

Markdown is supported

Paste, drop, or click to add files

Create pull request

A screenshot of the GitHub repository 'Github-Demo'. The repository has 2 branches and 0 tags. It shows a commit from 'deHank' titled 'Create index.html' made 6 minutes ago. The commit details show 'LICENSE', 'README.md', and 'index.html' were created. A green 'Compare & pull request' button is visible at the top right of the commit card.

Review the changes and click merge pull request

Create index.html #1

[! Open](#) deHank wants to merge 1 commit into [main](#) from [dev](#)

Conversation 0 Commits 1 Checks 0 Files changed 1

deHank commented 4 minutes ago

Creating a homepage

→ Create index.html ... 6b988cf

Require approval from specific reviewers before merging
[Rulesets](#) ensure specific people approve pull requests before they're merged.

Continuous integration has not been set up
[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Add a comment

Write Preview

Now your changes are in main!!!

 **Github-Demo** Public

 Pin  Unwatch 1

 main  2 Branches  0 Tags  Go to file  Add file  Code

 deHank Merge pull request #1 from deHank/dev  def2031 · now  3 Commits

 LICENSE	Initial commit	3 days ago
 README.md	Initial commit	3 days ago
 index.html	Create index.html	13 minutes ago

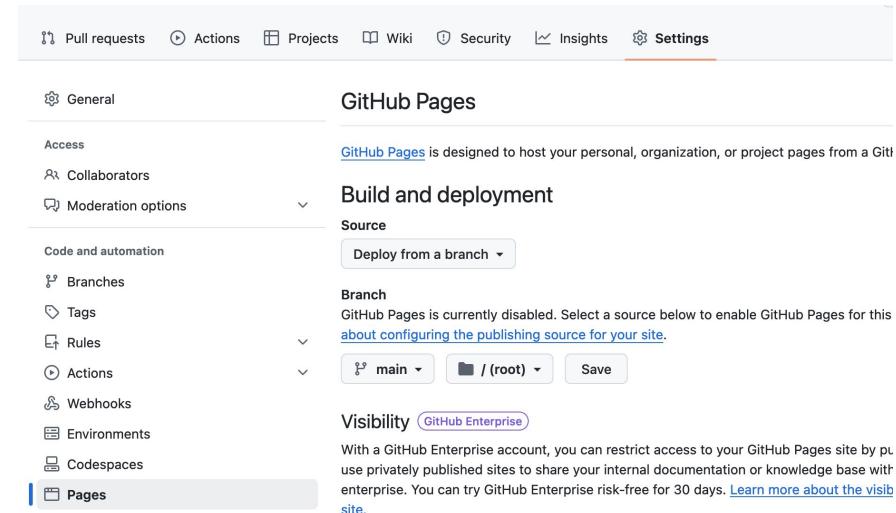
 README  GPL-3.0 license 

Github-Demo

This is a demonstration of GitHub

Want to see something cool? (Hosting your own site)

- GitHub will let you host your websites for free!
- Go to your repository → settings → Pages → Deploy from a branch
 - Set it to main



Click Deployments in the bottom right for the link

 **Github-Demo** Public

[Pin](#) [Unwatch](#) 1 [Fork](#) 0 [Star](#)

[main](#) [2 Branches](#) [0 Tags](#) [Add file](#) [Code](#)

 **deHank** Merge pull request #1 from deHank/dev [...](#)  def2031 · 5 minutes ago  [3 Commits](#)

 LICENSE	Initial commit	3 days ago
 README.md	Initial commit	3 days ago
 index.html	Create index.html	19 minutes ago

[Readme](#) [GPL-3.0 license](#)

Github-Demo

This is a demonstration of GitHub

About

This is a demonstration of GitHub

[Readme](#) [GPL-3.0 license](#)

[Activity](#) [0 stars](#) [1 watching](#) [0 forks](#)

Releases

No releases published [Create a new release](#)

Packages

No packages published [Publish your first package](#)

Deployments 1

 [github-pages now](#)

github-pages deployments

Latest deployments

✓ github-pages

Last deployed 1 minute ago

<https://dehank.github.io/Github-Demo/>



Filter

Filter deployments

1 deployments

✓ Merge pull request #1 from deHank/dev Active

Deployed to github-pages by 🎨 deHank via pages-build-deployment #1

main (#1)

Our website!!

- <https://dehank.github.io/Github-Demo/>
-

Common Questions / Mistakes

- My code isn't showing up on GitHub 😠
 - Did you make sure you pushed?
- My Website isn't loading or making a homepage
 - There is a certain format you must follow (that's why our first file was called index.html)
- What do I do with the branch I made after I put my changes into main?
 - You can delete it or continue to use it. Personally I delete it and make a new branch for each change
- I can't automatically merge! I have to manually select the changes I want
 - This can be caused by conflicting changes! I'll go over this next class



Any Questions?

