

# HHS GitHub Quickstart

---

The official Homestead High School GitHub Quickstart Guide by the wonderful TAs: Rohan Parikh and Neel Sudhakaran.

## Table of Contents

1. [Introduction](#)
2. [Creating A Github Account](#)
3. [Accepting an Assignment](#)
4. [Downloading GitHub Desktop](#)
5. [Cloning a Git Repository](#)
6. [Import a Project into Eclipse](#)
7. [Making a Git Commit](#)
8. [Pushing Changes to GitHub](#)
9. [Conclusion](#)

## Introduction

---

GitHub is a platform that allows you to store your code in the cloud and easily keep track of the changes that you make. You can think of it like Google Docs, but built specifically for developers like you! In this quickstart, you'll learn the essentials needed to get you up and started using GitHub for this class

**Fun Fact:** We're using GitHub right now to host and version control this quickstart guide! Feel free to [check us out](#) there!

## Creating a GitHub Account

To get started with GitHub, you need to create an account. Open [github.com](https://github.com) in your browser and it should take you to a page that looks like this:

The GitHub homepage features a large, glowing blue globe centered against a dark background. A small, stylized cartoon astronaut in a white spacesuit with a blue helmet is positioned to the right of the globe. At the top left, there's a navigation bar with links like 'Why GitHub?', 'Team', 'Enterprise', 'Explore', 'Marketplace', and 'Pricing'. On the top right, there are 'Search GitHub', 'Sign in', and 'Sign up' buttons. The main headline reads 'Where the world builds software' in large, bold, white letters. Below it, a sub-headline says 'Millions of developers and companies build, ship, and maintain their software on GitHub—the largest and most advanced development platform in the world.' There are two buttons: a white one labeled 'Email address' and a green one labeled 'Sign up for GitHub'. At the bottom left, there are four statistics: '65+ million Developers', '3+ million Organizations', '200+ million Repositories', and '72% Fortune 50'.

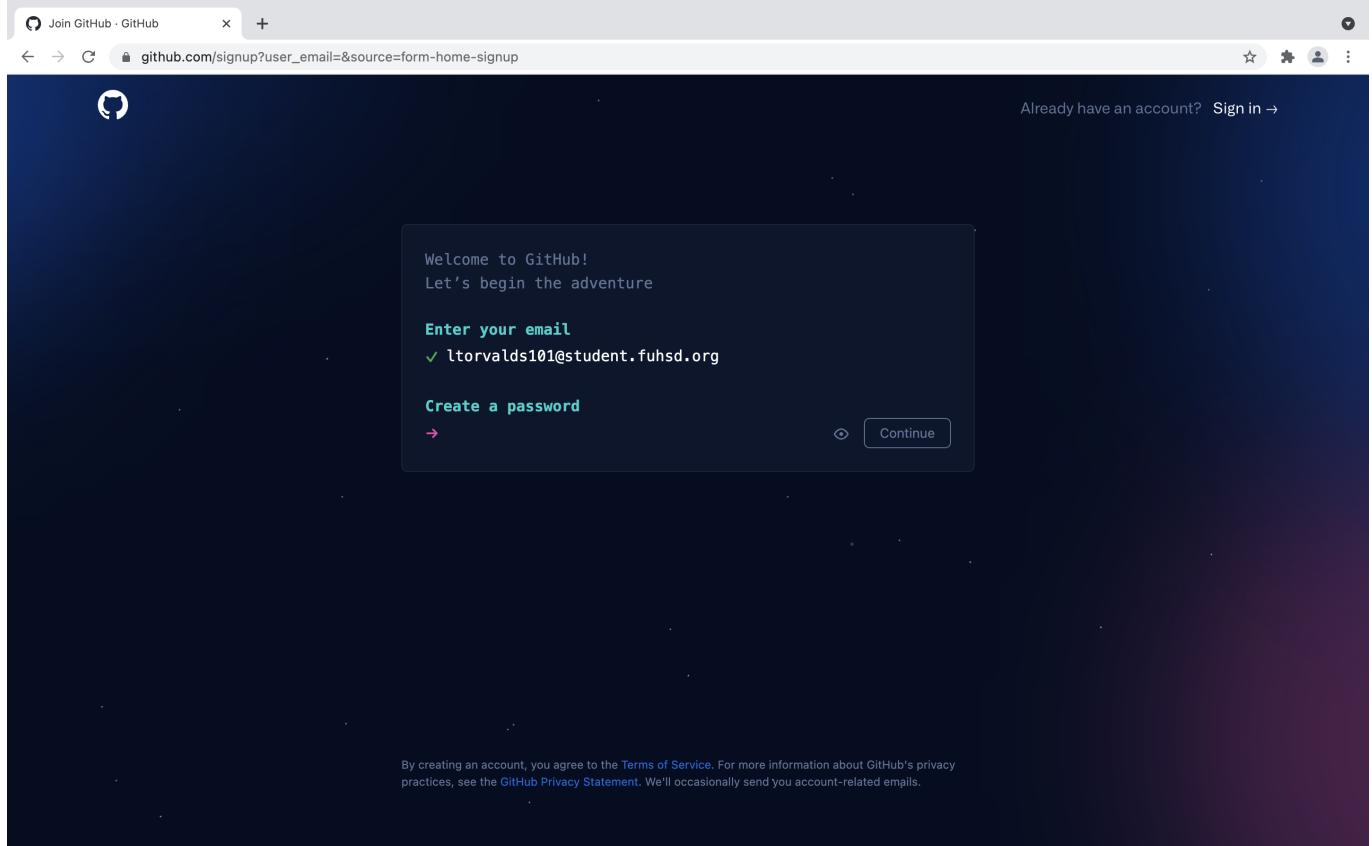
From here, click on the big green button saying "Sign up for GitHub". It should redirect you to a page asking for your email.

The screenshot shows the GitHub sign-up page titled 'Join GitHub'. The main heading is 'Welcome to GitHub! Let's begin the adventure'. Below it, there's a field labeled 'Enter your email' with a pink arrow pointing to it. To the right of the field is a 'Continue' button. In the top right corner, there's a link 'Already have an account? Sign in →'. At the bottom of the page, there's a small note: 'By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#). We'll occasionally send you account-related emails.'

Enter your **school email** and click continue.

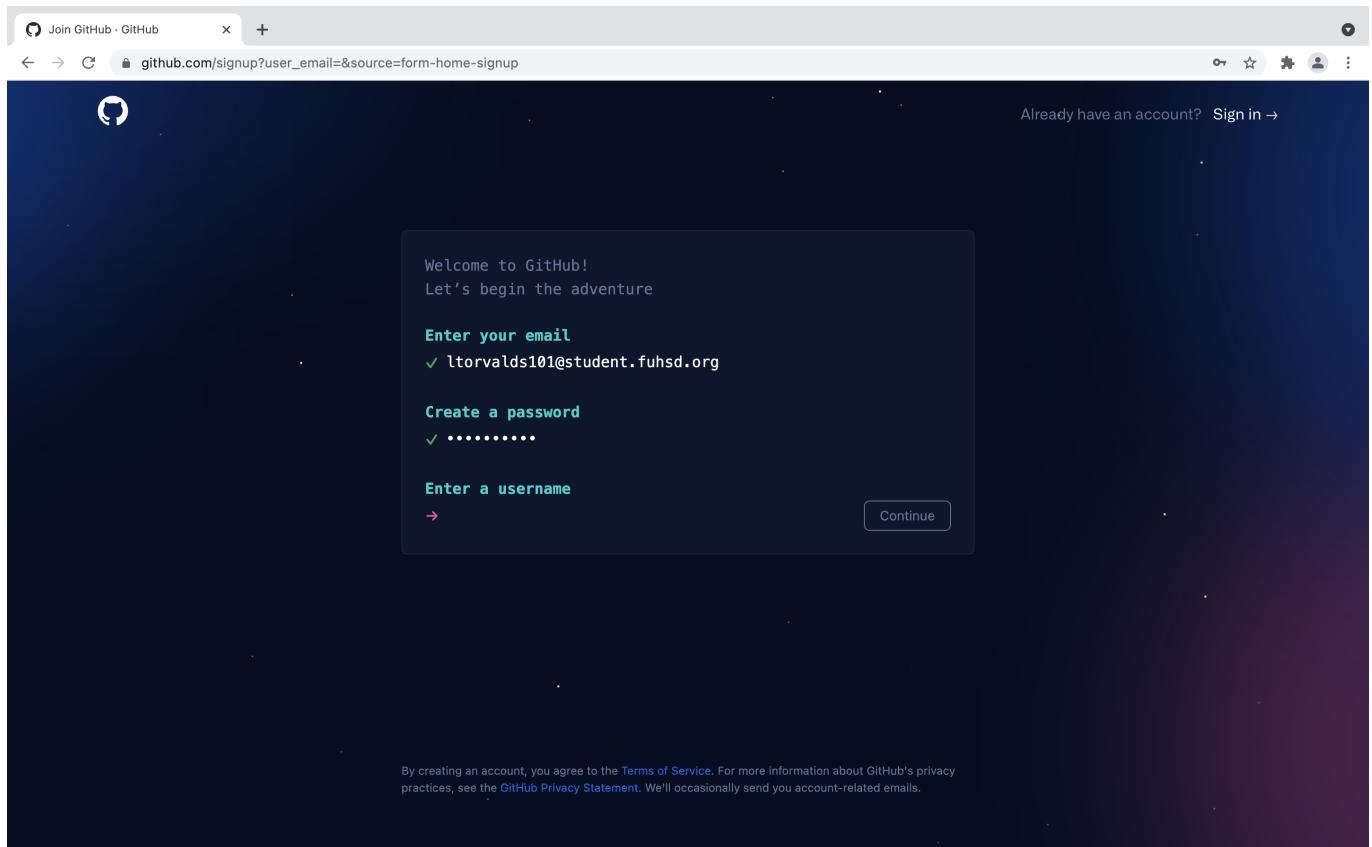
**Note:** Your school email is your "*first initial + first ten letters of your last name + last three digits of your student ID*" at *student.fuhsd.org*. For example, if your name was Linus Torvalds, and your

Student ID was 1100101, your email would be *ltorvalds101@student.fuhsd.org*



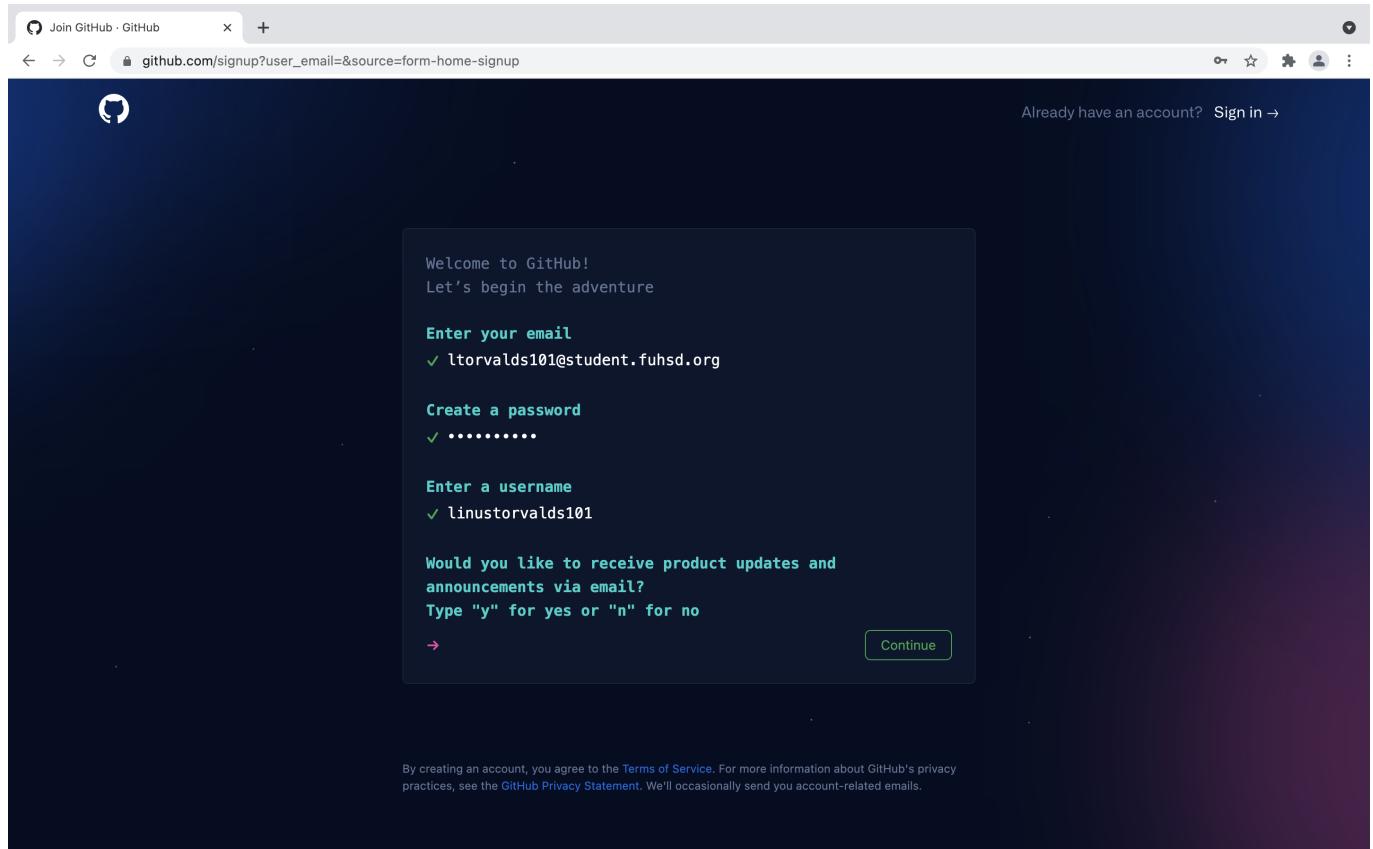
Enter in any password that you like as long as you'll be able to remember it for later use! Then, click continue.

**Note:** Make sure your password is secure! GitHub requires your passwords to have at least 15 characters or at least 8 characters including a number and a lowercase letter.

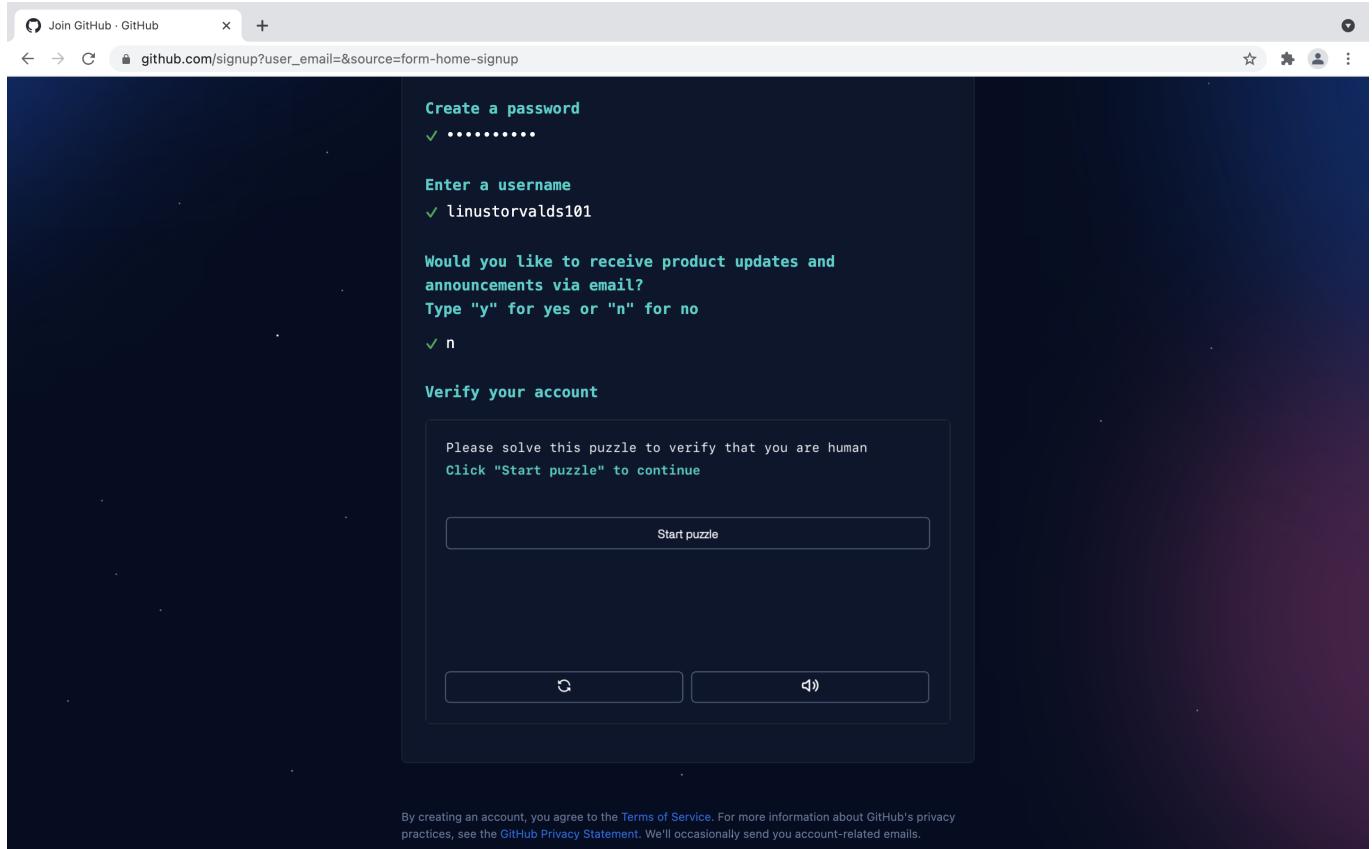


Enter in your username in the format "*first name + last name + last three digits of your Student ID*". After you're finished, click continue.

**Note:** Your GitHub username must be unique, meaning that nobody else who uses GitHub can have your username. In the case that somebody does have your username, try a variation of the suggested username format as long as it at least includes the information in your email. For example, you could try using your full Student ID instead of just the last three digits.

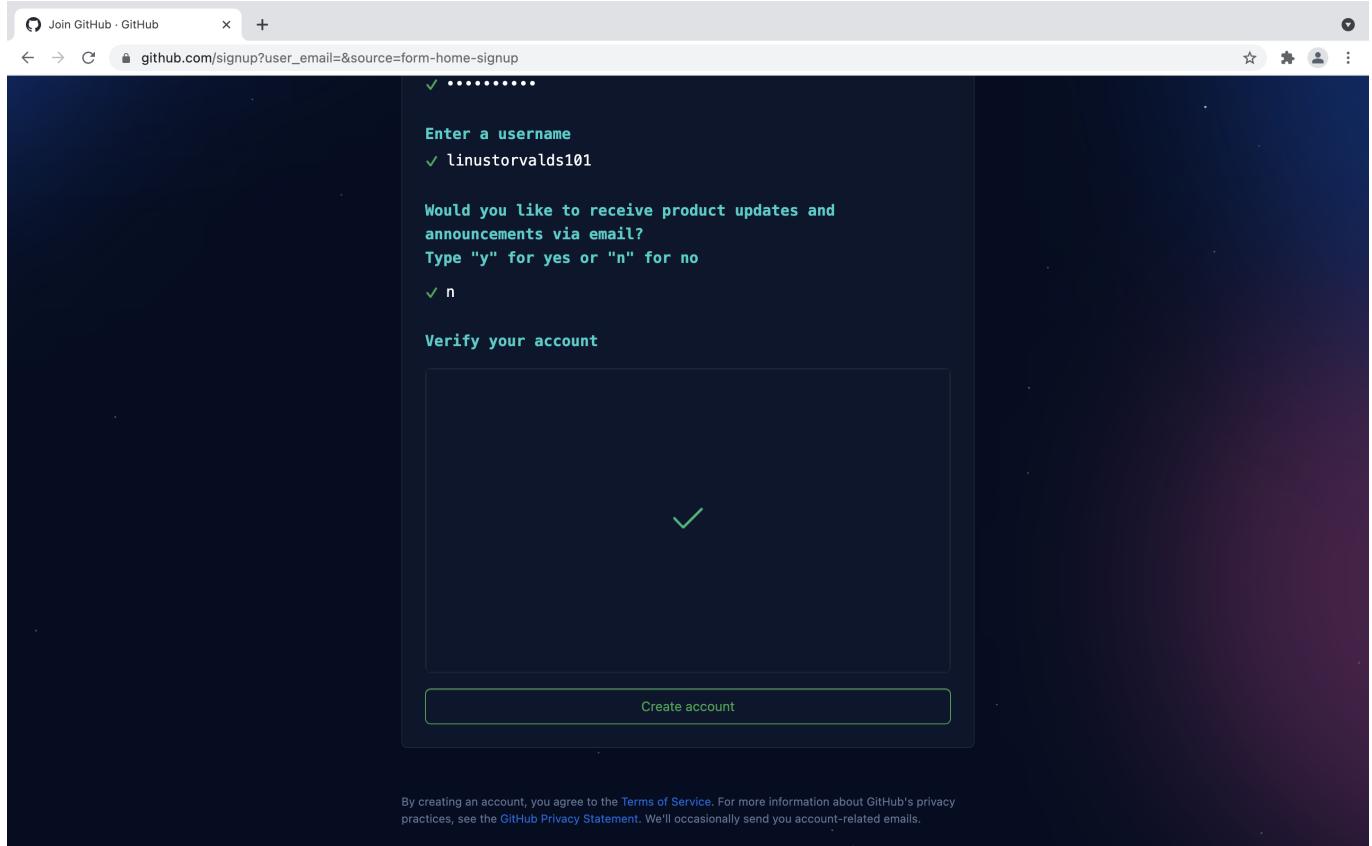


GitHub is now asking you whether or not you want to receive promotional emails from them. You're welcome to do whatever you'd like here, so type "n" if you don't want such emails and "y" if you do.

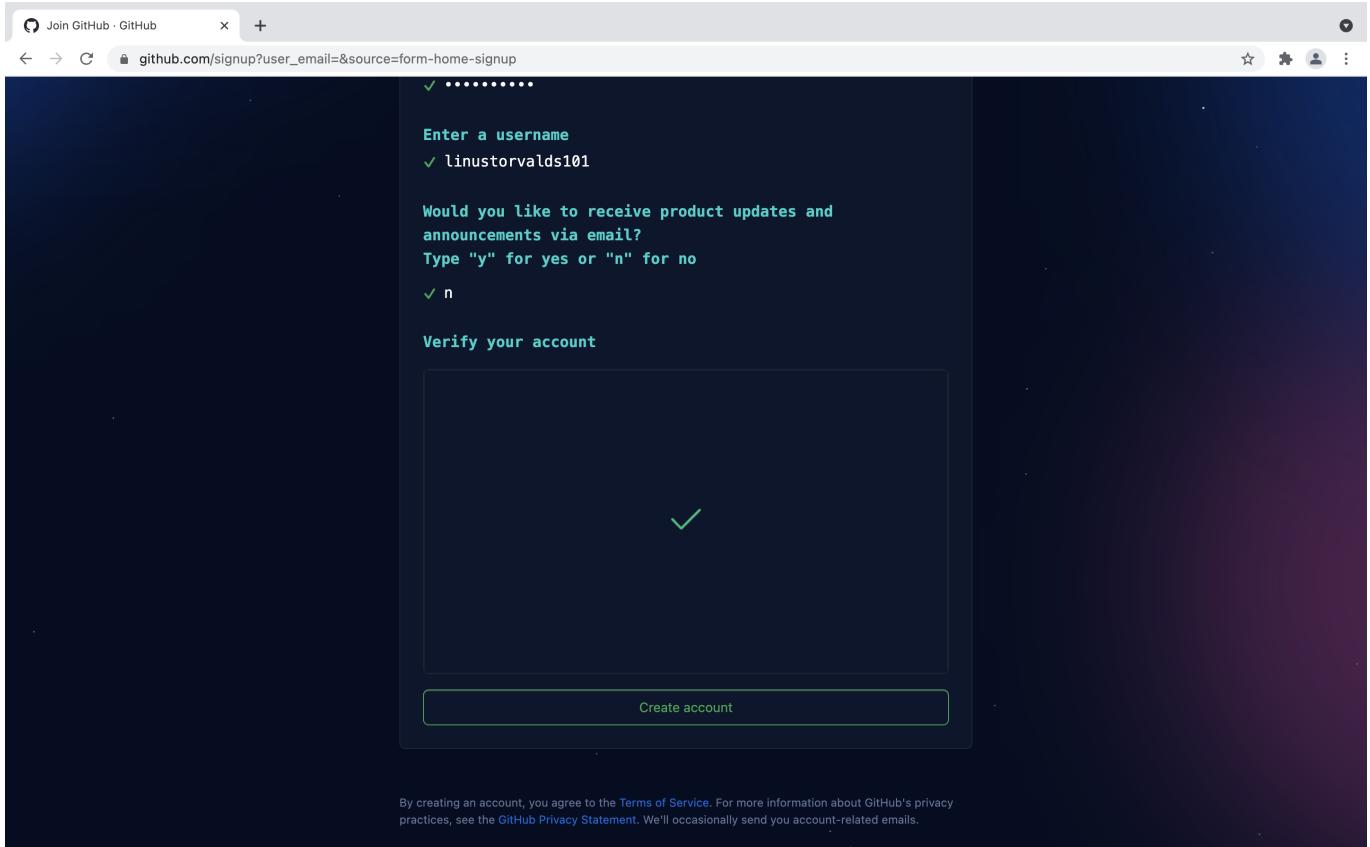


Before you can complete your account creation, you need to solve a simple puzzle to verify you aren't a robot. Click the "Start puzzle" button and complete the puzzle.

**Note:** This puzzle will vary for everyone, so just click the "Start puzzle" button and follow the instructions you are given.



We're almost finished creating your GitHub account! Now that you've entered in all of your information, click the big "Create account" button.



Now you need to verify your email by entering in the code sent to your inbox. Open your school email in a new tab, and then find the verification code email.



Here's your GitHub launch code,  
@linustorvalds101!



Continue signing up for GitHub by entering the code below:

882338

[Open GitHub](#)

Once completed, you can start using all of GitHub's features to explore, build, and share projects.

From here you have two options: You can either return to the tab you were just in previously and enter in the given code, or you can click the big green "Open GitHub" button in the email.

**Note:** If you can't find the email, try looking for emails from noreply@github.com or emails with the subject "🚀 Your GitHub launch code". Check your spam folder!

The screenshot shows the GitHub onboarding process. It starts with a 'Create your first project' section, followed by a 'Learn Git and GitHub without any code!' guide, and a 'Save the Date!' announcement for GitHub Universe. Below these are sections for 'Recent activity', 'All activity', and 'Discover interesting projects and people to populate your personal news feed.'

**Create your first project**  
Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

[Create repository](#) [Import repository](#)

**Recent activity**  
When you take actions across GitHub, we'll provide links to that activity here.

**Learn Git and GitHub without any code!**  
Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

[Read the guide](#) [Start a project](#)

**Save the Date!**  
GitHub Universe is coming October 27 and 28. From product deep dives to interactive roundtables, you'll gather the tips, tools, and connections to help you do the best work of your life.

[Learn more](#)

**All activity**

**Introduce yourself**  
The easiest way to introduce yourself on GitHub is by creating a README in a repository about you! You can start here:

linustorvalds101/README.md

```
1 - 🌟 Hi, I'm @linustorvalds101
2 - 💬 I'm interested in ...
3 - 🚧 I'm currently learning ...
4 - 💡 I'm looking to collaborate on ...
5 - 📩 How to reach me ...
6
```

[Dismiss this](#) [Continue](#)

**Discover interesting projects and people to populate your personal news feed.**  
Your news feed helps you keep up with recent activity on repositories you [watch](#) or [star](#) and people you [follow](#).

Either way, you should be taken to your GitHub dashboard. Congratulations! You've created your very own GitHub Account. You're now ready to move on to the next step which is [accepting an assignment](#).

## Accepting an Assignment

When you're using GitHub all of your project code and changes live inside something called a repository. Although you can create a repository from scratch, for this class you're mostly going to be coding using pre-created repositories given by your teacher.

To find the given repository for an assignment, start by opening up your main class page in Schoology.

Teacher Assistant - 9030: TaylorE p6 T1

Homestead High School

All Materials

- Unit\_01\_Quiz\_02\_TEST
- Unit 02 Quiz 01 Test
- Unit 02 Quiz 02 Test
- lab\_RedCross
- lab\_BullsEye

Upcoming

Wednesday, October 6, 2021

Things you noticed in Quiz 2 11:59 pm

English

Support | Schoology Blog | PRIVACY POLICY | Terms of Use

Schoology S

Now, open up your given lab folder.

**Note:** You may not see your lab folders in the top level view in Schoology because they might be nested inside various sub folders.

Teacher Assistant - 9030: TaylorE p6 T1

lab\_RedCross

https://classroom.github.com/a/1Jxawu1

Prev Next

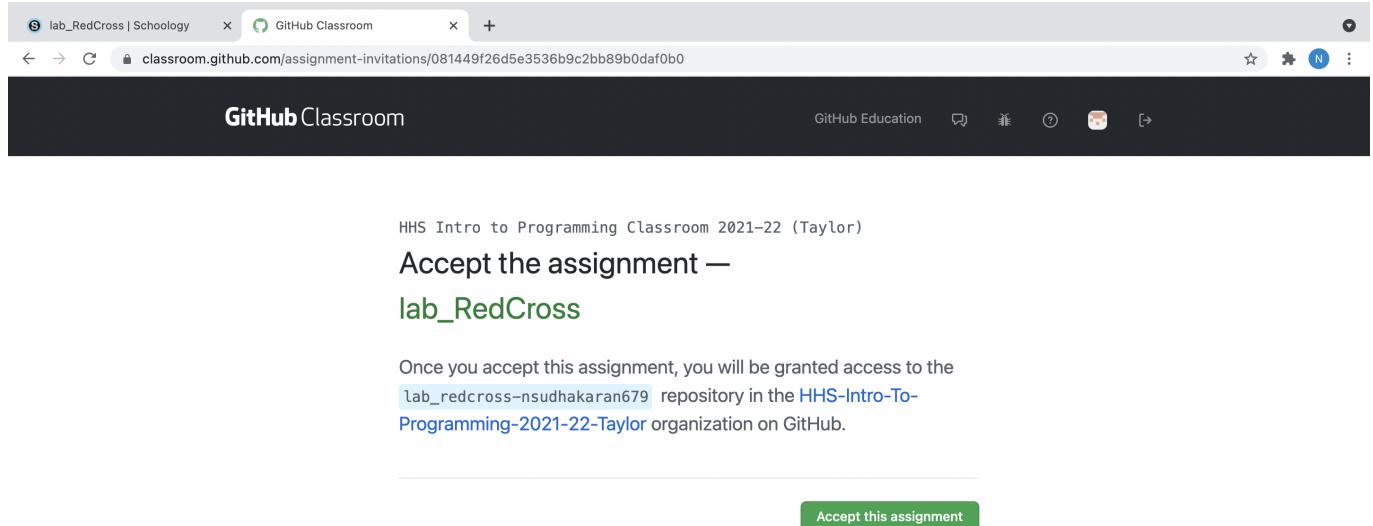
English

Support | Schoology Blog | PRIVACY POLICY | Terms of Use

Schoology S

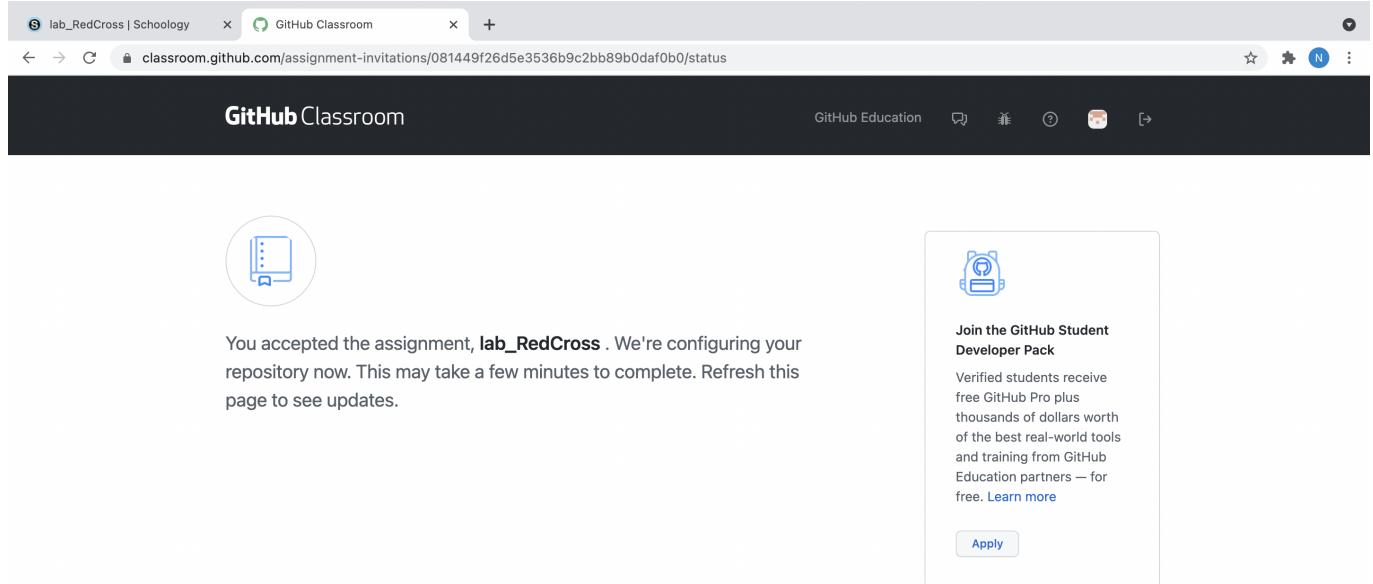
Double click on the [classroom.github.com](https://classroom.github.com/a/1Jxawu1) link inside the folder.

**Note:** You likely have more stuff inside the lab Schoology folder, as this is a simplified version created for this demo.



A screenshot of a web browser showing the GitHub Classroom interface. The title bar says "lab\_RedCross | Schoology" and "GitHub Classroom". The URL is "classroom.github.com/assignment-invitations/081449f26d5e3536b9c2bb89b0daf0b0". The main content area is titled "HHS Intro to Programming Classroom 2021-22 (Taylor)". It displays the message "Accept the assignment — lab\_RedCross". Below this, it says "Once you accept this assignment, you will be granted access to the lab\_redcross-nsudhakaran679 repository in the HHS-Intro-To-Programming-2021-22-Taylor organization on GitHub." At the bottom right is a green button labeled "Accept this assignment".

GitHub Classroom is now asking if you want to accept this assignment. Click the big green button that says "Accept assignment".



A screenshot of a web browser showing the GitHub Classroom interface. The title bar says "lab\_RedCross | Schoology" and "GitHub Classroom". The URL is "classroom.github.com/assignment-invitations/081449f26d5e3536b9c2bb89b0daf0b0/status". The main content area shows a circular icon with a book and a pencil. Below it, the text reads: "You accepted the assignment, lab\_RedCross. We're configuring your repository now. This may take a few minutes to complete. Refresh this page to see updates." To the right, there is a box with a backpack icon and the text: "Join the GitHub Student Developer Pack. Verified students receive free GitHub Pro plus thousands of dollars worth of the best real-world tools and training from GitHub Education partners — for free. [Learn more](#)". At the bottom of this box is a blue "Apply" button.

If you're at this point, GitHub is now creating your project repository for you. Wait a second and then reload the page (the keyboard shortcuts are **Ctrl + R** on Windows and **Cmd + R** on a Mac.)

The screenshot shows a browser window for GitHub Classroom. The URL is [classroom.github.com/assignment-invitations/081449f26d5e3536b9c2bb89b0daf0b0/status](https://classroom.github.com/assignment-invitations/081449f26d5e3536b9c2bb89b0daf0b0/status). The main content area displays a circular icon with a graduation cap and a checkmark, followed by the text "You're ready to go!". Below this, it says "You accepted the assignment, **lab\_RedCross**". It then states "Your assignment repository has been created:" and provides a link: [https://github.com/HHS-Intro-To-Programming-2021-22-Taylor/lab\\_redcross-nsudhakaran679](https://github.com/HHS-Intro-To-Programming-2021-22-Taylor/lab_redcross-nsudhakaran679). A note below the link says "We've configured the repository associated with this assignment ([update](#))". On the right side, there is a sidebar with a backpack icon and the text "Join the GitHub Student Developer Pack". It explains that verified students receive free GitHub Pro plus thousands of dollars worth of tools and training from GitHub Education partners for free, with a link to "Learn more". There is also an "Apply" button.

You're almost ready to go! The last step now is to click on the link, and it will take you to your newly created assignment repository!

**Note:** Your assignment repository link should have the following structure: "[github.com/HHS-Intro-To-Programming-2021-22-Taylor/](https://github.com/HHS-Intro-To-Programming-2021-22-Taylor/) + the lab name + your GitHub username"

The screenshot shows a GitHub repository page for 'HHS-Intro-To-Programming-2021-22-Taylor/lab\_redcross-nsudhakaran679'. The repository is private. The main navigation bar includes 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the navigation, there are tabs for 'Code', 'Issues', 'Pull requests' (1), 'Actions', 'Projects', 'Security', and 'Insights'. The 'Code' tab is selected. On the left, there are buttons for 'main' (with 2 branches and 0 tags), 'Go to file', 'Add file', and 'Code'. The commit history shows three commits from 'github-classroom': 'Setting up GitHub Classroom Feedback' (64b5a1c, 2 minutes ago), '.github' (GitHub Classroom Feedback, 2 minutes ago), and 'lab\_RedCross' (Initial commit, 2 minutes ago). A button 'Add a README' is visible. To the right, there are sections for 'About' (repository details like 'lab\_redcross-nsudhakaran679' created by GitHub Classroom), 'Releases' (no releases published, 'Create a new release'), 'Packages' (no packages published, 'Publish your first package'), and 'Languages' (Java 100%). The footer contains links to GitHub's terms, privacy, security, status, docs, contact, pricing, API, training, blog, and about sections.

Congratulations! You've now created your lab repository and can get started working on your lab! Move on to the next step which is [downloading GitHub Desktop](#).

## Downloading GitHub Desktop

Now that you've gotten your repository set up in GitHub, you need some way of accessing it on your local machine. The easiest way to get started with this is another GitHub tool, GitHub Desktop.

To get started, go to [desktop.github.com](https://desktop.github.com).

The screenshot shows the GitHub Desktop application window on a Windows desktop. At the top, there's a browser-like header with tabs, back/forward buttons, and a search bar. Below the header is the GitHub logo and navigation links for Overview, Release Notes, and Help. A large globe icon is on the right. The main content area features the title "GitHub Desktop" and a subtitle: "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." A prominent purple button labeled "Download for Windows (64bit)" is centered. Below it, smaller text provides links for "Download for macOS or Windows (msi)" and a note about accepting terms. On the left, there's a sidebar with a globe icon and a list of repository items. The main pane shows a list of commits in a pull request, with one commit highlighted.

This screenshot shows the GitHub Desktop application window on a Mac desktop. The interface is similar to the Windows version, with a browser header, GitHub logo, and navigation links. The main content area features the title "GitHub Desktop" and a subtitle: "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." A purple button labeled "Download for macOS" is centered. Below it, smaller text provides links for "Download for Windows" and a note about accepting terms. On the left, there's a sidebar with a globe icon and a list of repository items. The main pane shows a list of commits in a pull request, with one commit highlighted.

This screenshot shows a close-up of the GitHub Desktop interface. It displays the repository name "desktop" and the current branch "esc-pr". The "Changes" tab is selected, showing a single commit: "Add event handler to dropdown component" by iAmWillShepherd and Markus Olson. The commit message includes "Co-Authored-By: Markus Olson <niik@users.noreply.github.com>". The commit hash is c79e71c. The commit was last fetched 2 minutes ago. The "History" tab is also visible below the commit details.

Depending on whether you're on a Windows machine or on a Mac, click the big purple "Download for Windows (64bit)" button or the "Download for macOS". Or alternatively, use these links to download the [Windows version](#) and the [MacOS version](#).

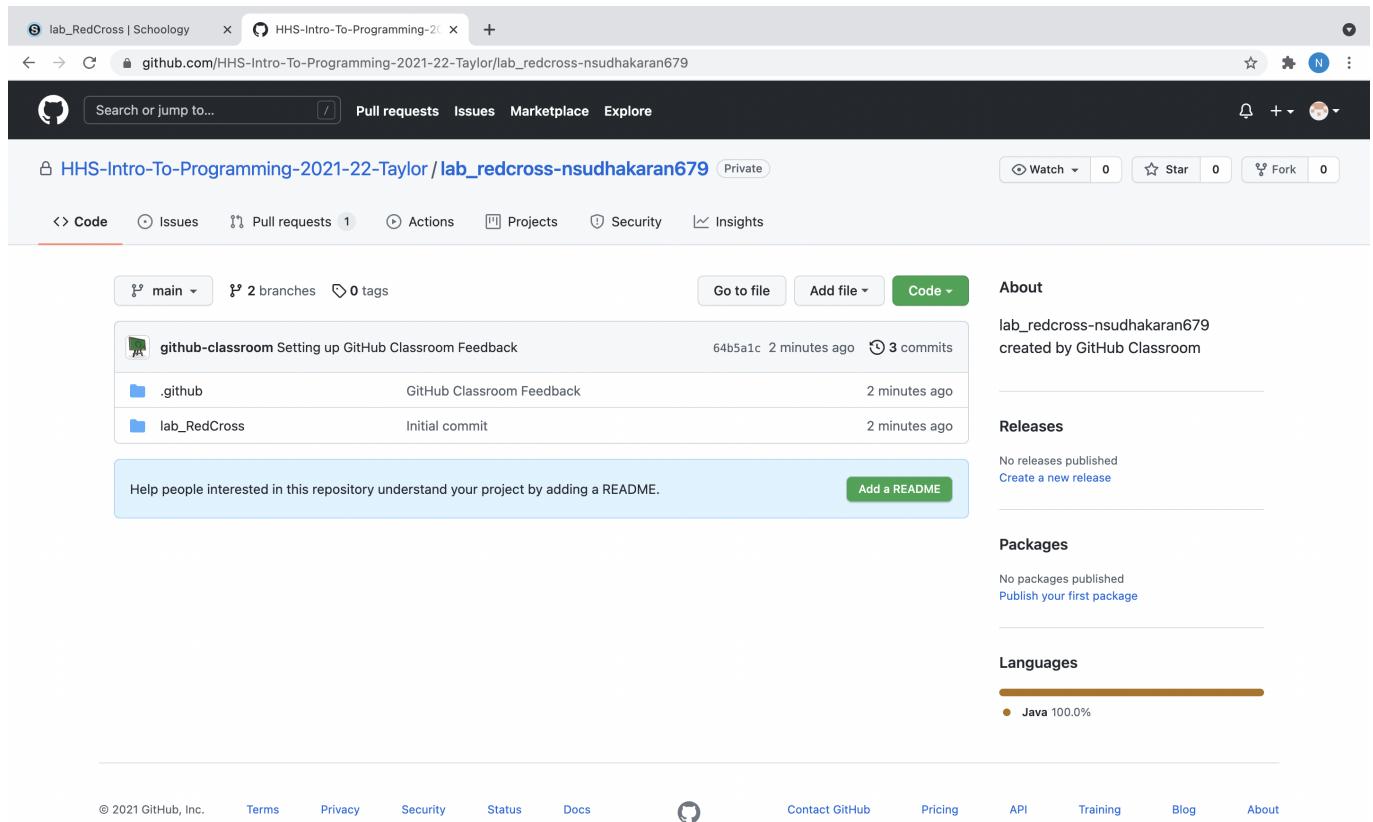
**Note:** If you are working on an Apple Silicon Machine, there's a special link to download for [the native M1 build](#).

Then, download as normal.

## Cloning a Git Repository

Now that we've downloaded GitHub Desktop, let's use it to access our assignment repository on our local machine. The process to do this is called **cloning**, as we're making a clone of the **remote** repository (the one on GitHub) to have access to it locally.

Start by pulling up your assignment repository on GitHub.



The screenshot shows a GitHub repository page for 'HHS-Intro-To-Programming-2021-22-Taylor/lab\_redcross-nsudhakaran679'. The 'Code' tab is selected. A green button labeled 'Code' with a dropdown arrow is visible. The page displays recent commits from 'github-classroom' and 'lab\_RedCross'. It includes sections for About, Releases, Packages, and Languages. The footer contains links to GitHub's terms, privacy, security, status, docs, contact, pricing, API, training, blog, and about pages.

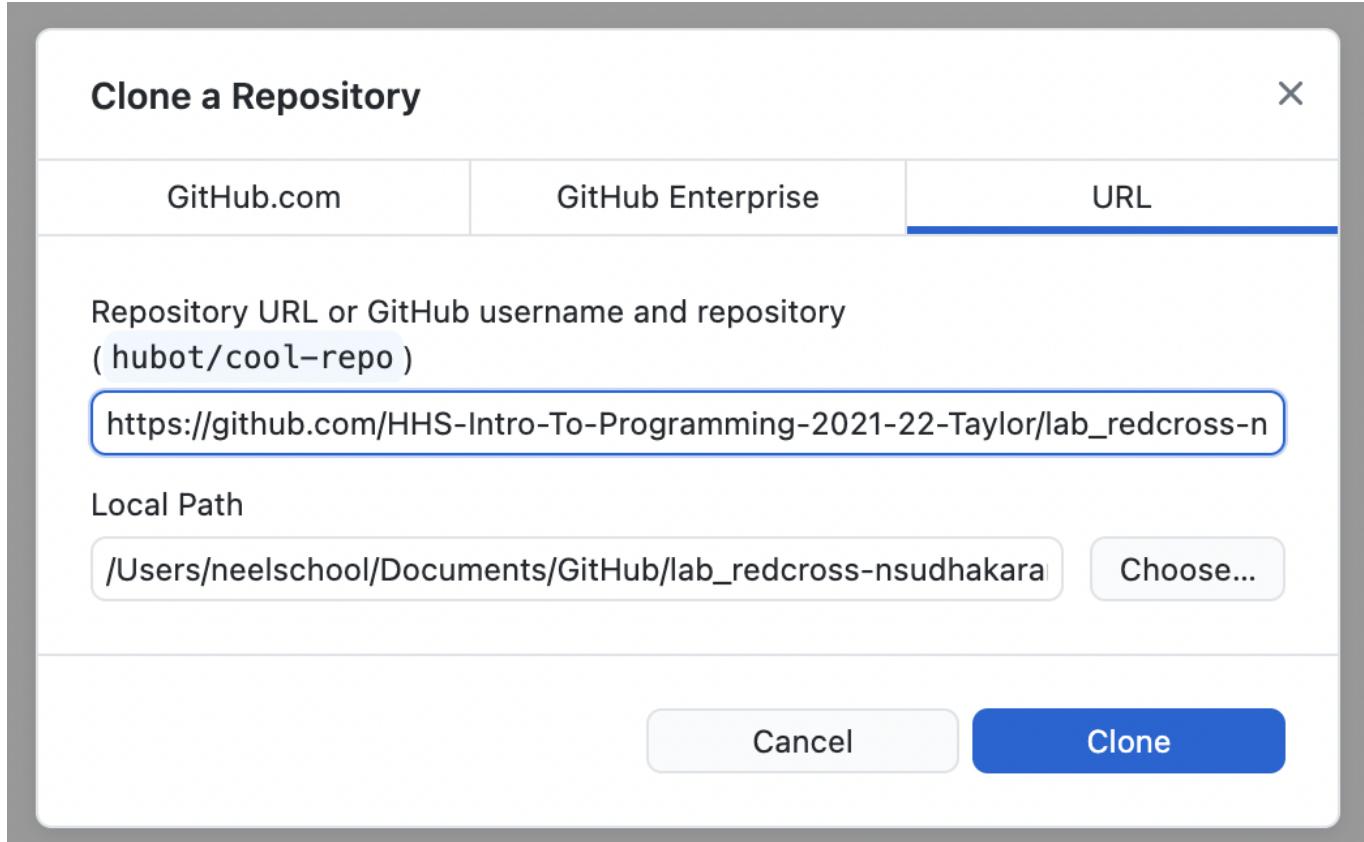
Click on the big green button saying "Code" and it should pull up a dropdown menu like the picture below.

A screenshot of a GitHub repository page for 'HHS-Intro-To-Programming-2021-22-Taylor/lab\_redcross-nsudhakaran679'. The 'Code' dropdown menu is open, showing options like 'Clone', 'HTTPS', 'SSH', 'GitHub CLI', 'Open with GitHub Desktop', and 'Download ZIP'. The 'Open with GitHub Desktop' option is highlighted.

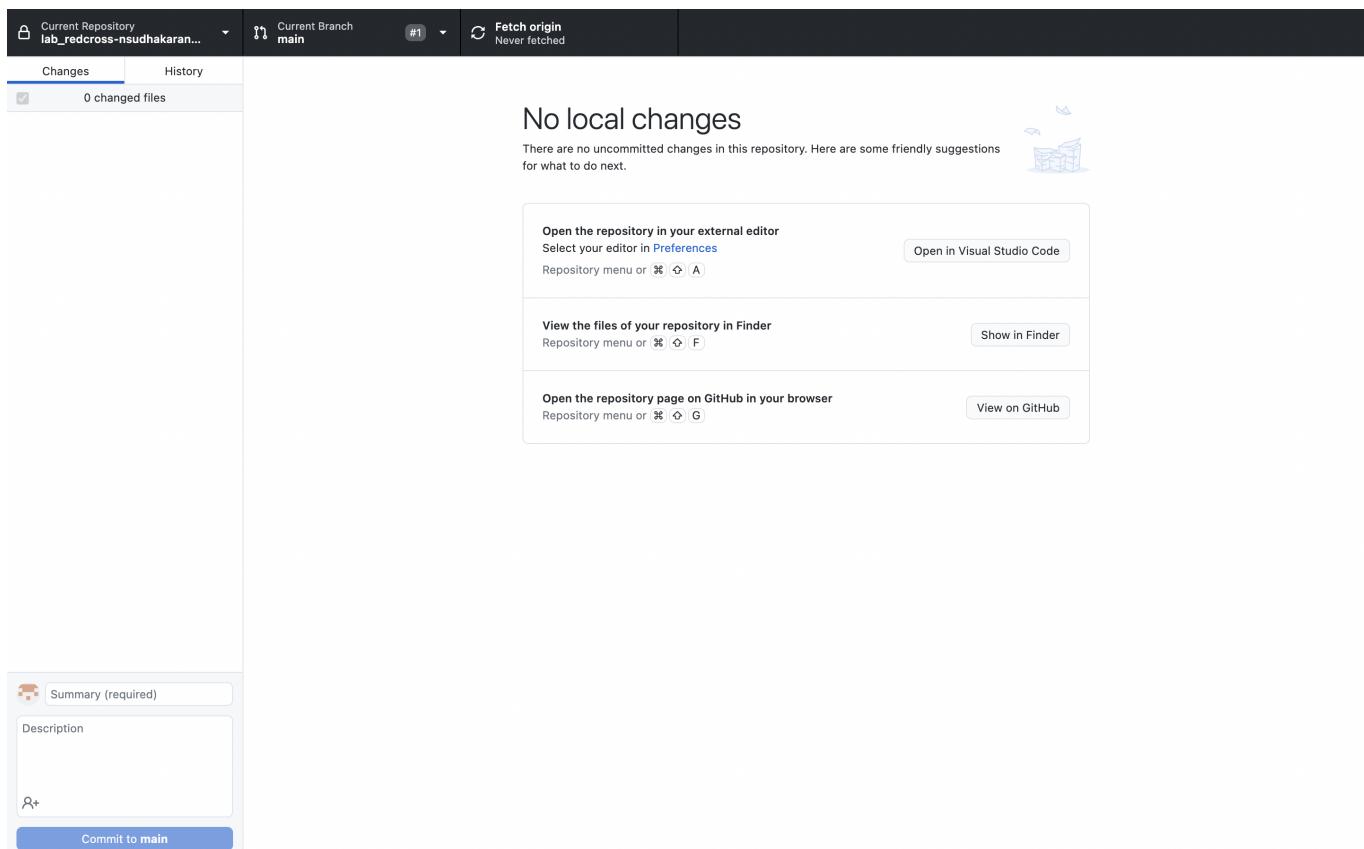
Click on the option that says "Open with GitHub Desktop" and it should prompt you to open GitHub Desktop.

A screenshot of a GitHub repository page for 'HHS-Intro-To-Programming-2021-22-Taylor/lab\_redcross-nsudhakaran679'. A modal window titled 'Open GitHub Desktop?' appears, asking if the user wants to open the application. It includes a checkbox for 'Always allow github.com to open links of this type in the associated app' and buttons for 'Cancel' and 'Open GitHub Desktop'. Below the modal, the GitHub desktop application interface is shown, displaying the same repository details.

Click "Open GitHub Desktop" and it should open GitHub Desktop with the following prompt:



Click the big blue "Clone" button, and GitHub Desktop should bring you to your a page specifically meant for your current lab.



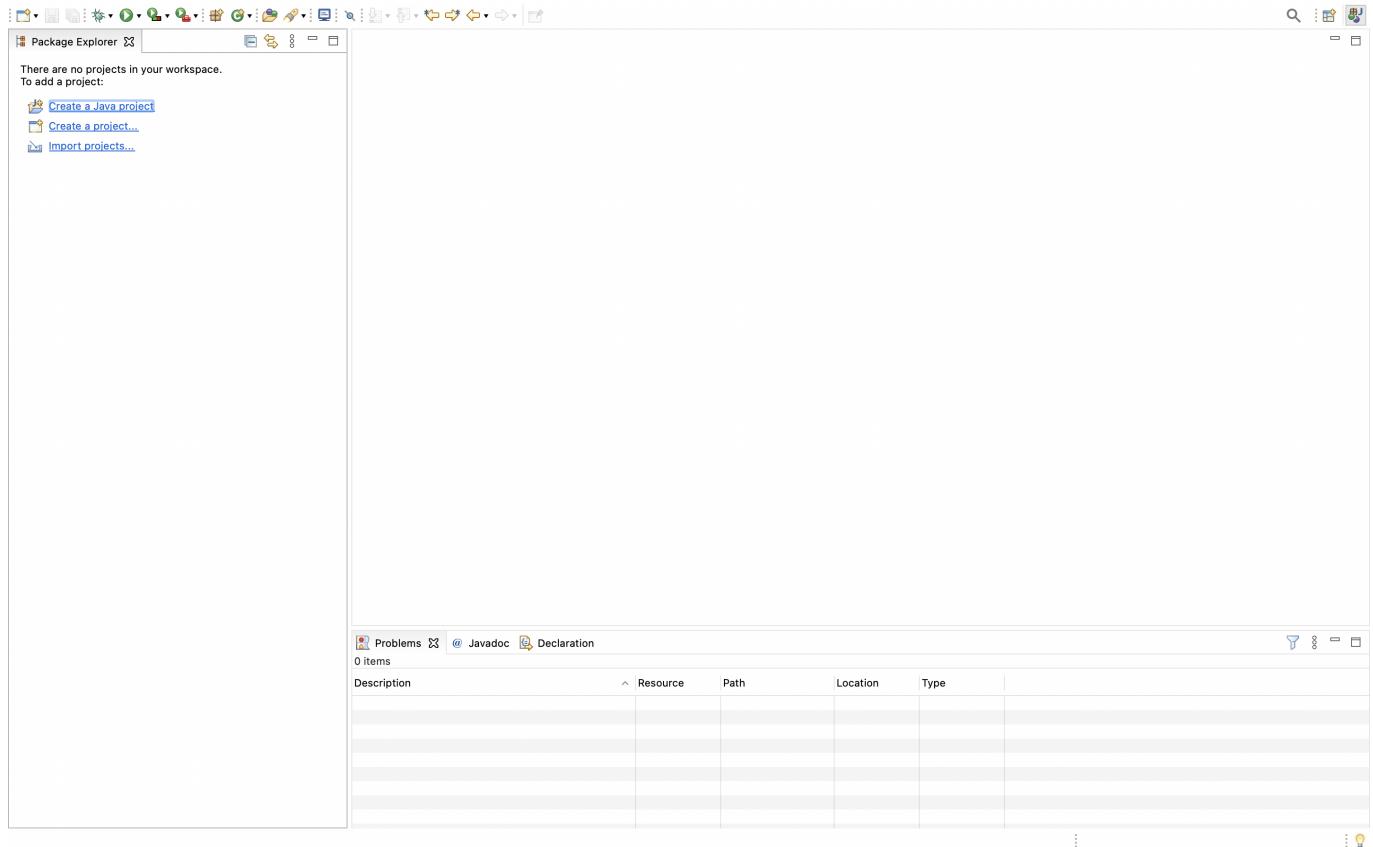
Congratulations! If you've gotten to this page, it means you've successfully cloned your lab repository to your local machine. Continue to the next step to learn how to [import a project into Eclipse](#).

# Import a Project into Eclipse

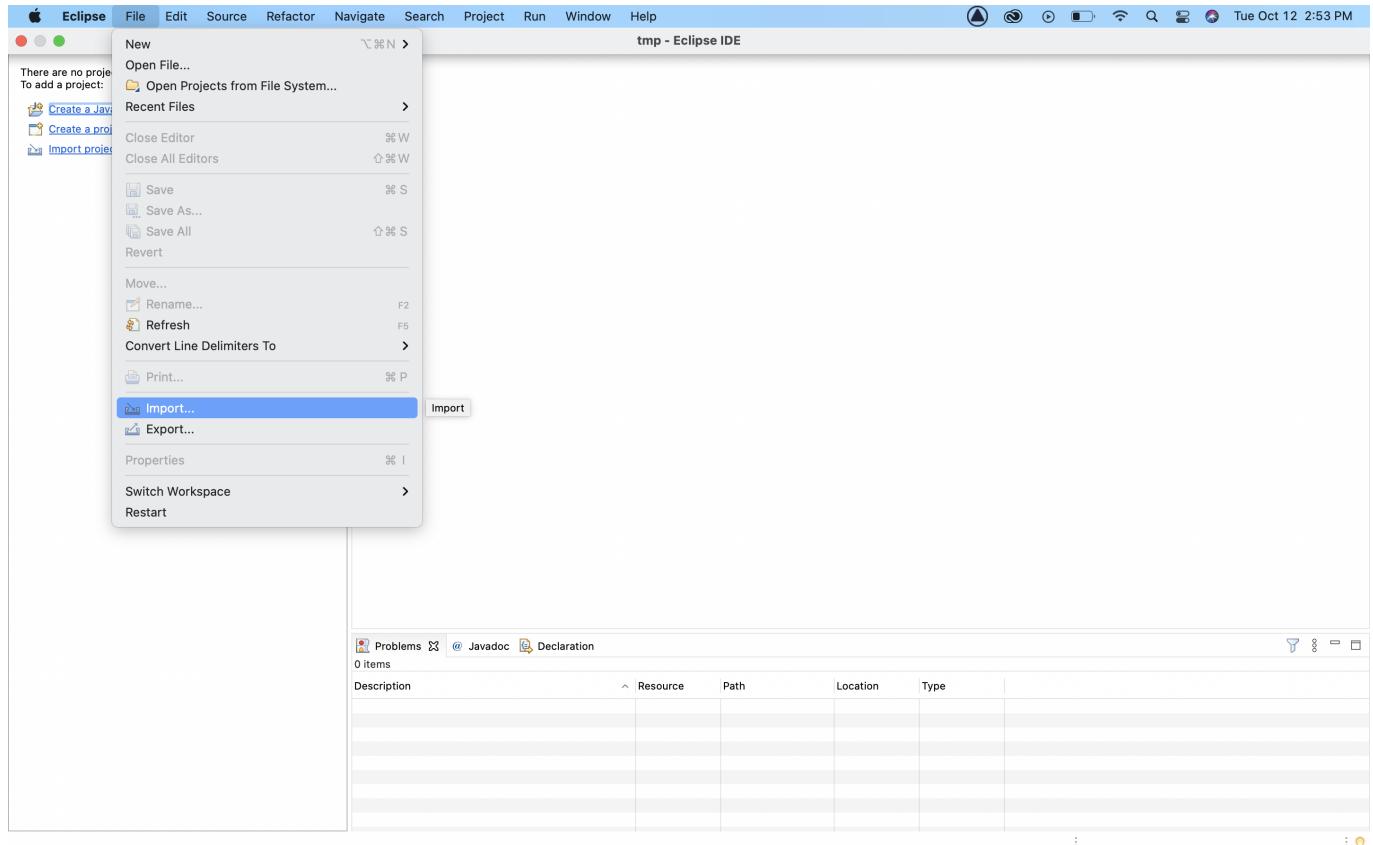
Having our project on our local machine is useless unless we can access the code in our IDE, Eclipse.

Start by opening up your Eclipse workspace.

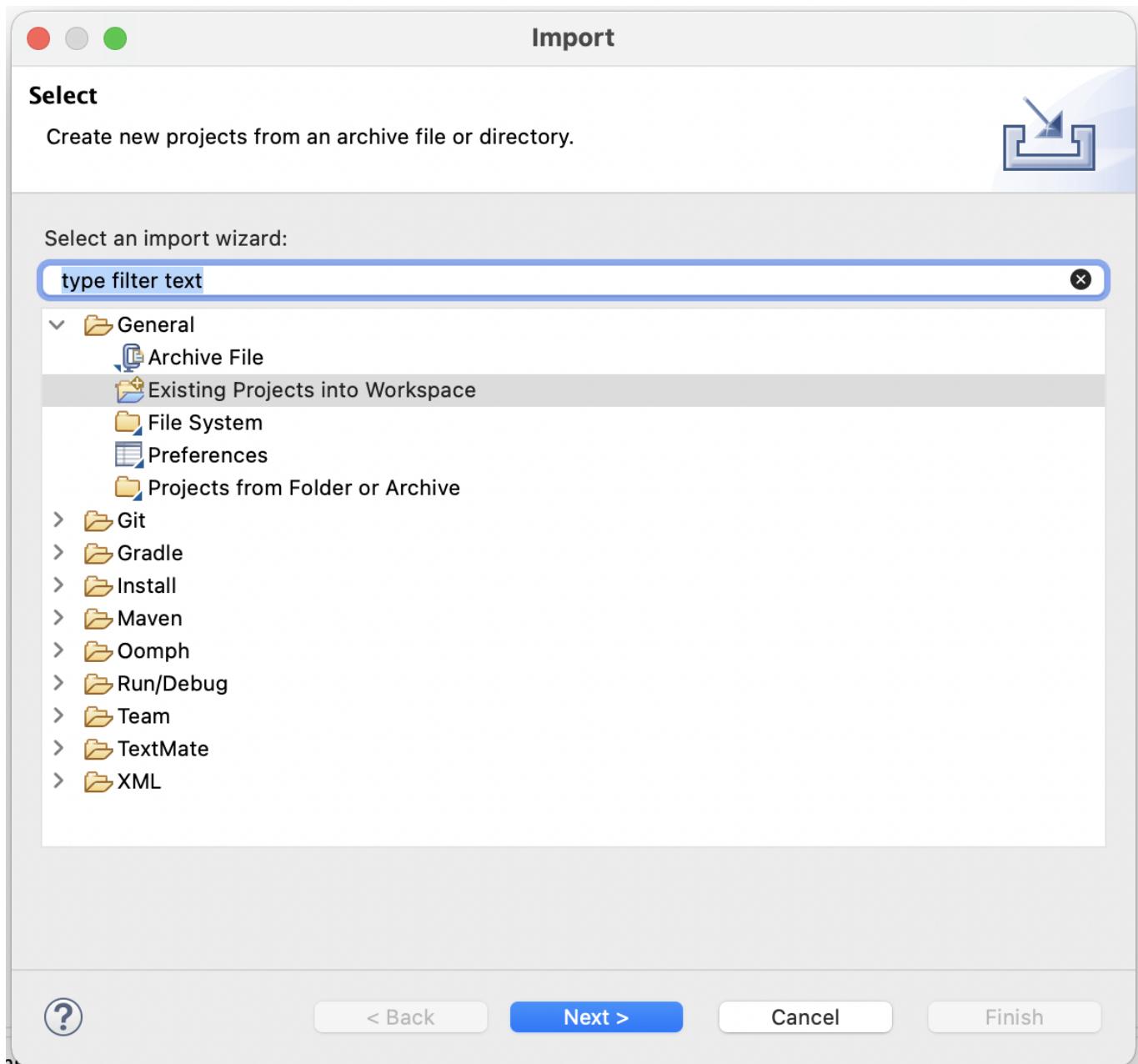
**Note:** You likely have more projects in your workspace, we just created this one as a demo.



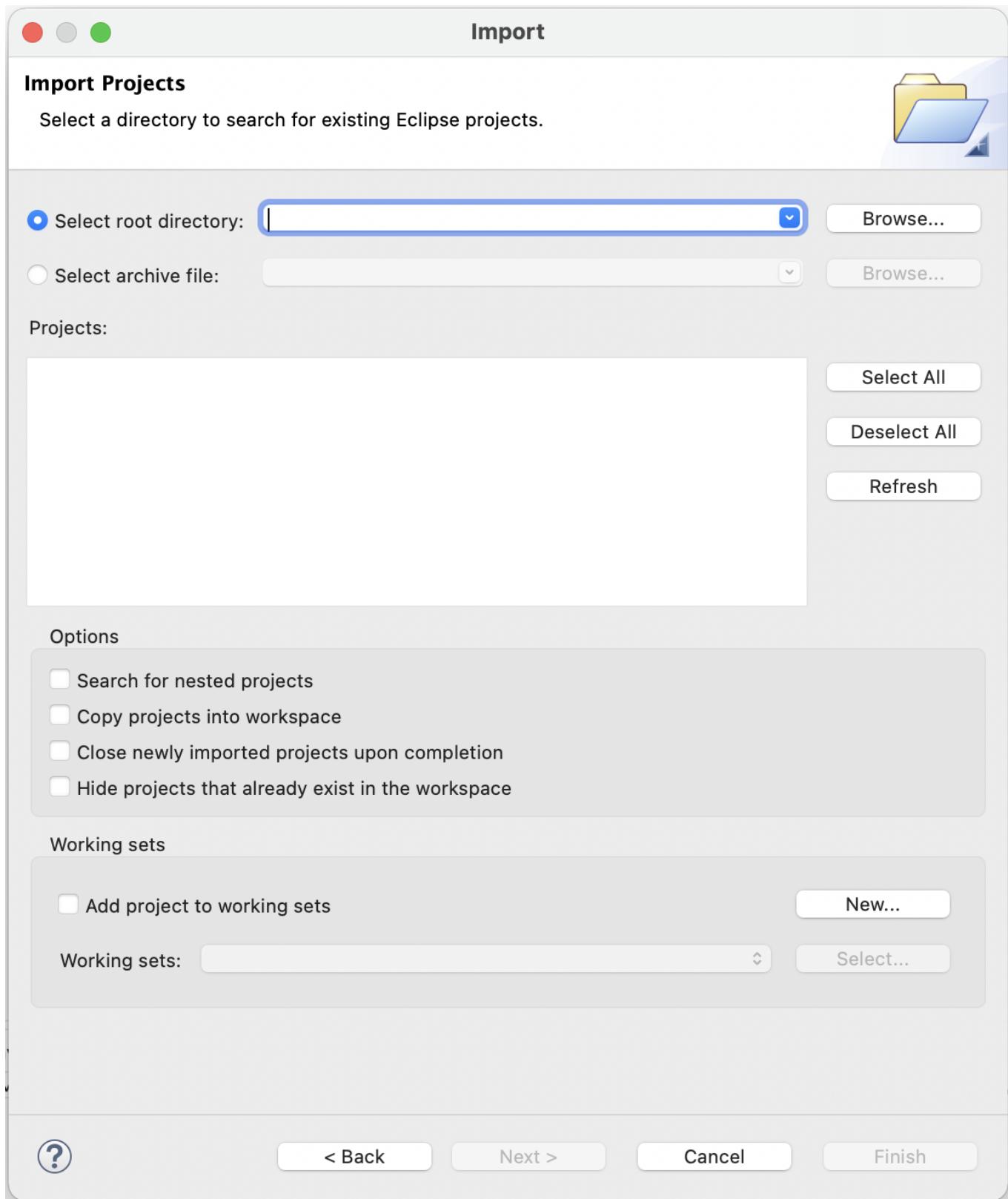
Now, go to the Eclipse menu bar and click "Import" within the "File" dropdown menu.



Once you click import, you should see a popup that prompts you to import a project.

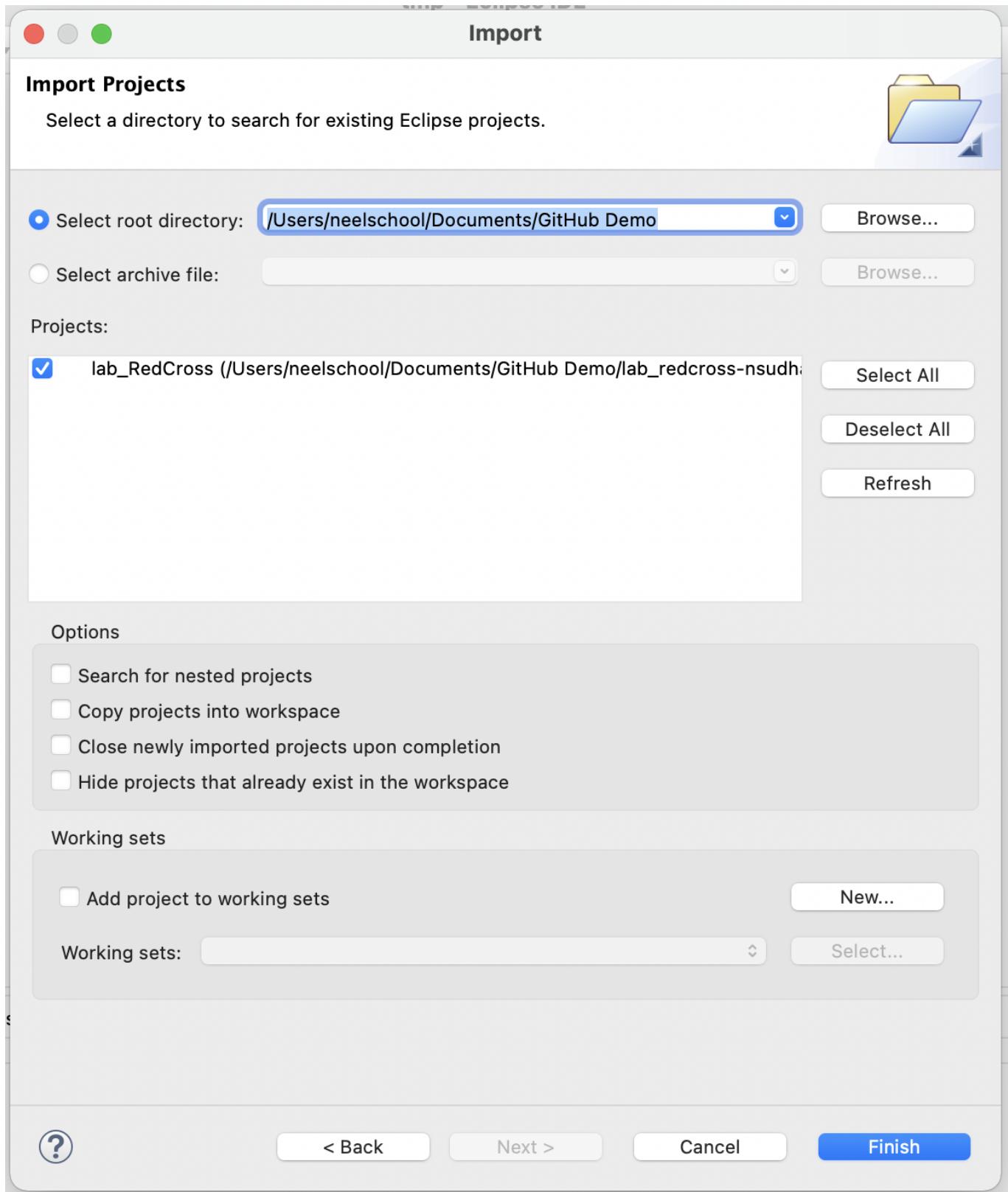


Click "Existing Projects into Workspace" and then hit the blue button that says "Next".

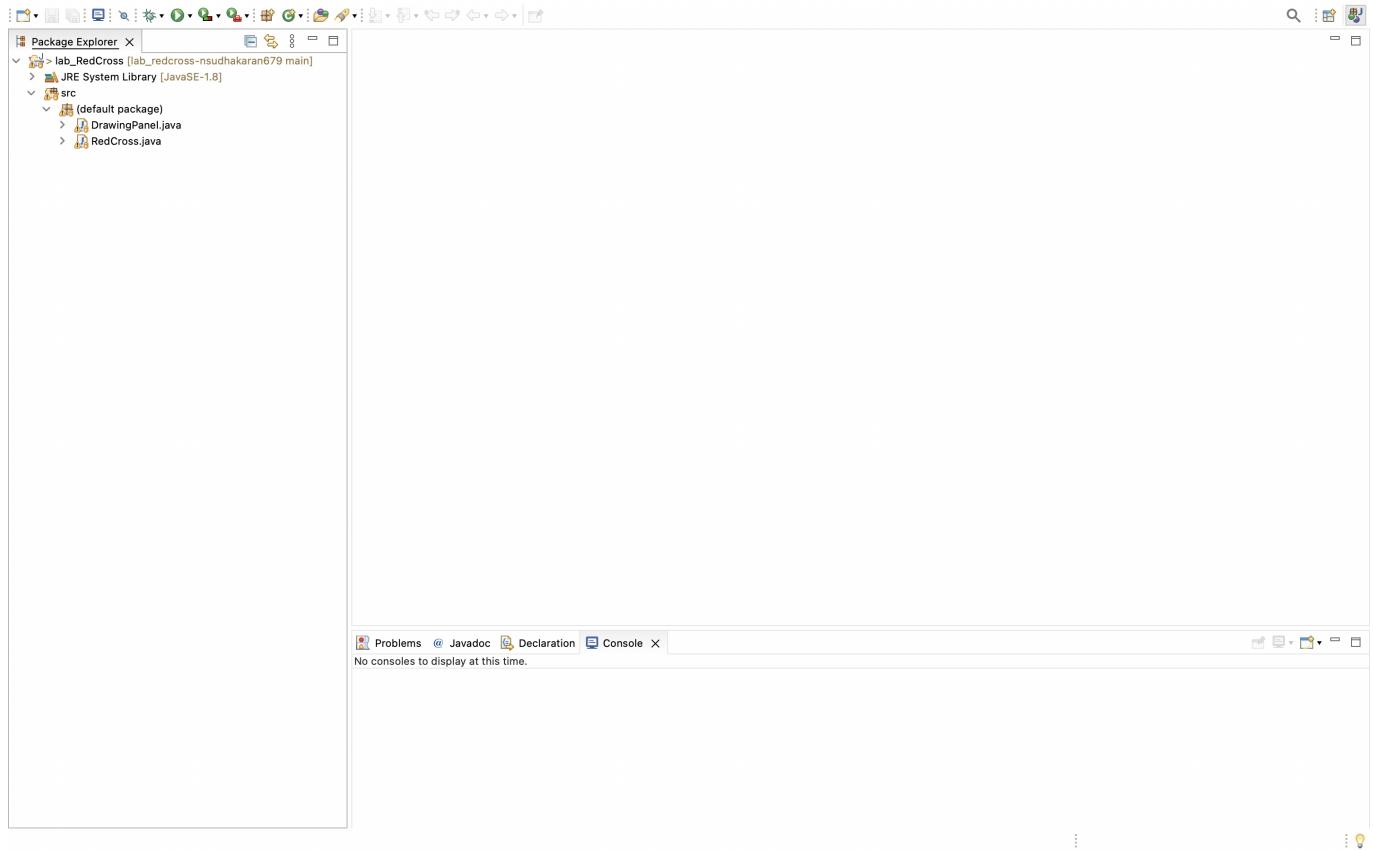


You should be taken to a page that asks you to select the root directory for your projects. By default, GitHub Desktop stores your projects inside a GitHub folder in the Documents directory. Hit browse and locate that folder.

**Note:** If you've modified the default directory for GitHub desktop like we have in the example, navigate to that directory instead.



Once you do that, Eclipse will automatically find and select any projects within your GitHub directory that have not already been imported into Eclipse. To import them, simply click the blue "Finish" button.



Congratulations! If the project you wanted to import into Eclipse is shown inside the Package Explorer, you've successfully completed this step! Continue on to learn [how to make a git commit](#).

## Making a Git Commit

Say you make a bunch of changes to your project and add a whole bunch of new functionality. You can save the individual files within this project, but there is no real way to save a snapshot of the current version of your project and also no way to rollback to a previous version in case you need to. To fix this issue, git has the concept of commits. Each commit is essentially a snapshot of changes that you have made, and you can easily view a history of your commits as well as rollback to a previous commit if necessary.

To make your first commit, make sure you have made changes to your project using Eclipse. Then open up GitHub Desktop to your project repository.

The screenshot shows a GitHub commit interface. At the top, it displays the current repository as "lab\_redcross-nsudhakaran679", the current branch as "main", and the last fetch time as "Last fetched 23 minutes ago". Below this, the "Changes" tab is selected, showing 3 changed files: ".gitignore", "RedCross.class", and "RedCross.java". The "RedCross.java" file is expanded, showing the following code snippet:

```
@@ -32,6 +32,9 @@ public class RedCross
    g.setColor(Color.RED);
    g.fillRect(xCenter, yCenter, 10, 50);
    g.fillRect(xCenter, yCenter, 50, 10);
+
+   System.out.println("Yay, I made it here!");
+
}
```

Below the code, there is a "Summary (required)" field containing "Description" and a "Commit to main" button.

In the bottom right hand corner of the screen, you should see the form inputs asking for a "Summary (required)" and "Description". These are respectively the "Summary" and "Description" values of the commit. Now, enter in your commit summary value into the text input. The commit summary should be an overall title for your commit. If you want, you may enter in a more detailed description into its text input.

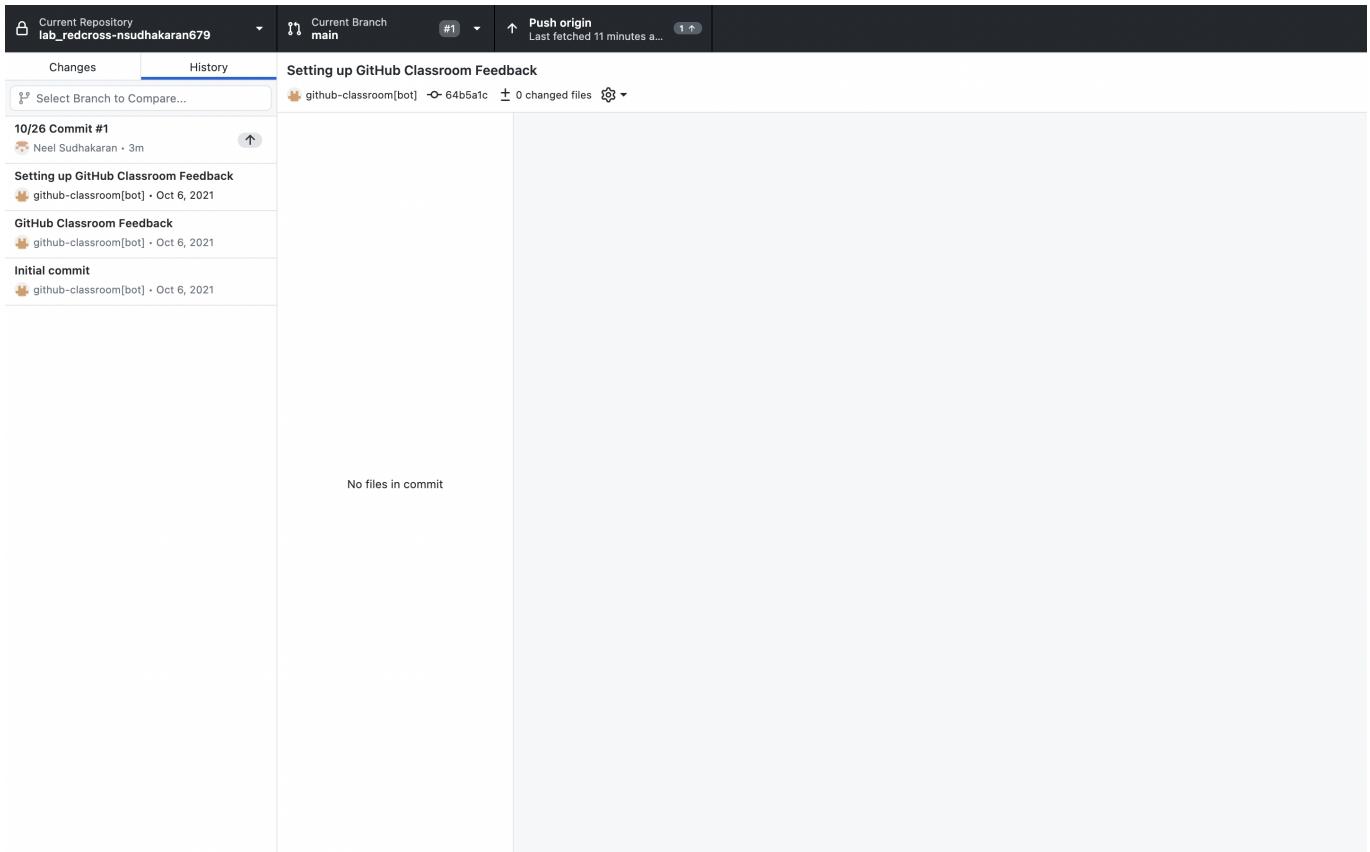
**Note:** A good commit summary format is the "*Date* + Commit + #*Commit Number*". For example, following this format, the first commit I make on October 31 would be called "10/31 Commit #1"

The screenshot shows the GitHub Desktop application interface. In the top left, the repository name is 'lab\_redcross-nsudhakaran679'. The top center shows 'Current Branch main'. On the right, there's a 'Fetch origin' button. The main area displays a diff of a Java file, 'RedCross.java', with three changes highlighted. The commit message in the bottom left says: 'Added a line to print out "Yay, I made it here!"'. A large blue 'Commit to main' button is at the bottom.

Once you've done that, click the big blue "Commit to main" button.

The screenshot shows the GitHub Desktop application interface after committing. The top left shows the repository name 'lab\_redcross-nsudhakaran679'. The top center shows 'Current Branch main'. On the right, there's a 'Push origin' button. The main area displays a message: 'No local changes' with a small icon of a person pushing a cart. Below it, a summary box says 'Summary (required)' and 'Description'. At the bottom, a 'Commit to main' button is visible.

You should see a screen like the one above, where in the "Changes" tab in the sidebar it says "0 changed files". Just to double check though, click on the "History" tab in the sidebar.



If the commit you made is at the top of the commit history in the "History" tab, you've successfully completed this step! Congratulations, you can now learn how to [push your local changes to GitHub](#).

## Pushing Changes to GitHub

Now that you've made a commit to your local repository, we need to sync these changes to GitHub. We call this action "pushing" our code.

To push your changes to GitHub navigate back to the "Changes" tab in GitHub Desktop sidebar.

The screenshot shows the GitHub Desktop application interface. At the top, the repository name is "lab\_redcross-nsudhakaran679" and the branch is "main". The "Changes" tab is selected, showing "0 changed files". In the center, a message says "No local changes" with a small icon of a person pushing a barbell. Below this, there are three suggestions: "Push commits to the origin remote" (with a "Push origin" button), "Open the repository in your external editor" (with a "Open in Visual Studio Code" button), and "View the files of your repository in Finder" (with a "Show in Finder" button). At the bottom left, there's a commit summary window with fields for "Summary (required)", "Description", and "Commit to main". It also shows "Committed just now" and "10/26 Commit #1".

Click the big blue "Push origin" button in the middle of the screen.

The screenshot shows the GitHub Desktop application interface after pushing the commit. The repository name is "lab\_redcross-nsudhakaran679" and the branch is "main". The "Changes" tab is selected, showing "0 changed files". In the center, a message says "No local changes" with a small icon of a person pushing a barbell. Below this, there are three suggestions: "Open the repository in your external editor" (with a "Open in Visual Studio Code" button), "View the files of your repository in Finder" (with a "Show in Finder" button), and "Open the repository page on GitHub in your browser" (with a "View on GitHub" button). At the bottom left, there's a commit summary window with fields for "Summary (required)", "Description", and "Commit to main". It also shows "Committed just now" and "10/26 Commit #1".

If the prompt to push your commits is gone, you've successfully pushed your code to GitHub! As an extra precaution though, take a look at the "Fetch Origin" tab in the header. If it says "Last fetched just now", you've completed everything!

## Conclusion

Congratulations! You've now finished every step in the GitHub Quickstart. However, keep it close by in case you ever need to look back through the guide as a reference. Happy Coding!