Git & GitHub Classroom

# Table of Contents

[Welcome To Your New Workflow!](#_hom1tpuwrum1)

[1. Accept your assignment on GitHub Classroom.](#_op9voobgob2)

[2. Set up your local repository.](#_g50wjn4zlcbu)

[3. Make changes to your code.](#_v5oj3jwtonjj)

[4. Stage your modified files and make a commit.](#_2o8007deh3yh)

[5. Push your changes to GitHub Classroom.](#_9qvpkdbunbus)

[Using Git via The Command Line](#_sjvlugnax65v)

[Setting Up A Local Repo](#_rld1afkm6pom)

[Stage Your Files](#_jr79esn643vq)

[Make A Commit](#_ai5mwt7g5fs)

[Push to GitHub Classroom](#_f1bcjfo8uas4)

[Using Git in Visual Studio Code](#_ayeh2jcvgah3)

[Setting Up A Local Repo](#_1kt2isgnsvm4)

[Stage Your Files, Make a Commit, and Push](#_dcyhoedgrf2b)

[Using GitHub Desktop](#_7ikiz3mq8xcm)

# 

# Welcome To Your New Workflow!

When using GitHub Classroom, your workflow will follow the steps below.

### 1. Accept your assignment on GitHub Classroom.

If you have not already done so, you will be prompted to connect your GitHub account to the class roster.

### 2. Set up your local repository.

You will need to clone the remote repository[[1]](#footnote-0) and set it up locally[[2]](#footnote-1). Once you have it set up, you will be able to work on your project with a text editor as you normally do, such as *Sublime Text* or *Visual Studio Code*.

### 3. Make changes to your code.

Work as you normally do – that is, make changes to your code, validate, save your files, check the output, and debug your code.

### 4. Stage your modified files and make a commit.

When you make changes to the files in your working directory (i.e. your project folder), you have to tell Git that you want to add those files to the ‘staging area’. Once your modified files are in the staging area, Git will know to include them in the next commit.

Next, make your commit. Think of a commit like saving a snapshot of all of your changes, or creating a checkpoint for your progress. When you make a commit, make sure to include a descriptive summary of all of the changes you made since the last commit, written in the present tense.

### 5. Push your changes to GitHub Classroom.

Pushing your local repository (i.e. the project on your computer) to GitHub Classroom will make sure that you will always have a copy of your latest commit available to you online and that your instructor will be able to see your progress.

Make sure that you push your final commit to GitHub Classroom before the deadline.

Now that you’re familiar with the general steps and overall workflow, let’s talk about three different tools and the specific steps you need to take in order to use git below.

# Using Git via The Command Line

The command line is a way of giving instructions or commands to your computer in a slightly different way than you may be used to. Instead of pointing, clicking, and dragging things in a graphic interface, all of the commands that you give your computer are typed.

In Windows, your command line tools include the **Command Prompt** and other applications; in macOS, your command line tool will be the **Terminal**. When using Git, all of the commands that you will use are operating system agnostic, or the same on any operating system.

## Setting Up A Local Repo

After accepting your assignment on GitHub Classroom, open your command line.

When using the command line, you need to navigate to the directory that you want to work in or make changes to. This means that you need to change your directory to the place that you want to put your project folder.

For example, you could type:

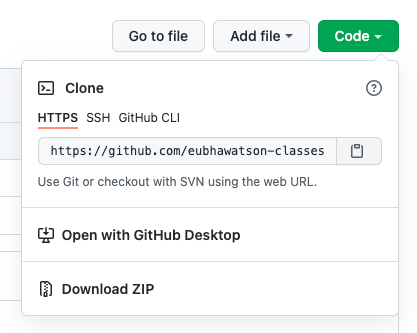
cd Users\watson\Desktop

cd just means ‘change directory’; after a space, you will type the file path to where you want to go. You can also click and drag a folder onto your window in order to copy its file path.

Next, you will need to clone (or make a copy of) your remote repository.

git clone URL

You will need to replace ‘URL’ with the actual address to your repository on GitHub Classroom. You can copy this value by clicking on the green ‘Clone’ button in the upper right-hand corner.



Now that you have your local repository all set up, you can begin working as you normally do.

## Stage Your Files

As you make changes to your files, you need to add them to the staging area. You can check which files have been modified with:

git status

This command will tell you which files have been modified and whether or not they have been added. You can then stage individual files with:

git add filename

Or, you can stage all modified files with:

git add -A

## Make A Commit

Finally, make a commit with the following command:

git commit -m "Message about this version."

Remember that your message should be written in the present tense and summarise what changes have been made since your last commit.

## Push to GitHub Classroom

Lastly, you must make sure that your code, including your latest commit, is uploaded to GitHub Classroom. You can do this with the push command:

git push -u origin main

Remember that if your current working branch is called something other than ‘main’, you will need to use your own branch name in place of ‘main’.

origin is the place where the code originally came from; if this isn’t set, you can always grab the URL from your remote repository and set it with the following command:

git remote add origin URL

If you have not already done so, you may have to enter your username and password. When you type your password, it may look as if nothing is being entered (i.e. there are no bullet operators or asterisks); this is normal.

You may also need to set up a personal access token and use this in place of your password. To create a personal access token, follow the steps outlined here:

<https://docs.github.com/en/github/authenticating-to-github/keeping-your-account-and-data-secure/creating-a-personal-access-token>

Make sure that you save your token, as you will never be able to see it again after you generate it.

Finally, refresh your repository on GitHub Classroom and make sure that all of your changes are available there.

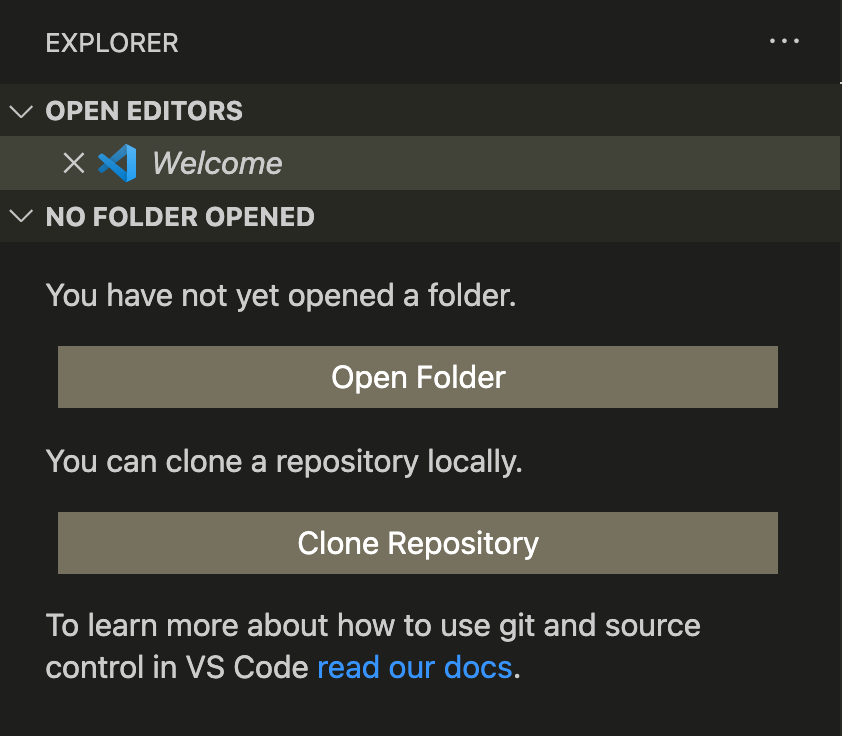
# 

# Using Git in Visual Studio Code

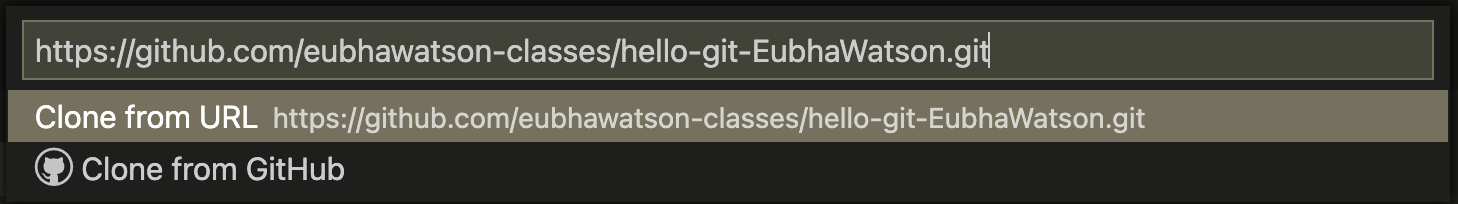
In Visual Studio Code, source control is built right into the editor. This means that you do not need to install any special extensions or plugins in order to use Git and GitHub from the text editor itself.

## Setting Up A Local Repo

From the Welcome Screen, you can clone a remote repository with the press of a button. To get started, click on ‘Clone Repository’.



You will then be prompted with the repository’s URL; you can get this from the green ‘Code’ button when visiting your repository in your browser. This is the same URL that you would use if you were using the command line.



VS Code will then ask you where you’d like to put your local repository on your computer; after, it will prompt you to open the repository in the editor.

## Stage Your Files, Make a Commit, and Push

As you make changes and save your files, you will notice an ‘M’ in your explorer panel. This means that the file has been modified.

You do not strictly need to stage your files in VS Code. This is because if you make a commit and you haven’t staged all of your modified files, the editor will prompt you to do so with a pop-up dialogue window.

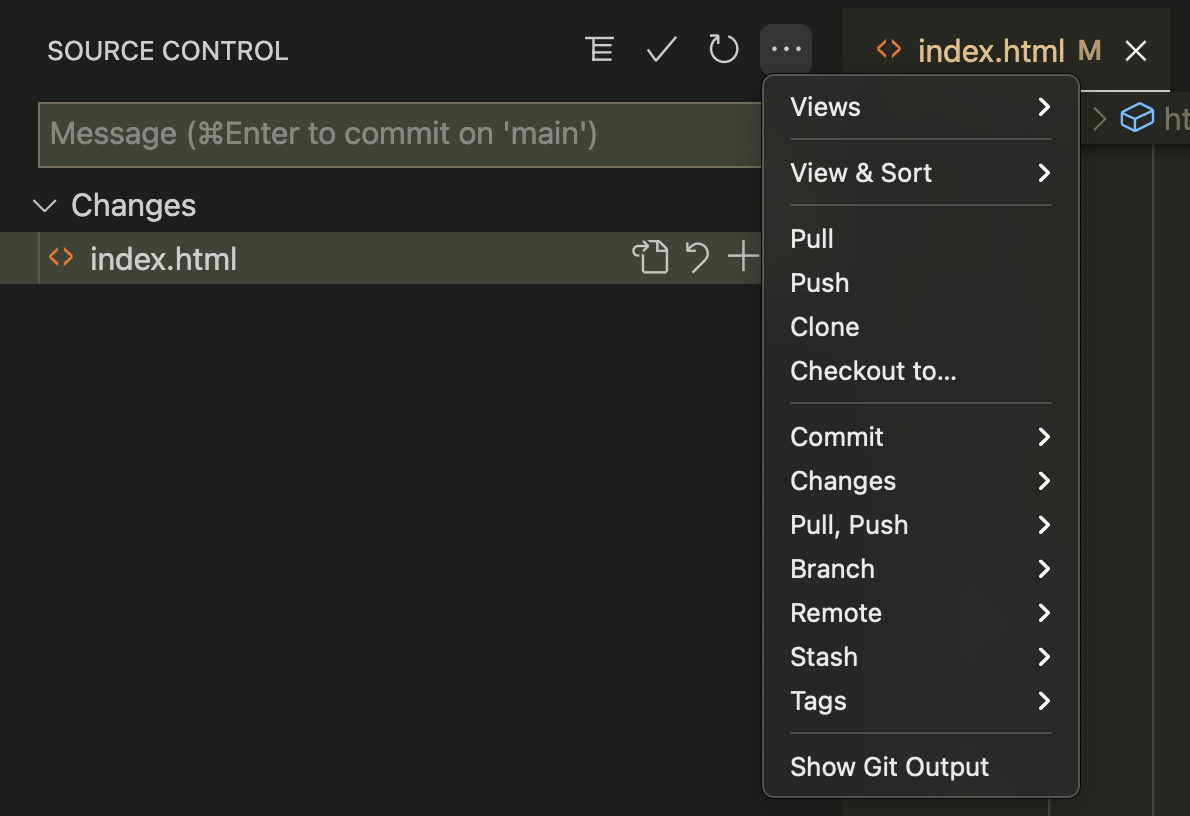
However, if you would like to stage them manually, click on the **Source Control** icon on the left-hand side of your editor window.

# 

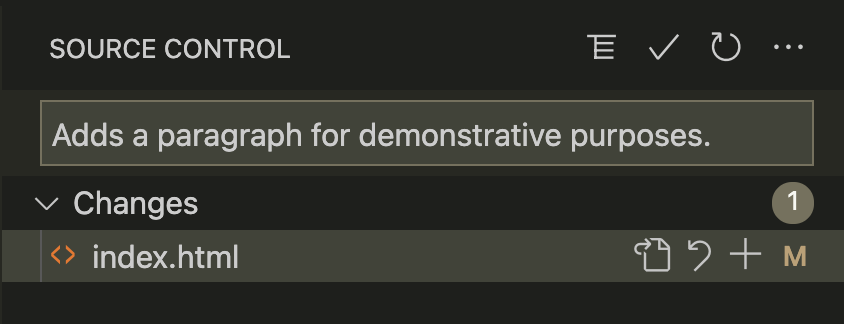
*In this screenshot, the Source Control icon currently has a ‘1’ badge*

*because there is one modified file in the current working directory.*

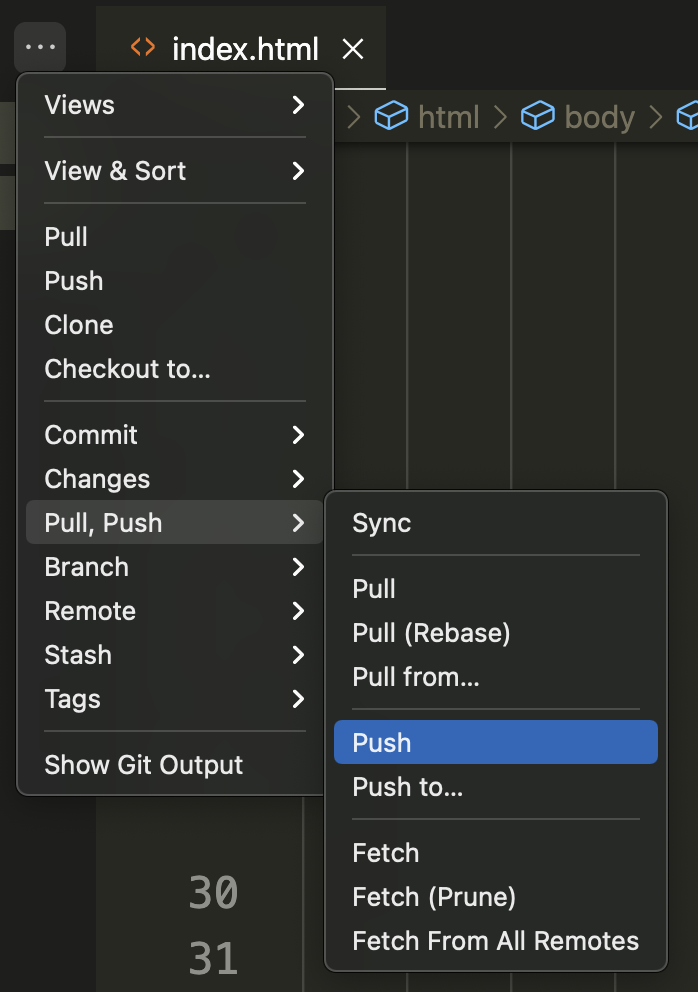
From Source Control, you can use Git through a menu interface.



You can also make a quick commit at any time by typing a message that summarises all of your changes in the ‘Message’ text field and hitting ⌘ + Enter when you are finished.



Do not forget to push your latest commit to GitHub Classroom and check the remote repository in your browser to make sure that the push was successful.

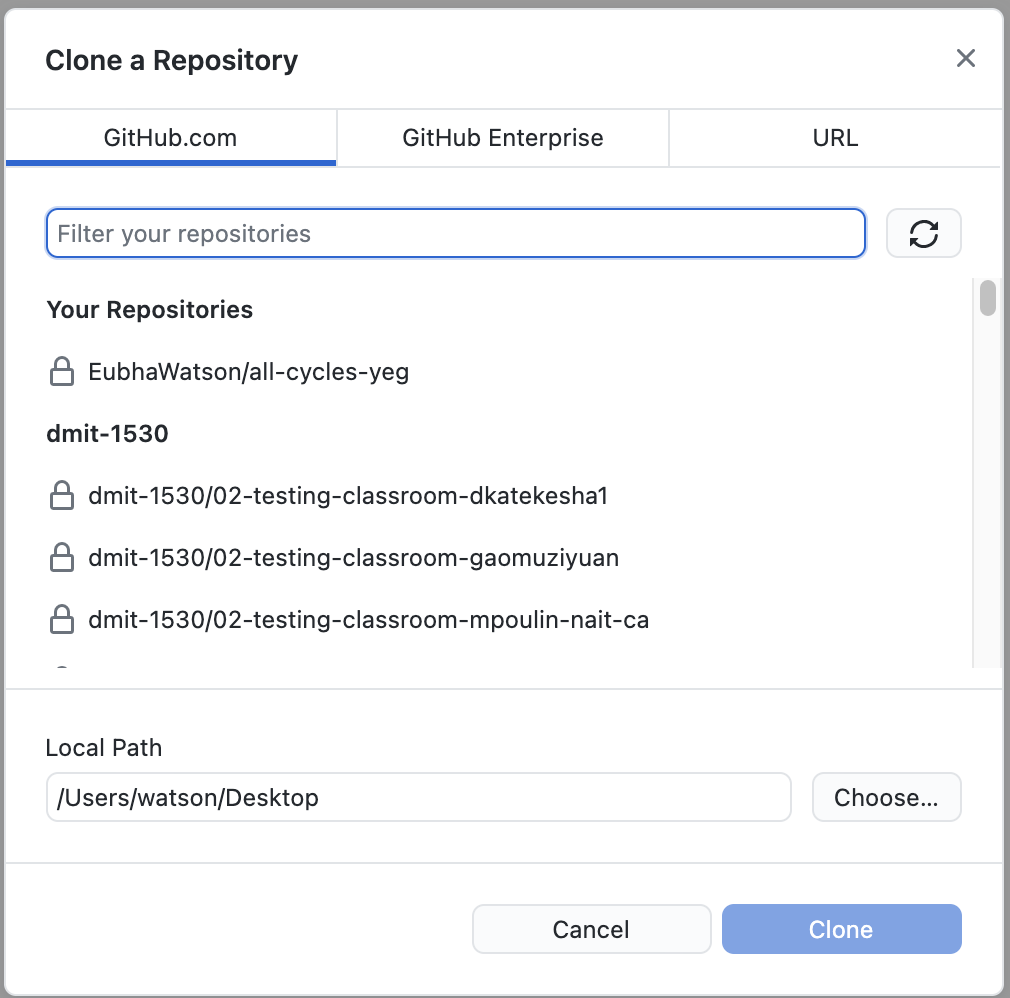


# Using GitHub Desktop

LIke Visual Studio Code, GitHub Desktop offers an alternative to using the command line. To get started, go to:

File > Clone Repository …

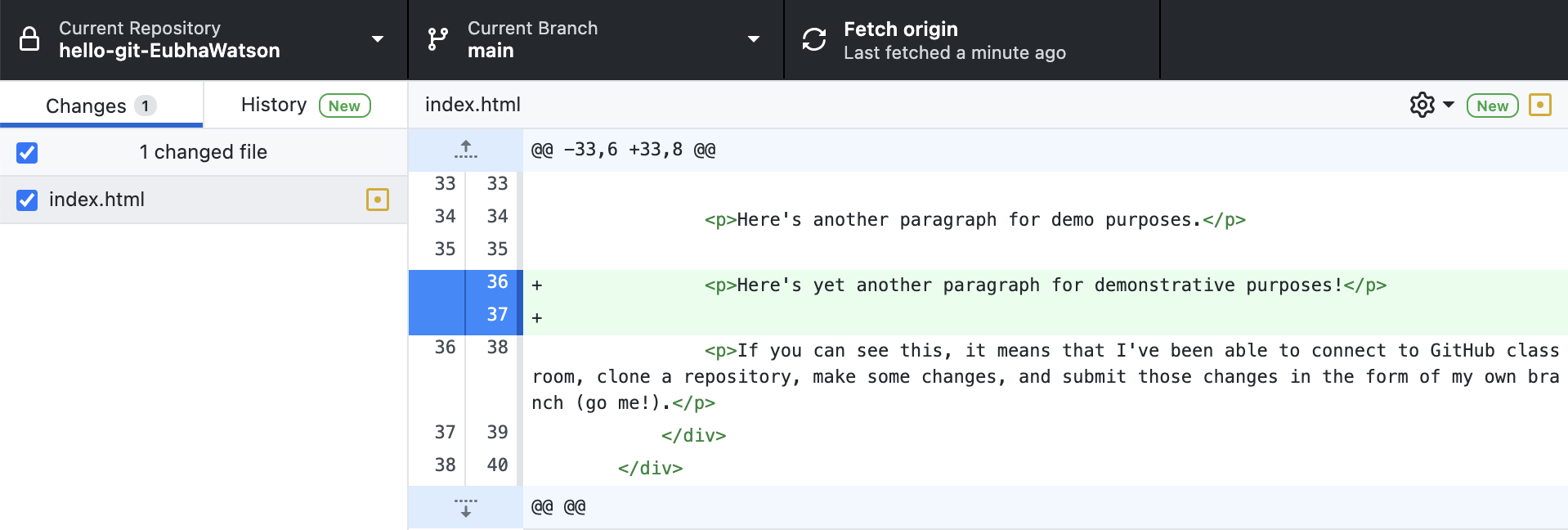
This will bring up a dialogue window with a number of options.



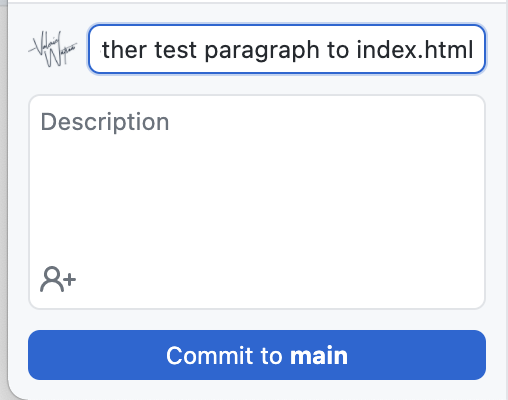
You can either search for your repository under the first tab, or simply paste the URL of the remote repository under the ‘URL’ tab.

After you’ve told GitHub Desktop where you’d like to clone your repository from, you will also need to tell it where you’d like to put it on your computer.

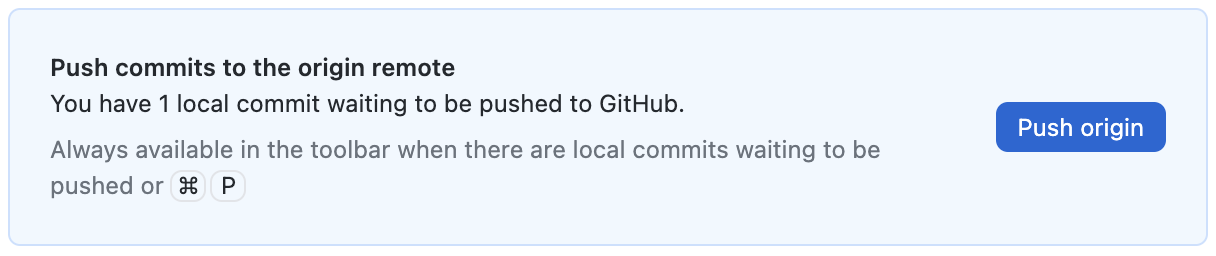
As you work, GitHub Desktop will keep track of all the changes you’ve made to your project files, including which lines of code you’ve added or deleted since your last commit.



When you are ready to make a commit, you can do so by typing your message in the lower left-hand corner and clicking ‘commit to main’.



After you’ve made your commit, GitHub Desktop will prompt you to push your repository to the origin (that is, where you originally cloned your repository from). Click ‘Push origin’ to execute this command.



Finally, check the remote repository in your browser window to make sure that your latest changes were uploaded properly.

1. A remote repository is a project, tracked by Git, hosted somewhere online; in this case, the remote repository will be on GitHub Classroom. [↑](#footnote-ref-0)
2. This means you need to set it up on your computer. [↑](#footnote-ref-1)