

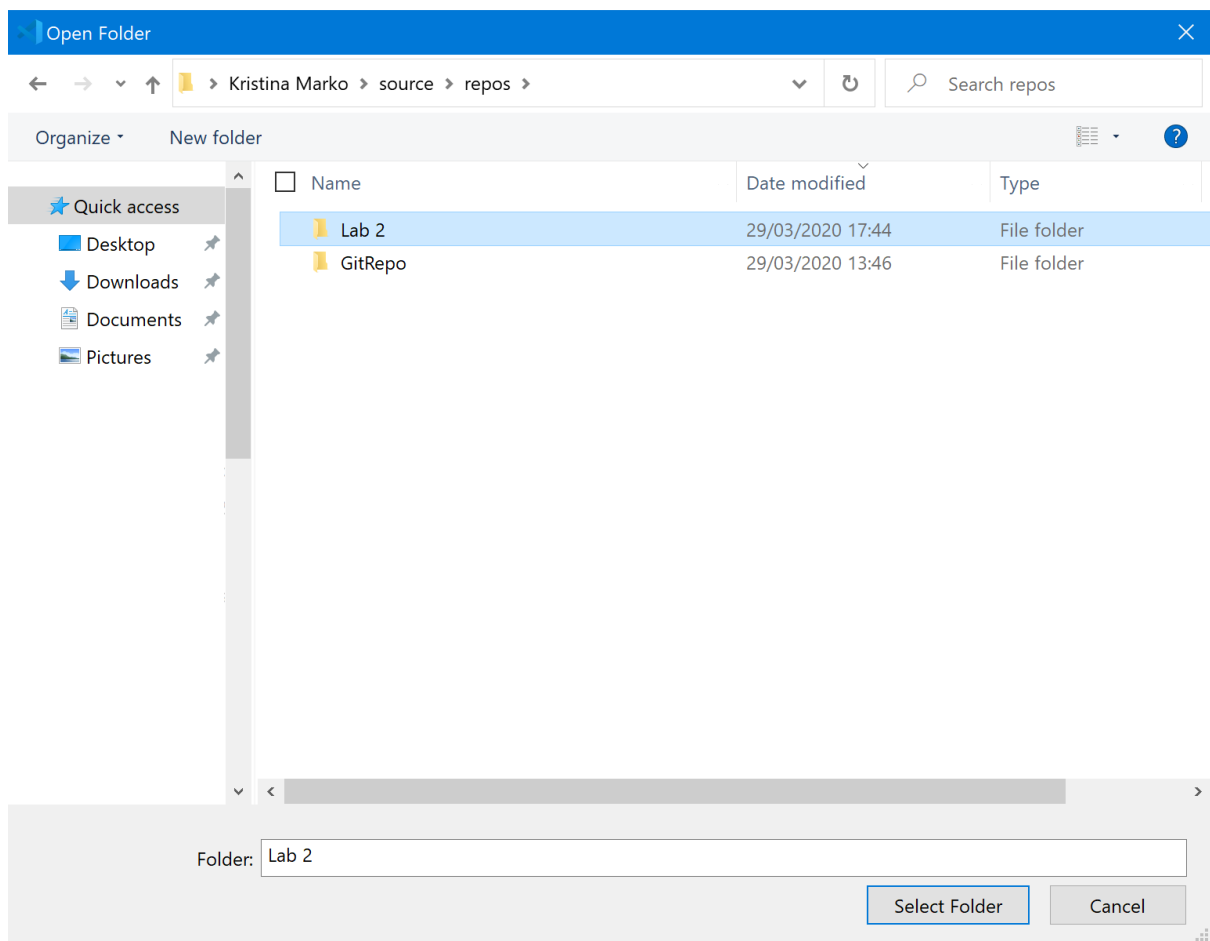
## Lab 2 – Local and Remote Git Repository using Visual Studio Code


### Objectives

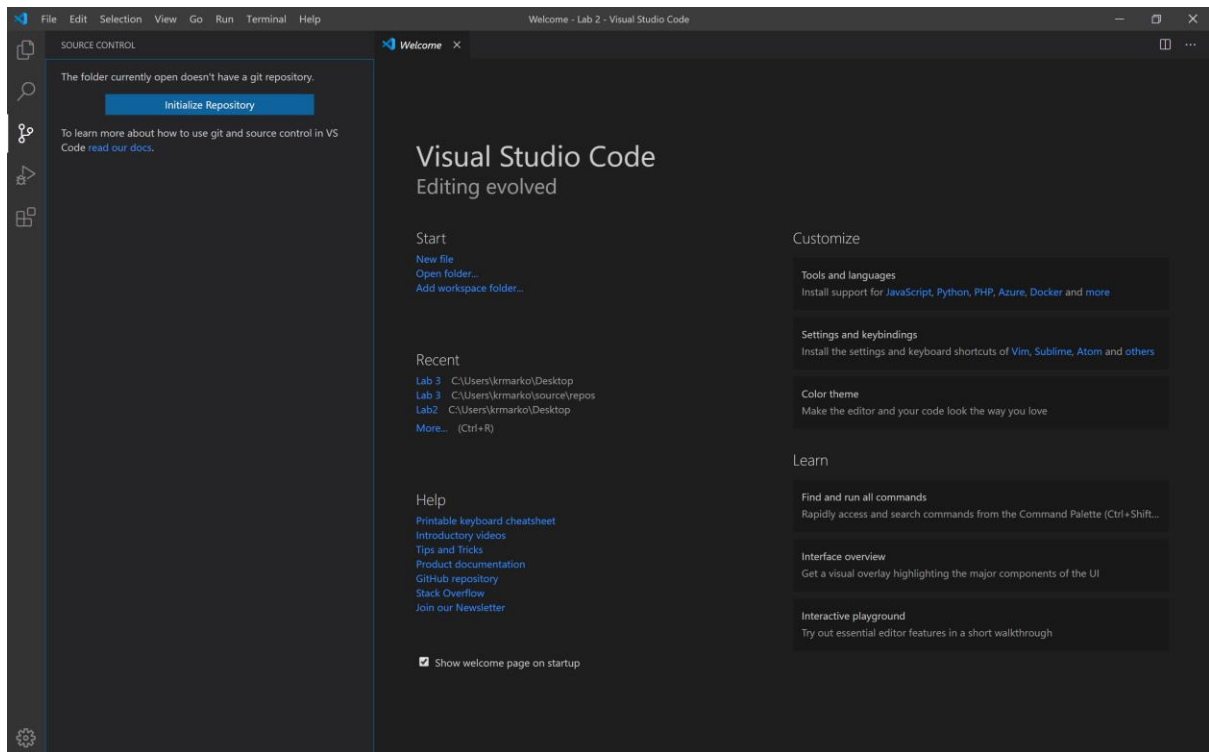
- Create a repository via VSCode
- Using VSCode to stage & commit
- Connect github to VSCode

### Initialise Local Git Repository

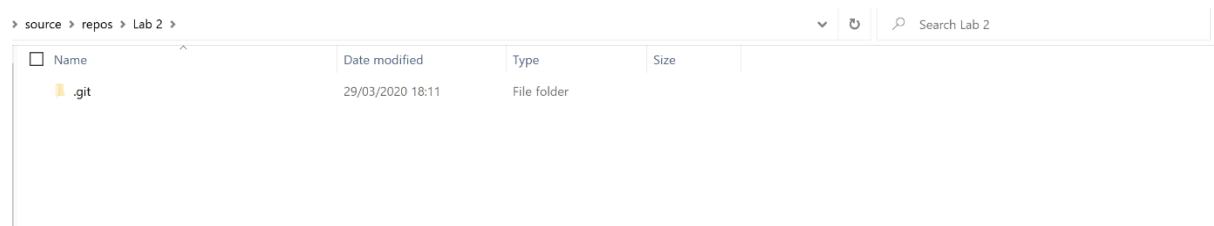
Open Visual Studio Code. Under File select “Open Folder”. Navigate to the same root where you created a repository in Lab 1. Right click and create a new Folder named Lab 2. Select that folder and hit enter.



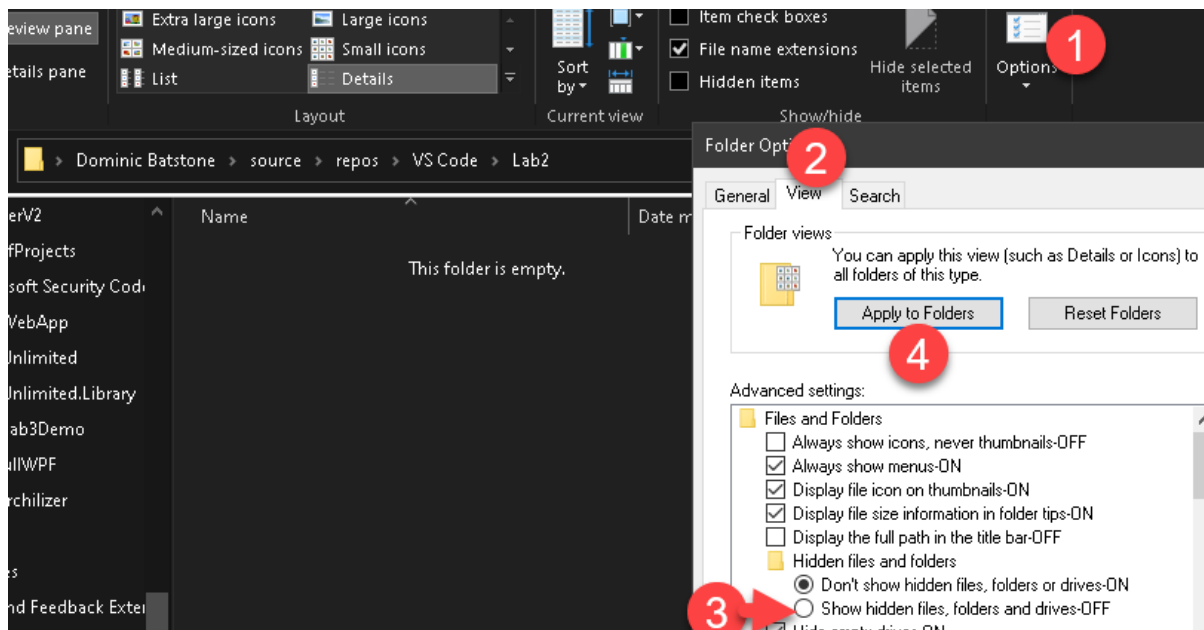
On the left pane, if you select the Source Control button  then initialize a repository. This is equivalent to the git init command.



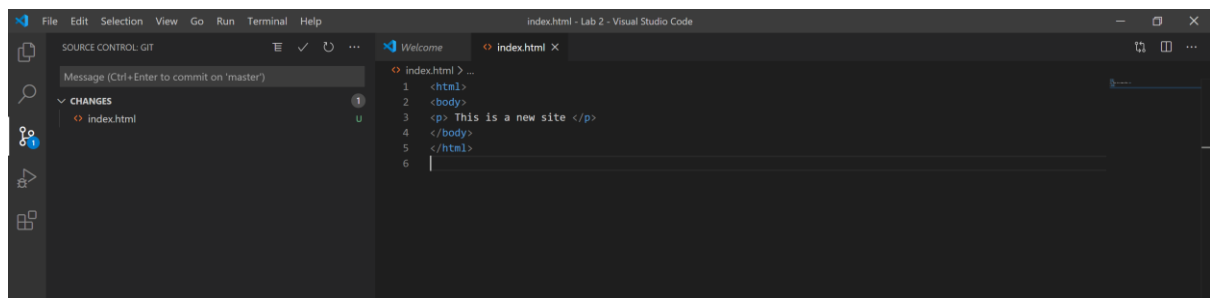
**Note:** Once you have initialized the repository, if you navigate to that folder in your file explorer and view the hidden files, you will be able to see a `.git` folder. This is what is used in order to track all the changes you make to your files.



[Optional – if you cant see the hidden files and want to try it, follow these steps (note: it’s possible that your company may not allow these changes)]



As we did in Lab 1, you will now create another index.html file. Click on “New File”, populate your html and then save it as an index.html file.

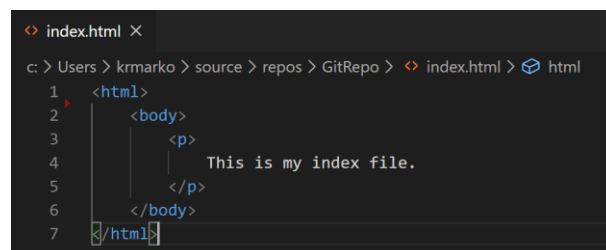


In the left pane you can now see that the index.html file is under changes. There is also a green U next to it meaning that these changes are untracked.

## Update your file in VSCode

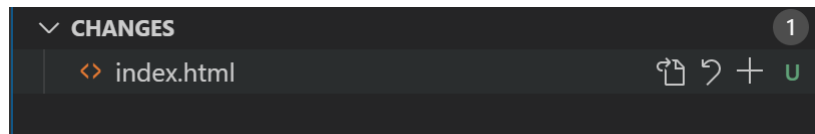
Navigate to your index.html file in Visual Studio Code and open it. Your file should be empty.

Populate your html file to contain some code (see below for sample) and save your changes.

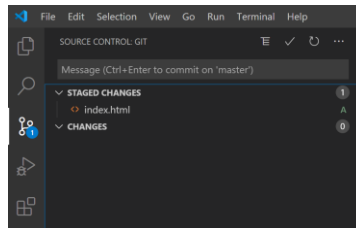


## Staging and Committing Changes

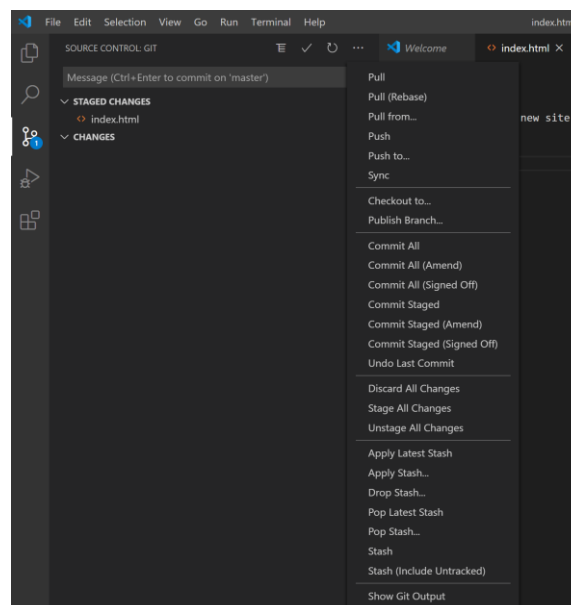
As mentioned earlier, the green U means that the changes to this file are untracked. You are able to open the file, discard the changes or stage them.



Click on the plus in order to stage your changes. You should now see this:

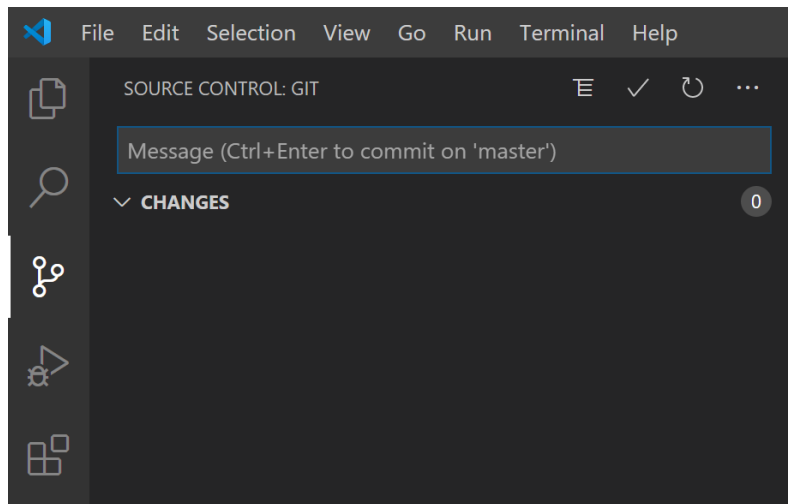


We can commit this change. On the left pane, click on the 3 dots. They will give you a list of various actions that you can perform. Select **Commit Staged**.



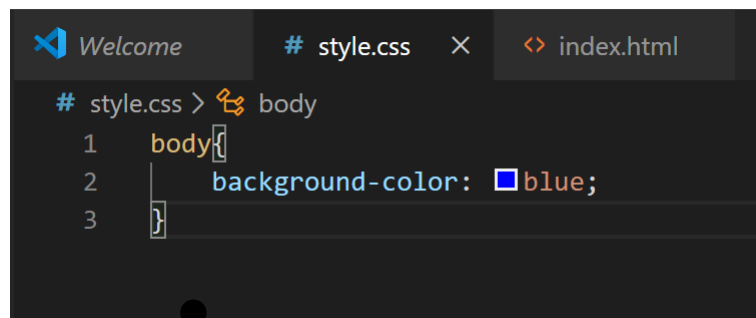
You will be prompted to enter a commit message. Your message must mention that you added an index.html file for this example.

Once you have done that your changes will be committed, and you should now see that there are no more changes for your repo.

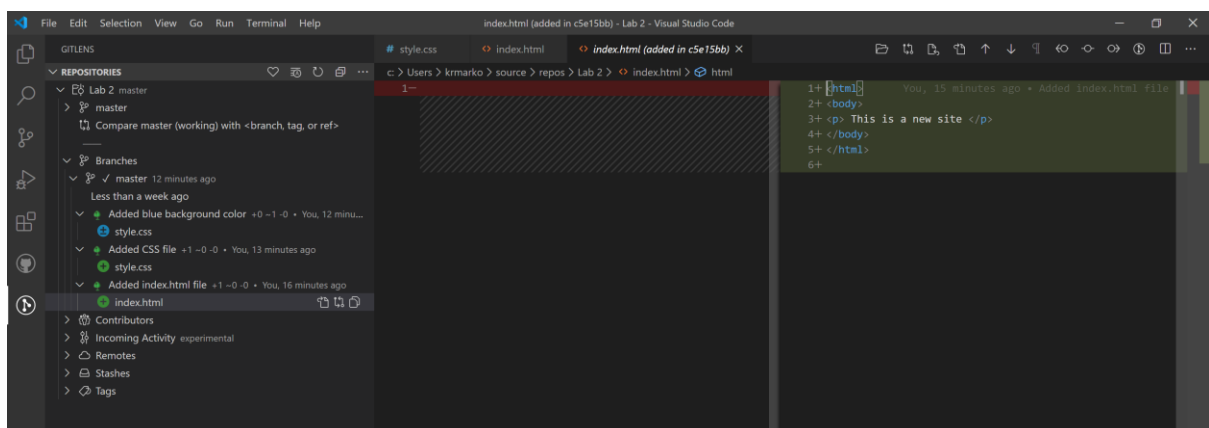


Now create a style.css file, stage and commit that change as well.

Once that has been committed, add code to your css to make the body of your index.html document blue. It should look something like this:



Now we have made 3 commits in total. If we wanted to see all commits made we can download an extension like GitLens. In order to do that go to Settings – Extensions – GitLens – Install. Once installed you'll see a GitLens image on the left sidebar and when you click on that you'll have more information about your source control. You'll be able to see all of the files that you made changes to and can see the specific changes made for each commit as shown below:



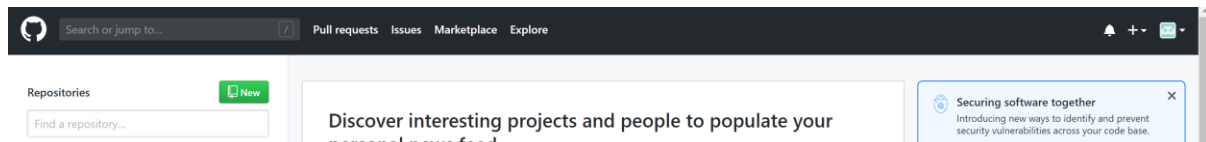
If you try to push your changes, you will be told that “Your repository has no remotes configured to push to”, so we must now create a remote repository and then configure it with our local one.

## Creating a Remote Repository

Up to now you have created a local repository and committed files to it. The next step is creating a remote repository that you can push your files to.

In order to do that we must first create a remote repository. Navigate to <https://www.github.com>

Create an account if you don't have one already. Navigate to the home page which should look something like the image below. On the left hand side you should see your "repositories". Click on "New".



This will enable you to create a new repository.

As we will be importing an existing repository, we want this one to be empty so provide a name for your repo along with a description. You don't need to tick the checkbox. Once completed, hit the "create repository" button.

Create a new repository

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

---

Owner: kristim96 / Repository name: GitLabApp2 ✓

Great repository names are short and memorable. Need inspiration? How about fluffy-system?

Description (optional): Application 2 for Git Demos

☒ Public  
Anyone can see this repository. You choose who can commit.

☐ Private  
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ Initialize this repository with a README  
This will let you immediately clone the repository to your computer.

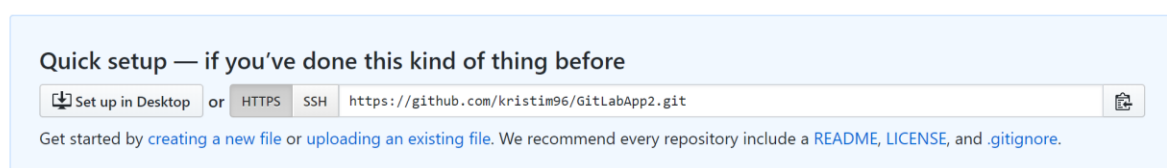
Add .gitignore: None Add a license: None ⓘ

[Create repository](#)

You now have created an empty remote repository. Copy the URL of your repo

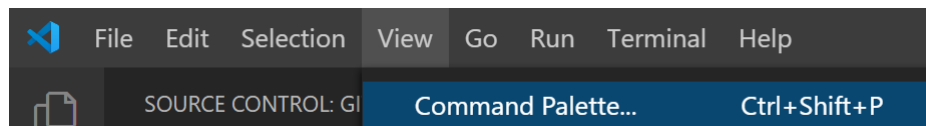
You now have 2 repos: One local repository where you have committed your changes and one remote repository that is empty.

You now need to push your local repository to the remote repo. In order to do that we will need the URL of your remote repo.

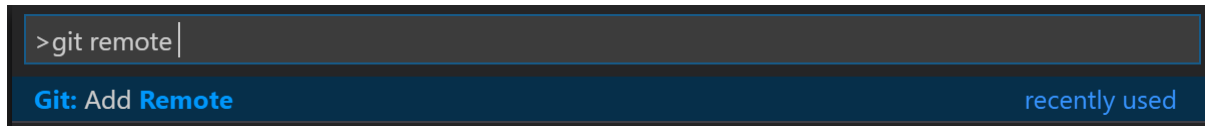


## Configuring your Repositories

Go back to VSCode and under the View bar select Command Palette.



Type in “git add remote” and select “Git: Add Remote”



Then enter a name and then the HTTPS URL of your remote repository.

Once completed, select “Push To” from your list of actions. You should be able to see your recently added remote repository.

If you check on GitHub now, you should see be able to see your local repository’s changes/files.

