

Essential Skills – Lesson 27 and 28

Version Control

It is a tool for tracking changes in your files and coordinating the work among all the programmers.


Today we will continue our discussion with Git – even though you will use it many times throughout the remainder of the term and into future terms – and look at the process of setting up a repository for a group project and how team members can use Pull and Push processes to keep the project code up to date within Visual Studio Code (VSC). Look at the following videos.

<https://www.youtube.com/watch?v=fJtyf62yAb8> – Defines Git (~ 9 minutes)

A short video on how to stage and commit files plus create branches.

https://youtu.be/i_23KUAEtUM?si=OYI9sq1sKjHfFk1L

As you start a new project you have a few options:

- Create a remote repository in GitHub
- Get the URL for the Project that has been set up for the team. – I would suggest making it Public with a ReadMe file and if you want to include a licence, I suggest the MIT License.
- This option is if you have not opted for 1 of the first 2 paths, select the source control icon  on the left-hand side of VSC. There will be 2 options, Initialize Repository or Publish to GitHub both will allow you to access source control features but only Publish to GitHub will take a new project locally and publish it to your GitHub account

Get familiar with using all 3 options.

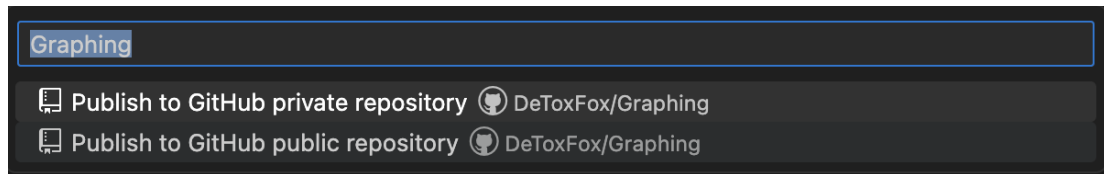
Previous lessons have shown how to create a repository in GitHub and how to clone a repository from GitHub. This will walk you through initializing and eventually pushing code to GitHub.

As you create a new project you will initialize this directory as a local repository by using selecting the source control icon and pressing the “Initialize Repository” button. To prove that it is a local repository open terminal in VSC and type in `ls -a` and notice that the file `.git` is there, also in the origin folder on Mac you can use Command + Shift + . and yes that is a period there, this will show all hidden file in any directory. The following will be creating time sensitive backups of your project that will be used for version control.

The folder currently open doesn't have a git repository. You can initialize a repository which will enable source control features powered by git.

Initialize Repository

You will now use the options in the source control menu to a series of Git commands including **stage changes** which is a **plus sign next to the file name** - to place files in the staging area, and **git commit plus adding a message** to commit the changes once there is enough content. Once you have added a commit comment you can hit the blue commit button. Use the ellipsis just above the Commit Message box, go down to branch and select Publish Branch. A drop down in to top center of VSC will pop up here you can Publish to GitHub Public Repository.



Git Practice Exercise

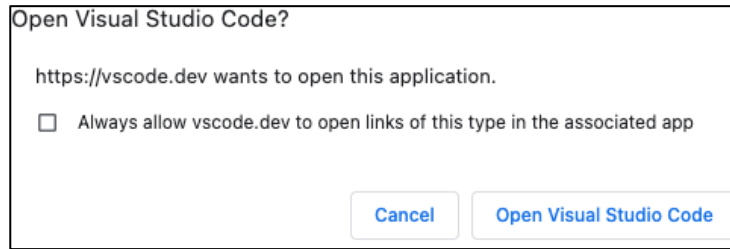
To practice this process, complete the following exercise.

1. In VSC, under the file menu open a New File with the name `main.py`, then selecting create. From there a window will open, in a location of your choice create a New Folder and ProjectName of choice – if you all want to be consistent for the first one use **SampleRepoProject** making sure it is selected before Create File button is pressed, from there a blue open folder should appear, select that to make sure you are in the right directory.

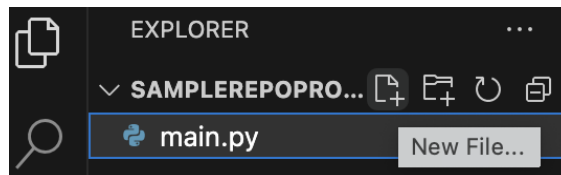
2. In VSC sign into GitHub though this icon.



3. You might have to go to your browser to allow permissions for use of the application.



4. To make this a local repository (for version control), use the Initialize Repository button in the source control menu. As mentioned above, you can prove that it is a local repository with **ls -a** and notice that the file **.git** is there. Also note that VSC gives a letter “U” for untracked and not in the staging area.
5. Add some code to the **main.py** file OR copy an existing program into the folder. Use the touch command in the terminal to create a second file called **Stuff.py** or use the New File button and add code to it. Create a third file called **MoreStuff.txt** – add some text to the file explaining the project.



6. From the Source Control menu you will notice that the files have been added or modified but are not in the staging area with the letter U.
7. In the Source Control menu add the two python files to the staging area using the plus symbol that is next to the file name. You can see now those files are under “Staged Changes” with the letter “A” for added.
8. Now you are ready to commit the files and adding a commit message using the commit button, make sure to add a message first before committing. You can see now that there are no longer any files left under “Staged Changes”.
9. Add a couple of new files to the project and repeat using steps 5, 6, 7 and 8. Notice this time that there are two commits that have been performed.
10. If you make a terrible error and screw everything up, you can revert to any previous commit, use the ellipsis button and hover over the commit menu and select “Undo Last Commit” and you will see the files committed come back as “U” for untracked in this instance.

Now use git to link to your remote repository.

If you did not create the GitHub repository the next steps will do this for you.

- To send the files to your repository or GitHub account make sure that are staged with a commit message and hit the commit button.
- Use the ellipsis button and hover over branch selecting Publish Branch. Once complete, go to the GitHub repository and refresh – you will see the files are now there.
- As members of the team add items to the repository, you can update your local repository using the pull which again is in the menu when you select the ellipsis button. This will add any new files to your local project and be up to date.

What is the whole idea of version control so important?

Git allows us to keep track of the different versions of our program as a backup of sorts and allows us to revert to any previous version if there is a catastrophic error.

Git with GitHub allows us to use a local repository on our own computer to keep up to date with the entire project stored in the remote GitHub repository using push and pull operations. The entire team can be active, and the project made available to everyone.