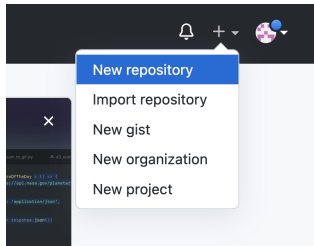


Setup GitHub:

1. Create GitHub repository.

Login to GitHub - <https://github.com/>

And click on the plus icon and select “New Repository”.





2. Create new repository.

Fill the details as circled in the screenshot below.

Create a new repository


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


Owner * **Repository name ***

 leocyriac / LearningPython 

Great repository names are short and memorable. Need inspiration? How about [congenial-octo-winner](#)?

Description (optional)

☒  **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:
Skip [this step](#) if you're importing an existing repository.


☒ **Add a README file**
This is where you can write a long description for your project. [Learn more](#).

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

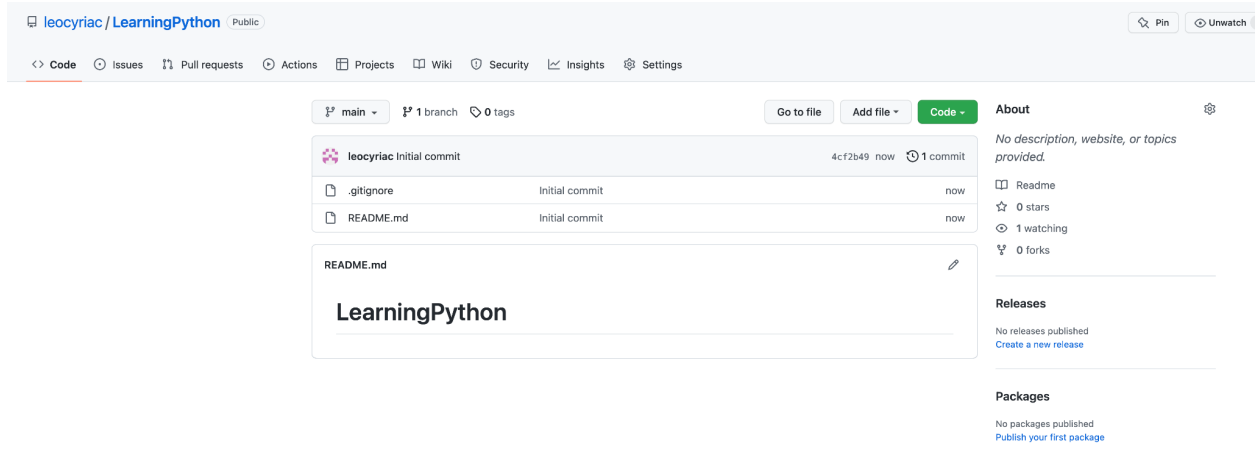
Choose a license
A license tells others what they can and can't do with your code. [Learn more](#).

License:

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.

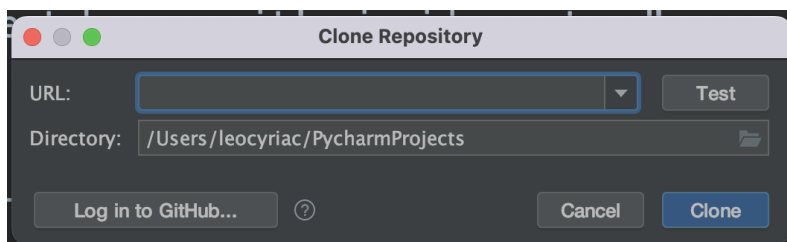
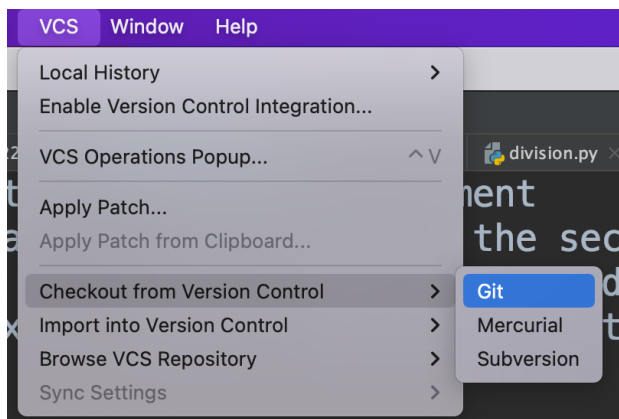
- Once you click “Create Repository”, you should be redirected to a page similar to below.



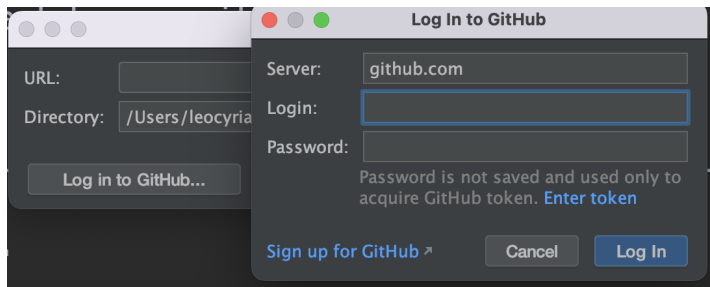
You are now ready to use this code repository.

Setup Git in PyCharm

- Open PyCharm and Select “VCS” from the menu list at the top, and click “Git”.

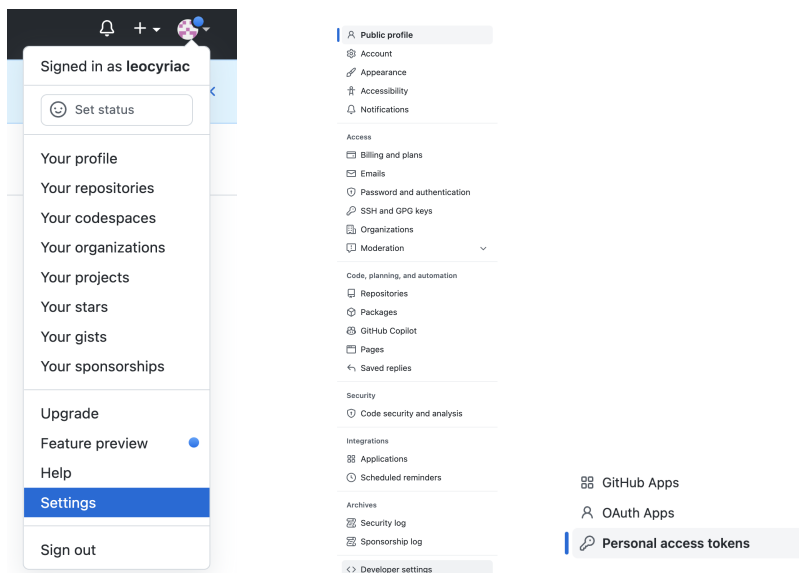


Click “Log in to GitHub” to login to GitHub from Pycharm. You can either enter your github login and password here to login,

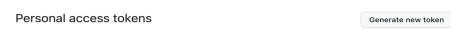


OR using “Enter Token”.

A token can be created by login to github web and follow below,



Click generate new token



Pycharm

What's this token for?

Expiration

30 days

The token will expire on Sat, Sep 10 2022

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo:deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input type="checkbox"/> workflow	Update GitHub Action workflows
<input type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input type="checkbox"/> read:packages	Download packages from GitHub Package Registry
<input type="checkbox"/> delete:packages	Delete packages from GitHub Package Registry
<input type="checkbox"/> admin:org	Full control of orgs and teams, read and write org projects
<input type="checkbox"/> write:org	Read and write org and team membership, read and write org projects
<input type="checkbox"/> read:org	Read org and team membership, read org projects
<input type="checkbox"/> admin:public_key	Full control of user public keys
<input type="checkbox"/> write:public_key	Write user public keys
<input type="checkbox"/> read:public_key	Read user public keys
<input checked="" type="checkbox"/> admin:repo_hook	Full control of repository hooks
<input checked="" type="checkbox"/> write:repo_hook	Write repository hooks
<input checked="" type="checkbox"/> read:repo_hook	Read repository hooks
<input type="checkbox"/> admin:org_hook	Full control of organization hooks
<input checked="" type="checkbox"/> gist	Create gists
<input type="checkbox"/> notifications	Access notifications
<input type="checkbox"/> user	Update ALL user data
<input type="checkbox"/> read:user	Read ALL user profile data
<input type="checkbox"/> user:email	Access user email addresses (read-only)
<input type="checkbox"/> user:follow	Follow and unfollow users
<input checked="" type="checkbox"/> delete_repo	Delete repositories
<input type="checkbox"/> write:discussion	Read and write team discussions
<input type="checkbox"/> read:discussion	Read team discussions
<input type="checkbox"/> admin:enterprise	Full control of enterprises
<input type="checkbox"/> manage_runners:enterprise	Manage enterprise runners and runner-groups
<input type="checkbox"/> manage_billing:enterprise	Read and write enterprise billing data
<input type="checkbox"/> read:enterprise	Read enterprise profile data
<input type="checkbox"/> project	Full control of projects
<input type="checkbox"/> read:project	Read access of projects
<input type="checkbox"/> admin:pgp_key	Full control of public user GPG keys
<input type="checkbox"/> write:pgp_key	Write public user GPG keys
<input type="checkbox"/> read:pgp_key	Read public user GPG keys

Generate token

Cancel





Copy the token created and paste in pycharm at the section where the token is asked.

Personal access tokens

[Generate new token](#)[Revoke all](#)


Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

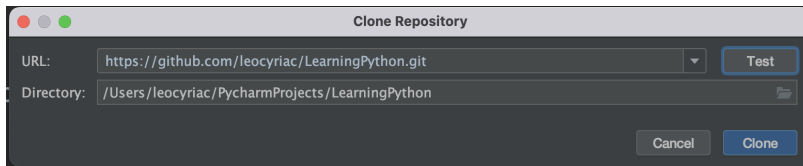
[Delete](#)

Pycharm1 — *admin:repo_hook, delete_repo, gist, repo* Last used within the last week [Delete](#)

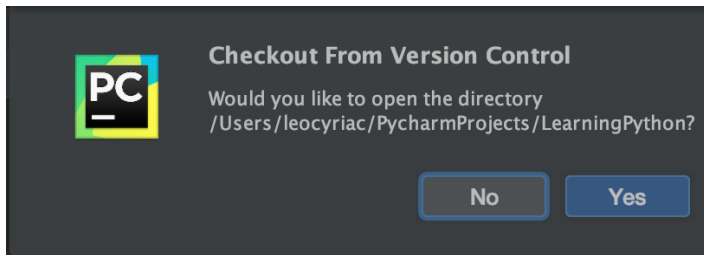
 This token has no expiration date.

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

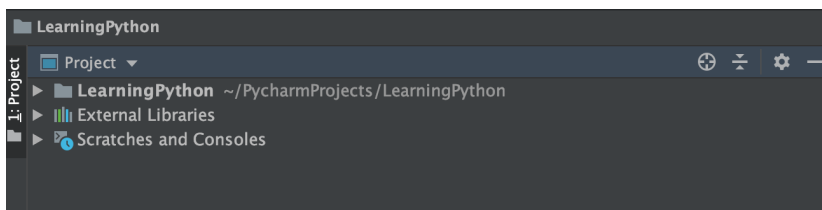
Once the connection is created with github, all repositories should be available under the dropdown button. Select “LearningPython” from the dropdown list and click “Clone”



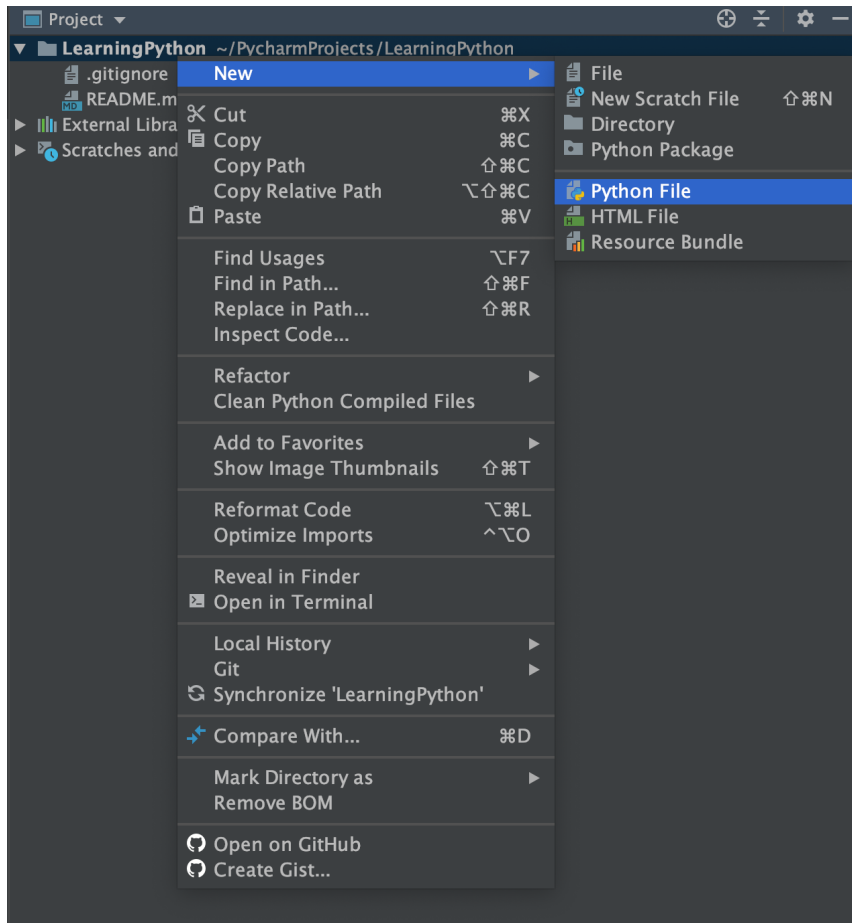
Click “Yes” below.



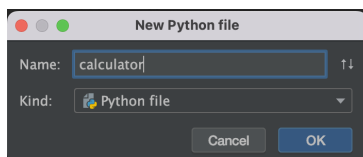
The repository is cloned and opened as below:



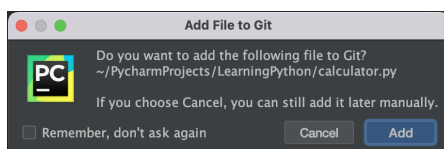
Right click on the “LearningPython” folder and select “New” “Python” file.



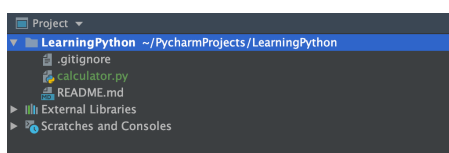
Enter file name and click ok,



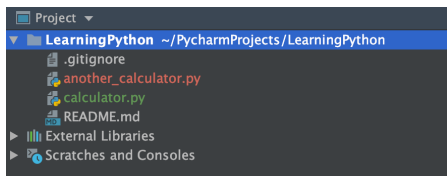
Click “Add”. Note: No issues if forgotten or accidentally clicked cancel.



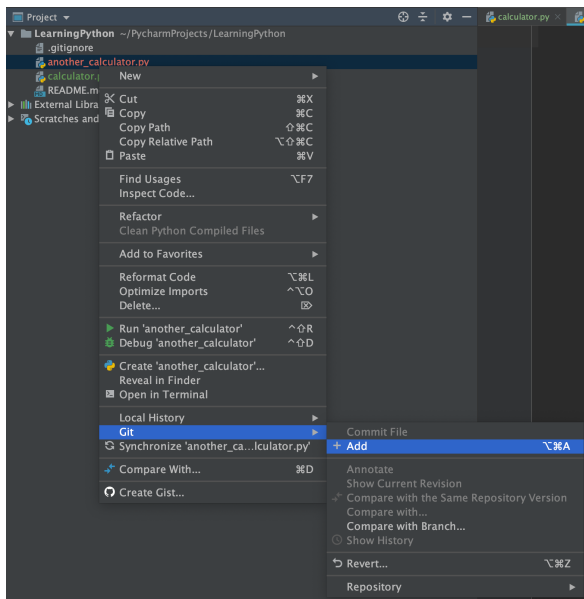
If you clicked “Add” the file will be shown as green in the left side panel,



If you clicked “Cancel” the file will be shown as red in the left side panel as below,

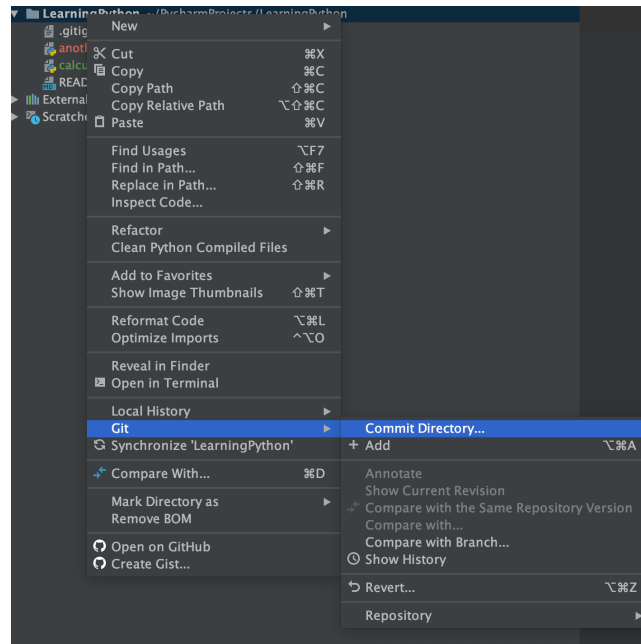


As the file in red is not tracked under git, first step would be to add it to git as shown below,

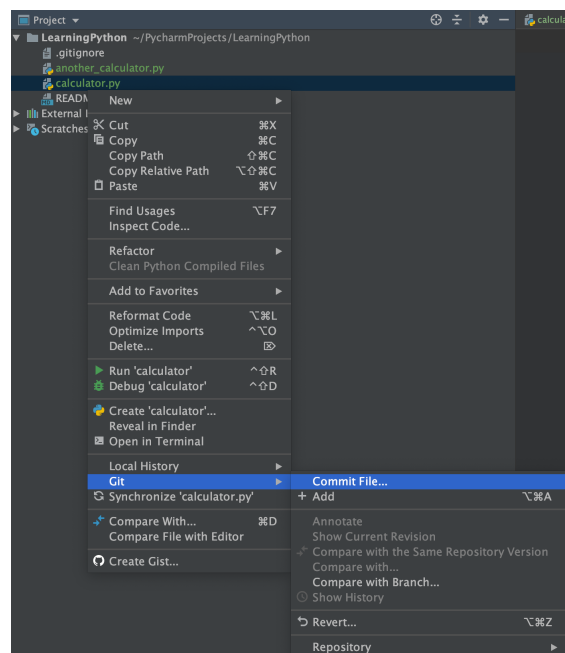


Once modifications are done to the python files and ready to checkin back to git - there are two options,

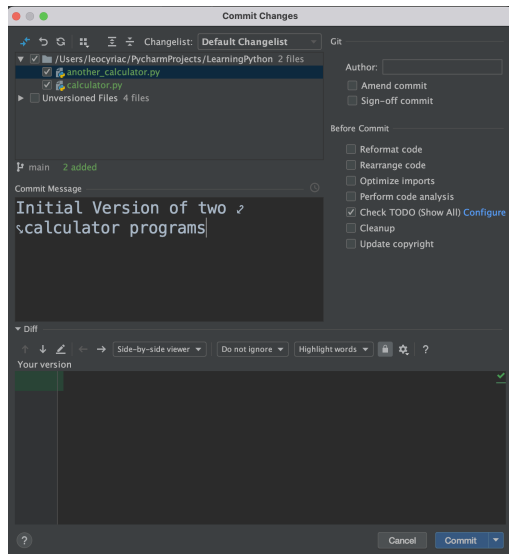
1. Checkin the entire directory



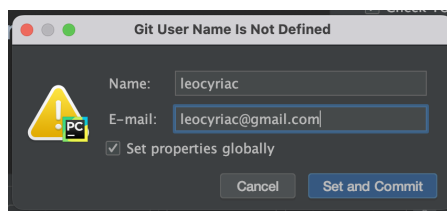
2. Commit each file



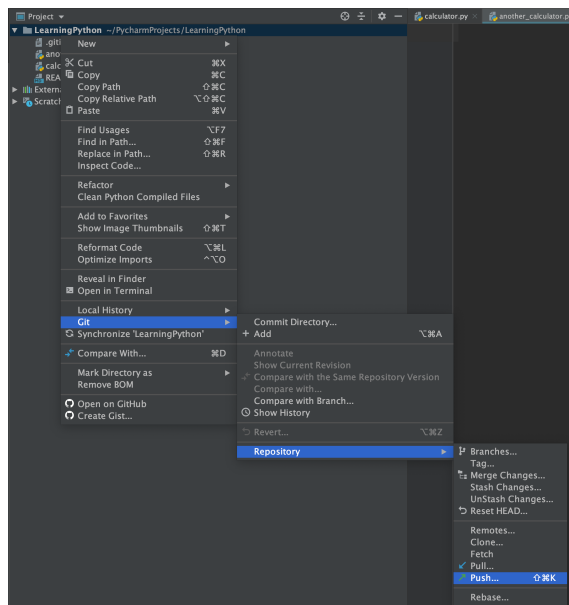
When you click on “Commit File/Directory” below screen appears. Enter appropriate commit message and click “Commit”.



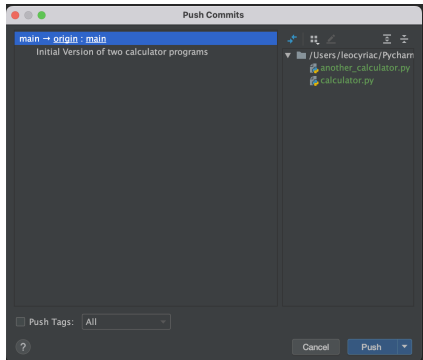
Enter details requested (this will be asked only once) and click “Set and Commit”



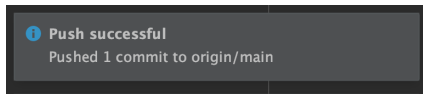
Now to push the changes to GitHub - Select Git → Repository → Push



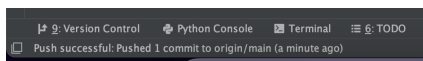
Click on “Push”



You should see a notification as below,



And should see the below at the bottom of pycharm,



Now check in GitHub if the changes are visible.

