GIS4x07 GitHub and exercise folder setup

Contents

[Intro 1](#_Toc80780192)

[Initial repository setup 1](#_Toc80780193)

[Create the GitHub repository 1](#_Toc80780194)

[Add collaborators 4](#_Toc80780195)

[Prepare for cloning 5](#_Toc80780196)

[Cloning the repository from GitHub to a local repository 6](#_Toc80780197)

[Pushing changes to the remote repository 8](#_Toc80780198)

[Stage changes 8](#_Toc80780199)

[Commit changes 9](#_Toc80780200)

[Push changes 10](#_Toc80780201)

[Pull changes 10](#_Toc80780202)

[Show hidden files, folders, and drives 11](#_Toc80780203)

# Intro

This document contains the instructions for creating remote (GitHub) and local repositories for GIS4x07 exercises. This example uses gis4107-week02. You will replace this with the appropriate course/week (e.g. gis4107-week04, gis4207-week02, etc.) as specified in the exercise documentation.

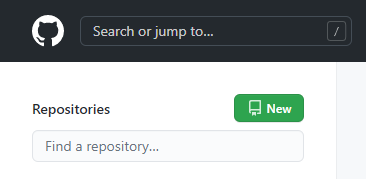
# Initial repository setup

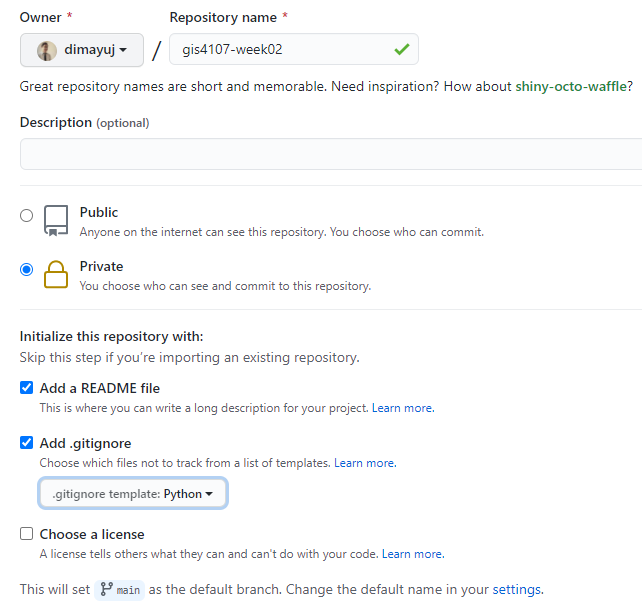
The repository owner specified by the instructor will do the initial repository setup. Ideally, the Owner would have done this before class. If not, their partner will standby/observe until this process is complete.

## Create the GitHub repository

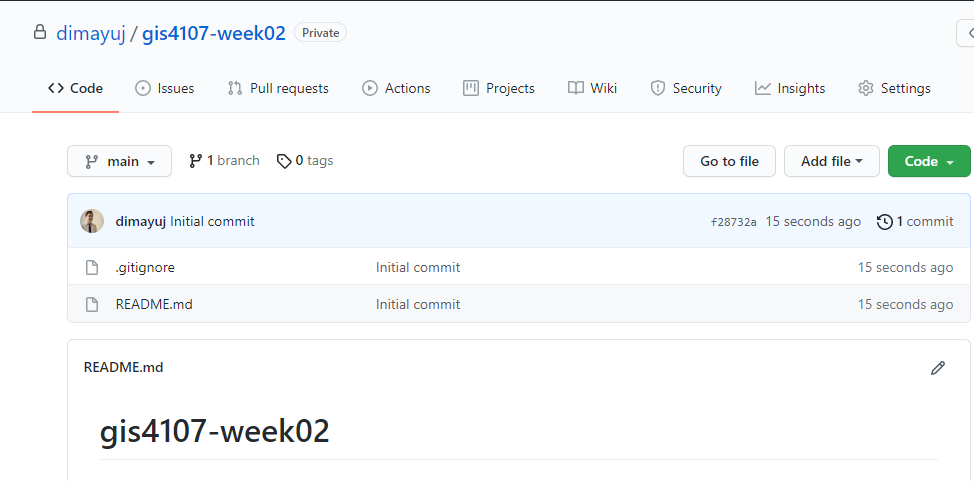
The owner will:

1. Navigate to GitHub.com and sign in with their credentials.
2. Click ‘New’ in the upper left of the web page to create a new repository.



1. Specify the name as instructed (e.g. gis4107-week02)
2. Select ‘Private’
3. Check the box for ‘Add a README file’
4. Check the box for ‘Add a .gitignore’ file
5. For the .gitignore template type ‘Python’ in the filter or select it from the drop-down list.
6. Click Create Repository  
   

In GitHub you will see the new repository as shown below.

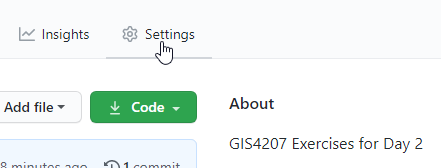


Note: dimayuj will be replaced with the GitHub username of the repository owner (e.g. acgis-abcd1234)

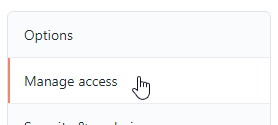
You have now successfully created a Private remote repository in GitHub.

## Add collaborators

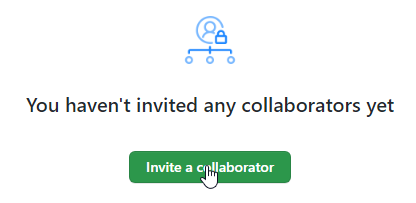
1. Add viljoed and your partner as collaborators as follows:  
     
   Select the Settings tab for the repository …



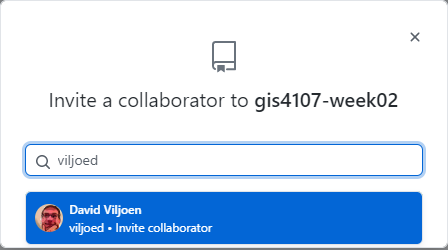
1. Select “Manage access” from left menu …



1. Click “Invite a collaborator”



1. Search / select /add collaborator …



1. Search / select / add your partner as a collaborator.

## Prepare for cloning

Copy the URL for the GitHub repository to your clipboard. You will use this for cloning in VS Code.

1. In the ‘Code’ tab you will find a green ‘Code’ drop-down. Click it then select the Copy to Clipboard icon to the right of the URL.

Graphical user interface, text, application, email

Description automatically generated

# Cloning the repository from GitHub to a local repository

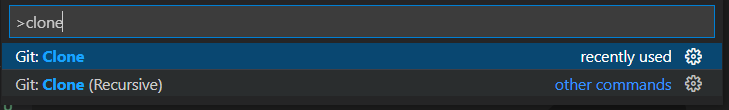
Now that we have a remote repository, we will use VS Code to clone that remote repository, locally.

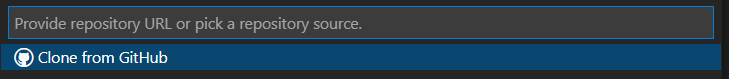
The owner and collaborator can both clone the GitHub repository to their computers.

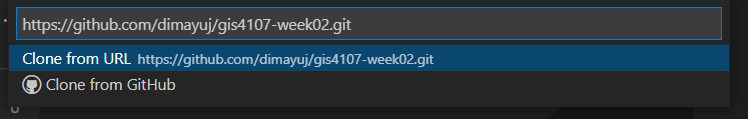
1. Open VS Code
2. Show the Command Palette by pressing either F1 or Ctrl+Shift+P. It should look like this at the very top with some drop-down options:



1. Start typing ‘clone’ and the Command Palette should autocomplete to Git: Clone as seen below:



1. Press ‘Enter’
2. You should now be prompted for a repository URL or source
3. You copied this to the clipboard earlier. If not, repeat the last step of the previous section and paste this URL into VS Code



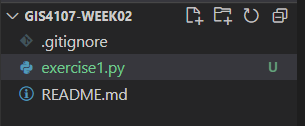
1. Press ‘Enter’. You should be prompted to specify a target directory for the local repository.
2. Navigate to your lab folder (D:\acgis\gis4107\_prog\week\_02\lab) and select that as your repository location.
3. VS Code will prompt you, giving you options to open, etc. Just select open.
4. You have now successfully cloned your repository on your local drive. You will find it in D:\acgis\gis4107\_prog\week\_02\lab\gis4107-week02. The git repository is in D:\acgis\gis4107\_prog\week\_02\lab\gis4107-week02\.git
5. Inside of D:\acgis\gis4107\_prog\week\_02\lab\gis4107-week02, create a folder with the first name and last name initial of the owner and collaborator separated by an underscore (e.g. DavidV\_JoseD). This will be referred to as your **exercise folder**. e.g.  
     
   D:\acgis\gis4107\_prog\week\_02\lab\gis4107-week02\DavidV\_JoseD  
     
   This will be where all your Python code for this exercise will be saved.
6. Use File > Open Folder to open the exercise folder in VS Code
7. Inside of D:\acgis\gis4107\_prog\week\_02\lab\gis4107-week02, add the following line to the top of the .gitignore file  
     
   .vscode
8. Make sure you save this change and stage/commit/push the updated .gitignore file to GitHub. The next section shows how you do this.

# Pushing changes to the remote repository

There are many ways to commit changes from your local repository to your remote repository.

One way is to administer your changes in the local repository and ‘push’ those changes back up to the remote repository. To illustrate how this will work, create a file for the first exercise this week in your working folder. For this example, let’s use exercise1.py.

The owner will:

1. Create the file (e.g. exercise1.py) in the working folder. I created mine in VS code. 

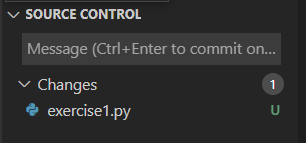
In order to push the changes back up to the remote repository we must stage the changes, commit those changes, and then push them.

## Stage changes

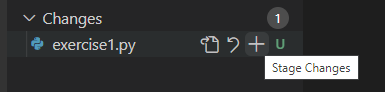
1. Click ‘Source Control’ on the left-hand menu.



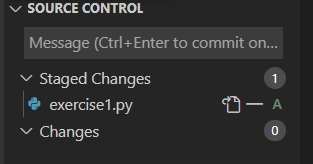
1. Click on the ‘Source Control’ menu and have a look inside ‘Changes’



1. If you hover over the exercise file you just created, you will see an option to ‘Stage Changes’

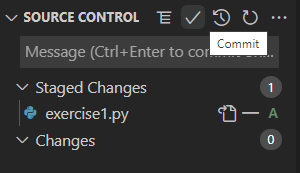


1. Click the plus icon to stage changes. Notice how the menu changes. You can always unstage changes if you need to by clicking on the unstage option.

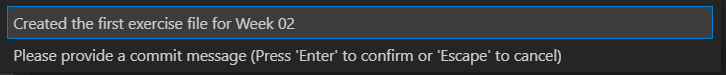


## Commit changes

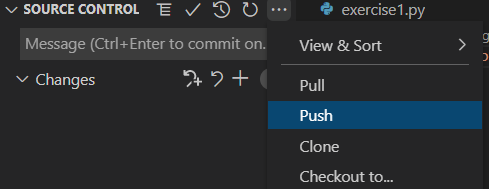
1. To commit the changes to your local repository, hover near Source Control and click on the check mark.



1. VS Code will prompt you for a commit message. Add something meaningful to document what you are committing to the local repository.



## Push changes

1. Lastly, in order to push these changes to the remote repository we need to click on the ‘…’ beside the Source Control menu and select ‘Push’. 
2. The changes will be reflected in the remote repository now. Refresh the web page in GitHub and you will see these changes.

## Pull changes

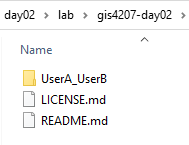
Assuming both the Owner and Collaborator have already cloned the repository, whoever did not **push** the changes can **pull** the changes to their local repository.

Graphical user interface, text, application

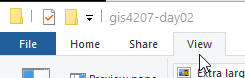
Description automatically generated

# Show hidden files, folders, and drives

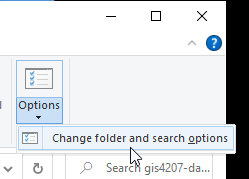
Local git repositories are stored in a hidden .git sub-folder. If you do not see it in File Explorer, e.g.



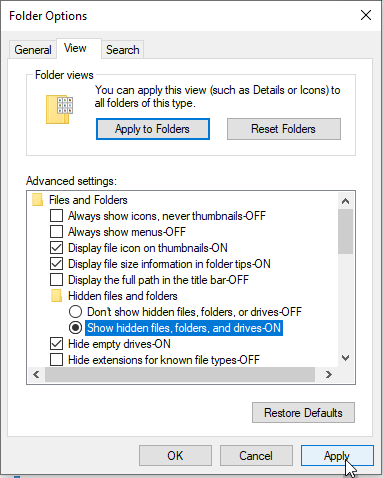
1. Select the View Tab in File Explorer



1. Select “Change folder and search options”



1. Select the View tab and “Show hidden files, folders, and drives” as shown below



1. Click Apply to see the change and click Ok to close the Folder Options dialog. The hidden .git folder will display with a slightly faded icon

