GIS4x07 Git Workflow

Intro

This document introduces version control with Git and outlines the process of using Git, SourceTree and GitHub for GIS4x07 Exercises. To make best use of this document, there are 3 prerequisites:

1. The reader has a GitHub Student account
2. Git for Windows is installed
3. SourceTree is installed

This document covers the simplest possible Git Workflow. It does not yet cover some of the other fundamental version control concepts such as branching, merging, reverting, diffing, etc.

To practice this “pull – edit – add – commit – push” workflow, a GitHub repository could be cloned into two separate folders to simulate two different users on different computers.

Contents

[What is Git 2](#_Toc492933148)

[Create a new repository in GitHub 2](#_Toc492933149)

[Add Collaborators 4](#_Toc492933150)

[Clone the GitHub repository to a local repository 5](#_Toc492933151)

[Stage, commit, push 7](#_Toc492933152)

[Pull changes from remote repository to local repository 8](#_Toc492933153)

# What is Git

Git is a version control system. The Git version control system has two main parts:

1. A repository – to store a set of files and the history of changes made to those files.
2. A Git client – a software that interacts with the repository

A local Git repository is a set of files in a .git sub-folder that keeps “snapshots” of the parent folder’s files and folders through time. The .git folder’s parent folder is called a “working directory/tree” or “working copy”.

GitHub is a web-based Git repository that stores the same set of files as the local Git repository. Changes to the local or remote repositories can be synchronized using Git client software.

The workflow for setting up remote and local repositories for exercises is:

1. **Create a new repository in GitHub**
2. **Add Collaborators**
3. **Clone the GitHub repository to a local repository**

After the setup of local and remote repositories, the workflow is:

1. **Pull changes from remote repository to local repository**
2. Make changes to your local repository (add/delete/modify files)
3. **Add and Commit changes to the local repository**
4. **Push changes from the local repository to the remote repository (GitHub)**

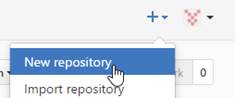
Pulling changes from the remote repository ensures you have any changes from your collaborators before you begin your work making changes to your local repository.

The bold items in the above will be covered in detail in the rest of this document.

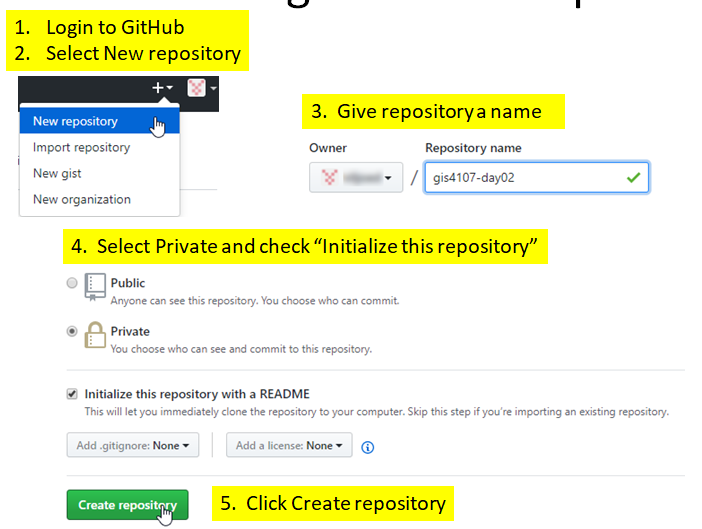
# Create a new repository in GitHub

Login to GitHub

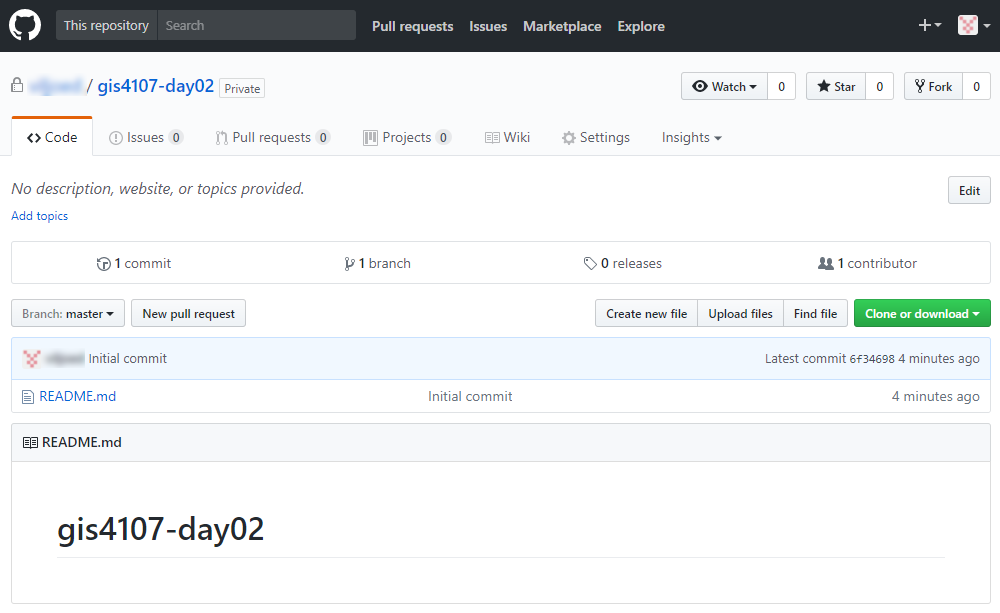
Select “New repository”:



1. For Repository name, use the pattern “course-dayN” (e.g. gis4107-day03)
2. Set it to be a Private repository
3. Check Initialize this repository with a README. This will allow you to clone this GitHub repository to a local repository.



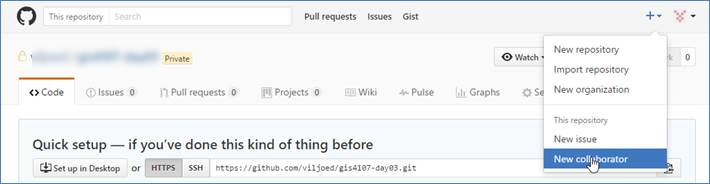
This creates the new repository shown below.



# Add Collaborators

You have now successfully created a Private repository and it will be the active repository.  Add viljoed and your partner as collaborators as follows:

Select New collaborator …



Search / select /add collaborator …

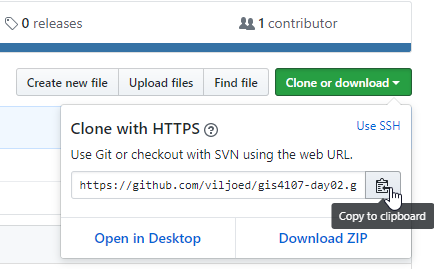


Search / select / add other collaborators (e.g. your partner)

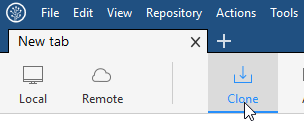
# Clone the GitHub repository to a local repository

A local repository will be created as a .git hidden folder in a folder of your choosing. In this example, E:\acgis\gis4107\_Intro2Prog\day02\lab.

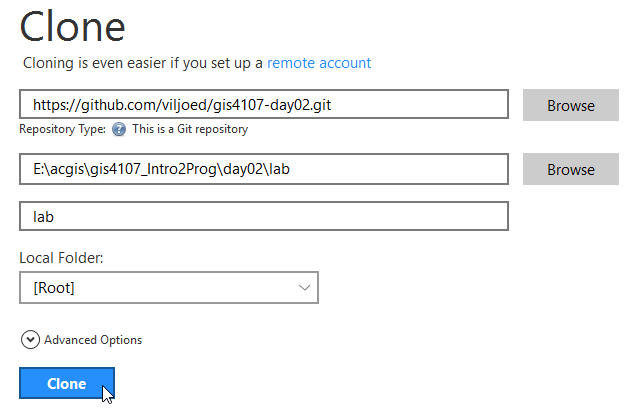
In GitHub, copy the URL for the repository you created above



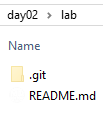
In SourceTree, click the Clone button



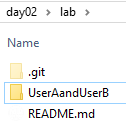
Fill out the Clone dialog and click Clone



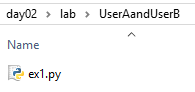
The Git repository is now cloned in E:\acgis\gis4107\_Intro2Prog\day02\lab



Create a folder in this working directory (tree) for you and your partner, e.g.

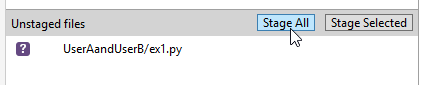


Where UserAandUserB will be replaced with the first and last initial of you and your partner. This folder will need a file before you stage, commit, and push to GitHub, e.g.



# Stage, commit, push

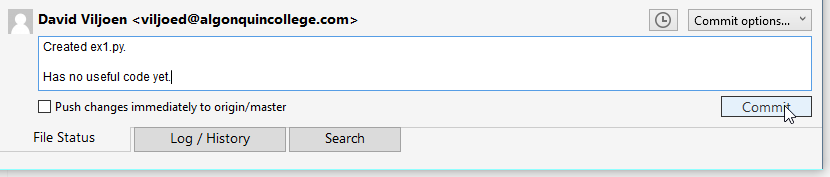
To stage the new file, in SourceTree click the File Status Tab. You will see your ex1.py file in Unstaged files. Click Stage All



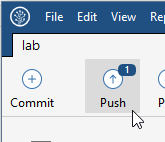
This will add the file to Staged files and show as an new (added) file:



In the commit comment box, add text to describe the commit, e.g.



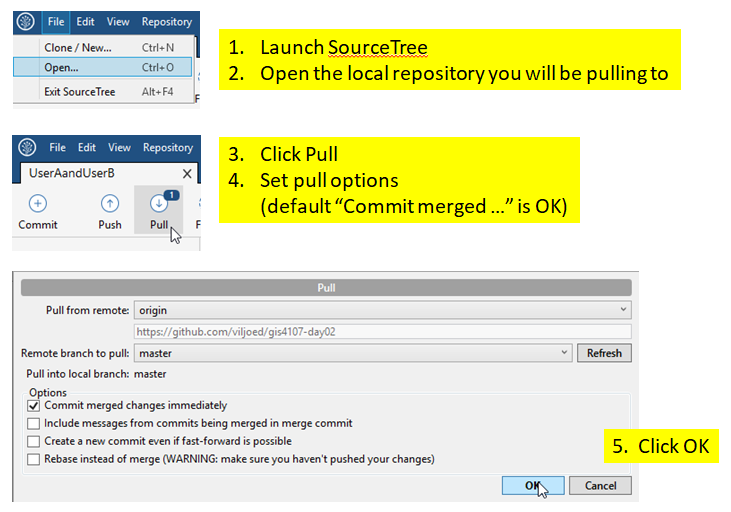
Now that the change has been committed, it can be pushed to GitHub. Click the Push button in SourceTree



This will present the push dialog that confirms the remote branch, etc. You will usually leave everything as-is and click Push in the bottom right corner. The code is now in GitHub and can be pulled by your partner.

# Pull changes from remote repository to local repository

The Git pull command will update the local repository with any changes that have been made to the GitHub (remote) repository. To Pull …



You can now edit existing files, add new ones, etc.