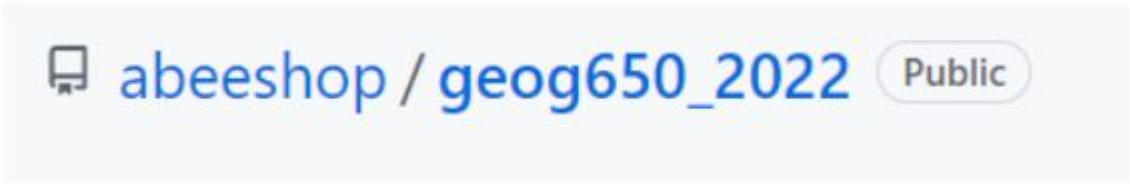
Guide to set up Git and Github

Compiled a quick and easy set up for GEOG650 which I am using for getting started with git and github. Workflow is referenced here
Processes are run from the terminal, I am using the cmd line shell Anaconda prompt
Note This tutorial will work for **Public** repositories only, additional verification steps needed for a private repository.

Installation and set up

1. Install git on local computer
2. Create your own github account and first public repository!



   Note: You can change to private at a later date, but will require additional step to push files to repository.

Part One: Connect your local directory with new repository

This step you will do once. This process creates a .git folder within your working directory which you will use to commit your files to github. The following coomands are run from the terminal.

1. Navigate to your working directory:

```
(geog650) H:\>cd "\jupyter\geog650"
```

2. Initate a new git project in your working directory - this will create the ".git" hidden folder:

```
(geog650) H:\jupyter\geog650> git init -b main
```

3. Add files to the staging area

   - This will add all files in the folder to be tracked. You can explore adding specific files or using .gitignore to change which files are tracked

```
(geog650) H:\jupyter\geog650> git add *
```

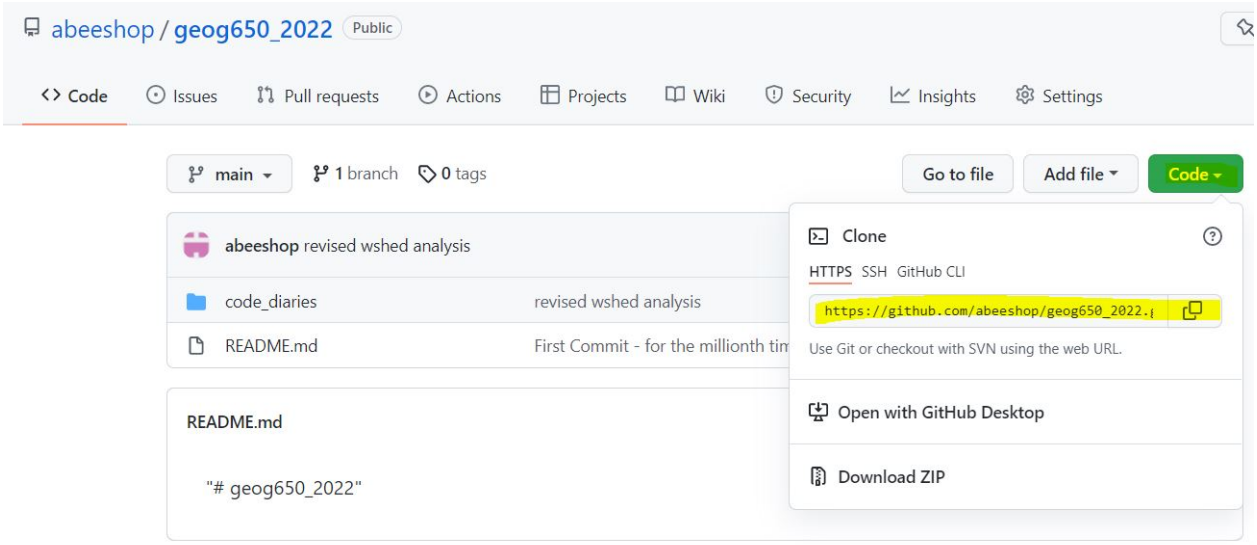
4. Commit the files within this directory to the staging area, ie. the .git folder

```
(geog650) H:\jupyter\geog650> git commit -m "First Commit of files"
```

5. Connect the github repository you created with this .git folder on local machine

- Provide the github url of your own repository followed with ".git"
- This can also be found by navigating to your github repository, and copying the link under code (image below):

```
(geog650) H:\jupyter\geog650> git remote add origin
https://github.com/abeeshop/geog650_2022.git
```



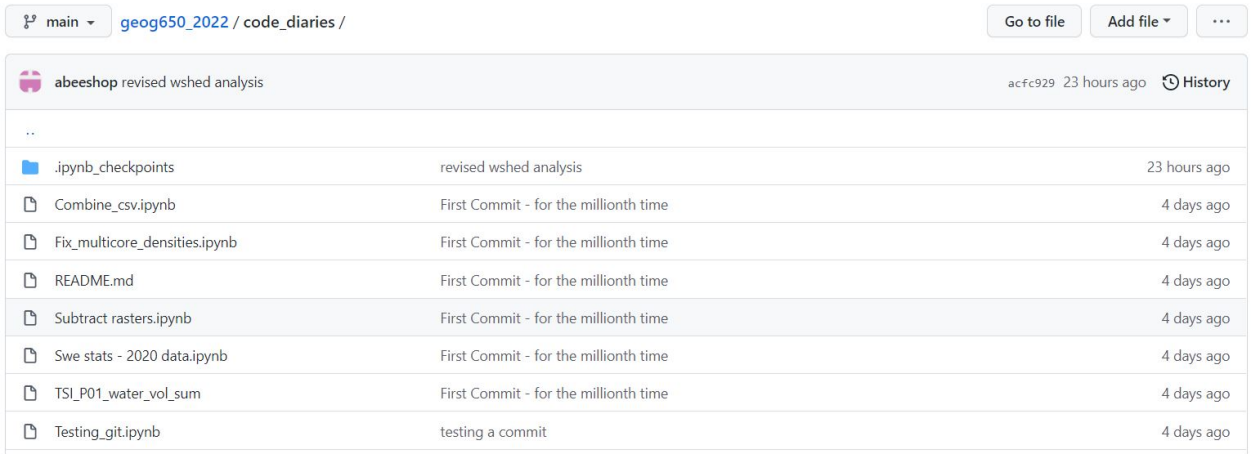
1. Create the first and main branch for your repostory.

(*Aside: I came across this article as to the coding conventionial words and inclusivity - interesting!*)

```
(geog650) H:\jupyter\geog650>git branch -M main
```

1. Push the first commit of files from staging area to github, and establish the tracking relationship between your directory and github
    - You should see your files alongside the commit message on your github repository

```
(geog650) H:\jupyter\geog650>git push -u origin main
```



Part Two: Version Control

This step you will repeat as often as you would like to back up your files. It's a simple four lines of code from the command line which will:

1. **Change directory** to project path
2. **Add** Files (only files which changed will be added)
3. **Commit** Files - with message in quotations
4. **Push** Files to your repository

```
(geog650) H:\>cd "\jupyter\geog650"
(geog650) H:\jupyter\geog650> git add *
(geog650) H:\jupyter\geog650> git commit -m "revised wshed analysis"
(geog650) H:\jupyter\geog650> git push
```

- Your Github should update to reflect your commit message, and you are in business!