

The present document aims at giving you all the info you need in order to get started here at CU. It covers a number of items which you should find useful during your PhD career here. All the examples in this document are guaranteed to work on a Unix platform (MacOS, Linux).

1 Git and Github

1.1 Brief description

Git is "a version control system that is used for system development". Not only can it keep track of all the changes imparted to code undergoing development by a single user, but its functionalities are extremely handy when several developers interact with the same piece of code concurrently.

Github is a website that offers a server architecture that you can synchronize with the local copy of the code. This remote code storage location is called "remote repository" in Git jargon. Repositories can be

- public: everybody can see the code, contribute or pull (i.e retrieve the most recent version of the code) from the repository
- private: the repository's owner is by default the only one able to perform any of the previously listed actions, unless authorized contributors are added to the repository access list.

Github permits one to create an infinite number of public repositories for free. Private repositories are typically not free, but you can upgrade your Github account to a Premium account (which allows you to create private repositories) thanks to your CU Boulder student status.

Note that alternatives to Github such as [Bitbucket](#) do exist. A summary of Git workflow is provided on Figure 1 along with the main Git commands.

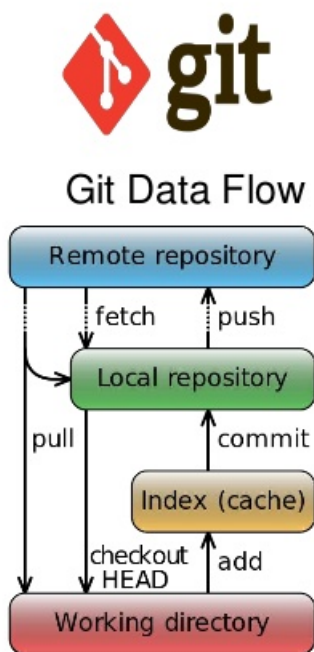


Figure 1: A summary of Git workflow taken from [here](#)

Figure 1 encompasses a good chunk of what you need to know about Git. The following sections will show you two important examples:

1. How to create a local repository on your computer, a remote repository on Github and synchronize the two. In particular, the *init*, *add*, *commit* and *push* commands will be exemplified.
2. How to contribute to code in a collaborative environment.

1.2 How to set up a local/remote repository

Create the local repository

We begin by opening a terminal window into the folder where we want to create our repository. In this example, the repository will be in a folder named *NewStudentHandout* located in the *Documents* folder of a Unix system. On a Mac, the terminal command *pwd* would return */Users/my_username/Documents/NewStudentHandout* when executed from *NewStudentHandout*. Once we have made sure that the window terminal is in the right folder, let's type

Listing 1: Creation of the local repository

```
git init
```

If everything goes well, the following should appear on the terminal window:

Listing 2: Successful git init message

```
Initialized empty Git repository in /Users/bbercovici/Documents/NewStudentHandout/.git/
```

This command initiates the local git repository by creating the hidden directory *./git* and populates it with a number of configuration files. These files are the backbone of your local repository, so make sure not to alter them.

As noted by Git, our local repository is empty. Note that the working directory content is unrelated to the local repository content as those are two separate entities as shown on Figure 1. We thus have to use *add* to add files to the index cache, and *commit* this index to have those files added to our local repository.

Let us start with the *add* command. It can be used in two different ways, as shown below (assuming that *myfile.txt* does exist in the working directory):

Listing 3: Add files to the cache

```
git add myfile.txt
# the index cache only contains myfile.txt
git add *
# the index cache now contains all files in the working directory that were not excluded in .gitignore
```

Another useful command is *status*, which highlights files depending on whether they have been modified since the last commit. Since *git status* lists *myfile.txt* in green, we are now ready to commit it to the local repository using the following command:

Listing 4: commit files to the local repository

```
git commit -m "first commit: myfile.txt added to the local repo"
```

The *-m* options allows us to append a message to the commit. After the content of the local repository is pushed to remote repository, the other developers will be able to read the message and quickly understand what this commit was about.

Create an account on Github

The title of this section is quite self-explanatory. Make sure you remember your username/password as you will need them shortly!

Create the remote repository on Github

Log on to Github and click on "New repository" in the drop-down list next to your profile picture as show on Figure 2.

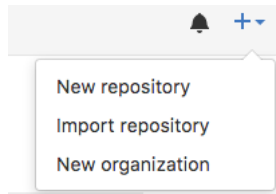


Figure 2: New repository

The following window (shown on Figure 3) allows you to choose several important settings for your remote repository.

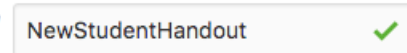
Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

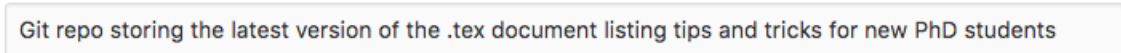


Repository name



Great repository names are short and memorable. Need inspiration? How about **ubiquitous-succotash**.

Description (optional)



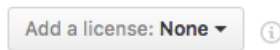
Anyone can see this repository. You choose who can commit.



You choose who can see and commit to this repository.

☐ Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.



Create repository

Figure 3: New repository settings

- Choose whether the repository will be made public or private (the latter being only available if you have upgraded to a premium account).
- Initialize the repository with a README.md file.
Leave this box unchecked if your local repository is already created, as this file can be added later on manually without risking conflicts
- Add a .gitignore file. This file lists all the file extensions that must be ignored by Git. Especially handy if there are specific file extensions that you never want to see included in a commit.
- Add a license file.

When you are ready, simply click on Create Repository and proceed to the next step.

Connect the local repository to the remote repository

You should now be seeing the same page as on Figure 4.

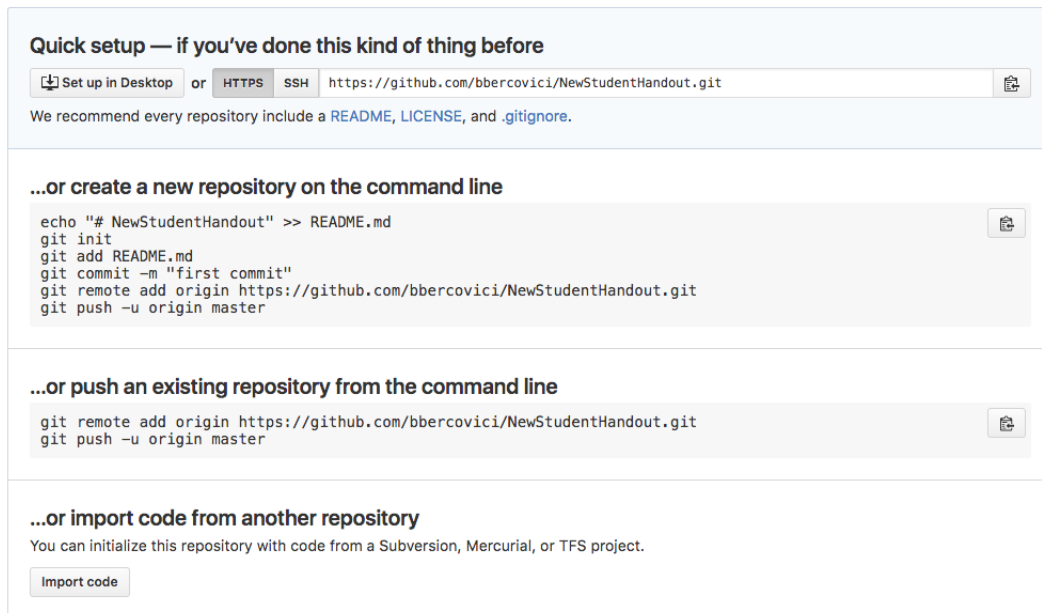


Figure 4: Final setup options

The third option is what we want to do, since we have already created a remote repository. We can copy and paste the two lines under "...push an existing repository from the command line" into our terminal. Start with the following:

Listing 5: Add the remote repository location

```
git remote add origin https://github.com/bbercovici/NewStudentHandout.git
```

This will add the remote repository address to one of the hidden configuration files. Finally, push the local repository to the remote.

Listing 6: Push the local repository to the remote

```
git push -u origin master
```

We are now all set! We can keep working in our local directory, add files containing new content to the index using *git add*, commit the changes to the local repository using *git commit*, and finally pushing the changes to the remote using *git push origin master* (note that the *-u* option was not used here). Now is also the time to create a *.gitignore* file and a *README.md* file.

2 Latex

3 Asymptote