GIT

Sign up to git

<https://github.com/>

* **What is git**
* **Commands at a glance**
* **Creating a Repository**
* **Pushing to Git**
  + **Adding files**
  + **Making a commit**
  + **Pushing to git**

**To be done**

**Branching**

**Merging**

**Publishing**

**What is Git?**

Git is a source control for many projects, it is used to manage a project over time, Git uses repository data structures to keep track of changes to its files. Git can be used for projects of all sizes and varying amount of contributors. A Git repository will allow users to clone a project, branch it and then later merge it back in if they wish.

Git allows for online and offline development, the globally accessible repository is stored online via github. Everyone who has a copy of the project on their local machine has the same content as the stored online one, they can view history and previous commits. This allows users flexibility to create commits locally and then later on push them to the global repository when they have internet connection.

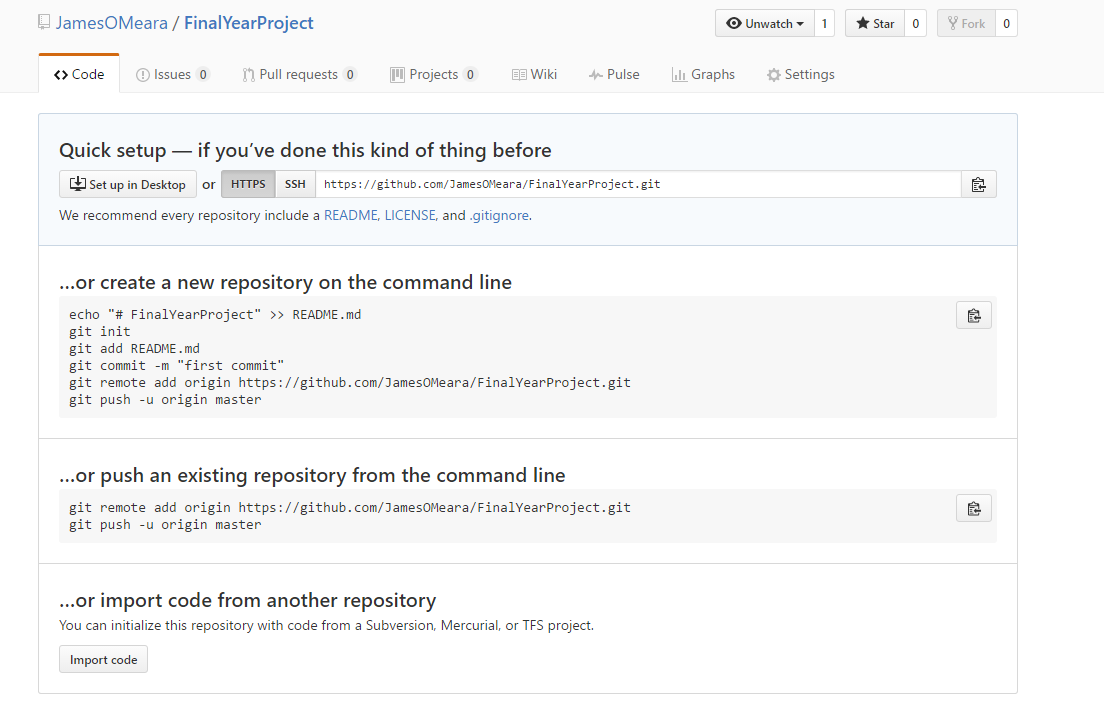
**Commands at a glance**

<https://services.github.com/kit/downloads/github-git-cheat-sheet.pdf>

* git status
* git commit -m “message here”
* git push
* git branch
* git reset

**Creating a Repository**

Create a repository and it will instruct you to follow the steps shown below.

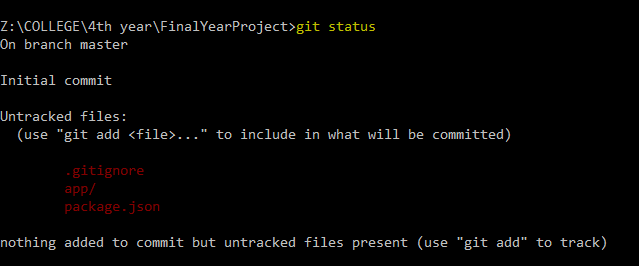


Once you download and install git on your local machine

CD into the directory you wish to save to source control, for this example I am inside the “FinalYearProject” directory and everything inside this will be saved to source control.



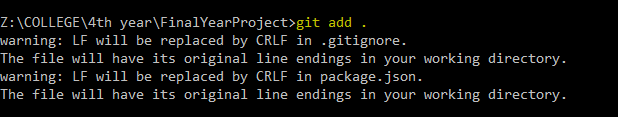
Using the command **“git status”** you can check to see what has been modified, has yet to be / or been committed.



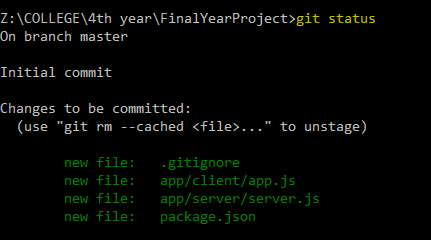
Now since we have seen what files have been changed and we want to commit and push these to source control we can add them to our local commit.

Use the command **“git add”** followed by the name of the file you would like

Or you can add all by: **“git add .”**

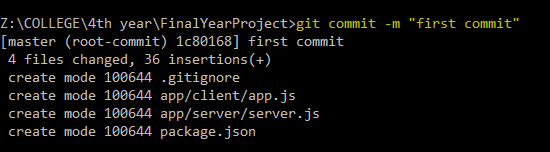


Call “git status” again and now you will see what files are added and ready to be committed, as you can see below they are showing up in green.



Finally, now since we are happy with the files we have added, we can now create a commit. To do this us the command **‘git commit -m “your message here”’.**

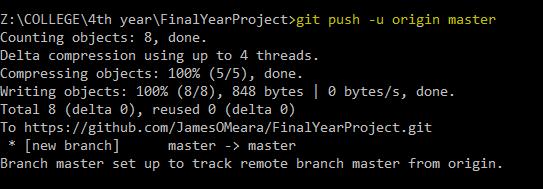
The -m stands for message to accompany the commit. It is good practice to do multiple commits for certain files and new features. Give the commits meaningful names, as this will help greatly in the future identify what changes were made.



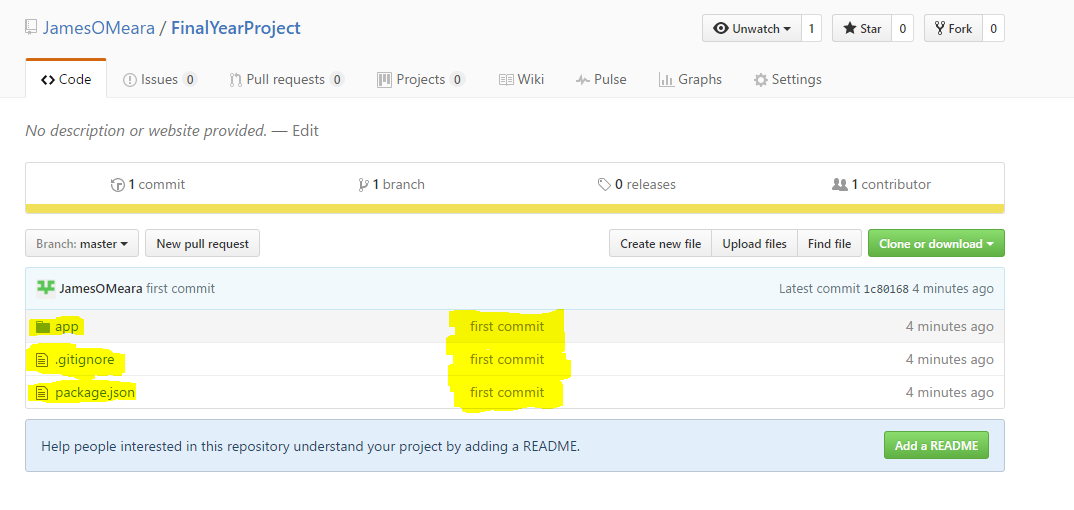
This next command is a one off command, since here we are first initializing the local repository, this command for your repo is shown in the browser when you create the repository, or in the first image in this doc.



Finally after we have all these steps complete, we can now push our local commits to our repository



And here we can see the changes made in our browser.



**Pushing to Git**

**Adding Files**

When we add files, we are adding them to our local repository (staging them, to be ready to be pushed).

To do this:

**“git add .”** for to add all modified and new files

Or specify with

**“git add someFolder/someFile.js”** etc

\*I’ve added files to be stages, but I want to remove them

Use **“git Reset”** – this will not do anything to your files, it will simply un-stage the files you just added.

**Making a Commit**

Now that we have files staged and ready, we must put them into the envelope, you could say and write and address/message on it. Here we are saving these stages files into a commit and attaching a message to it.

Use command:

**“git commit -m “your message here”**

**Pushing to Git Repo**

And finally we can push this to our global git repository

Use command

**“git push”**