Pushing to your Github Repository

When you want to push your code because you've either reached a good checkpoint or you've made some changes to your code that you want to be "saved" in version control, you will need to follow these commands:

1. git add [file/directory]

i.e. git add .

- this will tell Git to track these files or directories for changes. This way these changes will be pushed to the repository. Usually you'll be using "git add ." to add all of the current directory. To avoid adding files you don't want to be recorded, make use of the .gitignore file. Any file you add in that file will be ignored always.

2. git commit -am "insert useful commit message here"

i.e. git commit -am "Completed function x y z in file abc.py"

- this will commit the changes you have made to your local .git history. This "pushes" changes to your local repository (the folder where all your stuff is).

3. git push [remote] [branch]

i.e. git push

i.e. git push origin master

- this will push your local commits and changes to the Github Repository online located at the remote given in the command line. By default, you will be pushing to the remote, "origin" because that is the default remote that is created when you initialize the repository. The default branch is also "master". If you haven't manually changed branches, git push on its own will push to the default remote and branch.

Pulling Files Paul created on his Repository that you Forked From

When Paul has uploaded homework files or whatever, you can pull them from your repository if this repository is the one you forked from him.

1. git fetch upstream

This will fetch files from the remote "upstream" which is set to Paul's original repository.

2. git merge upstream

This will merge the changes in upstream to your own repository

Alternative to the above commands

This alternative method involves adding Paul's Directory as a remote called upstream. Whenever Paul makes changes, you can simply pull from this remote to get the files.

1. git remote add upstream [Paul's REPO URL]

- This adds the Paul's Repo to a remote named upstream

2. git pull upstream master

-This pulls from Paul’s Repo’s Master branch all the files you will need