Useful GIT/CMD Commands

- `$ mkdir name`: “Make Directory”. Creates a new folder with the specified name.
- `$ cd name`: “Change Directory”. Goes to specified directory.
- `$ git init`: “Git Initialise”. Initialises an empty Git repository.
- `$ pwd`: “Print Working Directory”. Prints current directory.
- `$ cd ~`: “Change Directory to Home”. Goes to home directory.
- `$ cd ..`: “Change Directory back”. Goes back by one directory.
- `$ ls`: “Listing”. Lists all connecting directories from current directory.
- `$ clear`: Clears current command prompt.
- `$ git add <file>`: Adds current changes to the repo from specified directory.
- `$ git commit -m “label”`: Creates a ‘snapshot’ of current working code, so that you can easily go back to it.
- `$ git status`: Tells you the current status of the repo i.e. if any changes have been made and you need to commit again, or if everything is the same as before.
- `$ git reset`: Undo any changes added to original commit
- `$ git add <file>`: Adds current file to the repo.
- `$ git add .`: Adds current changes to repo from current directory.
- `$ git log`: Shows you the history of your commits
- `$ git log --pretty=oneline`: Shows you the history of your commits in a more condensed ‘prettier’ format.
- `$ git checkout <hash>`: Lets you go back to a commit and make any experimental changes to it without having to actually affect your other branches.
- `$ cat <file>`:Prints out the current contents of the file in cmd
- `$ git checkout master`: Returns to the master branch (most recent commit).
- `$ git tag <tag>`: Tags current version of the program with the specified name
- `$ git checkout <tag>^`: Returns to the previous version than the one stated.
- `$ git tag`: Shows every tag available
- `$ git hist master --all`: Shows history with tag versions
- `$ git reset HEAD <file>`: resets the staging area to whatever is in HEAD.

Useful GIT/CMD Commands

- $ git revert HEAD: Reverts the most recent commit back to its previous version
- $ git reset --hard <tag>: Resets back to the version specified. Make sure you tag the version you want to get rid of to a different name, before you do this
- $ git tag -d <tag>: deletes the specified tag from the repository
- $ git commit --amend -m “comment”: amend last commit to new edit
- $ git mv <file> <folder>: moves repo for the specified file into the specified folder
- $ git rm <file>: removes file from repo without deleting it
- $ ls -C .git: prints out your git directory when used in the root of your project directory
- $ git hist --max-count=1: finds the latest commit
- $ git remote add origin <web address>: connects github repo to your home directory
- $ git push origin master: pushes branch to github
- $ git remote rm origin: clears the origin URL
- $ git fetch origin: downloads new data from remote repo
- $ git pull origin master: downloads new data from remote repo and integrates it into your current working copy files
- $ git checkout -b <label>: creates and goes to a new branch with the specified label
- $ git branch -d <label>: deletes the specified branch