Git Commands

Please note that in order to use GitHub now, having a regular password alone is not enough. You need to set 2 factor Authentication as well as set a Personal Access Token. Use the Personal Access Token instead of your regular password when using Git via your Terminal/ CMD.

Command	Function
Terminal commands	
$ ls	List the directories and files inside the current directory
$ ls -a	List the directories including the hidden files in the list of directories and files. This is helpful when trying to find hidden files like .git or .gitignore
$ cd [folder]	Go into the folder. e.g. cd Desktop/Developer
$ clear	When you need a fresh Terminal window
$ pwd	print working director
$ ~	home directory
$ ..	up one directory
$ -	previous working directory
$ mkdir	create new directory
$ ps	list all running processes
$ kill	terminate existing process
$ rmd	permanently delete file
$ rmdir	remove directory
Git Commands	
$ git status	List the files you've changed and those you still need to add or commit:
$ git commit -m "Here is my commit message"	Apply the changes to your local Git repository with a message briefly outlining the changes you made.
$ git push	Push changes you have made locally to update the remote repository
$ git pull	Pull changes that have been added to the remote repository by a collaborator to update your local repository.
$ git clone <URL>	Clone the repository.
$ git checkout -b <new-branch>	Create new local branch
$ git push -u origin <new-branch>	Sync local branch with remote
$ git push origin <branch>	Push branch to remote
$ git checkout <branch>	Checkout branch
$ git branch -d <branchname>	deletes local branch
$ git push origin :<branchname>	deletes remote branch
$ git subtree push --prefix docs origin gh-pages	push docs as subtree to gh-pages
$ git init	Create an empty Git repository in the current directory. By default it will have one branch named master.
$ git clone url	Clone the Git repository from url. This may be over HTTP, SSH, or the Git protocol, or it may be a path to another local repository.
$ git add .	add those 'unknown' files
$ git branch	show list of all branches (* is active)
$ git checkout master	go back to master branch
$ git branch -m <oldname> <newname	rename branch
$ git branch -m <newname>	rename current branch
$ git branch -d <branchname>	deletes local branch
$ git push origin :<branchname>	deletes remote branch
$ git remote prune <branchname>	update local/remote sync
$ git log master	show history of branch "master"

Git Commands

$ git merge <branchname>	To merge a different branch into your active branch:
$ git diff <sourcebranch> <targetbranch>	Preview changes, before merging:
$ git --version	Get the installed version
$ git gc	Cleanup unnecessary files and optimize the local repository
$ git remote -v	Verifies the new remote URL The respond should be something like origin https://yourname@github.com/yourname/yourrepo.git (fetch) origin https://yourname@github.com/yourname/yourrepo.git (push)
Create README.md	touch README.md nano README.md #### ADD YOUR INFORMATION #### Press: control + X #### Type: Y #### Press: enter
$ cat README.md	Show the contents of the README.md file

Add specific file to Github	$ git add README.md
	$ git commit -m "Adding readme file"
	$ git push -u origin master

$ git add file	Add or update file from the working tree into the Index.
$ git reset file	Unstage changes to file in the index, without touching the working tree.
$ git checkout file	Undo modifications to file in the working tree by reading it back from the index.
$ git rm file	Delete file from the index and the working tree.
$ git mv oldfile newfile	Shortcut for mv oldfile newfile plus the appropriate additions and removals in the index.
$ git commit	Make a commit out of the current index.
$ git commit -a	Shortcut for adding all modified files to the index and committing.
$ git log	List the commits on the current branch.
$ git show object	Show an object (e.g. the log information and patch for a commit, or the contents of a file).
$ git diff	Show the differences between the index and the working tree.
$ git diff --cached	Show the differences between HEAD and the index.
$ git diff commit	Show the differences between commit and the working tree.
$ git tag tag [commit]	Attach a new tag named tag to commit (defaulting to current master).
$ git tag -d tag	Delete the tag named tag.
$ git config --global user.name "Your Name " **$ git config --global user.email "your@email.edu "**	Configure your Git account.
$ git merge commit	Merge commit into Master.