Git Commands & Notes

//Common
git status
git add .
git commit -m ‘commit message here’
git push

//Notable
git pull origin (fetchs and merges)
git stash

-

//Create local repository
git init

//Create local mapping
git clone {url}

//Stage file
git add
git add .
git add *.html

//Create ignore file
touch .gitignore
(add *.log to the created .gitignore file)

//Get repository Status
git status

//Commit changes to repository
git commit
    i - add comment
    esc - escape after providing message
    :wq - finish comment
git commit -m ‘commit message here’
git commit -a -m ‘commit message here’ (-a: allows you to skip the staging step but only for files that are currently being tracked)

//Branching
git branch NameOfBranch

//Checking
git checkout NameOfBranch
git checkout master (master is always the name of the main branch)

//Merge (from the master branch)
git merge NameOfBranch
*Conflict will occur when merging, git adds comments to the file that has the conflict and will have to manually resolve the conflicts. Note, the git status will tell us what files have changes. Once resolved run the “git commit -a -m ‘commit message here’” command once again

git mergetool (for external merge tool to help this process)

//Create file
touch filename.extension

//Shelving or Stash pending changes
git stash

//Retrieve pending changes in stash
git stash apply

// View all remote repositories
git remote
git remote -v (-v: will show the urls for the remote repository)

/Creates a local map for repository
(Create GetHub repository - Note the clone https)
git clone ‘url’

//Changes to map
cd NameOfRepository

//Get Latest (origin is simply alias for the url)
git fetch origin
git pull origin (fetchs and merges)

//Check into Repository (master specifies the branch)
git push
git push origin master

//Add remote repositories
git remote add NameOfRepo url