####Git Setup - Configuring Username/Email

git config --global user.name "Aayush Tuladhar"'

git config --global user.email "aayush.tuladhar@gmail.com"'

####Creating an empty GIT repository git init

####Adding all files to GIT git add .

####Performing local commit with Message git commit -m "Message"

####Creates a Tag Object (-a = annotated text object) git tag -a v0.1

####Reverts changed you performed in the git checkout -- <filename>

####Adds file track list git add <filename>

####Performs difference with the HEAD git diff HEAD

####Perfoming a Commit (-a: All ),(-m: Message) git commit -a -m "Message"

####Take the last commit back. git reset --soft HEAD^

####Give's you a snapshot git difff ORIG HEAD

####Undoing the commit you performed (Getting the Original Head Back) git commit -a -c ORIG\_HEAD

####Cloning a Remote git clone git://githubaddr <directoryname>

####Fetches from the ORIGIN and merge to the current branch git pull

####Reverts the pull from the ORIGIN git reset --hard ORIG\_HEAD

####Shows status of current working directory git status

####Shows list of branch git show-branch

####Shows commit logs,-p (Performs diff on each commit log) git log -p

###Git Branching Explained

**Creates a new branch from the old branch**

git checkout -b <newbranch> <oldbranch>

**Switch to**

git checkout <branch>

git checkout -b # -b Causes new branch to be created

git checkout -f # -f Forcing to change branch. This is used to throw local changes

####Merge contents of to The --no-ff cause merge to always create a new commit object, even if the merge should be performed with a fast-forward.

git merge --no-ff <newbranch>

git push origin develop

####Creating a release version git checkout -b release-1.2 develop

####Making a commit git commit -a -m "Release Version"

####Finishing a release branch

git checkout master

git merge --no-ff release-1.2

git tag -a 1.2

####Merging contents of release1.2 to the develop branch

git checkout develop

git merge --no-ff release-1.2

####Deleting branch git branch -d release1.2

####Creating HotFix Branch git checkout -b fix-3.2.1 master

####Finishing a hotfix branch

git checkout master

git merge --no-ff fix-3.2.1`

git tag -a 1.2.1`

git checkout develop

git merge --no-ff fix-3.2.1`