

git restore --staged <filename> to unstage a file on the local branch

When you delete a remote branch from git and not from your command line, your local .git file is not updated with the delete branch. When you do a ‘git branch -a’ you are able to see the deleted remote branch in the list of remote branches.

Solution: git fetch -p ; where p means prune

*To set an origin for your empty repository –*

git remote add <variable name e.g., origin> <url to the repo git file e.g., [www.github.com/georgejohnchavady/georacle.git](http://www.github.com/georgejohnchavady/georacle.git)>

*Pushing a file to git -*

git push <remote repo e.g., origin> <local branch name e.g., testbranch>

*Stash –*

Stash to temporarily save the contents of your working branch to do a git pull.

git stash save “<name>” – to stash the changes

git stash pop – to restore the latest stashed changes

git stash list – to list the stashed changes

git stash apply <int>; 0 for the top most in the stack

*Branching statement –*

*git checkout -b <branch name>* to create a branch and switch to the same branch

*git branch -m <branch name>* to rename the current branch

git branch -d <branch name> to delete a branch; Note: you cannot delete the branch that you are currently in

git branch <branch name> to create a branch with name <branch name>

**Cleaning the Working Branch**

*git checkout -- .*  🡪 to restore all the modified files to the latest commit

*git clean -f* 🡪 to remove all the untracked files in the current branch

*git clean -fd* 🡪 to remove all untracked files and directory in the current branch

git restore –staged <file url> 🡪 to unstage a file

git diff 🡪 gives the changes made to a file in the current working directory

**Logs**

git log –oneline 🡪 helps you get the head of each commit require to revert/reset

Branches – master, dev(develop), prod

Environments – Dev(Develop), QA and production(prod)

**Merge**

git checkout <master branch>

git merge <feature branch name>