Complete Git and Git-hub

\*Git-It is a version control system. It allows to maintain the history of a project, at what particular time which person change which change where in the project.

\*Git hub- It is an online platform which allows us to host our git repositories.

\*Repository- It is just a folder where changes are saved.

\*Untracked files- relatives whose photos has not been taken with the couple.

Commands –

* .git
* Git status
* Git add . – to placing in files whose photos not taken yet (staging/ track). Adds all the untracked files in the present directory.
* Git add file-name – to place name of individual untracked file in the files whose photos is not taken yet. (Used when there more than one files in the working directory)
* Git commit -m “message” – to unstage/click photo of staged file.
* Add-content path
* Cat file-name
* Git restore --staged file-name - to remove file from staging area if added by mistake
* Git log – to see history of all commit
* Rm file-name : to delete file
* Restore “commit”: to remove all above commits
* Git stash: to place all the changes in staged area to back area. And we can call them whenever needed. It will let us clean history.
* Git stash pop : to call all the changes in the back side area to staging area.
* Git stash clear: to delete all the changes in the back stage area.
* Git remote add origin urlOfTheRepo: to host to git-hub
* Git clone url
* Git remote add urlOfFromwherefolded : to add upstream url
* Git push origin master/branch-name: to push the changes to git-hub
* Git remote -v: to see url of origin and upstream.
* Git branch name: to make new brach
* Git checkout nameofBranch: head will come to that branch
* Git push origin branchName -f: to force push a commit on git-hub. Used when online repo contains a commit that my repo doesn’t.

Note:

* Whenever you are working on new feature or resolving a bug always create a separate branch.
* For every new feature or new bug , create a new branch
* Origin is my url and upstream is url from where I have folked the project.
* Git config – global user.name “[Firstname]”
* Git config – global user.email “[email]”