GIT-HUB

1. Move to your project folder using cd/d:
2. To check git version type git –version;
3. If you want to send project to git-hub you first need to initiate it first time by running command git init
4. If you are on mac/ Linux terminal or using git bash then type ls –a to view directory files and folder including hidden files ;
5. After initiate git then run git status to check the status of files and folder
6. git add <file name> to add single file in staging area. A staging area is the area where you put those files which you want to push in your git-hub repository. And if you want to add all files and folder then run git add . it will put all the files and folder in staging area
7. After adding the files in staging area now make a commit. A commit tells others people what these changes are about. To add commit type git commit –m ‘msg’ –m flag means msg.
8. A very important and optional thing is generate ssh keys. To generate an ssh keys follow these steps,
    - ls –al ~/.ssh to find out if any key present.
    - If not type ssh-keygen –t rsa –b 4096 –C ‘<email>’ and enter and enter and enter.
    - id_rsa file contains private key don’t give to anyone
    - Now type eval “$(ssh-agent –s)” and it will generate agent id.
    - Now add id to rsa by typing ssh-add ^/.ssh/id_rsa.
9. Manually compare files using command line.
    - For Window=> FC <filename> <filename>
    - For Mac & Linux => diff –u <filename> <filename>
10. These are different version control system like manual saving, Dropbox, Google docs, Wikipedia, git, SUN, CVS.
11. git log to view all the commit made in repository.
12. git diff commit id ,commit id to view difference in files a.
13. git log –stat gives the static of each file change at every commit.

Staging area between working area and repository.

1. git diff without ids shows difference in working area and staging area.
2. git diff –staged shows difference in staged files and last commit.
3. git reset –hard to discard charges either in working directory or in staging area.

   Caution: its irreversible because if there no made.

Branches in GIT HUB:

1. git branch shows my current branch
2. git branch <branch name> create new branch with branch name.
3. *(steric) with branch means this branch currently check out.
4. git check-out <branch name> to switched to that branch.
5. git log –graph –online <branch 1> <branch 2> compare two branches.

   Note: --online to result shoter.

Merging Branches:

1. git merge (branch 1) (branch 2) it will merge into selected branch which I active .
2. git show (commit id) will show the changes between commit and its parent.
3. git branch –d (branch name) to delete a branch. It should be done after merging branch.

   Note: -d stand for del.
4. git remote, get remote –v to get data from and push data to.