Concepts of GIT

1. keeps track of code history

2. Takes "snapshots" of your files

3. you decide when to take a snapshot by making a "commit"

4.you can visit any snapshot at any point of time.

5.you can stage files before committing.

Basic Commands

1. $ git init //Initialize Local Git Respository

2. $ git add <file> //Add file (s) to index (Staging)

3. $ git status //check status of working tree (easily show number of files in staging status)

4. $ git commit //commit changes in Index

5. $ git push // Push to Remote Repository

6. $ git pull //pull latest from remote repository

7. $ git clone //clone repository into a new Directory

=====================================================================================

1. Go to Project folder.

2. git init //create git context default folder

3. git config --global user.name "CodeRepoUnder-Nitish"

4. git config --global user.email "nitishpanotra@outlook.com"

5. git add index.html //added in stage under track that need to commit

6.git status //check current tracked files for commit and untracked files and also tell the name of current branch u r on.

7. git rm --cached index.html // for removing tracked file

8.git add \*.html //will add all html file

9. git add . // (To add all file for commit then use DOT after Add And afterwards ,if there is changes in files ,then again we have to run git add . command otherwise changes will stay un staged.

10. git status

11. git commit // for committing staged changes , it will open VM ,Press "i" to enter in typing mode.

then add comment . To get out of VM : press escape key ,then :wq and enter.

It will give us details of how many files are committed

12. git commit -m "type message to related changes" //it will commit and show details.

=================================================================================

1. touch .gitignore // it will create a git ignore file .Always try from GIT cmd. useful for ignoring dir or files.

2. touch log.txt //will create sample text file ,now mention this file name inside .gitignore file.

3. git add .

4. git status //you will see only file in un stage is .gitignore and log.txt will not be shown and counted.

=================================================================================

Suppose there is separate functionality to developed for example login ,but not in main or master branch. So always create separate branch and merge it later.

1. git branch login //create new branch but not switch to it from current.

2.git checkout login //it will switch to login branch from master/current branch. Before switching

always commit current changes if its there in current branch.

3.touch login.html //it will create new file login.html and change or add what require

4. git add . //it will changes in stage for commit in login branch.

5. git commit -m 'login form'

6. git checkout master // branch switched to Master. but you will not see any changes that made in login branch is going to reflect in master until we merge login branch into master branch.

7. git merge login //again it will open VM to add comment before merging ,we are in master branch when we run merge command.

=====================================================================================

Pushing code to remote repo on distributed environment using git hub. Running below while in master branch.

1. git remote // will list of connected remote repository .if its there.

2. git remote add origin https://github.com/CodeRepo-Nitish/myappsample.git // add your remote repo link to connect

3. git remote // it will show origin as remote repo

4. git push -u origin master // it will push code to remote repo after login pop up to be shown to provide details.

5. touch READ.md

6. git add .

7. git commit -m 'added readme file'

8. git push //as we already connected to remote repo.

=====================================================================================

If someone want to get code from remote repo:

1. Go to new folder and open git cmd

2. git clone paste the github link //it will pull the entire folder from remote repo

3. git pull //to pull changes from remote repo , if there is any changes