

GIT NOTES

LOCAL VERSION CONTROL SYSTEM (VCS)

CENTRALISED VCS

DISTRIBUTED VCS (Example : GIT)

CHECK SUM(Different for every file)

GIT COMMANDS

USES OF GIT COMMANDS

git config user.name name	used to give a user name(name = Nikhil)
git config user.email email	used to give a user email(email = nikhil@gmail.com)
git config user.name	used to display the name of the user
git config user.email	used to display the email of the user
git init	To create a Repository (file in git) ,present working directory(pwd)
git add filename	To add a particular file to staging area
git add --a(git add .)	To add all the file in current directory to the staging area and to start tracking
git add filename	To add filename file to the staging area and to track the filename file(tracking)
git rm --cached filename	Used to untrack a filenameed file(untracking)
git commit -m "message"	To commit the files in staged area with a message (To save the files)
git commit -a -m "message"	To commit directly without staging a file or a folder , but it should be tracked.
-m , -a , --a	These are known as flags in GIT
git status	To view the status of present working directory
git log	To see all the commits with the time and date
rm -rf .git	To remove a File as a Repository, To stop tracking a file or repository
pwd	used to display the current directory on which we are working on
cd	Used to change the directory
ls	To see the list of the folder
mkdir foldername	To create a folder using command line
ls -lart	To see the hidden files in a directory
touch filename	Used to create an empty file
touch .gitignore	Used to create an empty file with .gitignore filename by using we can ignore the files
git clone url ourfilename	Used to clone any remote repository from github

Example : git clone <https://github.com/pandas-dev/pandas.git> pandas

Cloning into 'pandas'...

From example the pandas github repository is being cloned to pandas folder

NOTE : WATCH THE 7TH TUTORIAL OF GIT TO KNOW MORE

NOTES : By adding different files names (or) folder names to the file named as .gitignore we can ignore that files and folders which are not require in project.

GIT ignores the blank files. It doesn't show those files in git status.

Example : log files are not necessary in the project.

*.log (added to gitignore file)	All the files which contain .log will be neglected
/dir/	outer folder named as dir will be ignored or neglected
dir/	All the dir named folders will be neglected
static/dir/	static containing dir folder will be neglected or ignored

These are some of the commands which are added in the .gitignore to neglect or ignore the specific files or folders

GIT COMMANDS

USES OF GIT COMMANDS

git diff	Used to compare staged area file with the present working directory file
git diff --staged	Used to compare previous commit with the present staged area
git --help	To get the help from the git bash and to get commands.
git rm filename	To delete the file with filename

Example: git rm excel.txt, excel.txt will be deleted and this will be added to the staging area.

git mv oldfilename newfilename	To rename the file of oldfilename with the newfilename (move ~rename)
--------------------------------	---

Notes: If everything is in staging area then **"git diff"** command doesn't show any changes in the files and folders

Untracked files cannot be directly committed , it should be added or tracked and then only it can be committed directly

TO RENAME A FILE WATCH **TUTORIAL 10** @CODEWITHHARRY

GIT LOG COMMANDS AND ITS USES (DEEP DIVE)

GIT COMMAND

USES OF COMMANDS

git log	To see all the commits with the time and date
git log -p	Used to display the changes made by the person(committer) with dates and time with all the commits .
git log -p -n	Used to show n number of commits . where n=1,2,3.....n
git log --stat	Gives commits with a short information and all the insertions, deletions etc.
git log --pretty=oneline	Gives all the commits in oneline

git log --pretty=short	Gives all the commits in 3 or 4 lines (displays shortly)
git log --pretty=full	Gives all commits in 4 to 5 lines with author and committer
git log --since=n.years	Gives information about commits which were done in past n years
git log --since=n.months	Gives information about commits which were done in past n months
git log --since=n.weeks	Gives information about commits which were done in past n weeks
git log --since=n.days	Gives information about commits which were done in past n days
git log --pretty=format:"%h--%an"	Gives information related to %h(abbreviated commit hash) and %an(author name)
git log --pretty=format:"%x--%y"	Gives information related to %x and %y

NOTES: For more information of the [git log](#) commands go through the,

GOOGLE SEARCH: git scm useful option for git log

Or

WEBSITE: <https://git-scm.com/docs/git-log>

git commit --amend	Used to change the previously committed commit
git config --global alias.ANOC OCN	Used to give a alias name to the command ANOC → alias name of the command OCN → original command name
git restore filename	Used to remove the all changes made currently and restores all the information as committed in the last commit of the file with filename.
git --version	Used to see the version of the git
git checkout -b branch-name	Which creates a branch with branch-name and move the directory to branch-name.
git checkout master	which moves from branch develop to master branch (EX: branchname=develop)
git checkout develop	Used to switch from another branch to develop named branch
git branch	Used to list all the branches which are created
git branch -v	Gives the list of branches with commit hash and commit message
git branch --merged	Gives the list of merged files
git branch --no-merged	Gives the list of the non-merged files
git branch -d branchname	Used to delete the branch with the branchname with error(merged file)
git branch -D branchname	Used to delete the branch with the branchname without error(non-merged file)

USING GITHUB VIA GIT COMMAND LINE

git remote	To check whether the repo is remote or not
git remote add origin url	To add the repo to github
git remote set-url origin new-url	To add the repo which is pull by someone's account

VIM EDITOR COMMANDS

COMMAND : `git commit -amend`  enter , opens the vim command prompt

VIM COMMANDS

USES OF VIM COMMANDS

i	Uses to edit vim editor
esc	Used to enter into a command line
:wq	To exit from the vim editor

ghp_g7wzNyyLhfWmkl2keOHgnqq1AhYvU21HzUz1

GIT REPLY

Hi Everyone, who is reading this comment ! My name is **Nikhil Pandravesam** pursuing **B.TECH 2nd year** in **Electronics and Communication Engineering** stream . I have taken this course to get knowledge of GIT and GITHUB , this course also helps me to understand about distributed version control system . Expectations are simple , which may help me in dealing with the large project files in my daily live project and in my future IT field . I really love programming alot because of it functionality , which makes the work flow very simple. using Git help me to get hands on experience with the command line. I hope that , git and github are the most important in the evolving IT field.

HOBBIES : PLAYING CHESS , COMPLETING THE WORK ON TIME.

INTERESTS : Coding is my favorite work . yes, I found the person with same interest . The person is my school mate who always guides me about the new technology and future scopes of IT jobs.

I'm an INDIAN , live in Telagana , Hyderabad. The residential is so comfortable that I get all the requiried material in easy and affordable price range.

