Learning Git

15 October 2023   00:07

Git Commands:

pwd	present working directory
ls	list content
git --version	gives the git version
git config --global user.name "name"	declaring the name of the user
git config --global user.email "email.com"	declaring the user email
git config --list	will give out the list of things in git including the git username and email
git status	checking the status of the git repo
git init	initializing git
git add --a or git add .	adding all the files to staging area
git commit -m "some message"	commits all the staged files in to repo
git log	shows all the commits in the repo
git log -p	shows all the commits and the changes done in the previous commits
git log --stat	shows all the changes as a short summary about the previous commits
git log --pretty=oneline git log --pretty=short git log --pretty=full	shows all the changes as a short summary in the previous commits in a single line
git log --since=2.days	changesof the commits in the past 2 days
git-scm website and find the formats for log display	to see the ways in which the git logs can be formatted
q	exit log
git add filename.txt	adding the only file to the staging area instead of all the files
rm -rf .git	the folder will be removed as a git folder
git clone https.....git newname	cloning a git repo from github to the local folder
touch error.log	creating a file named error.log using the linux commands
touch .gitignore	creating a folder called .gitignore
*.log	if this command is written in the gitigniore file, all the files with the .log will be ignored
foldername/	if this command is given in the gitignore file. all the contents in the folder- foldername will be ignored
git diff	compares the things in the working directory and the staging area
git diff --staged	compares the things in the staging area and the last commit
git commit -a -m " message"	this function will skip the staging area and directly the changes will be committed --> but the newly added untracked file will not be committed
git rm	removing the file from the folder using the git command, once a file is removed in this way, git will add this in to stagingarea as well
git mv filename.txt filename_renamed.txt	renaming the file in the folder using git command, once does through this way, git will add it in to staging area
git rm --cached filename.txt	the file is removed from being tracked - this has to be run only after the file is put in the gitignore
rm -rf .git	removing the git repo
git commit -- amend	To do modifications in previous commit message
git restore --staged filename.txt	to unstage a staged file in repo
git restore filename.txt	to discard the changes in the working directory
git checkout -- filename.txt	**this command will restore the file to previously committed state, so any new changes that were made will be removed and lastcommitted file changes are restored**
git checkout -f	this command will restore all the files in the git repo and you'll lose all the changes done to all the files
git remote	a repository on the internet (website) - Github (Github is a website hosting our repo on internet)
git remote add origin <url>	adding a url to the current repo as a remote (Origin is an alias kept for the url being added, It can be added as anything)
git remote -v	it will show all the remote repos available right now from where we are currently pushing and pulling the code from and to
git push -u origin master	Pushing changes from the current repo to remote repo on Github website to master branch
git push origin branch_name	pushing all the changes from the current working directory to the new branch_name in the repo
git push origin branch_name:NewBranch	pushing all the changes to branch in remote while renaming the new branch_name in local to NewBranch name in remote
git config --global alias.alias_name actual_name	replacing the name of the command you use to something more convenient alias name
git checkout -b develop	a new branch is created called 'develop' and the working directory is switched to 'develop'
git checkout develop	Switched to an existing branch - develop (mot creating a new branch)
git branch	shows all the branches available in repo
git merge branch_name	merging the branch_name branch to the master/main branch of the git repo If merge conflicts occur, in VS code, the changes are displayed, you have to accept the incoming changes to finalize the newchanges that you made as the final changes
git branch -v	to view every branch's last commit
git branch --merged	this will display all the branches that are already merged
git branch --no-merged	this will display all the branches that are not yet merged
git branch -d branch_name	Deleting an existing branch --> this will give error if branch is not already merged
git branch -D branch_name	Deleting an existing branch --> no error irrespective of merging
git push -d origin branch_name	deleting the branch in remote repo in Github website (**origin** is the alias of the github website)

Vim Editor commands

i	insert --> command for writing in vim editor
escape button	you'll be able to enter the commands
:wq *enter*	will save the changes in the Vim editor

Generating ssh keys

ssh-keygen -t rsa -b 4096 -C "hashfaq176@gmail.com"	Generating a new ssh key
eval $(ssh-agent -s)	To run the ssh
ssh-add ~/.ssh/id_rsa	to add ssh private key to our ssh agent
tail ~/.ssh/id_rsa.pub	adding ssh key to github account

ssh-keygen -t rsa -b 4096 -C "hashfaq176@gmail.com"	Generating a new ssh key
eval $(ssh-agent -s)	To run the ssh
ssh-add ~/.ssh/id_rsa	to add ssh private key to our ssh agent
tail ~/.ssh/id_rsa.pub	adding ssh key to github account