Important Git commands

* pwd🡪 Print Working Directory 🡪 will print the address of the directory that you are currently working on…
* mkdir 🡪 creates a new folder within the directory that you are currently working on
* cd 🡪current directory 🡪 changes the current directory🡪 example I am in C:/YashWani….and I want to navigate to a directory named “hello” inside it…so I will type cd name….and then it will open the hello directory….--> now the command pwd will print…C:/YashWani/hello
* cd .. 🡪 will bring you out of current folder and take you to its immediate parent folder …(basically ‘back’)
* cd 🡪 will take back to the root folder/super parent
* touch 🡪 used to create a file in a repository 🡪 example…touch master.txt🡪 this will create a text file with the name of “master” in the given directory
* rm 🡪 used to remove a file in linux of local
* git rm 🡪 used to remove a file which is being tracked by git and will automatically stage the changes 🡪 simply commit the changes after using the command 🡪 and then push the changes on the server
* to open a particular file using a particular app…. 🡪 example: to open a txt file using notepad🡪 notepad filename.txt……example: to open a file using VScode 🡪

code .filename.extension…eg. code. Test.cpp

* git add . 🡪used to add the files for tracking…it means that the changes in the files are now tracked by git…these files can now be commited🡪bascially this file is now added to the “staging area”…whatever is part of the working area can be commited
* git commit –m “message put while commiting” 🡪 used for commiting the particular file that you are in
* git log 🡪 it will give you the log of all the commits that you have done
* git log - -pretty=oneline 🡪 will present all the changes made on the repository using one line per change
* git status 🡪 this will show how many files are commited, how many are not commited and bascically the status of all the files in the repository.
* git remote add origin <https://github.com/YashWani28/LearningGitUsingCommands.git> 🡪 this is done to to link the remote repository on the server with the local repo. 🡪 “origin ” is an alias git uses for the master repository on the server
* git remote –v 🡪 shows the links of the repos on github which are linked to the local repository
* git push –u origin master 🡪 used to push the master branch of a local repository on the server
* git push origin 🡪 will push the changes to the repo on the server
* suppose there were 5 commits: c1 -> c2 -> c3 -> c4 -> c5 ….now we want to revert the work done in commit c3 🡪 for this, we use the revert command…this will delete what was done in commit c3 and add that change in a new commit c6…so the commit log will now look like c1 -> c2 -> c3 -> c4 -> c5 -> c6 (this is the revert c3 commit) 🡪 command is:
* git revert 014e3c411d8b0110075557c8d268c2e042261957 🡪 this is done to revert to a previous commit. Instead of deleting the history, this command undo’s the changes done in some previous commit and adds it as a new commit….the address appearing after 🡪 address of the commit to be reversed
* git init🡪 The **git** **init** **command** is used to create a new blank repository. It is used to make an existing project as a **Git** project

Cloning a repo from server to a local machine

* create a repo on server and copy its ssh link
* then in terminal, type git clone *ssh-link*
* then enter the passphrase and it will be cloned on local
* no need of establishing origin as the repo was created on the server itself

SSH security protocol

Just a way to authorize your local computer to manipulate the repos on the server…it’s a mechanism to connect local to the git hub server

1. to create a key pair of public and private keys 🡪 ssh-keygen –t rds –C proprogramming28@gmail.com
2. then enter and re-enter the passphrase for the repository
3. After that 2 files will be created, go to the file having that pub in its title…now copy the ssh key in that file
4. Go to github.com 🡪 click settings 🡪 go to SSH and GFG keys 🡪 now add a new SSH key 🡪 paste the copied key in the description 🡪 click okay and enter the github password
5. Now come back to gitbash and do ssh –T [git@github.com](mailto:git@github.com) 🡪 put yes when prompted 🡪 then enter passphrase which was set up earlier.

Git branching

* git branch *branchname* 🡪 this creates a new called branch1
* git branch –v 🡪 will list the names of all the branches that exist
* git checkout *branchname* 🡪 will open that branch for working in it

how to add a change in a branch

1. create a new branch using git branch branchname
2. open the branch using git checkout branchname
3. then make whatever changes that need to be done
4. After that add those changes for staging using git add .
5. after that the changes should be commited

merging the changed branch

1. suppose want to merge branch1 with master, 🡪 go to master and type git merge branch1
2. then just push the merged master branch on the server by doing git push –u origin master
3. then you may delete the branch1 🡪 git branch –D branch1
4. after this the change needs to be pushed on github by the command 🡪 git push origin - -delete branch1