(BASICS)

Complete Git and Github Tutorial

14/07/2

Git → version control System

Github → Online platform allowing us to host our repositorie

git init → Initialize a git Repository.

git status → Displays the state of the working dir

git add * → Add everything in the project dir (Stage)

git add x.txt → Stage x.txt

git commit -m " " → Add commit to the staged files

git restore --staged → To remove from staged

git log → Displays the commit history of the Repository.

git reset <commit-hash> → Returns the Repository to the state of the particular commit & removes all the commits after the particular

git stash → Will save the dir state without commit to be used or retrieved whenever needed.

git stash → bring back the above saved dir state.

git stash clear → will remove the above saved dir. State

git remote add origin <url> → will connect the github repository with <url> with remote git.

git remote -v → will show all the <url>'s connected.

git push origin <branch-name> → will push the code to the particular branch.

git checkout <branch-name> → changes the initial branch to the new branch.(HEAD changes

git merge <branch-name> → creates a PR to merge two branches: one with the above name another the branch we were working on

git branch <branch-name> → will create a new brach.

git push origin <branch-name> -f → force push - used when the commit is stashed etc.

git fetch -all --prune → fetches all the change from the origin to remote. prune makes sure even the deletions are fetched.

git reset --hard upstream → will reset the origin with the state of upstream. hard deletes the extra commits made from origin.

git pull <branch1> <branch2> → Pulls the change from branch 1 to branch2.

git rebase <commit-hash> → will unstage all the other commits and send the HEAD back to the partiaular commit-hash state.

git commit -m " " --append → will change commit messag
of the recent commit.

git rebase -i <commit-hash> → Open an interactive
environment where we can either
pick or squash the commit. That is
done by changing the pick with the
s to represent squash & combine.

git pull → pulls the origin to remote.

So the concepts covered here are,

Creating a repository.

How to connect with remote

How to commit

How to push

How to pull

How to create a branch

How to merge two branches

How to fetch

How to reset the state of the dir
to any state needed with the
help of rebase

How to combine multiple commits into
single commit

How to change commit message of the most recent commit.

Things that may be useful.

Merge conflicts → Arrises one more than one modifications occur in the same branch. usually in the same file.

Squash commit → This means that we combine all the commits into one single commit.

PR → Pull request

One branch can have only one active PR at a time

fork → Copying the repository to be your own.

HEAD → Pointer pointing the branch