

Class-5

4/2/25

Git Reset:- It is cmd in Git that allows you to move the head and the current branch pointer to a specific commit, effectively "rewinding" or "Resetting" your project's state. It's commonly used to undo changes or to unstage commits.

git reset --hard HEAD~1: used to delete the latest commit along with the changes.

git reset --hard HEAD~3: used to delete the latest 3 commit along with the changes

git reset --soft HEAD~1: used to delete the only commit but not actions/changes/files.

git reset --soft HEAD~3: used to delete only ^{latest 3} commit but not actions/changes.

Git Revert:- The cmd that allows you to undo or reverse the changes made in a previous commit. It creates a new commit that undoes the changes introduced by the specified commit, effectively taking your code back to a previous state without deleting commit history.

git revert commit_id: used to delete particular commit action & add a new commit for the change.

Git Branch:-

A branch represents an independent line of development

The git branch cmd lets you create, list, rename &

delete branches.

The default branch name in git is master.

Allows you to work on different features & changes to your code independently, without affecting the main & other branches.

It's a way to organize & manage your code changes making it easier to collaborate and maintain your project.

git branch: used to see the list of branches

git branch branch-name: to create a branch

git checkout branch-name: To switch one branch to another.

git checkout -b branch-name: used to create & switch a branch at a time.

git branch -m old-branch newbranch: used to rename a branch

git branch -d branchname: To delete a branch

git branch branchname deleted-branch-id: used to get deleted branch id

git branch -D branch-name: To delete branch forcefully.

Git merge :-

git merge is a cmd used in the git version control to combine changes from one branch.

To merge: git merge branch-name.

git commit --amend --no-edit: used to add all commit all tracking files / stating other file will be added.

git update-ref -d HEAD: used to delete all the commits at a time without any changes in a file.

Git Conflict:

→ when ever we merge two branches of same file data we can get conflicts to resolve this @ overcome we want to again modified the data in the file present in master branch & we want to track & commit the file again after modifying.

→ To merge particular commit_id's we want use cherry-pick.

→ git cherry-pick commit_id: used to commit merge a particular commit's into the master.

Stash :-

→ when we add data in the file, for 2nd time without any commit we cannot checkout the branch it will be overwritten, Aborting the new data

in the file. Then we use `git stash` to delete temporarily.

`git stash` :- used to delete the uncommit data temporary.

`git stash apply` :- used to select the temporary

stash will apply for files which present in staging area & tracking files.

stash is applied for only one file at a time.

`Git stash list` :- To see total number of stash lists.

`git stash drop stash-id` :- To delete a particular stash from the list

`git stash clear` :- To remove all the stash in the list.

GitHub :- To store project files & easy to access users.

To connect git with github

`git remote add origin github_depository_link`

`git remote -v` :- To check git is connect with github_depository @ not.

`git push -u origin branch-name` :- To push all

tracking files from git to github

`-u` \Rightarrow forcefully.

while pushing git will use github username & github token as a password.

username: github username

Password: github token

For every changes in a file we need to do

Commit & push the file into github.

`git push -u origin branch1 branch2`: used to

Push multiple branches at a time.

`git push -u origin --all`: used to push all branches at a time.

`git pull origin branch_name`: used to pull the data from github to git

To create a folder we want to give file name/

SSH-keygen = To connect git & github with SSH
generate SSH key in .ssh folder.

`git remote rm origin`: To unlink git & github

`git pull origin`: To pull branch from github to git.

git push -u origin --delete branch_name:

It is used to delete a branch from git

Fetch: check the changes are done @ not

merge: changes \rightarrow local

git Pull = git fetch + git merge.

git fetch origin master: changes are done @ not

git merge origin/master: merge the github to git local.

git clone http_url_github_repo: To clone the repo by developer 2

git branch -a: To see list of branch

pull before push

git ~~re~~log: To ^{See} revert the delete commit

git cherry-pick Commit-id: To revert the delete commit's