Git/Github

`git clone <link>` To clone a existing repo

`git checkout branch1` To go to that branch

`git branch branchname` To create a new branch of a certain name

`git merge branchname` will merge your current branch with other branch

`git branch -d branchname` Deletes a existing branch

`git push origin –delete branchname` Deletes the branch from remote repo

`git fetch` To get the changes between current local Repo and Remote repo without merging them in

`git stash` Saves changes without using git add . so that you can switch branches without loosing your changes

`git stash list` Gives list of all stashes

`git stash apply stash@{2}` this will apply those changes in your current window

`git stash pop` removes latest stash

`git stash clear` Clears all stash

`git pull origin main` Pulls changes from the remote repo to the local repo (git fetch + git merge)

`git status` Gives status of each file U-Untracked git doesnt know about this file . M-Modified some changes have been made to this file . S-Staged ready to commit

`git diff branch1 branch2` gives all differences between the branches

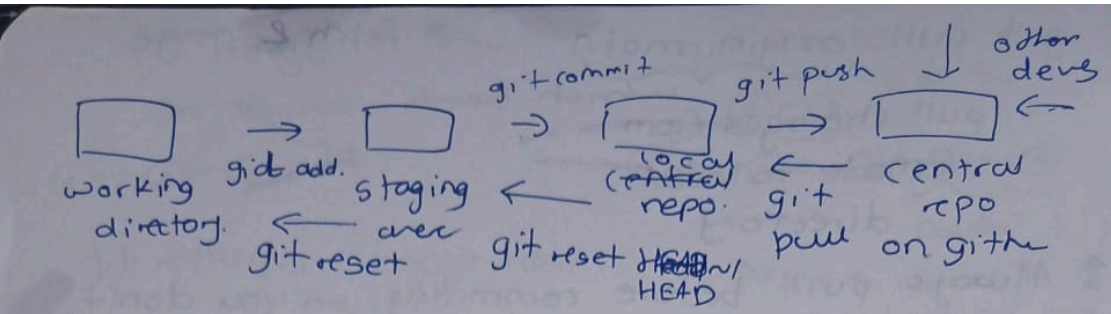
`git rebase branch` instead of merging all changes all the commit history of that merge gets attached to the current branch preserves the commit history

`git reflog` if you wanna go back

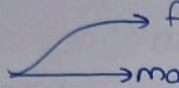
`git reset –soft or –mixed –hard` soft just moves head changes are still there –mixed just unstages those changes they are still there –hard removes those changes entirely

`git revert` creates another commit that does exact opposite of latest commit

Handwritten Notes



git clone <link> → to clone existing repo.



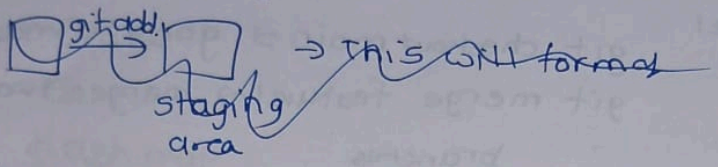
git checkout main → goes to main
git merge feature1 → merges two branches.

git branch name → creates new branch (or use git checkout -b 'hari')
git branch → shows all branches
git branch -d branchname → deletes branch. (local)
git push origin --delete branchname → remove from remote repo
git fetch → download latest changes from remote repo
git stash → save changes without using git add. before you switch branches
git stash save "This is uncommited"
git stash list → List all stashes
↳ helps in finding lade
git stash apply stash@{2} → code
↳ apply that change to current
git stash apply → most recent
git stash pop → remove ladest stash.
git stash clear → clear all stash

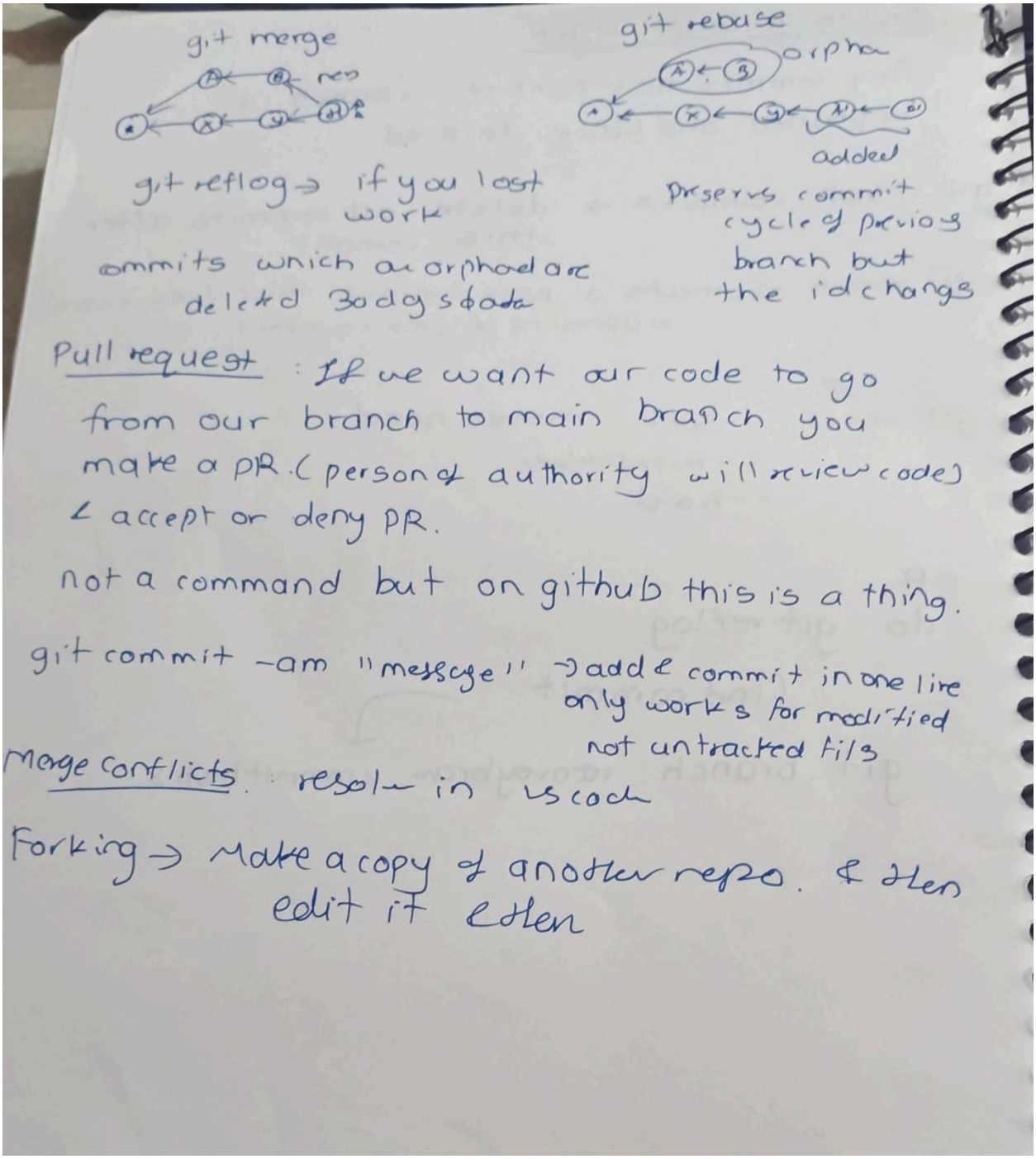
git remote add origin URL → connecting to a remote repo

git pull origin main → fetch & merge
↓ ↑ which branch
pull changes from
remote to local
directory.

* Always pull before commiting so you don't accidentally overwrite smth someone else coded



git add . → all files in current folder
git add -A → all files everywhere
git status → gives status of each file.
untracked → git has no history for this file (new file)
git log → see all commits with their id.
Head → Pointer to latest commit.
git diff → changes that aren't git add . yet
git diff --staged → shows what new things are changed than the last commit.
git diff <commit 1> <commit 2> → diff btwn 2 commits
git diff main branchname → compare 2 branches
+ → added - removed



git reflog → if you lost work

commits which are orphaned or deleted 30 days data

preserves commit cycle of previous branch but the id changes

Pull request : If we want our code to go from our branch to main branch you make a PR. (person of authority will review code) & accept or deny PR.

not a command but on github this is a thing.

git commit -am "message" → add & commit in one line only works for modified not untracked files

Merge conflicts : resolve in vscode

Forking → Make a copy of another repo. & then edit it often

How to go back in git

Any commit not part of a branch is orphaned and hence deleted

git reset commitX → delete all commits after this commit

git revert → creates a new commit that does exact opposite of prev commit

git reset --soft moves head
--mixed (default)
--hard

OR
do git reflog

find commit

git branch recoverybran <commithash>