git --version

git config

set username and email

git config --global user.name "your_name"

git config user.name "your_name" //locally set

git config --global user.email "a@domain.com"

check

git config --list

Change username/email

git config --global user.name "name2"

local, system & global

git config --local --list

git config --global --list

git config --system --list

unset a local or global or system

git config --global --unset user.name

ssh set up

cd ~ for root directory access

ssh-keygen -o -t rsa -C "git email"

code test.pub

try clone through ssh

create a directory and initialize git

mkdir directory_name

ls -a

git init

check git track or untracked files

git status

git staging & unstaging

git add file_name //to take the only file

git add -A to take all files and directories to stage

git add .

git add *.js //only files with .js extension will be in staging

git add **/*.js //directory and subdirectory files will be moved

git restore after add (back to previous code)

git restore file_name / git checkout

git unstaged any file

git rm --cached file_name

check status

difference check after staging

git diff file_name

stage to local repo

git commit -m "Significant_Message"

git commit -am "Significant_Message" to get stag + commit git add . && git commit -m "your_messages"

git log for audit purpose --oneline

uncommit or rollback

git reset --hard HEAD^ //go to next head and the previous head removed

git reset --soft HEAD^// uncommit but stagged

git reset HEAD^ //untracked in local

to know more about any commit history

git log --oneline

git show your_commit_id (7 or full)

version control

git checkout commit_id (7 digit)

git checkout master

Pull And Push and work with remote

git push origin main/git push -u origin main

git pull

branch and merge

git branch //existing branches in local

git branch branch_name // create a new branch in local

git checkout branch_name //move to a new branch

git branch -d branch_name // delete a branch

git checkout -b "branch_name" //create and switch to a new branch

git merge branch_name //after coming to main

fast-forward merge or two-way merge in git through git visualization

three way merge

merge conflict

Contribution

Git fork to contribute other project

Collaboration