

- Git is version control system

Configuring Git

```
git config --global user.name "My Name"
```

```
git config --global user.email "someone@email.com"
```

```
git config --list
```

↓
global
↓
local

- Global user will be use to make changes in any repo of account
- Local user will be use to make changes to specific repo
- To copy git on your local laptop use : clone command

untracked

new files that git doesn't yet track

modified

changed

staged

file is ready to be committed

unmodified

unchanged

change / new file
(modified) (untracked)
↓
add (staged)

- File which is added and ready to commit is staged file
- git add filename : add specific file
- git add . : adds all files
- git commit -m "message" : commit command format

Init Command

init - used to create a new git repo

`git init`

`git remote add origin <- link ->`

`git remote -v` (to verify remote)

`git branch` (to check branch)

`git branch -M main` (to rename branch)

`git push origin main`

git push -u origin main
↓
set upstream

- Git init : to initialize git
- Git remote add origin <link> : to set the default repo on which you will be working
- Default branch on repo is main(newly used by github)/master
- If you are continuously making changes on specific branch you can fix the same branch using command: `git push -u origin main/branchname` (u : upstream) after this you can use only `git push`

Branch Commands

`git branch` (to check branch)

`git branch -M main` (to rename branch)

`git checkout <- branch name ->` (to navigate)

`git checkout -b <- new branch name ->` (to create new branch)

`git branch -d <- branch name ->` (to delete branch)

Merging Code



Way 1

`git diff <- branch name->` (to compare commits, branches, files & more)

`git merge <- branch name->` (to merge 2 branches)

Way 2

Create a PR

(Pull Request)

Ex : you are on feature1 branch and want to compare this branch with main
then run command : `git diff main`
Same for merge : `git merge main`

Merging can be done by PR(Pull request)

Pull Command

`git pull origin main`

used to fetch and download content from a remote repo and immediately update the local repo to match that content.

Undoing Changes

Case 1 : staged changes (add →)

```
git reset <- file name ->
```

```
git reset
```

Case 2 : committed changes (for one commit)

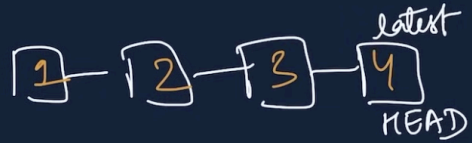
```
git reset HEAD~1
```

Case 3 : committed changes (for many commits)

```
git reset <- commit hash ->
```

```
git reset --hard <- commit hash ->
```

HEAD~1



Hard to reset from both vscode and git

Fork

A fork is a new repository that shares code and visibility settings with the original “upstream” repository.

Fork is a rough copy.