Git and Github

27 October 2024 02:45

Git

Version Control system is a tool that helps to track change in code

Git is a Version control system. It is:
Popular
Free & open source
Fast & scalable

It help
1. Trck the history
2. Collaborate

GitHub

Website that allows developers to store and manage their code using Git

https://github.com

GitHub Account

- Create a new repository
- Make our first commit

Add --> commit changes

Setting up Git

Visual studio code
Windows (Git Bash)
Mac (Terminal)

git --version
ls
psw

Configuring Git

```
git config --global user.name "My Name"
git config --global user.email "someone@email.com"
git config --list
```

Global - changes in system (all)
Local - changes in specific repository/project

Clone & Status

Clone - cloning repository on our local machine

(open github -> open repository -> open project/file -> click **code** -> select **HTTP** -> copy link)

git clone link

(remote [file in github] copy --> local [file in laptop/pc])

Cd -> change directory

cd file name (we can move inside that fle)
(ls -a -> list all files
Status - displays the state of the code

git status

- **untracked**
  new files that git doesn't yet track
- **modified**
  Changed
- **staged**
  File is ready to be committed
- **unmodified**
  unchanged

Change (modified) / newfile (untracked) --> add (staged) --> commit (modified)(unchange) (repeatation)

Add & Commit

add - adds new or changed files in your working directory to the git staging area.

git add <file name>

git add . --> all filed add using this command

commit - it is the record of change

git commit -m "some message"

e.g.: git commit -m "add new paragraph"

Push command

push - upload local(laptop/pc) repository content to remote(github) repository

git push origin main
(first time they ask permission --> click **Allow** --> display screen (Authorized GitHub for VS Code) --> click **Authorize Visual-Studio-Code** --> open visual studio code --> click open

Init Command

init - used to create a new git repository

git init

git remote add origin <link>
git remote -v *(to verify remote)*
git branch *(to check branch)*
(if we check branch we will get -master branch(default branch, not main branch))

git branch -m *(to rename branch)*
git push origin main
(we use **git push -u origin main ,** we are working in same file long time then we can use **git push** only (avoid repeatation **git push origin main**)

cd .. ---> to come outside directory
mkdir ---> to make new directory

WorkFlow

Local Git

1.GitHub rep
2.Clone
3.Changes
4.add
5.commit
6.push

Git Branches

In Git, a branch is a new/separate version of the main repository.

Merge branches ----> add/compine two branches (same code)

Branch Commands

git branch (to check branch)

git branch -M main (here main is new branch name) (to rename branch)

git checkout <- brnch 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)*

Pull Request

It lets tell others about changes you've pushed to a branch in a repository on GitHub.

our bracnch changed -----> mrge to --> main branch(pr review)

Open **github** ---> click **repository** ---> open **file** ---> click **commit pull request** ---> type **add new feature(file) -->** click **create pull request--->** type messge (for company senior) ---> click **merge pull request** --> click **confirm merge**

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

Resolving Merge conflicts

An event that takes place when Git is unable to automatically resolve differences in the code between two commits.

1. PR (Pull Request)
2. Git Merge

Git Merge

git merge main

Undoing Changes

Case1 : staged changes
git reset <- file name ->
git reset

Case 2 : commited changes (for one commit) **(undo one)**
git reset HEAD~ 1

Case 3 : commited changes (for many commits) (undo multiple change)

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

git log *(display all commit(change))*

git reset --hard

Fork

A fork is a new repository that shares code and visibility settings with the original "upstrem" repository

Fork is a rough copy.

Open GitHub ---> serch repository ----> click Fork ---> Create fork

We can make commits on our copied repo
If we want to merge changes in that base repo : create pull request(click **new pull request**)