Masterclass on

Git & GitHub

Topics to be covered

- Git & GitHub
- Git Setup and Configuration
- Git Initialization, Adding changes
- How to perform push, pull operation
- Github Account
- How to connect remote and local repository
- Gitignore file
- Branching and merging

Theory of Git & GitHub

Why do we need git?

- version control (Git, GitLab, Beanstalk, PerForce, Bitbucket)
- by git we clone repository and push changes
- change tracking
- collaborate on project

Theory of Git & GitHub

Why do we need github?

- Remote folder/repository is hosted on GitHub
- Local clone of repository
- It is owned by Microsoft

Theory of Git & GitHub

Difference Between Git & GitHub

Git

1. Tool/Software
2. Change control, version control
3. Provides CLI (Command Line Interface Tool)
4. Installed and maintained locally

GitHub

1. Service
2. Host git repository/folder/source code
3. Provides GUI (Graphical User Interface)
4. Maintained on cloud/web

Git Installation & Setup

Download: https://git-scm.com/downloads

- **git --version (On command line)**

- **To set your Git username, type this in your terminal:**

  git config --global user.name "Your Name"
- **To confirm that you have set your Git username correctly, type this:**

  git config --global user.name
- **To set your Git email, type this in your terminal:**

  git config --global user.email "youremail@gmail.com"
- **To confirm that you have set your Git email correctly, type this:**

  git config --global user.email

Working with Git

Working Directory	Staging Area/Index	Local Repository	Remote Repository
mkdir myfolder for making new folder	**git add filename**	**git commit -m"file is added"**	
cd myfolder for changing directory	**git add -A**	**git commit -am "flies added"**	
ls for all folder and files including hidden	**git add .**	**git add . && git commit -m"files are added"**	
ls -a for all folder and files including hidden	**git add *.php**	**git log**	
touch myfile.php for all folder and files including hidden	**git add **/*.js**		
git init **git status**	**git diff**		

Working with Git

Working Directory

Tracked and Untracked Floder & Files

git init ⟶ **a .git folder will be created**

for initializing git

git status

for checking tracked or untracked

Working with Git

Staging Area

git add filename

for adding in the staging area

git add -A

stage all changed files in the directory & sub folders

git add .

sstage all changed files in the folder/directory but no sub directories

git add *.php

directory wildcard

git add **/*.js

directory & sub directory wildcard

git diff

checking difference

git rm --cached filename

for unstaging a specific file

git restore filename

for going back to the previous state or discard the modification

Working with Git

Local Repository

(commit)

git commit -m"file is added"

for adding in the local repository

message should be clear to understand

git commit -am "flies added"

For doing both, staging and commiting

or

git add . && git commit -m"files added"

For doing both, staging and commiting

git log

To see the commit history

git log --oneline

To see the commit history in oneline

Working with Git

Local Repository

(uncommit)

git reset --soft HAED^

only reset the commit and but will reamin in the staging area

git reset --hard HEAD^

delete the recent commit and takes to the previous commit state

git reset HEAD^

reset the commit and also remove from the staging area

Working with .gitignore

touch .gitignore (for creating .gitignore file)

myfile.txt
.env
*.txt
!main.txt
myfile?.txt
vendors/
temp/
node_modules/

Working with GitHub

Must have a github account

Sign in from here:

https://github.com/signup

Working with GitHub

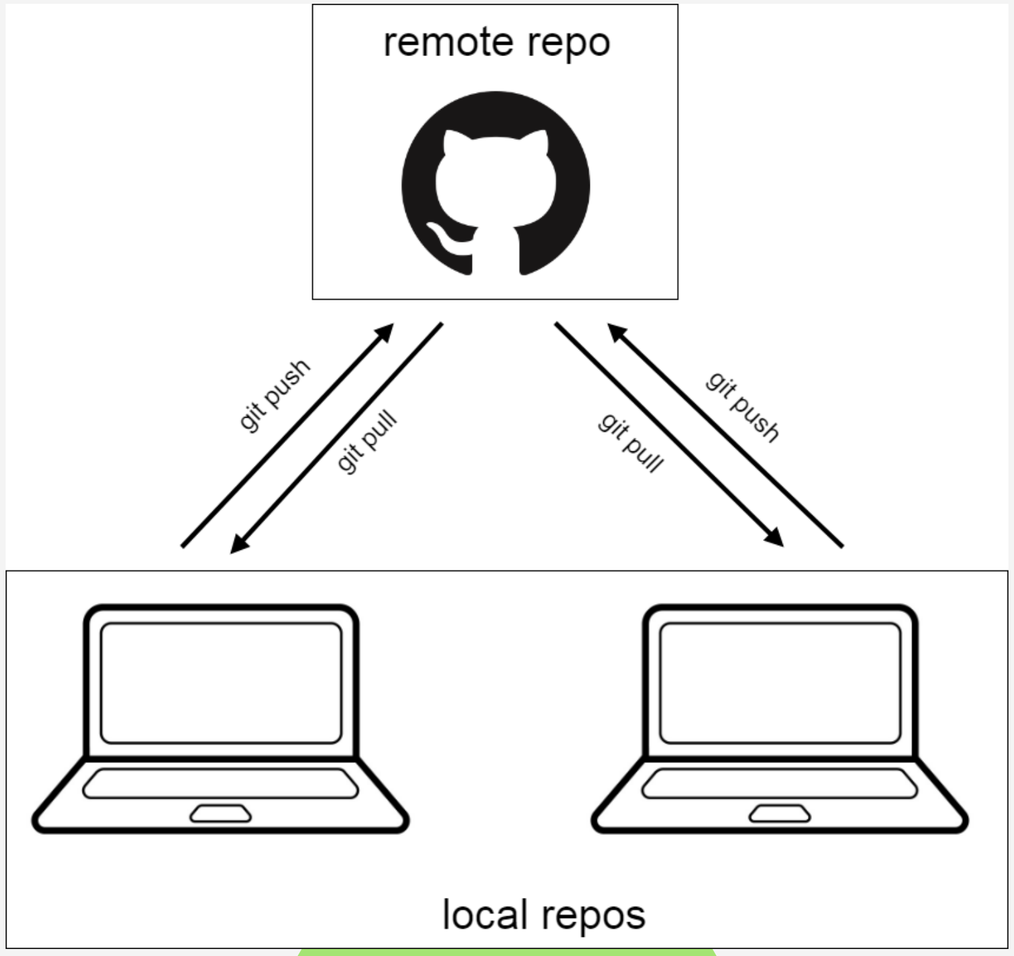

Image Source: google.com

Connection with local repository & Remote repository

git remote

to check the connection

git remote -v

shows the details

git remote add name <YOUR_REMOTE_URL>

git remote add origin <YOUR_REMOTE_URL>

Connection with local repository & Remote repository

Git Push & Git Pull

git push -u origin main

to push the local changes

git pull

to get the updated changes from remote repository

Connection with local repository & Remote repository

Git Branch

git branch

to show all branch

git branch test

to create a new branch

git checkout test

to move new branch

} **git checkout -b "test"**

to create and swtich

git branch -d test

to delete the created branch (must be switched to main branch)

Connection with local repository & Remote repository

Git merge

git merge test

to merge other branch with main branch (must be in the specified branch)

Thank you