**Introduction to Git:**

**🡪**Git stands for Global Information Tracker.

🡪It is also called as VCS (Version Control System).

🡪It is used to tracking changes in the source code, enabling multiple developers to work together.

🡪It is a central repository using which we can manage our project source code.

🡪It maintains all modifications happening to a specific file.

* Because of versions trouble shooting & fixing bugs is easy.
* If something goes wrong in current version we can rollback to previous version.

🡪It records the modifications, when modified(timestamp) and why it is modified.

🡪Git is distributed VCS.

**Git used for two reasons:**

1.VCS- Version Control System

2.Colloboration

**Why do we use git?**

* To maintain multiple versions of same file.

**Functionalities of VCS:**

1.It allows multiple developers to develop the code simultaneously.

2.It doesn’t allow overwriting each other changes.

3.It maintains a history of every version.

4.Git is fast when compare to other Version Controlling tools.

5.Multiple developers can easily collaborate and work on same project.

6.It also works as backing up our project code.

**Types of VCS:**

There are 2 types of VCS:

1.CVCS: Centralized version control system------>SVN-Sub version control system.

2.DVCS: Distributed VCS/Decentralized VCS----->Git

**SVN-Centralized Server**

It consists of current version data.

Each and every developer needs to connect to this server and then needs for develop their codes.

**Disadvantages of SVN:**

1.Servers can be accessed by anyone-------- any one can copy the code.

2.If server is destroyed, everything will be lost.

**Decentralized/Distributed VCS:** Github----->Server

* Each and every developer need not want to connect to the github (server).
* Create a repository in github ---remote repository.
* In github we have an option of opting the code-----cloning;

**Cloning:** Using cloning, we can bring our codes from remote repository(server—Github) to local repository (own laptops/systems)

Command for cloning

git clone URL (each project will have separate URL)

**Git:**

🡪Git is a client/server architecture.

🡪Git bash- client-----

🡪github- server---------====mirrored copy

**Repository:**

🡪Group of project files to store in one single area.

🡪Each project has one repository.

🡪Github has many number of repository.

**Local Repository:**

🡪Getting the remote repository to our local repository (our own laptops/systems).

**Remote Repository:**

🡪Hosted on a server or cloud platform.

🡪Synchronizes with local repositories via push/pull.

**Github: Server**

🡪**GitHub** is a **cloud-based platform** that hosts Git repositories and provides tools for version control, collaboration, and project management.

🡪It allows developers to store, manage, and share code, making it easier to collaborate on software projects.

Push: We will send the files to the remote repository.

Pull: We will bring/extract the files from remote repository to our local repository.

**Fork:**

🡪The projects are copied from one’s github account to others github account.

**Ex:** Santosh’s project will be copied to Durga’s repository by using fork.

**Git will follow two types of protocols:**

1. HTTPS-Hypertext Transfers Protocol Secure
2. SSH- Secure Shell

**1.HTTPS:**

In Git, **HTTPS (Hypertext Transfer Protocol Secure)** is one of the protocols used to securely communicate with remote repositories, like those hosted on **GitHub, GitLab, or Bitbucket**.

**2.SSH:**

In Git, **SSH (Secure Shell)** is a protocol used to securely communicate with remote repositories, like those on GitHub, GitLab, or Bitbucket. It provides an alternative to HTTPS for authentication and data transfer.

**Why Use SSH in Git?**

* **Security** – SSH encrypts data and uses key-based authentication, which is more secure than password-based login.
* **Convenience** – Once set up, SSH allows you to interact with remote repositories **without repeatedly entering a password** or personal access token (PAT).
* **Speed** – SSH is often faster than HTTPS, especially for large repositories.

Open powershell commands:

git --version🡪to check our git version

git config🡪used to check git package installed properly or not

git config --global --list

git config --global user.name “laxmi”

git config --global user.email [“puropaleprasanna1234@gmail.com”](mailto:\“puropaleprasanna1234@gmail.com\”)

Visual code Commands:

git --version

git int

git add hello.py

git commit -m “myfirstcommit”

git status

git branch -m main

git remote add origin http:link

git push -u origin main

Git Commands:

1. git init🡪to initialise git repository
2. git status🡪to know all necessary & current information of the current active branch
3. git pull🡪to get latest version of the repository
4. git push🡪sends local commits to the remote repository
5. git add🡪to add all new changes in the staging area
6. git commit🡪record the changes made to the file to a local repository
7. git branch🡪to know which one is the current active branch & how many branches are there
8. git merge🡪integrate branches together.git merge combine changes from one branch to another branch
9. git checkout🡪change the branch