**Q. What is Git?**

* Git is a **distribued** **version control system** that helps developers track and manage changes in their code. It allows:  
  ✅ Keeping a history of changes  
  ✅ Working on different features using branches  
  ✅ Undoing mistakes easily  
  ✅ Collaborating with multiple developers
* **Git is a local tool**, meaning it runs on your computer and doesn't require an internet connection.
* Git provides both🡪

- GIT GUI(Graphical User Interface)

- GIT BASH(Command Line)

* Developed by Linus Torvalds

**Q. What is Version Control ?**

Version control is a system that helps track changes to files over time. It allows multiple people to work on a project simultaneously, keeps a history of modifications, and enables users to revert to previous versions if needed.

**Types of Version Control Systems 🡪**

1. **Local Version Control (LVCS)**
   * Keeps versions of files on a local computer.
   * Example: Simple backups or local history in text editors.
2. **Centralized Version Control (CVCS)**
   * A single server stores all versions of files, and users check out files to work on.
   * Example: SVN (Subversion).
3. **Distributed Version Control (DVCS)**
   * Each user has a complete copy of the repository, including the full history.
   * Example: Git, Mercurial.

**Why Use Version Control?**

* **Tracks Changes** – Keeps a history of modifications.
* **Collaboration** – Multiple people can work on the same project.
* **Backup & Recovery** – Restores previous versions if needed.
* **Branching & Merging** – Allows working on different features separately and then merging them.

**Q. What is GitHub?**

GitHub is an **online platform** that hosts Git repositories and makes collaboration easier. It provides:  
✅ Remote storage for Git repositories  
✅ Tools for collaboration (issues, pull requests, code reviews)  
✅ Project management features  
✅ Integration with CI/CD pipelines and other tools

🔹 **GitHub is NOT Git.** It is a service that uses Git. Other similar services include **GitLab** and **Bitbucket**.

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Git | Git Bash | GitHub |
| What it is | Version control system | Command-line tool for Git | Online platform for Git repositories |
| Works on | Local computer | Windows (for running Git commands) | Online (remote storage) |
| Internet Required? | ❌ No | ❌ No | ✅ Yes |
| Purpose | Tracks code changes | Helps use Git on Windows | Stores & shares repositories |

Steps include in Git’s way of tracking and saving changes in the project 🡪

1. Modify: When you edit or change files in your project, they are considered modified but not yet saved in Git.
2. Staging: Staging means preparing modified files for a commit. You add the files to the staging area, telling Git that these changes should be included in the next commit.
3. Commit: A commit is a snapshot of the project at a specific point. Once committed, the changes are saved in Git’s history.

Commands in Git🡪

GIT IMPORTANT TERMS🡪

1. **Repository (Repo)** – Storage for your project files and history.
2. **Commit** – A snapshot of changes, like saving progress.
3. **Branch** – A separate line of development. (e.g., main, feature-branch)
4. **Merge** – Combines changes from one branch to another.
5. **Master**– Primary branch of all repos.
6. **Clone** – Copies a remote repository to your local machine.
7. **Pull** – Adding the changes to the master branch.
8. **Push** – Sends local commits to the remote repository.
9. **Remote** – The online version of your repo (e.g., GitHub, GitLab).
10. **Staging Area** – Temporary area where files are prepared before committing.
11. **Checkout** – Switches between branches or commits.
12. **Rebase** – Moves changes from one branch onto another cleanly.
13. **Fetch** – Downloading and copying file to your workstation.
14. **Index**– Files that have been modified will be staged withing index until files are ready to commit.
15. **Tag**– Most imp. Portions of project history are marked using tags.
16. **Origin**– Conventional name for the primary version of repo.
17. **Upstream**– This can be considered where you push your Git changes.