GIT:

Git is a popular version control system. It was created by Linus Torvalds in 2005, and has been maintained by Junio Hamano since then.

It is used for:

* Tracking code changes
* Tracking who made changes.
* Coding collaboration

How GIT WORKS:

* Create a "repository" (project) with a git hosting tool (like abc project)
* Copy (or clone) the repository to your local machine
* Add a file to your local repo and "commit" (save) the changes
* "Push" your changes to your main branch
* Make a change to your file with a git hosting tool and commit
* "Pull" the changes to your local machine
* Create a "branch" (version), make a change, commit the change
* Open a "pull request" (propose changes to the main branch)
* "Merge" your branch to the main branch

GIT COMMANDS:

Clone a Repository: git clone <repository-url>

Create a Branch: git branch <branch-name>

Switch to a Branch: git checkout <branch-name>

Add Changes to Staging: git add <file(s)>

Commit Changes: git commit -m "commit message"

Push Changes to Remote Repository: git push origin <branch-name>

Pull Changes from Remote Repository: git pull origin <branch-name>

GITHUB

GitHub is a web-based platform that uses Git for version control and collaboration.

Git is a distributed version control system that allows multiple developers to work on a project simultaneously.

Uses of GitHub:

Version Control: GitHub allows developers to track changes in their codebase over time.

Collaboration: Multiple developers can work on the same project simultaneously. GitHub provides tools for managing branches, merging changes, and resolving conflicts.

Code Hosting: GitHub serves as a remote repository for hosting your code. This allows you to access your code from anywhere and provides a backup in case of local failures.

Issue Tracking: GitHub has built-in issue tracking features, enabling teams to manage tasks, track bugs, and discuss project-related topics.

Code Review: Pull Requests on GitHub facilitate code review, enabling team members to review changes before they are merged into the main codebase.

Continuous Integration: GitHub can be integrated with CI/CD (Continuous Integration/Continuous Deployment) tools to automate the testing and deployment processes.

Documentation: GitHub provides a space for project documentation, README files, and wikis to help users understand and contribute to the project.

GIT CLONE:

Git clone is used to create a copy of a repository (a collection of files) on your computer. This can be useful if you want to work on the same project but don't have access to the original source code.

GIT BRANCH:

The Git branch command is used to keep track of the changes that have been made to a file or project. Use the Git branch command to see which commits are associated with which branch. This is useful for reviewing the changes that have been made to a project, and for knowing which commits are responsible for certain features or fixes.

GIT COMMIT:

The Git commit command is used to record a change you've made to a file and then store that change in the repository. Whenever someone else needs to access the changes you've made, they can use the Git commit command to retrieve them.

GIT MERGE:

Git merge is used to combine changes made in different branches into one single branch - this makes it easier for developers working on the same project to see updates quickly. It also helps avoid unintentional errors when multiple people are working on the same codebase simultaneously.

GIT STATUS:

The 'git status' command produces a list of files in the current working directory, along with information about the status of each file. Understanding the output of this command can help you track the progress of your files and make changes more effectively.

GIT PUSH

Git push commands allow you to send changes you have made to a repository to other people who are also working on that repository. The most common use for this is when you want to share changes you have made with a team of collaborators. With Git push, everyone on the team can receive the same version of the project at the same time.

GIT PULL:

The Git pull command allows you to fetch changes made by other developers on a project. When you use the Git pull command, Git will check out the latest changes from the remote repository and merge them into your local copy. By default, the Git pull command will merge all changes made to the current branch.

GIT INIT:

Git init is a command used to initialize a new Git repository. It is written in the shell, and typically runs from the root of a project directory.

GIT FETCH:

Git fetch is used to retrieve files from a Git repository. This can be useful if you want to get a copy of the latest version of a file without having to clone or checkout the repository first.