**GIT:**

Git is a distributed version control system that plays a fundamental role in modern software development. Git provides a robust and flexible mechanism for tracking changes in source code throughout the development lifecycle.

**GitHub:**

GitHub is a web-based platform for version control and collaboration using Git. It provides a hosting service for software development projects and facilitates collaboration among developers. GitHub is widely used for managing and sharing code, tracking changes, and collaborating on projects.

**Repository:**

In simple terms, a repository (repo) is like a folder or storage space where you keep all the files for your project. It not only stores your current work but also remembers every change you've ever made, acting like a detailed history log.

**Key Differences:**

1. **Local vs. Web-Based:** Git is a command-line tool installed and used locally on a developer's machine, managing the version control of a project. GitHub, on the other hand, is a web-based platform that hosts Git repositories and adds collaboration features.
2. **Version Control vs. Collaboration:** Git focuses on version control and managing the local repository, while GitHub provides a platform for collaboration, code hosting, and team collaboration.
3. **Repository Hosting:** Git itself does not provide a centralized server for hosting repositories. GitHub, on the other hand, is a remote server where developers can host their Git repositories.

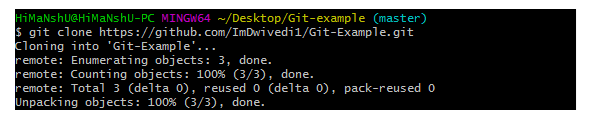
**GIT COMMANDS:**

1)Git clone command

This command is used to make a copy of a repository from an existing URL. If I want a local copy of my repository from GitHub, this command allows creating a local copy of that repository on your local directory from the repository URL.

**Syntax**

1. $ git clone URL



2) Git add command

This command is used to add one or more files to staging (Index) area.

**Syntax**

To add one file

1. $ git add Filename

To add more than one file

1. $ git add\*

Git Commands

3) Git commit command

Commit command is used in two scenarios. They are as follows.

**Git commit -m**

This command changes the head. It records or snapshots the file permanently in the version history with a message.

**Syntax**

1. $ git commit -m " Commit Message"

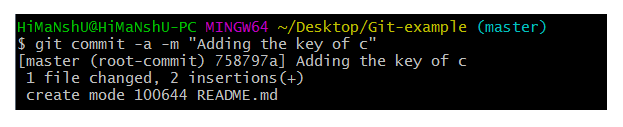
**Git commit -a**

ADVERTISEMENT

This command commits any files added in the repository with git add and also commits any files you've changed since then.

**Syntax**

1. $ git commit -a

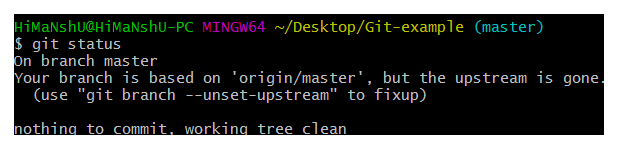


4) Git status command

The status command is used to display the state of the working directory and the staging area. It allows you to see which changes have been staged, which haven't, and which files aren?t being tracked by Git. It does not show you any information about the committed project history. For this, you need to use the git log. It also lists the files that you've changed and those you still need to add or commit.

**Syntax**

1. $ git status

Top of Form

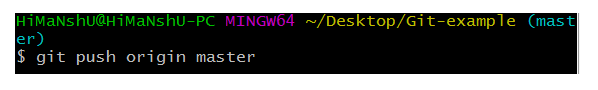
### 5) Git push Command

It is used to upload local repository content to a remote repository. Pushing is an act of transfer commits from your local repository to a remote repo.

ADVERTISEMENT

**Syntax**

1. $ git push [variable name] master

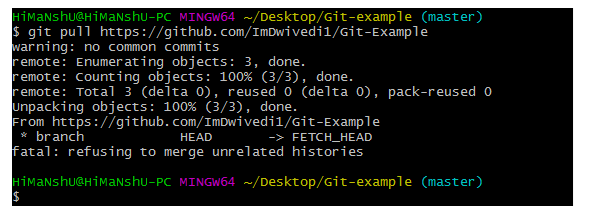


 6)Git pull command

Pull command is used to receive data from GitHub. It fetches and merges changes on the remote server to your working directory.

**Syntax**

1. $ git pull URL



7)Git log Command

This command is used to check the commit history.

**Syntax**

1. $ git log

