**Git**

Git is a actively maintained open source project.

The raw performance characteristics of Git are very strong when compared to many alternatives. Committing new changes, branching, merging and comparing past versions are all optimized for performance.

One of Git's key design objectives is flexibility. Git is flexible in several respects: in support for various kinds of nonlinear development workflows, in its efficiency in both small and large projects and in its compatibility with many existing systems and protocols.

Git has the functionality, performance, security and flexibility that most teams and individual developers need.

Everything in Git is checksummed before it is stored and is then referred to by that checksum. This means it’s impossible to change the contents of any file or directory without Git knowing about it.

Git has three main states that your files can reside in: modified, staged, and committed:

1. **Modified**: means that you have changed the file but have not committed it to your database yet.

2. **Staged**: means that you have marked a modified file in its current version to go into your next commit snapshot.

3. **Committed**: means that the data is safely stored in your local database.

==================================================================================

**GitHub**

GitHub is a website and cloud-based service that helps developers store and manage their code, as well as track and control changes to their code.

GitHub is an online software development platform. It's used for storing, tracking, and collaborating on software projects.

GitHub facilitates social coding by providing a hosting service and web interface for the Git code repository, as well as management tools for collaboration.

The developer platform can be thought of as a social networking site for software developers.

The following are some important terms GitHub developers use:

**1. git clone**: This command is used for downloading the latest version of a remote project and copying it to the selected location on the local machine.

**2. git fetch**: This Git command will get all the updates from the remote repository, including new branches.

**3. git checkout**: You can use the checkout command to switch the branch that you are currently working on.

**4. git init**: This is the command you need to use if you want to start a new empty repository or to reinitialize an existing one in the project root. It will create a .git directory with its subdirectories.

**5. git commit**: This one is probably the most used Git command. After changes are done locally, you can save them by “committing” them.

**6. git push**: Git push will push the locally committed changes to the remote branch.

**7. git diff**: You can use this command to see the unstaged changes on the current branch.

**8. git pull**: Using git pull will fetch all the changes from the remote repository and merge any remote changes in the current local branch.

**9. git add**: This is the command you need to use to stage changed files. You can stage individual files.

**10. git branch**: Using git branch will list all the branches of the repository.

==================================================================================

**Difference between Git and Github:**

Git is a version control system that allows developers to track changes in their code. GitHub is a web-based hosting service for git repositories.

In simple terms, you can use git without Github, but you cannot use GitHub without Git.

