Git

Git is a distributed version control system that allows multiple people to collaborate on the same codebase efficiently. It tracks changes in your codebase, allowing you to:

1. Keep Track of Changes: Git records every change you make, along with who made it and when.

2. Branching and Merging: It enables you to create different branches of your code for different features or experiments, and then merge them back together.

3. Local and Remote Repositories: You have a local copy of your code on your computer, and you can also have a remote copy on a server or a service like GitHub.

4. Snapshot-based: Git stores data as a series of snapshots, which allows for lightning-fast operations.

5. Decentralized: Unlike centralized version control systems, every developer's working copy of the code is also a repository that can contain the full history of all changes.

GitHub

GitHub is a web-based platform built around Git. It provides a user-friendly interface and a set of tools on top of Git to facilitate collaboration. Here are some key features:

1. Remote Hosting: GitHub allows you to host your Git repositories online, making it easy to share your code with others.

2. Collaboration: Multiple people can work on the same project simultaneously. GitHub offers tools for code review, issue tracking, and project management.

3. Pull Requests: A pull request is a way to propose changes to a repository. Other developers can review the proposed changes and discuss them before they are merged.

4. Issues and Projects: GitHub provides tools for tracking bugs, feature requests, and other tasks. It also offers project boards for managing workflows.

5. Community and Social Coding: It's a hub for open source development, allowing developers from all over the world to collaborate on projects.

Basic Workflow:

1. Initialize a Repository: You start by creating a new Git repository or cloning an existing one.

2. Make Changes: You modify files in your local repository.

3. Stage Changes: You choose which changes to include in the next commit by "staging" them.

4. Commit Changes: You create a snapshot of the staged changes along with a descriptive message.

5. Push to Remote Repository (GitHub): If you have a remote repository on GitHub, you can push your local changes to it.

6. Collaborate and Review: If working with others, you can create branches, make changes, and open pull requests for review.

7. Merge Changes: After review, changes can be merged back into the main branch.

8. Pull Changes: If others have made changes to the repository, you can pull those changes to your local copy.

This is a basic overview, and there's a lot more to learn about Git and GitHub. As you delve deeper, you'll discover advanced features and best practices that can greatly enhance your development workflow.

Commands :

git --version

Configure GIT on local machine :

git config --global user.name "______"

git config --global user.email "______"

git config --list

Clone and Status :

```
git clone < - - - Repository Link - - - >
```

```
git status
```

Untracked Files : new files that GIT doesn’t yet track.

Modified : Changed

Unmodified : Unchanged

Staged : File is ready to be Committed

Add : adds new files or changed files in your working directory to the GIT staging area.

```
git add < - - - File name - - - >    // Add any particular file in a current directory

git add .         // add all the new files and modifications in a current directory

git commit -m  “ - - - some message - - - ”

git push origin main
```