Introduction to Git

Git is a distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Created by Linus Torvalds in 2005, Git allows multiple people to work on a project simultaneously without interfering with each other's work. Here are some key features and concepts related to Git:

Key Features:

1. Distributed Version Control: Each developer has a local copy of the entire project history, making it fast and efficient.
2. Branching and Merging: Git supports branching and merging, allowing multiple lines of development to run in parallel and be integrated later.
3. Data Integrity: Git stores data as snapshots of the project over time, ensuring that the history is preserved accurately.
4. Lightweight and Fast: Git is designed to be fast, with most operations performed locally.

Core Concepts:

1. Repository (Repo): A Git repository is the database where your project files and the entire revision history are stored.
2. Commit: A commit is a snapshot of your project's files at a given point in time. Each commit has a unique ID and can have a message describing what was changed.
3. Branch: A branch is a parallel version of your project. By default, Git has a main branch (often called `main` or `master`), but you can create new branches to work on different features or fixes.
4. Merge: Merging is the process of integrating changes from different branches into a single branch.
5. Clone: Cloning creates a copy of a Git repository, allowing you to work on a local version of a

remote project.

6. Pull: Pulling updates your local repository with changes from a remote repository.

7. Push: Pushing updates a remote repository with changes from your local repository.

Basic Commands:

- git init: Initializes a new Git repository.

- git clone [url]: Clones a repository from a URL.

- git add [file]: Stages a file for a commit.

- git commit -m "message": Commits the staged changes with a message.

- git status: Shows the current status of your working directory and staging area.

- git push: Pushes changes to a remote repository.

- git pull: Fetches and merges changes from a remote repository.

Git is widely used in software development due to its powerful features and flexibility, and it serves as the foundation for many collaborative development workflows.