What Is Git?

Git is a **specific open-source version control system** created by Linus Torvalds in 2005.

Specifically, Git is a **distributed version control system**, which means that the entire codebase and history is available on every developer's computer, which allows for easy branching and merging.

Why Git?

GitHub's interface is user-friendly enough so even novice coders can take advantage of Git. Without GitHub, using Git generally requires a bit more technical savvy and use of the command line.

GitHub is so user-friendly, though, that some people even use GitHub to manage other types of projects.

Additionally, anyone can sign up and host a public code repository for free, which makes GitHub especially popular with open-source projects.

Git Commands

i. Git init - Initializes a new Git repository. If you want to place a project under revision control, this is the first command you need to learn.
ii. Git clone - Creates a copy of an existing Git repository. Cloning is the most common way for developers to obtain a working copy of a central repository.
iii. Get checkout - In addition to checking out old commits and old file revisions, git checkout is also the means to navigate existing branches. Combined with the basic Git commands, it's a way to work on a particular line of development.
iv. Git add - Moves changes from the working directory to the staging area. This gives you the opportunity to prepare a snapshot before committing it to the official history.
v. Git commit - Takes the staged snapshot and commits it to the project history. Combined with git add, this defines the basic workflow for all Git users.
vi. Git push - Pushing is the opposite of fetching (with a few caveats). It lets you move a local branch to another repository, which serves as a convenient way to publish contributions. This is like svn commit, but it sends a series of commits instead of a single changeset.