Git is a tool used to keep track of changes to files, especially the code of the projects**.** It is termed a distributed version control system because of its behaviour to allow multiple people to work on the same project, even if they are not connected to a common server.

It was created by a person named Linus Torvalds in 2005 for the development of the Linux kernel. In this article, we will discuss Git, its features, advantages, and disadvantages.

What is Git?

Git is a tool that helps you manage changes to your code over time. It keeps track of every little change or update you make, you can always look back at previous versions or undo mistakes if needed.

Features of Git

- **Version Control System**: Git keeps track of every change you make to your project files. You can go back to previous versions is any requirement arises.
- **Repositories**: A Git repository (or repo) is like a project's central hub where everything related to your work is stored and managed. There are two main types:
    - **Local Repository**: This is a copy of the repository which is stored in your computer. You can work on your project and make changes here.
    - **Remote Repository**: This is stored on a server, like GitHub, where you and others can share and work on a project.
- **Commits**: Every time you make changes and save them in Git, these are called commits. The repository keeps track of all the commits you have made.
- **Branches**: A repository allows you to create Branches and work on new features and fixes. For example, you can have a "main" branch for stable code and a "feature" branch for new features you're developing.
- **Merging**: Once you are done with the branches you have made, you can merge those branches into main branch. This adds the changes you have made in the rest of the project.
- **Cloning**: Cloning is a process of making a complete copy of Git repository. It's like copying the entire project from central location (like website) to your own computer.

Reason to Choose Git

- **Distributed System**: Every developer has a complete local copy of the project, allowing for offline work and increased resilience.
- **Performance**: Git is fast and efficient, handling large projects with ease.
- **Branching and Merging**: Git's lightweight branching and easy merging facilitate parallel development and feature isolation.
- **Collaboration**: Tools like GitHub enhance team collaboration, code reviews, and project management.
- **Track Changes**: Git tracks changes and maintains a history, making it easy to revert to previous versions.

Steps to Setup a Git

- **Install Git**: Download and install Git from the official Git website.
- **Configure Git**: Set up your username and email.

```
git config --global user.name "Your Name"
git config --global user.email "your.email@example.com"
```

- **Create a Repository**: Navigate to your project directory and initialize a Git repository.

```
git init
```

- **Make Your First Commit**: Add files to the staging area and commit your changes.

```
git add .
git commit -m "Initial commit"
```