Git

Git is the most popular SCM tool right now.

Git tracks the complete history of changes made to a file. So, developers can see who made what changes at what time.

You don't need to pay any subscription fee to use Git. It is an entirely free, open-source SCM tool.

Git supports non-linear development. Developers can remotely make changes to a particular part of the project without worrying about the rest.

Git creates backups of files with different branches. Developers can go back to any version of the file they want.

Git is scalable, so you won't have any problem collaborating, even when more developers start contributing to the project.

Instead of having a central repository (a virtual storage space for files), Git is distributed. Every developer has a local repository of files in their system. In addition, there is a remote repository in a hosting service like GitHub.

Why Use Git?

Git tracks all the changes made to a project, including information on who made the changes and when. This makes collaboration very easy.

If developers make mistakes or unwanted changes to a file, they can simply revert to a previous version.

Developers need to implement different branches to a file and implement different changes on individual branches. They might need to do this to control the code's quality or to test different changes to the code.

What is Version Control?

A control system is used for tracking content, and Git is a control system for source code.

Content in the source code keeps changing in the development process. So a new file version is constantly being made. This is why Git is called a Version Control System.

Thus, a version control system is a software system that tracks changes to a file over time.

In the case of Git, the code is not stored in a central server. Instead, it is stored on the user's computer and in a Git repository. This is why it is called a Distributed Version control system.

How to Use Git?

Git is pretty easy to learn if you have proper guidance. While Git can be used with a GUI client, we recommend using the terminal as many developers prefer it.

Git was designed for Unix systems, so we have GitBash, which provides you with a Linux-like environment for windows. You can use default terminals in Linux and macOS as they are Unix-based OS.

How to upload project on GIT.

Step 1: Install Git and Create a GitHub Account

https://git-scm.com/download/win

https://github.com/

Step 2: Create a Local Git Repo and Add a File

A repository ("repo" in short) is a virtual storage space that allows you to save different versions of your code. As a developer, you need to work with two kinds of repositories:

Local Repository - A repository on a developer's computer.

Remote Repository - A repository on a hosting server like GitHub.

create a local Git repo:

Create a folder with your project name. Right-click on the folder, and you will see a 'Git Bash Here' option. Click it.

A Git Bash terminal will open. Here you can type the git init command to create a repository. You will see a .git/ folder created on your project folder. Make sure to turn on the 'View Hidden files and folder' option to see this folder.

Step 3: Add a File to the Staging Area

You need to remember three things while using Git:

the working directory is where you are currently working or coding at (your project folder)

the staging area is a temporary space where you add files you made changes to

your Git repository is where you commit the files with changes

Let us add a file index.html to the staging area. For this, just type in the following command in the terminal:

git add index.html

Step 4: Create a Commit

You can now commit the changes you made to the index.html file.

For this, type in the following command:

git commit -m

Step 5: Create a New Branch

When you create a new branch in Git, it essentially creates a new version of your file. A developer must move between branches because different branches will have different file versions.

Step 6: Create a Remote Repo on GitHub

git remote add origin <link to your repo>

Step 7: Push the Local Repository to GitHub

To push your repo to GitHub, we can use the following command:

git push origin main

Upload Code

1 git config --global user.name ""

2 git config --global user.email ""

3. git clone(link)

5. cd folder

4. git init

5.git status

6. git add .

7. git commit -m ""

8. $ git remote add origin git@github.com:mamatageloth/ShareMarket.git

$ git push origin main

The authenticity of host 'github.com (20.207.73.82)' can't be established.

ED25519 key fingerprint is

SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.

This key is not known by any other names.

Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.

git@github.com: Permission denied (publickey).

fatal: Could not read from remote repository.

error: solution

$ git remote set-url origin https://github.com/mamatageloth/ShareMarket.git

continue with the code: captcha code

verification process

8. git push origin main

To https://github.com/mamatageloth/ShareMarket.git

 ! [rejected] main -> main (fetch first)

error: failed to push some refs to 'https://github.com/mamatageloth/ShareMarket.git'

hint: Updates were rejected because the remote contains work that you do not

hint: have locally. This is usually caused by another repository pushing to

hint: the same ref. If you want to integrate the remote changes, use

hint: 'git pull' before pushing again.

hint: See the 'Note about fast-forwards' in 'git push --help' for details.

error:solution

9. git push origin main --force

$ git checkout master(switched to new branch)

Frequently Asked Questions

1. What is a Git repository?

A Git repository is virtual storage that stores all the versions of your files. It acts as metadata for your Git project. When you initialize a repository, Git creates a .git/ folder in your project folder, which tracks the entire history of your project.

2. What is Git Bash?

Git Bash is an application that provides a terminal (in Windows) to use Git. Since Git is designed to be run in a Unix-style terminal, it feels natural with macOS and Linux as both are built using Unix. But you will need to install Git Bash on Windows to run Git in a Unix-like environment.

3. Are Git and GitHub the same?

Git is a version control system, while GitHub is a cloud-based hosting service to manage Git repositories. GitHub is like a remote hard disk where you store your repositories for collaboration, while Git is how you push and pull files from GitHub.

4. Is Git a programming language?

No, Git is a version control system meant for code collaboration. It is not a programming language.

5. What is a branch in Git?

A branch is basically a file version created after you commit some changes to the repo. The default branch in Git is called the master or main branch.

You can do your work on a new branch you created while the main branch (master) remains stable. After you finish your work, you can merge it with the main branch.

6. Who should use Git?

Git can be used by anyone looking to collaborate with others. While programmers mainly use Git, it can also be used by people from other areas like marketing and design. Since Git basically tracks changes to files, people working in any profession can use it for this purpose.

Git Commands

git init

The command git init is used to create an empty Git repository.

After the git init command is used, a .git folder is created in the directory with some subdirectories. Once the repository is initialized, the process of creating other files begins.

git add

Add command is used after checking the status of the files, to add those files to the staging area.

Before running the commit command, "git add" is used to add any new or modified files.

git commit

The commit command makes sure that the changes are saved to the local repository.

The command "git commit –m <message>" allows you to describe everyone and help them understand what has happened.

git status

The git status command tells the current state of the repository.

The command provides the current working branch. If the files are in the staging area, but not committed, it will be shown by the git status. Also, if there are no changes, it will show the message no changes to commit, working directory clean.

git remote

The git remote command is used to create, view, and delete connections to other repositories.

The connections here are not like direct links into other repositories, but as bookmarks that serve as convenient names to be used as a reference.

git push

The command git push is used to transfer the commits or pushing the content from the local repository to the remote repository.

The command is used after a local repository has been modified, and the modifications are to be shared with the remote team members.

git pull

The git pull command is used to fetch and merge changes from the remote repository to the local repository.

The command "git pull origin master" copies all the files from the master branch of the remote repository to the local repository.