# Git

Git is a version control system.

Git helps you keep track of code changes.

Git is used to collaborate on code.

git --version

## Git and Remote Repositories

Git and GitHub are different things.

In this tutorial you will understand what Git is and how to use it on the remote repository platforms, like GitHub.

You can choose, and change, which platform to focus on by clicking in the menu on the right:

### What does Git do?

* Manage projects with ****Repositories****
* ****Clone**** a project to work on a local copy
* Control and track changes with ****Staging**** and ****Committing****
* ****Branch**** and ****Merge**** to allow for work on different parts and versions of a project
* ****Pull**** the latest version of the project to a local copy
* ****Push**** local updates to the main project

### Working with Git

* Initialize Git on a folder, making it a ****Repository****
* Git now creates a hidden folder to keep track of changes in that folder
* When a file is changed, added or deleted, it is considered ****modified****
* You select the modified files you want to ****Stage****
* The ****Staged**** files are ****Committed****, which prompts Git to store a ****permanent**** snapshot of the files
* Git allows you to see the full history of every commit.
* You can revert back to any previous commit.
* Git does not store a separate copy of every file in every commit, but keeps track of changes made in each commit!

### Why Git?

* Over 70% of developers use Git!
* Developers can work together from anywhere in the world.
* Developers can see the full history of the project.
* Developers can revert to earlier versions of a project.

### What is GitHub?

* Git is not the same as GitHub.
* GitHub makes tools that use Git.
* GitHub is the largest host of source code in the world, and has been owned by Microsoft since 2018.
* In this tutorial, we will focus on using Git with GitHub.

## Using Git with Command Line

To start using Git, we are first going to open up our Command shell.

For Windows, you can use Git bash, which comes included in Git for Windows. For Mac and Linux you can use the built-in terminal.

The first thing we need to do, is to check if Git is properly installed:

## Configure Git

Now let Git know who you are. This is important for version control systems, as each Git commit uses this information:

git config --global user.name "w3schools-test" git

config --global user.email ["test@w3schools.com"](mailto:\"test@w3schools.com\")

### Working Directory and Staging Area or Index

### The working directory is the place where files are checked out. In other CVCS, developers generally make modifications and commit their changes directly to the repository. But Git uses a different strategy. Git doesn’t track each and every modified file. Whenever you do commit an operation, Git looks for the files present in the staging area. Only those files present in the staging area are considered for commit and not all the modified files.

### Let us see the basic workflow of Git.

### Step 1 − You modify a file from the working directory.

### Step 2 − You add these files to the staging area.

### Step 3 − You perform commit operation that moves the files from the staging area. After push operation, it stores the changes permanently to the Git repository.



## Benefits of Git

A version control application allows us to ****keep track**** of all the changes that we make in the files of our project. Every time we make changes in files of an existing project, we can push those changes to a repository. Other developers are allowed to pull your changes from the repository and continue to work with the updates that you added to the project files.

Some ****significant benefits**** of using Git are as follows:

## Why Git?

We have discussed many ****features**** and ****benefits**** of Git that demonstrate the undoubtedly Git as the ****leading version control system****. Now, we will discuss some other points about why should we choose Git.





# What is GitHub?

GitHub is a Git repository hosting service. GitHub also facilitates with many of its features, such as access control and collaboration. It provides a Web-based graphical interface.

GitHub is an American company. It hosts source code of your project in the form of different programming languages and keeps track of the various changes made by programmers.

It offers both ****distributed version control and source code management (SCM)**** functionality of Git. It also facilitates with some collaboration features such as bug tracking, feature requests, task management for every project.



## Features of GitHub

GitHub is a place where programmers and designers work together. They collaborate, contribute, and fix bugs together. It hosts plenty of open source projects and codes of various programming languages.

Some of its significant features are as follows.

* Collaboration
* Integrated issue and bug tracking
* Graphical representation of branches
* Git repositories hosting
* Project management
* Team management
* Code hosting
* Track and assign tasks
* Conversations
* Wikisc

## Benefits of GitHub

GitHub can be separated as the Git and the Hub. GitHub service includes access controls as well as collaboration features like task management, repository hosting, and team management.

The key benefits of GitHub are as follows.

* It is easy to contribute to open source projects via GitHub.
* It helps to create an excellent document.
* You can attract recruiter by showing off your work. If you have a profile on GitHub, you will have a higher chance of being recruited.
* It allows your work to get out there in front of the public.
* You can track changes in your code across versions.

The Git installation is completed. Now you can access the ****Git Gui**** and ****Git Bash****.

|  |  |
| --- | --- |
| **Centralized Version Control System** | **Distributed Version Control System** |
| In CVCS, The repository is placed at one place and delivers information to many clients. | In DVCS, Every user has a local copy of the repository in place of the central repository on the server-side. |
| It is based on the client-server approach. | It is based on the client-server approach. |
| It is the most straightforward system based on the concept of the central repository. | It is flexible and has emerged with the concept that everyone has their repository. |
| In CVCS, the server provides the latest code to all the clients across the globe. | In DVCS, every user can check out the snapshot of the code, and they can fully mirror the central repository. |
| CVCS is easy to administrate and has additional control over users and access by its server from one place. | DVCS is fast comparing to CVCS as you don't have to interact with the central server for every command. |
| The popular tools of CVCS are ****SVN**** (Subversion) and ****CVS****. | The popular tools of DVCS are ****Git**** and ****Mercurial****. |
| CVCS is easy to understand for beginners. | DVCS has some complex process for beginners. |
| If the server fails, No system can access data from another system. | if any server fails and other systems were collaborating via it, that server can restore any of the client repositories |

## Git configuration levels

The git config command can accept arguments to specify the configuration level. The following configuration levels are available in the Git config.

* local
* global
* system

****--local****

It is the default level in Git. Git config will write to a local level if no configuration option is given. Local configuration values are stored in ****.git/config**** directory as a file.

****--global****

The global level configuration is user-specific configuration. User-specific means, it is applied to an individual operating system user. Global configuration values are stored in a user's home directory. ****~ /.gitconfig**** on UNIX systems and ****C:\Users\\.gitconfig**** on windows as a file format.

****--system****

The system-level configuration is applied across an entire system. The entire system means all users on an operating system and all repositories. The system-level configuration file stores in a ****gitconfig**** file off the system directory. ****$(prefix)/etc/gitconfig**** on UNIX systems and ****C:\ProgramData\Git\config**** on Windows.

The order of priority of the Git config is local, global, and system, respectively. It means when looking for a configuration value, Git will start at the local level and bubble up to the system level.

# 12 Git Commands

There are many different ways to use Git. Git supports many command-line tools and graphical user interfaces. The Git command line is the only place where you can run all the Git commands.