118242

BRIAN MURIITHI KOOME

Git is an Open Source Distributed Version Control System.

Real life projects generally have multiple developers working in parallel. So a version control system like Git is needed to ensure there are no code conflicts between the developers.

Additionally, the requirements in such projects change often. So a version control system allows developers to revert and go back to an older version of the code.

Finally, sometimes several projects which are being run in parallel involve the same codebase. In such a case, the concept of branching in Git is very important.

GitHub is just a code repository. Git is the system, which does code version control. On the other hand, **Git Bash** is a feature in Git for Windows, which provides **BASH**emulation for running **Git commands** from the **command line**.

**Git Clone**

git clone is used to clone an existing remote repository into your computer. The command for this is:

git clone [repository url]

When you clone a remote repository using git clone, a remote connection called origin is automatically created, which connects back to the cloned repository.

**What is a git branch?**

A branch is an independent development route, which can later be connected back to the main branch.

We use git branch command to **create**, **list**, **rename** and **delete**branches.

When working on big projects, which takes a long time and involve many developers, it is advisable to define a strict branching model or design centered around project release.

## How to handle git merge conflict

What are git conflicts and when do they happen.

Lets say in out home.html file, one developer modifies lines 4 and 5 and another developer modifies lines 10 and 11. Well, everything is okay because there is no common line in the modifications.

However, if both developers modified code in home.html and somehow they happen to modify the same line, say they both change line 7, then a conflict is said to have happened.

When a git conflict happens, git will show you the **2 sections**, from which you need to choose the correct one.  It is up to you to decide what to keep and what to discard.

Now that [**GitHub**](https://github.com/)provides a [**free plan for unlimited private**](https://github.com/pricing) repositories and unlimited collaborators, you have a chance to use it to work on big projects with a big team.

GitHub is increasingly becoming attractive because it offers issue tracking, project management boards, group milestone tracking, team discussions and organization management in the free plan.

# How to rename the git "master" branch to "main"

The reason for renaming is to remove references to slavery and replace it with a more inclusive term. Using master in GitHub does not really mean that they refer to slavery, but it is the origin of the term **master**that is the concern.

This means any repository created from **October 1, 2020**, will automatically use **main**as the default branch and **NOT master**.

However, those repositories created before the above date will have the same default branch and this change will not affect them.

The **bad**or **good**news is that  by the **end of the year 2020**, all existing repositories still using master as their default branch will be changed to main.

Before that is done, you need to do it yourself to avoid a disruption on your **git workflow**, in case GitHub's automated tools make a mistake

**Proceed as follows;-**

**step 1:**Rename old branch, master to new branch, main using the -m option

git branch -m master main

**step 2**: Update remote repository using -u option to add upstream (tracking) reference.

git push -u origin main

step 4: Now you can **delete** the **master** branch, in the remote repository

git push origin --delete master