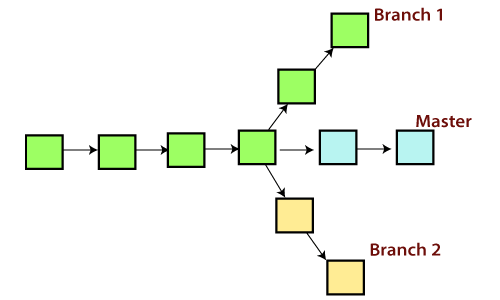
Git Branch

A branch is a version of the repository that diverges from the main working project. It is a feature available in most modern version control systems. A Git project can have more than one branch. These branches are a pointer to a snapshot of your changes. When you want to add a new feature or fix a bug, you spawn a new branch to summarize your changes. So, it is complex to merge the unstable code with the main code base and also facilitates you to clean up your future history before merging with the main branch.



Git Master Branch

The master branch is a default branch in Git. It is instantiated when first commit made on the project. When you make the first commit, you're given a master branch to the starting commit point. When you start making a commit, then master branch pointer automatically moves forward. A repository can have only one master branch.

Master branch is the branch in which all the changes eventually get merged back. It can be called as an official working version of your project.

Operations on Branches

We can perform various operations on Git branches. The **git branch command** allows you to **create**, **list**, **rename** and **delete** branches. Many operations on branches are applied by git checkout and git merge command. So, the git branch is tightly integrated with the **git checkout** and **git merge commands**.

**The Operations that can be performed on a branch:**

Create Branch

You can create a new branch with the help of the **git branch** command. This command will be used as:

**Syntax:**

1. $ git branch  **<branch** name**>**

**Output:**

Git Branch

This command will create the **branch B1** locally in Git directory.

List Branch

You can List all of the available branches in your repository by using the following command.

Either we can use **git branch - list** or **git branch** command to list the available branches in the repository.

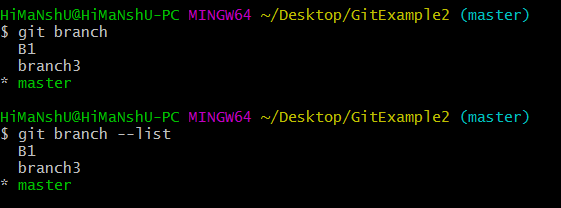
**Syntax:**

1. $ git branch --list

**or**

1. $ git branch

**Output:**



Here, both commands are listing the available branches in the repository. The symbol \* is representing currently active branch.

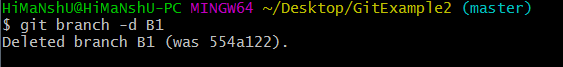
Delete Branch

You can delete the specified branch. It is a safe operation. In this command, Git prevents you from deleting the branch if it has unmerged changes. Below is the command to do this.

**Syntax:**

1. $ git branch -d**<branch** name**>**

**Output:**



This command will delete the existing branch B1 from the repository.

The **git branch d** command can be used in two formats. Another format of this command is **git branch D**. The '**git branch D**' command is used to delete the specified branch.

1. $ git branch -D **<branch** name**>**

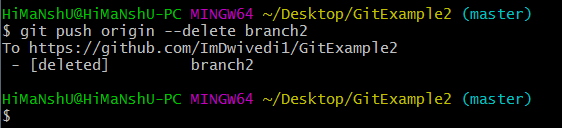
Delete a Remote Branch

You can delete a remote branch from Git desktop application. Below command is used to delete a remote branch:

**Syntax:**

1. $ git push origin -delete **<branch** name**>**

**Output:**



As you can see in the above output, the remote branch named **branch2** from my GitHub account is deleted.

Switch Branch

Git allows you to switch between the branches without making a commit. You can switch between two branches with the **git checkout** command. To switch between the branches, below command is used:

1. $ git checkout**<branch** name**>**

**Switch from master Branch**

You can switch from master to any other branch available on your repository without making any commit.

**Syntax:**

1. $ git checkout **<branch** name**>**

**Output:**

Git Branch

As you can see in the output, branches are switched from **master** to **branch4** without making any commit.

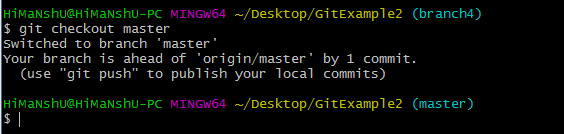
**Switch to master branch**

You can switch to the master branch from any other branch with the help of below command.

**Syntax:**

1. $ git branch -m master

**Output:**



As you can see in the above output, branches are switched from **branch1 to master** without making any commit.

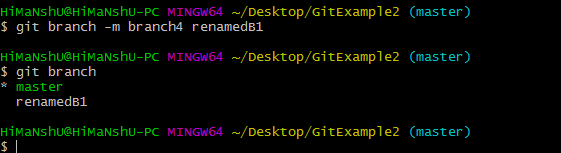
Rename Branch

We can rename the branch with the help of the **git branch** command. To rename a branch, use the below command:

**Syntax:**

1. $ git branch -m **<old** branch name**><new** branch name**>**

**Output:**



As you can see in the above output, **branch4** renamed as **renamedB1**.

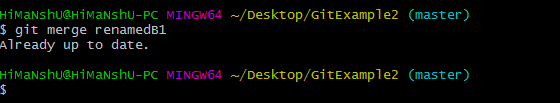
Merge Branch

Git allows you to merge the other branch with the currently active branch. You can merge two branches with the help of **git merge** command. Below command is used to merge the branches:

**Syntax:**

1. $ git merge **<branch** name**>**

**Output:**



From the above output, you can see that **the master** branch **merged** with **renamedB1**. Since I have made no-commit before merging, so the output is showing as already up to date.