

Module 3: Branching and Merging Git

Demo1: Demo on Create, delete, and manage branches with Git

Problem Statement:

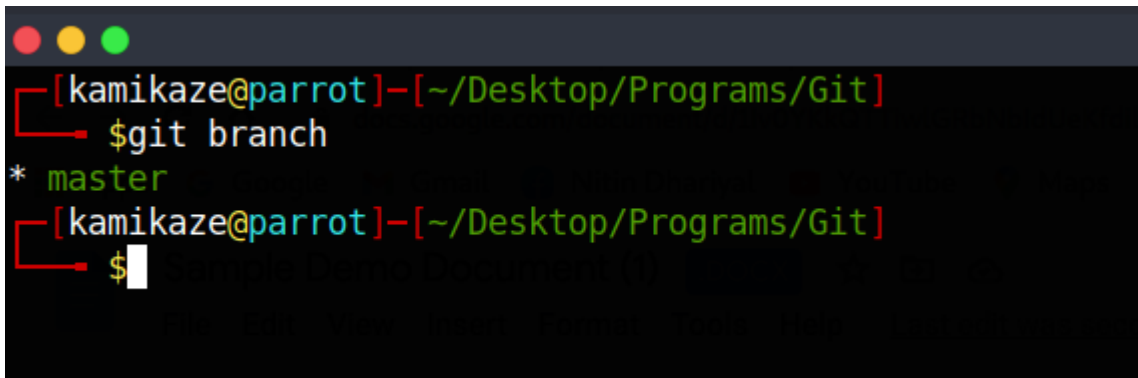
How we can create, delete, and merge branches using Git Bash

Solution:

Step 1: In This demo, we will go through GitHub Branch Creation, deletion, and managing them by our Git Command Line Interface.

Command Used: git branch

This command will do more than just create and delete branches. If you run it with no arguments, you will get a simple listing of your current branches:

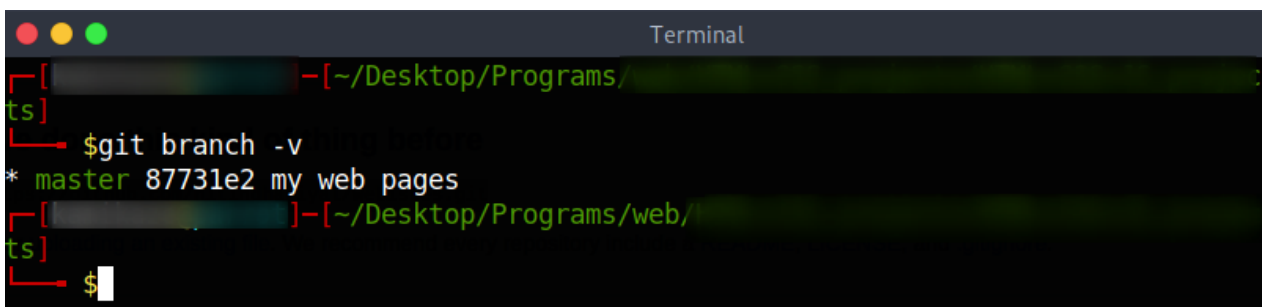


```
[kamikaze@parrot]-[~/Desktop/Programs/Git]
$git branch
* master
[kamikaze@parrot]-[~/Desktop/Programs/Git]
$
```

The ***** character here prefixes the **master** branch indicates the branch that you currently have checked out (i.e., the branch that **HEAD** points to). It means that if you commit at this point, the **master** branch will be moved forward with your new work.

Step 2: Let us the last commit on each branch,

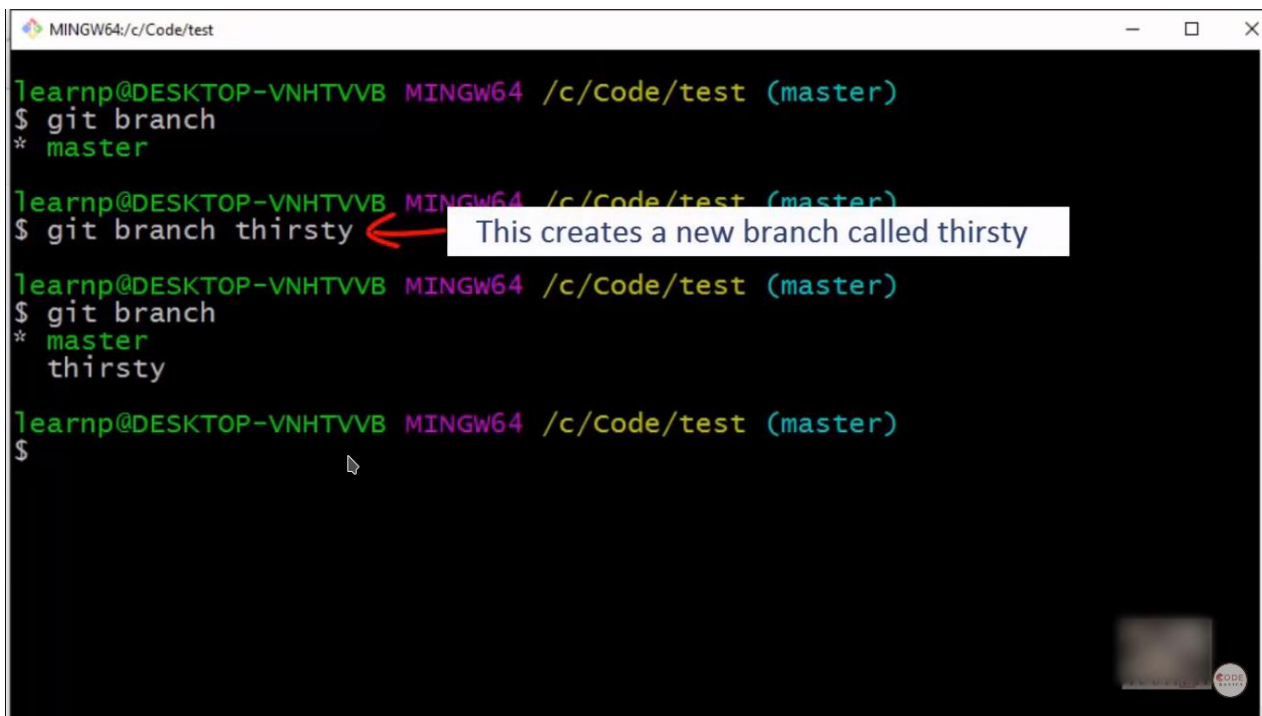
Command Used: git branch -v



```
Terminal
[kamikaze@parrot]-[~/Desktop/Programs/Git]
$git branch -v
* master 87731e2 my web pages
[kamikaze@parrot]-[~/Desktop/Programs/web/]
$
```

Step 3: Let us Create a new branch,

Command Used: git branch branchname

A terminal window titled 'MINGW64:/c/Code/test' showing the process of creating a new branch. The user is currently on the 'master' branch. They run 'git branch' which shows '* master'. Then they run 'git branch thirsty', which creates a new branch named 'thirsty'. A red arrow points from a text box to the command. Finally, they run 'git branch' again, which shows both '* master' and 'thirsty'.

```
learnp@DESKTOP-VNHTVVB MINGW64 /c/Code/test (master)
$ git branch
* master

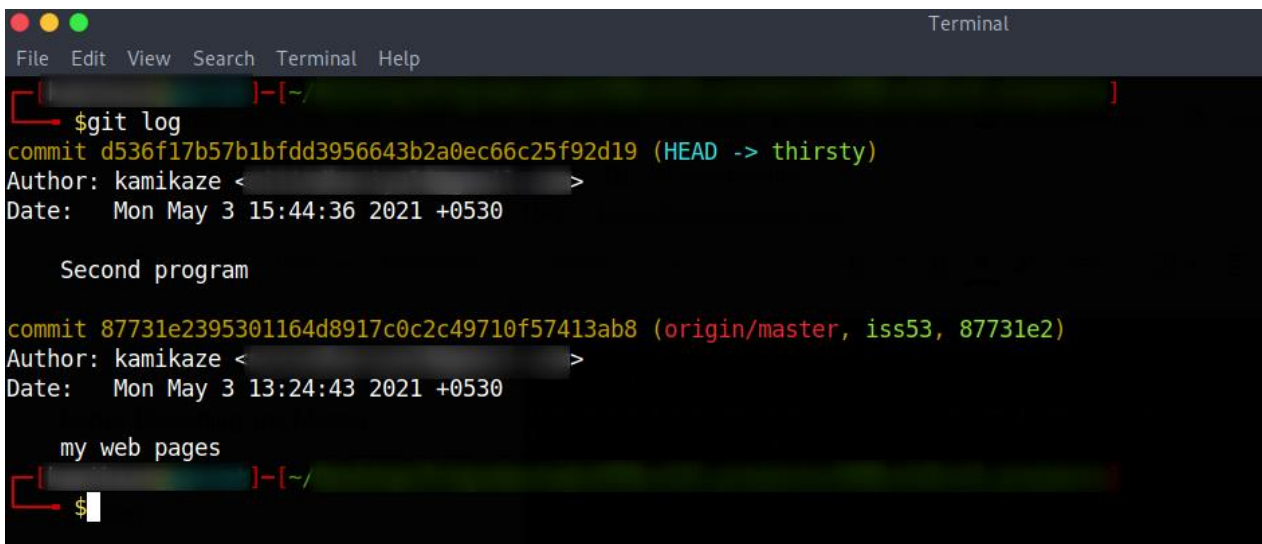
learnp@DESKTOP-VNHTVVB MINGW64 /c/Code/test (master)
$ git branch thirsty

learnp@DESKTOP-VNHTVVB MINGW64 /c/Code/test (master)
$ git branch
* master
  thirsty

learnp@DESKTOP-VNHTVVB MINGW64 /c/Code/test (master)
$
```

Step 4: Let us Check the logs.

Command Used: git log

A terminal window titled 'Terminal' showing the output of 'git log'. It displays two commits. The first commit is on the 'thirsty' branch, and the second commit is on the 'master' branch. The output includes commit hashes, branch names, author names, and dates.

```
Terminal
File Edit View Search Terminal Help

$git log
commit d536f17b57b1bfdd3956643b2a0ec66c25f92d19 (HEAD -> thirsty)
Author: kamikaze <>
Date: Mon May 3 15:44:36 2021 +0530

    Second program

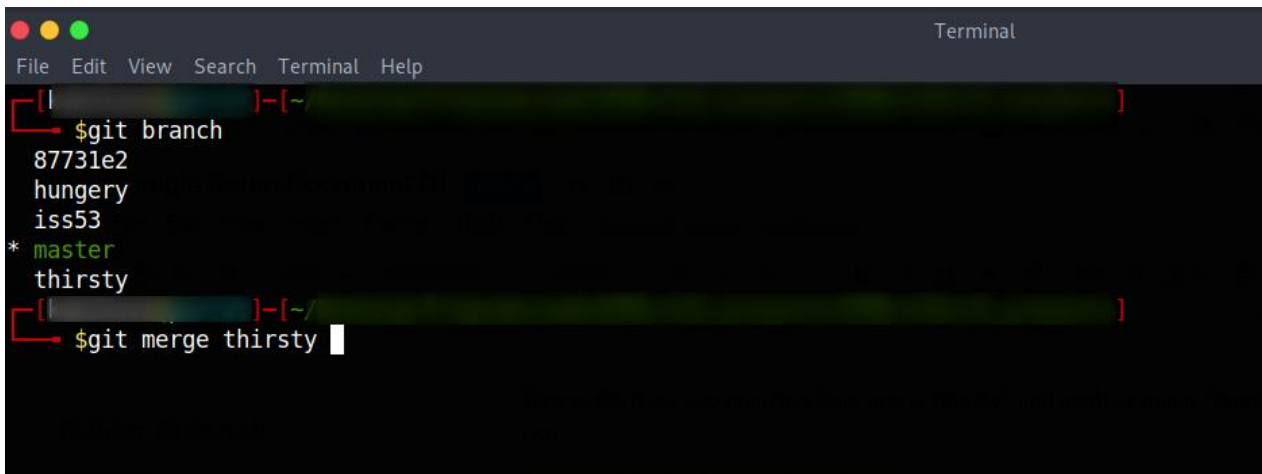
commit 87731e2395301164d8917c0c2c49710f57413ab8 (origin/master, iss53, 87731e2)
Author: kamikaze <>
Date: Mon May 3 13:24:43 2021 +0530

    my web pages

$
```

Step 5: We have two branches here one is “thirsty” and another one is “hungry”. Let us merge these two.

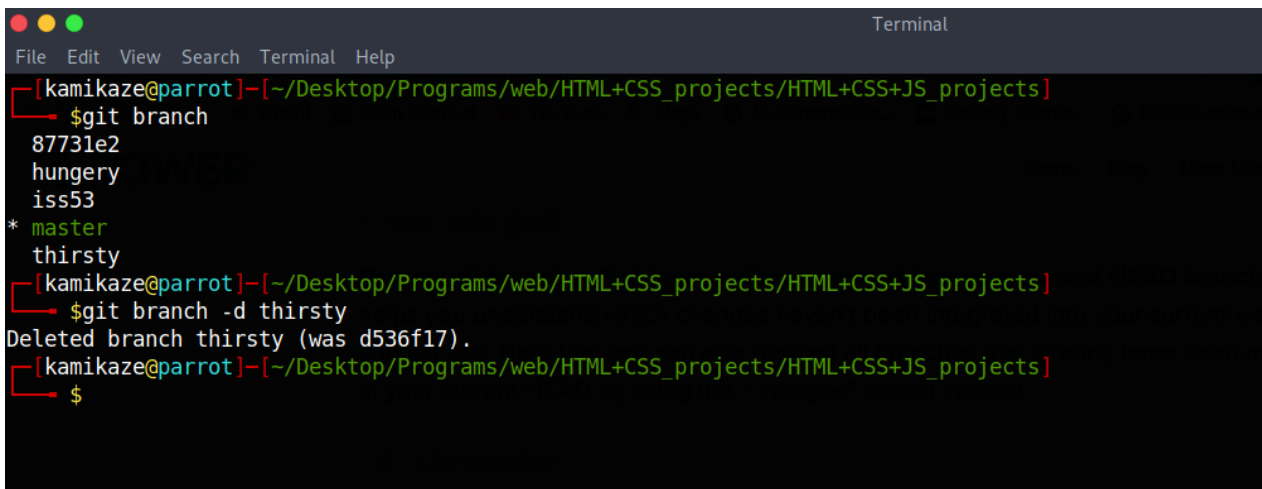
Command Used: git merge branchname



```
Terminal
File Edit View Search Terminal Help
[kamikaze@parrot]~$ git branch
87731e2
hungry
iss53
* master
thirsty
[kamikaze@parrot]~$ git merge thirsty
```

Step 6: Let us delete one branch:

Command Used: git branch -d branchname



```
Terminal
File Edit View Search Terminal Help
[kamikaze@parrot]~/Desktop/Programs/web/HTML+CSS_projects/HTML+CSS+JS_projects$ git branch
87731e2
hungry
iss53
* master
thirsty
[kamikaze@parrot]~/Desktop/Programs/web/HTML+CSS_projects/HTML+CSS+JS_projects$ git branch -d thirsty
Deleted branch thirsty (was d536f17).
[kamikaze@parrot]~/Desktop/Programs/web/HTML+CSS_projects/HTML+CSS+JS_projects$
```