

Module 3: Branching and Merging Git

Demo2: Demo on git merge and code conflicts

Problem Statement:

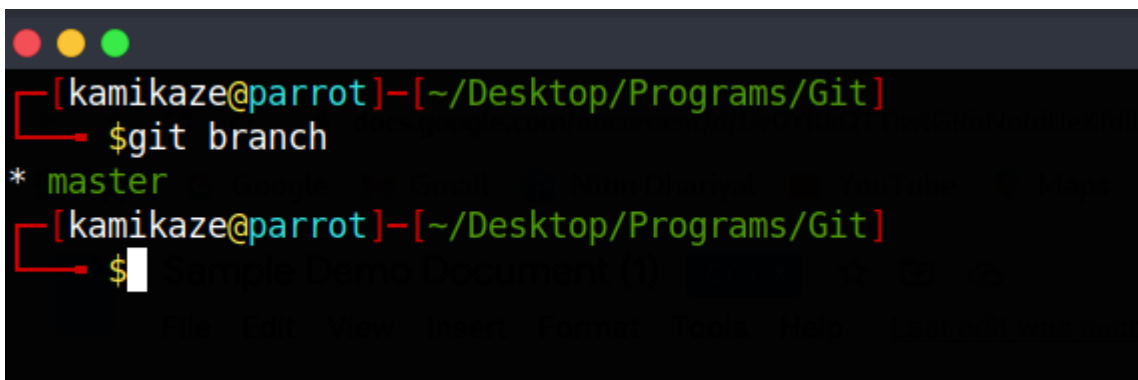
How we perform merging of two branches by git and manage code conflicts.

Solution:

Step 1: In This demo, we will go through GitHub Branch Creation then we merged them with the help of the Git Command Line Interface.

Command Used: git branch

This command will do more than just create and delete branches. You will get a simple listing of your current branches if you run it without any arguments:

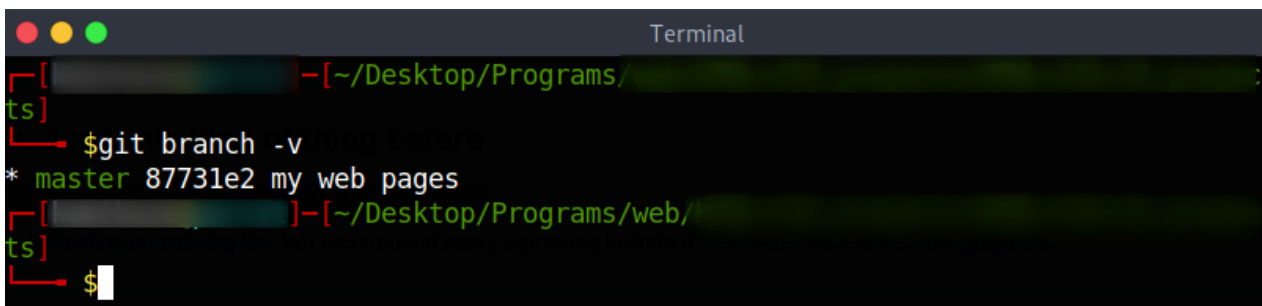


```
[kamikaze@parrot]--[~/Desktop/Programs/Git]
$git branch
* master
[kamikaze@parrot]--[~/Desktop/Programs/Git]
$
```

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 runs 'git branch' and sees '* master'. Then they run 'git branch thirsty', which is annotated with a red arrow and a text box saying 'This creates a new branch called thirsty'. Finally, they run 'git branch' again and see both '* master' and 'thirsty' listed.

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

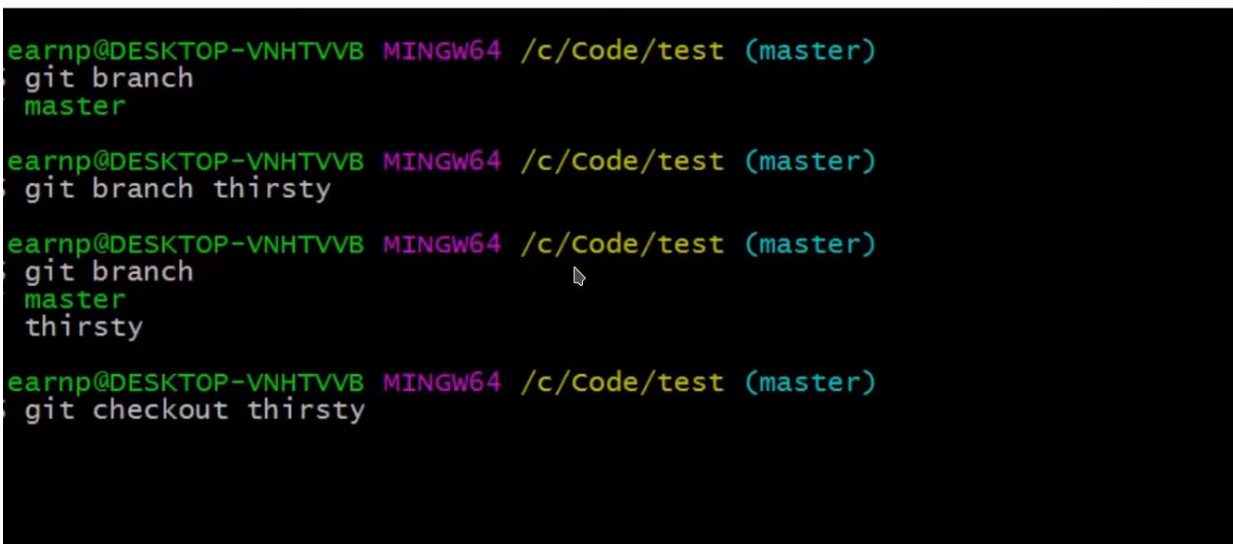
learnp@DESKTOP-VNHTVVB MINGW64 /c/Code/test (master)
$ git branch thirsty

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

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

Step 4: Let us Checkout into our branch.

Command Used: git checkout branchname

A terminal window showing the user checking out the 'thirsty' branch. They run 'git branch' twice to confirm the branches exist. Then they run 'git checkout thirsty', and the prompt changes from '(master)' to '(thirsty)'.

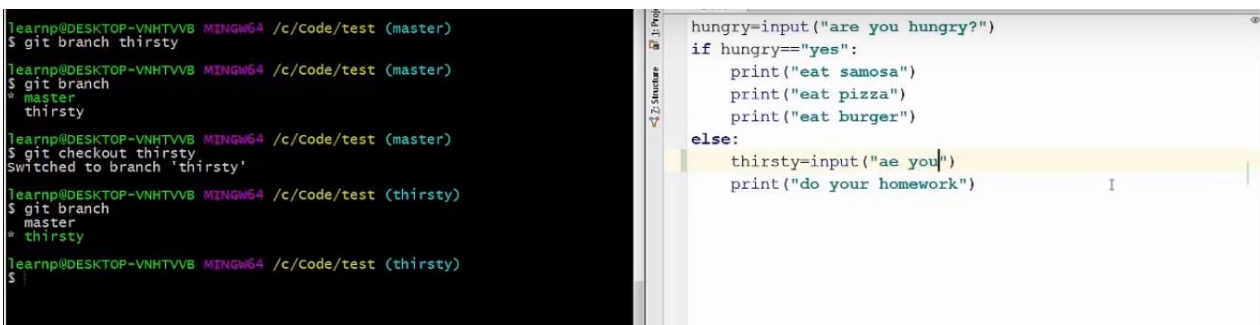
```
learnp@DESKTOP-VNHTVVB MINGW64 /c/Code/test (master)
$ git branch
master

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 (thirsty)
$
```

Step 5: Let us create a file in the “thirsty” branch. I created a thirsty.py file, you can create your own file.

Two side-by-side screenshots. The left one is a terminal window showing the user switching to the 'thirsty' branch with 'git checkout thirsty' and then running 'git branch' to confirm they are on 'thirsty'. The right one is a code editor showing the content of 'thirsty.py', which contains a simple Python program that asks if the user is hungry and prints responses based on the input.

```
learnp@DESKTOP-VNHTVVB MINGW64 /c/Code/test (master)
$ git branch thirsty
learnp@DESKTOP-VNHTVVB MINGW64 /c/Code/test (master)
$ git checkout thirsty
Switched to branch 'thirsty'

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

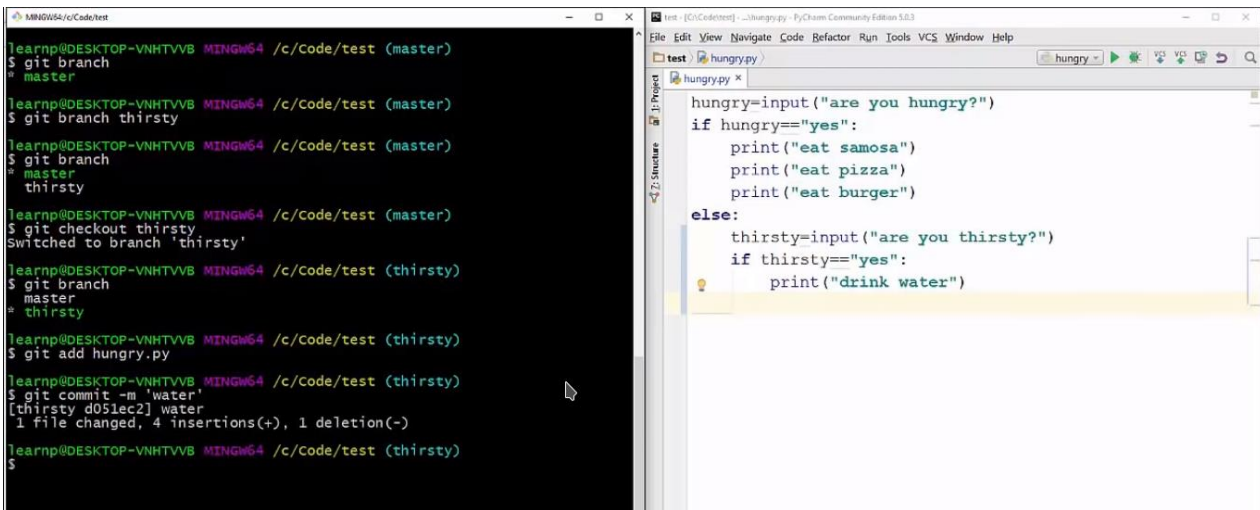
learnp@DESKTOP-VNHTVVB MINGW64 /c/Code/test (thirsty)
$
```

```
hungry=input("are you hungry?")
if hungry=="yes":
    print("eat samosa")
    print("eat pizza")
    print("eat burger")
else:
    thirsty=input("are you thirsty?")
    print("do your homework")
```

Step 6: Let us add this file into our branch and make a commit for it:

Command Used: git add filename

Command Used: git commit -m “your commit”



The terminal window shows the following commands and output:

```
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)
$ git checkout thirsty
Switched to branch 'thirsty'

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

learnp@DESKTOP-VNHTVVB MINGW64 /c/code/test (thirsty)
$ git add hungry.py

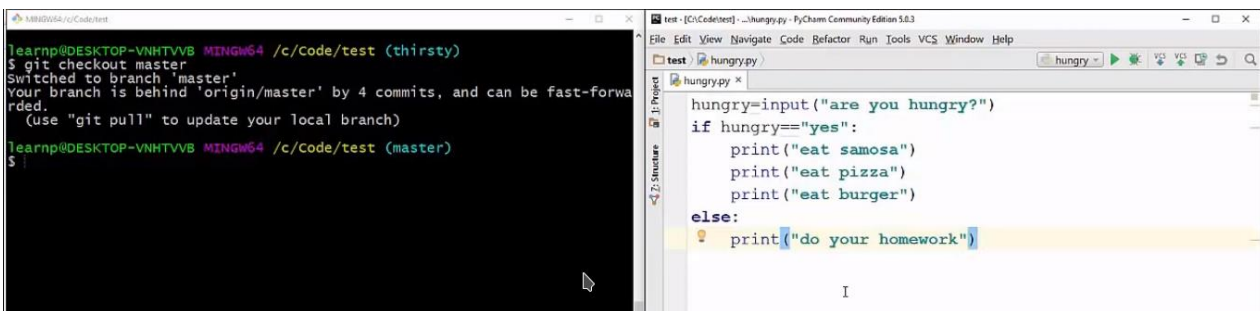
learnp@DESKTOP-VNHTVVB MINGW64 /c/code/test (thirsty)
$ git commit -m 'water'
[thirsty d051ec2] water
1 file changed, 4 insertions(+), 1 deletion(-)

learnp@DESKTOP-VNHTVVB MINGW64 /c/code/test (thirsty)
$
```

The code editor shows the following Python script:

```
hungry=input("are you hungry?")
if hungry=="yes":
    print("eat samosa")
    print("eat pizza")
    print("eat burger")
else:
    thirsty=input("are you thirsty?")
    if thirsty=="yes":
        print("drink water")
```

Step 7: Let us check our master branch, you see the code here got changed because the changes we did are only on the “thirsty” branch.



The terminal window shows the following commands and output:

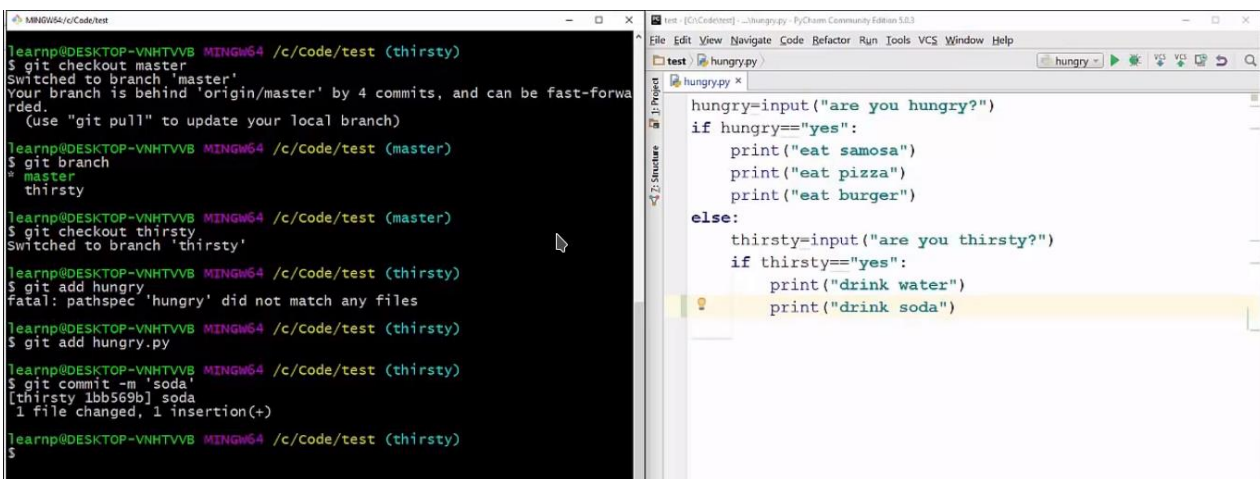
```
learnp@DESKTOP-VNHTVVB MINGW64 /c/code/test (thirsty)
$ git checkout master
Switched to branch 'master'
Your branch is behind 'origin/master' by 4 commits, and can be fast-forwarded.
(use "git pull" to update your local branch)

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

The code editor shows the following Python script:

```
hungry=input("are you hungry?")
if hungry=="yes":
    print("eat samosa")
    print("eat pizza")
    print("eat burger")
else:
    print("do your homework")
```

Step 8: Let us make a new branch over here, I make a branch name “hungry”. Let us add a file into it(hungry.py) and make commit into it.



The terminal window shows the following commands and output:

```
learnp@DESKTOP-VNHTVVB MINGW64 /c/code/test (thirsty)
$ git checkout master
Switched to branch 'master'
Your branch is behind 'origin/master' by 4 commits, and can be fast-forwarded.
(use "git pull" to update your local branch)

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

learnp@DESKTOP-VNHTVVB MINGW64 /c/code/test (master)
$ git checkout thirsty
Switched to branch 'thirsty'

learnp@DESKTOP-VNHTVVB MINGW64 /c/code/test (thirsty)
$ git add hungry
fatal: pathspec 'hungry' did not match any files

learnp@DESKTOP-VNHTVVB MINGW64 /c/code/test (thirsty)
$ git add hungry.py

learnp@DESKTOP-VNHTVVB MINGW64 /c/code/test (thirsty)
$ git commit -m 'soda'
[thirsty 1bb569b] soda
1 file changed, 1 insertion(+)

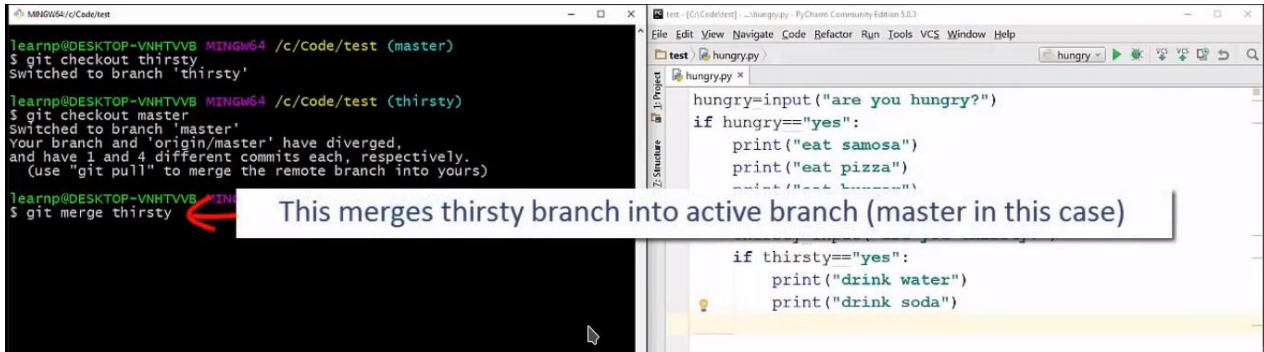
learnp@DESKTOP-VNHTVVB MINGW64 /c/code/test (thirsty)
$
```

The code editor shows the following Python script:

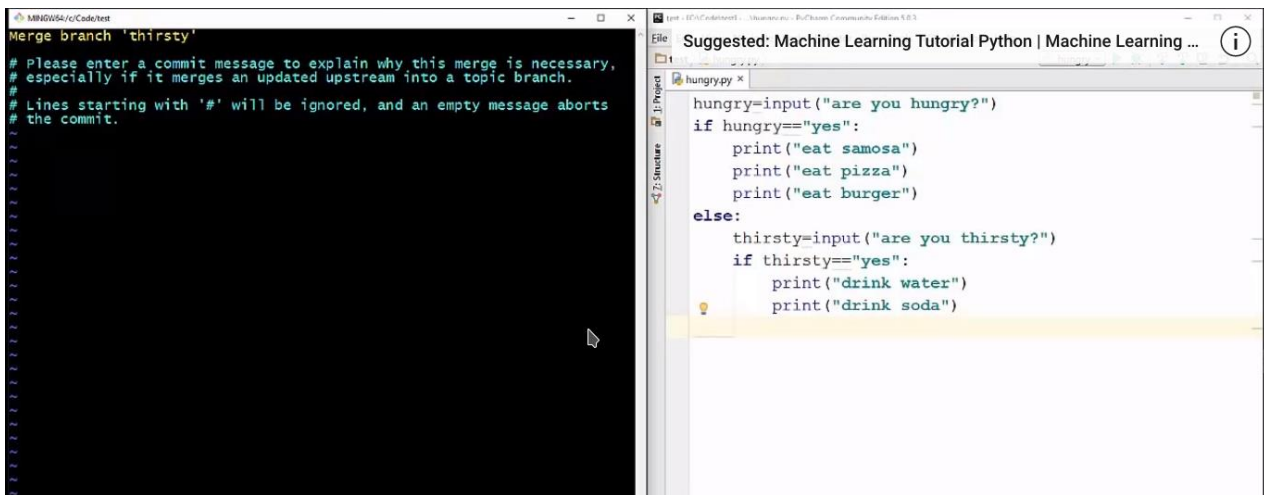
```
hungry=input("are you hungry?")
if hungry=="yes":
    print("eat samosa")
    print("eat pizza")
    print("eat burger")
else:
    thirsty=input("are you thirsty?")
    if thirsty=="yes":
        print("drink water")
        print("drink soda")
```

Step 9: Let us merge branch, I am merging the thirsty branch into my master branch for that we use,

Command Used: `git merge branchname`



Command Used: !wq



Command: git log

Finally, you can check, you have performed merging into git.