How does the branch and merge model in git work?

Any repository starts with a default master branch and a. By default the master branch points to the last commit made by the user.

So basically in order to do branch (i.e to have a new branch) we use the command line . The new branch will point to the last commit made just as the master branch. However in this case it is important to know which branch will move forward.

Here it is important to understand the concept of HEAD. It is a pointer to the current branch. By typing the command , the HEAD pointer will move from the master branch to the new branch created by the user.

To make it clearer the following two images show how HEAD, Master and the new Branch –assume it’s called testing- look like after each command

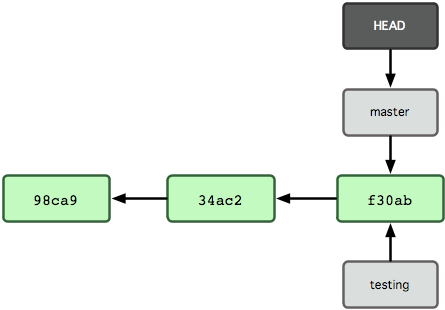


Figure : After "$ git branch" command [[1]](#footnote-1)

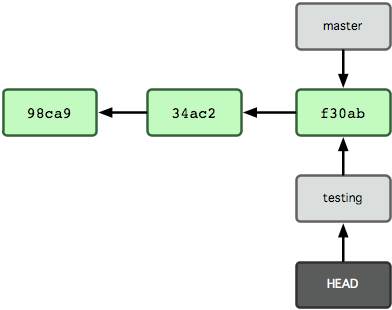


Figure : After $ git checkout [[2]](#footnote-2)

So now for any upcoming commits only testing branch –the current branch with HEAD referring to it- will move forward. There other git command that make it possible to change from branch to another, where any changes made at any branch are isolated from each other.

1. <http://git-scm.com/book/en/Git-Branching-What-a-Branch-Is> [↑](#footnote-ref-1)
2. <http://git-scm.com/book/en/Git-Branching-What-a-Branch-Is> [↑](#footnote-ref-2)