

# Branch Creation

 [coursera.org/learn/git-distributed-development/supplement/FI0Es/branch-creation](https://coursera.org/learn/git-distributed-development/supplement/FI0Es/branch-creation)

The basic command for creating a new branch is:

```
1  
$ git branch [branch_name] [starting_point]
```



If you do not give any arguments, you get a list of branches with the active one starred. A very detailed history of the branches can be obtained with:

```
1  
$ git show-branch
```



If you are creating a new branch, you must give it a name. There are some rules, like no blank spaces in the name, no special or control characters, no slashes at the end, etc. Keep it simple.

A branch is like a tag, but you can add commits to it (changes induced by git branch):

Command	Source Files	Index	Commit Chain	References
<b>git branch</b>	Unchanged	Unchanged	Unchanged	A new branch is created in <code>.git/refs/heads</code> <b>HEAD</b> for the new branch points to <b>HEAD</b> of the current branch; the current branch is set to the new branch

The starting point is any commit. If there is a tag that describes it, you can use that instead of the long string. If you do not give the argument, you create a copy of the active branch as of its last commit. So, you might do:

1

```
$ git branch devel
```



to create a new development branch off the mainline.

You can delete the **devel** branch with:

1

```
$ git branch -d devel
```



which cannot be your current working branch. Recovering an accidentally deleted branch is rather difficult, although not always impossible, so use care.