Git Branching - Basic Branching and Merging

Let’s go through a simple example of branching and merging with a workflow that you might use in the real world. You’ll follow these steps:

1. Do some work on a website.
2. Create a branch for a new user story you’re working on.
3. Do some work in that branch.

At this stage, you’ll receive a call that another issue is critical and you need a hotfix. You’ll do the following:

1. Switch to your production branch.
2. Create a branch to add the hotfix.
3. After it’s tested, merge the hotfix branch, and push to production.
4. Switch back to your original user story and continue working.

Common Options

```
# git branch
```

List all of the branches in your repository. This is synonymous with git branch --list.

```
# git branch <branch>
```

Create a new branch called ＜branch＞. This does not check out the new branch.

```
# git branch -d <branch>
```

Delete the specified branch. This is a “safe” operation in that Git prevents you from deleting the branch if it has unmerged changes.

```
# git branch -D <branch>
```

Force delete the specified branch, even if it has unmerged changes. This is the command to use if you want to permanently throw away all of the commits associated with a particular line of development.

```
# git branch -m <branch>
```

Rename the current branch to ＜branch＞.

```
# git branch -a
```

List all remote branches.

Deleting Branches

Once you’ve finished working on a branch and have merged it into the main code base, you’re free to delete the branch without losing any history:

```
# git branch -d crazy-experiment
```

However, if the branch hasn’t been merged, the above command will output an error message:

```
error: The branch 'crazy-experiment' is not fully merged. If you are sure you want to delete it, run 'git branch -D crazy-experiment'.
```

This protects you from losing access to that entire line of development. If you really want to delete the branch (e.g., it’s a failed experiment), you can use the capital -D flag:

```
# git branch -D crazy-experiment
```

This deletes the branch regardless of its status and without warnings, so use it judiciously.

The previous commands will delete a local copy of a branch. The branch may still exist in remote repos. To delete a remote branch execute the following.

```
# git push origin --delete crazy-experiment
```

Or

```
# git push origin :crazy-experiment
```

This will push a delete signal to the remote origin repository that triggers a delete of the remote crazy-experiment branch.