Git Branching

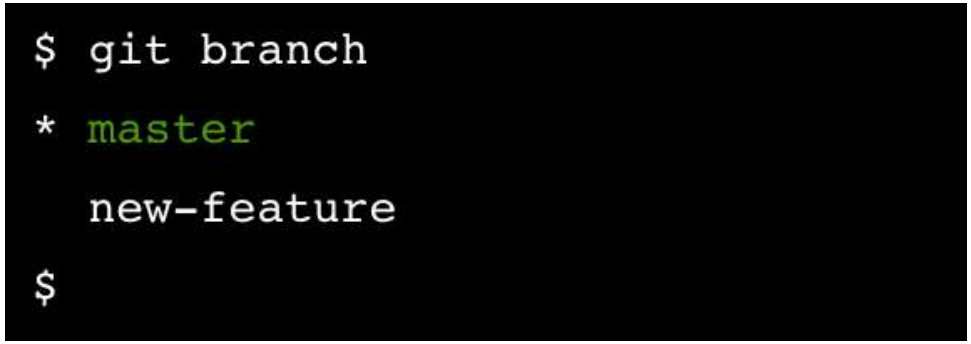
The Master Branch

In Git, the main project is completed on the `master` branch. Making your first commit in a new git repository will automatically create a `master` branch. Create new branches from the `master` branch to develop new features for a project. These branches can be merged into `master` at a later time to incorporate the new features. You can use `git branch` to check what branch you're on.

```
$ git init
Initialized empty Git repository in /home/ccuser/new-
project/.git/
$ echo "Hello World!" >> hello.txt
$ git add hello.txt
$ git commit -m 'initial commit'
[master (root-commit) bb0e565] initial commit
 1 file changed, 1 insertion(+)
 create mode 100644 hello.txt
$ git branch
* master
```

Viewing the Current Branch

In Git, the `git branch` command will display all of the branches. The current branch will display `*` before its name.



Creating a New Branch

In Git, the `git branch branch-name` command is used to create a new branch called `branch-name`. Branches should be named something that describes the purpose of the branch.

Note that branch names can't contain whitespace: `new-feature` and `new_feature` are valid branch names, but `new feature` is not.

```
$ git branch new-feature
$ git branch
* master
  new-feature
$
```

Merging Branches

In Git, the `git merge branch-name` command will add the changes from `branch-name` into the current branch. Use this command when you have finished building a feature in a separate branch and want to bring those changes into your current branch.

```
$ git merge resume-edits
Updating 86b8a77..c443513
Fast-forward
 resume.txt | 2 ++
 1 file changed, 2 insertions(+)
$
```

Merge Conflicts

In Git, a merge conflict occurs when the same file is changed on the current branch and the branch that is being merged. An error will appear displaying: `CONFLICT (content): Merge conflict in [filename]`.

Git will automatically edit the file with the conflict to show where the conflict is. The current branch's text will be between `<<<<<<< HEAD` and `=======`. The text from the branch that is being merged into the current branch will be between `=======` and `>>>>>>> branch-name`

To resolve a merge conflict, edit the file with the conflict, decide which parts of each branch's edits should be kept, then add and commit the file.

```
PETE PAN
<<<<<<< HEAD
Address: No 31 Kensington Hill Park, London, England
=======
Address: 113 Gloucester Rd Patchway, Bristol, England
>>>>>>> resume-edits
-------------------
```

Deleting a Branch

In Git, the `git branch -d branch_name` command is used to delete the `branch_name` branch. It's good practice to delete a branch after it has been merged into the `master` branch.

```
$ git branch
* master
  new-feature
$ git branch -d new-feature
Deleted branch new-feature (was 828ea2c).
$
```