2. Working from an individual branch

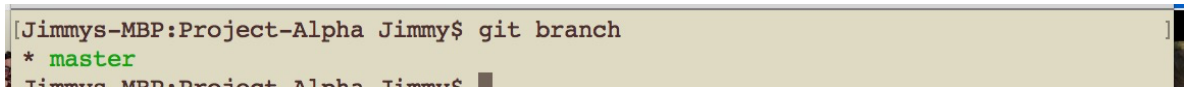
Before you read this documentation, please refer **1.Setup: Cloning the Repository** to setup the repository onto your local machine.

Why do we use Branches?

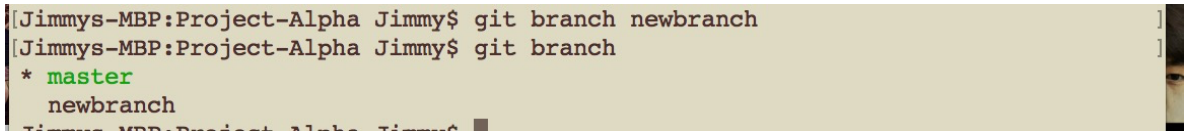
- Branches is a good way to keep our current work from messing up the origin files.
- The origin files are placed onto a 'master' branch, which everyone can download and access.
- When we want to write new code, we are essentially making a 'side branch' of the project so that we can edit the files.
- When these files are ready, we can push them up so they can be checked and merged into the origin master files.

Step 1 :

- Open your terminal. To check which branch you are currently in, type '**git branch**':



- As we can see, the current branch we are on is the 'master' branch. Let's first create a separate branch by typing ***'git branch + <new branch name>'***.

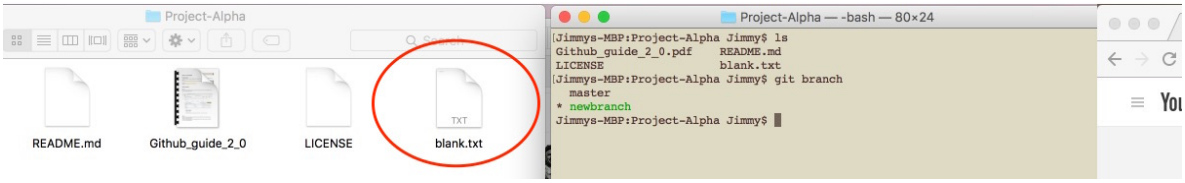


Step 2:

- To switch to this branch, type ***'git checkout'***, followed by the name of the branch we want to switch to.
- After we have done this, we can confirm we are now on the new branch by typing **git branch** once again.

```
[Jimmys-MBP:Project-Alpha Jimmy$ git checkout newbranch
D       Github_guide.pdf
Switched to branch 'newbranch'
[Jimmys-MBP:Project-Alpha Jimmy$ git branch
  master
* newbranch
```

Step 3:
Anything we do within this branch will not affect the files on the master branch. For example, I created a file here named *blank.txt*.



If we add and commit this file (covered in 3) **within our new branch**, then the *blank.txt* **will not appear** in our master branch.