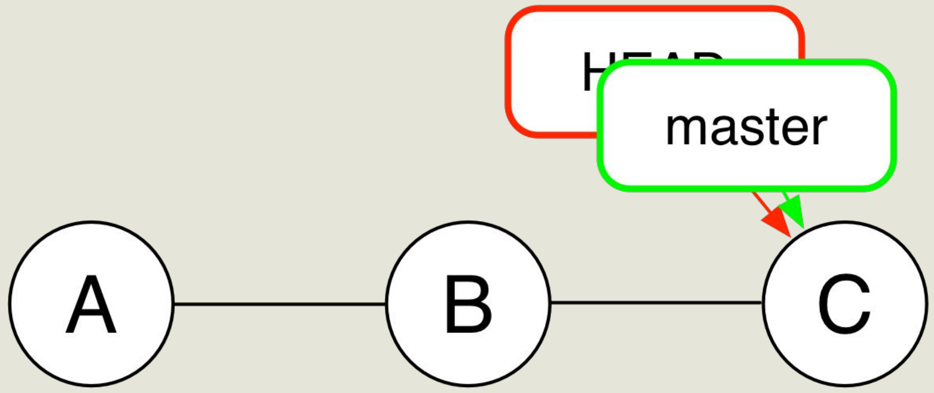
BRANCHING

A Picture of Git

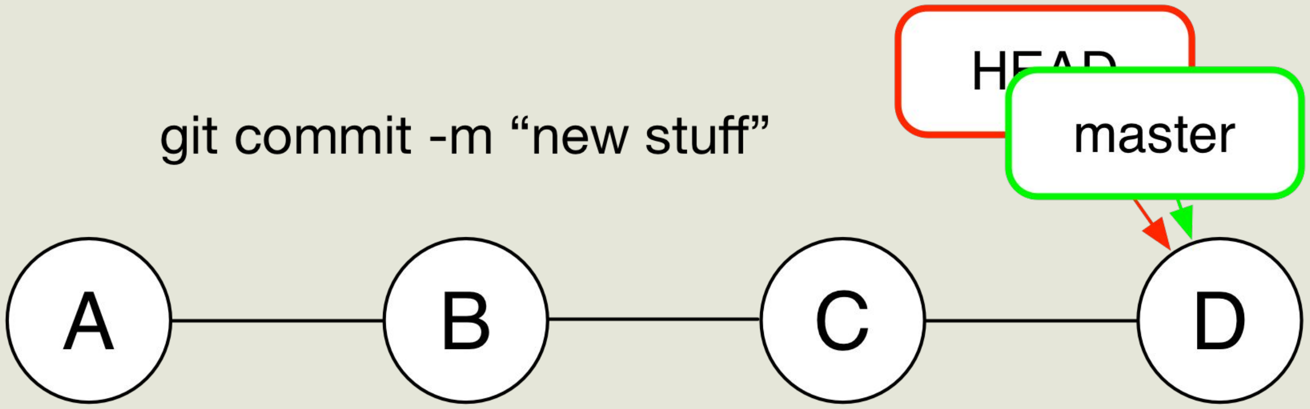


You may also be familiar with the label "master".

This is the name that Git automatically gives to the first branch in a repository.

A branch is actually just a label that points to a specific point in time.

A Picture of Git



When you make a commit in Git, you add a new point to the timeline.

The HEAD label moves to this new point.

So does the label for the branch you are on.

More Git Workflow

Remember the basic Git workflow: modify, add, commit.

Now imagine your repository is code for a vital website.

Further imagine that your production server is running using code on the master branch.

You wouldn't want anyone making willy-nilly changes to master.

It would be much better to have only tested, vetted code end up in master.

So, you ask your development team to implement fixes and features on branches.

For example:

NASA has code on GitHub. You can imagine they wouldn’t want untested code affecting their satellites and other toys.

GIT COMMAND: `checkout`

The command `checkout` has two uses.

The first takes a branch name as an argument and moves you to that branch (timeline).

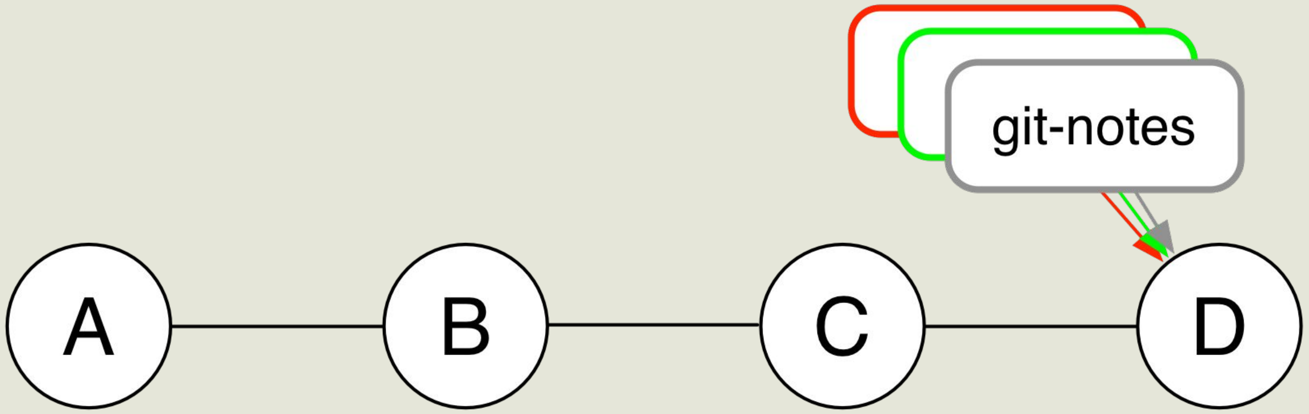
```
$ git checkout branch-name
```

But, how do you make a new branch? That’s the second use.

If you enter `-b` before the argument this will create a branch with that name and move you to it.

```
$ git checkout -b new-branch-name
```

Making a Branch



Checking out a new branch creates a new label. Since nothing has actually changed you are still on the same commit snapshot as you were previously.

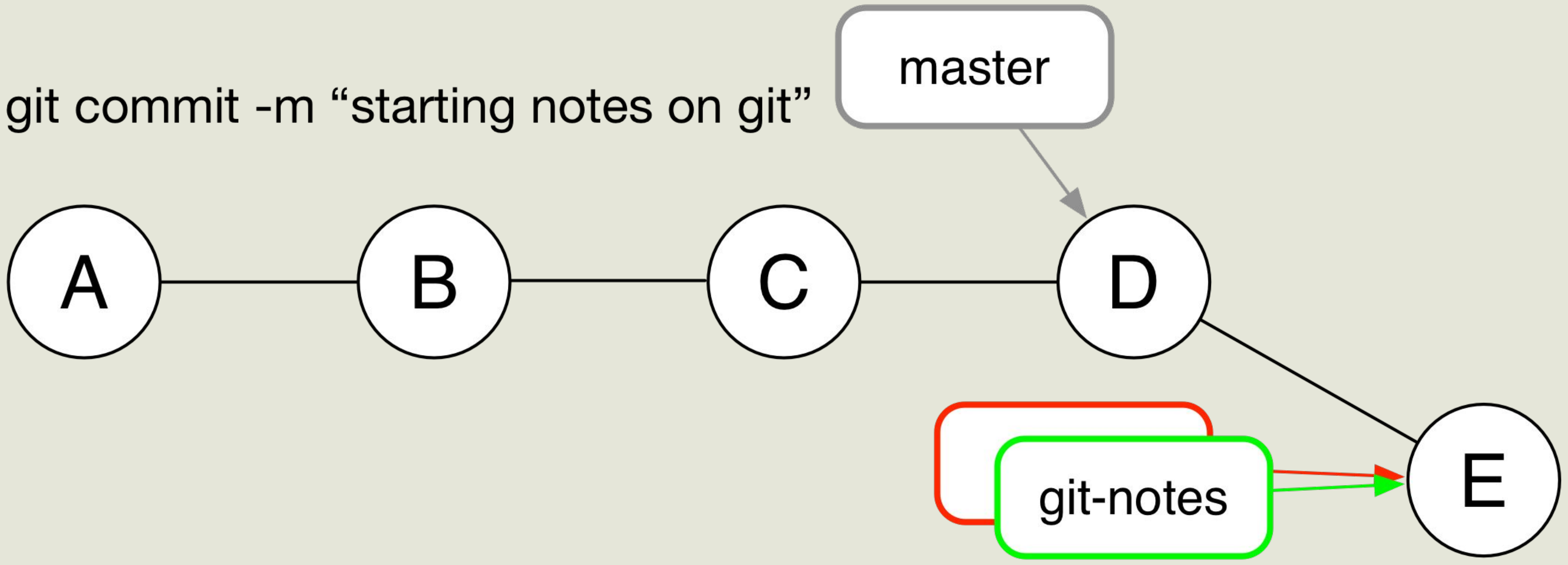
GIT COMMAND: `branch`

The command `branch` will list all the branches that exist in your local repo and indicate which one you are currently on.

```
$ git branch
  kitteh-feature
  git-notes
* master
```

Visualizing the Results

```
git commit -m "starting notes on git"
```



Once you have completed a Git workflow cycle the branches actually diverge.

Pushing Different Branches

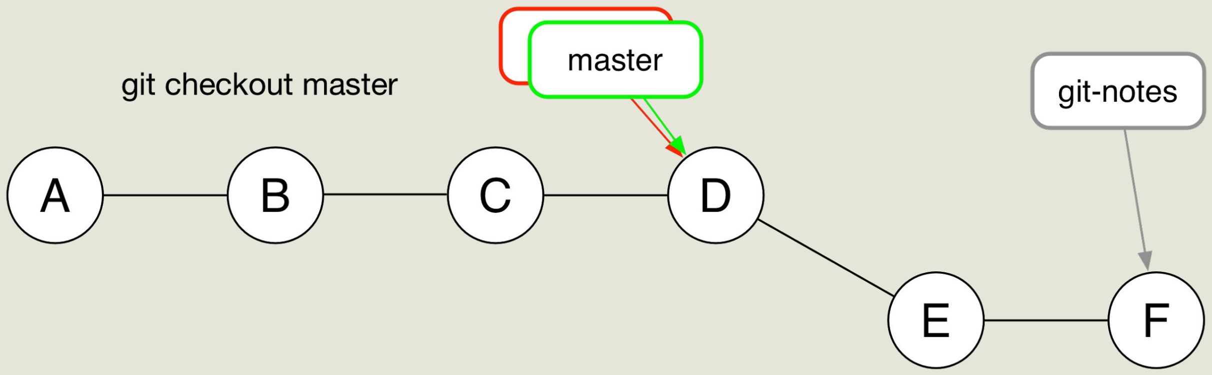
In order to push this new branch to GitHub we need to clarify in the `push` command which branch these commits should belong to.

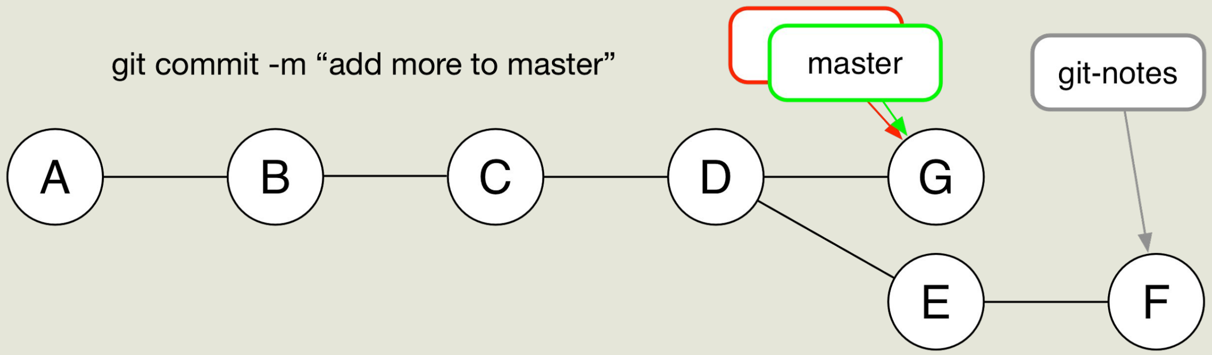
When you do a push you always include the name of the branch that you are on locally after the `origin` alias.

```
$ git push origin git-notes
…
To git@github.com:cewing/uge-workshop.git
 * [new branch]      git-notes -> git-notes
```

Isolating your Changes

You can switch back and forth between branches to keep changes isolated to their relevant branches (timelines).





ANY

QUESTIONS?

Git Commands

New terminology learned in this lesson:

git branch
View existing branches in your repository.

git checkout
Create new branch or change the active branch.