Branches

Become introduced to the concept of “branches” in Git.

We'll cover the following

- What are branches?
  - Repository with no branch
  - Adding branch to a repository

What are branches?

In case you have not looked at a source control tool before, “branching” is a core concept.

Repository with no branch

A series of changes to a repository might look like this:

Three commits to a repository

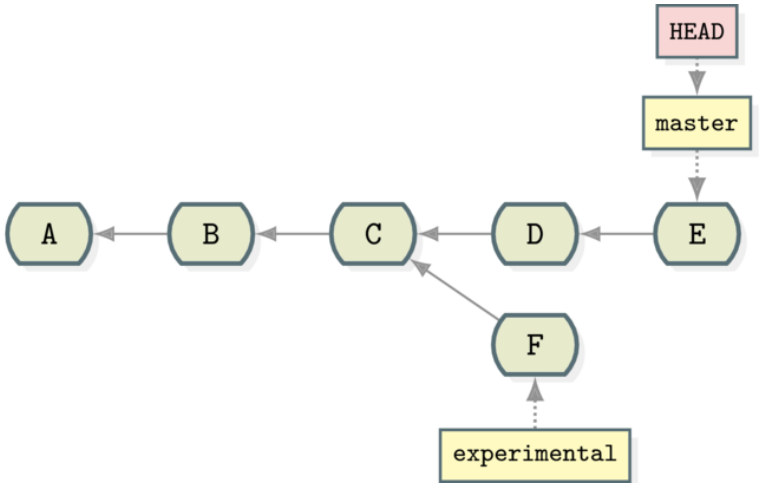
Direction of arrows in Git diagrams
The arrows in the above diagram point *backward* in time. This is a convention standard in Git diagrams. It indicates that each commit depends on the previous commit back to the initial commit. In the above diagram, the last change `C` depends on `B` which in turn is dependent on `A`.

The letters do not have a specific meaning here. They simply indicate multiple changes to lines that are grouped together.

Here, change `A` is made, then `B`, and then `C`. This might be informally called the *main line*. Most of the time, it's called the "master" in Git.

Adding branch to a repository

But let's say someone wants to make an experimental change but does not want to affect the *main line*. They might *branch* the code at point `C`:



Code branched at point C

Study the above diagram, and make sure you understand it.

While the main line has continued with changes `D` and `E`, another branch has been created from `C` that has one change: `F` made to it. This `experimental` branch does not include the `D` and `E` changes, and the `master` branch does not include the `experimental` changes.

That way, users can choose to get a view of the source on the *main line* branch or the *experimental* one.

That's all a branch is: a set of changes from a specific point in time.

Back

Next

The Four Phases of Git Content

What about GitHub?

Mark as Completed

Report an Issue

Ask a Question (https://discuss.educative.io/tag/branches__introduction-to-git__learn-git-the-hard-way)