

Git Tutorial



Backlog lets you manage projects with Git integration

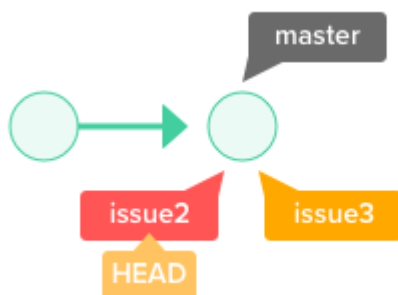
Learn more →

Work in parallel

Branching allows us to work in multiple parallel workspaces.

Let's create two branches. Create one with the name "issue2" and another with the name "issue3", then switch over to "issue2".

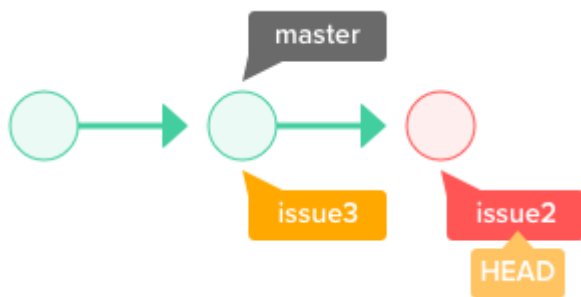
```
$ git branch issue2
$ git branch issue3
$ git checkout issue2
Switched to branch 'issue2'
$ git branch
* issue2
  issue3
  master
```



Add the bold text below to myfile.txt and commit the change.

Git Tutorial

```
$ git add myfile.txt
$ git commit -m "append description of the commit
command"
[issue2 8f7aa27] append description of the commit
command
1 files changed, 2 insertions(+), 0 deletions(-)
```



Now switch to "issue3" branch.

```
$ git checkout issue3
Switched to branch 'issue3'
```

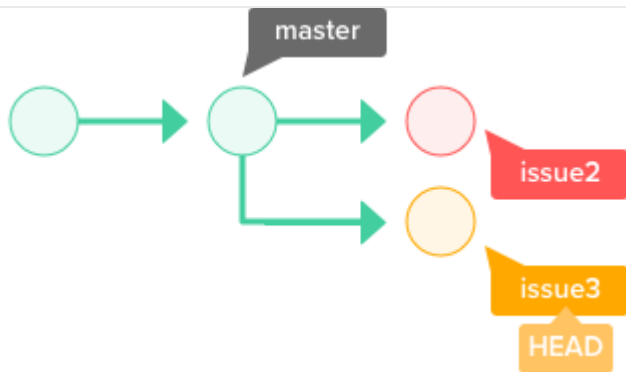
"issue3" currently has the same history/content as the master branch. It will not include the recent change that we have just made. This is because the recent change has been committed to the "issue2" branch.

Add the bold text below to myfile.txt and commit the change.

```
Git commands even a monkey can understand
add: Register a change in an index
pull: Obtain the content of the remote repository
```

```
$ git add myfile.txt
$ git commit -m "append description of the pull
command"
```

Git Tutorial



We have now added two different line of texts to two different branches in parallel.

[Prev](#)[Next](#)

Backlog has built-in Git repositories for every project

Try it free

Follow us on

[English](#)[Terms](#)[Privacy](#)

© 2021 Nulab, Inc. All rights reserved.