search

Confused by git? Here's a git crash course to fix that 🎉

Chris Achard Sep 10 · 3 min read

#git #beginners #tutorial

This was originally posted as a twitter thread:
https://twitter.com/chrisachard/status/1171124289128554498

NOTE: if you are looking for a very basic intro to git, I recommend reading this guide by Atlassian first.

Do you use git but still don't really understand it?

Here's a 🔥 git crash course 🔥 to fix that 🎉

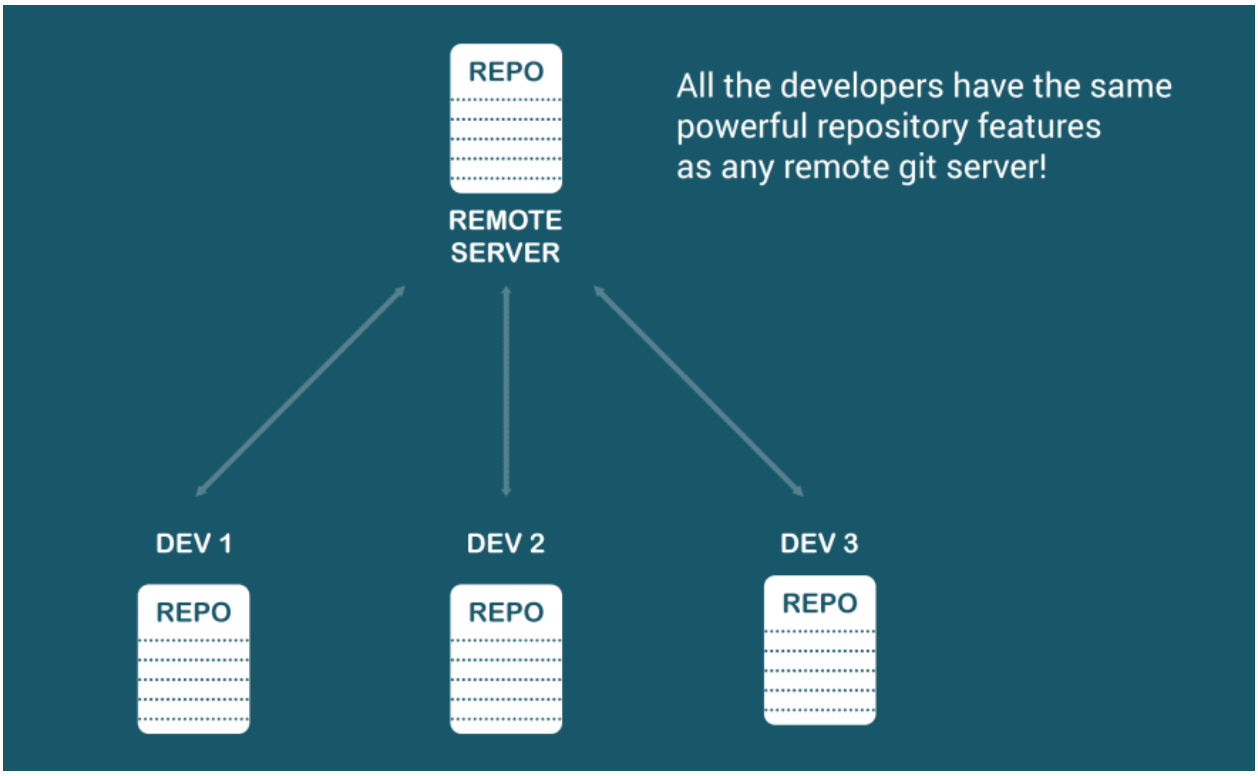
1.

 152  32  325 ■■■

search

This is different than other version control systems

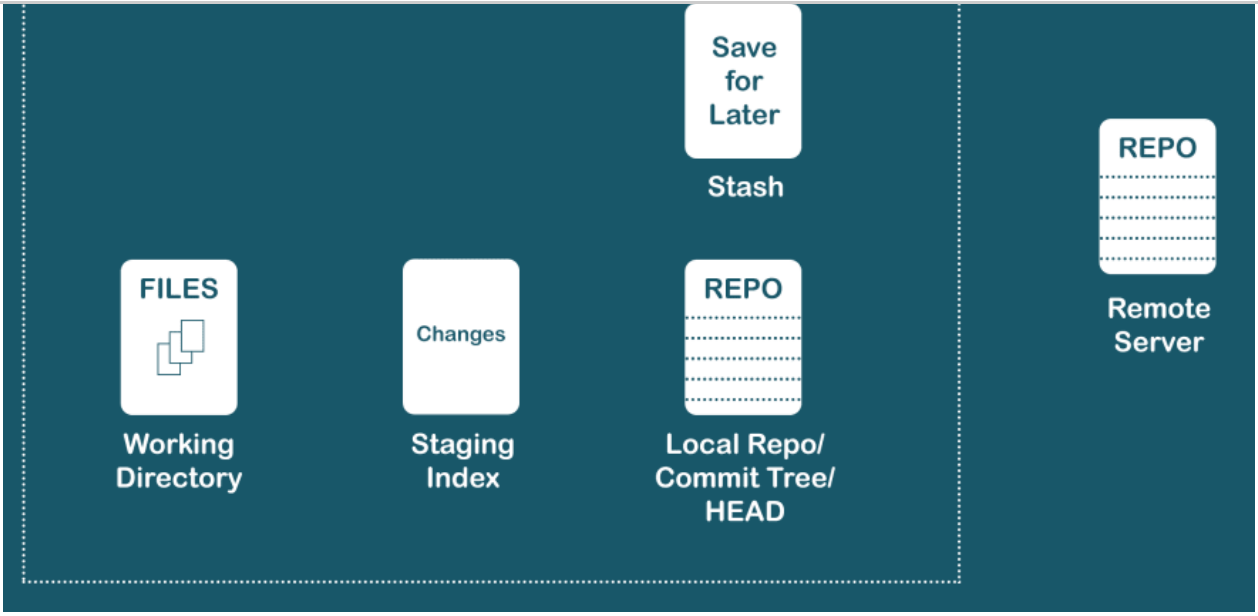
Once you embrace that, you can start demystifying some of the git 'magic'



2.

Think of files (and changes) as being in 5 different places, or "states"

- Working directory
- Staging (Index)
- Commit tree (local repo or HEAD)
- Stash
- Remote repo (github, Bitbucket, gitlab, etc)



3.

Think of moving files (or changes) between those places:

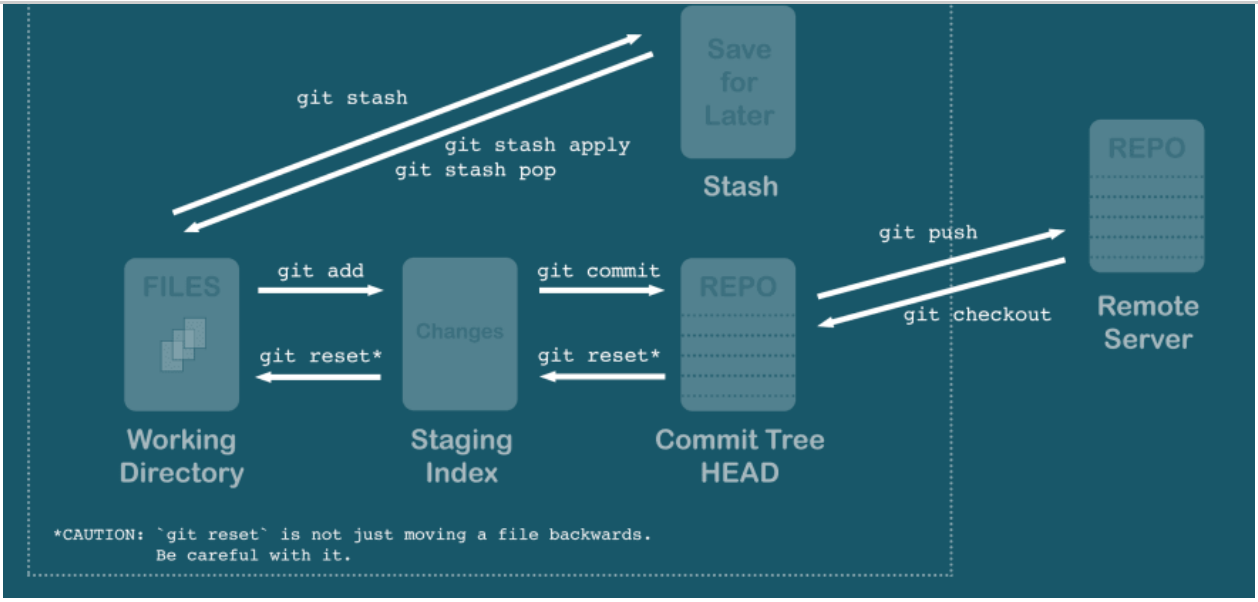
`git add` working dir => staging

`git commit` staging => HEAD

`git push` HEAD => remote repo

`git stash` working dir <=> stash

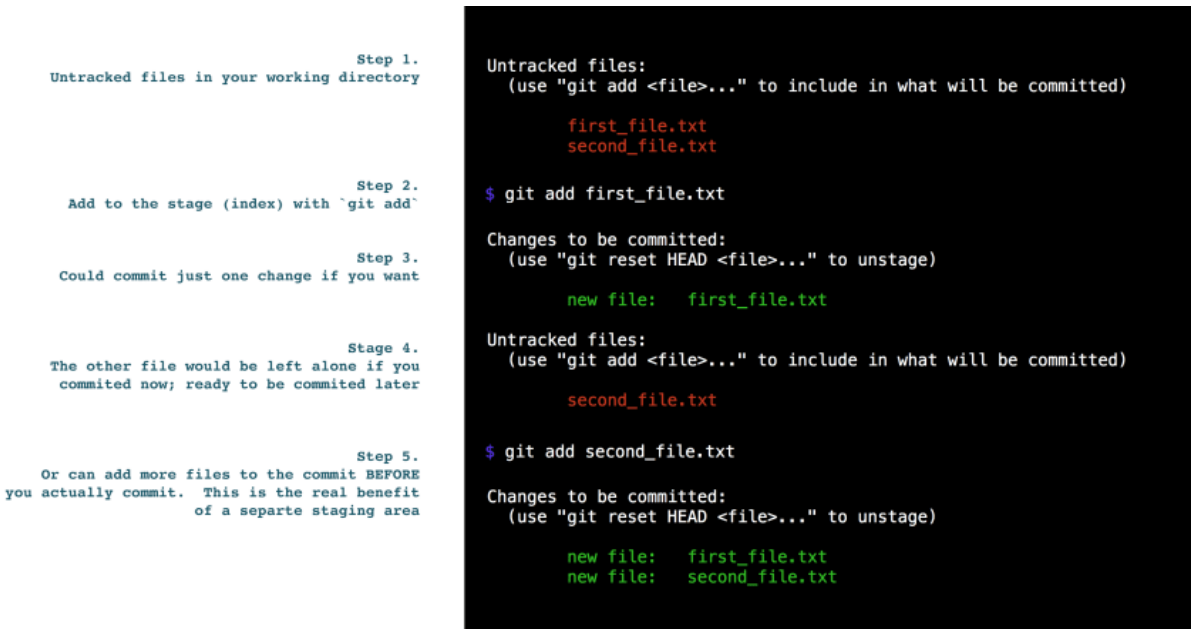
`git reset` and `git checkout` to pull from upstream



4.

Why have a dedicated staging area?

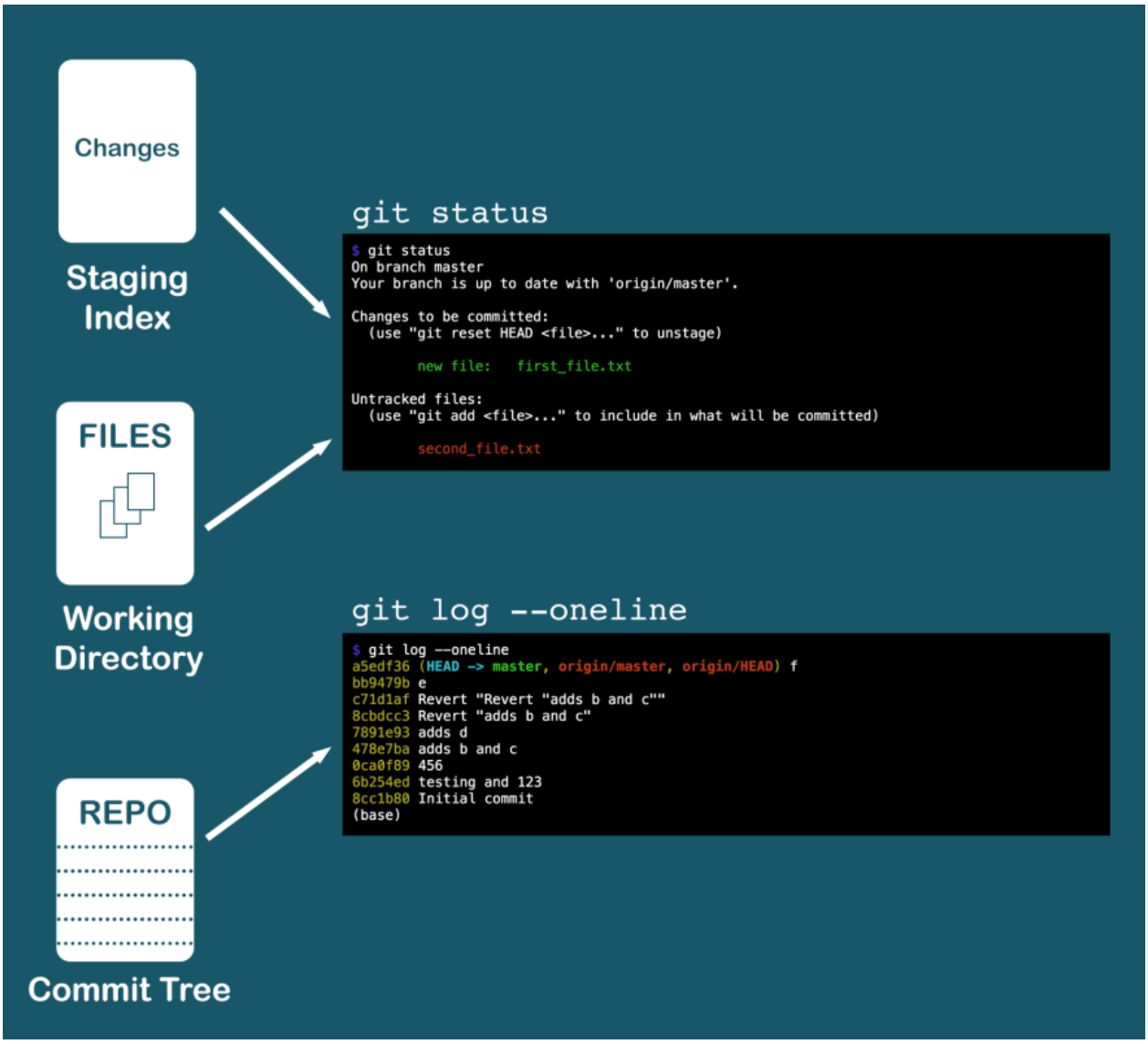
So that you can choose and review which files and changes to commit before committing.



5.

and staging, but think of them as separate things.

`git log` shows the history of commits your local repository



6.

Learn to love `git log`

It's a snapshot of repo state: shows past commits as well as local HEAD, local branch, remote HEAD and remote branch

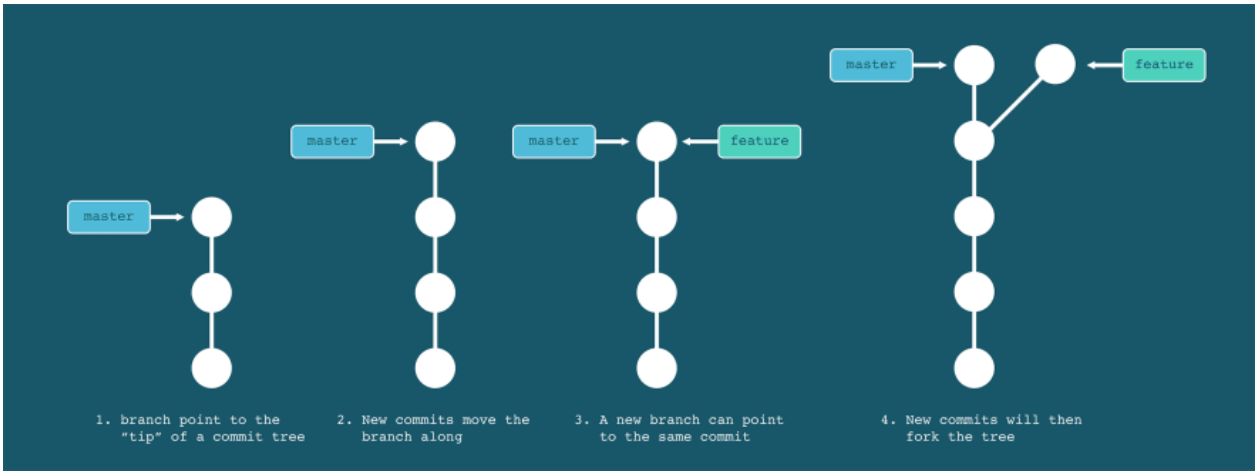
```
$ git log --oneline
a5edf36 (HEAD -> master, origin/master, origin/HEAD) f
bb9479b e
c71d1af Revert "Revert "adds b and c""
8cbdcc3 Revert "adds b and c"
7891e93 adds d
478e7ba adds b and c
0ca0f89 456
6b254ed testing and 123
8cc1b80 Initial commit
(base)
```

7.

A branch is a reference to the tip of a line of commits

It automatically updates when new commits are added to that line

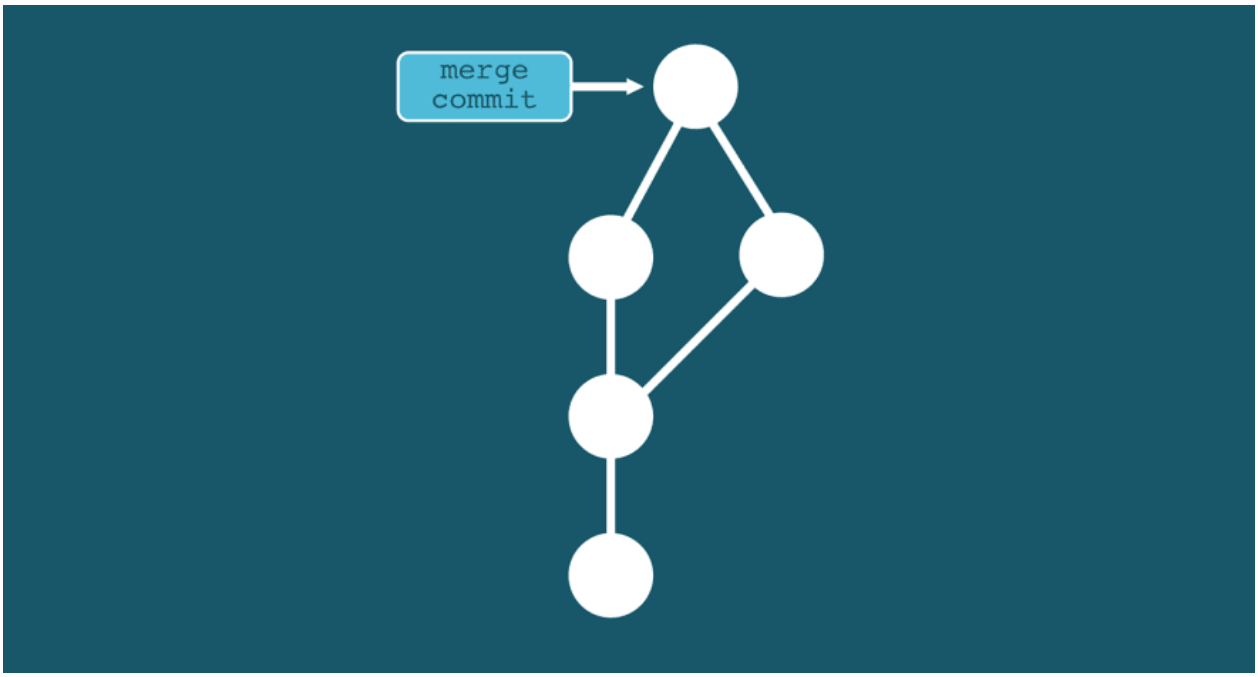
Making a new branch will diverge the tree at that point



8.

A merge takes two branches and makes a NEW commit which combines them

shortcuts!)



9.

`git rebase` lets you rewrite commit history

Applies your current commits directly to branch HEAD

Can squash all your commits into one to clean up history

Don't do this to public (remote) commits!

10.

Some people say you should only ever merge to keep your entire history

into master to keep a clean history tree

I say: do whatever works for you and your team

11.

HEAD can point to a branch or a specific commit

If it points to an old commit, that's called a "detached HEAD"

Editing in a detached HEAD state is dangerous (can lose work or cause problems combining work)

12.

Many git commands can operate on either: individual files, commits, or branches

This can cause a lot of confusion - so make sure you know what TYPE of object you're operating on

13.

There are many ways to undo unwanted actions in git
Here are the most common:

unstage a file: `git reset [file]`

undo local commit: `git reset [commit BEFORE the one to undo]`

undo remote commit: `git revert [commit to undo]`

14.

There's SO MUCH MORE I could have talked about!

What other things confuse you about git?

Comment below and I'll try to answer or find some resources for you 🙌

Like this crash course?

Follow me on twitter for more: @chrisachard

Or you can join the newsletter:
https://chrisachard.com/newsletter

Thanks for reading!

search

Chris Achard + FOLLOW

I'm trying to teach everything I know at chrisachard.com Instructor at egghead.io Mostly, I use JS, React, Rails, and Node

@chrisachard chrisachard chrisachard.com

Add to the discussion

PREVIEW SUBMIT

Ekanem

Sep 11

First time hearing about `git log --oneline` . A truly cleaner way to look at the history.

Thanks Chris

3 REPLY

Jérôme Gully

Sep 11

And there is much more... try

```
git log --all --decorate --oneline --graph
```

:)

3 REPLY

Chris Achard

Sep 11

Yes! there's a whole bunch :) Here's the full docs if you're interested: git-scm.com/docs/git-log (examples at the bottom)

1 REPLY

152 32 325

1 REPLY

James Bedford

Sep 11

On fire with these recent posts Chris!!

4 REPLY

Chris Achard

Sep 11

Thanks!

2 REPLY

Papidev

Sep 11

Love it, so clear

3 REPLY

Jérôme Gully

Sep 11

Best git summary ! Clear, concise, simple illustrations. I already master all these commands, but I never seen a so summary, so thanks. I will spread my team with your post :)

2 REPLY

marko

Sep 11

What is command when changes should be added under pushed commit?

2 REPLY

Chris Achard

Sep 11

If I understand correctly, what you want is to add the extra files to the staging area with:

152 32 325

search

and then you can add them to the most recent commit with

```
$ git commit --amend
```

Does that solve the issue?

1 REPLY

Blair Jersyer Sep 10

nice course...i'm however a bit confused about "stash".. first time i read this. What is that ?

2 REPLY

Chris Achard Sep 10

Thanks - and yeah, I'm realizing that I didn't explain stash... like at all in the post oops.

Stash is a temporary place you can put work in progress. Usually, the workflow goes something like this:

- You are working on something in your working directory
- a high priority bug comes in. To fix it, you have to switch branches and clear your working directory of the changes you already have made
- instead of trying to save your work for later in a commit or a special branch, etc, you can put it in the "stash" with `git stash`
- Then you go and fix the high priority bug
- later, you can re-apply what was in the stash with `git stash apply` or `git stash pop` (pop will remove it from the stash; apply just brings it back over)
- then you can continue to work on whatever you were working on from bullet 1.

Hope that helps a bit! I probably should have had a separate point for it in the guide

5 REPLY

152 32 325

Wow, thank you.

2 REPLY

AlexCodes Sep 10

Thank you!! I am currently learning git

2 REPLY

Chris Achard Sep 10

Glad it helped!

1 REPLY

Jason Sep 10

I was really getting confused, till I realized I was seeing the word 'comment' instead of 'commit'. Now it makes much more sense! Great article.

2 REPLY

Chris Achard Sep 10

1 REPLY

Jasterix Sep 10

Great article! I also recommend the atlassian git docs, which helped me get comfortable with git

2 REPLY

152 32 325

search

Git is the expression used for an old angry man.

Either git good, or git' out.

Either way someone is going to be frustrated.

1 REPLY

Meierreoi Sep 11

Just find something useful in this post, thanks.

3 REPLY

code of conduct - report abuse

Classic DEV Post from Apr 15

The Many Flavors of Technical Presentations

Jeremy Likness

Learn about various technical presentation formats to choose the best one for you

62

Another Post You Might Like

How to build an admin panel with React

Brian Neville-O'Neill

Introduction A good number of web applications have evolved from stati...

151 4

152 32 325

search

Best DEV.to Posts for Beginners: Week of May 26, 2019

Desi

These are some of the best beginner posts of the week!

58 2

Add to a Dictionary in D

Jesse Phillips - Sep 10

Never fail your type checking ever again

Amin NAIRI - Sep 10

Getting a job at Google after coding bootcamp

Matt Upham - Sep 10

PostgreSQL: How To Update & Delete Data

miku86 - Sep 10

152 32 325