

Git for ~~fun and~~ profit

Andy Boughton / [Center for Statistical Genetics](#)

July 14, 2017

git-interpret-stash(1) Manual Page

NAME

`git-interpret-stash` — interpret various staged stashes

SYNOPSIS

`git-interpret-stash` [`--collide-area` | `--prep`

DESCRIPTION

`git-interpret-stash` interprets a few non-indexed unstaged subtrees to many remotes, you can fail the ref of the stages.

(no, not really)

<https://git-man-page-generator.lokaltogether.net/>

“How can I delete a remote?”
“git remote rm”, replied Master Git.
“And how can I delete a branch?”
“git branch -d”, replied Master Git.

<http://stevelosh.com/blog/2013/04/git-koans/>

Lots of power

But running new commands is scary

git reflog









Your safety net

Reference logs, or "reflogs", record when the tips of branches and other references were updated in the local repository


```
def perfect_code():  
    return True
```

```
$ git init  
$ git add .  
$ git commit -m "O(1) sort algorithm"  
  
$ git log --oneline  
2e5b760 O(1) sort algorithm
```



```
def perfect_code():  
    """Documentation makes it even better!"""  
    return False
```

```
$ git add .  
$ git commit --amend
```



```
$ git log --oneline
d14023e O(1) sort algorithm
$ cat test.py
def perfect_code():
    """Documentation makes it even better!"""
    return False
```


Uh-oh

...Did I just lose the good version?


```
$ git reflog
d14023e HEAD@{0}: commit (amend): O(1) sort algorithm
2e5b760 HEAD@{1}: commit (initial): O(1) sort algorithm
$ git checkout 2e5b760
Previous HEAD position was d14023e... O(1) sort algorithm
HEAD is now at 2e5b760... O(1) sort algorithm
$ cat test.py
def perfect_code():
    return True
```


Summary

- You don't "lose" changes- even when you rewrite git history
- There is a copy of the change if you know where to look
- This makes it less scary to try new things

Things with reflogs

Git stash, change between branches, rebase, merges,
etc

Caveats

- Only works for changes Git knows about (unsaved files are still lost)
- Your reflog doesn't get pushed to GitHub- for your computer only^{*}
- The reflog is not permanent, so don't forever
 - ^{*}...Though GitHub has its own reflog, [via API](#)

git bisect

...When did that bug come in?



mariellevolz @mariellevolz · 1 May 2014

Using git **bisect** is like talking to my toddler. Git **bisect** bad! Git **bisect** bad!
Drawing on wall bad! Drawing on paper good!



1



8

*git-bisect - Use binary search to find the
commit that introduced a bug*

See: <http://webchick.net/node/99>

Caveats

- It's not magic
- Your job to find the last time something worked
- You define what "good" means- unit tests can help!

References

Git Gone Wild- SEM JS 2016

<https://www.youtube.com/watch?v=8uZF1qD-gd4>

How to undo (almost) anything with Git -

<https://github.com/blog/2019-how-to-undo-almost-anything-with-git>

