

INTRO TO GIT

MPLS JR DEVS

♥ February 13, 2018 ♥

Joshua Wehner
@jaw6

Git

command-line version control
git-scm.com

April 2005

GITHUB

Git hosting plus "social" features
www.github.com

February 2008

GIT

PROS

Fast

Offline

Decentralized

Widespread use

CONS

User-hostile interface

GITHUB

PROS

CONS

User-friendly

Centralized + online

Widespread use

TERMS AND CONCEPTS

REPOSITORY



TERMS AND CONCEPTS

COMMIT



TERMS AND CONCEPTS

BRANCH



TERMS AND CONCEPTS

MERGE



TERMS AND CONCEPTS

PULL REQUEST



TERMS AND CONCEPTS

HEAD



TERMS AND CONCEPTS

FORK



UNDO #1: AFTER PUSHING

**It's broken
... and it's been pushed**

```
$ git revert <SHA>
```

UNDO #2: NO REGERTS

**Bad commit message
... caught before push**

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

UNDO #3: IT'S NOT WORKING

Go back to what worked

```
$ git checkout -- <filename>
```

!!!

UNDO #4: WHOOOOPS

Bad commits are bad

```
$ git reset <last good SHA>
```


UNDO #5: THROW IT ALL AWAY

It's all rubbish

```
$ git reset --hard <last good SHA>
```

UNDO #6: ONE OF US

That one? That one was goooooood

\$ git cherry-pick <GOOD SHA>

UNDO #7: ON SECOND THOUGHT

Could I maybe get those commits back?

```
$ git reflog
```

THEN

```
$ git reset --hard <SHA>
```

OR

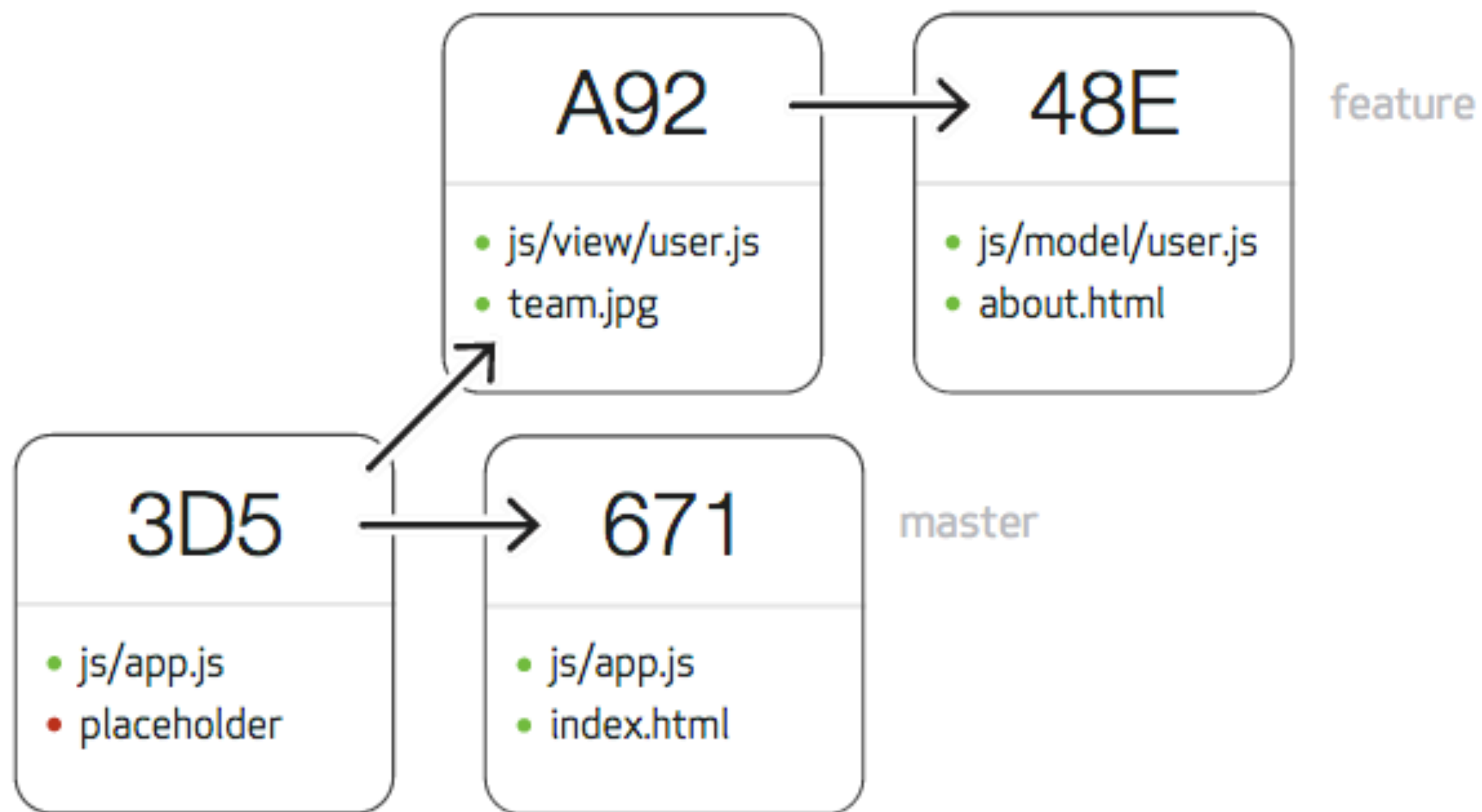
```
$ git reset cherry-pick <SHA>
```

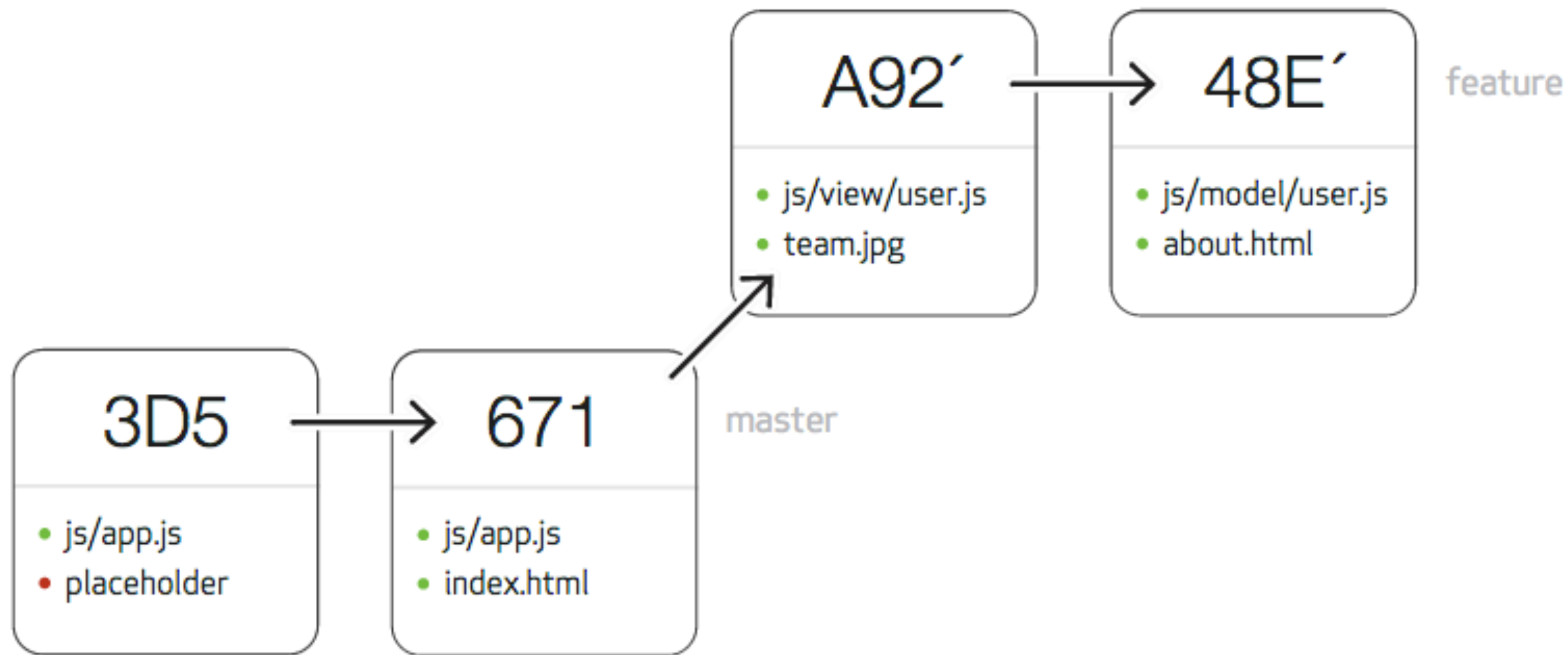
UNDO #8: BRANCH DIFFERENTER

Those commits, on this branch

```
$ git checkout <this branch>
```

```
$ git rebase <other branch>
```





⚡ UNDO: JEDI MASTER ⚡



```
$ git rebase -i <good SHA>
```

THANK YOU

Intro to Git @ MPLS JR DEVS

Joshua Wehner
@jaw6