

Demystifying Git

Adam Matan, SW Team Leader, Wix Code





```
> whoami
```

```
Adam Matan
```



Back in 2013
I started learning Git.

2013




Stick to
add,
commit,
push, and
pull

2013

Something
goes wrong

Stick to **add**,
commit, **push**,
and **pull**





You are in 'detached HEAD' state.

You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by performing another checkout.

2013



Something
goes wrong

Stick to **add**,
commit, **push**,
and **pull**

Found
stackoverflow
magic

2013

Stick to **add**,
commit, **push**,
and **pull**

Something
goes wrong

Found
stackoverflow
magic

Read the
manual





git-reset

Reset current HEAD to the specified state



git-reflog

Manage reflog information

2013

Stick to **add**,
commit, **push**,
and **pull**

Something
goes wrong

Found
stackoverflow
magic

Read the
manual

Trial and
error



A conceptual image featuring two hands reaching towards each other against a solid light blue background. The hand on the left is positioned higher and further from the center, while the hand on the right is lower and closer to the center. Both hands are oriented with their index fingers pointing towards the other hand, creating a sense of tension or a spark. The skin tone of the hands is light. The word 'Power' is written in black, sans-serif font, slanted upwards, next to the left hand. The word 'language' is written in the same style, slanted downwards, next to the right hand.

Power

language

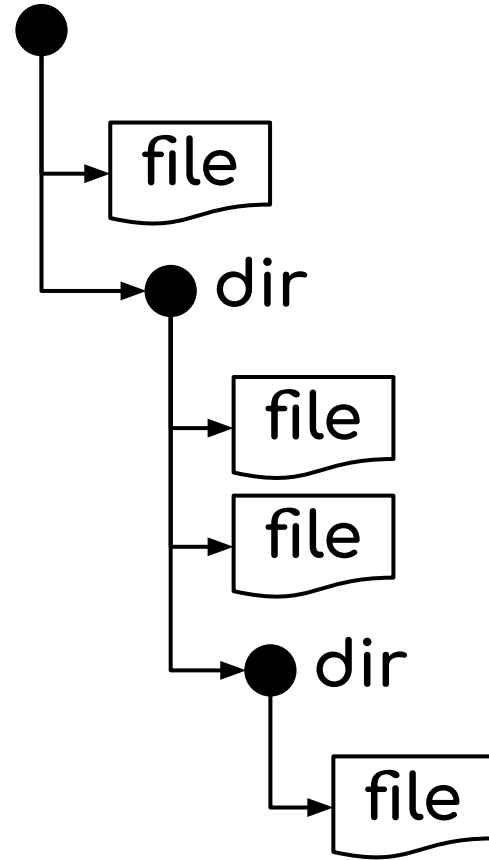
Git is...

Trees

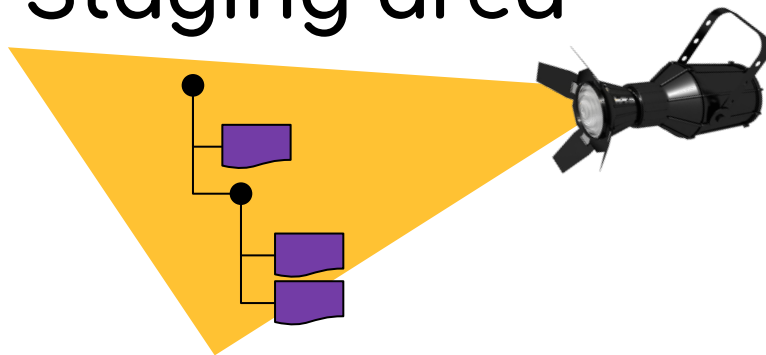
Objects

Pointers

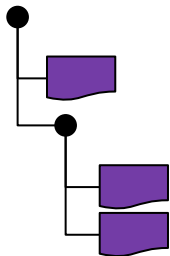
Trees



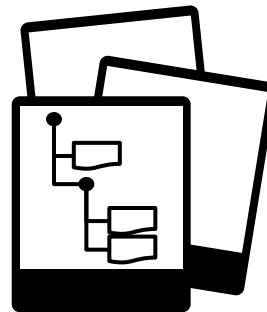
Staging area

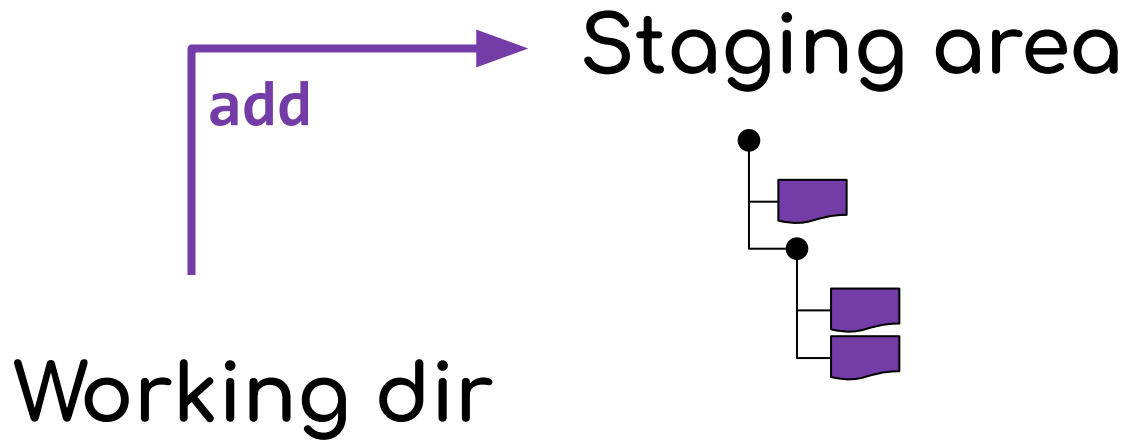


Working dir

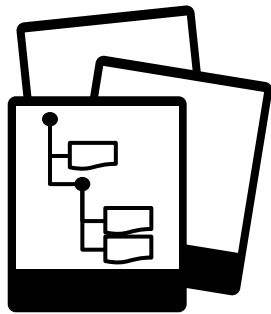


Commits





Commits



Trees

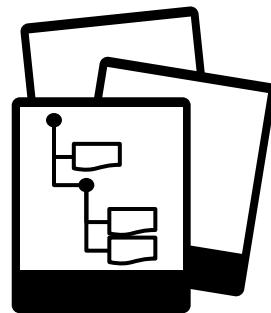
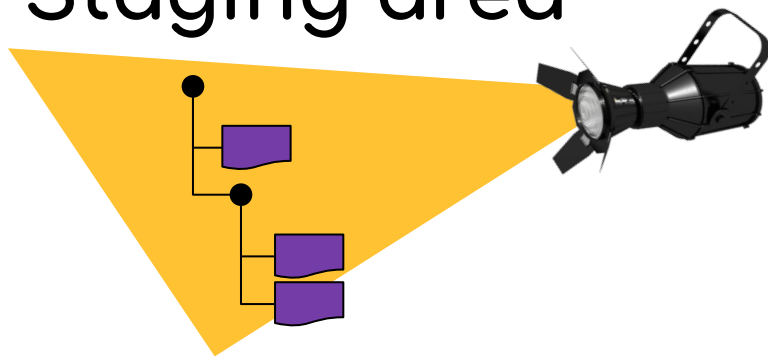
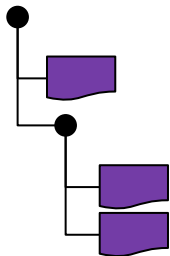
Staging area

add

commit

Working dir

Commits



Trees

Staging area

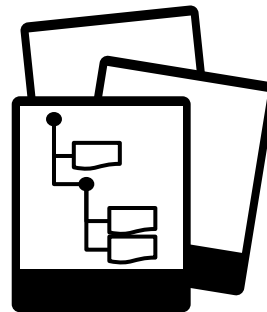
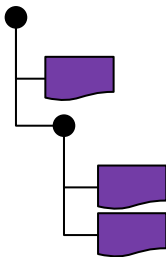
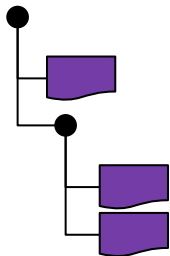
add

commit

Working dir

Commits

checkout

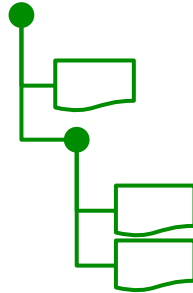


Objects

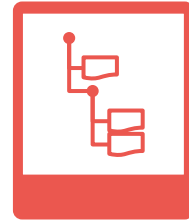
File (blob)



Tree



Commit



Objects

Are kept as compressed files.

File name is

`.git/objects/833592c9df1b118af602de54` SHA-1 hash

`.git/objects/5c60166876461ad757d80d3e`

`.git/objects/d5184ef704284477e24fc3c0`

`.git/objects/eab683a4a58ee4f9b2d917d0`

`.git/objects/cc7c6b3cd7b6512eb9724a49`

`.git/objects/e61ef7b965e17c62ca23b6ff`

`.git/objects/e898f7071923b09364a48f00`

File name is
.git/objects/833592c9df1b118af602de54 SHA-1 hash





```
.git/objects/833592c
```

```
# git cat-file -p 833592c
```

See a **file's** content



```
.git/objects/833592c
```

```
# git cat-file -p 833592c
```

```
#!/usr/bin/env python3
```

```
print("Hello!")
```

a file's content



```
.git/objects/cc7c6b3
```

```
# git cat-file -p cc7c6b3
```

See a **tree's** content



```
.git/objects/cc7c6b3
```

```
# git cat-file -p cc7c6b3
```

```
100644 blob 833592c code.py
100644 blob 146de1c README.md
040000 tree ab89a4d config
```

a **tree's** content



```
.git/objects/e898f70
```

```
# git cat-file -p e898f70
```

See a **commit's** content



```
.git/objects/e898f70
```

```
# git cat-file -p e898f70
```

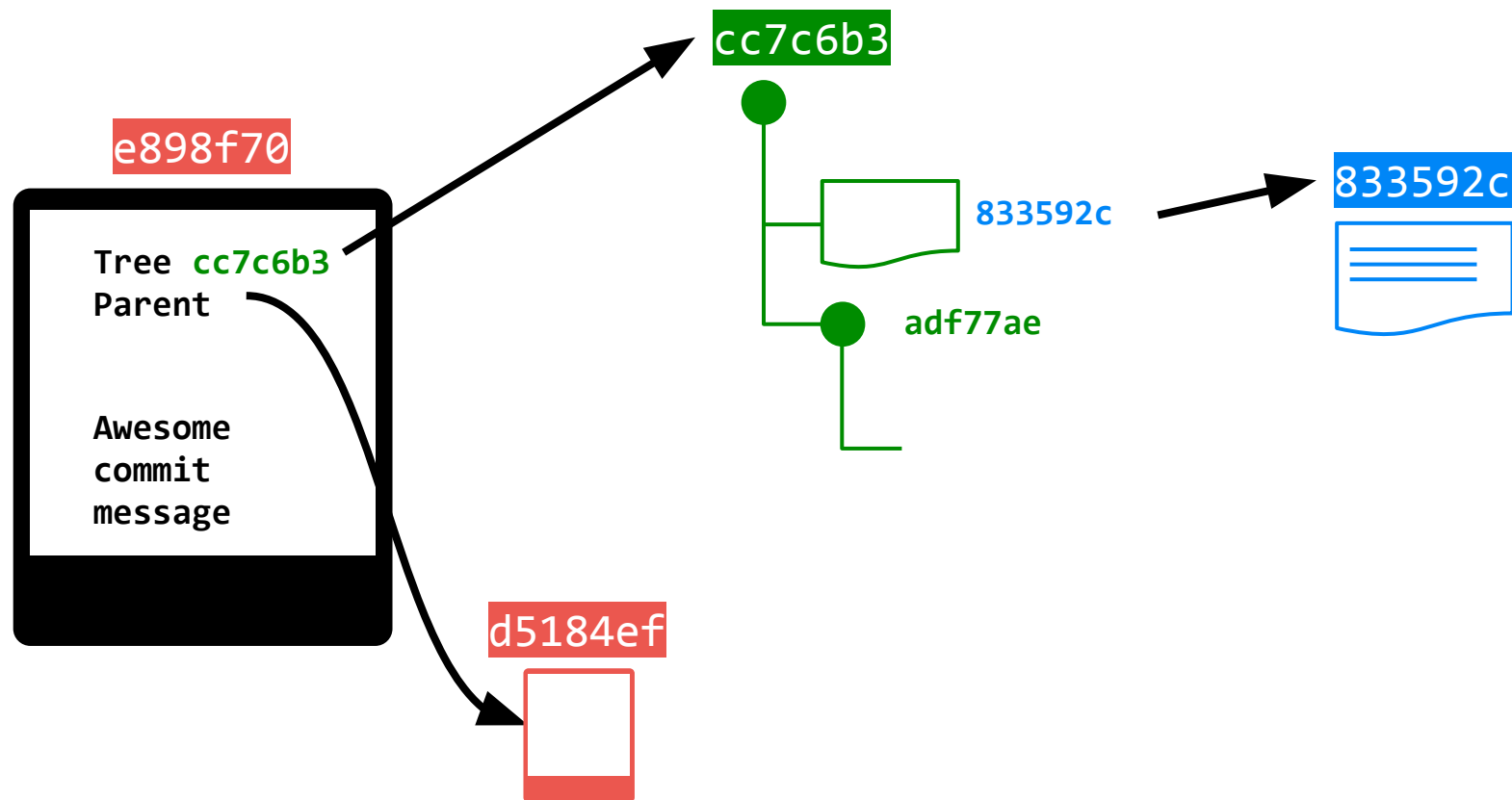
```
tree cc7c6b3
```

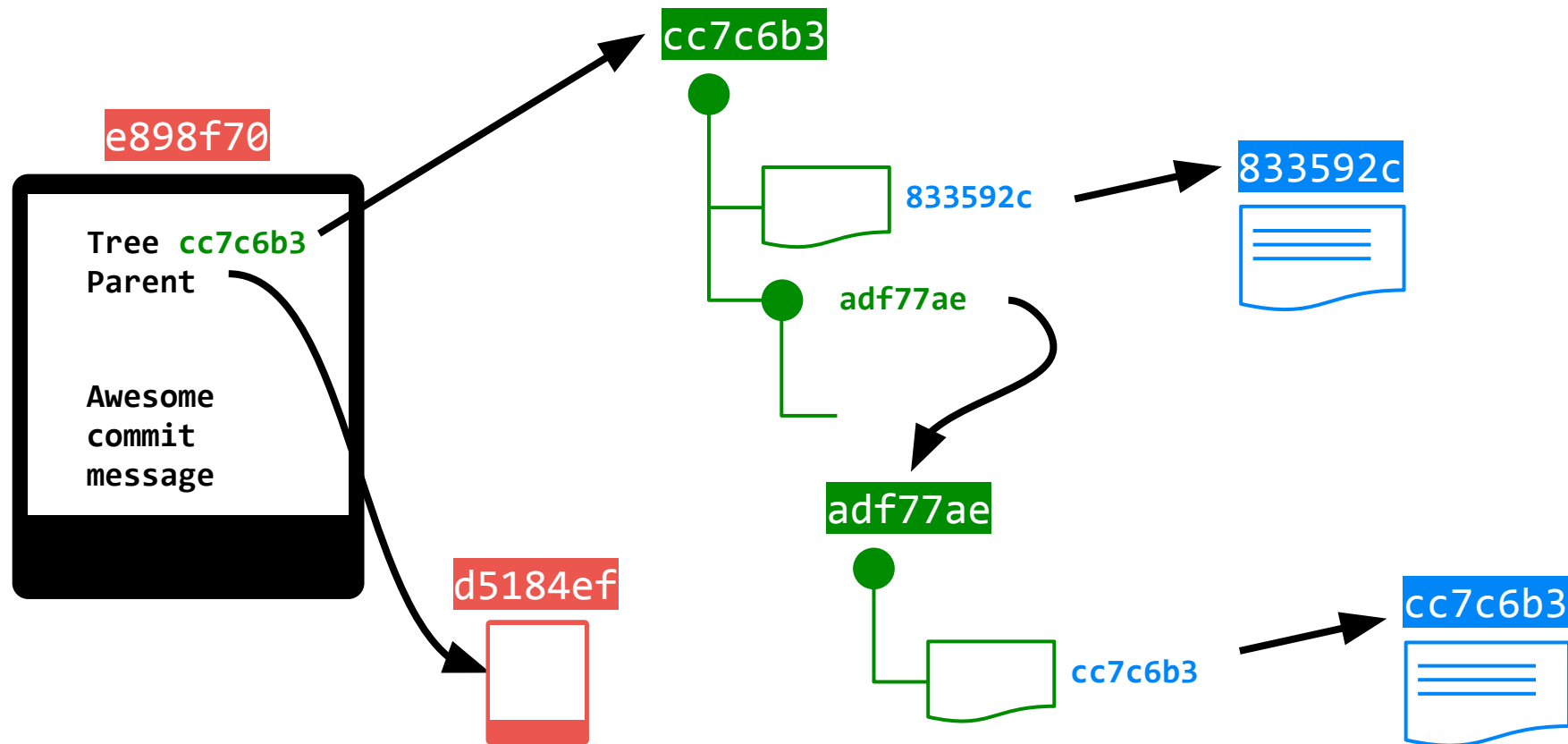
```
parent d5184ef
```

```
author Adam Matan 1557778029
```

```
committer Adam Matan 1557778029
```

```
Awesome commit message
```





Pointers

HEAD points
to the
current
branch

Branch points
to the current
commit



The HEAD content



```
# cat .git/HEAD  
ref: refs/heads/master
```


The HEAD content



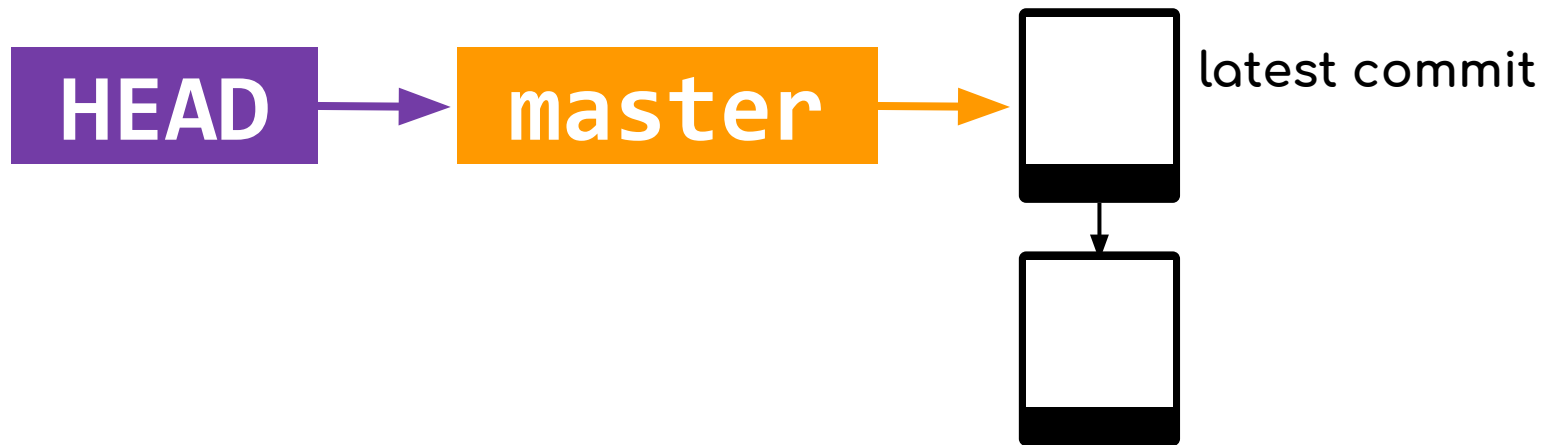
The branch content

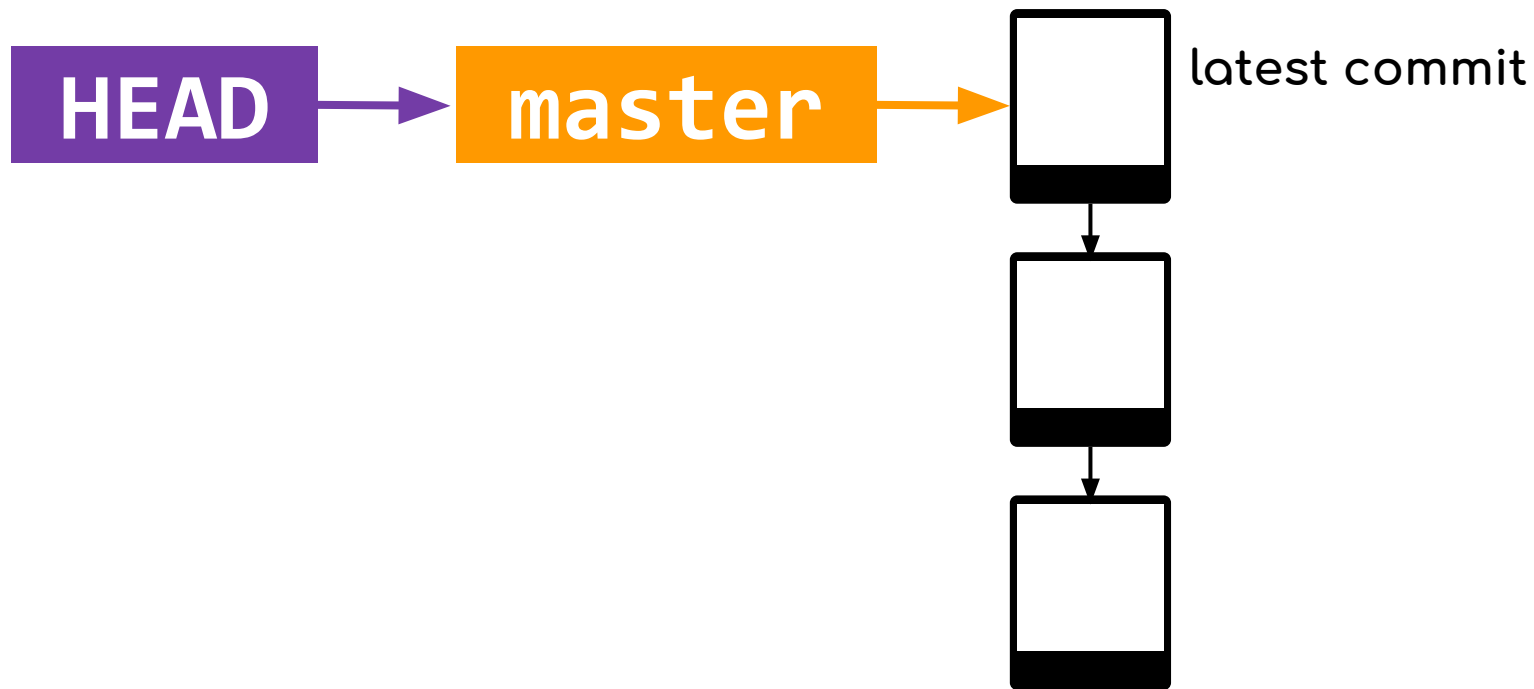


```
# cat .git/HEAD  
ref: refs/heads/master
```

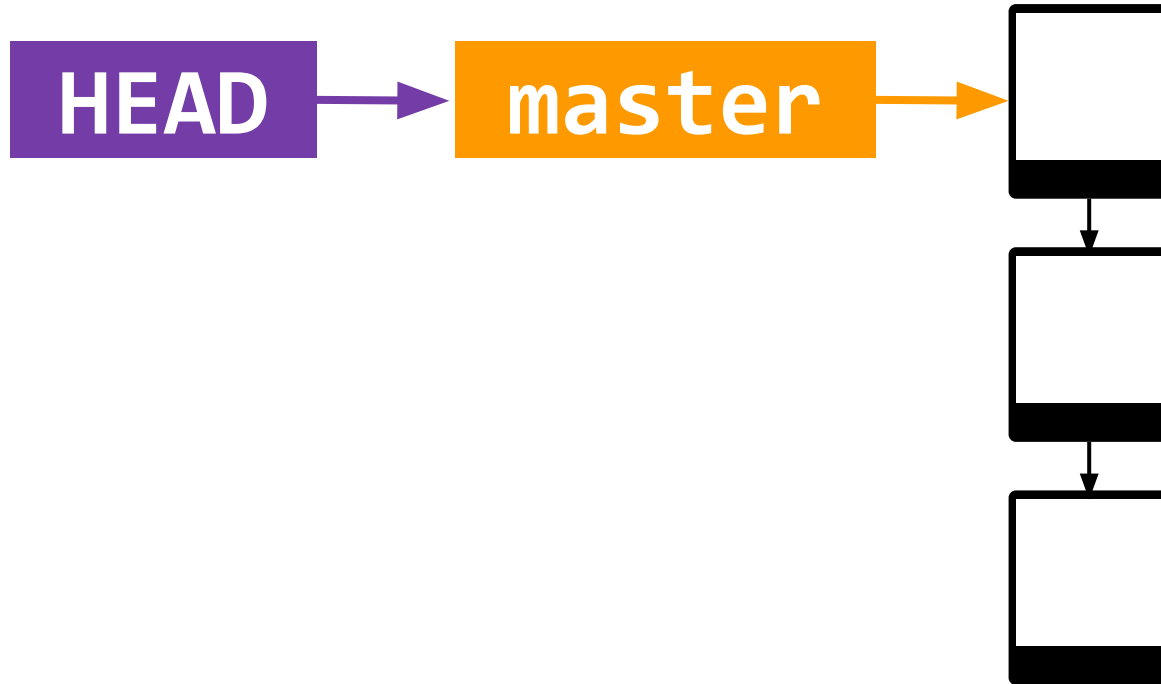
```
# cat .git/refs/heads/master  
e898f70
```







This is an attached HEAD.



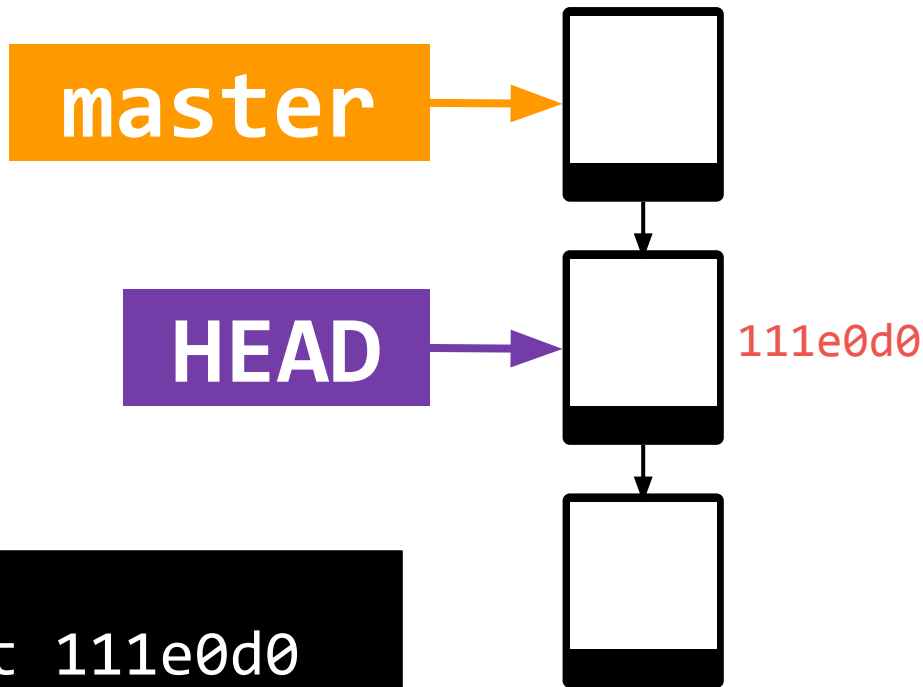


You are in 'detached HEAD' state.

You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by performing another checkout.

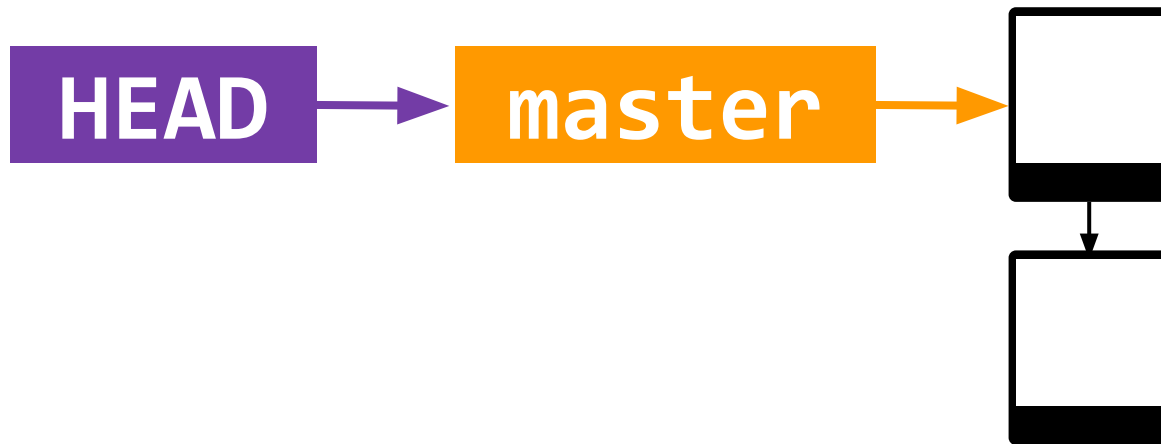


This is a detached HEAD.

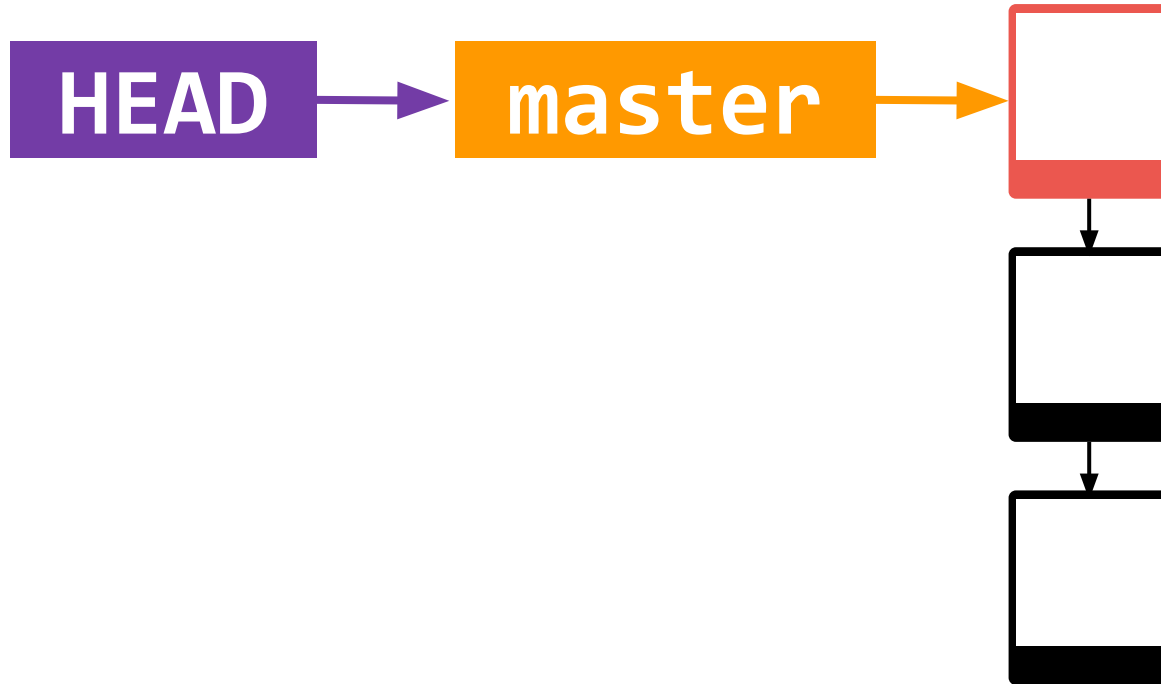


```
git checkout 111e0d0
```

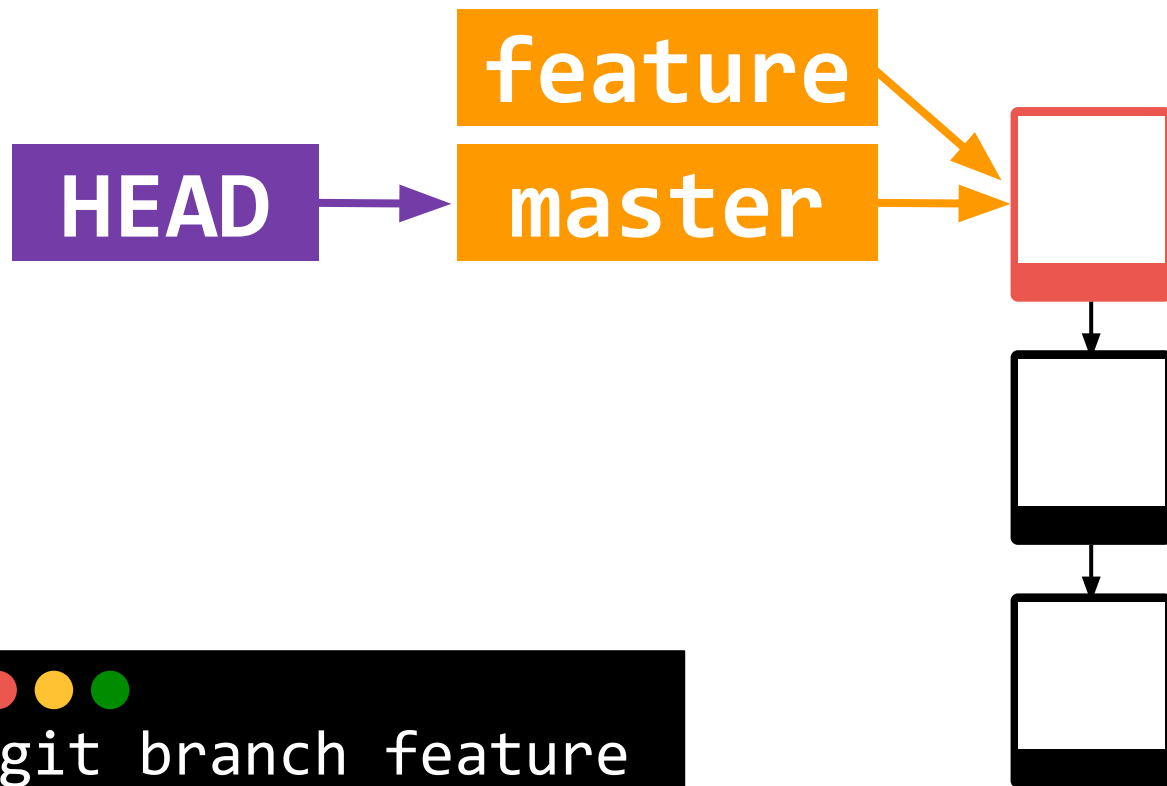
Accidental commit to master



Accidental commit to master



Accidental commit to master

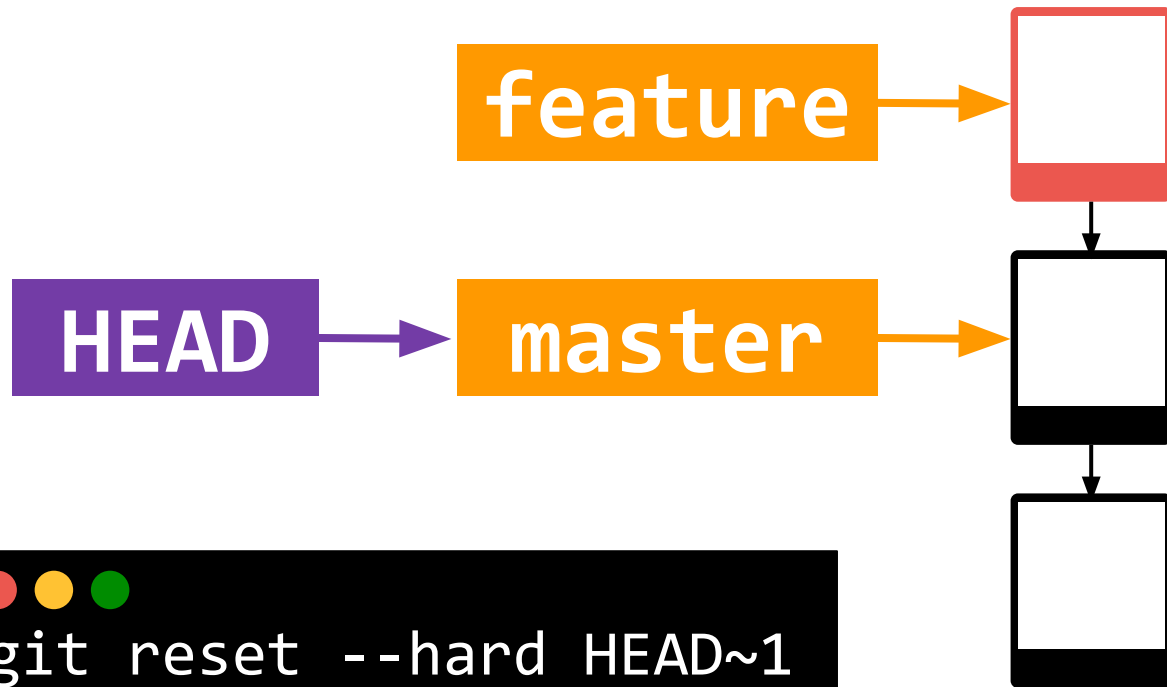




git-reset

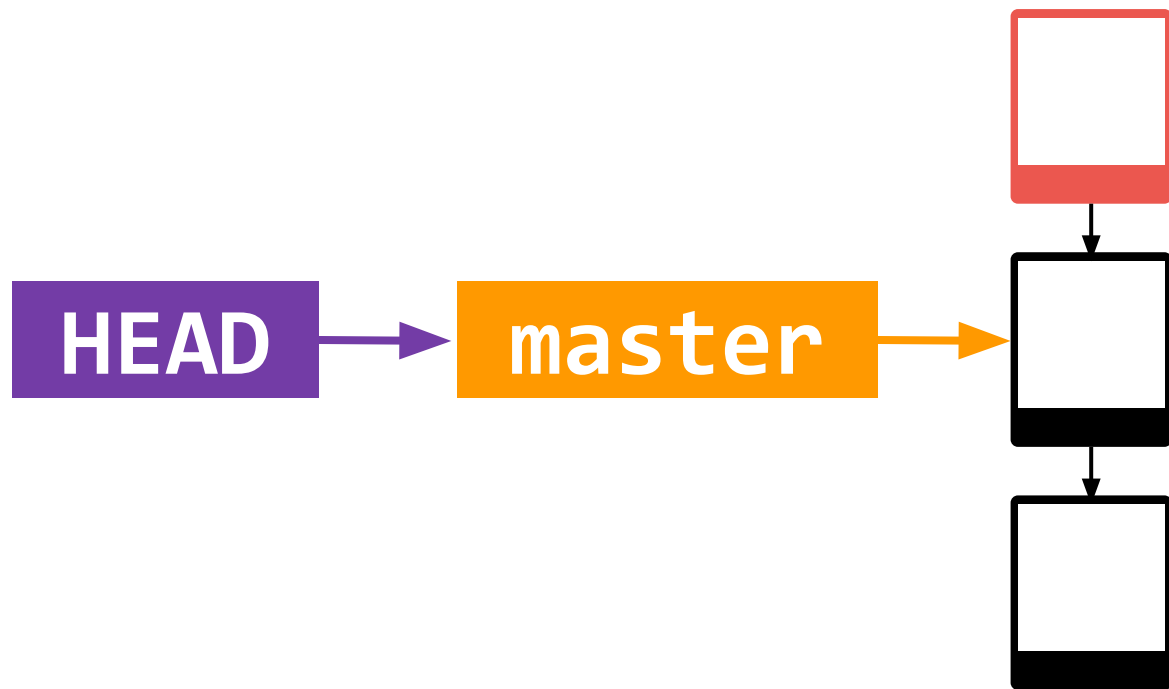
Reset current HEAD to the specified state

Accidental commit to master



```
git reset --hard HEAD~1
```

Lost commit





git-reflog

Manage reflog information

Git - reflog

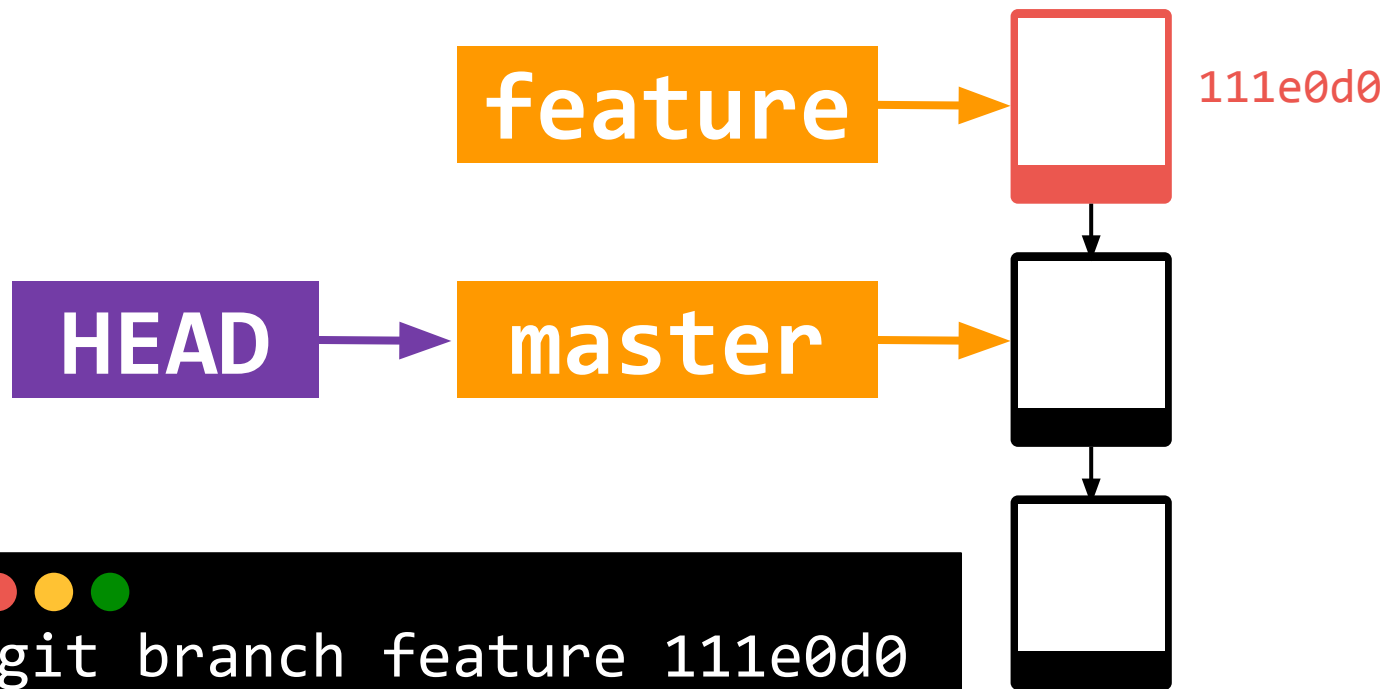


19d10e7 HEAD@{0}: commit: ...

111e0d0 HEAD@{1}: commit: moving...

6a40b7b HEAD@{2}: clone: from...

Lost commit



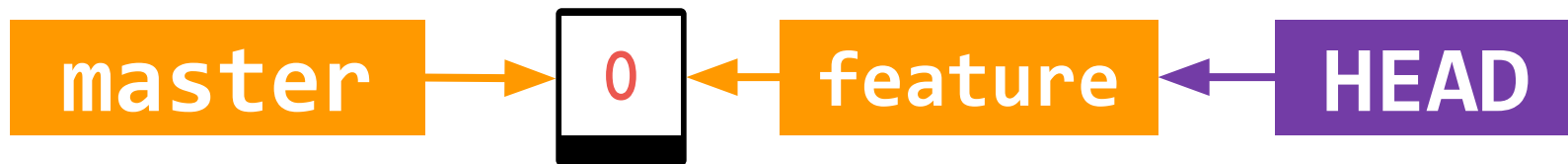
git branch feature 111e0d0

Now let's explain merge & rebase.

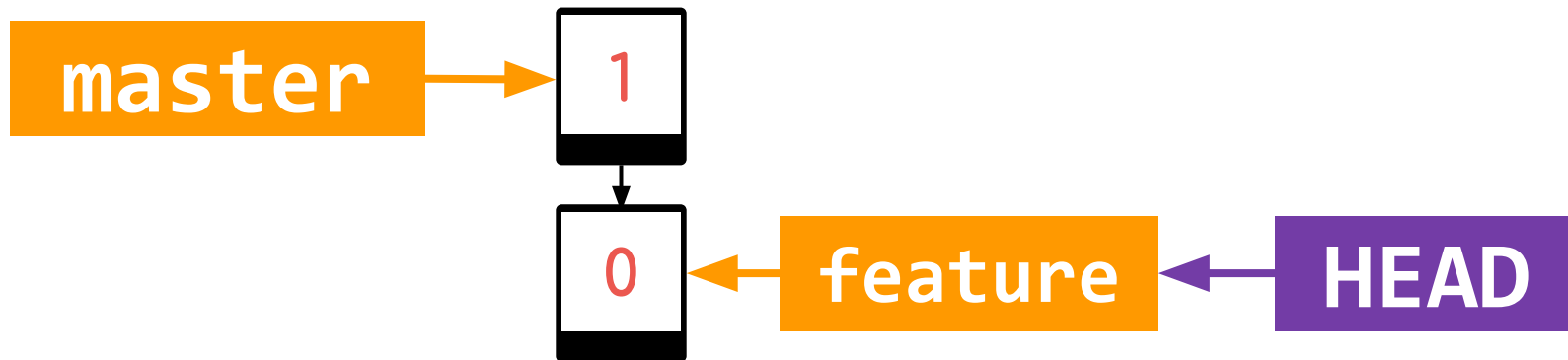




```
git checkout -b feature
```

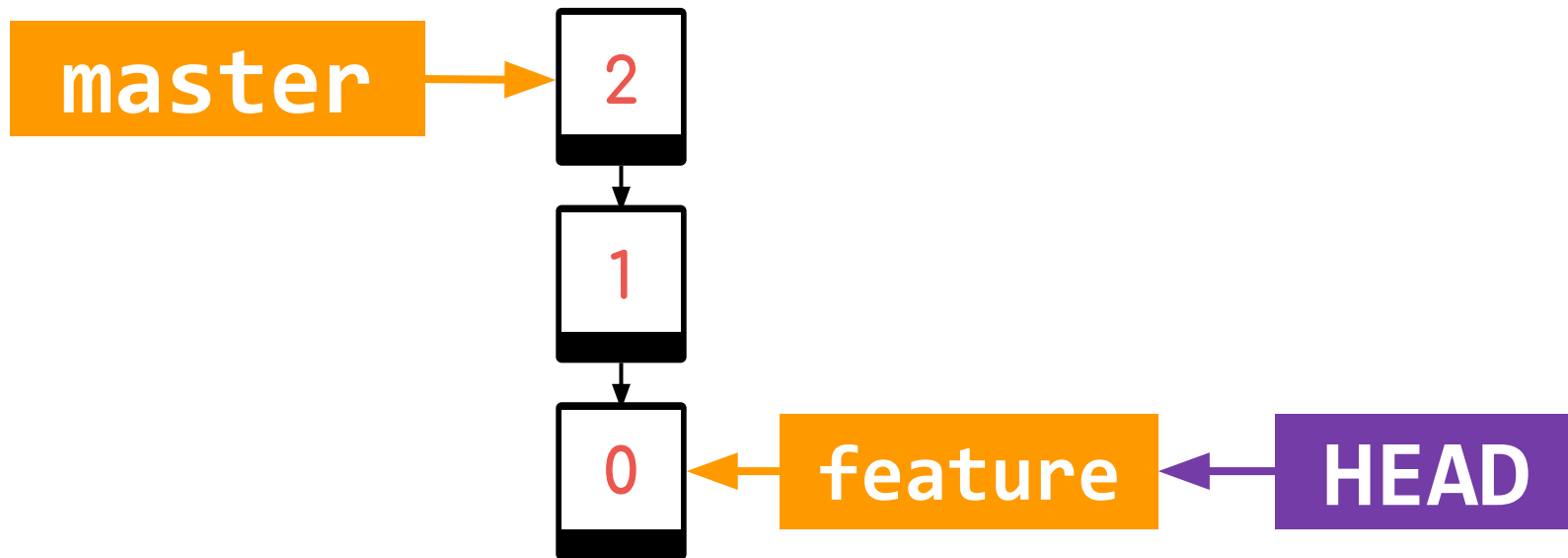


```
git commit -m message
```

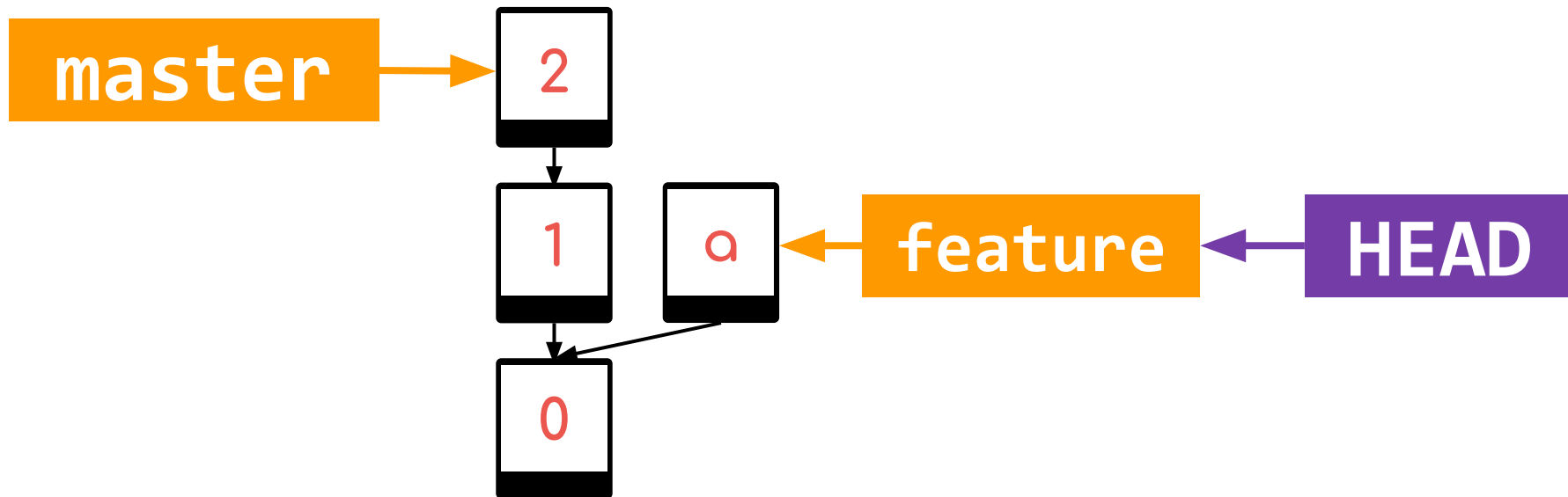




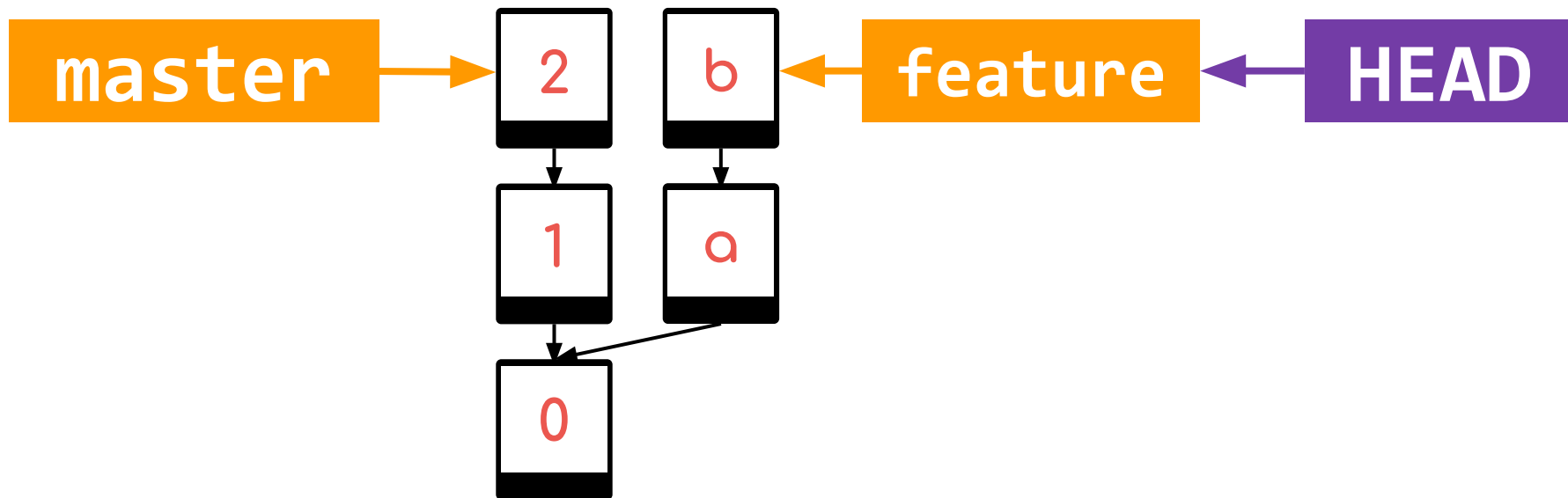
```
git commit -m message
```



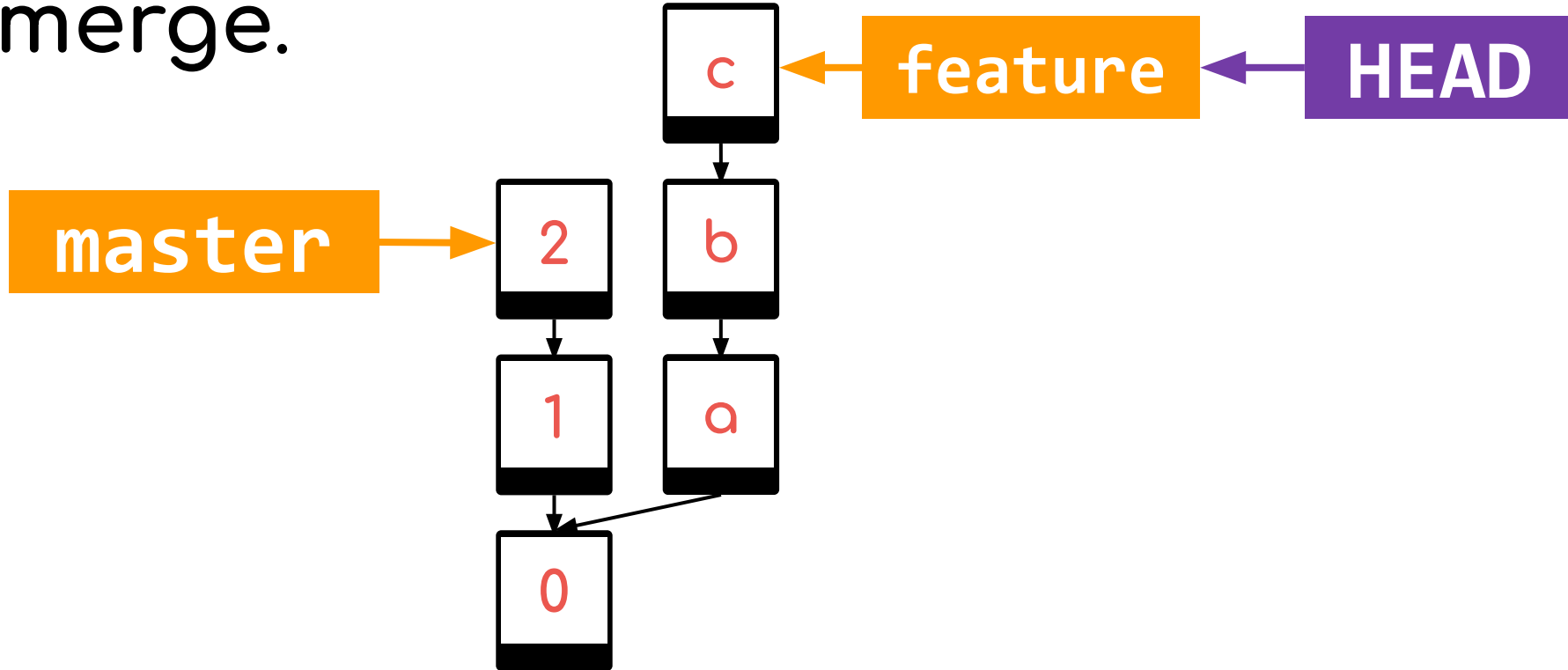
```
git commit -m message
```

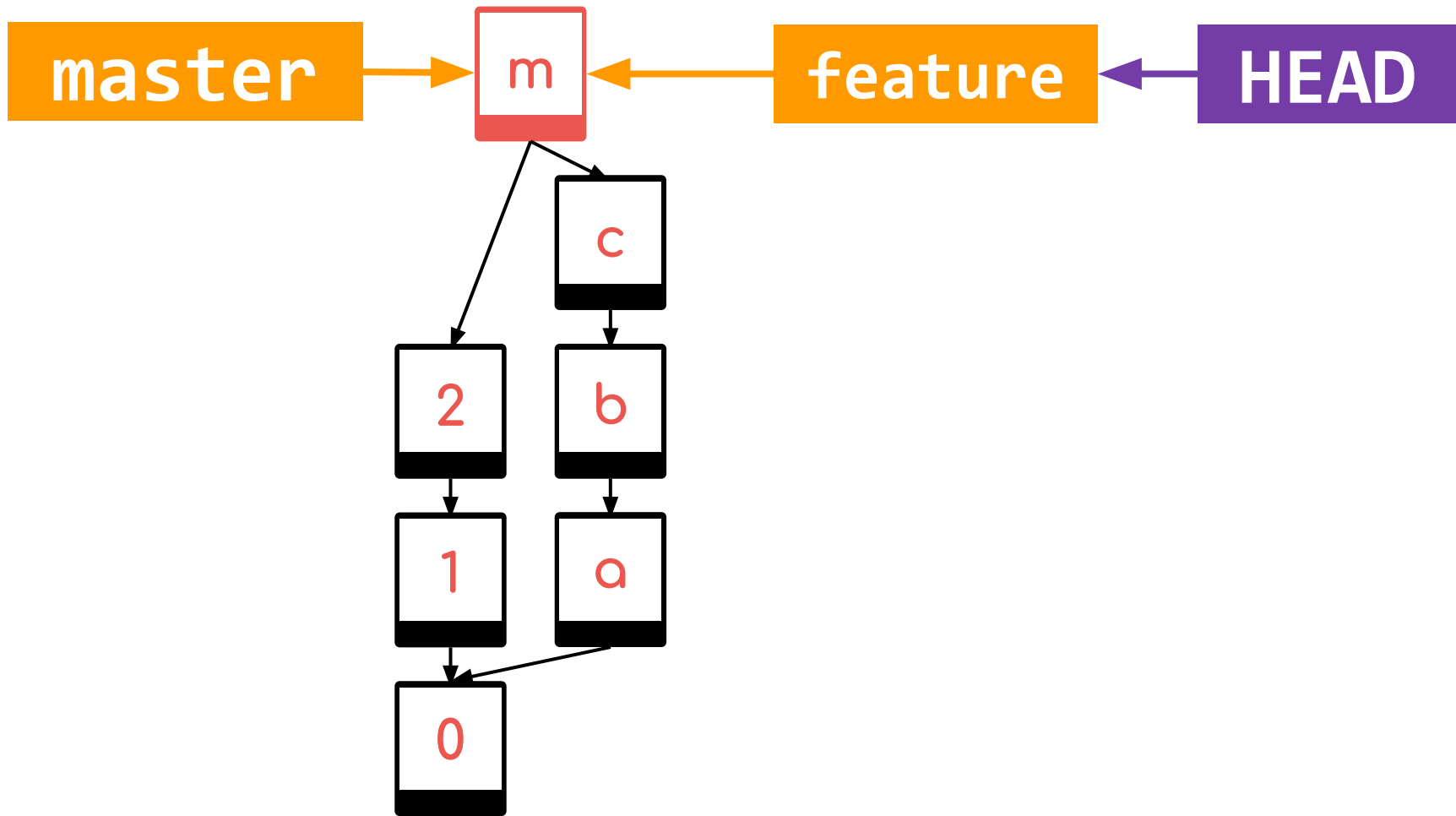


```
git commit -m message
```

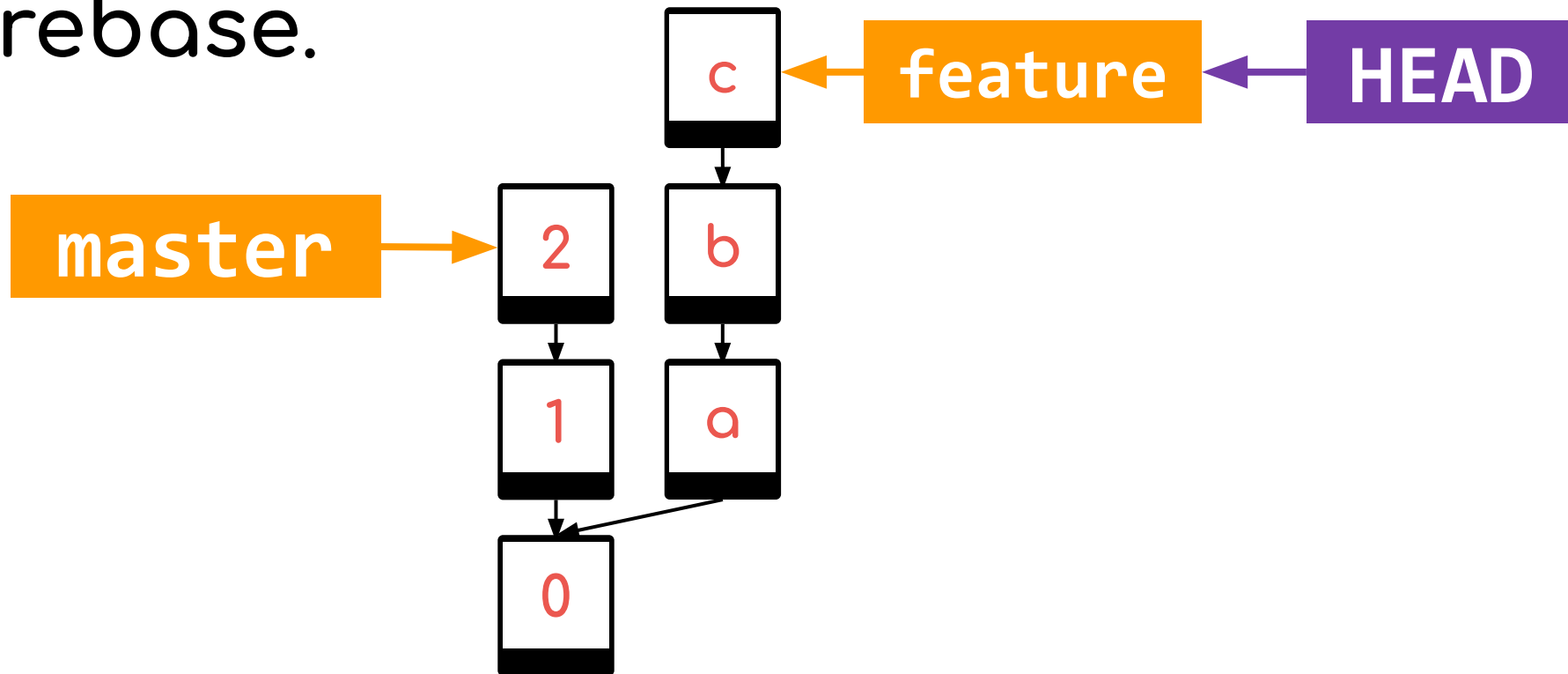


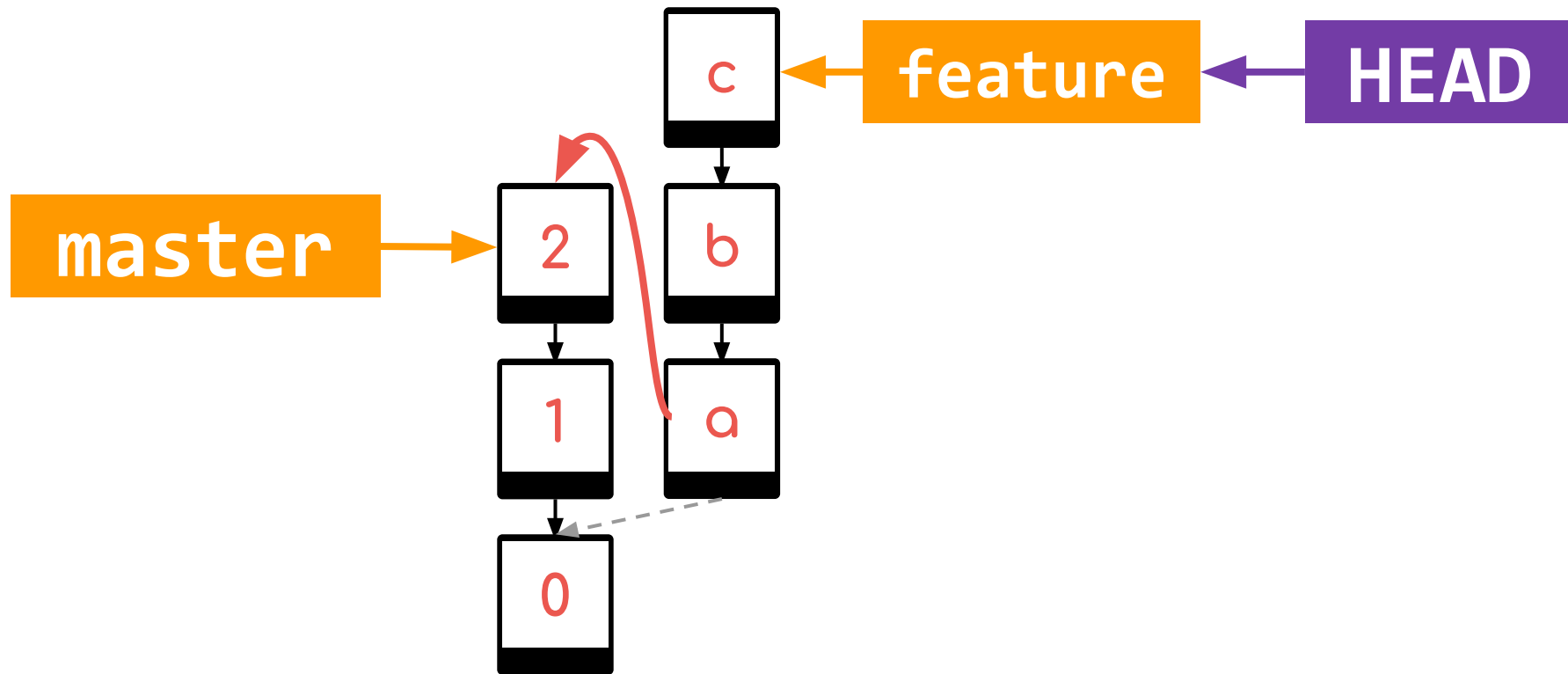
Let's
merge.

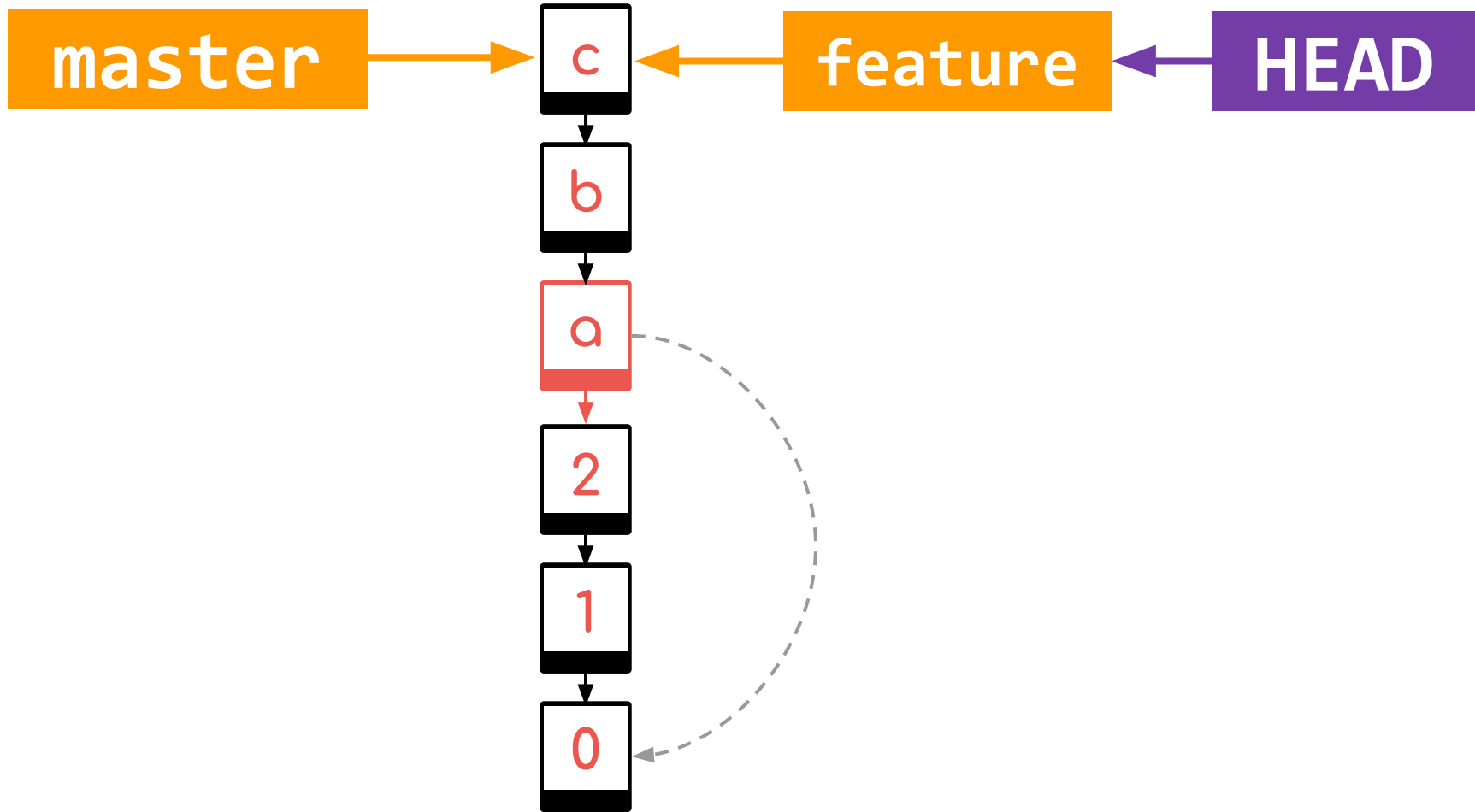


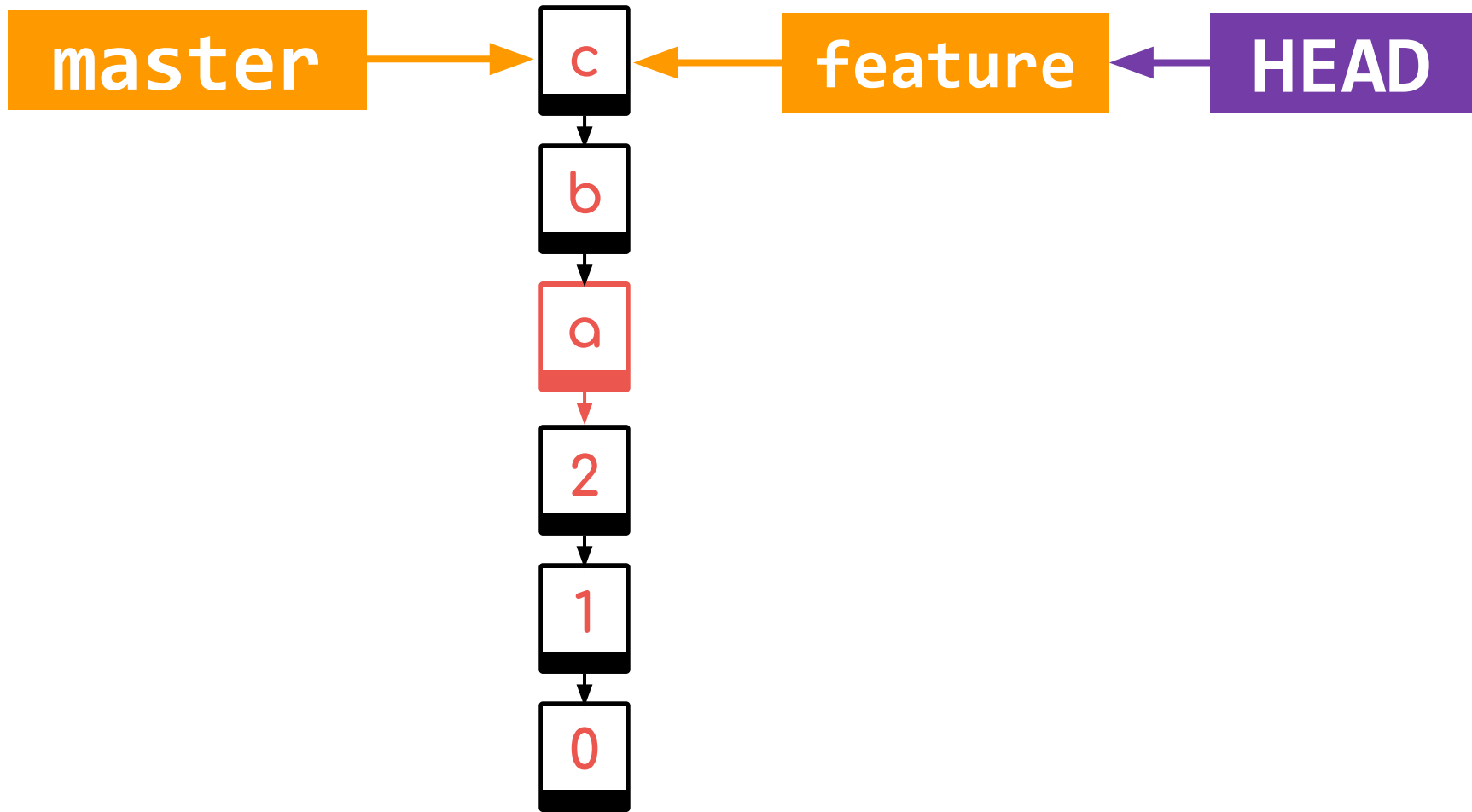


Let's
rebase.

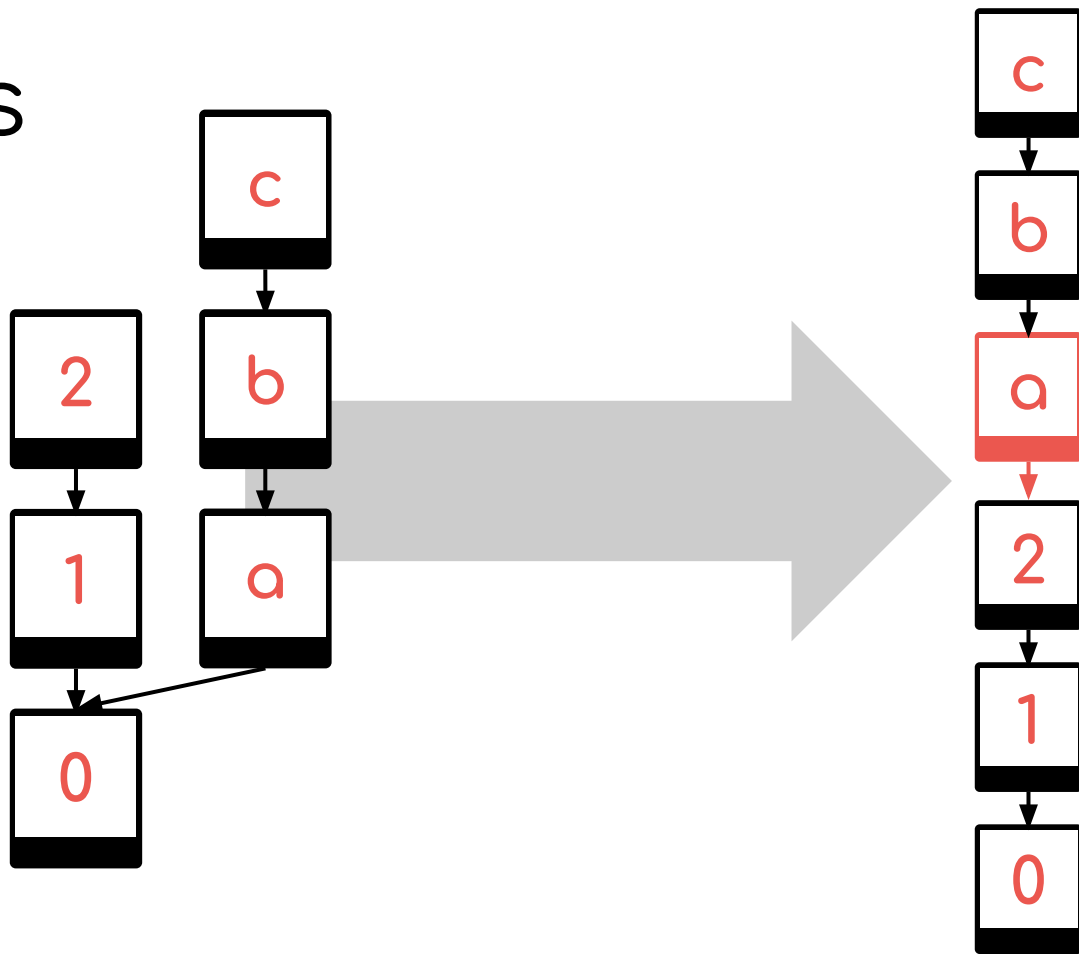




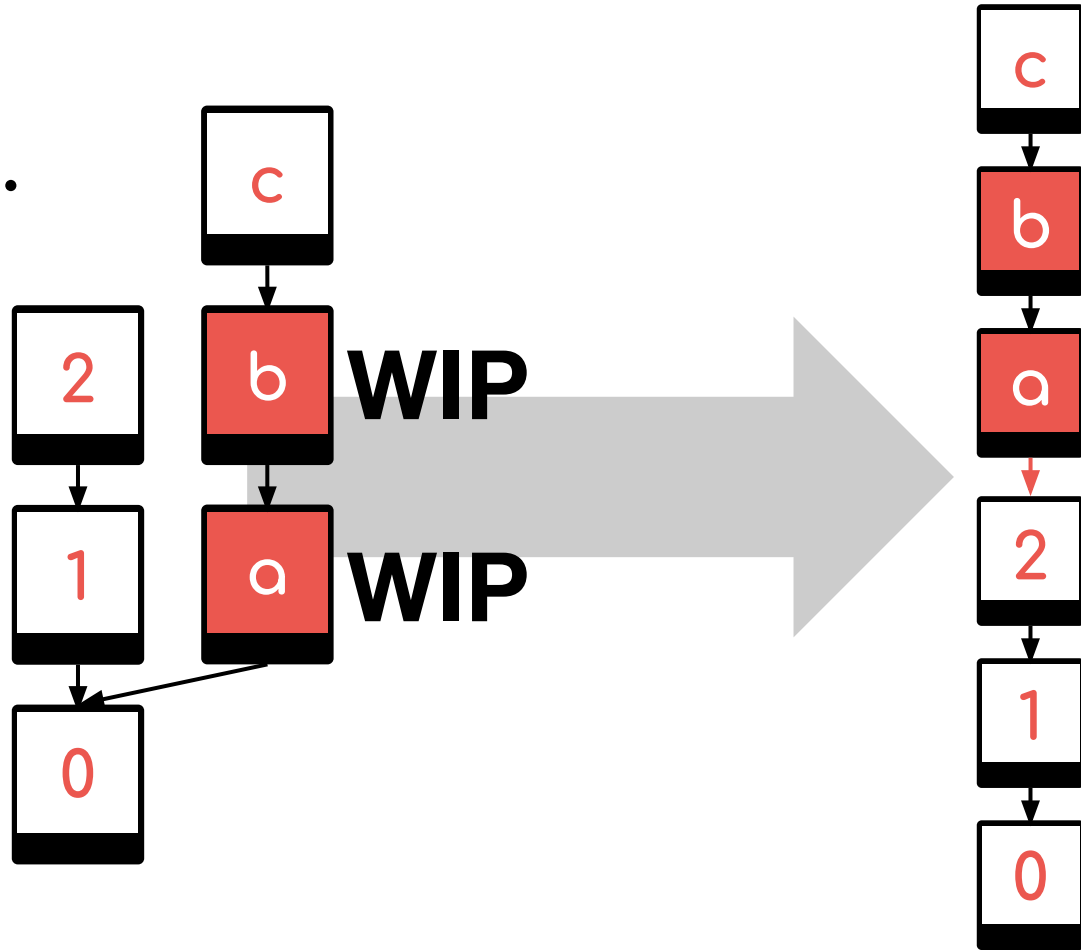


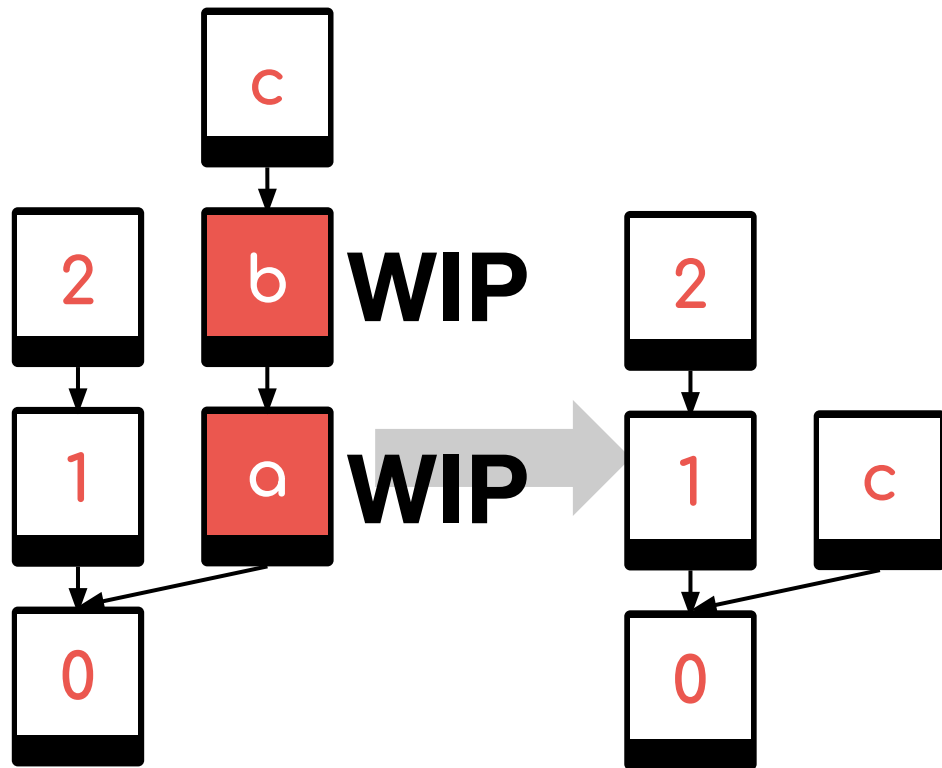


So this is
rebase.

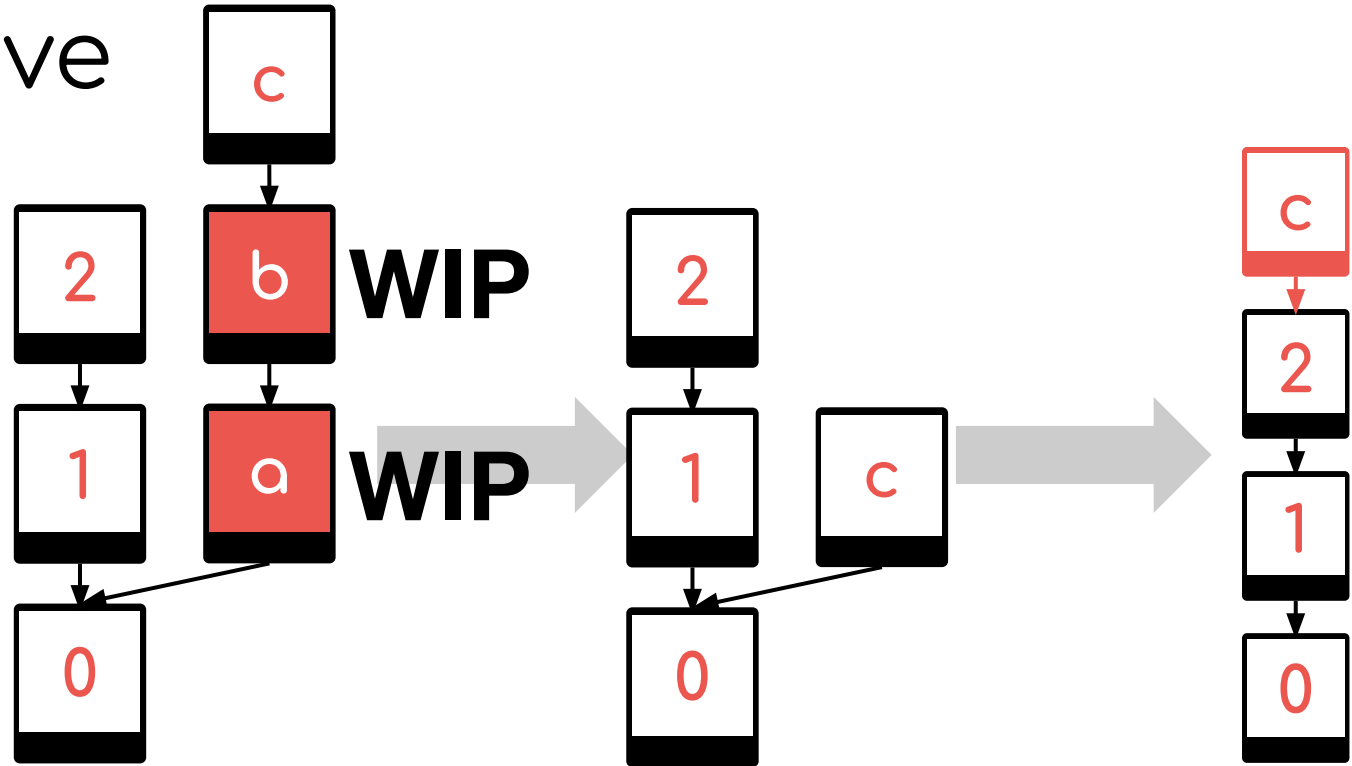


What if ...





This is
interactive
rebase.



Git is Trees, Objects and Pointers.

If you get that, you'll get git manuals, error messages,
and repo states - without losing your HEAD.

Your next step - Katas!

github.com/adamatan/git



Credits & Thanks



Sessions

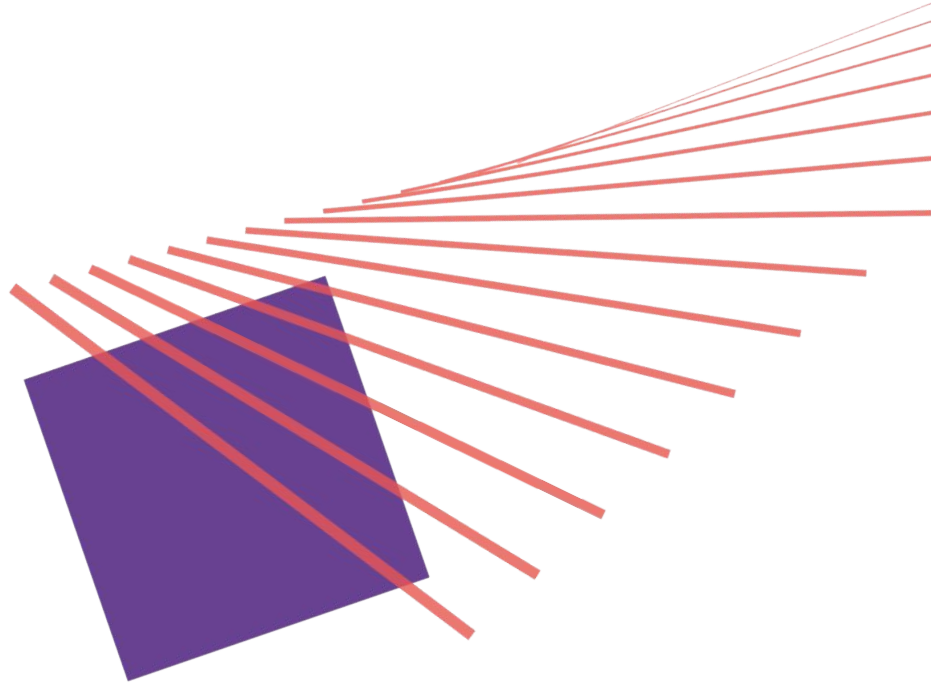
Jessica Kerr (@jessitron), Git happens
David Thomas, GitLab, Fear Not The SHA!



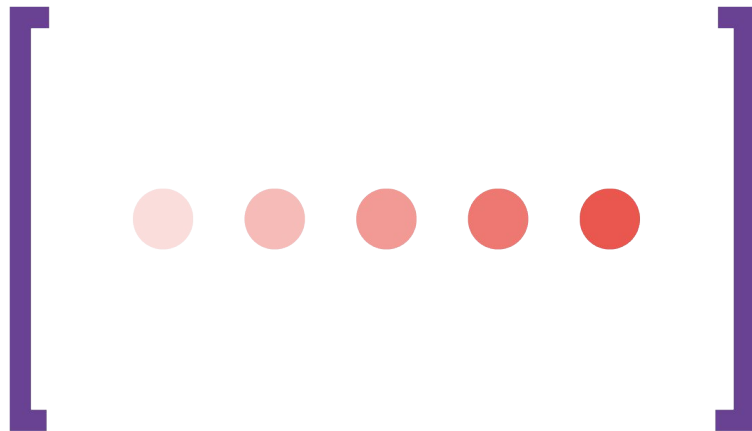
People

Lee Baror - for her wonderful work on the slide deck
Dror Kronstein - for his great advice
Assaf Lavie - for teaching me useful best practices
Issahar Weiss, Avraham Rosenzweig - For moderating me
Guy Kobrinsky - for useful feedback on my workshop at Outbrain

Thank You



Q&A



References

<https://insights.stackoverflow.com/survey/2017#most-popular-technologies>

<https://insights.stackoverflow.com/survey/2019#development-environments-and-tools>

<https://insights.stackoverflow.com/survey/2018>

[Fake git command generator](#)

Image credits

- <https://www.pexels.com/photo/fingers-touching-2172156/>
- [Capitoline Museums](#) [CC BY-SA 3.0]
(<https://creativecommons.org/licenses/by-sa/3.0>)