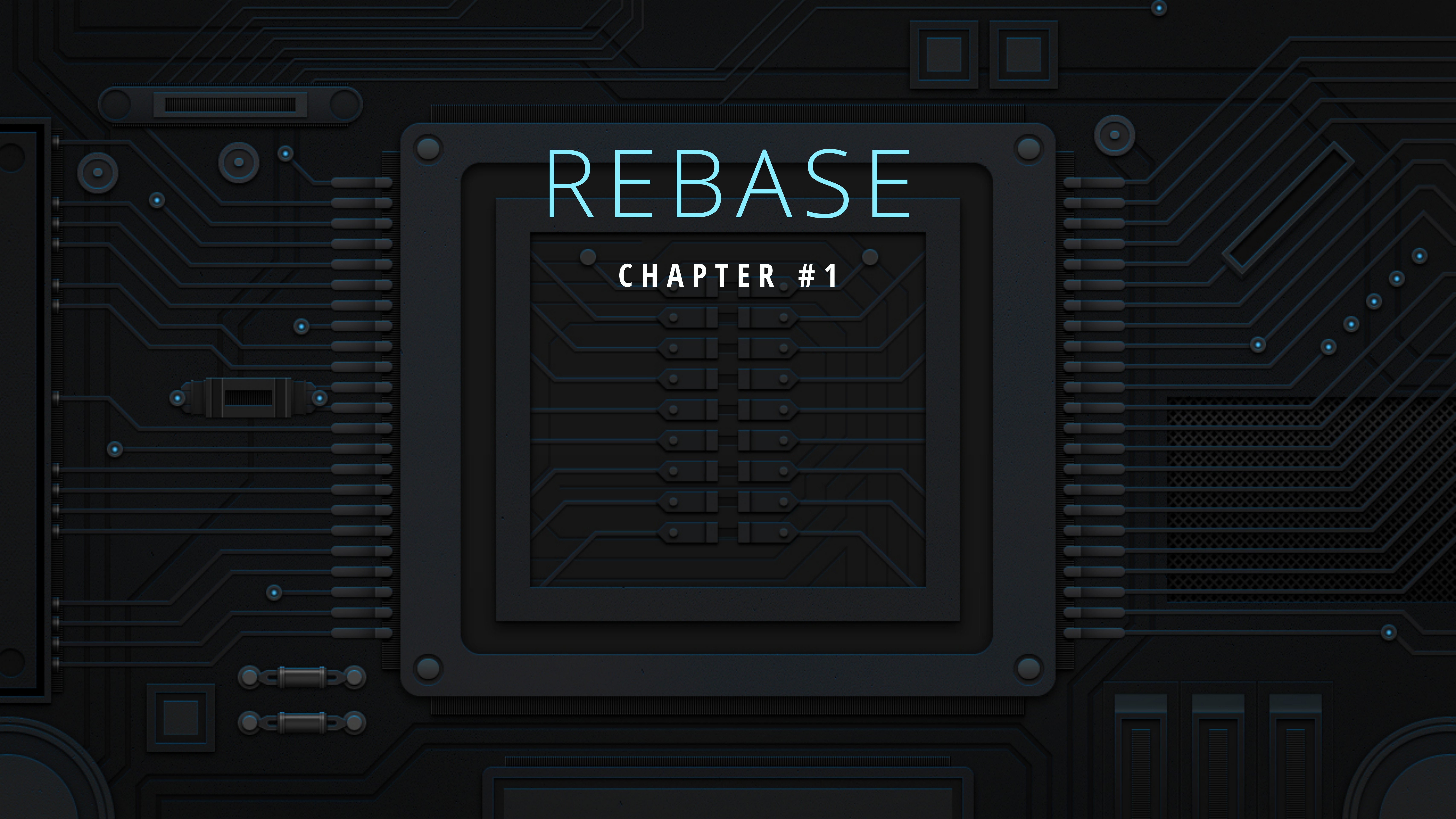
GIT REAL
2



REBASE

CHAPTER #1

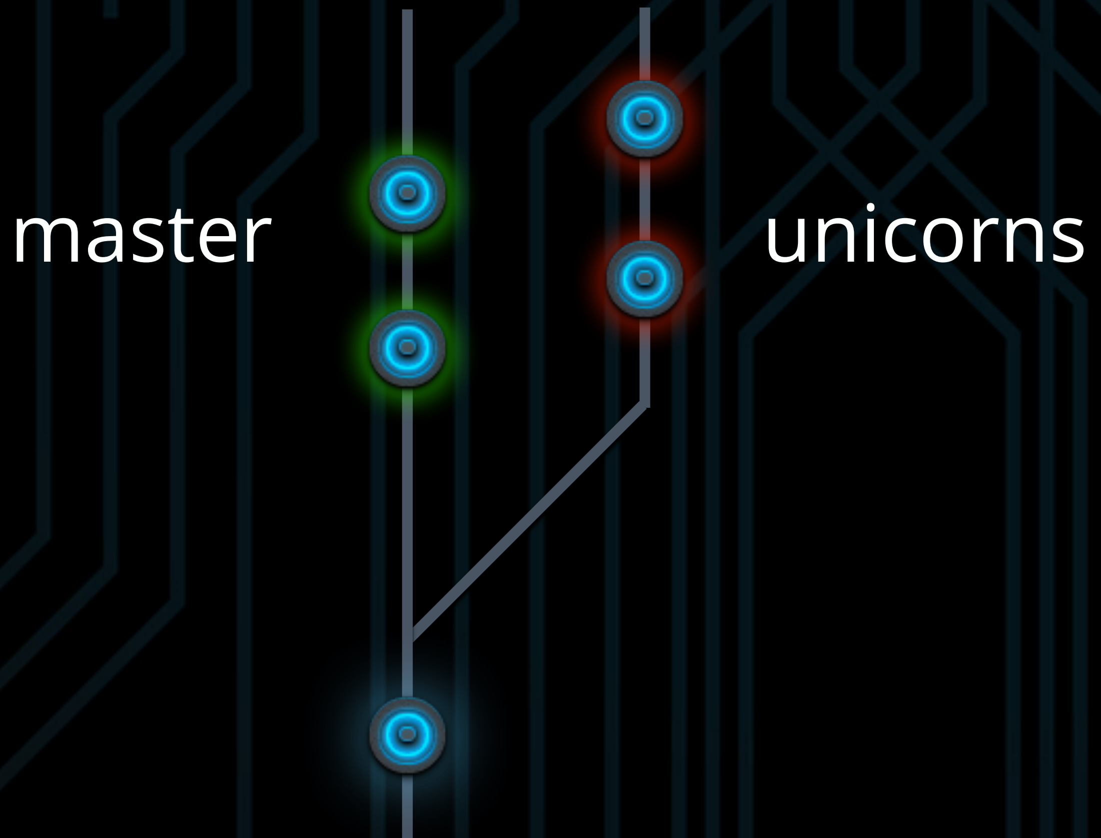
REBASE RECAP

"master" branch

```
$ git log --oneline
cd853a0 Fix pluralization.
b37fb82 Add index.
c60de92 Catalog pages
```

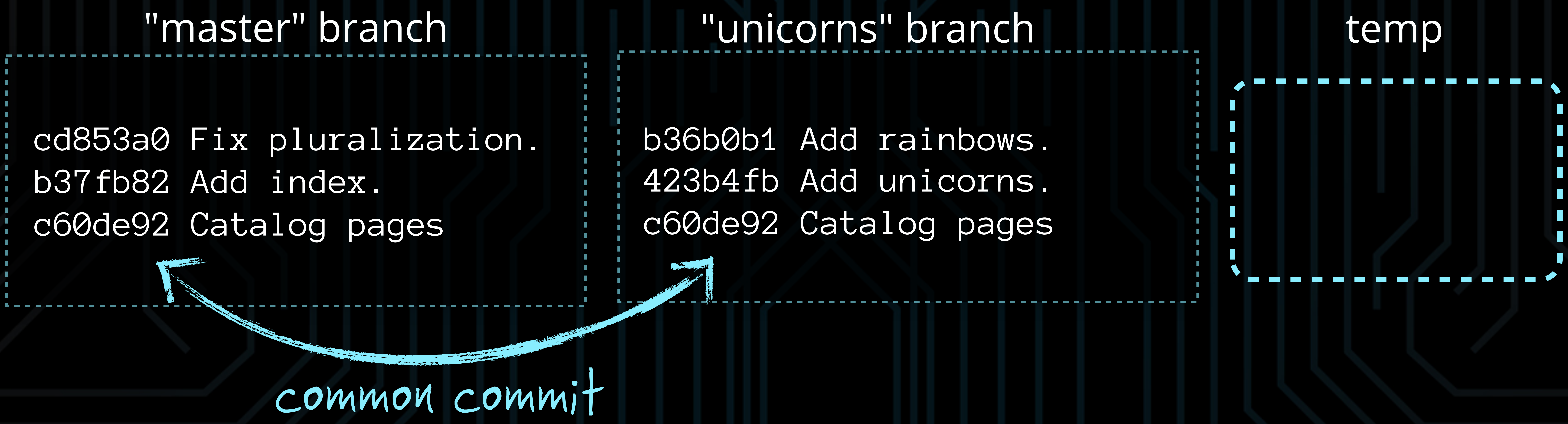
"unicorns" branch

```
$ git log --oneline
b36b0b1 Add rainbows.
423b4fb Add unicorns.
c60de92 Catalog pages
```



REBASE RECAP

```
$ git checkout unicorns
Switched to branch 'unicorns'
$ git rebase master
First, rewinding head to replay your work on top of it...
```



"master" branch

```
cd853a0 Fix pluralization.
b37fb82 Add index.
c60de92 Catalog pages
```

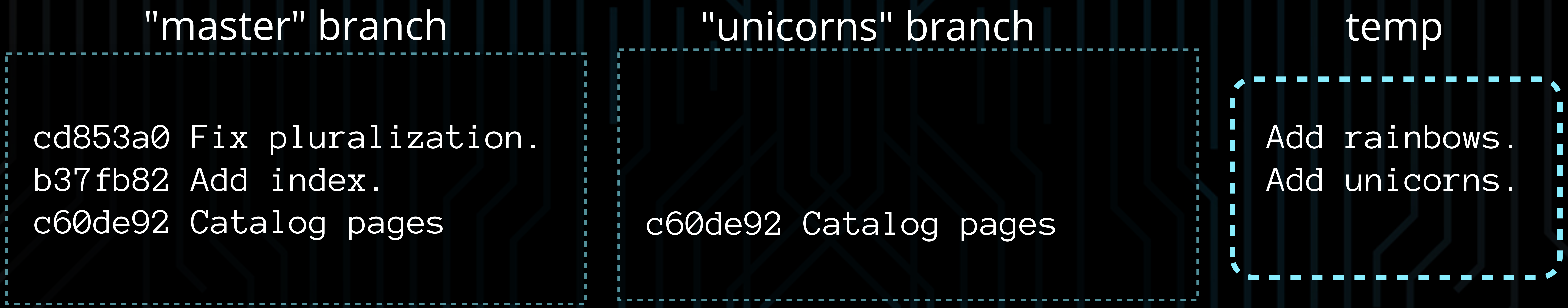
"unicorns" branch

```
b36b0b1 Add rainbows.
423b4fb Add unicorns.
c60de92 Catalog pages
```

temp

REBASE RECAP

```
$ git checkout unicorns
Switched to branch 'unicorns'
$ git rebase master
First, rewinding head to replay your work on top of it...
```



"master" branch

```
cd853a0 Fix pluralization.
b37fb82 Add index.
c60de92 Catalog pages
```

"unicorns" branch

```
c60de92 Catalog pages
```

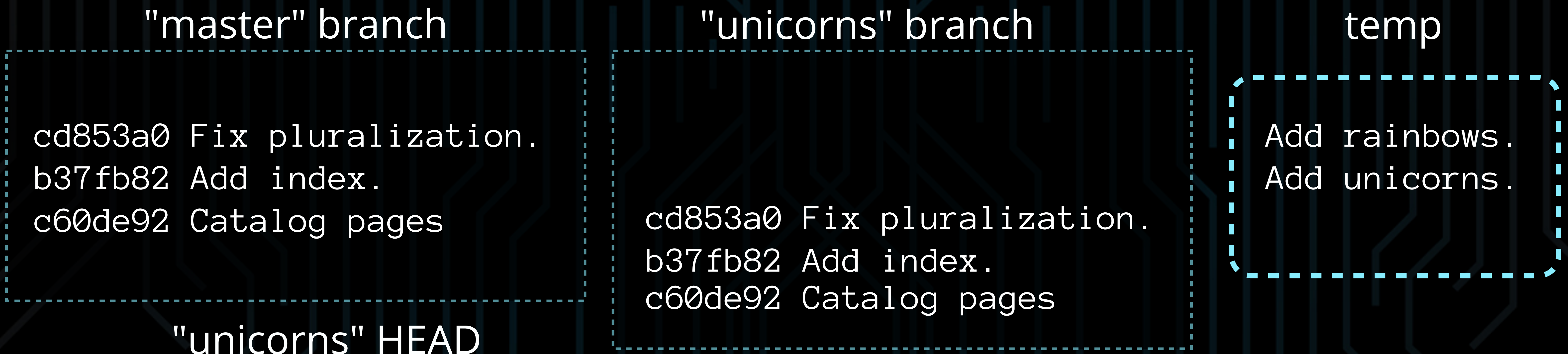
temp

```
Add rainbows.
Add unicorns.
```

Work that's on "unicorns", but not "master", is moved to a temp area

REBASE RECAP

```
$ git checkout unicorns
Switched to branch 'unicorns'
$ git rebase master
First, rewinding head to replay your work on top of it...
```



"master" branch

```
cd853a0 Fix pluralization.
b37fb82 Add index.
c60de92 Catalog pages
```

"unicorns" branch

```
cd853a0 Fix pluralization.
b37fb82 Add index.
c60de92 Catalog pages
```

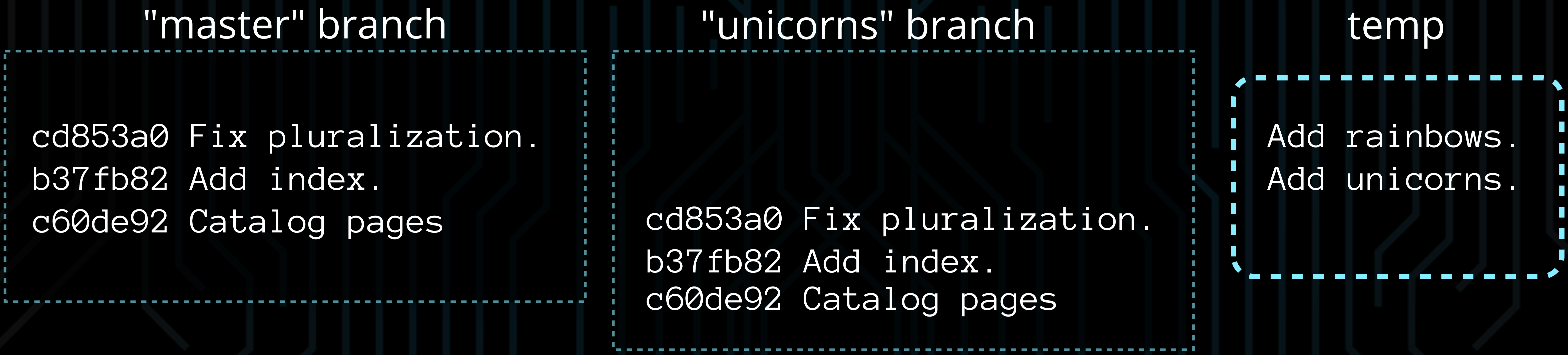
temp

```
Add rainbows.
Add unicorns.
```

"unicorns" HEAD moved to match "master" HEAD

REBASE RECAP

```
$ git checkout unicorns
Switched to branch 'unicorns'
$ git rebase master
First, rewinding head to replay your work on top of it...
```



"master" branch

```
cd853a0 Fix pluralization.
b37fb82 Add index.
c60de92 Catalog pages
```

"unicorns" branch

```
cd853a0 Fix pluralization.
b37fb82 Add index.
c60de92 Catalog pages
```

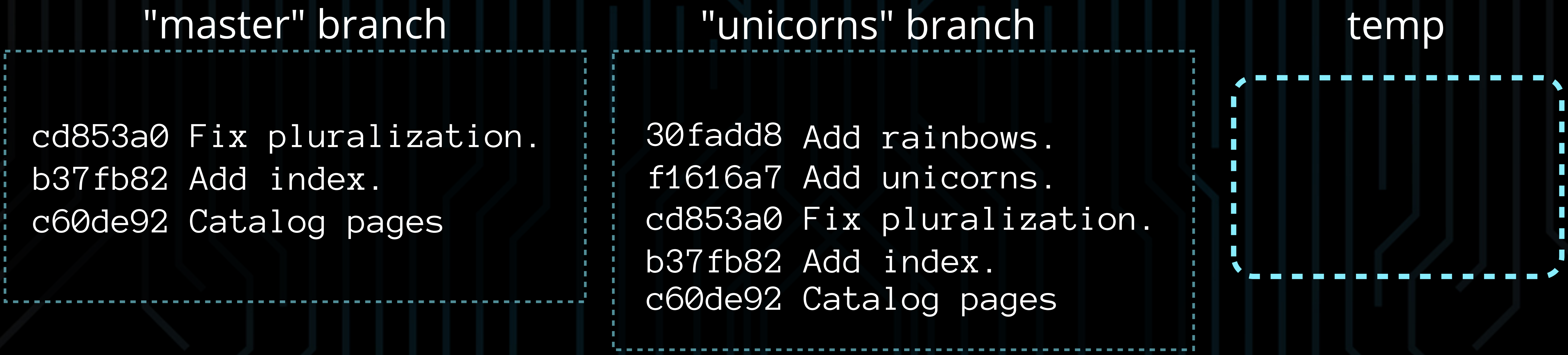
temp

```
Add rainbows.
Add unicorns.
```

Commits replayed,
one at a time

REBASE RECAP

```
$ git checkout unicorns
Switched to branch 'unicorns'
$ git rebase master
First, rewinding head to replay your work on top of it...
```



"master" branch

```
cd853a0 Fix pluralization.
b37fb82 Add index.
c60de92 Catalog pages
```

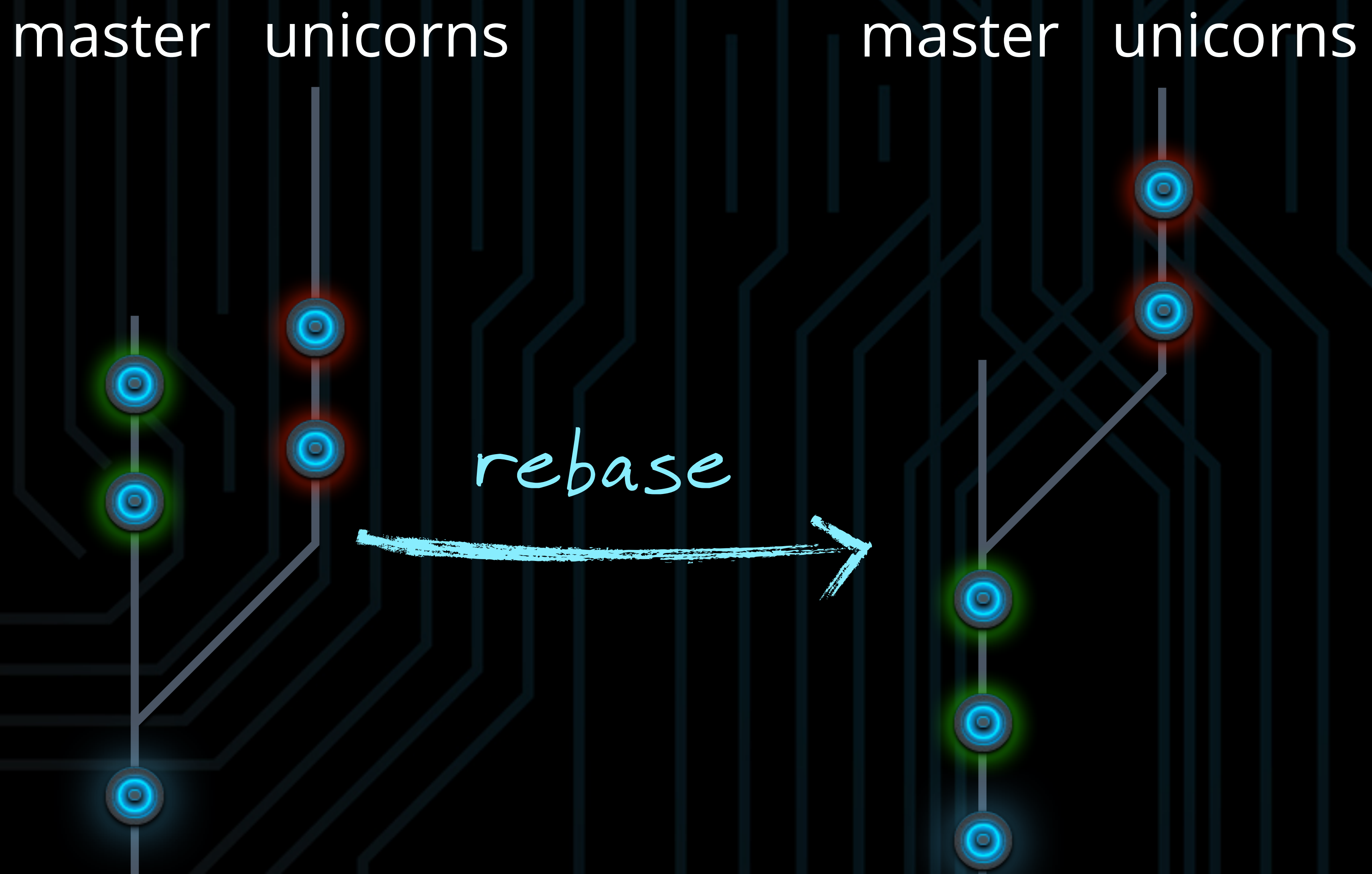
"unicorns" branch

```
30fadd8 Add rainbows.
f1616a7 Add unicorns.
cd853a0 Fix pluralization.
b37fb82 Add index.
c60de92 Catalog pages
```

temp

REBASE RECAP

```
$ git checkout unicorns
Switched to branch 'unicorns'
$ git rebase master
First, rewinding head to replay your work on top of it...
```



INTERACTIVE REBASE

What if we need to alter commits in the SAME branch?

```
$ git rebase -i HEAD~3
```

interactive mode

commit to replay onto

Pops up an editor with the rebase script

INTERACTIVE REBASE

```
$ git rebase -i HEAD~3
```

editor

```
pick 9629e9b Add capybaras to index.
pick 21e37b1 Add capybaras page.
pick eb7d5a0 Actually, the plural is 'capybara'.

# Rebase 4202447..eb7d5a0 onto 4202447
#
# Commands:
#  p, pick = use commit
#  r, reword = use commit, but edit the commit message
#  e, edit = use commit, but stop for amending
#  s, squash = use commit, but meld into previous commit
#  f, fixup = like "squash", but discard this commit's log message
#  x, exec = run command (the rest of the line) using shell
```

these commands will execute when you close the editor

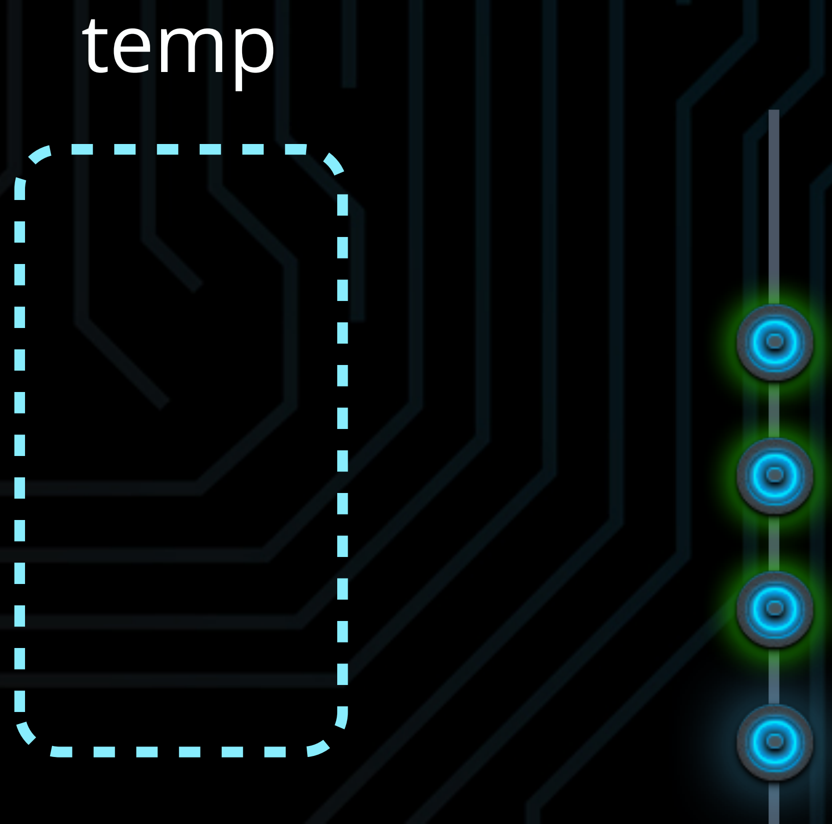
you can replace them with any of these

INTERACTIVE REBASE

editor

```
pick 9629e9b Add capybaras to index.
pick 21e37b1 Add capybaras page.
pick eb7d5a0 Actually, the plural is 'capybara'.
```

We didn't make any changes...



The default script just replays the same commits

INTERACTIVE REBASE

Alters every commit AFTER the one you specify

✗

```
$ git rebase -i HEAD
```

editor

```
noop

# Rebase 21e37b1..21e37b1 onto 21e37b1
```

There's no commits after HEAD!

Does nothing

✓

```
$ git rebase -i HEAD^
```

editor

```
pick 21e37b1 Add capybaras page.

# Rebase 9629e9b..21e37b1 onto 9629e9b
```

Did you mean the parent of HEAD?

REORDER COMMITS

```
$ git log --oneline
9afe987 Actually, the plural is 'capybara'.
1ee9572 Add capybaras page.
74e6f3e Add capybaras to index.
```

We want to re-order these two commits

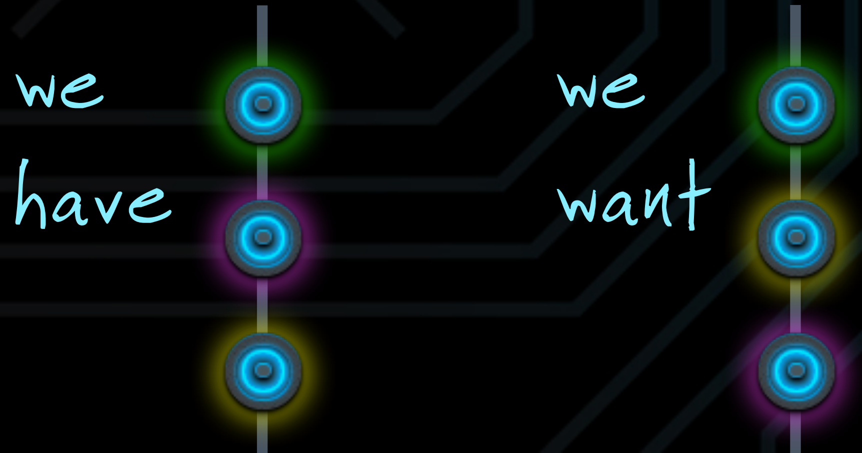
```
$ git rebase -i HEAD~3
```

editor

```
pick 74e6f3e Add capybaras to index.
pick 1ee9572 Add capybaras page.
pick 9afe987 Actually, the plural is 'capybara'.
```

"pick" chooses a commit for replay

Default script is in the original chronological order



REORDER COMMITS

```
$ git log --oneline
9afe987 Actually, the plural is 'capybara'.
1ee9572 Add capybaras page.
74e6f3e Add capybaras to index.
```

```
$ git rebase -i HEAD~3
```

editor

```
pick 1ee9572 Add capybaras page.
pick 74e6f3e Add capybaras to index.
pick 9afe987 Actually, the plural is 'capybara'.
```

Swap the order, save, and close the editor

```
$ git log --oneline
0511ab7 Actually, the plural is 'capybara'.
7d2edea Add capybaras to index.
44d59fa Add capybaras page.
```

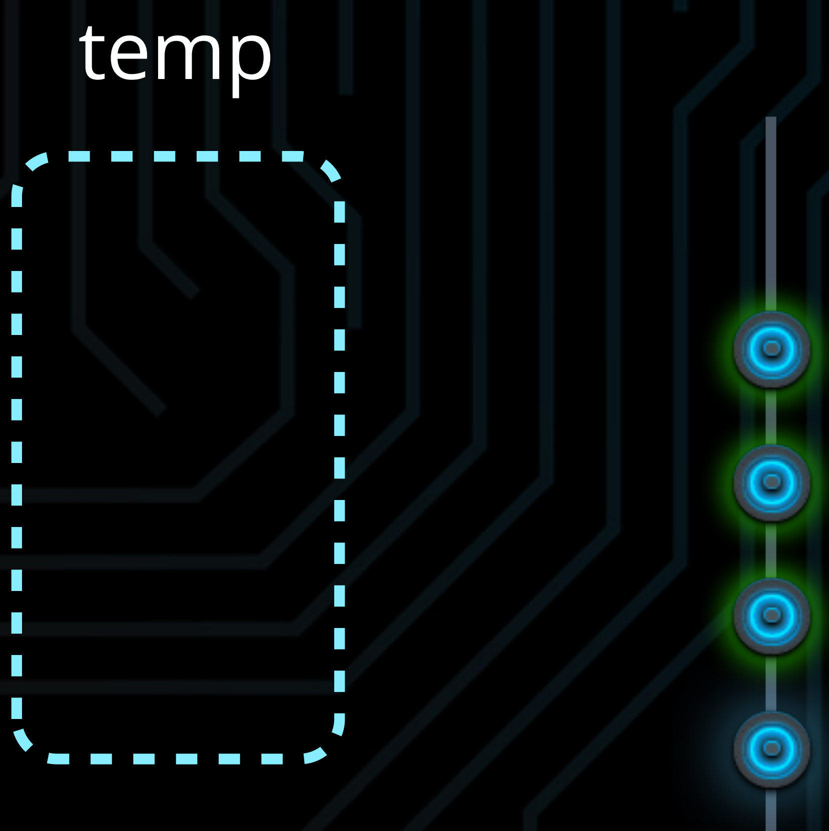
Commits get replayed in new order

INTERACTIVE REBASE

editor

```
pick 1ee9572 Add capybaras page.
pick 74e6f3e Add capybaras to index.
pick 9afe987 Actually, the plural is 'capybara'.
```

With the order swapped...



Commits will be replayed (picked) in the specified order

HOW TO CHANGE MESSAGES

editor

```
pick 1ee9572 Add capybaras page.
pick 74e6f3e Add capybaras to index.
pick 9afe987 Actually, the plural is 'capybara'.
```

CHANGE MESSAGES

editor

```
pick 1ee9572 Add capybaras page.
pick 74e6f3e Add capybaras to index.
reword 9afe987 Actually, the plural is 'capybara'.
```

Change "pick" to "reword", then save and exit

Another editor pops up, with the existing message as the default

editor

```
Actually, the plural is 'capybara'.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# Not currently on any branch.
# Changes to be committed:
#   (use "git reset HEAD^1 <file>..." to unstage)
#
#  modified:   capybara.html
#  modified:   index.html
```

CHANGE MESSAGES

editor

```
pick 1ee9572 Add capybaras page.
pick 74e6f3e Add capybaras to index.
reword 9afe987 Actually, the plural is 'capybara'.
```

editor

```
Change plural on index and details pages to 'capybara'.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# Not currently on any branch.
# Changes to be committed:
#   (use "git reset HEAD^1 <file>..." to unstage)
#
#  modified:   capybara.html
#  modified:   index.html
```

Enter new message, save, and exit

NOW THE MESSAGE IS UPDATED

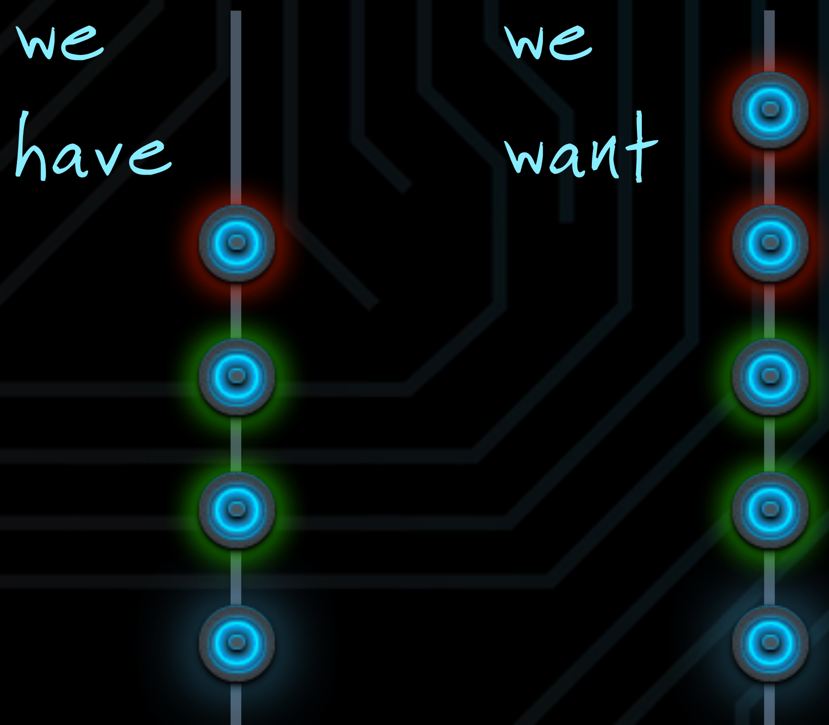
```
$ git log --oneline
9afe987 Change plural on index and details pages to 'capybara'.
1ee9572 Add capybaras page.
74e6f3e Add capybaras to index.
```

SPLIT COMMITS

editor

```
pick 1ee9572 Add capybaras page.
pick 74e6f3e Add capybaras to index.
pick 39b23ce Change plural on index and details pages to 'capybara'.
```

We want to split this into 2 commits



SPLIT COMMITS

editor

```
pick 1ee9572 Add capybaras page.
pick 74e6f3e Add capybaras to index.
edit 39b23ce Change plural on index and details pages to 'capybara'.
```

Change "pick" to"edit"

"edit" replays changes and stops after this commit

```
Stopped at 39b23ce... Change plural on index
and details pages to 'capybara'.
You can amend the commit now, with

   git commit --amend

Once you are satisfied with your changes, run

   git rebase --continue
$
```

Returns to prompt

SPLIT COMMITS

To split commits, we need to undo the changes that were just replayed

```
$ git reset HEAD^
Unstaged changes after reset:
M   capybara.html
M   index.html
```

since we don't specify "--hard", files stay in working directory

```
$ git add capybara.html
$ git commit -m "Change plural on detail page to 'capybara'."
[detached HEAD 6e8e5d6] Change plural on detail page to 'capybara'.
 1 file changed, 1 insertion(+), 1 deletion(-)
$ git add index.html
$ git commit -m "Change plural on index page to 'capybara'."
[detached HEAD e8005f4] Change plural on index page to 'capybara'.
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Commit them separately

SPLIT COMMITS

New commits are in place, now we need to finish the rebase

```
$ git rebase --continue
Successfully rebased and updated refs/heads/master.
```

Resumes replaying changes from temp area

```
$ git log --oneline
e8005f4 Change plural on index page to 'capybara'.
6e8e5d6 Change plural on detail page to 'capybara'.
7d2edea Add capybaras to index.
44d59fa Add capybaras page.
```

One commit is now two!

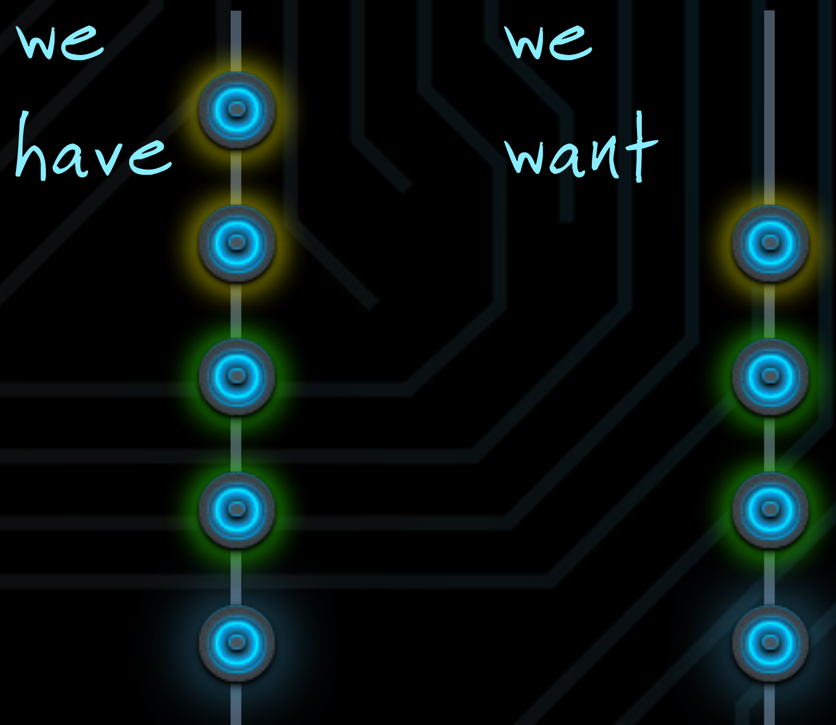
SQUASH COMMITS

No, wait... they were better off as one commit.

```
$ git rebase -i HEAD~4
```

editor

```
pick 44d59fa Add capybaras page.
pick 7d2edea Add capybaras to index.
pick 6e8e5d6 Change plural on detail page to 'capybara'.
pick e8005f4 Change plural on index page to 'capybara'.
```



SQUASH COMMITS

No, wait... they were better off as one commit.

```
$ git rebase -i HEAD~4
```

editor

```
pick 44d59fa Add capybaras page.
pick 7d2edea Add capybaras to index.
pick 6e8e5d6 Change plural on detail page to 'capybara'.
squash e8005f4 Change plural on index page to 'capybara'.
```

"squash" merges a commit with the previous commit

SQUASH COMMITS

editor

```
# This is a combination of 2 commits.
# The first commit's message is:

Change plural on detail page to 'capybara'.

# This is the 2nd commit message:

Change plural on index page to 'capybara'.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
```

Another editor pops up, with the commits being squashed

"master" branch

SQUASH COMMITS

editor

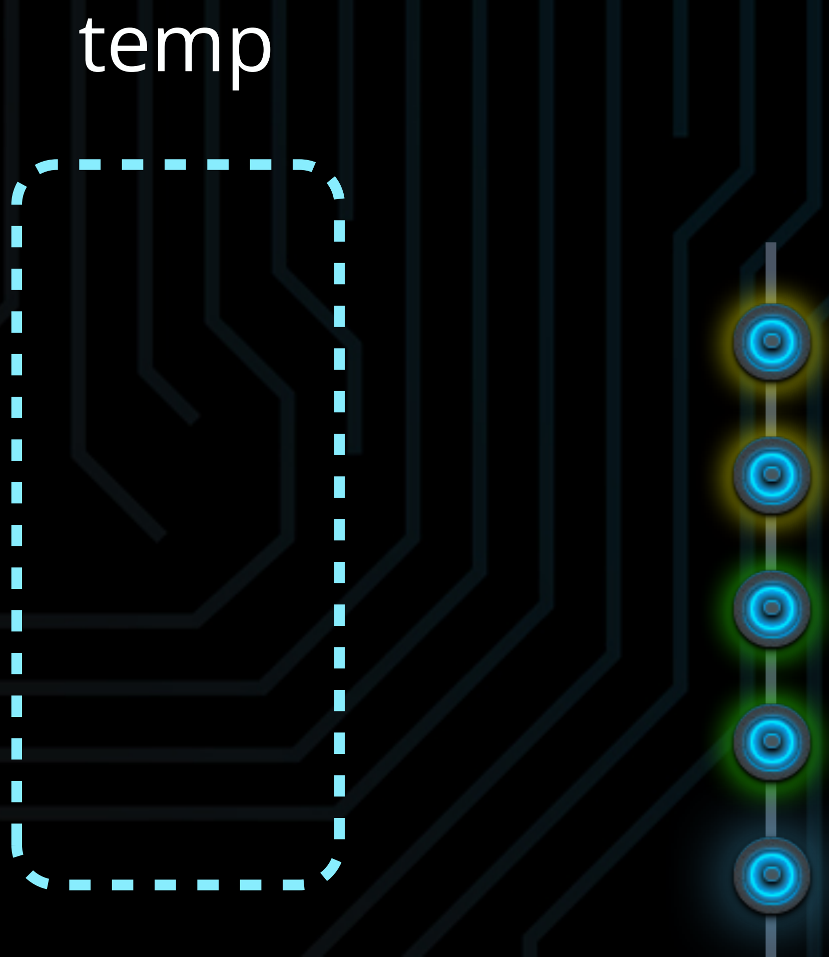
```
# This is a combination of 2 commits.
# The first commit's message is:

Change plurals to 'capybara'.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
```

Enter the message for the new commit, save, and exit

SQUASH COMMITS

```
$ git log --oneline
1c4614d Change plurals to 'capybara'.
7d2edea Add capybaras to index.
44d59fa Add capybaras page.
```

One commit, with the changes to both files



The content of two commits is replayed, but only one commit is made