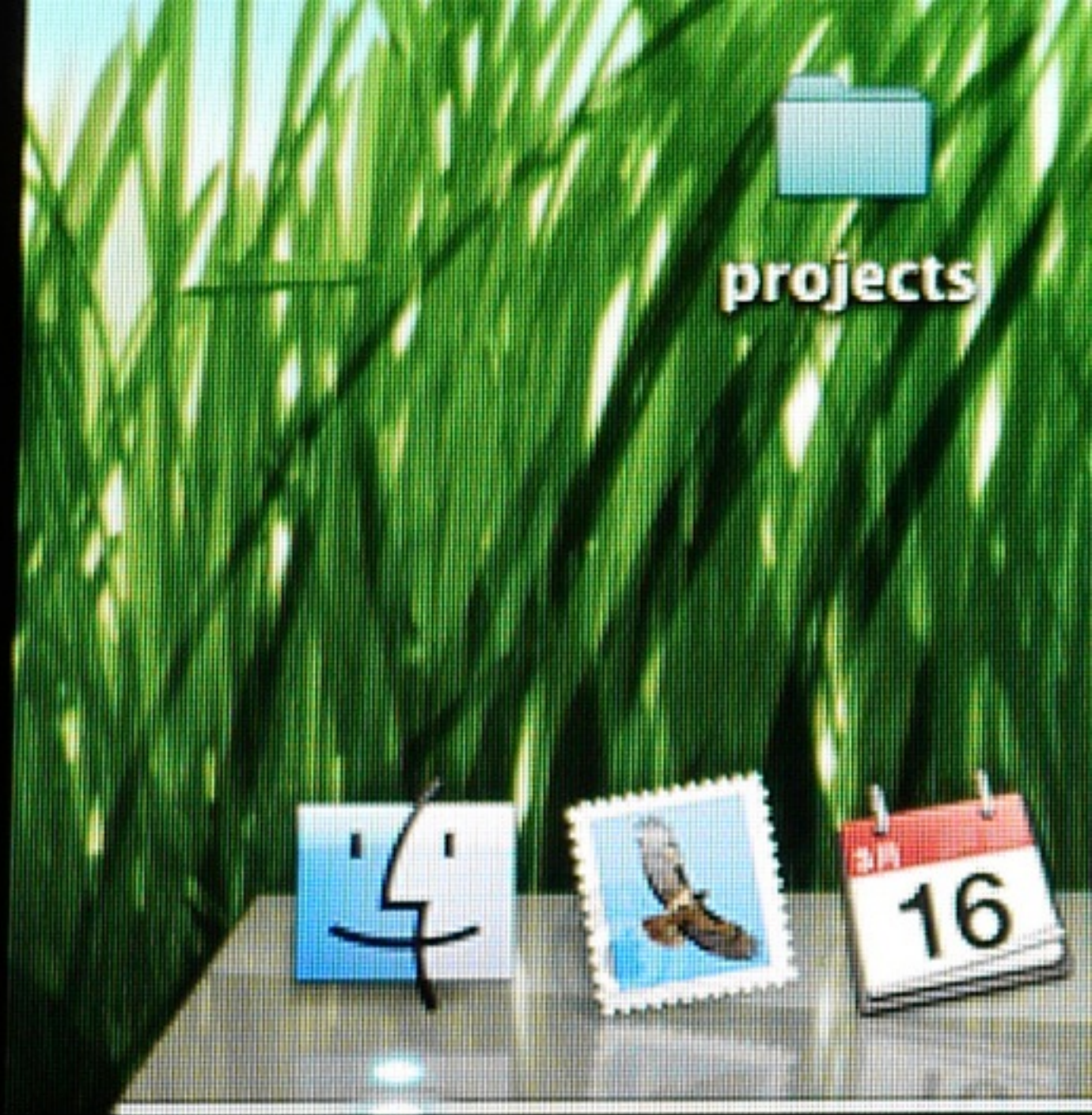


你知道 Git 是怎麼一回事嗎？

高見龍 @eddiekao

高見龍 @eddiekao a.k.a Eddie

- 📌 愛現! 喜歡冷門的玩具
- 📌 Ruby/Rails/iOS app 開發者、講師
- 📌 Ruby 技術推廣、教育、諮詢
- 📌 台灣、日本 Ruby 技術研討會講者
- 📌 目前於五倍紅寶石擔任紅寶石鑑定商職務
- 📌 部落格：<http://kaochenlong.com>



假的封面

第1版

為你自己學

Ruby on Rails

學習不為公司、不為別人，只為你自己！



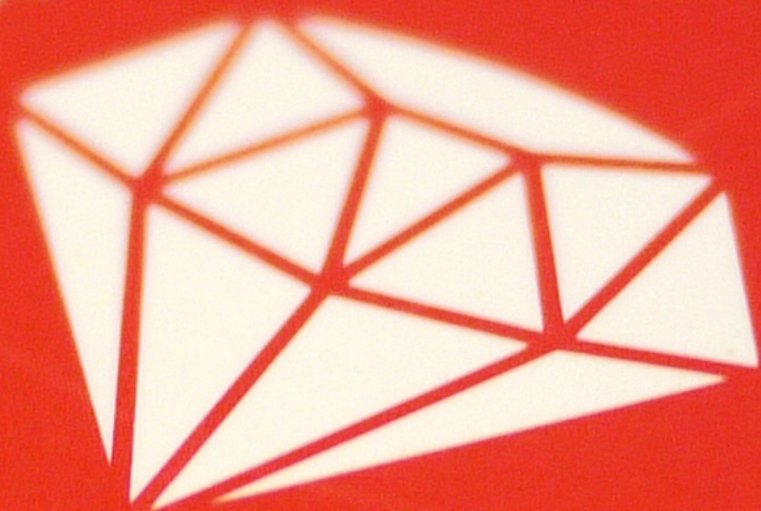
5xRuby
ファイブルビー

高見龍 著
たかみ りゅう 訳

- ✓ 這是一本 Ruby on Rails 的入門書。
- ✓ 沒意外的話，大概九月會上市(應該吧)。
- ✓ 封面當然不是長這樣！
- ✓ 網頁版可於線上免費閱讀。

網址：<http://railsbook.tw>

5x{



}.tw

五倍紅寶石股份有限公司



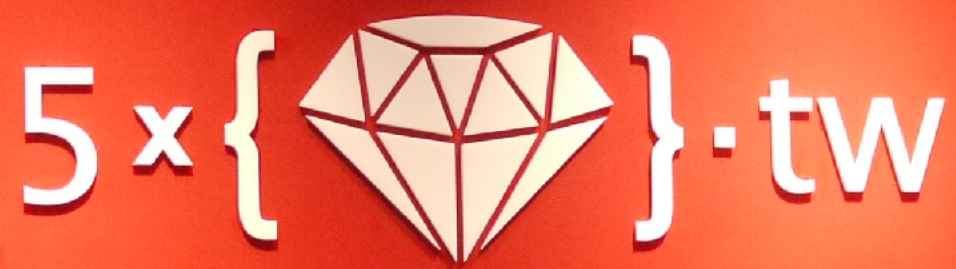
五倍紅寶石

<https://5xruby.tw/>



學妹

阿肥!



五倍紅寶石台北出礦坑

5xRuby Inc.

五倍紅寶石

<https://5xruby.tw/>



身為一位專業的碼農
身上隨時帶著紅蘿蔔也是很合理的...



有獎徵答時間!

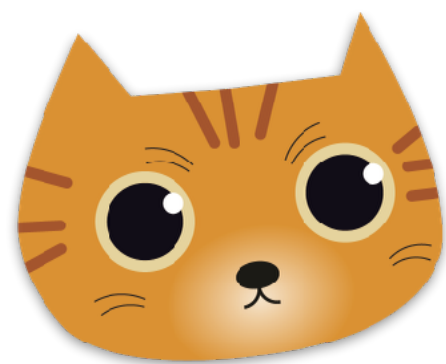
明年
4/27 ~ 28

RubyConf Taiwan



適合對象

平常工作有在用 Git
或曾經用過 Git



平常有在使用 Git
的朋友請舉個手！

Git 很簡單，我會！


```
$ git add .
```

```
$ git commit -m "done!"
```


打完收工!

Git 簡單嗎? 困難嗎?

Git 不算難，但真的不好學！

易學難精

瞭解運作的原理
遇到問題才能迎刃而解

底層命令(Plumbing)

v.s

高階指令(Porcelain)

20/80 法則

Git != SVN + Magic

有獎徵答



"git add ." 跟 "git add --all"
這兩個指令有什麼不一樣？

有獎徵答

在 Git 1.x 版的時候...

使用參數	新增檔案	修改檔案	刪除檔案
<code>--all</code>	O	O	O
<code>.</code>	O	O	X

有獎徵答

在 Git 2.x 版的時候...

使用參數	新增檔案	修改檔案	刪除檔案
<code>--all</code>	☐	☐	☐
<code>.</code>	☐	☐	☐

有獎徵答

"git add --all" 指令

不管在專案的哪一層目錄執行都有效果



那個...你有聽過 Git 嗎? 那是什麼?

啊就一種分散式的版本控制系統啊!



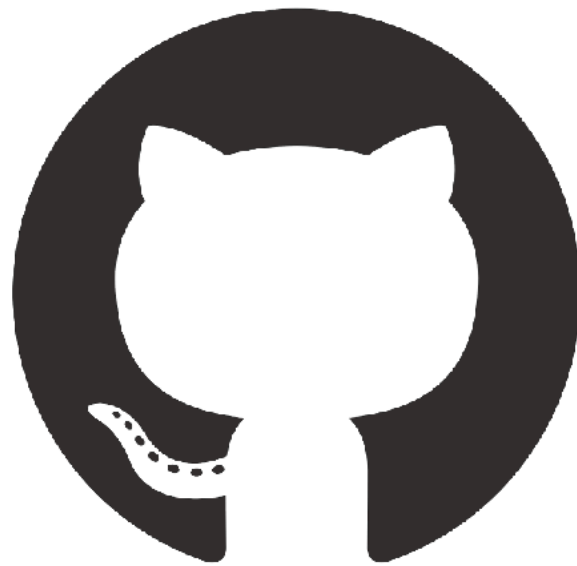
? ???



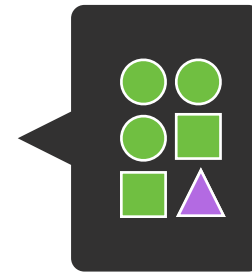
分散? 版本?

「分散式」？

分散式 != 放在 GitHub 上



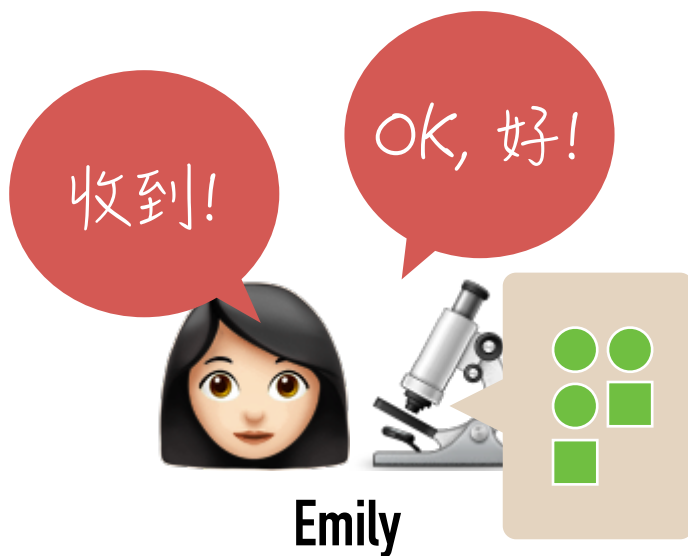
Octocat



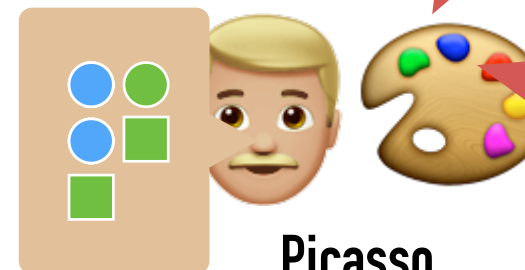
Sherly



Eddie



Emily



Picasso

大家可以
下班了(誤)

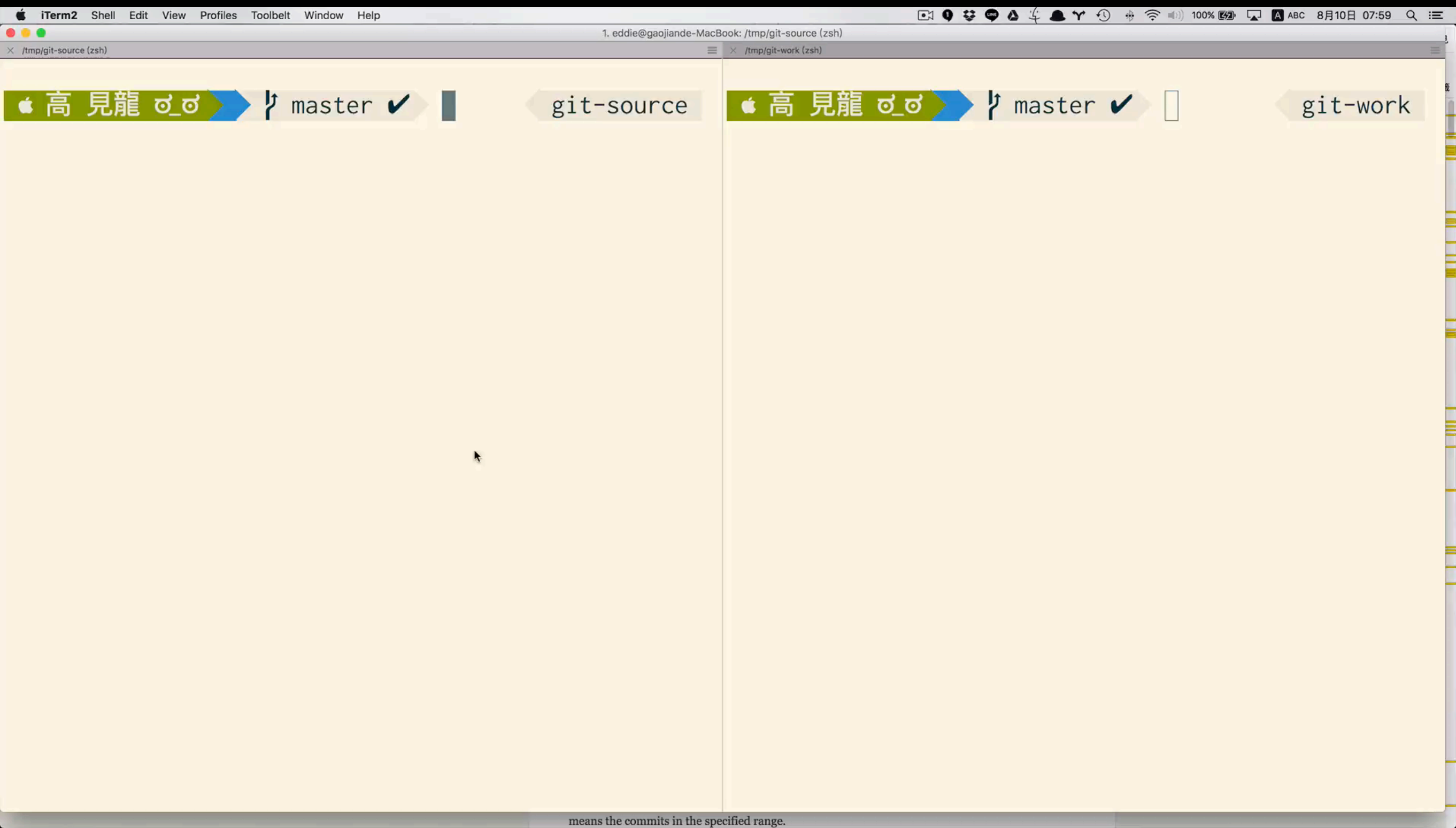


No server is currently available to service your request.

Sorry about that. Please try refreshing and contact us if the problem persists.

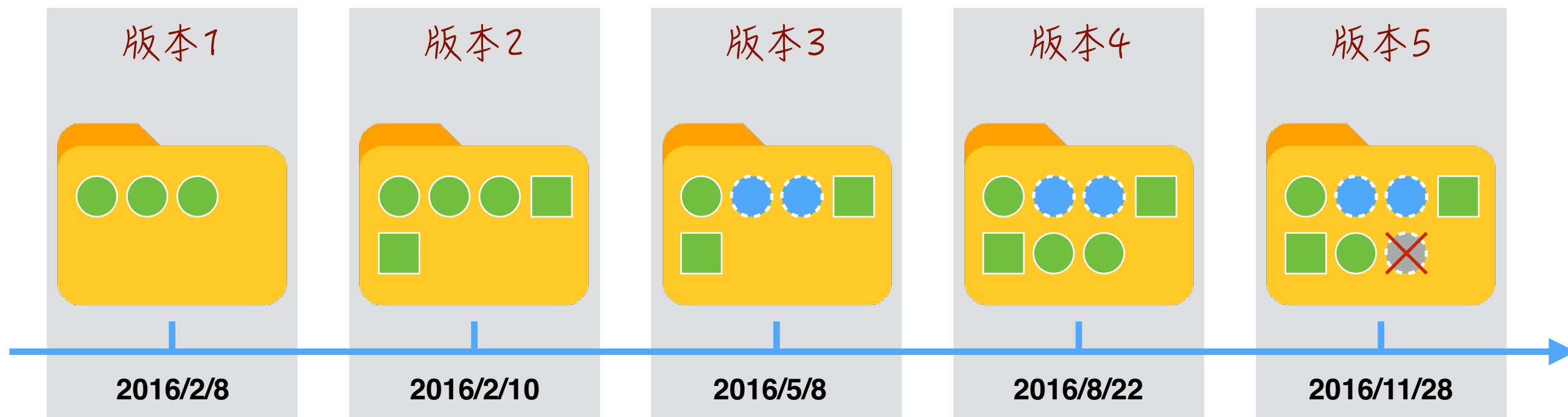
咦？

不用 push 就可以交換檔案？



means the commits in the specified range.

「版本」？



Git

跟其它的版控系統有什麼不同？

Git 只在乎檔案的內容
不在乎目錄或檔案名稱

Git 其實只是個內容追蹤軟體
(stupid content tracker)

“In many ways you can just see git as a filesystem – it’s content-addressable, and it has a notion of versioning, but I really really designed it coming at the problem from the viewpoint of a filesystem person (hey, kernels is what I do), and I actually have absolutely zero interest in creating a traditional SCM system.”

— Linus

Git 只在乎檔案的內容
不在乎目錄或檔案名稱

Git 根本不在乎你!

有獎徵答



既然 Git 只在乎檔案內容，不在乎檔案名稱... 那，檔案名稱存在哪裡？

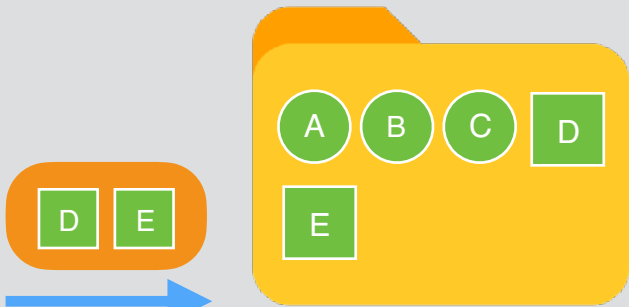
別的版控軟體記錄每次的差異

版本1



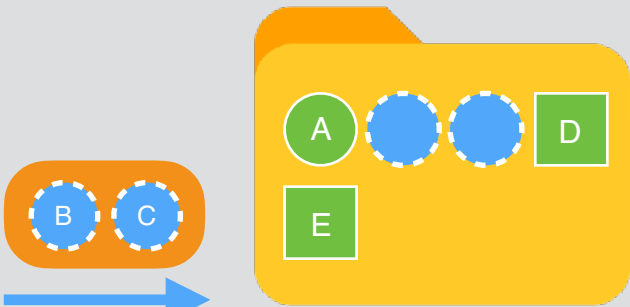
2016/2/8

版本2



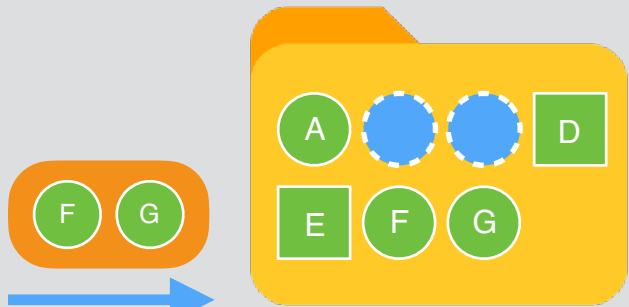
2016/2/10

版本3



2016/5/8

版本4



2016/8/22

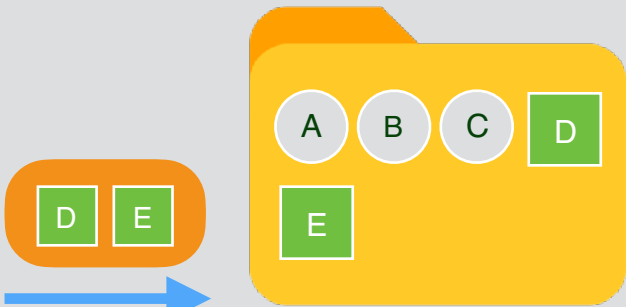
別的版控軟體記錄每次的差異

版本1



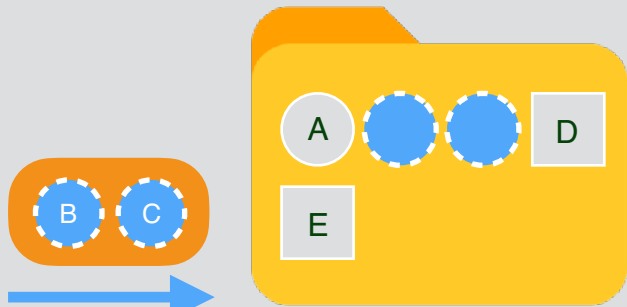
2016/2/8

版本2



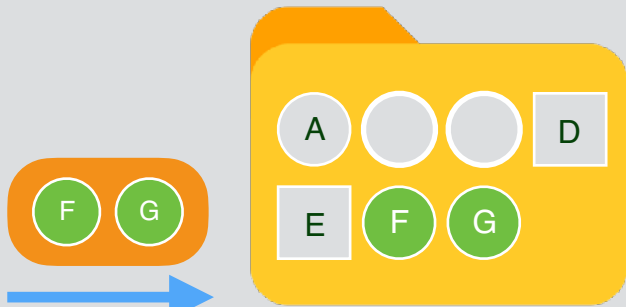
2016/2/10

版本3



2016/5/8

版本4



2016/8/22

先別說這個了

你聽過 "SHAT" 嗎?

\$ git log

commit 657fce783a23e26721ec4f778b9e0e108253e92d (HEAD -> master)

Author: Eddie Kao <eddie@5xruby.tw>

Date: Wed Aug 2 16:53:43 2017 +0800

add container

commit abb4f43814af7bcf47afa9b779aaba63599e562b

Author: Eddie Kao <eddie@5xruby.tw>

Date: Wed Aug 2 16:49:49 2017 +0800

update index page

commit cef6e4017eb1a16a7bb3434f12d9008ff83a821a

Author: Eddie Kao <eddie@5xruby.tw>

Date: Wed Aug 2 03:02:37 2017 +0800

create index page

就是這些看起來
像亂碼的傢伙啦!

科普時間

什麼是 SHA1 演算法？

科普時間

由 40 個 16 進位字元組成

科普時間

$$f(x) = 3x + 2$$

科普時間

輸入一樣的內容，就有一樣的輸出值
輸入不一樣的内容，就有不一樣的輸出值

廢話！



科普時間

SHA1 值會有重複的狀況發生嗎？

科普時間

碰撞！

科普時間

重複 $\neq$ 碰撞

科普時間

碰撞 =

輸入兩個不同的值，卻得到相同的結果

科普時間

也就是說，兩個內容不同的檔案，
卻得到一樣的 SHA1 值

科普時間

這個機會有多低？

科普時間

每秒幾十萬
上下!

就算你每秒 commit 100,000 次，在有限的人生也應該沒機會遇到碰撞

科普時間

等等... 聽說今年年初，Google 不是
破解 SHA1 什麼的嗎？

Collision attack: **same** hashes



Good doc



Sha-1



3713..42



Bad doc



Sha-1



3713..42

科普時間

聽說攻擊了 9 兆次

9,223,372,036,854,775,808

科普時間

Git 計算 SHA1 方式還有額外加料...



所以我說，

那個 Git 的 SHA1 是怎麼算出來的？

Blob 物件的 SHA1 計算公式：

1. 「"blob" 字樣」
2. 「1 個空白字元」
3. 「輸入內容的長度」
4. 「Null 結束符號」
5. 「輸入內容」

```
$ irb
>> require "digest/sha1"
=> true
>> content = "Hello, 5xRuby"
=> "Hello, 5xRuby"
>> input = "blob #{content.length}\0#{content}"
=> "blob 13\u0000Hello, 5xRuby"
>> Digest::SHA1.hexdigest(input)
=> "4135fc4add3332e25ab3cd5acabe1bd9ea0450fb"
```

輸入的內容

在這裡加料

算出答案!

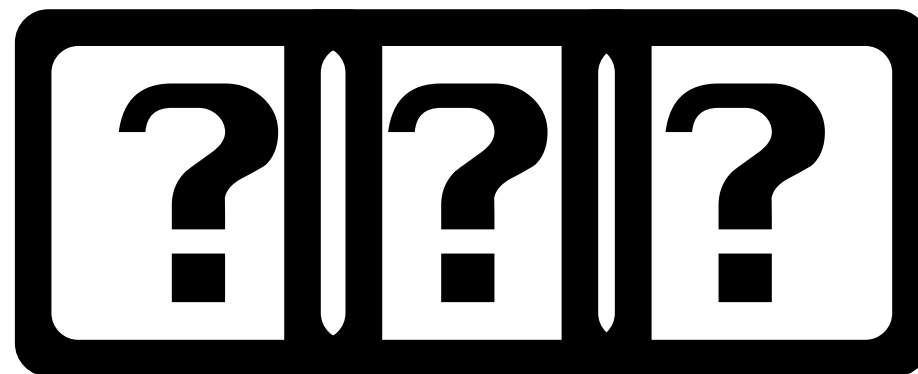
使用
`git hash-object`
指令

```
$ printf "Hello, 5xRuby" | git hash-object --stdin  
4135fc4add3332e25ab3cd5acabe1bd9ea0450fb
```

算出來的結果

在 Git 的世界裡
每個物件都有它自己的 SHA1

有獎徵答



請猜猜看

e69de29bb2d1d6434b8b29ae775ad8c2e48c5391

這個 SHA1 值，在 Git 裡是什麼內容的輸出值

有獎徵答

答案是「空字串」

```
$ printf "" | git hash-object --stdin  
e69de29bb2d1d6434b8b29ae775ad8c2e48c5391
```

Apple logo

高見龍

0_0

↗

master

✓

modern-web

.git

⏪

⏩

📁

🔍

⚙️

📁 喜好項目

📁 Dropbox

📁 AirDrop

📁 應用程式

📁 Google Drive

📁 桌面

📁 文件

📁 下載項目

📁 音樂

📁 影片

📁 圖片

📁 Presentations

📁 真 · 五倍紅寶石

📁 裏 · 五倍紅寶石

📁 Creative Cloud Files

名稱

📁 hooks

📁 info

📁 objects

📁 info

📁 pack

📁 refs

📄 config

📄 description

📄 HEAD

📄 index

Git 物件的檔名：SHA1

Git 物件的內容：檔案內容 + 壓縮

git commit 的過程發生了什麼事？

Git 裡的 4 種物件

檔案

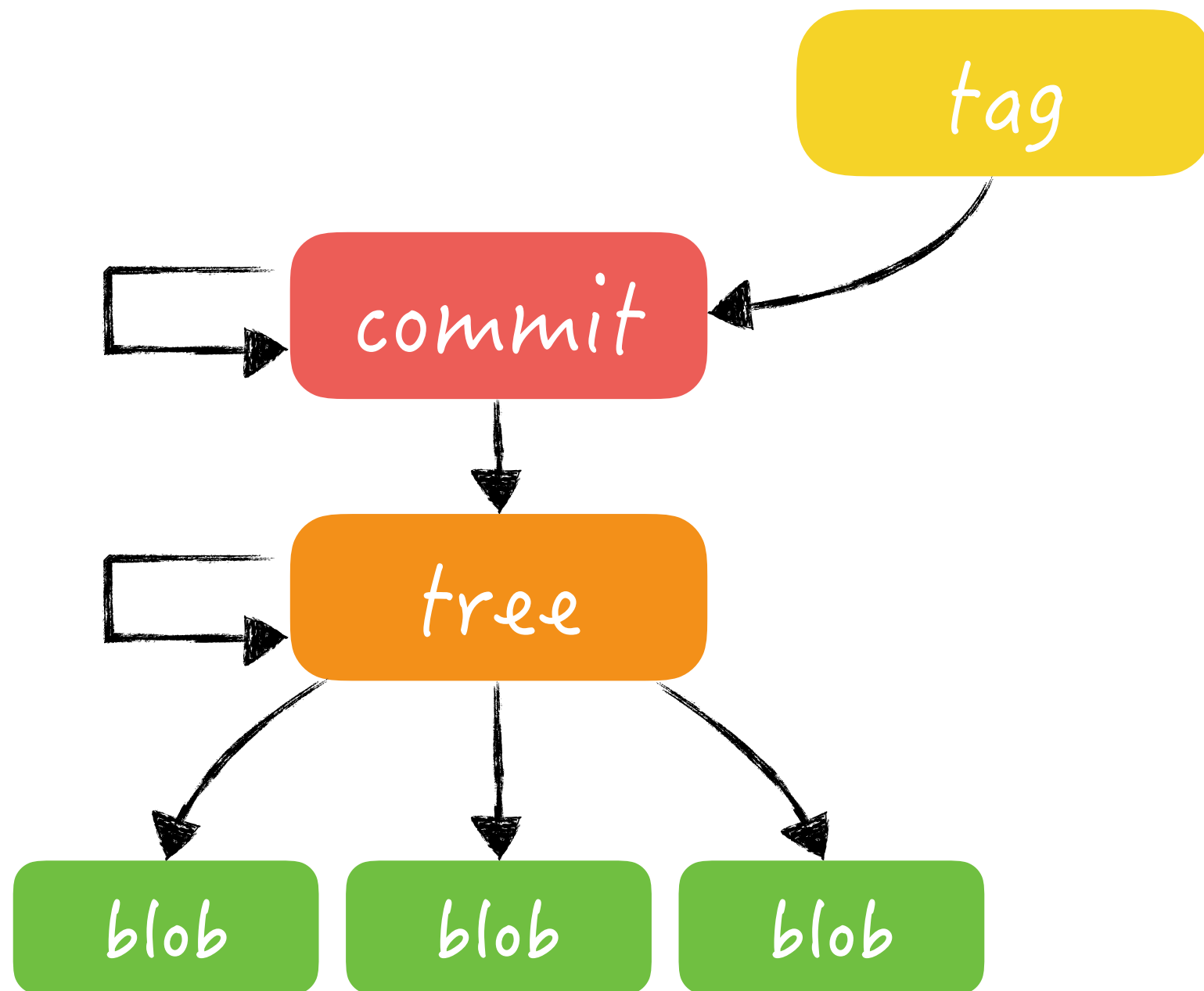
blob

commit

tree

tag

目錄



實際運作流程，GO!

```
$ echo "hello, 5xruby" > index.html
```

untracked
状態



我是一個 blob
物件喔

fa6556

```
$ git add index.html
```


fa6556

你進來了？

```
$ mkdir config
```

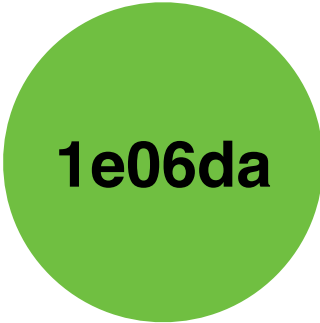
fa6556

untracked
狀態

```
$ echo "super-secret-password" > config/database.yml
```

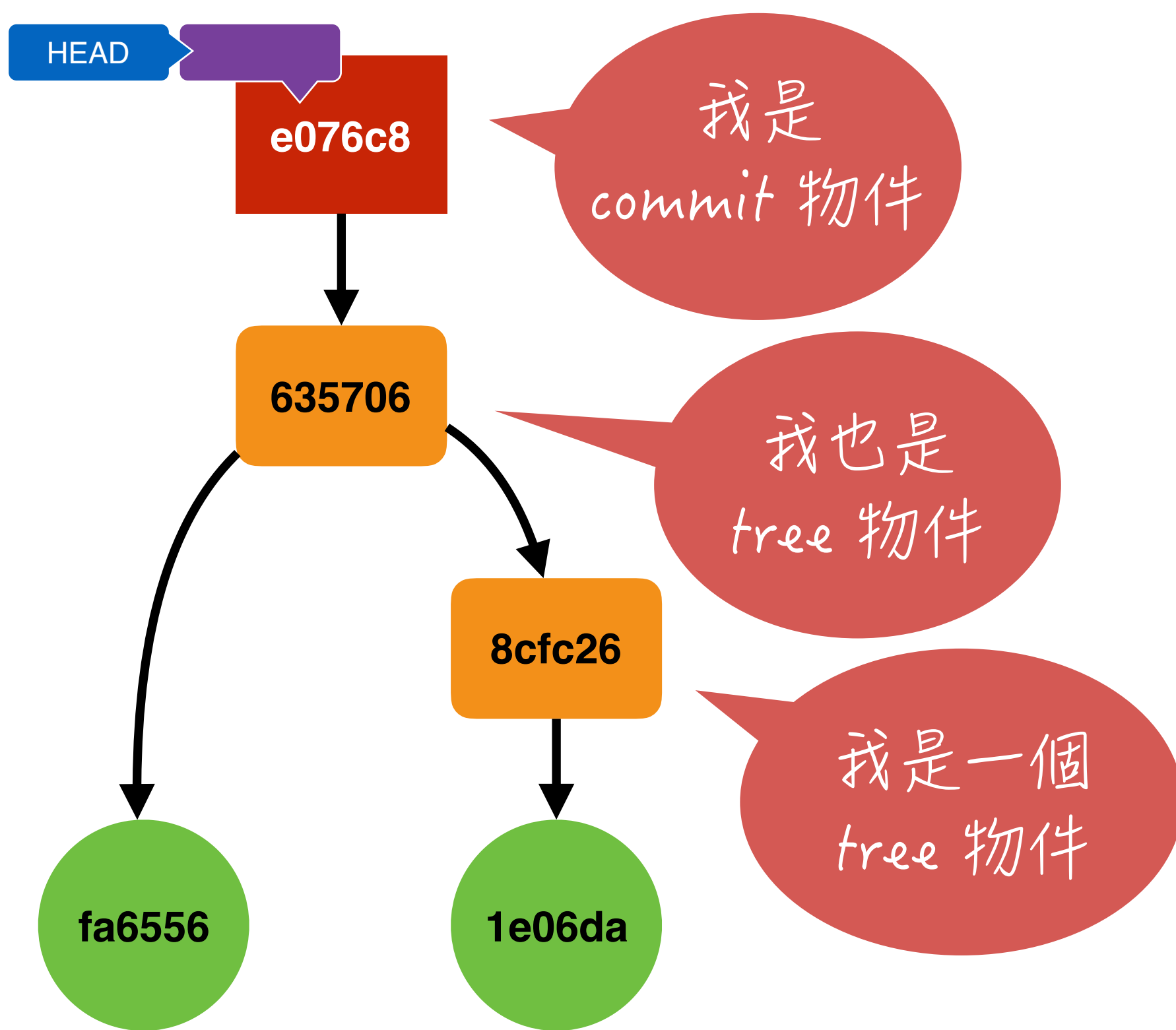


fa6556

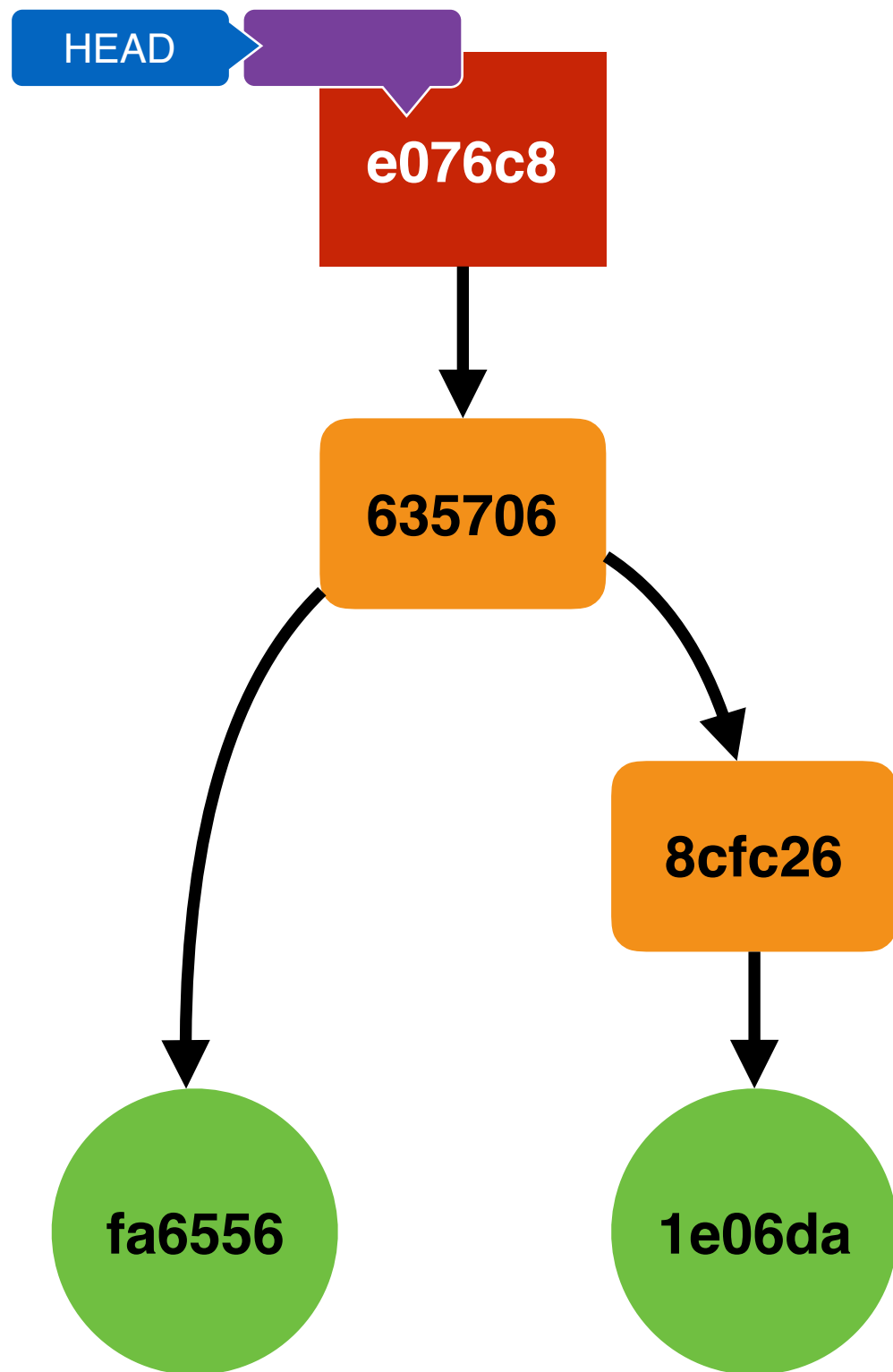


1e06da

```
$ git add config/database.yml
```

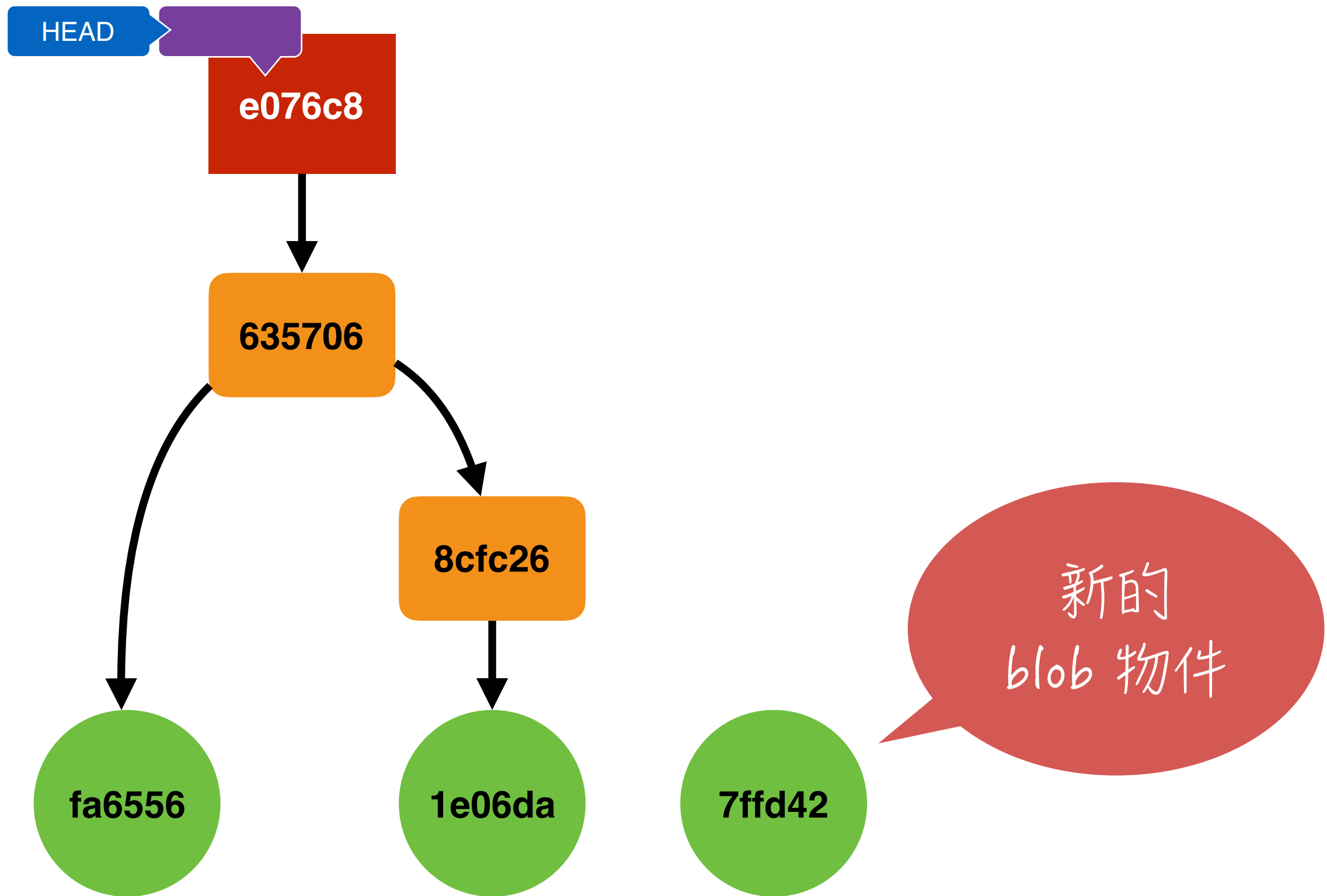


```
$ git commit -m "init commit"
```

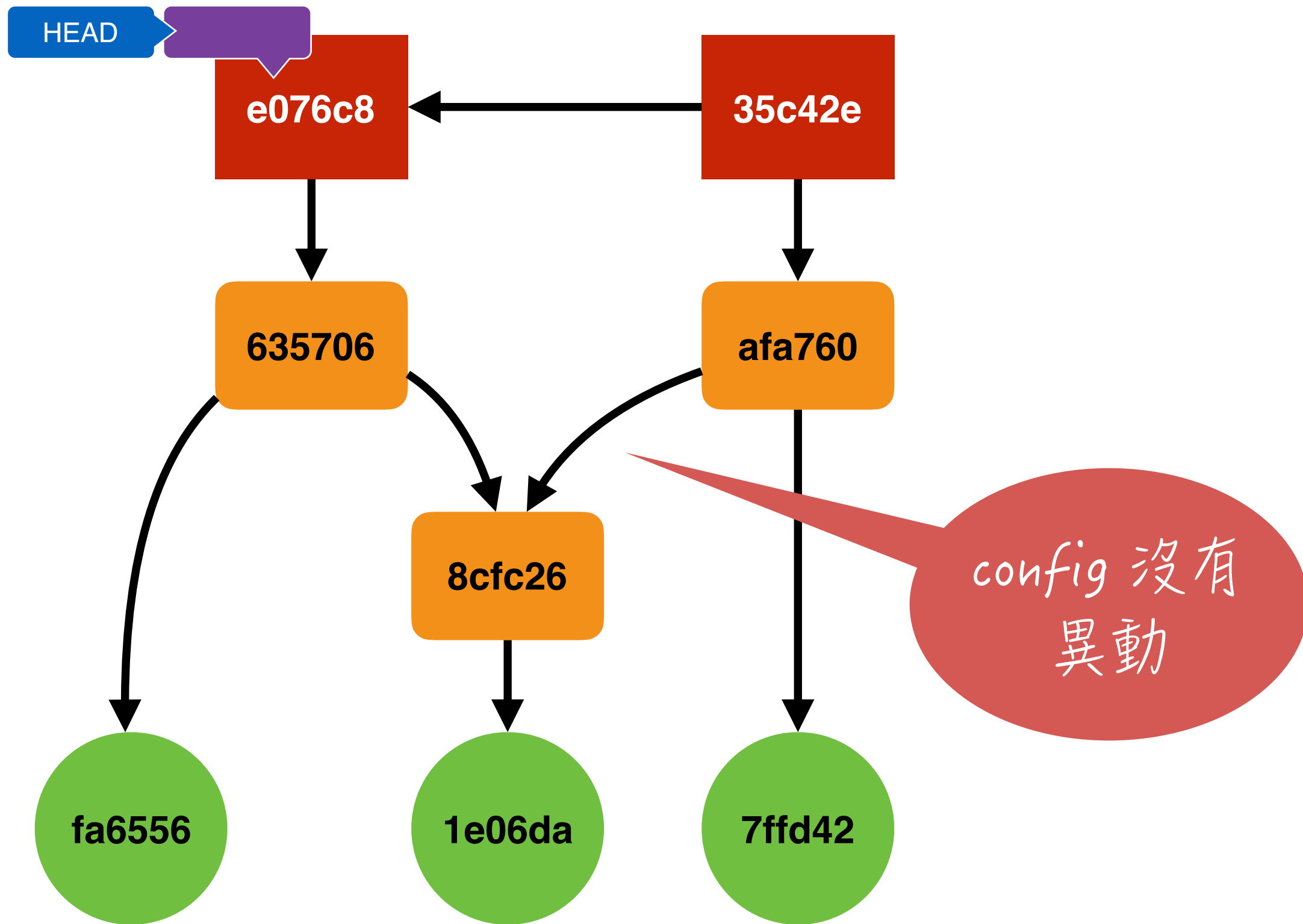


modified
状態

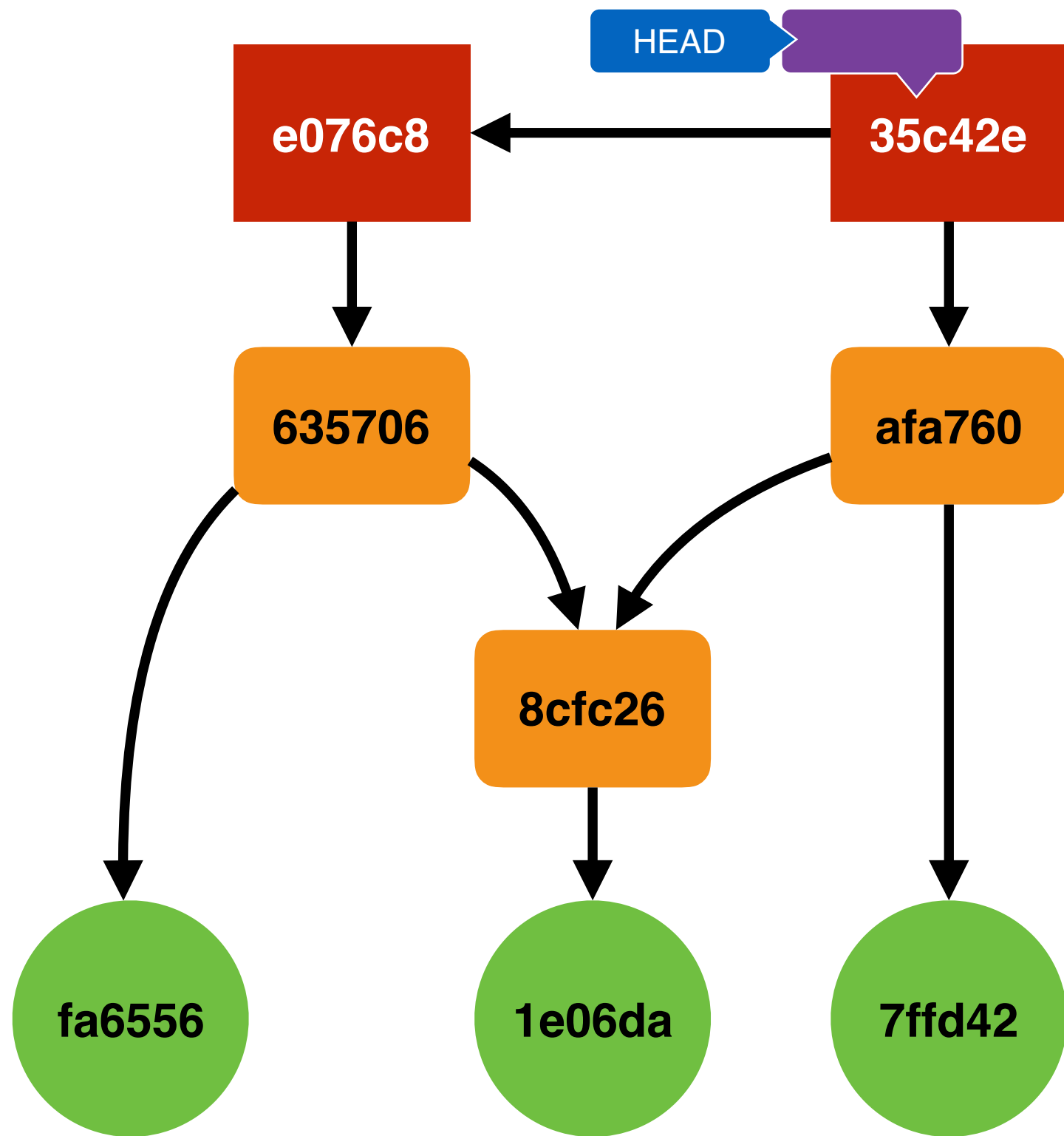
```
$ echo "this is a book" >> index.html
```



```
$ git add index.html
```

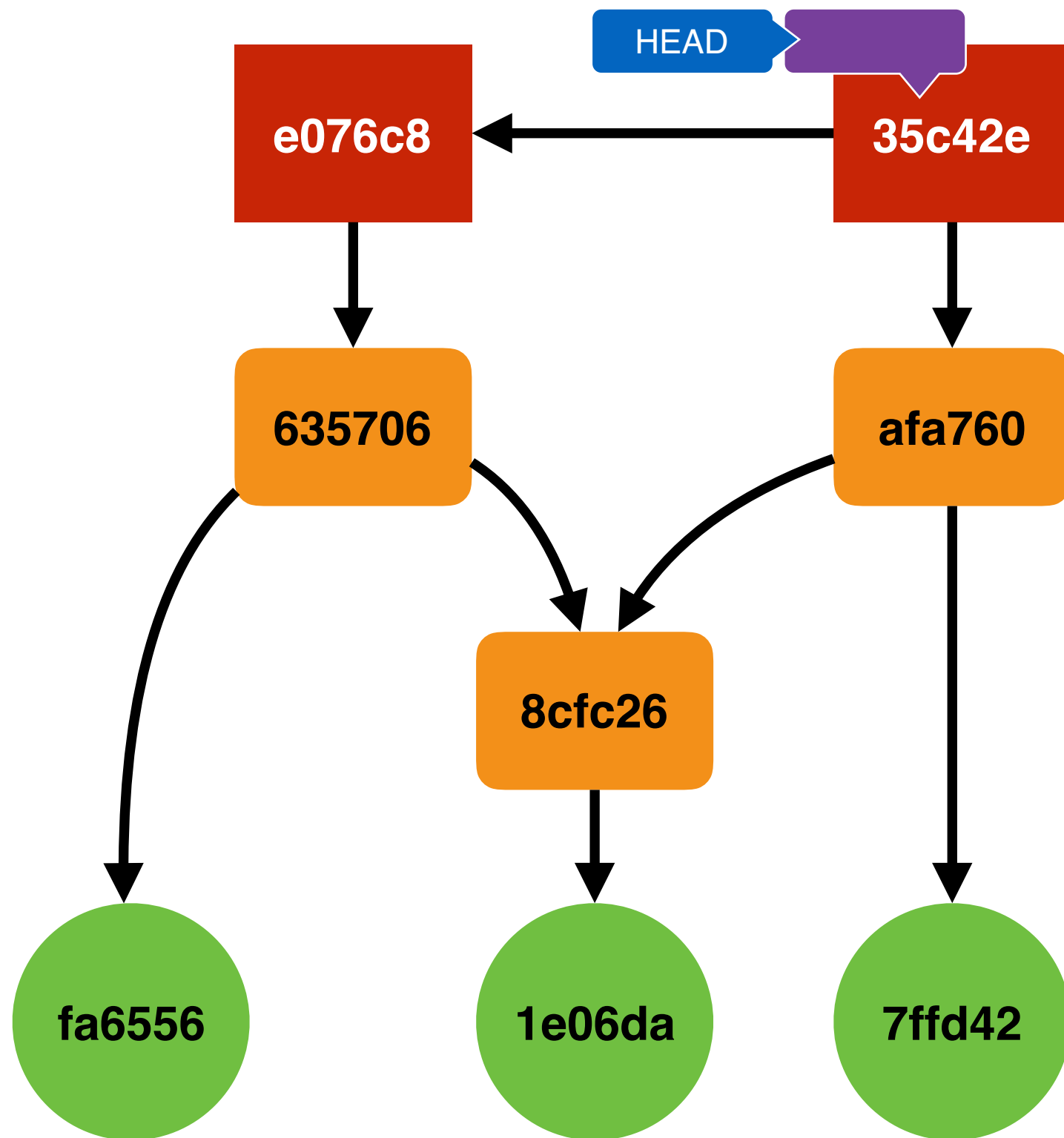



```
$ git commit -m "update index.html"
```



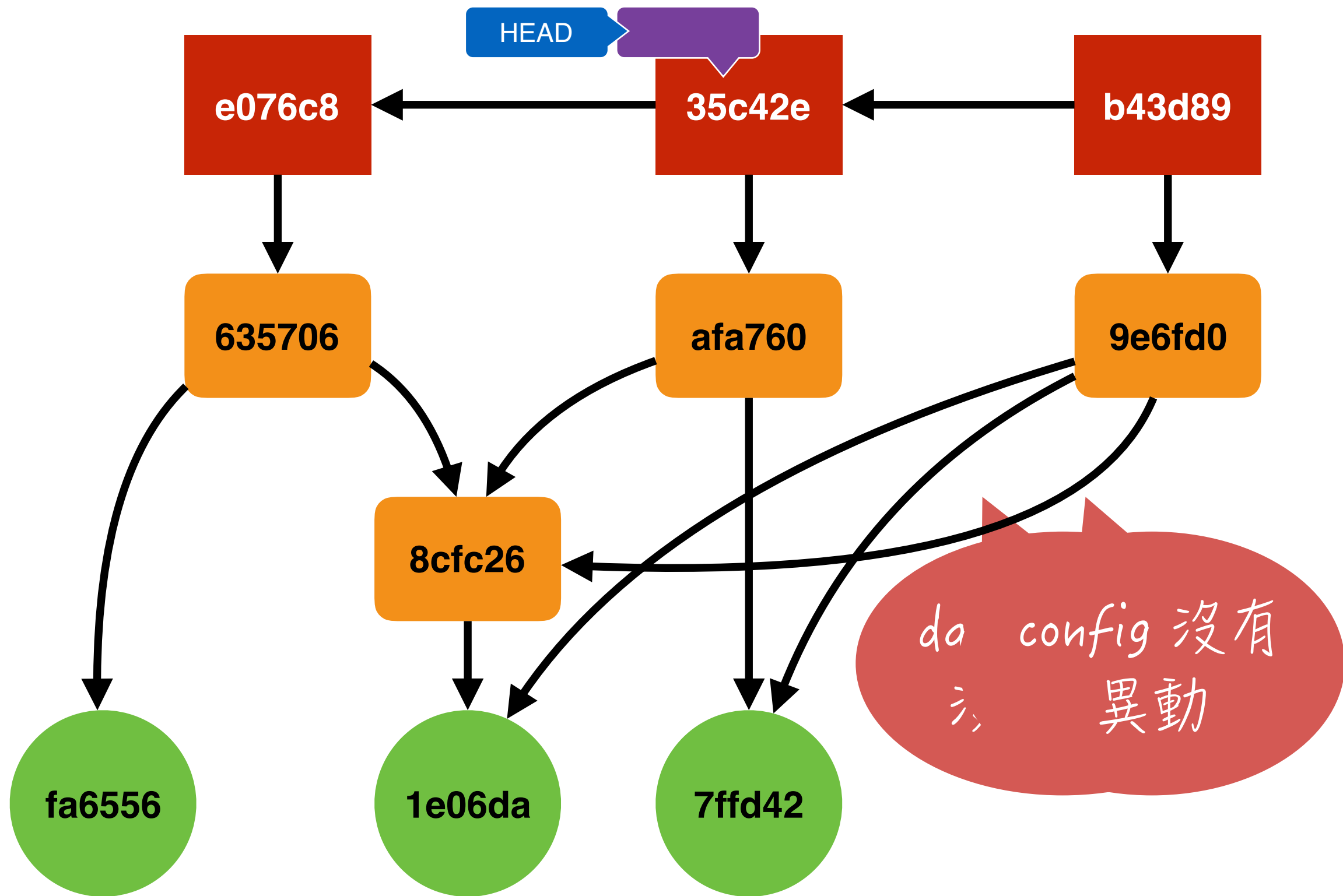
untracked
状態

```
$ echo "super-secret-password" > key.txt
```

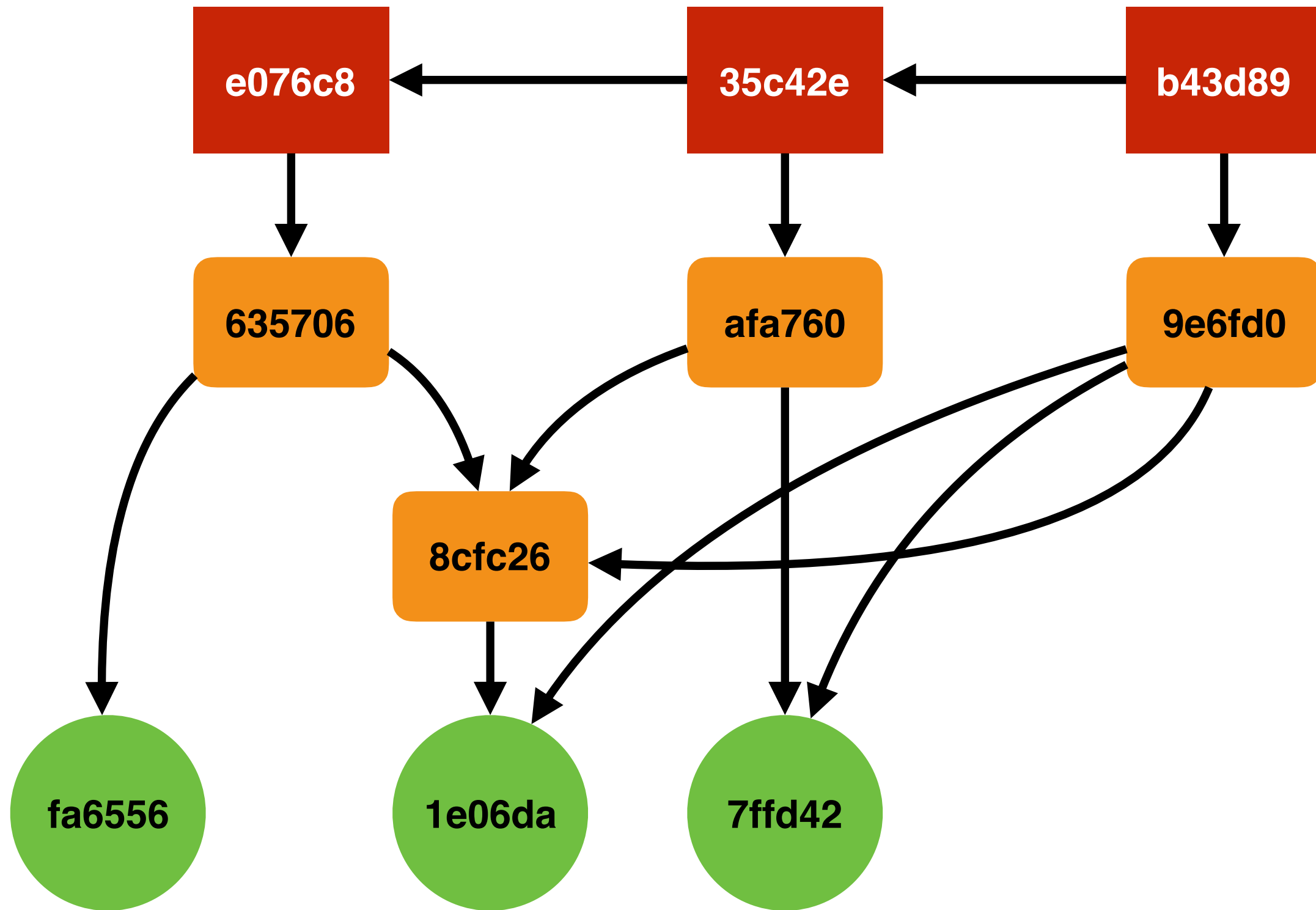


不會產生新的
Blob 物件!

\$ git add key.txt



```
$ git commit -m "add secret key"
```



\$ git count-objects
10

關於 reset!

有獎徵答

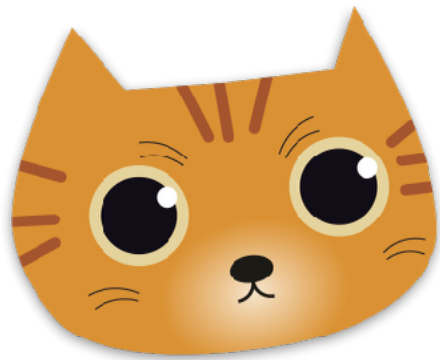


reset 的 3 個參數 `--mixed`, `--soft` 以及 `--hard` 這三個有什麼差別?

關於分支(branch)

對分支的誤解

有用過分支的
請舉個手！



你想像中的分支是什麼樣子？



photo by [Mark Fischer](#)



photo by Caroline

為什麼大家都說
「在 Git 開分支很便宜」？

其實分支只是一個
指向某個 Commit 的指標

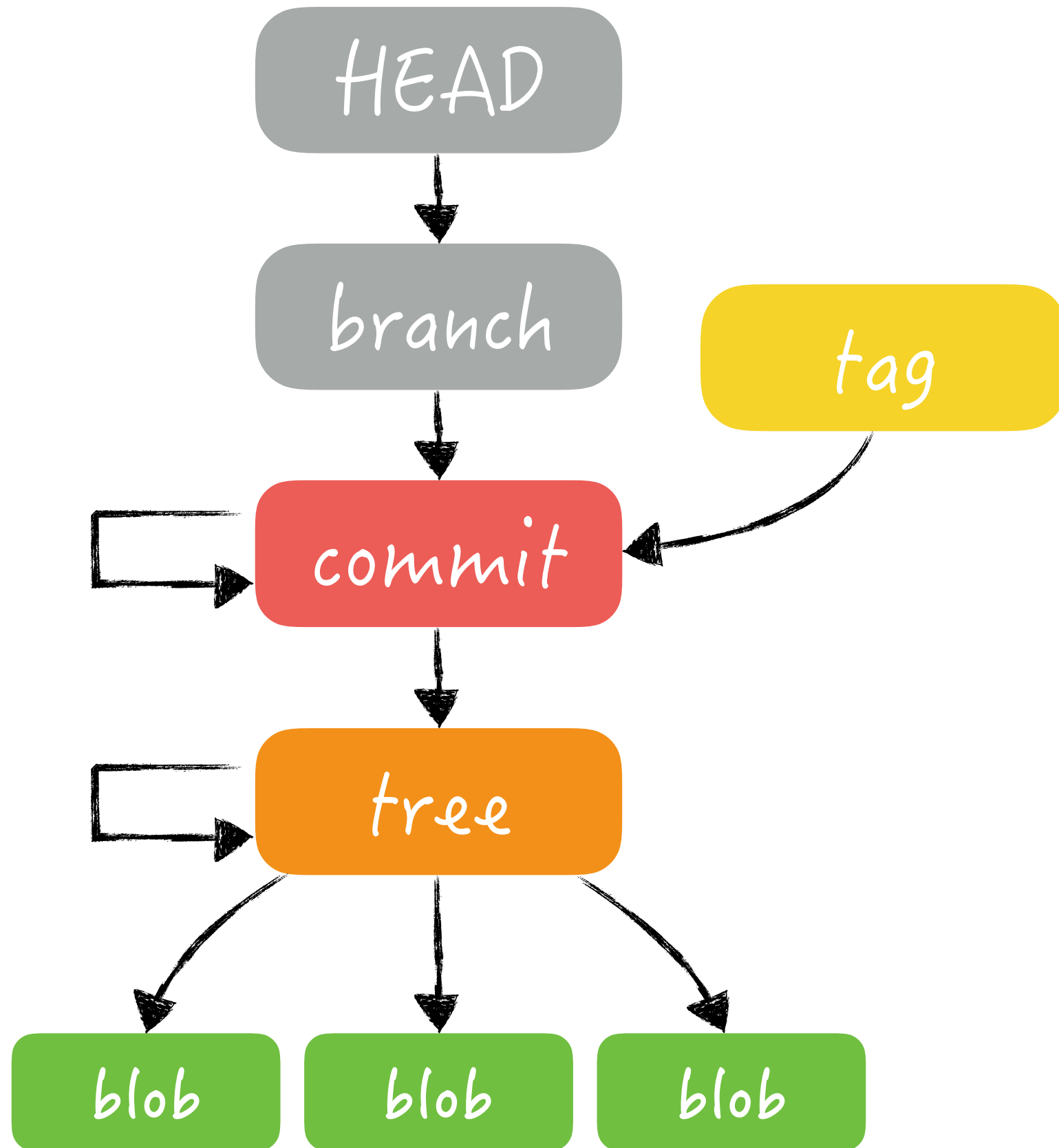
所謂的分支，說穿了就只是一個
有 40 個字元的檔案而已

高 見龍 0_0 master modern-web

{

關於 HEAD

HEAD 只是指向某個分支的指標



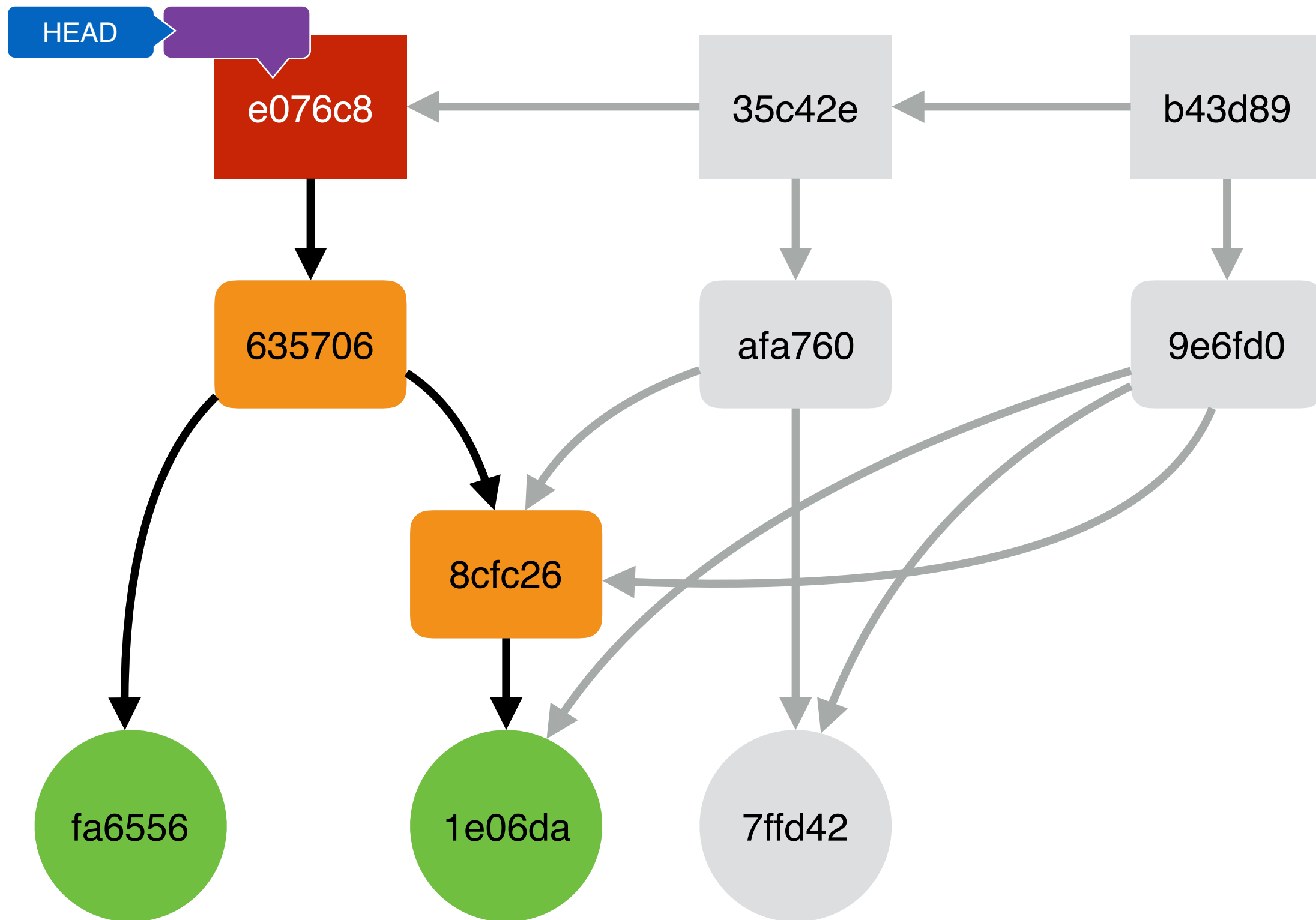
Git 怎麼知道現在是哪個分支？

modern-web

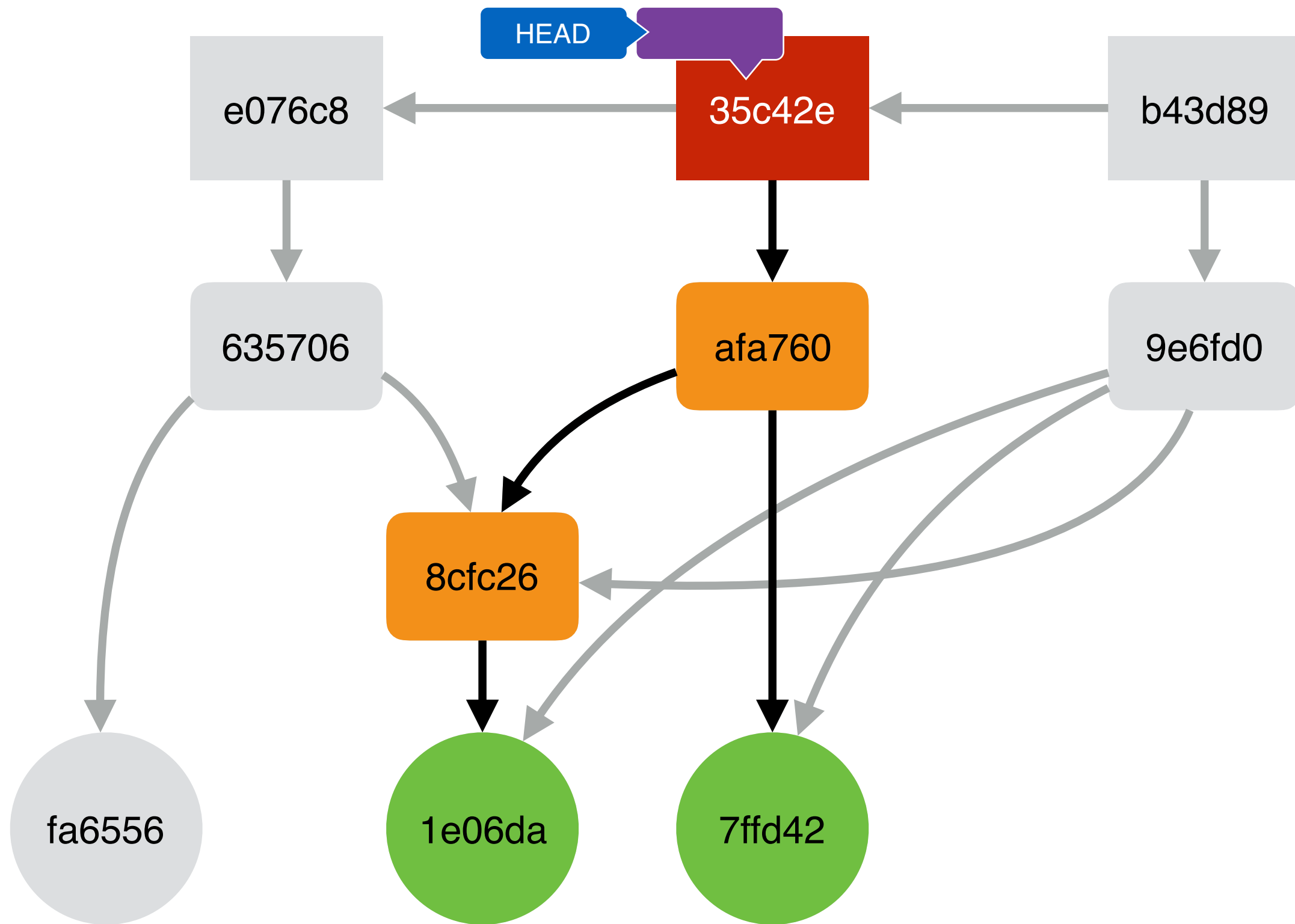
那手動改 `.git/HEAD` 的內容
會發生什麼事？

當執行 "git checkout 某個分支" 指令
的時候，發生了什麼事？

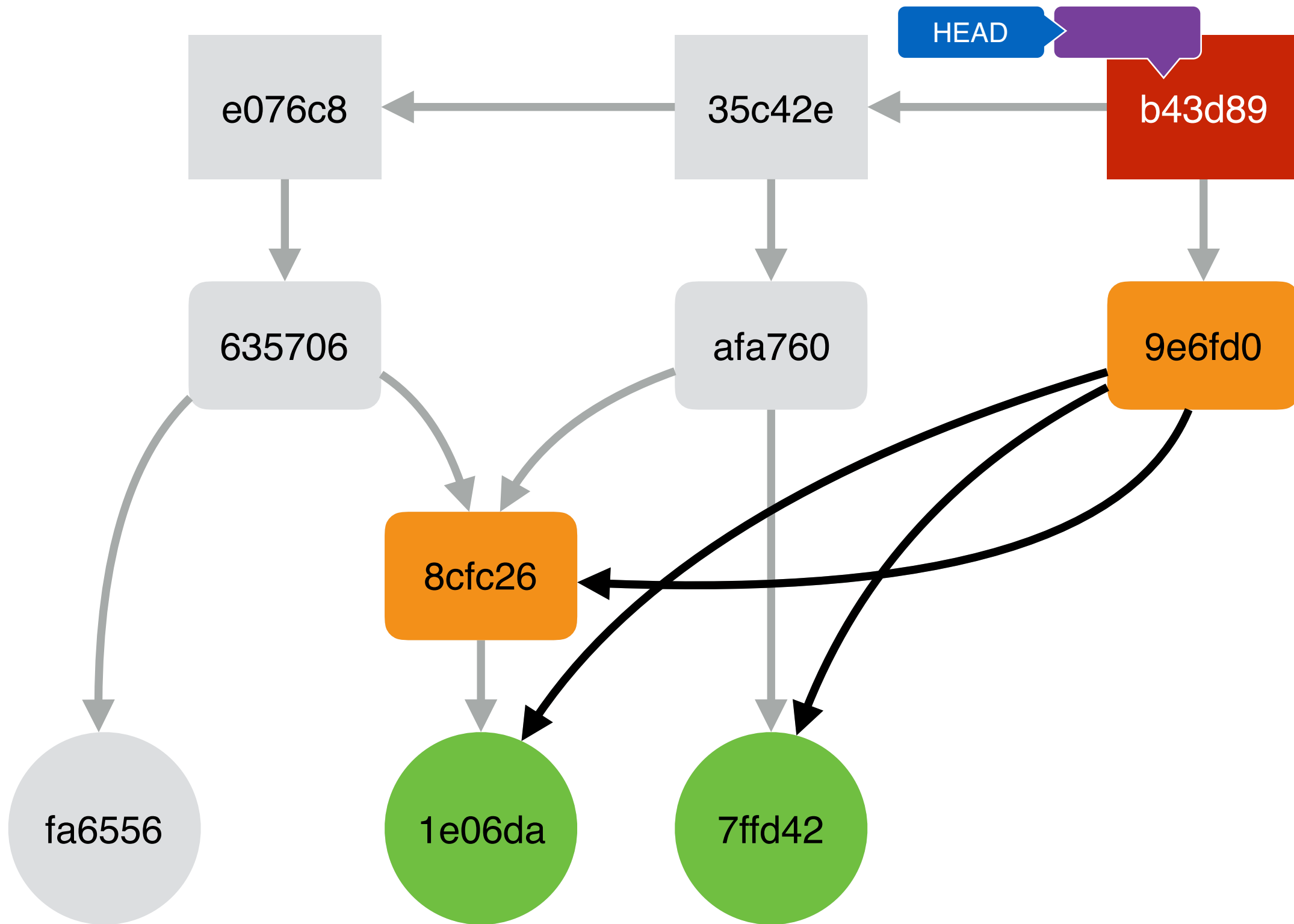
其實分支只是一個
指向某個 Commit 的指標



\$ git checkout e076c8



\$ git checkout 35c42e



\$ git checkout b43d89

1. 把 Repo 裡的東西搬一份出來到
工作目錄

2. 變更 .git/HEAD 內容

有獎徵答



A merge B

跟

B merge A

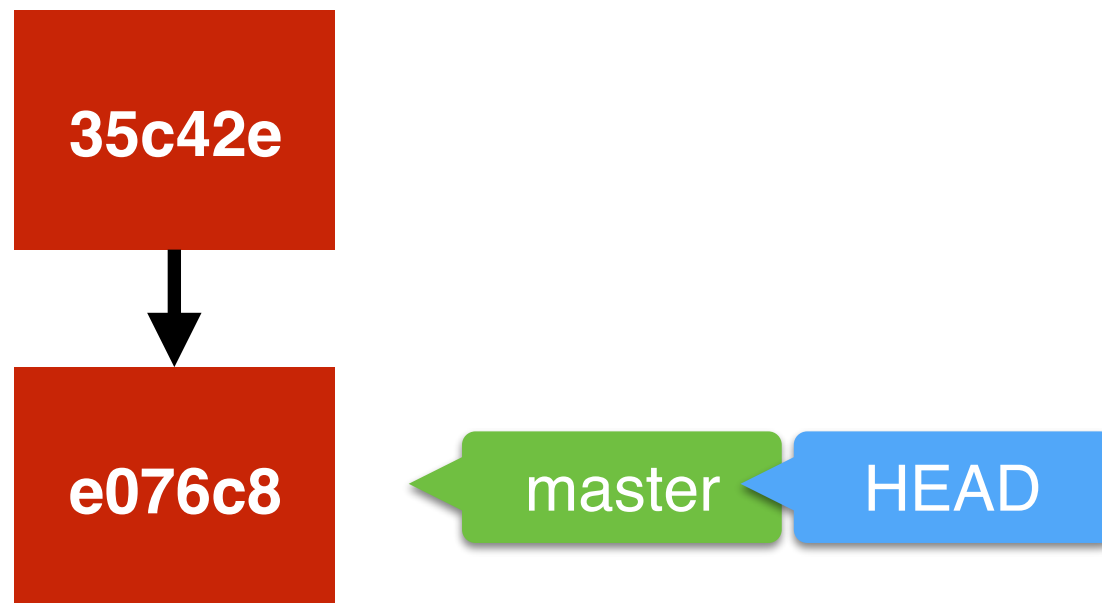
有什麼不一樣?

e076c8

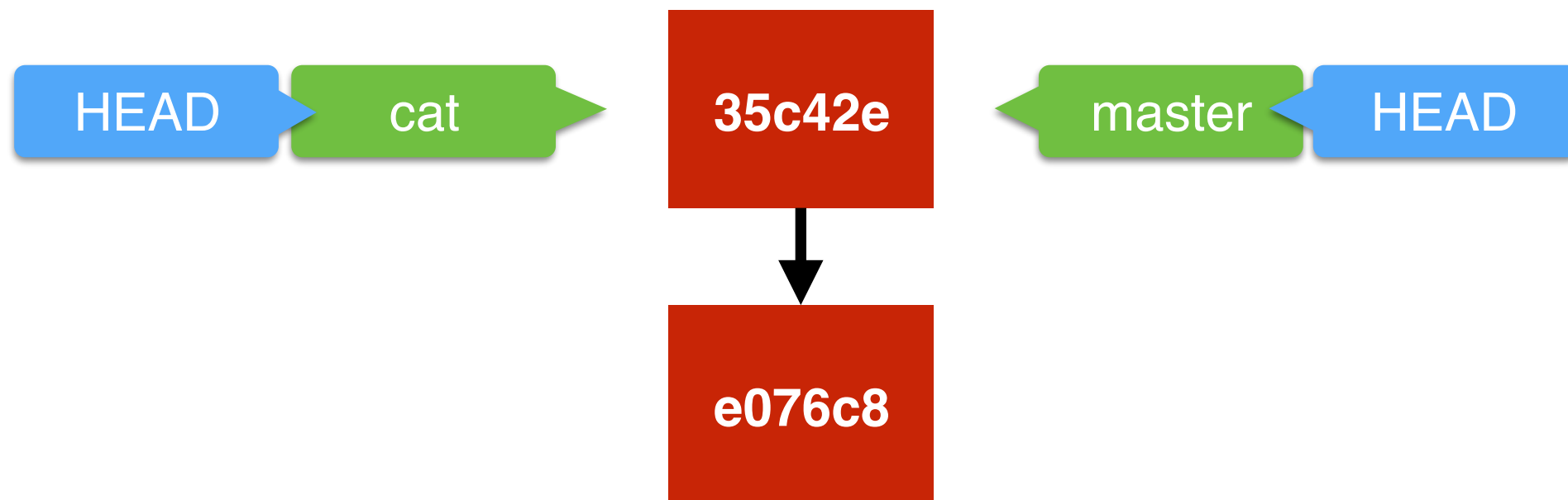
master

HEAD

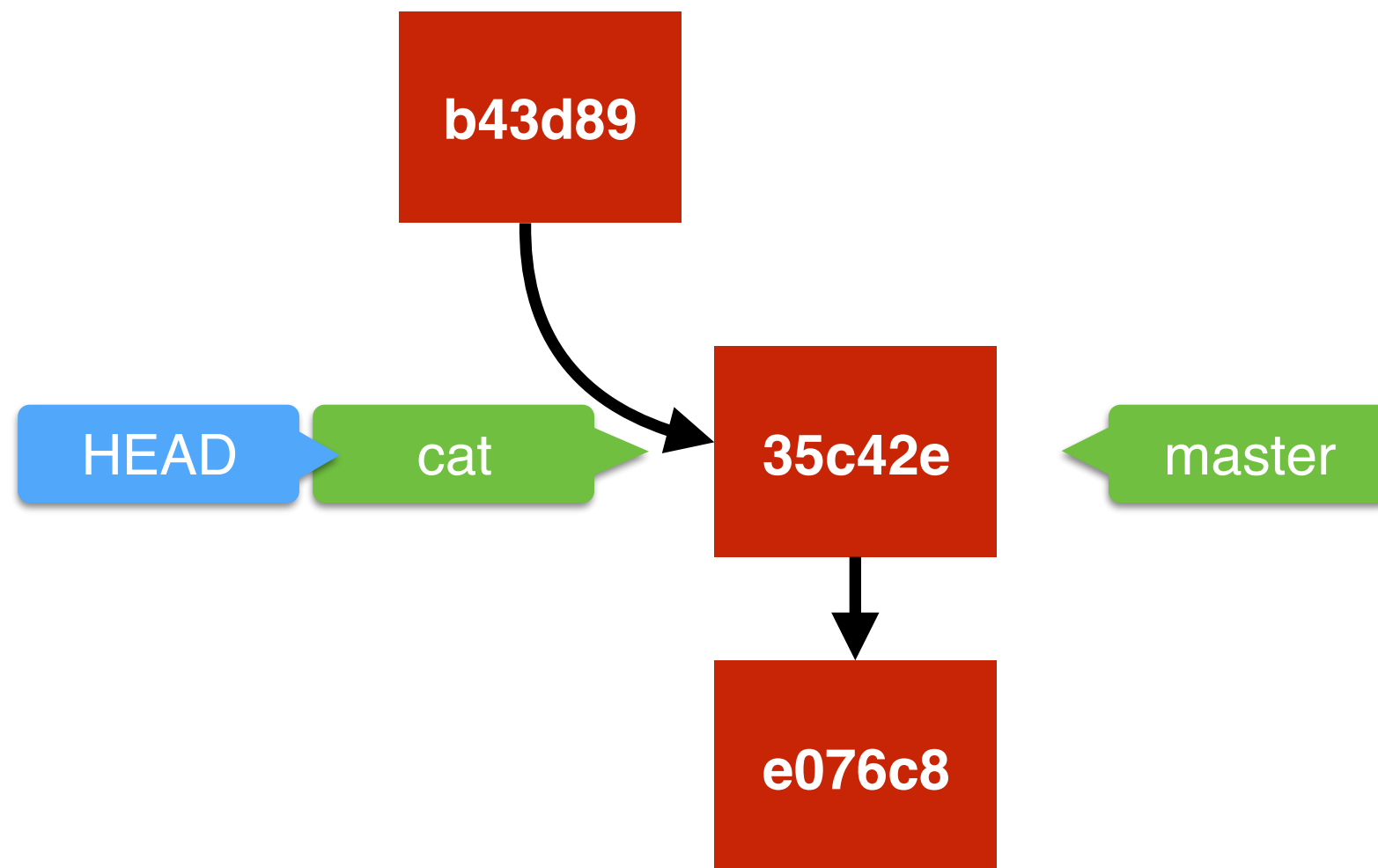
\$



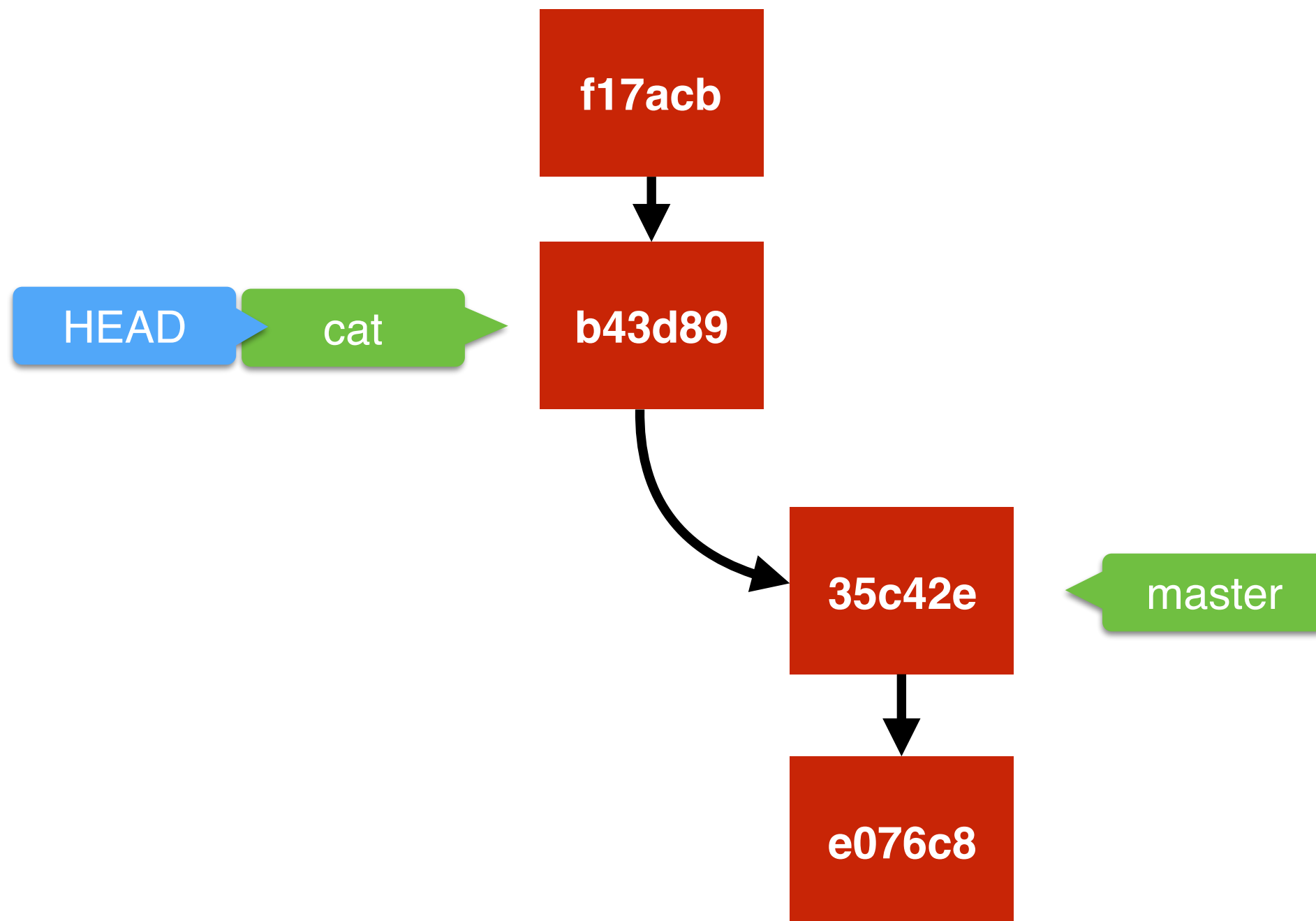
```
$ git commit -m "..."
```



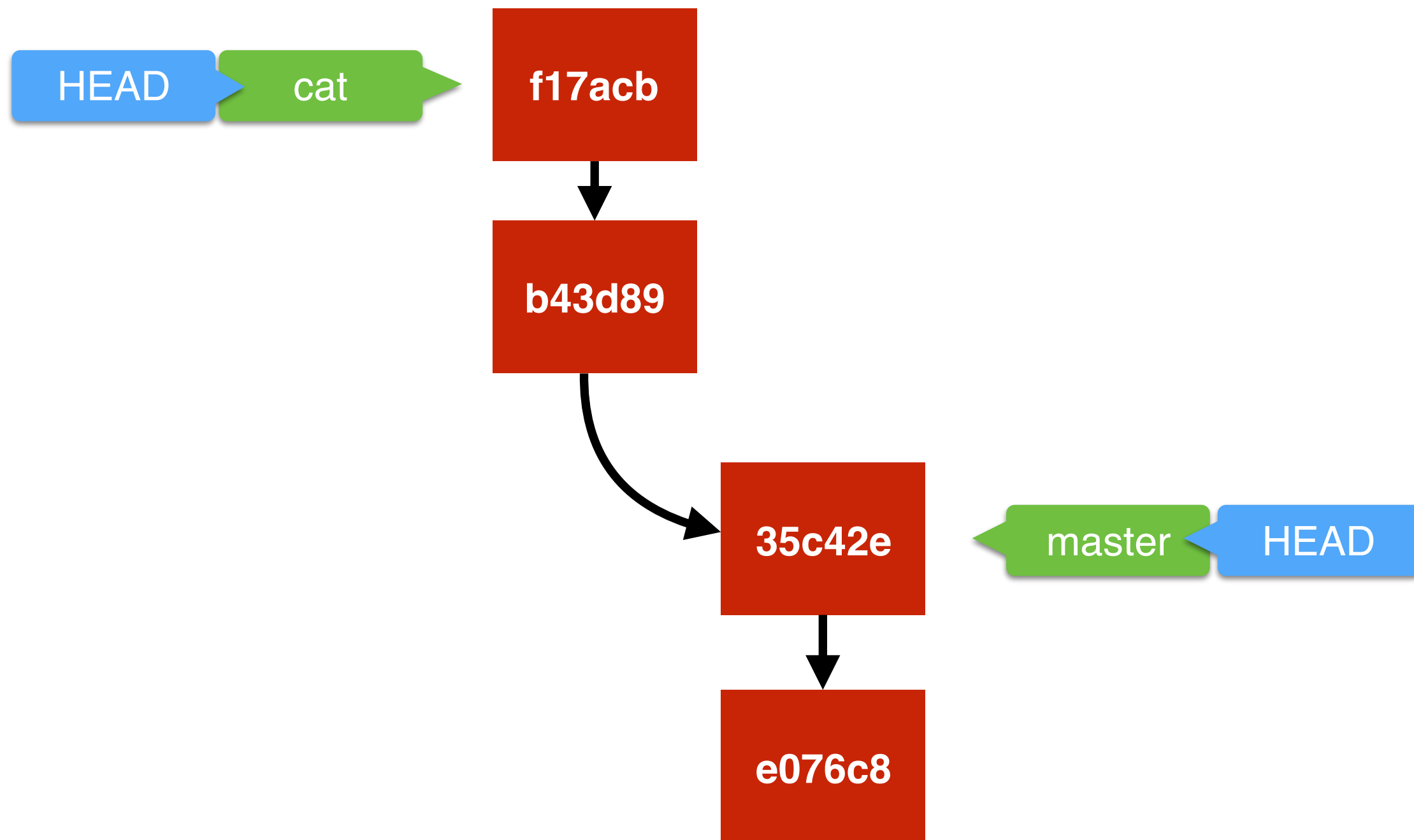
```
$ git checkout -b cat
```



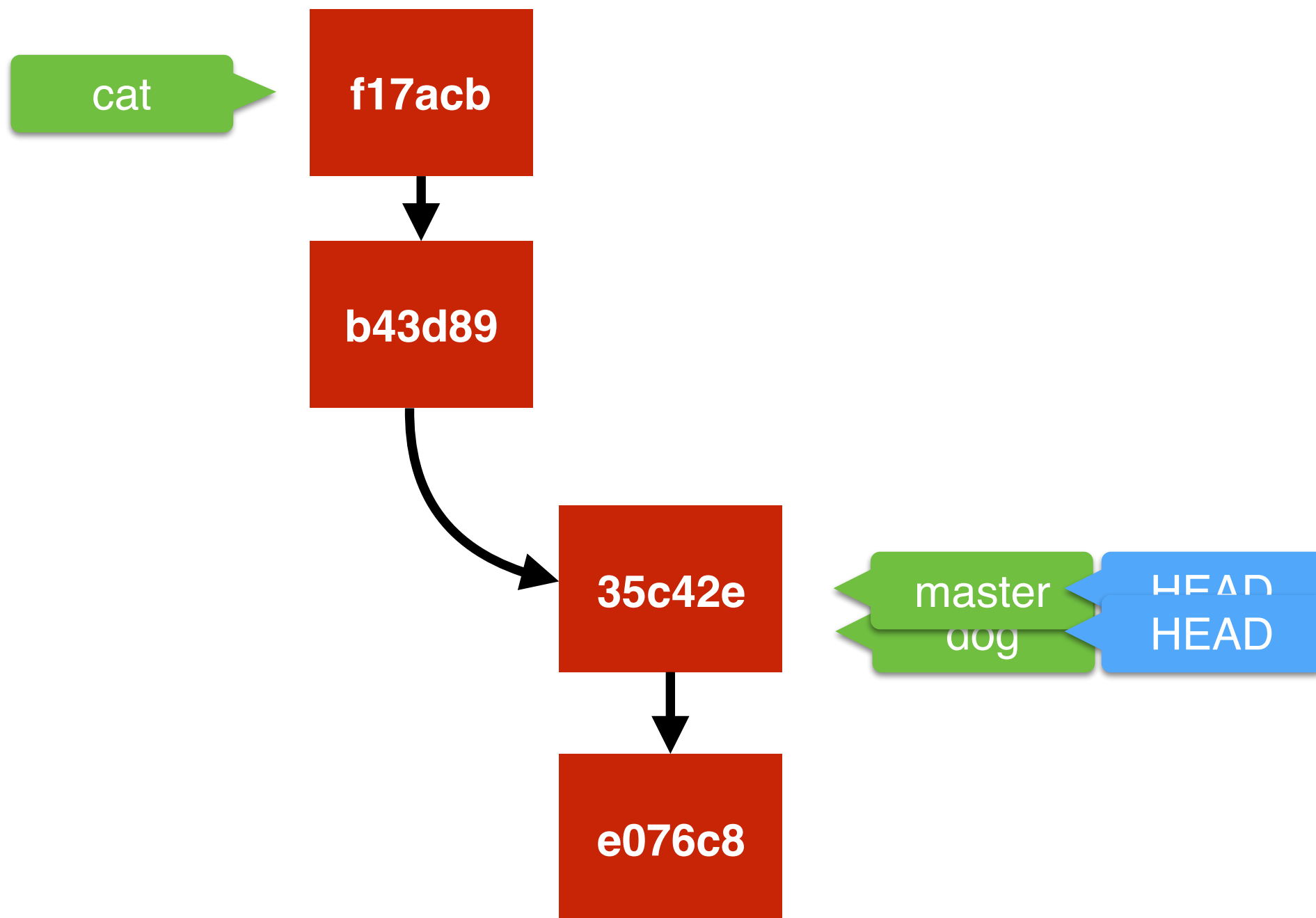
```
$ git commit -m "..."
```



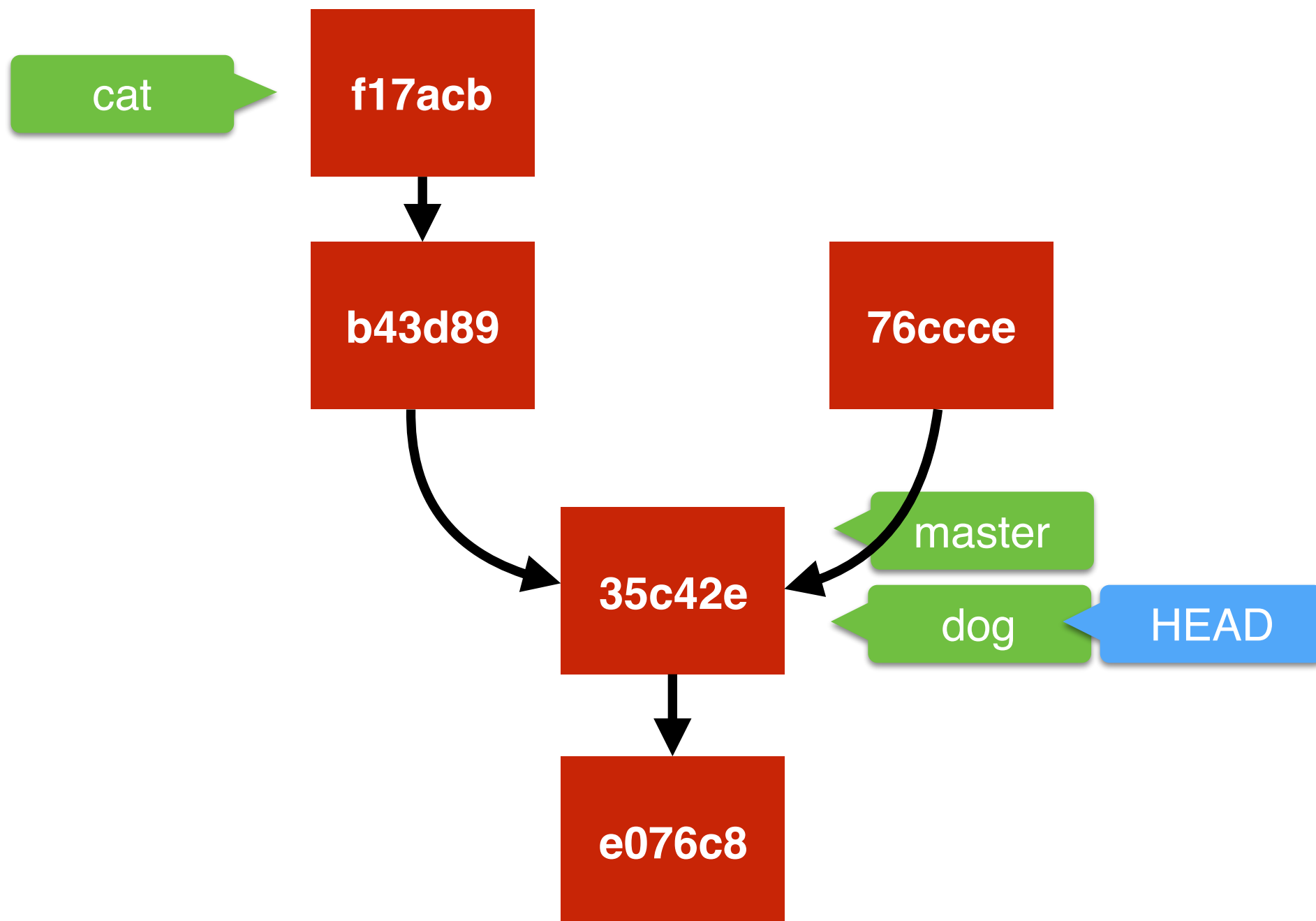
```
$ git commit -m "..."
```



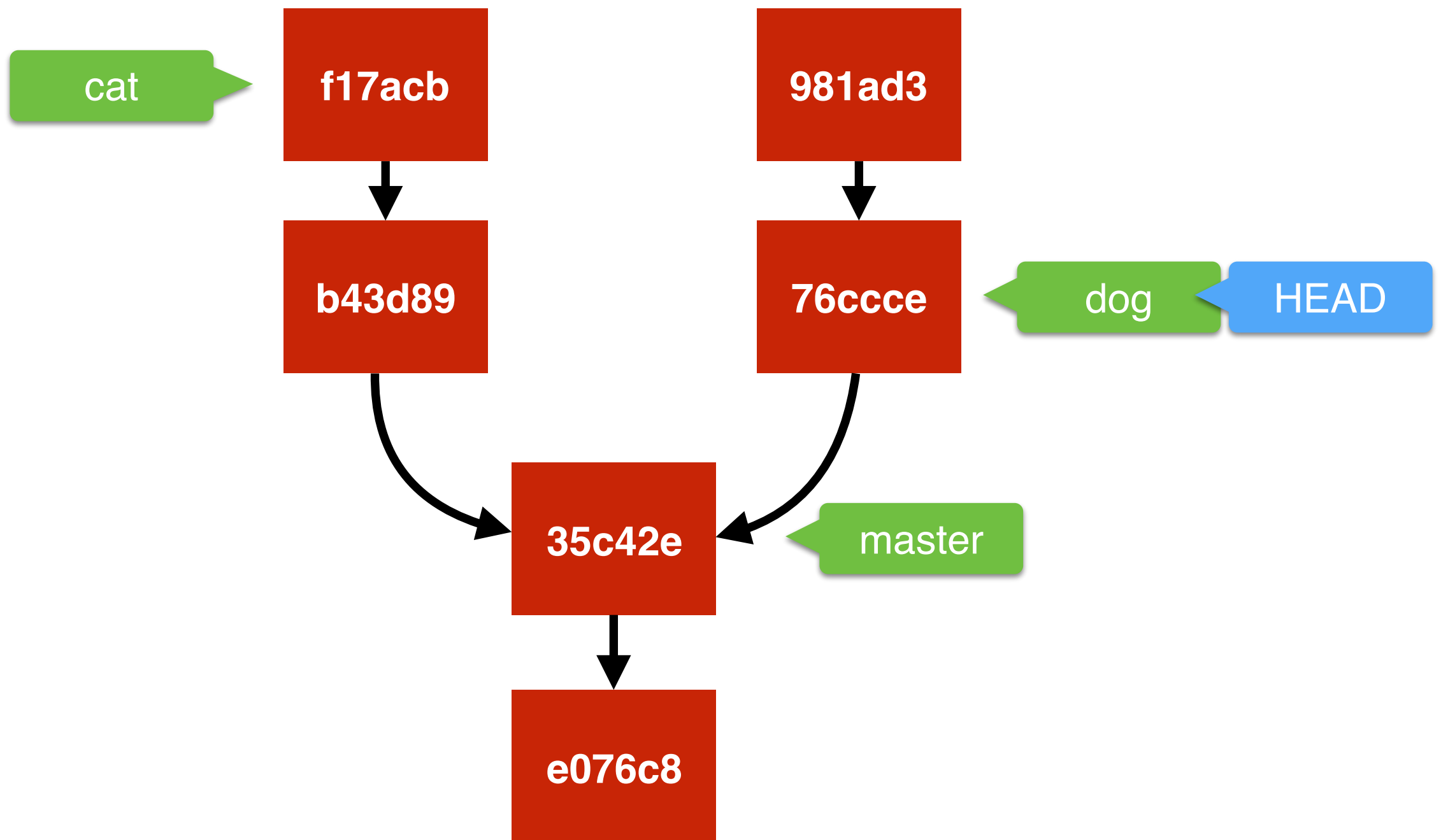
```
$ git checkout master
```



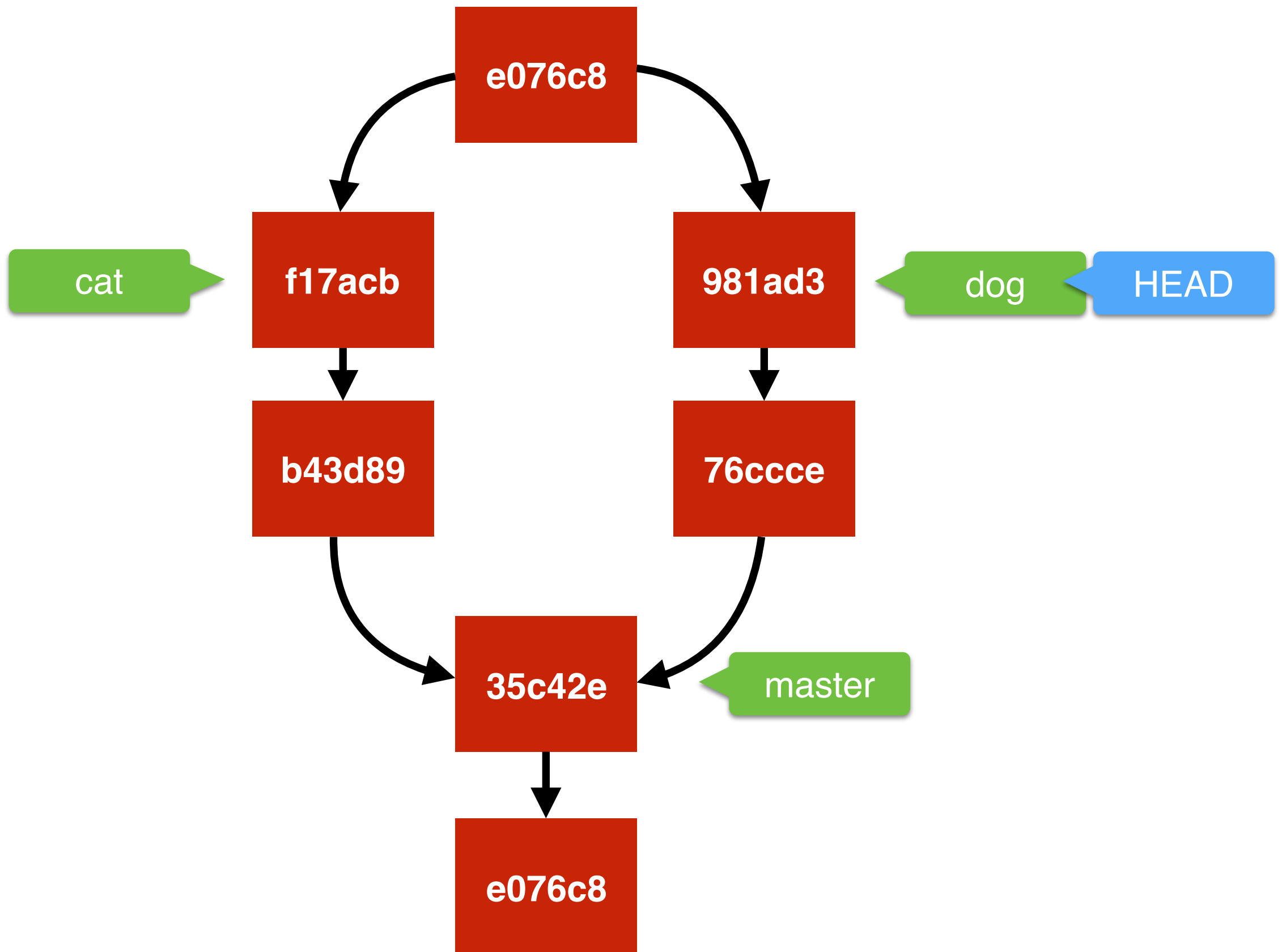
```
$ git checkout -b dog
```

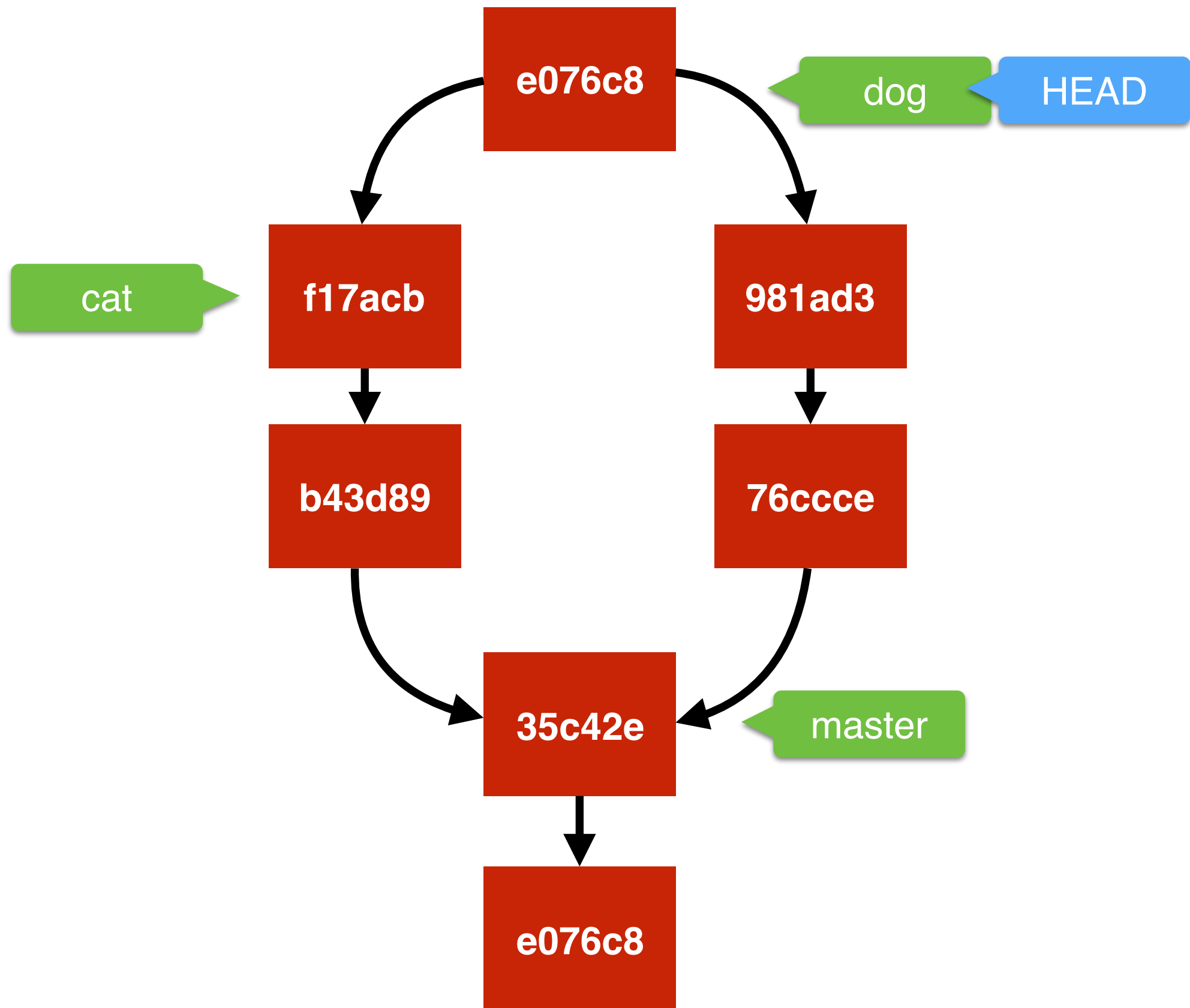
```
$ git commit -m "..."
```



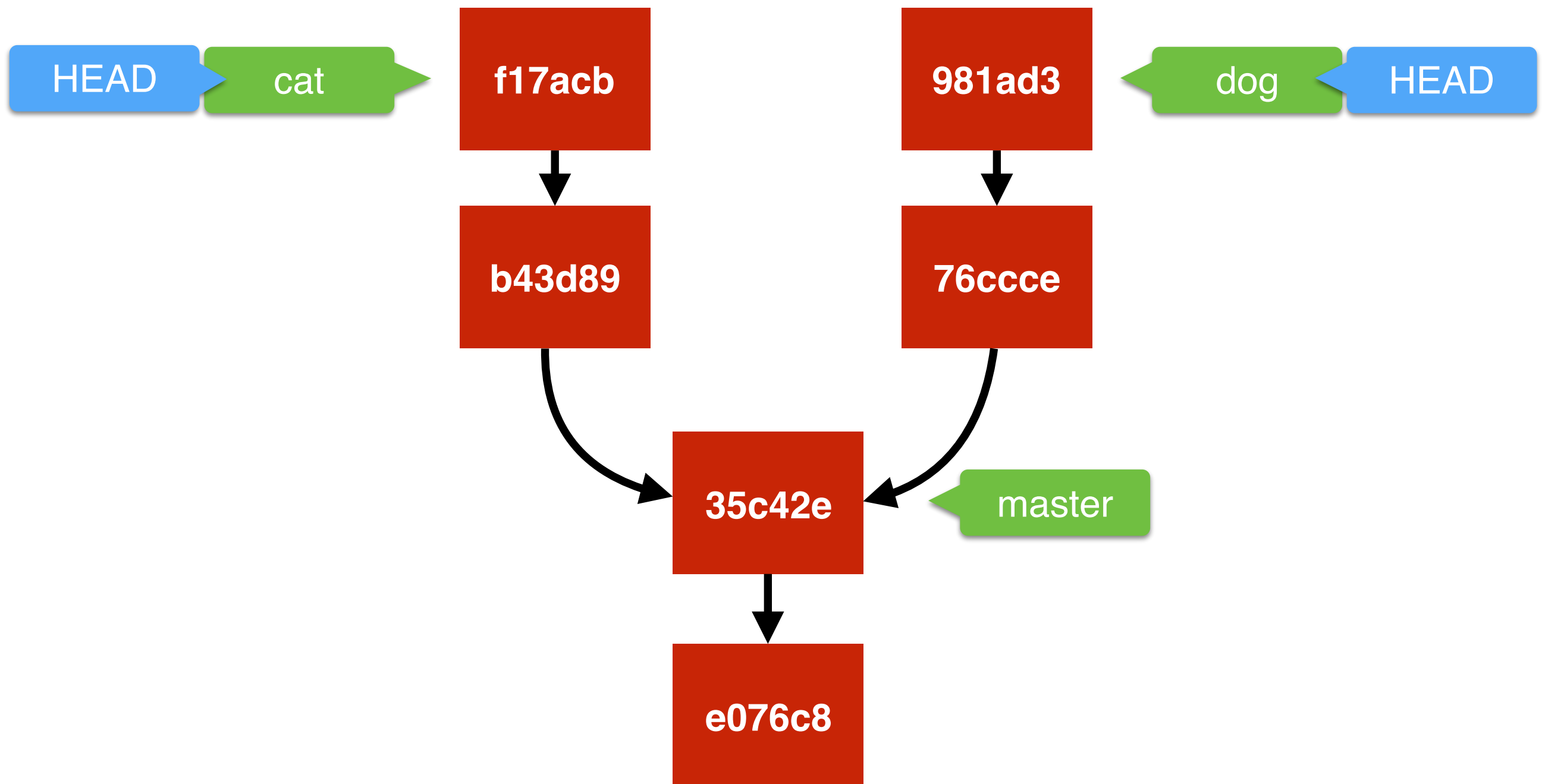
```
$ git commit -m "..."
```



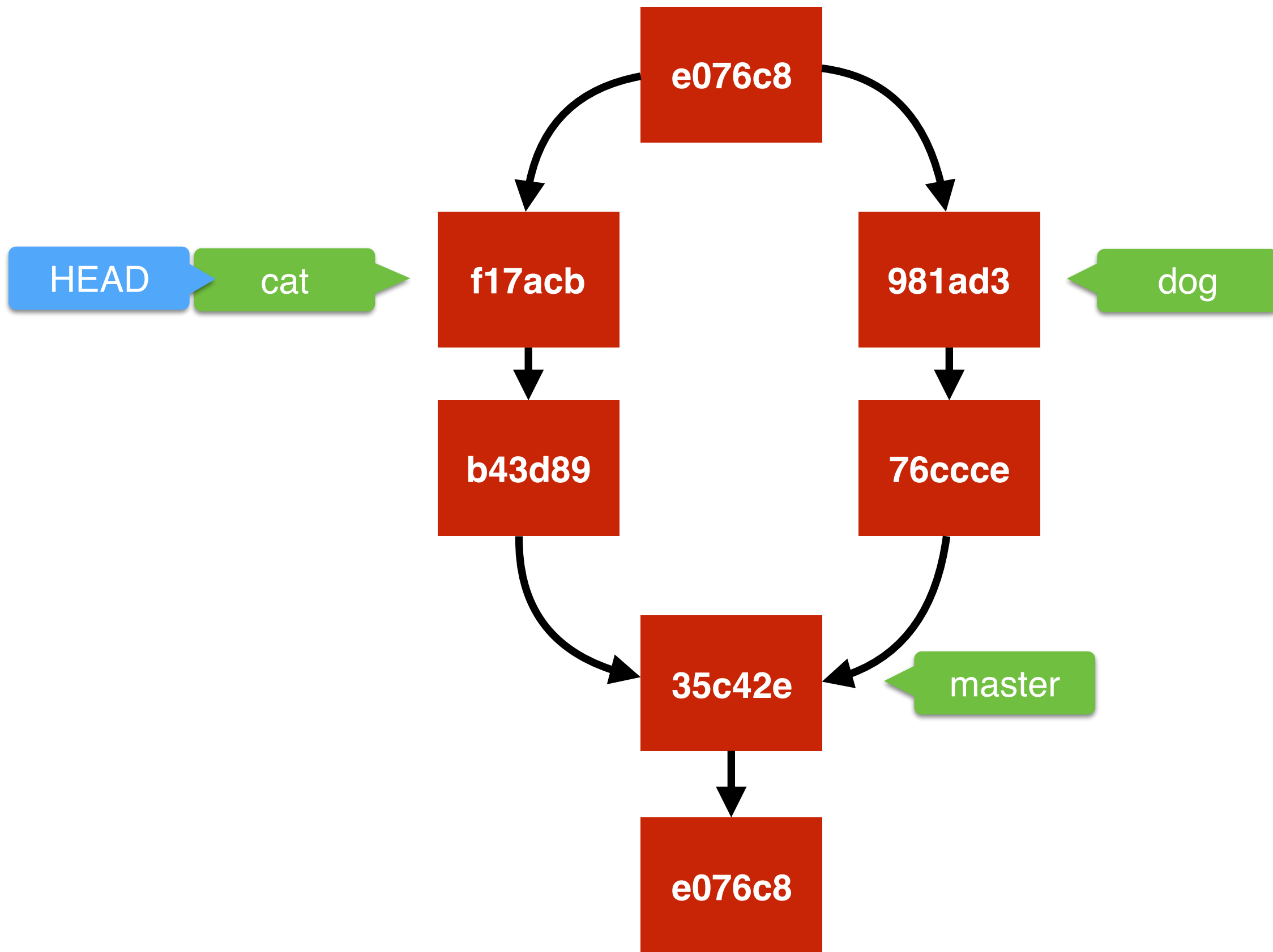
\$ git merge cat



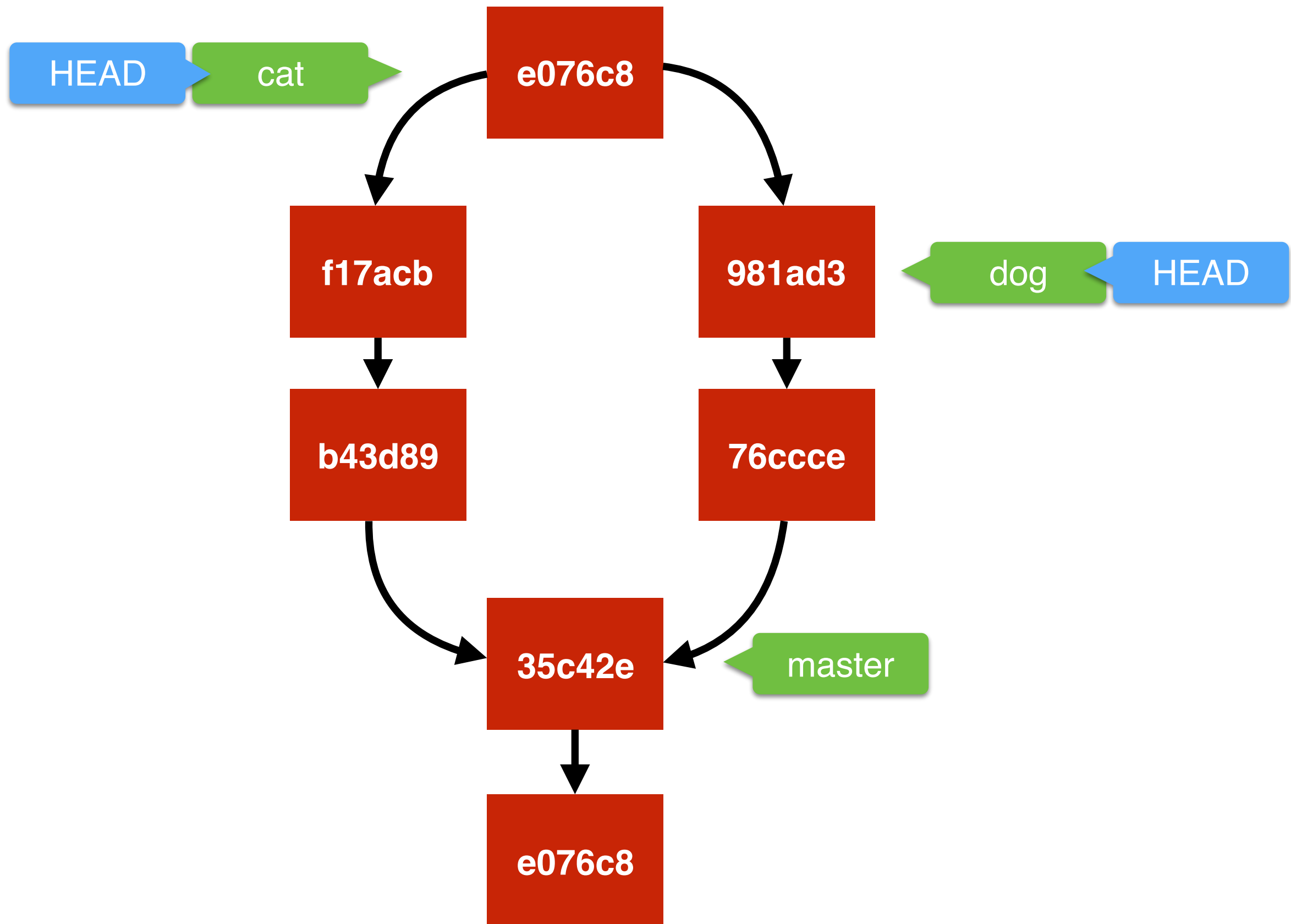
```
$ git reset HEAD^ --hard
```



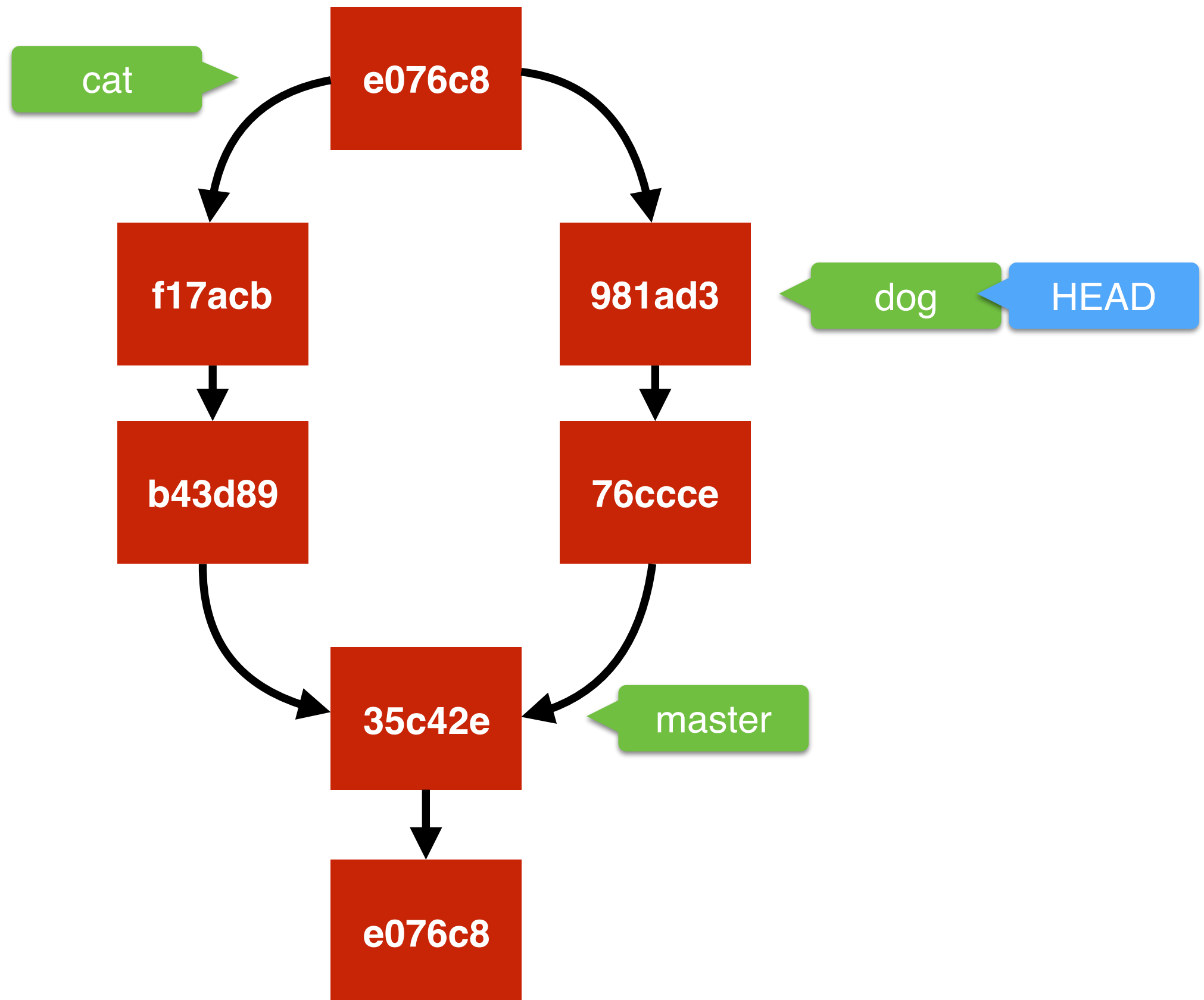
\$ git checkout cat



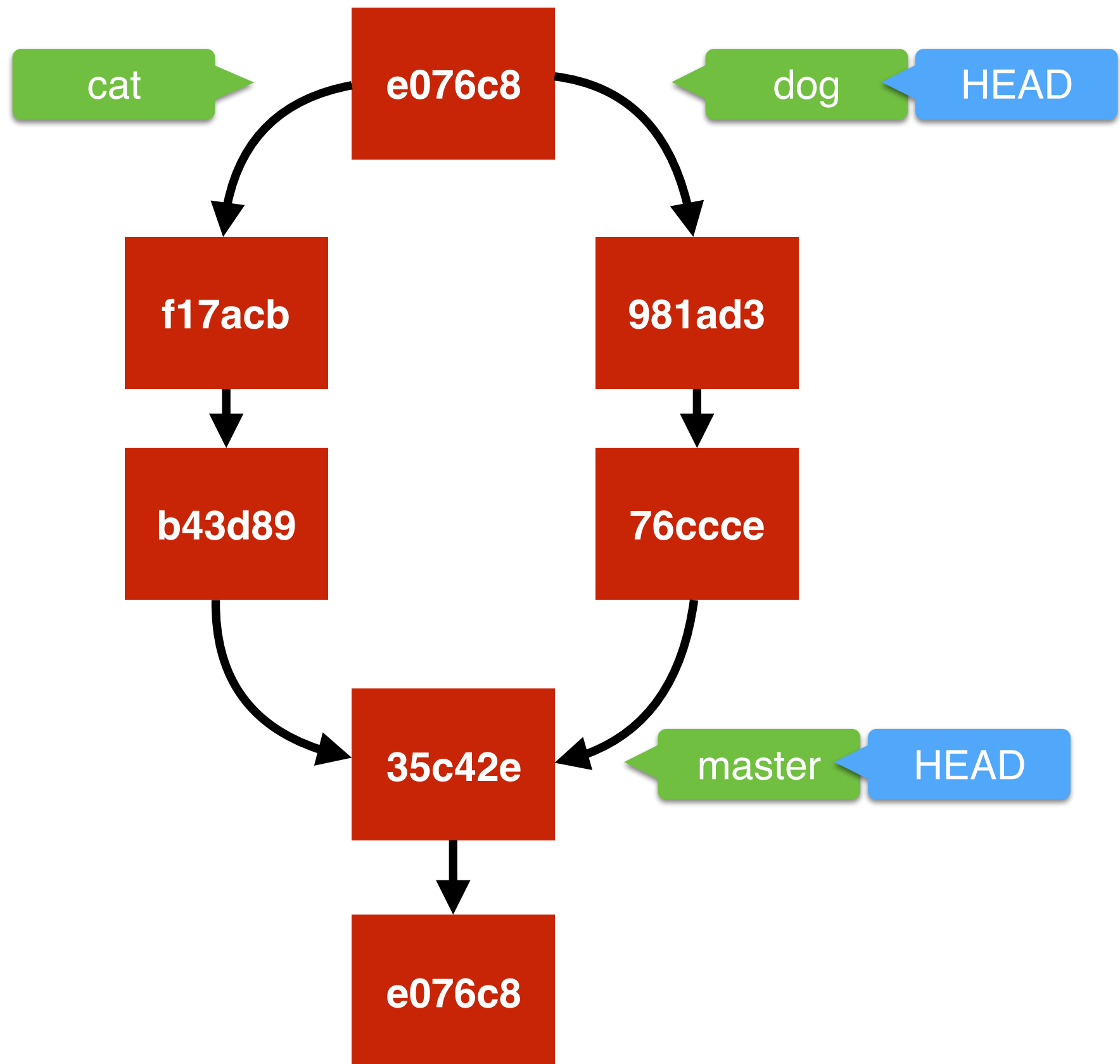
\$ git merge dog



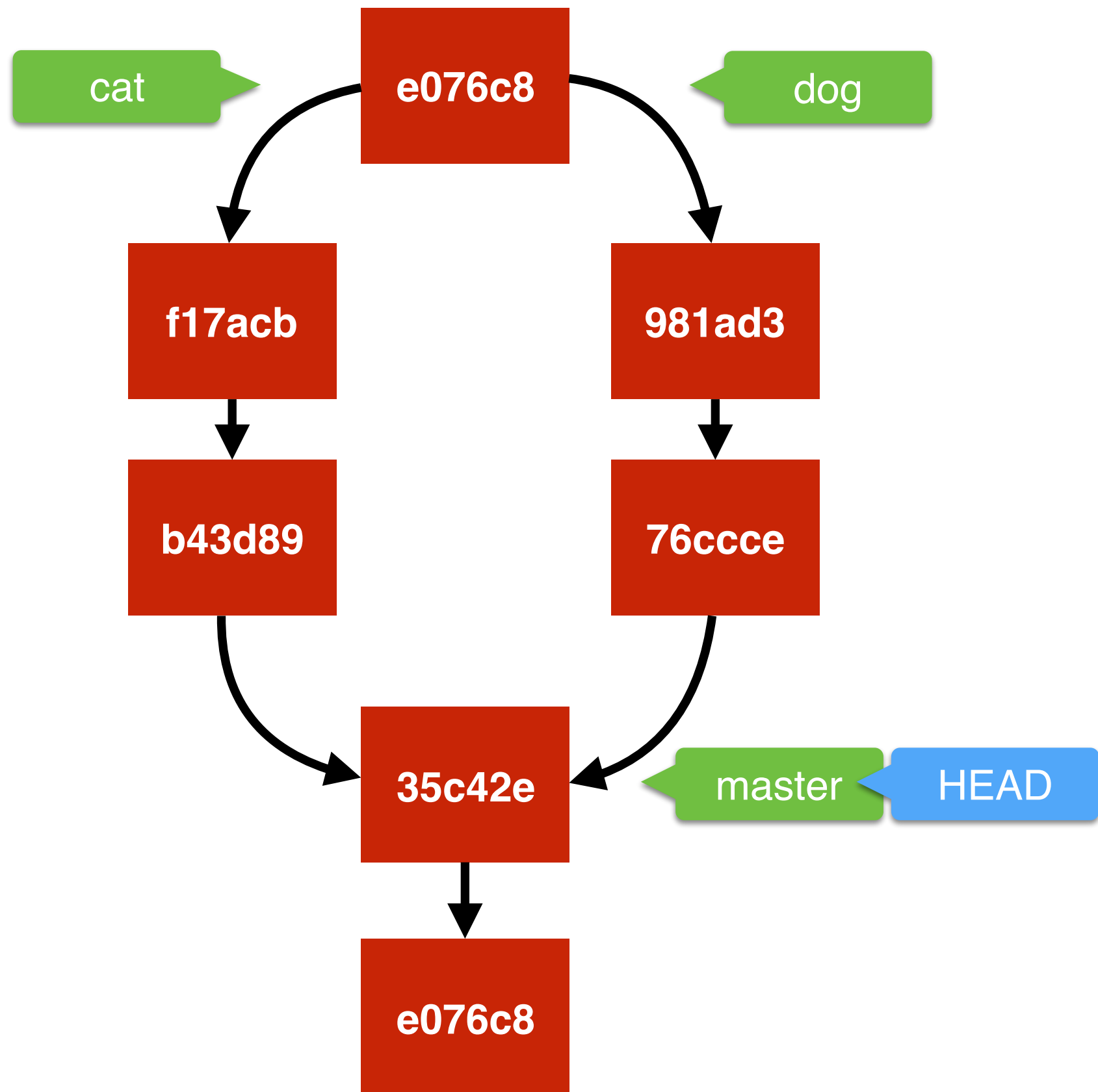
\$ git checkout dog



\$ git merge cat



\$ git checkout master

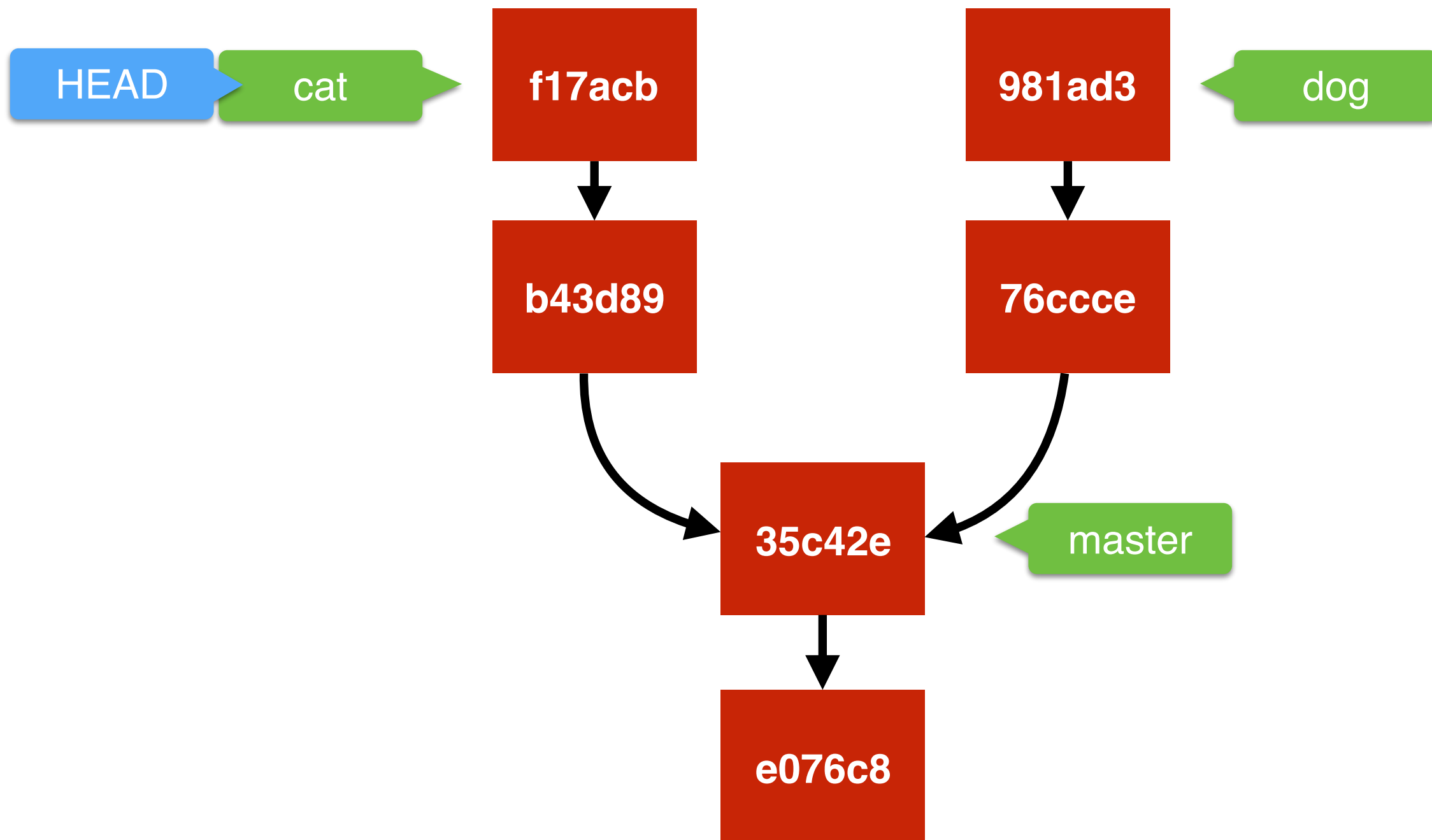


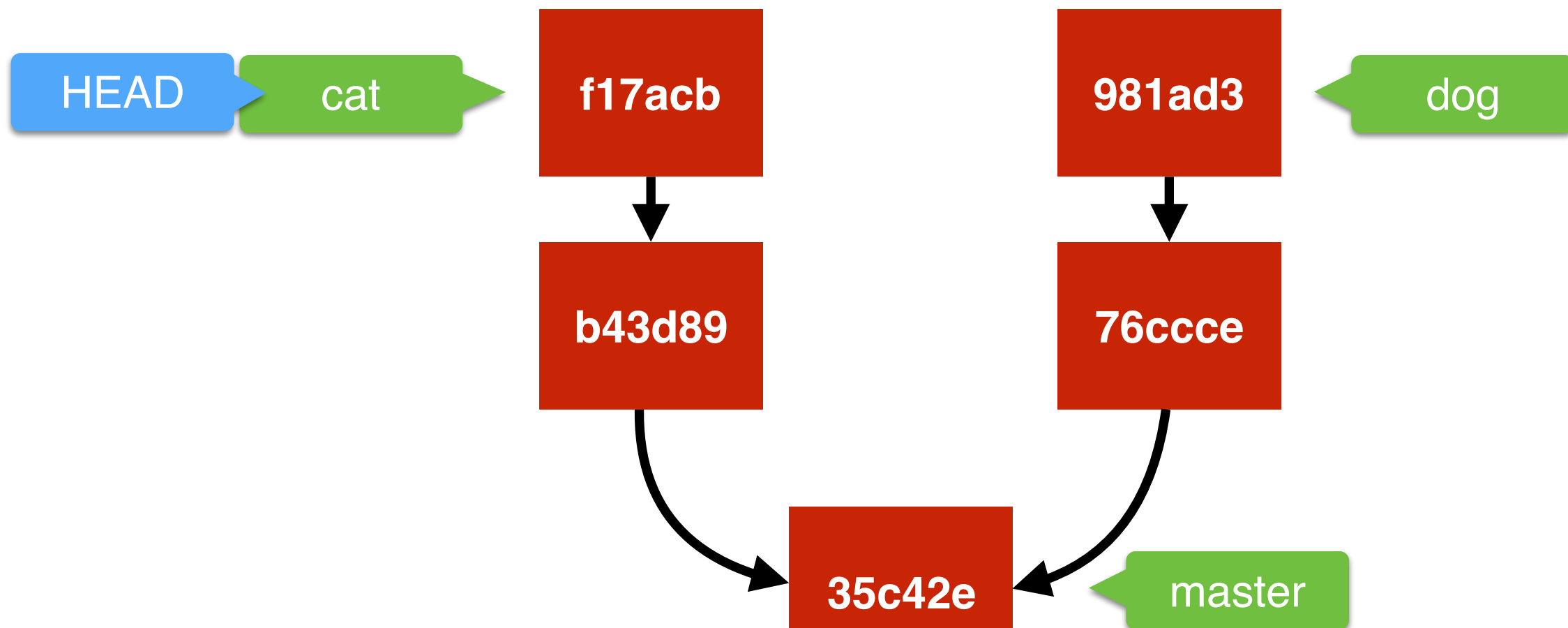
\$ git merge dog

有感受到 Checkout 跟 merge 的時候
有什麼變化嗎？

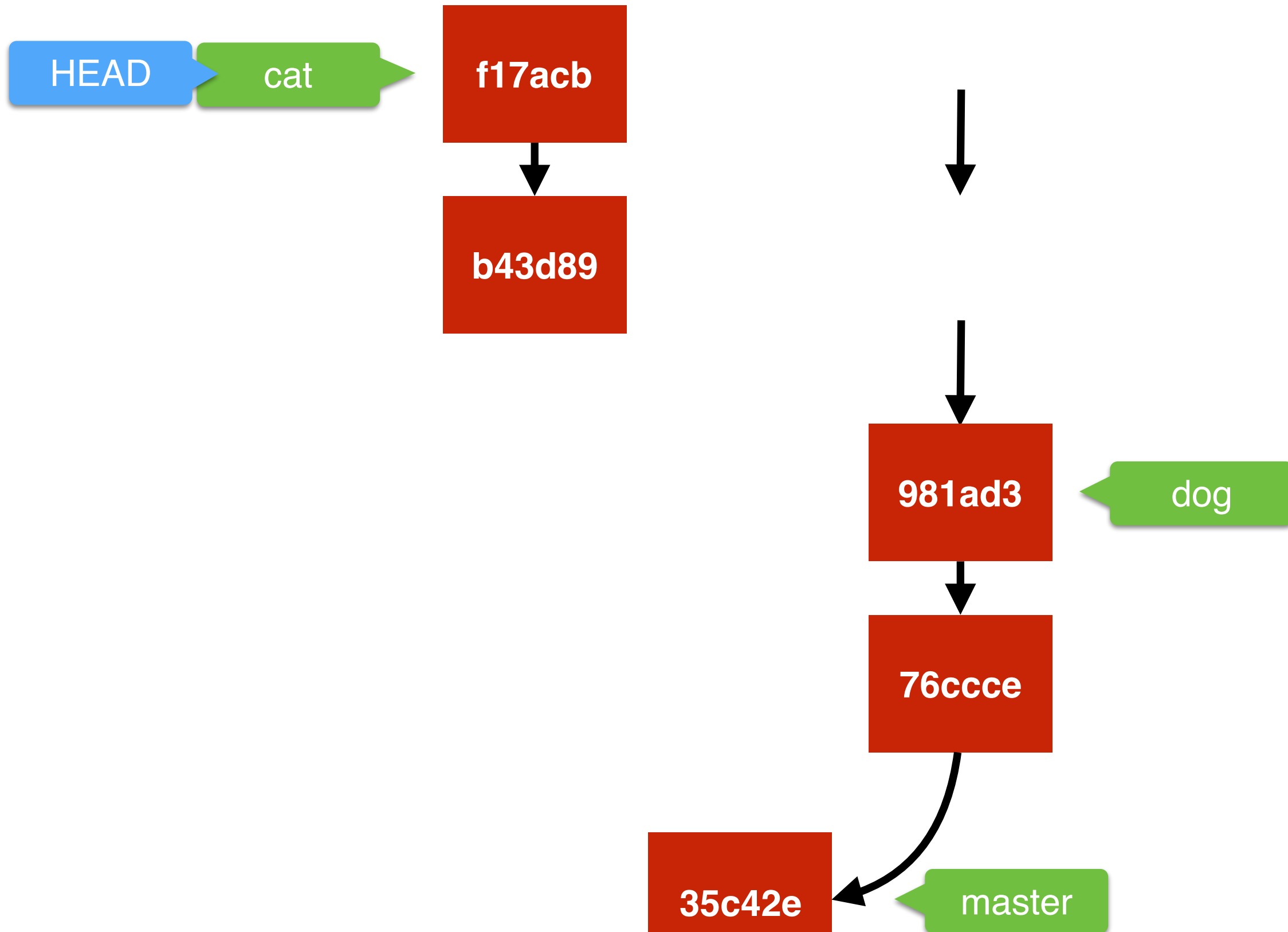
就是分支跟 HEAD 在移來移去而已

rebase v.s merge





\$ git rebase dog



\$ git rebase dog

使用時機

分支一定要叫 Master 嗎?

有獎徵答



合併過的分支可以砍掉嗎？
為什麼？

Master 分支可以砍掉嗎？

其它這些分支
就像一張貼紙一樣而已

其實東西進不了 Git，
想要刪還不太容易...

把還沒合併的分支刪掉
救得回來嗎？

1. eddie@gaojiande-MacBook: /private/tmp/modern-web (zsh)

高 見龍 0_0 master ✓

modern-web

被 Hard Reset 的話
救得回來嗎？

🍏 高見龍 0_0 ➡ ↶ master ✓

mo

關於標籤(tag)

標籤(tag) 跟分支是很像的東西

除了標籤不會隨著 Commit 而前進



留下來 或者我跟你走？

【冷知識】

你知道 Git 會做資源回收嗎？

Garbage Collection

資源回收的發動時機？

1. eddie@gaojiande-MacBook: /private/tmp/modern-web (zsh)

高 見龍 0_0 master ✓

modern-web



新增定制化的 Git 指令？

\$ git whatever
=> git-whatever

Apple logo 高 見龍 0_0

bin

小結

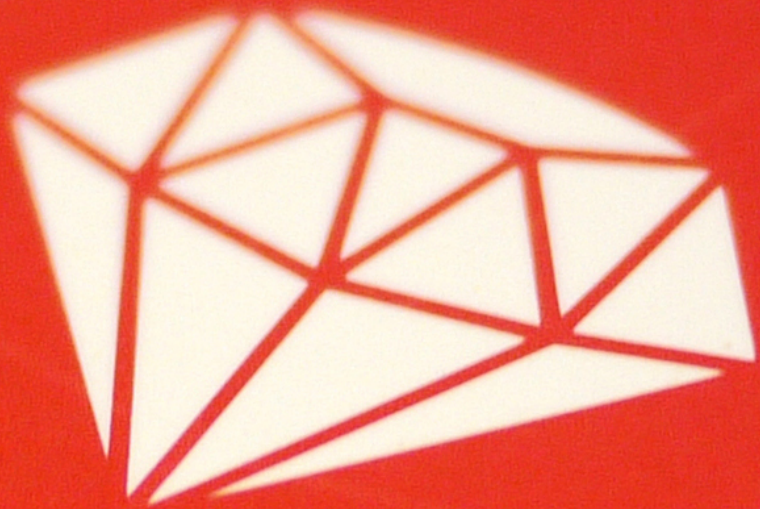
在各位離開這間會議廳之前...

0. 知道 Git 裡的 4 種物件
1. 知道什麼是 SHA1
2. 知道什麼是 Commit
3. 知道 Commit 的時候發生什麼事
4. 知道什麼是分支
5. 知道合併分支的時候發生什麼事
6. 知道東西進 Git, 要砍還不太容易

瞭解運作的原理
遇到問題才能迎刃而解

共勉之！

5x{



五倍紅寶石股份有限公司

}.tw

高見龍

📍 Blog

<http://kaochenlong.com>

📍 Facebook

<http://www.facebook.com/eddiekao>

📍 Twitter

<https://twitter.com/eddiekao>

📍 Email

eddie@5xruby.tw

📍 Mobile

+886-928-617-687

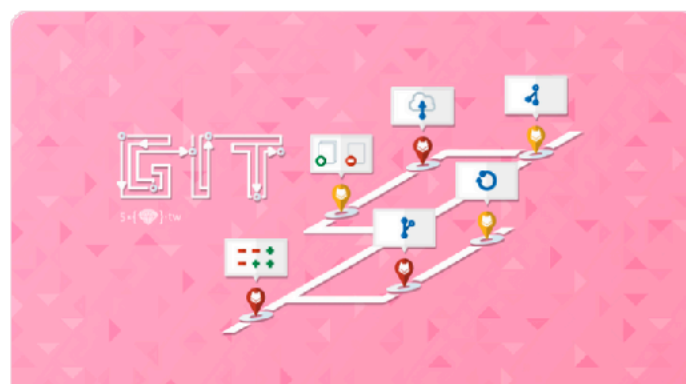
工商服務



歡迎來到 五倍紅寶石學院

Enroll Now

精選課程



<https://campus.5xruby.tw/>



人生不能重來，但 GIT
可以

🛒 現在就加入！

Git 是目前最流行的版本控制工具。本課程除介紹 Git 的基本原理及操作外，並將模擬各種在開發過程常遇到的狀況，引導大家快速上手。

線上課程預購中!!
課程將於八月底上線

折扣碼

"ModernWeb2017"

可折抵 300 元

有效期限

即日起至 8/19 止

<https://campus.5xruby.tw/>

0 - 入門篇

📖 0.1 - 什麼是 Git？為什麼要學習它？

📖 0.2 - 與其它版本控制系統的差異

1 - 環境安裝

📖 1.1 - 安裝在 Windows 作業系統

📖 1.2 - 在 Mac OSX 作業系統

📖 1.3 - 安裝在 Linux 作業系統

📖 1.4 - 圖形化介面工具

2 - 終端機/命令提示字元

📖 2.1 - 常用指令介紹

📖 2.2 - 超精簡 Vim 操作介紹

3 - 設定 Git

📖 3.1 - 使用者設定

📖 3.2 - 其它便利設定

4 - 開始使用 Git

📖 4.1 - 新增、初始 Repository

📖 4.2 - 把檔案交給 Git 控管

📖 4.3 - 工作區、暫存區與儲存庫

📖 4.4 - 檢視狀態與紀錄

📖 4.5 - 【狀況題】如何在 Git 裡刪除檔案或變更檔名？

📖 4.6 - 【狀況題】修改提交紀錄

📖 4.7 - 【狀況題】追加檔案到最近一次的提交

📖 4.8 - 【狀況題】提交目錄？

📖 4.9 - 【狀況題】有些檔案我不想放在 Git 裡面...

📖 4.10 - 【狀況題】檢視特定檔案的提交紀錄

📖 4.11 - 【狀況題】等等，這行程式誰寫的？

📖 4.12 - 【狀況題】啊！不小心把檔案或目錄刪掉了...

📖 4.13 - 【狀況題】剛才的提交後悔了，想要撤掉重做...

📖 4.14 - 【冷知識】HEAD 是什麼東西？

📖 4.15 - 【冷知識】那個長得很像亂碼 SHA1 是怎麼算出來的？

📖 4.16 - 【狀況題】可以只 Commit 一個檔案的部份的進度嗎？

📖 4.17 - 【冷知識】在 .git 目錄裡有什麼東西？Part 1

📖 4.18 - 【冷知識】在 .git 目錄裡有什麼東西？Part 2

5 - 使用分支

📖 5.1 - 為什麼要使用分支？

📖 5.2 - 開始使用分支

📖 5.3 - 對分支的誤解

📖 5.4 - 合併分支

📖 5.5 - 【狀況題】為什麼我的分支都沒有「小耳朵」？

📖 5.6 - 【常見問題】合併過的分支要留著嗎？

📖 5.7 - 另一種合併方式

📖 5.8 - 合併發生衝突了，怎麼辦？

📖 5.9 - 【冷知識】為什麼大家都說在 Git 開分支「很便宜」

📖 5.10 - 【冷知識】Git 怎麼知道現在是在哪一個分支？

6 - 修改歷史紀錄

📖 6.1 - 【狀況題】修改歷史訊息

📖 6.2 - 【狀況題】把多個提交合併成一個提交

📖 6.3 - 【狀況題】把一個提交拆解成多個提交

📖 6.4 - 【狀況題】想要刪除某幾個特定的提交

📖 6.5 - reset、revert 跟 rebase 指令有什麼差別？

7 - 標籤

 7.1 - 使用標籤

 7.2 - 【冷知識】 標籤跟分支有什麼不一樣？

8 - 其它常見狀況題

 8.1 - 【狀況題】 手邊的工作做到一半，臨時要切換到別的任務

 8.2 - 【狀況題】 不小心把帳號密碼放在 Git 裡了，而且已經推出去了。

 8.3 - 【狀況題】 如果你只想要某個分支的某幾個提交？

 8.4 - 【狀況題】 不小心使用 --hard 模式 reset 了某個提交

 8.5 - 【狀況題】 不小心把還沒合併的分支砍掉了，救得回來嗎？

 8.6 - 【冷知識】 怎麼樣把檔案真正的從 Git 裡移掉？

9 - 遠端共同協作 - 使用 GitHub

 9.1 - GitHub 是什麼？

 9.2 - push 上傳到 GitHub

 9.3 - pull 下載更新

 9.4 - 從伺服器上取得 Repository

 9.5 - 【常見問題】 clone 跟 pull 有什麼不一樣？

 9.6 - 與其它開發者的互動 - 使用 Pull Request(PR)

 9.7 - 【狀況題】 怎麼跟上當初 fork 專案的進度？

 9.8 - 【狀況題】 怎麼刪除遠端的分支？

 9.9 - 【狀況題】 聽說 git push -f 這個指令很可怕，什麼情況可以用它呢？

 9.10 - 使用 GitHub 免費製作個人網站

 9.11 - 【冷知識】 一定要有 GitHub 之前的 Git Server 才能得到別人更新的檔案

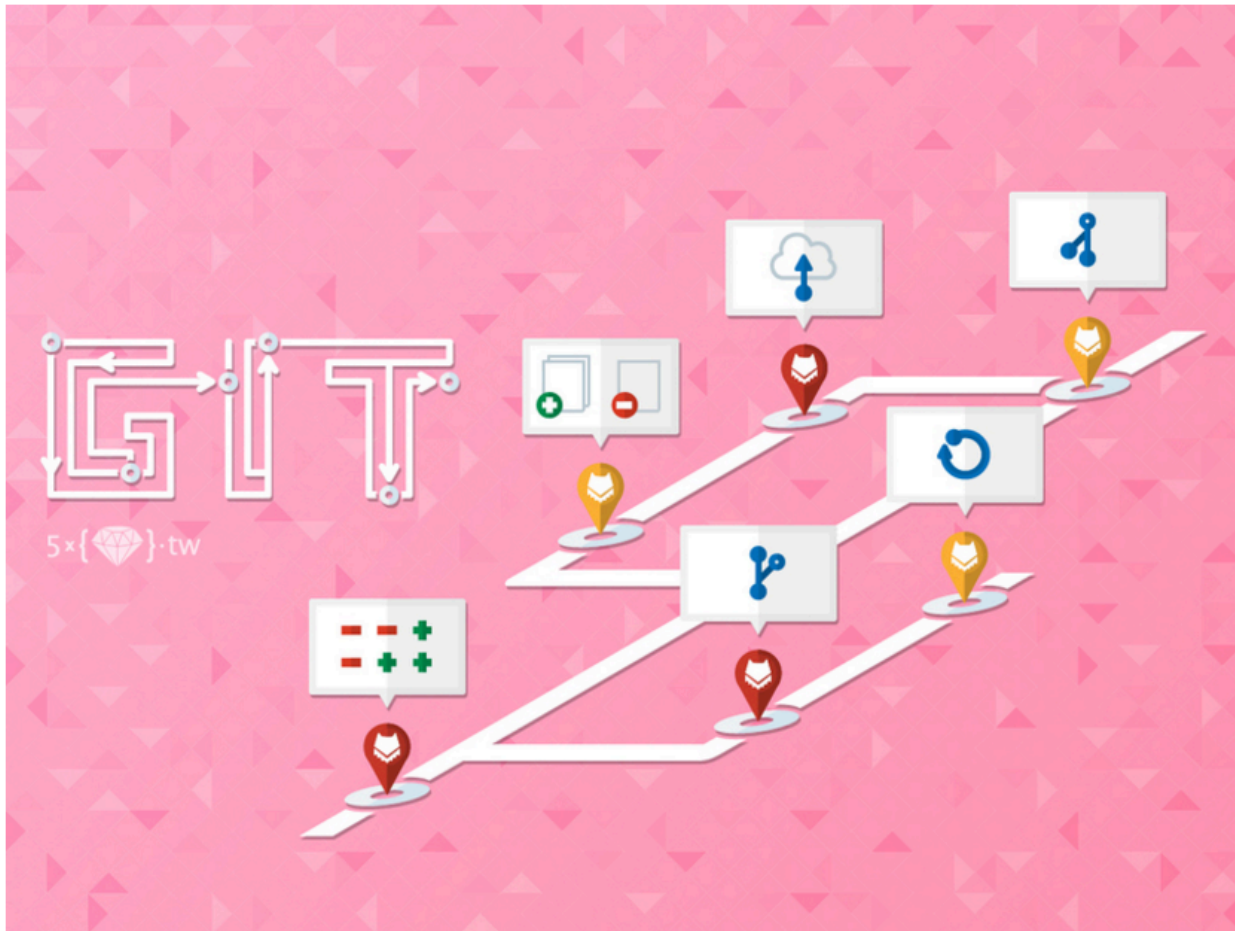
10 - 使用 Git flow

 10.1 - 為什麼需要這種東西？

 10.2 - 使用 Git Flow

實體課程

優惠即日起至 8/19 止

[首頁](#) / [講座資訊](#) / 人生不能重來，但 GIT 可以

Git

人生不能重來，但 GIT 可以

2017-08-23 ~ 2017-08-25

高見龍 (Eddie Kao) | 時數:6.0

Git 是目前最流行的版本控制工具。本課程除介紹 Git 的基本原理及操作外，並將模擬各種在開發過程常遇到的狀況，引導大家快速上手。

[立即報名](#)

謝謝！