

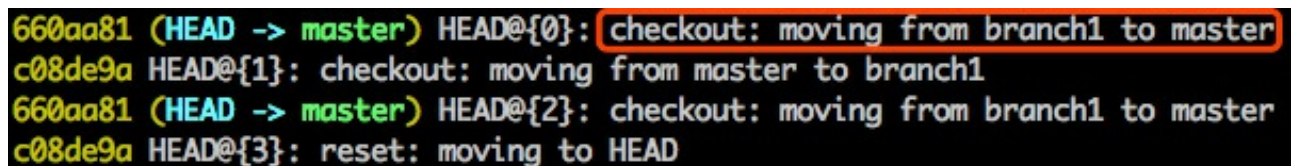
高级 10: branch 删过了才想起来有用?

branch 用完就删是好习惯，但有的时候，不小心手残删了一个还有用的 branch，或者把一个 branch 删掉了才想起来它还有用，怎么办？

reflog : 引用的 log

reflog 是 "reference log" 的缩写，使用它可以查看 Git 仓库中的引用的移动记录。如果不指定引用，它会显示 HEAD 的移动记录。假如你误删了 branch1 这个 branch，那么你可以查看一下 HEAD 的移动历史：

```
git reflog
```

A terminal window showing the output of the 'git reflog' command. The output lists four HEAD movements. The first line, '660aa81 (HEAD -> master) HEAD@{0}: checkout: moving from branch1 to master', is highlighted with a red box. The second line is 'c08de9a HEAD@{1}: checkout: moving from master to branch1'. The third line is '660aa81 (HEAD -> master) HEAD@{2}: checkout: moving from branch1 to master'. The fourth line is 'c08de9a HEAD@{3}: reset: moving to HEAD'.

```
660aa81 (HEAD -> master) HEAD@{0}: checkout: moving from branch1 to master
c08de9a HEAD@{1}: checkout: moving from master to branch1
660aa81 (HEAD -> master) HEAD@{2}: checkout: moving from branch1 to master
c08de9a HEAD@{3}: reset: moving to HEAD
```

从图中可以看出，HEAD 的最后一次移动行为是「从 branch1 移动到 master」。而在这之后，branch1 就被删除了。所以它之前的那个 commit 就是 branch1 被删除之前的位置了，也就是第二行的 c08de9a。

所以现在就可以切换回 c08de9a，然后重新创建 branch1：

```
git checkout c08de9a
git checkout -b branch1
```

这样，你刚删除的 branch1 就找回来了。

注意：不再被引用直接或间接指向的 commits 会在一定时间后被 Git 回收，所以使用 reflog 来找回删除的 branch 的操作一定要及时，不然有可能会由于 commit 被回收而再也找不回来。

查看其他引用的 reflog

reflog 默认查看 HEAD 的移动历史，除此之外，也可以手动加上名称来查看其他引用的移动历史，例如某个 branch：

```
git reflog master
```

```
660aa81 (master, branch2) master@{0}: commit: Add feature4
1c2841c (origin/master, origin/HEAD) master@{1}: reset: moving to HEAD^
fcf98c0 master@{2}: reset: moving to fcf98c06a423aa1a06460beb57691c14a922cc38
b8356c5 master@{3}: commit (merge): Merge branch 'feature1'
fcf98c0 master@{4}: commit: SSD bought
```