Git 原理详解及实用指南 - 扔物线 - 掘金小册

高级 5：想丢弃的也不是最新的提交？

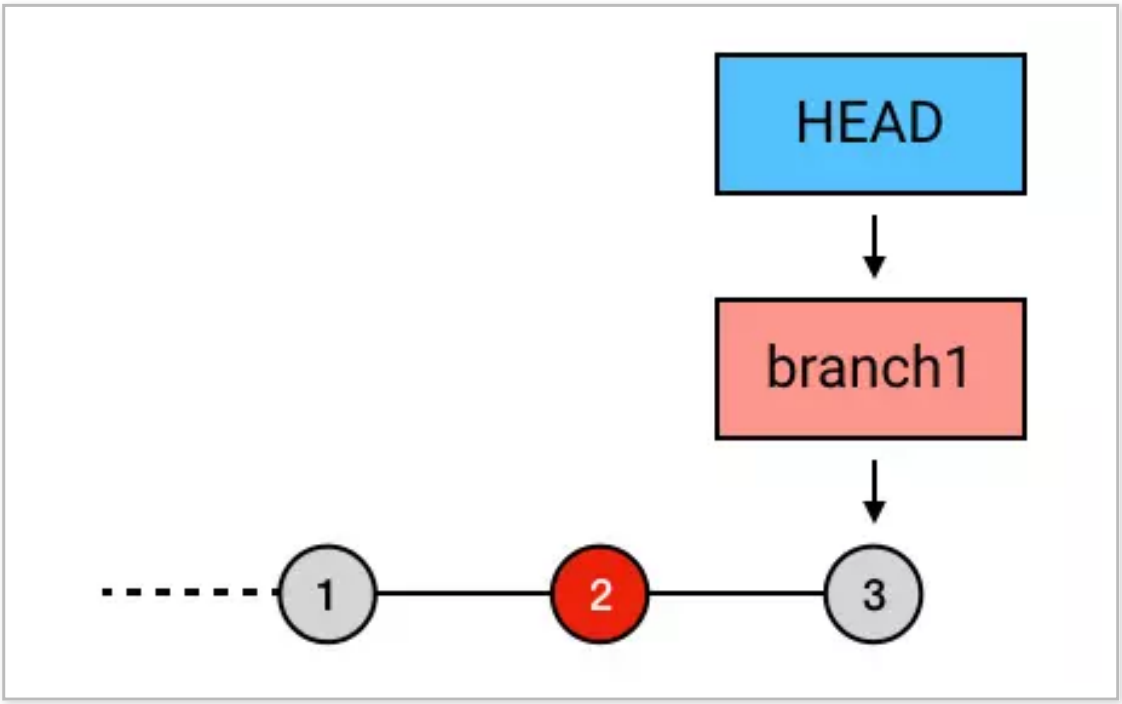
假如有一个 `commit` ，你在刚把它写完的时候并没有觉得它不好，可是在之后又写了几个提交以后，你突然灵光一现：「哎呀，那个 `commit` 不该写，我要撤销！」

不是最新的提交，就不能用 `reset --hard` 来撤销了。这种情况的撤销，就要用之前介绍过的一个指令：交互式 `rebase` —— `rebase -i` 。

用交互式 rebase 撤销提交

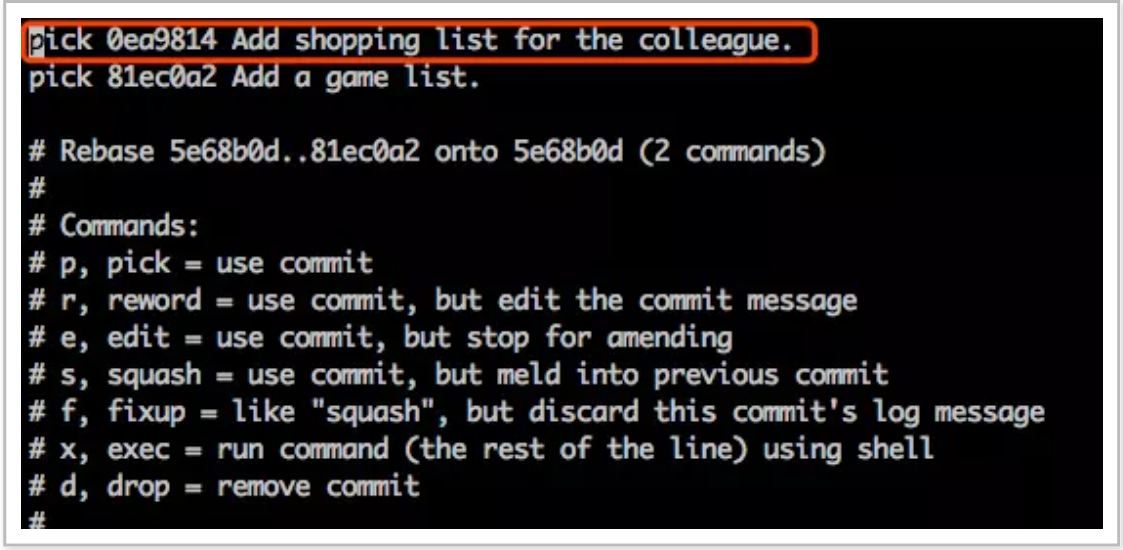
之前介绍过，交互式 `rebase` 可以用来修改某些旧的 `commit` s。其实除了修改提交，它还可以用于撤销提交。比如下面这种情况：

```
commit 81ec0a26286b431de68460108f182cfbad52a2e6 (HEAD -> branch1)
Author: Kai Zhu <rengwuxian@gmail.com>
Date:   Sun Nov 19 19:48:33 2017 +0800

    Add a game list.

commit 0ea98144cc321649dcfb21a71007b5b652577655
Author: Kai Zhu <rengwuxian@gmail.com>
Date:   Sun Nov 19 16:24:52 2017 +0800

    Add shopping list for the colleague.

commit 5e68b0d8d47ebd2a52977586f728f74335ae62c5
Author: Kai Zhu <rengwuxian@gmail.com>
Date:   Sun Nov 19 14:58:33 2017 +0800

    Remove a shameful part

commit 79c353da8263fd9e93d7eb68dfbd1c8190d3450c
```



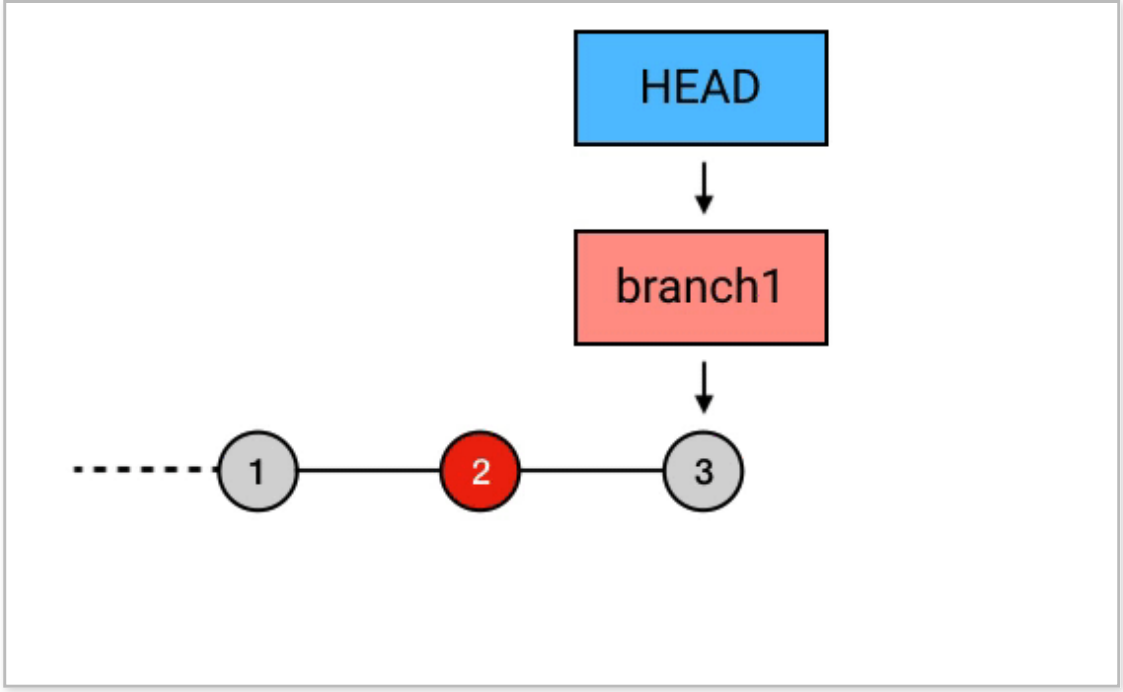
你想撤销倒数第二条 `commit` ，那么可以使用 `rebase -i` ：

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



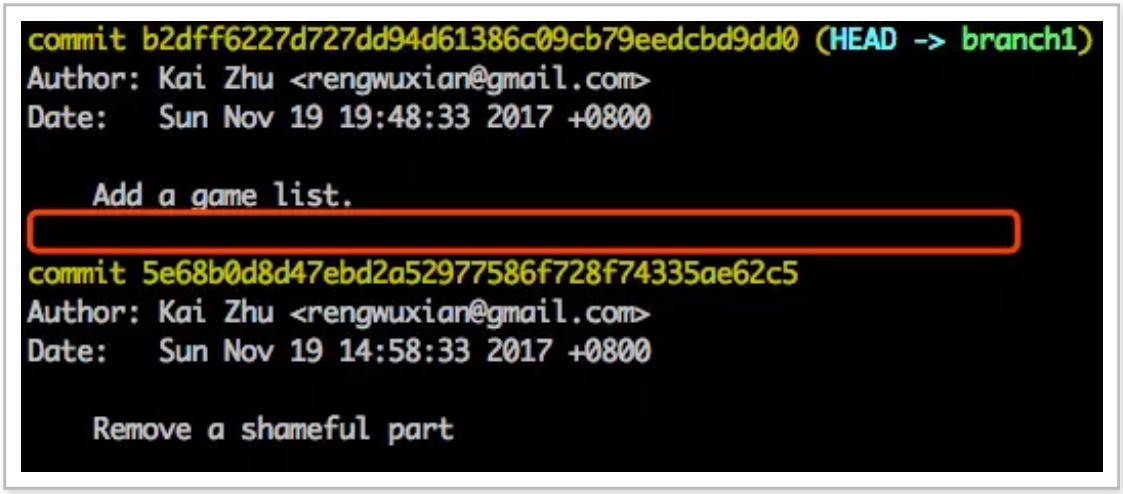
然后就会跳到 `commit` 序列的编辑界面，这个在之前的第 10 节里已经讲过了。和第 10 节一样，你需要修改这个序列来进行操作。不过不同的是，之前讲的修正 `commit` 的方法是把要修改的 `commit` 左边的 `pick` 改成 `edit` ，而如果你要撤销某个 `commit` ，做法就更加简单粗暴一点：直接删掉这一行就好。

`pick` 的直接意思是「选取」，在这个界面的意思就是应用这个 `commit` 。而如果把这一行删掉，那就相当于在 `rebase` 的过程中跳过了这个 `commit` ，从而也就把这个 `commit` 撤销掉了。



现在再看 `log` ，就会发现之前的倒数第二条 `commit` 已经不在了。

```
git log
```

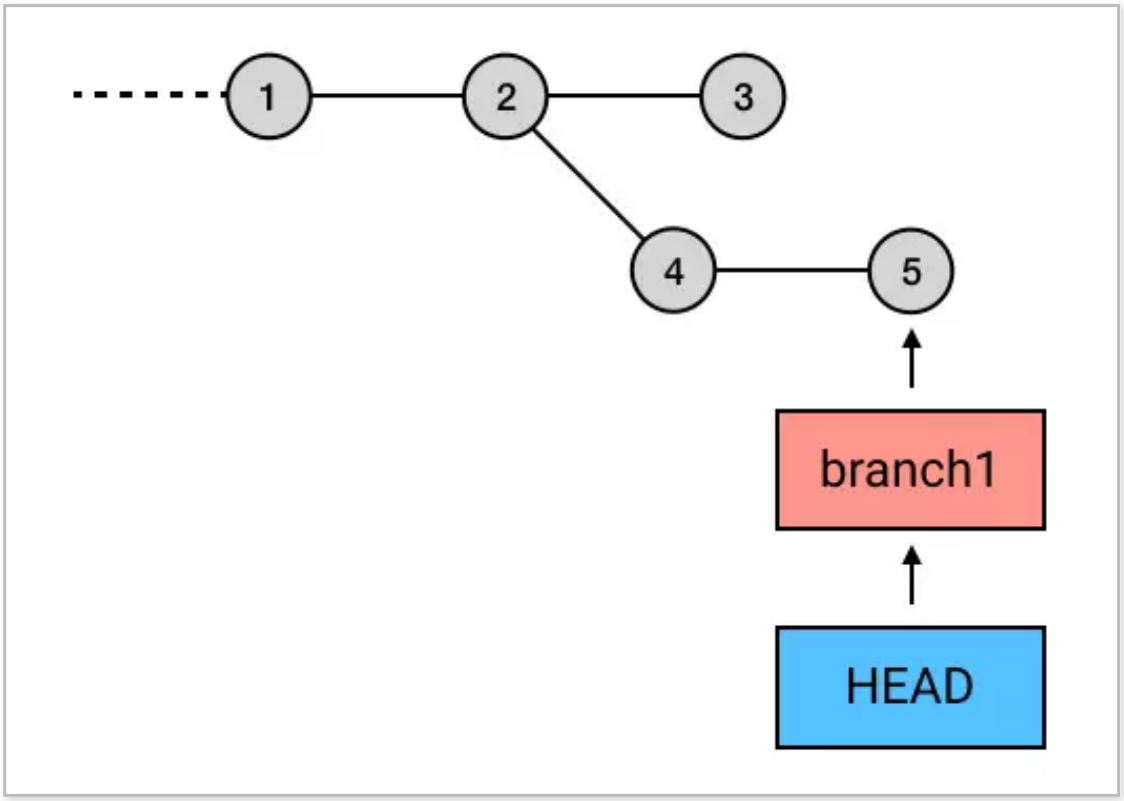


用 rebase --onto 撤销提交

除了用交互式 `rebase` ，你还可以用 `rebase --onto` 来更简便地撤销提交。

`rebase` 加上 `--onto` 选项之后，可以指定 `rebase` 的「起点」。一般的 `rebase` ，告诉 Git 的是「我要把当前 `commit` 以及它之前的 `commit` s 重新提交到目标 `commit` 上去，这其中， `rebase` 的「起点」是自动判定的：选取当前 `commit` 和目标 `commit` 在历史上的交叉点作为起点。

例如下面这种情况：



如果在这里执行：

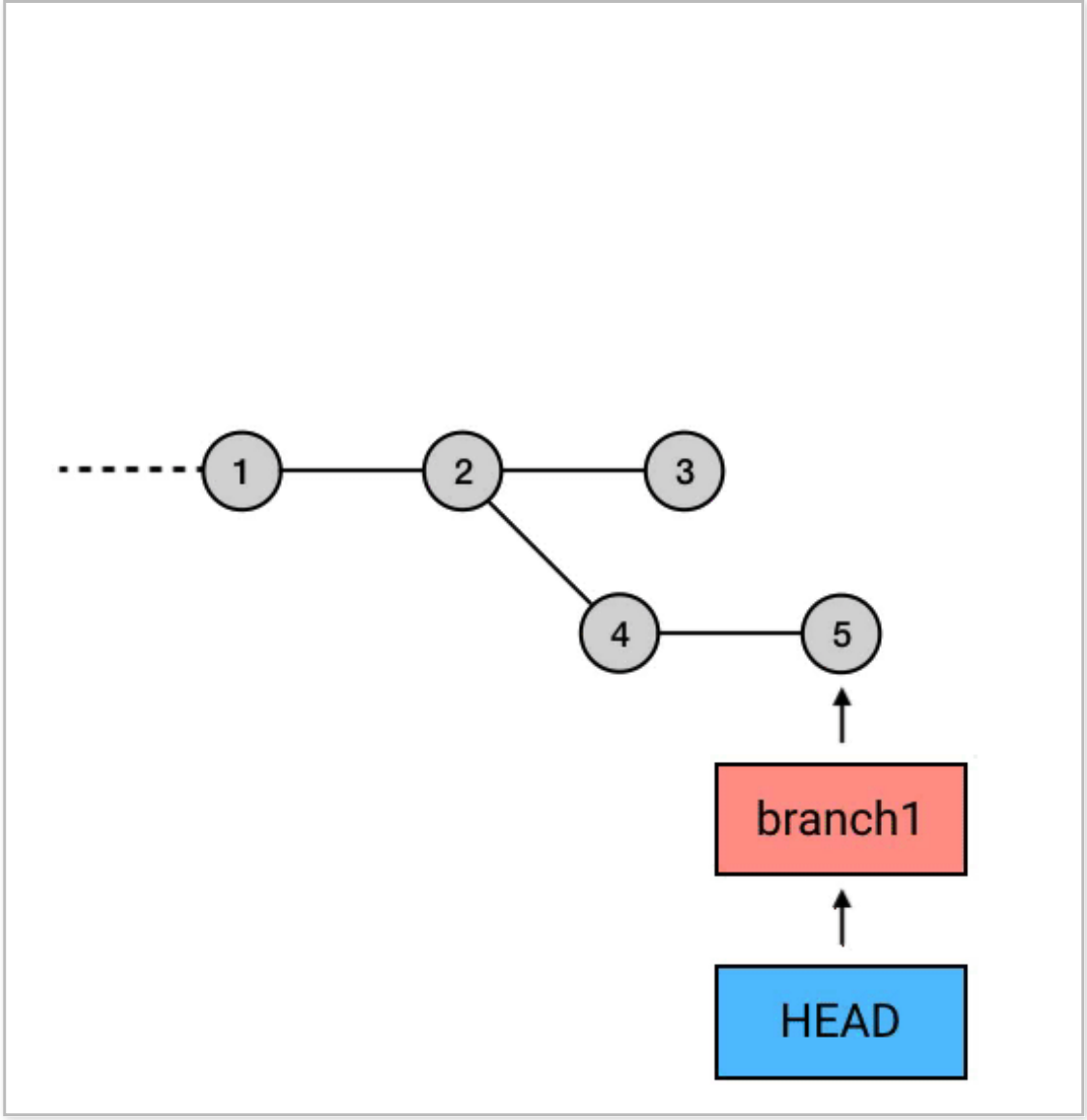
```
git rebase 第3个commit
```

那么 Git 会自动选取 3 和 5 的历史交叉点 2 作为 rebase 的起点，依次将 4 和 5 重新提交到 3 的路径上去。

而 --onto 参数，就可以额外给 rebase 指定它的起点。例如同样以上图为例，如果我只想把 5 提交到 3 上，不想附带上 4 ，那么我可以执行：

```
git rebase --onto 第3个commit 第4个commit branch1
```

--onto 参数后面有三个附加参数：目标 commit 、起点 commit （注意：rebase 的时候会把起点排除在外）、终点 commit 。所以上面这行指令就会从 4 往下数，拿到 branch1 所指向的 5 ，然后把 5 重新提交到 3 上去。

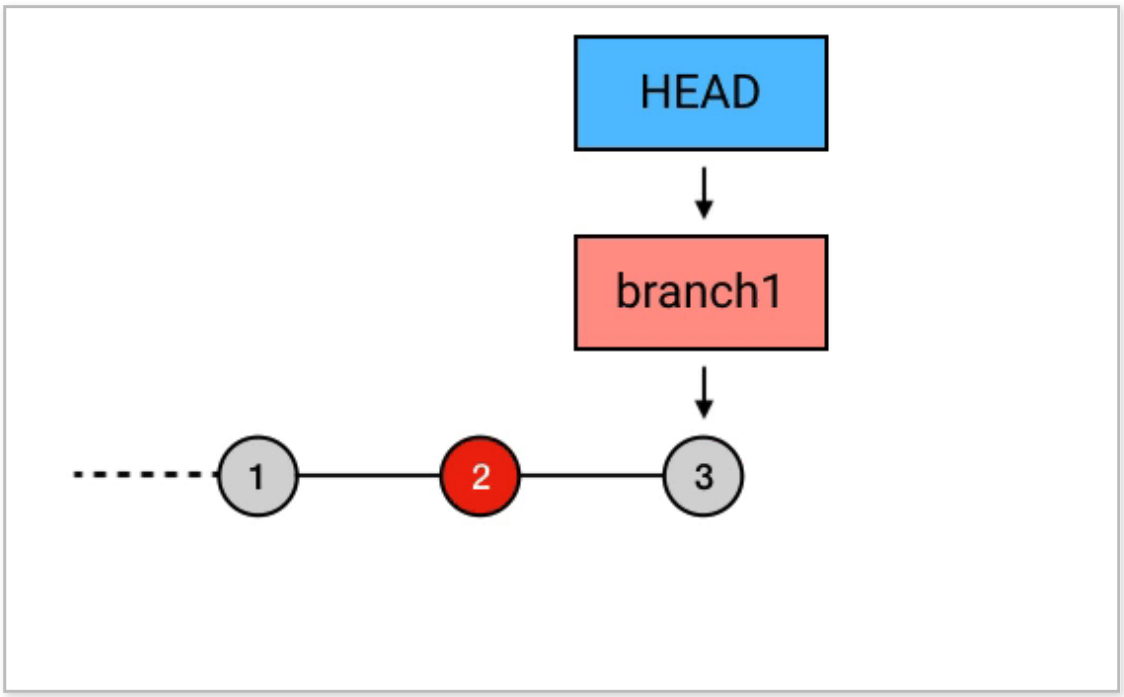


同样的，你也可以用 `rebase --onto` 来撤销提交：

```
git rebase --onto HEAD^^ HEAD^ branch1
```

上面这行代码的意思是：以倒数第二个 `commit` 为起点（起点不包含在 `rebase` 序列里哟），`branch1` 为终点，`rebase` 到倒数第三个 `commit` 上。

也就是这样：



小结

这节的内容是「撤销过往的提交」。方法有两种：

- 用 `git rebase -i` 在编辑界面中删除想撤销的 `commit` s
- 用 `git rebase --onto` 在 rebase 命令中直接剔除想撤销的 `commit` s

方法有两种，理念是一样的：在 `rebase` 的过程中去掉想撤销的 `commit` ，让他它消失在历史中。

留言

评论将在后台进行审核，审核通过后对所有人可见

- **sailingfaraway** java web

  用 reset 撤销当前 commit，commit 实际还在，用 rebase 是不是 commit 就再也找不回来了?

  ▲0 收起评论 1 月前

  - sailingfaraway
    java web
    真删除?
    1 月前

  评论审核通过后显示 评论

- **sailingfaraway** java web

  又是一个灵活的命令，不过同问怎么 push 到远端 XD

  ▲0 评论 1 月前

- **cjr0707**

  你好，我有一个疑惑，比如现在是一个分支情况是这样的 1-2-3-4。我 git rebase -i 1，然后删除 pick 2 这个 commit，过了一会我突然又想拿回刚删除的 commit，git reflog 查看 sha 1 值，无法看出该如何 reset --hard 到哪个 commit。请老师指点一下吧。

  ▲0　　评论　1 月前

- **普通分享**

  "用交互式 rebase 撤销提交" 中，调用完 “git rebase -i HEAD^^”，并把 pick 那行删完后，接下来怎么做呢?

  ▲0　　收起评论　4 月前

  - 普通分享

    我发现两次修改同一个文件，不能修改相邻行，当两次 commit，再用 “git rebase -i HEAD^^”，最后删

除前一个 commit 时，会有冲突。

4 月前

- cjr0707

  回复 普通分享：可能是你修改的同一个文件，和你之后的修改有冲突了吧

  1 月前

评论审核通过后显示

评论

- **这是一个昵称 Z**

  Android 开发工程师 @ 北京千橡网景...

  git rebase --onto 第 3 个 commit 第 4 个 commit branch1　这里的 branch1 这一个参数可以看作是指的是 branch1 上的最后一个 commit

  ▲0　评论　6 月前

-

ivotai

rebase onto 还是迷迷糊糊的

▲0　　评论　6 月前

- **月与海**　web 前端 @ 禾木林

  rebase 比 merg 灵活

  ▲0　　评论　6 月前

- **丁辉元 (Dean)**

  @仍物线 第一张图片加载不出来了

  ▲0　　评论　7 月前

- **George 吴逸云**

  这个看的好懵逼

  ▲0　　评论　8 月前

-

RottenWang

请问一下操作完之后 确实是没有在记录里了. 但是如何 push 到远端呢

▲0　　评论　8 月前

- **一直向往平淡**

我觉得还是用 -i 删除直观一点，onto 容易混

▲0　　评论　8 月前

- **L。同学**　前端工程师 @ 北京莱熙科技

假如现在是这样的 1-->2-->3-->4, 我删除第二个提交的话, 是不是 3,4 的内容也没了?

▲0　　收起评论　8 月前

  - 扔物线　Android 布道师（假装）@ HenCoder

    有的，你试试

    8 月前

- zsp109096
  学生 @ 南昌大学
  如果你用 onto 就没有了
  3 月前

评论审核通过后显示　评论

- **思樺**
  而 rebase 參數，就可以額外給 rebase 指定它的起点。 □<- 這裡的 rebase 參數應該是 --onto 參數才對
  ▲0　收起评论　9 月前

  - 扔物线
    Android 布道师（假装） @ HenCoder
    是的，感谢指正哈
    8 月前

评论审核通过后显示

评论

- **leiqingqiang**

  rebase --onto 真是万脸懵逼

  ▲0　　评论　9 月前

- **小葱**　前端

  留名

  ▲0　　评论　9 月前

- **Ashley_T**　Android 应用开发工程师

  在编辑界面删除一行，和使用 drop 是同样的效果吗?

  ▲0　　收起评论　9 月前

  - PPTingKJ

    我试了一下，效果上在编辑页面直接删除一行或者将 pick 改

成 drop 都可以将这个 commit 删除，但不知道实质上有没有区别

9 月前

- PPTingKJ

  Stack Overflow 上有这个问题的讨论 https://stackoverflow.com/questions/35846154/git-rebase-interactive-drop-vs-deleting-the-commit-line

  9 月前

评论审核通过后显示

评论

- **RTFSC**

  万脸懵逼

▲0 评论 9 月前

- **码青** 安卓软件开发 @ 无

  工作中还从来没用过 git rebase --onto ？

  ▲0 评论 9 月前

- **祺在掘金**

  这个有点蒙，留个底，下次学习

  ▲0 收起评论 9 月前

  - funnyzhao

    副业 CEO @ 天天向上

    大兄弟、你把我们逗乐了~~

    9 月前

  评论审核通过后显示 评论

全文完