

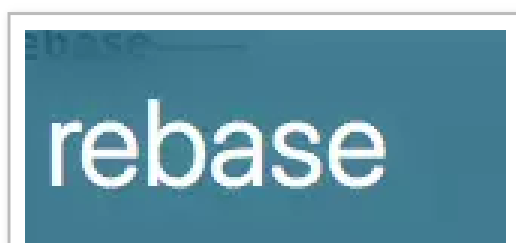
Git 原理详解及实用指南 - 扔物线 - 掘金小册

高级 1：不喜欢 merge 的分叉？ 用 rebase 吧

有些人不喜欢 `merge`，因为在 `merge` 之后，`commit` 历史就会出现分叉，这种分叉再汇合的结构会让有些人觉得混乱而难以管理。如果你不希望 `commit` 历史出现分叉，可以用 `rebase` 来代替 `merge`。

rebase——在新位置重新提交

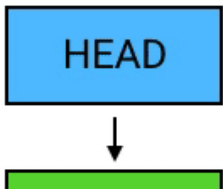
`rebase`，又是一个中国人看不懂的词。这个词的意思，你如果查一下的话：



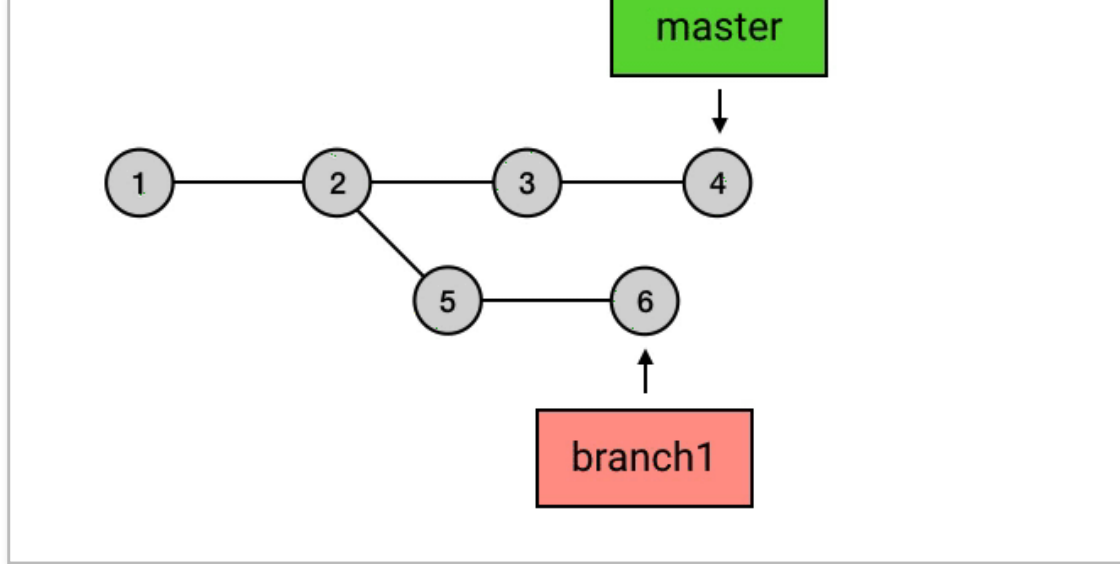
哈？玩个 Git 就弯了？

其实这个翻译还是比较准确的。`rebase` 的意思是，给你的 `commit` 序列重新设置基础点（也就是父 `commit`）。展开来说就是，把你指定的 `commit` 以及它所在的 `commit` 串，以指定的目标 `commit` 为基础，依次重新提交一次。例如下面这个 `merge`：

```
git merge branch1
```

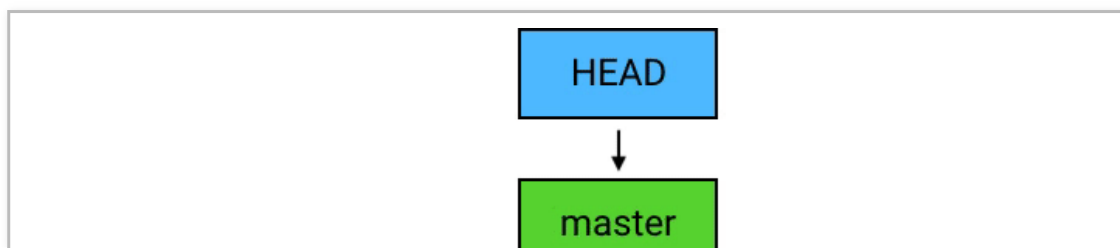


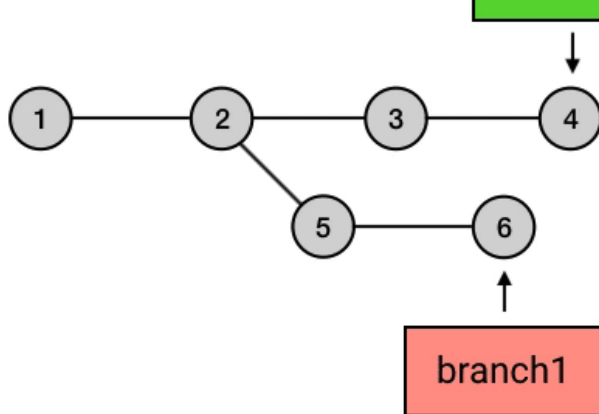
HEAD



如果把 `merge` 换成 `rebase` , 可以这样操作:

```
git checkout branch1
git rebase master
```



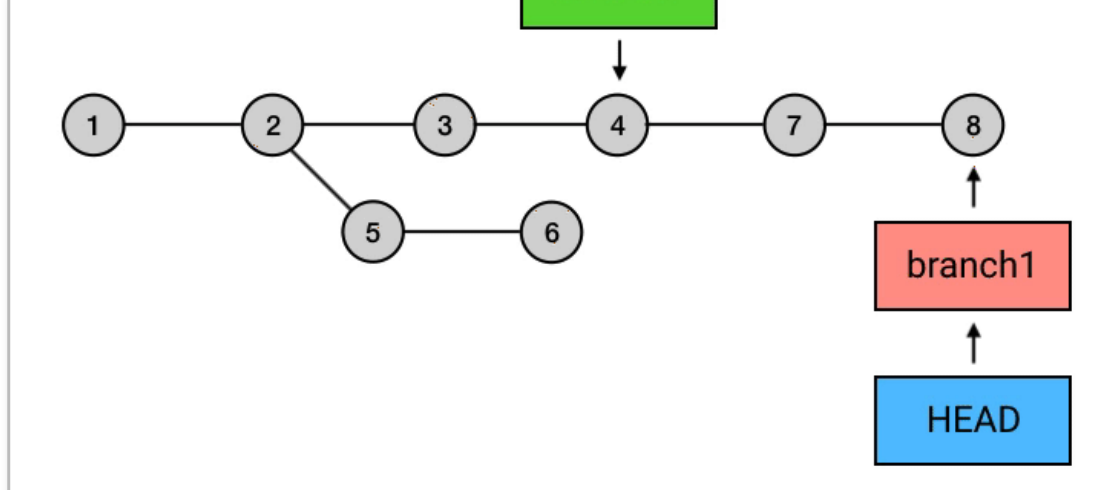


可以看出，通过 `rebase`，`5` 和 `6` 两条 `commit` s 把基础点从 `2` 换成了 `4`。通过这样的方式，就让本来分叉了的提交历史重新回到了一条线。这种「重新设置基础点」的操作，就是 `rebase` 的含义。

另外，在 `rebase` 之后，记得切回 `master` 再 `merge` 一下，把 `master` 移到最新的 `commit`：

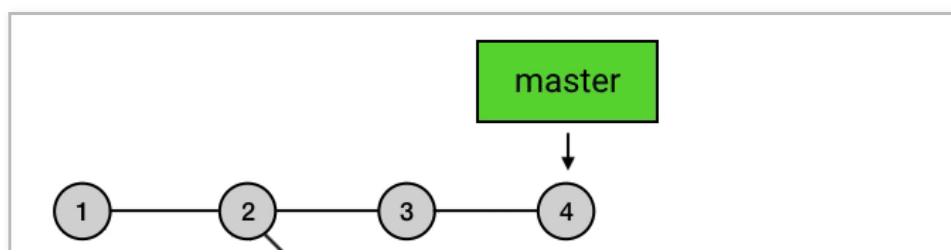
```
git checkout master
git merge branch1
```

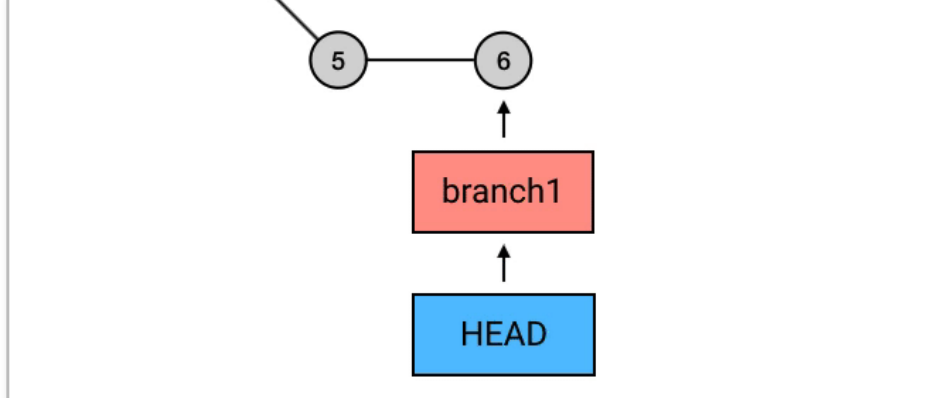
master



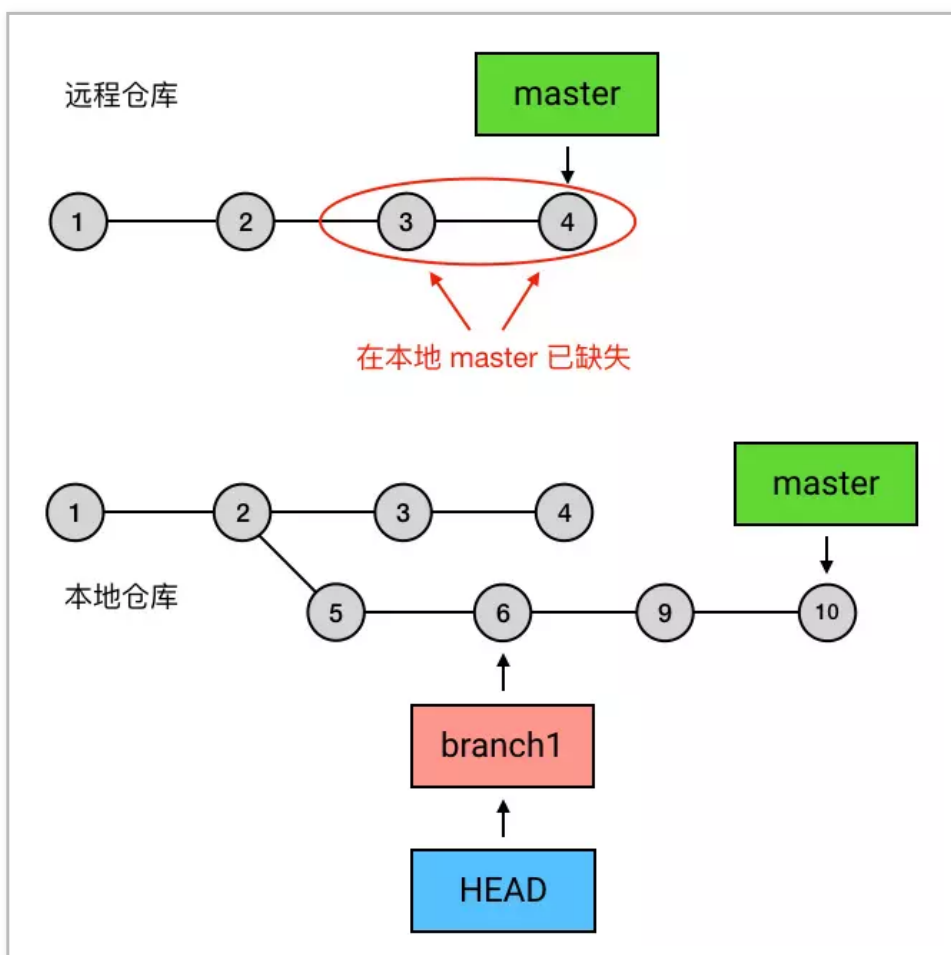
为什么要从 `branch1` 来 `rebase` , 然后再切回 `master` 再 `merge` 一下这么麻烦, 而不是直接在 `master` 上执行 `rebase` ?

从图中可以看出, `rebase` 后的 `commit` 虽然内容和 `rebase` 之前相同, 但它们已经是不同的 `commits` 了。如果直接从 `master` 执行 `rebase` 的话, 就会是这样:





这就导致 `master` 上之前的两个最新 `commit` 被剔除了。如果这两个 `commit` 之前已经在中央仓库存在，这就会导致没法 `push` 了：



所以，为了避免和远端仓库发生冲突，一般不要从 `master` 向其他 `branch` 执行 `rebase` 操作。而如果是 `master` 以外的

`branch` 之间的 `rebase` (比如 `branch1` 和 `branch2` 之间), 就不必这么多费一步, 直接 `rebase` 就好。

小结

本节介绍的是 `rebase` 指令, 它可以改变 `commit` 序列的基础点。它的使用方式很简单:

```
git rebase 目标基础点
```

需要说明的是, `rebase` 是站在需要被 `rebase` 的 `commit` 上进行操作, 这点和 `merge` 是不同的。

留言

评论将在后台进行审核, 审核通过后对所有人可见

●

杨伟康 (vigroid) 安卓开发 @ tictalk

扔哥也用 Alfred 啊

▲0 评论 1 天前

●

18500040836 前端

我的理解：

git checkout branch1

git rebase master

执行这两个命令之后，master 分支并没有变化还是包含了 1、2、3、4 四个 commit。

branch1 分支变成了 1、2、3、4、7（内容是原来的 5）、8（内容是原来的 6）。

此时两条分支共存，而且 branch1 分支比 master 分支长，多了两个 commit.... [展开全部](#)

▲0 评论 29 天前

●

Michael 翔 CID @ HW

变基：改变基点，也就是你这个分支开始分叉时的点。那基点变为什么呢？变为 git rebase

<NewBase> 命令中对应的 NewBase 最新的 commit 点。

所以，执行变基操作，需要在你改变基点的分支进行，而不是 master。

合并：要和哪个分支合并呢？master 想要和

branch 合并，那么在 master 分支，执行 git merge branch 即可。

▲0 评论 1 月前

●

Reci

不知道大家有没有这个疑问

git rebase master 执行之前 特意 checkout 到 branch1

但是 反向 git rebase branch1 的动态图，head 一直在 branch1 上面，不是应该现在 master 上，执行 rebase 之后带下来么

谁能解答下么

▲1 收起评论 1 月前

- 1874 重名了
应该是图没画对，执行 git rebase branch1 时，应该要先 checkout master，这时 head 应该指向 master 先。@凯哥
1 月前

- Michael 翔
CID @ HW

确实是，作者总结也说了，执行变基操作，要在被变基的分支上，git rebase branch1 时，是在 master 分支上，此时，HEAD 应该是指向的是 master，所以，图画错了

1 月前

评论审核通过后显示

评论

订阅公众号

“master 以外的 branch 可以直接 rebase”，难道其他两个分支就不能有冲突的情况了吗？？比如 branch1 rebase 到 branch2 中，branch1 也是会有最新的提交被剔除啊，这样不是也跟 master 的情况一样吗？（我的 branch2 是 branch1 上生成的），难道大佬说的 branch1 和 branch2 是完全没有关系的分支，才能直接 rebase 的意思是吗

▲0

评论 1 月前

●
高洋 iOS 开发工程师

看到这里有不明白的可以`git rebase --help`里面有个图，进去看看

▲1 评论 1 月前

●
毛毛大婶

看完再看评论区 理解起来就更容易了

▲0 评论 2 月前

●
碧海银剑

有点绕，需要多看几遍，点个赞！

▲0 评论 3 月前

●
air

牛逼

▲0 评论 3 月前

●

吴数

看了很多版本的 rebase 解释，看到这里完全明白了，3Q

▲0 评论 4 月前

●

卖牙膏的关口钉 Android 拱城狮 @ 数位科技

很棒

▲0 收起评论 5 月前

●

小菜

哪都可以看到大哥

2 月前

评论审核通过后显示

评论

仔细看最上面的图，merge 操作后是 7 个节点，而 rebase 操作后只有 6 个节点。
7 8 和 5 6 是不同的，不只是简单的迁移。如果分支存有冲突的话，7 8 中包含了 merge 的细节。

▲0

收起评论 6 月前

•

千百度 YM-GG

如果存在冲突，会要求解完冲突才能 rebase 成功的，所以在解冲突的时候就包含了 merge 细节了

3 月前

评论审核通过后显示

评论

•

xiaxiayang

思考和查询一番后，我的理解是：

rebase 是将 commit 修改，而不是复制。

也就是说，branch1 rebase master 操作是在 master 分支上是从 branch1 分支移过去两个 commit 节点（5，6 变成了 7，8，此时 5，6 已

经不存在，存在只有 7, 8，作者没有出于某种考虑没有在图中显示 5, 6 的消失，引起了误会)。反过来的话，master rebase branch1，是将 master 的新的节点移到 branch1 。... [展开全部](#)

▲0

收起评论 6 月前



大白酱

照你这样理解也可以。不过感觉作者没有在图中消掉 5, 6 并不会引起误会，因为照之前章节作者的意思 branch 和 head 也只是种引用。merge 之后 branch1 和 head 指向引用到最新的 8 这个 commit 上了，这时，branch1 代表的 commit 串就是 1, 2, 3, 4, 7, 8 了。而 5, 6 的两个 commit 因为没有任何东西引用它们，所以它们就变成“野生” commit 了 (照作者之前说的 ^_^)，然后 git 自身就会利用某种机制在

后面将它们回收。

3 月前

- hercule

回复 大白酱： 同学 你的理解好像有点问题.

2 月前

- 大白酱

回复 hercule： 你好，想问下问题的错误在哪里呢？麻烦纠正

2 月前

- hercule

回复 大白酱： 其实 5.6 并没变成 "野生", 前面作者也说了 branch 是一串 commits 的引用, 只要没删除 branch,

5.6 就不会是 "野生", 只有删除 branch, 5,6 相当于没依托了,

这个时候, 才是野生, 系统会回收 (个人理解, 所谓的野生 commit, 是没有依托的, 也就是没有引用的 commit).. 当时 rebase 变基的时候,,branch 上的 5,6 是不存在的了, 已经消失掉了,, 当你在 checkout 到 branch 分支再次进行开发的时候,,, 基点不是 6, 而是 8, 所以 我个人觉得 你用 "野生" 来理解. 是有点问题,,,, 纯属个人理解

1 月前

- 亚拉斯特尔

5, 6 怎么就丢了?
我从 git log 中还是可以看到 5, 6 两个 commit 啊?

1 月前

评论审核通过后显示

评论



xiaxiayang

在 master 上 rebase 会丢 3, 4 这句未理解。
在 branch1 上 rebase 不会丢, 为什么在 master
上 rebase 就丢了呢? 那 rebase
多出来的 9 和 10 是什么?

▲0

收起评论 6 月前



air

1.master 分支
(local) 上的 3,4 变
成了 9,10(commits
变了, 代码没变)
但是 master(远程)却
还是有 3,4
2. 本地的 commits
和远程的 commits
不在一条线上, pull
或者 push 会有很多
冲突, 导致操作无法
执行

3 月前



air

branch1 上 5, 6 其
实也丢了. rebase 之

后 branch 分支也是
操作 push/pull 操作
也会出现冲突

3 月前

评论审核通过后显示

评论

●

xiaxiayang

在 master 上 rebase 会丢 3, 4 这句未理解。在
branch1 上 rebase 就不会丢

▲0

评论 6 月前

●

骑着乌龟上高速

如果在 branch1 上 rebase, branch1 不是也同
步了 master 上的提交了!!! 如果此时回到
master 执行 merge, 此时两个分支的内容不是一
样的了吗! ?

▲0

收起评论 8 月前

●

BlackC0 安卓

类似的 相当于分支上

的 commit 在主支重新 commit 了 之前分支就没用了 两个重新变成一条支线了

8 月前

评论审核通过后显示

评论



子华宝宝萌萌哒

PHP

终于知道啥叫变基了

▲0

评论 8 月前



kkk 就是我 代码复制粘贴工程师 @ copyright

本地的 3、4 不是变成 9、10 了吗》怎么会丢失呢

▲0

收起评论 8 月前



c~同学

引用都变了啊，每个 commit 都会有它独立的引用

8 月前

- [月与海](#)
[web 前端 @ 禾木林](#)

内容是拿过来了，但是以前版本的索引丢了

6 月前

评论审核通过后显示

评论

•

一直向往平淡

merge 连接 master; rebase 不连接 master, 新开 commit

▲0

评论 8 月前

•

enumlin

写的确实不错呀！

▲0

评论 9 月前

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验。