

HenCoder Plus 讲义

Git 深入之常用指令和风骚指令以及它们的本质

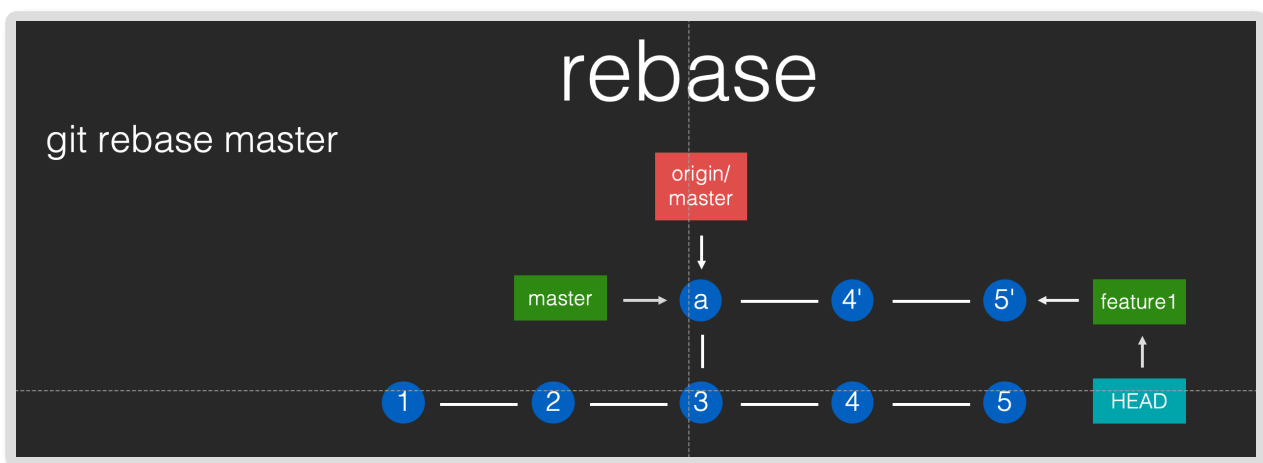
checkout

- 移动 HEAD，让它指向某个 commit 或某个 branch。
- `checkout --detach`：让 **HEAD** 脱离当前 branch，直接指向下面的 commit。 ,

rebase

把当前 commit（以及它之前的 commits）应用到指定的需要 rebase 的 commit 上。

Git 中的每一个 commit 都是不会改变的，所以 rebase 之后的每个 commit 都是新产生的，而不是对原先的 commit 进行「修改」



rebase 冲突

rebase 的冲突解决方法和 merge 冲突一样，只是把 `git merge --continue` 改成 `git rebase --continue` 就行了

reset

把当前 branch 指向指定的 commit。

- `git reset 指定commit`
移动到指定 commit，并保留 working tree 的内容
- `git reset --hard 指定commit`
移动到指定 commit，并重置 working tree

reset 和 checkout 的区别

它们都是移动 HEAD，但 checkout 移动的时候是自己移动，不带着 branch 一起；而 reset 会带着 branch 一起移动。

提交过的东西写错了

1. 最新的一条内容需要修改：

```
git commit --amend
```

「修改」只是概念行为，实质上会产生一个新的 commit

2. 旧的内容需要修改：

交互式 rebase

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

常用选项：

- `pick` 或 `p`：沿用
 - `reword` 或 `r`：修改 commit message
 - `edit` 或 `e`：修改 commit 内容
 - `drop` 或 `d`：删除
 - `squash` 或 `s`：和上一条融合，并且合并 commit message（可编辑）
 - `fixup` 或 `f`：和上一条融合，并且沿用上一条的 commit message
3. 已经 push 到 master 的内容需要删除：

不能用 rebase，因为 master 上的东西是不能强行修改的

可以用 `git revert 指定commit` 来撤销。它的原理是创建一个新的 commit, 内容是指定 commit 的「相反内容」

add -i 交互式 add

用法:

```
git add -i
```

常用选项:

- p (patch)

选取块时的常用选项:

- y: 选用当前块
- n: 不用当前块
- s: 把当前块做自动切分后再重新询问
- e: 手动选取修改内容

tag

另一种引用类型。

- 和 branch 区别之一: 不能改变
- 和 branch 区别之二: 不能被 HEAD 指向
- 用处: 设置持久标记, 例如版本号
- origin/master, origin/feature, origin/HEAD 和 tag 有相似之处: 也不能从本地改变位置, 也不能被 HEAD 指向

cherry-pick

用法:

```
git cherry-pick commit1 commit2
```

用途: 应用指定的某个或某些 commit 到 HEAD 处

- 比如我想抛弃某个 branch, 这个特性不要了, 但它的某两个 commit 我希望保留

- `git cherry-pick` 这两个 commit

reflog

用法：

```
git reflog <branch>
```

```
9c48604 (HEAD, tag: v3.0, tag: v1.0.1, tag: tag1, v1.0, feature1) HEAD@{0}: checkout: moving from v1.0 to v1.0.1
9c48604 (HEAD, tag: v3.0, tag: v1.0.1, tag: tag1, v1.0, feature1) HEAD@{1}: checkout: moving from feature1 to v1.0
9c48604 (HEAD, tag: v3.0, tag: v1.0.1, tag: tag1, v1.0, feature1) HEAD@{2}: commit: c
ed9faf0 HEAD@{3}: commit: abd
9ae1672 HEAD@{4}: commit: 79
33ed581 HEAD@{5}: commit: 8
5d4ac19 HEAD@{6}: commit: 1256
0ddd56b HEAD@{7}: commit: 34
6c5c2ba HEAD@{8}: reset: moving to HEAD
6c5c2ba HEAD@{9}: revert: Revert "增加想看也行"
fb03f40 HEAD@{10}: rebase -i (finish): returning to refs/heads/feature1
fb03f40 HEAD@{11}: rebase -i (continue): 增加想看也行
fd975e8 (origin/master, origin/HEAD) HEAD@{12}: rebase -i (start): checkout fd975e
19e19ea HEAD@{13}: rebase -i (finish): returning to refs/heads/feature1
19e19ea HEAD@{14}: rebase -i (continue): 增加想看也行
afab1a4 HEAD@{15}: commit (amend): 不给就是不给
2614991 (master) HEAD@{16}: rebase -i: fast-forward
fd975e8 (origin/master, origin/HEAD) HEAD@{17}: rebase -i (start): checkout fd975e
d3e26e7 HEAD@{18}: rebase -i (finish): returning to refs/heads/feature1
```

用途：

查看指定的引用（HEAD 或 branch）的移动历史，从而找到之前的某个特定 commit

问题和建议？

课上技术相关的问题，都可以去群里和大家讨论，对于比较通用的、有价值的问题，可以去我们的知识星球提问。

具体技术之外的问题和建议，都可以找丢物线（微信：diuwuxian），丢丢会为你解答技术以外的一切。



觉得好？

如果你觉得课程很棒，欢迎给我们好评呀！<https://ke.qq.com/comment/index.html?cid=381952>

一定要是你真的觉得好，再给我们好评。不要仅仅因为对扔物线的支持而好评（报名课程已经是你最大的支持了，再不够的话 B 站多来点三连我也很开心），另外我们也坚决不做好评返现等任何的交易。我们只希望，在课程对你有帮助的前提下，可以看到你温暖的评价。

更多内容：

- 网站：<https://hencoder.com>；<https://kaixue.io>
- 各大搜索引擎、微信公众号、微博、知乎、掘金、哔哩哔哩、YouTube、西瓜视频、抖音、快手、微视：统一账号「扔物线」，我会持续输出优质的技术内容，欢迎大家关注。
- 哔哩哔哩快捷传送门：<https://space.bilibili.com/27559447>

大家如果喜欢我们的课程，还请去扔物线的哔哩哔哩，帮我素质三连，感谢大家！