

git reset --hard HEAD^ 和 git reset HEAD^ 的区别

作者: [whuang](#)

原文链接: <https://ld246.com/article/1516671091985>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

git reset --hard HEAD^和git reset HEAD^的区别

即有没有--hard 有什么区别?

两个命令的功能都是回滚commit 过,但是还没有push的文件

git参数

有--hard

没有--hard
容不会丢失

命令执行后

文件是最新状态

文件是修改未commit的状态,即修改

git的其他命令

1,把修改的文件回滚到修改之前:

前提:该文件不是新增文件,并且没有提交(commit)

```
git checkout -- test.sh
```

```
git checkout -- readme.txt
```

回滚多个文件:

```
git checkout HEAD -- prod.sh src/main/java/com/chanjet/gov/controller/LogViewController.java
```

2,查看提交记录

```
git log --pretty=oneline
```

3,回滚commit过的文件

前提:commit过,但是没有push

```
git checkout -- test.sh 命令没用
```

```
git reset --hard 0248d8341472ab9d80d4e43ce89876f502fc975c
```

或者

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

4,创建本地分支

```
git checkout -b dev3333
```

5,把本地分支提交到新的远程分支

```
git -c core.quotepath=false push --progress --porcelain origin refs/heads/dev3333:dev33333  
3_remote --set-upstream
```

简化版本:

```
git push origin refs/heads/dev3333:dev33333333_remote --set-upstream
```

6,删除远程分支

```
git push origin --delete dev33333333_remote
```

7,拉去最新信息

```
git fetch origin --progress --prune
```

8,删除远程分支之后,如何删除本地分支和远程分支的关联关系?

```
git branch --unset-upstream
```

9,删除本地分支

```
git branch -d dev3333
```

```
w ~/code/mygit/back/chanjet_web_store ❯ prod_unitPrice_05_18 ❯ git branch -d prod_unitPrice_0518
```

```
error: The branch 'prod_unitPrice_0518' is not fully merged.
```

```
If you are sure you want to delete it, run 'git branch -D prod_unitPrice_0518'.
```

从某次提交拉出分支,

```
git 从某次提交创建分支
```

```
git checkout -b prod_12_7 c9b5ad1cc43b0cbf230a7a1d6054dfdc8c0c5c88
```

```
Switched to a new branch 'prod_12_7'
```

修改远程服务器地址

```
修改remote url:
```

```
tv_mobile git:(master) git remote -v
```

```
origin https://github.com/whuanghkl/tv_mobile.git (fetch)
```

```
origin https://github.com/whuanghkl/tv_mobile.git (push)
```

```
git remote set-url origin http://git.yhskyc.com/root/tv_mobile.git
```