

git回滚操作

<https://blog.csdn.net/ligang2585116/article/details/71094887>

场景一：

糟了，我刚把不想要的代码，commit到本地仓库中了，但是还没有做push操作！

场景二：

彻底完了，刚线上更新的代码出现问题了，需要还原这次提交的代码！

场景三：

刚才我发现之前的某次提交太愚蠢了，现在想要干掉它！

撤销

上述场景一，在未进行git push前的所有操作，都是在“本地仓库”中执行的。我们暂且将“本地仓库”的代码还原操作叫做“撤销”！

情况一：文件被修改了，但未执行git add操作(working tree内撤销)

```
git checkout fileName
git checkout .
```

情况二：同时对多个文件执行了git add操作，但本次只想提交其中一部分文件

```
$ git add *
$ git status
# 取消暂存
$ git reset HEAD <filename>
```

情况三：文件执行了git add操作，但想撤销对其的修改(index内回滚)

```
# 取消暂存
git reset HEAD fileName
# 撤销修改
git checkout fileName
```

情况四：修改的文件已被git commit，但想再次修改不再产生新的Commit

```
# 修改最后一次提交
$ git add sample.txt
$ git commit --amend -m"说明"
```

情况五：已在本地进行了多次git commit操作，现在想撤销到其中某次Commit

```
git reset [--hard|soft|mixed|merge|keep] [commit|HEAD]
具体参数和使用说明，请查看：Git Pro深入浅出（二）中的重置揭秘部分
```

回滚

上述场景二，已进行git push，即已推送到“远程仓库”中。我们将已被提交到“远程仓库”的代码还原操作叫做“回滚”！注意：对远程仓库做回滚操作是有风险的，需提前做好备份和通知其他团队成员！

如果你每次更新线上，都会打tag，那恭喜你，你可以很快的处理上述场景二的情况

```
git checkout <tag>
如果你回到当前HEAD指向
git checkout <branch_name>
```

情况一：撤销指定文件到指定版本

```
# 查看指定文件的历史版本
git log <filename>
# 回滚到指定commitID
git checkout <commitID> <filename>
```

情况二：删除最后一次远程提交

方式一：使用revert

```
git revert HEAD
git push origin master
```

方式二：使用reset

```
git reset --hard HEAD^
git push origin master -f
```

二者区别:

- **revert**是放弃指定提交的修改，但是会生成一次新的提交，需要填写提交注释，以前的历史记录都在；
- **reset**是指将**HEAD**指针指到指定提交，历史记录中不会出现放弃的提交记录。

情况三：回滚某次提交

```
# 找到要回滚的commitID
git log
git revert commitID
```

删除某次提交

```
git log --oneline -n5
```

```
1166140 test2
4fa6b49 test1
5b3ba7a test
8a5a3e4 es6 proxy
241eb4b //dataStructure sort
```

Git撤销&回滚操作-log

```
git rebase -i "commit id"^^
```

注意：需要注意最后的`号，意思是commit id的前一次提交

```
git rebase -i "5b3ba7a"^^
```

```
pick 5b3ba7a test
pick 4fa6b49 test1
pick 1166140 test2

# Rebase 8a5a3e4..1166140 onto 8a5a3e4 (3 command(s))
#
# Commands:
# p, pick = use commit
# r, reword = use commit, but edit the commit message
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
# f, fixup = like "squash", but discard this commit's log message
# x, exec = run command (the rest of the line) using shell
# d, drop = remove commit
#
# These lines can be re-ordered; they are executed from top to bottom
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
http://blog.csdn.net/ligang2585116
```

Git撤销&回滚操作-rebase

在编辑框中删除相关commit，如pick 5b3ba7a test2，然后保存退出（如果遇到冲突需要先解决冲突）！

```
git push origin master -f
```

通过上述操作，如果你想对历史多个commit进行处理或者，可以选择**git rebase -i**，只需删除对应的记录就好。rebase还可对 commit 消息进行编辑，以及合并多个commit。