

时间机器

在`test`文件中随意改动，然后提交

```
git commit -a -m "意外改动"
```

`git log`，增加了一条纪录：

```
commit d9b03125921d20482937f43ea0bdbfbfb7fe1745Author: kardinal <2999am@gmail.com>Date: Wed Nov 5
```

使用`git reset`命令回溯到历史版本：

```
git reset ❶ HEAD^❷ git log ❸git diff ❹
```

- ❶ `git reset`默认使用`--mixed`选项
- ❷ `HEAD`表示当前版本，`HEAD^`表示前一个版本，`HEAD^^`表示前两个版本，`HEAD~4`表示前四个版本；也可以使用“版本标签”或“版本ID”来指定版本（只要前几位就可以了）
- ❸ 可以看到版本纪录中最后一次提交已经取消
- ❹ 可以看到，`--mixed`选项回溯到提交到索引之前的状态

`git reset --soft`回溯到已提交到索引但未提交到版本库的状态

```
git commit -a -m "意外改动"❶git reset --soft HEAD^ ❷git log ❸git diff ❹git diff --cached ❺
```

- ❶ 再一次将这些改变提交
- ❷ 使用`--soft`选项回溯到上一版本
- ❸ 版本纪录中已取消该版本
- ❹ 改动纪录中没有任何内容
- ❺ 改动已被提交到索引，但是未提交到版本库，所以缓存的改动纪录还可以查看



注意

`git reset` 回溯到`git add`之前的状态；`git reset --soft`回溯到`git add`之后的状态

以上方法回溯到历史版本，只是回溯版本库和索引的纪录，而文件的内容并不会回溯到之前的状态，使用`git reset -hard`命令，将文件内容也一同回溯

```
git commit -a -m "意外改动"❶git reset --hard HEAD^ ❷git log ❸git diff --cached ❹cat test❺
```

- ❶还得提交一次，谁让它是“意外改动”
- ❷ 使用`--hard`选项回溯到上一版本
- ❸ 版本纪录中已取消该版本
- ❹ 没有任何改动纪录待提交
- ❺ 文件内容回溯到上一版本的状态

`--hard`选项存在一定风险，因为很多情况下，你不能确定内容算不算“意外改动”。这时，可以新建一个分支，在这个分支中进行回溯，处理完成后合并两个分支，参见“[分支管理](#)”一节