### 版本回退

* HEAD指向的版本就是当前版本，因此，Git允许我们在版本的历史之间穿梭，使用命令git reset --hard commit\_id。

--hard参数有啥意义？--hard会回退到上个版本的已提交状态，而--soft会回退到上个版本的未提交状态，--mixed会回退到上个版本已添加但未提交的状态。

另外一种写法：git reset --hard HEAD^

在Git中，用HEAD表示当前版本，也就是最新的提交1094adb...（注意我的提交ID和你的肯定不一样），上一个版本就是HEAD^，上上一个版本就是HEAD^^，当然往上100个版本写100个^比较容易数不过来，所以写成HEAD~100

* 穿梭前，用git log可以查看提交历史，以便确定要回退到哪个版本。
* 要重返未来，用git reflog查看命令历史，以便确定要回到未来的哪个版本。

### 撤销修改（版本回退会修改提交历史，撤销修改不会

git reset命令既可以回退版本，也可以把暂存区的修改回退到工作区。

git reset HEAD readme.txt

场景1：当你改乱了工作区某个文件的内容，想直接丢弃工作区的修改时，用命令git checkout -- file。

场景2：当你不但改乱了工作区某个文件的内容，还添加到了暂存区时，想丢弃修改，分两步，第一步用命令git reset HEAD <file>，就回到了场景1，第二步按场景1操作。

场景3：已经提交了不合适的修改到版本库时，想要撤销本次提交，参考[版本回退](https://liaoxuefeng.com/books/git/time-travel/reset/index.html)一节，不过前提是没有推送到远程库。

### 删除文件

rm test.txt

这个时候，Git知道你删除了文件，因此，工作区和版本库就不一致了，git status命令会立刻告诉你哪些文件被删除了：deleted: file.txt

使用rm删除后，可以使用**git restore file.txt**恢复文件

命令git rm用于删除一个文件，相当于rm+add。如果一个文件已经被提交到版本库，那么你永远不用担心误删，但是要小心，你只能恢复文件到最新版本，你会丢失****最近一次提交后****

