目录：

[一、 管理修改 1](#_Toc13009)

[二、 撤销修改 3](#_Toc20456)

[三、 删除文件 3](#_Toc1996)

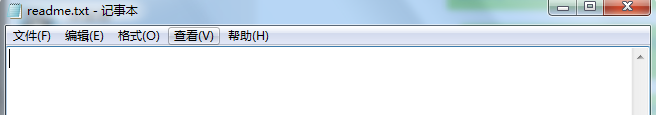
## 管理修改

前提：

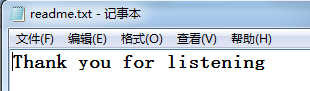
有一个版本库，里面包含一个**readme.txt** 文件

简单的例子：

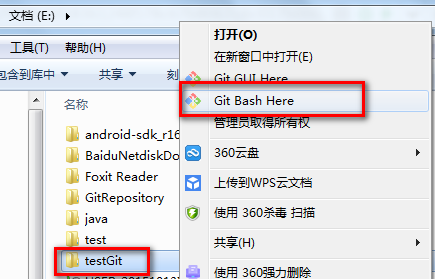
刚开始什么信息都没有：

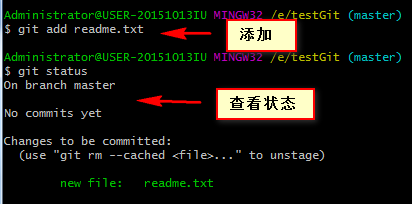


现在往里面添加一行信息（保存信息并退出）：

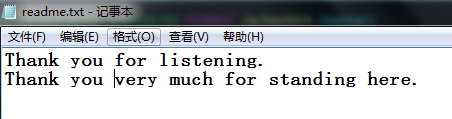


打开git Bash Here，将文件修改添加到git 缓存区。

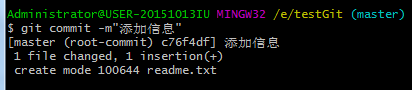




再打开**readme.txt** 文件，往里面添加一行信息（保存信息并退出）。



在git Bash Here中提交信息。



可以看到提示信息，只有一行信息被 insertion(+)。

好了，到这里我们就会有个疑问了，**为什么第二次的修改没有被提交？**

我们来回顾一下以上的整个过程：

第一次修改 -> **git add** -> 第**二**次修改 -> **git commit**

我们前面讲了，Git管理的是修改，当你用git add命令后，在工作区的第一次修改被放入暂存区，准备提交，但是，在工作区的第二次修改并没有放入暂存区，所以，git commit只负责把暂存区的修改提交了，也就是第一次的修改被提交了，第二次的修改不会被提交。

**结论：**

每一次文件的修改，都要先将修改的内容添加到git的缓存区（通过命令 git add 文件名）完成修改后，再通过git commit命令提交修改。

## **撤销修改**

一般要进行撤销修改的几个常见的场景：

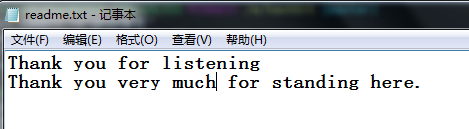
场景1：当你改乱了工作区某个文件的内容，想直接丢弃工作区的修改时，用命令git checkout -- file。

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（不过前提是没有推送到远程库）。

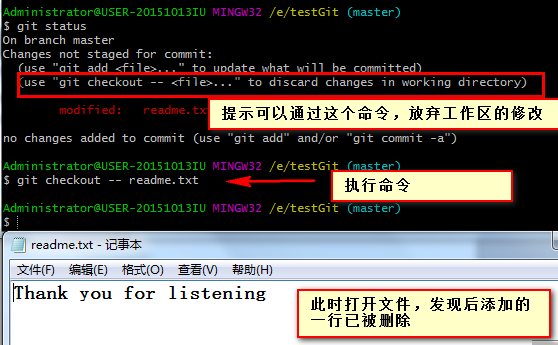
场景一、

比如在readme.txt文件后添加了一句话’Thank you very much for standing here’。



后面你发现不应该添加。这个时候有两个方法：

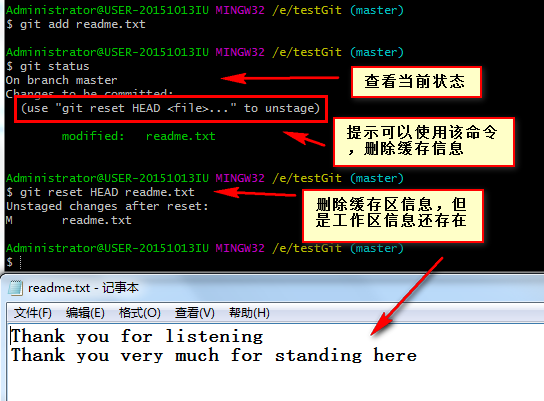
1. 手动打开readme.txt文件，将多余信息删除。
2. 通过命令“git checkout -- readme.txt”



场景二、

在readme.txt文件后添加了一句话’Thank you very much for standing here’。

通过命令git add readme.txt将修改的信息加入了缓存，但是没有执行 git commit，此时想撤销缓存区的修改。



此时，就相当于回到场景一。

我们再通过“git checkout -- readme.txt”删除工作区信息，就可以了。

场景三、

参考版本回退步骤。

## **删除文件**

在Git中，删除也是一个修改操作，我们实战一下，先添加一个新文件test.txt到Git并且提交：

$ git add test.txt

$ git commit -m "add test.txt"

[master b84166e] add test.txt

1 file changed, 1 insertion(+)

create mode 100644 test.txt

一般情况下，你通常直接在文件管理器中把没用的文件删了，或者用rm命令删了：

$ 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：

$ git status

On branch master

Changes not staged for commit:

(use "git add/rm <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

deleted: test.txt

no changes added to commit (use "git add" and/or "git commit -a")

现在你有两个选择，一是确实要从版本库中删除该文件，那就用命令git rm删掉，并且git commit：

$ git rm test.txt

rm 'test.txt'

$ git commit -m "remove test.txt"

[master d46f35e] remove test.txt

1 file changed, 1 deletion(-)

delete mode 100644 test.txt

现在，文件就从版本库中被删除了。

 小提示：先手动删除文件，然后使用git rm <file>和git add<file>效果是一样的。

另一种情况是删错了，因为版本库里还有呢，所以可以很轻松地把误删的文件恢复到最新版本：

$ git checkout -- test.txt

git checkout其实是用版本库里的版本替换工作区的版本，无论工作区是修改还是删除，都可以“一键还原”。

小结：

命令git rm用于删除一个文件。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你修改的内容。