**Git后悔药 - 版本回退操作（一）**

git reset命令可以实现Git版本回退，其有三个选项，可以完成三种不同效果的回退。

**（1）git reset --soft命令**

git reset --soft commit-id命令：回退到指定版本。（soft：柔软的）

**该命令仅仅修改分支中的HEAD指针的位置，不会改变工作区与暂存区中的文件的版本。**

**实现上是只做了一件事情，就是移动HEAD指针的指向，指向了指定的提交版本。**

示例开始：

首先在版本库中的readme.txt文件中添加一行内容，并提交该内容。我们的目的就是要再回退到该版本。

**1）查看本地版本库日志**

|  |
| --- |
| Bash # 1.使用git log查看历史版本记录 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git log --oneline f4da0ae (HEAD -> master) 第3次提交，新增内容：readme.txt file v3 05f5ff9 第2次提交，新增内容：readme.txt file v2 75b4466 第1次提交，创建readme.txt文件  # 2.使用git reflog查看历史版本记录 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git reflog f4da0ae (HEAD -> master) HEAD@{0}: commit: 第3次提交，新增内容：readme.txt file v3 05f5ff9 HEAD@{1}: commit: 第2次提交，新增内容：readme.txt file v2 75b4466 HEAD@{2}: commit (initial): 第1次提交，创建readme.txt文件  # 3.查看readme.txt文件的内容 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ cat readme.txt readme.txt file v1 readme.txt file v2 readme.txt file v3 |

**2）向readme.txt文件中新增一行数据，并提交到本地版本库**

|  |
| --- |
| Bash # 1.新增数据 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ echo "readme.txt file v4" >> readme.txt  # 2.查看readme.txt文件内容 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ cat readme.txt readme.txt file v1 readme.txt file v2 readme.txt file v3 readme.txt file v4  # 3.提交到本地版本库 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git commit -a -m '第4次提交，新增内容：readme.txt file v4' warning: LF will be replaced by CRLF in readme.txt. The file will have its original line endings in your working directory [master 2c4401f] 第4次提交，新增内容：readme.txt file v4  1 file changed, 1 insertion(+)   # 4.现在查看此时本地版本库日志 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git log --oneline 2c4401f (HEAD -> master) 第4次提交，新增内容：readme.txt file v4 f4da0ae 第3次提交，新增内容：readme.txt file v3 05f5ff9 第2次提交，新增内容：readme.txt file v2 75b4466 第1次提交，创建readme.txt文件 |

**3）现在比对工作区与暂存区、暂存区与本地版本库的差异**

|  |
| --- |
| Bash # 1.比对工作区与暂存区中文件的差异 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git diff readme.txt  # 2.比对暂存区与本地版本库中文件的差异 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git diff --cached readme.txt |

我们可以看到此时，工作区、暂存区与本地版本库中的readme.txt文件状态无差异。

**4）开始回退操作，退回到V3版本**

使用git reset --soft HEAD^命令，退回到前一个版本。

|  |
| --- |
| Bash # 1.回退一个提交版本 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git reset --soft HEAD^ |

**5）回退后，对比工作区、暂存区与本地库中版本中文件的差异**

|  |
| --- |
| Bash # 1.比对工作区与暂存区中文件的差异 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git diff readme.txt  # 2.比对暂存区与本地版本库中文件的差异 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git diff --cached readme.txt diff --git a/readme.txt b/readme.txt index 1e6534a..47b238c 100644 --- a/readme.txt +++ b/readme.txt @@ -1,3 +1,4 @@  readme.txt file v1  readme.txt file v2  readme.txt file v3 +readme.txt file v4  # 3.比对工作区与本地版本库中文件的差异 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git diff HEAD readme.txt diff --git a/readme.txt b/readme.txt index 1e6534a..47b238c 100644 --- a/readme.txt +++ b/readme.txt @@ -1,3 +1,4 @@  readme.txt file v1  readme.txt file v2  readme.txt file v3 +readme.txt file v4 |

回退后，我们再次对比了工作区、暂存区与本地库中版本中文件的差异：

* 发现工作区与暂存区内容没有差异，
* 暂存区与本地库中的版本出现了差异，
* 工作区与本地库中的版本出现了差异。

说明：工作区和暂存区中的内容没有回退，但是本地库中的内容回退到了之前的版本。（重要）

**6）查看本地版本库的提交日志信息**

|  |
| --- |
| Bash # 1.使用git log查看历史版本记录 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git log --oneline f4da0ae (HEAD -> master) 第3次提交，新增内容：readme.txt file v3 05f5ff9 第2次提交，新增内容：readme.txt file v2 75b4466 第1次提交，创建readme.txt文件  # 2.使用git reflog查看历史版本记录 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git reflog f4da0ae (HEAD -> master) HEAD@{0}: reset: moving to HEAD^ 2c4401f HEAD@{1}: commit: 第4次提交，新增内容：readme.txt file v4 f4da0ae (HEAD -> master) HEAD@{2}: commit: 第3次提交，新增内容：readme.txt file v3 05f5ff9 HEAD@{3}: commit: 第2次提交，新增内容：readme.txt file v2 75b4466 HEAD@{4}: commit (initial): 第1次提交，创建readme.txt文件 |

从上我们可以看到：（重点）

* 使用git log命令查看历史版本记录，发现已经看不到第4次提交了。
* 使用git reflog命令查看历史版本记录，第四次提交这个版本仍然存在的。

**7）恢复到回退前版本**

因为前面说了，git reset --soft命令回退，只是移动HEAD指针，也就是本地版本库进行退回，而工作区和暂存区的内容都不回退。

所以若要恢复到回退之前的版本，可以直接将暂存区中的数据commit提交到本地版本库即可。

|  |
| --- |
| Bash # 1.查看工作目录中文件的状态 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git status On branch master Changes to be committed:  (use "git restore --staged <file>..." to unstage)  modified: readme.txt # 我们可以看到readme.txt文件是修改已暂存状态，提交即可。 # 即当前的Git状态为暂存区中的版本尚未提交时的状态。  # 2.提交操作 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git commit -m '第5次提交，append v4 again!' [master 4399da4] 第5次提交，append v4 again!  1 file changed, 1 insertion(+)   # 3.查看工作目录中文件状态 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git status On branch master nothing to commit, working tree clean   # 4.使用git log查看历史版本记录 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git log --oneline 4399da4 (HEAD -> master) 第5次提交，append v4 again! f4da0ae 第3次提交，新增内容：readme.txt file v3 05f5ff9 第2次提交，新增内容：readme.txt file v2 75b4466 第1次提交，创建readme.txt文件 # 已经看不到第四次提交了。所以git log命令是看不到全部历史版本的。  # 5.使用git reflog查看历史版本记录 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/learngit (master) $ git reflog 4399da4 (HEAD -> master) HEAD@{0}: commit: 第5次提交，append v4 again! f4da0ae HEAD@{1}: reset: moving to HEAD^ 2c4401f HEAD@{2}: commit: 第4次提交，新增内容：readme.txt file v4 f4da0ae HEAD@{3}: commit: 第3次提交，新增内容：readme.txt file v3 05f5ff9 HEAD@{4}: commit: 第2次提交，新增内容：readme.txt file v2 75b4466 HEAD@{5}: commit (initial): 第1次提交，创建readme.txt文件 # 使用git reflog命令，可以看到全部的历史版本记录。 |

提示：我们可以通过git reset --soft命令，回退到第4次提交。

执行命令：$ git reset --soft 2c4401f

也会生成一个新的commit提交，日志信息如下：

2c4401f (HEAD -> master) HEAD@{0}: reset: moving to 2c4401f

看到最前的的2c4401f和第四次提交的commit一致，说明已经退回到第四次提交了。

作者：繁华似锦Fighting 链接：https://www.jianshu.com/p/1bb377559cd2 来源：简书 著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。