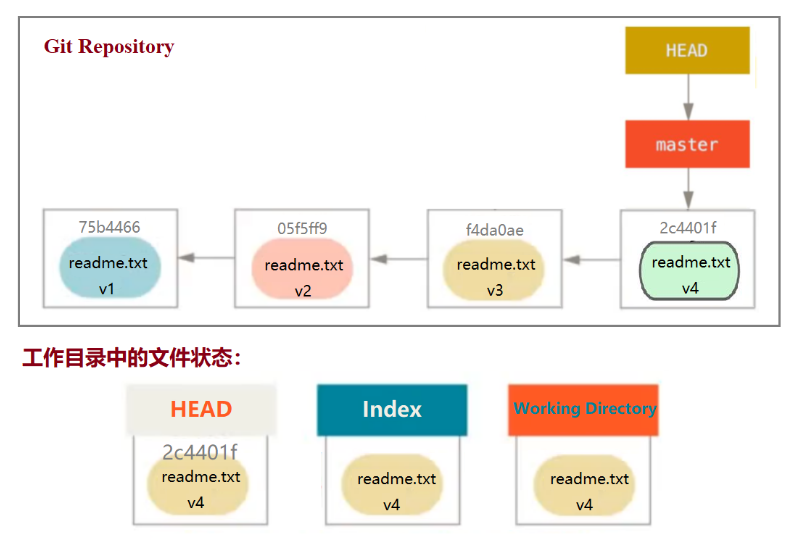
**Git后悔药 - reset版本回退总结**

在Git中进行版本回退需要使用git reset命令。

以前面文章中的示例为例，当我准备在V4版本，回退到V3版本的时候，分支中的提交和工作目录中文件的状态如下图：



我们分别执行了三种回退方式：

* git reset --soft HEAD^：温柔的回退。
* git reset --mixed HEAD^：中等回退。
* git reset --hard HEAD^：强硬的回退。

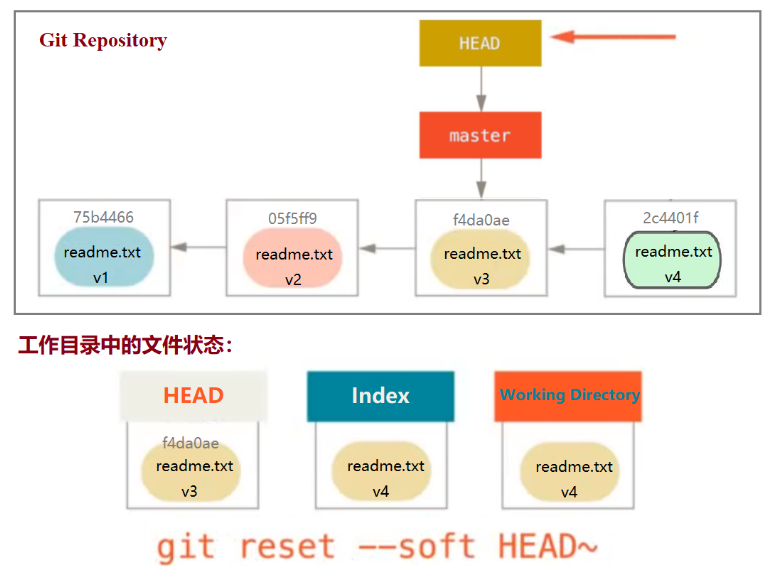
（我们从英文中就可以看出，一个比一个回退的多。）

下面我们一一进行总结。

**1、git reset --soft回退**

当我在V4版本的时候，执行git reset --soft HEAD^命令回退到V3版本。

Git中发生的变化如下图：



依据上图，理解一下发生的事情：本质上就发生了，把HEAD指针指向了V3版本。而工作区和暂存区中的readme.txt文件是没有做任何变动的。

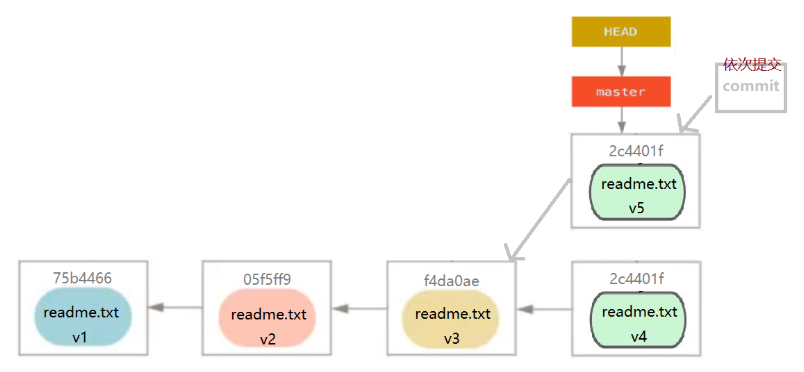
所以你查看本地版本库中的readme.txt文件是V3版本，工作区和暂存区中的readme.txt文件是V4版本。

就等于回滚到了git commit之前的状态。

（我前面文章中有详细的演示）

**拓展：**

当我继续修改readme.txt文件之后，再次提交，会在V3版本之上在创建一个新的commit提交，并移动HEAD指针指向的分支来使其指向该commit提交，这样依次提交下去，如下图：



如果我们使用git log命令查看本地版本库的历史提交信息的时候，就不会出现V4版本提交的信息。会是V1、V2、V3、V5。（我们从前面文章中已经演示了）

但是V4版本是不会在Git中删除的，会永远的存储在Git的本地版本库中。我们可以使用git reflog命令，可以查看该V4版本的提交信息。

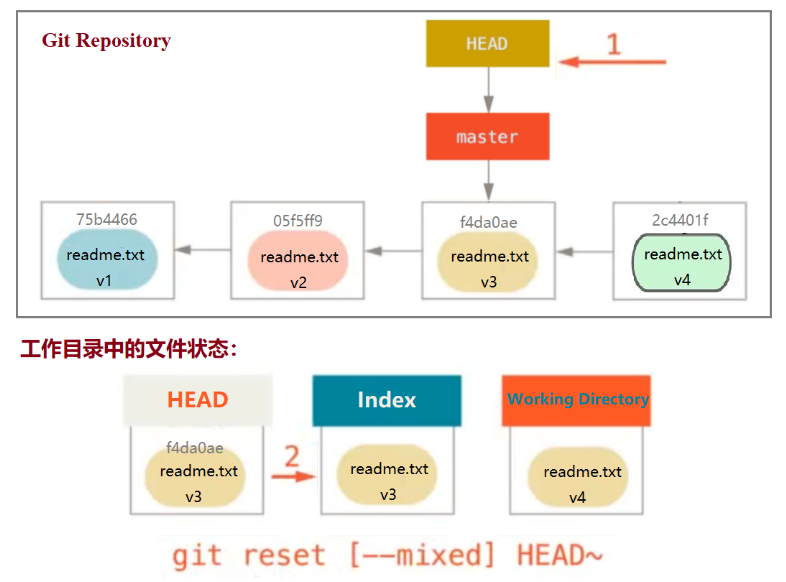
|  |
| --- |
| 提示：只要是本地版本库中HEAD有过的变化，那么git reflog命令就能够显示出来。 |

（关于这点，下面同理，所以下面就不说了。）

**2、git reset --mixed命令**

当我在V4版本的时候，执行git reset --mixed HEAD^命令回退到V3版本。

Git中发生的变化如下图：



理解一下发生的事情，我们可以看到上图中，完成了两步操作：

1. 把HEAD指针指向了V3版本（也就是版本库回退了）。
2. 把暂存区中的readme.txt文件也回退到了V3版本。

而只有工作区中的readme.txt文件内容没有变化。

这说明git reset --mixed命令比git reset --soft命令，多回退了暂存区中的内容。

就等于回滚到了git commit和git add之前的状态。

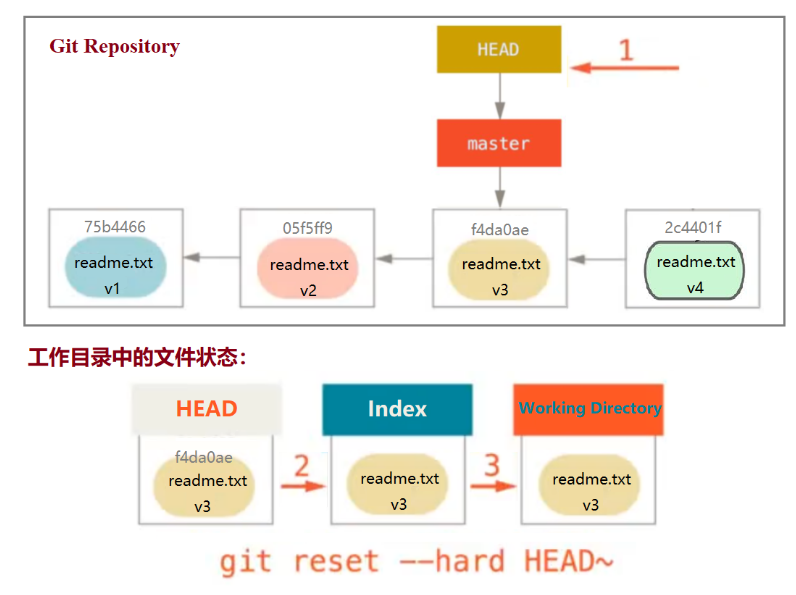
（我前面文章中有详细的演示）

|  |
| --- |
| 提示：因为--mixed参数是git reset命令的默认选项，也就是不写任何参数就默认使用--mixed参数。即git reset HEAD^等同于git reset --mixed HEAD^命令 |

**3、git reset --hard命令**

当我在V4版本的时候，执行git reset --hard HEAD^命令回退到V3版本。

Git中发生的变化如下图：



理解一下发生的事情，我们可以看到上图中，完成了三步操作：

1. 把HEAD指针指向了V3版本（也就是版本库回退了）。
2. 把暂存区中的readme.txt文件也回退到了V3版本。
3. 把工作区中readme.txt文件的修改也复原了。

所以执行完git reset --hard HEAD^命令，是完全回退一个版本。

此时工作区、暂存区、本地版本库中的文件状态都是一致的，都是V3版本。

就等于回滚了一个“编辑文件，添加到暂存区，提交版本库”的整个流程。

（我前面文章中有详细的演示）

**4、注意点**

必须注意，--hard参数是git reset命令唯一的危险用法，是能够使Git会真正地销毁数据的仅有的几个操作之一。

其他任何形式的git reset操作都可以轻松撤消，但是--hard选项不能，因为它强制覆盖了工作目录中的文件。

在这种特殊情况下，我们的Git数据库中的一个提交内，还留有该文件的v4版本，我们可以通过git reflog来找回它。但是若该文件还未提交，Git仍会覆盖它从而导致无法恢复。

作者：繁华似锦Fighting 链接：https://www.jianshu.com/p/73108169d9fa 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。