版本回退

首先，Git必须知道当前版本是哪个版本，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。

使用git reset命令让版本回退

```
$ git reset --hard HEAD^
```

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。

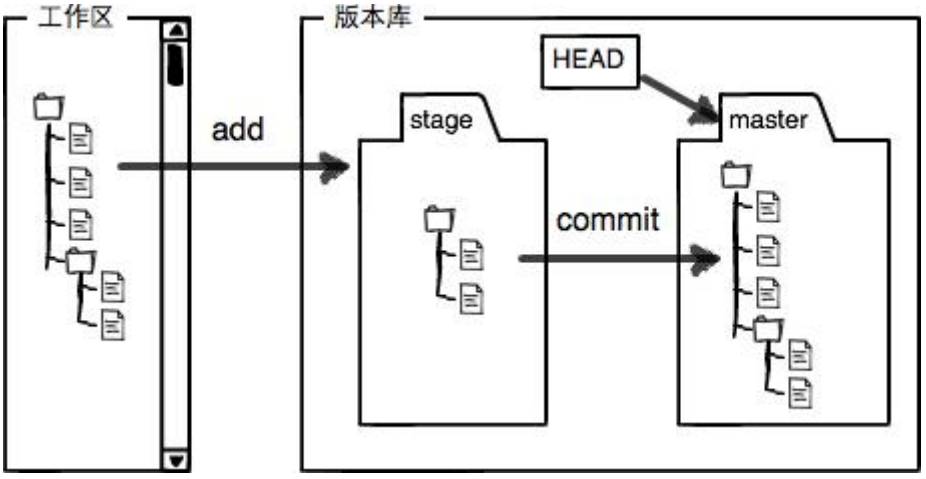
穿梭前，用git log可以查看提交历史，以便确定要回退到哪个版本。

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。

版本回退后强制更新服务器

```
$ git push --force
```

工作区和暂存区关系如下图



stage为暂存区，master为.git文件夹中Git工具为我们创建的第一个分支master和指向master的指针HEAD

提交后，用git diff HEAD -- readme.txt命令可以查看工作区和版本库里面最新版本的区别：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。

从版本库中删除该文件，那就用命令git rm删掉，并且git commit：

Git鼓励大量使用分支：

查看本地分支：git branch

获取所有分支： git fetch

查看本地及远程仓库的分支: git branch -a

查看本地及远程仓库的分支：git branch --all

创建分支：git branch <name>

切换分支：git checkout <name>

创建+切换分支：git checkout -b <name>

合并某分支到当前分支：git merge <name>

删除分支：git branch -d <name>

查看版本： git tag

Git分支十分强大，在团队开发中应该充分应用。

合并分支时，加上--no-ff参数就可以用普通模式合并，合并后的历史有分支，能看出来曾经做过合并，而fast forward合并就看不出来曾经做过合并。