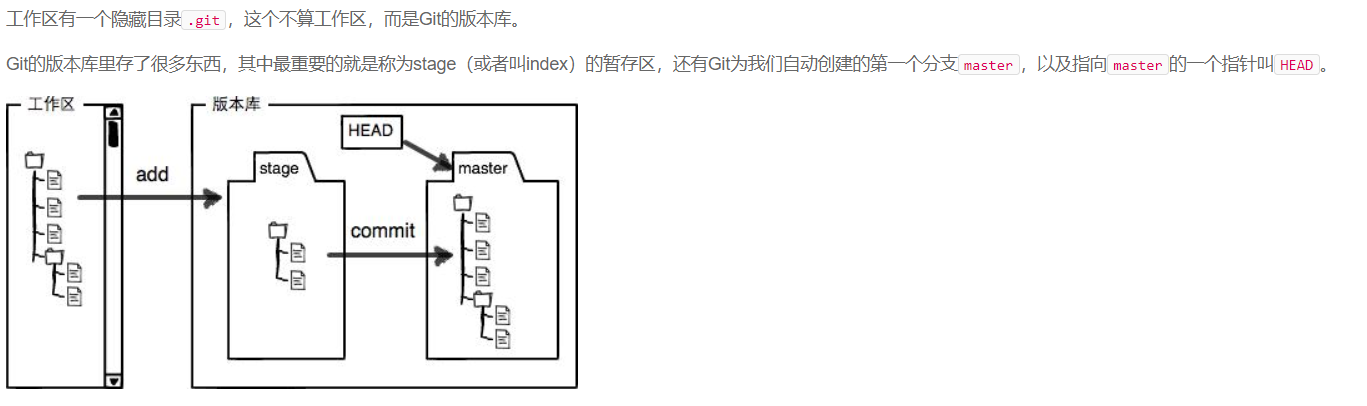
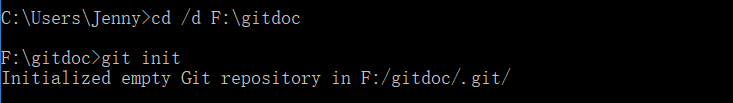
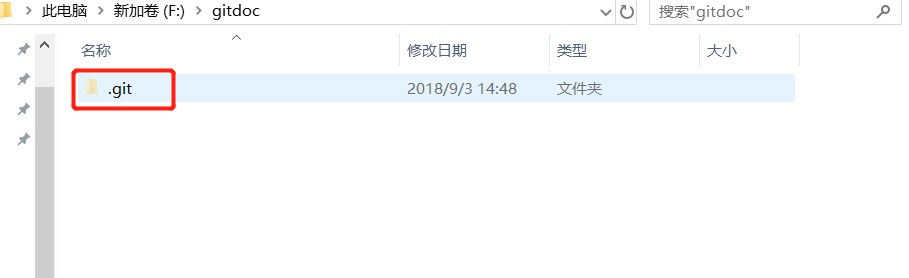
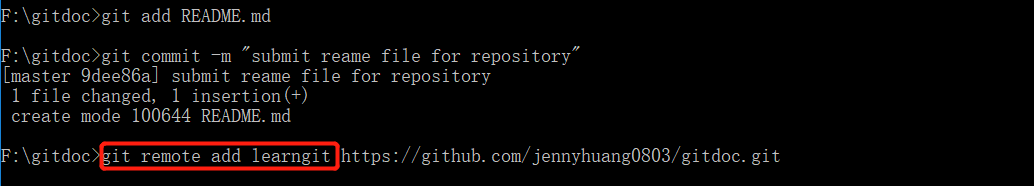
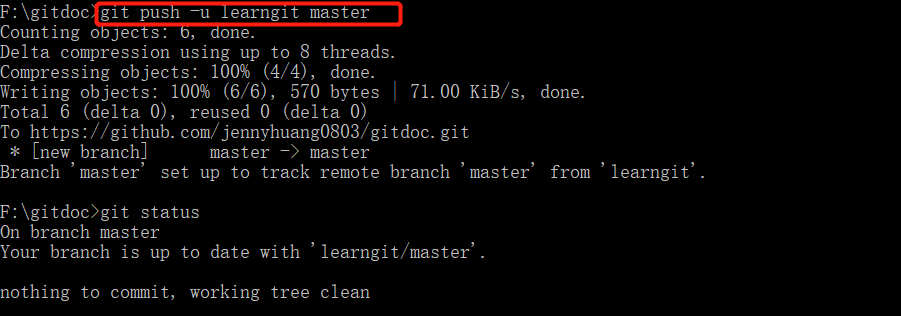
Git基本操作命令

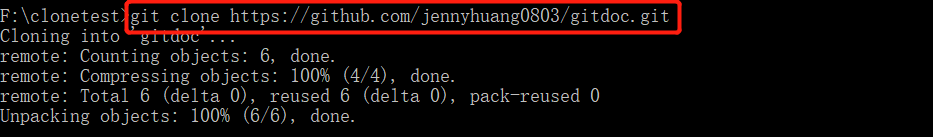
* 仓库结构  
  
* 本地新建数据仓库初始化 ，生成隐藏文件.git，仓库初始化成功  
  command : git init   
    
  
* 创建文件 -> 同步到Github步骤，示例：

1. 空仓库创建README.md文件  


1. 添加到暂存区 -> 发布到仓库 -> 建立GitHub仓库 ->获取仓库URL -> 本地remote创建连接  
     
     
   Note: 把文件添加至暂存区 git add <file>

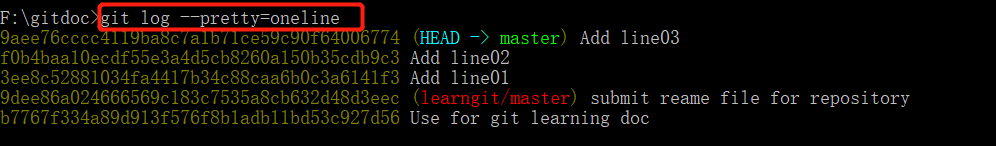
把文件提交至仓库内存 git commit -m “提交的说明”  
Learngit为远程仓库名称

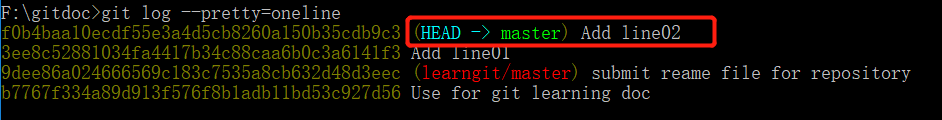
1. 把本地库的内容推送到远程，用git push命令，实际上是把当前分支master推送到远程  
     
   【加上了-u参数，Git不但会把本地的master分支内容推送的远程新的master分支，还会把本地的master分支和远程的master分支关联起来，在以后的推送或者拉取时就可以简化命令】
2. Cool ,同步完成  
   

* 从GitHub同步仓库到本地  
  git clone <仓库URL>  
  
* 版本控制

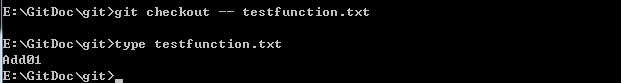
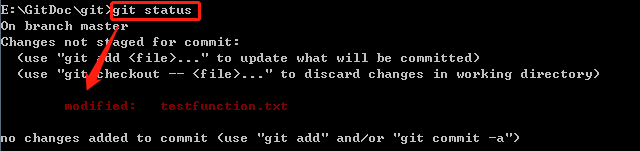
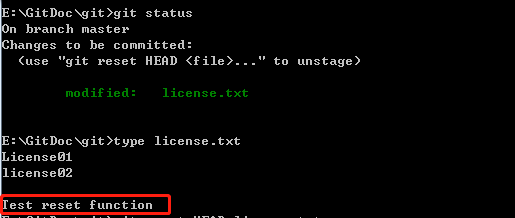
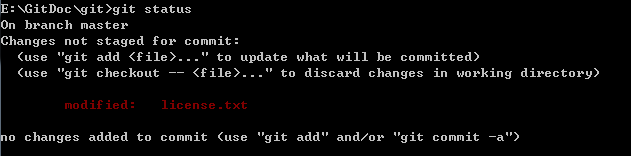
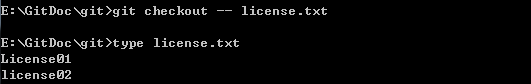
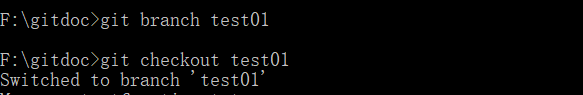
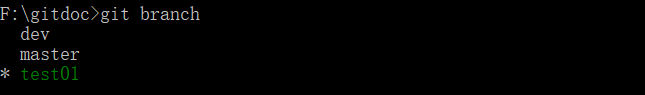
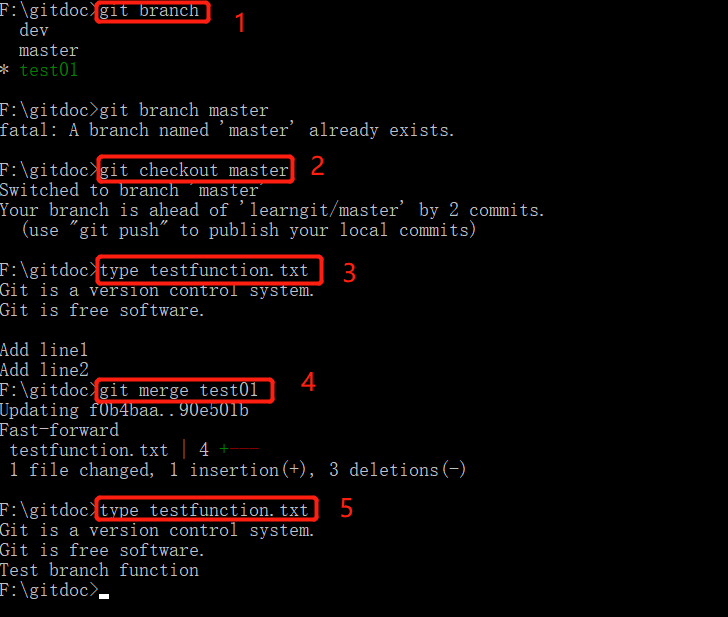
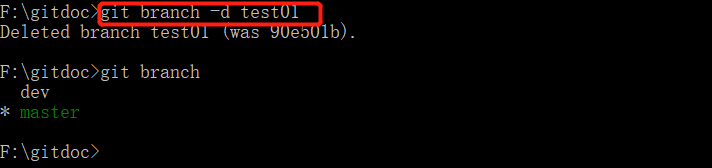
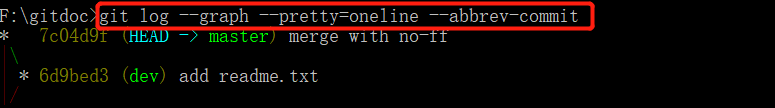
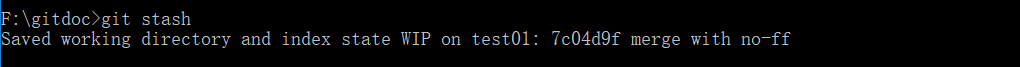
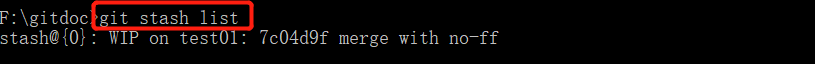
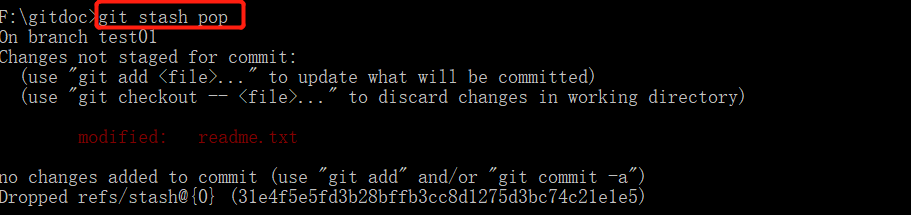
1. 提交至本地仓库后回到之前版本

【在Git中，HEAD表示当前版本，上一个版本就是HEAD^，上上一个版本就是HEAD^^】

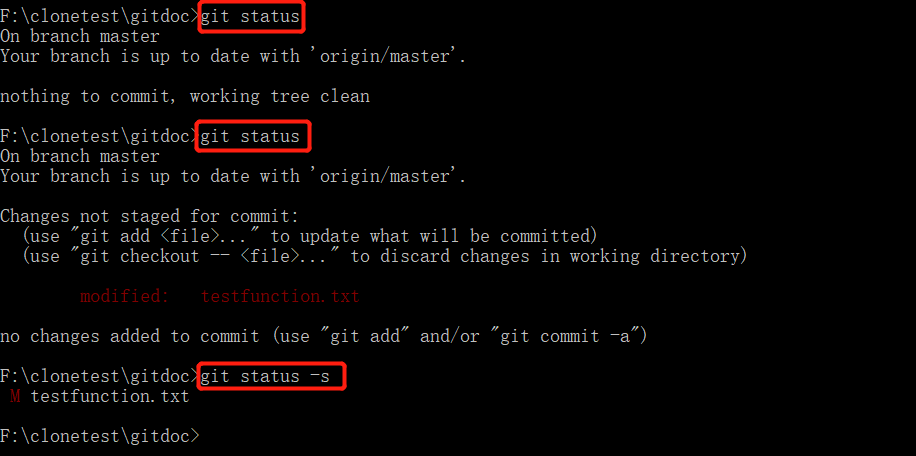


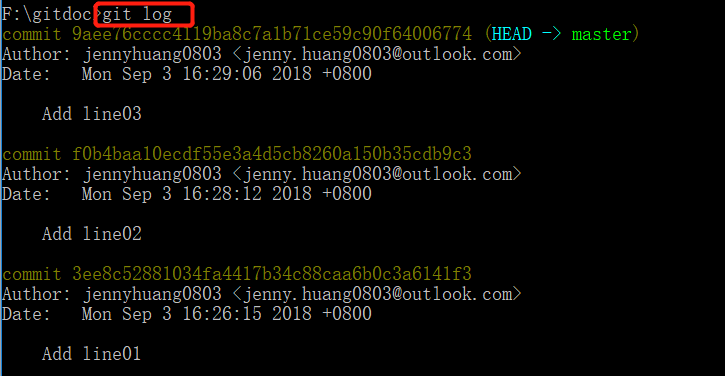
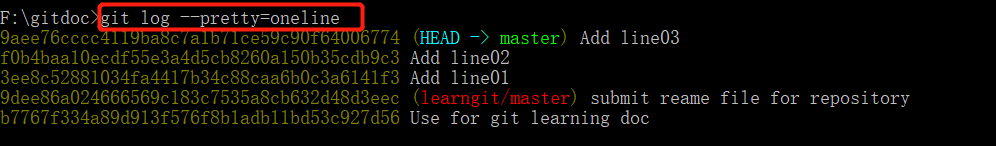
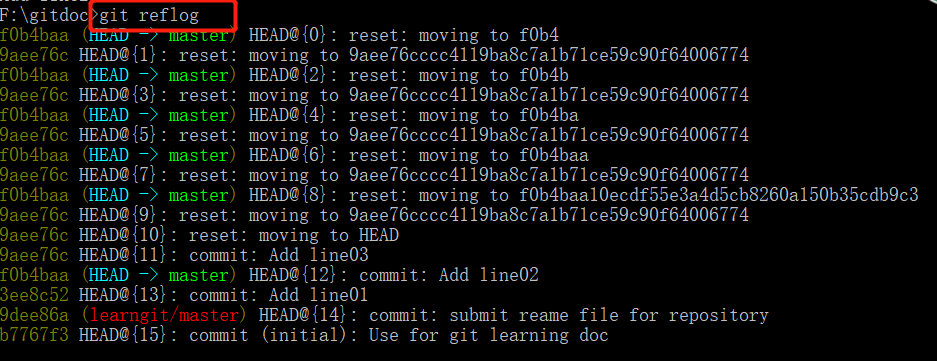
eg: 从“Add line03”-> “Add line02”版本  
command : git reset --hard “command id ”  
  


Note:  
 可用reflog查看历史操作command id记录  


2. 未提交撤销修改  
eg: 工作区域有文件更改未提交  
  
3. 已提交至暂存区撤销至工作区  
  
git reset HEAD <filename.txt> 放回至工作区  
  
显示工作区有修改，暂存区无修改  
  
撤销修改命令  
  
  
分支管理  
1. 创建dev分支，然后切换到dev分支  
  
【git checkout命令加上-b参数表示创建并切换，相当于以下两条命令】  
  
【git branch命令查看当前分支，git branch命令会列出所有分支，当前分支前面会标一个\*号】  
  
  
2. 创建分支 --> 合并修改  
  
 步骤：查看当前分支 -> 在当前分支做修改 - > 切换到Master分支（查看文件并不是修改后得文件） -> Merge 分支 （查看文件为修改后得文件）  
【git merge命令合并指定分支到当前分支，Fast-forward合并是“快进模式”，也就是直接把master指向test01的当前提交】  
  
  
合并后删除分支  
【git branch -D <name>强行删除】  
git branch -d <分支>  
  
  
  
【如果要强制禁用Fast forward模式，Git就会在merge时生成一个新的commit，这样，从分支历史上就可以看出分支信息】  
git log --graph --pretty=oneline --abbrev-commit  
  
  
  
  
3. 储蓄改动  
储存改动 ->切回分支查看储存 -> 恢复  
【stash功能，可以把当前工作现场“储藏”起来】  
  
  
【git stash pop，恢复的同时把stash内容也删了，也可以用git stash apply恢复，但是恢复后，stash内容并不删除，需要用git stash drop来删除】  
  
  
  
通用命令

1. Git status  
   【git status 以查看在上次提交之后是否有修改，加 -s 参数，以获得简短的结果输出。如果没加该参数会详细输出内容】



1. Git log   
   【git log命令显示从最近到最远的提交日志，加上--pretty=oneline参数 减少输出信息，HEAD表示当前版本】  
       
     
     
   Git reflog   
   【Git提供了一个命令git reflog用来记录每一次命令】  
   
2. Git remote

【加上 -v 参数 ，可以看到每个别名的实际链接地址】 