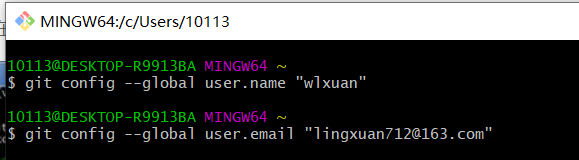
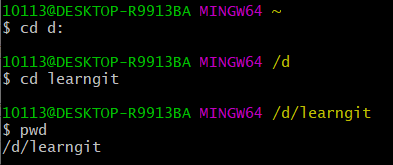
01\_Git

1. Git简介
2. Git是用C语言开发的
3. Git是分布式版本控制系统，没有中央服务器，每个人的电脑就是一个完整的版本库，工作时就无需联网
4. 安装Git



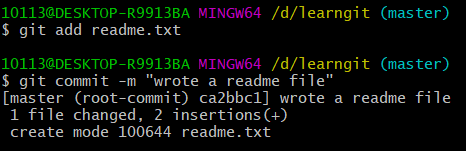
1. 创建版本库
2. 创建并进入learngit目录



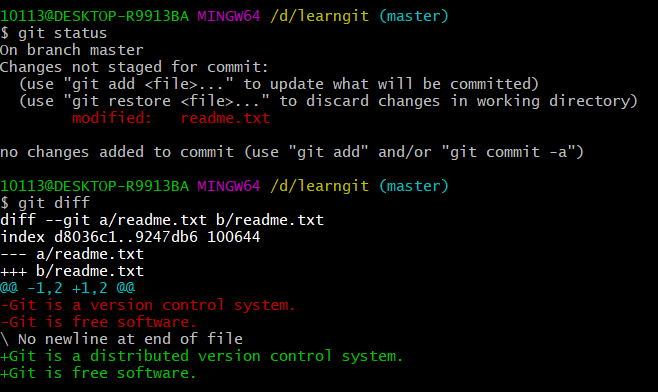
1. 通过git init命令将该目录变成Git可以管理的仓库



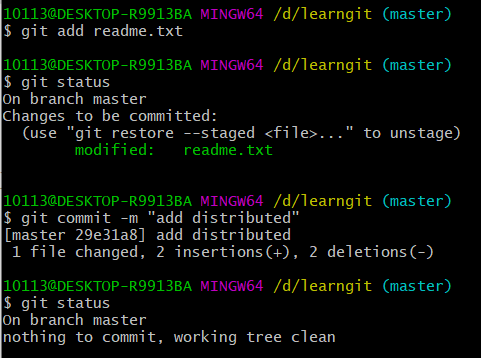
1. 将文件添加至版本库



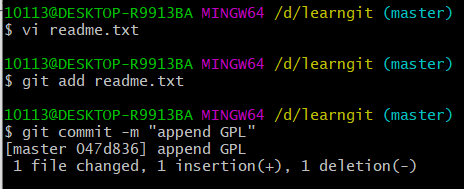
1. 修改文件后查看diff



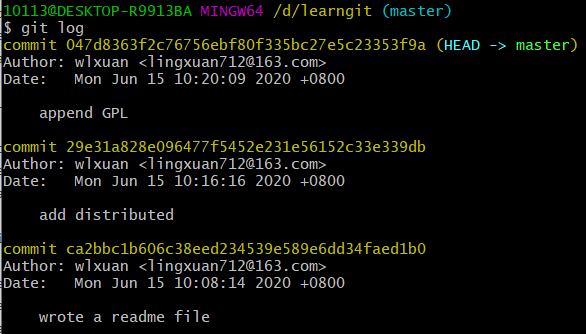
1. 重新上传后查看状态



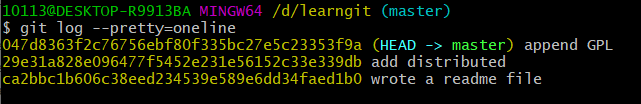
1. 版本回退
2. 再次进行修改



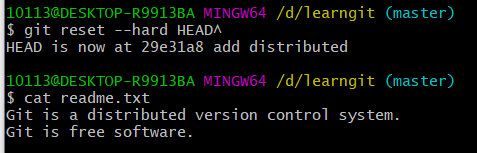
1. 查看历史记录



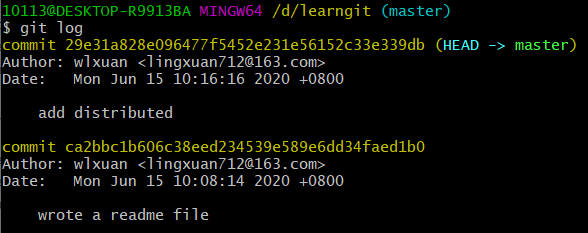
查看简易版本信息



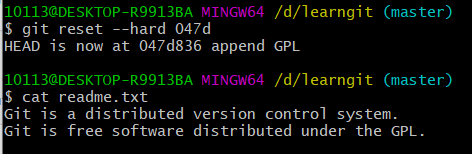
1. 回溯到上一版本



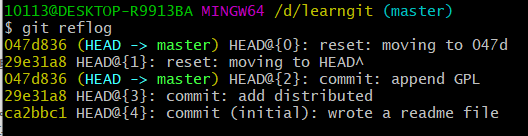
查看当前记录



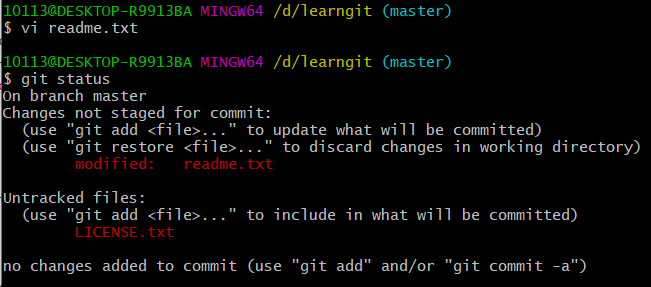
1. 回到上一版本



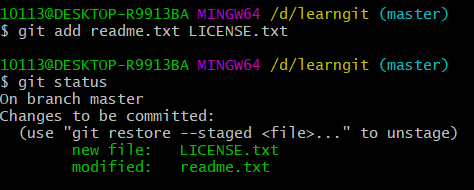
1. 查看命令记录



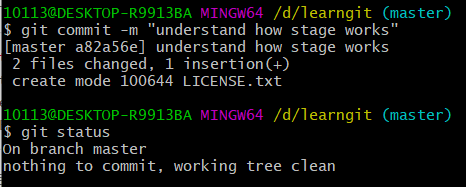
1. 工作区与暂存区
2. 再次修改readme并创建LICENSE



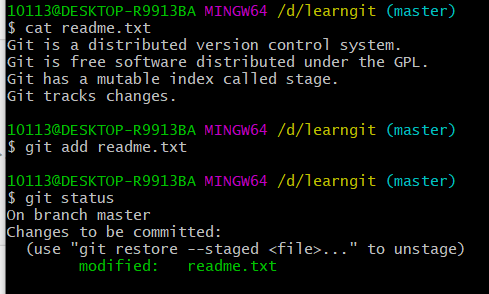
1. readme和LICENSE进入暂存区



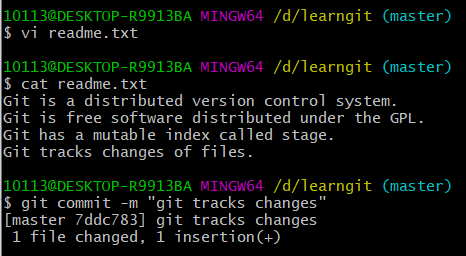
1. commit之后，暂存区清空



1. 管理修改
2. 进行修改并添加

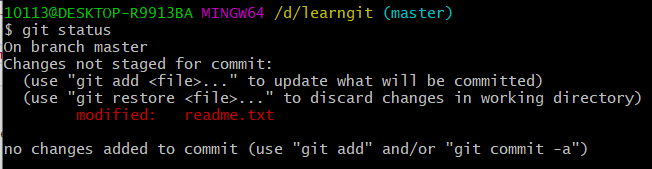


1. 再次修改并提交

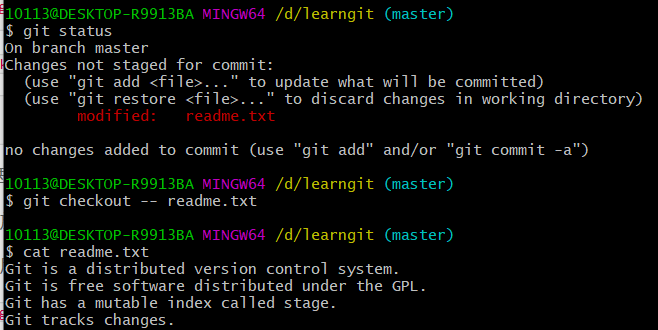


1. 查看状态可知第二次修改并未提交

由此可知，git commit命令提交的是暂存区的内容，git add加入暂存区的修改才会被提交，第二次修改没有add因此也不会被提交。

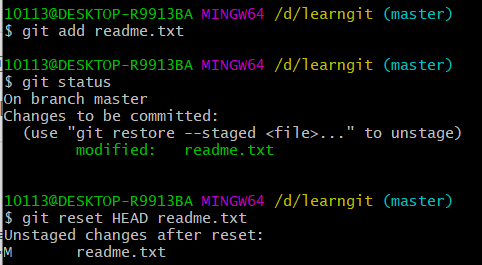


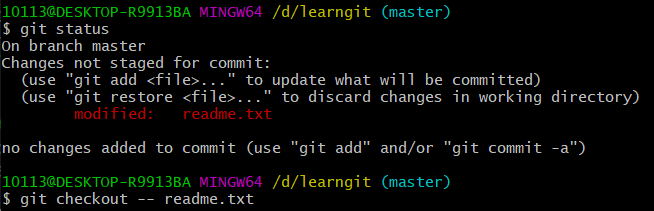
1. 撤销修改
2. 丢弃工作区的修改，使用git checkout -- file



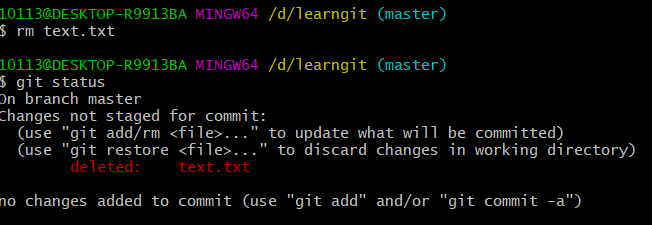
1. 丢弃暂存区的修改

上传到暂存区的内容用reset退回，之后再用checkout丢弃修改

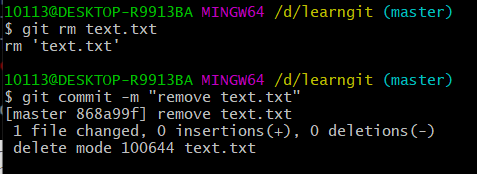




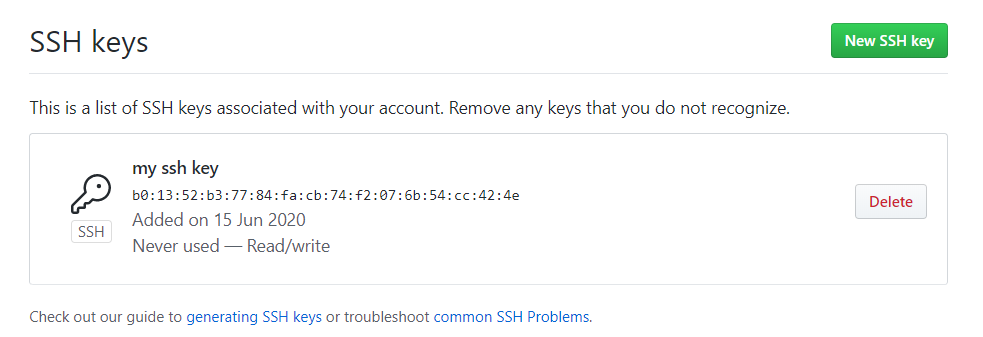
1. 删除文件
2. 删除后与版本库不一致



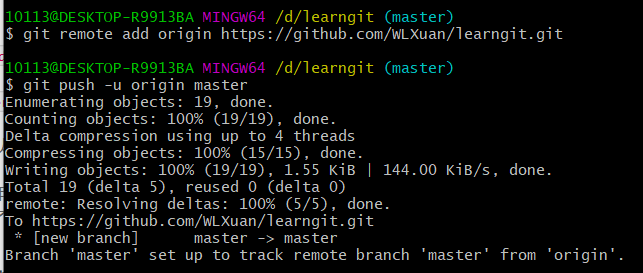
1. 在版本库中删除该文件



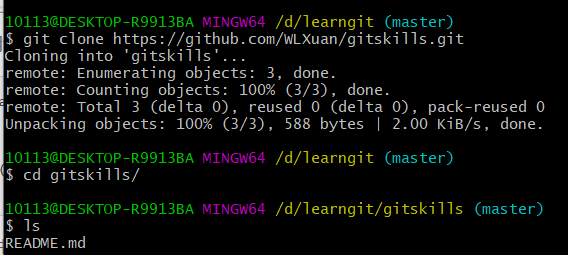
1. 远程仓库
2. add ssh key



1. 上传远程仓库



1. 从远程库克隆文件

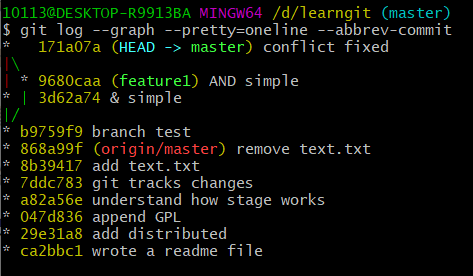


1. 分支管理
2. 创建于合并分支



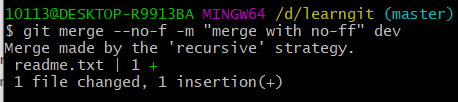
1. 解决冲突

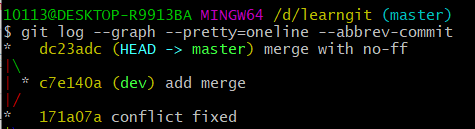
git log --graph命令可以看到分支合并图



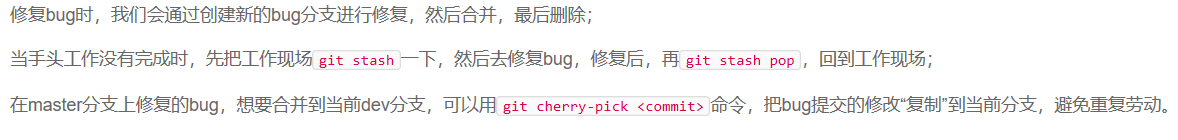
1. 分支管理策略

用no-ff合并分支

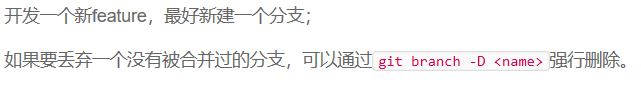




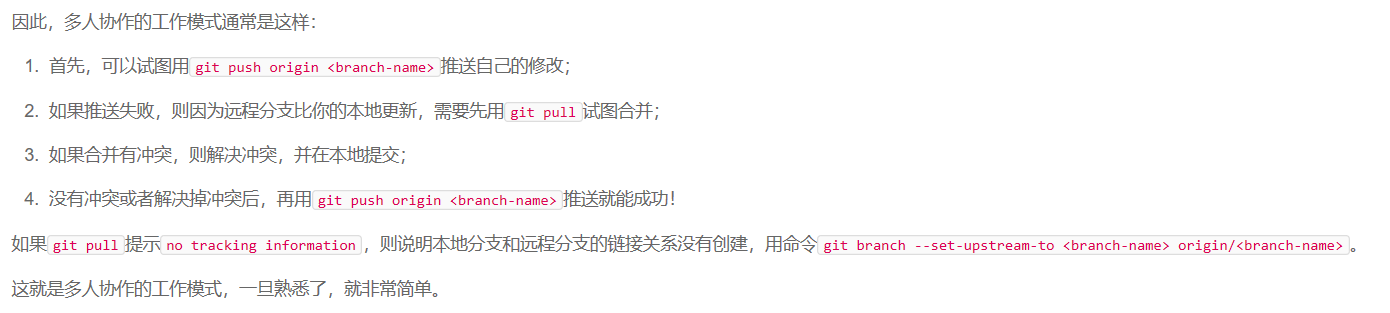
1. bug分支



1. Feature分支

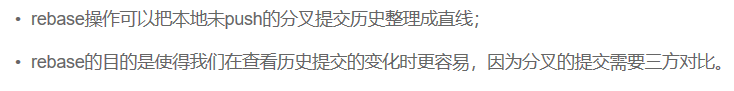


1. 多人协作

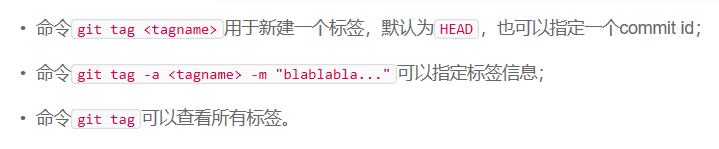


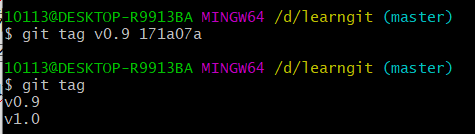


1. Rebase



1. 标签管理
2. 创建标签

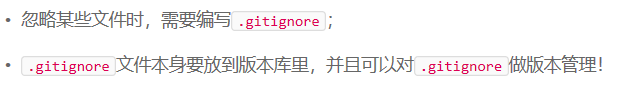




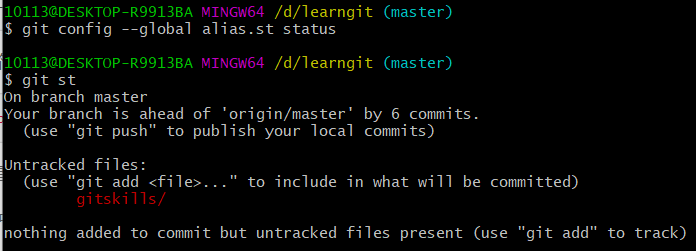
1. 操作标签



1. 使用github
2. 忽略特殊文件



1. 配置别名



1. 使用sourcetree

