9.1

**版本控制系统VCS**

*--version control system*

概述：一种文件管理系统，被它管理的文件内容发生任何变化都能被记录下来，可以根据这个记录对一个文件溯源与特定的场景操作。

应用方式：不仅可以应用于软件源代码的文本文件，而且可以对任何类型的文件进行版本控制

常见的版本控制软件：SVN、GIT

GIT：分布式多人协作版本控制系统(每个人都有一个本地仓库，再上传到远程仓库)

SVN：集中式多人协作版本控制系统(有个专门的服务区去管理代码)

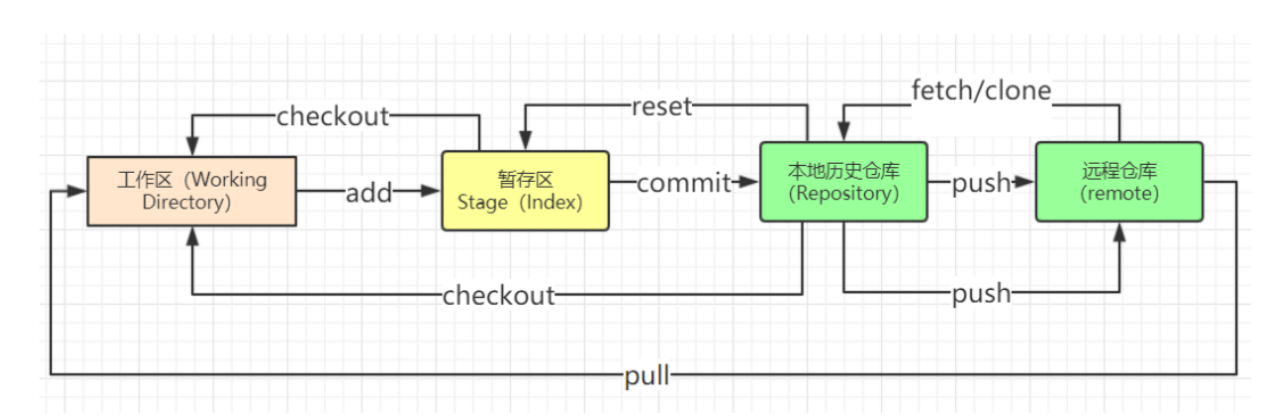
GIT的工作模式：

概括：分为四个区进行控制与管理

工作区：改写代码 add->暂存区

暂存区：暂时存放文件 commit->本地仓库

本地仓库：



切换盘符



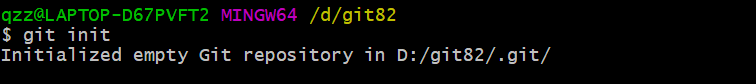
创建文件夹



进入文件夹



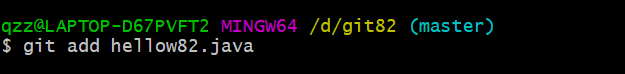
初始化



创建文件



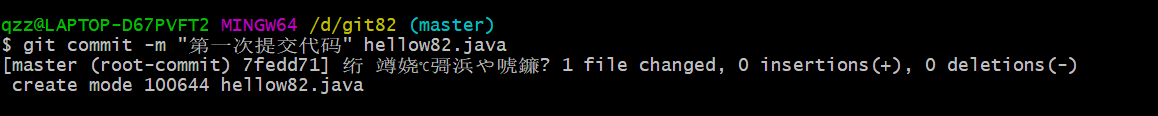
添加到暂存区 文件夹为空时，添加不了



需强制添加 Git add -f 文件名/文件夹



提交到本地仓库

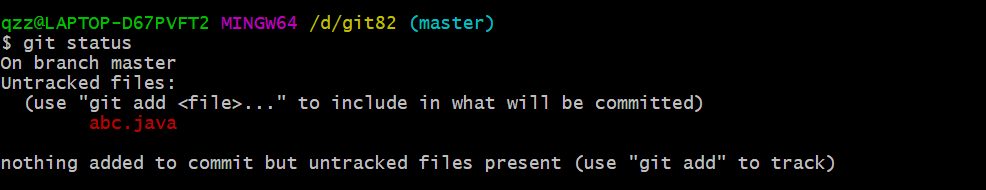


如报错，则执行以下代码：

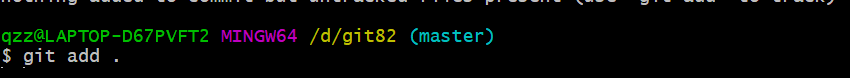
指定git客户端用户名：git config --global user.name "自定义名"

指定git客户端的邮箱（账号）：git config –global user.email "邮箱账号"

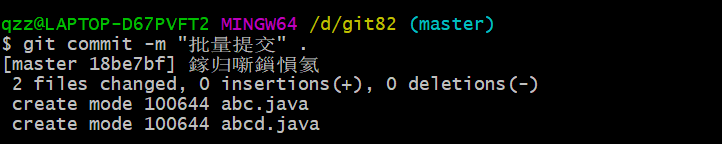
判断提交还是未提交(爆红是未提交的) .(当前文件夹下的所有文件)或者\*都可以



未提交的所有文件 提交到暂存区去 .(当前文件夹下的所有文件)或者\*都可以



把暂存区中的所有文件提交到本地仓库

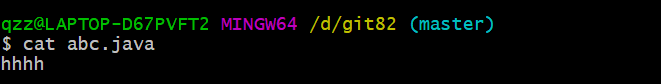


编辑文件

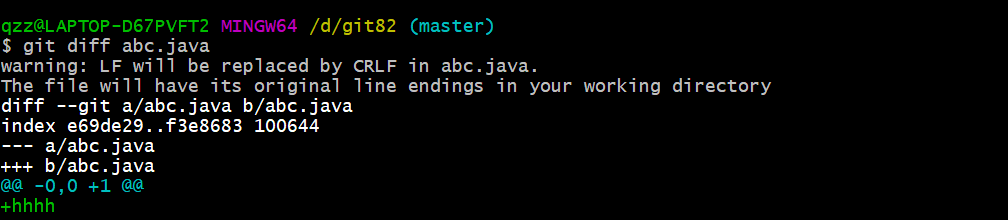
i切换编辑模式 esc退出编辑模式 shift+: wq保存并退出



查看文件内的内容



查看对文件的更改 git diff \* 查看所有文件的修改

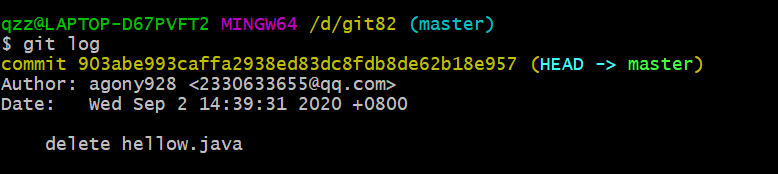


9.2

强制递归删除 git rm -f 文件名



查看详情/简略(查看所有的版本号信息)





恢复

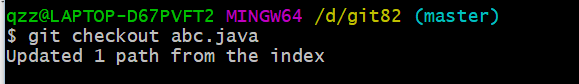
1. 回退到上一个版本(一个^就是回退一个版本，^^就是回退两个)

Git reset head^^

1. 根据版本号回退版本

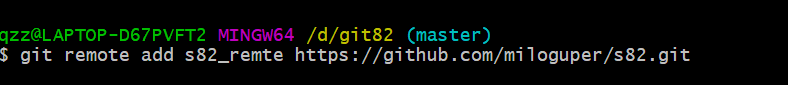
git reset --hard 版本号(eb35a67)

放弃工作区的修改，从暂存区拷贝一份最新的 (放弃本次修改的内容)

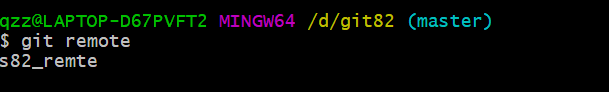


远程仓库：

在本地仓库建立远程仓库 s82\_remte自定义名



查看本地有哪些远程仓库



上传到远程仓库

*更新本地仓库的文件：*

git pull 仓库名 master --allow-unrelated-histories

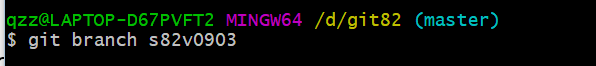
*上传本地仓库的文件：*

git push 仓库名 master

9.3

分支：基于主干或者某一个版本而创立的一个独立的版本

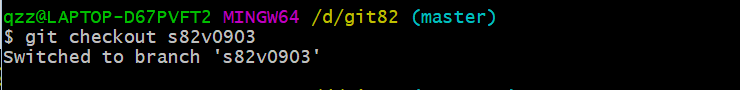
创建分支



查看分支



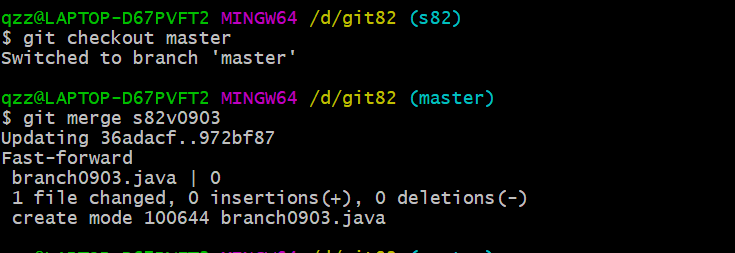
从主干切换到分支

****

从分支提交到远程仓库



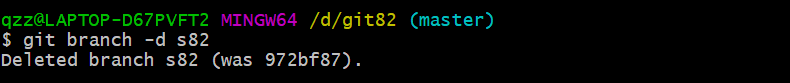
合并分支到master主干



强制提交(轻易不用)

git push -f 仓库名 master

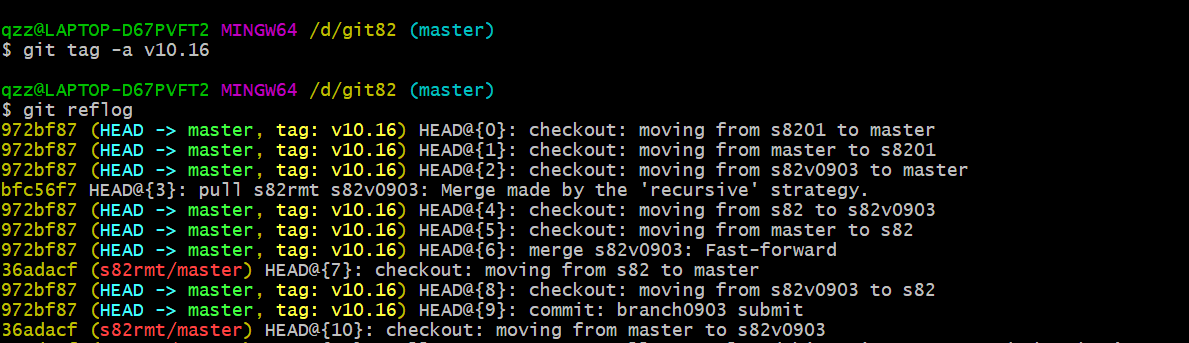
删除分支



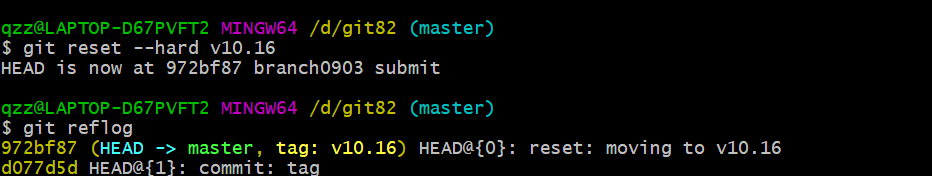
创建并且切换到分支



创建标签



给新文件添加标签



克隆clone：从远程仓库克隆

