**GIT学习**

**版本控制**：

SVM：集中式版本控制系统

GIT：分布式版本控制系统，git可以直接看到更新了哪些代码和文件

**环境配置：略** 环境变量只是为了在任何地方都能使用，默认已经配置完成

**基本操作：**

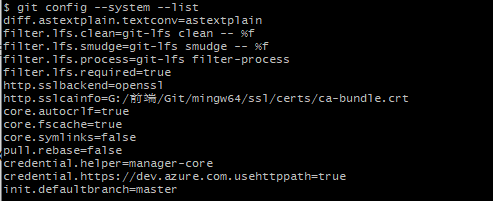
基本命令和linux一样 touch xx.xx 建文件（mkdir建文件夹） reset重新初始化终端

history查看命令历史

**GIT配置：**

查看用户配置git config global --list 用户名、邮箱

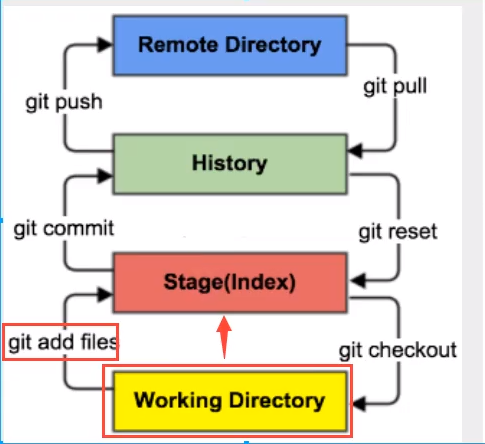
查看系统配置：

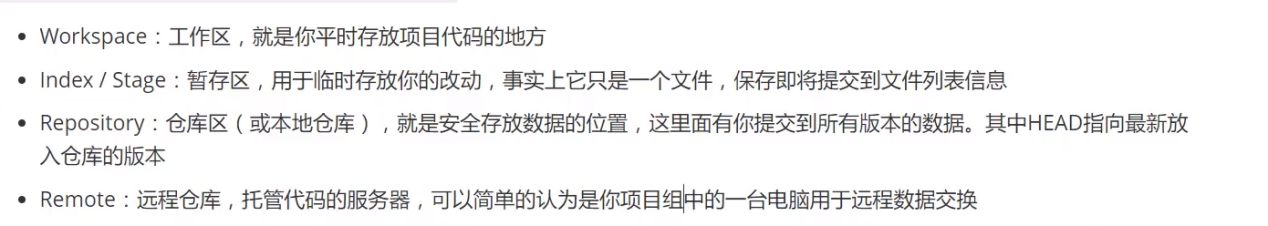


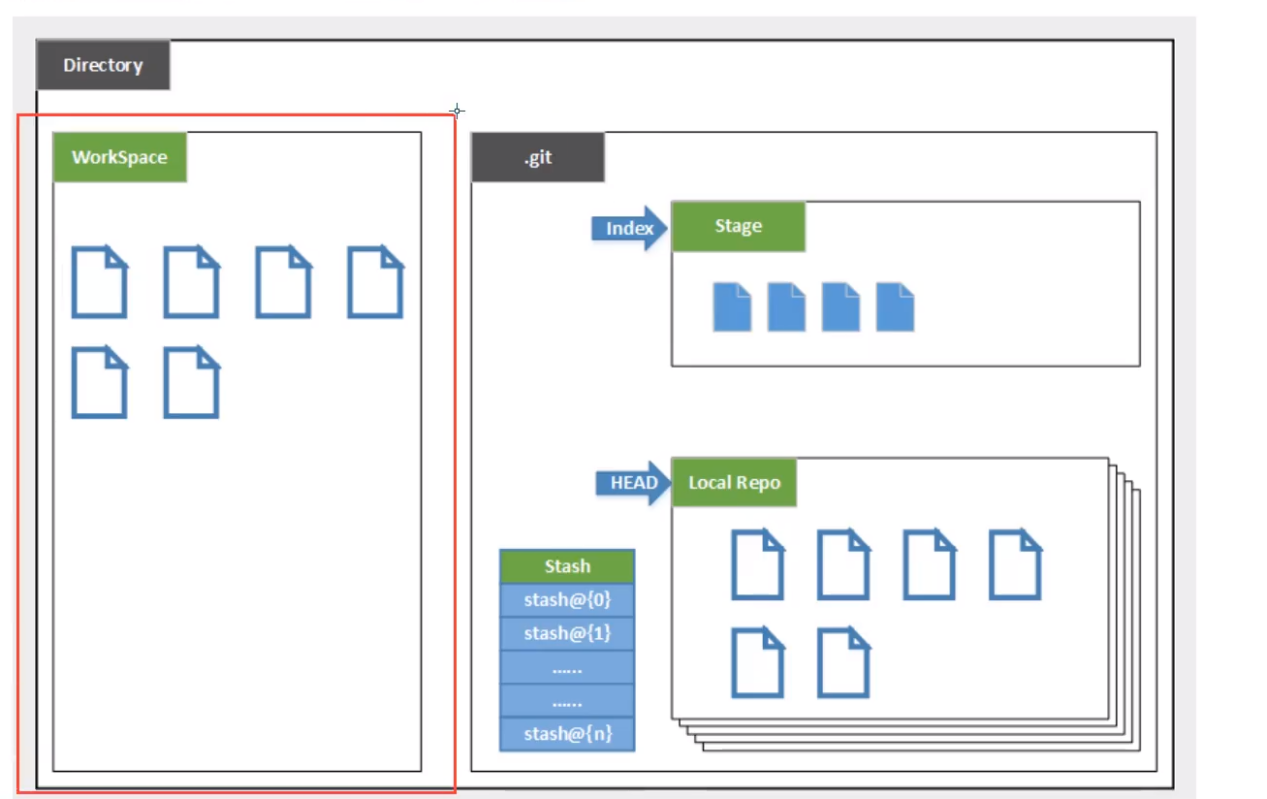
本质上都是在本地上自己所配置的文件

**GIT基本理论**

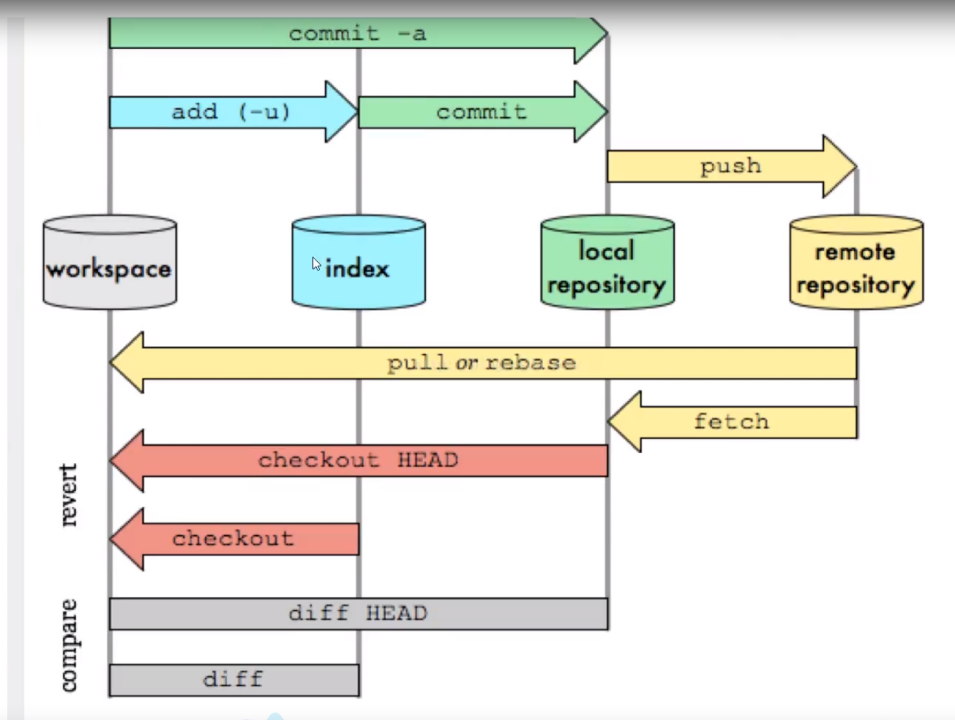
本地的三个工作目录：工作目录、暂存区、资源库、 远程git仓库



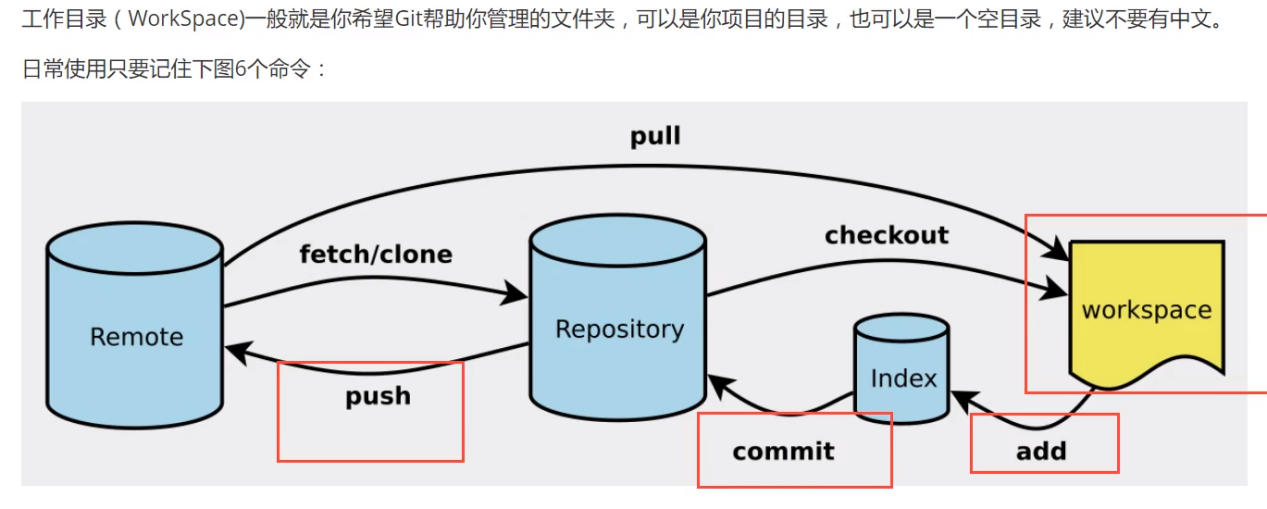






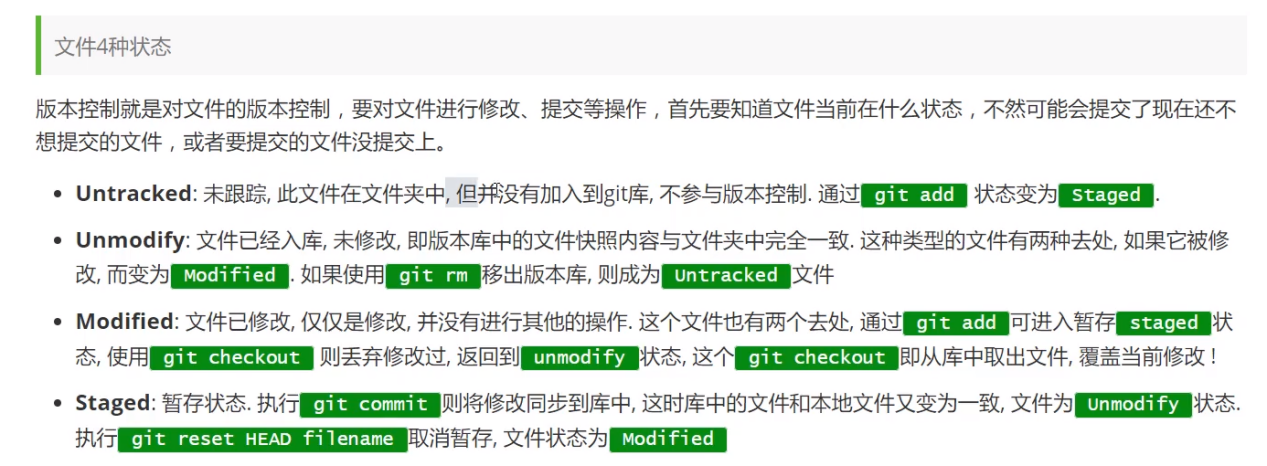


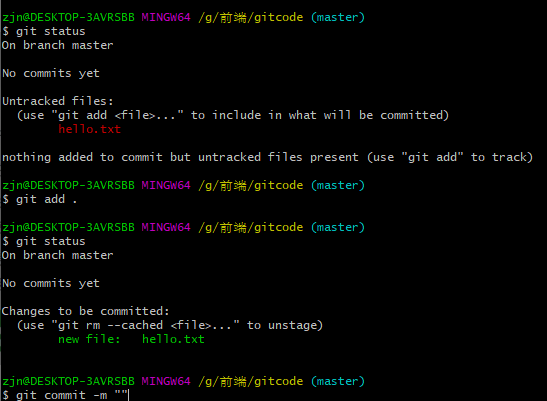
**GIT项目搭建**



1. 新建本地仓库Git init 初始化本地
2. 克隆远程目录：git clone [url] 远程仓库的文件就可以下载下来了

***GIT的文件操作***





$ git add .提交到暂存区，add可以自动判断目录下的文件是否有改动以及新增

$ git status查看工作目录和暂存区的状态，是否能追踪到

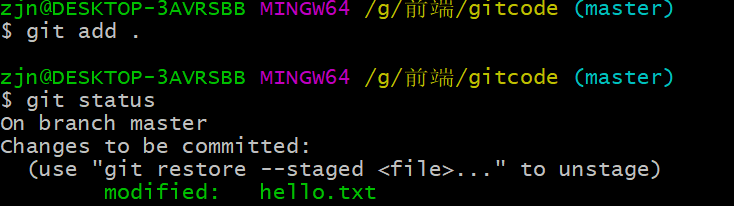
$ git commit -m "" 将暂存区中的文件提交到本地仓库

修改文件以及文件提交

$ git status 检测到有文件未被追踪

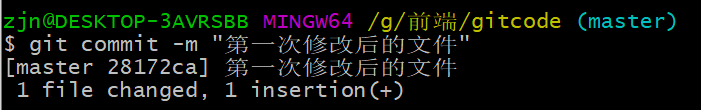


$ git add . 提交到暂存区

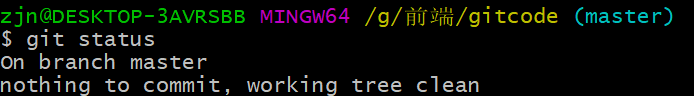


在提交到暂存区之后，再次git status查看工作状态可以发现工作树上已经存在了一个被追踪的文件(变绿)

$ git commit -m "第一次修改后的文件" 提交到本地仓库并备注信息 -m(message)。



$ git status 再次查询文件状态



工作树已经clean了。

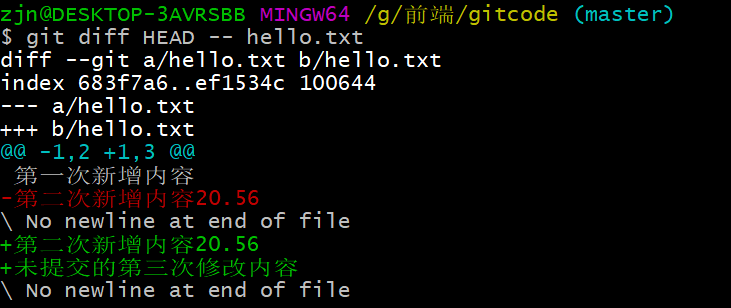
$ git log



提交文件必须要先放入暂存区

与未提交的文件与提交至版本库的文件进行比较进行比较：

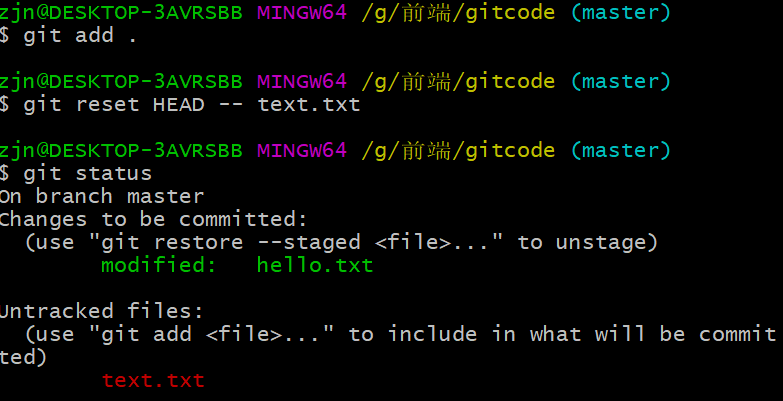
$ git diff HEAD -- hello.txt





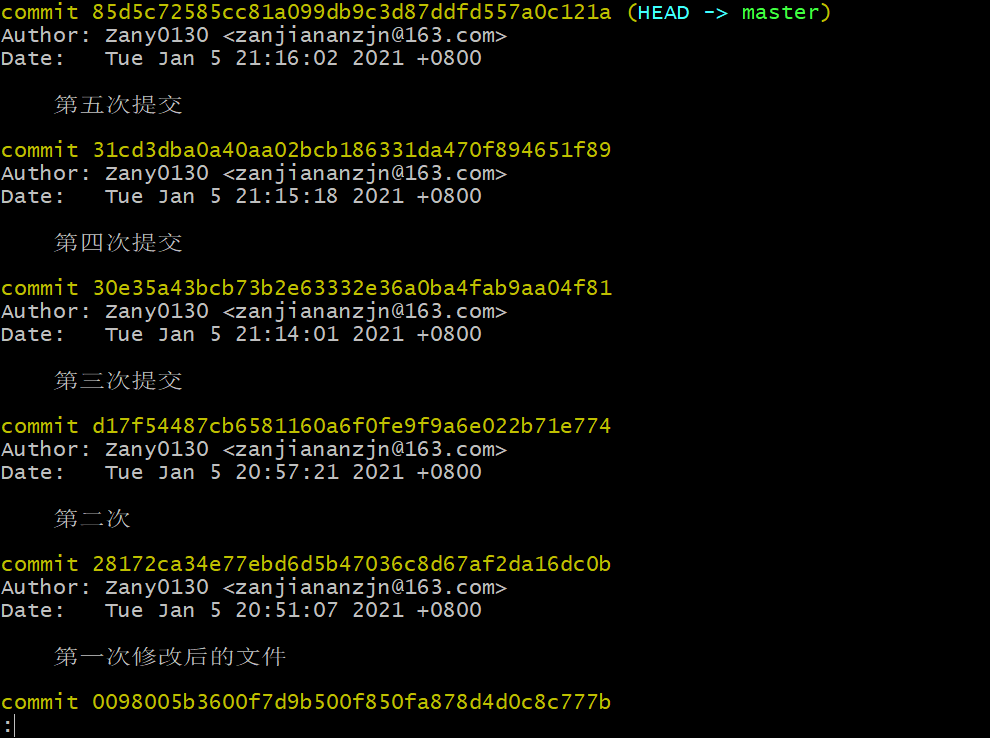


$ git reset HEAD -- text.txt 可以发现从暂存区中取回的文件再次变成红色



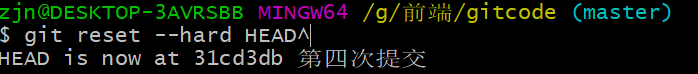
**版本回退**

**五次版本**



回退到上一个版本

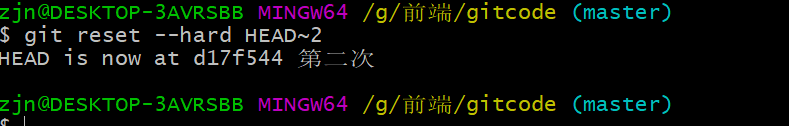
$ git reset --hard HEAD^





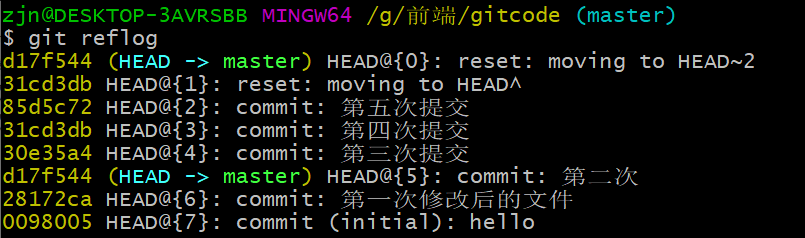
第五次修改内容不见了

$ git reset --hard HEAD~2 往前回两个版本



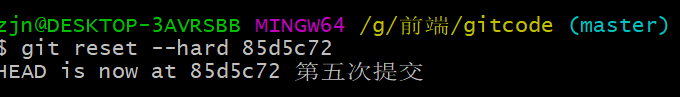


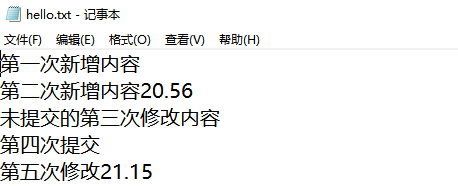
$ git reflog 查看所有字符串



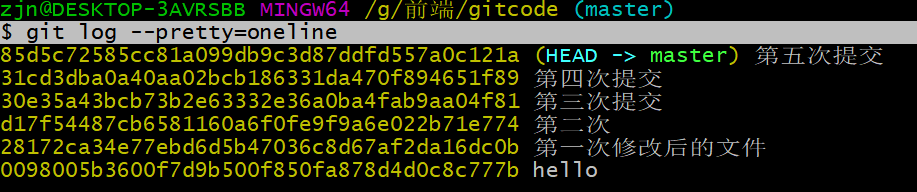
跳转至任意版本

$ git reset --hard 85d5c72





$ git log --pretty=oneline 单行显示所有的版本



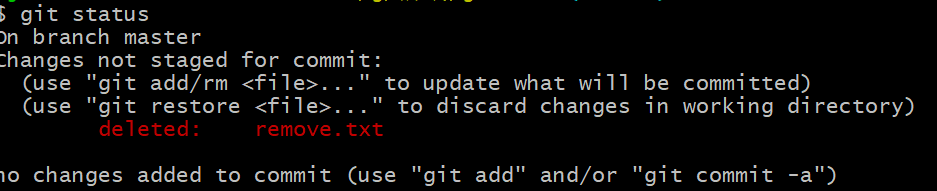
**文件删除**

$ git add remove.txt

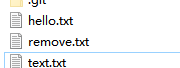
$ git commit -m "删除操作"

**模拟工作区误删remove.txt文件**

$ git status 发现出现了删除操作



$ git restore remove.txt 文件又回来了



**或者使用**$ git checkout remove.txt **去版本库拉去文件**

**完全从版本库里面删除：**

现在工作区删掉后(也算一次修改)，再次在git中提交一遍

$ git add remove.txt $ git commit -m "删除操作"

**或使用**

$ git rm remove.txt版本库与工作区中都进行了删除 但是使用版本回退依旧可以拿到之前的我文件

**使用远程仓库**

*/或者自己搭建gitlab/*

设置本机绑定SSH公钥(对应本地的电脑)，实现免密码登录具体操作见git使用说明。

**远程仓库下载**

$ git clone <https://github.com/Zany0130/DOM-Scripting.git>

$ git clone url

Push到自己的远程仓库

$ git ls-file 查看本地仓库内的文件



**初始化 添加说明 提交到本地仓库 添加远程仓库并且取别名”origin”(和远程绑定联系)**

**另一种步骤：**

**1 github上新建仓库**

**2在本地需要提交的代码的文件夹下右键用gitbash打开**

**3首先git clone url 将本地仓库和远程仓库绑定，以及获取远程仓库的内容，避免提交时发生冲突(一般新建的远程仓库仅有LICENSE以及README.md)。**

**4相同步骤，首先使用git add .将所有内容提交到缓存区**

**5然后再git commit -m“提交说明(别瞎写，会显示在很醒目的位置)” 提交到本地仓库**



**6接着**$ git push -u origin main

**将本地仓的文件push到远程仓库(github)。**

**忽略文件**  
