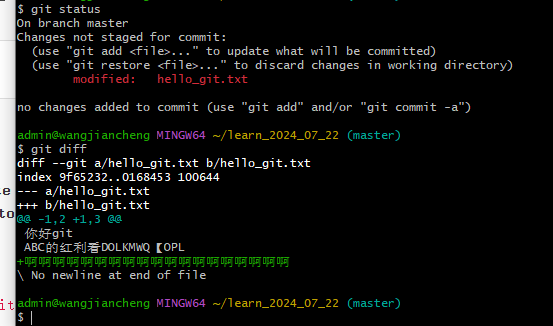
Git

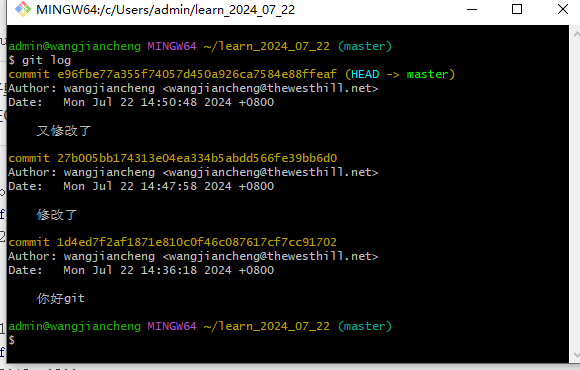
创建文件 **mkdir** --->切换目录 **cd--**->显示当前目录 **pwd** ---> 把这个目录变成Git可以管理的仓库 **git init** ---> 文件添加到仓库 **git add --->** 文件提交到仓库 **git commit**

**git status命令可以让我们时刻掌握仓库当前的状态，上面的命令输出告诉我们，readme.txt被修改过了，但还没有准备提交的修改。**

**git diff：查看不同**



git log命令显示从最近到最远的提交日志

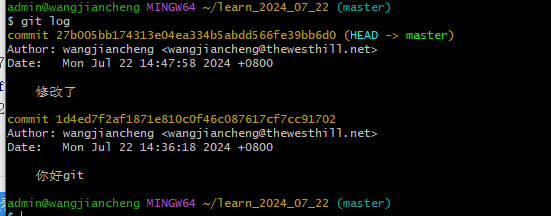
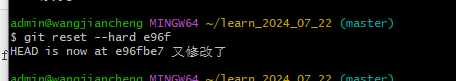


在Git中，用HEAD表示当前版本，上一个版本就是HEAD^，上上一个版本就是HEAD^^，往上100个版本写HEAD~100

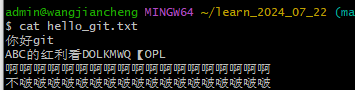
回退使用git reset命令



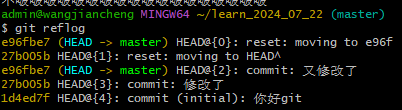
此时，再使用git log，最新的那个版本已经没有了，不过，在命令行没有关闭之前，可翻找上面的内容，查看到其id，再使用git reset 指定回到未来的某个版本（id写几位就好）

cat查看文件



git reflog用来记录你的每一次命令



git add命令实际上就是把要提交的所有修改放到暂存区（Stage），然后，执行git commit就可以一次性把暂存区的所有修改提交到分支。

Git跟踪并管理的是修改，而非文件！

git diff HEAD --命令可以查看工作区和版本库里面最新版本的区别

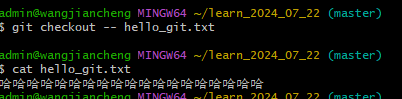
git checkout -- file 可撤销修修改

命令git checkout -- readme.txt意思就是，把readme.txt文件在工作区的修改全部撤销，这里有两种情况：

一种是readme.txt自修改后还没有被放到暂存区，现在，撤销修改就回到和版本库一模一样的状态；

一种是readme.txt已经添加到暂存区后，又作了修改，现在，撤销修改就回到添加到暂存区后的状态。

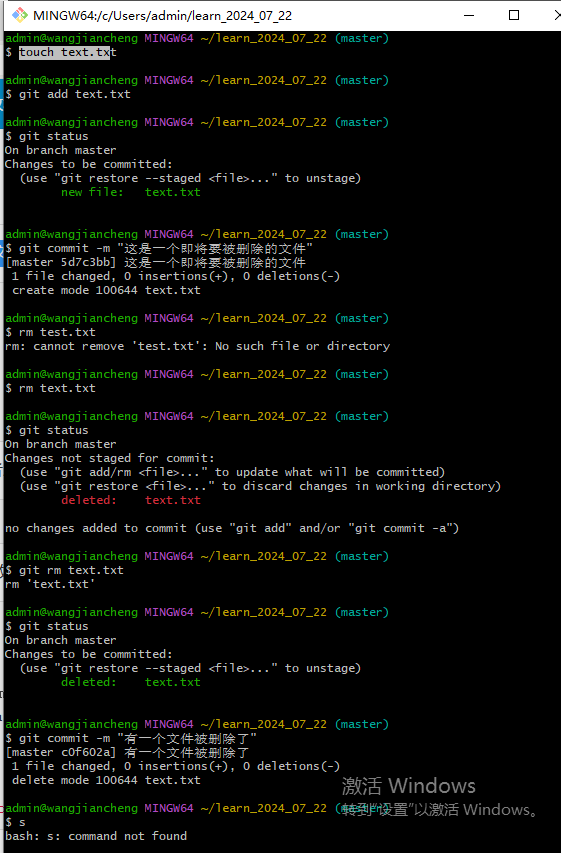
总之，就是让这个文件回到最近一次git commit或git add时的状态。

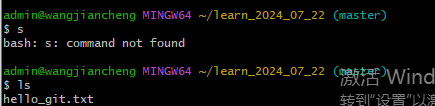


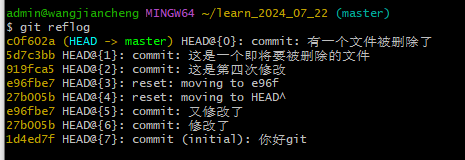
撤销修改后返回到了未修改前

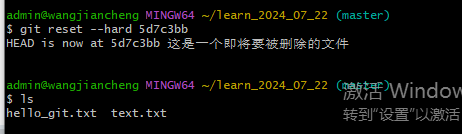
命令git reset HEAD <file>可以把暂存区的修改撤销掉（unstage），重新放回工作区：

rm删除：





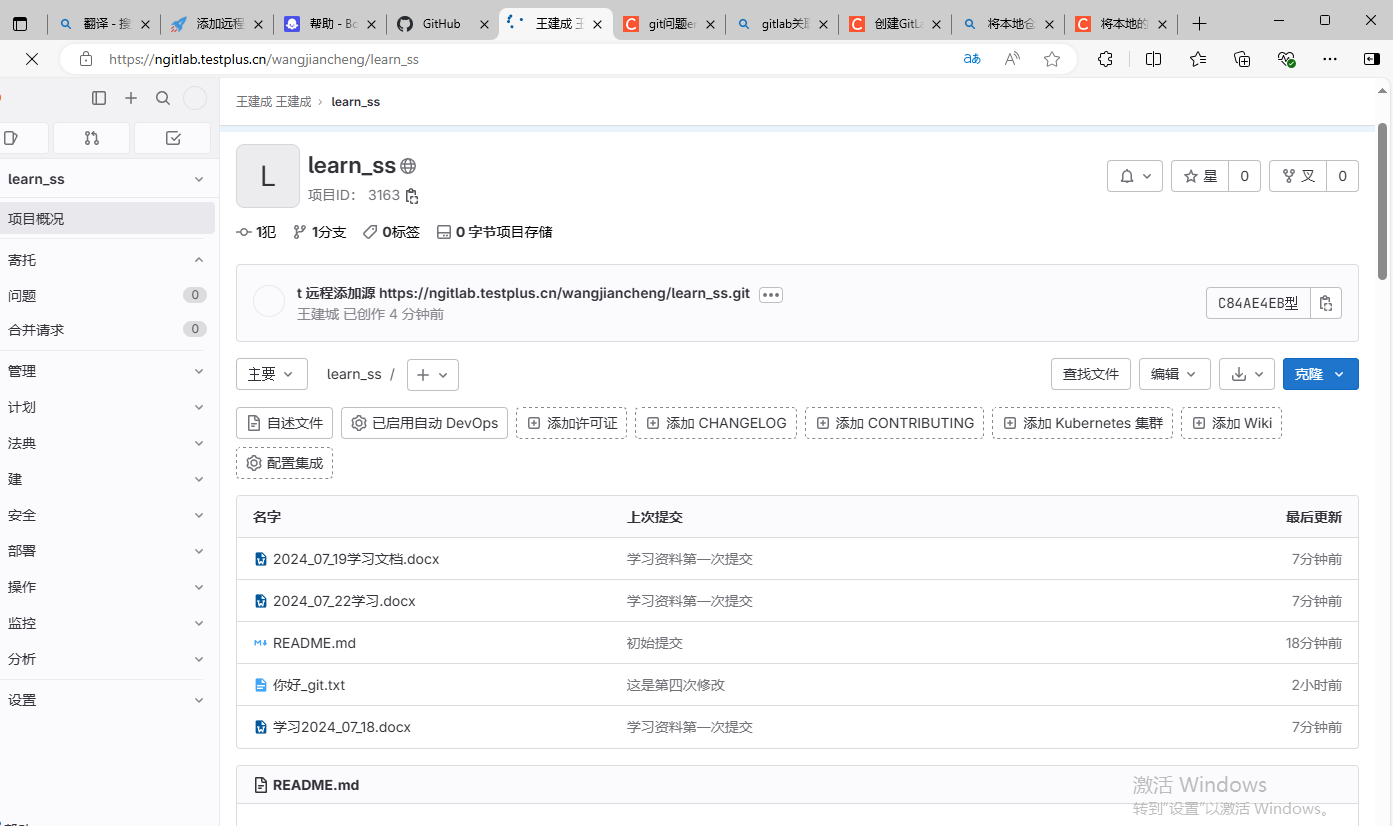
Git reflog + git reset 可回退已经被删除的文件  




**远程仓库**

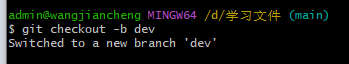
ssh密码：$ ssh-keygen -t rsa -C "youremail@example.com"

之后在，C:\Users\admin\.ssh中可以找到，id\_rsa.pub为公钥，另一个为私钥



**分支**

$ git checkout -b dev 创建dev分支



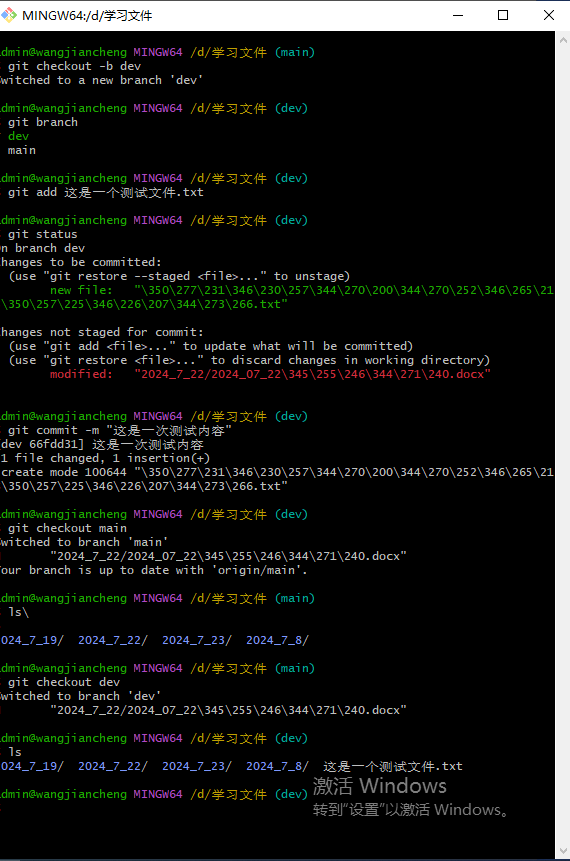
git checkout命令加上-b参数表示创建并切换，相当于以下两条命令：

$ git branch dev

$ git checkout dev

用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。



切换分支后添加的文件在主分支里不可见



合并后，在主分支里可见了

$ git merge dev 把dev分支合并到当前分支上