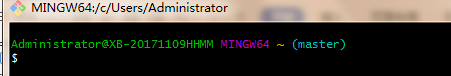
**一、git的设置**

——进入git界面（选择git bash）：进入时，显示默认进入home目录

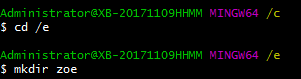
选择git bash的原因：可以使用linux命令来进行git的操作



——进入E盘，并在E盘创建一个zoe目录做练习

Cd /e

Mkdir zoe

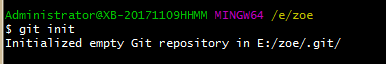


——进入zoe目录，并在zoe目录下创建版本库，创建好的版本库会在zoe目录下产生一个隐藏文件夹，此文件夹即版本库；

Cd zoe

Git init

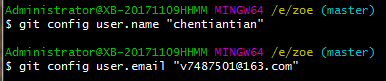




——进行用户设置（没有弹出任何提示则为设置成功）

Git config user.name “chentiantian”

Git config user.email “v7487501@163.com”



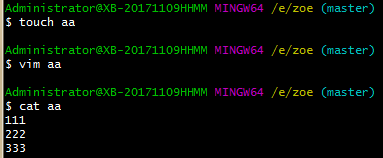
**二、git的使用**

——创建一个文件aa，并编辑，用于练习使用git文件版本管理

Touch aa

Vim aa

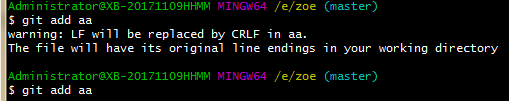
Cat aa



（注：用cat命令查看我输入了多少行数据，共三行）

——开始添加至缓存区

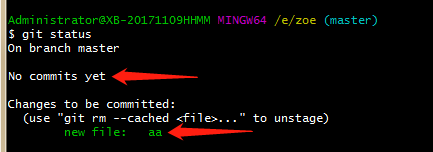
Git add aa



（注：第一次添加报错，是提示bash命名和cmd命令有冲突，但并不妨碍我们进行添加，此时我们再提交一次git add命令即可）

——用git stauts 查看aa文件的状态

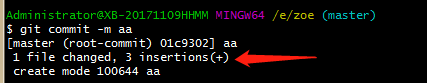
Git status



（注：可以看到提示，有新文件（new files）添加了，但是没有提交（not commit yet）)

——把aa文件进行提交至我们创建的版本库

Git commit -m aa



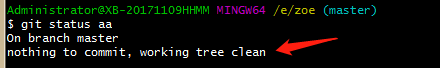
（注：提示我们提交成功，1个文件已改变，新增三行数据，此时aa文件已提交到版本库）

（注2：git add命令仅把文件添加到缓存，但并未实际提交到版本库）

（注3：git commit命令会把在缓存区的所有文件提交到版本库）；

关系如图：

——提交完成后，我们回头看看aa文件的状态



（注：这里提示我们，本地文件夹和版本库的aa文件是一致的，不需要提交，工作树是干净的）

1. **git与github的连接**

Github介绍：https://github.com/



（注：github像一个程序代码图书馆，很多程序员把他们写的代码上传到github，公开并供别人下载，我们的操作就是连接好后，把自己写的文档上传至自己的github上）

——先进行注册在hub上的账户；



——注册完毕后登录；建立个人的存储库



——在我的存储库下建立一个新的存储库：



——填写资料，填完后，告诉我们创建成功：



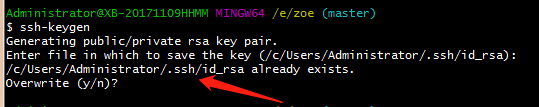
——返回我的存储库，看新建的存储的url



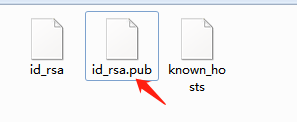
（注：这就是我们的存储库url（git@github.com:dayday001/-sir-.git），通过url我们可以让我们的git连接到我们的版本库）

——在git上获取密钥

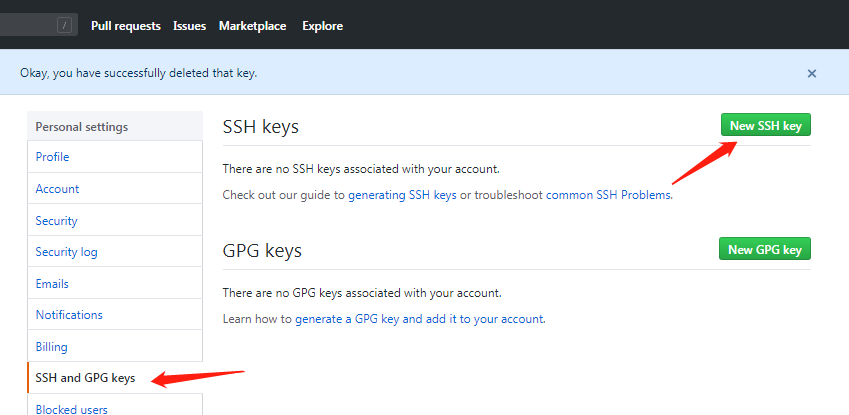
Ssh-keygen （输入后连按三次回车）（箭头指向为密钥在所在路径）



——去对应路径下找到密钥，用记事本打开



——进入github添加我们的密钥





以上git与github的连接就成功了

1. **git与github之间互传文件**

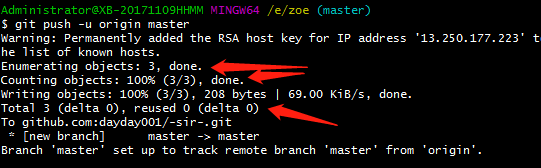
——将git与github上前面新创建的存储库连接（我的url：[git@github.com:dayday001/-sir-.git）](mailto:git@github.com:dayday001/-sir-.git）)

Git remote add origin [git@github.com:dayday001/-sir-.git](mailto:git@github.com:dayday001/-sir-.git)

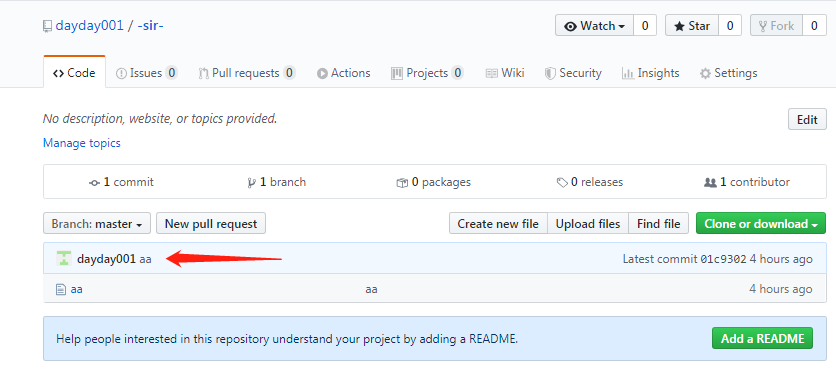


——将提交的aa文件上传至github

Git push -u origin master

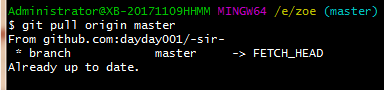


此时github上面已有我们上传的文件



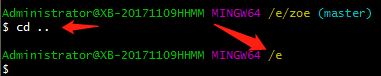
——拉取github上的文件

git pull origin master



——下载整个存储库（可以复制自己的，也可以复制别人的公开存储库，复制自己的其实和git pull命令差不多作用，在这里演示如何下载别人的存储库下来）

返回到E盘 cd ..



在github上随便找个别人的存储库

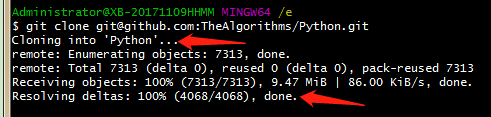
复制别人存储库url



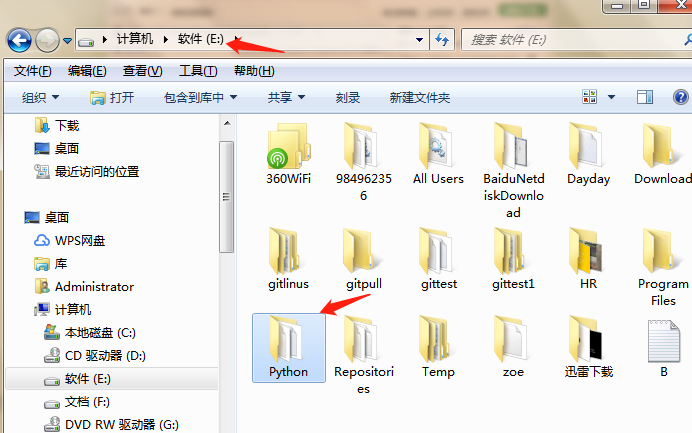


使用命令： git clone 别人的url

Git clone [git@github.com:TheAlgorithms/Python.git](mailto:git@github.com:TheAlgorithms/Python.git)



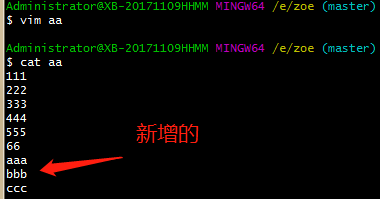
可以看到E盘已经下载到了我在github上随便找到的存储库文件夹



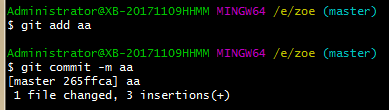
1. **版本回退与撤销：**

——版本回退（类似于word文档编辑后的撤销操作，但版本回退命令是回退commit后版本库）

首先我们对aa进行编辑，新增几行数据（aaa bbb ccc）

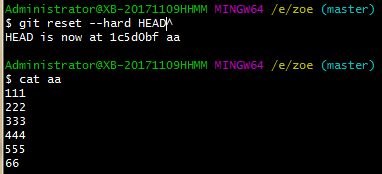


添加到缓存库，并提交至版本库



假设现在突然发现插的三行的写错了，需要变回原样，用版本回退命令：

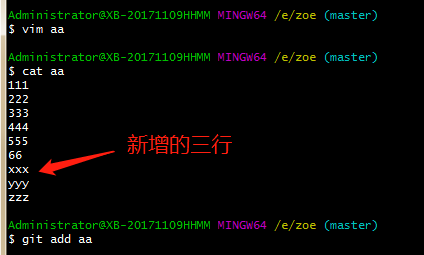
Git reset --hard head^



可以看到插入的三行已消失

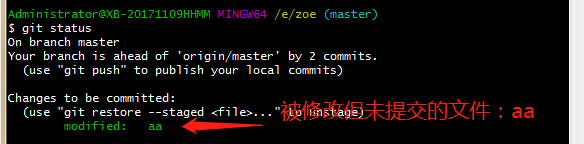
——撤销修改（和版本回退有所不同，撤销修改撤销的是在缓存区的修改后文件）

首先对aa文件编辑，插入 xxx yyy zzz三行，并添加到缓存区，但不提交



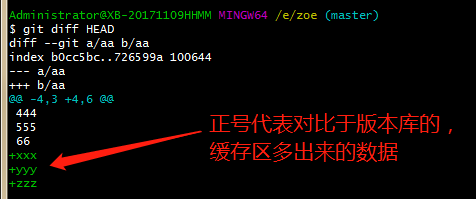
这个时候缓存区的aa和版本库的aa有差别（因为我没有commit） ，差了三行，我们可以用命令来查看

Git status



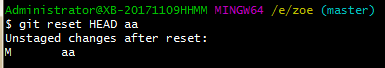
我们看下差异在哪里：

Git diff HEAD

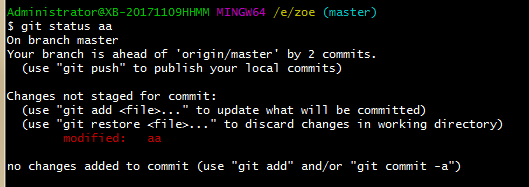


现在撤销掉xxx yyy zzz这三行：

Git reset HEAD aa



再看文件状态：git status aa



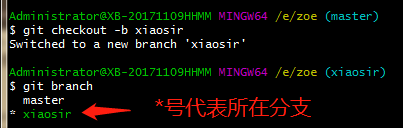
（添加后，modifed是绿色的，现在我们已经撤销修改，modified变成了红色，表示我们的本地库aa文件已经修改，但是没有添加到缓存）

1. **分支管理：**

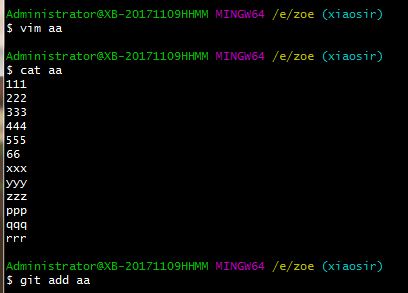
——查看当前分支：git branch

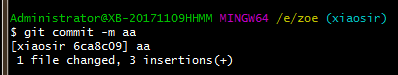


——创建并切换到一条分支：git checkout -b xiaosir



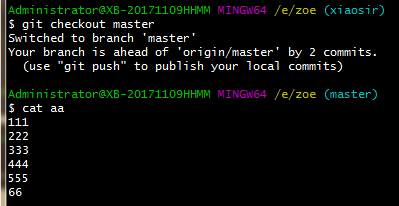
——我们在xiaosir分支下对aa文件进行修改（增加ppp qqq rrr），并添加到缓存区且提交





——我们在master分支下查看 aa文件，可以看到master没有刚刚添加的内容：

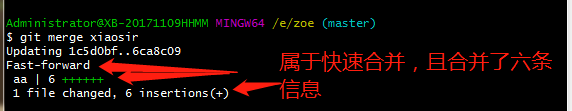
Git checkout master

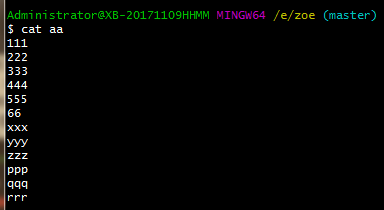


（注：这充分解释了：分支即工作线，即如果你在一条工作线进行文档修改编辑工作，那是不会影响到另一条工作线的，其中，一般以master为主分支）

——把xiaosir分支的修改结果合并到master分支上；

Git merge xiaosir （注：需要在master分支下进行操作）

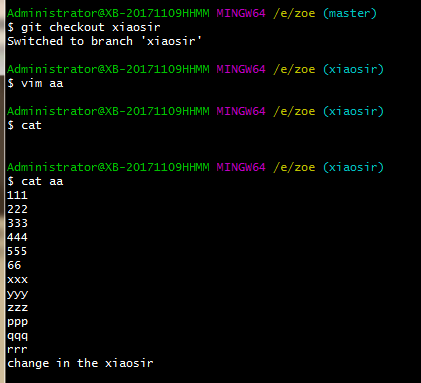




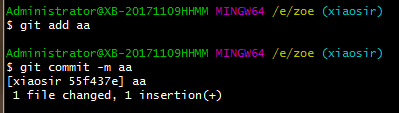
——合并分支过程产生的冲突：

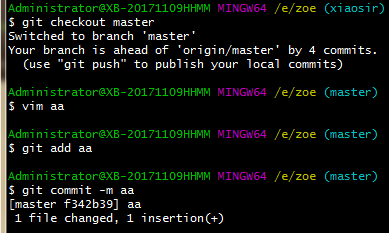
上面的步骤只是单独对一条分支进行修改并合并，但是在两条分支修改并合并的时候，会有冲突产生；

在xiaosir分支下修改编辑：git checkout xiaosir；



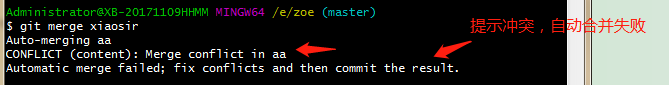
在xiaosir分支下添加并提交：



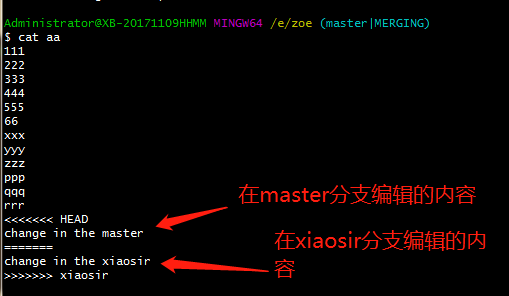
切换至master再进行修改并添加提交： 

此时，我们知道，我们在xiaosir分支增加了一行文字，master分支也增加一行文字（文字内容与xiaosir分支上的不同），也就是我们在不同分支上做了不同修改，现在我们对这个操作进行合并：

Git merge xiaosir （该命令在master分支下操作）

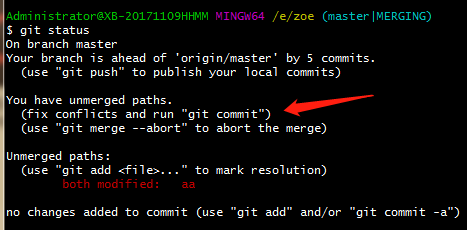


现在来看合并的结果：



可以看到，内容有合并，但是用<<<<<<< ========== >>>>>>>>>分隔符分开了；

Git status 命令也告诉我们冲突了



——合并冲突修复：

Git add aa

Git commit -m “conflict fixed”

