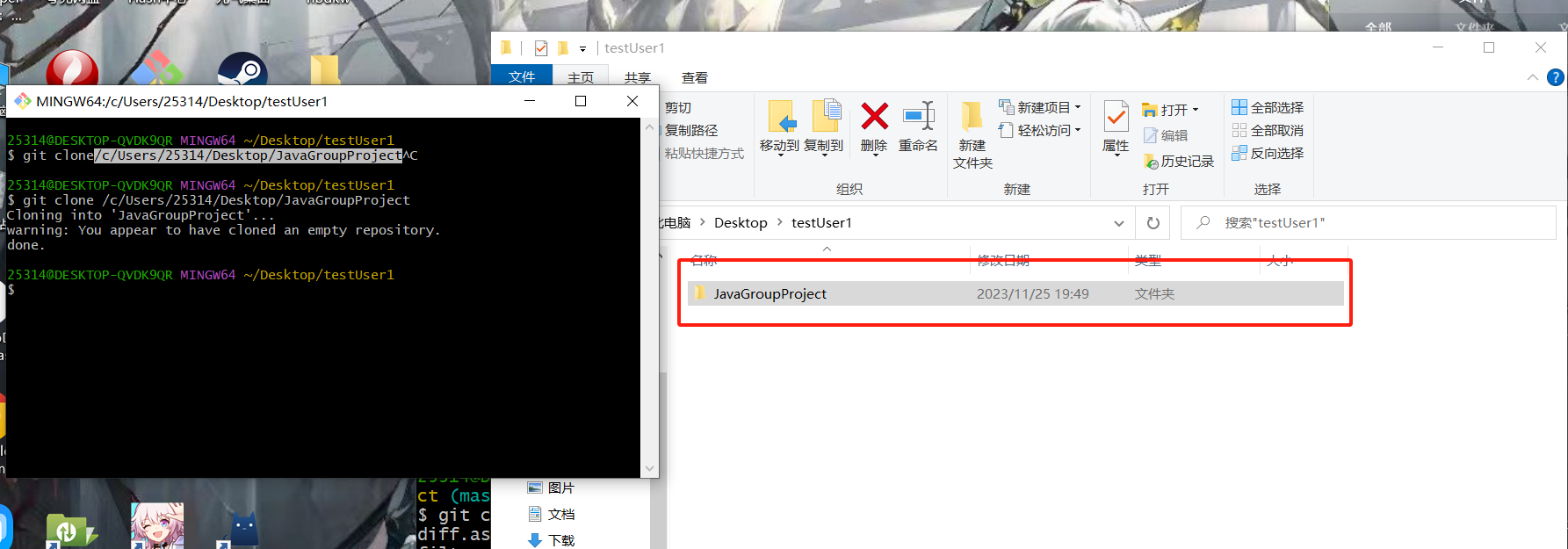
## 第一步，克隆项目

代码：

git clone <仓库地址>

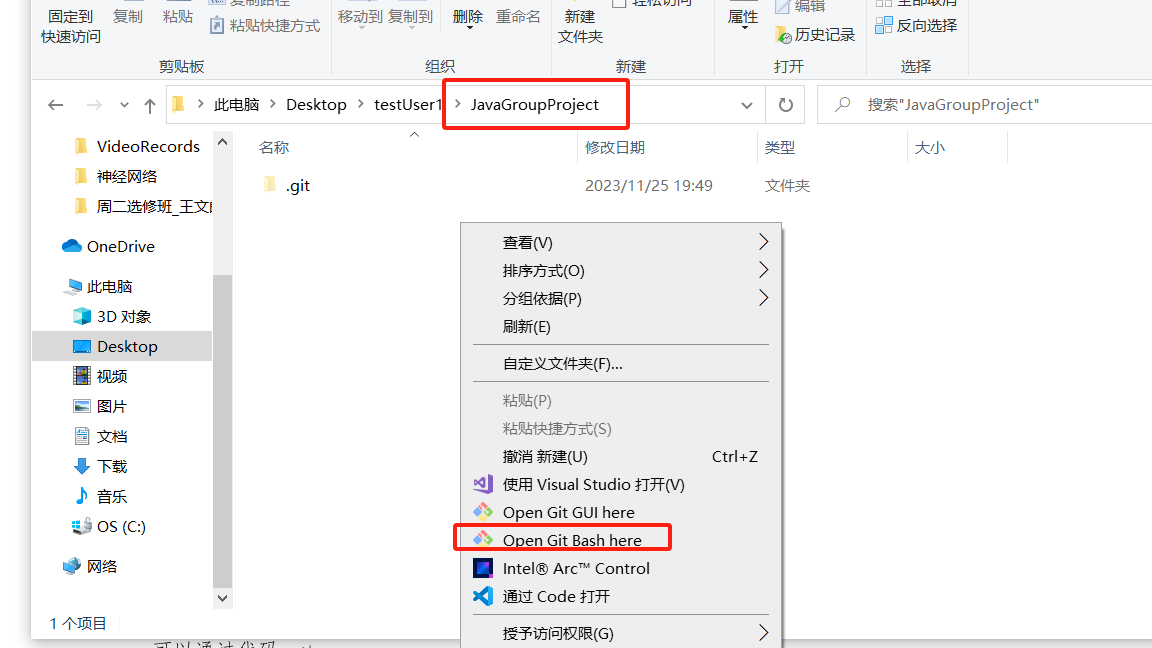
假如现在有一项目名：JavaGroupProject

运行图片:



## 第二部，进入project，添加姓名与邮箱

直接进入项目目录中，然后右键运行git bash:



代码：

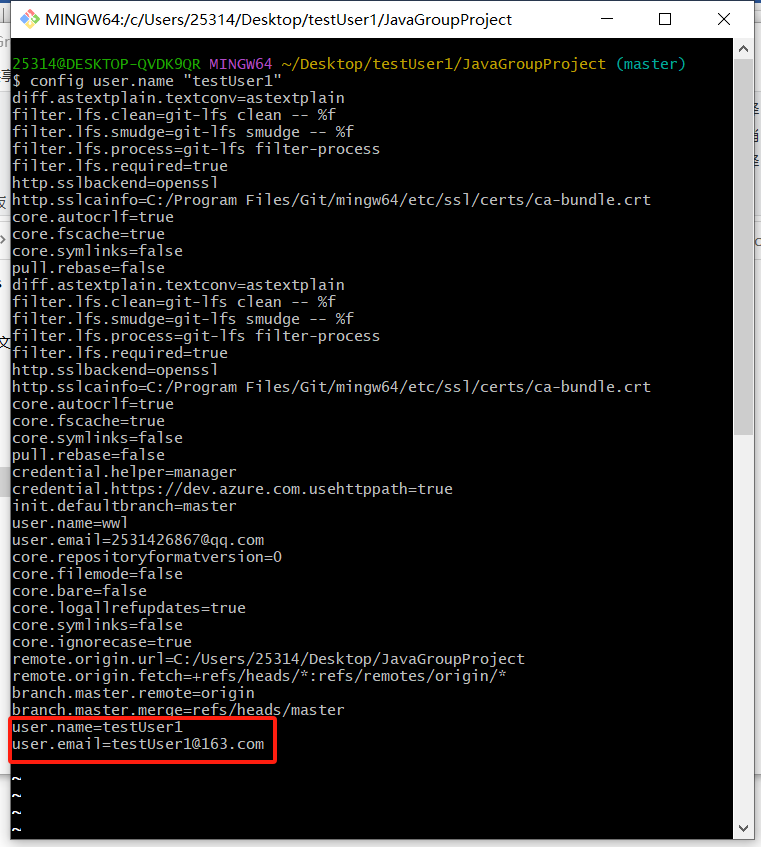
git config user.name ""

git config user.email ""

可以通过代码：

git config -l

来查看用户名与邮箱是否被添加（按q可以退出）:



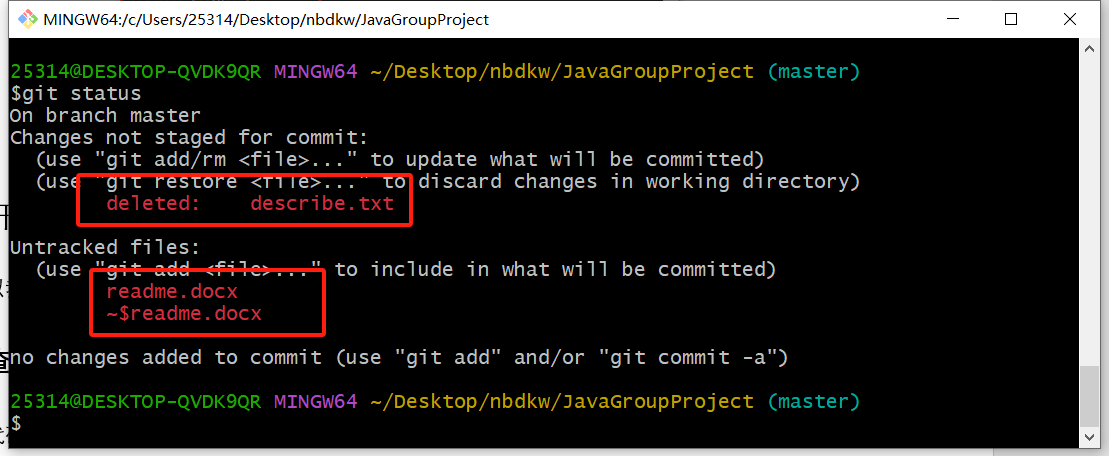
## 第三步，开始更改或是继续开发项目

以我现在为例，我刚刚删除了名为describe.txt的文档，且现在正在编写名为readme.docx文档为例子：

### 查看更改状态

代码：

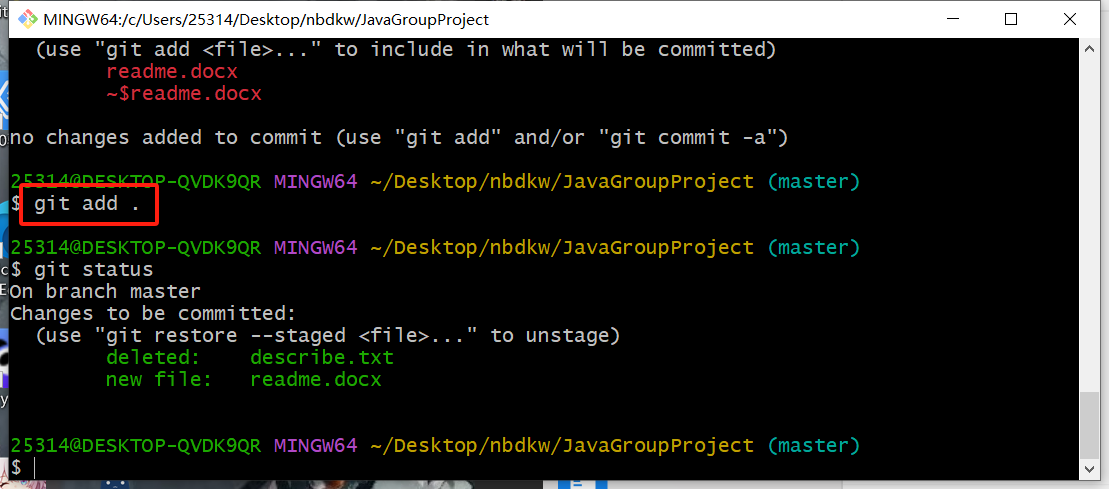
git status



### 添加要更新的代码

代码：

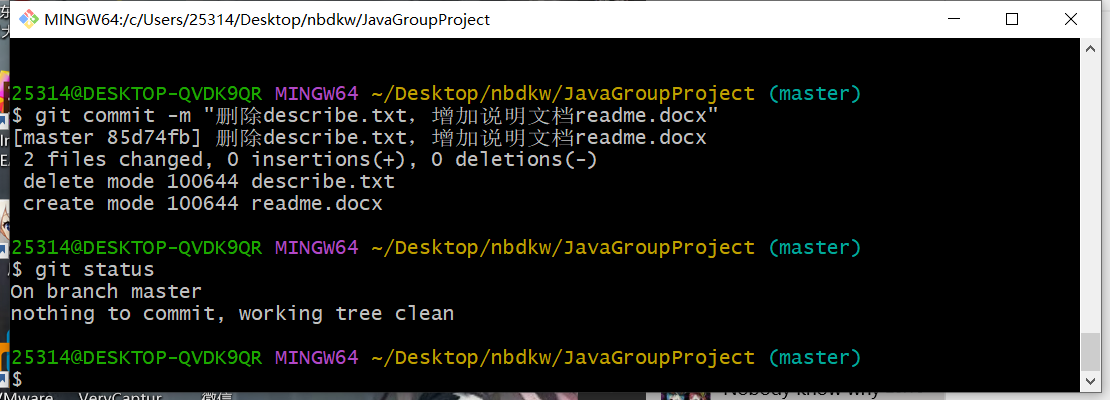
git add .



### 确认提交项目的更改并提交注释到本地的版本库（注释是必须的，没有注释无法提交）

代码：

git commit -m "注释内容"

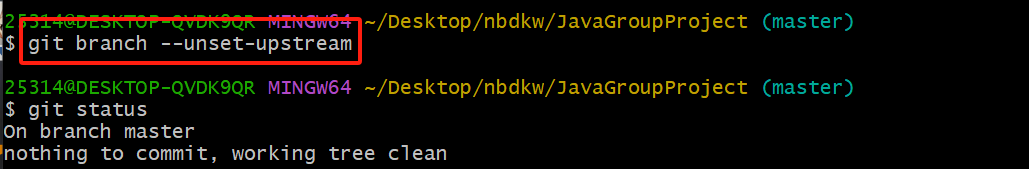


若此时运行命令git status时遇到情况：

Your branch is based on 'origin/master', but the upstream is gone.

(use "git branch --unset-upstream" to fixup)

则运行命令git branch --unset-upstream即可



### 将本地的版本库提交到远程的共享版本库当中完成最后阶段提交

代码：

git push

若出现错误：

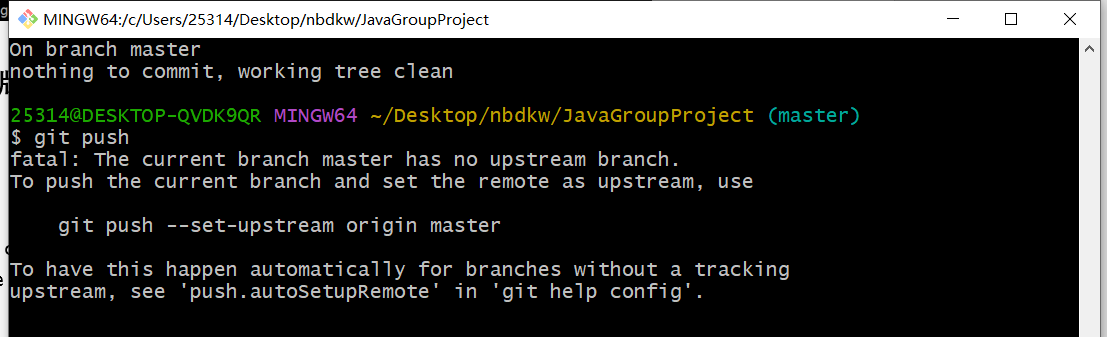
fatal: The current branch master has no upstream branch.

To push the current branch and set the remote as upstream, use

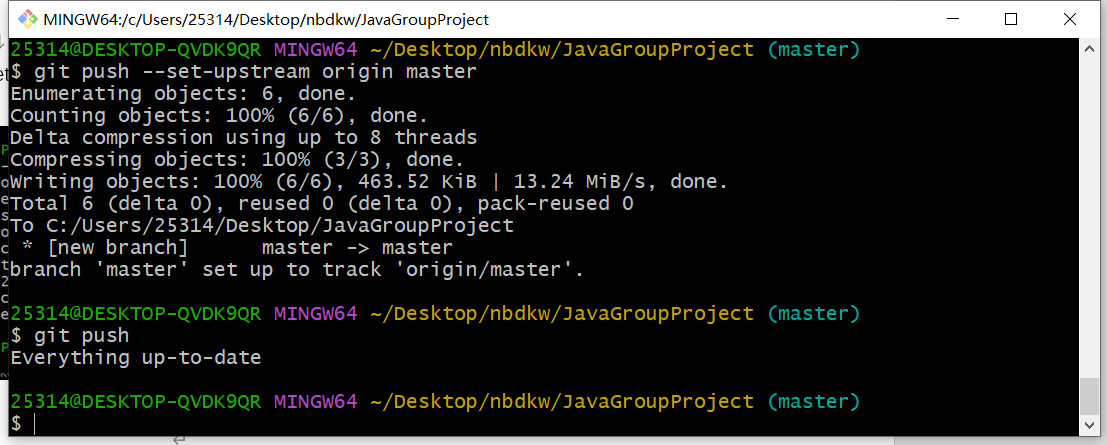
git push --set-upstream origin master

To have this happen automatically for branches without a tracking

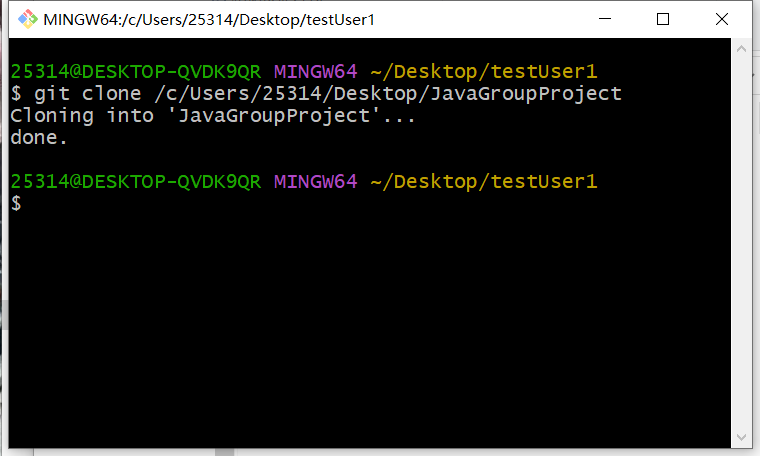
upstream, see 'push.autoSetupRemote' in 'git help config'.

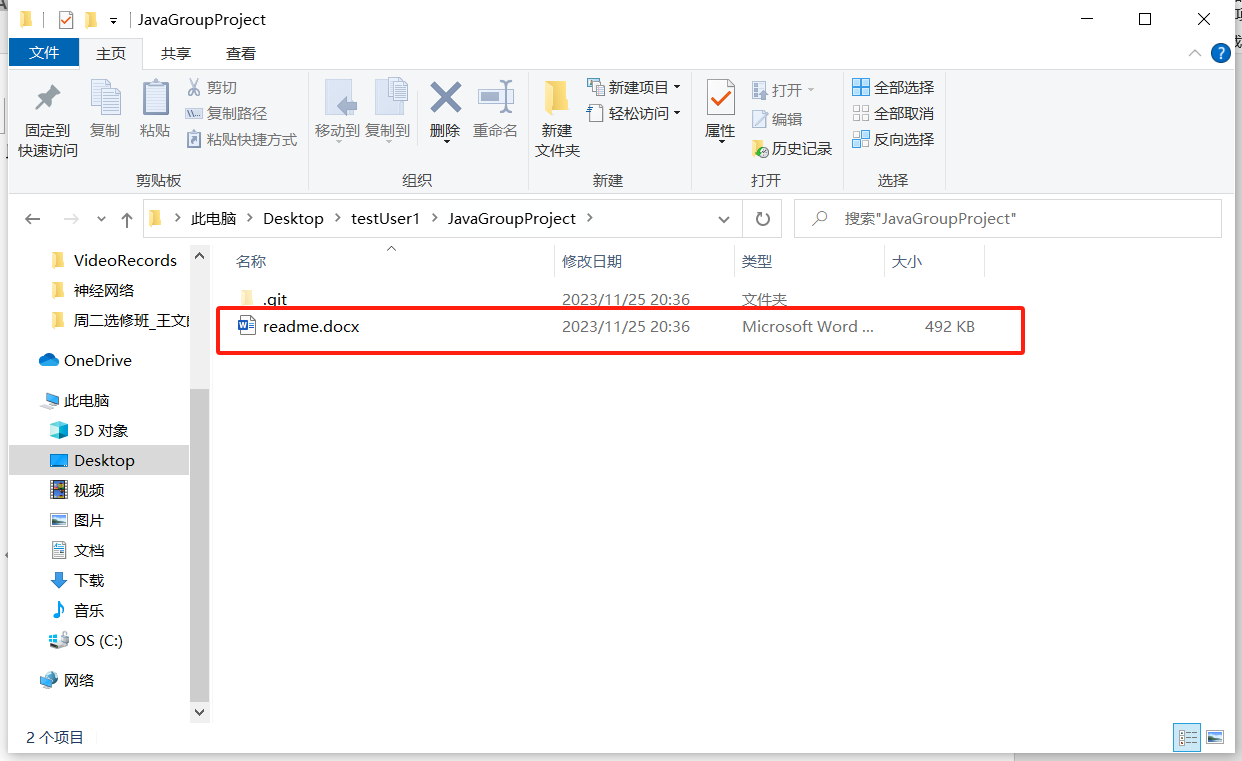


则运行代码:  
git push --set-upstream origin master然后再运行git push



此时在其他人的文件夹再次下载文件时，就会下载更新后的文件：

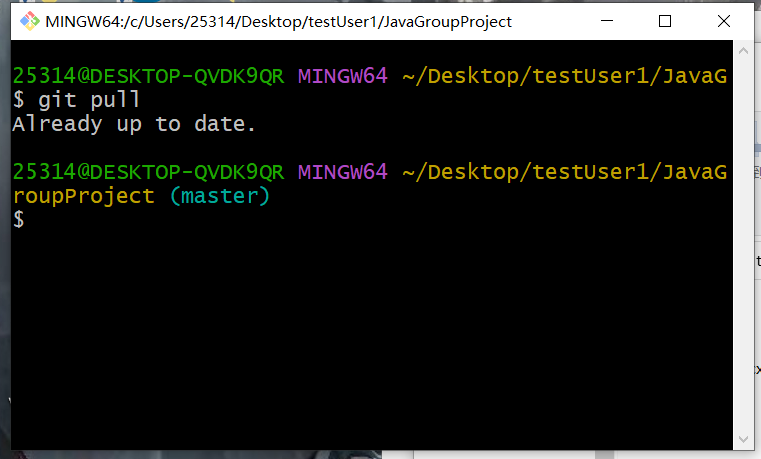




## 更新别人修改好的项目（之前已经有这个项目了）

代码：

git pull



## 关于查看，创建新分支（分支文件一般用来测试代码）

代码：

git branch（查看分支）

git branch dev1（创建名为dev1的分支）

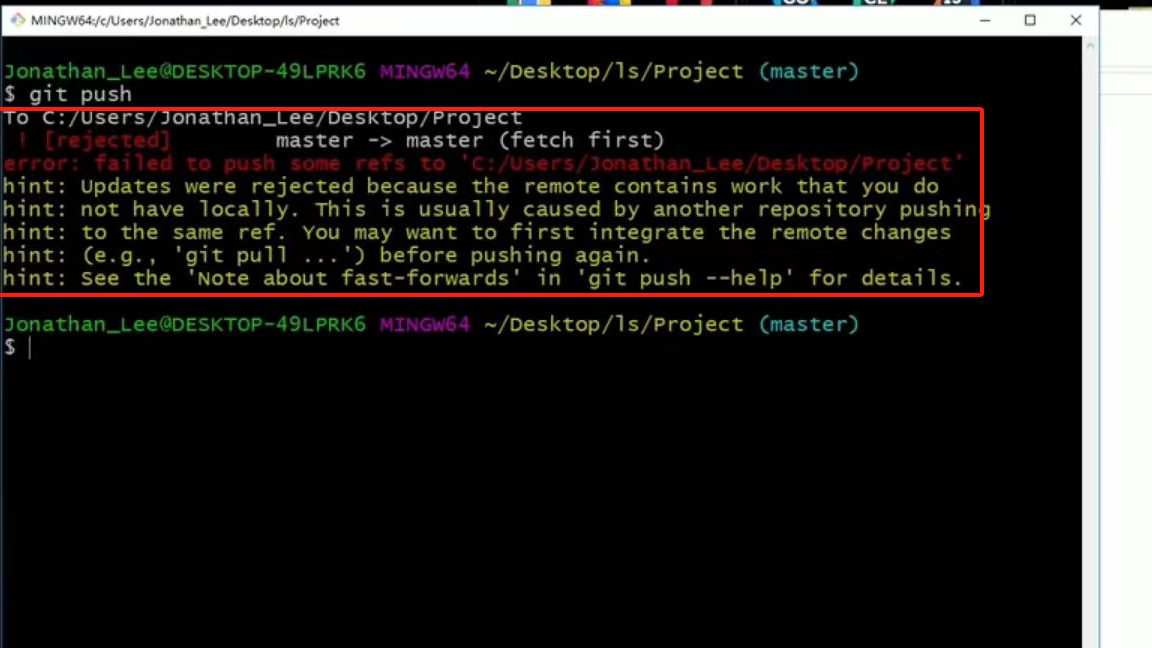
git switch dev1（进入dev1分支）

后续操作一致

## 常见问题

### 关于git pull命令，必须在进行代码撰写前运行一次，确保拿到的时最新版本的代码

### git push 命令错误：



原因：

由于其他用户已经执行了git push命令对数据进行了更新，因此你的数据版本过低了，需要重新更新：git pull然后再进行git push提交

### 若同一文件中同一函数进行了修改

无需担心，git一般会直接添加不同的函数进入代码当中，若两个人提交的代码发生冲突，git会在冲突的代码段当中给出对应的冲突信息