目录占位

提出概念

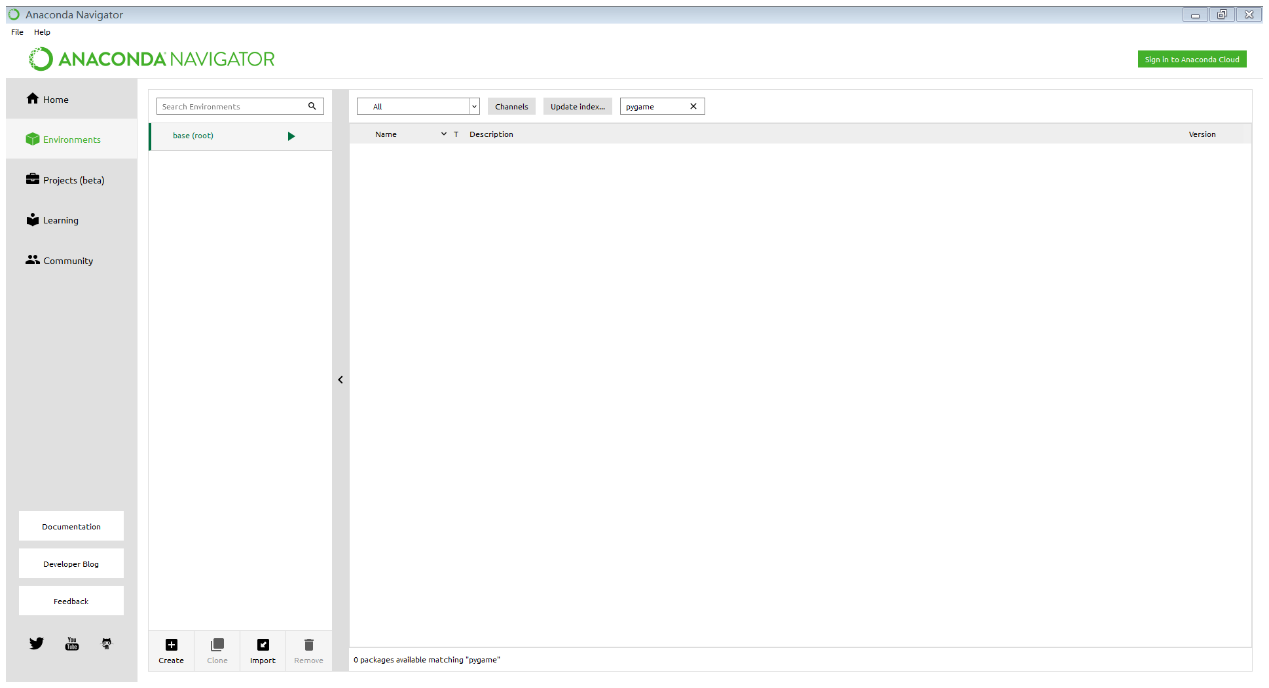
解释

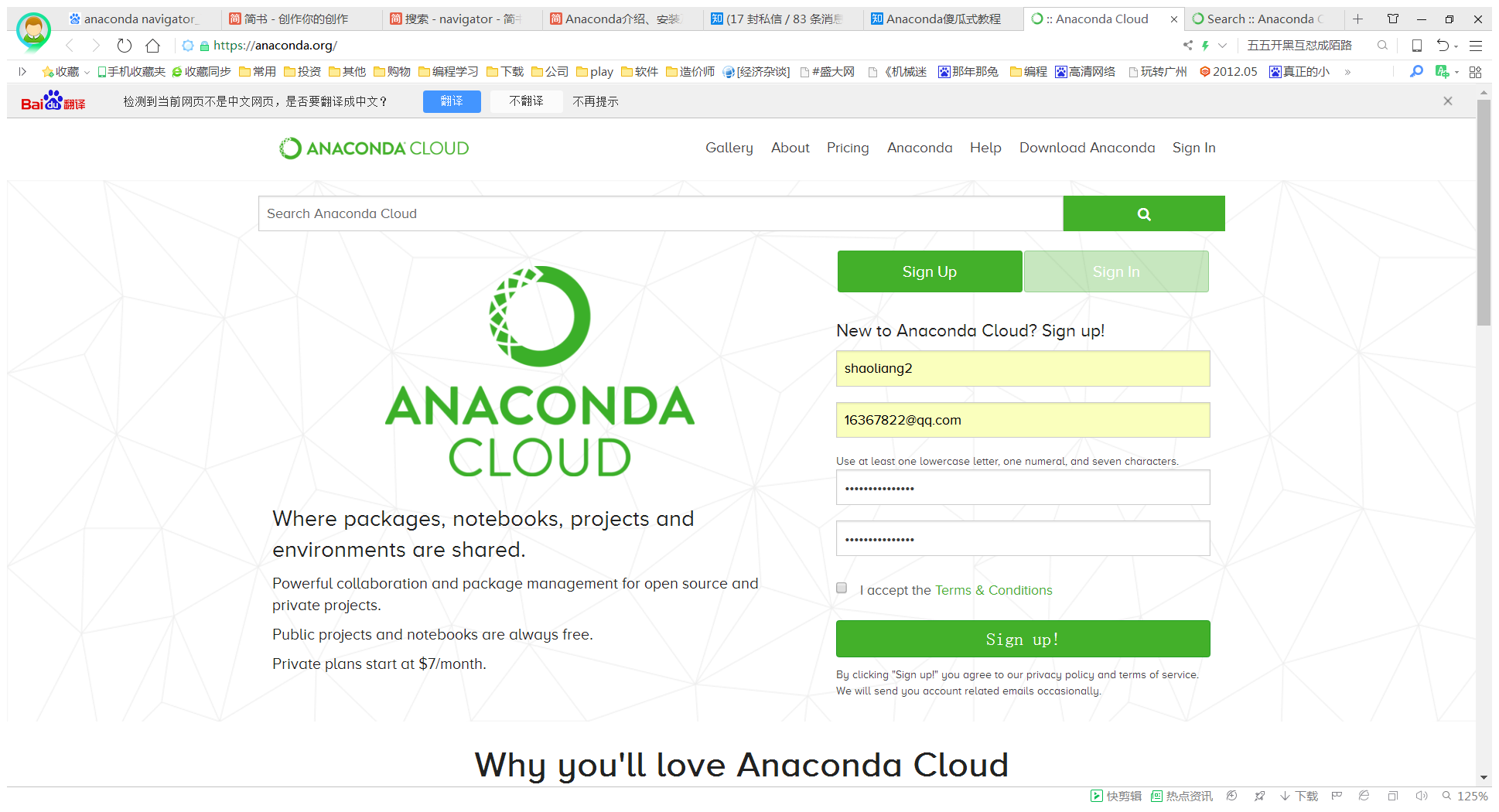
练习

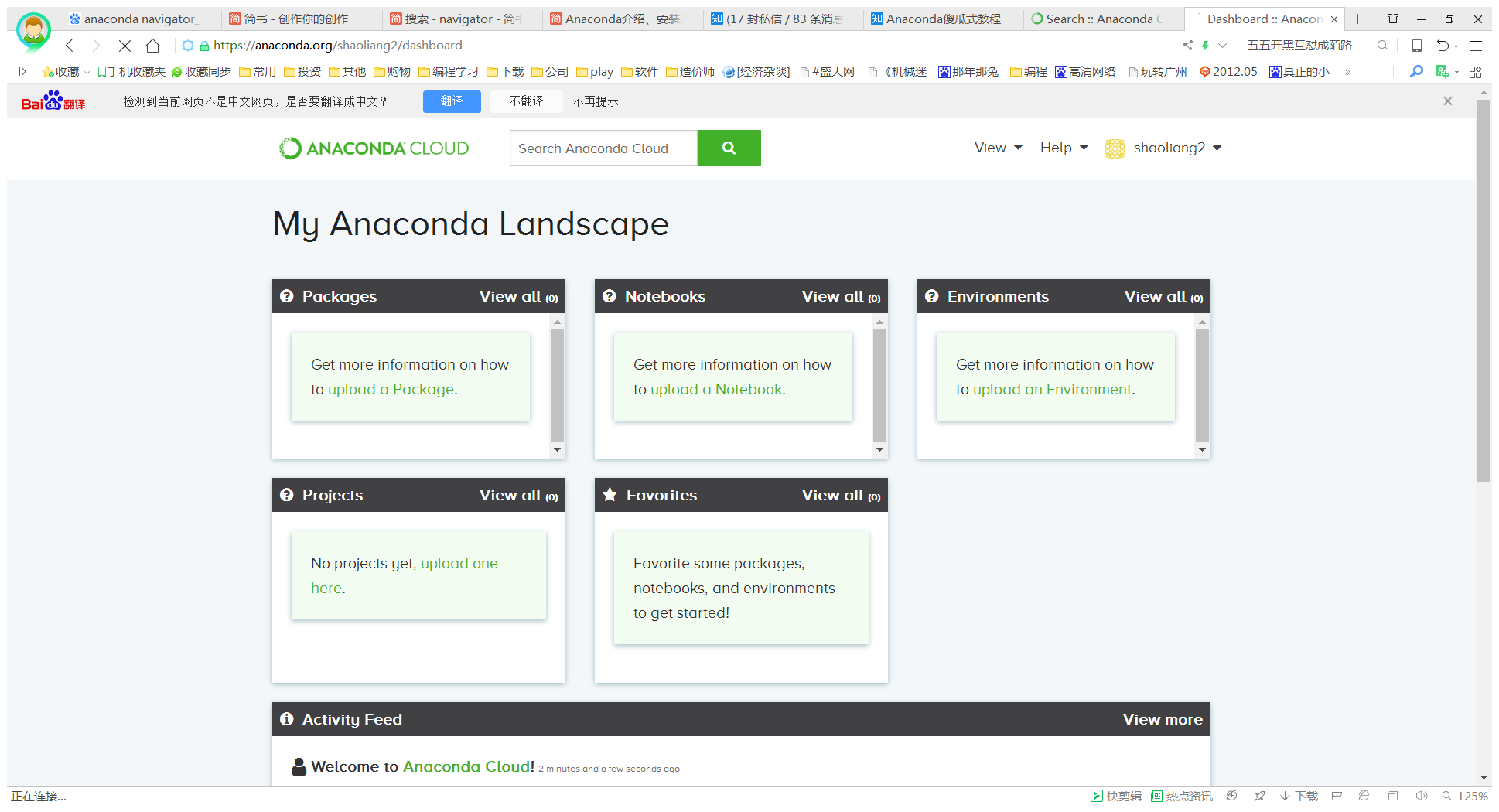
1.2

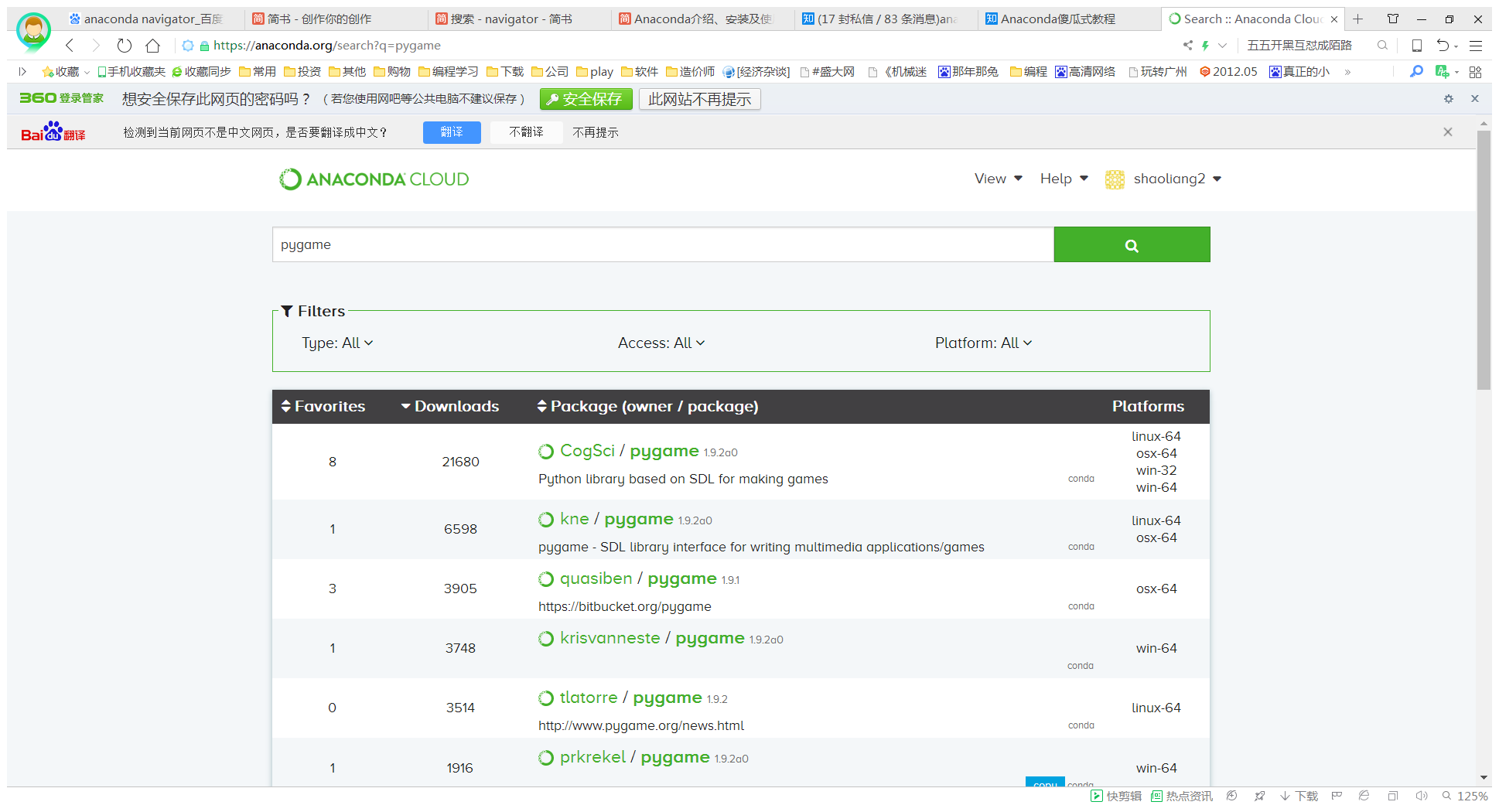
1.2.1 清华源

anaconda navigator









第二章 python基础

附录一 git

可以参考廖雪峰的GIT教程（https://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/001373962845513aefd77a99f4145f0a2c7a7ca057e7570000），在这里我们选择常用部分进行记录。

1. 安装

下载github，安装。安装之后会在开始菜单中出现一个Git目录，里面有三个快捷方式：git Bash、git CMD、git GUI，我们就用这个git bash。安装完成后，在开始菜单里找到“Git”->“Git Bash”，蹦出一个类似命令行窗口的东西，就说明Git安装成功！



1. 仓库

在学习使用git命令之前，我们应该先理解仓库的概念，仓库可以分为本地仓库和远程仓库。

本地仓库：就是本地的一个文件夹。

远程仓库：就是github(或者coding，或者码云，或者...)的一个线上文件夹。

对Git操作实际就是本地库和远程库之间的各种操作。（请注意：本地文件夹和远程文件夹请都避免使用中文名——包括所有的父目录和子目录，以免出现各种不可预料的问题。）

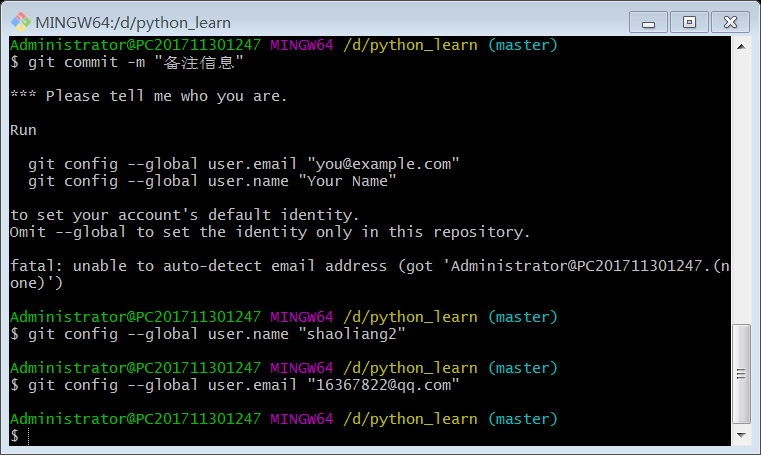
但是这个文件夹和普通的文件夹是由区别的，普通的文件夹只能存东西，仓库这个文件夹，不只能存东西，还有各种状态。

大家可以这么理解：git命令，其实就是切换和查看这个文件夹的各种状态而已。

1. 命令

在使用git之前，需要首先在github上注册一个账号（国内的码云或其他类似网站大同小异），网址<https://github.com/>，具体操作自行查阅。

在git bash中输入注册的信息：

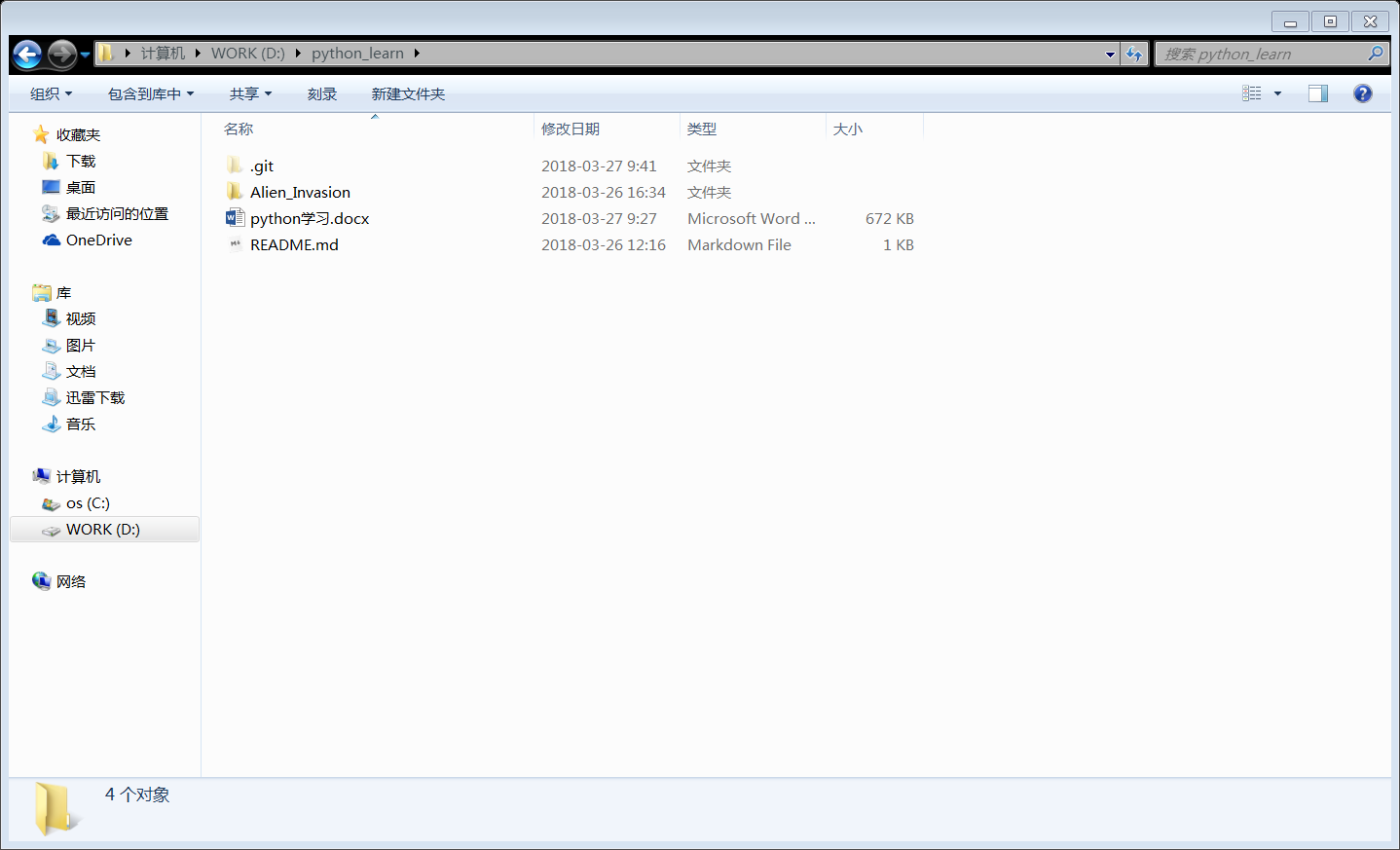


这样每次需要连接远程git库时，就会自动输入的信息登录了。

理解了仓库的概念，git命令就很好学习了。

第一步：创建一个新的文件夹，在里面点击鼠标右键，然后点击“Git Bash Here”,就会在当前的目录下打开一个命令行工具。

第二步：执行"git init"，可以将当前文件夹编程一个git仓库（从此他就有状态了）。



Git文件夹是隐藏的，请勿删除或修改，否则git状态会错乱。

第三步：在这个文件夹中添加一个新的文件（仓库有变化），然后执行"git status",可以查看仓库的状态，提示信息中会有红色标注的修改内容。

到这里，应该特殊说明一下，仓库有三个区域：编辑区、暂存区、分支。仓库中的文件虽然看似都安静地待在仓库这个文件夹中，但是他们可能所在不同的区域。执行"git status"可以查看文件所在区域，红色的是编辑区，绿色的是暂存区，如果查不到，说明文件在分支上。

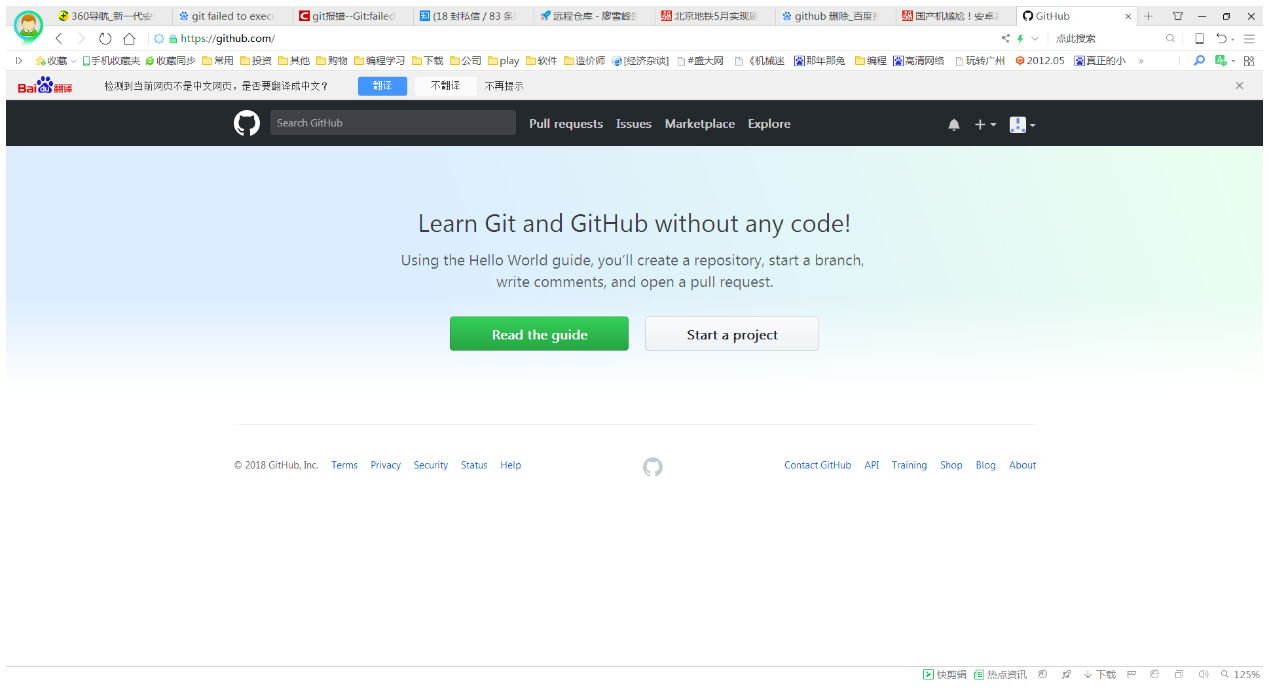
还有：只有分支上的内容才能同步到远程仓库。

那我们接下来要做的事儿就明确了，就是将编辑区的内容放到暂存区，再放到分支上。

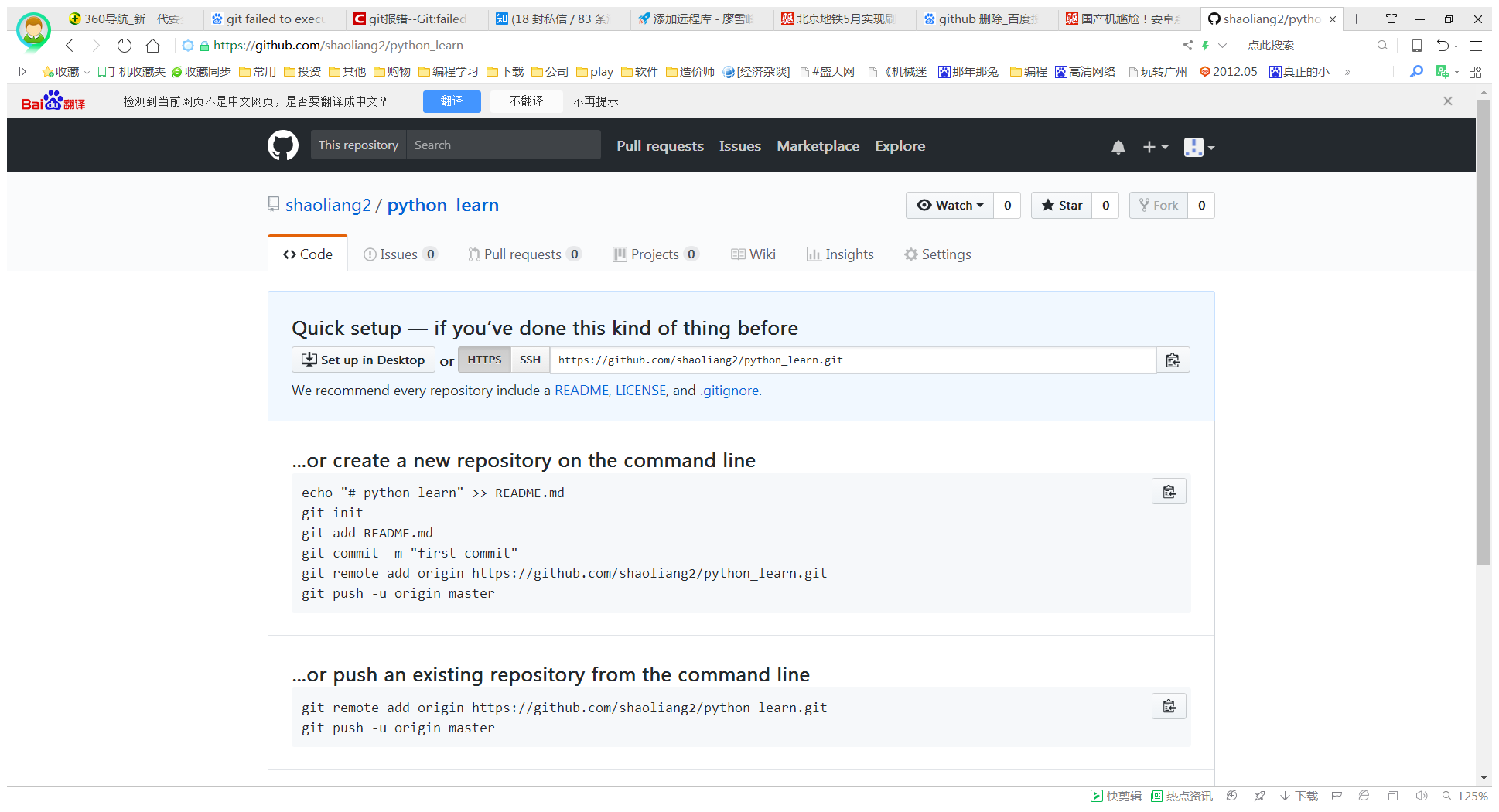
第四部：执行"git add .",将所有暂存区的内容添加到暂存区。

第五部：执行git commit -m "备注信息"，将暂存区内容提交到分支上。

第六步：创建远程仓库。在github网站上，start a project



输入一个名字，然后creat repository（创建一个仓库），就建立了一个远程库——在这里我没有选择建立一个readme.md文档。



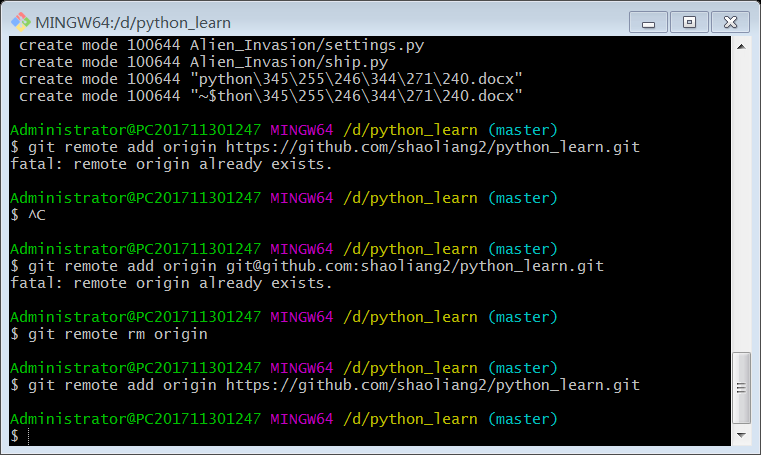
记录下<https://github.com/shaoliang2/python_learn.git>的地址，这个就是远程库的地址，我们需要将本地库和这个远程库地址链接起来，那么每次本地库的改动就可以上传到远程库上了。

第七步，执行"git remote",定义远程主机名，这一步完整的命令会在创建远程仓库的时候提示（git remote每个仓库执行一次就可以了，再次同步远程仓库不用再次执行）。

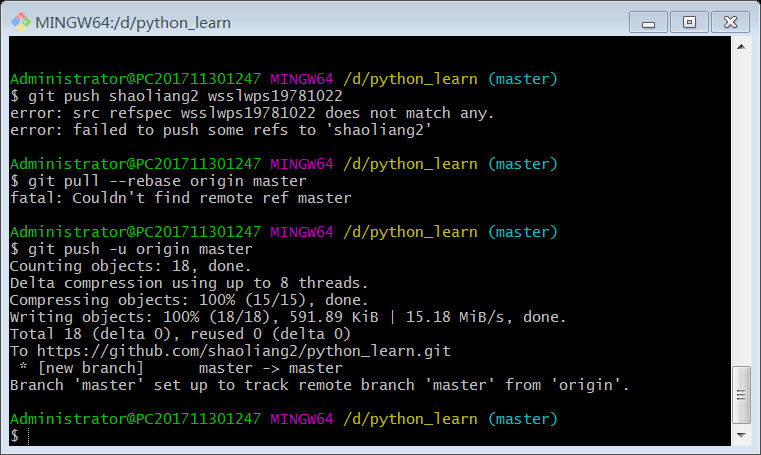
先输入：git remote rm origin

然后输入：git remote add origin <https://github.com/shaoliang2/python_learn.git>

本地仓库和远程仓库就连接起来了。



输入git push -u origin master，将本地仓库和远程仓库合并。



此时我们可以对比一下本地文件夹和远程仓库：

