学习GITHUB

目的：借助github托管项目代码

基本概念：

1. 仓库（repository）：本人的项目，用来存放项目代码，多个开源项目则有多个仓库
2. 收藏（star）：收藏别人的项目，方便下次收看
3. 复制克隆项目（fork）：复制张三的仓库到我的仓库里，并且独立存在
4. 发起请求（pull request）：在我复制了别人的仓库里，我又更新了，我想发送给张三，等待张三查看，若张三感觉不错，他会合并到自己仓库
5. 关注（watch）：类似于关注，人家再发布动态我能看见
6. 事务卡片（issue）：发现代码bug，发送给别人，一起讨论
7. GitHub主页：账号创建成功都可进入，该页左侧主要显示用户动态以及关注用户或仓库的动态，右侧显示所有的git库
8. 仓库主页：显示项目信息，如项目代码，版本，收藏关注fork等情况
9. 个人主页：头像，个人简历，关注我的，我关注的人、git库，我的开源项目，我贡献的开源项目等信息

注册账号：GitHub。Com

翻墙软件：lantern服务器

创建心新仓库的时候勾选方框是在仓库霞多一个reddme的说明文件，来详细描述项目

搜算文件快捷键：t

Github issues：发现代码bug，

ls：查看当前文件

pwd：查看当前工作目录

git status：查看文件当前处于哪个状态

git config –global user.name ‘ccccbeibei’ #设置基本信息

git config –global user.email ‘841586984@qq.com’

vi ：编辑文件的意思

初始化：初始化一次以后就不用初始化了

1. mkdir 文件名 #创建文件夹
2. cd 文件名 #进入该文件
3. git init #在文件夹内初始化git（创建git仓库），存储仓库信息的

向仓库中添加文件

1、touch 文件名 #得先创建一个文件

2、git add 文件名 #把文件提交到暂存区

3、git commit -m “” #提交描述将文件从暂存区提交到仓库

修改文件

1、vi 文件名 #修改文件

2、git status #查看状态

3、git add 文件名 #从工作区添加到暂存区

4、git commit -m “第一次通过git修改文件并提交到仓库”

5、git status #若有working directory clean 表示提交成功

删除仓库文件

1. rm -rf 文件名 #删除文件
2. git rm文件名 #从git中删除文件
3. git commit -m ‘通过git删除仓库文件

远程仓库代码

Git克隆操作

目的：将远程仓库（github对应的项目）复制到本地

在github中点下载的按钮，然后复制地址

git clone 仓库地址（刚刚复制的那个地址）

git push #将文件从本地仓库提交到远程仓库

**Git Basics Exercise**

Now that you have learned the basics of Git workflow, try running through this a couple of times on your own:

1. Create a folder called learn\_git\_again. **mkdir learn\_git\_again**
2. cd into the learn\_git\_again folder.**cd learn\_git\_again**
3. Create a file called third.txt. **touch third.txt**
4. Initialize an empty git repository. **git init**
5. Add third.txt to the staging area.**git add third.txt**
6. Commit with the message "adding third.txt".**git commit -m "adding third.txt"**
7. Check out your commit with git log.**git log**
8. Create another file called fourth.txt.**touch fourth.txt**
9. Add fourth.txt to the staging area.**git add fourth.txt**
10. Commit with the message "adding fourth.txt"**git commit -m "adding fourth.txt"**
11. Remove the third.txt file. **rm third.txt**
12. Add this change to the staging area. **git add third.txt**
13. Commit with the message "removing third.txt". **git commit -m "removing third.txt"**
14. Check out your commits using git log. **git log**
15. Change your global setting to core.pager=cat - you can read more about that [here](https://git-scm.com/book/en/v2/Customizing-Git-Git-Configuration). **git config --global core.pager "cat"**
16. Write the command to list all of the global configurations for git on your machine. You can type git config --globalto find out how to do this - **git config --global --list**