这个文件是介绍爬虫文件的使用方法：

环境搭配：

首先你需要拥有python2.7的环境，只需要在官网上下载对应版本的文件，然后改一下系统环境变量即可。

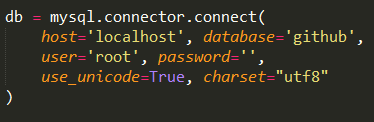
需要安装requests,mysql-connector才能使用我们的爬虫文件，在安装前需要安装pip，如果使用系统为Centos参考：

<https://blog.csdn.net/yulei_qq/article/details/52984334>

然后使用pip install requests, pip install mysql-connector即可（linux和windows类似）

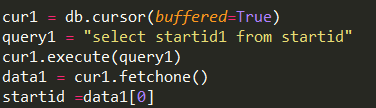
github-repo.py：（主要是用API获取信息）

首先是数据库的连接：

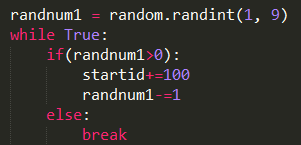


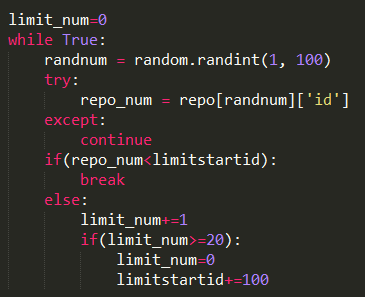
数据库的创建可以参考create\_table中的github.sql(确保数据库包括contributors, repositories\_data,users,startid四个表)

当首次使用这个文件时，你需要在startid表中把startid设置为你想开始爬取的repo\_id，同时这个startid也是为了保存每次运行爬取到哪个id了，便于之后的爬取

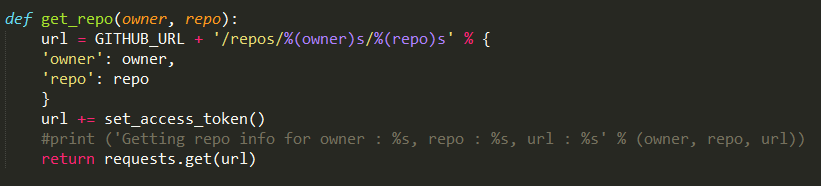


一下两图的代码主要是为了实现从1000个public repos中获取一个repos的实现





之后便可以根据这个repos的login,repo\_name进一步通过api获取repos的详细信息，使用的函数是：

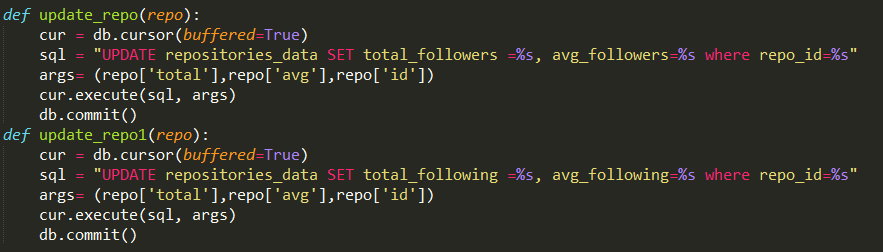


至此，便可以按照需求爬取repos的信息了。

caclulateFollow.py

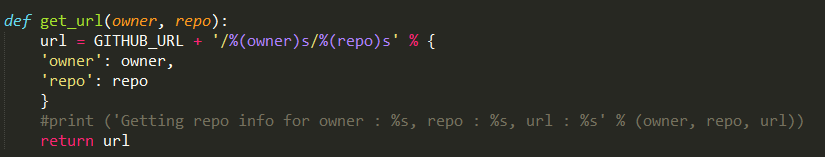
这个爬虫文件主要是计算repos的followers，followings的total以及avg信息，主要是通过contributors以及users的关联，通过sql语句即可计算出对应的followers，followings，

然后通过函数进行更新：

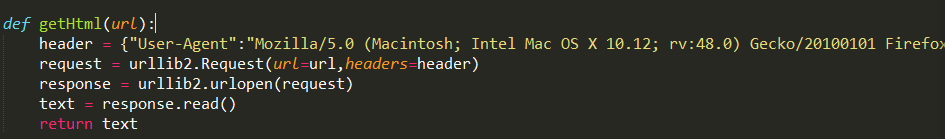


downloadreadme.py

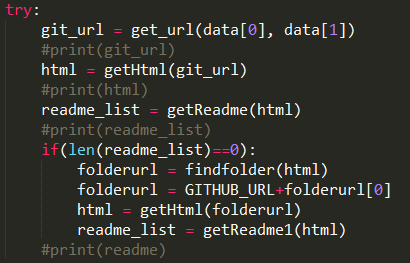
这个爬虫文件主要是通过repositories\_data数据库里的login,repo\_name去通过获取到github.com/{login}/{repo\_name}的仓库中，

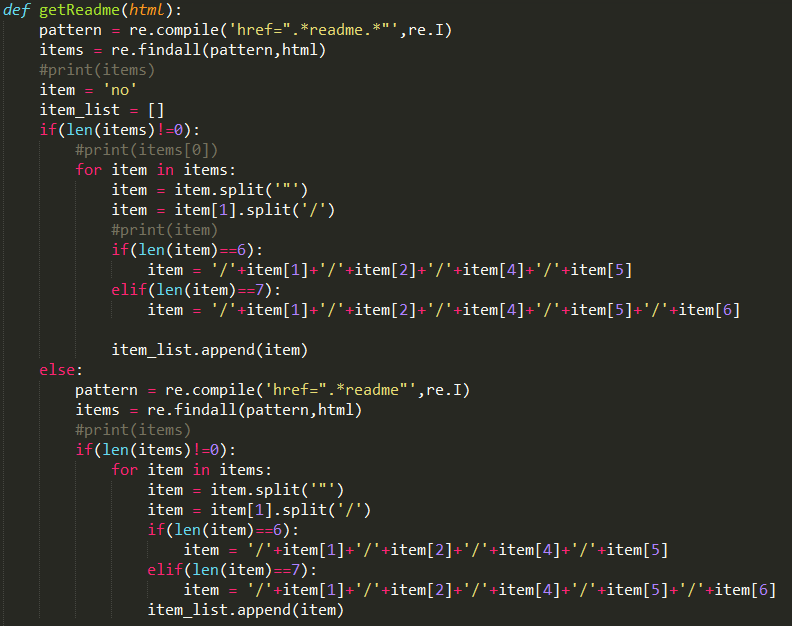


然后获取该网页的html文件：

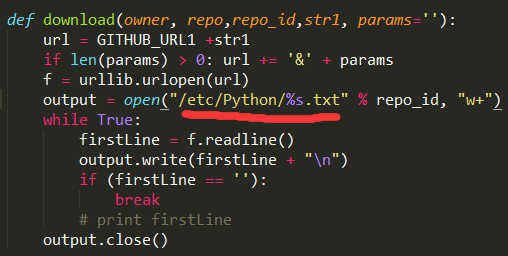


通过re模式匹配去找到readme的链接地址，具体的搜寻方法:



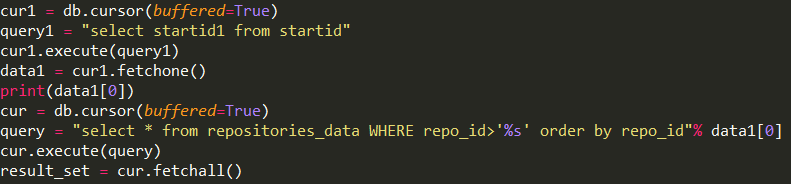


等代码中的对应函数，主要是通过re匹配的，最终返回readme的url，通过getHTML的函数获取对应的信息，即可通过download函数下载到本地，（注意地址的填写）

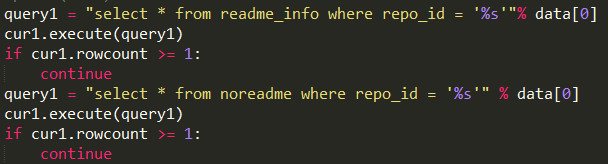


至此，该文件便可运行去爬取之前通过git-repo爬去的repos的readme文件了，而且只需有repo\_id,login,repo\_name即可实现readme的爬取

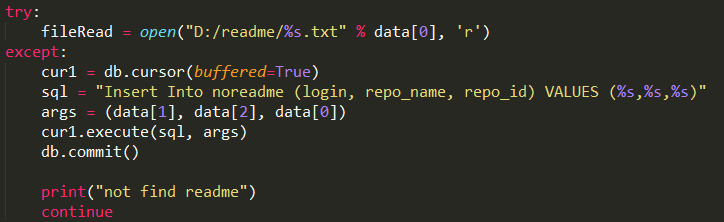
analysisReadme.py



获取repos的信息，在这里repo\_id,login,repo-name对应的index可能与代码写的有偏差，请读者print查看



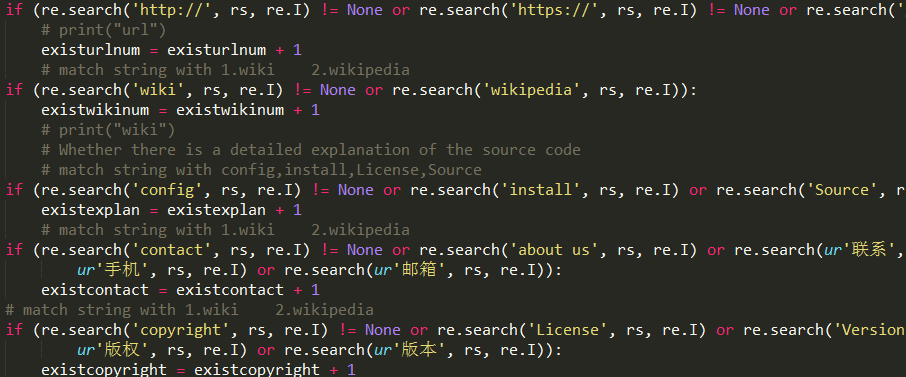
在开始之前需要查看是否已经存在相同的repos的信息在readme\_info，或者noreadme中（只是为了确保程序的健壮性，如果顺序执行便可忽略此步骤）



获取repos对应的readme，以文件的形式打开.



计算readme文件的大小



然后通过re的匹配对文件的每一行进行匹配，匹配的方面主要包括：1.URL 2.wiki 3.command 4.contact 5.copyright

读者可自行添加需要匹配的信息

然后把对应的repo\_id，url等信息insert into到数据库中即可.

有的repos不存在readme，会被添加到noreadme表中，在之后需要去确认一下是不是真的不存在readme文件，还是漏抓了

使用[url:github.com/{login}/{repo\_name}](url:github.com/%7blogin%7d/%7brepo_name%7d)在页面中去查看即可

至此，以上四个爬虫文件便介绍完了，希望可以对您起到帮助，谢谢参考！