实现文档

小组：猫和老鼠

组长：李佳骏 1613368

组员：潘巧巧 1613415

（1） 小组分工

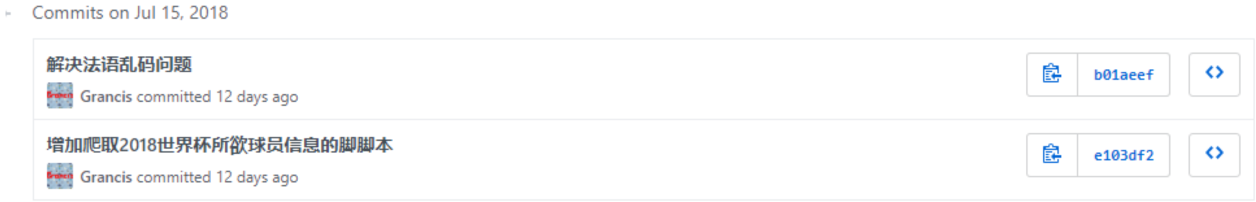
李佳骏：前端页面的设计制作，后端与前台网页的交互，对数据库的查询，查询信息的呈现，前后端交互（通过 get 与 post 请求）

潘巧巧：数据库的设计和建立、数据爬虫代码的编写和操作、后台登录界面和操作界面的代码实现，所有文档的后台与数据库部分。

（2） 爬虫实现

主要数据来源：FIFA 官网 www.FIFA.com

爬虫代码 GITHUB 上传记录：



Commits on Jul 15, 2018

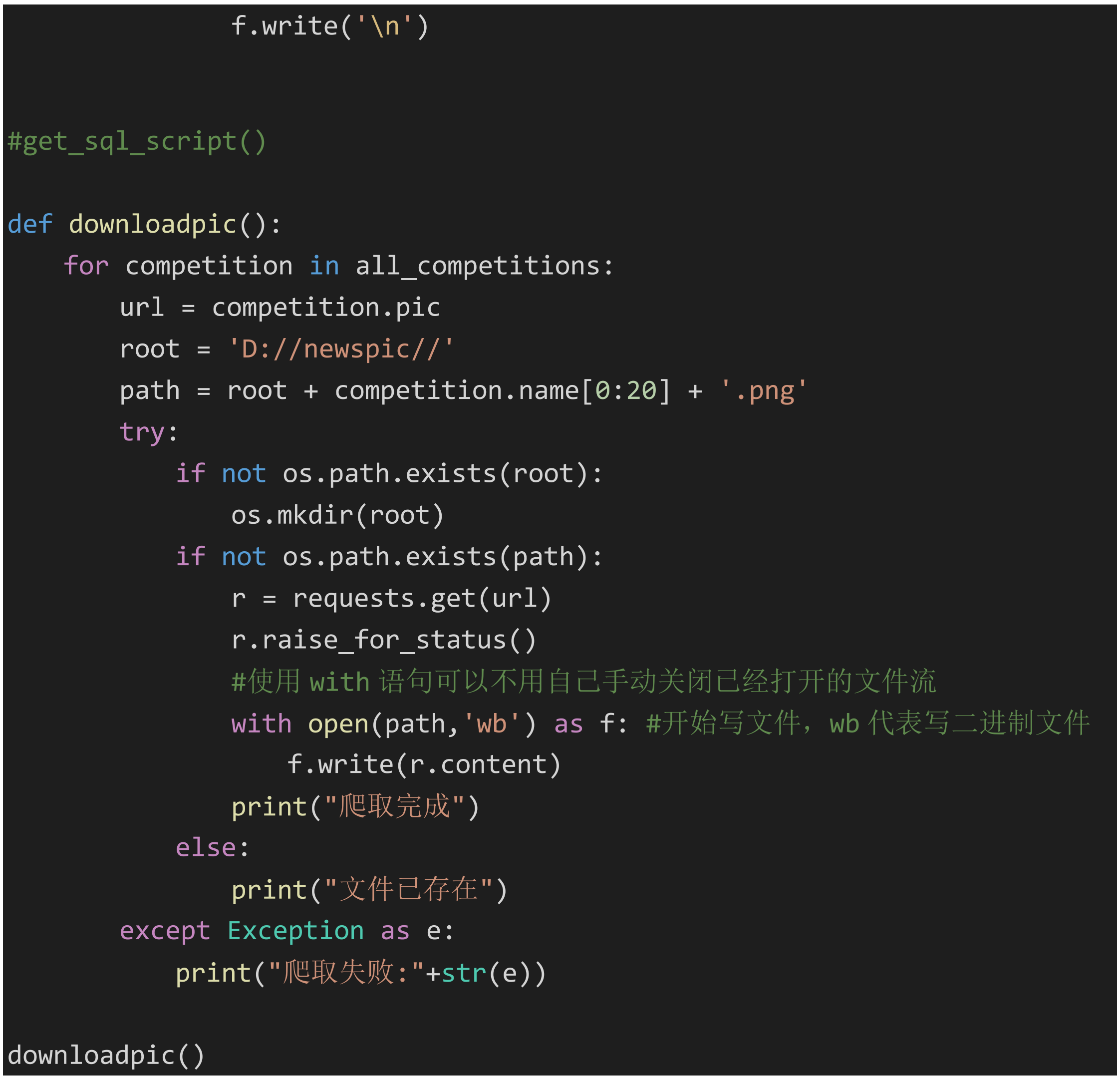
解决法语乱码问题
Grancis committed 12 days ago
b01aeef

增加爬取2018世界杯所欲球员信息的脚脚本
Grancis committed 12 days ago
e103df2

爬虫代码实例：功能为爬取世界杯相关新闻的标题、发表时间、链接、图片

```
#!/usr/bin/env python3
#-*-encoding=utf-8-*-

#time
#city

import requests
from html.parser import HTMLParser
import json
```

```python
import re
import os


#类 Competition
class Competition(object):
    def __init__(self,name,time,link,pic): #构造函数，生成对象时自动执行
        self.name=name
        self.time=time
        self.link=link
        self.pic=pic



class CompetitionParser(HTMLParser):
    def __init__(self):#构造函数
        HTMLParser.__init__(self)
        self.__flag=None
        self.name=[]
        self.time=[]
        self.link=[]
        self.pic=[]


    def handle_starttag(self,tag,attrs):
        if tag=='img':
            if len(attrs)==3 and attrs[0][0]=='data-src':
                if attrs[2][1]=='img-responsive lazyload':
                    self.pic.append(attrs[0][1])

        if tag=='p':
            if len(attrs)==1 and attrs[0][1]=='d3-o-media-object__date fi-o-media-object__date':
                self.__flag='time'

        if tag=='a':
            if len(attrs)==3 and attrs[2][1]=='fi-o-media-object__link':
                self.__flag='link'

        if tag=='h3':
            if len(attrs)>=1 and attrs[0][1]=='d3-o-media-object__title fi-o-media-object__title':
                    self.__flag='name'
    def handle_endtag(self,tag):
        if tag=='a' or tag =='img' or tag == 'h3' or tag=='p':
            self.__flag=None

    #压栈
    def handle_data(self,data):
        if self.__flag=='name':
            self.name.append(data)
        elif self.__flag=='time':
            data=re.split(r'[\s]',data)
            self.time.append(data)
        elif self.__flag=='link':
            self.link.append(data)




#对每场比赛进行逐个解析
#para: url
#return Player[]
def getCompetitionList(url):
    competitions=[]
    res=requests.get(url)
    parser=CompetitionParser()
    parser.feed(res.text)
    for i in range(len(parser.name)):

competition=Competition(parser.name[i],parser.time[i],parser.link[i],parser.pic[i])
        competitions.append(competition)
    return competitions


url = 'https://www.fifa.com/worldcup/news/'
all_competitions=getCompetitionList(url)


def get_sql_script():
    with open('./study1.sql','w',encoding='utf-8') as f:
        for competition in all_competitions:
                f.write('INSERT INTO `news`(`news_title`, `news_time`, `news_link`) VALUES (\"%s\",\"%s\",\"%s\");' %(competition.name,competition.time,competition.link))
```



```
                f.write('\n')


#get_sql_script()

def downloadpic():
    for competition in all_competitions:
        url = competition.pic
        root = 'D://newspic//'
        path = root + competition.name[0:20] + '.png'
        try:
            if not os.path.exists(root):
                os.mkdir(root)
            if not os.path.exists(path):
                r = requests.get(url)
                r.raise_for_status()
                #使用 with 语句可以不用自己手动关闭已经打开的文件流
                with open(path,'wb') as f: #开始写文件，wb 代表写二进制文件
                    f.write(r.content)
                print("爬取完成")
            else:
                print("文件已存在")
        except Exception as e:
            print("爬取失败:"+str(e))

downloadpic()
```