**工作总结**

**时间：2017.10.8——2017.10.15**

**组别：BlogRunner**

**报告人：前端设计开发工程师：郭莹婷**

报告内容：

1. 学习Python语言：
2. 学习Python语言基本语法与使用规则；
3. 函数与条件处理
4. 嵌套语句
5. Python GUI编程(Tkinter)
6. 导入 Tkinter 模块
7. 创建控件
8. 指定这个控件的 master， 即这个控件属于哪一个
9. Python JSON
10. 使用 Python 语言来编码和解码 JSON 对象
11. JSON函数
12. 学习Python语言使用范围与应用范围；
13. 用Python语言进行基本的使用。
14. 学习爬取网页
15. 学习Python案例：国庆哪里最堵；
16. 爬取输入区域的景点名称、地址、销量、售价等信息并存入excel表格；
17. 根据景点位置获取景点经纬度等信息生成与百度地图api、echarts示例对接的json文件；
18. 对百度地图api、echarts进行修改可引入本地json文件。
19. 学习网页爬取，读取json文件；
20. 通过慕课网学习网页爬取内容。
21. 分析一个网站[落网：http://luoo.net] 对网站内容进行提取来进一步了解：

第一步 确定目的

1. 抓取目标网站的某一期所有音乐

第二步 分析页面结构  
访问落网的某一期刊，通过Chrome的开发者模式查看播放列表中的歌曲

3． 代码：

学习网址：

<https://github.com/imchenkun/ick-spider/blob/master/luoospider.py>

#-\*- coding: utf-8 -\*-

'''by sudo rm -rf  http://imchenkun.com'''

import os

import requests

from bs4 import BeautifulSoup

import random

from faker import Factory

import Queue

import threading

fake = Factory.create()

luoo\_site = 'http://www.luoo.net/music/'

luoo\_site\_mp3 = 'http://luoo-mp3.kssws.ks-cdn.com/low/luoo/radio%s/%s.mp3'

proxy\_ips = [    '27.15.236.236'    ] # 替换自己的代理IP

headers = {

    'Connection': 'keep-alive',

    'User-Agent': fake.user\_agent()

    }

def random\_proxies():

    ip\_index = random.randint(0, len(proxy\_ips)-1)

    res = { 'http': proxy\_ips[ip\_index] }

    return res

def fix\_characters(s):

    for c in ['<', '>', ':', '"', '/', '\\\\', '|', '?', '\*']:

        s = s.replace(c, '')

    return s

class LuooSpider(threading.Thread):

    def \_\_init\_\_(self, url, vols, queue=None):

        threading.Thread.\_\_init\_\_(self)

        print '[luoo spider]'

        print '=' \* 20

        self.url = url

        self.queue = queue

        self.vol = '1'

        self.vols = vols

    def run(self):

        for vol in self.vols:

            self.spider(vol)

        print '\\ncrawl end\\n\\n'

        def spider(self, vol):

        url = luoo\_site + vol

        print 'crawling: ' + url + '\\n'

        res = requests.get(url, proxies=random\_proxies())

                soup = BeautifulSoup(res.content, 'html.parser')

        title = soup.find('span', attrs={'class': 'vol-title'}).text

        cover = soup.find('img', attrs={'class': 'vol-cover'})['src']

        desc = soup.find('div', attrs={'class': 'vol-desc'})

        track\_names = soup.find\_all('a', attrs={'class': 'trackname'})

        track\_count = len(track\_names)

        tracks = []

        for track in track\_names:

            \_id = str(int(track.text[:2])) if (int(vol) < 12) else track.text[:2]  # 12期前的音乐编号1~9是1位（如：1~9），之后的都是2位 1~9会在左边垫0（如：01~09）

            \_name = fix\_characters(track.text[4:])

            tracks.append({'id': \_id, 'name': \_name})

            phases = {

                'phase': vol,                         # 期刊编号

                'title': title,                       # 期刊标题

                 'cover': cover,                      # 期刊封面

                 'desc': desc,                        # 期刊描述

                 'track\_count': track\_count,          # 节目数

                 'tracks': tracks                     # 节目清单(节目编号，节目名称)

            }

            self.queue.put(phases)

class LuooDownloader(threading.Thread):

    def \_\_init\_\_(self, url, dist, queue=None):

        threading.Thread.\_\_init\_\_(self)

        self.url = url

        self.queue = queue

        self.dist = dist

        self.\_\_counter = 0

     def run(self):

        while True:

            if self.queue.qsize() <= 0:

                pass

            else:

                phases = self.queue.get()

                self.download(phases)

    def download(self, phases):

        for track in phases['tracks']:

            file\_url = self.url % (phases['phase'], track['id'])

            local\_file\_dict = '%s/%s' % (self.dist, phases['phase'])

            if not os.path.exists(local\_file\_dict):

                os.makedirs(local\_file\_dict)

            local\_file = '%s/%s.%s.mp3' % (local\_file\_dict, track['id'], track['name'])

            if not os.path.isfile(local\_file):

                print 'downloading: ' + track['name']

                res = requests.get(file\_url, proxies=random\_proxies(), headers=headers)

                with open(local\_file, 'wb') as f:

                    f.write(res.content)

                    f.close()

                print 'done.\\n'

            else:

                print 'break: ' + track['name']

if \_\_name\_\_ == '\_\_main\_\_':

    spider\_queue = Queue.Queue()

    luoo = LuooSpider(luoo\_site, vols=['680', '721', '725', '720'],queue=spider\_queue)

    luoo.setDaemon(True)

    luoo.start()

    downloader\_count = 5

    for i in range(downloader\_count):

        luoo\_download = LuooDownloader(luoo\_site\_mp3, 'D:/luoo', queue=spider\_queue)

        luoo\_download.setDaemon(True)

        luoo\_download.start()

1. 学习前端开发
2. 在Linux环境下搭建前端开发环境（正在搭建）；
3. 设计前端开发界面结构；
4. 学习前端开发要点及内容;
5. 熟悉前端开发环境element+vue2.0

**HTML元素**

每个HTML元素都存在的标准特性

* id 元素在文档中的唯一标识符
* title 有关元素的附加说明信息
* lang 元素内容的语言代码
* dir 语言方向。值为"ltr"【从左至右】| "rtl"【从右至左】
* className 与元素的class特性对应。

**获取特性**

* getAttribute()  
  【注意】传递给getAttribute的特性名与实际特性名相同（不区分大小写）。获取class需传入"class"而不是"className"。可获取**自定义特性**。通常只有在取得自定义特性值的情况下使用getAttribute()方法。【特殊情况如下】
  + style 通过getAttribute访问时返回的style特性值中包含的是css文本，通过属性访问则会返回一个对象
  + onclick 返回相应代码的字符串 通过属性访问时会返回一个js函数
* setAttribute()   
  如果特性已经存在，会以指定值替换现有的值；如果特性不存在，则创建该属性并设置相应值。
* removeAttribute()  
  用于彻底删除元素特性。不仅会清除特性的值，也会从元素中完全删除特性。

### 创建元素

* document.createElement()只接受一个参数【要创建元素的标签名】
* 将新元素添加到文档树
  + appendChild()
  + insertBefore()
  + replaceChild(0
* 需要注意的问题
  + 不能设置动态创建的<iframe>元素的name特性
  + 不能通过表单的reset方法重设动态创建的<input>元素
  + 动态创建的type特性值为"reset"的<button>元素重设不了表单
  + 动态创建的一批name相同的单选按钮彼此毫无关系。
* 针对上述问题的**解决方案**：在createElement中指定完整的HTML标签来解决。