python学习知识点

**一、函数方法**

**精英日志**

**1.以r或R开头的python中的字符串表示（非转义的）原始字符串**

**2.以u或U开头的字符串表示unicode字符串**

**3.range（start， end， scan):**

**4.sys.getcwd() 输出当前目录地址**

**5.sys.path 输出当前目录现所有文件名称数组**

**6.重复操作符( \* )**

**7.strip() 去除空格**

**列表**

**i = [ i \* 2 for i in [8, -2, 5] ]**

**print i**

**8.extract后会把selector对象转换成list类型了**

**9.Python通过re模块提供对正则表达式的支持。使用re的一般步骤是先使用re.compile()函数，将正则表达式的字符串形式编译为Pattern实例**

**import re**

**pattern = re.compile('[a-zA-Z]')**

**result = pattern.findall('as3SiOPdj#@23awe')**

**print result**

**# ['a', 's', 'S', 'i', 'O', 'P', 'd', 'j', 'a', 'w', 'e']**

**10.zlib.decompress 解压缩文件**

**11.模块 WordCloud 词云**

**12.yield scrapy.Request**

**13.python中，platform模块给我们提供了很多方法去获取操作系统的信息**

**如：**

**import platform**

**platform.platform() #获取操作系统名称及版本号，'Windows-7-6.1.7601-SP1'**

**platform.version() #获取操作系统版本号，'6.1.7601'**

**platform.architecture() #获取操作系统的位数，('32bit', 'WindowsPE')**

**platform.machine() #计算机类型，'x86'**

**platform.node() #计算机的网络名称，'hongjie-PC'**

**platform.processor() #计算机处理器信息，'x86 Family 16 Model 6 Stepping 3, AuthenticAMD'**

**platform.uname() #包含上面所有的信息汇总，uname\_result(system='Windows', node='hongjie-PC',**

**14.使用代理文章 https://github.com/chenjiandongx/stackoverflow/blob/master/stackoverflow/middleware/httpproxy.py**

**15.使用 userAgant**

**https://github.com/chenjiandongx/stackoverflow/blob/master/stackoverflow/middleware/useragent.py**

**16.Python strip() 方法用于移除字符串头尾指定的字符（默认为空格）。**

**17.unicodePage = myPage.decode("utf-8", 'ignore') decode 转换字符集**

**18.resp.text返回的是Unicode型的数据。**

**resp.content返回的是bytes型也就是二进制的数据。**

**也就是说，如果你想取文本，可以通过r.text。**

**如果想取图片，文件，则可以通过r.content。**

**（resp.json()返回的是json格式数据）**

**19.requirements.txt**

**安装 (venv) $ pip install -r requirements.txt**

**生成 pip freeze > requirements.txt**

**20.join 以空格相连**

**REDIS.set('keywords', ' '.join(keywords))**

**21.for i in range(1, 21): for 数组 从 1 到 21**

**22. if a and b**

**if a or b**

**23. Python split()通过指定分隔符对字符串进行切片，如果参数num 有指定值，则仅分隔 num 个子字符串**

**24. exec()是Python的built-in函数。其作用很好描述，就是执行以string类型存储的Python代码。话不多说举个例子。**

**25. exit(0)：无错误退出**

**exit(1)：有错误退出**

**退出代码是告诉解释器的（或操作系统）**

**项目：music-163**

**来源：origin https://github.com/RitterHou/music-163.git (push)**

**知识笔记：**

**模块 artists.py 获取 文章 id 存入 数据库**

**1.python 2.7**

**2.使用 requests抓取，BeautifulSoup 网页分析，存储 mysql 数据库**

**3.request.get(url,params),抓取， params = {id: group\_id,'initial':initial}**

**网页分析**

**soup = BeautifulSoup(r.text,'html.parser') //r.text 网页内容 r.content 二进制内容 编码 r.content.decode()**

**# print hot\_artists.decode('gbk','ignore').encode('utf-8')**

**body = soup.body**

**模块 album\_by.py 根据 文章 id 获取 详细信息**

**1.params = {'id':artist\_id,'limit':'200'}**

**r = requests.get(url,headers=self.headers,params=params)**

**#网页分析**

**soup = BeautifulSoup.soup(r.text,'html.parser')**

**body = soup.body**

**for 一维数组a in 二维数组b**

**try:**

**print a.i**

**except Exception e:**

**print str(i) + ':' + str(e)**

**time.sleep(5) //暂停 5秒**

**模块 music\_by\_album.py 获取个人专辑**

**musics = body.find('ul',attrs={'class':'f-hide'}).find\_all('li')**

**if \_\_name\_\_=='\_\_main\_\_'**

**模块 comments\_by\_music.py 获取评论**

**threading 进程**

**pymysql.cursors mysql 游标**

**proxies = {'http':'http://12.0.0.1:10800'}**

**r = request.post(url,headers,params,data,proxies)**

**#pymysql 链接**

**connection = pymysql.connect(host,user,password,charset,cursorclass)**

**t1 = threading.Thread()**

**t1.start()**

**模块 sql**

**import pymysql.cursors**

**connection = pymysql.connect(host,user,db,charset,cursorclass)**

**def insert\_comments():**

**with connection.cursor() as cursor:**

**sql = "insert into `comments` (`music\_id`,`coments`,`details`) values (%s,%s,%s)"**

**cursor.execute(sql,(music\_id,comments,detail))**

**connection.commit()**

**def dis\_connect():**

**connection.close()**

**项目：github-search**

**1.db.py 链接redis**

**import redis**

**REDIS=get\_redis()**

**REDIS.SET('kk',''.join(arr))**

**REDIS.GET('kk').decode('utf-8');**

**2.app.py**

**import flask**

**app = Flask(\_\_name\_\_)**

**@app.route('/')**

**@app.route('/del\_repo',methods=['DELETE'])**

**app.run()**

**知识点：列表生成器**

**1.循环列表**

**>>> [x \* x for x in range(1, 11)]**

**[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]**

**2.二层循环**

**>>> [x \* x for x in range(1, 11) if x % 2 == 0]**

**[4, 16, 36, 64, 100]**

**>>> [m + n for m in 'ABC' for n in 'XYZ']**

**['AX', 'AY', 'AZ', 'BX', 'BY', 'BZ', 'CX', 'CY', 'CZ']**

**flask 知识点**

**1.@app.route("/") 设置路由**

**return reder\_template("index.html",argv=argv)**

**3.github\_search.py 获取信息，存入redis**

**REDIS.hsetnx('repos',key,value) 不存在的时候设置内容**

**项目：douban-spider**

**# from openpyxl import Workbook #写入Excel中**

**链家地产**

**https://github.com/lanbing510/LianJiaSpider**

**1.opner url参数，获取 cookies信息**

**2. re.compile(r'JSESSIONID=(.\*)') python 正则，获取sessionid**

**3. threading.Lock() 线程锁**

**4. sqlite 数据库**

**5. cookie = cookielib.CookieJar()**

**爬取策略-京东商品评论**

**http://samray.xyz/%E4%B8%A4%E5%8F%AA%E8%A0%A2%E8%90%8C%E7%9A%84%E7%88%AC%E8%99%AB**

**众所周知，爬虫比较难爬取的就是动态生成的网页，因为需要解析 JS, 其中比较典型的 例子就是淘宝，天猫，京东，QQ 空间等。所以在我爬取京东网站的时候，首先需要确 定的就是爬取策略。因为我想要爬取的是商品的信息以及相应的评论，并没有爬取特定 的商品的需求。所以在分析京东的网页的 url 的时候, 决定使用类似全站爬取的策略。**

**1.提取数据**

**2.反爬虫策略**

**3.禁用 cookie**

**4.轮转 user-agent**

**5.伪装成搜索引擎**

**6.代理 IP**

**7.扩展成分布式爬虫**

**8.爬虫监控**

**9.爬虫拆分**

**项目：news-spider**

**import tools.Global as Global**

**import os**

**import Global**

**if not os.path.exists(path):**

**os.makedirs(path)**

**f = open(file,'w')**

**f.readline()**

**f.close()**

**print 'Done';**

**项目：github-trending**

**print datetime.datetime,datetime.datetime.now()**

**datetime.datetime.now() //时间格式 2018-01-05 12：30：00**

**strdate = datetime.datetime.now().strftime('%Y-%m-%d') //输出年月日 2018-01-05**

**filename = '{date}.md'.format(date=strdate) //{date} 被 strdate替换为2018-01-05.md**

**format //格式替换**

**//os 执行 git 命令**

**def git\_add\_commit\_push(date, filename):**

**cmd\_git\_add = 'git add {filename}'.format(filename=filename)**

**cmd\_git\_commit = 'git commit -m "{date}"'.format(date=date)**

**cmd\_git\_push = 'git push -u origin master'**

**os.system(cmd\_git\_add)**

**os.system(cmd\_git\_commit)**

**os.system(cmd\_git\_push)**

**assert //assert断言语句的语法格式**

**assert 1==1**

**assert 2+2==2\*2**

**assert r.status\_code == 200 //如果返回不是200状态，就报错误**

**项目：news-feed**

**celery 定时任务**

**tornado web服务器**

**difflib 差异比较**

**项目：51job-log**

**jieba（结巴）是一个强大的分词库，完美支持中文分词，本文对其基本用法做一个简要总结。**

**生成词云之python中WordCloud包的用**

**python学习：counter计数**

**counter = dict() 字典**

**import csv**

**import matplotlib.pyplot as plt**

**Matplotlib.pyplot绘图实例 做图表**

**项目：tech163news-spider**

**包引入 相对路径**

**from ..items import Tech163Item**

**import pymongo 引入mongodb**

**// 定义爬虫url规则**

**rules = (**

**Rule(LinkExtractor(allow=r"/14/08\d+/\d+/\*"),**

**callback="parse\_news",follow=True),**

**)**

**strip() str.strip(' aaa ') 去除字符串两边的字符，默认为空格**

**\*未完需要再看一次并且实践**

**项目：tumblr-spider**

**import queue 安装失败，用from multiprocessing import Queue代替**

**get 返回dict对象中的指定值**

**dict = {'Name': 'Zara', 'Age': 27}**

**print "Value : %s" % dict.get('Age')**

**print "Value : %s" % dict.get('Sex', "Never")**

**Python find() 方法检测字符串中是否包含子字符串 str**

**str.find(str, beg=0, end=len(string))**

**Python Signal 信号**

**signal.signal(signal.SIGINT, handler)**

**转载 2014年04月25日 09:55:19 标签：python /编程 /通讯 20443**

**信号的概念**

**信号（signal）-- 进程之间通讯的方式，是一种软件中断。一个进程一旦接收到信号就会打断原来的程序执行流程来处理信号。**

**几个常用信号:**

**SIGINT 终止进程 中断进程 (control+c)**

**SIGTERM 终止进程 软件终止信号**

**SIGKILL 终止进程 杀死进程**

**SIGALRM 闹钟信号**

**项目：Tmall1212**

**1.Python startswith() 方法用于检查字符串是否是以指定子字符串开头，如果是则返回 True，否则返回 False。如果参数 beg 和 end 指定值，则在指定范围内检查。**

**语法**

**startswith()方法语法：**

**str.startswith(str, beg=0,end=len(string));**

**2.r.content.decode('gbk', 'ignore') 编码转化**

**3.设置编码**

**import sys**

**reload(sys)**

**sys.setdefaultencoding('utf8')**

**4.打开文件，添加内容**

**with open('failure.txt', 'a') as f:**

**f.write('%s erroe\n' % sid)**

**5.mongoDB**

**import pymongo**

**client = pymongo.MongoClient('localhost', 27017)**

**db = client['1212']**

**collection\_items = db['Tmall\_items']**

**collection\_items\_failure = db['Tmall\_items\_failure']**

**collection\_details = db['Tmall\_details']**

**collection\_details.insert(routine)**

**项目：cup\_size**

**1.set 集合**

**a = set('boy')**

**a.add('python')**

**set(['y','python','b','o'])**

**2.Python中append和extend的区别**

**list.append(object) 向列表中添加一个对象**

**object**

**list.extend(sequence) 把一个序列seq的内容添加到列表中**

**3.JSON 函数**

**使用 JSON 函数需要导入 json 库：import json。**

**函数 描述**

**json.dumps 将 Python 对象编码成 JSON 字符串**

**json.loads 将已编码的 JSON 字符串解码为 Python 对象**

**4.enumerate 枚举**

**enumerate还可以接收第二个参数，用于指定索引起始值，如：**

**list1 = ["这", "是", "一个", "测试"]**

**for index, item in enumerate(list1, 1):**

**print index, item**

**>>>**

**1 这**

**2 是**

**3 一个**

**4 测试**

**补充**

**如果要统计文件的行数，可以这样写：**

**count = len(open(filepath, 'r').readlines())**

**1**

**这种方法简单，但是可能比较慢，当文件比较大时甚至不能工作。**

**可以利用enumerate()：**

**count = 0**

**for index, line in enumerate(open(filepath,'r'))：**

**count += 1**

**5.open file**

**def save(self):**

**""" 保存数据到本地 """**

**with open(self.\_path, "w+", encoding="utf-8") as f:**

**f\_csv = csv.writer(f)**

**f\_csv.writerows(self.\_result)**

**6.cvs 导出**

**with open(self.\_path, "r", encoding="utf-8") as f:**

**fin\_csv = csv.reader(f)**

**for row in fin\_csv:**

**s.add(tuple(row))**

**with open("cup\_all.csv", "w+", encoding="utf-8") as f:**

**fout\_csv = csv.writer(f)**

**fout\_csv.writerows(s)**

**print(len(s))**

**7.staticmethod 静态方法**

**python staticmethod 返回函数的静态方法。**

**该方法不强制要求传递参数，如下声明一个静态方法：**

**电影网站抓取**

**1.import threading**

**from Queue**

**2.thread 示例**

**def mythread(site,num=20):**

**'''**

**num: 结合队列，跑多线程的抓取，默认线程数是20个**

**site: 这是一个关于某个站点的引用，比如 t = dytt8()**

**'''**

**queue = Queue()**

**for i in range(num):**

**t= ThreadUrl(queue,site)**

**t.setDaemon(True)**

**t.start()**

**for url in site.http\_url():**

**queue.put(url)**

**queue.join()**

**for ftp\_url in ftp\_urls:**

**print ftp\_url**

**项目：zhihu-girl**

**Bloomfilter就是将去重对象映射到几个内存“位”，通过几个位的 0/1值来判断一个对象是否已经存在。**

**ConfigParser模块在python3中修改为configparser.这个模块定义了一个ConfigParser类，该类的作用是使用配置文件生效，配置文件的格式和windows的INI文件的格式相同**

**由于python中默认的编码是ascii，如果直接使用open方法得到文件对象然后进行文件的读写，都将无法使用包含中文字符（以及其他非ascii码字符），因此建议使用utf-8编码。**

**使用方法**

**读**

**下面的代码读取了文件，将每一行的内容组成了一个列表。**

**import codecs**

**file = codecs.open('test.txt','r','utf-8')**

**lines = [line.strip() for line in file]**

**file.close()**

**写**

**下面的代码写入了一行英文和一行中文到文件中。**

**import codecs**

**file = codecs.open('test.txt','w','utf-8')**

**file.write('Hello World!\n')**

**file.write('哈哈哈\n')**

**file.close()**

**文件读写模式**

**最为常见的三种模式，见下表，其中模式就是指获取文件对象时传入的参数，最常用的是前三个。**

**|模式|描述|**

**|:-:|:-:|**

**|r|仅读，待打开的文件必须存在|**

**|w|仅写，若文件已存在，内容将先被清空|**

**|a|仅写，若文件已存在，内容不会清空|**

**|r+|读写，待打开的文件必须存在|**

**|w+|读写，若文件已存在，内容将先被清空|**

**|a+|读写，若文件已存在，内容不会清空|**

**|rb|仅读，二进制，待打开的文件必须存在|**

**|wb|仅写，二进制，若文件已存在，内容将先被清空|**

**|ab|仅写，二进制，若文件已存在，内容不会清空|**

**|r+b|读写，二进制，待打开的文件必须存在|**

**|w+b|读写，二进制，若文件已存在，内容将先被清空|**

**|a+b|读写，二进制，若文件已存在，内容不会清空|**

**random 随机数**

**for i in range(1, 33):**

**# 生成0-15的整数，并将其转换为16进制形式**

**n = random.randint(0, 15)**

**# hex将整数转化为16进制形式"0xc"，取第三个字符**

**n = hex(n)**

**n = n[2]**

**cnkiUserKey += n**

**# 在第8，12，16，20个字符后面加上连字符(-)**

**if i == 8 or i == 12 or i == 16 or i == 20:**

**cnkiUserKey += "-"**

**return cnkiUserKey**

**二、安装方法**

**1.安装 openpyxl（excel插件）错误，请用 pip install openpyxl==2.3.5 指定版本**

**2.本地离线安装包，进入安装包文件目录 python setup.py install / pip install openpyxl**

**三、项目笔记**

**比较完整的抓取步骤**

**存储内容**

**图片**

**url列表**

**使用beautifulSoup抓取**

**1.更新**

**2.查询**

**3.查找**

**# python-scrapy**

**python 爬虫脚本**

**python zlib 压缩文件**

**github项目**

**抓取地址列表：https://github.com/facert/awesome-spider**

**简单：自如实时房源提醒、网易云音乐**

**github\_search https://github.com/facert/github\_search**

**中等：豆瓣读书、链家、京东商品+评论、今日头条网易腾讯等新闻、github trending、新闻监控、前程无忧Python招聘岗位信息爬取分析、Shadowsocks 账号爬虫**

**网易新闻（mongo）**

**tumblr 多线程、天猫双12爬虫、Tmall 女性文胸尺码爬虫、电影网站、乌云公开漏洞**

**高级内容：https://github.com/yijingping/unicrawler 分布式抓取框架**

**https://github.com/bowenpay/wechat-spider 公众号抓取**

**微博主题搜索分析**

**知乎妹子**

**中国知网爬虫**

**总结**

**框架**

**flask django Scrapy**

**存储**

**sqlite msyql redis mongo elasticsearch**

**web服务器**

**uWSGI是一个Web服务器、 Tornado、Nginx**

**模块**

**cookies session queue队列 threading线程 proxy代理**

**大数据**

**knn kmeans 最小二次差 pandas数据分析numpy matplotlib绘图**

**重点**

**1.window 执行 python 命令**

**from scrapy import cmdline**

**cmdline.execute("scrapy crawl spider".split()) window 执行 python 方法**

**2.加载编码问题**

**import sys**

**reload(sys)**

**sys.setdefaultencoding('gbk') 或 'utf8'**

**3.xpath语法**

**Selector有四个基本的方法(点击相应的方法可以看到详细的API文档):**

**xpath(): 传入xpath表达式，返回该表达式所对应的所有节点的selector list列表 。**

**css(): 传入CSS表达式，返回该表达式所对应的所有节点的selector list列表.**

**extract(): 序列化该节点为unicode字符串并返回list。**

**re(): 根据传入的正则表达式对数据进行提取，返回unicode字符串list列表。**

**4.yield的用法**

**带有 yield 的函数不再是一个普通函数，而是一个生成器generator，可用于迭代，工作原理同上。**

**yield 是一个类似 return 的关键字，迭代一次遇到yield时就返回yield后面(右边)的值。重点是：下一次迭代时，从上一次迭代遇到的yield后面的代码(下一行)开始执行。**

**简要理解：yield就是 return 返回一个值，并且记住这个返回的位置，下次迭代就从这个位置后(下一行)开始。**

**5.判断为空**

**if nexturl is not None:**

**pass**

**6.pillow**

**PIL：Python Imaging Library，已经是Python平台事实上的图像处理标准库了。PIL功能非常强大，但API却非常简单易用。**

**由于PIL仅支持到Python 2.7，加上年久失修，于是一群志愿者在PIL的基础上创建了兼容的版本，名字叫Pillow，支持最新Python 3.x，又加入了许多新特性，因此，我们可以直接安装使用Pillow。**

**7.23种设计模式**

**简单工厂模式**

**1.工厂方法模式 2.抽象工厂模式 3.单例模式 4.建造者模式 5.原型模式**

**6.适配器模式 7.装饰模式 8.代理模式 9.外观模式 10.桥接模式**

**11.组合模式 12.享元模式 13.策略模式 14.模版方法模式 15.观察者模式**

**16.迭代子模式 17.责任链模式 18.命令模式 19.备忘录模式 20.状态模式**

**21.访问者模式 22.中介者模式 23.解释器模式**

2017-11-25  
<h1>1.编程笔记</h1>  
1.github.com 和 github.io的区别是为了禁止cookies的攻击，不同域名限制cookies攻击。  
github.io 存放pages服务。  
  
2.github.io 建立博客<br/>  
jetyll 生成html md文件<br/>  
liquid 模版引擎<br/>  
  
3.会话回放脚本<br/>  
监控网站的操作记录，一种Linux脚本  
  
<h1>2.英语学习</h1>  
specialize 专门从事 专攻<br/>  
pitch 投掷 扎营<br/>  
vital 必须的 必不可少的<br/>  
  
<h1>3.微语录</h1>  
在商业的经营中，以江湖身份进入，以商人身份退出。避免伤和气。

**成长笔记**

**github.io 个人博客**

**liquid 模版引擎，使用双中括号{{}}**

**jetyll 生成html md文件**

**建立自己的博客**

**记录学习笔记**

**vi pull 拉去最新**

**软件设计模式 工厂模式**

**扩展**

绘图工具：matplotlib  
好的项目：今日头条网易腾讯新闻 NewsSpider

python 经验

python zlib 压缩文件