

#### 主要内容:

- 网络爬虫基本概念
- · requests 库
- urllib库
- **beautifulsoup4**库
- re库



#### 网络爬虫基本概念:

Clar

沿为首面 手机就泊网

移动客户端 🗸

受录 微博 → 博習

№箱 🗸 网站导航

⊠关

**参** Sina新浪网 sina.com.cn

文庫 大家正在搜: 巴西总统称自己已!

超 博客 专栏 历史 天皇

时尚 女性 医药 育儿 教育 高考 公益 佛学 举报专区

微博 城市 上海 学投资 交易 旅游 文化 彩票 高尔夫 理财 游戏 手游 邮箱 English 更多

sina 新浪教育

新浪2020五星金牌<u>企</u> 教师评选开启

> 共赴 【() 年之约 等待热爱教育的每一个你 报名时间: 6月5日-7月15日



新浪2020万星金牌教师评选》

新浪产品

< >>

#### 新浪众测

众测问答帮帮团上线 魔爪稳定器百台试用



が哈: 数量: 100 免费申请

考 热卖 **新款** 淘



图片 专栏 热点







- · 🔀 英国威尔士3家肉食加工厂出现聚集性新冠
- ・図 比利时将进入"解封"第四阶段
- ·図 杨洋西装造型绅士清爽 戚薇卷发红唇气场

视频 综艺



- ・□ 世卫点赞中国新冠疫苗
- ⇨ 武汉以各式活动欢度端
- 🖼 南宁遭遇史上最强暴雨
- 第日後後北水鉄川会面
- ※ 衆国政府並然前側家門□ 第二起不让告告事前主
- · 🖼 孙杨就禁赛提出上诉
- ・ 👨 国丘会唱《我姿像由

猜你喜欢 🔇 • • • • 🌖

- •1分钟可擦地1000次,米家无线手持擦地机分
- ・800万"后浪"求职季: 有人先就业再择业 有
- · 频发虐狗视频 虐狗狂引公愤
- ・2020年上海中考语文写作题目公布 附17年写
- •玻璃打磨机"咬住"工人手臂
- 天津这个区叒添一处生态好去处
- 声网Agora美国上市: 首日涨153% 市值超50亿
- · 孩士探切另士所马应医领型获 义者。孩士教 。本主动会6年早悔氏独选出 俚理108万90年期

新闻 看上海

2020. 6. 27

- 全面小康 一个少数民族也不能少 理上网
- 暴雨!暴雨!暴雨!受灾群众怎么样了 脱
- 专访:多数外商真心欢迎涉港国安立法
- •美因涉港问题对中方官员实施签证限制 我使
- 昨日北京新增17例确诊 丰台15例 大兴2例 疫
- 我驻加拿大使馆就孟晚舟事件发表谈话
- •印度陆军司令向国防部长:在中印实控线做长
- ・陈方安生宣布会退出政坛 网友: 想走 没那么
- •香港教育局局长做出一个重要表态!
- ・校园现"港独"标语 港大却不出手
- ・湖北省政府领导再有调整
- •中国一总领事遭澳大利亚警方调查 居然是这
- ・默克尔: 或将不得不"从根本上思考"与美国
- ・不丹"断水" 印度媒体慌了
- •美国最大保健品公司破产!这家中国公司损失
- ・班主任带大汉跨省堵门 两次被顶替女子: 老
- ・□ 特朗普夸边境墙阻止新冠传播:多亏它杜绝
- ・孕妈突发晕厥医护市民合力救助 5秒视频温暖
- ・财经 | 外商投资准入负面清单再压减17.5%
- □ 端午假期前两日国内游客达3707.7万人次
   1-5月22省份房地产投资重回正增长 西部省份
- •1节月22省份房地产投资重回正增长 四部省份 •股票 | 扇贝逃亡真相:证监会出动北斗卫星 獐
- 黄光裕的三张地产牌 潘伟明兄弟的千亿执念
- ·科技 | 瑞幸咖啡发布声明: 将于6月29日停牌
- ·美政府: 科技公司CEO雲确保社交平台不被用
- ·塑料瓶的"快乐水"为什么没内味了? GIF:
- ・汽车 | 48寸大屏梦碎一地 拜腾还没腾飞就要
- ・时尚 | 宋西教你学基础款 时尚博王川的显高・执占 | 和关略必二起失到启度 范录录的发展

(IDUCTYPE html
(I published at 2020-06-27 11:39:00 )
(Atml> (hand>

《meta name="keywords" content="新浪,新浪网,SINA, sina, sina, com. cm,新浪首页,门户,资讯(//)) 《meta name="description" content="新浪网为全球用户24小时提供全面及时的中文资讯,内容覆盖国内外突发新闻事件、体坛赛事、娱乐

动交流空间。" />

《meta content="always" name="referrer">
《link rel="mask-icon" sizes="any" href="//www.sina.com.cn/favicon.svg" color="red">
《meta name="stencil" content="PGLS000022" />
《meta name="publishid" content="30,131,1" />
《meta name="verify-v1" content="30+ PHTM-VATMOUQBI2TW8+CfkYCoyeB8IDbn8=" />
《meta name="application-name" content="新浪首页"/>

(meta name="msapplication=TileColor" content="新浪首页"/>
(meta name="msapplication=TileColor" content="//i1. sinaimg. cn/dy/deco/2013/0312/logo. png"/>
(meta name="msapplication=TileColor" content="#ffbf27"/>
(link rel="apple=touch=icon" href="//i3. sinaimg.cn/home/2013/0331/U586P30DT20130331093840.png" />

<script type="text/javascript">

//is异步加载管理 (function() [var w=this, d=document, version='1.0.7', data={}, length=0, cbkLen=0; if(w. jsLoader) {if(w. jsLoader.version)=version) {r [obj. attachEvent("on" +eventType, func)}else[obj. addEventListener(eventType, func, false)}];var\_domReady=false, ondomReady=function() <525) {doReadvStateCheck()}else{d.addEventListener("DOMContentLoaded", function(){d.removeEventListener("DOMContentLoaded", arguments) {return}; try{d. documentElement. doScroll("left")}catch(e){return}; ondomReady()}; function doReadyStateCheck(){if(domReady){return} {ondomReady(); return}else{setTimeout(doReadyStateCheck, 1); return}}; function createPosNode() {if(jsLoader, caller) {return}; cbkLen+ <del>id=</del>"jl\_pos\_'+obkLen+'"></div>');s=d.getElementById('\_jl\_pos\_'+obkLen)}catch(e){var\_s=d.createElement('div');s.id='\_jl\_pos\_'+obk s=d. createElement('div');s.id='jl\_pos'+cbkLen;s.style.display='none';d.body.appendChild(s)};return s};function getScript(url, div) {scriptNode.charset=charset}; scriptNode.onreadyStatechange=scriptNode.onload=function() {if(!this.readyState||this.readyState="1 {dispose()}; scriptNode. onreadystatechange=scriptNode. onload=null; scriptNode. parentNode. removeChild(scriptNode)}}; scriptNode. src= cWrite(str){if(posNode.childNodes.length>0){return};if(posNode.innerHTML!=''){while(posNode.childNodes.length){posNode.parentNod {posNode.parentNode.insertBefore(posNode.childNodes[0], posNode)}}; var JsObj=function(name, url) {this.name=name; this.url=url; this. i=0;i<this.callback.length;i++){if(typeof this.callback[i]='function'){try{if(this.callback[i].posNode){posNode=this.callback[i].posNode}} d.write=write:this.callback[i].posNode.parentNode.removeChild(this.callback[i].posNode)}}catch(e){errors.push(e)}};this.callba callback=cfg.callback||"";var.charset=cfg.charset||"";if(name){if(!data[name]){if(!url){data[name]=new JsObj(name);data[name].st {data[name].status='init'}:if(cfg.status){data[name].status=cfg.status};if(data[name].status='loading'||data[name].status='wai

if(data[name] status='ok') {if(typeof callback='function') {callback()} return}}else{if(!url) {return}; for(var item in data) {if(data[name] status='ok') {if(typeof callback='function') {callback()} return}}else{if(!url) {return}; for(var item in data) {if(data[name] status='ok') {if(typeof callback='function') {callback()} return}}else{if(!url) {return}; for(var item in data) {if(data[name] status='ok') {if(typeof callback='function') {callback()} return}}else{if(!url) {return}; for(var item in data) {if(data[name] status='ok') {if(typeof callback='function') {callback()} return}}else{if(!url) {return}; for(var item in data) {if(data[name] status='ok') {if(typeof callback='function') {callback()} return}}else{if(!url) {return}; for(var item in data) {if(data[name] status='ok') {if(data[name] status='ok

JsObj(name, url); length++}; if(data[name]. status='loading') {if(typeof callback='function') {callback.posNode=createPosNode(); data

{callback()};return}}; if(typeof callback='function') {callback.posNode=createPosNode();data[name].callback.push(callback)}; getSc{data[name].onload()}, charset);data[name].status='loading'}; w.jsLoader.version=version; w.jsLoader.getData=function(){return data

(/script)

<!--iplook接口如果故障,首页舌签切换将失效,此为iplook接口的容错,默认为北京市一> <soript>
//空对象返回true

#### requests 库

Oflow

• requests 库是一个简洁的处理HTTP请求的第三方库,它的程序编写过程更接近正常URL 访问过程。

In [6]: import requests

In [7]: r=requests(get("http://www.baidu.com")

In [8]: type(r)

Out[8]: requests.models.Response

Income?

7632

渡一教旨 Duyi Educatio

### requests 库 response的属性

| 属性                 | 描述                                 |
|--------------------|------------------------------------|
| status_code        | HTTP请求返回状态,200表示正常,404表示失败         |
| text               | HTTP响应内容                           |
| encoding           | HTTP响应编码方式                         |
| content            | HTTP响应的二进制方式                       |
| json               | 如果HTTP响应内容包含JSON数据,该方法解析<br>JSON数据 |
| raise_for_status() | 如果status_code不是200,则报异常。           |



#### requests 库

```
In [20]: response=requests.get('http://www.baidu.com')

In [20]: response.text

Out[20]: '<!DOCTYPE html>\r\n<!--STATUS OK--><html> <head><meta http-equiv=content-type
content=text/html;charset=utf-8><meta http-equiv=X-UA-Compatible content=IE=Edge><meta
content=always name=referrer><link rel=stylesheet type=text/css href=http://s1.bdstatic.com/r/www/
cache/bdorz/baidu.min.css><title>ç\x99¼åº!ä\x80ä\x80ä\x80ä¼\x8cä¼\x80å°tç\x9f¥é\x81\x93</title></head>
<body link=#0000cc> <div id=wrapper> <div id=head> <div class=head_wrapper> <div class=s_form> <div
class=s_form_wrapper> <div id=lg> <img hidefocus=true src=//www.baidu.com/img/bd_logo1.png
width=270 height=129> </div> <form id=form name=f action=//www.baidu.com/s class=fm> <input
type=hidden name=bdorz_come value=1> <input type=hidden name=ie value=utf-8> <input type=hidden
name=f value=8> <input type=hidden name=rsv_bp value=1> <input type=hidden name=rsv_idx value=1>
<input type=hidden name=tn value=baidu><span class="bg s_ipt_wr"><input id=kw name=wd class=s_ipt
value maxlength=255 autocomplete=off autofocus></span><span class="bg s_btn_wr"><input type=submit
id=su_value=c\x00¼åº!ä\x80ä\x80 class="bg s_btn"></span> </form> </f
```

In [22]: response.encoding='utf-8'

In [22]: response.text

Out[22]: '<!DOCTYPE html>\r\n<!--STATUS OK--><html> <head><meta http-equiv=content-type
content=text/html;charset=utf-8><meta http-equiv=X-UA-Compatible content=IE=Edge><meta
content=always name=referrer><link rel=stylesheet type=text/css href=http://s1.bdstatic.com/r/www/
cache/bdorz/baidu.min.css><title>百度一下,你就知道</title></head> <body link=#0000cc> <div
id=wrapper> <div id=head> <div class=head\_wrapper> <div class=s\_form> <div class=s\_form\_wrapper>
<div id=lg> <img hidefocus=true src=//www.baidu.com/img/bd\_logo1.png width=270 height=129> </div>
<form id=form name=f action=//www.baidu.com/s class=fm> <input type=hidden name=bdorz\_come value=1>
<input type=hidden name=ie value=utf-8> <input type=hidden name=f value=8> <input type=hidden
name=rsv\_bp value=1> <input type=hidden name=tn
value=baidu><span class="bg s\_ipt\_wr"><input id=kw name=wd class=s\_ipt value maxlength=255
autocomplete=off autofocus></span><span class="bg s\_btn\_wr"><input type=submit id=su value=百度一下



### requests 库 post 方法

```
90°C 200 = 200-
```

```
In [25]: data = { 'key1': 'value1', 'key2': 'value2' }
In [26]: r = requests.post('http://httpbin.org/post', data)
In [27]: r.status_code
Out[27]: 200
```



hep. (hlob)

GET和POST两种基本请求方法的区别



Urllib库

requests
in its tare

• Urllib是Python提供的一个用于操作URL的模块,我们爬取网页的时候,经常需要用到这个库。

#### Urllib库

Vancol

- urllib.request---请求模块,用于发起网络请求
- urllib.parse-解析模块,用于解析URL
- urllib.error---异常处理模块,用于处理request引起的异常



#### urllib.request模块

request模块主要负责构造和发起网络请求,并在其中添加Headers, Proxy等。 利用它可以模拟浏览器的请求发起过程。

- 发起网络请求
- 添加Headers



12 BAR

• HTML 建立的Web 页面一般非常复杂,除了有用的内容信息外,还包括大量用于页面格式的元素,直接解析一个Web 网页需要深入了解HTML 语法,而且比较复杂。beautifulsoup4 库将专业的Web

页面格式解析部分封装成函数,提供了若干有用且便捷的处理函数。

Sheul

@ a a s

- 使用requests 库获取HTML 页面并将其转换成字符串后,需要进一步解析HTML页面格式,提取有用信息 ,这需要处理HTML 和XML 的函数库。
- beautifulsoup4 库,也称为Beautiful Soup 库或bs4 库,用于解析和处理HTML和XML。



• beautifulsoup4 库采用面向对象思想实现,简单说,它把每个页面当做一个对象,通过<a>.<b>的方式调用对象的属性(即包含的内容),或者通过<a>.<b>()的方式调用方法(即处理函数)。



| 属性    | 描述  |
|-------|---|
| head  | HTML页面的 <head>内容</head>   |
| title | HTML页面标题 <title>&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;body&lt;/td&gt;&lt;td&gt;HTML页面&lt;br/&gt;body&gt;&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;р&lt;/td&gt;&lt;td&gt;HTML页面第一个内容&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;strings&lt;/td&gt;&lt;td&gt;HTML页面所有标签的内容&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;stripped_strings&lt;/td&gt;&lt;td&gt;HTML页面所有非空格字符串&lt;/td&gt;&lt;/tr&gt;&lt;/tbody&gt;&lt;/table&gt;</title> |



#### BeautifulSoup 对象的常用属性

- 其中,尖括号(<>)中的标签的名字是name,尖括号内其他项是attrs,尖括号之间的内容是string。因此,可以通过Tag对象的name、attrs和string属性获得相应内容,采用<a>.<b>的语法形式。
- 标签Tag 有4 个常用属性



| 属性       | 描述               |
|----------|------------------|
| name     | 字符串,标签的名字        |
| attrs    | 字典,包含了原有页面tag的属性 |
| contents | 列表, 这个tag下的所有内容  |
| strings  | tag中包含的文本        |



#### find\_all方法查找所有

# 268 Parte la

BeautifulSoup find\_all (name, attrs, recursive, string, limit)

作用:根据参数找到对应标签,返回列表类型。

参数:

name:按照 Tag 标签名字检索,名字用字符串形式表示,例如:div,li;

attrs:按照 Tag 标签属性值检索,需要列出属性名称和值,采用 JSON 表示;

recursive: 设置查找层次,只查找当前标签下一层时使用 recursive=False;

string: 按照关键字检索 string 属性内容,采用 string=开始;

limit: 返回结果的个数,默认返回全部结果。



#### re库

#### 什么是正则表达式:

正则表达式,又称规则表达式,通常被用来检索,替换那些符合某个模式(规则)的文本。

正则表达式是对字符串操作的一种逻辑公式,就是用事先定义好的一些特定字符、及这些特定字符的组合,组成一个"规则字符串",这个"规则字符串"用来表达对字符串的一种过滤逻辑。



常用操作符(1)







| 属性       | 描述             |                               |
|----------|----------------|-------------------------------|
| 0        | 表示任意单个字符       |                               |
| []       | 字符集            | [abc]表示a、b、c, [a-z]表示a到z的单个字符 |
| [^]      | 非字符集           | (Yabc)表示 非a非b非c的单个字符          |
|          | 前一个字符0次或者无限次扩展 | abc*表示 ab, abc abcc abccc等    |
| $\Theta$ | 前一个字符1次或者无限次扩展 | abc→表示 abcc, abccc等           |
| ?        | 前一个字符1次或者0次扩展  | abc? 表示 abc 或者ab              |
| 1        | 左右表达式任意一个      | abc def 表示 abc 或者 def         |
|          |                |                               |
|          |                |                               |

#### 常用操作符的组合



| 操作符    | 描述           | In [41]:    |
|--------|--------------|-------------|
| *?     | 重复任意次,尽量少重复  | In [42]:    |
| +?     | 重复1次以上,尽量少重复 | Out[42]:    |
| ? ?    | 重复0到1次,尽量少重复 | In [43]:    |
| {n,m}? | 重复n到m次,尽量少重复 | In [44]:    |
| {n,}?  | 重复n次以上,尽量少重复 | Out[44]:    |
|        | Japan Path   | rezino Bold |

```
In [41]: ls=re.findall("a.*b","abcdb")
In [42]: ls
Out[42]: ['abcdb']
In [43]: ls=re.findall("a.*?b","abcdb")
In [44]: ls
Out[44]: ['ab']
```

#### 常用操作符(2)

8 3

| 1000 | -2 |
|------|----|
|      |    |
|      |    |

# A Ca-23864C

| 属性          | 描述                   |                           |
|-------------|----------------------|---------------------------|
| {m}         | 扩展一个字符m次             | Ab{4}c 表示 Abbbbc          |
| {m,n}       | 扩展一个字符m至n次           | Ab{2,4}c 表示 Abbc, Abbbc   |
| $\bigcirc$  | 匹配字符串开头              | ^abc 表示以abc 开头的字符串        |
| <b>(\$)</b> | 匹配字符串结尾              | \$abc 表示以abc 结尾 约字符串      |
| <b>(</b>    | 前一个字符1次或者无限次扩展       | abc+ 表示 abc, abcc, abccc等 |
| 0           | 前一个字符1次或者0次扩展        | abc? 表示 abc 或者ab          |
| <u>/a</u>   | 数字,等价于[0-9]          |                           |
| \w          | 单词字符,等价于[a-zA-Z0-9-) | 1th                       |
|             |                      |                           |



#### re.match

- re.match(pattern, string)

```
In [49]: re.match('\d{3}', abc123,444hhh,weather0000')
```

```
In [50]: re.match('\d{3}','123abc,444hhh,weather0000')
```

```
Out[50]: kre.Match object; span=(0, 3), match='(123')>
```



#### 邮箱的正则表达式



• ^[a-zA-Z0-9\_]+@[a-zA-Z0-9\_-]+(\.[a-zA-Z0-9\_-]+)+\$



```
In [36]: re.match(r'^[a-zA-Z0-9 -]+@[a-zA-Z0-9 -]+(\.[a-zA-Z0-9 -]+)+
$','ilovepython@qq.com')
Out[36]: <re.Match object; span=(0, 18), match='ilovepython@qq.com'>
In [37]: re.match(r'^[a-zA-Z0-9_-]+@[a-zA-Z0-9_-]+(\.[a-zA-Z0-9_-]+)+
$','ilove python@qq.com')
In [38]: re.match(r'^[a-zA-Z0-9 -]+@[a-zA-Z0-9 -]+(\.[a-zA-Z0-9 -]+)+
$',' python@qq.com')
Out[38]: <re.Match object; span=(0, 15), match='__python@qq.com'>
In [39]: re.match(r'^[a-zA-Z0-9 -]+@[a-zA-Z0-9 -]+(\.[a-zA-Z0-9 -]+)+
$','__python@qq.com.cn')
Out[39]: <re.Match object; span=(0, 18), match='__python@qq.com.cn'>
In [40]: re.match(r'^[a-zA-Z0-9_-]+@[a-zA-Z0-9_-]+(\.[a-zA-Z0-9_-]+)+
$','__python@qq.com.cn.org')
Out[40]: <re.Match object; span=(0, 22), match='__python@qq.com.cn.org'>
In [41]: re.match(r'^[a-zA-Z0-9_-]+@[a-zA-Z0-9_-]+(\.[a-zA-Z0-9_-]+)+
$','__python@.org')
```



#### re.search

re.search(pattern, string)

· 在一个字符串中搜索匹配正则表达式的第一个位置,返回match对象

·pattern:正则表达式的字符串或原生字符串表示

·string:待匹配字符串

In [48]: re.search('\d{3}','abc123,444hhh,weather0000').group(0)
Out[48]: '123'



#### re.findall

re.findall(pattern, string)

搜索字符串,以列表类型返回全部能匹配的子串

- ·pattern:正则表达式的字符串或原生字符串表示
- ·string:待匹配字符串

```
In [24]: import re
In [25]: ls=re.findall(r'\d{6}','ABCDE1000000, FGHIJG234567')
In [26]: ls
Out[26]: ['100000', '234567']
In [27]: ls=re.findall(r'\d{5}','ABCDE1000000, FGHIJG234567')
In [28]: ls
Out[28]: ['10000', '23456']
In [29]: ls=re.findall(r'\d{3}),'ABCDE100000, FGHIJG234567')
In [30]: ls
Out[30]: ['100', '000', '234', '567']
```



#### compile 函数

• compile 函数用于编译正则表达式,生成一个 Pattern 对象,它的一般使用形式如下:

```
In [53]: pattern=re.compile('\d{3}')
In [54]: re.match(pattern,'123abc,444hhh,weather00000')
Out[54]: <re.Match object; span=(0, 3), match='123'>
```



案例一: 从电影页面爬取豆瓣评论数据



案例二:从百度图片爬取图片



# 作业

·编写一个电影评论爬虫程序,要求用户能够输入电影名称,然后将评论和评级存入一个txt文件。

