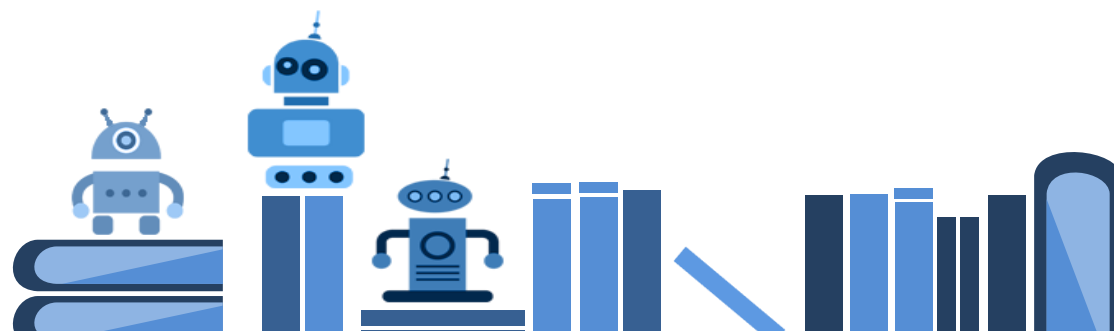




# HTML文本解析

主讲老师：张涛



1

Requests介绍

2

Xpath解析数据

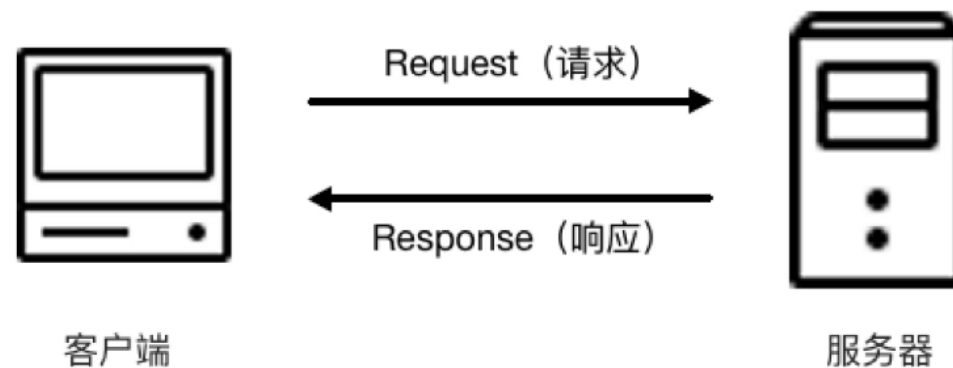
3

正则表达式

# 目录



回顾一下访问网页的过程：浏览器发送请求给网站服务器，网站服务器响应浏览器的请求，将网页发送给浏览器显示。



Python如何实现将Request请求发送给服务器呢？





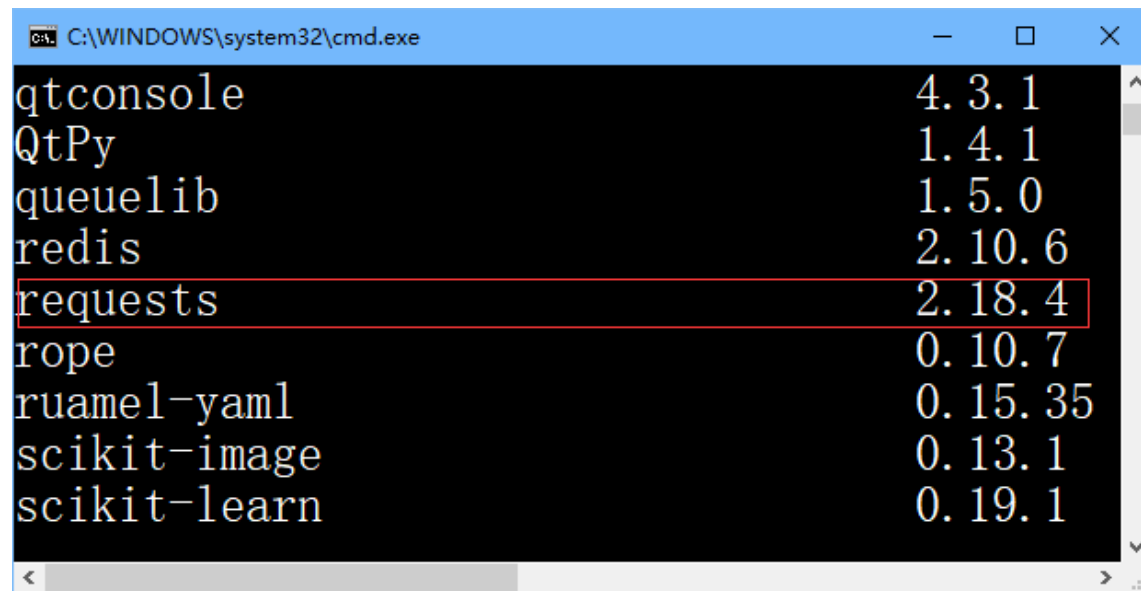
Requests是Kenneth Reitz编写的一个优雅、易用的HTTP库。Requests的底层基于Python官方库urllib，但Requests良好的API设计更适合人类使用。

Requests最核心的两个类：

- ◆ Request: 对 HTTP 请求的封装
- ◆ Response: 对 HTTP 返回结果的封装



- ◆ Windows下: pip install requests
- ◆ Linux下: pip install requests



```
C:\WINDOWS\system32\cmd.exe

qtconsole          4.3.1
QtPy               1.4.1
queuelib           1.5.0
redis              2.10.6
requests           2.18.4
rope                0.10.7
ruamel-yaml         0.15.35
scikit-image       0.13.1
scikit-learn       0.19.1
```



当通过Requests库获取到HTML代码后，就需要从页面中获取想要的數據了。解析数据方式通常有三种：

- ◆ 正则表达式
- ◆ Xpath
- ◆ BeautifulSoup





网络爬虫最核心、最重要的工作是从HTML文档中将想要的的数据提取出来，而使用XPath就可以很轻松地解析HTML文档信息。

XPath概念：

XPath全称**XML Path Language**，即XML路径语言。它是一门在XML文档中查找信息的语言。HTML与XML结构类似，也可以在HTML中查找信息。

XPath 使用路径表达式来选取HTML文档中的节点或者节点集。这些路径表达式和我们在常规的电脑文件系统中看到的表达式非常相似。





#### ➤ 安装lxml库

```
pip install lxml
```

➤ `from lxml import etree`

➤ `Selector=etree.HTML(网页源代码)`







XPath 使用路径表达式来选取HTML文档中的节点或者节点集

常用的XPath路径表达式如下表所示：

表达式	描述	示例
<b>nodename</b>	选取此节点的所有子节点	div,p,h1
<b>/</b>	从根节点选取（描述绝对路径）	/html
<b>//</b>	不考虑位置，选取页面中所有子孙节点	//div
<b>.</b>	选取当前节点（描述相对路径）	./div
<b>..</b>	选取当前节点的父节点（描述相对路径）	h1/..
<b>@属性名</b>	选取属性的值	@href,@id
<b>text()</b>	获取元素中的文本节点	//h1/text()



## 带谓语的XPath路径表达式

谓语表达式	说明
<code>//div[@id='content']</code>	选取属性id为content的div元素
<code>//div[@class]</code>	选取所有带有属性class的div元素
<code>//div/p[1]/text()</code>	选取div节点中的第一个p元素的文本
<code>//div/p[2]/text()</code>	选取div节点中的第二个p元素的文本
<code>//div/p[last()]/text()</code>	选取div节点中的最后一个p元素的文本

练习：使用XPath实现豆瓣电影信息提取2。





- ✓ 正则表达式是一个特殊的字符序列，它能帮助你方便的检查一个字符串是否与某种模式匹配。
- ✓ 如果说网页爬虫爬取的网页信息是数据大海的话，正则表达式就是我们进行“大海捞针”的工具。



正则表达式的组件可以是单个的字符、字符集合、字符范围、字符间的选择或者所有这些组件的任意组合。

符号	描述	符号	描述
\w	匹配字母、数字、下划线	.	匹配任意字符，包括汉字
\W	匹配不是字母、数字、下划线的字符	[m]	匹配单个字符串
\s	匹配空白字符	[m1m2...n]	匹配多个字符串
\S	匹配不是空白的字符	[m-n]	匹配m到n区间内的数字、字母
\d	匹配数字	[^m]	匹配除m以外的字符串
\D	匹配非数字的字符	()	对正则表达式进行分组，一对圆括号表示一组
*	重复0或N次	{m}	重复m次
+	重复1或N次	{m,n}	该限定符的意思是至少有 m 个重复，至多到 n 个重复
?	重复0或1次		



## 网络爬虫用得最多的正则表达式：

- . : 匹配任意字符，换行\n除外
- \* : 匹配前一个字符0次或无限次
- ? : 匹配前一个字符0次或一次
- .\* : 贪心算法
- .\*? : 非贪心算法
- () 内的数据作为结果输出



Python中自带re模块，可以使用re模块来直接调用正则表达式来实现正则匹配。

**1.re.findall函数：**匹配所有符合规则的内容，返回包含结果的列表。

```
re.findall(pattern, string, flags=0)
```

参数	描述
pattern	匹配的正则表达式
string	要被查找的原始字符串
flags	标志位，用于控制正则表达式的匹配方式，如：是否区分大小写，多行匹配等



Python中自带re模块，可以使用re模块来直接调用正则表达式来实现正则匹配。

**3. re.sub函数：**用于替换字符串中的匹配项。返回替换后的值。

```
re.sub(pattern, repl, string, count=0, flags=0)
```

参数	描述
pattern	匹配的正则表达式
repl	替换的字符串，也可为一个函数
string	要被查找替换的原始字符串
count	模式匹配后替换的最大次数，默认0表示替换所有的匹配
flags	标志位，用于控制正则表达式的匹配方式，如：是否区分大小写，多行匹配等





**结束 谢谢收看**

