

9.4 网络爬虫

- 随着网络的迅速发展，万维网成为大量信息的载体，如何有效地提取并利用这些信息成为一个巨大的挑战。
- Python有一个非常著名的HTTP库requests，现在requests库的作者又发布了一个新库，叫做requests-html，新的库把网页抓取和信息提取两个功能合二为一。
- 安装requests-html非常简单，一行命令即可做到。需要注意的是，requests-html全部功能只支持Python 3.6及以后的版本。。

Requests-html模块功能

- ◎ **Full JavaScript support!**
- ◎ *CSS Selectors* (a.k.a jQuery-style, thanks to PyQuery).
- ◎ *XPath Selectors*, for the faint at heart.
- ◎ Mocked user-agent (like a real web browser).
- ◎ Automatic following of redirects.
- ◎ Connection—pooling and cookie persistence.
- ◎ The Requests experience you know and love, with magical parsing abilities.
- ◎ **Async Support**

模块的方法

- ④ >>> import requests_html
- ④ >>> dir(requests_html)
- ④ ['AsyncHTMLSession', 'BaseParser', 'BaseSession', 'Cleaner', 'DEFAULT_ENCODING', 'DEFAULT_NEXT_SYMBOL', 'DEFAULT_URL', 'DEFAULT_USER_AGENT', 'Element', 'HTML', 'HTMLResponse', 'HTMLSession', 'HtmlElement', 'List', 'MaxRetries', 'MutableMapping', 'Optional', 'PyQuery', 'Result', 'Set', 'ThreadPoolExecutor', 'TimeoutError', 'Union', 'UserAgent', '_Attrs', '_BaseHTML', '_Containing', '_DefaultEncoding', '_Encoding', '_Find', '_HTML', '_LXML', '_Links', '_Next', '_NextSymbol', '_RawHTML', '_Result', '_Search', '_Text', '_URL', '_UserAgent', '_XPath', '__builtins__', '__cached__', '__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__', '_get_first_or_list', 'asyncio', 'cleaner', 'etree', 'findall', 'html_to_unicode', 'lxml', 'lxml_html_tostring', 'parse_search', 'partial', 'pyppeteer', 'requests', 'soup_parse', 'sys', 'urljoin', 'urlparse', 'urlunparse', 'user_agent', 'useragent']
- ④ >>>

获取网页

◎ 获取 “etf50.pythonanywhere.com” 的主页，只需五行代码

```
from requests_html import HTMLSession

session = HTMLSession()    #创建会话

url='https://etf50.pythonanywhere.com' #设置网址

r = session.get(url) #返回获取的网页对象

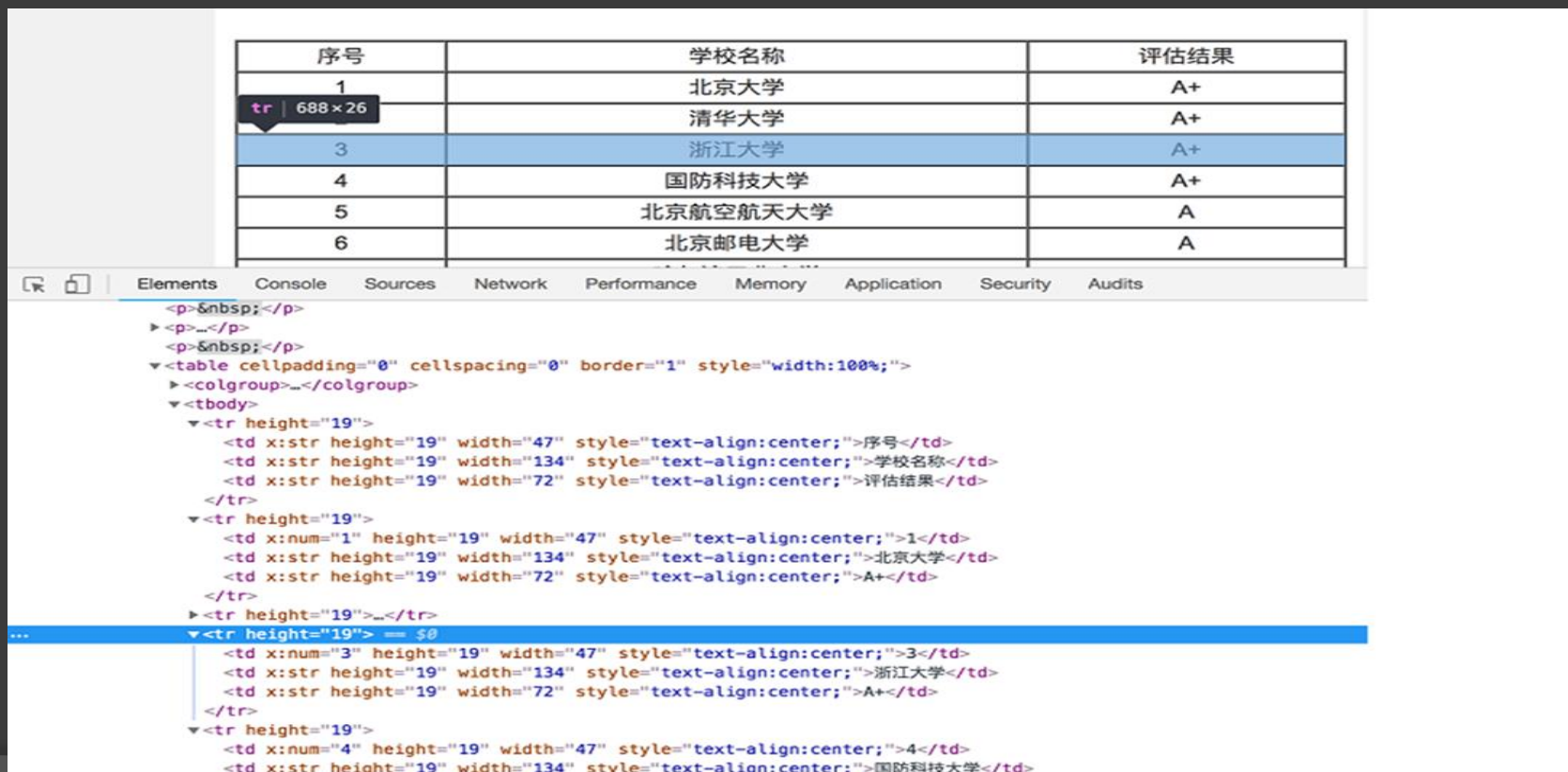
print(r.html.html) #调用 r 的 html 方法获取网页
```

核心功能---r.html

- >>> dir(r.html)
- ['__aiter__', '__anext__', '__class__', '__delattr__', '__dict__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattr__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__', '__lt__', '__module__', '__ne__', '__new__', '__next__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', '__weakref__', '_async_render', '_encoding', '_html', '_lxml', '_make_absolute', '_pq', 'absolute_links', 'add_next_symbol', 'arender', 'base_url', 'default_encoding', 'element', 'encoding', 'find', 'full_text', 'html', 'links', 'lxml', 'next', 'next_symbol', 'page', 'pq', 'raw_html', 'render', 'search', 'search_all', 'session', 'skip_anchors', 'text', 'url', 'xpath']

获取网页内容

- ◎ 教育部2017-2018计算机科学与技术专业大学排名。
- ◎ “<https://www.dxsbb.com/news/7566.html>”，在chrome选择开发者工具，



The image shows a screenshot of a web browser displaying a table of university rankings. The table has three columns: 序号 (Serial Number), 学校名称 (School Name), and 评估结果 (Evaluation Result). The rows list universities such as 北京大学 (Peking University), 清华大学 (Tsinghua University), 浙江大学 (Zhejiang University), 国防科技大学 (National University of Defense Technology), 北京航空航天大学 (Beihang University), and 北京邮电大学 (Beihang University). Below the table, the browser's developer tools are open, showing the HTML source code for the table. The code includes a table element with attributes like cellpadding, cellspacing, border, and style, and a tbody containing the table rows with their respective data and styling attributes.

序号	学校名称	评估结果
1	北京大学	A+
2	清华大学	A+
3	浙江大学	A+
4	国防科技大学	A+
5	北京航空航天大学	A
6	北京邮电大学	A

```
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<table cellpadding="0" cellspacing="0" border="1" style="width:100%;">
  <tbody>
    <tr height="19">
      <td x:str height="19" width="47" style="text-align:center;">序号</td>
      <td x:str height="19" width="134" style="text-align:center;">学校名称</td>
      <td x:str height="19" width="72" style="text-align:center;">评估结果</td>
    </tr>
    <tr height="19">
      <td x:num="1" height="19" width="47" style="text-align:center;">1</td>
      <td x:str height="19" width="134" style="text-align:center;">北京大学</td>
      <td x:str height="19" width="72" style="text-align:center;">A+</td>
    </tr>
    <tr height="19">
      <td x:num="2" height="19" width="47" style="text-align:center;">2</td>
      <td x:str height="19" width="134" style="text-align:center;">清华大学</td>
      <td x:str height="19" width="72" style="text-align:center;">A+</td>
    </tr>
    <tr height="19">
      <td x:num="3" height="19" width="47" style="text-align:center;">3</td>
      <td x:str height="19" width="134" style="text-align:center;">浙江大学</td>
      <td x:str height="19" width="72" style="text-align:center;">A+</td>
    </tr>
    <tr height="19">
      <td x:num="4" height="19" width="47" style="text-align:center;">4</td>
      <td x:str height="19" width="134" style="text-align:center;">国防科技大学</td>
      <td x:str height="19" width="72" style="text-align:center;">A+</td>
    </tr>
    <tr height="19">
      <td x:num="5" height="19" width="47" style="text-align:center;">5</td>
      <td x:str height="19" width="134" style="text-align:center;">北京航空航天大学</td>
      <td x:str height="19" width="72" style="text-align:center;">A</td>
    </tr>
    <tr height="19">
      <td x:num="6" height="19" width="47" style="text-align:center;">6</td>
      <td x:str height="19" width="134" style="text-align:center;">北京邮电大学</td>
      <td x:str height="19" width="72" style="text-align:center;">A</td>
    </tr>
  </tbody>
</table>
```

获取排名程序

```
from requests_html import HTMLSession

session = HTMLSession()

url='https://www.dxsbb.com/news/7566.html'

r = session.get(url)

table=r.html.find('tbody>tr' )

for row in table[:21]:    #取前 20 行

    l=row.text.split()    #row.text 取 3 列，返回字符串。split 变为列表

    s=''

    for i in l:

        s=s+'{0:^14}'.format(i)

    print(s)
```

程序运行结果

序号	学校名称	评估结果
1	北京大学	A+
2	清华大学	A+
3	浙江大学	A+
4	国防科技大学	A+
5	北京航空航天大学	A
6	北京邮电大学	A
7	哈尔滨工业大学	A
8	上海交通大学	A
9	南京大学	A
10	华中科技大学	A
11	电子科技大学	A
12	北京交通大学	A-
13	北京理工大学	A-
14	东北大学	A-
15	吉林大学	A-
16	同济大学	A-
17	中国科学技术大学	A-
18	武汉大学	A-
19	中南大学	A-
20	西安交通大学	A-

验证码识别

https://login.bce.baidu.com

百度智能云

返回首页 English

云生态狂欢季

智能ABC 共拓未来

热门产品 1折起 助力领跑智能时代

立即抢购

百度账号

云账号(推广账号)

zhangjunyu1963

.....

kata

kata 换一张

登录

安全控件常见问题 立即注册 忘记密码

温馨提示：
使用百度推广帐号或新注册的云帐号可直接登录。

调用智能云

- ① 1. `pip install baidu-aip`
- ② 2. 创建账号和相关应用
- ③ <https://login.bce.baidu.com/>
- ④ 获取 `app_id`, `api_key`, `secret_key`
- ⑤ 3. 调用api

创建OCR应用

产品服务 / 文字识别 - 应用列表

应用列表

[+ 创建应用](#)

	应用名称	AppID	API Key	Secret Key	创建时间	操作
1	OCR示例	16668272	uVgVRFzmZIP7WZg8trhWMkV	***** 显示	2019-06-28 14:34:31	报表 管理 删除

< 1 >

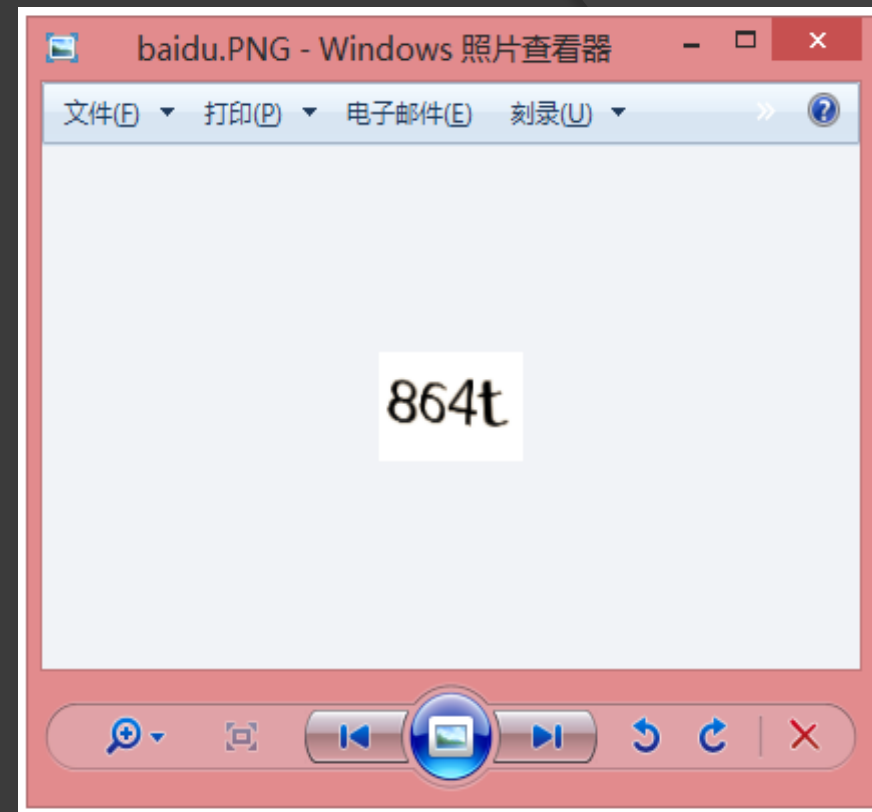
OCR程序

- ⦿ `app_id='16668272'`
- ⦿ `api_key='uVgVRFzImZIP7WZg8trhWMkV'`
- ⦿ `secret_key='*****'`

- ⦿ `from aip import AipOcr`
- ⦿ `client = AipOcr(app_id, api_key, secret_key)`

- ⦿ `f=open("baidu.png",'rb') #打开图形文件`
- ⦿ `image=f.read()`
- ⦿ `f.close()`

- ⦿ `dict1=client.general(image)`
- ⦿ `for i in dict1['words_result']:`
- ⦿ `print (i['words'])`



程序运行结果

```
1 app_id='16668272'
2 api_key='uVgVRFzlmZIP7WZg8trhWMkV'
3 secret_key='bvEq8DbG8lYWQ1t0lG6hjlENlZ8jGCLm'
4
5
6 from aip import AipOcr
7 client = AipOcr(app_id, api_key, secret_key)
8
9 f=open("baidu.png",'rb') #打开图形文件
10 image=f.read()
11 f.close()
12
13 dict1=client.general(image)
14 for i in dict1['words_result']:
15     print (i['words'])
```

Shell ×

Python 3.7.2 (bundled)

```
>>> %cd 'C:\Users\lenovo\Desktop\python的MOOC\第9章\OCR文字识别'
```

```
>>> %Run '3.6.0cr文字识别.py'
```

864t

```
>>> |
```