

# 北京林业大学

## 2022 学年—2023 学年第 2 学期 Python 应用 实验报告书

专 业： 大数据 班 级： 大数据 212

姓 名： 余睿捷 学 号： 211002328

实验地点： 机房 N07 任课教师： 王春玲

实验题目： 实验 3 爬取中国工程院院士信息

实验环境： Python、PyCharm 等

### 一、实验目的

1. 熟练使用标准库 urllib 读取网页内容。
2. 熟练使用正则表达式提取文本中感兴趣的信息。
3. 熟练使用内置函数 open() 创建文本文件和二进制文件。
4. 熟悉 HTML 语法以及常见的 HTML 标签。

### 二、实验内容

爬取中国工程院网页，把每位院士的简介保存为本地文本文件，并把每位院士的照片保存为本地图片，文本文件和图片文件都以院士的姓名为主文件名。

实验步骤如下：

(1) 使用 Google Chrome 或其他浏览器打开下面的网址，然后在页面上右击，在弹出的菜单中选择“查看网页源代码”。

[http://www.cae.cn/cae/html/main/col48/column\\_48\\_1.html](http://www.cae.cn/cae/html/main/col48/column_48_1.html)

(2) 分析网页源代码，确定每位院士的姓名和链接所在的 HTML 标签，为后面编写正则表达式做准备，如图 1 所示。

```
<li class="name_list"><a href="/cae/html/main/colys/63775817.html" target="_blank">曹喜滨</a></li>
<li class="name_list"><a href="/cae/html/main/colys/35791989.html" target="_blank">陈学东</a></li>
<li class="name_list"><a href="/cae/html/main/colys/01567139.html" target="_blank">邓宗全</a></li>
<li class="name_list"><a href="/cae/html/main/colys/25235806.html" target="_blank">丁荣军</a></li>
<li class="name_list"><a href="/cae/html/main/colys/46604755.html" target="_blank">董春鹏</a></li>
<li class="name_list"><a href="/cae/html/main/colys/42793697.html" target="_blank">樊会涛</a></li>
<li class="name_list"><a href="/cae/html/main/colys/15623030.html" target="_blank">冯培德</a></li>
<li class="name_list"><a href="/cae/html/main/colys/48072917.html" target="_blank">冯焯芳</a></li>
<li class="name_list"><a href="/cae/html/main/colys/71311121.html" target="_blank">甘晓华</a></li>
<li class="name_list"><a href="/cae/html/main/colys/15801057.html" target="_blank">高金吉</a></li>
```

图 1 每位院士的链接

(3) 使用浏览器打开任意一位院士的链接，然后查看并分析网页源代码，确定

简介信息和照片所在的 HTML 标签，为后面编写正则表达式做准备，如图 2 所示。

```
<div class="mdbg_w">
  <div class="wenz_md">
    <div class="right_md_top">当前位置: <a href="/cae/html/main/index.html" class="grey12">首页</a> > <a href="/cae/html/main/index.html" class="grey12">沈国防</a>
    <div class="right_md_name">沈国防</div>
    <div class="right_name_big_clearfix">
      <div class="info_img">
        <a href="http://ysg.ckeest.cn/html/details/461/index.html" target=" _blank">
          
        </a>
      <div class="cms_ysg_title"><a href="http://ysg.ckeest.cn/html/details/461/index.html" target=" _blank">沈国防院士百科</a></div>
    </div>
    <div class="intro">
      <p>&ensp;&ensp;&ensp;&ensp;&ensp;沈国防（1933.11.15- ）林学与生态学专家。出生于上海市，原籍浙江省嘉善县。1956年毕业于前苏联列宁格勒林学院，北京林业大学教授，</p>
    </div>
  </div>
</div>
```

图 2 院士个人简介和照片

(4) 编写代码，爬取信息并创建本地文件。

### 三、实验步骤及结果

```
import requests
from lxml import etree
import os

# 发送 HTTP 请求获取页面内容
url = 'https://www.cae.cn/cae/html/main/col48/column_48_1.html'
response = requests.get(url)
html = response.content.decode('utf-8')
# 解析 HTML
tree = etree.HTML(html)

# 获取每位院士的姓名和链接
academicians = {}
academicians_elem =
tree.xpath('/html/body/div[3]/div/div[2]/div/div[2]/div/ul/li[@class="name_list"]/a')
for elem in academicians_elem:
    name = elem.text
    link = elem.get('href')
    # academicians.append({'name': name, 'link': link})
    academicians[name] = link

# 遍历每位院士的链接，获取简介和照片
for name, link in academicians.items():
    # 发送 HTTP 请求获取院士页面内容
    response = requests.get('https://www.cae.cn/' + link)
    html = response.content.decode('utf-8')
    tree = etree.HTML(html)

    # 获取院士简介
    # intro = tree.xpath('//div[@class="intro"]/p/text()')
    intro = tree.xpath('/html/body/div[3]/div/div[3]/div[2]/p/text()')
```

```

intro = ".join(intro).strip()































# 获取院士照片
# img_url = tree.xpath('//div[@class="intro"]/div[@class="pic"]/a/img/@src')[0]
img_url = tree.xpath('/html/body/div[3]/div/div[3]/div[1]/a/img/@src')[0]
img_content = requests.get("https://www.cae.cn/" + img_url).content

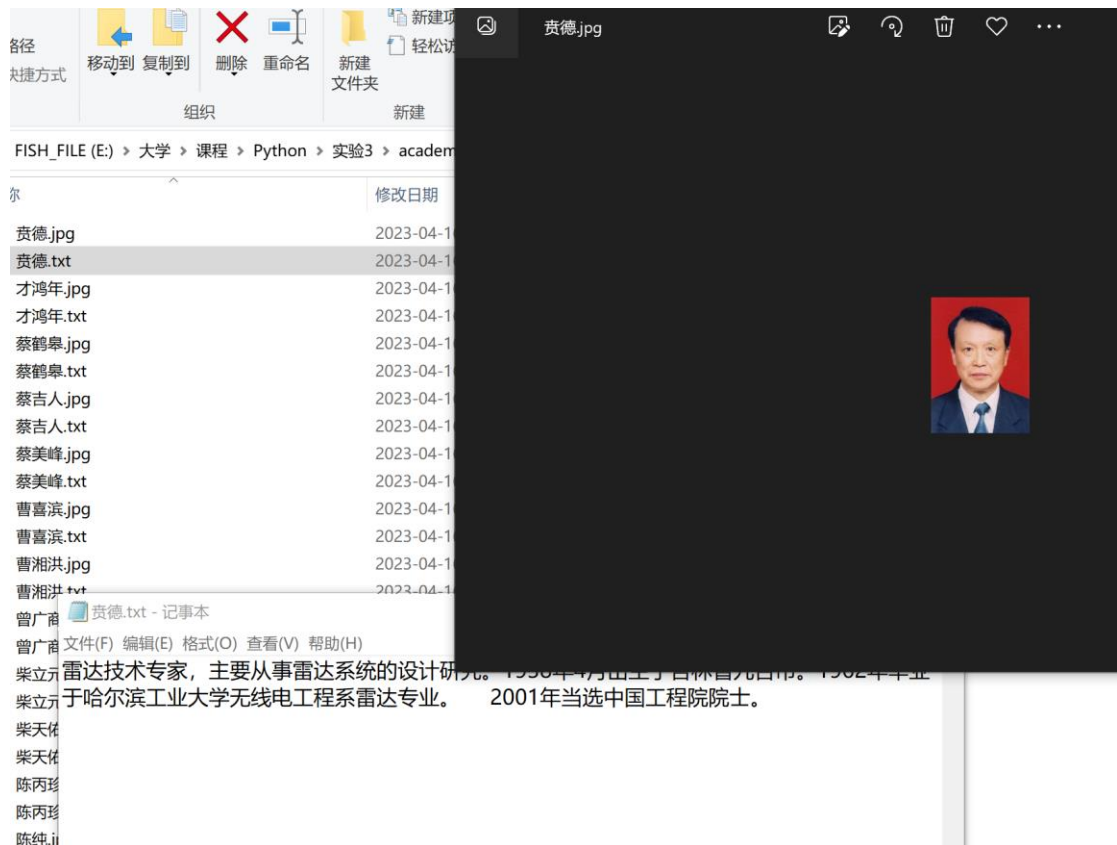
# 创建文件夹（如果不存在）
if not os.path.exists('academicians'):
    os.mkdir('academicians')

# 保存院士简介到文本文件
with open(f'academicians/{name}.txt', 'w', encoding='utf-8') as f:
    f.write(intro)
# 保存院士照片到图片文件
with open(f'academicians/{name}.jpg', 'wb') as f:
    f.write(img_content)

```

## 实验结果：

 曹喜滨.jpg	2023-04-16 18:27	JPG 文件	193 KB
 曹喜滨.txt	2023-04-16 18:27	文本文档	1 KB
 陈懋章.jpg	2023-04-16 18:28	JPG 文件	27 KB
 陈懋章.txt	2023-04-16 18:28	文本文档	1 KB
 陈学东.jpg	2023-04-16 18:27	JPG 文件	7 KB
 陈学东.txt	2023-04-16 18:27	文本文档	1 KB
 陈一坚.jpg	2023-04-16 18:28	JPG 文件	11 KB
 陈一坚.txt	2023-04-16 18:28	文本文档	1 KB
 陈予恕.jpg	2023-04-16 18:28	JPG 文件	24 KB
 陈予恕.txt	2023-04-16 18:28	文本文档	1 KB
 单忠德.jpg	2023-04-16 18:28	JPG 文件	301 KB
 单忠德.txt	2023-04-16 18:28	文本文档	1 KB
 邓宗全.jpg	2023-04-16 18:27	JPG 文件	10 KB
 邓宗全.txt	2023-04-16 18:27	文本文档	1 KB
 丁衡高.jpg	2023-04-16 18:28	JPG 文件	34 KB
 丁衡高.txt	2023-04-16 18:28	文本文档	1 KB
 丁荣军.jpg	2023-04-16 18:27	JPG 文件	28 KB
 丁荣军.txt	2023-04-16 18:27	文本文档	1 KB
 董春鹏.jpg	2023-04-16 18:28	JPG 文件	27 KB
 董春鹏.txt	2023-04-16 18:28	文本文档	1 KB
 杜善义.jpg	2023-04-16 18:28	JPG 文件	26 KB
 杜善义.txt	2023-04-16 18:28	文本文档	1 KB
 朵英贤.jpg	2023-04-16 18:28	JPG 文件	20 KB
 朵英贤.txt	2023-04-16 18:28	文本文档	1 KB
 樊会涛.jpg	2023-04-16 18:27	JPG 文件	640 KB
 樊会涛.txt	2023-04-16 18:27	文本文档	1 KB
 范本尧.jpg	2023-04-16 18:28	JPG 文件	38 KB
 范本尧.txt	2023-04-16 18:28	文本文档	1 KB
 冯培德.jpg	2023-04-16 18:28	JPG 文件	29 KB
 冯培德.txt	2023-04-16 18:28	文本文档	1 KB



#### 四、 实验分析

问题 1: 请求的 URL 缺少协议: MissingSchema: Invalid URL  
'/cae/html/main/colys/63775817.html': No scheme supplied. Perhaps you meant  
`https:///cae/html/main/colys/63775817.html?`  
解决方法: 将 `response = requests.get(link)` 改成 `response = requests.get('https://www.cae.cn/' + link)`, 即手动将 `'https://www.cae.cn/'` 补上。

问题 2: 简介如果直接获取会有 `['\u2002\u2002\u2002\u2002 ..... ', '\xa0', '\u2002\u2002\u2002\u2002\u2002 ..... ']`  
解决方法: 用 `".join(intro).strip()` 处理。

问题 3: 若直接通过 `xpath('/html/body/div[3]/div/div[2]/div/div[2]/div/ul/li/a')` 获取院士信息, 会将题头 abcd 那些便捷检索也收录进来。  
解决方法: 将 `li` 的属性 `class="name_list"` 加上: `xpath('/html/body/div[3]/div/div[2]/div/div[2]/div/ul/li[@class="name_list"]/a')`

问题 4: 一开始错将 `academicians = {}` 写成 `academicians = []`  
解决方法: 主要问题在于获取所有院士姓名和链接时, 代码中将字典 `academicians` 和列表 `academicians` 搞混了, 应该使用字典 `academicians` 来存储姓名和链接。此外, 在遍历字典时应该使用 `items()` 方法来同时获取键和值。

问题 5: XPath 表达式写法不够准确

解决方法： 写了两份 XPath 表达式，经测试都能使用，不过注释中的 XPath 更为稳定。

问题 6： 写完发现使用的是 request 库，而不是 urllib

解决方法： 详细了解两个库的区别，并在之后再写一份用 urllib 写的代码。

收获：

这次实验我意识到如果 XPath 表达式写法不准确，或文件名不合法都无法实现想要的理想结果：

XPath 表达式是用来解析 HTML 文档中的元素和属性的工具，如果写法不准确就可能无法正确获取到目标信息。如果官网的页面结构发生变化，例如元素的层级、类名、id 等属性变化，就需要修改 XPath 表达式。如果官网的 HTML 文档中存在多个相同的元素，例如多个 class 为“intro”的 div，XPath 表达式需要加上更具体的限定条件。

而如果院士的姓名包含一些非法字符（例如空格、/、\等），就可能导致文件名不合法，从而无法保存到本地文件夹中。这种情况可以使用 Python 中的一些字符串处理方法，例如 replace()、strip()等，将文件名中的非法字符替换成合法字符。

我还了解到 BeautifulSoup 和 lxml 都是 Python 中常用的解析 HTML 和 XML 的库，它们之间的主要区别在于：

解析器：BeautifulSoup 默认使用 Python 标准库中的 html.parser 解析器，也支持 lxml、html5lib 等其他解析器。而 lxml 则是使用 C 语言编写的解析器，速度更快。

API：BeautifulSoup 的 API 更加简单易用，可以像使用字典一样访问 HTML 或 XML 中的标签，也可以使用 CSS 选择器或正则表达式等方式来查找元素。而 lxml 则提供了更多底层的 API，例如 XPath 查询等，可以更加灵活地处理 XML 或 HTML 文档。

性能：由于 lxml 是使用 C 语言编写的，因此其解析速度通常要快于 BeautifulSoup。

总的来说，如果对解析速度有较高的要求，并且需要进行高级的 XML 或 HTML 处理操作，可以使用 lxml 库；如果您需要进行基本的 HTML 或 XML 解析，并且希望 API 简单易用，可以使用 BeautifulSoup 库。

这次实验还让我了解到很多优秀的院士，激励我进步。而在看到女院士后面要专门标注女时，也深刻意识到平权道路仍然任重道远。