python 中 HTML 页面编写和解析

杨晨
学号 2021212171
北京邮电大学计算机学院

日期：2024 年 2 月 26 日

1 概述

1.1 实验内容

1. 用基本的 html 标签，编写一页个人简介的 html。要求有文字、链接、图片等。
2. 安装 lxml 库，并使用 etree 查找你编写的 html 中的不同元素。
3. 使用 etree，查找并输出 https://www.bupt.edu.cn/yxjg1.htm 页面中的各个学院的名字和对应链接。

1.2 开发环境

- Windows10
- PyCharm 2023.2.1

2 实验过程

2.1 个人简介 HTML 页面的编写

根据实验要求，使用 HTML 代码编写个人简介的 HTML 页面。

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>个人简介</title>
    <style>
        body {
            position: relative;
            background-image: url("https://yangchen-1318434888.cos.ap-beijing.
                myqcloud.com/images%2FOHR.MoonlightRainier_ZH-CN6263832605_UHD.jpg")
                ;
            background-size: cover;
        }
```

```
        body::after {
            content: "";
            position: absolute;
            top: 0;
            left: 0;
            width: 100%;
            height: 100%;
            background-color: rgba(255, 255, 255, 0.5); /* 调整透明度值 */
            z-index: -1;
        }
    </style>
</head>
<body>
<h1>个人简介</h1>
<p>我是一个热爱编程的人，喜欢使用Python语言进行开发。</p>
<p>以下是一些有关我的信息：</p>
<ul>
    <li>姓名：杨晨</li>
    <li>年龄：20岁</li>
    <li>邮箱：1369792882@bupt.edu.cn</li>
</ul>
<p>欢迎访问我的<a href="https://blog.yangchen-pro.com/">个人网站</a>。</p>
</body>
</html>
```

在上述代码中，我们保留了原有的 HTML 结构和内容，同时添加了一些样式。

首先，我们在<head>标签中添加了一个<style>块，用于设置页面的背景样式。通过 body 选择器，我设置了页面的背景图片为我在腾讯云对象存储中，存储的一张图片，并将背景图片的大小调整为覆盖整个页面。

然后，使用 body::after 伪元素为页面添加了一个半透明的遮罩层，以提高背景图片上文字的可读性。这里使用了 rgba() 函数来设置半透明的背景颜色，其中的最后一个参数控制透明度值。

在<body>标签中，保留了原有的标题、段落、列表和链接等内容，没有进行修改。

这样，就完成了个人简介 HTML 页面的编写，同时添加了背景样式来增强页面的美观性和可读性。

2.2 使用 lxml 库查找 HTML 中的元素

根据实验要求，安装了 lxml 库，并使用其 etree 模块来查找我编写的 HTML 中的不同元素。

首先，从文件中读取 HTML 内容，并创建了一个 HTML 解析器。

```
from lxml import etree

# 读取HTML文件内容
with open("homepage.html", "r", encoding="utf-8") as file:
    html_string = file.read()
```

```
# 创建HTML解析器
parser = etree.HTMLParser()
```

接下来，使用解析器解析 HTML 文档，并得到一个解析树对象。

```
# 解析HTML文档
tree = etree.fromstring(html_string, parser)
```

然后，使用 XPath 表达式来查找不同的元素。在这里，使用了以下 XPath 表达式：

- //h1：查找所有<h1>元素
- //p：查找所有<p>元素
- //ul：查找所有<ul>元素
- //li：查找所有<li>元素
- //a：查找所有<a>元素

```
# 使用XPath查找元素
h1_element = tree.xpath("//h1")[0]
p_elements = tree.xpath("//p")
ul_element = tree.xpath("//ul")[0]
li_elements = tree.xpath("//li")
a_element = tree.xpath("//a")[0]
```

最后，输出查找到的元素的文本内容或属性值。

```
# 输出查找到的元素文本内容
print("h1元素文本内容:", h1_element.text)
print("p元素文本内容:")
for p in p_elements:
    print("  -", p.text)
print("ul元素文本内容:")
for li in li_elements:
    print("  -", li.text)
print("a元素链接地址:", a_element.attrib["href"])
```

输出结果如下

```
h1元素文本内容: 个人简介
p元素文本内容:
  - 我是一个热爱编程的人，喜欢使用Python语言进行开发。
  - 以下是一些有关我的信息:
  - 欢迎访问我的
ul元素文本内容:
  - 姓名: 杨晨
  - 年龄: 20岁
  - 邮箱: 1369792882@bupt.edu.cn
a元素链接地址: https://blog.yangchen-pro.com/
```

这样，使用 lxml 库的 etree 模块成功查找了自己编写的 HTML 中的不同元素，并输出了它们的文本内容或属性值。

2.3 查找并输出学院名称和链接

根据实验要求，使用 lxml 库的 etree 模块和 requests 库来获取并解析 URL 页面，然后使用 XPath 表达式查找学院的名称和链接。

首先，使用 requests 库发送 HTTP GET 请求获取页面内容，并指定编码为 UTF-8。

```
from lxml import etree
import requests

# 发送HTTP GET请求获取页面内容，并指定编码为UTF-8
url = "https://www.bupt.edu.cn/yxjg1.htm"
response = requests.get(url)
response.encoding = "utf-8"  # 指定编码为UTF-8
html_string = response.text
```

接下来，将 HTML 字符串传递给 etree.HTML() 方法创建一个可解析的 HTML 树。

```
# 将HTML字符串传递给etree.HTML()方法创建一个可解析的HTML树
html_tree = etree.HTML(html_string)
```

然后，使用 XPath 表达式定位目标元素。根据查看 HTML 结构，可以通过以下 XPath 表达式定位到包含学院名称和链接的<ul>元素。

```
# 使用XPath表达式定位目标元素
ul_element = html_tree.xpath('//ul[@class="linkPageList"]/li[4]/div/ul')[0]
```

接下来，遍历目标<ul>元素下的子元素，并提取学院名称和链接。由于可能有多个<a>元素，使用循环进行提取。

```
# 遍历目标 ul 元素下的子元素，提取内容
for li_element in ul_element.xpath(".//li"):
    # 提取院系名称和链接
    a_elements = li_element.xpath(".//a")
    department_name = []
    department_link = []

    # 可能有多个 <a>
    for a_element in a_elements:
        department_name.append(a_element.text.strip())
        department_link.append(a_element.get("href"))

    # 输出院系名称和链接
    for i in range(len(department_name)):
        print("院系名称:", department_name[i])
        print("院系链接:", department_link[i])
        print()
```

这样，成功使用 lxml 库的 etree 模块和 requests 库获取并解析了 URL 页面，并输出了学院的名称和链接。

输出结果如下

```
院系名称: 信息与通信工程学院
院系链接: https://sice.bupt.edu.cn/

院系名称: 电子工程学院
院系链接: https://see.bupt.edu.cn/

院系名称: 计算机学院（国家示范性软件学院）
院系链接: http://scs.bupt.edu.cn/

院系名称: 网络空间安全学院
院系链接: http://scss.bupt.edu.cn/

院系名称: 人工智能学院
院系链接: https://ai.bupt.edu.cn/

院系名称: 现代邮政学院（自动化学院）
院系链接: https://smp.bupt.edu.cn/

院系名称: 集成电路学院
院系链接: https://ic.bupt.edu.cn/

院系名称: 经济管理学院
院系链接: http://sem.bupt.edu.cn/

院系名称: 理学院
院系链接: https://science.bupt.edu.cn/

院系名称: 未来学院
院系链接: https://www.bupt.edu.cn/yxjg1.htm

院系名称: 人文学院
院系链接: http://sh.bupt.edu.cn/

院系名称: 数字媒体与设计艺术学院
院系链接: http://sdmda.bupt.edu.cn/

院系名称: 马克思主义学院
院系链接: http://mtri.bupt.edu.cn/

院系名称: 国际学院
院系链接: http://is.bupt.edu.cn/

院系名称: 应急管理学院
院系链接: #

院系名称: 网络教育学院
```

```
院系链接: http://www.buptnu.com.cn/

院系名称: 继续教育学院
院系链接: https://sce.bupt.edu.cn/

院系名称: 玛丽女王海南学院
院系链接: https://qmsh.bupt.edu.cn/

院系名称: 体育部
院系链接: http://ped.bupt.edu.cn/
```