实验题目： 实验 4 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

(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。

(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。

(4)编写代码，爬取信息并创建本地文件。

三、 实验步骤及结果

1. 实验的源代码

主要使用的 Python 包有：

- urllib 根据指定的 url 获取网页数据
- re 使用正则表达式进行文字匹配，获取需要的信息
- os 创建文件夹，将爬取到的信息存储到本地
- bs4 BeautifulSoup 将获取到的网页源代码进行解析，定位指定的标签

实现实验目的算法设计步骤主要分 3 步：

(1) 从任务书中给出的主页面获取到每位院士的详情页链接；

(2) 对每一位院士的详细信息界面进行解析，爬取需要的信息

(3) 将获取到的信息保存

实现以上功能和步骤的源代码如下：

```python
# -*- coding = utf-8 -*-
# @Time : 2022-04-21 13:46
# @Author : wxy
# @File : Exp4.py
# @Software : PyCharm
from bs4 import BeautifulSoup          # 网页解析，获取数据
import re                              # 正则表达式，进行文字匹配
import urllib.request, urllib.error    # 指定 url，获取网页数据
import os                              # 创建文件夹

mainurl = r"https://www.cae.cn"

def main():
    """
    从主页面进入详情页面，得到每一位院士的信息
    """
    baseurl =
r"https://www.cae.cn/cae/html/main/col48/column_48_1.html"
    # 1.从主页面得到院士详情页地址
    LinkList = getLink(baseurl)
    # 2.进入每一位院士的详细信息界面，爬取需要的信息并保存
    for link in LinkList:
        getData(link)
    # getData(LinkList[0])

findName = re.compile(r'<div class="right_md_name">(.*)</div>')
findImgsrc = re.compile(r'<img src="(.*)"
style="width:150px;height:210px;"/>')

def getData(url):
    """
    在每一位院士的详细信息页面得到需要的信息，包括院士的姓名、照片和简介
    """
    html = askURL(url).decode('utf-8')
    # print(html)
    datalist = []
    # 使用正则表达式匹配需要的信息
    Name = re.findall(findName, html)[0]
    Imgsrc = re.findall(findImgsrc, html)[0]
    Imgsrc = re.sub(r'\s+',"%20", Imgsrc)
```

```
        datalist.append(Name)
        datalist.append(mainurl + Imgsrc)

        soup = BeautifulSoup(html, "html.parser")
        Intro = soup.find_all("div", class_ = "intro")[0]
        p = Intro.find_all("p")
        inq = ""
        for item in p:
            try:
                item = str(item.string).replace("\u2002", ' ', re.U)
                item = item.replace("\xa0", ' ')
                item = str(item.string).replace("\u2022", ' ')
            except:
                pass
            if item != '':
                inq = inq + item + '\n'
        datalist.append(inq)
        # print(datalist)
        saveData(datalist)

    def saveData(datalist):
        """
        将获取到的信息保存到指定路径，图片与简介均以院士的姓名命名
        """
        path = r'./Introduction/'
        if os.path.exists(path):
            pass
        else:
            os.mkdir(path)
        fileIntro = open(path + datalist[0] + ".txt", 'w+', errors='ignore')
        fileIntro.write(datalist[2])

        fileImg = open(path + datalist[0] + ".png", "wb")
        Img = askURL(datalist[1])
        fileImg.write(Img)

    def getLink(baseurl):
        """
        从主界面获得每位院士详细信息界面的 url，存储到列表中返回
        """
        Linklist = []
        # 访问页面得到页面所有内容
        html = askURL(baseurl)
        findLink = re.compile(r'<li class="name_list"><a href="(.*)"
```

```
target="_blank">.*</a></li>', re.S)

    soup = BeautifulSoup(html, "html.parser")
    for item in soup.find_all("li", class_ = "name_list"):
        # print(item)
        item = str(item)
        Link = re.findall(findLink, item)[0]
        Linklist.append(mainurl+Link)

    # print(Linklist)
    return Linklist

def askURL(url):
    """
    得到指定一个URL的网页内容
    """
    head = {  # 模拟浏览器头部信息，向服务器发送信息
        "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/100.0.4896.127
Safari/537.36 Edg/100.0.1185.44"
    }
    # 用户代理，表示告诉服务器我们是什么类型的机器（本质上是告诉浏览器可以接收什么
水平的信息
    request = urllib.request.Request(url, headers=head)
    html = ""
    try:
        response = urllib.request.urlopen(request, timeout = 10)
        html = response.read()
        # print(html)
    except urllib.error.URLError as e:
        if hasattr(e, "code"):  # 判断是否有code这个属性（有的错误信息可能没
有code）
            print(e.code)
        if hasattr(e, "reason"):
            print(e.reason)

    return html

if __name__ == "__main__":
    main()
```

在爬取时输出存储每位院士信息的列表，如果发生错误，能够更清楚地定位错误原因，便于调试。一部分的输出结果如下图 1 所示。

图 1　部分输出结果

将爬取到的院士简介和照片存储到代码运行目录下的 Introduction 文件夹中，均以院士姓名命名，图片保存为 PNG 格式，简介信息保存在 txt 文件中。文件夹内容如下图 2 所示。

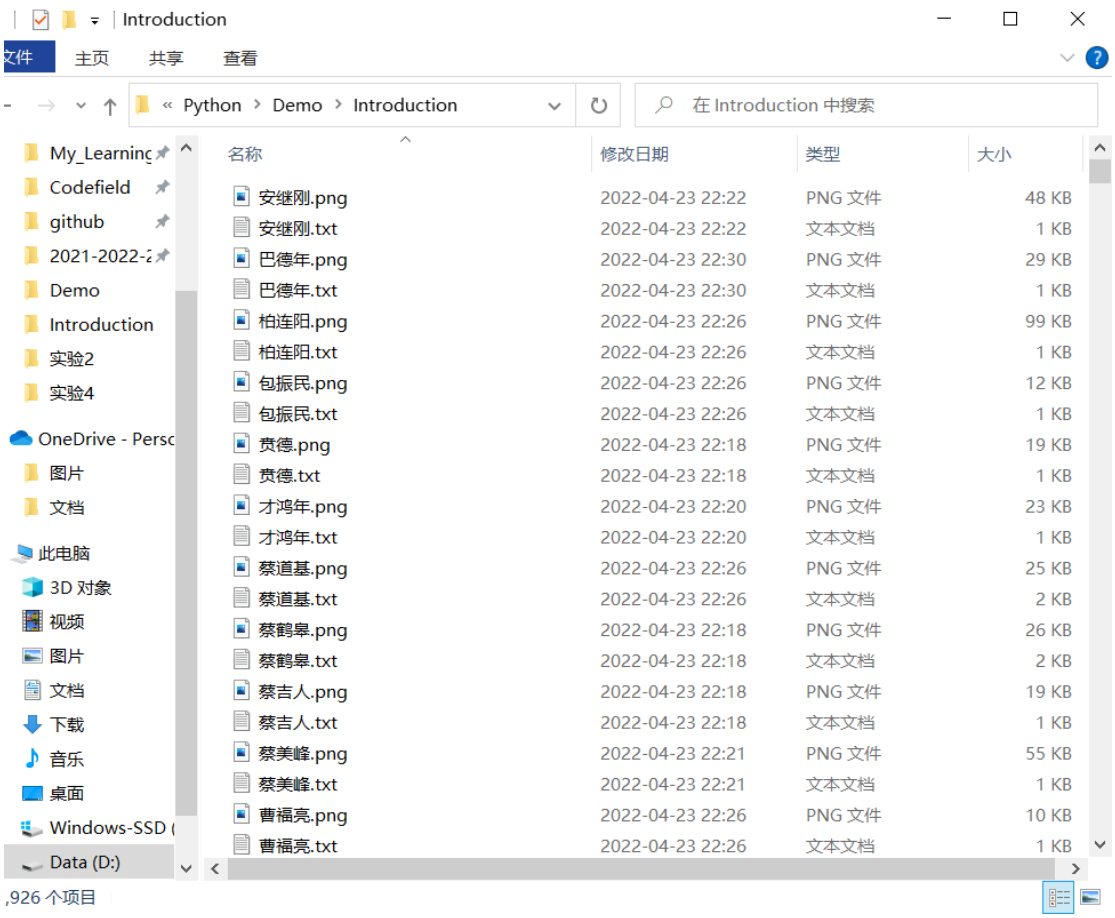

图 2　保存结果的文件夹内容

四、 实验分析

完成实验遇到的困难主要有 3 点，分别是 urllib、BeautifulSoup、re 的使用。

urllib 用于访问指定 url 的网页，根据上课讲的内容有较充分的理论知识但是缺乏实践，因此在开始做实验时通过翻看 PPT 复习，在实验过程中查阅 PPT 或者上网查询。

BeautifulSoup 是 Python 包 bs4 中的一个对象，用于解析获得的 html 源代码的标签，主要通过网上学习了解和使用。在对信息进行定位时，仅通过正则表达式感到有些繁琐，因此通过查找资料了解到了 BeautifulSoup 可以用来解析 html 从而可以非常方便地定位某个标签，在此基础上使用正则表达式匹配需要的信息。

re 是 Python 中使用正则表达式的 API，在实验之前对正则表达式的使用不

多，在实验过程中主要通过目的导向查询使用。

缅怀李三立院士。