**实验10 正则表达式与网络爬虫**

**姓名：朱俊哲 学号：2212190220 班级：计科2202 成绩：**

**1. 实验内容与目标**

本次实验旨在了解正则表达式相关基本概念以及在网络爬虫中的应用，掌握Python正则表达式模块re的常见用法，初步学会运用基本的爬虫模块urllib，以及第三方库requests和Beautiful Soup，进行web数据的自动抓取实践。需掌握的相关知识点主要包括：

**1) 正则表达式：**特殊字符的用法

**2) re模块：**

(a)正则表达式对象、匹配对象等概念

(b)核心函数与方法：compile、findall、finditer、match、search、group、sub、split

**3) 网络爬虫：**

(a) url的概念（包括http协议、ftp协议等）、爬虫的工作流程

(b) urllib基本爬虫库：掌握urllib.request模块的基本用法，结合正则表达式学会web数据的自动抓取方法。

(c)简单入门requests和Beautiful Soup等第三方爬虫库

请完成下列实验练习题，报告书写要求同第1次实验作业。注意，请在本报告中将自己的姓名、学号、班级书写正确。在规定时间内完成本次实验，并将报告的word文档(非pdf版本)以附件形式提交至学习通平台。

**2. 实验练习题**

**答题要求**：将正确运行的**源代码书写在题目下方**，并紧跟着贴上一份**源代码的截图**和运行结果的**截图**。注：如题目要求保存结果数据到文件，则该相关数据也视作运行结果，需截图粘贴。但提交报告时无需提供数据文件。

**2.1** 运用re.findall函数查找所有匹配。

(1) 设计一个正则表达式，提取出字符串中含字母’o’的全部单词(数字和标点符号都不是单词部分)。这个字符串为s="I like Python programming 123 because it is 456 simple and elegant."

源代码：

**import** re

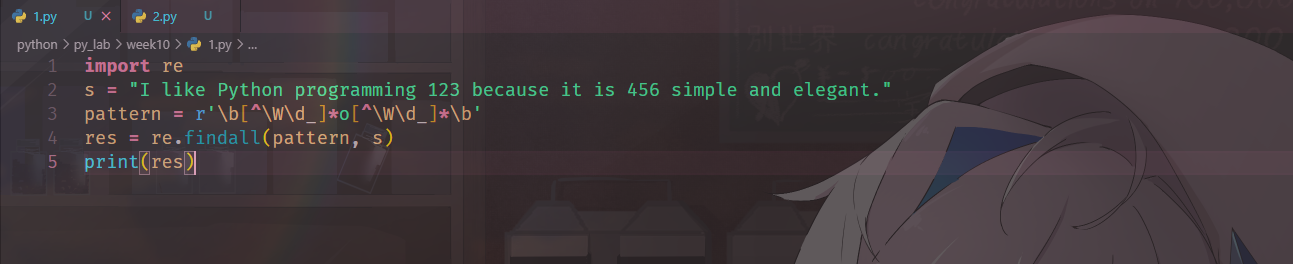
s **=** "I like Python programming 123 because it is 456 simple and elegant."

pattern **=** r'\b[**^**\W\d\_]**\***o[**^**\W\d\_]**\***\b'

res **=** re**.**findall(pattern**,** s)

print(res)

源代码截图：



运行结果截图：



(2) 提取下列文本每行中完整的年月日和时间字段：

s='''see 123, 1987-02-09 07:30:00

1987-02-15 07:25:00'''

源代码：

**import** re

s **=** '''see 123, 1987-02-09 07:30:00

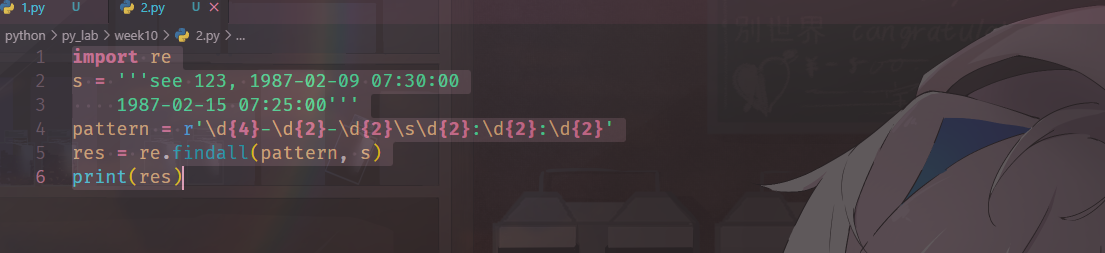
    1987-02-15 07:25:00'''

pattern **=** r'\d**{4}**-\d**{2}**-\d**{2}**\s\d**{2}**:\d**{2}**:\d**{2}**'

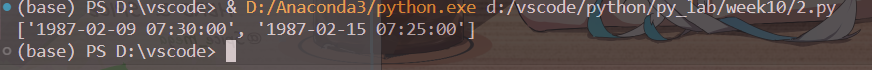
res **=** re**.**findall(pattern**,** s)

print(res)

源代码截图：



运行结果截图：



(3) 使用正则表达式找出下列文本中全体合法的邮件地址：

s="""xiasd@163.com, sdlfkj@.com sdflkj@180.com solodfdsf@123.com sdlfjxiaori@139.com saldkfj.com oisdfo@.sodf.com.com"""

源代码：

**import** re

s **=** """xiasd@163.com, sdlfkj@.com sdflkj@180.com solodfdsf@123.com sdlfjxiaori@139.com saldkfj.com oisdfo@.sodf.com.com"""

pattern **=** r'\w**+**@\w**+**\.\w**{2,}**'

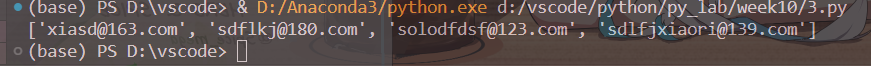
res **=** re**.**findall(pattern**,** s)

print(res)

源代码截图：



运行结果截图：



(4) 令s="see 123, 1987-02-09 07:30:00, 1986-02-15 07:25:00"

执行下面代码：

fa1=re.findall("\d+-\d+-\d+",s)

fa2=re.findall("(\d+)-\d+-\d+",s)

fa3=re.findall("(\d+)-(\d+)-(\d+)",s)

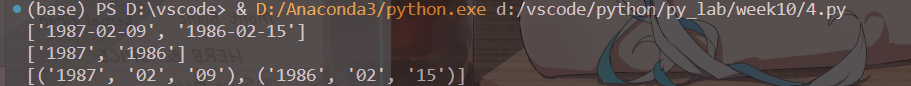
print(fa1)

print(fa2)

print(fa3)

程序输出结果是什么？通过考察结果，请学习掌握findall在正则式有否分组情形下输出结果的不同格式。

运行结果截图：



findall不同情形说明：

fa1匹配由’-’字符分隔的数字序列

fa2和fa1匹配一样但只捕获第一个数字串

fa3和fa1匹配一样但分别捕获每一个数字串

**2.2** 给定一个包含电话号码列表（一行一个电话号码）的文本文件 text.txt，写一个程序输出所有有效的电话号码。假设一个有效的电话号码必须满足以下两种格式： (xxx) xxx-xxxx 或 xxx-xxx-xxxx。（x 表示一个数字）。请输出该文本文件里所有的正确电话号码。

源代码：**import** re

pattern **=** r'\(\d**{3}**\)\d**{3}**-\d**{4}|**\d**{3}**-\d**{3}**-\d**{4}**'

valid\_phone\_numbers **=** []

**with** open('text.txt'**,** 'r') **as** file**:**

**for** line **in** file**:**

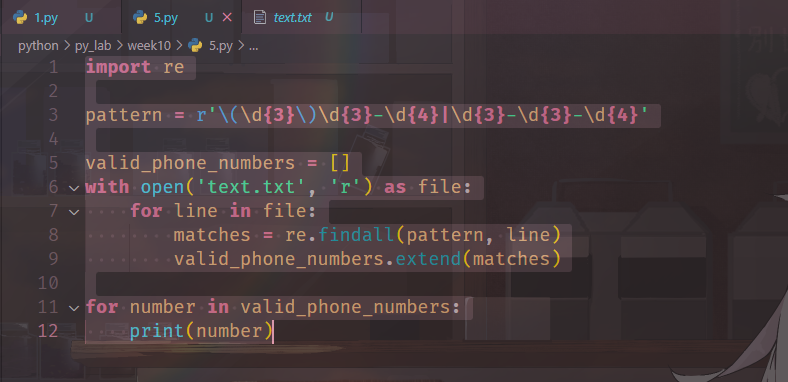
        matches **=** re**.**findall(pattern**,** line)

        valid\_phone\_numbers**.**extend(matches)

**for** number **in** valid\_phone\_numbers**:**

    print(number)

源代码截图：



运行结果截图：



**2.3** 利用re.sub函数进行字符串的替换任务。

(1) 将下列文本s中的qq邮箱地址全部都替换为你自己的电子邮件地址([hime@163.com](mailto:hime@163.com)):

s="""hi69@qq.com, werksdf@163.com, sdf@sina.com

    sfjsdf@139.com, soifsdfj@qq.com

[pwoeir23@126.com](mailto:pwoeir23@126.com)"""

源代码：

**import** re

s **=** """hi69@qq.com, werksdf@163.com, sdf@sina.com

    sfjsdf@139.com, soifsdfj@qq.com

    pwoeir23@126.com"""

pattern **=** r'\b\w**+**@qq.com\b'

replaced\_text **=** re**.**sub(pattern**,** '2827354494@qq.com'**,** s)

print(replaced\_text)

源代码截图：



运行结果截图：



(2) 利用re.sub函数，将以下文本s中的数字加上212，重新输出。

s="hi69@qq.com, werksdf@163.com, sdf@sina.com"

源代码：

**import** re

s **=** "hi69@qq.com, werksdf@163.com, sdf@sina.com"

**def** add\_212(**match**)**:**

    num **=** int(match**.**group())

**return** str(num **+** 212)

replaced\_text **=** re**.**sub(r'\d**+**'**,** add\_212**,** s)

print(replaced\_text)

源代码截图：



运行结果截图：



**2.4** 在给定的字符串中搜索有效的用户名，各个用户名以分号分隔。有效的用户名定义为以字母开头，紧随其后的是字母、数字或下划线的组合，整个用户名的长度不超过10个字符。使用search函数找到第一个这样的有效用户名，并使用group函数打印找到的用户名。

输入：“123\_john;Robert\_Michael;Mary123;Susan”

输出：Mary123

源代码：

**import** re

s **=** input()

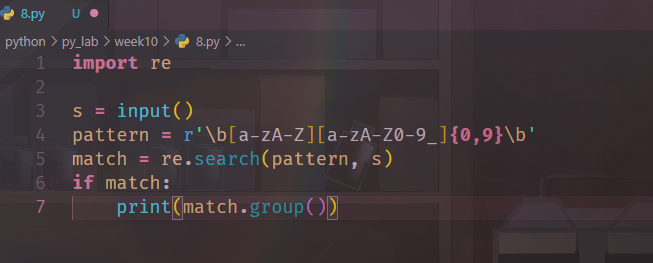
pattern **=** r'\b[a-zA-Z][a-zA-Z0-9\_]**{0,9}**\b'

match **=** re**.**search(pattern**,** s)

**if** match**:**

    print(match**.**group())

源代码截图：



运行结果截图：



**2.5**给定包含HTML页面源代码的字符串，你需要使用正则表达式来找到<title>标签及其对应的内容。请使用search函数定位这个标签，然后用group函数提取标签内的文本内容。

给定的HTML页面代码如下：

<html>  
<head>  
 <title>Sample Page</title>  
</head>  
<body>  
 <h1>Welcome to the Sample Page</h1>  
 <p>This *is* a paragraph on the sample page.</p>  
</body>  
</html>

源代码：

**import** re

html\_content **=** """

<html>

<head>

   <title>Sample Page</title>

</head>

<body>

   <h1>Welcome to the Sample Page</h1>

   <p>This is a paragraph on the sample page.</p>

</body>

</html>

"""

pattern **=** r'<title>(.**\*?**)</title>'

match **=** re**.**search(pattern**,** html\_content**,** re**.**IGNORECASE)

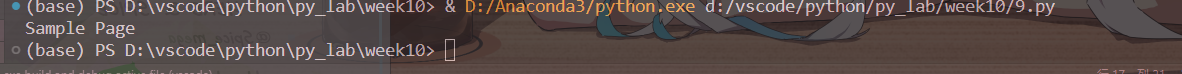
title\_content **=** match**.**group(1)

print(title\_content)

源代码截图：



运行结果截图：



**2.6** 简单网络爬虫实践。

按下列两种方式, 将网页(https://qq.yh31.com/qt/fj/)中的图片分别爬取保存至本地文件目录“image1/”和“image2/”中（请自行先创建好文件夹）。

(1) 仅利用urllib模块完成图片爬取任务，可参考“picture\_crawler.py”。

源代码：

**import** urllib**.**request

**import** re

**import** os

**def** fetch\_images(**url,** **folder**)**:**

**if** **not** os**.**path**.**exists(folder)**:**

        os**.**mkdir(folder)

    response **=** urllib**.**request**.**urlopen(url)

    html **=** response**.**read()**.**decode('utf-8')

    img\_urls **=** re**.**findall(r'src="(https**?**://.**\*?**\.jpg)"'**,** html)

**for** img\_url **in** img\_urls**:**

        img\_name **=** img\_url**.**split('/')[**-**1]

        urllib**.**request**.**urlretrieve(img\_url**,** os**.**path**.**join(folder**,** img\_name))

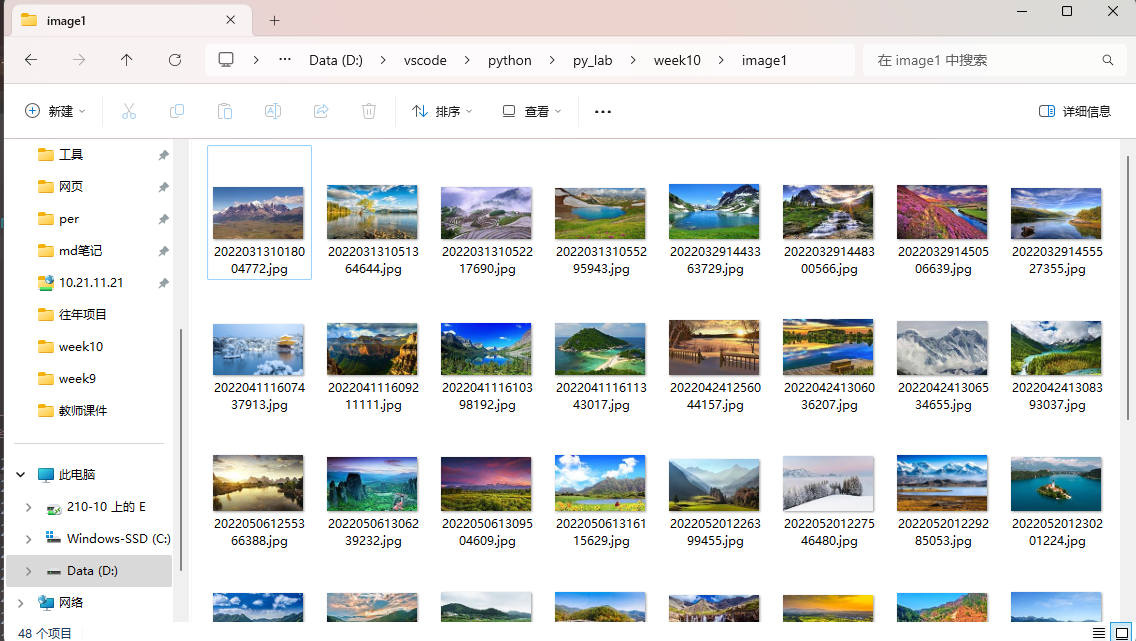
        print(f"Downloaded {img\_name}")

fetch\_images('https://qq.yh31.com/qt/fj/'**,** 'image1')

源代码截图：



图片数据截图：



(2) 结合第三方库Beautiful Soup和requests完成图片爬取任务，可参考“bs4\_pics.py”。

源代码：

**import** requests

**from** bs4 **import** BeautifulSoup

**import** os

**import** time

**import** hashlib

**def** download\_images(**url,** **save\_folder**)**:**

**if** **not** os**.**path**.**exists(save\_folder)**:**

        os**.**makedirs(save\_folder)

    response **=** requests**.**get(url)

    soup **=** BeautifulSoup(response**.**content**,** 'html.parser')

    images **=** soup**.**find\_all('img'**,** **attrs=**{'data-src'**:** lambda **src:** src **and** src**.**endswith('.jpg')})

**for** img **in** images**:**

        img\_url **=** img['data-src']

        filename **=** img\_url**.**split('/')[**-**1]

        unique\_suffix **=** hashlib**.**md5(img\_url**.**encode())**.**hexdigest()[**:**8]

        unique\_filename **=** f"{time**.**strftime('%Y%m%d%H%M%S')}\_{unique\_suffix}\_{filename}"

        save\_path **=** os**.**path**.**join(save\_folder**,** unique\_filename)

**try:**

            img\_data **=** requests**.**get(img\_url)**.**content

**with** open(save\_path**,** 'wb') **as** file**:**

                file**.**write(img\_data)

            print(f"Image saved: {save\_path}")

**except** Exception **as** e**:**

            print(f"Failed to download {img\_url}: {str(e)}")

url **=** "https://qq.yh31.com/qt/fj/"

download\_images(url**,** r"image2")

源代码截图：



图片数据截图：

