专栏：003：正则表达式

E – 爬虫技术　专栏

> 系列爬虫专栏
>
> 崇尚的学习思维是：输入，输出平衡，且维持平衡点不断精进的地步
>
> 曾经有大神告诫说：没事别瞎写文章；为此写的都是，在我能力范围内的

1：框架

序号	章节	解释
01	概念解释	概念是理解和精进的第一步
02	语法解释	2/8法则，解释使用最频繁的语法
03	代码实例	对博客内容进行使用正则表达式匹配
04	参考及说明	参考列表

2：概念

- 什么是正则表达式？

正则表达式，又称正规表示式、正规表示法、正规表达式、规则表达式、常规表示法（英语：Regular Expression，在代码中常简写为regex、regexp或RE），计算机科学的一个概念。正则表达式使用单个字符串来描述、匹配一系列匹配某个句法规则的字符串。

用自己的话复述：正则表达式是一种匹配文本的工具，由字符串和特殊字符组成。相当于一种过滤器，不符合要求的被过滤掉。

- 眼见为实

`\bhi\b.*\bLucy\b` 这是个正则表达式

3：语法

记号	说明	示例
literal	匹配字符串的值	wuxiaoshen
re1 | re2	匹配任意之一	wuxiaoshen | wuxiaoxiaoshen
.	匹配任意字符(除换行符之外)	wu.iaoshen
^	匹配字符串的开始	^wuxiaoshen
$	匹配字符串的结尾	wuxiaoshen$
*	匹配前面出现0次或者多次的	wu*xiaoshen
+	匹配前面出现1次或者多次	wu+xiaoshen
?	匹配前面出现零次或者1次	wu?xiaoshen
{N}	匹配前面出现的正则表达式N次	[0-9]{2}
{M,N}	匹配前面出现的正则表达式M到N次之间	[0-9]{3,8}
[ ]	匹配里面内容的任意一个字符	wu[xyz]iaoshen
[x-y]	匹配任意之间的一个值	[0-9]
[^..]	不匹配里面内容任意值	[^0-9]
()	匹配封闭括号中正则表达式，并保存为子组	（wuxiaoshen）
特殊字符	特殊字符	特殊字符
\d	匹配数字	data\d.txt
\w	匹配任何数字字母字符	[wuxiao]\w+
\s	匹配空白符	of\sthe
\b	匹配单词的边界	\bwuxiaoshen\b
\D	不匹配数字	
\W	不匹配数字字母字符	
\S	匹配任意不是空白符的字符	
\B	匹配不是单词开头或结束的位置	

> 看不懂，那算了。

通杀型组合： （.*? ）
括号里是你想要的内容，那就使用这个，后面会代码演示。

正则里有个贪婪还是非贪婪的概念：白话点说，贪婪就是匹配的尽可能长，非贪婪就是匹配符合要求的最短的。

- 眼见为实

```
pattern = "http://mindhacks.cn/"

mind_pattern_1 = "mind"
mind_pattern_2 = "[m].*?d"
mind_pattern_3 = r"//(.*?)h"
mind_pattern_4 = r"[mind]{4}"

还可以想出各种，都是上面的基本语法的组合

上面是已知匹配信息，想出匹配规则，匹配出规定字符
```

- 实例:

主观题：
匹配QQ号码:

匹配出手机号码：

匹配出IP：

```
QQnumber_pattern = '[1-9][0-9]{4,}'

tellnumber_pattern = '0?(13[0-9]|15[012356789]|17[0678]|18[0-9]|14
[57])[0-9]{8}'

IPnumber_pattern = '((?:(?:25[0-5]|2[0-4]\d|((1\d{2})|([1-9]?
\d))).){3}(?:25[0-5]|2[0-4]\d|((1\d{2})|([1-9]?\d))))'

IPnumber_pattern_2 = '\d+\.\d+\.\d+\.\d+'

个人理解，能匹配出绝大数，但可能还不够完美.
```

4：代码实例

先介绍个python模块：re

模块函数	**描述**
match(pattern, string, flag)	匹配以pattern开始的字符串
search(pattern, string, flag)	匹配第一符合要求的字符串，其他还符合，不管
findall(pattern, string, flag)	匹配全部符合要求的字符串
split(pattern, string, flag)	按格式进行切分
sub(pattern, repl, string, flag)	替换掉符合要求的字符串，常用来替换网址的组成

```
# 假设你对下面这个博客首页的文章的标题感兴趣.
import re
import requests
# 先缩小范围，再在缩小的范围内进行匹配

url_one = "http://www.geekonomics10000.com/author/admin"
html = requests.get(url_one)
response = html.text
#<h3 id="post-967" class="post-title"><a href="http://www.geekonomics10000.com/967" rel="bookmark" title="Permanent Link to 特朗普是极右狂人？其实共和党候选人里，他最温和">特朗普是极右狂人？其实共和党候选人里，他最温和</a></h3>

content = r'h3\sid(.*?)</h3>'

#title="Permanent Link to 特朗普是极右狂人？其实共和党候选人里，他最温和">特朗普是极右狂人？其实共和党候选人里，他最温和</a></h3>

little_title = r'title=.*?>(.*?)</a>'
all_title = re.findall(content, response, re.S)
title_content = re.findall(little_title, str(all_title), re.S)
for one in title_content:
    print(one)
# output
---
别指望灵感，还是要靠汗水 ——“创造性思维”的三个迷信
特朗普是极右狂人？其实共和党候选人里，他最温和
超强记忆力是个邪道功夫
我的新书《智识分子：做个复杂的现代人》
2016新年荐书
美国人说的圣贤之道
---

查看网页源代码：推荐chrome浏览器
```

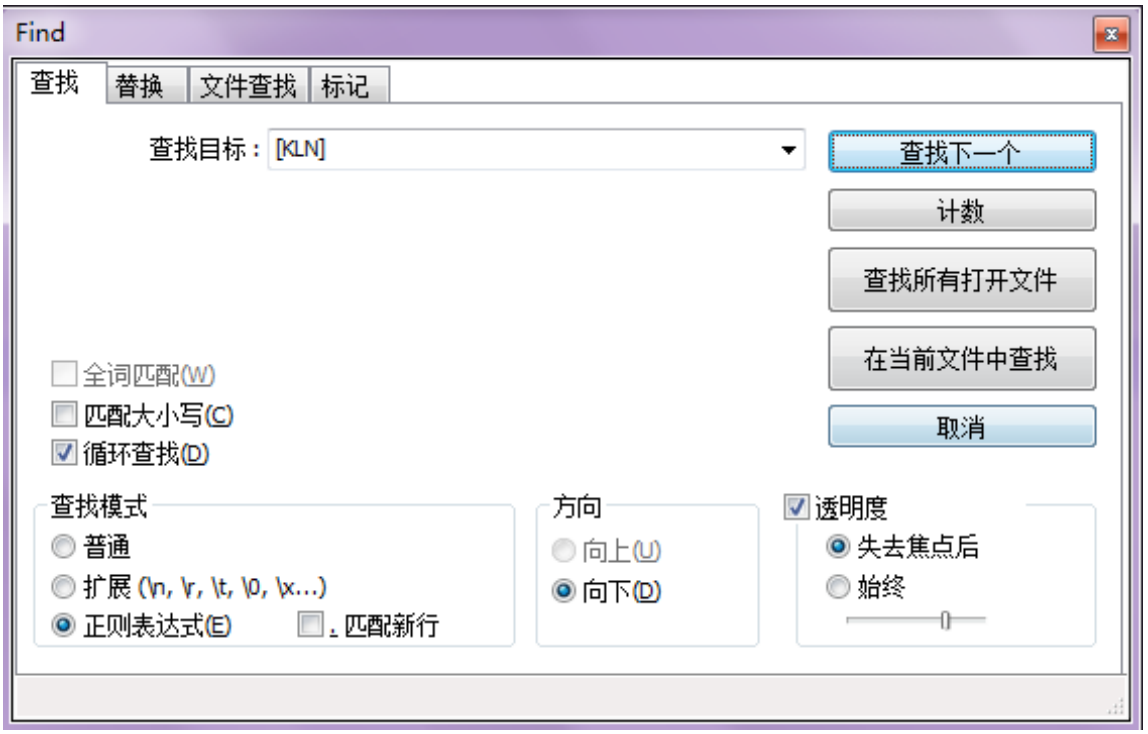
```
# 假设你想匹配首页的课程图片
# -*- coding:utf-8 -*-
# To: regular expression
# Author: wuxiaoshen

import re
import requests
class TestRe(object):
    """
    使用正则表达式抓取imooc课首页网站的图片：并下载至002 JPG文件夹下
    """
    def __init__(self):

        pass

    def download(self):
        url = "http://www.imooc.com/course/list"
        html = requests.get(url)
        response = html.text
        listurl = re.findall(r'http://.+.jpg',response)
        print(listurl)
        i = 0
        for one in listurl:
            with open("002 JPG\\"+str(i)+".jpg","wb") as f:
                cont = requests.get(one)
                print(cont)
                f.write(cont.content)
                i += 1
            f.close()
        pass

if __name__=="__main__":
    download = TestRe()
    download.download()
    pass
```

4：参考及备注

参考资料：正则表达式

如何练习正则表达式：

1. Notepad++ 文本编辑器 的查找可以使用正则匹配



2. 在线的正则表达式测试工具
3. chrome 正则匹配的插件Regular Expression Checker

> 关于本人：
> 国内小硕，半路出家的IT学习者
> 感兴趣领域：爬虫与数据科学
> 理念： 持续精进
> Github:wuxiaoshen
> weibo:乌小小申