正则表达式

常见匹配模式

模式	描述
\w	匹配字母数字及下划线
\W	匹配非字母数字下划线
\s	匹配任意空白字符，等价于 [\t\n\r\f].
\S	匹配任意非空字符
\d	匹配任意数字，等价于 [0-9]
\D	匹配任意非数字
\A	匹配字符串开始
\Z	匹配字符串结束，如果是存在换行，只匹配到换行前的结束字符串
\z	匹配字符串结束
\G	匹配最后匹配完成的位置
\n	匹配一个换行符
\t	匹配一个制表符
^	匹配字符串的开头
$	匹配字符串的末尾。
.	匹配任意字符，除了换行符，当re.DOTALL标记被指定时，则可以匹配包括换行符的任意字符。
[...]	用来表示一组字符,单独列出：[amk] 匹配 'a'，'m'或'k'
[^...]	不在[]中的字符：[^abc] 匹配除了a,b,c之外的字符。
*	匹配0个或多个的表达式。
+	匹配1个或多个的表达式。
?	匹配0个或1个由前面的正则表达式定义的片段，非贪婪方式
{n}	精确匹配n个前面表达式。
{n, m}	匹配 n 到 m 次由前面的正则表达式定义的片段，贪婪方式
a|b	匹配a或b
( )	匹配括号内的表达式，也表示一个组

re.match

re.match 尝试从字符串的起始位置匹配一个模式，如果不是起始位置匹配成功的话，match()就返回none。 re.match(pattern, string, flags=0)

最常规的匹配

```
import re

content = 'Hello 123 4567 World_This is a Regex Demo'
print(len(content))
# 逐个精确匹配 和 指定{10}精确匹配
result = re.match('^Hello\s\d\d\d\s\d{4}\s\w{10}.*Demo$', content)
print(result)
print(result.group())
print(result.span())
```

```
41
<_sre.SRE_Match object; span=(0, 41), match='Hello 123 4567 World_This is a
Regex Demo'>
Hello 123 4567 World_This is a Regex Demo
(0, 41)
```

泛匹配

```
import re

content = 'Hello 123 4567 World_This is a Regex Demo'
# 匹配开头，结尾
result = re.match('^Hello.*Demo$', content)
print(result)
print(result.group())
print(result.span())
```

```
<re.Match object; span=(0, 41), match='Hello 123 4567 World_This is a Regex
Demo'>
Hello 123 4567 World_This is a Regex Demo
(0, 41)
```

匹配目标

```
import re

content = 'Hello 1234567 World_This is a Regex Demo'
# 用括号标记要匹配的目标 \d+表示至少1个数字
result = re.match('^Hello\s(\d+)\sWorld.*Demo$', content)
print(result)
print(result.group(1))
print(result.span())
```

```
<re.Match object; span=(0, 40), match='Hello 1234567 World_This is a Regex
Demo'>
1234567
(0, 40)
```

贪婪匹配

```
import re

content = 'Hello 1234567 World_This is a Regex Demo'
# .*贪婪匹配，所以会尽可能的多匹配，而\d+至少一个即可，即匹配7
result = re.match('^He.*(\d+).*Demo$', content)
print(result)
print(result.group(1))
```

```
<re.Match object; span=(0, 40), match='Hello 1234567 World_This is a Regex
Demo'>
7
```

非贪婪匹配

```
import re

content = 'Hello 1234567 World_This is a Regex Demo'
# .*? 非贪婪匹配，所以会尽可能的少匹配，而\d+至少一个，至多没有上限，所以\d+匹配尽可能多的数字
# .*?非贪婪，意味着其他要贪婪
result = re.match('^He.*?(\d+).*Demo$', content)
print(result)
print(result.group(1))
```

```
<re.Match object; span=(0, 40), match='Hello 1234567 World_This is a Regex
Demo'>
1234567
```

匹配模式

```
import re

content = '''Hello 1234567 World_This
is a Regex Demo
'''
# re.S表示包含换行符
# result = re.match('^He.*?(\d+).*?Demo$', content, re.S)
result = re.match('^He.*?(\d+).*?Demo$', content)
print(result.group(1))
```

```
---------------------------------------------------------------------------

AttributeError                            Traceback (most recent call last)

<ipython-input-25-7bfc6a037fe1> in <module>
      7 # result = re.match('^He.*?(\d+).*?Demo$', content, re.S)
      8 result = re.match('^He.*?(\d+).*?Demo$', content)
----> 9 print(result.group(1))
```

```
AttributeError: 'NoneType' object has no attribute 'group'
```

转义

```
import re
content = 'price is $5.00'
# 特殊字符匹配失败
result = re.match('price is $5.00', content)
print(result)
```

```
None
```

```
import re

content = 'price is $5.00'
# 转义后可以匹配
result = re.match('price is \$5\.00', content)
print(result.group(0))
```

```
price is $5.00
```

总结：尽量使用泛匹配、使用括号得到匹配目标、尽量使用非贪婪模式、有换行符就用re.S

re.search

re.search 扫描整个字符串并返回第一个成功的匹配。

```
import re

content = 'Extra stings Hello 1234567 World_This is a Regex Demo Extra stings'
# match必须从起始位置开始匹配
result = re.match('Hello.*?(\d+).*?Demo', content)
# 匹配成功
# result = re.match('Extra stings Hello.*?(\d+).*?Demo', content)
print(result)
```

```
<re.Match object; span=(0, 53), match='Extra stings Hello 1234567 World_This
is a Regex >
```

```
import re

content = 'Extra stings Hello 1234567 World_This is a Regex Demo Extra stings'
# 不要求从头开始匹配，只要内容符合匹配即可
result = re.search('Hello.*?(\d+).*?Demo', content)
print(result)
print(result.group(1))
```

```
<re.Match object; span=(13, 53), match='Hello 1234567 World_This is a Regex
Demo'>
1234567
```

总结：为匹配方便，能用search就不用match

匹配演练

```
import re

html = '''<div id="songs-list">
    <h2 class="title">经典老歌</h2>
    <p class="introduction">
        经典老歌列表
    </p>
    <ul id="list" class="list-group">
        <li data-view="2">一路上有你</li>
        <li data-view="7">
```

```
            <a href="/2.mp3" singer="任贤齐">沧海一声笑</a>
        </li>
        <li data-view="4" class="active">
            <a href="/3.mp3" singer="齐秦">往事随风</a>
        </li>
        <li data-view="6"><a href="/4.mp3" singer="beyond">光辉岁月</a></li>
        <li data-view="5"><a href="/5.mp3" singer="陈慧琳">记事本</a></li>
        <li data-view="5">
            <a href="/6.mp3" singer="邓丽君"><i class="fa fa-user"></i>但愿人长久
</a>
        </li>
    </ul>
</div>'''
# 匹配齐秦<li data-view="4" class="active">
#             <a href="/3.mp3" singer="齐秦">往事随风</a>
#         </li>
result = re.search('<li.*?active.*?singer="(.*?)">(.*?)</a>', html, re.S)
if result:
    print(result.group(1), result.group(2))
```

```
齐秦 往事随风
```

```
import re

html = '''<div id="songs-list">
    <h2 class="title">经典老歌</h2>
    <p class="introduction">
        经典老歌列表
    </p>
    <ul id="list" class="list-group">
        <li data-view="2">一路上有你</li>
        <li data-view="7">
            <a href="/2.mp3" singer="任贤齐">沧海一声笑</a>
        </li>
        <li data-view="4" class="active">
            <a href="/3.mp3" singer="齐秦">往事随风</a>
        </li>
        <li data-view="6"><a href="/4.mp3" singer="beyond">光辉岁月</a></li>
        <li data-view="5"><a href="/5.mp3" singer="陈慧琳">记事本</a></li>
        <li data-view="5">
            <a href="/6.mp3" singer="邓丽君">但愿人长久</a>
        </li>
    </ul>
</div>'''
# 要匹配:
#         <li data-view="7">
```

```
#             <a href="/2.mp3" singer="任贤齐">沧海一声笑</a>
#         </li>
# <li.*?singer="(.*?)">(.*?)
result = re.search('<li.*?singer="(.*?)">(.*?)</a>', html, re.S)
if result:
    print(result.group(1), result.group(2))
```

```
任贤齐 沧海一声笑
```

```
import re

html = '''<div id="songs-list">
    <h2 class="title">经典老歌</h2>
    <p class="introduction">
        经典老歌列表
    </p>
    <ul id="list" class="list-group">
        <li data-view="2">一路上有你</li>
        <li data-view="7">
            <a href="/2.mp3" singer="任贤齐">沧海一声笑</a>
        </li>
        <li data-view="4" class="active">
            <a href="/3.mp3" singer="齐秦">往事随风</a>
        </li>
        <li data-view="6"><a href="/4.mp3" singer="beyond">光辉岁月</a></li>
        <li data-view="5"><a href="/5.mp3" singer="陈慧琳">记事本</a></li>
        <li data-view="5">
            <a href="/6.mp3" singer="邓丽君">但愿人长久</a>
        </li>
    </ul>
</div>'''
result = re.search('<li.*?singer="(.*?)">(.*?)</a>', html)
if result:
    print(result.group(1), result.group(2))
```

```
beyond 光辉岁月
```

re.findall

搜索字符串，以列表形式返回全部能匹配的子串。

```
import re

html = '''<div id="songs-list">
```

```
    <h2 class="title">经典老歌</h2>
    <p class="introduction">
        经典老歌列表
    </p>
    <ul id="list" class="list-group">
        <li data-view="2">一路上有你</li>
        <li data-view="7">
            <a href="/2.mp3" singer="任贤齐">沧海一声笑</a>
        </li>
        <li data-view="4" class="active">
            <a href="/3.mp3" singer="齐秦">往事随风</a>
        </li>
        <li data-view="6"><a href="/4.mp3" singer="beyond">光辉岁月</a></li>
        <li data-view="5"><a href="/5.mp3" singer="陈慧琳">记事本</a></li>
        <li data-view="5">
            <a href="/6.mp3" singer="邓丽君">但愿人长久</a>
        </li>
    </ul>
</div>'''
results = re.findall('<li.*?href="(.*?)".*?singer="(.*?)">(.*?)</a>', html,
re.S)
print(results)
print(type(results))
for result in results:
    print(result)
    print(result[0], result[1], result[2])
```

```
[('/2.mp3', '任贤齐', '沧海一声笑'), ('/3.mp3', '齐秦', '往事随风'), ('/4.mp3',
'beyond', '光辉岁月'), ('/5.mp3', '陈慧琳', '记事本'), ('/6.mp3', '邓丽君', '但愿人
长久')]
<class 'list'>
('/2.mp3', '任贤齐', '沧海一声笑')
/2.mp3 任贤齐 沧海一声笑
('/3.mp3', '齐秦', '往事随风')
/3.mp3 齐秦 往事随风
('/4.mp3', 'beyond', '光辉岁月')
/4.mp3 beyond 光辉岁月
('/5.mp3', '陈慧琳', '记事本')
/5.mp3 陈慧琳 记事本
('/6.mp3', '邓丽君', '但愿人长久')
/6.mp3 邓丽君 但愿人长久
```

```
import re

html = '''<div id="songs-list">
    <h2 class="title">经典老歌</h2>
```

```
    <p class="introduction">
        经典老歌列表
    </p>
    <ul id="list" class="list-group">
        <li data-view="2">一路上有你</li>
        <li data-view="7">
            <a href="/2.mp3" singer="任贤齐">沧海一声笑</a>
        </li>
        <li data-view="4" class="active">
            <a href="/3.mp3" singer="齐秦">往事随风</a>
        </li>
        <li data-view="6"><a href="/4.mp3" singer="beyond">光辉岁月</a></li>
        <li data-view="5"><a href="/5.mp3" singer="陈慧琳">记事本</a></li>
        <li data-view="5">
            <a href="/6.mp3" singer="邓丽君">但愿人长久</a>
        </li>
    </ul>
</div>'''
# \w+ 至少一个数字字母下划线
# \s*?若干个空白字符：空格/回车/制表
results = re.findall('<li.*?>\s*?(<a.*?>)?(\w+)(</a>)?\s*?</li>', html, re.S)
print(results)
for result in results:
    print(result[1])
```

```
[('', '一路上有你', ''), ('<a href="/2.mp3" singer="任贤齐">', '沧海一声笑',
'</a>'), ('<a href="/3.mp3" singer="齐秦">', '往事随风', '</a>'), ('<a
href="/4.mp3" singer="beyond">', '光辉岁月', '</a>'), ('<a href="/5.mp3"
singer="陈慧琳">', '记事本', '</a>'), ('<a href="/6.mp3" singer="邓丽君">', '但愿
人长久', '</a>')]
一路上有你
沧海一声笑
往事随风
光辉岁月
记事本
但愿人长久
```

re.sub

正则替换：替换字符串中每一个匹配的子串后返回替换后的字符串。

```python
import re

content = 'Extra stings Hello 1234567 World_This is a Regex Demo Extra stings'
# 第一个参数：正则匹配的对象
# 第二个参数：要替换成为的新对象
# 第三个参数：原字符串
# 把数字全部替换成空
content = re.sub('\d+', '', content)
print(content)
```

```
Extra stings Hello  World_This is a Regex Demo Extra stings
```

```python
import re

content = 'Extra stings Hello 1234567 World_This is a Regex Demo Extra stings'
# 把数字替换成指定字符串
content = re.sub('\d+', 'Replacement', content)
print(content)
```

```
Extra stings Hello Replacement World_This is a Regex Demo Extra stings
```

```python
import re

content = 'Extra stings Hello 1234567 World_This is a Regex Demo Extra stings'
# 替换之后，使用 \1 代表替换后的结果，继续在后面添加数字
# 注意使用r'' 表示纯字符
content = re.sub('(\d+)', r'\1 110', content)
print(content)
```

```
Extra stings Hello 1234567 110 World_This is a Regex Demo Extra stings
```

- 不规则的数据，先进行数据一致化后，再进行数据提取

```python
import re

html = '''<div id="songs-list">
    <h2 class="title">经典老歌</h2>
    <p class="introduction">
        经典老歌列表
    </p>
    <ul id="list" class="list-group">
```

```
        <li data-view="2">一路上有你</li>
        <li data-view="7">
            <a href="/2.mp3" singer="任贤齐">沧海一声笑</a>
        </li>
        <li data-view="4" class="active">
            <a href="/3.mp3" singer="齐秦">往事随风</a>
        </li>
        <li data-view="6"><a href="/4.mp3" singer="beyond">光辉岁月</a></li>
        <li data-view="5"><a href="/5.mp3" singer="陈慧琳">记事本</a></li>
        <li data-view="5">
            <a href="/6.mp3" singer="邓丽君">但愿人长久</a>
        </li>
    </ul>
</div>'''
# 删除a标签
html = re.sub('<a.*?>|</a>', '', html)
print(html)
# 处理完成后，再进行数据提取，发现数据全部在li标签里
results = re.findall('<li.*?>(.*?)</li>', html, re.S)
print(results)
for result in results:
    # 通过strip()进行去除前后空格
    print(result.strip())
```

```
<div id="songs-list">
    <h2 class="title">经典老歌</h2>
    <p class="introduction">
        经典老歌列表
    </p>
    <ul id="list" class="list-group">
        <li data-view="2">一路上有你</li>
        <li data-view="7">
            沧海一声笑
        </li>
        <li data-view="4" class="active">
            往事随风
        </li>
        <li data-view="6">光辉岁月</li>
        <li data-view="5">记事本</li>
        <li data-view="5">
            但愿人长久
        </li>
    </ul>
</div>
['一路上有你', '\n            沧海一声笑\n        ', '\n            往事随风\n    ', '光辉岁月', '记事本', '\n            但愿人长久\n        ']
一路上有你
沧海一声笑
往事随风
```

```
光辉岁月
记事本
但愿人长久
```

re.compile

将正则字符串编译成正则表达式对象

> 将一个正则表达式串编译成正则对象，以便于复用该匹配模式

```
import re

content = '''Hello 1234567 World_This
is a Regex Demo'''
pattern = re.compile('Hello.*Demo', re.S)
result = re.match(pattern, content)
#result = re.match('Hello.*Demo', content, re.S)
print(result)
```

```
<_sre.SRE_Match object; span=(0, 40), match='Hello 1234567 World_This\nis a
Regex Demo'>
```

实战练习

```
import re
html = '''        <li class="">
            <div class="cover">
              <a href="https://book.douban.com/subject/34820857/?icn=index-
latestbook-subject" title="巨浪下的小学">
                <img
src="https://img3.doubanio.com/view/subject/m/public/s33496763.jpg" class=""
                  width="115px" height="172px" alt="巨浪下的小学">
              </a>
            </div>
            <div class="info">
              <div class="title">
                <a class="" href="https://book.douban.com/subject/34820857/?
icn=index-latestbook-subject"
                  title="巨浪下的小学">巨浪下的小学</a>
              </div>
              <div class="author">
                [英]理查德·劳埃德·帕里
              </div>
              <div class="more-meta">
                <h4 class="title">
                  巨浪下的小学
```

```
                </h4>
                <p>
                  <span class="author">
                    [英]理查德·劳埃德·帕里
                  </span>
                  /
                  <span class="year">
                    2019-10
                  </span>
                  /
                  <span class="publisher">
                    文汇出版社
                  </span>
                </p>
                <p class="abstract">

                  一旦发生不幸，有些社会的第一反应就是掩盖真相，对于这样的社会而言，《巨浪下的小学》是发人深省的一堂课。今年你再也不会读到比这本书更好的非虚构故事。——《经济学人》
·
☆ 6年追踪调查3·11地震：海啸并不是问题所在，日本本身就是问题。
☆ GQ | 美国国家公共电台 | 亚马逊 |《卫报》|《经济学人》年度好书
☆ 被誉为“灾难新闻写作未来的经典” | Amazon评分4.4/5 |...
                </p>
              </div>
            </div>
          </li>'''
# 测试1
# pattern = re.compile('<li class=.*?title="(.*?)"', re.S)
# 测试2
# pattern = re.compile('<li class=.*?title="(.*?)">.*?<span class="author">
(.*?)</span>', re.S)
# 测试3
pattern = re.compile('<li class=.*?title="(.*?)">.*?<span class="author">(.*?)
</span>.*?<span class="year">(.*?)</span>', re.S)
results = re.findall(pattern, html)
for i in results:
    name, author, date = i
    # 修一修数据，可以用sub，或者 strip()
#     author = re.sub('\s', '', author)
#     date = re.sub('\s', '', date)
    print(name, author.strip(), date.strip())
```

```
巨浪下的小学 [英]理查德·劳埃德·帕里 2019-10
```

```
import requests
import re
# r = requests.get("https://www.baidu.com")
# print(r.status_code)
content = requests.get('https://book.douban.com/').text
print(content)
# pattern = re.compile('<li.*?cover.*?href="(.*?)".*?title="(.*?)".*?more-
meta.*?author">(.*?)</span>.*?year">(.*?)</span>.*?</li>', re.S)
# results = re.findall(pattern, content)
# for result in results:
#     url, name, author, date = result
#     author = re.sub('\s', '', author)
#     date = re.sub('\s', '', date)
#     print(url, name, author, date)
```