

# 第4天-发送消息和正则表达式

## 一、预备知识正则

### 1. 正则介绍

Python 中的正则，本质上是嵌入在Python中的一种微小的、高度专业化的编程语言，可通过 `re` 这个内置模块获得。正则表达式模式几乎和 shell 中的一样，更接近 `grep -P` 的效果，因为 Python 中的 `re` 模块提供的是类似 Perl 语言中的正则表达式。正则表达式模式会被编译成一系列字节码，然后由用 C 编写的匹配引擎执行。

### 2. 陷阱

友情提示：

正则表达式语言相对较小且受限制，因此并非所有可能的字符串处理任务都可以使用正则表达式完成。

还有一些任务可以用正则表达式完成，但表达式变得非常复杂。在这些情况下，你最好编写 Python 代码来进行处理；虽然 Python 代码比精心设计的正则表达式慢，但它也可能更容易理解。

### 3. 特殊的字符

在 Python 中有一些特殊的字符，在正则表达式模式中的作用和 shell 和 `grep -P` 时候有一些细微的差别

正则特殊字符	匹配内容
<code>\w</code>	匹配单个字母、数字、汉字(shell中没有)或下划线 类似于 <code>[a-zA-Z0-9_]</code>
<code>\d</code>	匹配单个数字 类似于 <code>[0-9]</code>
<code>\s</code>	匹配单个任意的空白符，这等价于 <code>[\t\n\r\f\v]</code>
<code>\S</code>	匹配任何非空白字符， <code>[^\t\n\r\f\v]</code>

## 二、re 模块的方法

其实，前面在 shell 中我们已经学习正则，这里我们主要学习的是 Python 中如何使用正则的，就是 re 模块中都有哪些方法。接下来我们就学习几个常用的方法，更多请移步[正则扩展知识完整版](#)

### 1 常用方法

**match()** 就看开头有没有

只在整个字符串的起始位置进行匹配

示例字符串

```
1 s = "isinstance yangge enumerate www.qfedu.com 1997"
```

示例演示：

```
1 import re
2 In [4]: r = re.match("is\\w+", s)
3
4 In [8]: r.group() # 获取匹配成功的结果
5 Out[8]: 'isinstance'
```

## search() 只查到第一个匹配的

从整个字符串的开头找到最后，当第一个匹配成功后，就不再继续匹配。

```
1 In [9]: r = re.search("a\\w+", s)
2
3 In [10]: r.group()
4 Out[10]: 'ance'
```

## findall() 查到所有

搜索整个字符串，找到所有匹配成功的字符串，比把这些字符串放在一个列表中返回。

```
1 In [16]: r = re.findall("a\\w+", s)
2
3 In [17]: r
4 Out[17]: ['ance', 'angge', 'ate']
```

## sub() 替换

把匹配成功的字符串，进行替换。

```

1 # 语法:
2 """
3 ("a\\w+", "100", s, 2)
4 匹配规则, 替换成的新内容, 被搜索的对象, 有相同的话替换的次数
5
6
7 """
8 In [24]: r = re.sub("a\\w+", "100", s, 2)
9
10 In [25]: r
11 Out[25]: 'isinst100 y100 enumerate www.qfedu.com
12         1997'
13 # 模式不匹配时, 返回原来的值

```

## split() 分割

和 `awk -F '[d]'` 一样效果, 以匹配到的字符进行分割, 返回分割后的列表

```

1 In [26]: s
2 Out[26]: 'isinstance yangge enumerate www.qfedu.com
3         1997'
4 In [27]: r = re.split("a", s, 1)

```

使用多个界定符分割字符串

```
1 >>> line = 'asdf fjdk; afed, fjek,asdf, foo'
2 >>> import re
3 >>> re.split(r'[;,\s]\s*', line)
4 ['asdf', 'fjdk', 'afed', 'fjek', 'asdf', 'foo']
```

## 2. 正则分组

就是从已经成功匹配的内容中，再去把想要的取出来

```
1 # match
2 In [64]: s
3 Out[64]: 'isinstance yangge enumerate www.qfedu.com
4 1997'
5 In [65]: r = re.match("is(\w+)", s)
6
7 In [66]: r.group()
8 Out[66]: 'isinstance'
9
10 In [67]: r.groups()
11 Out[67]: ('instance',)
12
13
14
15 # search
16 # 命名分组
17 In [87]: r = re.search("is\w+\s(?P<name>y\w+e)", s)
18
19 In [88]: r.group()
20 Out[88]: 'isinstance yangge'
21
```

```
22 In [89]: r.groups()
23 Out[89]: ('yangge',)
24
25 In [90]: r.groupdict()
26 Out[90]: {'name': 'yangge'}
27
28
29 # findall
30
31 In [98]: s
32 Out[98]: 'isinstance yangge enumerate www.qfedu.com
1997'
33
34 In [99]: r = re.findall("a(\\w+)", s)
35
36 In [100]: r
37 Out[100]: ['nce', 'ngge', 'ny', 'te']
38
```