

# re模块

那么接下来的问题是, 正则我会写了, 怎么在python程序中使用正则呢? 答案是re模块

re模块中我们只需要记住这么几个功能就足够我们使用了.

## 1. findall 查找所有. 返回list

```
1 lst = re.findall("m", "mai le fo len, mai ni  
   mei!")  
2 print(lst)      # ['m', 'm', 'm']  
3 lst = re.findall(r"\d+", "5点之前. 你要给我5000  
   万")  
4 print(lst)      # ['5', '5000']
```

## 2. search 会进行匹配. 但是如果匹配到了第一个结果. 就会返回这个结果. 如果匹配不上search返回的则是None

```
1 ret = re.search(r'\d', '5点之前. 你要给我5000  
   万').group()  
2 print(ret) # 5
```

## 3. match 只能从字符串的开头进行匹配

```
1 ret = re.match('a', 'abc').group()
2 print(ret)      # a
```

4. finditer, 和findall差不多. 只不过这时返回的是迭代器(重点)

```
1 it = re.finditer("m", "mai le fo len, mai ni mei!")
2
3 for el in it:
4     print(el.group()) # 依然需要分组
```

5. compile() 可以将一个长长的正则进行预加载. 方便后面的使用

```
1 obj = re.compile(r'\d{3}') # 将正则表达式编译成为一个 正则表达式对象, 规则要匹配的是3个数字
2 ret = obj.search('abc123eeee') # 正则表达式对象调用search, 参数为待匹配的字符串
3 print(ret.group()) # 结果: 123
```

6. 正则中的内容如何单独提取?

单独获取到正则中的具体内容可以给分组起名字

```
1 s = """
2 <div class='西游记'><span id='10010'>中国联通
   </span></div>
3 """
4 obj = re.compile(r"<span id='(?P<id>\d+) '>(?P<name>\w+)</span>", re.S)
5
6 result = obj.search(s)
7 print(result.group()) # 结果: <span
   id='10010'>中国联通</span>
8 print(result.group("id")) # 结果: 10010 # 获取
   id组的内容
9 print(result.group("name")) # 结果: 中国联通 #
   获取name组的内容
```

这里可以看到我们可以通过使用分组. 来对正则匹配到的内容进行进一步的筛选.

关于正则, 还有一个重要的小点, 也非常的简单, 在本节中就不继续扩展了. 下一小节的案例中会把这个小点进行简单的介绍.