

PYTHON

编程基础

re模块使用-compile 和match

compile

compile 函数用于将一个字符串形式的正则表达式编译成一个正则表达式对象，供 **match**、**search**以及其他一些函数使用。**compile**函数的语法格式为：

```
re.compile(pattern, flags=0)
```



其中，**pattern**是一个字符串形式的正则表达式；**flags**指定了匹配选项，可以使用按位或（|）运算符将多个选项连接起来；**flags**的默认值为0，表示没有任何匹配选项。

flags参数对应的匹配选项

匹配选项	描述
re.A或re.ASCII	使特殊序列中的\w、\W、\b、\B、\d、\D、\s、\S仅作ASCII码的匹配，而不是Unicode码的匹配
re.DEBUG	显示被编译正则表达式的调试信息
re.I或re.IGNORECASE	匹配时不区分大小写
re.L或re.LOCATE	使特殊序列中的\w、\W、\b、\B和不区分大小写的匹配取决于当前的语言环境（不建议使用）
re.M或re.MULTILINE	使匹配模式中的^能够匹配每行的开头若干个字符，\$能够匹配每行的结尾若干个字符
re.S或re.DOTALL	使匹配模式中的.（点）能够匹配任一字符（包括换行符）
re.X或re.VERBOSE	忽略正则表达式中的空格和#后面的注释

match

re模块中的match函数用于对字符串开头的若干字符进行正则表达式的匹配。
re.match函数的语法格式为：

```
re.match(pattern, string, flags=0)
```



其中，pattern是要匹配的正则表达式；string要作正则表达式匹配的字符串；flags参数的含义与compile函数中的flags参数相同。如果匹配成功，则返回一个Match对象；如果匹配失败，则返回None。

match

例

▶ re.match函数使用示例。

```
1 import re
2 result1=re.match(r'python', 'Python是一门流行的编程语言', re.I)
3 result2=re.match(r'python', '我喜欢学习Python', re.I)
4 result3=re.match(r'python', "我喜欢学习Python
5 Python是一门流行的编程语言", re.I|re.M)
6 print('result1:',result1)
7 print('result2:',result2)
8 print('result3:',result3)
```

match

例

▶ re.match函数使用示例。

```
result1: <re.Match object; span=(0, 6), match='Python'>
```

```
result2: None
```

```
result3: None
```

match



提示

match函数返回的是一个Match对象，其中span是匹配的字符序列在字符串中的位置信息，而match中保存了匹配到的字符序列信息。

即便对flags参数指定了匹配选项re.MULTILINE或re.M，re.match函数也只会对字符串开头的若干字符作匹配，而不对后面行的开头字符作匹配。

match

除了直接调用re模块中的match函数外，也可以使用compile函数生成的正则表达式对象中的match方法实现同样的功能，其语法格式为：

```
Pattern.match(string[,pos[,endpos]])
```



其中，Pattern是compile函数返回的正则表达式对象；string是要做正则表达式匹配的字符串；可选参数pos指定了从string的哪个位置开始进行匹配，默认为0；可选参数endpos指定了string的结束位置，match函数将对string中pos至endpos-1范围的子串进行正则表达式匹配。

match

例

► Pattern.match方法使用示例。

```
1 import re
2 pattern=re.compile(r'python', re.I) #生成正则表达式对象
3 result1=pattern.match('Python是一门流行的编程语言')
4 result2=pattern.match('我喜欢学习Python ! ',5)
5 print('result1:',result1)
6 print('result2:',result2)
```

result1: <re.Match object; span=(0, 6), match='Python'>

result2: <re.Match object; span=(5, 11), match='Python'>

match



提示

使用compile函数的优点在于：当一个正则表达式在程序中被多次使用时，通过compile函数生成的正则表达式对象可重复使用、从而提高效率。