

Python 中常见的正则匹配符

1、正则表达式模式

模式	描述
<code>^</code>	匹配字符串的开头
<code>\$</code>	匹配字符串的末尾
<code>.</code>	匹配任意字符，除了换行符
<code>[...]</code>	用来表示一组字符,单独列出: <code>[amk]</code> 匹配 'a', 'm'或'k'
<code>[^...]</code>	不在[]中的字符: <code>[^abc]</code> 匹配除了 a,b,c 之外的字符
<code>re*</code>	匹配 0 个或多个的表达式
<code>re+</code>	匹配 1 个或多个的表达式
<code>re?</code>	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
<code>re{ n}</code>	匹配 n 个前面表达式
<code>re{ n,}</code>	精确匹配 n 个前面表达式
<code>re{ n, m}</code>	匹配 n 到 m 次由前面的正则表达式定义的片段，贪婪方式
<code>a b</code>	匹配 a 或 b
<code>\w</code>	匹配字母数字及下划线
<code>\W</code>	匹配非字母数字及下划线
<code>\s</code>	匹配任意空白字符，等价于 <code>[\t\n\r\f]</code>
<code>\S</code>	匹配任意非空字符
<code>\d</code>	匹配任意数字，等价于 <code>[0-9]</code>
<code>\D</code>	匹配任意非数字

[0-9]	匹配任何数字。类似于 [0123456789]
[a-z]	匹配任何小写字母
[A-Z]	匹配任何大写字母
[a-zA-Z0-9]	匹配任何字母及数字
[^0-9]	匹配除了数字外的字符

2、正则表达式修饰符

修饰符	描述
re.I	使匹配对大小写不敏感
re.L	做本地化识别 (locale-aware) 匹配
re.M	多行匹配, 影响 ^ 和 \$
re.S	使 . 匹配包括换行在内的所有字符
re.U	根据 Unicode 字符集解析字符。这个标志影响 \w, \W, \b, \B
re.X	该标志通过给予你更灵活的格式以便你将正则表达式写得更易于理解

