

字符	说明
\	将下一字符标记为特殊字符、文本、反向引用或八进制转义符。例如，“n”匹配字符“n”。“\n”匹配换行符。序列“\\”匹配“\”，“\(”匹配“(”。
^	匹配输入字符串开始的位置。如果设置了 <b>Regex</b> 对象的 <b>Multiline</b> 属性，^ 还会与“\n”或“\r”之后的位置匹配。
\$	匹配输入字符串结尾的位置。如果设置了 <b>Regex</b> 对象的 <b>Multiline</b> 属性，\$ 还会与“\n”或“\r”之前的位置匹配。
*	零次或多次匹配前面的字符或子表达式。例如，zo* 匹配“z”和“zoo”。* 等效于 {0,}。
+	一次或多次匹配前面的字符或子表达式。例如，“zo+”与“zo”和“zoo”匹配，但与“z”不匹配。+ 等效于 {1,}。
?	零次或一次匹配前面的字符或子表达式。例如，“do(es)?”匹配“do”或“does”中的“do”。? 等效于 {0,1}。
{n}	<i>n</i> 是非负整数。正好匹配 <i>n</i> 次。例如，“o{2}”与“Bob”中的“o”不匹配，但与“food”中的两个“o”匹配。
{n,}	<i>n</i> 是非负整数。至少匹配 <i>n</i> 次。例如，“o{2,}”不匹配“Bob”中的“o”，而匹配“fooooood”中的所有 o。“o{1,}”等效于“o+”。“o{0,}”等效于“o*”。
{n,m}	<i>M</i> 和 <i>n</i> 是非负整数，其中 $n \leq m$ 。匹配至少 <i>n</i> 次，至多 <i>m</i> 次。例如，“o{1,3}”匹配“fooooood”中的头三个 o。’o{0,1}’ 等效于 ’o?’。注意：您不能将空格插入逗号和数字之间。
?	当此字符紧随任何其他限定符 (*、+、?、{n}、{n,}、{n,m}) 之后时，匹配模式是“非贪心的”。“非贪心的”模式匹配搜索到的、尽可能短的字符串，而默认的“贪心的”模式匹配搜索到的、尽可能长的字符串。例如，在字符串“oooo”中，“o+?”只匹配单个“o”，而“o+”匹配所有“o”。
.	匹配除“\n”之外的任何单个字符。若要匹配包括“\n”在内的任意字符，请使用诸如“[\s\S]”之类的模式。
(pattern)	匹配 <i>pattern</i> 并捕获该匹配的子表达式。可以使用 <b>\$0...\$9</b> 属性从结果“匹配”集合中检索捕获的匹配。若要匹配括号字符 ( )，请使用“\(”或者“\)”。
(?:pattern)	匹配 <i>pattern</i> 但不捕获该匹配的子表达式，即它是一个非捕获匹配，不存储供以后

	使用的匹配。这对于用“or”字符 ( ) 组合模式部件的情况很有用。例如， 'industr(?:y ies) 是比 'industry industries' 更经济的表达式。
(?=pattern)	执行正向预测先行搜索的子表达式，该表达式匹配处于匹配 <i>pattern</i> 的字符串的起始点的字符串。它是一个非捕获匹配，即不能捕获供以后使用的匹配。例如， 'Windows (?:95 98 NT 2000)' 匹配“Windows 2000”中的“Windows”，但不匹配“Windows 3.1”中的“Windows”。预测先行不占用字符，即发生匹配后，下一匹配的搜索紧随上一匹配之后，而不是在组成预测先行的字符后。
(?!pattern)	执行反向预测先行搜索的子表达式，该表达式匹配不处于匹配 <i>pattern</i> 的字符串的起始点的搜索字符串。它是一个非捕获匹配，即不能捕获供以后使用的匹配。例如， 'Windows (?!95 98 NT 2000)' 匹配“Windows 3.1”中的“Windows”，但不匹配“Windows 2000”中的“Windows”。预测先行不占用字符，即发生匹配后，下一匹配的搜索紧随上一匹配之后，而不是在组成预测先行的字符后。
x y	匹配 <i>x</i> 或 <i>y</i> 。例如，'z food' 匹配“z”或“food”。'(z f)ood' 匹配“zood”或“food”。
[xyz]	字符集。匹配包含的任一字符。例如，“[abc]”匹配“plain”中的“a”。
[^xyz]	反向字符集。匹配未包含的任何字符。例如，“[^abc]”匹配“plain”中的“p”。
[a-z]	字符范围。匹配指定范围内的任何字符。例如，“[a-z]”匹配“a”到“z”范围内的任何小写字母。
[^a-z]	反向范围字符。匹配不在指定的范围内的任何字符。例如，“[^a-z]”匹配任何不在“a”到“z”范围内的任何字符。
\b	匹配一个字边界，即字与空格间的位置。例如，“er\b”匹配“never”中的“er”，但不匹配“verb”中的“er”。
\B	非字边界匹配。“er\B”匹配“verb”中的“er”，但不匹配“never”中的“er”。
\cx	匹配 <i>x</i> 指示的控制字符。例如，\cM 匹配 Control-M 或回车符。 <i>x</i> 的值必须在 A-Z 或 a-z 之间。如果不是这样，则假定 <i>c</i> 就是“c”字符本身。
\d	数字字符匹配。等效于 [0-9]。
\D	非数字字符匹配。等效于 [^0-9]。
\f	换页符匹配。等效于 \x0c 和 \cL。

<code>\n</code>	换行符匹配。等效于 <code>\x0a</code> 和 <code>\cJ</code> 。
<code>\r</code>	匹配一个回车符。等效于 <code>\x0d</code> 和 <code>\cM</code> 。
<code>\s</code>	匹配任何空白字符，包括空格、制表符、换页符等。与 <code>[\f\n\r\t\v]</code> 等效。
<code>\S</code>	匹配任何非空白字符。与 <code>[^\f\n\r\t\v]</code> 等效。
<code>\t</code>	制表符匹配。与 <code>\x09</code> 和 <code>\cI</code> 等效。
<code>\v</code>	垂直制表符匹配。与 <code>\x0b</code> 和 <code>\cK</code> 等效。
<code>\w</code>	匹配任何字类字符，包括下划线。与 <code>"[A-Za-z0-9_]"</code> 等效。
<code>\W</code>	与任何非单词字符匹配。与 <code>"[^A-Za-z0-9_]"</code> 等效。
<code>\xn</code>	匹配 <i>n</i> ，此处的 <i>n</i> 是一个十六进制转义码。十六进制转义码必须正好是两位数长。例如， <code>"\x41"</code> 匹配 <code>"A"</code> 。 <code>"\x041"</code> 与 <code>"\x04"</code> & <code>"1"</code> 等效。允许在正则表达式中使用 ASCII 代码。
<code>\num</code>	匹配 <i>num</i> ，此处的 <i>num</i> 是一个正整数。到捕获匹配的反向引用。例如， <code>"(.)\1"</code> 匹配两个连续的相同字符。
<code>\n</code>	标识一个八进制转义码或反向引用。如果 <code>\n</code> 前面至少有 <i>n</i> 个捕获子表达式，那么 <i>n</i> 是反向引用。否则，如果 <i>n</i> 是八进制数（0-7），那么 <i>n</i> 是八进制转义码。
<code>\nm</code>	标识一个八进制转义码或反向引用。如果 <code>\nm</code> 前面至少有 <i>nm</i> 个捕获子表达式，那么 <i>nm</i> 是反向引用。如果 <code>\nm</code> 前面至少有 <i>n</i> 个捕获，则 <i>n</i> 是反向引用，后面跟有字符 <i>m</i> 。如果两种前面的情况都不存在，则 <code>\nm</code> 匹配八进制值 <i>nm</i> ，其中 <i>n</i> 和 <i>m</i> 是八进制数字（0-7）。
<code>\nml</code>	当 <i>n</i> 是八进制数（0-3）， <i>m</i> 和 <i>l</i> 是八进制数（0-7）时，匹配八进制转义码 <i>nml</i> 。
<code>\un</code>	匹配 <i>n</i> ，其中 <i>n</i> 是以四位十六进制数表示的 Unicode 字符。例如， <code>\u00A9</code> 匹配版权符号（©）。