芒果笔记

记录生活和技术。

正则表达式语法

2015-11-08
技术 (http://note.mango.im/category/2)
正则 (http://note.mango.im/tag/15) Roy

表达式	匹配
/^\s*$/	匹配空行。
/\d{2}-\d{5}/	验证由两位数字、一个连字符再加 5 位数字组成的 ID 号。
/<\s*(\S+)(\s[^>]*)?>[\s\S]*<\s*\/\1\s*>/	匹配 HTML 标记。
[\s\n]*?	空格，空行
([\s\S].*?)	任意字符

下表包含了元字符的完整列表以及它们在正则表达式上下文中的行为：

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 RegExp 对象的 Multiline 属性，^ 还会与“\n”或“\r”之后的位置匹配。

Sponsor

阿里云9折优惠
码：DXGS73

Categories

默认分类
(http://note.mango.im/category/1)
技术
(http://note.mango.im/category/2)
php
(http://note.mango.im/category/8)

Elsewhere

GitHub
(https://github.com/mangoim)

Friends

集智博客
(http://www.tpec.cn/)
隐的博客
(http://blog.seaning.com/

$	匹配输入字符串结尾的位置。如果设置了 RegExp 对象的 Multiline 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*}	*n* 是非负整数。正好匹配 *n* 次。例如，“o{2}”与“Bob”中的“o”不匹配，但与“food”中的两个“o”匹配。
{*n*,}	*n* 是非负整数。至少匹配 *n* 次。例如，“o{2,}”不匹配“Bob”中的“o”，而匹配“foooood”中的所有 o。“o{1,}”等效于“o+”。“o{0,}”等效于“o*”。
{*n*,*m*}	*M* 和 *n* 是非负整数，其中 *n* <= *m*。匹配至少 *n* 次，至多 *m* 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 *pattern* 并捕获该匹配的子表达式。可以使用 $0…$9 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 *pattern* 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。

	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 *pattern* 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
(?=*pattern*)	(?=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。
	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 *pattern* 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
(?!*pattern*)	(?!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 *x* 或 *y*。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。