

正则表达式

正则表达式包括几个部分:定位符, 元字符, 限定符

定位符: 用于定位匹配内容的位置

^	匹配字符串的开始
\$	匹配字符串的结束
\b	匹配单词的开始或结束

元字符(匹配符): 对特定字符实现特定的匹配功能

.	匹配除换行符以外的任意字符
\d	匹配数字
\w	匹配字母或数字或下划线或汉字
\s	匹配任意的空白符
[abcd]	匹配 abcd 中任意一个字符
[0-9a-zA-Z]	匹配数字 0-9, 字母 a-z 以及大写字母 A-Z
\W	匹配任意不是字母, 数字, 下划线, 汉字的字符
\S	匹配任意不是空白符的字符
\D	匹配任意非数字的字符
[^abcd]	匹配除了 a、b、c、d 这几个字母以外的任意字符

限定符: 对某个匹配方法的若干匹配次数

- 1.通常与元字符结合使用, 默认是贪婪匹配
- 2.对限定符前面的单个字符, 元字符, 分组表达式等进行若干次相同匹配

贪婪匹配

当正则表达式中包含能接受重复的限定符时, 通常的行为是 (在使整个表达式能得到匹配的前提下) 匹配尽可能多的字符。考虑这个表达式: `a*b`, 它将会匹配最长的以 `a` 开始, 以 `b` 结束的字符串。如果用它来搜索 `aabab` 的话, 它会匹配整个字符串 `aabab`。这被称为贪婪匹配

*	重复零次或更多次, 等价于{0,n}	
+	重复一次或更多次, 等价于{1,n}	
{	重复特定次数	
	{n,m}	重复 n 到 m 次
	{n,}	重复 n 次或 n 次以上
	{n}	重复 n 次
?	重复零次或一次, 等价于{0,1}	

懒惰匹配

当正则表达式中包含能接受重复的限定符时, 匹配尽可能少的字符。前面给出的限定符都可以被转化为懒惰匹配模式, 只要在其后面加上一个问号“?”。像这样: `a.*?` 就意味着匹配任意数量的重复, 但是在能使整个匹配成功的前提下使用最少的重复。考虑这个表达式: `a.*?b` 匹配最短的以 `a` 开始, 以 `b` 结束的字符串。如果用它来搜索 `aabab` 的话, 它会匹配 `aab`(第一个到第三个字符)`ab`(第四个到第五个字符)

*?	重复零次或更多次，但尽可能少重复
+?	重复一次或更多次，但尽可能少重复
??	重复零次或一次，但尽可能少重复
{}	重复特定次数，但尽可能少重复

规则匹配(分枝条件)

匹配规则 a | 匹配规则 b

\d{5}-\d{4}|\d{5}这个表达式能匹配规则形式为 xxxxx-xxxx 的数字以及规则形式为 xxxxx

分组表达式

(待匹配的子表达式)

参考链接: http://help.locoy.com/Document/Learn_Regex_For_30_Minutes.htm