

正则表达式入门之分组匹配

Tom

分组的概念

将子表达式做成子集

使用 () 进行分组

方便对匹配字符串时进行划分以及处理时方便提取

例如: `(\d{1,3}\.){3}\d{1,3}`

匹配IP地址

分组的命名

语法: `(?<groupname>exp)`

`(?:exp)` 匹配exp, 不捕获匹配的文本, 也不给次分组匹配组号

分组序号, 默认从0开始

子集序号从1开始

贪婪与懒惰

贪婪（尽可能多）

a. *b (aabab) aabab

懒惰（尽可能少）

匹配任意数值的重复，但是在能使整个匹配成功的前提下使用最少的重复

a. *?b (aabab) aab

贪婪与懒惰语法

语法	说明
$*?$	重复任意次，但尽可能少重复
$+?$	重复1次或更多次，但尽可能少重复
$??$	重复0次或1次，但尽可能少重复
$\{n,m\}?$	重复n到m次，但尽可能少重复
$\{n,\}$	重复n次以上，但尽可能少重复

正则表达式处理选项 (RegexOptions)

语法	说明
IgnoreCase	匹配时不区分大小写
Multiline	更改^和\$的含义，使它们分别在任意一行的行首和行尾匹配，而不仅仅在整个字符串的开头和结尾匹配。 (在此模式下，\$的精确含义是：匹配\n之前的位置以及字符串结束前的位置。)
Singleline	更改.的含义，使它与每一个字符匹配（包括换行符\n）
IgnorePatternWhitespace	忽略表达式中的非转义空白并启用由#标记的注释
ExplicitCapture	仅捕获已被显式命名的组