正则表达式

元字符

代码	说明
.	匹配除换行符以外的任意字符
\w	匹配字母或数字或下划线或汉字
\s	匹配任意的空白符
\d	匹配数字
\b	匹配单词的开始或结束
^	匹配字符串的开始
$	匹配字符串的结束

反义

代码	说明
\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
\B	匹配不是单词开头或结束的位置
[^x]	匹配除了 x 以外的任意字符
[^aeiou]	匹配除了 aeiou 这几个字母以外的任意字符

重复

代码	说明
*	重复零次或更多次
+	重复一次或更多次
?	重复零次或一次
{n}	重复 n 次
{n,}	重复 n 次或更多次

代码	说明
{n,m}	重复 n 到 m 次

如：\w+ 表示匹配多个字母（即单词），空格或换行为止。

> 懒惰匹配：在重复后加 ? 则表示尽可能少地匹配

字符范围

用 [ ] 表示匹配字符的范围

- [aeiou] 表示匹配含有aeiou中的任何一个
- [3-9] 表示 在数字中匹配 3 ~ 9 范围内的任何一个

分组

用 ( ) 表示分组

- ([1-3]){3} 表示匹配字符 1 2 3 中的一个并重复 3 次
  - 即：23461322 的结果是 132

零宽断言

- (?=exp) 也叫零宽度正预测先行断言，它断言自身出现的位置的后面能匹配表达式 exp。
  - 比如 \b\w+(?=ing\b)，匹配以 ing 结尾的单词的前面部分(除了 ing 以外的部分)，如查找 I'm singing while you're dancing.时，它会匹配 sing 和 danc。
- (?<=exp) 也叫零宽度正回顾后发断言，它断言自身出现的位置的前面能匹配表达式 exp。
  - 比如 (?<=\bre)\w+\b 会匹配以 re 开头的单词的后半部分(除了 re 以外的部分)，例如在查找 reading a book 时，它匹配 ading。
- 一个更复杂的例子：(?<=<(\w+)>).*(?=<\/\1>) ：匹配不包含属性的简单 HTML 标签内里的内容。
  - (?<=<(\w+)>)指定了这样的前缀：被尖括号括起来的单词
  - .* 表示任意的字符串
  - 最后是一个后缀(?=<\/\1>)。\/ 是"/"的转译。\1 则是一个反向引用，引用的正是捕获的第一组，前面的(\w+)匹配的内容，这样如果前缀实际上是的话，后缀就是了。
  - 整个表达式匹配的像是是和之间的内容(**不包括前缀和后缀本身**)。

贪婪与懒惰

- 贪婪匹配：如 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
- 懒惰匹配：在表达式后加 ? 表示尽可能少地匹配。如 a.*?b，则只会匹配 aab

或

| 表示或

- `th(e|in|at)` 匹配 `this is the day` 中的 this 和 the
- "三目运算符"：`(exp?yes|no)`
    - `((A)?A\d{2}\b|\b\d{3}\b)` 匹配 "A10 C103 910" 中的 "A10" 和 "910"