

正则表达式进阶之高级用法

Tom

反向引用

将要匹配目标字符串 gupao gupao, home home 这样的情况如何处理？

`\b(\w+)\b\s+\1\b`

`\b(?<Word>\w+)\b\s+\k<Word>\b` 这是命名后的另一种写法

零宽断言

`?=exp` 零宽度正预测先行断言，自身出现的位置的后面能匹配表达式`exp`

`\b\w+(?=ing\b)`

I'm **singing** while you're **dancing**.

`?<=exp` 零宽度正回顾后发断言，自身出现的位置前面能匹配表达式`exp`

`<(?<=\bre)\w+\b`

re**ading** a book

负向零宽断言

1、查找这样的单词-它里面出现了字母q，但是q后面跟的不是字母u

`\b\w*q[^\u]\w*\b`

上面的表达式有没有问题呢？使用零宽度负预测先行断言

`\b\w*q(?!u)\w*\b` 并不消费任何字符

`\d{3}(?!\d)` 匹配三位数字，而且这三位数字的后面不能是数字

`\b((?!abc)\w)+\b` 匹配不包含连续字符串abc的单词

2、同样的，还有零宽度负回顾后发断言

`(?<![a-z])\d{7}`

匹配前面不是小写字母的七位数

`(?<=(\w+)>).*?(?<=\/\1>)`

不包括前缀和后缀本身

平衡组、递归匹配

如何把xx <aa <bbb> <bbb> aa> yy这样一个字符串里，最长的配对的尖括号内的内容捕获出来？

(?'group') 把捕获的内容命名为group，并压入堆栈（Stack）

(?'-group') 从堆栈上弹出最后压入堆栈的名为group的捕获内容，如果堆栈本来为空，则本身分组的匹配失效

(?(group)yes|no) 如果堆栈上存在以名为group的捕获内容的话，继续匹配yes部分的表达式，否则继续匹配no部分

(?!) 零宽负向先行断言，由于没有后缀表达式，试图匹配总是失败