正则表达式

1. 基础介绍：

jump:匹配”jump” jump(ing)?匹配jump或jumping ？匹配0或1次

保留字：[()\\*+?{}^$.|/]

如果想搜索这些字符中一个，需用反斜杠\转义

代码中\.将显示为\\.，表定义了字符串”\.”

截获圆括号：3(pm|am)会匹配，当需选择性匹配特定字符串时，如文中查”November”,但可能简写Nov,则可定义模式Nov(ember)表括号内容可选

字符组：相当于一组字符中匹配单个字符。如t[aeiou]匹配te或ti，10[0-9]相当于10[0123456789]，但更简洁，匹配100，101，t[^0]匹配包含t并且后面紧跟非0字符

操作符：|或 \*匹配 0或多次 +匹配1或多次优先多次 ?匹配0或1次优先1次 \*?匹配0或多次 +?匹配1或多次优先1次 ??匹配0或1次优先0 {n} {n}? {n}+ {n,} {n,m} {n,m}? 对于”0000” “0+?”匹配单个0 而“0＋”匹配0000

Anchors：^行始 $行止 \A \Z \z \

.匹配任一字符，如P.P匹配POP等，但不包括\r,\n 如果要匹配全部字符[\s\S]

\d匹配数字[0-9]，例:\d\d:\d\d可匹配9:30或12:45等

\b匹配空格、标点 例:空格换行 to \b会匹配to the moon或to !在单词匹配方便

\s匹配空白字符，如空格: hello\s匹配 “hello “

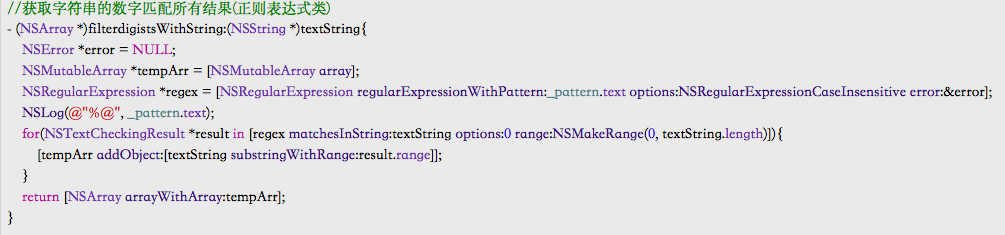
{}包含匹配最大和最小数，10{1,2}1会匹配 101，或1001

He[LI]{2,}会匹配HeLLO和HeLLIO等

1. 用法示例







1. 零宽断言
2. 简介

零宽断言用于查找在某些内容(但不包括这些内容)之前或之后但东西，也就是说，指定一个位置，这个位置应满足一定条件(断言)，断言用来声明一个应为真的事实。正则表达式只当断言为真时才会继续进行匹配。

1. 几种断言理解
2. 零宽度正预测先行断言(?=exp)匹配符合exp前的位置

例：[a-z]\*(?=ing)可以匹配cooking和singing中的cook和sing

注：先行断言执行步骤为先从匹配字符串最后面表达式，如不匹配往右继续查第二个ing

1. 零宽度正回顾后发断言(?<=exp)匹配exp后的位置

例：\*(?=ing)匹配cookingsinging中的cookingsing，而后发断言(?<=ing匹配singing)

注：先从最左端找到第一个ing，再匹配后面表达式，如不匹配，往右查第二个ing

1. 零宽度负预测先行断言(?!exp)匹配 exp前的位置

例：\d{3}(?!\d)匹配后面不是数字的三位数字

1. 零宽度负回顾后发断言(?<!exp)匹配前面不是exp的位置

例：(?<![a-z])\d{7}匹配前面不是小写字母的七位数字

口诀：先行由后往前匹配前面正则，后发由前往后匹配后面正则；正预测正回顾?=，负预测负回顾?!