正则表达：

1. 正则表达式是由普通字符（例如字符 a 到 z）以及特殊字符（称为"元字符"）组成的文字模式
2. 普通字符

[ xxx ] 匹配[ ]中所有的xxx

[^xxx] 匹配除了[ ]中的所有xxx

[ xx - xx] 匹配xx到xx的区间

[ . ] 匹配除换行符（\n、\r）之外的任何单个字符，相等于 [^\n\r]

[\s\S] 匹配所有。\s 是匹配所有空白符，包括换行，\S 非空白符，不包括换行。

[\w] 匹配字母、数字、下划线。等价于 [A-Za-z0-9\_]

1. 非打印字符

\cx 匹配由x指明的控制字符。例如， \cM 匹配一个 Control-M 或回车符。x 的值必须为 A-Z 或 a-z 之一。否则，将 c 视为一个原义的 'c' 字符。

\f 匹配一个换页符。等价于 \x0c 和 \cL。

\n 匹配一个换行符。等价于 \x0a 和 \cJ。

\r 匹配一个回车符。等价于 \x0d 和 \cM。

\s 匹配任何空白字符，包括空格、制表符、换页符等等。等价于 [ \f\n\r\t\v]。注意 Unicode 正则表达式会匹配全角空格符。

\S 匹配任何非空白字符。等价于 [^ \f\n\r\t\v]。

\t 匹配一个制表符。等价于 \x09 和 \cI。

\v 匹配一个垂直制表符。等价于 \x0b 和 \cK。

1. 特殊字符

$ 匹配输入字符串的结尾位置。

( ) 标记一个子表达式的开始和结束位置。子表达式可以获取供以后使用。

\* 匹配前面的子表达式零次或多次。

+ 匹配前面的子表达式一次或多次。

. 匹配除换行符 \n 之外的任何单字符。

[ 标记一个中括号表达式的开始。

? 匹配前面的子表达式零次或一次，或指明一个非贪婪限定符。

\ 将下一个字符标记为或特殊字符、或原义字符、或向后引用、或八进制转义符。

^ 匹配输入字符串的开始位置，

{ 标记限定符表达式的开始。

| 指明两项之间的一个选择。

1. 限定符

\* 匹配前面的子表达式零次或多次。

+ 匹配前面的子表达式一次或多次。

? 匹配前面的子表达式零次或一次。

{n} n 是一个非负整数。匹配确定的 n 次。

{n,} n 是一个非负整数。至少匹配n 次。

{n,m} m 和 n 均为非负整数，其中n <= m。最少匹配 n 次且最多匹配 m 次。

1. 定位符

^ 匹配输入字符串开始的位置。如果设置了 RegExp 对象的 Multiline 属性，^ 还会与 \n 或 \r 之后的位置匹配。

$ 匹配输入字符串结尾的位置。如果设置了 RegExp 对象的 Multiline 属性，$ 还会与 \n 或 \r 之前的位置匹配。

\b 匹配一个单词边界，即字与空格间的位置。

\B 非单词边界匹配。

1. 选择

用圆括号 () 将所有选择项括起来，相邻的选择项之间用 | 分隔。