

正则表达式

1.是什么？

Regular Expression

匹配想要的东西：把人类所认可的一种规则使用计算机能读懂的语言表达出来

正则正能操作字符串

是一种正常人类很难读懂的语法格式

2.正则表达式对象的创建

语法

var reg = new RegExp(pattern, modifiers);

var reg = /pattern/modifiers;

方法

test()

获取到匹配的字符返回true，否则返回false

exec()

获取到匹配的字符并返回，如果没有则返回null

3.字符串的方法

str.search(/haha/)

根据正则表达式进行查找，返回出现的位置，找不到就返回-1

str.replace(/haha/, "hehe")

根据正则表达式进行查找替换，例如：敏感词替换

str.match(/\d/g)

找到匹配的字符并返回

4.正则语法规则

modifiers（修饰词）

i

匹配时忽略大小写

g

执行时全局匹配

查找所有匹配项，而非查找到一个即停止

小括号

(abc)

查找abc这个单词

()：字符组合

(abc|bcd|def)

查找abc、bcd、def这三个单词其中的一个

|：或

中括号

[12345]

包括1、2、3、4、5其中的某一个

[0-9]

包括所有数字

[abc]

包括a、b、c其中的一个

[a-z]

包括所有的小写字母

[A-Z]

包括所有的大写字母

[A-z]

包括所有的字母

[0-9a-z]

包含所有的数字和小写字母

[^ab]

除了字符ab

元字符

.

代表任何的字符

注意：如果需要加.的时候，需要进行转义

\\.

\\w

任何字母、数字、下划线

\\W

任何非字母、数字、下划线

\\d

任何数字

\\D

任何非数字

\\s

空白字符

\\S

非空白字符

\\b

查找单词边界

\\B

查找非单词边界

\\0

查找NULL字符

\\n

查找换行符

\\f

查找分页符

\\r

查找回车符

\\t

查找制表符

\\v

查找垂直制表符

量词

^n

匹配开头为n的字符串

n\$

匹配结尾为n的字符串

n+

匹配任何包含至少一个n的字符串

n{x}

匹配任何包含x个n的字符串

n{x, y}

匹配包含x到y个n的字符串

n\*

匹配任何包含0个或者多个n的字符串

n?

匹配任何包含0个或1个n的字符串

?=n

匹配任何其后紧接指定字符串n的字符串

?!n

匹配任何其后没有紧接字符串n的字符串

5.实例

判断全部数字

判断文件后缀为PDF

判断手机号

判断邮箱

判断IP地址

6.工具网站

<https://regexper.com>