正则RegExp

1、定义

检索某个文本，使用正则模式。可以是单独的字符，字符串解析、格式检查、替换等，可规定字符串的检索位置、检索类型等。可以用new RegExp()

2、方法

RegExp对象有3个方法：test()、exec()、compile()

**test() 检索字符串中的指定值。regexObj.test(str)。返回值是true or false**

例如： /^135/.test(‘13566668888’) 返回 true

**exec() 检索字符串中的指定值。regexObj.exec(str)。返回值是被找到的值。**

例如： do{

var result = /o/g.exec(‘hello world’);

document.write(result);

}

while(result!=null) // 循环输出所有值

或者： /e/.exec(‘hello world’); 返回e

**complie() 既可以改变检索模式，也可以添加或删除第二个参数。返回值是true or false**

例如：var patt = new RegExp(“e”);

patt.test(“hello world”);

patt.compile(“s”);

patt.test(“hello world);

3、修饰符

i：执行对大小写不敏感的匹配

g：执行全局匹配

m：执行多行匹配

4、方括号

[abc] 查找方括号中的任何字符

[^abc] 查找不在方括号中的任何字符

[0-9] 查找从0至9的数字

[^0-9] 查找非数字的一个字符

[a-z] 查找从小写a至小写z的一个字母

[A-z] 查找从大写A到大写Z的一个字母

[red|blue|green] 查找任何指定的选项 | 管道符：表示 或，即选分支。

5、元字符

. 查找单个字符，除了换行和行结束符

\d 查找数字 等价于[0-9]

\D 查找非数字

\s 查找空白符

\S 查找非空白符

\w 查找单词字符 等价于[A-Za-z0-9]

\W 查找非单词字符

\b 匹配单词边界 例如：/\bm/ moon 只匹配边界

\B 匹配非单词边界

\0 查找NUL字符

\ 斜杠表示转义字符。 大多数符号在正则表达式中需转义！

\n 查找换行符

\f 查找换页符

\r 查找回车符

\t 查找制表符

\v 查找垂直制表符

\xxx 查找以八进制数xxx规定的字符

\xdd 查找以16进制数dd规定的字符

\uxxxx 查找以16进制数xxxx规定的unicode字符

6、量词

+ 一个或多个

\*零个或多个

? 零个或一个

{m} m次

{m,n} m到n次

{m,} 至少m次

^n 以n开头的字符串

n$ 以n结尾的字符串

?=n 匹配任何其后紧跟指定字符串n的字符串

?!n 匹配任何其后没有紧跟指定字符串n的字符串

7、支持正则表达式的String对象方法

() 括号表示捕获。 使用 $1,$2, … ; 可获取捕获值

match() 找到一个或多个正则表达式的匹配

replace() 替换正则表达式匹配的子串

search() 检索与正则表达式相匹配的值

split() 把字符串分割成数组

例如： 关于url 的匹配和捕获

var reg = / (https?:)\/\/([^\/]+)(\/[^\?]\*)?(\?[^\#]\*)? /

(https?:) 表示匹配http或https

\/\/ 转义符，转义url中的双斜杠//

([^\/]+) 表示非斜杠之前的字符串

(\/[^\?]\*)? 表示斜杠之后，非？之前的字符串

(\?[^\#]\*)? 表示问号之后，非# 之前的字符串

后两者，都是可能出现0次或者1次