正则表达式

1. 是描述一种字符串匹配的模式,正则表达式作为一个模板,将某个字符模式与搜索的字符进行匹配.
2. 简写为RegExp,主要是用来有效性规则匹配
3. 两种实例化方法

Var Exp1=new RegExp(pattern,flags);

Var Exp2=/pattern/flags;

Pattern:正则表达式

Flags:参数,

g :全局匹配,如果你没有设置这个参数,它只会找到最先出现的,而不会全部找到

i:不区分大小写

m:多行匹配

1. RegExp的方法

Test方法：检测字符串是否存在创建的正则表达式的字符串,有则返回true,否则false

Exec方法,指定正则模式搜索字符串,返回一个包含结果的数组.

1. 正则表达式元字符

限定符

{n,}规定前面的元素至少连续出现N次

{n,m}规定前面的元素至少连s续出现N次,最多连续出现m次

+:规定前面的元素至少连续出现1次或连续多次等效于{1,}

\*:等效于{0,}

?:出现0次或一次等效于{0,1}

1. 选择匹配符

只有一个 |

(nan|nv)22 只能匹配 nan22或者nv22

1. 字符匹配符

1.[…]:匹配方括号里面的任意一个字符[abc],只能匹配a,b,c

2.[^…]:匹配方括号中未包含的任何字符,[^abc],除开a,b,c其他字符都可以匹配

3.\d:匹配任意数字字符等效于[0-9]

4.\D:匹配任意一个非数字字符等效于[^0-9]

5.\s:匹配任何空白字符,包括空格,制表符等

6.\S匹配任何非空白字符.

7.\w:匹配英文字符,数字,下划线等效于[A-Za-z0-9\_]

8.\w与上面相反[^A-Za-z0-9\_]

9..匹配除换行符\n之外的任何单字符

8.定位符

要实现字符串的完全匹配,表示从字符串开始到结束的位置进行匹配

开始^

结束$

例如只能输入男或女 /^男$|^女$/

9.匹配中文[\u4e00-\u9fa5]

10.查找替换

Replace()方法

返回根据正则表达式进行文本替换后的字符串的复制

Stringobj.replace(rgExp,replaceText);

11.查找字符串

Search()方法

返回与正则表达式查找内容匹配的第一个字符串的位置

Stringobj.search(regexp);

查找成功则返回位置,没有则返回-1

12.全局查找

Match()方法

使用正则表达式模式对字符串执行查找,并将包含查找的结果作为数组返回

Stringobj.match(regExp);