**十四 、javascript-正则表达式**

**一、认识正则表达式**

**1.认识**

正则表达式(regular expression)是一个描述字符模式的对象。ECMAScript 的 RegExp 类 表示正则表达式，而 String 和 正则表达式都定义了进行强大的【模式匹配】和【文本检索】与【替换】的函数。

RegExp 类 正则表达式对象。

【注】强大的字符串。

**2.创建正则表达式**

修饰符：

i 忽略大小写

g 全局匹配

m 换行匹配

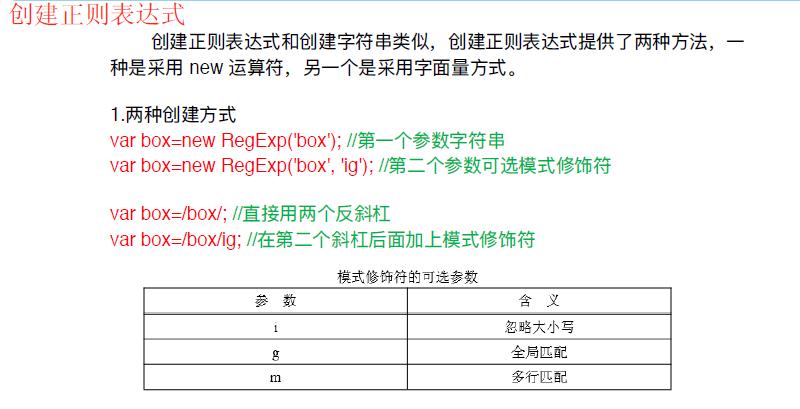
(1)通过new运算符创建正则表达式

var box = new RegExp("box", "ig");

alert(box); // /box/gi

(2)直接通过常量进行赋值

var box = /box/ig;



**3.测试正则表达式**

正则表达式 函数

test()

【格式】正则表达式.test(匹配的字符串)

【功能】匹配成功，返回true，否则返回false

exec()

【格式】正则表达式.exec(匹配的字符串)

【功能】匹配成功，返回装有匹配成功的子串的数组，否则返回null。

var str = "This is a Box";

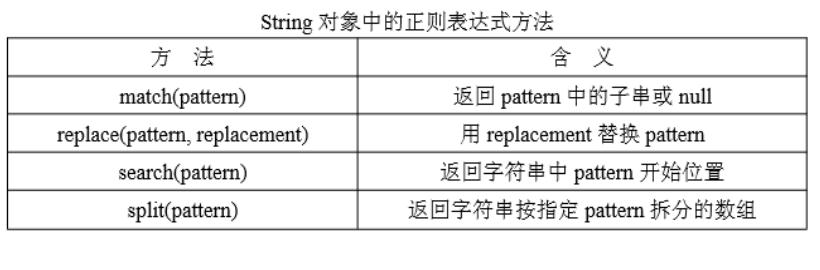
var pattern = /boxx/ig;



alert(pattern.test(str));

alert(pattern.exec(str));

**二、字符串的方法**



**1.match()**

【格式】字符串.match(正则表达式/子字符串);

【返回值】匹配成功，所有匹配成功的子串，组成的数组，匹配失败，返回null。

**2.replace()替换**

**3.search()**

【功能】查找子串在父串中出现的位置，有且只能查找一次。

**4.split()**

var str = "This is a box Box";

alert(str.split(/b/ig));

//结果为This is a, ox , ox ;

**三、元字符**

正则表达式

元字符：具有一些特殊意义的字符

**1.单个字符**

【**.**】 可以匹配任意字符

[0-9] 约束可以匹配括号内约定的单个字符

[0-9a-zA-Z\_] 数字字母

[^0-9] 约束可以匹配**除了**括号内约定的单个字符

\d 匹配数字的单个字符

\D 匹配**非**数字的单个字符

\w 匹配数字、字母、下划线的单个字符 [0-9a-zA-Z\_]

\W 匹配数字、字母、下划线**除外**的单个字符 [^0-9a-zA-Z\_]

**空白字符：**

\s 匹配空白字符

\S 匹配**非**空白字符

**2.重复字符**

x 代表上述的单个元字符

x? 代表匹配0个或者1个x字符。

x\* 代表匹配0个或者任意个x字符。

x+ 代表匹配1个或者大于1个的人一个x字符

x{m,n} 代表最少m个最大n个x字符

x{num} 代表匹配num个x字符。

(xyz)+ 小括号括起来的元字符，会被当做是一个字符进行匹配

**3.锚字符**

^ 行首匹配 约定以哪种方式开头

$ 行尾匹配 约定以哪种方式结尾

**4.换行模式m**

var pattern = /^\d+/gm;

var str = "1.baidu\n2.google\n3.bing";

alert(str);

alert(str.replace(pattern, "\*"));

**5.替代字符**

**|**  替代字符

转义字符 **\ \. \\***

.com

.cn

.net

.org

//匹配是否是一个压缩包

alert(/^\w+\.zip|rar|gz/.test("1.gz"));



