**正则表达式**

语法

/正则表达式主体/修饰符(可选)

如：

/.\*?Abc/i

.\*?Abc为主体，i为修饰符

**使用字符串方法**

在 JavaScript 中，正则表达式通常用于两个字符串方法 : search() 和 replace()。

search() 方法 用于检索字符串中指定的子字符串，并返回子字符串的起始位置。

replace() 方法 用于在字符串中用一些字符替换另一些字符

search() 方法使用正则表达式

使用正则表达式搜索 "w3cschool" 字符串，且不区分大小写：

var str = "Visit w3cschool";

var n = str.search(/w3cschool/i);

输出结果为：

6

search() 方法使用字符串

检索字符串中 "w3cschool" 的子字符串：

var str = "Visit w3cschool!";

var n = str.search("w3cschool");

replace() 方法使用正则表达式

使用正则表达式且不区分大小写将字符串中的 Microsoft 替换为 w3cschool :

var str = "Visit Microsoft!";

var res = str.replace(/microsoft/i, "w3cschool");

结果输出为:

Visit w3cschool!

replace() 方法使用字符串

replace() 方法将接收字符串作为参数：

var str = "Visit Microsoft!";

var res = str.replace("Microsoft", "w3cschool");

**正则表达式修饰符**

修饰符 描述

i 执行对大小写不敏感的匹配。

g 执行全局匹配（查找所有匹配而非在找到第一个匹配后停止）。

m 执行多行匹配。

**正则表达式模式**

方括号用于查找某个范围内的字符：

表达式 描述

[abc] 查找方括号之间的任何字符。

[0-9] 查找任何从 0 至 9 的数字。

(x|y) 查找任何以 | 分隔的选项。

元字符是拥有特殊含义的字符：

元字符 描述

\d 查找数字。

\s 查找空白字符。

\b 匹配单词边界。

\uxxxx 查找以十六进制数 xxxx 规定的 Unicode 字符。

量词:

量词 描述

n+ 匹配任何包含至少一个 n 的字符串。

n\* 匹配任何包含零个或多个 n 的字符串。

n? 匹配任何包含零个或一个 n 的字符串。

**RegExp 对象**

在 JavaScript 中，RegExp 对象是正则表达式对象，其具有的方法如下：

test()

test() 方法用于检测一个字符串是否匹配某个模式

以下实例用于搜索字符串中的字符 "e"：

var patt = /e/;

patt.test("The best things in life are free!");

字符串中含有 "e"，所以该实例输出为：

true

exec()

exec() 方法用于检索字符串中的正则表达式的匹配。

该函数返回一个数组，其中存放匹配的结果。如果未找到匹配，则返回值为 null。

以下实例用于搜索字符串中的字母 "e":

/e/.exec("The best things in life are free!");

字符串中含有 "e"，所以该实例输出为:

[“e”]