# JavaScript 正则表达式

正则表达式（英语：Regular Expression，在代码中常简写为regex、regexp或RE）使用单个字符串来描述、匹配一系列符合某个句法规则的字符串搜索模式。

搜索模式可用于文本搜索和文本替换。

## 什么是正则表达式？

正则表达式是由一个字符序列形成的搜索模式。

当你在文本中搜索数据时，你可以用搜索模式来描述你要查询的内容。

正则表达式可以是一个简单的字符，或一个更复杂的模式。

正则表达式可用于所有文本搜索和文本替换的操作。

## 语法

/正则表达式主体/修饰符(可选)

其中修饰符是可选的。

## 实例：

var patt = /runoob/i

实例解析：

**/runoob/i**  是一个正则表达式。

**runoob**  是一个**正则表达式主体** (用于检索)。

**i**  是一个**修饰符** (搜索不区分大小写)。

## 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，或检索与正则表达式相匹配的子字符串，并返回子串的起始位置。

**replace() 方法** 用于在字符串中用一些字符替换另一些字符，或替换一个与正则表达式匹配的子串。

## search() 方法使用正则表达式

var str = "Visit Runoob!";

var n = str.search(/Runoob/i);

console.log(n);

## search() 方法使用字符串

search 方法可使用字符串作为参数。字符串参数会转换为正则表达式：

var str = "Visit Runoob!";

var n = str.search( );

console.log(n);

## replace() 方法使用正则表达式

var str = "hello javascript"

var n = str.replace(/java/,"world");

console.log(n);

replace() 方法使用字符串

var str = "hello javascript"

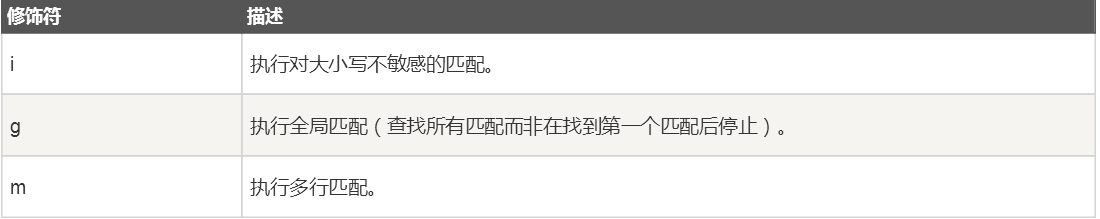
var n = str.replace(patt,"world");

console.log(n);

正则表达式参数可用在以上方法中 (替代字符串参数)。  
正则表达式使得搜索功能更加强大(如实例中不区分大小写)。

## 正则表达式修饰符

**修饰符** 可以在全局搜索中不区分大小写



var str = "hello jAVascript";

var patt = /java/i;

var pos = str.search(patt);

var str = "hello javascriptjava";

var patt = /java/g;

var newstr = str.replace(patt,"aaa");

console.log(newstr);

## 正则表达式模式

方括号用于查找某个范围内的字符：

## 

## /ABC/1234

## /0-9/abc

## /（abc|xyz）123/

## / [a-zA-Z]1234/

## 元字符是拥有特殊含义的字符：

## 

## \d == /[0-9]/

## \s 匹配任何空白字符，包括空格、制表符、换页符等等。等价于 /[ \f\n\r\t\v]/

## 量词:

## 

## 限定符

限定符用来指定正则表达式的一个给定组件必须要出现多少次才能满足匹配。有\*或+或?或{n}或{n,}或{n,m}共6种。

正则表达式的限定符有：

## 

## 特殊字符

所谓特殊字符，就是一些有特殊含义的字符，如上面说的"\*.txt"中的\*，简单的说就是表示任何字符串的意思。如果要查找文件名中有\*的文件，则需要对\*进行转义，即在其前加一个\。ls \\*.txt。

许多元字符要求在试图匹配它们时特别对待。若要匹配这些特殊字符，必须首先使字符"转义"，即，将反斜杠字符 (\) 放在它们前面。下表列出了正则表达式中的特殊字符：

## 

## 