正则表达式：

语法：/w3cschool/

1、

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是，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。

1. .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，并返回子字符串的起始位置；
2. .replace()方法：用于在字符串中用一些字符替换，或替换与正则表达式相匹配的字符串

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!

## 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:

|  |  |
| --- | --- |
| **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。 |
|  | 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。 |

## 使用 test()

test() 方法是一个正则表达式方法。

test() 方法用于检测一个字符串是否匹配某个模式，如果字符串中含有匹配的文本，则返回 true，否则返回 false。

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

## 使用 exec()

exec() 方法是一个正则表达式方法。

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

JQuery中JSONP如何解决跨域问题：

第一种方法是在ajax函数中设置dataType为'jsonp'：

1. $.ajax({
2. dataType: 'jsonp',
3. url: 'http://www.a.com/user?id=123',
4. success: function(data){
5. //处理data数据
6. }
7. });

第二种方法是利用getJSON来实现，只要在地址中加上callback=?参数即可：

1. $.getJSON('http://www.a.com/user?id=123&callback=?', function(data){
2. //处理data数据
3. });

也可以简单地使用getScript方法：

1. //此时也可以在函数外定义foo方法
2. function foo(data){
3. //处理data数据
4. }
5. $.getJSON('http://www.a.com/user?id=123&callback=foo');