

JavaScript RegExp 对象

RegExp：是正则表达式 (regular expression) 的简写。

完整 RegExp 对象参考手册

请查看我们的 [JavaScript RegExp 对象的参考手册](#)，其中提供了可以与字符串对象一同使用的所有的属性和方法。这个手册包含的关于每个属性和方法的用法的详细描述和实例。

什么是 RegExp ?

正则表达式描述了字符的模式对象。

当您检索某个文本时，可以使用一种模式来描述要检索的内容。RegExp 就是这种模式。

简单的模式可以是一个单独的字符。

更复杂的模式包括了更多的字符，并可用于解析、格式检查、替换等等。

您可以规定字符串中的检索位置，以及要检索的字符类型，等等。

语法

```
var patt=new RegExp(pattern,modifiers);
```

或更简单的方法

```
var patt=/pattern/modifiers;
```

- 模式描述了一个表达式模型。
- 修饰符(modifiers)描述了检索是否是全局，区分大小写等。

注意：当使用构造函数创建正则对象时，需要常规的字符转义规则（在前面加反斜杠\）。比如，以下是等价的：

```
var re = new RegExp("\\w+");  
var re = /\w+/;
```

RegExp 修饰符

修饰符用于执行不区分大小写和全文的搜索。

i - 修饰符是用来执行不区分大小写的匹配。

g - 修饰符是用于执行全文的搜索（而不是在找到第一个就停止查找,而是找到所有的匹配）。

实例 1

在字符串中不区分大小写找"runoob"

```
var str = "Visit RUnooB";  
var patt1 = /runoob/i;
```

以下 **标记** 的文本是获得的匹配的表达式：

Visit **RUnooB**

尝试一下 »

实例 2

全文查找 "is"

```
var str="Is this all there is?";  
var patt1=/is/g;
```

以下 **标记** 的文本是获得的匹配的表达式：

Is **this** all there **is**?

尝试一下 »

实例 3

全文查找和不区分大小写搜索 "is"

```
var str="Is this all there is?";  
var patt1=/is/gi;
```

以下 **标记** 的文本是获得的匹配的表达式：

Is **this** all there **is**?

尝试一下 »

test()

test()方法搜索字符串指定的值，根据结果并返回真或假。

下面的示例是从字符串中搜索字符 "e"：

实例

```
var patt1=new RegExp("e");  
document.write(patt1.test("The best things in life are free"));
```

由于该字符串中存在字母 "e"，以上代码的输出将是：

true

尝试一下 »

当使用构造函数创建正则对象时，需要常规的字符转义规则（在前面加反斜杠 \）

实例

```
var re = new RegExp("\\w+");
```

[尝试一下 »](#)

exec()

exec() 方法检索字符串中的指定值。返回值是被找到的值。如果没有发现匹配，则返回 null。

下面的示例是从字符串中搜索字符 "e"：

实例 1

```
var patt1=new RegExp("e");  
document.write(patt1.exec("The best things in life are free"));
```

由于该字符串中存在字母 "e"，以上代码的输出将是：

```
e
```

[尝试一下 »](#)[← JavaScript Math \(算数 \) 对象](#)[JavaScript Window →](#)[✎ 点我分享笔记](#)