

## JavaScript 参考手册

## 概览

## JavaScript 对象

[JavaScript Array 对象](#)[JavaScript Boolean 对象](#)[JavaScript Date 对象](#)[JavaScript Math 对象](#)[JavaScript Number 对象](#)[JavaScript String 对象](#)[JavaScript RegExp 对象](#)[JavaScript 全局属性/函数](#)[JavaScript 运算符](#)

## Browser 对象

[Window 对象](#)[Navigator 对象](#)[Screen 对象](#)[History 对象](#)[Location 对象](#)

## DOM 对象

[← JavaScript String 对象](#)[JavaScript 全局属性/函数 →](#)

# JavaScript RegExp 对象

## RegExp 对象

正则表达式是描述字符模式的对象。

正则表达式用于对字符串模式匹配及检索替换，是对字符串执行模式匹配的强大工具。

## 语法

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

或者更简单的方式：

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

- pattern (模式) 描述了表达式的模式
- modifiers(修饰符) 用于指定全局匹配、区分大小写的匹配和多行匹配

**注意：**当使用构造函数创造正则对象时，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+");
```

## 关注微信



## 分类导航

[HTML / CSS](#)[JavaScript](#)[服务端](#)[数据库](#)[移动端](#)[XML 教程](#)[ASP.NET](#)[Web Services](#)[反馈](#)

HTML DOM Document 对象

HTML DOM 元素对象

HTML DOM 属性对象

HTML DOM 事件对象

HTML 对象

<a>

<area>

<audio>

<base>

<blockquote>

<body>

<button>

<canvas>

<col>

<colgroup>

<datalist>

<del>

<details>

<dialog>

<embed>

<fieldset>

<form>

<iframe>

<frameset >

<img>

<ins>

<input> - button

```
var re = /\w+/;
```

更多关于 RegExp 对象请阅读我们的 [JavaScript RegExp 对象教程](#)。

修饰符

修饰符用于执行区分大小写和全局匹配:

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。

方括号

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。
[^abc]	查找任何不在方括号之间的字符。
[0-9]	查找任何从 0 至 9 的数字。
[a-z]	查找任何从小写 a 到小写 z 的字符。
[A-Z]	查找任何从大写 A 到大写 Z 的字符。
[A-z]	查找任何从大写 A 到小写 z 的字符。
[adgk]	查找给定集合内的任何字符。
[^adgk]	查找给定集合外的任何字符。
(red blue green)	查找任何指定的选项。

开发工具

网站建设

<input> - checkbox  
 <input> - color  
 <input> - date  
 <input> - datetime  
 <input> - datetime-local  
 <input> - email  
 <input> - file  
 <input> - hidden  
 <input> - image  
 <input> - month  
 <input> - number  
 <input> - range  
 <input> - password  
 <input> - radio  
 <input> - reset  
 <input> - search  
 <input> - submit  
 <input> - text  
 <input> - time  
 <input> - url  
 <input> - week  
 <keygen>  
 <link>  
 <label>  
 <legend>  
 <li>  
 <map>  
 <menu>  
 <menuitem>

## 元字符

元字符 ( Metacharacter ) 是拥有特殊含义的字符：

元字符	描述
.	查找单个字符，除了换行和行结束符。
\w	查找单词字符。
\W	查找非单词字符。
\d	查找数字。
\D	查找非数字字符。
\s	查找空白字符。
\S	查找非空白字符。
\b	匹配单词边界。
\B	匹配非单词边界。
\0	查找 NUL 字符。
\n	查找换行符。
\f	查找换页符。
\r	查找回车符。
\t	查找制表符。
\v	查找垂直制表符。
\xxx	查找以八进制数 xxx 规定的字符。
\xdd	查找以十六进制数 dd 规定的字符。
\uxxxx	查找以十六进制数 xxxx 规定的 Unicode 字符。

```

<meta>
<meter>
<object>
<ol>
<optgroup>
<option>
<param>
<progress>
<q>
<script>
<select>
<source>
<style>
<table>
<td>
<th>
<tr>
<textarea>
<title>
<time>
<track>
<video>

```

## 量词

量词	描述
<u><a href="#">n+</a></u>	匹配任何包含至少一个 n 的字符串。
<u><a href="#">n*</a></u>	匹配任何包含零个或多个 n 的字符串。
<u><a href="#">n?</a></u>	匹配任何包含零个或一个 n 的字符串。
<u><a href="#">n{X}</a></u>	匹配包含 X 个 n 的序列的字符串。
<u><a href="#">n{X,Y}</a></u>	匹配包含 X 或 Y 个 n 的序列的字符串。
<u><a href="#">n{X,}</a></u>	匹配包含至少 X 个 n 的序列的字符串。
<u><a href="#">n\$</a></u>	匹配任何结尾为 n 的字符串。
<u><a href="#">^n</a></u>	匹配任何开头为 n 的字符串。
<u><a href="#">?=n</a></u>	匹配任何其后紧接指定字符串 n 的字符串。
<u><a href="#">?!n</a></u>	匹配任何其后没有紧接指定字符串 n 的字符串。

## RegExp 对象方法

方法	描述	FF	IE
<u><a href="#">compile</a></u>	编译正则表达式。	1	4
<u><a href="#">exec</a></u>	检索字符串中指定的值。返回找到的值，并确定其位置。	1	4
<u><a href="#">test</a></u>	检索字符串中指定的值。返回 true 或 false。	1	4

## 支持正则表达式的 String 对象的方法

方法	描述	FF	IE
<u><a href="#">search</a></u>	检索与正则表达式相匹配的值。	1	4
<u><a href="#">match</a></u>	找到一个或多个正则表达式的匹配。	1	4

<a href="#">replace</a>	替换与正则表达式匹配的子串。	1	4
<a href="#">split</a>	把字符串分割为字符串数组。	1	4

[← JavaScript String 对象](#)[JavaScript 全局属性/函数 →](#)

### 在线实例

- [HTML 实例](#)
- [CSS 实例](#)
- [JavaScript 实例](#)
- [Ajax 实例](#)
- [jQuery 实例](#)
- [XML 实例](#)
- [Java 实例](#)

### 字符集&工具

- [HTML 字符集设置](#)
- [HTML ASCII 字符集](#)
- [HTML ISO-8859-1](#)
- [HTML 实体符号](#)
- [HTML 拾色器](#)
- [JSON 格式化工具](#)

### 最新更新

- [Python3 File te...](#)
- [Python3 File se...](#)
- [Python3 File re...](#)
- [Python3 File re...](#)
- [Python3 File re...](#)
- [Python3 File ne...](#)
- [Python3 File is...](#)

### 站点信息

- [意见反馈](#)
- [免责声明](#)
- [关于我们](#)
- [文章归档](#)

### 关注微信

