# **Expresiones Regulares**

Las expresiones regulares son patrones que se utilizan para hacer coincidir combinaciones de caracteres en cadenas. En JavaScript, las expresiones regulares también son objetos. Estos patrones se utilizan con los métodos <a href="mailto:exec()">exec()</a> y <a href="mailto:test()">test()</a> de <a href="mailto:RegExp">RegExp</a>, y con <a href="mailto:matchAll()</a>, <a href="mailto:replace()">replace()</a>, <a href="mailto:replace()">replace()</a>, <a href="mailto:search()">search()</a> y <a href="mailto:splitt()">split()</a> métodos de <a href="mailto:string">String</a>. Este capítulo describe las expresiones regulares de JavaScript.

# Crear una expresión regular

Construyes una expresión regular en una de estas dos formas:

• Usando una expresión regular literal, que consiste en un patrón encerrado entre barras, como sigue:

```
let re = /ab+c/;
```

Las expresiones regulares literales proporcionan la compilación de la expresión regular cuando se carga el script. Si la expresión regular permanece constante, su uso puede mejorar el rendimiento.

• O llamando a la función constructora del objeto RegExp, de la siguiente manera:

```
let re = new RegExp('ab+c');
```

El uso de la función constructora proporciona una compilación en tiempo de ejecución de la expresión regular. Usa la función constructora cuando sepas que el patrón de la expresión regular cambiará, o no conoces el patrón y lo obtienes de otra fuente, como la entrada del usuario.

# Escribir un patrón de expresión regular

Un patrón de expresión regular se compone de caracteres simples, como /abc/, o una combinación de caracteres simples y especiales, como /ab\*c/ o /Capítulo (\d)\.\d\*/. El último ejemplo incluye paréntesis, que se utilizan como dispositivos de memoria. La coincidencia realizada con esta parte del patrón se recuerda para su uso posterior, como se describe en <u>Uso de grupos</u>.

**Nota**: Si ya estás familiarizado con las formas de una expresión regular, también puedes leer <u>la hoja de</u> referencia para una búsqueda rápida de un patrón/construcción específica.

## Usar patrones simples

Los patrones simples se construyen con caracteres para los que deseas encontrar una coincidencia directa. Por ejemplo, el patrón /abc/ coincide con combinaciones de caracteres en cadenas solo cuando ocurre la secuencia exacta "abc" (todos los caracteres juntos y en ese orden). Tal coincidencia tendría éxito en las cadenas "Hola, ¿conoces tu abc?" y

"Los últimos diseños de aviones evolucionaron a partir de slabcraft". En ambos casos, la coincidencia es con la subcadena "abc". No hay ninguna coincidencia en la cadena "Grab crab" porque aunque contiene la subcadena "ab c", no contiene la subcadena "abc" exacta.

## Usar caracteres especiales

Cuando la búsqueda de una coincidencia requiere algo más que una coincidencia exacta, como por ejemplo buscar una o más 'b', o encontrar espacios en blanco, puedes incluir caracteres especiales en el patrón. Por ejemplo, para hacer coincidir *una sola "a" seguida de cero o más "b" s seguidas de "c"*, usarías el patrón /ab\*c/: el \* después de "b" significa "0 o más apariciones del elemento anterior". En la cadena "cbbabbbbcdebc", este patrón coincidirá con la subcadena "abbbbc".

Las siguientes páginas proporcionan listas de los diferentes caracteres especiales que encajan en cada categoría, junto con descripciones y ejemplos.

#### **Aserciones**

Las aserciones incluyen límites, que indican el comienzo y el final de líneas y palabras, y otros patrones que indican de alguna manera que el reconocimiento es posible (incluidas las expresiones anticipadas, inversas y condicionales).

#### Clases de caracteres

Distingue diferentes tipos de caracteres. Por ejemplo, distinguir entre letras y dígitos.

#### Grupos y rangos

Indica grupos y rangos de caracteres de expresión.

#### **Cuantificadores**

Indica el número de caracteres o expresiones que deben coincidir.

#### Escapes de propiedades Unicode

Distinguir según las propiedades de los caracteres Unicode, por ejemplo, letras mayúsculas y minúsculas, símbolos matemáticos y de puntuación.

Si deseas ver todos los caracteres especiales que se pueden usar en expresiones regulares en una sola tabla, consulta lo siguiente:

### Caracteres especiales en expresiones regulares.

Caracteres/construcciones	Artículo correspondiente
., \cX, \d, \D, \f, \n, \r, \s, \S, \t, \v, \w, \W, \0, \xhh, \uhhhh, \uhhhhh, [\b]	Clases de caracteres
^, \$, x(?=y), x(?!y), (?<=y)x, (? y)x, \b, \B</td <td>Aserciones</td>	Aserciones
(x), (?:x), (? <name>x), x y, [xyz], [^xyz], \Number</name>	<u>Grupos y rangos</u>
*, +, ?, x{n}, x{n,}, x{n,m}	Cuantificadores
\p{UnicodeProperty}, \P{UnicodeProperty}	Escapes de propiedades Unicode

**Nota**: <u>También está disponible una hoja de referencia más grande</u> (solo agregando partes de esos artículos individuales).

# Escapando

Si necesitas usar literalmente cualquiera de los caracteres especiales (en realidad buscando un "\*", por ejemplo), lo debes escapar colocando una barra invertida delante de él. Por ejemplo, para buscar "a" seguido de "\*" seguido de "b", usarías /a\\*b/ — la barra invertida "escapa" de "\*", volviéndola literal en lugar de especial.

De manera similar, si estás escribiendo un literal de expresión regular y necesitas buscar una barra inclinada ("/"), la debes escapar (de lo contrario, esta termina el patrón). Por ejemplo, para buscar la cadena "/ejemplo/" seguida de uno o más caracteres alfabéticos, usarías /\/ejemplo\/[a-z]+/i: las barras invertidas antes de cada barra, las hace literales.

Para hacer coincidir una barra invertida literal, debes escapar de la barra invertida. Por ejemplo, para encontrar la cadena "C:\" donde "C" puede ser cualquier letra, usarías /[A-Z]:\\/ — la primera barra invertida escapa a la que sigue, por lo que la expresión busca una sola barra invertida literal.

Si usas el constructor RegExp con un literal de cadena, recuerda que la barra invertida es un escape en los literales de cadena, por lo que para usarlo en la expresión regular, debes escapar en el nivel del literal de cadena. /a\\*b/ y new RegExp("a\\\*b") crean la misma expresión, que busca "a" seguida de un "\*" literal seguido de "b".

Si las cadenas de escape aún no forman parte de tu patrón, puedes agregarlas usando <a href="String.replace">String.replace</a>:

```
function escapeRegExp(string) {
  return string.replace(/[.*+\-?^${}()|[\]\\]/g,'\\$&'); // $& significa toda
}
```

La "g" después de la expresión regular es una opción o indicador que realiza una búsqueda global, buscando en toda la cadena y devolviendo todas las coincidencias. Se explica en detalle a continuación en **Búsqueda avanzada con indicadores**.

¿Por qué no está integrada en JavaScript? Existe una propuesta para agregar esta función a RegExp, pero fue rechazada por TC39. ☐

# Usando paréntesis

Los paréntesis alrededor de cualquier parte del patrón de expresión regular hacen que se recuerde esa parte de la subcadena coincidente. Una vez reconocida, la subcadena se puede recuperar para otro uso. Consulta <u>Grupos y rangos</u> para obtener más detalles.

# Usar expresiones regulares en JavaScript

Las expresiones regulares se utilizan con los métodos RegExp test() y exec() y con los métodos de String, match(), replace(), search() y split(). Estos métodos se explican en detalle en la referencia de JavaScript.

### Métodos que usan expresiones regulares

Método	Descripción
exec()	Ejecuta una búsqueda por una coincidencia en una cadena. Devuelve un arreglo de información o null en una discrepancia.
test()	Prueba una coincidencia en una cadena. Devuelve true o false.

Método	Descripción
match()	Devuelve un arreglo que contiene todas las coincidencias, incluidos los grupos de captura, o null si no se encuentra ninguna coincidencia.
matchAll()	Devuelve un iterador que contiene todas las coincidencias, incluidos los grupos de captura.
search()	Prueba una coincidencia en una cadena. Devuelve el índice de la coincidencia, o -1 si la búsqueda falla.
replace()	Ejecuta una búsqueda por una coincidencia en una cadena y reemplaza la subcadena coincidente con una subcadena de reemplazo.
replaceAll() (en-US)	Ejecuta una búsqueda de todas las coincidencias en una cadena y reemplaza las subcadenas coincidentes con una subcadena de reemplazo.
split()	Utiliza una expresión regular o una cadena fija para dividir una cadena en un arreglo de subcadenas.

Cuando desees saber si un patrón se encuentra en una cadena, utiliza los métodos test() o search(); para obtener más información (pero una ejecución más lenta) utiliza los métodos exec() o match(). Si usas exec() o match() y si la búsqueda tiene éxito, estos métodos devuelven un arreglo y actualizan las propiedades del objeto expresión regular asociado y también del objeto de expresión regular predefinido, el objeto RegExp. Si la búsqueda falla, el método exec() devuelve null (que coacciona a false).

En el siguiente ejemplo, el script utiliza el método exec () para encontrar una coincidencia en una cadena.

```
var myRe = /d(b+)d/g;
var myArray = myRe.exec('cdbbdbsbz');
```

Si no necesitas acceder a las propiedades de la expresión regular, una forma alternativa de crear myArray es con este script:

```
var myArray = /d(b+)d/g.exec('cdbbdbsbz');
   // similar a "cdbbdbsbz" .match(/d(b+)d/g); sin embargo,
   // "cdbbdbsbz" .match (/d(b+)d/g) genera Array ["dbbd"], mientras
   // /d(b+)d/g.exec('cdbbdbsbz') produce Array ['dbbd', 'bb', index: 1, input:
```

(Consulta <u>diferentes comportamientos</u> para obtener más información sobre los diferentes comportamientos).

Si deseas construir la expresión regular a partir de una cadena, otra alternativa más es este script:

```
var myRe = new RegExp('d(b+)d', 'g');
var myArray = myRe.exec('cdbbdbsbz');
```

Con estos scripts, la búsqueda se realiza correctamente, devuelve el arreglo y actualiza las propiedades que se muestran en la siguiente tabla.

## Resultado de la ejecución de expresiones regulares.

Objeto	Propiedad o índice	Descripción	En este ejemplo
		La cadena coincidente y todas las subcadenas recordadas.	['dbbd', 'bb', index: 1, input: 'cdbbdbsbz']
myArray	índice	El índice basado en 0 de la coincidencia en la cadena de entrada.	1
	entrada	La cadena original.	'cdbbdbsbz'
	[0]	Los últimos caracteres encontrados.	'dbbd'

Objeto	Propiedad o índice	Descripción	En este ejemplo
myRe	lastIndex	El índice en el que comenzará la siguiente búsqueda. (Esta propiedad se establece solo si la expresión regular usa la opción g , descrita en Búsqueda avanzada con banderas).	5
	fuente	El texto del patrón. Actualizado en el momento en que se crea la expresión regular, no se ejecuta.	'd(b+)d'

Como se muestra en la segunda forma de este ejemplo, puedes usar una expresión regular creada con un iniciador de objeto sin asignarla a una variable. Sin embargo, si lo hace, cada aparición es una nueva expresión regular. Por este motivo, si utilizas esta forma sin asignarla a una variable, no podrás acceder posteriormente a las propiedades de esa expresión regular. Por ejemplo, supongamos que tienes este script:

```
var myRe = /d(b+)d/g;
var myArray = myRe.exec('cdbbdbsbz');
console.log('El valor de lastIndex es ' + myRe.lastIndex);
```



```
// "El valor de lastIndex es 5"
```

Sin embargo, si tienes este script:

```
var myArray = /d(b+)d/g.exec('cdbbdbsbz');
console.log('El valor de lastIndex es ' + /d(b+)d/g.lastIndex);

// "El valor de lastIndex es 0"
```

Las apariciones de /d(b+)d/g en las dos declaraciones son objetos de expresión regular diferentes y, por lo tanto, tienen valores diferentes para su propiedad lastIndex. Si necesitas acceder a las propiedades de una expresión regular creada con un iniciador de objeto, primero debes asignarla a una variable.

## <u>Búsqueda avanzada con banderas</u>

Las expresiones regulares tienen seis indicadores opcionales que permiten funciones como la búsqueda global y que no distinga entre mayúsculas y minúsculas. Estos indicadores se pueden usar por separado o juntos en cualquier orden y se incluyen como parte de la expresión regular.

### Indicadores de expresión regular

Bandera	Descripción	Propiedad correspondiente	
g	Búsqueda global.	RegExp.prototype.global (en-US)	
i	Búsqueda que no distingue entre mayúsculas y minúsculas.  RegExp.prototype.ignoreCase		
m	Búsqueda multilínea.	RegExp.prototype.multiline (en-US)	
S	Permite que el . coincida con caracteres de nueva línea.	RegExp.prototype.dotAll (en-US)	
u	"unicode"; tratar un patrón como una secuencia de puntos de código Unicode.	RegExp.prototype.unicode (en-US)	
У	Realiza una búsqueda "pegajosa" que coincida a partir de la posición actual en la cadena de destino. Consulta <a href="sticky">sticky</a> (en-US).	<u>RegExp.prototype.sticky (en-US)</u>	

Para incluir una bandera con la expresión regular, usa esta sintaxis:

```
var re = /patrón/banderas;
```

0

```
var re = new RegExp('patrón', 'banderas');
```

Ten en cuenta que las banderas son parte integral de una expresión regular. No se pueden agregar ni eliminar más tarde.

Por ejemplo, re = /\w+\s/g crea una expresión regular que busca uno o más caracteres seguidos de un espacio y busca esta combinación en toda la cadena.

```
var re = /\w+\s/g;
var str = 'fee fi fo fum';
var myArray = str.match(re);
console.log(myArray);

// ["fee ", "fi ", "fo "]
```

Podrías reemplazar la línea:

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

con:

```
var re = new RegExp('\\w+\\s', 'g');
```

y obtener el mismo resultado.

El comportamiento asociado con el indicador g es diferente cuando se usa el método .exec(). Los roles de "clase" y "argumento" se invierten: En el caso de .match(), la clase cadena (o tipo de datos) posee el método y la expresión regular es solo un argumento, mientras que en el caso de .exec(), es la expresión regular la que posee el método, siendo la cadena el argumento. Compara esto str.match(re) con re.exec(str). El indicador g se usa con el método .exec() para obtener una progresión iterativa.

```
var xArray; while(xArray = re.exec(str)) console.log(xArray);
// produce:
// ["fee ", index: 0, input: "fee fi fo fum"]
```

```
// ["fi ", index: 4, input: "fee fi fo fum"]
// ["fo ", index: 7, input: "fee fi fo fum"]
```

La bandera m se utiliza para especificar que una cadena de entrada de varias líneas se debe tratar como varias líneas. Si se usa el indicador m, ^ y \$ coinciden al principio o al final de cualquier línea dentro de la cadena de entrada en lugar del inicio o el final de toda la cadena.

# **Ejemplos**

Nota: También hay varios ejemplos disponibles en:

- Las páginas de referencia para exec(), test(), match(), match(), search(),
   replace(), split()
- Artículos de esta guía: <u>clases de caracteres</u>, <u>aserciones</u>, <u>grupos y rangos</u>, <u>cuantificadores</u>, <u>escapes</u>
   <u>de propiedades Unicode</u>

Usar caracteres especiales para verificar la entrada

Report problems with this compatibility data on GitHub ☐

RegExp	
Chrome	1
Edge	12
Firefox	1
Internet Explorer	4
Opera	5
Safari	1
WebView Android	1
Chrome Android	18
Firefox for Android	4
Opera Android	10.1
Safari on iOS	1
Samsung Internet	1.0

Deno	1.0
Node.js	0.10.0
RegExp() constructor	
Chrome	1
Edge	12
Firefox	1
Internet Explorer	4
Opera	5
Safari	1
WebView Android	1
Chrome Android	18
Firefox for Android	4
Opera Android	10.1
Safari on iOS	1
Samsung Internet	1.0
Deno	1.0
Node.js	0.10.0
compile	
Chrome	1
Edge	12
Firefox	1
Internet Explorer	4
Opera	6
Safari	3.1
WebView Android	1
Chrome Android	18
Firefox for Android	4

Opera Android	10.1
Safari on iOS	2
Samsung Internet	1.0
Deno	1.0
Node.js	0.10.0
dotAll	
Chrome	62
Edge	79
Firefox	78
Internet Explorer	No
Opera	49
Safari	12
WebView Android	62
Chrome Android	62
Firefox for Android	79
Opera Android	46
Safari on iOS	12
Samsung Internet	8.0
Deno	1.0
Node.js	8.10.0
<u>exec</u>	
Chrome	1
Edge	12
Firefox	1
Internet Explorer	4
Opera	5
Safari	1
WebView Android	1

Chrome Android	18
Firefox for Android	4
Opera Android	10.1
Safari on iOS	1
Samsung Internet	1.0
Deno	1.0
Node.js	0.10.0
<u>flags</u>	
Chrome	49
Edge	79
Firefox	37
Internet Explorer	No
Opera	39
Safari	9
WebView Android	49
Chrome Android	49
Firefox for Android	37
Opera Android	41
Safari on iOS	9
Samsung Internet	5.0
Deno	1.0
Node.js	6.0.0
global	
Chrome	1
Edge	12
Firefox	1
Internet Explorer	5.5
Opera	5

Safari	1
WebView Android	1
Chrome Android	18
Firefox for Android	4
Opera Android	10.1
Safari on iOS	1
Samsung Internet	1.0
Deno	1.0
Node.js	0.10.0
Prototype accessor property (ES2015)	
Chrome	48
Edge	12
Firefox	38
Internet Explorer	5.5
Opera	35
Safari	1.3
WebView Android	48
Chrome Android	48
Chrome Android Firefox for Android	38
Firefox for Android	38
Firefox for Android  Opera Android	38 35
Firefox for Android  Opera Android  Safari on iOS	38 35 1
Firefox for Android  Opera Android  Safari on iOS  Samsung Internet	38 35 1 5.0
Firefox for Android  Opera Android  Safari on iOS  Samsung Internet  Deno	38 35 1 5.0 1.0
Firefox for Android  Opera Android  Safari on iOS  Samsung Internet  Deno  Node.js	38 35 1 5.0 1.0
Firefox for Android  Opera Android  Safari on iOS  Samsung Internet  Deno  Node.js  hasIndices	38 35 1 5.0 1.0 6.0.0

Internet Explorer	X	No
Opera	X	No
Safari		15
WebView Android		90
Chrome Android		90
Firefox for Android		88
Opera Android	X	No
Safari on iOS		15
Samsung Internet		15.0
Deno		1.8
Node.js	X	No
<u>ignoreCase</u>		
Chrome		1
Edge		12
Firefox		1
Internet Explorer		5.5
Opera		5
Safari		1
WebView Android		1
Chrome Android		18
Firefox for Android		4
Opera Android		10.1
Safari on iOS		1
Samsung Internet		1.0
Deno		1.0
Node.js		0.10.0
Prototype accessor property (ES2015)		
Chrome		48

Edge	12
Firefox	38
Internet Explorer	5.5
Opera	35
Safari	1.3
WebView Android	48
Chrome Android	48
Firefox for Android	38
Opera Android	35
Safari on iOS	1
Samsung Internet	5.0
Deno	1.0
Node.js	6.0.0
( <u>\$</u> ).	
Chrome	1
Edge	12
Firefox	1
Internet Explorer	5.5
Opera	15
Safari	3
WebView Android	1
Chrome Android	18
Firefox for Android	4
Opera Android	14
Safari on iOS	1
Samsung Internet	1.0

Deno	1.0
Node.js	0.10.0
<u>lastIndex</u>	
Chrome	1
Edge	12
Firefox	1
Internet Explorer	5.5
Opera	5
Safari	1
WebView Android	1
Chrome Android	18
Firefox for Android	4
Opera Android	10.1
Safari on iOS	1
Samsung Internet	1.0
Deno	1.0
Node.js	0.10.0
RegExp.lastMatch ( \$& )	
Chrome	1
Edge	12
Firefox	1
Internet Explorer	5.5
Opera	10.5
Safari	3
WebView Android	1
Chrome Android	18

Firefox for Android	4
Opera Android	11
Safari on iOS	1
Samsung Internet	1.0
Deno	1.0
Node.js	0.10.0
RegExp.lastParen (  ( \$+	
). 	
Chrome	1
Edge	12
Firefox	1
Internet Explorer	5.5
Opera	10.5
Safari	3
WebView Android	1
Chrome Android	18
Firefox for Android	4
Opera Android	11
Safari on iOS	1
Samsung Internet	1.0
Deno	1.0
Node.js	0.10.0
RegExp.leftContext (  \$` )	
Chrome	1
Edge	12

Firefox	1
Internet Explorer	5.5
Opera	8
Safari	3
WebView Android	1
Chrome Android	18
Firefox for Android	4
Opera Android	10.1
Safari on iOS	1
Samsung Internet	1.0
Deno	1.0
Node.js	0.10.0
(?<= ) and (? ) )</th <th></th>	
Chrome	62
Edge	
Edge	79
Firefox	79 78
Firefox	78
Firefox  Internet Explorer	78 No
Firefox  Internet Explorer  Opera	78 No 49
Internet Explorer Opera Safari	78 No 49 No
Internet Explorer Opera Safari WebView Android	78 No 49 No 62
Firefox  Internet Explorer  Opera  Safari  WebView Android  Chrome Android	78 No 49 No 62
Internet Explorer Opera Safari WebView Android Chrome Android Firefox for Android	78 No 49 No 62 62 79
Internet Explorer Opera Safari WebView Android Chrome Android Firefox for Android Opera Android	78 No 49 No 62 62 79

Deno	1.0
Node.js	8.10.0
multiline	
Chrome	1
Edge	12
Firefox	1
Internet Explorer	5.5
Opera	5
Safari	1
WebView Android	1
Chrome Android	18
Firefox for Android	4
Opera Android	10.1
Safari on iOS	1
Samsung Internet	1.0
Deno	1.0
Node.js	0.10.0
Prototype accessor property (ES2015)	
Chrome	48
Edge	12
Firefox	38
Internet Explorer	5.5
Opera	35
Safari	1.3
WebView Android	48
Chrome Android	48
Firefox for Android	38
Opera Android	35

Safari on iOS	1
Samsung Internet	5.0
Deno	1.0
Node.js	6.0.0
RegExp.\$1-\$9	
Chrome	1
Edge	12
Firefox	1
Internet Explorer	4
Opera	5
Safari	1
WebView Android	1
Chrome Android	18
Firefox for Android	4
Opera Android	10.1
Safari on iOS	1
Samsung Internet	1.0
Deno	1.0
Node.js	0.10.0
Named capture groups	
Chrome	64
Edge	79
Firefox	78
Internet Explorer	No
Opera	51
Safari	11.1
WebView Android	64
Chrome Android	64

Firefox for Android	79
Opera Android	47
Safari on iOS	11.3
Samsung Internet	9.0
Deno	1.0
Node.js	10.0.0
Unicode property escapes (  )	
Chrome	64
Eige	79
Firefox  Internet Evaluation	78 No
Internet Explorer	
Opera	51
Safari	11.1
WebView Android	64
Chrome Android	64
Firefox for Android	79
Opera Android	47
Safari on iOS	11.3
Samsung Internet	9.0
Deno	1.0
Node.js	10.0.0
RegExp.rightContext (  ( \$' )	
Chrome	1
Edge	12

Firefox	1
Internet Explorer	5.5
Opera	8
Safari	3
WebView Android	1
Chrome Android	18
Firefox for Android	4
Opera Android	10.1
Safari on iOS	1
Samsung Internet	1.0
Deno	1.0
Node.js	0.10.0
<u>source</u>	
Chrome	1
Edge	12
Firefox	1
Internet Explorer	4
Opera	5
Safari	1
WebView Android	1
Chrome Android	18
Firefox for Android	4
Opera Android	10.1
Safari on iOS	1
Samsung Internet	1.0
Deno	1.0
Node.js	0.10.0
"(?:)" for empty regexps	

Chrome	6
Edge	12
Firefox	38
Internet Explorer	No
Opera	15
Safari	5
WebView Android	37
Chrome Android	18
Firefox for Android	38
Opera Android	14
Safari on iOS	4.2
Samsung Internet	1.0
Deno	1.0
Node.js	0.10.0
Escaping	
Chrome	73
Edge	12
Firefox	38
Internet Explorer	10
Opera	60
Safari	6
WebView Android	73
Chrome Android	73
Firefox for Android	38
Opera Android	52
Safari on iOS	6
Samsung Internet	11.0

Node.js	12.0.0
Prototype accessor property (ES2015)	
Chrome	48
Edge	12
Firefox	41
Internet Explorer	4
Opera	35
Safari	1.3
WebView Android	48
Chrome Android	48
Firefox for Android	41
Opera Android	35
Safari on iOS	1
Samsung Internet	5.0
Deno	1.0
Node.js	6.0.0
<u>sticky</u>	
Chrome	49
Edge	13
Firefox	3
Internet Explorer	No
Opera	36
Safari	10
WebView Android	49
Chrome Android	49
Firefox for Android	4
Opera Android	36
Safari on iOS	10

Samsung Internet	5.0
Deno	1.0
Node.js	6.0.0
Anchored sticky flag behavior per ES2015	
Chrome	49
Edge	13
Firefox	44
Internet Explorer	No
Opera	36
Safari	10
WebView Android	49
Chrome Android	49
Firefox for Android	44
Opera Android	36
Safari on iOS	10
Samsung Internet	5.0
Deno	1.0
Node.js	6.0.0
Prototype accessor property (ES2015)	
Chrome	49
Edge	13
Firefox	38
Internet Explorer	No
Opera	36
Safari	10
WebView Android	49
Chrome Android	49
Firefox for Android	38

Opera Android	36
Safari on iOS	10
Samsung Internet	5.0
Deno	1.0
Node.js	6.0.0
<u>test</u>	
Chrome	1
Edge	12
Firefox	1
Internet Explorer	4
Opera	5
Safari	1
WebView Android	1
Chrome Android	18
Firefox for Android	4
Opera Android	10.1
Safari on iOS	1
Samsung Internet	1.0
Deno	1.0
Node.js	0.10.0
toSource A	
Chrome	No
Edge	No
Firefox	1–74 🛨
Internet Explorer	No
Opera	No
Safari	No
WebView Android	No

Chrome Android	X	No
Firefox for Android		4
Opera Android	X	No
Safari on iOS	X	No
Samsung Internet	X	No
Deno	$\sim$	No
Node.js		No
toString		
Chrome		1
Edge		12
Firefox		1
Internet Explorer		4
Opera		5
Safari		1
WebView Android		1
Chrome Android		18
Firefox for Android		4
Opera Android		10.1
Safari on iOS		1
Samsung Internet		1.0
Deno		1.0
Node.js		0.10.0
Escaping		
Chrome		73
Edge		12
Firefox		38
Internet Explorer		9
Opera		60

Safari	6
WebView Android	73
Chrome Android	73
Firefox for Android	38
Opera Android	52
Safari on iOS	6
Samsung Internet	11.0
Deno	1.0
Node.js	12.0.0
unicode	
Chrome	50
Edge	12 🛨
Firefox	46
Internet Explorer	No
Opera	<b>No</b> 37
Opera	37
Opera Safari	10
Opera Safari WebView Android	37 10 50
Opera Safari WebView Android Chrome Android	37 10 50 50
Opera Safari WebView Android Chrome Android Firefox for Android	37 10 50 50 46
Opera Safari WebView Android Chrome Android Firefox for Android Opera Android	37 10 50 50 46 37
Opera Safari WebView Android Chrome Android Firefox for Android Opera Android Safari on iOS	37 10 50 50 46 37
Opera Safari WebView Android Chrome Android Firefox for Android Opera Android Safari on iOS Samsung Internet	37 10 50 50 46 37 10
Opera Safari WebView Android Chrome Android Firefox for Android Opera Android Safari on iOS Samsung Internet Deno	37 10 50 50 46 37 10 5.0 1.0
Opera Safari WebView Android Chrome Android Firefox for Android Opera Android Safari on iOS Samsung Internet Deno Node.js	37 10 50 50 46 37 10 5.0 1.0
Opera Safari  WebView Android Chrome Android Firefox for Android Opera Android Safari on iOS Samsung Internet Deno Node.js	37 10 50 50 46 37 10 5.0 1.0 6.0.0 ★

Internet Explorer	No
Opera	37
Safari	10
WebView Android	50
Chrome Android	50
Firefox for Android	49
Opera Android	37
Safari on iOS	10
Samsung Internet	5.0
Deno	1.0
Node.js	6.0.0
@@matchAll	
Chrome	73
Edge	79
Firefox	67
	67 <b>No</b>
Firefox	
Firefox Internet Explorer	No
Firefox  Internet Explorer  Opera	<b>No</b> 60
Firefox Internet Explorer Opera Safari	No 60 13
Internet Explorer Opera Safari WebView Android	No  60  13  73
Firefox  Internet Explorer  Opera  Safari  WebView Android  Chrome Android	No 60 13 73 73
Internet Explorer Opera Safari WebView Android Chrome Android Firefox for Android	No  60  13  73  73  67
Internet Explorer Opera Safari WebView Android Chrome Android Firefox for Android Opera Android	No  60  13  73  73  67  52
Internet Explorer Opera Safari WebView Android Chrome Android Firefox for Android Opera Android Safari on iOS	No  60  13  73  73  67  52  13
Internet Explorer Opera Safari WebView Android Chrome Android Firefox for Android Opera Android Safari on iOS Samsung Internet	No         60         13         73         73         67         52         13         5.0
Internet Explorer Opera Safari WebView Android Chrome Android Firefox for Android Opera Android Safari on iOS Samsung Internet Deno	No  60  13  73  73  67  52  13  5.0  1.0

Firefox         49           Internet Explorer         No           Opera         37           Safari         10           WebView Android         50           Chrome Android         50           Firefox for Android         49           Opera Android         37           Safari on IOS         10           Samsung Internet         5.0           Deno         1.0           Node.js         6.0.0           S#Search         50           Edge         13           Firefox         49           Internet Explorer         No           Opera         37           Safari         10           WebView Android         50           Chrome Android         50           Firefox for Android         49           Opera Android         37           Safari on IOS         10           Samsung Internet         5.0           Deno         1.0           Node.js         6.0.0	Edge	79
Opera         37           Safari         10           WebView Android         50           Chrome Android         50           Firefox for Android         49           Opera Android         37           Safari on IOS         10           Samsung Internet         5.0           Deno         1.0           Node Js         6.0.0           Øesearch         13           Edge         13           Firefox         49           Internet Explorer         No           Opera         37           Safari         10           WebView Android         50           Chrome Android         50           Firefox for Android         49           Opera Android         49           Opera Android         37           Safari on IOS         10           Samsung Internet         5.0           Deno         1.0	Firefox	49
Safari         10           WebView Android         50           Chrome Android         50           Firefox for Android         49           Opera Android         37           Safari on iOS         10           Samsung Internet         5.0           Deno         1.0           Node js         6.0.0           @search         60.0           Edge         13           Firefox         49           Internet Explorer         No           Opera         37           Safari         10           WebView Android         50           Firefox for Android         49           Opera Android         37           Safari on iOS         10           Samsung Internet         5.0           Deno         1.0	Internet Explorer	No
WebView Android         50           Chrome Android         50           Firefox for Android         49           Opera Android         37           Safari on IOS         10           Samsung Internet         5.0           Deno         1.0           Node.js         6.0.0           Mesearch         6.0.0           Edge         13           Firefox         49           Internet Explorer         No           Opera         37           Safari         10           WebView Android         50           Chrome Android         50           Firefox for Android         49           Opera Android         37           Safari on IOS         10           Samsung Internet         5.0           Deno         1.0	Opera	37
Chrome Android         50           Firefox for Android         48           Opera Android         37           Safari on iOS         10           Samsung Internet         5.0           Deno         1.0           Node.js         6.0.0           ##Search         60           Edge         13           Firefox         49           Internet Explorer         No           Opera         37           Safari         10           WebView Android         50           Chrome Android         50           Firefox for Android         49           Opera Android         37           Safari on IOS         10           Samsung Internet         5.0           Deno         1.0	Safari	10
Firefox for Android         49           Opera Android         37           Safari on iOS         10           Samsung Internet         5.0           Deno         1.0           Node Js         6.0.0           ### Search         6.0.0           Edge         13           Firefox         49           Internet Explorer         No           Opera         37           Safari         10           WebView Android         50           Chrome Android         50           Firefox for Android         49           Opera Android         37           Safari on iOS         10           Samsung Internet         5.0           Deno         1.0	WebView Android	50
Opera Android         37           Safari on IOS         10           Samsung Internet         5.0           Deno         1.0           Node js         6.0.0           @search           Chrome         50           Edge         13           Firefox         49           Internet Explorer         No           Opera         37           Safari         10           WebView Android         50           Firefox for Android         49           Opera Android         37           Safari on iOS         10           Samsung Internet         5.0           Deno         1.0	Chrome Android	50
Safari on IOS         10           Samsung Internet         5.0           Deno         1.0           Node js         6.0.0           Besearch           Chrome         50           Edge         13           Firefox         49           Internet Explorer         No           Opera         37           Safari         10           WebView Android         50           Chrome Android         50           Firefox for Android         49           Opera Android         37           Safari on IOS         10           Samsung Internet         5.0           Deno         1.0	Firefox for Android	49
Samsung Internet         5.0           Deno         1.0           Node.js         6.0.0           @@sear.ch           Chrome         50           Edge         13           Firefox         49           Internet Explorer         No           Opera         37           Safari         10           WebView Android         50           Chrome Android         50           Firefox for Android         49           Opera Android         37           Safari on IOS         10           Samsung Internet         5.0           Deno         1.0	Opera Android	37
Deno         1.0           Node.js         6.0.0           @@search           Chrome         50           Edge         13           Firefox         49           Internet Explorer         No           Opera         37           Safari         10           WebView Android         50           Chrome Android         50           Firefox for Android         49           Opera Android         37           Safari on iOS         10           Samsung Internet         5.0           Deno         1.0	Safari on iOS	10
Node.js         6.0.0           @0search         50           Edge         13           Firefox         49           Internet Explorer         No           Opera         37           Safari         10           WebView Android         50           Chrome Android         50           Firefox for Android         49           Opera Android         37           Safari on iOS         10           Samsung Internet         5.0           Deno         1.0	Samsung Internet	5.0
Chrome         50           Edge         13           Firefox         49           Internet Explorer         No           Opera         37           Safari         10           WebView Android         50           Chrome Android         50           Firefox for Android         49           Opera Android         37           Safari on iOS         10           Samsung Internet         5.0           Deno         1.0	Deno	1.0
Chrome         50           Edge         13           Firefox         49           Internet Explorer         No           Opera         37           Safari         10           WebView Android         50           Chrome Android         50           Firefox for Android         49           Opera Android         37           Safari on iOS         10           Samsung Internet         5.0           Deno         1.0	Node.js	6.0.0
Edge       13         Firefox       49         Internet Explorer       No         Opera       37         Safari       10         WebView Android       50         Chrome Android       50         Firefox for Android       49         Opera Android       37         Safari on iOS       10         Samsung Internet       5.0         Deno       1.0	@@search	
Firefox         49           Internet Explorer         No           Opera         37           Safari         10           WebView Android         50           Chrome Android         50           Firefox for Android         49           Opera Android         37           Safari on iOS         10           Samsung Internet         5.0           Deno         1.0	Chrome	50
Internet Explorer         No           Opera         37           Safari         10           WebView Android         50           Chrome Android         50           Firefox for Android         49           Opera Android         37           Safari on iOS         10           Samsung Internet         5.0           Deno         1.0	Edge	13
Opera       37         Safari       10         WebView Android       50         Chrome Android       50         Firefox for Android       49         Opera Android       37         Safari on iOS       10         Samsung Internet       5.0         Deno       1.0	Firefox	49
Safari       10         WebView Android       50         Chrome Android       50         Firefox for Android       49         Opera Android       37         Safari on iOS       10         Samsung Internet       5.0         Deno       1.0	Internet Explorer	No
WebView Android50Chrome Android50Firefox for Android49Opera Android37Safari on iOS10Samsung Internet5.0Deno1.0	Opera	37
Chrome Android 50  Firefox for Android 49  Opera Android 37  Safari on iOS 10  Samsung Internet 5.0  Deno 1.0	Safari	10
Firefox for Android 49  Opera Android 37  Safari on iOS 10  Samsung Internet 5.0  Deno 1.0	WebView Android	50
Opera Android 37 Safari on iOS 10 Samsung Internet 5.0 Deno 1.0	Chrome Android	50
Safari on iOS  Samsung Internet  5.0  Deno  1.0	Firefox for Android	49
Samsung Internet 5.0 Deno 1.0	Opera Android	37
Deno 1.0	Safari on iOS	10
	Samsung Internet	5.0
Node.js 6.0.0	Deno	1.0
	Node.js	6.0.0

## <u>@@species</u> Chrome 50 Edge 13 Firefox 49 **Internet Explorer** No Opera 37 Safari 10 WebView Android 50 Chrome Android 50 Firefox for Android 49 Opera Android 37 Safari on iOS 10 Samsung Internet 5.0 Deno 1.0 Node.js 6.5.0 @@split Chrome 50 Edge 79 Firefox 49 **Internet Explorer** No Opera 37 Safari 10 WebView Android 50 Chrome Android 50 Firefox for Android 49 Opera Android 37 Safari on iOS 10 Samsung Internet 5.0

Deno 1.0

Node.js 6.0.0

Full support

No support

Non-standard. Check cross-browser support before using.

Deprecated. Not for use in new websites.

★ See implementation notes.

User must explicitly enable this feature.

En el siguiente ejemplo, se espera que el usuario ingrese un número de teléfono. Cuando el usuario presiona el botón "Comprobar", el script verifica la validez del número. Si el número es válido (encuentra la secuencia de caracteres especificada por la expresión regular), la secuencia de comandos muestra un mensaje agradeciendo al usuario y confirmando el número. Si el número no es válido, el guión informa al usuario que el número de teléfono no es válido.

Entre paréntesis que no capturan  $(?:, la expresión regular busca tres caracteres numéricos <math>\d{3}\ O \ |$  un paréntesis izquierdo  $\c$  seguido de tres dígitos  $\d{3}\$ , seguido de un paréntesis cerrado  $\c$ ), (finaliza el paréntesis no capturador  $\c$ )), seguido de un guión, una barra diagonal o un punto decimal y cuando lo encuentre, recuerde el carácter  $([-\c])$ , seguido de tres dígitos  $\d{3}$ , seguido de la coincidencia recordada de un guión, una barra diagonal o un punto decimal  $\c$ 1, seguida de cuatro dígitos  $\c$ 4}.

El evento Change activado cuando el usuario presiona Enter establece el valor de RegExp.input.

#### HTML

## JavaScript

```
var re = /(?:\d{3}|\(\d{3}\\))([-\/\.])\d{3}\1\d{4}/;
function testInfo(phoneInput) {
  var OK = re.exec(phoneInput.value);
  if (!OK) {
    console.error(phoneInput.value + ' isn\'t a phone number with area code!');
  } else {
    console.log('Gracias, tu número de teléfono es ' + OK[0]);}
}
```

## Resultado

# Live sample failed!

An error occurred trying to render this live sample. Consider filing an issue or trying your hands at a fix of your own.

**Error details:** 

## Probador de expresiones regulares □

Un constructor/depurador de expresiones regulares en línea

## Visualizador de expresiones regulares □

Un probador de expresiones regulares visual en línea.

#### Especificación

#### ECMAScript (ECMA-262)

La definición de 'RegExp' en esta especificación.