

MDN Plus now available in your country! Support MDN and make it your own. [Learn more](#) 🌟

This page was translated from English by the community. [Learn more](#) and join the MDN Web Docs community.

Hoja de referencia de sintaxis de expresiones regulares

Esta página proporciona una hoja de referencia general de todas las capacidades de la sintaxis de RegExp agregando el contenido de los artículos en la guía RegExp . Si necesitas más información sobre un tema específico, sigue el enlace del título correspondiente para acceder al artículo completo o dirígete a [la guía](#).

Clases de caracteres

| Caracteres | Significado |
|------------|-------------|
|------------|-------------|

| Caracteres | Significado |
|-----------------|--|
| . | <p>Tiene uno de los siguientes significados:</p> <ul style="list-style-type: none"> Encuentra cualquier caracter único <i>excepto</i> terminadores de línea: <code>\n</code>, <code>\r</code>, <code>\u2028</code> o <code>\u2029</code>. Por ejemplo, <code>/.y/</code> reconoce "my" y "ay", pero no "yes", en "yes make my day". Dentro de un juego de caracteres, el punto pierde su significado especial y concuerda con un punto literal. <p>Ten en cuenta que el indicador multilínea <code>m</code> no cambia el comportamiento del punto. Por lo tanto, para buscar en un patrón multilínea, puedes usar el juego de caracteres <code>[^]</code> — este encontrará con cualquier caracter, incluidas las nuevas líneas.</p> <p>ES2018 agregó el indicador <code>s</code> <i>"dotAll"</i>, que permite que el punto también concuerde con los terminadores de línea.</p> |
| <code>\d</code> | <p>Busca cualquier dígito (número arábigo). Equivalente a <code>[0-9]</code>. Por ejemplo, <code>/\d/</code> o <code>/[0-9]/</code> encuentra el "2" en "B2 es el número de suite".</p> |
| <code>\D</code> | <p>Busca cualquier caracter que no sea un dígito (número arábigo). Equivalente a <code>[^0-9]</code>. Por ejemplo, <code>/\D/</code> o <code>/[^0-9]/</code> encuentra la "B" en "B2 es el número de suite".</p> |
| <code>\w</code> | <p>Busca cualquier caracter alfanumérico del alfabeto latino básico, incluido el caracter de subrayado. Equivalente a <code>[A-Za-z0-9_]</code>. Por ejemplo, <code>/\w/</code> encuentra la "m" en "manzana", el "5" en "\$5.28" y el "3" en "3D".</p> |

| Caracteres | Significado |
|-----------------|---|
| <code>\w</code> | Busca cualquier caracter que no sea un caracter de palabra del alfabeto latino básico. Equivalente a <code>[^A-Za-z0-9_]</code> . Por ejemplo, <code>/\w/</code> o <code>/[^A-Za-z0-9_]/</code> encuentra el caracter "%" en "50%". |
| <code>\s</code> | Busca un solo caracter de espacio en blanco, incluido el espacio, tabulación, avance de página, avance de línea y otros espacios Unicode. Equivalente a <code>[\f\n\r\t\v\u00a0\u1680\u2000-\u200a\u2028\u2029\u202f\u205f\u3000\ufeff]</code> . Por ejemplo, <code>/\s\w*/</code> reconoce " bar" en "foo bar". |
| <code>\S</code> | Busca un solo caracter que no sea un espacio en blanco. Equivalente a <code>[^ \f\n\r\t\v\u00a0\u1680\u2000-\u200a\u2028\u2029\u202f\u205f\u3000\ufeff]</code> . Por ejemplo, <code>/\S\w*/</code> encuentra "foo" en "foo bar". |



| | |
|-------------------|---|
| <code>\n</code> | Coincide con un salto de línea. |
| <code>\v</code> | Coincide con una tabulación vertical. |
| <code>\f</code> | Coincide con un caracter de avance de página. |
| <code>[\b]</code> | Coincide con un caracter de retroceso. Si estás buscando el caracter de límite de palabra (<code>\b</code>), consulta Límites (en-US) . |
| <code>\0</code> | Coincide con un caracter NUL . No sigue a este con otro dígito. |

| Caracteres | Significado |
|--|--|
| <code>\cX</code> | Coincide con un caracter de control usando notación de acento circunflejo , donde "X" es una letra de la A a la Z (correspondiente a los puntos de código U+0001 - U+001F). Por ejemplo, <code>/\cM/</code> reconoce el caracter <code>"\r"</code> en <code>"\r\n"</code> . |
| <code>\xhh</code> | Busca el caracter con el código <code>hh</code> (dos dígitos hexadecimales). |
| <code>\uhhhh</code> | Busca una unidad de código UTF-16 con el valor <code>hhhh</code> (cuatro dígitos hexadecimales). |
| <code>\u{hhhh}</code> o <code>\u{hhhhh}</code> | (Solo cuando se establece el indicador <code>u</code>). Busca el caracter con el valor Unicode <code>U+hhhh</code> o <code>U+hhhhh</code> (dígitos hexadecimales). |

| Caracteres | Significado |
|------------|--|
| \ | <p>Indica que el siguiente caracter se debe tratar de manera especial o "escaparse". Se comporta de dos formas.</p> <ul style="list-style-type: none"> Para los caracteres que generalmente se tratan literalmente, indica que el siguiente caracter es especial y no se debe interpretar literalmente. Por ejemplo, <code>/b/</code> reconoce el caracter "b". Al colocar una barra invertida delante de "b", es decir, usando <code>/\b/</code>, el caracter se vuelve especial para significar que concuerda con el límite de una palabra. Para los caracteres que generalmente se tratan de manera especial, indica que el siguiente caracter no es especial y se debe interpretar literalmente. Por ejemplo, "*" es un caracter especial que significa que deben reconocer 0 o más ocurrencias del caracter anterior; por ejemplo, <code>/a*/</code> significa reconocer 0 o más "a"s. Para emparejar el * literal, precédelo con una barra invertida; por ejemplo, <code>/a*/</code> concuerda con "a*". <p>Ten en cuenta que algunos caracteres como <code>:</code>, <code>-</code>, <code>@</code>, etc. no tienen un significado especial cuando se escapan ni cuando no se escapan. Las secuencias de escape como <code>\:</code>, <code>\-</code>, <code>\@</code> serán equivalentes a sus equivalentes de caracteres literales sin escapar en expresiones regulares. Sin embargo, en las expresiones regulares con indicador Unicode, esto provocará un error de <i>escape de identidad no válido</i>. Esto se hace para asegurar la compatibilidad con el código existente que usa nuevas secuencias de escape como <code>\p</code> o <code>\k</code>.</p> <div> <p>Para reconocer este caracter literalmente, escápalo consigo mismo. En otras palabras, para buscar <code>\</code> usa <code>/\\</code>.</p> </div> |

Aserciones

Aserciones de tipo límite

| Caracteres | Significado |
|------------|--|
| ^ | <p>Coincide con el comienzo de la entrada. Si el indicador multilínea se establece en <code>true</code> , también busca inmediatamente después de un caracter de salto de línea. Por ejemplo, <code>/^A/</code> no reconoce la "A" en "an A", pero encuentra la primera "A" en "An A".</p> <div><p>Este caracter tiene un significado diferente cuando aparece al comienzo de un <u>grupo</u>.</p></div> |
| \$ | <p>Coincide con el final de la entrada. Si el indicador multilínea se establece en <code>true</code> , también busca hasta inmediatamente antes de un caracter de salto de línea. Por ejemplo, <code>/a\$/</code> no reconoce la "t" en "eater", pero sí en "eat".</p> |

| Caracteres | Significado |
|-----------------|---|
| <code>\b</code> | <p>Marca el límite de una palabra. Esta es la posición en la que un caracter de palabra no va seguido o precedido por otro caracter de palabra, por ejemplo, entre una letra y un espacio. Ten en cuenta que el límite de una palabra encontrada no se incluye en el resultado. En otras palabras, la longitud de un límite de palabra encontrada es cero.</p> <p>Ejemplos:</p> <ul style="list-style-type: none"><code>/\bm/</code> reconoce la "m" en "moon".<code>/oo\b/</code> no reconoce "oo" en "moon", porque "oo" va seguido de "n", que es un caracter de palabra.<code>/oon\b/</code> encuentra "oon" en "moon", porque "oon" es el final de la cadena, por lo que no va seguido de un caracter de palabra.<code>/\w\b\w/</code> nunca encontrará nada, porque un caracter de palabra nunca puede ir seguido de un caracter que no sea de palabra y otro de palabra. <p>Para encontrar un caracter de retroceso (<code>[\b]</code>), consulta Clases de caracteres.</p> |
| <code>\B</code> | <p>Coincide con un límite sin palabra. Esta es una posición en la que el caracter anterior y siguiente son del mismo tipo: ambos deben ser palabras o ambos deben ser no palabras, por ejemplo, entre dos letras o entre dos espacios. El principio y el final de una cadena se consideran no palabras. Igual que el límite de palabras encontradas, el límite sin palabras reconocidas tampoco se incluye en el resultado. Por ejemplo, <code>/\Bon/</code> reconoce "on" en "at noon", y <code>/ye\B/</code> encuentra "ye" en "possibly yesterday".</p> |

Otras aserciones

Nota: El caracter `?` también se puede utilizar como cuantificador.

| Caracteres | Significado |
|-------------------------|---|
| <code>x(?:y)</code> | <p>Aserción anticipada: Coincide con "x" solo si "x" va seguida de "y". Por ejemplo, <code>/Jack(?:=Sprat)/</code> reconocerá a "Jack" solo si va seguida de "Sprat".</p> <p><code>/Jack(?:=Sprat Frost)/</code> encontrará a "Jack" solo si va seguida de "Sprat" o "Frost". Sin embargo, ni "Sprat" ni "Frost" forman parte del resultado.</p> |
| <code>x(?:!y)</code> | <p>Aserción de búsqueda anticipada negativa: reconoce la "x" solo si la "x" no va seguida de "y". Por ejemplo, <code>/\d+(?!\.)/</code> reconoce un número solo si no va seguido de un punto decimal. <code>/\d+(?!\.)/.exec('3.141')</code> halla el "141" pero no el "3".</p> |
| <code>(?<=y)x</code> | <p>Aserción de búsqueda inversa: encontrará "x" solo si "x" está precedida por "y". Por ejemplo, <code>/(?<=Jack)Sprat/</code> reconoce a "Sprat" solo si está precedido por "Jack". <code>/(?<=Jack Tom)Sprat/</code> empareja "Sprat" solo si está precedido por "Jack" o "Tom". Sin embargo, ni "Jack" ni "Tom" forman parte del resultado.</p> |
| <code>(?<!y)x</code> | <p>Aserción de búsqueda inversa negativa: Reconoce la "x" solo si "x" no está precedida por "y". Por ejemplo, <code>/(?<!\-)\d+/</code> encuentra un número solo si no está precedido por un signo menos. <code>/(?<!\-)\d+/.exec('3')</code> encuentra el "3". <code>/(?<!\-)\d+/.exec('-3')</code> no lo reconoce porque el número está precedido por el signo menos.</p> |

Grupos y rangos

| Caracteres | Significado |
|--|---|
| <code>x y</code> | Coincide con "x" o "y". Por ejemplo, <code>/verde roja/</code> reconoce el "verde" en "manzana verde" y "roja" en "manzana roja". |
| <code>[xyz]</code> <code>[a-c]</code> | <p>Un juego de caracteres. Coincide con cualquiera de los caracteres incluidos. Puedes especificar un rango de caracteres mediante el uso de un guión, pero si el guión aparece como el primero o último carácter entre corchetes, se toma como un guión literal para incluirse en el juego de caracteres como un carácter normal. También es posible incluir una clase de caracteres en un juego de caracteres.</p> <p>Por ejemplo, <code>[abcd]</code> es lo mismo que <code>[a-d]</code>. Coincide con la "b" en "brisket" y la "c" en "chop".</p> <p>Por ejemplo, <code>[abcd-]</code> y <code>[-abcd]</code> reconoce la "b" en "brisket", la "c" en "chop" y el "-" (guión) en "non-profit".</p> <p>Por ejemplo, <code>[\w-]</code> es lo mismo que <code>[A-Za-z0-9_-]</code>. Ambos reconocen la "b" en "brisket", la "c" en "chop" y la "n" en "non-profit".</p> |

| Caracteres | Significado |
|--|--|
| <code>[^xyz]</code> <code>[^a-c]</code> | <p>Un juego de caracteres negado o complementado. Es decir, hallan cualquier cosa que no esté encerrada entre corchetes. Puedes especificar un rango de caracteres mediante el uso de un guión, pero si el guión aparece como el primero o último carácter entre corchetes, se toma como un guión literal para incluirse en el juego de caracteres como un carácter normal. Por ejemplo, <code>[^abc]</code> es lo mismo que <code>[^a-c]</code>. Inicialmente halla la "o" en "bacon" y la "h" en "chuleta".</p> <div><p>El carácter <code>^</code> además puede indicar el <u>comienzo de la entrada</u> <u>(en-US)</u>.</p></div> |
| <code>(x)</code> | <p>Grupo de captura: Encuentra la <code>x</code> y la recuerda. Por ejemplo, <code>/(foo)/</code> encuentra y recuerda "foo" en "foo bar".</p> <p>Una expresión regular puede tener varios grupos de captura. En los resultados, coincide con los grupos capturados normalmente en un arreglo cuyos miembros están en el mismo orden que los paréntesis de la izquierda en el grupo capturado. Este suele ser solo el orden de los propios grupos capturados. Esto se vuelve importante cuando los grupos capturados están anidados. Se accede a las coincidencias utilizando el índice de los elementos del resultado (<code>[1]</code>, <code>...</code>, <code>[n]</code>) o desde las propiedades predefinidas del objeto <code>RegExp</code> (<code>\$1</code>, <code>...</code>, <code>\$9</code>).</p> <p>Los grupos de captura tienen una penalización de rendimiento. Si no necesitas que se recupere la subcadena coincidente, prefiere los paréntesis que no capturen (ve más abajo).</p> <p>String.match() no devolverá grupos si el indicador <code>/.../g</code> está configurado. Sin embargo, aún puedes usar String.matchAll() para obtener todas los encontrados.</p> |

| Caracteres | Significado |
|-------------------------------|---|
| <code>\n</code> | <p>Donde "n" es un número entero positivo. Una referencia posterior a la última subcadena que coincide con el paréntesis n en la expresión regular (contando los paréntesis izquierdos). Por ejemplo, <code>/apple(,)\sorange\1/</code> coincide con "apple, orange" en "apple, orange, cherry, peach".</p> |
| <code>\k<Name></code> | <p>Una referencia inversa a la última subcadena encontrada con el grupo de captura Nombrado especificado por <code><Name></code>.</p> <p>Por ejemplo, <code>/(?<title>\w+), yes \k<title>/</code> concuerda con "Sir, yes Sir" en "Do you copy? Sir, yes Sir!".</p> <div> <p><code>\k</code> aquí se usa literalmente para indicar el comienzo de una referencia a un grupo de captura nombrado.</p> </div> |
| <code>(?<Name>x)</code> | <p>Grupo de captura nombrado: reconoce la "x" y la almacena en la propiedad <code>group</code> del resultado devuelto bajo el nombre especificado por <code><Name></code>. Los corchetes angulares (<code><</code> y <code>></code>) son obligatorios para el nombre del grupo.</p> <p>Por ejemplo, para extraer el código de área de Estados Unidos de un número de teléfono, podríamos usar <code>/\((?<area>\d\d\d)\)/</code>. El número resultante debería aparecer en <code>matches.groups.area</code>.</p> |
| <code>(?:x)</code> | <p>Grupo sin captura: reconoce la "x" pero no recuerda el resultado. La subcadena encontrada no se puede recuperar de los elementos del arreglo resultante (<code>[1]</code>, ..., <code>[n]</code>) o de las propiedades predefinidas del objeto <code>RegExp</code> (<code>\$1</code>, ..., <code>\$9</code>).</p> |

Cuantificadores

Nota: A continuación, *elemento* se refiere no solo a caracteres singulares, sino que también incluye clases de caracteres, escapes de propiedad Unicode, grupos y rangos.

| Caracteres | Significado |
|-------------------|---|
| <code>x*</code> | Concuerda 0 o más veces con el elemento "x" anterior. Por ejemplo, <code>/bo*/</code> reconoce a "boooo" en "Un fantasma boooood" y "b" en "A bird warbled", pero nada en "Una cabra gruñó". |
| <code>x+</code> | Encuentra 1 o más veces el elemento "x" anterior Equivalente a <code>{1,}</code> . Por ejemplo, <code>/a+/</code> encuentra la "a" en "candy" y todas las "a"es en "caaaaaaandy". |
| <code>x?</code> | Halla 0 o 1 vez el elemento "x" anterior. Por ejemplo, <code>/e?Le?/</code> reconoce a "el" en "ángel" y a "le" en "angle". Si se usa inmediatamente después de cualquiera de los cuantificadores <code>*</code> , <code>+</code> , <code>?</code> o <code>{ }</code> , hace que el cuantificador no codicioso (que reconoce el número mínimo de veces), a diferencia del predeterminado, que es codicioso (que reconoce el número máximo de veces). |
| <code>x{n}</code> | Donde "n" es un número entero positivo, concuerda exactamente con "n" apariciones del elemento "x" anterior. Por ejemplo, <code>/a{2}/</code> no reconoce la "a" en "candy", pero reconoce todas las "a"s en "caandy" y las dos primeras "a"s en "caaandy ". |

| Caracteres | Significado |
|--|---|
| $x\{n,\}$ | Donde "n" es un número entero positivo, concuerda con al menos "n" apariciones del elemento "x". Por ejemplo, <code>/a{2,}/</code> no reconoce la "a" en "candy", pero reconoce todas las "a" en "caandy" y en "caaaaaaandy". |
| $x\{n,m\}$ | Donde "n" es 0 o un número entero positivo, "m" es un número entero positivo y $m > n$, reconoce por lo menos "n" y como máximo "m" apariciones del elemento "x" anterior. Por ejemplo, <code>/a{1,3}/</code> no reconoce nada en "cndy", la "a" en "caramelo", las dos "a" en "caandy" y las tres primeras "a" está en "caaaaaaandy". Observa que al comparar "caaaaaaandy", las "aaa" encontradas, aunque la cadena original tenía más "a" s. |
| $x^*?$ $x^+?$ $x^{*?}$ $x\{n\}?$ $x\{n,\}?$ $x\{n,m\}?$ | <p>De manera predeterminada, los cuantificadores como <code>*</code> y <code>+</code> son "codiciosos", lo cual significa que intentan hacer coincidir la mayor cantidad de cadena posible. El carácter <code>?</code> después del cuantificador hace que este sea "no codicioso": lo cual significa que se detendrá tan pronto como encuentre una concordancia. Por ejemplo, dada una cadena "algo como <code><foo> <bar> new </bar> </foo></code>":</p> <ul style="list-style-type: none"> <code><.*></code> reconocerá "<code><foo> <bar> nuevo </bar> </foo></code>" <code><.*?></code> encajará "<code><foo></code>" |

Escapa la propiedad Unicode

```
// Valores no binarios
```

```
\p{UnicodePropertyValue}
```

```
\p{UnicodePropertyName=UnicodePropertyValue}
```

```
// Valores binarios y no binarios
```

`\p{UnicodeBinaryPropertyName}`

// Negación: `\P` is negado `\p`

`\P{UnicodePropertyValue}`

`\P{UnicodeBinaryPropertyName}`

UnicodeBinaryPropertyName

El nombre de una [propiedad binaria](#) . Por ejemplo: [ASCII](#) , [Alpha](#) , `Math` , [Diacrítica](#) , [Emoji](#) , [Hex_Digit](#) , `Math` , [Espacio_blanco](#) , etc. Consulta [Unicode Data PropList.txt](#) para obtener más información.

UnicodePropertyName

El nombre de una propiedad [no binaria](#) :

- [General_Category](#) (`gc`)
- [Script](#) (`sc`)
- [Script_Extensions](#) (`scx`)

Consulta también [PropertyValueAliases.txt](#)

UnicodePropertyValue

Uno de los fragmentos enumerados en la sección Valores, más adelante. Muchos valores tienen alias o abreviaturas (por ejemplo, el valor `Decimal_Number` para la propiedad `General_Category` se puede escribir cómo `Nd` , `digit` , o `Decimal_number`). Para la mayoría de los valores, la parte *UnicodePropertyName* y el signo igual se pueden omitir. Si se especifica un *UnicodePropertyName* , el valor debe corresponder al tipo de propiedad proporcionado.

Nota: Puesto que hay muchas propiedades y valores disponibles, no las describiremos exhaustivamente aquí, sino que proporcionaremos varios ejemplos.

Last modified: 11 feb 2021, [by MDN contributors](#)

