JavaScript RegExp Reference

## RegExp Object

A regular expression is a **pattern** of characters.

The pattern is used to do pattern-matching **"search-and-replace"** functions on text.

In JavaScript, a **RegExp Object** is a pattern with **Properties** and **Methods**.

## Syntax

/pattern/modifier(s);

Example explained:

|  |  |
| --- | --- |
| **w3schools** | The pattern to search for |
| **/w3schools/** | A regular expression |
| **/w3schools/i** | A case-insensitive regular expression |

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Regular Expressions</h2>

<p>Do a case-insensitive search for "w3schools" in a string:</p>

<p id="demo"></p>

<script>

let text = "Visit W3Schools";

let pattern = /w3schools/i;

let result = text.match(pattern);

document.getElementById("demo").innerHTML = result;

</script>

</body>

</html>

## Modifiers

Modifiers are used to perform case-insensitive and global searches:

|  |  |
| --- | --- |
| **Modifier** | **Description** |
| [g](https://www.w3schools.com/jsref/jsref_regexp_g.asp) | Perform a global match (find all matches rather than stopping after the first match) |
| [i](https://www.w3schools.com/jsref/jsref_regexp_i.asp) | Perform case-insensitive matching |
| [m](https://www.w3schools.com/jsref/jsref_regexp_m.asp) | Perform multiline matching |

# JavaScript RegExp m Modifier

## <!DOCTYPE html>

## <html>

## <body>

## <h2>JavaScript Regular Expressions</h2>

## <p>Do a multiline search for "is" at the beginning of each line in a string:</p>

## <p id="demo"></p>

## <script>

## let text = `Is this

## all there

## is`

## let pattern = /^is/m;

## let result = text.match(pattern);

## document.getElementById("demo").innerHTML = result;

## </script>

## </body>

## </html>

## <!DOCTYPE html>

## <html>

## <body>

## <h2>JavaScript Regular Expressions</h2>

## <p>Do a global, multiline search for "is" at the end of each line in a string.</p>

## <p id="demo"></p>

## <script>

## let text = `Is this

## all there

## is`

## let pattern = /is$/gm;

## let result = text.match(pattern);

## document.getElementById("demo").innerHTML = result;

## </script>

## </body>

## </html>

## Brackets

Brackets are used to find a range of characters:

|  |  |
| --- | --- |
| **Expression** | **Description** |
| [[abc]](https://www.w3schools.com/jsref/jsref_regexp_charset.asp) | Find any character between the brackets |
| [[^abc]](https://www.w3schools.com/jsref/jsref_regexp_charset_not.asp) | Find any character NOT between the brackets |
| [[0-9]](https://www.w3schools.com/jsref/jsref_regexp_0-9.asp) | Find any character between the brackets (any digit) |
| [[^0-9]](https://www.w3schools.com/jsref/jsref_regexp_not_0-9.asp) | Find any character NOT between the brackets (any non-digit) |
| [(x|y)](https://www.w3schools.com/jsref/jsref_regexp_xy.asp) | Find any of the alternatives specified |

## <!DOCTYPE html>

## <html>

## <body>

## <h2>JavaScript Regular Expressions</h2>

## <p>A global search for the character "h" in a string:</p>

## <p id="demo"></p>

## <script>

## let text = "Is this all there is?";

## let pattern = /[h]/g;

## let result = text.match(pattern);

## document.getElementById("demo").innerHTML = result;

## </script>

## </body>

## </html>

## <!DOCTYPE html>

## <html>

## <body>

## <h2>JavaScript Regular Expressions</h2>

## <p>A global search for the numbers 1 to 4:</p>

## <p id="demo"></p>

## <script>

## let text = "123456789";

## let pattern = /[1-4]/g;

## let result = text.match(pattern);

## document.getElementById("demo").innerHTML = result;

## </script>

## </body>

## </html>

## Definition and Usage

The [0-9] expression is used to find any character between the brackets.

The digits inside the brackets can be any numbers or span of numbers from 0 to 9.

****Tip:**** Use the [[^0-9]](https://www.w3schools.com/jsref/jsref_regexp_not_0-9.asp) expression to find any character that is NOT a digit.

## <!DOCTYPE html>

## <html>

## <body>

## <h2>JavaScript Regular Expressions</h2>

## <p>A global search for numbers that are NOT from 1 to 4:</p>

## <p id="demo"></p>

## <script>

## let text = "123456789";

## let pattern = /[^1-4]/g;

## let result = text.match(pattern);

## document.getElementById("demo").innerHTML = result;

## </script>

## </body>

## </html>

## <!DOCTYPE html>

## <html>

## <body>

## <h2>JavaScript Regular Expressions</h2>

## <p>A global search for the specified alternatives (red|green):</p>

## <p id="demo"></p>

## <script>

## let text = "re, green, red, green, gren, gr, blue, yellow";

## let pattern = /(red|green)/g;

## let result = text.match(pattern);

## document.getElementById("demo").innerHTML = result;

## </script>

## </body>

## </html>

## Metacharacters

Metacharacters are characters with a special meaning:

|  |  |
| --- | --- |
| **Metacharacter** | **Description** |
| [.](https://www.w3schools.com/jsref/jsref_regexp_dot.asp) | Find a single character, except newline or line terminator |
| [\w](https://www.w3schools.com/jsref/jsref_regexp_wordchar.asp) | Find a word character |
| [\W](https://www.w3schools.com/jsref/jsref_regexp_wordchar_non.asp) | Find a non-word character |
| [\d](https://www.w3schools.com/jsref/jsref_regexp_digit.asp) | Find a digit |
| [\D](https://www.w3schools.com/jsref/jsref_regexp_digit_non.asp) | Find a non-digit character |
| [\s](https://www.w3schools.com/jsref/jsref_regexp_whitespace.asp) | Find a whitespace character |
| [\S](https://www.w3schools.com/jsref/jsref_regexp_whitespace_non.asp) | Find a non-whitespace character |
| [\b](https://www.w3schools.com/jsref/jsref_regexp_begin.asp) | Find a match at the beginning/end of a word, beginning like this: \bHI, end like this: HI\b |
| [\B](https://www.w3schools.com/jsref/jsref_regexp_begin_not.asp) | Find a match, but not at the beginning/end of a word |
| [\0](https://www.w3schools.com/jsref/jsref_regexp_nul.asp) | Find a NULL character |
| [\n](https://www.w3schools.com/jsref/jsref_regexp_newline.asp) | Find a new line character |
| [\f](https://www.w3schools.com/jsref/jsref_regexp_formfeed.asp) | Find a form feed character |
| [\r](https://www.w3schools.com/jsref/jsref_regexp_carriagereturn.asp) | Find a carriage return character |
| [\t](https://www.w3schools.com/jsref/jsref_regexp_tab.asp) | Find a tab character |
| [\v](https://www.w3schools.com/jsref/jsref_regexp_vtab.asp) | Find a vertical tab character |
| [\xxx](https://www.w3schools.com/jsref/jsref_regexp_octal.asp) | Find the character specified by an octal number xxx |
| [\xdd](https://www.w3schools.com/jsref/jsref_regexp_hex.asp) | Find the character specified by a hexadecimal number dd |
| [\udddd](https://www.w3schools.com/jsref/jsref_regexp_unicode_hex.asp) | Find the Unicode character specified by a hexadecimal number dddd |

## <!DOCTYPE html>

## <html>

## <body>

## <h2>JavaScript Regular Expressions</h2>

## <p>A global search for word characters:</p>

## <p id="demo"></p>

## <script>

## let text = "Give 100%!";

## let pattern = /\w/g;

## let result = text.match(pattern);

## document.getElementById("demo").innerHTML = result;

## </script>

## </body>

## </html>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## <!DOCTYPE html>

## <html>

## <body>

## <h2>JavaScript Regular Expressions</h2>

## <p>A global search for non-word characters:</p>

## <p id="demo"></p>

## <script>

## let text = "Give 100%!";

## let pattern = /\W/g;

## let result = text.match(pattern);

## document.getElementById("demo").innerHTML = result;

## </script>

## </body>

## </html>

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Regular Expressions</h2>

<p>A global search for digits:</p>

<p id="demo"></p>

<script>

let text = "Give 100%!";

let pattern = /\d/g;

let result = text.match(pattern);

document.getElementById("demo").innerHTML = result;

</script>

</body>

</html>

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Regular Expressions</h2>

<p>A global search for non-digit characters:</p>

<p id="demo"></p>

<script>

let text = "Give 100%!";

let pattern = /\D/g;

let result = text.match(pattern);

document.getElementById("demo").innerHTML=result;

</script>

</body>

</html>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Quantifiers

|  |  |
| --- | --- |
| **Quantifier** | **Description** |
| [n+](https://www.w3schools.com/jsref/jsref_regexp_onemore.asp) | Matches any string that contains at least one n |
| [n\*](https://www.w3schools.com/jsref/jsref_regexp_zeromore.asp) | Matches any string that contains zero or more occurrences of n |
| [n?](https://www.w3schools.com/jsref/jsref_regexp_zeroone.asp) | Matches any string that contains zero or one occurrences of n |
| [n{X}](https://www.w3schools.com/jsref/jsref_regexp_nx.asp) | Matches any string that contains a sequence of *X* *n*'s |
| [n{X,Y}](https://www.w3schools.com/jsref/jsref_regexp_nxy.asp) | Matches any string that contains a sequence of X to Y *n*'s |
| [n{X,}](https://www.w3schools.com/jsref/jsref_regexp_nxcomma.asp) | Matches any string that contains a sequence of at least X *n*'s |
| [n$](https://www.w3schools.com/jsref/jsref_regexp_ndollar.asp) | Matches any string with n at the end of it |
| [^n](https://www.w3schools.com/jsref/jsref_regexp_ncaret.asp) | Matches any string with n at the beginning of it |
| [?=n](https://www.w3schools.com/jsref/jsref_regexp_nfollow.asp) | Matches any string that is followed by a specific string n |
| [?!n](https://www.w3schools.com/jsref/jsref_regexp_nfollow_not.asp) | Matches any string that is not followed by a specific string n |

## <!DOCTYPE html>

## <html>

## <body>

## <h2>JavaScript Regular Expressions</h2>

## <p>A global search for at least one "o" in a string:</p>

## <p id="demo"></p>

## <script>

## let text = "Hellooo World! Hello W3Schools!";

## let pattern = /o+/g;

## let result = text.match(pattern);

## document.getElementById("demo").innerHTML = result;

## </script>

## </body>

## </html>

## <!DOCTYPE html>

## <html>

## <body>

## <h2>JavaScript Regular Expressions</h2>

## <p>A global search for an "l", followed by zero or more "o" characters:</p>

## <p id="demo"></p>

## <script>

## let text = "Hellooo World! Hello W3Schools!";

## let pattern = /lo\*/g;

## let result = text.match(pattern);

## document.getElementById("demo").innerHTML = result;

## </script>

## </body>

## </html>

## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## RegExp Object Properties

|  |  |
| --- | --- |
| **Property** | **Description** |
| [constructor](https://www.w3schools.com/jsref/jsref_regexp_constructor.asp) | Returns the function that created the RegExp object's prototype |
| [global](https://www.w3schools.com/jsref/jsref_regexp_global.asp) | Checks whether the "g" modifier is set |
| [ignoreCase](https://www.w3schools.com/jsref/jsref_regexp_ignorecase.asp) | Checks whether the "i" modifier is set |
| [lastIndex](https://www.w3schools.com/jsref/jsref_regexp_lastindex.asp) | Specifies the index at which to start the next match |
| [multiline](https://www.w3schools.com/jsref/jsref_regexp_multiline.asp) | Checks whether the "m" modifier is set |
| [source](https://www.w3schools.com/jsref/jsref_regexp_source.asp) | Returns the text of the RegExp pattern |

## RegExp Object Methods

|  |  |
| --- | --- |
| **Method** | **Description** |
| [compile()](https://www.w3schools.com/jsref/jsref_regexp_compile.asp) | Deprecated in version 1.5. Compiles a regular expression |
| [exec()](https://www.w3schools.com/jsref/jsref_regexp_exec.asp) | Tests for a match in a string. Returns the first match |
| [test()](https://www.w3schools.com/jsref/jsref_regexp_test.asp) | Tests for a match in a string. Returns true or false |
| [toString()](https://www.w3schools.com/jsref/jsref_regexp_tostring.asp) | Returns the string value of the regular expression |

# JavaScript RegExp exec()

The exec() method tests for a match in a string.

If it finds a match, it returns a result array, otherwise it returns null.

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript RegExp</h2>

<p>The exec() method tests for a match in a string:</p>

<p>Search a string for the character "e":</p>

<p id="demo"></p>

<script>

let text = "The best things in life are free";

let result = /e/.exec(text);

document.getElementById("demo").innerHTML = result;

</script>

</body>

</html>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# JavaScript RegExp test()

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Regular Expressions</h2>

<p>The test() method returns true if it finds a match, otherwise false.</p>

<p>Search a string for the character "e":</p>

<p id="demo"></p>

<script>

let text = "The best things in life are free";

let pattern = /e/;

let result = pattern.test(text);

document.getElementById("demo").innerHTML = result;

</script>

</body>

</html>

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Regular Expressions</h2>

<p>toString() returns the string value of a RegExp object:</p>

<p id="demo"></p>

<script>

let pattern = new RegExp("Hello World", "g");

let text = pattern.toString();

document.getElementById("demo").innerHTML = text;

</script>

</body>

</html>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Regular Expressions</h2>

<p>Do a global search for "is" in a string:</p>

<p id="demo"></p>

<script>

let text = "Is this all there is?";

let pattern = /is/g;

let result = text.match(pattern);

document.getElementById("demo").innerHTML = result;

</script>

</body>

</html>

### Example

A global search for the character "h" in a string:

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Regular Expressions</h2>

<p>A global search for the character "h" in a string:</p>

<p id="demo"></p>

<script>

let text = "Is this all there is?";

let pattern = /[h]/g;

let result = text.match(pattern);

document.getElementById("demo").innerHTML = result;

</script>

</body>

</html>

# JavaScript RegExp Group [^abc]

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Regular Expressions</h2>

<p>Do a global search for that characters that are not "h":</p>

<p id="demo"></p>

<script>

let text = "Is this all there is?";

let pattern = /[^h]/g;

let result = text.match(pattern);

document.getElementById("demo").innerHTML = result;

</script>

</body>

</html>