JavaScript Regular Expressions

A regular expression is a sequence of characters that forms a search pattern.

The search pattern can be used for text search and text replace operations.

What Is a Regular Expression?

A regular expression is a sequence of characters that forms a **search pattern**.

When you search for data in a text, you can use this search pattern to describe what you are searching for.

A regular expression can be a single character, or a more complicated pattern.

Regular expressions can be used to perform all types of **text search** and **text replace** operations.

Syntax

```
/pattern/modifiers;
```

Example

```
var patt = /w3schools/i;
```

Example explained:

/w3schools/i is a regular expression.

w3schools is a pattern (to be used in a search).

i is a modifier (modifies the search to be case-insensitive).

Using String Methods

In JavaScript, regular expressions are often used with the two **string methods**: `search()` and `replace()`.

The `search()` method uses an expression to search for a match, and returns the position of the match.

The `replace()` method returns a modified string where the pattern is replaced.

Using String search() With a String

The `search()` method searches a string for a specified value and returns the position of the match:

Example

Use a string to do a search for "W3schools" in a string:

```
var str = "Visit W3Schools!";
var n = str.search("W3Schools");
```

Try it Yourself »

Using String search() With a Regular Expression

Example

Use a regular expression to do a case-insensitive search for "w3schools" in a string:

```
var str = "Visit W3Schools";
var n = str.search(/w3schools/i);
```

The result in *n* will be:

```
6
```

Try it Yourself »

Using String replace() With a String

The `replace()` method replaces a specified value with another value in a string:

```
var str = "Visit Microsoft!";
var res = str.replace("Microsoft", "W3Schools");
```

Try it Yourself »

Use String replace() With a Regular Expression

Example

Use a case insensitive regular expression to replace Microsoft with W3Schools in a string:

```
var str = "Visit Microsoft!";
var res = str.replace(/microsoft/i, "W3Schools");
```

The result in *res* will be:

```
Visit W3Schools!
```

Try it Yourself »

Did You Notice?

Regular expression arguments (instead of string arguments) can be used in the methods above.
Regular expressions can make your search much more powerful (case insensitive for example).

Regular Expression Modifiers

Modifiers can be used to perform case-insensitive more global searches:

Modifier	Description	Try it
i	Perform case-insensitive matching	Try it »
g	Perform a global match (find all matches rather than stopping after the first match)	Try it »
m	Perform multiline matching	Try it »

Regular Expression Patterns

Brackets are used to find a range of characters:

Expression	Description	Try it
[abc]	Find any of the characters between the brackets	Try it »
[0-9]	Find any of the digits between the brackets	Try it »
(x|y)	Find any of the alternatives separated with |	Try it »

Metacharacters are characters with a special meaning:

Metacharacter	Description	Try it
\d	Find a digit	Try it »
\s	Find a whitespace character	Try it »
\b	Find a match at the beginning of a word like this: \bWORD, or at the end of a word like this: WORD\b	Try it » Try it »
\uxxxx	Find the Unicode character specified by the hexadecimal number xxxx	Try it »

Quantifiers define quantities:

Quantifier	Description	Try it

n+	Matches any string that contains at least one *n*	Try it »
n*	Matches any string that contains zero or more occurrences of *n*	Try it »
n?	Matches any string that contains zero or one occurrences of *n*	Try it »

Using the RegExp Object

In JavaScript, the RegExp object is a regular expression object with predefined properties and methods.

Using test()

The `test()` method is a RegExp expression method.

It searches a string for a pattern, and returns true or false, depending on the result.

The following example searches a string for the character "e":

Example

```
var patt = /e/;
patt.test("The best things in life are free!");
```

Since there is an "e" in the string, the output of the code above will be:

```
true
```

Try it Yourself »

You don't have to put the regular expression in a variable first. The two lines above can be shortened to one:

```
/e/.test("The best things in life are free!");
```

Using exec()

The `exec()` method is a RegExp expression method.

It searches a string for a specified pattern, and returns the found text as an object.

If no match is found, it returns an empty *(null)* object.

The following example searches a string for the character "e":

Example 1

```
/e/.exec("The best things in life are free!");
```

Try it Yourself »

Complete RegExp Reference

For a complete reference, go to our Complete JavaScript RegExp Reference.

The reference contains descriptions and examples of all RegExp properties and methods.

JavaScript RegExp Reference

RegExp Object

A regular expression is an object that describes a pattern of characters.
Regular expressions are used to perform pattern-matching and "search-and-replace" functions on text.

Syntax

/pattern/modifiers;

Example

```
var patt = /w3schools/i
```

Try it Yourself »

Example explained:

- **/w3schools/i** is a regular expression.
- **w3schools** is a pattern (to be used in a search).
- **i** is a modifier (modifies the search to be case-insensitive).

For a tutorial about Regular Expressions, read our JavaScript RegExp Tutorial.

Modifiers

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

Modifier	Description
g	Perform a global match (find all matches rather than stopping after the first match)
i	Perform case-insensitive matching
m	Perform multiline matching

Brackets

Brackets are used to find a range of characters:

Expression	Description
[abc]	Find any character between the brackets
[^abc]	Find any character NOT between the brackets
[0-9]	Find any character between the brackets (any digit)
[^0-9]	Find any character NOT between the brackets (any non-digit)
(x|y)	Find any of the alternatives specified

Metacharacters

Metacharacters are characters with a special meaning:

Metacharacter	Description

.	Find a single character, except newline or line terminator
\w	Find a word character
\W	Find a non-word character
\d	Find a digit
\D	Find a non-digit character
\s	Find a whitespace character
\S	Find a non-whitespace character
\b	Find a match at the beginning/end of a word, beginning like this: \bHI, end like this: HI\b
\B	Find a match, but not at the beginning/end of a word
\0	Find a NULL character
\n	Find a new line character
\f	Find a form feed character
\r	Find a carriage return character
\t	Find a tab character
\v	Find a vertical tab character
\xxx	Find the character specified by an octal number xxx
\xdd	Find the character specified by a hexadecimal number dd

Quantifiers

Quantifier	Description
n+	Matches any string that contains at least one *n*
n*	Matches any string that contains zero or more occurrences of *n*
n?	Matches any string that contains zero or one occurrences of *n*
n{X}	Matches any string that contains a sequence of *X* *n*'s
n{X,Y}	Matches any string that contains a sequence of X to Y *n*'s
n{X,}	Matches any string that contains a sequence of at least X *n*'s
n$	Matches any string with *n* at the end of it
^n	Matches any string with *n* at the beginning of it
?=n	Matches any string that is followed by a specific string *n*

?!n	Matches any string that is not followed by a specific string *n*

RegExp Object Properties

Property	Description
constructor	Returns the function that created the RegExp object's prototype
global	Checks whether the "g" modifier is set
ignoreCase	Checks whether the "i" modifier is set
lastIndex	Specifies the index at which to start the next match
multiline	Checks whether the "m" modifier is set
source	Returns the text of the RegExp pattern

RegExp Object Methods

Method	Description
compile()	Deprecated in version 1.5. Compiles a regular expression
exec()	Tests for a match in a string. Returns the first match
test()	Tests for a match in a string. Returns true or false
toString()	Returns the string value of the regular expression

JavaScript RegExp constructor Property

❮ JavaScript RegExp ObjectNext ❯

Example

The constructor property returns a regular expression's constructor function:

```
var patt = new RegExp("Hello World", "g");
var res = patt.constructor;
```

Try it Yourself »

Definition and Usage

In JavaScript, the constructor property returns the constructor function for an object.
The return value is a reference to the function, not the name of the function:
For JavaScript **regular expressions** the constructor property returns **function RegExp() { [native code] }**
For JavaScript **numbers** the constructor property returns **function Number() { [native code] }**
For JavaScript **strings** the constructor property returns **function String() { [native code] }**

Browser Support

Property	Chrome	Edge	Firefox	Safari	Opera
constructor	Yes	Yes	Yes	Yes	Yes

Syntax

RegExpObject.constructor

Technical Details

Return Value:	function RegExp() { [native code] }
JavaScript Version:	ECMAScript 1

JavaScript compile() Method

Example

Do a global search for "man" in a string, and replace it with "person". Then change the regular expression and replace either "man" or "woman" with "person", with the compile() method:

```
var str = "Every man in the world! Every woman on earth!";
var patt = /man/g;
var str2 = str.replace(patt,"person");
document.write(str2 + "<br>");

patt = /(wo)?man/g;
patt.compile(patt);
str2 = str.replace(patt, "person");
document.write(str2);
```

Try it Yourself »

Definition and Usage

The compile() method was deprecated in JavaScript version 1.5.

The compile() method is used to compile a regular expression during execution of a script.

The compile() method can also be used to change and recompile a regular expression.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
[abc]	Yes	Yes	Yes	Yes	Not supported

Syntax

RegExpObject.compile(*regexp*, *modifier*)

Parameter Values

Parameter	Description
regexp	A regular expression
modifier	Specifies the type of matching. "g" for a global match, "i" for a case-insensitive match and "gi" for a global, case-insensitive match

JavaScript exec() Method

Example

Search a string for the character "e":

```
var str = "The best things in life are free";
var patt = new RegExp("e");
var res = patt.exec(str);
```

Try it Yourself »

Definition and Usage

The exec() method tests for a match in a string.
This method returns the matched text if it finds a match, otherwise it returns null.

Browser Support

Method	Chrome	Edge	Firefox	Safari	Opera
exec()	Yes	Yes	Yes	Yes	Yes

Syntax

RegExpObject.exec(*string*)

Parameter Values

Parameter	Description
string	Required. The string to be searched

Return Value

Type	Description
Array	An array containing the matched text if it finds a match, otherwise it returns null

Technical Details

JavaScript Version:	ECMAScript 1

More Examples

Example

Do a global search, and test for "Hello" and "W3Schools" in a string:

```
// The string:
var str = "Hello world!";

// Look for "Hello"
var patt = /Hello/g;
var result = patt.exec(str);

// Look for "W3Schools"
var patt2 = /W3Schools/g;
result2 = patt2.exec(str);
```

The output of the code above will be:

```
Hello // match for "Hello"
null // no match for "W3Schools"
```

Try it Yourself »

JavaScript RegExp g Modifier

Example

Do a global search for "is":

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

Try it Yourself »

Definition and Usage

The g modifier is used to perform a global match (find all matches rather than stopping after the first match).

Tip: To perform a global, case-insensitive search, use this modifier together with the "i" modifier.

Tip: Use the global property to specify whether or not the g modifier is set.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
g	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("regexp", "g")

or simply:

/regexp/g
```

More Examples

Example

Do a global, case-insensitive search for "is":

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

Try it Yourself »

JavaScript global Property

Example

Check whether or not the "g" modifier is set:

```
var str = "Visit W3Schools!";
var patt1 = /W3S/g;
var res = patt1.global;
```

Try it Yourself »

Definition and Usage

The global property specifies whether or not the "g" modifier is set.
This property returns true if the "g" modifier is set, otherwise it returns false.

Browser Support

Property	Chrome	Edge	Firefox	Safari	Opera
global	Yes	Yes	Yes	Yes	Yes

Syntax

RegExpObject.global

Return Value

Type	Description
Boolean	Returns true if the "g" modifier is set, false otherwise

Technical Details

JavaScript Version:	ECMAScript 1

JavaScript RegExp i Modifier

Example

Do a case-insensitive search for "w3schools" in a string:

```
var str = "Visit W3Schools";
var patt1 = /w3schools/i;
```

Try it Yourself »

Definition and Usage

The i modifier is used to perform case-insensitive matching.

Tip: Use the ignoreCase property to check whether or not the "i" modifier is set.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
i	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("regexp", "i")

or simply:

/regexp/i
```

JavaScript ignoreCase Property

Example

Check whether or not the "i" modifier is set:

```
var str = "Visit W3Schools!";
var patt1 = /W3S/i;
var res = patt1.ignoreCase;
```

Try it Yourself »

Definition and Usage

The ignoreCase property specifies whether or not the "i" modifier is set.
This property returns true if the "i" modifier is set, otherwise it returns false.

Browser Support

Property	Chrome	Edge	Firefox	Safari	Opera
ignoreCase	Yes	Yes	Yes	Yes	Yes

Syntax

RegExpObject.ignoreCase

Return Value

Type	Description
Boolean	Returns true if the "i" modifier is set, false otherwise

JavaScript lastIndex Property

Example

Do a global search for "ain" in a string, and output the index after a match is found:

```
var str = "The rain in Spain stays mainly in the plain";
var patt1 = /ain/g;

while (patt1.test(str) == true) {
  document.write("'ain' found. Index now at: "+patt1.lastIndex);
  document.write("<br>");
}
```

Try it Yourself »

Definition and Usage

The lastIndex property specifies the index at which to start the next match.

Note: This property only works if the "g" modifier is set.

This property returns an integer that specifies the character position immediately after the last match found by exec( ) or test( ) methods.

Note: exec( ) and test( ) reset lastIndex to 0 if they do not get a match.

Browser Support

Property	Chrome	Edge	Firefox	Safari	Opera
lastIndex	Yes	Yes	Yes	Yes	Yes

Syntax

RegExpObject.lastIndex

Return Value

Type	Description
Number	An integer that specifies the character position immediately after the last match found by exec( ) or test( ) methods

JavaScript RegExp m Modifier

Example

Do a multiline search for "is" at the beginning of each line in a string:

```
var str = "\nIs th\nis it?";
var patt1 = /^is/m;
```

Try it Yourself »

Definition and Usage

The m modifier is used to perform a multiline match.

The m modifier treat beginning (^) and end ($) characters to match the beginning or end of **each line** of a string (delimited by \n or \r), rather than just the beginning or end of the string.

Note: The m modifier is case-sensitive and will stop the search after the first match. To perform a global, case-insensitive, multiline search, use this modifier together with "g" and "i".

Tip: Use the multiline property to specify whether or not the m modifier is set.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
m	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("regexp", "m")

or simply:

/regexp/m
```

More Examples

Example

Do a global, multiline search for "is" at the beginning of each line in a string:

```
var str = "\nIs th\nis h\nis?";
var patt1 = /^is/gm;
```

Try it Yourself »

Example

Do a global, case-insensitive, multiline search for "is" at the beginning of each line in a string:

```
var str = "\nIs th\nis h\nis?";
var patt1 = /^is/gmi;
```

Try it Yourself »

Example

Do a global, multiline search for "is" at the end of each line in a string:

```
var str = "Is\nthis\nhis\n?";
var patt1 = /is$/gm;
```

Try it Yourself »

JavaScript multiline Property

Example

Check whether or not the "m" modifier is set:

```
var str = "Visit W3Schools!";
var patt1 = /W3S/gi; // "g" and "i" is set, "m" is not.
var res = patt1.multiline;
```

Try it Yourself »

Definition and Usage

The multiline property specifies whether or not the m modifier is set.
This property returns true if the "m" modifier is set, otherwise it returns false.

Browser Support

Property	Chrome	Edge	Firefox	Safari	Opera
multiline	Yes	Yes	Yes	Yes	Yes

Syntax

RegExpObject.multiline

Return Value

Type	Description
Boolean	Returns true if the "m" modifier is set, false otherwise

JavaScript RegExp + Quantifier

Example 1

Do a global search for at least one "o":

```
var str = "Hellooo World! Hello W3Schools!";
var patt1 = /o+/g;
```

Try it Yourself »

Definition and Usage

The *n*+ quantifier matches any string that contains at least one *n*.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
+	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("n+")

or simply:

/n+/
```

Syntax with modifiers

```
new RegExp("n+", "g")

or simply:

/\n+/g
```

More Examples

Example 2

Do a global search for at least one word character:

```
var str = "Hellooo World! Hello W3Schools!";
var patt1 = /\w+/g;
```

Try it Yourself »

JavaScript RegExp * Quantifier

Example 1

Do a global search for an "l", followed by zero or more "o" characters:

```
var str = "Hellooo World! Hello W3Schools!";
var patt1 = /lo*/g;
```

Try it Yourself »

Definition and Usage

The *n** quantifier matches any string that contains zero or more occurrences of *n*.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
*	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("n*")

or simply:

/n*/
```

Syntax with modifiers

```
new RegExp("n*", "g")

or simply:

/\n*/g
```

More Examples

Example 2

Do a global search for a "1", followed by zero or more "0" characters:

```
var str = "1, 100 or 1000?";
var patt1 = /10*/g;
```

Try it Yourself »

JavaScript RegExp ? Quantifier

Example

Do a global search for a "1", followed by zero or one "0" characters:

```
var str = "1, 100 or 1000?";
var patt1 = /10?/g;
```

Try it Yourself »

Definition and Usage

The *n*? quantifier matches any string that contains zero or one occurrences of *n*.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
?	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("n?")

or simply:

/n?/
```

Syntax with modifiers

```
new RegExp("n?", "g")

or simply:

/\n?/g
```

JavaScript RegExp {X} Quantifier

Example

Do a global search for a substring that contains a sequence of four digits:

```
var str = "100, 1000 or 10000?";
var patt1 = /\d{4}/g;
```

Try it Yourself »

Definition and Usage

The *n*{*X*} quantifier matches any string that contains a sequence of *X* *n*'s.
X must be a number.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
{X}	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("n{X}")

or simply:

/n{X}/
```

Syntax with modifiers

```
new RegExp("n{X}", "g")

or simply:

/\n{X}/g
```

JavaScript RegExp {X,Y} Quantifier

Example

Do a global search for a substring that contains a sequence of three to four digits:

```
var str = "100, 1000 or 10000?";
var patt1 = /\d{3,4}/g;
```

Try it Yourself »

Definition and Usage

The *n*{*X*,*Y*} quantifier matches any string that contains a sequence of X to Y *n*'s.
X and *Y* must be a number.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
{X,Y}	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("n{X,Y}")

or simply:

/n{X,Y}/
```

Syntax with modifiers

```
new RegExp("n{X,Y}", "g")

or simply:

/\n{X,Y}/g
```

JavaScript RegExp {X,} Quantifier

Example

Do a global search for a substring that contains a sequence of at least three digits:

```
var str = "100, 1000 or 10000?";
var patt1 = /\d{3,}/g;
```

Try it Yourself »

Definition and Usage

The *n*{*X*,} quantifier matches any string that contains a sequence of at least X *n*'s.
X must be a number.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
{X,}	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("n{X,}")

or simply:

/n{X,}/
```

Syntax with modifiers

```
new RegExp("n{X,}", "g")

or simply:

/\n{X,}/g
```

JavaScript RegExp $ Quantifier

Example

Do a global search for "is" at the end of a string:

```
var str = "Is this his";
var patt1 = /is$/g;
```

Try it Yourself »

Definition and Usage

The *n*$ quantifier matches any string with *n* at the end of it.

Tip: Use the ^*n* quantifier to match any string with *n* at the BEGINNING of it.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
$	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("n$")
```

or

```
/n$/
```

Syntax with modifiers

```
new RegExp("n$", "g")
```

or simply:

```
/\n$/g
```

More Examples

Example

Do a global, multiline search for "is" at the end of each line in a string:

```
var str = "Is\nthis\nhis\n?";
var patt1 = /is$/gm;
```

Try it Yourself »

JavaScript RegExp ^ Quantifier

Example

Do a global search for "Is" at the beginning of a string:

```
var str = "Is this his";
var patt1 = /^Is/g;
```

Try it Yourself »

Definition and Usage

The ^*n* quantifier matches any string with *n* at the beginning of it.
Tip: Use the *n*$ quantifier to match any string with *n* at the END of it.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
^	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("^n")
```

or

```
/^n/
```

Syntax with modifiers

```
new RegExp("^n", "g")
```

or simply:

```
/\^n/g
```

More Examples

Example

Do a global, case-insensitive, multiline search for "is" at the beginning of each line in a string:

```
var str = "\nIs th\nis h\nis?";
var patt1 = /^is/gmi;
```

Try it Yourself »

JavaScript RegExp ?= Quantifier

Example

Do a global search for "is" followed by " all":

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

Try it Yourself »

Definition and Usage

The ?=*n* quantifier matches any string that is followed by a specific string *n*.

Tip: Use the ?!*n* quantifier to match any string that is NOT followed by a specific string *n*.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
?=	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("regexp(?=n)")
```

or

```
/regexp(?=n)/
```

Syntax with modifiers

```
new RegExp("regexp(?=n)", "g")
```

or simply:

```
/\regexp(?=n)/g
```

JavaScript RegExp ?! Quantifier

Example

Do a global, case insensitive search for "is" not followed by " all":

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

Try it Yourself »

Definition and Usage

The ?!*n* quantifier matches any string that is not followed by a specific string *n*.
Tip: Use the ?=*n* quantifier to match any string that IS followed by a specific string *n*.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
?!	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("regexp(?!n)")
```

or

```
/regexp(?!n)/
```

Syntax with modifiers

```
new RegExp("regexp(?!n)", "g")
```

or simply:

```
/\regexp(?!n)/g
```

JavaScript source Property

Example

Return the text of the RegExp pattern:

```
var str = "Visit W3Schools";
var patt1 = /W3S/g;
var res = "The text of the RegExp is: " + patt1.source;
```

Try it Yourself »

Definition and Usage

The source property returns the text of the RegExp pattern.

Browser Support

Property	Chrome	Edge	Firefox	Safari	Opera
source	Yes	Yes	Yes	Yes	Yes

Syntax

RegExpObject.source

Return Value

Type	Description
String	The text of the RegExp pattern

JavaScript test() Method

Example

Search a string for the character "e":

```
var str = "The best things in life are free";
var patt = new RegExp("e");
var res = patt.test(str);
```

Try it Yourself »

Definition and Usage

The test() method tests for a match in a string.
This method returns true if it finds a match, otherwise it returns false.

Browser Support

Method	Chrome	Edge	Firefox	Safari	Opera
test()	Yes	Yes	Yes	Yes	Yes

Syntax

RegExpObject.test(*string*)

Parameter Values

Parameter	Description
string	Required. The string to be searched

Return Value

Type	Description
Boolean	Returns true if it finds a match, otherwise it returns false

Technical Details

JavaScript Version:	ECMAScript 1

More Examples

Example

Do a global search, and test for "Hello" and "W3Schools" in a string:

```
// The string:
var str = "Hello world!";

// Look for "Hello"
var patt = /Hello/g;
var result = patt.test(str);

// Look for "W3Schools"
patt2 = /W3Schools/g;
result2 = patt2.test(str);
```

Try it Yourself »

JavaScript RegExp toString Method

Example

Return the string value of the regular expression:

```
var patt = new RegExp("Hello World", "g");
var res = patt.toString();
```

Try it Yourself »

Definition and Usage

The toString() method returns the string value of the regular expression.

Browser Support

Method	Chrome	Edge	Firefox	Safari	Opera
toString()	Yes	Yes	Yes	Yes	Yes

Syntax

RegExpObject.toString()

Parameters

None.

Return Value

Type	Description
String	The string value of the regular expression

JavaScript RegExp [abc] Expression

Example

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

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

Try it Yourself »

Definition and Usage

The [abc] expression is used to find any character between the brackets.
The characters inside the brackets can be any characters or span of characters:

- [abcde..] - Any character between the brackets
- [A-Z] - Any character from uppercase A to uppercase Z
- [a-z] - Any character from lowercase a to lowercase z
- [A-z]- Any character from uppercase A to lowercase z

Tip: Use the [^abc] expression to find any character NOT between the brackets.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
[abc]	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("[abc]")

or simply:

/[abc]/
```

Syntax with modifiers

```
new RegExp("[abc]", "g")

or simply:

/[abc]/g
```

More Examples

Example

Do a global search for the characters "i" and "s" in a string:

```
var str = "Do you know if this is all there is?";
var patt1 = /[is]/gi;
```

Try it Yourself »

Example

Do a global search for the character-span from lowercase "a" to lowercase "h" in a string:

```
var str = "Is this all there is?";
var patt1 = /[a-h]/g;
```

Try it Yourself »

Example

Do a global search for the character-span from uppercase "A" to uppercase "E":

```
var str = "I SCREAM FOR ICE CREAM!";
var patt1 = /[A-E]/g;
```

Try it Yourself »

Example

Do a global search for the character-span from uppercase "A" to lowercase "e" (will search for all uppercase letters, but only lowercase letters from a to e.)

```
var str = "I Scream For Ice Cream, is that OK?!";
var patt1 = /[A-e]/g;
```

Try it Yourself »

Example

Do a global, case-insensitive search for the character-span [a-s]:

```
var str = "I Scream For Ice Cream, is that OK?!";
var patt1 = /[a-s]/gi;
```

Try it Yourself »

Example

A demonstration of "g" and "gi"-search for characters:

```
var str = "THIS This this";
var patt1 = /[THIS]/g;

var str = "THIS This this";
var patt1 = /[THIS]/gi;
```

Try it Yourself »

JavaScript RegExp [^abc] Expression

Example

Do a global search for characters that are NOT inside the brackets [h]:

```
var str = "Is this all there is?";
var patt1 = /[^h]/g;
```

Try it Yourself »

Definition and Usage

The [^abc] expression is used to find any character NOT between the brackets.
The characters inside the brackets can be any characters or span of characters:

- [abcde..] - Any character between the brackets
- [A-Z] - Any character from uppercase A to uppercase Z
- [a-z] - Any character from lowercase a to lowercase z
- [A-z]- Any character from uppercase A to lowercase z

Tip: Use the [abc] expression to find any character between the brackets.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
[^abc]	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("[^xyz]")

or simply:

/[^xyz]/
```

Syntax with modifiers

```
new RegExp("[^xyz]", "g")

or simply:

/\[^xyz]/g
```

More Examples

Example

Do a global search for characters that are NOT "i" and "s" in a string:

```
var str = "Do you know if this is all there is?";
var patt1 = /[^is]/gi;
```

Try it Yourself »

Example

Do a global search for the character-span NOT from lowercase "a" to lowercase "h" in a string:

```
var str = "Is this all there is?";
var patt1 = /[^a-h]/g;
```

Try it Yourself »

Example

Do a global search for the character-span NOT from uppercase "A" to uppercase "E":

```
var str = "I SCREAM FOR ICE CREAM!";
var patt1 = /[^A-E]/g;
```

Try it Yourself »

Example

Do a global search for the character-span NOT from uppercase "A" to lowercase "e":

```
var str = "I Scream For Ice Cream, is that OK?!";
var patt1 = /[^A-e]/g;
```

Try it Yourself »

Example

Do a global, case-insensitive search for the character-span that's NOT [a-s]:

```
var str = "I Scream For Ice Cream, is that OK?!";
var patt1 = /[^a-s]/gi;
```

Try it Yourself »

JavaScript RegExp [0-9] Expression

Example

Do a global search for the numbers 1, 2, 3 and 4 in a string:

```
var str = "123456789";
var patt1 = /[1-4]/g;
```

Try it Yourself »

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] expression to find any character that is NOT a digit.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
[0-9]	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("[0-9]")

or simply:

/[0-9]/
```

Syntax with modifiers

```
new RegExp("[0-9]", "g")

or simply:

/\[0-9]/g
```

More Examples

Example

Do a global search for the number "1" in a string:

```
var str = "12121212";
var patt1 = /[1]/g;
```

Try it Yourself »

JavaScript RegExp [^0-9] Expression

Example

Do a global search for the numbers that are NOT 1 to 4 in a string:

```
var str = "123456789";
var patt1 = /[^1-4]/g;
```
Try it Yourself »

Definition and Usage

The [^0-9] expression is used to find any character that is NOT a digit.
The digits inside the brackets can be any numbers or span of numbers from 0 to 9.
Tip: Use the [0-9] expression to find any character between the brackets that is a digit.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
[^0-9]	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("[^0-9]")

or simply:

/[^0-9]/
```

Syntax with modifiers

```
new RegExp("[^0-9]", "g")

or simply:

/\[^0-9]/g
```

More Examples

Example

Do a global search for numbers that are NOT "1" in a string:

```
var str = "12121212";
var patt1 = /[^1]/g;
```
Try it Yourself »

Example

Do a global search for numbers that are NOT 5 to 8 in a string:

```
var str = "123456789";
var patt1 = /[^5-8]/g;
```
Try it Yourself »

JavaScript RegExp (x|y) Expression

Example

Do a global search to find any of the specified alternatives (red|green):

```
var str = "re, green, red, green, gren, gr, blue, yellow";
var patt1 = /(red|green)/g;
```

Try it Yourself »

Definition and Usage

The (x|y) expression is used to find any of the alternatives specified.
The alternatives can be of any characters.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
(x|y)	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("(x|y)")

or simply:

/(x|y)/
```

Syntax with modifiers

```
new RegExp("(x|y)", "g")

or simply:

/(x|y)/g
```

More Examples

Example

Do a global search to find any of the specified alternatives (0|5|7):

```
var str = "01234567890123456789";
var patt1 = /(0|5|7)/g;
```

Try it Yourself »

JavaScript RegExp . Metacharacter

Example

Do a global search for "h.t" in a string:

```
var str = "That's hot!";
var patt1 = /h.t/g;
```

Try it Yourself »

Definition and Usage

The . metacharacter is used to find a single character, except newline or other line terminators.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
.	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("regexp.")

or simply:

/regexp./
```

Syntax with modifiers

```
new RegExp("regexp.", "g")

or simply:

/regexp./g
```

JavaScript RegExp \w Metacharacter

Example

Do a global search for word characters in a string:

```
var str = "Give 100%!";
var patt1 = /\w/g;
```

Try it Yourself »

Definition and Usage

The \w metacharacter is used to find a word character.
A word character is a character from a-z, A-Z, 0-9, including the _ (underscore) character.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
\w	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("\\w")

or simply:

/\w/
```

Syntax with modifiers

```
new RegExp("\\w", "g")

or simply:

/\w/g
```

JavaScript RegExp \W Metacharacter

Example

Do a global search for non-word characters in a string:

```
var str = "Give 100%!";
var patt1 = /\W/g;
```

Try it Yourself »

Definition and Usage

The \W metacharacter is used to find a non-word character.
A word character is a character from a-z, A-Z, 0-9, including the _ (underscore) character.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
\W	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("\\W")

or simply:

/\W/
```

Syntax with modifiers

```
new RegExp("\\W", "g")

or simply:

/\W/g
```

JavaScript RegExp \d Metacharacter

Example

Do a global search for digits:

```
var str = "Give 100%!";
var patt1 = /\d/g;
```

Try it Yourself »

Definition and Usage

The \d metacharacter is used to find a digit from 0-9.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
\d	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("\\d")

or simply:

/\d/
```

Syntax with modifiers

```
new RegExp("\\d", "g")

or simply:

/\d/g
```

JavaScript RegExp \D Metacharacter

Example

Do a global search for non-digit characters:

```
var str = "Give 100%!";
var patt1 = /\D/g;
```

Try it Yourself »

Definition and Usage

The \D metacharacter is used to find a non-digit character.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
\D	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("\\D")

or simply:

/\D/
```

Syntax with modifiers

```
new RegExp("\\D", "g")

or simply:

/\D/g
```

JavaScript RegExp \s Metacharacter

Example

Do a global search for whitespace characters in a string:

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

Try it Yourself »

Definition and Usage

The \s metacharacter is used to find a whitespace character.

A whitespace character can be:

- A space character
- A tab character
- A carriage return character
- A new line character
- A vertical tab character
- A form feed character

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
\s	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("\\s")

or simply:

/\s/
```

Syntax with modifiers

```
new RegExp("\\s", "g")

or simply:

/\s/g
```

JavaScript RegExp \S Metacharacter

Example

Do a global search for non-whitespace characters in a string:

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

Try it Yourself »

Definition and Usage

The \S metacharacter is used to find a non-whitespace character.

A whitespace character can be:

- A space character
- A tab character
- A carriage return character
- A new line character
- A vertical tab character
- A form feed character

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
\S	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("\\S")

or simply:

/\S/
```

Syntax with modifiers

```
new RegExp("\\S", "g")

or simply:

/\S/g
```

JavaScript RegExp \b Metacharacter

Example

Do a search for "LO" at the beginning of a word in a string:

```
var str = "HELLO, LOOK AT YOU";
var patt1 = /\bLO/;
```

Try it Yourself »

Definition and Usage

The \b metacharacter is used to find a match at the beginning or end of a word.

Search for the pattern at the beginning of a word like this:

`\bLO`

Search for the pattern at the end of a word like this:

`LO\b`

If no match is found, it returns null.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
\b	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("\\bregexp")

or simply:

/\bregexp/
```

Syntax with modifiers

```
new RegExp("\\bregexp", "g")

or simply:

/\bregexp/g
```

More Examples

Example

Do a search for "LO" at the END of a word in a string:

```
var str = "HELLO, LOOK AT YOU";
var patt1 = /LO\b/;
```

Try it Yourself »

JavaScript RegExp \B Metacharacter

Example

Find the first occurrence of "LO" where it is NOT at the beginning of a word:

```
var str = "HELLO, LOOK AT YOU!";
var patt1 = /\BLO/;
```

Try it Yourself »

Definition and Usage

The \B metacharacter is used to find a match, but where it is NOT at the beginning/end of a word.

Search for the pattern NOT at the beginning of a word like this:

`\BLO`

Search for the pattern NOT at the end of a word like this:

`LO\B`

If no match is found, it returns null.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
\B	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("\\Bregexp")

or simply:

/\Bregexp/
```

Syntax with modifiers

```
new RegExp("\\Bregexp", "g")

or simply:

/\Bregexp/g
```

More Examples

Example

Find the first occurrence of "LO" where it is NOT at the END of a word:

```
var str = "HELLO, LOOK AT YOU";
var patt1 = /LO\B/;
```

Try it Yourself »

JavaScript RegExp \0 Metacharacter

Example

Search for a NUL character in a string:

```
var str = "Visit W3Schools.\0Learn Javascript.";
var patt1 = /\0/;
```

Try it Yourself »

Definition and Usage

The \0 metacharacter is used to find NUL character.
\0 returns the position where the NUL character was found. If no match is found, it returns -1.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
\0	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("\\0")

or simply:

/\0/
```

JavaScript RegExp \n Metacharacter

Example

Search for a newline character in a string:

```
var str = "Visit W3Schools.\nLearn Javascript.";
var patt1 = /\n/;
```

Try it Yourself »

Definition and Usage

The \n character is used to find a newline character.

\n returns the position where the newline character was found. If no match is found, it returns -1.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
\n	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("\\n")

or simply:

/\n/
```

JavaScript RegExp \f Metacharacter

Example

Search for a form feed character in a string:

```
var str = "Visit W3Schools.\fLearn Javascript.";
var patt1 = /\f/;
```

Try it Yourself »

Definition and Usage

The \f metacharacter is used to find a form feed character.
\f returns the position where the form feed character was found. If no match is found, it returns -1.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
\f	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("\\f")

or simply:

/\f/
```

JavaScript RegExp \r Metacharacter

Example

Search for a carriage return character in a string:

```
var str = "Visit W3Schools.\rLearn Javascript.";
var patt1 = /\r/;
```

Try it Yourself »

Definition and Usage

The \r metacharacter is used to find a carriage return character.
\r returns the position where the carriage return character was found. If no match is found, it returns -1.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
\r	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("\\r")

or simply:

/\r/
```

JavaScript RegExp \t Metacharacter

Example

Search for a tab character in a string:

```
var str = "Visit W3Schools.\tLearn Javascript.";
var patt1 = /\t/;
```

Try it Yourself »

Definition and Usage

The \t metacharacter is used to find a tab character.
\t returns the position where the tab character was found. If no match is found, it returns -1.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
\t	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("\\t")

or simply:

/\t/
```

JavaScript RegExp \v Metacharacter

Example

Search for a vertical tab character in a string:

```
var str = "Visit W3Schools.\vLearn Javascript.";
var patt1 = /\v/;
```

Try it Yourself »

Definition and Usage

The \v metacharacter is used to find a vertical tab character.
\v returns the position where the vertical tab character was found. If no match is found, it returns -1.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
\v	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("\\v")

or simply:

/\v/
```

JavaScript RegExp \xxx Metacharacter

Example

Do a global search for octal number 127 (W) in a string:

```
var str = "Visit W3Schools. Hello World!";
var patt1 = /\127/g;
```

Try it Yourself »

Definition and Usage

The \xxx character is used to find the Latin character specified by an octal number xxx.
If no match is found, it returns null.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
\xxx	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("\\xxx")

or simply:

/\xxx/
```

Syntax with modifiers

```
new RegExp("\\xxx", "g")

or simply:

/\xxx/g
```

JavaScript RegExp \xdd Metacharacter

Example

Do a global search for the hexadecimal number 57 (W) in a string:

```
var str = "Visit W3Schools. Hello World!";
var patt1 = /\x57/g;
```
Try it Yourself »

Definition and Usage

The \xdd character is used to find the Latin character specified by a hexadecimal number dd.
If no match is found, it returns null.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
\xdd	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("\\xdd")

or simply:

/\xdd/
```

Syntax with modifiers

```
new RegExp("\\xdd", "g")

or simply:

/\xdd/g
```

JavaScript RegExp \udddd Metacharacter

Example

Do a global search for the hexadecimal number 0057 (W) in a string:

```
var str = "Visit W3Schools. Hello World!";
var patt1 = /\u0057/g;
```
Try it Yourself »

Definition and Usage

The \udddd character is used to find the Unicode character specified by a hexadecimal number dddd.
If no match is found, it returns null.

Browser Support

Expression	Chrome	Edge	Firefox	Safari	Opera
\udddd	Yes	Yes	Yes	Yes	Yes

Syntax

```
new RegExp("\\udddd")

or simply:

/\udddd/
```

Syntax with modifiers

```
new RegExp("\\udddd", "g")

or simply:

/\udddd/g
```