JavaScript RegExp

❮ Previous Next ❯

Regular Expressions

A **Regular Expression** is a sequence of characters that forms a **search pattern**.

Regex is a common shorthand for a regular expression.

JavaScript **RexExp** is an **Object** for handling Regular Expressions.

RegExp are be used for:

- Text searching
- Text replacing
- Text validation

Example

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

```
let text = "Visit W3Schools";
let n = text.search(/w3schools/i);
```

Try it Yourself »

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

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

Regular Expression Syntax

```
/pattern/modifier flags;
```

Using String Methods

Regular expressions are often used with the **string methods**:

Method	Description
match(*regex*)	Returns an Array of results
replace(*regex*)	Returns a new String
search(*regex*)	Returns the index of the first match

Using String match()

Search for "W3schools" in a string:

```
let text = "Visit W3Schools";
let n = text.match(/W3schools/);
```

Try it Yourself »

Using String replace()

```
let result = text.replace(/Microsoft/i, "W3Schools");
```

Try it Yourself »

Using String search()

Search for "W3Schools" in a string:

```
let text = "Visit W3Schools";
let n = text.search(/W3Schools/);
```

Try it Yourself »

RexExp Alteration (OR)

In a regular expression an **alteration** is denoted with a vertical line character **|**.

An alteration matches any of the alternatives separated with **|**.

Example

A global search for the alternatives (red|green|blue):

```
let text = "Black, white, red, green, blue, yellow.";

let result = text.match(/red|green|blue/g);
```

Try it Yourself »

```
/pattern/flags
```

Regular expression flags are parameters that can modify how a pattern is used, such as making it case-insensitive or global.

These are the most common:

Flag	Description
/g	Performs a global match (find all)
/i	Performs case-insensitive matching
/u	Enables Unicode support (new 2015)

The /g Flag (Global)

The **/g** flag matches all occurrences of the pattern, rather than just the first one.

Example

A global search for "is" in a string:

```
let text = "Is this all there is?";
const pattern = /is/g;

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

Try it Yourself »

The /i Flag (Insensitive)

The **/i** flag makes a match case-insensitive: /abc/i matches "abc", "AbC", "ABC".

```
let text = "Visit W3Schools";
const pattern = /w3schools/i;
let result = text.match(pattern);
```

Try it Yourself »

Learn More:

JavaScript RegExp Flags

RexExp Metacharacters

```
// Match words
const pattern = /\w/;
```

Metacharacters are characters with a special meaning.

They can be used to match digts, words, spaces, and more.

These are the most common:

Meta	Description
\d	Matches Digits
\w	Matches Words
\s	Matches Spaces

Example

A global search for digits in a string:

```
let text = "Give 100%!";
const pattern = /\d/g;

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

Try it Yourself »

RegExp \w (word) Metacharacter

The \w metacharacter matches word characters.

A word character is a character a-z, A-Z, 0-9, including _ (underscore).

Example

A global search for word characters:

```
let text = "Give 100%!";
const pattern = /\w/g;

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

Try it Yourself »

Learn More:

JavaScript RegExp Quantifiers

```
// Match at least one zero
const pattern = /0+/;
```

Quantifiers define the numbers of characters or expressions to match.

These are the most common:

Code	Description
x*	Matches zero or more occurrences of x
x?	Matches zero or one occurrences of x
x{n}	Matches n occurences of x

The n? Quantifier

x? matches zero or one occurrences of x.

Example

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

```
let text = "1, 100 or 1000?";
const pattern = /10?/g;

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

Try it Yourself »

JavaScript RegExp Quantifiers

Regular Expression Assertions

```
// Match beginning of string
const pattern = /^W3Schools/;

// Match end of string
const pattern = /W3Schools$/;
```

Assertions matches **Boundaries** and **Lookarounds**:

String Boundaries and Word Boundaries.

Lookarounds: Lookaheads and Lookbehinds.

These are the most common:

Syntax	Name	Description
^	String boundary	Matches the beginning of a string
$	String boundary	Matches the end of a string
\b	Word boundary	Matches the beginning or end of a word
(?=...)	Lookahead	Matches the subsequent string
(?<=...)	Lookbehind	Matches the previous string

RegExp ^ Metacharacter

Examples

Test if a string starts with W3Schools:

```
const pattern = /^W3Schools/;
let text = "W3Schools Tutorial";

let result = pattern.test(text); // true
```

Try it Yourself »

```
const pattern = /^W3Schools/;
let text = "Hello W3Schools";

let result = pattern.test(text); // false
```

Try it Yourself »

RegExp $ Metacharacter

The $ metacharacter matches the end of a string.

Test if a string ends with W3Schools:

```
const pattern = /W3Schools$/;
let text = "Hello W3Schools";

let result = pattern.test(text); // true
```

Try it Yourself »

```
const pattern = /W3Schools$/;
let text = "W3Schools tutorial";

let result = pattern.test(text); // false
```

Learn More:

JavaScript RegExp Assertions

JavaScript RegExp Character Classes

```
// Match Digits
const pattern = /[0-9]/;
```

Character Classes are characters enclosed in square brackets **[]**.

A character class matches any character from a set within brackets.

These are the most common:

Class	Description
[a]	Matches the character between the brackets
[abc]	Matches all characters between the brackets
[a-z]	Matches all characters in the range from a to z
[0-9]	Matches all characters in the range from 0 to 9

Example [0-9]

A global search for the characters "0" to "9" in a string:

```
let text = "More than 1000 times";
const pattern = /[0-9]/g;
```

Try it Yourself »

Learn More:

JavaScript RegExp Character Classes

See Also:

JavaScript RegExp Patterns

JavaScript RegExp Objects

JavaScript RegExp Methods

Exercise ?

What does the `i` modifier represent in Regular Expressions?

○ Perform global matching (find all occurences)

○ Perform case-sensitive matching

○ Perform case-insensitive matching

Submit Answer »

COLOR PICKER

GET CERTIFIED FOR TEACHERS

FOR BUSINESS CONTACT US

Top Tutorials

HTML Tutorial
CSS Tutorial
JavaScript Tutorial
How To Tutorial
SQL Tutorial
Python Tutorial
W3.CSS Tutorial
Bootstrap Tutorial
PHP Tutorial
Java Tutorial
C++ Tutorial
jQuery Tutorial

Top References

HTML Reference
CSS Reference
JavaScript Reference
SQL Reference
Python Reference
W3.CSS Reference
Bootstrap Reference
PHP Reference
HTML Colors
Java Reference
Angular Reference
jQuery Reference

Top Examples

HTML Examples
CSS Examples
JavaScript Examples
How To Examples
SQL Examples
Python Examples
W3.CSS Examples
Bootstrap Examples
PHP Examples
Java Examples
XML Examples
jQuery Examples

Get Certified

HTML Certificate
CSS Certificate
JavaScript Certificate
Front End Certificate
SQL Certificate
Python Certificate
PHP Certificate
jQuery Certificate
Java Certificate
C++ Certificate
C# Certificate
XML Certificate

schools

Tutorials ▾ Exercises ▾ Services ▾ Sign In

≡ . CSS JAVASCRIPT SQL PYTHON JAVA PHP HOW TO W3.CSS C

FORUM ABOUT ACADEMY

W3Schools is optimized for learning and training. Examples might be simplified to improve reading and learning.
Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness
of all content. While using W3Schools, you agree to have read and accepted our terms of use, cookie and privacy policy.