**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.**
* 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:**

/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:

let text = "Visit W3Schools!";  
let n = text.search("W3Schools"); // Result n=6

**Using String search() With a Regular Expression**

**Example:**

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

let text = "Visit W3Schools";  
let n = text.search(/w3schools/i); // Result n=6

**Using String replace() With a String**

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

let text = "Visit Microsoft!";  
let result = text.replace("Microsoft", "W3Schools");

So, result = “Visit W3Schools!”

**Use String replace() With a Regular Expression**

**Example:**

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

let text = "Visit Microsoft!";  
let result = text.replace(/microsoft/i, "W3Schools");

The result will be:

Visit W3Schools!

**Regular Expression Modifiers**

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

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

**Regular Expression Patterns**

**Brackets** are used to find a range of characters:

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

**Metacharacters** are characters with a special meaning:

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

**Quantifiers define quantities:**

|  |  |  |
| --- | --- | --- |
| **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* |  |

**Using the RegExp Object**

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

**Using test() function**

* 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:**

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

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

The above two lines can be written as

/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:**

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

Result: The best things in life are free!

# JavaScript String match()

### Examples

A search for "ain" using a string:

let text = "The rain in SPAIN stays mainly in the plain";  
text.match("ain");

A search for "ain" using a regular expression:

let text = "The rain in SPAIN stays mainly in the plain";  
text.match(/ain/);

A global search for "ain":

let text = "The rain in SPAIN stays mainly in the plain";  
text.match(/ain/g);

A global, case-insensitive search:

let text = "The rain in SPAIN stays mainly in the plain";  
text.match(/ain/gi);

## Definition and Usage

The match() method matches a string against a regular expression \*\*

The match() method returns an array with the matches.

The match() method returns null if no match is found.

## Note

\*\* If the search value is a string, it is converted to a regular expression.

## Syntax

*string*.match(match)

|  |  |
| --- | --- |
| Parameter | Description |
| match | Required. The search value. A regular expression (or a string that will be converted to a regular expression). |

## Return Values

|  |
| --- |
|  |
| Type | Description |
| An array or null | An array containing the matches. null if no match is found. |

## The Difference Between String match() and String search()

* The match() method returns an array of matches.
* The search() method returns the position of the first match.

## Regular Expression Search Methods

In JavaScript, a regular expression text search, can be done with different methods.

With a **pattern** as a regular expression, these are the most common methods:

|  |  |
| --- | --- |
| **Example** | **Description** |
| text.match(**pattern**) | [The String method match()](https://www.w3schools.com/jsref/jsref_match.asp) |
| text.search(**pattern**) | [The String method search()](https://www.w3schools.com/jsref/jsref_search.asp) |
| **pattern**.exec(text) | [The RexExp method exec()](https://www.w3schools.com/jsref/jsref_regexp_exec.asp) |
| **pattern**.test(text) | [The RegExp method test()](https://www.w3schools.com/jsref/jsref_regexp_test.asp) |