

# Array.prototype.join()

**Summary:** in this tutorial, you'll learn how to use the JavaScript Array `join()` method to concatenate all elements of an array into a string separated by a separator.

## Introduction to the JavaScript Array `join()` method

The `join()` method allows you to concatenate all elements of an [array](#) and return a new [string](#).

Here's the syntax of the `join()` method:

```
const str = array.join(separator);
```

In this syntax:

- `separator` is an optional parameter that separates each pair of adjacent elements in the result string. The default is a comma ( `,` ).

If the array has one element, the `join()` method returns that element as a string without the trailing `separator` . When the array is empty, the `join()` method returns an empty string. In the case of an empty array, the `join()` method converts them to strings before joining.

Note that the `join()` method converts `undefined` , `null` , and empty array `[]` to an empty string ( `''` ).

## JavaScript Array `join()` method examples

Let's take some examples of using the `join()` method.

### 1) Basic Array `join()` method example

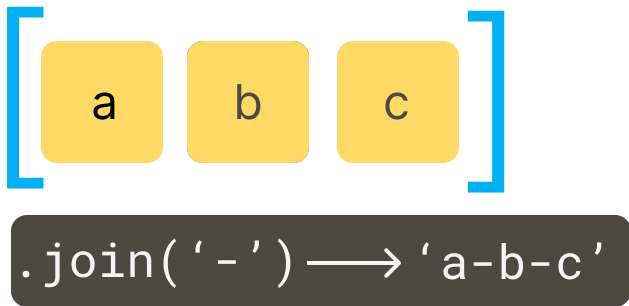
The following example uses the `join()` method to concatenate elements in an array into a string:

```
const array = ['a', 'b', 'c'];
const str = array.join('-');

console.log({ str });
```

Output:

```
{ str: 'a-b-c' }
```



## 2) Using the `join()` method to join CSS classes

The following example uses the `join()` method to join CSS classes:

```
const cssClasses = ['btn', 'btn-primary', 'btn-active'];
const btnClass = cssClasses.join(' ');

console.log(btnClass);
```

Output:

```
btn btn-primary btn-active
```

How it works.

First, define an array that stores the CSS classes:

```
const cssClasses = ['btn', 'btn-primary', 'btn-active'];
```

Second, join the CSS classes into a string separated by a space using the `join()` method:

```
const btnClass = cssClasses.join(' ');
```

Third, display the result to the console:

```
console.log(btnClass);
```

### 3) Generating HTML using the `map()` and `join()` methods

The following example uses the `map()` and `join()` method to render HTML based on an array:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>JavaScript Array join() method</title>
  </head>
  <body>
    <ul class="root"></ul>
    <script>
      const todos = [
        { id: 1, text: 'Learn HTML' },
        { id: 2, text: 'Learn CSS' },
        { id: 3, text: 'Learn JavaScript' },
      ];

      const renderedTodos = todos
        .map((todo) => `<li>${todo.text}</li>`)
        .join('');

      document.querySelector('.root').innerHTML = renderedTodos;
    </script>
  </body>
</html>
```

Output:

- Learn HTML
- Learn CSS
- Learn JavaScript

How it works.

First, declare an ul with the root class in the HTML document:

```
<ul class="root"></ul>
```

Second, map each item in the todos array into a list item and join them into a single string:

```
const renderedTodos = todos
  .map((todo) => `<li>${todo.text}</li>`)
  .join('');
```

Third, set the innerHTML of the root element to the HTML string:

```
document.querySelector('.root').innerHTML = renderedTodos;
```

### 3) Using the JavaScript Array join() method to replace all occurrences of a string

This example uses the JavaScript Array `join()` method to [replace all occurrences](#) of the space `' '` by the hyphen `( - )`:

```
const title = 'JavaScript array join example';
const url = title.split(' ').join('-').toLowerCase();

console.log(url);
```

Output:

How it works:

- First, split the **title** string by the space into an array by using the `split()` string method.
- Second, concatenate all elements in the result array into a string by using the `join()` method.
- Third, convert the result string to lowercase by using the `toLowerCase()` method.

## Summary

- Use the JavaScript Array `join()` method to concatenate all elements of an array into a string separated by a separator.