JavaScript Array.prototype.join()

The JavaScript join method converts an array into a string.

Converting an array to a string in JavaScript may seem like an unlikely scenario at first, since they are dissimilar types, but this is not as uncommon as one might think. I've found that when I want to build a complex string, organizing the various parts of that string into array elements can be a helpful approach. For example: I have to build many vanilla JavaScript applications that feature custom CSS. The CSS is completely managed by the JavaScript code, but I ultimately inject a new **STYLE** element into the page, with the custom CSS as the content of that element. In building these custom styles, I organize each CSS rule into a string that is an element of an array. Then, after all of the custom CSS rules are created, I use the **join()** method to combine all of those array elements into one long string, and make that string the content of the **STYLE** element that is injected into the page.

At first glance, it may seem a bit odd to use a space, hyphen or forward slash as the separator, but as a developer, you are likely to find yourself in many situations in which the business requirements require you to solve unexpected problems. Converting a number of array elements to a string and separating each element with an odd character will be a challenge you will run into, so be prepared; if it has not happened yet, it will! Fortunately, the **Array.prototype.join()** provides an elegant solution to this problem.

If you pass no arguments to the JavaScript **join** method, then there will be no space between the characters of the string. Or, you can pass an argument that determines how to join the elements of the array. Which character(s) you provide is up to you. The most common practice is to use the default, which is a comma (**" , "**), but again, the choice is completely yours.

Try it yourself !

Edit in JSFiddle

- JavaScript
- Result

```
var foo = ['a', 'b', 'c', 'd', 'e', 'f'];

//the join method combines all elemetns of an array
//and returns them as a string

//the default separator is a comma: ','
foo.join(); // 'a,b,c,d,e,f'

console.info('join using the default separator');
console.log(foo.join()); // 'a b c d e f'

//of you can provide a specific separator:

console.info('join using a space');
console.log(foo.join(' ')); // 'a b c d e f'

console.info('join using: ","');
console.log(foo.join(',')); // 'a,b,c,d,e,f'

console.info('join using: "-"');
console.log(foo.join('-')); // 'a-b-c-d-e-f'

console.info('join using: "/"');
console.log(foo.join('/')); // 'a/b/c/d/e/f'
```

Click the **JavaScript** tab in the above example. We have an array with six elements: **['a', 'b', 'c', 'd', 'e', 'f']**. When we call the **join** method on that array in the first **console.log** statement, it returns the string: **"a,b,c,d,e,f"**. This is the default behavior. That is to say: when you do not provide a separator argument, the characters in the returned string are separated by a comma (**" , "**). In the following examples we do provide a separator argument. In each case, you will see that the separator is used to create the returned string.

Video Example Code

If you want to download the example code, visit this Github page, and then follow the instructions: bit.ly/kcv-javascript-array-join-fiddle

Summary

String manipulation in JavaScript can be tedious, as converting an array to a string is a problem that might tempt one to use a **for-loop** as a solution. But the **Array.prototype.join()** method was specifically designed to solve this problem, as it negates the need for any kind of for-loop or other iteration patterns. You can simply chain the **join()** method from your array reference and then pass a character as an argument, which tells the **join()** method how you want to separate the elements of the array when converting to a string. In the long run, it really is a smooth way to go.