KEVIN CHISHOLM – BLOG

Web Development Articles and Tutorials

JavaScript String.prototype.split()

Home › JavaScript › String.prototype › JavaScript String.prototype.split()

The JavaScript split method turns a string into an array. You simply need to pass one or more characters as the separator.

Some may find it odd that all JavaScript strings have a **prototype** object. After all, isn't a string a primitive type? Well, the answer is "yes": a string is a primitive in JavaScript, but the string primitive has a prototype object "**wrapper**". This wrapper prototype object actually wraps the primitive string temporarily, providing a number of properties and methods. Now one of the properties provided is the **split()** method, which allows you to turn the string into an array whose elements are specific parts of the string. And those parts are determined by the separator character (or characters) that you provide as the first argument.

When you need to convert a string into an array, you first need to decide how to split that string. In other words: what *character* in the string indicates a new array element? For example, if you determine that any single space is a separator, then every word in the string will become a new array element.

Now this is a fairly typical scenario (and perfectly valid). Another common case is when you have a string of words or phrases separated by commas, and you need to turn each word or phrase into an array element. In this case, the comma is your separator character, and you'll need to pass it to the **split()** method. So keep two things in mind: **1)** If you do not provide an argument to the **split()** method, it will return an array with one element: the original string (i.e. the split() method did not understand how to split the array!). **2)** If you provide two quotes as the argument to the **split()** method, it will return an array and every single character in the original string will be an array element (this could potentially be a large array).

Try it yourself !

Edit in JSFiddle

- Result
- JavaScript

```
var testString = 'I want to be an array';

//the split method turns a string into an array.

console.info('if we pass an empty string as an argument, then every letter becomes an array element');
console.dir(testString.split(''));

console.info('if we pass an argument, then that determines where the string is split');
console.dir(testString.split(' '));

console.info('if we split on " be ", then whatever is on either side of it becomes an array element');
console.dir(testString.split(' be '));
```

In above example click the **JavaScript** tab. There are a series of **console.dir** statements. We use **console.dir** because the JavaScript **split** method returns an array, so we want to inspect that array. We are using the string: **“I want to be an array”** for each example.

In the first **console.dir** statement, we can see that passing no arguments to the **split** method returns an array where every character in the string is an element (including spaces). In the second **console.dir** statement we use a single space as the separator character **(” “)**. This returns an array were *every* word in our string becomes an element. This is a very common approach, and quite useful. The third and fourth **console.dir** statements show other ways that you can provide a separator. The results are a bit odd, but the main point here is that it is really up to you: you can use any character(s) in that string in order to convert the string to an array.

Summary

So, to recap, every JavaScript string primitive has a temporary object wrapper that provides various properties and methods. That string primitive wrapper prototype includes a **split()** method, which allows you to convert that string into an array. So, you need to let the **split()** method know how to “split” the string into the array elements, and that is accomplished by the one or more characters you provide as the first argument.