

An array is a special variable, which can hold more than one value:

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>

<p id="demo"></p>

<script>
const cars = ["Saab", "Volvo", "BMW"];
document.getElementById("demo").innerHTML = cars;
</script>

</body>
</html>
```

JavaScript Arrays

Saab, Volvo, BMW

Why Use Arrays?

If you have a list of items (a list of car names, for example), storing the cars in single variables could look like this:

```
let car1 = "Saab";
let car2 = "Volvo";
let car3 = "BMW";
```

However, what if you want to loop through the cars and find a specific one? And what if you had not 3 cars, but 300?

The solution is an array!

An array can hold many values under a single name, and you can access the values by referring to an index number.

Creating an Array

Using an array literal is the easiest way to create a JavaScript Array.

Syntax:

```
const array_name = [item1, item2, ...];
```

```

<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>

<p id="demo"></p>

<script>
const cars = [];
cars[0]= "Saab";
cars[1]= "Volvo";
cars[2]= "BMW";
document.getElementById("demo").innerHTML = cars;
</script>

</body>
</html>

```

JavaScript Arrays

Saab, Volvo, BMW

Using the JavaScript Keyword new

The following example also creates an Array, and assigns values to it:

```

<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>

<p id="demo"></p>

<script>
const cars = new Array("Saab", "Volvo", "BMW");
document.getElementById("demo").innerHTML = cars;
</script>

</body>
</html>

```

JavaScript Arrays

Saab, Volvo, BMW

Accessing Array Elements

You access an array element by referring to the **index number**:

```

<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>Bracket Indexing</h2>

<p>JavaScript array elements are accessed using numeric indexes (starting from 0).</p>

<p id="demo"></p>

<script>
const cars = ["Saab", "Volvo", "BMW"];
document.getElementById("demo").innerHTML = cars[0];
</script>

</body>
</html>

```

JavaScript Arrays

Bracket Indexing

JavaScript array elements are accessed using numeric indexes (starting from 0).

Saab

Changing an Array Element

This statement changes the value of the first element in cars:

```

<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>Bracket Indexing</h2>

<p>JavaScript array elements are accessed using numeric indexes (starting from 0).</p>

<p id="demo"></p>

<script>
const cars = ["Saab", "Volvo", "BMW"];
cars[0] = "Opel";
document.getElementById("demo").innerHTML = cars;
</script>

</body>
</html>

```

JavaScript Arrays

Bracket Indexing

JavaScript array elements are accessed using numeric indexes (starting from 0).

Opel,Volvo,BMW

Converting an Array to a String

The JavaScript method `toString()` converts an array to a string of (comma separated) array values.

```

<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>
<h2>The toString() Method</h2>

<p>The toString() method returns an array as a comma separated string:</p>

<p id="demo"></p>

<script>
const fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits.toString();
</script>

</body>
</html>

```

JavaScript Arrays

The toString() Method

The `toString()` method returns an array as a comma separated string:

Banana,Orange,Apple,Mango

Access the Full Array

With JavaScript, the full array can be accessed by referring to the array name:

```

<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Arrays</h1>

<p id="demo"></p>

<script>
const cars = ["Saab", "Volvo", "BMW"];
document.getElementById("demo").innerHTML = cars;
</script>

</body>
</html>

```

JavaScript Arrays

Saab,Volvo,BMW

Arrays are Objects

Arrays are a special type of objects. The `typeof` operator in JavaScript returns "object" for arrays.

But, JavaScript arrays are best described as arrays.

Arrays use **numbers** to access its "elements". In this example, `person[0]` returns John:

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Arrays</h2>

<p>Arrays use numbers to access its elements.</p>

<p id="demo"></p>

<script>
const person = ["John", "Doe", 46];
document.getElementById("demo").innerHTML = person[0];
</script>

</body>
</html>
```

JavaScript Arrays

Arrays use numbers to access its elements.

John