

# JavaScript Arrays

- description : JavaScript Arrays
- author :
- email : shlim@repia.com
- lastupdate : 2021-04-26

## The source of this article

[[[https://www.w3schools.com/js/js\\_events.asp](https://www.w3schools.com/js/js_events.asp)[https://www.w3schools.com/js/js\\_arrays.asp](https://www.w3schools.com/js/js_arrays.asp) JavaScript

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

## What is an Array?

( : ) ,

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

?

!

? 가 3 가 300

## Creating an Array

JavaScript

가

## Syntax

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

```
var cars = ["Saab", "Volvo", "BMW"];
```

(line breaks)

```
let cars = [  
  "Saab",  
  "Volvo",  
  "BMW"  
];
```

## Using the JavaScript Keyword new

```
let cars = new Array("Saab", "Volvo", "BMW");
```

, new Array() 가 , the array literal method)

## Access the Elements of an Array

(index number)

cars

```
let name = cars[0];
```

```
let cars = ["Saab", "Volvo", "BMW"];  
document.getElementById("demo").innerHTML = cars[1]; // Volvo
```

**Note:** 0 . [0] . [1]

# Changing an Array Element

cars

```
cars[0] = "Opel";
```

```
let cars = ["Saab", "Volvo", "BMW"];
cars[2] = "Opel";
document.getElementById("demo").innerHTML = cars; // Saab, Volvo, Opel
```

# Access the Full Array

JavaScript

```
let cars = ["Saab", "Volvo", "BMW"];
document.getElementById("demo").innerHTML = cars; // Saab, Volvo, BMW
```

# Arrays are Objects

. JavaScript typeof " "

, JavaScript

가

" "

person[0] John

## Array:

```
let person = ["John", "Doe", 46];
document.getElementById("demo").innerHTML = person[0]; // John
```

person.firstName John (members)

## Object

```
let person = { firstName: "John", lastName: "Doe", age: 46 };
```

```
document.getElementById("demo").innerHTML = person["firstName"]; // John
```

## Array Elements Can Be Objects

JavaScript

가

가

가

가

가

```
myArray[0] = Date.now;  
myArray[1] = myFunction;  
myArray[2] = myCars;
```

## Array Properties and Methods

JavaScript

```
let x = cars.length; // The length property returns the number of elements  
let y = cars.sort(); // The sort() method sorts arrays
```

## The length Property

length

(

)

```
let fruits = ["Banana", "Orange", "Apple", "Mango"];  
document.getElementById("demo").innerHTML = fruits.length;  
// the length of fruits is 4
```

length

가

## Accessing the First Array Element

```
let fruits = ["Banana", "Orange", "Apple", "Mango"];  
let first = fruits[0];
```

```
document.getElementById("demo").innerHTML = first; // Banana
```

## Accessing the Last Array Element

```
let fruits = ["Banana", "Orange", "Apple", "Mango"];
let last = fruits[fruits.length - 1];
// fruits[fruits.length-1] = fruits[4-1] = fruits[3]
document.getElementById("demo").innerHTML = last; // Mango
```

## Looping Array Elements

가 for loop .

```
let fruits, text, fLen, i; //
fruits = ["Banana", "Orange", "Apple", "Mango"]; // fruits
fLen = fruits.length; // 4

text = "<ul>";
for (i = 0; i < fLen; i++) {
  text += "<li>" + fruits[i] + "</li>";
}
text += "</ul>";

document.getElementById("demo").innerHTML = text;
```

Array.forEach() .

```
let fruits, text;
fruits = ["Banana", "Orange", "Apple", "Mango"];

text = "<ul>";
fruits.forEach(myFunction);
text += "</ul>";
document.getElementById("demo").innerHTML = text;

function myFunction(value) {
  text += "<li>" + value + "</li>";
}
```

## Adding Array Elements

가 가 "push()" .

```
let fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits;

function myFunction() {
  fruits.push("Lemon");
  document.getElementById("demo").innerHTML = fruits;
} // Banana, Orange, Apple, Mango, Lemon
```

length

가

.

```
let fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits;

function myFunction() {
  fruits[fruits.length] = "Lemon";
  document.getElementById("demo").innerHTML = fruits;
}
```

**WARNING!**

가

가

(undefined) "

(holes)"

```
let fruits, text, fLen, i;
fruits = ["Banana", "Orange", "Apple", "Mango"];
\fruits[6] = "Lemon";

fLen = fruits.length;
text = "";
for (i = 0; i < fLen; i++) {
  text += fruits[i] + "<br>";
}
document.getElementById("demo").innerHTML = text;
// Banana Orange Apple Mango undefined undefined Lemon
```

## Associative Arrays( )

가

.

가

( hashes)

.

JavaScript

가

.

JavaScript

가

.

```

let person = [];
person[0] = "John";
person[1] = "Doe";
person[2] = 46;
let x = person.length;
let y = person[0];
console.log(x); // person.length will return 3
console.log(y); // person[0] will return "John"
document.getElementById("demo").innerHTML = person[0] + " " + person.length;
// John 3

```

**WARNING!!**

, JavaScript

```

let person = {};
person["firstName"] = "John";
person["lastName"] = "Doe";
person["age"] = 46;
document.getElementById("demo").innerHTML = person[0] + "" + person.length;
// undefined()

```

## The Difference Between Arrays and Objects

JavaScript (arrays)\* 가 (numbered indexes)

JavaScript (objects) (named indexes)

가 가

## When to Use Arrays. When to use Objects.

- JavaScript
- ( )
- 

## Avoid new Array()

JavaScript new Array() 가

[] 가

```
let points = new Array(); // Bad  
let points = []; // Good
```

2

6

가

```
//  
let points = new Array(40, 100, 1, 5, 25, 10);  
let points = [40, 100, 1, 5, 25, 10];  
document.getElementById("demo").innerHTML = points[0];  
console.log(points); // 40
```

new

가

```
let points = new Array(40, 100); // Creates an array with two elements ( 40  
and 100)
```

```
let points = new Array(40);  
document.getElementById("demo").innerHTML = points[0];  
// Creates an array with 40 undefined elements!!!
```

## How to Recognize an Array

: 가 ?

JavaScript typeof가 "object"

```
let fruits = ["Banana", "Orange", "Apple", "Mango"];  
document.getElementById("demo").innerHTML = typeof fruits; // object
```

typeof JavaScript

### Solution 1:

, ECMAScript 5 Array.isArray()

```
let fruits = ["Banana", "Orange", "Apple", "Mango"];
```

```
document.getElementById("demo").innerHTML = Array.isArray(fruits); // true
```

ECMAScript 5가

### Solution 2:

, isArray() :

```
let fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = isArray(fruits);

function isArray(myArray) {
  return myArray.constructor.toString().indexOf("Array") > -1;
} // true
```

가 true .  
"Array" 가 , true .

### Solution 3:

instanceof 가 true .

```
let fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits instanceof Array; // true
```

, Javascript, Arrays

From: <http://rwiki.repia.com/> -

. - 2023.12

Permanent link: [http://rwiki.repia.com/doku.php?id=wiki:javascript:javascript\\_note:js\\_arrays&rev=1619412919](http://rwiki.repia.com/doku.php?id=wiki:javascript:javascript_note:js_arrays&rev=1619412919)



Last update: 2022/03/10 19:52