

# JavaScript Arrays

## 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";
```

## Creating an Array

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

Syntax:

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

## Example

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

## Example

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

```
12
13 <script>
14 let list=[10,20,30,40,50]
15 console.log(list)
16
17
18 </script>
19
20
21
22
```

No Issues

▼ (5) [10, 20, 30, 40, 50] ⓘ

- 0: 10
- 1: 20
- 2: 30
- 3: 40
- 4: 50

length: 5

▶ [[Prototype]]: Array(0)

>

# Accessing Array Elements

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

```
<script>
//You access an array element by | index number

let list=[10,20,30,40,50]
// console.log(list)

console.log(list[0])
console.log(list[1])
console.log(list[2])
console.log(list[3])
console.log(list[4])

</script>
```

Default levels ▼

10

20

30

40

50

>

```
<script>
// Stor string value in array
let fruits=['banana','apple','orange']
console.log(fruits)

console.log(fruits[0])
console.log(fruits[1])
console.log(fruits[2])

</script>
```

Default levels ▾ No Issues

▶ (3) ['banana', 'apple', 'orange']

banana

apple

orange

>

# The length Property

The `length` property of an array returns the length of an array (the number of array elements).

## Example

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

```
<script>  
// how to display length of an array  
let fruits=['banana','apple','orange']  
  
let count=fruits.length  
  
console.log(count)  
|
```

Default level

3

>

# Adding Array Elements

The easiest way to add a new element to an array is using the `push()` method:

```
// adding array element
let fruits = ["banana", "apple", "orange"];
fruits.push("lemon")

length=fruits.length

console.log(fruits)
console.log(length)
```

Default levels ▾ | No Issues

[index.ht](#)  
▼ (4) ['banana', 'apple', 'orange', 'lemon'] ⓘ  
 0: "banana"  
 1: "apple"  
 2: "orange"  
 3: "lemon"  
 length: 4  
 ► [[Prototype]]: Array(0)

4

[index.ht](#)

>

## Remove element from last

```
// Remove array element from last
let fruits = ["banana", "apple", "orange"];
fruits.pop()

length=fruits.length

console.log(fruits)
console.log(length)
```

Default levels ▾ | No Issues

▶ (2) ['banana', 'apple']

2

>

## Add array element from beginning

```
// Add array element from beginning
let fruits = ["banana", "apple", "orange"];
fruits.unshift("mango")

length=fruits.length

console.log(fruits)
console.log(length)
```

index.html:64

▼ (4) ['mango', 'banana', 'apple', 'orange'] ⓘ

- 0: "mango"
- 1: "banana"
- 2: "apple"
- 3: "orange"

length: 4

► [[Prototype]]: Array(0)

4 index.html:65

>



Remove array element from beginning

```
// Remove array element from beginning
let fruits = ["banana", "apple", "orange"];
fruits.shift()

length=fruits.length

console.log(fruits)
console.log(length)
```

▼ (2) ['apple', 'orange'] ⓘ  
 0: "apple"  
 1: "orange"  
 length: 2  
 ► [[Prototype]]: Array(0)

2

>

## display array element using for loop

```
// display array element using for loop
```

```
let fruits = ["banana", "apple", "orange"];  
length=fruits.length
```

```
for(let i=0;i<length;i=i+1)  
console.log(i+ " " +fruits[i])
```

Default levels ▾

No Issues

0 banana

1 apple

2 orange

>

```
// sort array element

let fruits = ["banana", "apple", "orange"];

// fruits = fruits.sort();
// fruits=fruits.sort().reverse();

length = fruits.length;
for (let i = 0; i < length; i = i + 1) console.log(i + " " + fruits[i]);
</script>
/html>
```