

JavaScript Array Iteration

- description : JavaScript Array Iteration
- author :
- email : shlim@repia.com
- lastupdate : 2021-05-06

The Source of this article

[JavaScript Array Iteration](#)
(Array iteration methods)

Array.forEach()

forEach() (, a callback function)

Example

```
let txt = "";
let numbers = [45, 4, 9, 16, 25];
numbers.forEach(myFunction);
document.getElementById("demo").innerHTML = txt;

function myFunction(value, index, array) {
  txt = txt + value + ", " // 45, 4, 9, 16, 25,
}
```

3 (arguments) 가

- (The item value)
- (The item index)
- (The array itself)

(value parameter)

Example

```
let txt = "";
let numbers = [45, 4, 9, 16, 25];
```

```
numbers.forEach(myFunction);
document.getElementById("demo").innerHTML = txt;

function myFunction(value) {
  txt = txt + value + "\," // 45, 4, 9, 16, 25,
}
```

Array.map()

map()

map()

map()

2

Example

```
let numbers1 = [45, 4, 9, 16, 25];
let numbers2 = numbers1.map(myFunction);

document.getElementById("demo").innerHTML = numbers2;
// 90,8,18,32,50
function myFunction(value, index, array) {
  return value * 2;
}
console.log(numbers1); // [45, 4, 9, 16, 25]
console.log(numbers2); // [90, 8, 18, 32, 50]
```

3

가

- value (The item value)
- index (The item index)
- array (The array itself)

가 value , index array

Example

```
let numbers1 = [45, 4, 9, 16, 25];
let numbers2 = numbers1.map(myFunction);

document.getElementById("demo").innerHTML = numbers2;
// 90,8,18,32,50
```

```
function myFunction(value) {
    return value * 2;
}
console.log(numbers1); // [45, 4, 9, 16, 25]
console.log(numbers2); // [90, 8, 18, 32, 50]
```

Array.filter()

filter()

18

Example

```
let numbers = [45, 4, 9, 16, 25];
let over18 = numbers.filter(myFunction);

document.getElementById("demo").innerHTML = over18;
// 45,25
function myFunction(value, index, array) {
    return value > 18;
}

console.log(numbers); // [45, 4, 9, 16, 25]
console.log(over18); // [45, 25]
```

3

가

- *(The item value)*
- *(The item index)*
- *(The array itself)*

,

Example

```
let numbers = [45, 4, 9, 16, 25];
let over18 = numbers.filter(myFunction);

document.getElementById("demo").innerHTML = over18;
// 45,25
function myFunction(value) {
    return value > 18;
}

console.log(numbers); // [45, 4, 9, 16, 25]
```

```
console.log(over18); // [45, 25]
```

Array.reduce()

reduce() , ()
reduce() . reduceRight()
reduce()

Example

```
let numbers = [45, 4, 9, 16, 25];
let sum = numbers.reduce(myFunction);

document.getElementById("demo").innerHTML = "The sum is " + sum; // The
sum is 99

function myFunction(total, value, index, array) {
    return total + value;
}

console.log(numbers); // [45, 4, 9, 16, 25]
console.log(sum); // 99
```

4

- (/) (The total(the initial value / previously returned value))
- (The item value)
- (The item index)
- (The array itself)

:

Example

```
let numbers = [45, 4, 9, 16, 25];
let sum = numbers.reduce(myFunction);

document.getElementById("demo").innerHTML = "The sum is " + sum; // The
sum is 99

function myFunction(total, value) {
```

```

        return total + value;
    }

console.log(numbers); // [45, 4, 9, 16, 25]
console.log(sum);    // 99

```

reduce() (initial value)

Example

```

let numbers = [45, 4, 9, 16, 25];
let sum = numbers.reduce(myFunction, 100);

document.getElementById("demo").innerHTML = "The sum is " + sum; // The
sum is 199

function myFunction(total, value) {
    return total + value;
}

console.log(numbers); // [45, 4, 9, 16, 25]
console.log(sum);    // 199

```

Array.reduceRight()

reduceRight() ()
 reduceRight() . reduce()
 reduceRight() .

Example

```

let numbers = [45, 4, 9, 16, 25];
let sum = numbers.reduce(myFunction);

document.getElementById("demo").innerHTML = "The sum is " + sum; // The
sum is 99

function myFunction(total, value, index, array) {
    return total + value;
}

console.log(numbers); // [45, 4, 9, 16, 25]

```

```
console.log(sum); // 99
```

4

- (/)
-
-
-

Example

```
let numbers = [45, 4, 9, 16, 25];
let sum = numbers.reduce(myFunction);

document.getElementById("demo").innerHTML = "The sum is " + sum; // The
sum is 99

function myFunction(total, value) {
    return total + value;
}

console.log(numbers); // [45, 4, 9, 16, 25]
console.log(sum); // 99
```

Array.every()

every()

18

Example

```
let numbers = [45, 4, 9, 16, 25];
let allOver18 = numbers.every(myFunction);

document.getElementById("demo").innerHTML = "All over 18 is " +
allOver18;
// All over 18 is false

function myFunction(value, index, array) {
    return value > 18;
}
console.log(numbers); // [45, 4, 9, 16, 25]
```

```
console.log(allOver18); // false
```

3

가

-
-
-

가

()

,

Example

```
let numbers = [45, 4, 9, 16, 25];
let allOver18 = numbers.every(myFunction);

document.getElementById("demo").innerHTML = "All over 18 is " +
allOver18;
// All over 18 is false

function myFunction(value) {
  return value > 18;
}
console.log(numbers); // [45, 4, 9, 16, 25]
console.log(allOver18); // false
```

Array.some() Internet Explorer 8

(Chrome Yes, Edge 9.0, Firefox Yes, Safari Yes, Opera Yes)

Array.some()

some()

18

Example

```
let numbers = [45, 4, 9, 16, 25];
let someOver18 = numbers.some(myFunction);

document.getElementById("demo").innerHTML = "Some over 18 is " +
someOver18;
// Some over 18 is true

function myFunction(value, index, array) {
  return value > 18;
}
```

```
console.log(numbers); // [45, 4, 9, 16, 25]
console.log(someOver18); // true
```

3

-
-
-

Array.some() Internet Explorer 8
(Chrome Yes, Edge 9.0, Firefox Yes, Safari Yes, Opera Yes)

Array.indexOf()

indexOf()

Note: 0 , 1

Example

“Apple” :

```
let fruits = ["Apple", "Orange", "Apple", "Mango"];
let a = fruits.indexOf("Apple");
document.getElementById("demo").innerHTML = "Apple is found in position
" + a;
// Apple is found in position 0
console.log(fruits); // ["Apple", "Orange", "Apple", "Mango"]
console.log(a); // 0
```

Array.indexOf() Internet Explorer 8
(Chrome Yes, Edge 9.0, Firefox Yes, Safari Yes, Opera Yes)

Syntax

```
array.indexOf(item, start)
```

item	.
start	.

Array.indexOf() -1

Array.lastIndexof()

Array.lastIndexof() Array.indexOf()

Example

“Apple” :

```
let fruits = ["Apple", "Orange", "Apple", "Mango"];
let a = fruits.lastIndexOf("Apple");
document.getElementById("demo").innerHTML = "Apple is found in position
" + (a + 1);
// Apple is found in position 3
```

Array.lastIndexof() Internet Explorer 8
 (Chrome Yes, Edge 9.0, Firefox Yes, Safari Yes, Opera Yes)

Syntax

`array.lastIndexOf(item, start)`

item	.
start	.

Array.find()

find()

18 ().

Example

```
let numbers = [4, 9, 16, 25, 29];
let first = numbers.find(myFunction);

document.getElementById("demo").innerHTML = "First number over 18 is " +
first;

function myFunction(value, index, array) {
  return value > 18;
}
```

3

가

-
-
-

Array.find()

(Chrome 45, Edge 12, Firefox 25, Safari 8, Opera 32)

Array.findIndex()

findIndex()

18

Example

```
let numbers = [4, 9, 16, 25, 29];
let first = numbers.findIndex(myFunction);

document.getElementById("demo").innerHTML = "First number over 18 has
index " + first;
// First number over 18 has index 3
function myFunction(value, index, array) {
    return value > 18;
}
```

3

가

-
-
-

Array.findIndex()

(Chrome 45, Edge 12, Firefox 25, Safari 8, Opera 32)

, Javascript, Array, Iteration, Methods

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

- 2023.12



Permanent link:
http://rwiki.requia.com/doku.php?id=wiki:javascript:javascript_note:js_array_iteration&rev=1620266512

Last update: 2022/03/10 19:52