JOIN US & LEARN
WEB DEVELOPMENT

MASTER ARRAYS IN **JAVASCRIPT** NOW

8 WAYS TO LOOP OVER AN ARRAY

- for loop
- while
- forEach
- every
- map
- filter
- reduce
- some

Learn all in one place →

```
const array = [10, 20, 30];
```

@developers_community_._ X @gowsami.dev

FOR LOOP

Basic

```
array = [ 1, 2, 3, 4, 5, 6 ];
for (index = 0; index < array.length; index++) {
    console.log(array[index]);
}
// OUTPUT
// 1 2 3 4 5 6
```

script.js

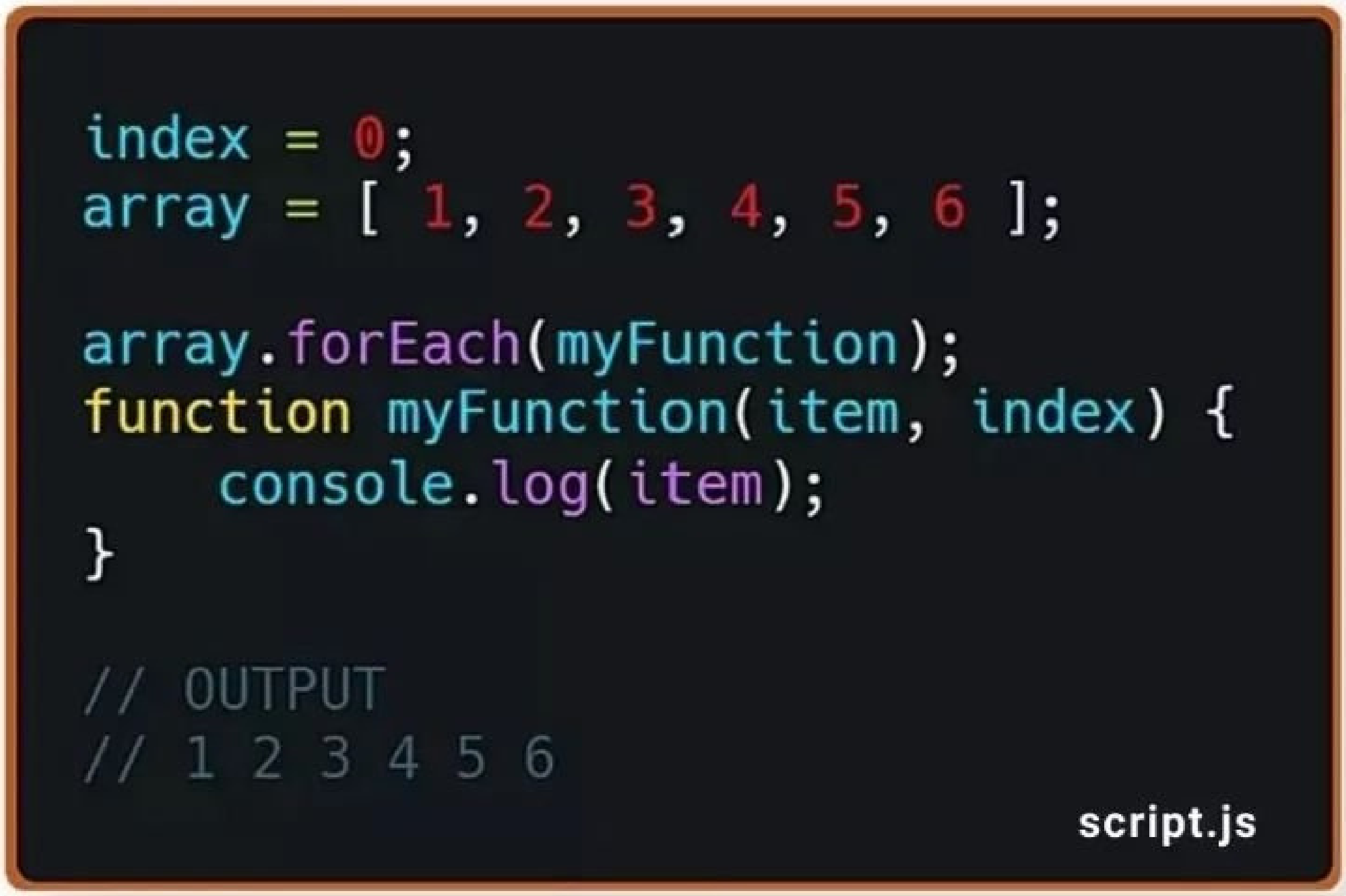
WHILE LOOP

```
index = 0;
array = [ 1, 2, 3, 4, 5, 6 ];

while (index < array.length) {
    console.log(array[index]);
    index++;
}
// OUTPUT
// 1 2 3 4 5 6
```

script.js

FOR EACH LOOP

The **forEach** method calls the provided function once for every array element **in the order.**

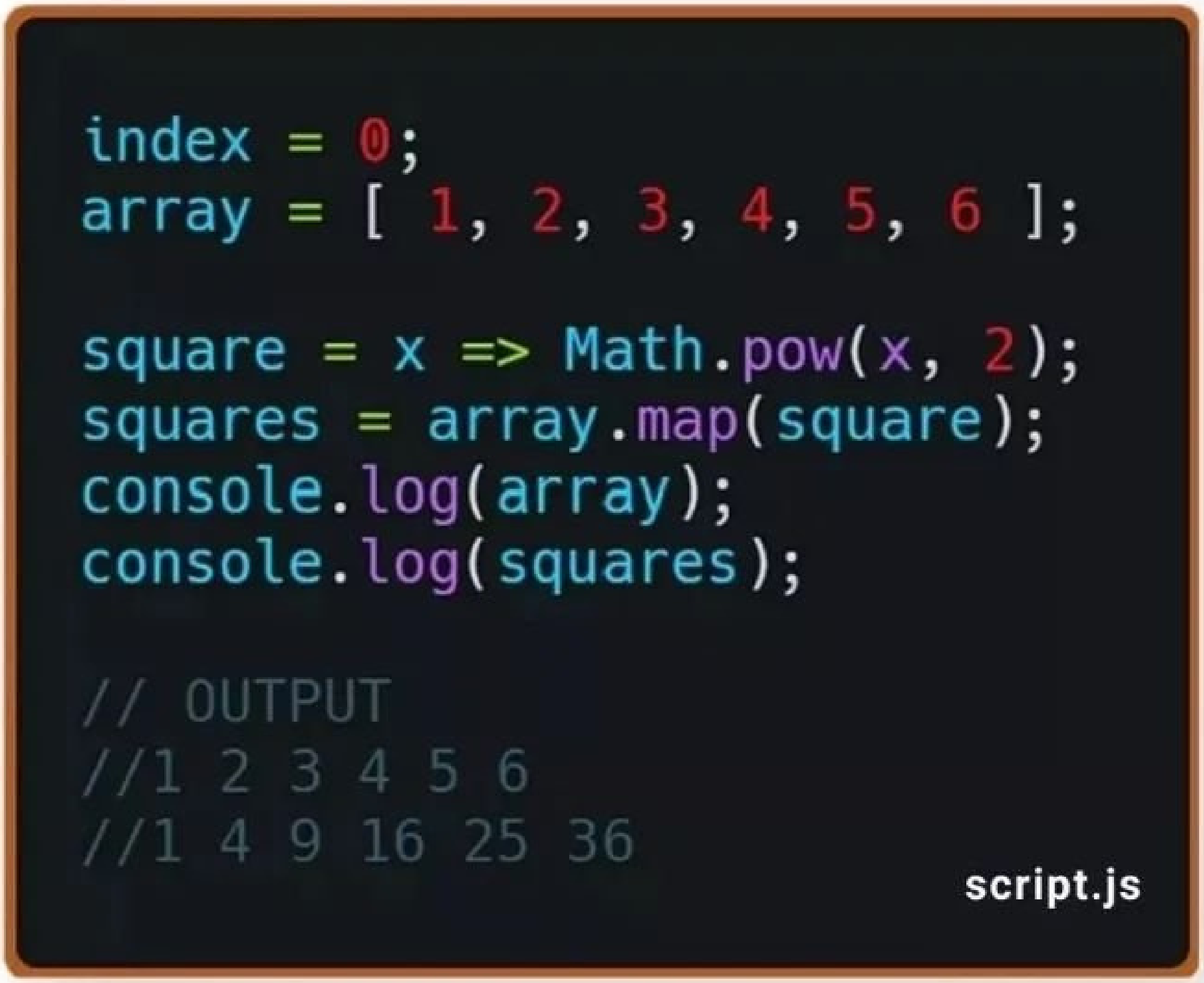


```
index = 0;
array = [ 1, 2, 3, 4, 5, 6 ];

array.forEach(myFunction);
function myFunction(item, index) {
    console.log(item);
}

// OUTPUT
// 1 2 3 4 5 6
```

script.js

MAP

A **map** applies a function over every element and then returns the **new array.**

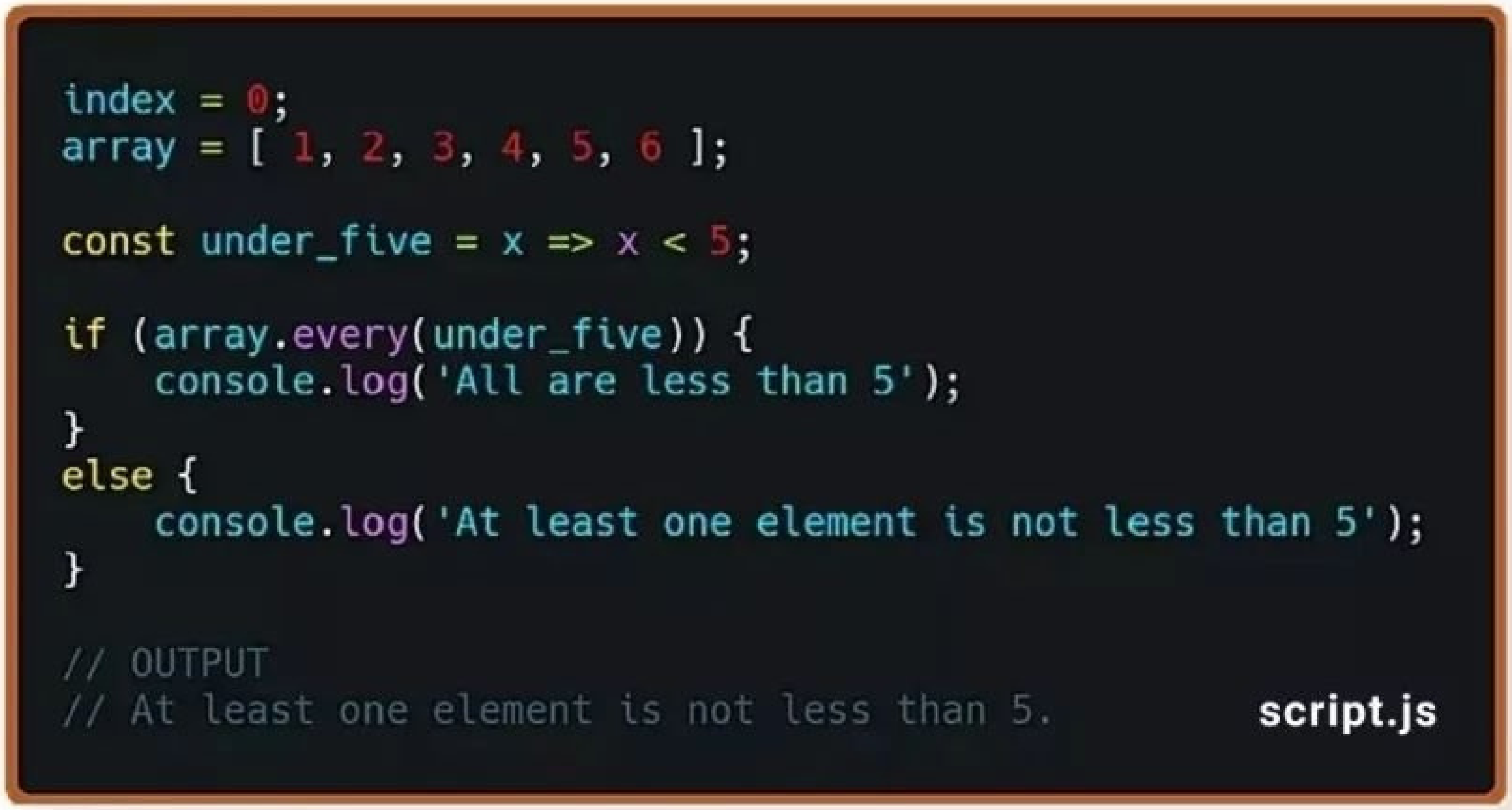


```
index = 0;
array = [ 1, 2, 3, 4, 5, 6 ];

square = x => Math.pow(x, 2);
squares = array.map(square);
console.log(array);
console.log(squares);

// OUTPUT
//1 2 3 4 5 6
//1 4 9 16 25 36
```

FILTER

EVERY

The **every()** method checks if all elements in an array pass a test **(provided as a function).**



```
index = 0;
array = [ 1, 2, 3, 4, 5, 6 ];

const under_five = x => x < 5;

if (array.every(under_five)) {
    console.log('All are less than 5');
}
else {
    console.log('At least one element is not less than 5');
}

// OUTPUT
// At least one element is not less than 5.
```

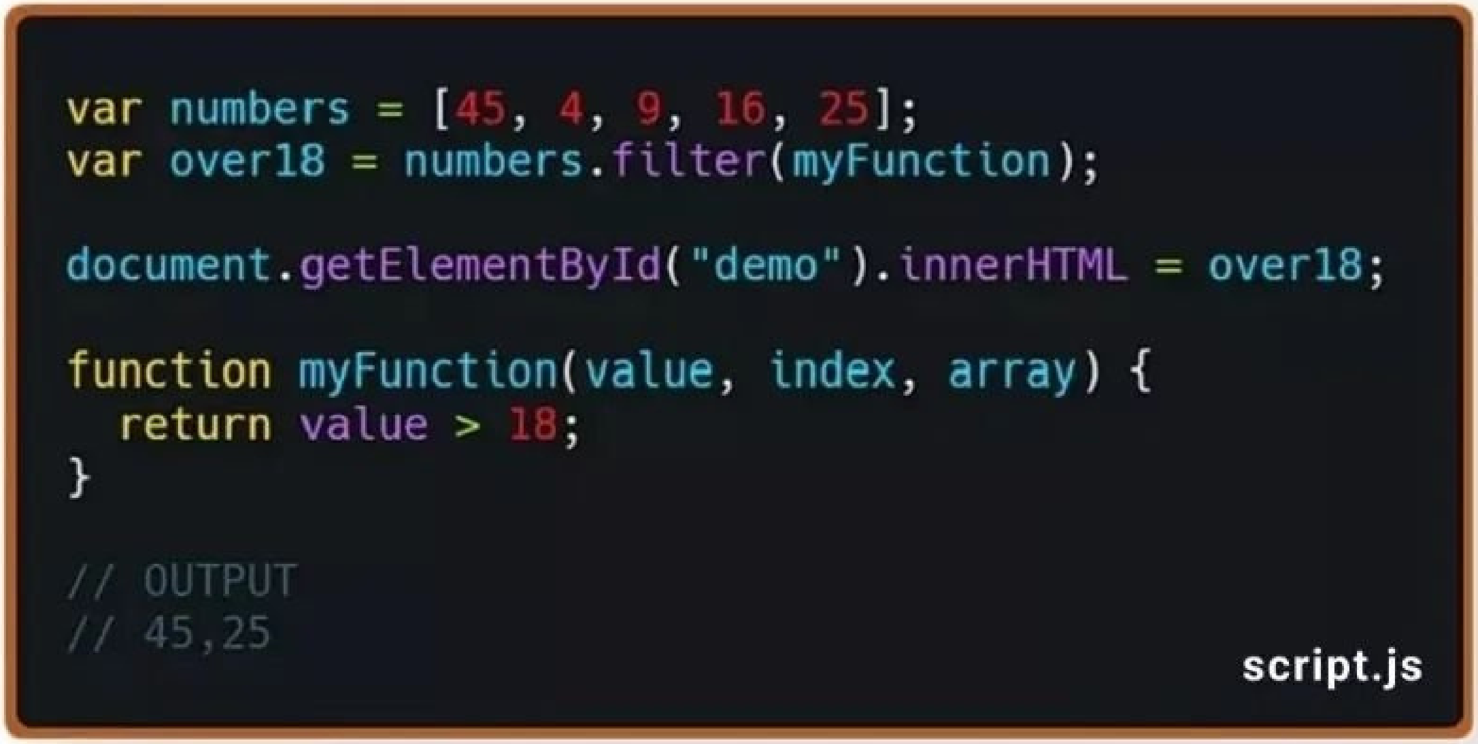
MAP →

SAVE THIS POST

JOIN US & LEARN
WEB DEVELOPMENT

FILTER

The **filter()** method **creates a new array** with array elements that **passes a test.**



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

document.getElementById("demo").innerHTML = over18;

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

// OUTPUT
// 45,25
```

REDUCE →

@developers_community_._ X @gowsami.dev

SAVE THIS POST

JOIN US & LEARN
WEB DEVELOPMENT

REDUCE

The **reduce()** method runs a function on each array element to produce **(reduce it to) a single value.**

The **reduce()** method **does not reduce** the original array.

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

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

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

// OUTPUT
// The sum is 99
```

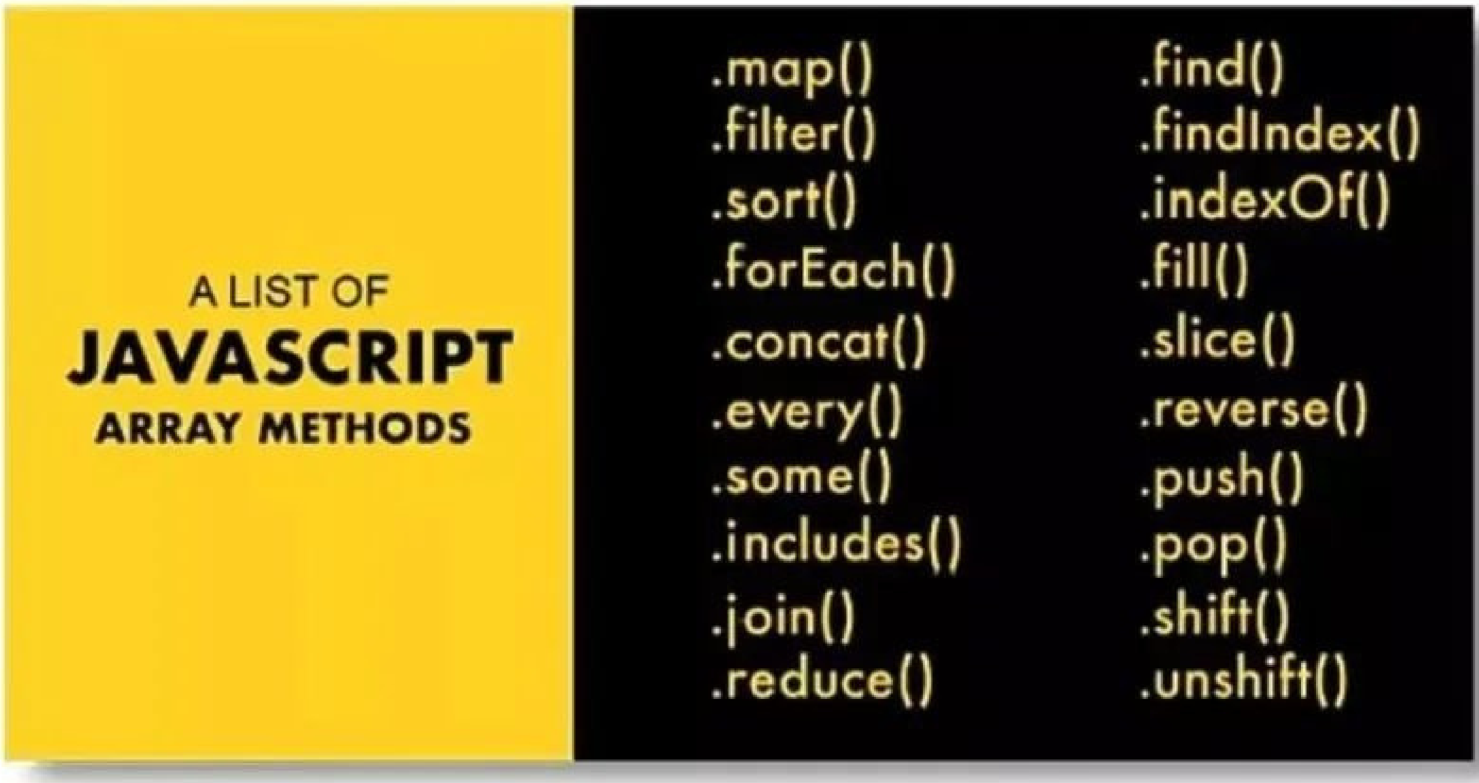
script.js

SOME →

@developers_community_._ X @gowsami.dev

ARRAY METHODS

"To make JavaScript array manipulation easier, we should use array methods to make our work easier and the code cleaner."



A LIST OF
JAVASCRIPT
ARRAY METHODS

.map()	.find()
.filter()	.findIndex()
.sort()	.indexOf()
.forEach()	.fill()
.concat()	.slice()
.every()	.reverse()
.some()	.push()
.includes()	.pop()
.join()	.shift()
.reduce()	.unshift()