Array Iterators

forEach()

In the last chapter, we saw that a for loop could be used to loop through each value in an array like so:

An alternative is to use the forEach() method.

const numbers = [1,2,3,4,5].forEach(function (value,index,array) {

return console.log(value,index,array.length)

})

console.log(numbers)

map()

method is very similar to the forEach() method. The difference is that it returns a new array that replaces each value with the return value of the callback function.

let colors = ['Red', 'Green', 'Blue']

for(let color of colors) {

console.log(color);

}

let values =colors.map(function(color) {

return color

})

console.log(values)

Reduce()

The reduce() method is another method that iterates over each value in the array, but this time it cumulatively combines each result to return just a single value. . This is often used to calculate statistics such as averages from data stored in an array. It usually takes two parameters.

const result = [1,2,3,4,5].reduce(function(acc,val) {

return acc + val

})

console.log(result)

Filter()

The filter() method returns a new array that only contains items from the original array that return true when passed to the callback. For example, we can filter an array of numbers to just the even numbers using the following code

const numbers = [ 2, 7, 6, 5, 11, 23, 12 ]

let even = numbers.filter(function (value) {

value%2 == 0

})

console.log(even)