JavaScript Iterators

code|cademy

The `.reduce()` Method

The .reduce() method iterates through an array and returns a single value.
In the above code example, the .reduce() method will sum up all the elements of the array. It takes a callback function with two parameters (accumulator, currentValue) as arguments. On each iteration, accumulator is the value returned by the last iteration, and the currentValue is the current element. Optionally, a second argument can be passed which acts as the initial value of the accumulator.

```javascript
const arrayOfNumbers = [1, 2, 3, 4];

const sum = arrayOfNumbers.reduce((accumulator, currentValue) => {
  return accumulator + currentValue;
});


console.log(sum); // 10
```

The `.forEach()` Method

The .forEach() method executes a callback function on each of the elements in an array in order.
In the above example code, the callback function containing a console.log() method will be executed 5 times, once for each element.

```javascript
const numbers = [28, 77, 45, 99, 27];

numbers.forEach(number => {
  console.log(number);
});
```

code|cademy

The `.filter()` Method

The .filter() method executes a callback function on each element in an array. The callback function for each of the elements must return either true or false . The returned array is a new array with any elements for which the callback function returns true .

In the above code example, the array filteredArray will contain all the elements of randomNumbers but 4 .

```javascript
const randomNumbers = [4, 11, 42, 14, 39];
const filteredArray = randomNumbers.filter(n => {
  return n > 5;
});
```

The `.map()` Method

The .map() method executes a callback function on each element in an array. It returns a new array made up of the return values from the callback function.

The original array does not get altered, and the returned array may contain different elements than the original array.

In the example code above, the .map() method is used to add ' joined the contest.' string at the end of each element in the finalParticipants array.

```javascript
const finalParticipants = ['Taylor', 'Donald', 'Don',
'Natasha', 'Bobby'];

// add string after each final participant
const announcements = finalParticipants.map(member => {
  return member + ' joined the contest.';
})

console.log(announcements);
```

⬇ **Print**