How do map(), filter() & reduce() work in JavaScript ?

@NamasteDevOfficial

0. Array.prototype.map()

- It takes 2 arguments, a callback, and an optional context (will be considered as this in the callback)
- Creates a new array populated with the results of calling a provided function on every element in the calling array.
- The resulting array will always be the same length as the original array.

```
> var series = [
    { id: 1, name: 'Sherlock Holmes' },
    { id: 2, name: 'F.R.I.E.N.D.S' },
    { id: 3, name: 'Harry Potter' },
    { id: 4, name: 'Suits' }
  ];
  var seriesName = series.map(function (series) {
    return series.name
  });
  console.log(seriesName);

  ▼(4) ["Sherlock Holmes", "F.R.I.E.N.D.S", "Harry Potter", "Suits"]
     0: "Sherlock Holmes"
     1: "F.R.I.E.N.D.S"
     2: "Harry Potter"
     3: "Suits"
     length: 4
   ▶ __proto__: Array(0)
```

1. Array.prototype.reduce()

- Just like .map(), .reduce() also runs a callback for each element of an array resulting in single output value

```
const array3 = [0,1, 2, 3, 4];
const reducer3= (accumulator, currentValue) => accumulator + currentValue;

// 1 + 2 + 3 + 4
console.log(array3.reduce(reducer3));
// expected output: 10

// 5 + 1 + 2 + 3 + 4
console.log(array3.reduce(reducer3, 5));
// expected output: 15
10
15
```

- **A Secret**:
  Using .reduce() is an easy way to generate a single value or object from an array.

2. Array.prototype.filter()

- What if you have an array, but only want some of the elements in it?
  That's where .filter() comes in!

- The filter() method creates a new array with all elements that pass the test implemented by the provided function.

- Array elements that do not pass the callback test are skipped and are not included in the new array

```
const words = ['spray', 'limit', 'elite', 'exuberant', 'destruction', 'present'];

const result = words.filter(word => word.length > 6);

console.log(result);
// expected output: Array ["exuberant", "destruction", "present"]
```

```
▼(3) ["exuberant", "destruction", "present"]
   0: "exuberant"
   1: "destruction"
   2: "present"
   length: 3
 ▶ __proto__: Array(0)
```

Secrets

- Using Array.filter(), then Array.map() traverses the array twice. **But you can achieve the same effect while traversing only once with Array.reduce()**, thereby being more efficient. Try and find this out.

- Try replacing some of your for loops with .map(), .reduce(), .filter() where it seems to fit.
  You will see that your code will be way less clumsy and with much better readability.

- None of the above-mentioned methods mutate the original array.

Browser Support:

The numbers in the table specify the first browser version that fully supports the method.

Method					
map()	Yes	9.0	1.5	Yes	Yes
reduce()	Yes	9.0	3.0	4	10.5
filter()	Yes	9.0	1.5	Yes	Yes

FOLLOW

@NamasteDevOfficial

on Instagram

for more amazing updates!

DOUBLE TAP to show your ❤️

Mention your feedback in the comments

@NamasteDevOfficial