Title: How does map method work on JavaScript Arrays?

Answer:

The `map()` method creates a new array with the results of calling a provided function on every element in the calling array.

```
var array1 = [1, 4, 9, 16];

// pass a function to map
const map1 = array1.map(function(x) {
 return x * 2
});

console.log(map1);
// expected output: Array [2, 8, 18, 32]
```

`map` calls a provided `callback` function **once for each element** in an array, in order, and constructs a new array from the results. `callback` is invoked only for indexes of the array which have assigned values, including undefined.

Examples

Mapping an array of numbers to an array of square roots

The following code takes an array of numbers and creates a new array containing the square roots of the numbers in the first array.

```
var numbers = [1, 4, 9];
var roots = numbers.map(Math.sqrt);
// roots is now [1, 2, 3]
// numbers is still [1, 4, 9]
```

Mapping an array of numbers using a function containing an argument

The following code shows how map works when a function requiring one argument is used with it. The argument will automatically be assigned from each element of the array as map loops through the original array.

```
var numbers = [1, 4, 9];
var doubles = numbers.map(function(num) {
  return num * 2;
});

// doubles is now [2, 8, 18]
// numbers is still [1, 4, 9]
```

Tags: javascript, arrays