map

Download Demo Code

Goals

- Understand what map does
- Write your own version of map

map

- Creates a new array
- Iterates through an array
- Runs a callback function for each value in the array
- Adds the result of that callback function to the new array
- Returns the new array
- map always returns a new array of the same length

An Example

```
let numbers = [1,2,3];

numbers.map(function(value, index, array){
  return value * 10;
});

// [10,20,30]
```

How Does It Work?

```
function map(array, callback){
  let newArray = [];
  for(let i = 0; i < array.length; i++){
    newArray.push(callback(array[i], i, array));
  }
  return newArray;
}
```

- Creates a new array
- Loops through an array
- Runs a callback function for each value in the array
- Pushes result of the callback function to the new array
- Returns the new array

Using map In A Function

```
function squareValues(array){
  return array.map(function(value){
    return value ** 2;
  });
}

squareValues([2,3,4]) // [4,9,16]
function extractCourse(array){
  return array.map(function(value){
    return value.course;
  });
}

extractCourse([
  { author: "Billy Banks", course: "Tai Bo" },
  { author: "Colt Steele", course: "JavaScript 101" },
  { author: "Gordon Ramsey", course: "Cooking and Yelling" },
])

// ["Tai Bo", "JavaScript 101", "Cooking and Yelling"]
```

When You Would Use Map

- You want to “transform” an array into another array of the same length
- You do not want to overwrite an existing array and instead return a new copy

Recap

- map creates a new array
- map runs a callback on each value and pushes the result of each callback in the new array