Array Methods

A JavaScript reference of Array methods returning new array/value rather than modifying existing array 😀

a. **map**
b. **filter**
c. **flat**
d. **slice**
e. **join**
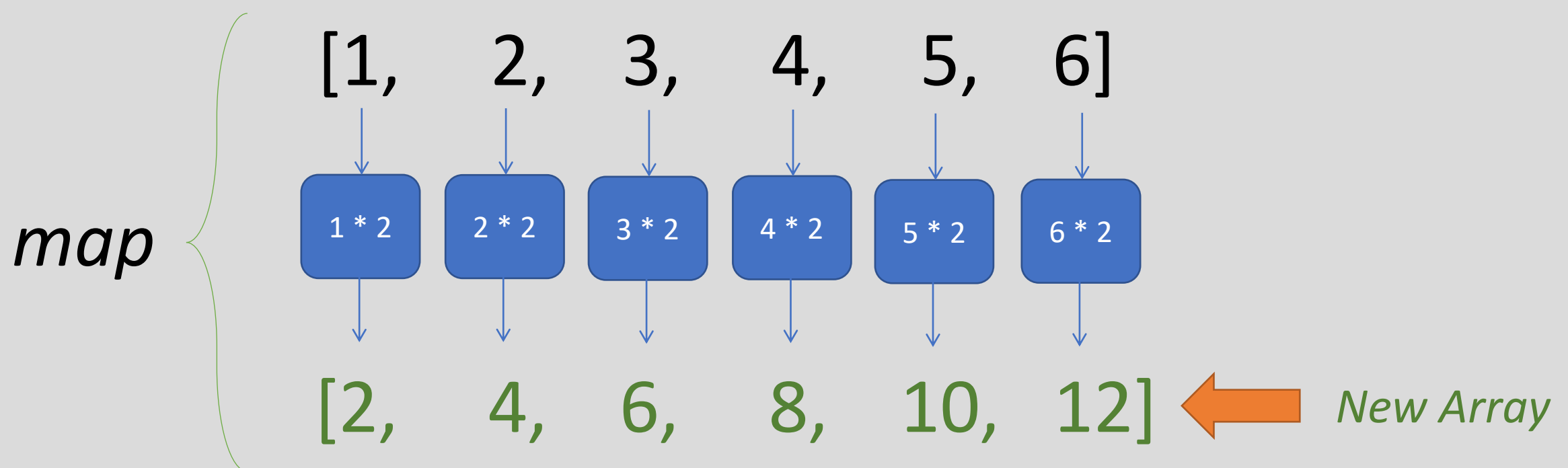f. **reduce**

map

map - returns a new array by iterating each element and modifying them based on a provided function.

```javascript
const numbers = [1, 2, 3, 4, 5, 6];

const result = numbers.map(n => n * 2);

// result = [2, 4, 6, 8, 10, 12]
```

[1,   2,   3,   4,   5,   6]

map

| 1 * 2 | 2 * 2 | 3 * 2 | 4 * 2 | 5 * 2 | 6 * 2 |

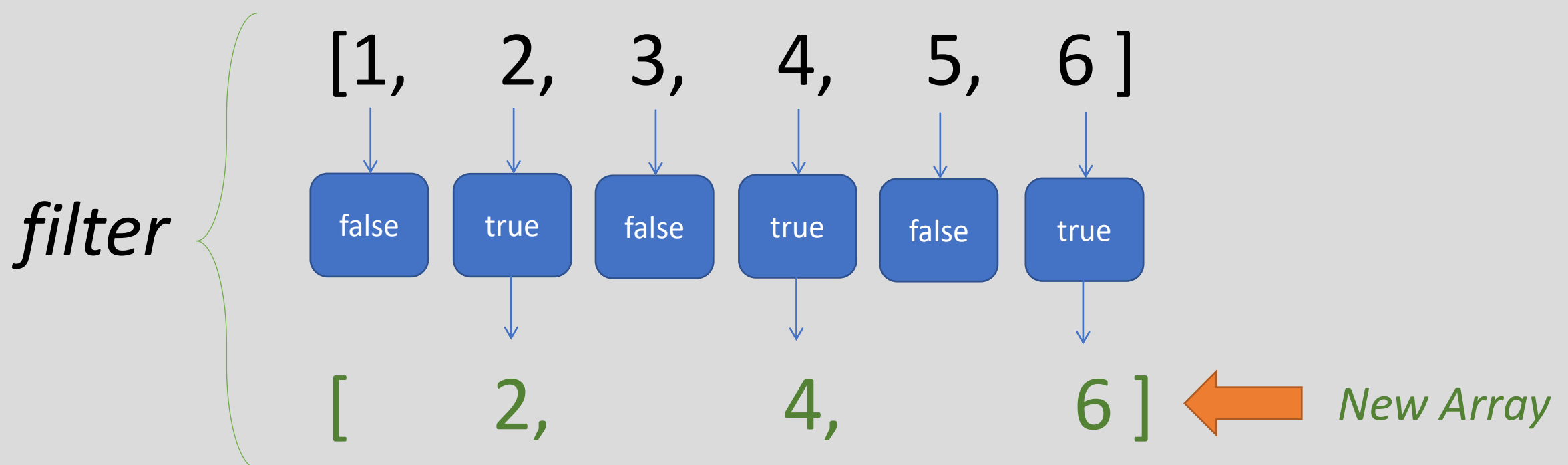[2,   4,   6,   8,   10, 12] ← *New Array*

filter

filter - returns a new array containing elements for which filtering function returns *true*

```javascript
const numbers = [1, 2, 3, 4, 5, 6];

const filtered = numbers.filter(n => n % 2 === 0);

// filtered = [2, 4, 6]
```

[1,   2,   3,   4,   5,   6]

filter {

| false | true | false | true | false | true |

[2,   4,   6] ⬅ *New Array*

flat

flat – returns new array by moving nested ones level higher. You can also specify the nested depth.

```javascript
const numbers = [1, 2, 3, 4, 5, 6];

const flatNumbers = numbers.flat();

// flatNumbers = [ 1, 2, 3, 4, 5, 6 ]
```

flat

[1, 2, 3, 4, [5, 6]]

[1, 2, 3, 4, 5, 6] ⬅ *New Array*
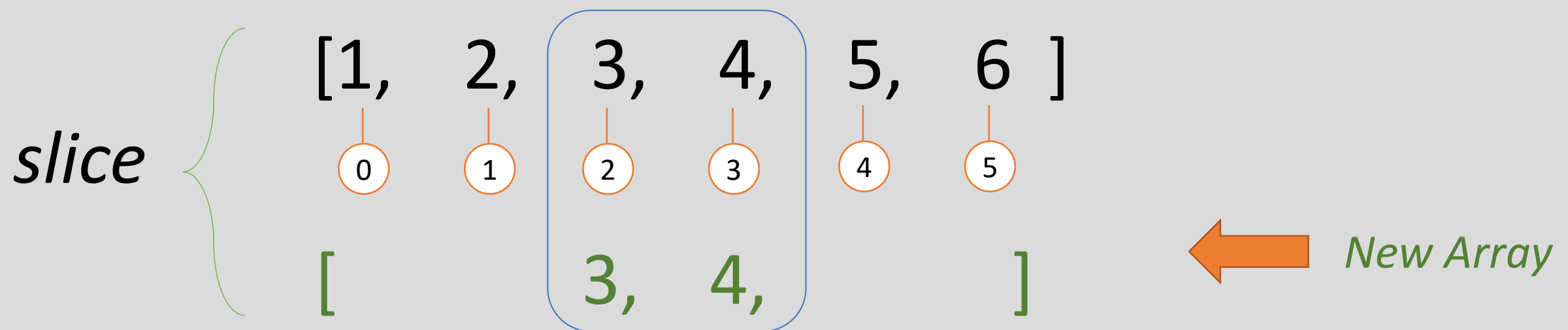
Follow **me for more tips**

slice

slice – returns a new array based on provided range, the start index is included but end index is not.

```javascript
const numbers = [1, 2, 3, 4, 5, 6];

const sliced = numbers.slice(2,4);

// sliced = [ 3, 4 ]
```

slice
```
[1,  2,  3,  4,  5,  6 ]
   0    1    2    3    4    5

[        3,  4,          ]          ← New Array
```

Follow me for more tips

join

join – returns a new string by concatenating elements with provided separator.

```js
const numbers = [1, 2, 3, 4, 5, 6];

const joined = numbers.join("-");

// joined = 1-2-3-4-5-6
```

[1, 2, 3, 4, 5, 6]

join

1-2-3-4-5-6 ⬅ *New string*

Follow me for more tips

reduce

reduce – returns the value of accumulated
function result.
Note: You can also specify initial value.

```js
const numbers = [1, 2, 3, 4, 5, 6];

const sum = numbers.reduce((prev,next) => prev + next);

// sum = 21
```

	[1, 2,	3,	4,	5,	6]
reduce	1+2 → 3 + 3 → 6 + 4 → 10 + 5 → 15 + 6				

21 ⬅ *New value*

JS

To know more refer to my array search methods post link in description

Follow me for more tips