**Массивы - map**

Функция map() проходит по элементам массива и возвращает обработанное значение каждого элемента.

<имя массива>.map(Callback(<текущий элемент массива>))

Рассмотрим на примере

const array = [2, 4, 6]

const newArr = array.map((item) => item \* 2)

console.log(newArr) //в консоли отобразится массив [4, 8, 12]

**item** - это просто имя аргумента, оно может быть любым!

const newArr = array.map(elem => elem \* 2)

const newArr = array.map(x => x \* 2)

const newArr = array.map(smth => smth \* 2)

Думаю, прицнип ясен.

В примере выше мы брали каждый элемент массива и умножали его на 2, а результат складывали в переменную newArr.

В лекции используется чуть более сложный пример, но принцип работы тот же.

В качестве callback-функции мы можем использовать стороннюю функцию.

const array = [2, 4, 6]

const func = (item) => item \* 2

const newArr = array.map(func)

console.log(newArr) //в консоли увидим [4,8,12]

У callback-функции внутри map может быть 3 аргумента:

* item - текущий элемент массива
* index - индекс текущего элемента
* array - массив

Из функций map можно строить цепочки, чтобы произвести несколько последовательных преобразований элементов массива.

const array = [2, 4, 6]

const newArr = array.map((item) => item \* 2).map((item) => item + 10)

console.log(newArr) //в консоли увидим [14,18,22]