

Evgen959 / JavaScript Public

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#)

master JavaScript / Lesen 014 / code / example_01 / index.js



Evgen959 new

now



79 lines (69 loc) · 2.82 KB

Code

Blame

Raw



```
1 // forEach
2 // const ff = (a, b, c) => {    // a - элемент массива b - индекс элемента
3 // c - весь массив
4 //     console.log(`${a} + ${b} + ${c}`);
5 // }
6 const arrF = [1, 2, 3, 5, 4, 10, 6];
7 console.log(arrF);
8 // let b =0;
9 // const ff = (a) => {
10 //     b = b + a;
11 // }
12
13 let arrayA = [];
14 const ff = (a) => {
15     arrayA.push(a + 3); // пушим в пустой массив
16 }
17
18 arrF.forEach(ff); // запускает функцию для каждого элемента массива столько раз сколько
19 // элементов массива
20
21 // map !!!
22 const callbackForMap = (element) => {
23     return element + 4;
24 }
25 const arrayAMap = arrF.map(callbackForMap); // возвращает новый переделанный массив
26 // console.log(arrayAMap);
27
28 const callbackForMapA = (element) => {
29     return element * element;
30 }
31 const arrayBMap = arrF.map(callbackForMapA); // возвращает новый переделанный массив
32 // console.log(arrayBMap);
33
34 // filter
35 const callbackForFilter = (element) => {
36     // if (element > 4) {
```

```
37     //     return true;
38     // } else {
39     //     return false;
40     // }
41     return element > 4 ? true: false;
42 }
43 const arrayAFilter = arrF.filter(callbackForFilter); // возвращает новый массив
44 // отфильтрованный по заданным параметрам
45 console.log(arrayAFilter);
46
47 // slice
48 const arrayASliceResult = arrF.slice(0, 2); // возвращает новый массив, обрезает массив
49 // до указанного диапазона (указывающий элемент на конец диапазона не входит в него)
50 // console.log(arrayASliceResult);
51
52 // sort
53 ✓ const callBackForSort = (a, b) => {
54     // if (a < b) {
55     //     return -1;
56     // } else {
57     //     return 1;
58     // }
59
60     // return a > b ? -1: 1;
61
62     return b - a
63 }
64 const arrFcopy = [...arrF]; // делаем копию!!!! массива
65 console.log(arrFcopy.sort(callBackForSort)); // сортирует массив и !!!!!!!возвращает
66 // исходный!!!!!! массив уже отсортированный
67
68 // классная работа: отсортировать массив из игроков по росту "height"
69 const callBackForSortC = (object1, object2) => {
70     return object1.height < object2.height ? -1: 1;
71 }
72 const arrayC = [
73     {"name": "John", "surname": "Doe", "height": 198, "effectiveness": 15.6},
74     {"name": "Jane", "surname": "Smith", "height": 183, "effectiveness": 12.3},
75     {"name": "Mike", "surname": "Brown", "height": 202, "effectiveness": 18.4},
76     {"name": "Emily", "surname": "Davis", "height": 190, "effectiveness": 14.8},
77     {"name": "Chris", "surname": "Wilson", "height": 205, "effectiveness": 20.1}
78 ];
79 console.log(arrayC.sort(callBackForSortC));
```