



JavaScript

Урок №13: Основы. Функции высшего порядка.

Автор Бахшиллоев Бехзод

Ментор по Frontend в **PRO** UNITY



Функции высшего порядка.

Функция высшего порядка — функция, принимающая в качестве аргументов другие функции или возвращающая другую функцию в качестве результата.

Функция принимает другую функцию в качестве аргумента:

```
function highFunction( func ) {  
  func();  
}
```

```
highFunction( function () { console.log(123) } ) // 123
```

Функция возвращает другую функцию в качестве результата:

```
function anotherHighFunction() {  
  return function () { return 123 }  
}
```

В этом случае anotherHighFunction вернет другую функцию:

```
anotherHighFunction() → function () { return 123 }
```

Так как возвращаемый результат - функция, её тоже можно вызывать:

```
anotherHighFunction()() → ( function () { return 123 } )()
```



Функции высшего порядка.

```
function processString(string, handler) {  
  let newString = "";  
  
  for (let i = 0; i < string.length; i++) {  
    newString += handler(string[i]);  
  }  
  
  return newString;  
}
```

```
processString('Hello, easycodel',  
function (symbol) {  
  return symbol.toUpperCase();  
});  
processString('Hello, easycodel',  
function (symbol) {  
  return symbol + '_';  
});  
processString('Hello, easycodel',  
function (symbol) {  
  return symbol.charCodeAt();  
});
```



Задачи:

1. Создать две функции и дать им осмысленные названия:
 - первая функция принимает массив и колбэк (**одна для всех вызовов**)
 - вторая функция (колбэк) обрабатывает каждый элемент массива (**для каждого вызова свой callback**)

Первая функция возвращает строку "New value: " и результат обработки:

firstFunc(['my', 'name', 'is', 'Trinity'], **handler1**) → "New value: MyNameIsTrinity"

firstFunc([10, 20, 30], **handler2**) → "New value: 100, 200, 300, "

firstFunc([{age: 45, name: 'John'}, {age: 20, name: 'Aaron'}], **handler3**) →
"New value: John is 45, Aaron is 20,"

firstFunc(['abc', '123'], **handler4**) → "New value: cba, 321," // строки инвертируются

Подсказка: **secondFunc** должна быть представлена функцией, которая принимает один аргумент (каждый элемент массива) и возвращает результат его обработки



Задачи:

2. Написать аналог метода **every**. Создайте функцию **every**, она должна принимать первым аргументом массив чисел (**обязательно проверьте что передан массив**) вторым аргументом **callback** (**обязательно проверьте что передана функция**) функция должна возвращать **true** или **false** в зависимости от результата вызова **callback** (**проверить число больше 5**). **Callback** должен принимать один элемент массива, его индекс в массиве и весь массив.

Что такое метод **every** можно прочитать [здесь](#) и ниже в презентации тоже о них идет речь.