

ПАРАМЕТРЫ И ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ



НИКОЛАЙ ЛОПИН



НИКОЛАЙ ЛОПИН

Senior JavaScript Developer в MyTaxi



lopinopulos@gmail.com



[nikolay.lopin](#)



[@nlopin](#)

ПЛАН ЗАНЯТИЯ

1. Повторение – мать учения
2. Аргументы функции
3. Функция как подпрограмма возвращающая значение
4. Больше информации!

ПОВТОРЕНИЕ – МАТЬ УЧЕНИЯ

ВСПОМНИМ ПРОШЛЫЙ МАТЕРИАЛ

```
1 let name = "Petro";  
2  
3 function sayMyName() {  
4     console.log(name);  
5 }  
6  
7 sayMyName();
```

Что выведет код?

OTBET

Petro

ВЕРНЕМСЯ К НАСТОЯЩЕМУ

В нашей программе часто нужно вычислять среднее значение массива чисел.

```
1 let numbers = [12, 14, 14];
2
3 let sum = 0;
4 let count = 0;
5
6 for (let index = 0; index < numbers.length; index++) {
7     count += 1;
8     sum += numbers[index];
9 }
10
11 console.log(sum / count);
```

ЕСТЬ ПРОБЛЕМА

Нехорошо, когда мы повторяем такой простой код сотни раз.

Мы можем вынести этот код в функцию! Но нужно как-то передать ей массив с числами.

Для этого используются **аргументы** функций. Это такой способ явно передать данные в функцию.

РЕШЕНИЕ

```
1 function average(numbers) {  
2     let sum = 0;  
3     let count = 0;  
4     for (let index = 0; index < numbers.length; index++) {  
5         count += 1;  
6         sum += numbers[index];  
7     }  
8  
9     console.log(sum / count);  
10 }  
11  
12 average([12, 14, 14]);  
13  
14 let array = [1, 2, 3, 4, 5];  
15 average(array);
```

Что выведет этот код?

ОТВЕТ

То же самое!

13 , 33333333

Мы передали массив чисел как **аргумент**.

АРГУМЕНТЫ

Аргументов может быть сколько угодно!

```
1 function sayHello(firstName, lastName) {  
2     console.log('Hello, ' + firstName + ' ' + lastName);  
3 }  
4  
5 sayHello('Petro', 'Egorov');
```

ЧТО, ЕСЛИ НЕ ПЕРЕДАТЬ АРГУМЕНТЫ?

```
1 function sayHello(firstName, lastName) {  
2   console.log('Hello, ' + firstName + ' ' + lastName);  
3 }  
4  
5 sayHello('Petro');
```

Что выведет этот код?

ОТВЕТ

```
Hello, Petro undefined
```

Если не передать аргумент, то ему будет присвоено значение `undefined`.

ЗНАЧЕНИЕ ПО УМОЛЧАНИЮ

```
1 function sayHello(firstName, lastName = 'Surname') {  
2   console.log('Hello, ' + firstName + ' ' + lastName);  
3 }  
4  
5 sayHello('Petro');
```

Hello, Petro Surname

ВОПРОС!

Какие применения для функций с аргументами вы можете найти прямо сейчас?

МЫ ХОТИМ ОТ ФУНКЦИИ БОЛЬШЕЙ УНИВЕРСАЛЬНОСТИ!

После вычисления среднего значения, его не всегда нужно выводить в консоль. Иногда хочется записать в файл, иногда вывести в `alert`.



Что делать?

ПРОСТОЕ РЕШЕНИЕ

```
1 function averageInConsole(numbers) {  
2     // ...  
3     console.log(sum / count);  
4 }  
5  
6 function averageInAlert(numbers) {  
7     // ...  
8     alert(sum / count);  
9 }
```

КОПИРОВАНИЕ КОДА...

Для этого случая подходит **возвращаемое значение**.

Функция может **вернуть** какое-то значение тому коду, который ее вызвал.

```
1 function average(numbers) {  
2     let sum = 0;  
3     let count = 0;  
4     for (let index = 0; index < numbers.length; index++) {  
5         count += 1;  
6         sum += numbers[index];  
7     }  
8  
9     return sum / count;  
10 }  
11  
12 let avg = average([12, 14, 14]);  
13 console.log(avg);
```

ЭТО ОБЫЧНОЕ ЗНАЧЕНИЕ!

С возвращенным значением мы можем делать уже что угодно.

```
1 let avgForConsole = average([12, 14, 14]);
2 console.log(avgForConsole);
3
4 let avgForAlert = average([120, 1, 543]);
5 alert(avgForAlert);
6
7 let alertForFile = average([54, 32, 12]);
8 writeFile(alertForFile);
```

ЗАВЕРШЕНИЕ

`return` завершает работу функции.

```
1 | function average(numbers) {
2 |   let sum = 0;
3 |   let count = 0;
4 |   for (let index = 0; index < numbers.length; index++) {
5 |     count += 1;
6 |     sum += numbers[index];
7 |
8 |
9 |   if (count === 0) {
10 |     return;
11 |   }
12 |
13 |   return sum / count;
14 | }
```

Как только встретился `return`, функция прекращает выполняться.

ВОПРОС

Сколько значений можно вернуть из функции?

ОТВЕТ

Технически – одно.

```
1 function average(numbers) {  
2     // ...  
3  
4     return sum / count;  
5 }
```

НО ХОЧЕТСЯ НЕСКОЛЬКО...

Тогда можно вернуть массив или объект.

```
1 function average(numbers) {  
2     // ...  
3  
4     let avg = sum / count;  
5  
6     return { avg: avg, sum: sum, total: count };  
7 }
```

ВОЗВРАЩАЕМЫЕ ЗНАЧЕНИЯ

Какие еще применения функциям с возвращаемыми значениями вы можете найти?

ИТАК!

**ФУНКЦИЯМ МОЖНО
ПЕРЕДАВАТЬ ДАННЫЕ
ЧЕРЕЗ АРГУМЕНТЫ.**

**ФУНКЦИИ МОГУТ
ВОЗВРАЩАТЬ ДАННЫЕ**

ФУНКЦИИ МОГУТ ВОЗВРАЩАТЬ ДАННЫЕ

1. Узнали, как передавать данные в функции;
2. Узнали, как получать данные из функций.

РАДУЕМСЯ!

ДОМАШНЕЕ ЗАДАНИЕ

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задаем в группе Facebook!
- Задачи можно сдавать по частям.
- Зачет по домашней работе проставляется после того, как приняты все **3 задачи**.

ЧТО ПОЧИТАТЬ

- <https://learn.javascript.ru/function-basics>
- <http://javascript.ru/basic/functions>
- <https://habr.com/post/261723/>
- https://www.youtube.com/watch?v=e-5obm1G_FY



Спасибо за внимание! Время задавать вопросы

НИКОЛАЙ ЛОПИН



lopinopulos@gmail.com



[nikolay.lopin](#)



[@nlopin](#)