

ФУНКЦИИ БЕЗ ПАРАМЕТРОВ. ОБЛАСТЬ ВИДИМОСТИ



НИКОЛАЙ ЛОПИН



НИКОЛАЙ ЛОПИН

Senior JavaScript Developer в MyTaxi



lopinopulos@gmail.com



[nikolay.lopin](#)



[@nlopin](#)

ПЛАН ЗАНЯТИЯ

1. Повторение – мать учения
2. Подпрограмма
3. Функция? Функция
4. Зачем? Как?
5. Область видимости

ПОВТОРЕНИЕ – МАТЬ УЧЕНИЯ

ВСПОМНИМ ПРОШЛОЕ

```
1 const locker = {  
2     undies: ['pants1', 'pants2'],  
3     bras: ['bra1', 'bra2'],  
4     isLocked: false  
5 };  
6  
7 console.log(locker.bras);  
8 console.log(locker.undies[0]);
```

Что выведет код?

OTBET

```
["bra1", "bra2"]  
pants1
```

ВЕРНЕМСЯ К НАСТОЯЩЕМУ

Есть код:

```
1 let length = 12;  
2  
3 if (length > 10) {  
4     console.log('Wow!');  
5     console.log('So large!');  
6 }  
7  
8 if (length > 11) {  
9     console.log('Wow!');  
10    console.log('So large!');  
11 }
```

ЕСТЬ ПРОБЛЕМА

Теперь заказчик хочет так:

```
1 let length = 12;  
2  
3 if (length > 10) {  
4     alert('Wow!');  
5     alert('So large!');  
6 }  
7  
8 if (length > 11) {  
9     alert('Wow!');  
10    alert('So large!');  
11 }
```

Представьте, для того чтобы задернуть шторы, нужно просто перевесить каждый крючок каждой шторы.

РЕШЕНИЕ

Нужно вынести куда-то этот код.

```
1 | console.log('Wow!');  
2 | console.log('So large!');
```

Тогда можно заменить его в одном месте и не грустить.

ФУНКЦИИ

В JavaScript есть функции. Это вынесенные кусочки кода, которые можно использовать повторно.

Функцию сначала объявляют. Назвать функцию можно как угодно:

```
1 | function functionName() {  
2 |     // code  
3 | }
```

А затем вызывают по имени:

```
1 | functionName();
```

При вызове происходит выполнение *тела функции* – блока кода внутри фигурных скобок.

ПРИМЕР ФУНКЦИИ

```
1 let length = 12;  
2  
3 function sayLarge() {  
4     console.log('Wow!');  
5     console.log('So large!');  
6 }  
7  
8 if (length > 10) {  
9     sayLarge();  
10 }  
11  
12 if (length > 11) {  
13     sayLarge();  
14 }
```

ФУНКЦИЯ

Теперь, чтобы заменить `console.log` на `alert` достаточно сделать это только внутри функции.

```
1 let length = 12;  
2  
3 function sayLarge() {  
4     alert('Wow!');  
5     alert('So large!');  
6 }  
7  
8 if (length > 10) {  
9     sayLarge();  
10 }  
11  
12 if (length > 11) {  
13     sayLarge();  
14 }
```

ЛЕГКО И УДОБНО!

```
1 function sayMyName() {  
2   console.log('Heisenberg');  
3 }  
4  
5 sayMyName();  
6 sayMyName();  
7 sayMyName();
```

Выведет в консоль **Heisenberg** три раза.

ВОПРОС!



Для каких действий могут использоваться функции?

НОВОЕ ЗАДАНИЕ ОТ ЗАКАЗЧИКА...

Нужно выводить три сообщения, последнее должно иметь вид “15 meters”.

ЕСТЬ ИДЕИ?

ЕСТЬ ИДЕИ?

Нужно как-то использовать переменную `length`.

```
1 let length = 12;
2
3 function sayLarge() {
4     console.log('Wow!');
5     console.log('So large!');
6     console.log(length + ' meters');
7 }
8
9 if (length > 10) {
10    sayLarge();
11 }
12
13 if (length > 11) {
14    sayLarge();
15 }
```

А ЧТО, ЕСЛИ ПЕРЕМЕННАЯ ОПРЕДЕЛЕНА ВНУТРИ ДРУГОЙ ФУНКЦИИ?

```
1 function sayLarge() {
2     console.log('Wow!');
3     console.log('So large!');
4     console.log(len + ' meters');
5 }
6
7 function checkLarge() {
8     let len = 12;
9
10    if (len > 10) {
11        sayLarge();
12    }
13
14    if (len > 11) {
15        sayLarge();
16    }
17 }
18
19 checkLarge();
```

ОШИБКА!

```
ReferenceError: len is not defined
```

Доступа к переменной нет, и программа ломается. Это не то что нам нужно!

ОБЛАСТЬ ВИДИМОСТИ

У переменной есть **область видимости**. Это тот код, который видит переменную и может получить к ней доступ.

Определяется блоком кода вокруг объявления переменной переменной. Блоки кода обозначаются фигурными скобками.

ЧТО БУДЕТ В КОНСОЛИ?

```
1 function sayMyName() {  
2     let name = 'Heisenberg';  
3     console.log(name);  
4 }  
5  
6 let age = 53;  
7 if (age > 50) {  
8     let message = 'Okay';  
9     console.log(message);  
10  
11     sayMyName();  
12 }  
13  
14 console.log(name);  
15 console.log(message);
```

ВЫВОД, ПУСТАЯ СТРОКА И ОШИБКА

```
Okay
```

```
Heisenberg
```

```
Uncaught ReferenceError: message is not defined  
at :15:13
```

ОБЛАСТЬ ВИДИМОСТИ

Переменная доступна внутри тех фигурных скобок, где она объявлена.

**МОЖНО ЕЕ ИЗМЕНЯТЬ ИЗ
«ДОЧЕРНИХ» БЛОКОВ
КОДА?**

МОЖНО

```
1 let age = 25;  
2  
3 function birthday() {  
4     age = age + 1;  
5     console.log(age);  
6 }  
7  
8 birthday();  
9 birthday();  
10 birthday();
```

РЕЗУЛЬТАТ

26

27

28

А ЧТО ЕСЛИ ПЕРЕЗАПИСАТЬ ЕЕ?

```
1 let age = 25;  
2  
3 function birthday() {  
4     let age = 23;  
5     console.log(age);  
6 }  
7  
8 birthday();  
9 birthday();  
10 birthday();
```

ПЕРЕЗАПИШЕТСЯ

23

23

23

Лучше так не делать. Это снижает читаемость кода!

И ТАК!

**ФУНКЦИИ –
ИМЕНОВАННЫЕ БЛОКИ
КОДА.**

**ОБЛАСТЬ ВИДИМОСТИ –
КОД КОТОРЫЙ ИМЕЕТ
ДОСТУП ДО
ПЕРЕМЕННОЙ.**

ЧЕМУ МЫ НАУЧИЛИСЬ?

1. Узнали, что такое подпрограммы и как их реализовать в JS;
2. Узнали, к каким переменным есть доступ у функции;
3. Попробовали переопределить переменную из другой области видимости.

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>
- <https://habr.com/post/240349/>



Спасибо за внимание! Время задавать вопросы

НИКОЛАЙ ЛОПИН



lopinopulos@gmail.com



[nikolay.lopin](#)



[@nlopin](#)