



# Лекция 3

Функции

## Цели на сегодня

- Начать использовать строгий режим;
- Разобраться с синтаксисом функций;
- Стрелочные функции;
- Функциональные выражения;
- Область видимости;
- IIFE;

# Типы данных

## 1. Прimitives

- a. Числа (Numbers);
- b. Строки (Strings);
- c. Логические (Boolean)
- d. Символ (Symbol);
- e. null;
- f. undefined.

## 2. Сложные

- a. Объекты (Objects);
- b. Массивы (Arrays);
- c. Функции (Functions).

## **Функции (Functions) --**

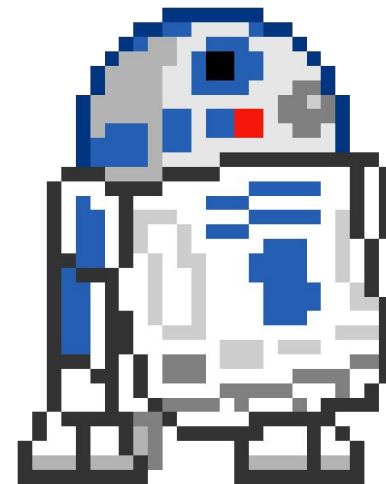
это лучший способ повторять один и тот же код в разных частях программы.

Мы имеем какие-то данные и хотим произвести над ними действия.  
Мы знаем, что эти действия придется выполнять часто, поэтому  
создаем для этого помощника.

Для начала мы научим его выполнять простые  
расчеты, постепенно усложняя его ПО.

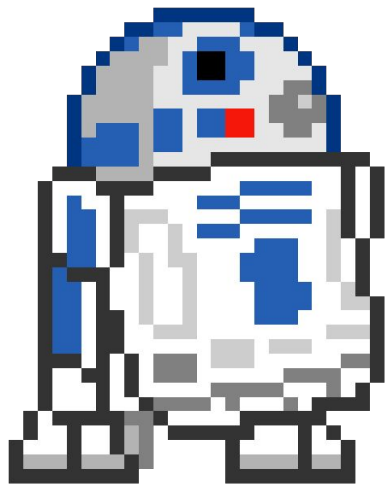
Пусть сегодня он просто будет рассчитывать  
площадь поверхности сферы по формуле  **$S=4\pi R^2$** .

В домашнем задании к четвертому занятию вы  
научите его большому количеству полезностей.



**R2D2()**

Указываем  
радиус



Получаем  
результат

# Синтаксис функций

```
//home/knyazevich/Courses/ec-develop/func/common.js - Sublime Text (UNREGISTERED)
common.js x
1 // Функцию можно объявить таким образом:
2
3 function имя_функции(параметры) {
4     тело_функции
5 }
6
7 /*
8     При этом, после объявления функция не
9     выполнится, а просто запишется в память.
10
11     Вызывается функция таким образом:
12 */
13
14 имя_функции(параметры); // Какой-то результат
15 имя_функции(другие_параметры); // Другой результат
16
```

Line 8, Column 31 UTF-8 Tab Size: 4 JavaScript



```
//home/knyazevich/Courses/ec-develop/func/common.js - Sublime Text (UNREGISTERED)
common.js x
1  /*
2     Функции -- конструкции, которые позволяют нам
3     переиспользовать код, повторять одно и то
4     же действие в разных частях программы.
5  */
6
7  function addThree(number) {
8      return number + 3;
9  }
10
11 // Вызов функции происходит так:
12
13 addThree(10); // 13
14 addThree(5); // 8
15 addThree(100); // 103
16
```

Line 8, Column 23      UTF-8      Tab Size: 4      JavaScript

**Параметры (аргументы)**

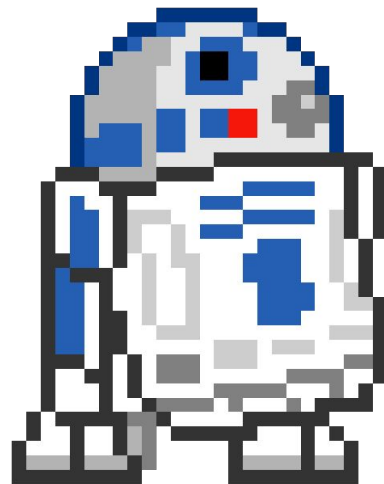
```
//home/knyazevich/Courses/ec-develop/func/common.js - Sublime Text (UNREGISTERED)
common.js x
1  /*
2     При вызове функции ей можно передать данные,
3     которые та использует по своему усмотрению.
4     Параметры записываются во внутренние переменные
5     функции.
6  */
7
8  function greeting(name, text) {
9      const greetText = `
10         Hello, ${name}, there is a message for you:
11         ${text}.
12     `;
13
14     console.log(greetText);
15 }
16
17 greeting('Masha', 'You are so cute :3');
```

Line 17, Column 41

UTF-8 Tab Size: 4 JavaScript

## Самостоятельная работа

1. Напишите функцию `square(x)`, которая считает квадрат переданного из prompt числа, и возвращает в консоль значение "Квадрат числа **x** равен **y**".
2. Если ничего не передано, то возвращает в консоль значение "Число не передано".
3. Если NaN, то возвращает "Значение **x** не является числом".



**R2D2()**

```
//home/knyazevich/Courses/ec-develop/func/common.js - Sublime Text (UNREGISTERED)
common.js x
1  /*
2     Параметров можно передавать больше или меньше,
3     чем необходимо. При этом никакой ошибки не будет.
4  */
5
6  function greeting(name, text, login) {
7      const greetText = `
8          Hello, ${name}, there is a message for you:
9          ${text}.
10         Your login is ${login}.
11     `;
12
13     console.log(greetText);
14 }
15
16 greeting('Masha', 'You are so cute :3');
17 // ...Your login is undefined.
```

Line 17, Column 31      UTF-8      Tab Size: 4      JavaScript

# ES6

```
~/Courses/ec-develop/func/common.js - Sublime Text (UNREGISTERED)
common.js x
1  /*
2     Если нам не нравится видеть в сообщении
3     'undefined', то мы можем задать значение по
4     умолчанию.
5  */
6
7  function greeting(name, text, login = 'user') {
8      const greetText = `
9          Hello, ${name}, there is a message for you:
10         ${text}.
11         Your login is ${login}.
12         `;
13
14     console.log(greetText);
15 }
16
17 greeting('Masha', 'You are so cute :3');
18 // ...Your login is user.
```

Line 19, Column 1      UTF-8      Tab Size: 4      JavaScript

# ES6

```
~/Courses/ec-develop/func/common.js - Sublime Text (UNREGISTERED)
common.js x
1  /*
2     Параметры функции можно получить при
3     помощи оператора ...someName (rest)
4
5     Через несколько занятий вы научитесь удобно с
6     ними работать
7  */
8
9  const makeSomeNoize = (...args) => {
10     alert(args);
11     return args;
12 }
13
14 makeSomeNoize('Test', 'Some sound', 23, true);
15 // ["Test", "Some sound", 23, true]
```

Line 15, Column 36 UTF-8 Tab Size: 4 JavaScript

**Возврат значения (return)**



## **return --**

используется для того, чтобы функция возвращала значение.

Функции могут иметь директиву return, а могут и не иметь. Если return не указан или указан без значения, то возвращается undefined. При этом никакой ошибки не возникает.

~/Courses/ec-develop/func/common.js - Sublime Text (UNREGISTERED)

common.js

```
1 function square(num) {
2     num * num;
3 }
4 square(2); // undefined
5
6
7 function square(num) {
8     return num * num;
9 }
10 square(2); // 4
11
12
13 function square(num) {
14     console.log(num * num);
15 }
16 square(3); // undefined, но в консоли '9'
17
```

Line 16, Column 41

UTF-8 Tab Size: 4 JavaScript

```
//home/knyazevich/Courses/ec-develop/func/common.js - Sublime Text (UNREGISTERED)
common.js x
1  /*
2     return завершает выполнение функции.
3     Никакой код после этого не выполнится.
4  */
5
6  function add(a, b) {
7
8     const result = +a + +b;
9     return
10
11     // Код ниже никогда не выполнится
12
13     console.log('Test');
14     console.log('Why it is not working? :C');
15 }
16
17 add(2,10); // unreachable code after return statement
```

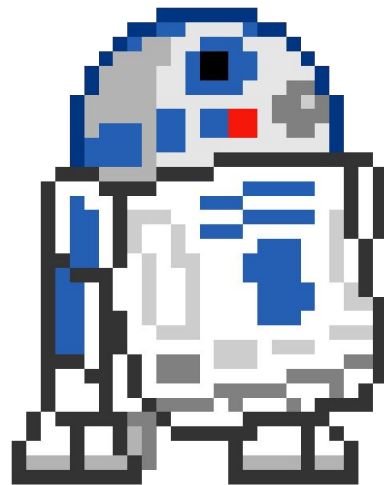
Line 17, Column 54      UTF-8      Tab Size: 4      JavaScript

```
~/Courses/ec-develop/func/common.js - Sublime Text (UNREGISTERED)
common.js x
1  /*
2     Если функция возвращает значение, его
3     можно записать в переменную
4  */
5
6  let result; // undefined
7
8  function square(number) {
9     return number * number;
10 }
11
12 result = square(2); // 4
13
```

Line 13, Column 1      UTF-8      Tab Size: 4      JavaScript

## Самостоятельная работа

1. Напишите функцию `R2D2()`, которая принимает в качестве параметра радиус сферы и возвращает значение в консоль. Формула:  
 **$S=4\pi R^2$** .
2. Напишите функцию `greetings()`, которая в зависимости от переданного значения (`user`, `admin` или ничего не передано) будет по-разному приветствовать пользователя.



**`R2D2()`**

# Стрелочные функции (Arrow functions)

# ES6

```
~/Courses/ec-develop/func/common.js - Sublime Text (UNREGISTERED)
common.js x
1 // Сейчас часто используется такой синтаксис:
2
3 // Раньше
4 const addTwo = function(num) {
5     return +num + 2;
6 }
7
8 // Сейчас
9 const addTwo = (num) => {
10     return +num + 2;
11 }
12
13 // Или еще компактнее
14 const addTwo = (num) => +num + 2;
15
```

Line 15, Column 1 UTF-8 Tab Size: 4 JavaScript

**Область видимости**



~/Courses/ec-develop/func/common.js - Sublime Text (UNREGISTERED)

common.js x

```
1 // Тут все переменные глобальные
2 // Тут не видно ничего из внутренних областей
3
4 const number = 25;
5
6 function test() {
7     // Тут все переменные локальные
8     const another = 10;
9     console.log(number);
10 }
11
12 test(); // 25
13 console.log(another); // another is not defined
14
```

Line 14, Column 1 UTF-8 Tab Size: 4 JavaScript

# **Немедленно вызываемые функции (IIFE)**

## **Немедленно вызываемая функция (IIFE) --**

это синтаксическая конструкция, позволяющая вызвать функцию сразу же в месте ее определения.

```
~/Courses/ec-develop/func/common.js - Sublime Text (UNREGISTERED)
common.js x
1 // Немедленно вызываемая функция выглядит так:
2
3 (function () {
4     // Тело функции
5 }());
6
7
8 // Или так:
9
10 (function () {
11     // Тело функции
12 }());
13
```

Line 13, Column 1 UTF-8 Tab Size: 4 JavaScript

# **Функциональные выражения**

```
~/Courses/ec-develop/func/common.js - Sublime Text (UNREGISTERED)
common.js x
1 // Объявление Function Declaration
2
3 function addThree(num) {
4     return num + 3
5 }
6
7 // Объявление Function Expression
8
9 const addThree = function(num) {
10     return num + 3
11 }
12
```

Line 11, Column 1      UTF-8      Tab Size: 4      JavaScript

```
~/Courses/ec-develop/func/common.js - Sublime Text (UNREGISTERED)
common.js x
1  /*
2     Функции, созданные с помощью FD
3     создаются сразу, поэтому их
4     можно вызывать раньше объявления
5  */
6
7  addThree(2); // 5
8
9  function addThree(num) {
10     console.log(num + three);
11 }
12
```

Line 11, Column 2

Такое поведение называют термином “поднятие” (hoisting)

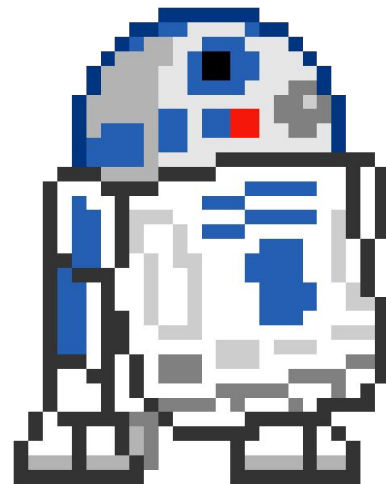
common.js

```
1  /*
2      Функции, созданные с помощью FE
3      нельзя вызвать до оглашения
4  */
5
6  addThree(2); // ReferenceError
7
8  const addThree = function(num) {
9      console.log(num + three);
10 }
11
12 addThree(2); // 5
13
```



## Самостоятельная работа

1. Напишите стрелочную самовызывающуюся функцию, которая принимает два числа и возвращает меньшее из них.



**R2D2()**