



Frontender[1.0] JavaScript - Functions. Коллбэки (callbacks), стрелочные функции, iife, call stack

YouTube	https://youtu.be/mZGJ-AjT_Xc
Telegram	https://t.me/Dmitry_Kolotilshikov
Github	https://github.com/DmitryKolotilshikov/
Boosty	https://boosty.to/dmitry_ko
# Номер урока	18

💡 Задачи к этому уроку тут https://boosty.to/dmitry_ko

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="script.js"></script>
  <title>Functions - callbacks, iife, arrow fn</title>
</head>
<body>
</body>
</html>
```

```
// Functions - callbacks, iife, arrow functions

/*
Аргумент функция-колбэк (call back).
Когда передается функция в виде аргумента, то ожидается, что она вызовется обратно
*/

const log = console.log;

function greeting() {
  log("Hello my friends!");
}
```

```

function showGreeting(callback) {
    callback();
}

showGreeting(greeting);

//-----

function getInfo(name, age) {
    return `Name: ${name}; Age: ${age}; 📎`;
}

function getInfoWithCurrentDate(cb) {
    const now = new Date();

    log(`Today: ${now.toISOString()}; \n ${cb("Dmitry", 33)}`);
}

getInfoWithCurrentDate(getInfo);

//-----

function survey(question, agreeFn, disagreeFn) {
    if (confirm(question)) {
        agreeFn();
    } else {
        disagreeFn();
    }
}

// survey(
//     "Ты мой друг?",
//     function () { log("Ты согласился, что ты мой друг 😎"); },
//     function () { log("Ты не согласился, получается не друг ❌"); },
// )

// ----- Стрелочные функции (arrow functions) -----

const sum = (a, b) => {
    return a + b;
};

log(sum(10, 20));

const multiplication = (a, b) => a * b; // без return

log(multiplication(10, 100));

const showName = name => `Name: ${name}`; // можно без круглых скобок, т.к один параметр

log(showName("Alex"));

// ----- IIFE -----

// Immediately-invoked function expressions
/*

```

В прошлом, поскольку существовал только var, а он не имел блочной области видимости, программисты придумали способ её эмулировать.

Этот способ получил название «Immediately-invoked function expressions» (сокращенно IIFE).

!!! Это не то, что мы должны использовать сегодня, но, так как вы можете встретить это в старых скриптах, полезно понимать принцип работы.

```
*/

// function () {
//     log("...")
// }

(function () {
    var message = "Hello";
    log(message);
})();

// log(message) // Ошибка: message is not defined

(function (name) {
    var message = "Hello";
    log(`${message} ${name}!`);
})("Anna");
```

```
// Call stack (Стек вызовов)

const log = console.log;

function one() {
    log("One");
}
function two() {
    log("Two");
}
function three() {
    log("Three");
}

one();
two();
three();

// -----

function one() {
    log("One");
    two();
}
function two() {
    log("Two");
    three();
}
function three() {
    log("Three");
}

one();
```

