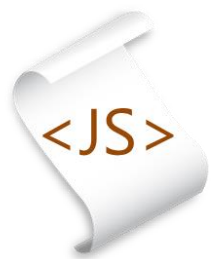


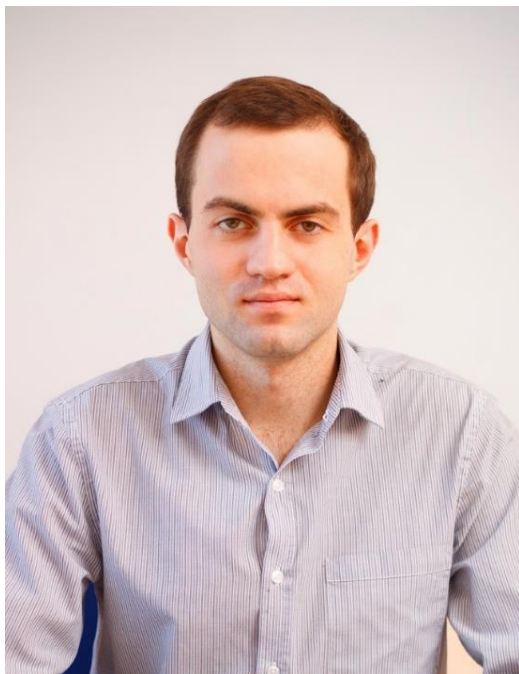
# JavaScript Essential



Функции

# JavaScript Essential

Автор курса



Дмитрий Охрименко  
МСТ



MCID: 9210561

# JavaScript Essential

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](http://itvdn.com)



Проверьте как Вы усвоили данный материал на [TestProvider.com](http://testprovider.com)

# JavaScript Essential

Тема

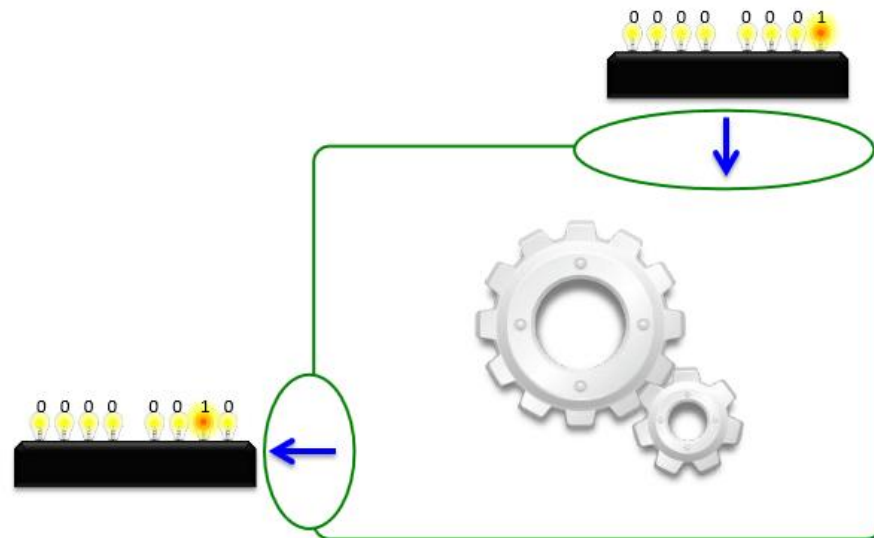
Функции

# Функции

## Функция

**Функция** – это блок кода, который можно многократно вызывать для выполнения.

Функция может принимать аргументы и возвращать значение.



Функции являются основными «**строительными блоками**» программы.

# Функции

## Создание функции

Ключевое слово  
объявления функции

Имя  
функции

Параметры (не обязательно)



```
function MyFunction(a, b) {  
  
};
```

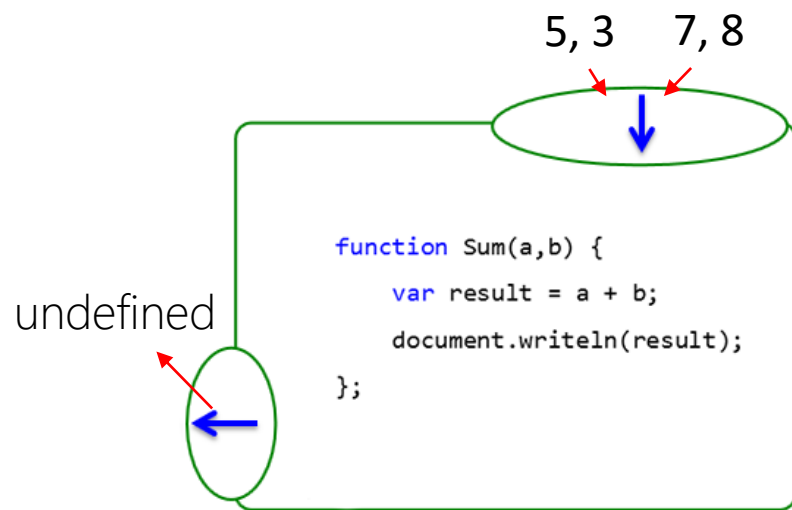
Функции являются основными «**строительными блоками**» программы.

# Функции

## Вызов функции

Будучи один раз определенной, функцию можно вызвать указав ее имя и операторные скобки «( )».

### Определение функции

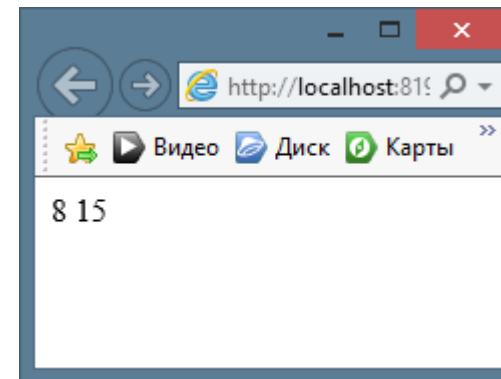


### Вызов функции

Sum(5,3);

Sum(7,8);

### Результат



**JavaScript** – язык не строгой типизации, поэтому тип параметров функции указывать не требуется.

# Функции

## Возвращаемые значения

Ключевое слово `return` – завершает выполнение функции и возвращает некоторое значение (по умолчанию – `undefined`).

```
function pow (x) {  
    return x * x;  
};
```

Если явно не указано ключевое слово `return`, оно неявно вызывается по завершению работы функции, и возвращает значение `undefined`.



# Функции

## Вложенные функции

В **JavaScript** допускается вложенные определения функций в другие функции.

```
function hypotenuse(a, b) {  
    function pow(x) {  
        return x * x;  
    };  
    return Math.sqrt(pow(a) + pow(b));  
};
```

Вложенные функции объявленные с помощью ключевого слова **function** могут находиться только внутри функции верхнего уровня.

# Функции

## Функциональные литералы

JavaScript позволяет определять функции в виде функциональных литералов.

```
let f = function(a, b) {  
    document.write(a + b);  
};  
f(4, 6);
```

ошибка



```
let f = function Func(a, b) {  
    document.write(a + b);  
};  
f(4, 6);  
  
Func(a, b);
```

# Функции

## Глобальные и локальные переменные

Переменные, созданные в функции, являются локальными и не доступны за пределами тела функции.

```
var f = function () {  
    var local_a = 4; // локальная  
    local_b = 5;    // глобальная  
};
```

Если переменная в теле функции объявлена без ключевого слова `var`, она будет создана как глобальная.

# Функции

## Аргументы функции

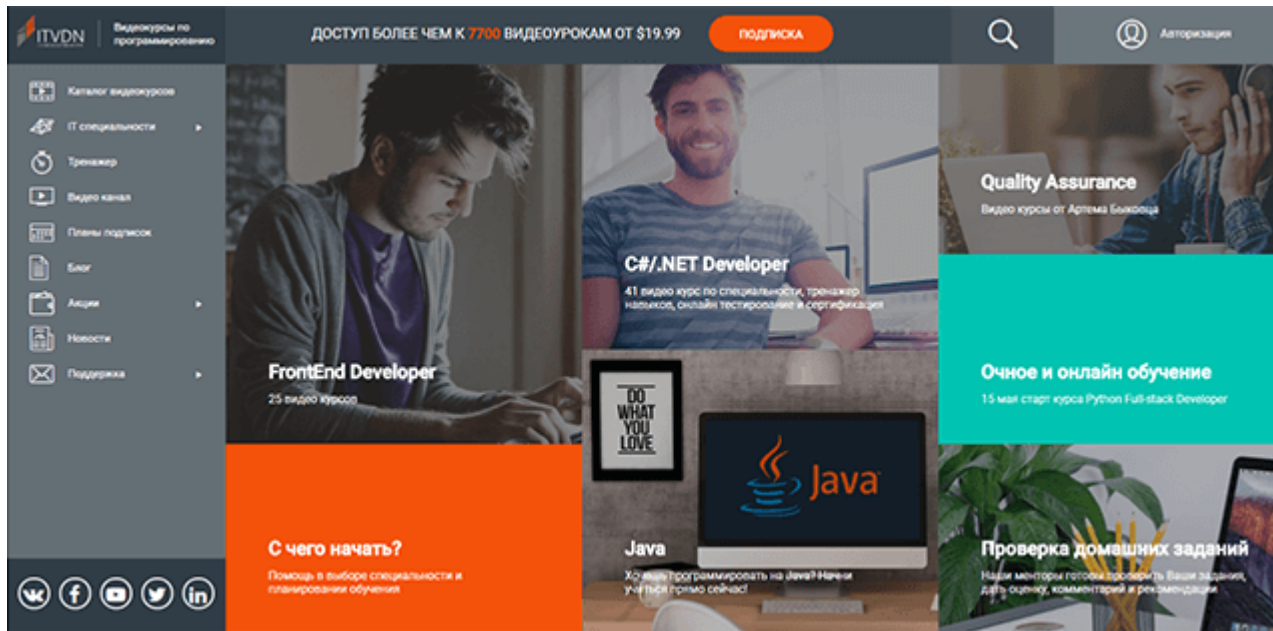
Функции в **JavaScript** могут вызываться с произвольным числом аргументов.

```
var f = function (a, b) {  
    document.writeln(a);  
    document.writeln(b);  
};  
f(12);  
f(12, 54, 32);
```

Если в функцию передать меньше аргументов чем ожидается, остальные будут инициализироваться как – **undefined**.

# Смотрите наши уроки в видео формате

ITVDN.com



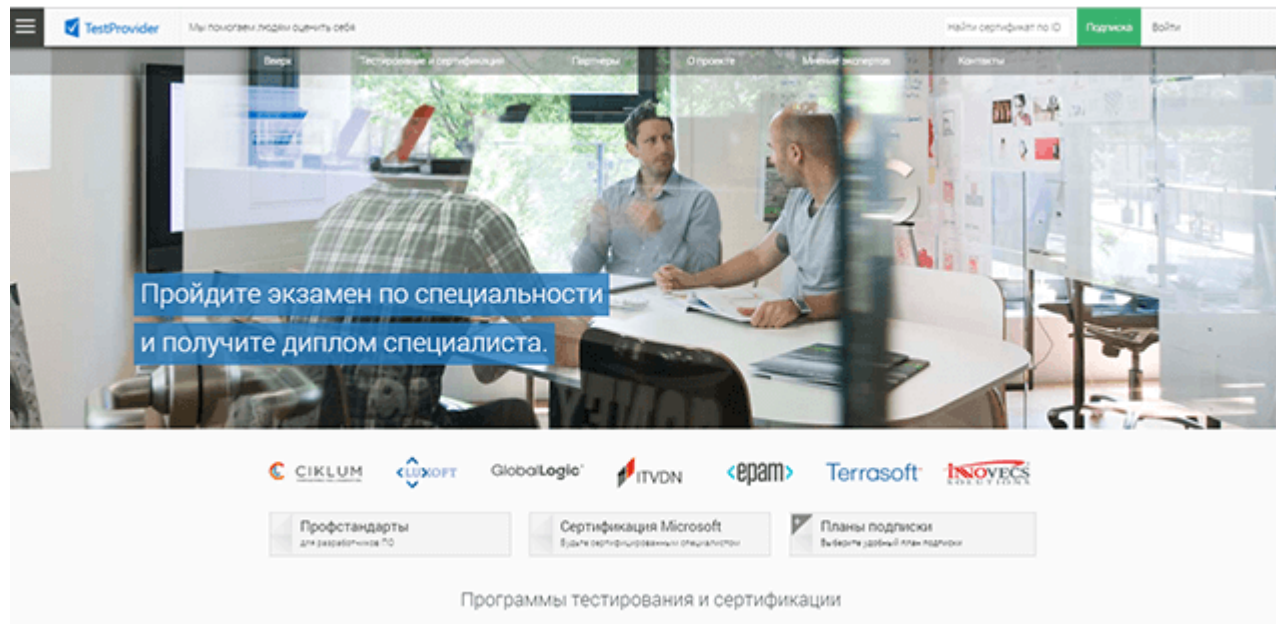
Посмотрите этот урок в видео формате на образовательном портале [ITVDN.com](http://ITVDN.com) для закрепления пройденного материала.

Курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics и другими высококвалифицированными разработчиками.



# Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на [TestProvider.com](https://testprovider.com)

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



# JavaScript Essential

Q&A

# Информационный видеосервис для разработчиков программного обеспечения

