

# Функции

**№ урока:** 3    **Курс:** CoffeeScript Fundamentals

**Средства обучения:** Компьютер с установленной Microsoft Visual Studio

## Обзор, цель и назначение урока

1. Рассмотреть работу функций

## Изучив материал данного занятия, учащийся сможет:

1. Понимать работу функций
2. Создавать функции различной сложности
3. Вызывать функции
4. Возвращать аргументы из функции

## Содержание урока

1. Обзор функций
2. Правила создания функций
3. Функции с опциональными аргументами
4. Создание функций с групповыми аргументами
5. Области видимости

## Резюме

**Функция** – это блок программного кода на языке CoffeeScript, который определяется один раз и может многократно вызываться. Функции могут иметь параметры, или аргументы – локальные переменные, значения которых определяются при вызове функции.

Для определения функции в языке CoffeeScript необходимо:

1. Создать переменную;
2. Созданной переменной присвоить конструкцию которая состоит из круглых скобок, в которых перечислены параметры принимаемые функцией, и знака `->`, который указывает что это функция и компилируется в ключевое слово **function**;
3. На следующей строке с отступом в несколько пробелов определить тело функции.

В качестве имени функции может использоваться любой допустимый CoffeeScript идентификатор

Ключевое слово **return** – завершает выполнение функции и возвращает некоторое значение (по умолчанию – **undefined**). Если явно не указано ключевое слово **return**, возвращаемым значением будет последняя инструкция в теле функции.

В языке CoffeeScript допускается вложение определений функции в другие функции.

Вложенные функции могут определяться только в коде функции верхнего уровня. Это значит, что определения функции не могут находиться, например, внутри циклов или условных инструкций.

Если одна функция вложена в другую функцию, то внутренняя функция имеет доступ к локальным переменным внешней функции. Такая техника называется – захват локальной переменной или замыкание.

Когда неизвестно количество аргументов, которые функция должна принимать, в языке CoffeeScript используется групповой аргумент, который является массивом, и записывается с помощью троеточия ([имя\_переменной]...). Групповой аргумент заменяет использование объекта arguments.

В языке CoffeeScript параметры функций могут принимать значения по умолчанию, для этого необходимо при объявлении параметра функции присвоить ему начальное значение. Если при вызове функции не передать значение параметру, который имеет значение по умолчанию, то в теле функции у него будет задано начальное значение.

### Закрепление материала

- Что такое функция?
- Для чего используются групповые аргументы в функциях?
- Зачем нужны функции?
- Как работает ключевое слово return?

### Дополнительное задание

Создайте четыре функции которые будут принимать два параметра, и выполнять арифметические операции. Функции должны возвращать значение полученное арифметической операцией. Предусмотрите возможность того, что пользователь может не передать один из параметров.

### Самостоятельная деятельность учащегося

#### Задание 1

Создайте функцию принимающую три числа. Функция должна находить среднее арифметическое параметров и выводить результат на экран.

#### Задание 2

Создать функцию, которая принимает число от 0 до 999 999. Функция должна находить сумму трех первых цифр и сумму последних трех цифр числа, и при равенстве сумм этих цифр, функция выводит в тело документа строку «Вы выиграли в лотерею», иначе – «Сегодня не Ваш день». Если число имеет меньше шести разрядов, старшие разряды считать нулями.

#### Задание 3

Создайте функцию, которая принимает один числовой параметр. Функция должна вычислять число в последовательности Фибоначи. Полученное значение функцией выведите на экран.

Последовательность Фибоначи – это последовательность чисел, в которой каждое следующее число равно сумме двух предыдущих. При этом считается что первый элемент последовательности равен 0, а первый – 1.

При решении данного задания используйте рекурсию.

### Рекомендуемые ресурсы

<http://coffeescript.org> – официальный сайт CoffeeScript