

Функции

№ урока: 4 Курс: JavaScript

Средства обучения: Компьютер с установленной Microsoft Visual Studio

Обзор, цель и назначение урока

- Рассмотрение функций.

Изучив материал данного занятия, учащийся сможет:

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

Содержание урока

1. Обзор функций.
2. Правила создания функций.
3. Рассмотрение управляющей структуры `return`.
4. Вложенные функции.

Резюме

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

Самый распространенный способ определения функции – использование инструкции `function`. Она состоит из ключевого слова `function`, за которым следуют:

- Имя функции;
- Заключенный в круглые скобки необязательный список имен параметров, разделенных запятыми;
- JavaScript-инструкции, составляющие тело функции, заключенные в фигурные скобки.

JavaScript позволяет определять функции в виде функциональных литералов.

Синтаксис функционального литерала во многом напоминает синтаксис инструкции `function`, за исключением того, что он используется как выражение, а не как инструкция, и ему не требуется имя функции.

Функция-конструктор – это функция, которая выполняет инициализацию свойств объекта и предназначена для использования совместно с инструкцией `new`.

В качестве имени функции может использоваться любой допустимый JavaScript-идентификатор.

Чаще всего в качестве имен функций выбираются глаголы или фразы, начинающиеся с глаголов. По общепринятому соглашению имена функций начинаются со строчной буквы. Если имя состоит из нескольких слов, в соответствии с одним из соглашений они отделяются друг от друга символом подчеркивания, примерно так: `like_this()`, по другому соглашению все слова, кроме первого, начинаются с прописной буквы, примерно так: `likeThis()`.

Объект Arguments – это объектное представление массива, позволяющего извлекать переданные функции значения по номеру, а не по имени.

В теле функции идентификатор **arguments** всегда имеет особый смысл; **arguments** это специальное свойство объекта вызова, ссылающееся на объект, известный как объект **Arguments**.

Функции в JavaScript представляют собой комбинацию исполняемого программного кода и области видимости, в которой этот код выполняется.

Функции в JavaScript имеют не динамическую, а статическую природу. Это означает, что они выполняются в области видимости, которая была создана на момент определения функции, а не на момент ее исполнения.

Ключевое слово **return** – завершает выполнение функции и возвращает некоторое значение (по умолчанию – **undefined**). Если явно не указано ключевое слово **return**, оно не явно вызывается по завершению работы функции, и возвращает значение **undefined**.

В JavaScript допускается вложение определений функций в другие функции.

Вложенные функции могут определяться только в коде функций верхнего уровня. Это значит, что определения функций не могут находиться, например, внутри циклов или условных инструкций. Эти ограничения распространяются только на объявления функций с помощью инструкции **function**. Функциональные могут присутствовать внутри любых выражений.

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* (*closure*).

Закрепление материала

1. Что такое функция?
2. Что такое объект **Arguments**?
3. Зачем нужны функции?
4. Как вызывается функция?
5. Сколько аргументов может принимать функция?
6. Для чего используется ключевое слово **return**?

Дополнительное задание

Создайте метод с именем Calculate, который принимает в качестве параметров три целочисленных значения и выводит на экран среднее арифметическое этих параметров.

Самостоятельная деятельность учащегося

Задание 1

Создайте четыре функции для арифметических действий: (Add – сложение, Sub – вычитание, Mul – умножение, Div – деление). Каждая функция должна принимать два параметра и выводить на экран результат вычисления. Функция деления должна делать проверку деления на ноль. Пользователь вводит значения, над которыми хочет произвести операцию и выбрать саму операцию.

Задание 2

Напишите функцию, которая будет принимать число и определять:

1. Является ли введенное число положительным или отрицательным.
2. Является ли оно простым
3. Делится ли оно на 2, 5, 3, 6, 9 без остатка

Рекомендуемые ресурсы

Функции

<http://learn.javascript.ru/function-basics>