



SHIFT
Training center

JavaScript. Уровень 2. Расширенные возможности

JavaScript. Уровень 2. Расширенные ВОЗМОЖНОСТИ

Модуль 1. Углубленная работа с функциями

Модуль 2. Методы массивов

Модуль 3. Объектная модель браузера

Модуль 4. DOM

Модуль 5. События

Модуль 6. Практическая работа

Модуль 7. Введение в асинхронный JS

Модуль 8. Введение в модули и Node.js

Модуль 9. Тестирование и сборка

Модуль 10. Практическая работа

Преподаватель

Валентин Степанов

- 20 лет в разработке
- 10 лет в автоматизации бизнес-процессов
- 10 лет в веб-программировании
- 3 года преподавания

mail@v-stepanov.ru

<https://t.me/valentinstepanov>

Модуль 1. Углубленная работа с функциями

- Методы `apply`, `call`, `bind`
- Замыкания
- Рекурсия
- Функции высшего порядка
- Каррирование

Функции

- Функция – это блок кода JavaScript, который определяется однажды, но может выполняться, или вызываться, любое количество раз
- Определение функции может включать список идентификаторов – параметры
- При вызове функций для параметров предоставляются значения, или аргументы
- Возвращаемое значение – значение выражения вызова функции
- Контекст вызова – экземпляр функции, значение ключевого слова `this`

Объявление функции

- Ключевое слово `function`
- Выражения функций
- Стрелочные функции
- Вложенные функции

Вызов функции

- Как функции
- Как методы
- Как конструкторы
- Косвенно через `call()` и `apply()`
- Неявно

Параметры функции

- Необязательные параметры
 - Значение `undefined` для отсутствующих параметров
 - Стандартные значения для параметров (по умолчанию)
- Параметры остатка
 - Во время описания `...args`
- Присвоение параметров через распространение
 - Во время вызова `...args`
- Деструктуризация параметров
 - Объект или массив в составе параметров

Функция как объект

- Имя функции – имя переменной
- Собственные свойства функции
- Пространство имен при вызове функции
 - Все локальные переменные остаются в пространстве имен функции

Замыкания

- Замыкание – это комбинация функции и лексического окружения, в котором эта функция была определена. Другими словами, замыкание даёт вам доступ к Scope (en-US) внешней функции из внутренней функции. В JavaScript замыкания создаются каждый раз при создании функции, во время её создания.
- При вызове функции создается локальная область видимости
- Окружение функции сохраняется между вызовами

Свойства и методы функций

- `length` – арность, количество ожидаемых аргументов
- `name` – имя функции
- `prototype` – ссылка на объект прототипа (кроме стрелочных функций)
- `call`, `apply` – косвенный вызов функции
- `bind` – привязка функции к объекту
- `toString` – исходный код или другое выражение функции
- Конструктор `Function`

Функции высшего порядка

- Функция, которая оперирует функциями, принимая одну или большее количество функций в качестве аргументов и возвращая новую функцию
- Функции в качестве аргументов функции
- Функции в качестве возвращаемого значения
- Мемоизация – сохранение кешированного значения функций
- Рекурсия – функция, вызывающая саму себя

Модуль 2. Методы массивов

- Методы map, filter, reduce
- Остальные методы

Методы итераторов

- `forEach` – проход по массиву
- `map` – выполнение функции над каждым элементом массива
- `filter` – применение предиката к каждому элементу массива, получение нового массива на основе предиката
- `find`, `findIndex` – поиск первого элемента массива, удовлетворяющего условию предиката
- `every`, `some` – проход по массиву с предикатом, если предикат возвращает истину на всех или одном элементе массива
- `reduce`, `reduceRight` – объединение элементов массива через функцию

Другие методы массивов

- flat, flatMap – выравнивание массивов
- concat – присоединение массивов
- push, pop, shift, unshift – работа с массивом как со стеком
- slice, splice, fill copyWithin – работа с подмассивами, срезами
- indexOf, lastIndexOf – поиск элементов в массиве
- includes – проверка нахождения элемента в массиве
- sort – сортировка массива
- reverse – изменение порядка элементов на противоположный
- join, toLocaleString – преобразование массива в строку

Модуль 3. Объектная модель браузера

- Введение в BOM/DOM
- Событийные HTML-атрибуты
- Таймеры и requestAnimationFrame
- Интерфейсы
- Поиск элементов и коллекции

Объектная модель браузера

- Объектная модель браузера (Browser Object Model, BOM) – ядро JavaScript для использования в браузере
- В основе BOM лежит объект `window` – экземпляр браузера
- `window` так же является ECMAScript-объектом `Global`

Объект location

- Сведения о текущем загруженном документе
- Функционал навигации
- Адрес документа разбирается на свойства hash, host, hostname, href, pathname, port, protocol, search, username, password, origin