

# Разработка на JavaScript

**Код:** WEB-007

**Длительность:** 40 ч.

## Описание:

Курс посвящен современному JavaScript (EcmaScript 2015+), использованию JS для работы в браузере (события, манипуляция HTML и CSS), работе с модулями JS (NPM, Webpack), работе с сервером (HTTP, REST, AJAX, Axios, WebSocket). Курс является основой для изучения веб-фреймворков Angular и React. Делается упор на применение ООП, работу с JSON, функциональное программирование. Детально изучается работа с классами, объектами, массивами, датами, иммутабельность и асинхронность.

## Цели:

После прохождения курса слушатели будут хорошо разбираться в возможностях языка JavaScript, получат основу для дальнейшего использования веб-фреймворков, таких как React или Angular. Слушателями будет подробно изучена структура и синтаксис языка, его объектная модель. Подробно рассматривается DOM модель HTML документа, дается введение в CSS, изучается техника управления DOM и CSS с помощью JavaScript. Изучаются модели обработки событий. Изучаются вопросы создания форм. Рассматривается технология JSON и технология Ajax, асинхронность и работа с сервером.

## Разбираемые темы:

### 1. Язык JavaScript:

- Версии и реализации JavaScript;
- Области применения JavaScript (браузер, сервер, мобильная разработка);
- Типы данных: String, Number, Boolean (понятие truthy и falsy);
- Операторы, приоритет операторов;
- Объявление переменных: var, const, let;
- Циклы: while, for, for ... in, for ... of;
- Работа с числами и строками;
- Значения null и undefined: отличия и цели применения;
- Работа с консолью;
- Исключения в JavaScript.

### 2. JSON, объекты и массивы:

- Формат JSON, примеры;
- JSON.parse и stringify;
- Объекты, работа с классом Object;
- Создание объектов (с помощью литерала {}, оператор new, метод create);

- Прототипическое наследование, ссылка `__proto__`;
- Геттеры и сеттеры;
- Оператор `in` и `hasOwnProperty`, понятие перечислимых свойств;
- Метод `toString()`, вызов и переопределение метода;
- Метод `valueOf()`, вызов и переопределение метода;
- Создание массивов: литералы и `new Array()`;
- Автоматическое изменение длины массива;
- Получение и установка длины массива `length`, усечение массива;
- Применение методов `join()` и `concat()`;
- Использование метода `sort`, задание порядка сортировки массива;
- Использование методов `slice` и `splice`;
- Использование методов `push`, `pop`, `shift`, `unshift`: создание стека или очереди.

### 3. Работа с датами и строками:

- Применение конструктора `new Date()`;
- Понятие компьютерного представления даты как времени с Epoch;
- Использование `getTime()` и `setTime()`;
- Парсинг строкового представления дат;
- Варианты создания строк в JS: использование одинарных и двойных кавчек;
- Интерполяция строк (строка в апострофах);
- Экранирование специальных символов в строке;
- Конкатенация строк, перевод в верхний и нижний регистр;
- Поиск подстроки с помощью `indexOf()` и `lastIndexOf()`;
- Выделение подстроки с помощью `substring()` и `substr()`;
- Применение методов `includes`, `endsWith`, `startsWith`.

### 4. Применение JavaScript в браузере: модель DOM:

- Управление окном браузера, документом, событиями и оформлением (CSS-стилями);
- Объект `window` и его применение;
- Подключение кода JavaScript к HTML-странице;
- Порядок выполнения скриптов на странице;
- Кэширование скриптов, CDN-серверы, принудительная перезагрузка JavaScript;
- Работа с ссылками, подключение JavaScript к ссылкам;
- Событие `onload` и его использование;
- XSS-атаки на сайт;
- Работа с DOM: использование `window`, `document`, `childNodes`;
- Манипуляции документом с помощью DOM: создание, удаление, перемещение элементов;
- Использование `getElementById()`, `getElementsByTagName()`, `getElementsByClassName()` для поиска элементов на странице;
- Работа с методами и свойствами `HTMLDocument` и `HTMLElement`.

### 5. Работа с событиями в браузере:

- Понятие событийного программирования;

- Обработчики и слушатели событий;
- Подключение обработчиков с помощью `addEventListener()`;
- Свойства и методы класса `Event`;
- Распространение событий вверх (bubbling) и вниз (capturing);
- Остановка распространения;
- События окна;
- События мыши и клавиатуры, специальные свойства событий;
- События формы;

#### 6. Работа с CSS и управление стилями:

- Стандарт CSS, основные свойства CSS;
- Общие стили и стили отдельных элементов;
- Каскадирование стилей;
- CSS-селекторы;
- Изменение CSS с помощью JavaScript;
- Отступы: `margin`, `padding`, `border`;
- Управление видимостью элементов: `visibility` и `display`;
- Позиционирование элементов: `absolute`, `fixed`, `relative`, установка `top` и `left`;
- Единицы измерения `px`, `pt` и другие;
- Переполнение элемента и управление через свойство `overflow`;
- Управление слоями документа: свойство `zIndex`;
- Анимация, перемещение элементов;
- Работа с классами, установка `className`;
- Работа с прозрачностью и переходами (transition);
- Метод `querySelectorAll()`.

#### 7. Работа с модулями в JavaScript, менеджер пакетов NPM, сборщик Webpack:

- Причины использования модулей;
- Экспорт из модуля;
- Импорт из модуля;
- Реэкспорт;
- Работа с NPM, файл `package.json`;
- Папка `node_modules`, загрузка пакетов с помощью NPM;
- Глобальная и локальная установка пакетов;
- Применение webpack для сборки проекта;
- Загрузчики CSS и TypeScript;
- Конфигурация `webpack.config.js`;
- Создание бандла, сжатие;
- Плагины webpack;
- Применение старых систем модульности – CommonJS и AMD;
- Использование `CommonsChunkPlugin`, несколько точек входа;
- Горячая замена модулей HMR.

#### 8. Классы и наследование в JavaScript:

- Прототипное наследование, ссылка `__proto__`;
- Прототип объекта;
- Использование `instanceof`;

- Ключевое слово `class`, создание классов;
- Ключевое слово `super`;
- Применение метода `Object.assign()`;
- Применение метода `Object.defineProperty()`;
- Геттеры и сеттеры;
- Spread-оператор для работы с объектами и массивами;
- Деструктурирование массива и объекта.

#### 9. Функциональное программирование в JavaScript:

- Функции как переменные;
- Массив `arguments`;
- Класс `Function`, методы `call()` и `apply()`;
- Метод `bind()`, установка `this` и части параметров с помощью `bind`;
- Замыкания в JavaScript;
- Инкапсуляция значений с помощью замыканий;
- Создание функции через конструктор `Function`: конструирование строки;
- Стрелочные функции, примеры их применения;
- Функциональная манипуляция массивом: `find`, `findIndex`, `filter`;
- Примеры использования `map/filter/reduce` для обработки данных в массиве.

#### 10. Иммутабельность, тестирование, паттерн MVC:

- Понятие иммутабельности, ее преимущества;
- Мутабельные и иммутабельные методы изменения объектов и массивов;
- Понятие и преимущества `Virtual DOM`;
- Тестирование иммутабельных функций, применение `Jest`;
- Мемоизация, ее преимущества;
- Основные ошибки и рекомендации при использовании функционального программирования;
- Паттерн MVC, создание пользовательского интерфейса;
- Обзор основных библиотек для разработки интерфейса: `Angular`, `React`, `Vue`.

#### 11. Асинхронность в JavaScript. Промисы.:

- Понятие и необходимость асинхронности;
- Работа с `setTimeout`, `setInterval`, `clearTimeout`;
- Функции обратного вызова (коллбэки);
- Потеря `this` при работе с функцией обратного вызова. Варианты решения (использование `bind`, стрелочных функций и др.);
- Понятие промисов, возвращение `Promise` из асинхронной функции;
- Метод `Promise.all()`, его применение;
- Синтаксис `async/await`, варианты применения, ограничения;
- Обработка внештатных ситуаций: `Promise reject`, обработка исключений при асинхронных вызовах.

#### 12. Работа с сервером. Протокол HTTP. REST-сервисы:

- Протокол HTTP;
- Заголовки и методы HTTP (`GET`, `POST`, `PUT`, `DELETE`, другие), передача данных;

- HTTP-серверы (обзор);
- Коды ответа HTTP-сервера;
- Использование форм для отправки данных;
- Передача данных на сервер с помощью AJAX;
- Архитектура REST: преимущества и особенности;
- Реализация CRUD (операций Create/Read/Update/Delete) при работе с REST;
- Применение функции `fetch()` для получения и отправки данных;
- Документирование REST-сервисов с помощью Swagger;
- Понятие HATEOAS, применение;

### 13. Библиотека AXIOS. Протокол WebSocket:

- Сравнение высокоуровневой библиотеки Axios и применения низкоуровневого `fetch()`;
- GET- и POST- запросы в Axios;
- Применение `async/await` с Axios;
- Обработка ошибок при использовании Axios;
- Отмена запроса с помощью Axios;
- Протокол WebSocket;
- Примеры использования WebSocket, сравнение с HTTP;
- HTML-анимация;

### Целевая аудитория:

Веб-разработчики.

### Предварительная подготовка - общее:

Слушатели знакомы с HTML и CSS, желателен опыт программирования на каком-либо языке с C-подобным синтаксисом (C, C++, Java, C#).

### Примечание:

Материалы курса представлены на английском языке.