

## Modern JavaScript

### Course Summary

#### Description

Having an in-depth knowledge of JavaScript makes it easier to learn a variety of frameworks, including React, Angular, and related tools and libraries. This course is designed to help you cover the core JavaScript concepts you need to build modern web-based applications.

You will learn to code comprehensive JavaScript from the ground up to the advanced and latest features. You will also learn how to write JavaScript in a professional environment using the new JavaScript syntax added in ES6, ES7, and ES8.

The course covers the fundamentals of JavaScript including syntax, variables, conditionals, functions, arrays and loops. The course also covers the latest features of JavaScript and advanced concepts, such as modularity, web storage and asynchronous nature using callbacks and promises. Within the course, the JavaScript code will run in both Node.js and the browser. Learn how to represent an HTML document in the Document Object Model (DOM), handle events and display JSON from the server to an HTML page. Learn techniques how to convert JSON to CSV to JS strings and vice-versa, how to set up test suites and test your code using Jasmine Framework, and how to debug JavaScript code using VS Code.

#### Objectives

At the end of this course, students will be able to:

- Examine major features in ES5, ES6, ES7 and ES8 and implement those features to build applications
- Understand data types, variables and scope
- Write functions, arrow functions and IIFE
- Modules in ES5 and ES6
- Become familiar with array methods
- Destructuring Assignment, Spread and Default Parameter
- Run JavaScript in both in Node.js and the browser
- Manipulate the DOM with JavaScript
- Handle JavaScript browser events
- Use web storage effectively
- Explore regular expressions in JavaScript
- How to debug using VS Code and handle errors
- Create Promise and Callback handlers to work with asynchronous processes
- Develop asynchronous flows using Promise chaining
- Use of await and async for asynchronous flows
- Unit Testing Using Jasmine Framework

#### Topics

- Getting Started
- Variables
- Operators
- Conditionals
- Functions
- Arrays
- JavaScript Object Notation
- JavaScript in the Browser
- Regular Expression
- Error Handling and Debugging
- Asynchronous Programming
- Unit Testing

#### Audience

This course is designed for those who are brand new to JavaScript and worked with another programming/scripting languages before. This course is also designed for those who currently use JavaScript but want a better sense of mastery over the language. Now you're ready to master JavaScript as well as explore the latest from ES6 to ES8.

#### Prerequisites

Basic understanding of HTML, CSS and JavaScript and programming concepts.

#### Duration

Five days

## Modern JavaScript

---

### Course Outline

#### I. *Getting Started*

- A. What is JavaScript?
- B. ECMAScript
  - 1. ECMAScript 5
  - 2. ECMAScript 6
  - 3. ECMAScript 7
  - 4. ECMAScript 8
- C. Babel
- D. Displaying Text
- E. JavaScript and HTML
- F. JavaScript and Node.js

#### II. *Variables*

- A. Variables
- B. JavaScript Data Types
- C. Scope
- D. The var Keyword
- E. Hoisting in JavaScript
- F. Comments

#### III. *Operators*

- A. Assignment Operator
- B. Increment and Decrement Operator
- C. Arithmetic Operator
- D. Comparison Operator
- E. Logical Operator
- F. Short Circuit Operators
- G. Truthy and Falsey

#### IV. *Conditionals*

- A. if
- B. if-else
- C. switch
- D. ternary operator
- E. The switch Statement

#### V. *Functions*

- A. Basic Functions
- B. Function Expressions
- C. Modules
  - 1. Modules ES5 in Node.js
  - 2. Modules ES5 in the Browser
  - 3. Modules ES6 in Node.js
  - 4. Modules ES6 in the Browser
- D. Arrow Functions
- E. Immediately Invoked Functions (IIFE)
- F. Callback Functions

#### VI. *Arrays*

- A. Array Basics

- 1. Accessing Elements of an Array
- 2. Adding Elements in an Array
- 3. Removing Elements in an Array
- 4. Insert, Delete or Replace Elements in Array

#### B. Loops Construct and Iterating Arrays

- 1. while
- 2. do-while
- 3. for
- 4. continue and break
- 5. for ... of

#### C. Array Methods

- 1. Array.forEach()
- 2. Array.map()
- 3. Array.filter()
- 4. Array.find()
- 5. Array.every()
- 6. Array.some()
- 7. Array.reduce()

#### D. Rest Parameters

#### E. Spread Operator

#### F. Destructuring Assignment

#### VII. *JavaScript Object Notation*

##### A. JavaScript Objects

##### B. Creating JavaScript Objects

- 1. Literal Notation
- 2. Constructor Notation

##### C. for ... in loops

##### D. Object Destructuring

##### E. Merge Objects Using Spread

##### F. JSON Structure

##### G. JSON as Data Transmission

- 1. JSON.stringify()
- 2. JSON.parse()

##### H. JSON vs JS Object Literal Notation

##### I. JS Date and Methods

##### J. JS String and Methods

#### VIII. *HTML Forms*

##### A. What is DOM?

##### B. DOM tree

##### C. innerHTML

##### D. DOM Queries

##### E. Looping HTML Collection

- 1. for ... of
- 2. Array.forEach() method
- 3. Classical for loop
- 4. Spread Operator

## Modern JavaScript

---

### Course Outline (cont'd)

- F. Looping Node List
  - 1. Classical for loop
  - 2. Array.forEach() method
- G. Advanced CSS Style Selectors
- H. Event Handling
  - 1. HTML Event Handler Attribute
  - 2. Traditional DOM Event Handlers
  - 3. Event Listener
- I. JavaScript Object Notation in HTML oage
- J. Web Storage
  - 1. window.localStorage
  - 2. window.sesionStorage

#### IX. Regular Expression

- A. Methods to Find Patterns in String
  - 1. search()
  - 2. test()
  - 3. match()
- B. Create Regular Expression
- C. The Patterns
  - 1. The [] Character
  - 2. Qualifiers
  - 3. Groups and Ranges
  - 4. Metacharacter
  - 5. Special Characters
  - 6. Grouping and Capturing
- D. LookAround, LookAhead and LookBehind
- E. JS Global Methods
  - 1. parseInt()
  - 2. parseFloat()
  - 3. toFixed()

#### X. Error Handling and Debugging

- A. How Program Works
  - 1. Error
  - 2. Error Handling
  - 3. try-catch-finally
- B. throw
- C. Debugging
- D. VS Code Debugging Tools

#### XI. Asynchronous Programming

- A. What is Asynchronous JS?
- B. Revisiting Callback Functions
  - 1. setTimeout(), setInterval()
- C. Async JavaScript
- D. Promise
  - 1. Create a Promise
  - 2. Using a Promise
  - 3. then().catch() and finally()
- E. sync and await keywords

#### XII. Unit Testing – Jasmine Framework

- A. Why Unit Test?
- B. JavaScript – Truthy and Falsey
- C. Jasmine Framework
  - 1. Configuration
  - 2. Basic Elements of Jasmine Testing
  - 3. Builtin Matcher
- D. Mocking Code in Jasmine
  - 1. spyOn()
  - 2. Jasmine.createSpy()
  - 3. Jasmine.createSpyObj()