JavaScript Programming

Duration: 35 hours

Prerequisites: Basic computer skills and knowledge of HTML fundamentals including use of Cascading Style Sheets.

Course Description: JavaScript is a scripting language that is commonly used to create and control dynamic Website content along with its use in the Node.js runtime. This hands on JavaScript training course provides the fundamental knowledge necessary to design and develop dynamic Web pages using JavaScript and to be able to grasp JavaScript libraries and frameworks such as jQuery, React and Angular. Students will learn the syntax of the JavaScript language and how to use JavaScript in a Web browser.

Topics include ways to declare variables, use of intrinsic JavaScript objects such as Math, Date and Array, declaring and calling functions, defining custom objects, error handling, working with arrow functions, and using conditional logic. Students will learn how to include JavaScript in a Web page and how to use browser-based APIs such as the Document Object Model (DOM), Geolocation and Web Storage. Upon completion students will be able to utilize event handling, form validation, JSON and Ajax.

Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency.

Students Will Learn

- Including JavaScript in a Web page
- Using the Document Object Model (DOM) to access the HTML elements on the page dynamically
- Using JavaScript objects effectively, including the `window` object and the `navigator` object
- Creating and using variables
- Using event handlers to handle user-triggered events
- Using regular expressions with string methods to perform basic validation
- Working with primitive data types
- Passing arguments to a function
- Using conditional constructs and loops
- Debugging JavaScript code
- Using Geolocation and Web Storage JavaScript APIs
- Working with JSON objects
- Using Ajax to make asynchronous calls to a Web server

Overview

Overview of the JavaScript Language

- Typical Uses of JavaScript
- Evolution of the JavaScript Language
- ECMAScript Version Support
- What is Core JavaScript?

JavaScript Fundamentals

- Using Strict Mode
- JavaScript and Whitespace
- Adding Comments
- Declaring Variables

What are JavaScript APIs?
- Writing JavaScript in Visual Studio Code

`var`, `let` and `const` Keywords
    - Local vs. Global Variables
    - Declaring Block-scoped Variables
- JavaScript DataTypes
    - Primitive Types
    - Objects
- Performing Data Conversions
- Working with JavaScript Operators
    - Assignment Operators
    - Type Operators
    - Bitwise Operators
    - Conditional Operator
- Working with Dates

Working with Numeric Data

- Solving Precision Problems with Numbers
- Rounding Numbers
- Creating Random Numbers
- Arithmetic Operation
- Working with Exponents
- Converting to Numbers
    - `parseInt`, `parseFloat` and `Number` Functions

Dealing with Errors

- Types of JavaScript Errors
- Error Handling
    - `try`
    - `catch`
    - `finally`
- Throwing Custom Errors with throw

Controlling Flow

- Creating Boolean Expressions
    - Using Standard and Strict Equality Operators
    - Using Logical and Comparison Operators
- Conditional Constructs
    - `if else`
    - `switch`
- Looping Constructs
    - `do while`
    - `while`
    - `for`
    - `for in`
    - `for…of`
- Branching Statements
    - `break`
    - `continue`
    - Using Labels

Working with Arrays

- Creating Arrays
    - Using the Array Constructor
    - Using Literal Notation
    - `String.split()`
    - `Array.of()`
- Accessing Array Elements
- Common Array Methods
    - `forEach()`
    - `includes()`
    - `find() and findIndex()`
- Adding and Removing Elements
- Sorting Arrays
    - Creating a Custom Sort Function
- Looping through an Array

Working with Strings

- Creating String Objects
- Using String Methods
    - Searching within a String
    - Returning a Substring
    - Converting Character Case

Implementing Functions

- Declaring Functions
- Function Parameters
    - Default Parameters
    - Rest Parameters
- Invoking Functions

Template Literals
- Escape Sequences

- Passing Arguments
  - Spread Operator
- Named and Anonymous Functions
- Local vs. Global Variables
- Returning Values from a Function
- The `this` Keyword
- IIFEs
- Closures
- Callback Functions
- Arrow Functions

Using JavaScript in the Browser

- Embedding JavaScript in a Page
- Referencing an External Script File
- Using the JavaScript Console
  - Logging Messages in the Console
  - Debugging with Console
- Deferring Script Loading
- Where to Include JavaScript
- `DOMContentLoaded` vs Load Events

Browser Object Model

- Overview of the Browser Object Model
- Using the `window` Object
  - Opening and Closing Windows
  - Using Timers
  - Interacting with the User with Dialogs
- Using the document Object
  - Writing to the Web Page
  - Useful Properties of the Document
- `navigator` Object
- Feature detection
- `location` Object
- `screen` Object

Introduction to the Document Object Model (DOM)

- Overview of Common DOM Types
- Selecting DOM Elements
  - Selecting Elements By Id and Name
  - Selecting Elements By Class
  - Using CSS Selectors to Select Slements
- Modifying Page Content with the DOM
  - `innerHTML` and `textContent` Properties
  - `createElement`
  - `createDocumentFragment`
- Accessing Attributes

Event Handling in JavaScript

- Overview of JavaScript Events
  - The `event` Object Members
  - Event Capturing and Bubbling
- Adding Event Handlers
  - Traditional (Inline) Model
  - Registering Events in JavaScript
  - Adding Multiple Handlers with `addEventListener`
- Mouse and Keyboard Events
- Preventing Default Behavior
- Canceling Events

Scripting CSS with JavaScript

- Using the style Property
- `CSSStyleDeclaration` Objects
- `getComputedStyle`
- Modifying Classes with the `classList` API

Working with Forms

- Enhancing Forms with JavaScript
  - Preventing a Double Submission
  - Selecting All Checkboxes
  - Conditionally Showing Fields
- Working with Form Elements

Form Validation

- Client-Side vs Server-Side Validation
- Validating Form Fields
  - Checking Required Fields

Using Regular Expressions

- Overview of Regular Expressions
- Using Regular Expressions in JavaScript
  - Creating a RegExp Object

Checking Length of Input
- Overview of HTML 5 Validation
- Constraint API
- Canceling Form Submission

Working with JavaScript Objects and JSON
- Understanding Objects
- Defining Custom Objects
  - Object Literal Notation
  - Defining Properties and Methods
  - Creating a Constructor
- Using the `in` and `instanceof` Operators
- Overview of JSON
- Using `JSON.stringify` and `JSON.parse`

HTML5 JavaScript APIs
- Overview of APIs
- Using Modernizr
- Geolocation API
  - Using the `getCurrentPosition()` Method
  - The Position Object
- Using the Google Maps JavaScript API
  - Obtaining an API Key
  - Creating a Map
  - Adding Custom Markers to a Map
  - Adding Info Windows to a Map
- Web Storage API
- Local Storage
- Session Storage

Using `i` and `g` Flags
- Pattern Matching with Regular Expressions
- Using Backreferences

Using Ajax
- Overview of Ajax
- The `XMLHttpRequest Object`
- Configuring an Ajax Request
- Processing JSON with Ajax
- Using Ajax to Call a Web Service

Related Bootcamp

Track	Duration	Price
Web Developer	119 hours	$3,580

Contact Us

Address: 6 Hemlock Drive Chelmsford, MA 01824

Phone: 978.250.4983

Mon - Thur: 9 am - 5 pm EST
Fri: 9 am - 4 pm EST

E-mail: info@developer-bootcamp.com

Copyright© 2025 Developer Bootcamp