

Overview of Client-Side JavaScript

- Including JavaScript in a Page
 - Using the `<script>` Tag
 - Referencing an External Script File
 - Using an Event Handler
 - Using a URL
- Evolution of the JavaScript Language
- Common Misconceptions About JavaScript
- ES5 vs ES6
 - New ES6 Data Types
 - `Map` and `Set` Collections
 - ES6 Arrow Functions
 - ES6 String Template Literals
- Using the JavaScript Console
 - Logging Messages in the Console
 - Debugging with Console

Controlling Flow

- Conditional Constructs
 - `if else`
 - `switch`
- Looping Constructs
 - `do while`
 - `while`

JavaScript Fundamentals

- Using Strict Mode
- Adding Comments
- Debugging JavaScript Code
- Declaring Variables
 - `var`, `let` and `const` Keywords
 - Local vs. Global Variables
 - Declaring Block-scoped Variables
- Using JavaScript Primitive Types
 - String, Number and Boolean
 - `undefined`
 - `null`
- Performing Data Conversions
- Interacting with the User
 - `alert`
 - `prompt`
 - `confirm`
- Working with JavaScript Operators
 - Arithmetic Operators
 - Assignment Operators
- Using the Math and Date Objects

Working with Strings, Numbers and Arrays

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

- `for`
- `for in`
- Branching Statements
 - `break`
 - `continue`
- Equality and Identity Operators
- Logical and Comparison Operators
- Error Handling
 - `try`
 - `catch`
 - `throw`
 - `finally`

Implementing Functions

- Declaring Functions
- Invoking Functions
- Named and Anonymous Functions
- Passing Arguments
- Local vs. Global Variables
- Returning Values from a Function
- The `this` Keyword
- Default Parameters
- Explaining Closures

Introduction to the Document Object Model (DOM)

- Selecting DOM Elements
 - `getElementById`

- Formatting Numbers
- Creating Arrays
 - Using the Array Constructor
 - Using Literal Notation
- Accessing Array Elements
- Common Array Methods
- Defining and Using Associative Arrays
- Predefined Object Arrays

Browser Object Model

- Overview of the Browser Object Model
- Using the `window` Object
 - Opening and Closing Windows
 - Using Timers
 - Passing Data Between Windows
- Using the `document` Object
 - Writing to the Webpage
 - Accessing Properties of the Document
- `navigator` Object
- `location` Object
- `screen` Object

Event Handling in JavaScript

- Overview of JavaScript Events
 - Identifying Differences Between Browsers

- `getElementsByTagName`
- `getElementsByName`
- `getElementsByClassName`
- `querySelector`
- `querySelectorAll`
- Modifying Page Content with the DOM
- Manipulating CSS using JavaScript

Working with Forms

- Accessing Forms from JavaScript
- Working with Form Elements
- Validating Form Fields
 - Checking Required Fields
 - Checking Length of Input
- Canceling Form Submission

Working with JavaScript Objects and JSON

- Understanding Objects
- Defining Custom Objects
 - Object Literal Notation
 - Defining Properties and Methods
 - Creating a Constructor
 - ES6 Enhanced Object Literals
- Extending an Existing Object with `prototype`
- Using the `in` and `instanceof` Operators

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

Using Regular Expressions

- Overview of Regular Expressions
- Using Regular Expressions in JavaScript
 - Creating a `RegExp` Object
 - Using `i` and `g` Flags
- Validating Strings with Regular Expressions
- Using Backreferences

Using Ajax

- Overview of Ajax
- The `XMLHttpRequest` Object
- Configuring an Ajax Request
- Handling Errors on `readystatechange`
- Using JSON with Ajax

- Overview of JSON
- Using `JSON.stringify` and `JSON.parse`

HTML5 JavaScript APIs

- Web Storage API
 - Local Storage
 - Session Storage
- Geolocation API