

INTRODUCTION TO JAVASCRIPT (3 Day Course)

Technical
Course
Outline

Course Summary

This introductory course will immerse students into the JavaScript client-side programming language. Gaining an understanding of the JavaScript programming language, its syntax, and how it is processed are the core goals of the class. Students will gain an understanding of the Browser's Object Model (BOM) as well as the W3C Document Object Model (DOM) standards to be able to write cross-browser, standards compliant code. Cross-browser event handling, working with and validating XHTML forms via JavaScript, and using JavaScript to modify CSS properties to create Dynamic HTML (DHTML) are covered.

Intended Audience:

This class is intended for **web page developers** who want to make their web pages more dynamic, interactive, and interesting using programming code embedded inside XHTML. Anyone wishing to learn AJAX programming should take this course.

Prerequisites:

To ensure an effective learning environment *for all attendees*, it is imperative that *each student* meet minimum course prerequisites of the successful completion of, or 90% proficiency with, the topics from the following course:

- "Advanced HTML, XHTML, & CSS" (Please refer to the course outline to make sure you meet this requirement):

Course Contents:

Introducing JavaScript

- JavaScript vs. Java
- JavaScript vs. JScript vs. VBScript vs. ECMAScript
- The XHTML `<script>` Tag
- Hiding JavaScript From Older Browsers
- Ensuring JavaScript Can Be Read As XML
- JavaScript Comments
- Configuring Internet Explorer For JavaScript Debugging

The Object-Based Nature of JavaScript

- Objects, Methods, Properties
- Browser Supplied Objects
- Language Supplied Objects
- W3C Supplied Objects

JavaScript Language Syntax

- When To Use The Semi-Colon
- JavaScript Code Blocks and { }
- JavaScript Case-Sensitivity & Best Practices
- JavaScript Variables
 - The JavaScript Primitive Types
 - Declaration & Implicit Data Typing
 - Scope
 - Implicit & Explicit Type Conversions
 - `parseFloat()` & `parseInt()`
 - The Dangers of `parseInt()` & `parseFloat()`
 - Specifying A Radix
- Decision Making Code
 - The `if()` & `switch()` Constructs
- Iteration - Looping
 - `for()`, `for...in`, and `do` Loops
- User-Defined Functions (UDF's)
 - Creating The Function
 - Without Parameters
 - With Parameters
 - Parameter Scope
 - Returning From A Function
 - Returning A Value
 - Calling Your Function
 - Functions Embedded In The Web Page
 - Functions Kept In An External '.js' File

JavaScript Language Syntax (continued)

- JavaScript Operators
 - Logical, Comparison, Unary, & Mathematical Operators
 - Operator Precedence
- Dealing With `null`, `undefined`, `infinity` and `NaN`

The Browser's Object Model (BOM)

- Browser Compatibility
- The "document", "window", & "navigator" Objects
 - Properties & Methods
- Browser Object Arrays
 - Using Array Methods & Properties

The W3C Document Object Model

- Thinking In Terms Of "Nodes"
- Node Types
- The DOM "document" Object vs. The BOM "document" Object
- Parsing Nodes
 - `getElementById()` & `getElementsByTagName()`

JavaScript Events & Event Handling

- What HTML Elements Have Which Event Handlers?
- DOM Level 0 vs. Level 2 Event Handling
- Wiring Event Handlers To Functions
- Event Bubbling and Canceling Events

JavaScript In The Real-World

- Alerts, Prompts & Feedback
 - `alert()`, `prompt()`, `confirm()`
- Form Validation
- Feature Detection
- Debugging

JavaScript Core Objects

- **String**, **Date**, & **Array** Objects
 - Properties & Methods
- Instantiating Objects
 - Implicitly vs. Explicitly

Dynamic Page Content (DHTML)

- What Is DHTML?
- Accessing An (x)HTML Element's CSS Characteristics in JavaScript