

Course Summary

Students will continue to learn how to leverage the JavaScript language to perform more powerful tasks in a web page. Students will take their next steps with the Browser Object Model (BOM) and the Document Object Model (DOM), and gain proficiency with some of the more advanced JavaScript language elements like timers, object creation, and asynchronous page updates via AJAX technologies to be able to build more rich and dynamic web applications.

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 HTML.

Prerequisites:

To ensure an effective learning environment *for all attendees*, it is imperative that *each student* meet minimum course prerequisites.

Successful completion of, or 90% proficiency with, the topics from the following courses (please refer to the course outlines to make sure you meet this requirement):

- "Advanced HTML, XHTML, & CSS"
- "Introduction To JavaScript"

Course Contents:

JavaScript Reviewed

- Inline Scripting
- The `<script>` Tag
- External JavaScript Libraries

Advanced JavaScript Language Concepts & Elements

- The `setTimeout()` and `clearTimeout()` Methods
- The `setInterval()` and `clearInterval()` Methods
- Advanced String & Date/Time Methods
- Implicit & Explicit Arrays
- Compacting `if()` Logic With JavaScript's Ternary Operator
- Extending Types Via `prototype`

Debugging & Error Handling

- Handling Errors With `try...catch`
- The Integrated IE 8 Debugger
- Firebug For Firefox

The W3C Document Object Model (DOM)

- The "Core" DOM vs. The "HTML" DOM
- Thinking In Terms Of "Nodes" and "NodeLists"
- The `documentElement` Node
- Navigating The Document's Nodes
- Dynamically Creating New Nodes
- DOM Event Handling
 - DOM Level 0 and Level 2 Event Handling
 - Event Bubbling & Cascading
 - Cross-Browser Event Handling
 - Accessing The `event` Object In A Cross-Browser Way

DHTML

- HTML + CSS + JavaScript = DHTML
- CSS Properties Accessible Via JavaScript Properties
- Setting CSS Property Values From JavaScript
 - The `style` Object
- Reading CSS Property Values From JavaScript
 - `element.currentStyle` For IE and Opera
 - `window.getComputedStyle` for Mozilla

User-Defined Functions

- Types Of Functions
 - Declarative Functions, Function Literals, & Anonymous Functions
- Functions & Recursion
- Functions As Objects
- Function Context & The `this` Keyword
- Closures
- Avoiding Global Scope With Wrapper Functions

Writing Cross-Browser Supported Scripts

- Checking For Browser Compatibility Using Feature Detection
- Graceful Degredation

Asynchronous JavaScript And XML (AJAX)

- What is AJAX & Why Should I Care?
- The `XmlHttpRequest` (XHR) Object
 - Preparing & Sending The Request
 - Processing the Results
 - The `onreadystatechange` Callback Event Handler
 - The `readyState` Property
 - The `responseText` Property
 - The `responseXml` Property

JavaScript Object Notation (JSON)

- Creating An Object Out Of Thin Air With `{}`, `:`, and `[]`
- Processing Result Strings With `eval()`
 - Why `eval()` is Evil!
- The `JSON` Object
 - The `parse` and `stringify` Methods

Using Public JavaScript Libraries

- The Resurgence of JavaScript
- JQuery (overview)
 - Accessing The JQuery Object with `$`
 - The `$(document).ready` Event Handler
 - JQuery's Way Of Getting Element References
 - Show & Hide Animations