ProTech
protechtraining.com

"Charting the Course ...

... to Your Success!"

Advanced JavaScript with DOM

Course Summary

Description

This course provides indoctrination in the practical use of the umbrella of technologies that are on the leading edge of web development.

Working within in a dynamic, hands-on learning environment, guided by our expert web development team, attendees will understand:

- Complex and powerful Javascript objects and functionality
- How to traverse and manipulate the DOM and how to handle events in ways that work in all browsers
- Closures and prototypes and other exotic features of JavaScript
- Best practices for using Javascript so that it works unobtrusively and performs well

This course is “skills-focused”, designed to train attendees in robust web development and design skills, coupling the most current, effective techniques with the soundest development and design practices. Throughout the course, students will be led through a series of progressively advanced topics, where each topic consists of lecture, group discussion, comprehensive hands-on lab exercises, and lab review.

Topics

- Advanced JavaScript
- Working with DOMs
- Javascript Techniques

Audience

This is an intermediate and beyond-level web development course, designed for experienced developers who need to extend their knowledge of web design and development.

Prerequisites

Attendees should have previous experience or working knowledge of developing software applications. Basic HTML and CSS development skills are required. This is a programming class. Familiarity with an object-oriented language is helpful, and real world programming experience is a must.

Duration

Three days

ProTech
protechtraining.com

"Charting the Course ...

... to Your Success!"

Advanced JavaScript with DOM

Course Outline

I. Session: Advanced JavaScript

A. Lesson: JavaScript Reviewed
B. JavaScript Defined
C. Embedding Scripts in HTML
D. Identifiers and literals
E. Non-Standard Operators
F. Loop Structures
G. Iteration
H. Conditional Structures
I. The with Statement
J. Functions in JavaScript
K. Invoking Functions
L. Functions as Data
M. Functions as Methods
N. Function Constructor
O. Function Scope and Closures
P. Lesson: JavaScript Objects
Q. JavaScript Object
R. JavaScript Literals
S. Associative array
T. Object Universal Properties
U. JavaScript Arrays
V. Arrays
W. Array Methods
X. Working with Arrays
Y. Strings
Z. String Objects
AA. JavaScript and Regular Expressions
BB. Regex Content
CC. Character Sets and Alternation
DD. Additional Constructs
EE. Repetition
FF. Example:
GG. Using Regular Expressions in JavaScript
HH. Creating Regular Expressions
II. Using Regular Expressions – String Match
JJ. Using Regular Expressions – String Replace
KK. Using Regular Expressions – String Split
LL. Using Regular Expressions – String Search
MM. Working with the RegEx Object
NN. Lesson: JavaScript Classes
OO. JavaScript Classes are NOT
PP. Classes as Functions or Functions as Classes
QQ. Function Constructors
RR. Creating New Objects in JavaScript
SS. JavaScript Classes
TT. Creating Multiple Objects
UU. Prototypes
VV. Prototype Property
WW. Extending Classes With Prototype
XX. Extending Classes
YY. Prototyping
ZZ. Extending Built-in Objects
AAA. Reusable, Flexible Classes
BBB. Lesson: JavaScript Event Handling
CCC. JavaScript Event Handling
DDD. Event Models
EEE. Basic Event Handling
FFF. HTML Events
GGG. Handling HTML Events
HHH. Event Handling With Dom Level 2
III. Three Phases
JJJ. Event Handler Registration
KKK. InnerHTML
LLL. DOM 2 Event Propagation:

II. Session: Working with DOMs

A. Lesson: HTML DOM Mechanics
B. The HTML Document Object Model (DOM)
C. Introduction to Objects in Documents
D. DOM Structure
E. Structure of a Page in Browser Memory
F. Dynamic Document Content
G. The HTML DOM
H. Accessing Nodes
I. Accessing Element Nodes
J. getElementsByTagName() Code Sample
K. getElementsByTagName() Code Explanation
L. getElementById()
M. getElementById() Code Explanation
N. Properties for Accessing Element Nodes
O. The this Object
P. Attaching Events
Q. Code Explanation and Walkthrough
R. DOM Element Properties
S. Event Handler by Function Assignment
T. Event Handler by Closure
U. Unobtrusive JavaScript
V. Accessing Attribute Nodes
W. Accessing Nodes by Type, Name or Value
X. Node Type Constants
Y. nodeName
Z. nodeValue
AA. Accessing Nodes by Class Name
BB. getElementsByClassName() Method
CC. removeChild() Method
DD. DOM Differences: The Whitespace Problem
EE. Creating New Nodes
FF. Methods for Inserting Nodes
GG. Rendered Code and HTML in Example

Advanced JavaScript with DOM

Course Outline (cont'd)

HH. Using Node Manipulations
II. setAttribute() Method
JJ. Review of Forms
KK. Form Object
LL. Scripting Form Elements

III. Session: Javascript Techniques
A. Lesson: Pushing Javascript
B. JavaScript Security Limits Capability
C. Same Origin Policy
D. Timers Within Browser Windows
E. Scheduling in JavaScript
F. setTimeout() in Action
G. setInterval() in Action
H. Uses for JavaScript Timers
I. Window and Screen Information
J. Opening And Manipulating Windows
K. Retrieving X,Y Location of Mouse Click
L. JavaScript Execution: Global Object
M. JavaScript Execution: Call Object
N. JavaScript Execution Context
O. Scope Chain
P. Window Object – Global Context
Q. Executing Scripts
R. The onload Event Handler
S. Window Object As Execution Context
T. Threading (Actually Non-Threading)
U. Manipulating Document During Loading
V. Lesson: JavaScript Best Practices
W. JavaScript Code Convention Overview
X. Managing JavaScript
Y. Formatting and Structure
Z. Comments, Comments, Comments
AA. Variable Declarations
BB. Function Declarations
CC. Names in JavaScript
DD. Drupal Naming Conventions
EE. Statement Structure
FF. return and if Statement Conventions
GG. Control Statement Conventions
HH. Managing Whitespace
II. JavaScript Best Practices Overview
JJ. Naming Best Practices
KK. Globals are a Bad Thing
LL. Resolving Shared Resources
MM. Enforce Quality Code
NN. Never Use eval or Similar Constructs
OO. Enhance Progressively
PP. Modularize and Encapsulate
QQ. Do NOT Trust Data Returning From the End Point
RR. Additional JavaScript Best Practices
SS. Lesson: Performance and Optimization
TT. Code Optimization
UU. Adding Elements to a Page
VV. JavaScript Verbosity Tradeoffs
WW. Minimizing Dot-Notation
XX. The Venkman Profiler
YY. Venkman Profiling Results
ZZ. Profiling Kayak
AAA. A Snippet from the Results
BBB. Using the Results
CCC. Memory Considerations
DDD. JavaScript Memory
EEE. Tools for Measuring Memory Footprint
FFF. Removal of DOM Nodes
GGG. Cyclic References
HHH. Reusing DOM Nodes
III. Interactions Between Techniques
JJJ. Lesson: Security
KKK. Misconception #1
LLL. Security: The Complete Picture
MMM. Verizon's 2011 Data Breach Report
NNN. US Secret Service Continues to Battle
OOO. Verizon AppSec Recommendations
PPP. Unvalidated Input: Description
QQQ. Unvalidated Input: Fixes
RRR. Define System and Trust Boundaries
SSS. Focus on Each Trust Boundary
TTT. Injection Flaws: Description
UUU. SQL Injection Example
VVV. Or
WWW. Heartland – Expensive SQL Injection Exploit
XXX. Initial Goal of XSS Attack: Insert Content
YYY. Ultimate Goal of Attacker: Get User to "touch" Location
ZZZ. XSS: Description
AAAA. Spoofing: Description
BBBB. Targeted Spoofing Attacks Can be Damaging
CCCC. Attacks are Constant and Changing
DDDD. Spoofing: Fixes
EEEE. JavaScript is Integral Part of Rich Interfaces
FFFF. How Attackers See JavaScript Applications
GGGG. Attack Surface Change When Moving to AJAX

Advanced JavaScript with DOM

Course Outline (cont'd)

HHHH. JavaScript Privacy Concerns
IIII. Factors that Increase Attack Surface
JJJJ. Injection and Cookie Tampering
KKKK. Data Tampering
LLLL. Cross Site Scripting
MMMM. Dangerous Developer Assumptions
NNNN. High Level Recommendations (Client Side)
OOOO. Three Basic Tenets for Safe JavaScripting
PPPP. Common Vulnerabilities and Exposures
QQQQ. OWASP Top Ten for 2010
RRRR. How Many Principles Can be Violated?
SSSS. CWE/SANS Top 25 Most Dangerous SW Errors
TTTT. Monster Mitigations
UUUU. Useful Links
VVVV. JavaScript Security Resources