

Advanced JavaScript

OVERVIEW

This workshop teaches the participants how to take advantage of some of the advanced features offered by JavaScript for building better maintainable applications large scale applications. It also introduces some of the frameworks that can assist in this process.

PAST OCCURANCES OF THIS PROGRAM

- Amdocs, Pune
- Honeywell, Bangalore (3 occurrences)

OBJECTIVES

At the end of this training course, the participants will:

- Understand and appreciate the dynamic typed nature of JavaScript language
- Be able to understand the exploit some of the functional programming techniques in JavaScript
- Understand and resolve scope issues in JavaScript
- Techniques for modularizing JavaScript code
- Know how to modularize and load JavaScript files using Require.js
- Be aware of some of the quirks in the JS language
- Be able to use Handlebar templates for UI templating

SUGGESTED AUDIENCE

Web developer who wants to build best-of-breed web UIs with the simplicity and elegance of JavaScript.

DURATION

2 Days

PARTICIPANT PREREQUISITES

Must Have

- Web development experience with HTML, CSS, JavaScript

Good To Have

- Exposure to any server technology (J2EE, .NET, Ruby etc)

INFRASTRUCTURE REQUIREMENTS

- Any Web Server (IIS or Tomcat)
- Chrome
- Internet Connection

DAY WISE SYLLABUS

- Expressive JavaScript
 - The Flexibility of JavaScript

- JavaScript as a Loosely Typed Language
 - Functions as First-Class Objects
 - Object Mutability
- Functions In Depth
 - Function Objects
 - Function Literal
 - Function Invocation Patterns
 - Augmenting Types
 - Recursion & Closures
 - Functions as Callbacks
 - Function Currying & Memorization
- Closures In Depth
 - How Closures Work
 - Private Variables, Callbacks and Timers
 - Binding Function Contexts
 - Overriding Function Behavior
- Object Orientation With Prototypes
 - Object Instantiation
 - Constructors
 - Inheritance and Prototype Chain
 - Extending Object
 - Extending Number
 - Instantiation Issues
 - Writing class-like code
- JavaScript Gotchas
 - Global Variables
 - Scope
 - Semicolon Insertions
 - Type Coercion
- Using Templates
 - Avoid string concatenation for dynamic generation of UI
 - Template syntax
 - Exploring Handlebar.js
 - Binding data to templates
 - Using pre-compiled templates
- Modular JavaScript
 - Techniques for modularizing JavaScript code
 - Using Require.js for loading dependent modules
- JavaScript Design Patterns
 - Common Design Pattern Implementations
 - Module Pattern
 - Revealing Module Pattern
 - Observer Pattern

- Factory Pattern
 - Decorator Pattern
 - Strategy Pattern
- Introduction to Unit Testing in JavaScript
 - Overview of Unit Testing Frameworks
 - Using Jasmine for Unit Testing
- Performance Profiling JavaScript Code
 - Understanding Performance Problems
 - Profiling using Chrome and Firebug
 - Identifying bottlenecks and resolving them
- Debugging Techniques
 - Using “console”
 - Developer tools in the browser
 - Using breakpoints
 - Advanced breakpoints
- Overview of JavaScript MVC Frameworks
 - Backbone.js
 - Knockout.js
 - Angular.js