|  |
| --- |
| 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) * Pega Systems, Bangalore * 12 other instances   **Code repository**  **@** <https://github.com/tkmagesh?tab=repositories> |
| **Trainer** |
| **K. magesh**  GitHub - <http://github.com/tkmagesh>  LinkedIn - <http://in.linkedin.com/in/tkmagesh> |
| **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 loadJavaScript files using Require.js * Be aware of some of the quirks in the JS language * Be able to use Handlebar templates for UI templating * Using ES6 |
| **suggested audience** |
| Web developer who wants to build best-of-breed web UIs with the simplicity and elegance of JavaScript. |
| **DURATION** |
| 3 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 (Mandatory) * Node.js * Any Text Editor |
| **DAY WISE SYLLABUS** |
| **Day-1**   * 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   **Day-2**   * 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 JavaScirpt code * Debugging Techniques   + Using “console”   + Developer tools in the browser   + Using breakpoints   + Advanced breakpoints * JavaScript Design Patterns   + Common Design Pattern Implementations   + Model View Separation   + Module Pattern   + Revealing Module Pattern   + Observer Pattern   + Factory Pattern   + Decorator Pattern   + Flyweight Pattern   + Strategy Pattern   + Publisher Subscriber   + PubSub Implementation * MVC Pattern in JavaScript   + Creating Models   + Creating Views   + Manipulating Collections   **Day-3**   * Introduction to Unit Testing in JavaScript   + Overview of Unit Testing Frameworks   + Using Jasmine for Unit Testing   + Builtin matchers   + Covering Before and After   + Using Spies for Mocking * Karma Test Runner   + Overview   + Configuration   + Auto watching   + Configuring Browsers * Gulp Task Runner   + Overview   + Configuration   + Installing and using plugins   + Commonly used plugins ex., JSLint, minification, uglification etc * ES6   + New Features in ES6 Overview   + Using Transpilers   + Array Functions   + Variable Declarations (let, const)   + Creating Classes   + Accessor Methods   + Modularity in ES6 * Asynchronous Programming in JavaScript   + Callbacks   + Events   + Advantages and Disadvantages of Callbacks and Events   + Using Promises   + ES6 Promises and polyfills   + Promise Chaining   + Deferred Objects   + 'fetch' Api * Performance   + Guidelines for High Performance JavaScript   + Local Variables   + With Statement   + Impact of Closures   + Objects and Array usage   + Loop Optimization   + Handling DOM   + CSS Styles and Classes |