|  |
| --- |
| Building Rich Internet Applications |
| **Overview** |
| This workshop aims at equipping the participants with the necessary knowledge and skills required to build rich internet applications using cutting edge technologies, frameworks and tools. |
| **Past occurances of this program** |
| * Amdocs, Pune * Honeywell, Bangalore (3 occurrences) |
| **OBJECTIVES** |
| At the end of this training course, the participants will:   * Understand and exploit the HTML5 features offered by browser for data storage, multi-threading capabilities and real time updates from the server. * Be able to understand the exploit the functional and object-oriented 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 the quirks in the JS language * Manipulate DOM using jQuery * Be able build RIA using Angular.js |
| **Case Studies** |
| Following are the case studies that will be developed by the participants during the workshop:   * An “Expense Management” application using Angular.js |
| **suggested audience** |
| Web developer who wants to build best-of-breed web applications with the simplicity and elegance of JavaScript. |
| **DURATION** |
| 3 Days |
| **PARTICIPANT PREREQUISITES** |
| **infrastructure requirements** |
| * Node.js * Chrome * Any good text editor (EditPlus/Notepad++/Sublime Text) * Internet Connection (Mandatory) |
| **DAY WISE SYLLABUS** |
| **Day – 1JavaScript In Depth**   * ~~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 * jQuery Overview   + Creating Wrapped set   + Using Selectors   + Manipulating Styles   + Handling Events   **Day –2 & 3Angular.js**   * Introduction to Angular.js   + How Angular.js is opinionated   + Difference between Backbone.js and Angular.js * Angular.js Building Blocks   + Controller Component   + Model Component   + View Component   + Directives   + Filters   + Services   + DI in Angular.js * Anatomy of an Angular.js Applications   + Creating Boundaries using ng-app   + Model View Controller   + Templates and Data Binding   + Repeating elements in templates   + Using Expressions, CSS Classes and Styles   + Using Controllers for UI responsibility separation   + Responding to model changes * Data Binding in Angular.js   + Understanding Built-in Directives   + Scope resolution   + One way and Two way data binding * Using Filters   + Filters Overview   + Understanding Filter Expressions   + Building custom Filters * Services   + Services Overview   + Modularity using Services   + Injecting Services * Directives   + Directives Overview   + Creating Directives   + The Directive Definition Object   + Compilation and Linking   + Creating Components * Communicating with Servers   + Communicating over $http   + Configuring the requests   + Sending Http Headers   + Caching Responses   + Request and Response Transformation   + Using RESTful Resources   + Communication over WebSockets * Unit Testing   + Testing Models using Jasmine   + Test considerations for Directives and Filters   + Using Angular Mocks   + Installing and Using Karma Test Runner * Modular JavaScript   + Techniques for modularizing JavaScirpt code   + Using Require.js for loading dependent modules * HTML5   + Using Offline Application Cache for caching application resources   + Using LocalStorage and SessionStorage for persisting data * Miscellaneous   + Overview of OSS Angular Modules   + Customizing Angular Modules |