**HTML5**

**Duration : 6 days**

**Pre requisite : Good knowledge of Javascript OOPS,DOM,HTML & CSS would be helpful but not mandatory**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Day 1:**

**Introduction:**

**Overview of HTML5**

**History of HTML5**

**The Myth of 2022 and Why It Doesn’t Matter**

**Who Is Developing HTML5?**

**A New Vision**

**Compatibility and Paving the Cow Paths**

**Utility and the Priority of Constituencies**

**Interoperability Simplification**

**Universal Access**

**A Plugin–Free Paradigm**

**What’s In and What’s Out?**

**What’s New in HTML5?**

**New DOCTYPE and Character Set**

**New and Deprecated Elements**

**Semantic Markup**

**Simplifying Selection Using the Selectors API**

**JavaScript Logging and Debugging**

**window JSON**

**DOM Level 3**

**Using WAI-ARIA with HTML5 for Accessibility**

**CANVAS API:**

**Overview of HTML5 Canvas**

**Canvas Coordinates& registering the Canvas dimensions**

**Drawing on Cavas with paths,curves etc.**

**Working with Solid colors,Gradients & Transparancy**

**Importing External Images & Setting the background**

**Working with Color & Geometrical transformations**

**Creating text,graphs & charts**

**Animating a Vertical Bar-Chartwith fine tuning.**

**Working with Pixel Data**

**CSS and Canvas**

**Create High-Res, Retina-Display-Ready Media with Canvas**

**Clipping Cavas drawings & saving them to a file.**

**When Not to Use Canvas**

**Fallback Content**

**Implementing Canvas Security**

**Implementing techniquesfor Backward compatibility.**

**Building an Application with HTML5 Canvas**

**SVG API :**

**Understanding SVG**

**Creating 2D Graphics with SVG**

**Adding SVG to a Page**

**Simple Shapes & Text**

**Transforming SVG Elements**

**Reusing Content**

**Patterns and Gradients**

**SVG Paths**

**Building an Interactive Application with SVG**

**Adding the CSS Styles**

**Implementing techniques for Backward compatibility.**

**Building an Application with SVG**

**GEOLOCATION:**

**Comparing Geolocation techniques in the past & modern day Geolocation**

**Understanding GPS/ IP Address/ Cell IDs/ WiFi and Bluetooth**

**LBS (Location based services)**

**Understanding Latitude,Longitude,Speed,Course & Accuracy**

**Getting you current location**

**Browser compatibility & Fallbacks.**

**Reverse geocoding&Mapping location**

**Getting Distance & Directions between two places.**

**Following a moving location**

**Combing geolocation with google maps**

**Triggering the Privacy Protection Mechanism**

**Saving Geographical information**

**Geolocation usage – Geo Marketing,Geo social,Geo tagging,Geo tagging & Geo applications.**

**Building a Real-Time Application with HTML5 Geolocation**

**Implementing techniques for Backward compatibility.**

**Alternative methods when Native Geolocation fails (Geo.js & MaxMind)**

**Media API (video & audio):**

**Flash V/s HTML5 video**

**Adding Video & Audio to a page**

**Supported Audio & Video formats& Codecs**

**Lossy & Lossless compression**

**Media specific attributes Vs Global attributes**

**Deployment challenges on Mobiles Converting Audio & Video to supported formats using open source & commercial software**

**Using a Frame grabber**

**Custom Controls,Seekbar,Progressbar with Javascript & CSS**

**Applying CSS skins & transforms**

**Working with multiple tracks,Subtitles & Captions with Captionator,Playr & the Leanback Player**

**Integrating Video with Canvas & SVG**

**Applying Visual filters using Canvas & SVG**

**Debuggin,Browser support &Licensing issues.**

**Implementing techniques for Backward compatibility.**

**Building an Application with HTML5 Media API**

**WEB FORMS API:**

**HTML forms fundamentals:**

**Introduction**

**Form Basics**

**Demo Form Basics**

**Form Settings**

**Demo Form Settings**

**Summary**

**HTML form inputs:**

**Introduction**

**Text inputs**

**Demo Text inputs**

**Selections**

**Demo Selections**

**Input attributes**

**Demo Input attributes**

**Input commands**

**Demo Input commands**

**Summary**

**Organizing HTML forms:**

**Introduction**

**Labels**

**Fieldsets**

**Tab index**

**Access keys**

**Summary**

**HTML form scenarios:**

**Introduction**

**Scripting forms**

**Handling multiple forms**

**Uploading files**

**Summary**

**Constraint Validation: Native Client Side Validation for Web Forms**

**What is Constraint Validation?**

**DOM API**

**willValidate**

**validity**

**validationMessage**

**checkValidity()**

**setCustomValidity()**

**HTML Attributes**

**Novalidate**

**Formnovalidate**

**CSS Hooks**

**:invalid and :valid**

**Resetting Default Styling**

**Inline Bubbles**

**Removing the Default Bubble**

**Current Implementation Issues and Limitations**

**setCustomValidity**

**Declarative Error Messages**

**Title Attribute**

**Polyfilling**

**Webshims**

**H5F**

**Day 2**

**HTML form server processing:**

**Introduction**

**Accessing Data**

**Demo ASP.NET Forms**

**Demo PHP Forms**

**Uploading Files**

**Demo ASP.Net Files**

**Demo PHP Files**

**Demo Ruby Files**

**Required Processing!**

**Model View Controller**

**Demo MVC**

**Summary**

**Building an Application with HTML5 Forms API**

**WEB WORKERS API:**

**What are web workers ?**

**Possibilities & Limitations of web workers**

**Inline,Dedicated & Shared Workers**

**Creating a worker,Assign roles & Deploying the same.**

**Leveragin a Shared Worker**

**Worker support in modern browsers**

**Managing multiple workers**

**Parsing data with workers**

**Perform Heavy array computations**

**Using timers in conjunction with worker**

**Work with pixel manipulations**

**Make twitter JSONP requests**

**Connect to share workers at same time with multiple browser windows**

**Transferable objects**

**Debuging Your Workers**

**Implementing techniques for Backward compatibility.**

**Building an Application with HTML5 Web Workers API**

**HTML5 Custom Data Attributes (data-\*)**

**Attribute Name**

**Attribute Value**

**How can I use data attributes?**

**What shouldn’t I use data attributes for?**

**Using data- attributes with JavaScript**

**A word of warning**

**CROSS DOCUMENT MESSAGING API:**

**Understanding Origin Security**

**Browser Support for Cross Document Messaging**

**Building an Application Using the postMessage API**

**XMLHttpRequest Level**

**Cross-Origin XMLHttpRequest**

**Progress Events**

**Browser Support for HTML XMLHttpRequest Level**

**Building an Application Using XMLHttpRequest Structured Data&Framebusting**

**Implementing techniques for Backward compatibility.**

**Building an Application with HTML5 CDM**

**Server Sent Events (SSE)**

**Possible Applications**

**Overview of the API**

**new EventSource(url)**

**Properties of Server-Sent Events**

**Message Format**

**Typical Server**

**Polyfills and Tweaks to the Server**

**Why Not Use WebSockets**

**WEB SOCKET API:**

**Understanding WebSocket**

**WebSocket API**

**WebSocket Protocol**

**Writing a Simple Echo WebSocket Server**

**Using the WebSocket API**

**Checking for Browser Support**

**Building a WebSocket Application**

**Adding the Geolocation Code**

**Combining Geolocation & Web sockets together.**

**Building Instant Messaging and Chat over WebSocket with XMPP**

**Using Messaging over WebSocket with STOMP**

**VNC with the Remote Framebuffer Protocol**

**WebSocket Security**

**Deployment Considerations**

**Inspecting WebSocket Traffic**

**Implementing techniques for Backward compatibility.**

**Web RTC API:**

**Introduction to Web Real-time Communication (WebRTC)**

**Introduction**

**History of Real-time Communication on the web**

**What you can do with WebRTC**

**Where WebRTC is supported**

**Architecture of a WebRTC Application**

**Introduction**

**Security with WebRTC**

**The Full WebRTC Environment**

**Understanding Server Technologies for WebRTC**

**Why Would We Need Servers?**

**Introducing ICE, STUN and TURN**

**Signaling Options for WebRTC**

**Options for Server Setup and Hosting**

**Recap of the Module**

**Introducing the WebRTC API**

**Overview of the WebRTC APIs**

**Accessing Webcam and Microphone with MediaStream (getUserMedia)**

**Establishing a Peer Connection with RTCPeerConnection**

**Understanding Data Communication with RTCDataChannel**

**Recap of the Module**

**Setting Up Your Development Environment**

**Introduction to Setting Up Your Environment**

**Software and tools for WebRTC development**

**Recap of the module**

**Create a Two-person Video Chat Using Peer.js**

**Introduction to Peer.js Library**

**Set Up the HTML for Two-person Video Chat**

**Write JavaScript Calls to Peer.js**

**Test the Two-person Chat Application**

**Recap of the Module**

**Create a Multi-person Chat Application Using SimpleWebRTC**

**Introduction to SimpleWebRTC Framework**

**Set up the HTML for Multi-person Video Chat**

**Write JavaScript Calls to SimpleWebRTC**

**Test the Multi-person Chat Application**

**Recap of the Module**

**WEB STORAGE & File System API:**

**Introduction:**

**Technology Landscape**

**Tradional storage mediums Vs the new Web Storage**

**Practical Uses**

**Narrowing Down the Scope**

**Concepts Immediately Invoked Function Expressions**

**Concepts $$result Module**

**Concepts mockJSON**

**Concepts Knockout.js**

**Development Environment**

**No Guarantees**

**Clearing Local Data**

**Browser Support**

**Summary**

**Web Storage:**

**Introduction**

**What's in a Name?**

**What is Web Storage?**

**Capacity**

**Testing Capacity**

**Features**

**Browser Support**

**Fallbacks and Polyfills**

**Caveats**

**Getting and Setting Values**

**Remove Item**

**Keys and Length**

**Clear**

**Session Storage**

**Exceed Quota**

**Storage Event**

**Persistent Form Demonstration**

**HTML Markup**

**localDataService JavaScript Module**

**viewModel JavaScript Module**

**Stepping Through the Code**

**Summary**

**Day 3:**

**Building an Application with HTML5 local/session storage**

**IndexedDB Introduction and Concepts:**

**Introduction**

**What is IndexedDB?**

**Event Lifecycle**

**Features**

**Capacity**

**Browser Support**

**Fallbacks and Polyfills**

**Caveats**

**Summary**

**IndexedDB Initialization & CRUD :**

**Introduction**

**Demo Opening a Database**

**Demo Deleting a Database**

**Demo The db Model Object**

**Demo Create Object (Insert)**

**Demo Read Object**

**Demo Update Object**

**Demo Delete Object**

**Summary**

**IndexedDB Cursors, Indexes and Ranges :**

**Introduction**

**Demo db Model for Cursor, Index and Range Demos**

**Cursor Concepts**

**Demo Cursors - Selecting Sets of Data**

**Demo Indexes - Selecting Individual Objects**

**Range Concepts**

**Demo Numeric Range**

**Demo String Range**

**Demo Controlling Cursor Direction**

**Summary**

**IndexedDB Keys, Capacity, Performance and Versions:**

**Introduction**

**Unique Identifier (Keys) Concepts**

**Demo Creating Object Store Keys**

**Demo Loading k Objects into a Database**

**Detecting When a Cursor is 'Done'**

**Demo Working with Large Sets of Data**

**Demo Managing Database Versions**

**Demo Capacity Capabilities**

**Summary**

**IndexedDB Abstractions & Implementing an Edit Screen:**

**Introduction**

**Demo Introduction to the Homes List Screen**

**Demo Homes List Markup**

**Demo Homes List db Model**

**Demo Abstracting IndexedDB - Error Handling**

**Demo Abstracting IndexedDB - Delete and Open Database**

**Demo Abstracting IndexedDB - Get All**

**Demo Abstracting IndexedDB - Insert, Update and Delete**

**Demo Homes List View Model**

**Demo Stepping Through the Code**

**Summary**

**Building an Application with HTML5 Indexed DB API**

**File System Introduction, Concepts & Initialization:**

**Introduction**

**What is the HTML File System?**

**Features**

**Capacity**

**Browser Support**

**Fallbacks and Polyfills**

**Caveats**

**Storage Types**

**Demo Initialization (Temporary Storage)**

**Demo Initialization (Permanent Storage)**

**Demo Wrapping Up Initialization Code**

**HTML File System Explorer (Chrome Extension)**

**Summary**

**File System Directories - Create, List, Delete, Move & Copy :**

**Introduction**

**Demo Create and Read Directory**

**Demo Create Sub Directories**

**Demo List Directory Contents**

**Demo Delete and Recursive Delete**

**Demo Move, Copy and Rename**

**Summary**

**File System Building an Abstraction Layer over Directories :**

**Introduction**

**Demo localFileSystem Module - Error Handling**

**Demo localFileSystem Module - Request File System**

**Demo localFileSystem Module - Create Directory**

**Demo localFileSystem Module - Directory Exists**

**Demo localFileSystem Module - Get Directory Entries**

**Demo localFileSystem Module - Delete**

**Demo localFileSystem Module - Move, Rename and Copy**

**Summary**

**File System Files - Create, Read, Write, Delete, Move & Copy :**

**Introduction**

**Demo Create and Get File**

**Demo Read, Write and Update File**

**Demo Delete, Move, Rename and Copy File**

**Demo File Abstractions Overview**

**Demo localFileSystem Module - Create File**

**Demo localFileSystem Module - Get and File Exists**

**Demo localFileSystem Module - Read, Prepend and Append**

**Demo localFileSystem Module - Delete and Replace File**

**Demo localFileSystem Module - Move, Rename and Copy File**

**Summary**

**File System Testing Capacity Limits & Implementing a File Editor:**

**Introduction**

**Demo File System Capacity Limits**

**Demo File Editor Demonstration**

**Demo File Editor Markup**

**Demo File Editor View Model**

**Summary**

**Libraries:**

**Introduction**

**What is store.js?**

**Demo store.js**

**What is amplify.js?**

**Demo amplify.js**

**What is lawnchair.js?**

**Demo lawnchair.js**

**Summary**

**Building an Application with HTML5 File System API**

**OFFLINE APPLICATIONS API:**

**Understanding Offline or Occasionally connected applications**

**Cache manifest - Cache,Network & fallback in detail**

**Create &Use the manifest to detect connectivity**

**Updating cache with the manifest**

**Application Cache API**

**Understading Events under the AppCache API :**

**i.e, load,checking,noupdate,downloading,progress,**

**cached,updated ready,obsolete & error events**

**Disk Space & Expiration**

**Deleting the local cache**

**Implementing techniques for Backward compatibility.**

**Building an Application with HTML5 Offline Applications API**

**NOTIFICATIONS API:**

**Notification Permissions**

**Browser Compatibility**

**Displaying a Simple Notification**

**Creating a Tweet Notification Page**

**Implementing techniques for Backward compatibility.**

**DEVICE API’s - CONTACTS,MESSAGING & NETWORK INTERFACE API:**

**Retrieving All Contacts and Mobile Numbers**

**Battery Status Events**

**HTML Media Capture with File Input**

**Device Orientation and Motion Events**

**Creating a Bubble level**

**Implementing techniques for Backward compatibility.**

**Building an Application with HTML5 Device API**

**Browser History API (Session):**

**Introduction:**

**Setup**

**Manipulating browser history: an overview**

**Detecting History API support**

**Pushing a new URL onto the stack**

**Handling back and forward buttons with the popstate event**

**Updating the stack with replaceState()**

**Basic Offline**

**Changing Browser History**

**History Basics &Browser Compatibility**

**Adding to History with pushState**

**Creating an Image Viewer**

**Popping State & changing history in the Image Viewer**

**Using Advanced State Data Objects to Pass Informationacross Pages**

**Testing History Security**

**Helpful Libraries**

**Building an Application with HTML5 Browser History API**

**Feedback:**

**Invalidation**

**Progress & Debugging**

**Javascript:**

**JavaScript API**

**Status**

**API Data Storage**

**Network**

**Gotchas**

**HTML5 Vulnerabilities & Protecting apps :**

**XSS,Javascript & more**

**Communication in HTML5**

**Local Storage options**

**Clickjacking**

**Protecting your HTML5 Applications.**

**Dealing with Backward compatibility & cross browser issues :**

**Need for the same ?**

**Implementing backward & cross browser compatibility for HTML5 API's includes:**

**CANVAS API**

**SVG API**

**Geolocation API**

**Media API**

**Forms API**

**Browser History API**

**Web Workers API**

**Cross Document Messaging API**

**Web Sockets API**

**Web Storage API**

**Offline Applications API**

**Notification API**

**File system API**

**Device API API**

**The Current State of Web Polyfills**

**Polyfilling: Past, Present, and Future**

**What Is a Polyfill (and What Is It Not)?**

**Types of Polyfills**

**Why Polyfills Still Matter**

**Building an Custom DropIn Polyfill for Forms Cross Browser functionatlity Applications API**

**Principles and Practices of Polyfill Development**

**Building Responsible Polyfills**

**Principles of Responsible Polyfill Development**

**Building Your First Polyfill, Part 1: Getting Started**

**The HTML5 Forms Polyfill**

**Setting Up Your Polyfill Project**

**Specifying the API**

**Deciding What to Build**

**Adding Basic Features**

**Beefing Up Your Polyfill with Additional Features**

**Building Utility Polyfills**

**Polyfilling Visual Features With CSS**

**Testing Your Work Across Browsers**

**Building Your First Polyfill, Part 2: Build Workflows and Cross-Browser Testing**

**Setting Up Your Project Workflow**

**Jump-Starting Your Workflow**

**Adding Unit Tests to Your Polyfill**

**Automating Cross-Browser Polyfill Testing**

**Configuring Cross-Browser Tests with Karma**

**The Bottom Line: Use What Works for You!**

**Building Your First Polyfill, Part 3: Performance and Edge-Case Testing**

**Building for Performance**

**Dealing with Browser-Specific Edge Cases**

**Mobile-Specifiuiopc Considerations**

**Prollyfilling and the Future of the Web Platform**

**The Future of Polyfilling**

**Polyfilling: Not Just For Older Browsers**

**Prollyfilling: Extending the Web Forward**

**Prollyfills In Action: Prollyfilling CSS**

**Prollyfills in Action: ServiceWorker**

**Prollyfills in Action: Web Components**

**Strategies for Polyfilling Experimental APIs**

**Building Your First Prollyfill**

**Prollyfills vs. Polyfills: What’s the Difference?**

**The Resource Priorities Prollyfill**

**Specifying the API and Deciding What to Build**

**Setting Up Your Prollyfill Project**

**Adding Prollyfill Features**

**What’s Next?**

**Colophon**

**Future of HTML5 ( overview)**

**Browser Support for HTML5**

**HTML Evolves**

**WebGL**

**Devices**

**Audio Data API**

**Video Improvements**

**Touchscreen Device Events**

**Peer-to-Peer Networking**

**Ultimate Direction**

**Best Practices for a Faster Web App with HTML5**

Use web storage in place of cookies

Use CSS Transitions instead of JavaScript animation

Use client-side databases instead of server roundtrips

JavaScript improvements lend considerable performance advantages

Use cache manifest for live sites, not just offline apps

Enable hardware acceleration to enhance visual experience

For CPU-heavy operations, Web Workers deliver

HTML5 Form attributes and input types

Use CSS3 effects instead of requesting heavy image sprites

WebSockets for faster delivery with less bandwidth than XHR

HTML5 Design Patterns

1. Main element design pattern
2. Navigation design pattern
3. Section design pattern
4. Details and summary design pattern
5. Figure and figcaption design pattern

|  |  |
| --- | --- |
| **Pre-requites for Participants** | |
| Tools | 1. Install NodeJS 4.4.x LTS from <https://nodejs.org/en/download/> 2. Google Chrome Version 50 or later 3. Chrome Plug-ins (Live HTTP Header, Postman) 4. Git (install from <https://git-scm.com/)> 5. Visual Studio Code |
| IDE | Eclipse Neon |
| Developer Skill | Basic JavaScript, HTML and/or jQuery, Web Technologies |
| Internet | Course need fulltime internet connectivity for Developers. Trainer shall use DataCard or Company given WIFI connectivity |
| Firewall | No firewall blocking. Let trainer know lab setup 2 days in advance. Trainer can inspect the system through live meet/team viewer/web ex |
| System | Laptop/Desktop PCs with one of the operating system   1. Windows 7.0 or later 2. Mac OS 110.11 3. Ubuntu Linux Desktop 15.10 or later |
| Administrator Account | All the systems must have Administrator privileges for developers attending training |

Below are deliverables for Angular 2.0 Advanced Training.

|  |  |
| --- | --- |
| **Deliverables** | |
| Presentations | All course presentations as PDF documents at end of the day |
| Example Source Code | All Working Example source codes at end of the day |
| Instructor Demo Source Code | Source code work out by instructor by end of the training |
| Feedbacks | Feedback forms to measure training effectiveness at end of the day |
| Assignments | Assignments for participants at the end of the day |
| Starter Templates | Starter Templates and Files shall be provided to participant for hands-on though github |

|  |  |
| --- | --- |
| **Day 1** | |
| Topic |  |
| Introduction | Course Objective  Angular 2 Introduction  Single Page Application  MVC Pattern  Single Page vs Multi-page application  Introduction to Components |
| Environment Setup | Node.js  HTTP RESTful API Server for Workshop  Setup Gulp  Setup up NPM project  Setup Karma  Visual Studio Code  Chrome remote debugging options  Debugging with Visual Studio Code  Angular 2 cli introduction and features |
| Angular 2 Get Started | Minimal Angular 2 Application with Components, List Page, Detail Page, Services  HTTP backend REST API Communication with GET method  File Structure for large Angular Application |
| Type Script &  ECMA 2016/ES6 | Introduction to Type Script, data types  Template String  Block Scope, Constant, Variables  TypeScript Classes, Inheritance, Interfaces  TypeScript modules, export, import  TypeScript static type checking  ES6 features, arrow functions  TypeScript Compiler  Typings, DefinitelyTyped, TypeScript Definition Manager |
| Angular 2 Components | Creating Components  Providers  Component selector  Component Templates  Component Styles  Property Binding [One way, Two way]  Event Binding  Get data into Component through Input  Subscribing to components events through output  Nested Component  Deeper Nesting of Components  Components Events  Component Life Cycle  Accessing Other Components |
| Directives | Attribute Directives (NgStyle, NgClass)  Structural Directives (NgIf, NgFor, NgSwitch) |
| Angular Templates | Template Syntax  Function Calls  Property Binding Syntax  Two Way Binding  One way Binding  Interpolation  textcontent  Syntatic Sugars in Angular 2  bing-prop, on-event, bindon-twowayprop |
| Pipes | Pipes Introduction  Creating custom Pipes  Passing Arguments to Pipes  Registering Pipes  Chaining Pipes  Http Web Service Calls with Pipes |
| Services | Implementing Services in Angular  Service Dependency  Dependency Injection  Injectable decorators  Provider definition  Provider configuration at Angular Bootup  Http Web Service calls with Services |

|  |  |
| --- | --- |
| **Day 2** | |
| Topic |  |
| Observables | Introduction to Reactive Extension (RxJS) Library  What is Reactiveness  Observables  Subscription  Events  Streaming in Observables  Http Web Service with Observables |
| Promises | Promises Introduction  Custom Promises  Promises vs Observable  Using HTTP Web Services with Promises |
| Custom Directives | Creating Custom Attribute Directives  Creating Custom Structural Directives  Event handlings in Custom directives  Using Input |
| Forms | Form Controls  Control Groups  Form Validation  Form Builder  Form Builder Validation  Custom Validation |
| Routing | Route Introduction  Configuring Routes  Redirection  Linking  Passing values/parameters between Routes  Router Outlet  Nested Routing /Child Routes  Handling Query Parameters |
| HTTP | Angular 2 HTTP  HTTP backend REST API/Web API Communication with GET, POST, PUT, DELETE methods using Angular 2 HTTP  HTTP GET, POST, PUT and DELETE  URL Query Parameters  Promises  HTTP with Observables  Cross Origin Request/CORS Headers  Cross Origin Requests/JSONP |
| Application Architecture, Component Tree, Bootstraping | Angular tree of components  Angular Observables  Angular Immutables  Angular 2 bootstrapping process |
| Gulp | Introduction to Gulp  Gulp task, watch, src and dest apis  Gulp streaming  Gulp Best Practices |

|  |  |
| --- | --- |
| **Day 3** | |
| Topic |  |
| TypeScript  Advanced | TypeScript Custom Decorator  Property Decorator  Class Decorator  Parameter Decorators |
| HTML5  Storage | Introduction to localStorage  Introduction to sessionStorage  Security with HTML5 storage  Cookie fundamentals |
| Web Workers | Web Threading  Web Workers |
| Unit Testing Angular 2 | Test Driven Development  Introduction to Jasmine  Introduction to Karma  Jasmine Test Suites, Specs, Expectations  Jasmine Matchers  Jasmine Custom Matchers  Setup Injection, Environment for Angular 2 testing  Testing a Components  Testing a Service  Testing with HTTP  Mocking HTTP with MockBackend  Testing Directives  Testing Pipes  Testing Routes (Introduction) |
| SASS | Working with SASS  Gulp and SASS  Generate CSS from SASS |
| Module Loaders | Introduction to Webpack  Introduction to SystemJS  Loading Angular Application with WebPack  Loading Angular Application with SystemJS |
| Other useful libraries | 1. Lodash 2. RxJS 3. Redux, State Management 4. ImmutableJS |
| Setup Release Environments | Gulp  Minification  SASS/CSS  Release Build  Source map generation  Managing, versioning files  Caching Files for performance |