

## Angular Training Course Outline

---

### Day 1 Start

---

- Introduction to HTML/CSS
- Introduction to HTML
- HTML By Examples
  - Example 1: Basic Layout of an HTML Document
  - Example 2: Lists and Hyperlinks
  - Example 3: Tables and Images
  - HTML Template
  - HTML Document Validator
  - Debugging HTML
- HTML Basics
  - HTML Tags, Attribute and Elements
  - The Basic Document Skeleton
  - HTML Comment `<!-- ... -->`
  - Block Elements vs. Inline Elements
  - Block Elements
  - Inline Elements - Logical Style vs. Physical Style
  - Lists
  - Tables
  - Anchors and Hyperlinks
  - Images
  - Hands on: -
    - Created forms
    - Sample application with home, about us and contact us navigation
- Introduction to CSS
- CSS By Examples
  - Example 1: CSS Syntax and Tag-Selectors
  - Example 2: CSS Class-Selector, ID-Selector with `<div>` and `<span>`
  - CSS template
  - Validating CSS
  - Inspecting Element's Styles
- CSS Basics
  - CSS Syntax
  - Inline, Internal and External Styles
  - Resolving Conflicting Rules
  - How to Use CSS for Styling HTML Document?
  - Types of CSS Selectors

- Style Properties
- Color Properties
- Length Measurements
- Box Model - Margin, Border, Padding and Content Area
- Font Properties
- Text Properties
- Background Properties
- List Properties
- Table Properties
- Image Properties
- Hands on: -
  - Used CSS properties in previously created project
  - Created student information form
- More HTML
  - Image Maps
  - Frames
  - Embedding Programs
  - HTML Global Attributes
  - HEAD Section's Tags
- More CSS
  - Positioning the HTML Block-Level Elements
  - Floating an Element Left or Right
  - Other Miscellaneous Properties

---

## *Day 2 Start*

- JavaScript Basics
  - Evolution of JavaScript
  - JavaScript Syntax
  - Datatypes and Variables in JavaScript
  - Strings and Arrays in JavaScript
  - Type Conversions in JavaScript
  - Operators in JavaScript Spread Operator
  - Conditional Statements in JavaScript
  - Looping Statements in JavaScript
  - Hands On:
    - String Manipulations in JavaScript Arrays in JavaScript
    - Type Conversions in JavaScript Operators in JavaScript
    - Conditional Statements in JavaScript

- Looping Statements in JavaScript
- Objects and Event Handling
  - JavaScript Objects Object Properties
  - JavaScript Object Methods
  - JavaScript Object Constructors
  - Object Prototypes
  - Class Methods Inheritance
  - Browser Object Model (BOM)
  - BOM Location and History Object
  - HTML Document Object Model (DOM)
  - DOM Events and Event Listeners
  - HTML Form Validations
  - Debugging using ChromeDev Tools
  - Hands On:
    - JavaScript Objects
    - JavaScript Object Constructors JavaScript Classes
    - BOM Objects
    - DOM Event Handling
    - HTML Form Validations
- ES6 & Typescript – Essentials
  - Introduction
  - Types & Strict Typing
  - Built-in types
  - Interfaces & Type Alias
  - Classes & Modules
  - Constructors
  - Inheritance
  - Annotations
  - Tuples
  - Nuance of 'this' keyword
  - Fat Arrow Functions
  - Template Strings
  - Generic
  - Hands on: -
    - Problem statements of Teacher and student
    - Problem statement of Photobook
    - Problem statement of Shape

---

### *Day 3 Start*

- Introduction to Angular
  - Angular Architecture & Building Blocks

- Architecture Overview
- Bootstrap our project with Angular-cli
- Architecture & Walkthrough
- Bootstrapping our App with Angular-cli
- Adding Structure for our App
- Introducing component
- Component Decorator Properties
- Introducing Template
- Interpolation
- Introducing Angular Module
- Decorator Properties
- Registering our component
- Understanding the Bootstrapping process
- Root Vs Feature components & modules
- Hands On: -
  - Creating starter project
  - Created project on customer and supplier
- Models and Directives
  - Creating Model & Adding data
  - Working with Bindings
  - Introducing Inbuilt directives
  - Property & Event bindings
  - Attribute Vs Structural directives
  - Working with Inbuilt directives
  - Inverse Data flows & Two-way data bindings
  - Filtering our data
  - Hands On: -
    - Added bindings into project

---

## *Day 4 Start*

- Form Validations
  - Angular Forms & Validation
  - Adding a Angular Form to our App
  - Adding Validation Criteria
  - Adding Styles for Error Handling
  - Throwing Error Notifications
  - Form Control & Group Validations
  - Reactive Forms/Model driven Forms
  - Delegating Template features to Model
  - Adding our Feature component
  - Importing the Reactive Forms module

- Programmatically Generate Form controls & validation
- Handling Errors & Custom validators
- Introducing Form Builder
- Driving the Form with Form Builder features
- Hands On: -
  - Added form validation into project
  - Use bootstrap with angular
  - Create customer application with bootstrap
- Routing
  - Breaking our App into Components & Modules
  - Configuring & Integrating Bootstrap.js
  - Working with Components, Services & Routing
  - Preparing our Feature components & template
  - Adding Property & Event bindings
  - Preparing other feature components
  - Understanding Route elements
  - Navigation with Sub routing features
  - Hands On: -
    - Added routing into the project
    - Create sub modules and added their sub routes
- Understanding Dependency Injection
  - What is DI
  - Dependency Injection with Services
  - Role of Injector service
  - Hands On: -
    - Created Logger in application
- Http Calls in Angular
  - Introducing HTTP client & injection
  - REST Data with Observables & Operators
  - Facilitating CRUD operations
  - Adding REST API
  - Introducing Service
  - Sending Form to Server
  - REST API Interactions
  - Hands On: -
    - Launch Dummy server
    - Send data to dummy server by adding http calls
- Rxjs
  - RXJS Core concepts
  - Observables & Observer
  - Hands On: -

- Create Employee app and use Rx js
- Event Emitters
  - Gather dynamic data with @input
  - Introduction to Event emitter
  - Throwing events with @output
  - Creating Shared Modules
  - Hands On: -
    - Created Grid-Ui common component
    - Created shared module for Grid-Ui to reuse in multiple components
- Introducing Pipes
  - Using Inbuilt Pipes
  - Creating a Custom Pipe
  - Pure Vs impure pipes
  - Hands On: -
    - Use Inbuilt Pipes
    - Create Custom Pipe

---

### *Day 5 Start*

- Advanced Routing
  - Lazy loading an Angular module
  - Dynamic load components via Routing Securing routes with Auth guard
  - Restricting Access with Authentication with JWT (JSON web token)
  - Preparing the feature components & services
  - Hand On: -
    - Create public private events project and use Dot net server as the backend
- Angular Hooks & Component Intercommunication
  - Understanding Component Lifecycle process
  - Using the ngOninit & ngOnchange
  - Using the ngDoCheck hook
  - Using the ngDestory Hook
  - Hands On: -
    - Create sample project in that understand the concept of hooks
- Unit Testing
  - Introduction to Jasmine and Karma
  - Benefits of Unit Testing
  - The 'AAA' pattern of testing
  - Creating our Test suite & Write test
  - Working with Jasmine matchers
  - Hands On: -
    - Create sample project in that understand the Karma and Jasmine in that
- Best Practices & Performance

- Utilizing Angular-cli, Typescript & ES6 features Angular Routing
- Modular development for Clean Structure
- Utilizing Lazy loading for reducing initial load times
- Breaking down an Application in Components
- Clean-code Practices to easily locate file & Code structures
- Hands On: -
  - Implement while doing labs