



Angular: Introduction to Angular 12

Course ID #: 7000-624-ZZ-Z

Hours: 21

Course Content

Course Description:

In this course, you will cover all the essential topics and is designed to get students up and running with Angular 12 development in the shortest amount of time.

Developers and Application Designers as well as those who are tasked with supporting Angular 12 applications will gain valuable knowledge required for success in their job roles while taking this course.

Get started with the new Angular 12 today!

Course Objectives:

- Develop single page Angular applications using Typescript
- Set up a complete Angular development environment
- Create Components
- Component Templates
- Inter Component Communication
- Template Driven Forms
- Reactive Forms
- Services and Dependency Injection
- HTTP Client
- Pipes and Data Formatting
- Introduction to Single Page Applications
- The Angular Component Router

Prerequisites:

Web development experience using HTML, CSS and JavaScript is required to get the most out of this Angular 12 course.

Knowledge of the browser DOM is also useful. Prior Angular experience, with AngularJS or the current version of Angular, is not required.



Angular: Introduction to Angular 12

Course ID #: 7000-624-ZZ-Z

Hours: 21

Topics:

Lesson 1: Introducing Angular

- What is Angular?
- Central Features of the Angular Framework
- Appropriate Use Cases
- Building Blocks of an Angular Application
- Basic Architecture of an Angular Application
- Installing and Using Angular
- Anatomy of an Angular Application
- Running the Application
- Building and Deploying the Application
- Summary

Lesson 2: Introduction to TypeScript

- Programming Languages for Use with Angular
- TypeScript Syntax
- Programming Editors
- The Type System – Defining Variables
- The Type System – Defining Arrays
- Type in Functions
- Type Inference
- Defining Classes
- Class Methods
- Class Constructors
- Class Constructors – Alternate Form
- Interfaces
- Working with ES6 Modules
- Visibility Control
- var vs let
- Arrow Functions
- Arrow Function Compact Syntax
- Arrow Function and Caller Context
- Template Strings
- Generics in Class
- Generics in Function
- Generics - Restricting Types
- TypeScript Transpilation
- Summary

Lesson 3: Components

- What is a Component?
- An Example Component
- Creating a Component Using Angular CLI
- The Component Class
- The @Component Decorator
- Registering a Component to Its Module
- Component Template
- Example: HelloComponent Template
- Example: The HelloComponent Class
- Using a Component
- Run the Application
- Component Hierarchy
- The Application Root Component
- The Bootstrap File
- Component Lifecycle Hooks
- Example Lifecycle Hooks
- CSS Styles
- Summary

Lesson 4: Component Templates

- Templates
- Template Location
- The Mustache {{ }} Syntax
- Setting DOM Element Properties
- Setting Element Body Text
- Event Binding
- Expression Event Handler
- Prevent Default Handling
- Attribute Directives
- Apply Styles by Changing CSS Classes
- Example: ngClass
- Applying Styles Directly
- Structural Directives
- Conditionally Execute Template
- Example: ngIf
- Looping Using ngFor
- ngFor Local Variables
- Manipulating the Collection
- Example - Deleting an Item
- Item Tracking with ngFor



Angular: Introduction to Angular 12

Course ID #: 7000-624-ZZ-Z

Hours: 21

- Swapping Elements with ngSwitch
- Grouping Elements
- Template Reference Variable
- Summary

Lesson 5: Inter Component Communication

- Communication Basics
- The Data Flow Architecture
- Preparing the Child to Receive Data
- Send Data from Parent
- More About Setting Properties
- Firing Event from a Component
- @Output() Example - Child Component
- @Output() Example - Parent Component
- Full Two Way Binding
- Setting up Two Way Data Binding in Parent
- Summary

Lesson 6: Template Driven Forms

- Template Driven Forms
- Importing Forms Module
- Basic Approach
- Setting Up a Form
- Getting User Input
- Omitting ngForm Attribute
- Initialize the Form
- Two Way Data Binding
- Form Validation
- Angular Validators
- Displaying Validation State Using Classes
- Additional Input Types
- Checkboxes
- Select (Drop Down) Fields
- Rendering Options for Select (Drop Down)
- Date fields
- Radio Buttons
- Summary

Lesson 7: Reactive Forms

- Reactive Forms Overview
- The Building Blocks
- Import ReactiveFormsModule
- Construct a Form
- Design the Template

- Getting Input Values
- Initializing the Input Fields
- Setting Form Values
- Subscribing to Input Changes
- Validation
- Built-In Validators
- Showing Validation Error
- Custom Validator
- Using a Custom Validator
- Supplying Configuration to Custom Validator
- Sub FormGroups - Component Class
- Sub FormGroups - HTML Template
- Why Use Sub FormGroups
- Summary

Lesson 8: Services and Dependency Injection

- What is a Service?
- Creating a Basic Service
- The Service Class
- What is Dependency Injection?
- Injecting a Service Instance
- Injectors
- Injector Hierarchy
- The Root Injector
- Registering a Service with a Component's Injector
- Where to Register a Service?
- Dependency Injection in Other Artifacts
- Providing an Alternate Implementation
- Dependency Injection and @Host
- Dependency Injection and @Optional
- Summary

Lesson 9: HTTP Client

- The Angular HTTP Client
- Using The HTTP Client - Overview
- Importing HttpClientModule
- Service Using HttpClient
- Making a GET Request
- What does an Observable Object do?
- Using the Service in a Component
- The PeopleService Client Component
- Error Handling
- Customizing the Error Object



Angular: Introduction to Angular 12

Course ID #: 7000-624-ZZ-Z

Hours: 21

- Making a POST Request
- Making a PUT Request
- Making a DELETE Request
- Summary

Lesson 10: Pipes and Data Formatting

- What are Pipes?
- Built-In Pipes
- Using Pipes in HTML Template
- Chaining Pipes
- Internationalized Pipes (i18n)
- Loading Locale Data
- The date Pipe
- The number Pipe
- Currency Pipe
- Create a Custom Pipe
- Custom Pipe Example
- Using Custom Pipes
- Using a Pipe with ngFor
- A Filter Pipe
- Pipe Category: Pure and Impure
- Pure Pipe Example
- Impure Pipe Example
- Summary

Lesson 11: Introduction to Single Page Applications

- What is a Single Page Application (SPA)
- Traditional Web Application
- SPA Workflow
- Single Page Application Advantages
- HTML5 History API
- SPA Challenges
- Implementing SPA's Using Angular
- Summary

Lesson 12: The Angular Component Router

- The Component Router
- View Navigation
- The Angular Router API
- Creating a Router Enabled Application
- Hosting the Routed Components
- Navigation Using Links and Buttons
- Programmatic Navigation
- Passing Route Parameters
- Navigating with Route Parameters
- Obtaining the Route Parameter Values
- Retrieving the Route Parameter Synchronously
- Retrieving a Route Parameter Asynchronously
- Query Parameters
- Supplying Query Parameters
- Retrieving Query Parameters Asynchronously
- Problems with Manual URL entry and Bookmarking
- Summary

Lab Exercises

- Lab 1. Introduction to Angular
- Lab 2. Introduction to TypeScript
- Lab 3. Introduction to Components
- Lab 4. Component Template
- Lab 5. Create a Photo Gallery Component
- Lab 6. Template Driven Form
- Lab 7. Create an Edit Form
- Lab 8. Reactive Form
- Lab 9. Develop a Service
- Lab 10. Develop an HTTP Client
- Lab 11. Use Pipes
- Lab 12. Basic Single Page Application Using Router
- Lab 13. Build a Single Page Application (SPA)

Register for this class by visiting us at:

www.tcworkshop.com or calling us at 800-639-3535