

## Developing Web Applications Using AngularJS

**Duration:** 4 Days (*Face-to-Face & Remote-Live*), or 28 Hours (*On-Demand*)

**Price:** CDN\$2,775 (*Face-to-Face & Remote-Live*), or CDN\$1,995 (*On-Demand*)

**Discounts:** We offer multiple discount options. [Click here](#) for more info.

**Delivery Options:** Attend face-to-face in the classroom, [remote-live](#) or [on-demand training](#).

### Students Will Learn

- Referencing AngularJS in a Web page
- Building Single Page Applications using AngularJS
- Integrating forms with AngularJS
- Organizing code using modules
- Injecting dependencies into a Module
- Configuring routes with `ngRoute` and `$routeProvider`
- Refactoring reusable code into factories and services
- Defining business logic using controllers
- Performing Ajax requests with the `$http` and `$resource` services
- Communicating with RESTful web services
- Reading data from remote servers using `$http`
- Providing new behaviors to HTML using AngularJS directives
- Enhancing the UI using animations and Bootstrap
- Extending AngularJS with custom directives, filters and services

### Course Description

**This course introduces AngularJS 1.x. Students interested in Angular versions 2 through 6 should take the [Developing Web Applications Using Angular](#) course instead.**

This hands on programming course provides a thorough introduction to the AngularJS JavaScript Framework. Attendees will learn the fundamental skills necessary to build Web Applications using AngularJS and the MV\* (Model View Whatever) design pattern. Topics include creating controllers, using scope to manage data, designing views/templates, routing, data binding and filters, applying directives, as well as form integration and validation. Students will also use AngularJS' built-in services to communicate with RESTful web services and provide CRUD database operations.

Students will learn how to use CSS animations and Bootstrap to enhance the UI as well as learn to employ third-party components such as modal dialogs ("modals"), progress bars

and navbars. In addition, students will learn to extend AngularJS with custom directives, services and filters.

Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency.

## Course Prerequisites

Knowledge of HTML, CSS and JavaScript equivalent to attending the [Website Development with HTML5, CSS and Bootstrap](#) and [JavaScript Programming](#) courses. Knowledge of jQuery is helpful, but not required.

## Course Overview

### Overview of AngularJS

- Features and Benefits of AngularJS
- MV\* Design Pattern Overview
- Downloading AngularJS
- Referencing AngularJS Using a CDN
- The AngularJS Digest Cycle
  - `$watch`
  - `$apply`
- Overview of jqLite
- Architecting an Application with AngularJS

### Building Modules in AngularJS

- Overview of Modules
- Organizing Code Using Modules
- Defining an Application with `angular.module()`
- Bootstrapping an Application with `ngApp`
- Managing Dependencies with Dependency Injection
- Dealing with Minification Issues

### Using Built-In AngularJS Directives

- AngularJS Directive Overview
- Behavior-Driven Directives
  - `ngChange`
  - `ngClick`
  - `ngSubmit`
- DOM-Driven Directives
  - `ngShow/ngHide`
  - `ngIf/ngSwitch`
  - `ngSrc/ngHref`

### Creating Single Page Applications (SPAs)

- Single Page Application Model
- AngularJS Programming Model for SPAs
- Building the Shell HTML
  - Designing a Common UI
  - Defining Navigation
- Managing Feature Containers
  - Designing Partial Views
  - Organizing Business Logic in Controllers
  - Programming the Data Model
- Marrying Views, Controllers and URLs

### Creating Controllers

- Defining a Controller with `module.controller()`
- Creating Properties and Methods in a Controller
- Using the `ngController` Directive
- Defining `$scope` in a Controller
- Avoiding Scope Conflicts Using "Controller As"

### Using AngularJS Services

- Built-In Services and Factories
- Comparing Factories and Services
- Using the `$http` and `$resource` Services
  - Connecting to RESTful Web Services
  - Performing CRUD Operations
  - Retrieving JSON data Asynchronously

- Data-Driven Directives
  - `ngBind`
  - `ngInit`
  - `ngModel`
  - `ngClass/ngStyle`

## AngularJS Routing

- Overview of Routing
- Configuring Routes with `$routeProvider`
- Defining Route Parameters
- Designing a Shell Page with `ngView`
- Defining Templates
- Mapping Route URLs to Templates

## Integrating Forms in AngularJS

- Using the Form Controller
- Binding to Input Fields
- Toggling Control State Using Directives and Expressions
  - `ngChecked`
  - `ngShow`
  - `ngDisabled`

## Validating Forms in AngularJS

- Setting HTML5 Validation Attributes
  - `required`
  - `pattern`
  - `min/max`
- Using AngularJS Properties in Expressions
  - `$error`
  - `$dirty`
  - `$invalid`
- Styling the Form with Angular CSS classes
  - `ng-valid`
  - `ng-invalid`
  - `ng-dirty`
  - `ng-pristine`

## Animating AngularJS Applications

- CSS3 Animations
  - CSS3 Transitions
  - CSS3 Animations
- Using the `$animate` Service
- Adding Animations on Enter, Leave and Move Events

- Logging Errors with `$log`
- Using the `$location` and `$window` Services
- Deferred and Promise API

## Data Binding

- Overview of Data Binding
- The `ngModel` Directive
- Displaying Sets of Data with `ngRepeat`
- Applying Filters
  - `currency`
  - `date`
  - `orderBy`
  - `filter`
- AngularJS Expressions

## Extending AngularJS

- Defining Custom Directives
  - Shared and Isolate Scope
  - Defining One- and Two-Way Data Bindings
  - Local Scope Properties (`@`, `=` and `&`)
- Defining Custom Services
  - Refactoring Code into a Service
  - Defining Properties and Methods
  - Returning a Service
- Defining Custom Factories
  - Refactoring Code into a Factory
  - Defining Properties and Methods
  - Returning a Factory Object
- Defining Constants and Values
- Defining Custom Filters

## Integrating Third Party Frameworks

- Creating a Responsive UI with the Bootstrap Grid System
- Applying Bootstrap CSS Classes
- Using Bootstrap Components
  - Progress bar
  - Modal

- Injecting the `ngAnimate` Module
- Tabs
- Navbar
- Using AngularStrap and AngularUI Components

Hands On Technology Transfer  
The Best Way to Transfer Technology Skills

1 Village Square, Suite 8  
14 Fletcher Street  
Chelmsford, MA 01824

Copyright © 2021 Hands On Technology Transfer, Inc.