www.logicalimagination.com
800.657.1494

# Angular Introduction

**Course #:** AN-400 **Duration:** 3 days

## Prerequisites

Student should have a thorough understanding of HTML, CSS and JavaScript.

## Details

From prototype through global deployment, Angular delivers the productivity and scalable infrastructure that supports Google's largest applications. Learn one way to build applications with Angular and reuse your code and abilities to build apps for any deployment target: web, mobile web, native mobile, and native desktop.

In this course, you will learn the basics of Angular, including components, routing, Angular CLI, data binding, pipes, directives, service and dependency injection, HTTP, and template-driven forms and validation. You will also get a brief introduction to TypeScript.

This course has been updated for Angular 19.

## Software Needed

- One or more browsers
- NodeJS
- A text editor (we strongly recommend Microsoft's Visual Studio Code, which is free)

## Outline

Angular Introduction

- **Introduction**
  - What is Angular?
  - Angular vs. AngularJS
  - Scope and Goals of Angular
  - Performance
  - Future of Web Development
  - Mobile Device Support
  - Ease of Use
  - Architecture (Big Picture / Concepts)
  - Architecture
  - Setting up Your Environment
  - A Basic Angular Application
- **TypeScript Introduction**
  - Programming Languages for Angular
  - How TypeScript Works
  - Transpiling TypeScript
  - TypeScript Syntax

    - The Type System
    - Classes
    - Interfaces
    - Arrow Functions
    - Modules
    - Generics

- **Components**
    - What is a component?
    - Angular Modules
    - Developing a Simple Component
    - HTML Templates
    - Models
    - Nesting Components

- **Routing**
    - Routing and Navigation
    - Router Terminology
    - Router Setup
    - Programmatic Navigation
    - Passing Data While Navigating
    - Using Route Parameter Values

- **Angular CLI**
    - Installation
    - Usage
    - Generating a New Project
    - Generating Components, Directives, Pipes, and Services
    - Generating a Route
    - Creating a Build
    - Build Targets and Environment Files
    - Bundling

- **Data Binding**
    - Unidirectional Data Flow
    - Binding Syntax
    - Template Expressions
    - One-Way Binding
    - Event Binding
    - Template Statements
    - Two-Way Binding

- **Pipes**
    - What are Pipes?
    - Using Pipes
    - Built-in Pipes
    - Pipe Syntax
    - Chaining Pipes
    - Creating a Custom Pipe
    - Pure vs Impure Pipes

- **Directives**
    - What is a Directive?
    - Kinds of Directives
    - Attribute Directives
    - Setting Attributes Dynamically
    - Structural Directives
    - Adding and Removing Elements Dynamically

- **Service and Dependency Injection**
    - What is a Service?
    - Service Example
    - What is Dependency Injection?
    - Why do I need Dependency Injection?
    - Registering a Service
    - Injecting a Service
    - @Injectable Decorator
    - Multiple Service Instances

- @Optional and @Host Decorators

- **HTTP**
  - The Angular HTTP Client
  - Setting Up the Root Component
  - HTTP Client Using Promises
  - Observables and Reactive Programming
  - HTTP Client in Using Observables
  - Enhancing Services with .map() and .catch()

- **Template-Driven Forms and Validation**
  - Template-Driven Forms
  - Binding Input Fields
  - Accessing the Form Object
  - Binding the Submit Event
  - HTML5 Validation
  - Angular Validation
  - Defining Validation Directives
  - Tracking Change State of Form Controls
  - Displaying Validation State
  - Disabling Submit When Invalid