

# Angular 7.2 Fundamentals

## Tour, CLI, Building Components, Directives, Databinding, HTTP Client, Forms, Bootstrapping

Angular 7.2 is the most advanced web client framework available for production use. It provides a very solid foundation for development of modern web applications and is increasingly being selected by start-ups, cloud vendors and global enterprises for their large scale solutions with demanding needs and tight development time frames.

Angular is a vibrant open source project and is built by a large team from Google and elsewhere. They work to further evolve the framework and they regularly add interesting enhancements. This course covers the latest

Angular 7.2 that can be used to build web desktop, web mobile, native desktop, native mobile and even server applications (as it has plenty of non-UI functionality).

Angular is a big project, best approached in stages by developers wishing to start building apps. Before this course app developers should learn TypeScript (Angular itself is written in TypeScript, as are most Angular apps). Then app developers should attend this course as their first contact with Angular, optionally followed by our *Advanced Angular 7.2* course, which looks in more detail at some specialist Angular topics.

<b>Contents of One-Day Training Course</b>	
<p><b>Target Audience</b> Developers seeking to quickly get up to speed with the best web framework in the world.</p> <p><b>Prerequisites</b> Developers experienced with the TypeScript language and web programming in general.</p> <p>No previous Angular experience required.</p> <p>All demos and lab exercises will be in TypeScript.</p>	<p><b>Angular Framework Tour</b> Collection of packages that work together to deliver a wonderful web framework Overview of how it works Introduction to each major module Many parts to an Angular app</p> <p><b>Angular CLI 7.2</b> Command line interface to creating, building, serving and testing Angular apps Angular CLI automates the creation of a good boilerplate source tree for your app that you can later enhance What you might like to customize (e.g. versions in package.json) CLI Prompts (new in Schematics in v7)</p> <p><b>Building Components</b> Exploring how we build components Event handling – firing and listening Input and output properties Metadata for components</p> <p><b>Angular Template Syntax</b> Enhancing HTML syntax with custom directives and expressions Which HTML concepts not permitted Interpolation Expression syntax Attribute directives ngModel</p> <p><b>Structural Directives</b> ngFor / ngIf ngSwitch Microsyntax ng-template / ng-container</p> <p><b>Databinding in-depth</b> Event, Property, Attribute, Class, Style, Two-Way</p> <p><b>HTTP Client</b> How to use the various HTTP request types Role of in-memory-web-api for testing Use of services in Angular app architecture Asynchronous stream of events (objects) delivered to your components (RxJS)</p> <p><b>Animation</b> Transitions States &amp; Triggers</p> <p><b>Advanced Components</b> Styling for Angular components Lifecycle hooks Pipes Deeper look at how you build services</p> <p><b>Introduction to Forms</b> Template-driven forms Error handling Change tracking Structuring form handling code Reactive forms</p> <p><b>Validation</b> Validating forms input Correctly displaying error information Built-in and custom validators Role of CSS in reflecting control status</p> <p><b>Bootstrapping</b> How an Angular app bootstraps Intro to how rendering works Use of platform-browser</p>