

Angular ist ein TypeScript-basiertes Front-End-Webapplikationsframework. Es wird von einer Community aus Einzelpersonen und Unternehmen, angeführt durch Google, entwickelt und als Open-Source-Software publiziert.

Ihr Nutzen

In diesem Seminar lernen Sie Angular mit Node.js / .NET Core einzusetzen und damit "Single Page Applications" (SPA's) zu erstellen. Sie werden dabei alle Bereiche von Angular inklusive Responsive UI Design, Security und Testing behandeln.

Preis pro Teilnehmer

EUR 2250,- exklusive der gesetzlichen MwSt.

Seminardauer

4 Tag(e)/Day(s)

Seminarinhalte

Tag 1

- * Das Angular Ecosystem
 - TypeScript in the Angular Ecosystem
 - Databinding & Completing Basic Tasks
 - Implementing Nested Components
 - Routing & Dependency Injection
 - Designing a responsive User Interface
 - Forms Design & Validation
 - Reactive Programming using Signals & RxJS
 - Managing client State & Sharing Events
 - Testing Angular Applications
 - Securing & Publishing Angular
- * Einführung in Angular
 - Angular Introduction & Technology Stack
 - Angular CLI Essentials
 - Debugging Angular
 - Bootstrapping & Configuration
 - Maintaining & Updating projects
- * TypeScript im Angular Ecosystem
 - TypeScript Overview, ECMA Script Standards
 - Types, Variables, Literal Types, Functions
 - Objects, Classes, Interfaces
 - Immutability, Cloning & Object Composition
 - Async Operations, Observables & Signals
 - Calculated Signals & Effects

Tag 2

- * Databinding & Completing Basic Tasks
 - Components & Dependency Injection
 - Services & Data Request
 - Expressions, Templates & Directives
 - String Interpolation, Attribute- & Event-Binding, Two-Way Binding
 - Built-in Control Flow & Deferrable Views
 - Data binding Observables & Signals
 - Built in and Custom Pipes & Directives
- * Implementing Nested Components
 - Benefits of Nested Components

Voraussetzungen

C# Programming~7644

Kenntnisse von HTML, CSS und Java Script

Hinweise

Version: 2019

- Container vs Presentational Components
- Data binding & Events using @Input & @Output
- Nesting using Signal inputs & model outputs
- View Child, View Children & Signal Queries

Tag 3

- * Routing & Dependency Injection
 - Register Providers & Dependency Injection
 - Routing & Navigation Basics
 - Working with Parameterized Routes
 - Component Input Bindings
 - Modules vs Standalone Components
 - Modules Use Cases & Lazy Loading
 - Route Guards & Data Preloading
- * Designing a responsive User Interface
 - Implementing a CSS Reset
 - Global & Component Styles
 - Responsive Web Design using Media Queries
 - Layout using Flexbox, CSS Grid and Areas
 - Angular Material Overview
 - Common controls: Tables, Dialogs & Form Controls ...
 - Angular CDK & 3rd Party Components (Virtual Lists, Drag & Drop, Markdown, ...)

* Reactive Forms Design & Validation

- Forms Introduction
- Template Driven Forms vs Reactive (Model Based) Forms
- FormBuilder & Typed Forms (Nullability, Nested Objects, Arrays)
- FormControl, FormGroups & FormArrays
- Form Validation Basics (Synchronous / Asynchronous)
- Custom- & Code-Based Validators
- Signals & FormControls

Tag 4

- * Reactive Programming using Signals & RxJS
 - What is Reactive Programming / Benefits
 - Imperative vs Reactive Programming Styles
 - Observable, Observer, Async pipe & Unsubscribing
 - Common RxJS Operators
 - Creating Observables & Casting to Observables

capturing mouse & DOM Events as Observables
Observable & Signal Interoperability

