



ANGULAR GİRİŞ EĞİTİMİ

3 GÜN



Digital Vizyon
Akademi

www.digitalvizyon.net



İçindekiler

Eğitim Hakkında.....	3
Neler Öğreneceksiniz?	3
Ön Koşullar	4
Kimler Katılmalı.....	4
Outline	5
TypeScript fundamentals.....	5
Reactive Programming with RxJS (Observables)	5
Async Requests to a REST API (HTTP)	5
Navigation and Routing	6
Forms and Validation.....	6
Using HttpClient to Update or Delete data on the server	6
Authentication.....	6
TypeScript fundamentals.....	6
Reactive Programming with RxJS (Observables)	6
Async Requests to a REST API (HTTP)	7
Navigation and Routing	7
Forms and Validation.....	7
Using HttpClient to Update or Delete data on the server	7
Authentication.....	7
TypeScript fundamentals.....	8
Reactive Programming with RxJS (Observables)	8
Async Requests to a REST API (HTTP)	8
Navigation and Routing	9
Forms and Validation.....	9
Using HttpClient to Update or Delete data on the server	9
Authentication.....	9

Eđitim Hakkında

Angular Giriş Eđitimi, web uygulamaları geliřtirmek isteyenlerin ya da temelleri öğrenmekte olanların tercih ettiđi deđerli bir kaynaktır. Bu eđitim, Angular framework'ünün temel kavramlarına ve özelliklerine odaklanır, ayrıca web uygulama geliřtirmede kullanılan genel prensipleri de ele alır.

Eđitimin ilk bölümünde, Angular'ın temel yapı taşları üzerinde duruyoruz. Bu yapı taşları arasında bileşenler, modüller, servisler ve direktifler yer alır. Katılımcılara, bu yapı taşlarının Angular uygulamalarının temelini nasıl oluşturduđunu detaylı bir şekilde anlatıyoruz. Öğrencilere, bu öğelerin Angular'ın gücü ve esnekliđi açısından neden önemli olduđu açıklanır.

Daha sonrasında, Angular ile bir web uygulaması geliřtirmenin temel adımlarını keřfederiz. Eđitim sırasında, katılımcılar temel form elemanları, veri bađlama ve HTTP isteklerini nasıl kullanacaklarını öğrenirler. Ayrıca, uygulamalar arası geçiřleri düzenlemek için Angular'daki yönlendirme (routing) özelliklerini de inceleyeceđiz.

Eđitimde, Angular'ın performans ve kullanıcı deneyimi açısından nasıl avantajlar sađladığına deđiniyoruz. Katılımcılar, sayfa yükleme hızını artırmak ve kullanıcı etkileřimini iyileřtirmek için basit optimizasyon tekniklerini öğrenirler. Temiz ve etkili kod yazmanın yanı sıra, Angular projelerinde çalışma süreçlerinin nasıl optimize edebileceđinizi de gösteriyoruz.

Son olarak, bu eđitimde, katılımcılar takım çalışması ve iřbirliđi hakkında bilgi edinirler. Angular projelerinin nasıl etkili bir şekilde yönetilebileceđi, kaynak kodunun nasıl sürdürülebilir ve okunabilir kılınacađı, ve versiyon kontrol sistemlerinin nasıl kullanılacađı anlatılır. Ayrıca, katılımcılar çeřitli geliřtirme araçlarına ve Angular ekosisteminde kullanılan en iyi uygulamalara ařına olurlar.

Amacımız, katılımcılara Angular kullanarak modern ve kullanıcı dostu web uygulamaları geliřtirmek için gereken bilgileri sađlamaktır. Eđitim aynı zamanda, katılımcıların Angular ile ilgili daha ileri konseptlere ve tekniklere geçiř yapmaları için sađlam bir temel oluşturur.

Angular Giriş Eđitimi, Angular ile web uygulamaları geliřtirmeye yeni bařlayanlar veya mevcut bilgilerini güncellemek ve derinleřtirmek isteyenler için harika bir bařlangıç noktasıdır. Öğrenciler, Angular'ın sunduđu potansiyeli anlayacak ve modern web geliřtirme dünyasına sađlam bir adım atmıř olacaklardır.

Neler Öğreneceksiniz?

Angular Giriş Eđitimi sırasında, řu konuları öğrenebilirsiniz:

- Angular framework'ünün temel kavramları ve prensipleri
- Angular component yapısı ve veri gösterme ve güncelleme
- Angular routing ve navigation sistemi
- Angular ile HTTP istekleri yapma
- Angular ile veri yönetimi ve form validasyonu



- Angular ile modüler ve esnek uygulama geliştirme
- Angular ile birlikte kullanılan diğer teknolojiler ve araçlar
- Angular ile uygulama performansının artırılması
- Angular ile uygulama güvenliğinin sağlanması

Bu konuları öğrenerek, Angular ile web uygulamalarının nasıl geliştirilebileceğini ve Angular'ın nasıl kullanılabileceğini anlayabilir ve Angular ile ilgili daha fazla bilgi ve beceri kazanabilirsiniz.

Ön Koşullar

Angular Giriş Eğitimi için ön koşullar şunlardır:

- HTML, CSS ve JavaScript temel bilgisi: Angular ile uygulama geliştirme, HTML, CSS ve JavaScript gibi web geliştirme dilleri ile yakından ilgilidir.
- Programlama deneyimi: Angular ile uygulama geliştirme, programlama deneyimi gerektirir. Öncelikle JavaScript dili hakkında temel bilgiye sahip olmanız önerilir.

Ek olarak, Angular Giriş Eğitimi'ne katılacak kişilerin aşağıdaki becerilerle de desteklenmesi önerilir:

- Proje yönetimi becerileri: Angular ile uygulama geliştirme, proje yönetimi becerilerinin kullanımını gerektirir.
- Kod okuma becerileri: Angular uygulamalarının geliştirilmesi sırasında, mevcut kodları okumak ve anlamak gerekir.
- Debugging becerileri: Angular uygulamalarının geliştirilmesi sırasında, hata ayıklama becerilerinin kullanımı gerekir.
- Veritabanı ve RESTful API'ler hakkında bilgi: Angular uygulamaları veritabanı ve RESTful API'ler ile iletişim kurar, bu konular hakkında da bilgi sahibi olmanız faydalı olacaktır.

Bu beceriler, Angular Giriş Eğitimi'ne katılacak kişilerin Angular ile uygulama geliştirme sürecinde daha başarılı olmasını ve eğitimi daha verimli bir şekilde tamamlamasını sağlayacaktır. Eğitimi almak isteyen kişilerin, HTML, CSS ve JavaScript dilleri hakkında temel bilgiye sahip olması ve programlama deneyimi bulunması ön koşullardır. Bu ön koşulların sağlanması, eğitim sürecinde daha verimli ve kolay bir şekilde öğrenmeyi mümkün kılacaktır.

Kimler Katılmalı

Angular Giriş Eğitimi, web geliştirme alanında çalışan veya çalışmak isteyen, HTML, CSS ve JavaScript dilleri hakkında temel bilgiye sahip, programlama deneyimi bulunan kişiler tarafından katılabilir. Ayrıca, proje yönetimi, kod okuma, hata ayıklama ve veritabanı/API konuları hakkında da bilgi sahibi olmak önerilir. Angular ile uygulama geliştirme konusunda ilgi duyan herkes, Angular Giriş Eğitimi'ne katılabilir.



Outline

TypeScript fundamentals

- The TypeScript compiler
 - Type safety with type annotation
 - Classes and Interfaces
 - Generics
- Setup and Tooling
- Getting started with the Angular CLI
 - Popular commands and schematics
- Introduction to Angular
- Structure of an app
- Components Model
 - Decorators, Templating Syntax, Data Binding, Directives, Input/Output, Events, Pipes, Encapsulation
- Services and Dependency Injection

Reactive Programming with RxJS (Observables)

- Understanding RxJS and its use with Angular
- Subscribe() function
- The async pipe
- Pipe() function
- Using popular RxJS operators
- RxJS best practices
- Handling errors in RxJS code

Async Requests to a REST API (HTTP)

- The HttpClient service
- A first async GET request
- Consume an Observable response
- Implementing a local cache with ShareReplay()
- Debugging an application
- Implementing pagination, sorting and filtering on a list
- Modules
 - Organizing a modular app
 - Modules imports/exports
- Deployment
 - Optimizations for production
 - ng build and ng deploy commands
- Migrating a project to a new version of Angular



Navigation and Routing

- Using the Angular Router
- routerLink directive
- The router-outlet placeholder
- Router service
- Nested routes
- Route parameters
- Handling routing errors
- Lazy-loading

Forms and Validation

- Reactive Forms
- Validation API
- POST request

Using HttpClient to Update or Delete data on the server

- PUT and DELETE requests

Authentication

- Login component
- Authentication Service
- Security with JSON Web Tokens (JWT)
- Http Interceptors
- Protecting access to components with Route Guards

TypeScript fundamentals

- The TypeScript compiler
 - Type safety with type annotation
 - Classes and Interfaces
 - Generics
- Setup and Tooling
- Getting started with the Angular CLI
 - Popular commands and schematics
- Introduction to Angular
- Structure of an app
- Components Model
 - Decorators, Templating Syntax, Data Binding, Directives, Input/Output, Events, Pipes, Encapsulation
- Services and Dependency Injection

Reactive Programming with RxJS (Observables)

- Understanding RxJS and its use with Angular



- Subscribe() function
- The async pipe
- Pipe() function
- Using popular RxJS operators
- RxJS best practices
- Handling errors in RxJS code

Async Requests to a REST API (HTTP)

- The HttpClient service
- A first async GET request
- Consume an Observable response
- Implementing a local cache with ShareReplay()
- Debugging an application
- Implementing pagination, sorting and filtering on a list
- Modules
 - Organizing a modular app
 - Modules imports/exports
- Deployment
 - Optimizations for production
 - ng build and ng deploy commands
- Migrating a project to a new version of Angular

Navigation and Routing

- Using the Angular Router
- routerLink directive
- The router-outlet placeholder
- Router service
- Nested routes
- Route parameters
- Handling routing errors
- Lazy-loading

Forms and Validation

- Reactive Forms
- Validation API
- POST request

Using HttpClient to Update or Delete data on the server

- PUT and DELETE requests

Authentication

- Login component
- Authentication Service



- Security with JSON Web Tokens (JWT)
- Http Interceptors
- Protecting access to components with Route Guards

TypeScript fundamentals

- The TypeScript compiler
 - Type safety with type annotation
 - Classes and Interfaces
 - Generics
- Setup and Tooling
- Getting started with the Angular CLI
 - Popular commands and schematics
- Introduction to Angular
- Structure of an app
- Components Model
 - Decorators, Templating Syntax, Data Binding, Directives, Input/Output, Events, Pipes, Encapsulation
- Services and Dependency Injection

Reactive Programming with RxJS (Observables)

- Understanding RxJS and its use with Angular
- Subscribe() function
- The async pipe
- Pipe() function
- Using popular RxJS operators
- RxJS best practices
- Handling errors in RxJS code

Async Requests to a REST API (HTTP)

- The HttpClient service
- A first async GET request
- Consume an Observable response
- Implementing a local cache with ShareReplay()
- Debugging an application
- Implementing pagination, sorting and filtering on a list
- Modules
 - Organizing a modular app
 - Modules imports/exports
- Deployment
 - Optimizations for production
 - ng build and ng deploy commands
- Migrating a project to a new version of Angular



Navigation and Routing

- Using the Angular Router
- routerLink directive
- The router-outlet placeholder
- Router service
- Nested routes
- Route parameters
- Handling routing errors
- Lazy-loading

Forms and Validation

- Reactive Forms
- Validation API
- POST request

Using HttpClient to Update or Delete data on the server

- PUT and DELETE requests

Authentication

- Login component
- Authentication Service
- Security with JSON Web Tokens (JWT)
- Http Interceptors
- Protecting access to components with Route Guards