

Resource Links

00_03. Using the Example Code and Codespaces

- GitHub Codespaces overview: <https://docs.github.com/en/codespaces/overview>

01_05. Workflow and Tooling

- Angular CLI overview: <https://angular.io/cli>

02_02. Angular Modules

- Introduction to modules: <https://angular.io/guide/architecture-modules>
- NgModules: <https://angular.io/guide/ngmodules>

02_03. Angular Component Decorator

- Angular components overview: <https://angular.io/guide/component-overview>
- Component decorator: <https://angular.io/api/core/Component>

02_04. Angular Component Classes

- Angular components overview: <https://angular.io/guide/component-overview>
- Component decorator: <https://angular.io/api/core/Component>

02_05. Reference Variables and Data Binding in Angular Templates

- Binding syntax: <https://angular.io/guide/binding-syntax>
- Displaying values with interpolation: <https://angular.io/guide/interpolation>
- Introduction to components and templates: <https://angular.io/guide/architecture-components>
- Template syntax: <https://angular.io/guide/template-syntax>
- Two-way binding: <https://angular.io/guide/two-way-binding>
- Understanding template variables: <https://angular.io/guide/template-reference-variables>

02_06. Structural Directives in Angular Templates

- Binding syntax: <https://angular.io/guide/binding-syntax>
- Displaying values with interpolation: <https://angular.io/guide/interpolation>

- Introduction to components and templates: <https://angular.io/guide/architecture-components>
- Template syntax: <https://angular.io/guide/template-syntax>
- Two-way binding: <https://angular.io/guide/two-way-binding>
- Understanding template variables: <https://angular.io/guide/template-reference-variables>

02_07. Angular Standalone Components

- Getting started with standalone components: <https://angular.io/guide/standalone-components>
-

03_01. Overview of Angular Pipes

- Standard pipes: <https://angular.io/api/common#pipes>
- Transforming data using pipes: <https://angular.io/guide/pipes>

03_02. Using Built-In Angular Pipes

- PercentPipe: <https://angular.io/api/common/PercentPipe>
- SlicePipe: <https://angular.io/api/common/SlicePipe>

03_03. Using Custom Angular Pipes

- Pipe decorator: <https://angular.io/api/core/Pipe>

03_04. Angular Services and the Injectable Decorator

- Creating an injectable service: <https://angular.io/guide/creating-injectable-service>
- Introduction to services and dependency injection: <https://angular.io/guide/architecture-services>
- Singleton services: <https://angular.io/guide/singleton-services>

03_05. Angular Service Classes

- Creating an injectable service: <https://angular.io/guide/creating-injectable-service>
- Introduction to services and dependency injection: <https://angular.io/guide/architecture-services>
- Singleton services: <https://angular.io/guide/singleton-services>
- RxJS pipe function: <https://rxjs.dev/api/index/function/pipe>
- RxJS delay function: <https://rxjs.dev/api/index/function/delay>

03_06. Angular Directive Decorator

- Directive decorator: <https://angular.io/api/core/Directive>

03_07. Angular Directive Classes

- Directive decorator: <https://angular.io/api/core/Directive>
 - ElementRef class: <https://angular.io/api/core/ElementRef>
-

04_01. Angular Project Configuration

- Angular workspace configuration: <https://angular.io/guide/workspace-config>
- What is a tsconfig.json: <https://www.typescriptlang.org/docs/handbook/tsconfig-json.html>

04_02. Application Environments and Build Targets

- Building and serving Angular apps: <https://angular.io/guide/build>

04_03. Injecting Dependencies Using Dependency Injection

- Injecting a dependency:
<https://angular.io/guide/dependency-injection#injecting-a-dependency>

04_04. Declaring Dependencies Using Dependency Injection

- Injectable decorator: <https://angular.io/api/core/Injectable>

04_05. Providing Dependencies Using Dependency Injection

- Configuring dependency providers:
<https://angular.io/guide/dependency-injection-providers>
- InjectionToken class: <https://angular.io/api/core/InjectionToken>
- Providing dependency:
<https://angular.io/guide/dependency-injection#providing-dependency>

04_06. Angular Change Detection Strategies

- Angular change detection and runtime optimization:
<https://angular.io/guide/change-detection>

04_07. Angular Component Lifecycle Hooks

- Component lifecycle: <https://angular.io/guide/lifecycle-hooks>
-

05_01. Defining Routes for Angular Router

- Angular routing: <https://angular.io/guide/routing-overview>

- Angular Router: <https://angular.io/api/router>

05_02. Using Routes with Angular's Router Module

- Angular routing: <https://angular.io/guide/routing-overview>
- Angular Router: <https://angular.io/api/router>

05_03. Lazy Loading Angular Routes

- Lazy-loading feature modules: <https://angular.io/guide/lazy-loading-ngmodules>
- Dynamic import expressions:
<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/import>

05_04. Dynamic Route Parameters in Angular Routes

- Angular routing: <https://angular.io/guide/routing-overview>
- Angular Router: <https://angular.io/api/router>

05_05. Preloading Strategies for Angular Router

- Angular routing: <https://angular.io/guide/routing-overview>
- Angular Router: <https://angular.io/api/router>

05_06. Using Angular Router with Angular Templates

- RouterOutlet directive: <https://angular.io/api/router/RouterOutlet>
- RouterLink directive: <https://angular.io/api/router/RouterLink>
- RouterLinkActive directive: <https://angular.io/api/router/RouterLinkActive>

06_01. Angular Signals and the Signal Function

- Angular Signals: <https://angular.io/guide/signals>

06_02. Angular Signals and the Computed Function

- Angular Signals: <https://angular.io/guide/signals>

06_03. Angular Signals and the Effect Function

- Angular Signals: <https://angular.io/guide/signals>

06_04. Angular's Async Pipe

- AsyncPipe: <https://angular.io/api/common/AsyncPipe>

06_05. RxJS and Reactive Programming in Angular

- Using observables to pass values: <https://angular.io/guide/observables>
 - RxJS API list: <https://rxjs.dev/api>
 - RxJS npm package: <https://www.npmjs.com/package/rxjs>
-

07_01. Overview of Unit Testing

- ng test command: <https://angular.io/cli/test>
- Karma test runner: <https://karma-runner.github.io/latest/index.html>

07_02. Unit Testing without Dependency Injection

- Testing pipes: <https://angular.io/guide/testing-pipes>
- Jasmine API docs: <https://jasmine.github.io/api/edge/global>

07_03. Unit Testing with Dependency Injection

- Testing services: <https://angular.io/guide/testing-services>
- Jasmine API docs: <https://jasmine.github.io/api/edge/global>

07_04. Static Code Linting in Angular

- Basic ESLint configuration: <https://www.linkedin.com/learning/angular-testing-and-debugging-10201318/basic-eslint-configuration>
 - ng lint: <https://angular.io/cli/lint>
 - ESLint homepage: <https://eslint.org/>
-