

RANGLE.IO

# Rangle's Angular 2 Training Book



Published with  
GitBook



---

# Table of Contents

Introduction	1.1
License	1.2
Why Angular?	1.3
The Architect's Guide to Angular	2.1
Creating Functional Forms	2.1.1
EcmaScript 6 and TypeScript Features	3.1
ES6	3.1.1
Classes	3.1.1.1
Refresher on 'this'	3.1.1.2
Arrow Functions	3.1.1.3
Template Strings	3.1.1.4
Inheritance	3.1.1.5
Delegation	3.1.1.6
Constants and Block Scoped Variables	3.1.1.7
...spread and ...rest	3.1.1.8
Destructuring	3.1.1.9
Modules	3.1.1.10
TypeScript	3.1.2
Getting Started With TypeScript	3.1.2.1
Working With tsc	3.1.2.2
Typings	3.1.2.3
Linting	3.1.2.4
TypeScript Features	3.1.2.5
TypeScript Classes	3.1.2.6
Interfaces	3.1.2.7
Shapes	3.1.2.8
Type Inference	3.1.2.9
Type Keyword	3.1.2.10
Decorators	3.1.2.11
Property Decorators	3.1.2.12

---

Class Decorators	3.1.2.13
Parameter Decorators	3.1.2.14
The JavaScript Toolchain	4.1
Source Control: git	4.1.1
The Command Line	4.1.2
Command Line JavaScript: NodeJS	4.1.3
Back-End Code Sharing and Distribution: npm	4.1.4
Module Loading, Bundling and Build Tasks: Webpack	4.1.5
Chrome	4.1.6
Bootstrapping an Angular Application	5.1
Understanding the File Structure	5.1.1
Bootstrapping Providers	5.1.2
Components in Angular	6.1
Creating Components	6.1.1
Application Structure with Components	6.1.2
Passing Data into a Component	6.1.2.1
Responding to Component Events	6.1.2.2
Using Two-Way Data Binding	6.1.2.3
Accessing Child Components from Template	6.1.2.4
Projection	6.1.3
Structuring Applications with Components	6.1.4
Using Other Components	6.1.5
Directives	7.1
Attribute Directives	7.1.1
NgStyle Directive	7.1.1.1
NgClass Directive	7.1.1.2
Structural Directives	7.1.2
NgIf Directive	7.1.2.1
NgFor Directive	7.1.2.2
NgSwitch Directives	7.1.2.3
Using Multiple Structural Directives	7.1.2.4
Advanced Components	8.1
Component Lifecycle	8.1.1
Accessing Other Components	8.1.2

---

---

View Encapsulation	8.1.3
ElementRef	8.1.4
Observables	9.1
Using Observables	9.1.1
Error Handling	9.1.2
Disposing Subscriptions and Releasing Resources	9.1.3
Observables vs Promises	9.1.4
Using Observables From Other Sources	9.1.5
Observables Array Operations	9.1.6
Cold vs Hot Observables	9.1.7
Summary	9.1.8
Angular Dependency Injection	10.1
What is DI?	10.1.1
DI Framework	10.1.2
Angular's DI	10.1.3
@Inject() and @Injectable	10.1.3.1
Injection Beyond Classes	10.1.3.2
Avoiding Injection Collisions: OpaqueToken	10.1.3.3
The Injector Tree	10.1.3.4
Http	11.1
Making Requests	11.1.1
Catching Rejections	11.1.2
Catch and Release	11.1.2.1
Cancel a Request	11.1.2.2
Retry	11.1.2.3
Search with flatMap	11.1.3
Enhancing Search with switchMap	11.1.4
Requests as Promises	11.1.5
Change Detection	12.1
Change Detection Strategies in Angular 1 vs Angular 2	12.1.1
How Change Detection Works	12.1.2
Change Detector Classes	12.1.3
Change Detection Strategy: OnPush	12.1.4

---

---

Enforcing Immutability	12.1.5
Additional Resources	12.1.6
Zone.js	13.1
Advanced Angular	14.1
Directives	14.1.1
Creating an Attribute Directive	14.1.1.1
Listening to an Element Host	14.1.1.1.1
Setting Properties in a Directive	14.1.1.1.2
Creating a Structural Directive	14.1.1.2
View Containers and Embedded Views	14.1.1.2.1
Providing Context Variables to Directives	14.1.1.2.2
AoT	14.1.2
AoT limitations	14.1.2.1
AoT Configuration	14.1.2.2
Immutable.js	15.1
What is Immutability?	15.1.1
The Case for Immutability	15.1.2
JavaScript Solutions	15.1.3
Object.assign	15.1.3.1
Object.freeze	15.1.3.2
Immutable.js Basics	15.1.4
Immutable.Map	15.1.4.1
Map.merge	15.1.4.1.1
Nested Objects	15.1.4.2
Deleting Keys	15.1.4.2.1
Maps are Iterable	15.1.4.2.2
Immutable.List	15.1.4.3
Performance and Transient Changes	15.1.4.4
Official Documentation	15.1.4.5
Pipes	16.1
Using Pipes	16.1.1
Custom Pipes	16.1.2
Stateful Pipes	16.1.3
Forms	17.1

---

[Click here to download full PDF material](#)