

## Flutter & Dart Kochbuch

Cross-Platform-Apps für die  
Cloud entwickeln

# DAS INHALTS- VERZEICHNIS

» Hier geht's  
direkt  
zum Buch

<b>Vorwort</b> .....	<b>11</b>
<b>Einleitung</b> .....	<b>13</b>
Wer dieses Buch lesen sollte .....	13
Warum ich dieses Buch geschrieben habe .....	14
Wie dieses Buch aufgebaut ist .....	14
Konventionen, die in diesem Buch genutzt werden .....	15
Danksagung .....	16
<b>1 Dart-Variablen kennenlernen</b> .....	<b>17</b>
Eine Dart-Anwendung ausführen .....	18
Mit Integerwerten arbeiten .....	19
Mit Doublewerten arbeiten .....	20
Mit booleschen Werten arbeiten .....	21
Mit Strings arbeiten .....	22
Informationen an der Konsole ausgeben .....	23
Eine konstante Variable hinzufügen (Kompilierungszeit) .....	24
Eine konstante Variable hinzufügen (Laufzeit) .....	24
Mit Nullvariablen arbeiten .....	25
<b>2 Den Programmablauf steuern</b> .....	<b>27</b>
Überprüfen, ob eine Bedingung erfüllt wurde .....	27
Iterieren, bis eine Bedingung erfüllt ist .....	29
Über eine Reihe von Elementen iterieren .....	31
Aktionen abhängig von einem Wert ausführen .....	32
Werte mit einem Enumerator repräsentieren .....	34
Exception-Handling implementieren .....	35
<b>3 Funktionen implementieren</b> .....	<b>37</b>
Funktionen deklarieren .....	38
Eine Funktion mit Parametern ausstatten .....	39

Optionale Parameter verwenden . . . . .	40
Werte von Funktionen zurückgeben . . . . .	41
Funktionen in Kurzform deklarieren . . . . .	42
Mit einem Future eine Funktion verzögert aufrufen . . . . .	43
<b>4 Umgang mit Listen und Maps . . . . .</b>	<b>45</b>
Listen mit Daten erstellen . . . . .	46
Eine Liste mit Daten erweitern . . . . .	47
Listen mit komplexen Typen verwenden . . . . .	48
Schlüssel-Wert-Paare per Map handhaben . . . . .	50
Datenstrukturen aus einer Map ausgeben . . . . .	51
Prüfen, ob Inhalte in einer Map existieren . . . . .	52
Komplexe Datentypen ausgeben . . . . .	53
<b>5 Einstieg in das objektorientierte Dart . . . . .</b>	<b>55</b>
Einstieg in das objektorientierte Dart . . . . .	55
Eine Klasse erstellen . . . . .	57
Eine Klasse mit einem Konstruktor initialisieren . . . . .	58
Klassenvererbung hinzufügen . . . . .	60
Ein Klassen-Interface hinzufügen . . . . .	62
Ein Klassen-Mixin hinzufügen . . . . .	65
<b>6 Testfälle in Dart . . . . .</b>	<b>69</b>
Das Dart-Test-Paket zu Ihrer Anwendung hinzufügen . . . . .	70
Eine Beispiel-Testanwendung erstellen . . . . .	71
Unit-Tests in Ihrer Dart-Anwendung ausführen . . . . .	72
Mehrere Unit-Tests zusammenfassen . . . . .	74
Mock-Daten für Tests hinzufügen . . . . .	77
<b>7 Einführung in Flutter . . . . .</b>	<b>81</b>
Eine Anwendungsoberfläche mocken . . . . .	82
Eine Projektgrundlage in Flutter erstellen . . . . .	83
Das Debug-Banner von Flutter entfernen . . . . .	85
Widgets verstehen . . . . .	87
Den Widget-Baum verstehen . . . . .	87
Die Rendering-Performance von Widgets verbessern . . . . .	88
<b>8 Assets hinzufügen . . . . .</b>	<b>91</b>
Die Datei pubspec.yaml verwenden . . . . .	91
Einen Assets-Ordner hinzufügen . . . . .	93
Ein Bild referenzieren . . . . .	94
Das Google-Font-Paket einbinden . . . . .	96
Ein Paket importieren . . . . .	97

<b>9</b>	<b>Mit Widgets arbeiten</b> . . . . .	<b>99</b>
	Ein zustandsloses Widget in Flutter erstellen . . . . .	100
	Ein zustandsbehaftetes Widget in Flutter erstellen . . . . .	101
	Flutter-Widgets refaktorisieren . . . . .	104
	Die Scaffold-Klasse einsetzen. . . . .	107
	Einen AppBar-Header hinzufügen. . . . .	110
	Mit einem Container arbeiten . . . . .	112
	Ein Center-Widget verwenden . . . . .	115
	Eine SizedBox nutzen . . . . .	116
	Eine Column verwenden . . . . .	119
	Eine Row verwenden . . . . .	123
	Ein Expanded-Widget verwenden. . . . .	125
<b>10</b>	<b>Benutzeroberflächen entwickeln</b> . . . . .	<b>129</b>
	Das Google-Font-Paket verwenden . . . . .	129
	RichText einsetzen. . . . .	131
	Die Host-Plattform ermitteln. . . . .	132
	Ein Placeholder-Widget verwenden . . . . .	134
	Einen LayoutBuilder verwenden . . . . .	136
	Mit MediaQuery auf Bildschirmdimensionen zugreifen . . . . .	140
<b>11</b>	<b>Mit Daten auf dem Bildschirm arbeiten</b> . . . . .	<b>143</b>
	Eine vertikale ListView implementieren . . . . .	144
	Eine horizontale ListView implementieren . . . . .	147
	Eine SliverAppBar hinzufügen. . . . .	150
	Eine SliverList hinzufügen . . . . .	152
	Ein GridView mit Elementen hinzufügen . . . . .	156
	Eine SnackBar (Popup-Benachrichtigung) hinzufügen . . . . .	159
<b>12</b>	<b>Seitennavigation in Flutter.</b> . . . . .	<b>163</b>
	Seitennavigation über Routen (imperativ) hinzufügen . . . . .	163
	Seitennavigation über Routen (deklarativ) hinzufügen. . . . .	167
	Einen Navigator Drawer implementieren . . . . .	171
	Mit Tabs arbeiten . . . . .	175
	Eine Bottom Navigation Bar hinzufügen. . . . .	179
	Informationen mit Schlüsseln weitergeben . . . . .	181
<b>13</b>	<b>Mit Daten-Assets arbeiten</b> . . . . .	<b>185</b>
	Strategien beim Zugriff auf Daten . . . . .	186
	Daten refaktorisieren . . . . .	188
	Dart-Klassen aus JSON erzeugen . . . . .	190
	JSON-Daten asynchron verwenden. . . . .	192
	Einen JSON-Datensatz aus dem Assets-Ordner holen . . . . .	197
	Auf Remote-JSON-Daten zugreifen. . . . .	200

<b>14</b>	<b>Die Flutter-Benutzeroberfläche testen</b> . . . . .	<b>203</b>
	Automatisierte Widget-Tests in Flutter . . . . .	203
	Automatisierte Widget-Tests ausführen . . . . .	206
	Integrationstests mit Flutter Driver durchführen . . . . .	207
	Die Kompatibilität mit Android/iOS-Geräten testen . . . . .	209
<b>15</b>	<b>Mit Firebase und Flutter arbeiten</b> . . . . .	<b>211</b>
	Die Firebase-Plattform mit Flutter verwenden . . . . .	212
	Ein Firebase-Projekt aufsetzen . . . . .	213
	Das Firebase-SDK für eine lokale Entwicklung initialisieren . . . . .	215
	Firebase-Emulatoren konfigurieren . . . . .	217
	flutterfire_cli zu einer Entwicklungsumgebung hinzufügen . . . . .	220
	Eine Firestore-Datenbank integrieren . . . . .	222
	Daten in eine Firestore-Datenbank schreiben . . . . .	225
	Daten aus Cloud Firestore lesen . . . . .	229
	Firebase Authentication zu Flutter hinzufügen . . . . .	234
	Flutter Web mit Firebase Hosting nutzen . . . . .	240
<b>16</b>	<b>Einführung in Cloud-Services</b> . . . . .	<b>243</b>
	Einstieg mit Cloud-Providern . . . . .	244
	Mit Identity and Access Management arbeiten . . . . .	244
	Ein Objekt mit Cloud Storage hosten . . . . .	246
	Einen Backend-HTTP-Server mit Dart entwickeln . . . . .	247
	Einen Dart-Container bauen . . . . .	249
	Einstieg in Serverless mit Dart . . . . .	251
<b>17</b>	<b>Einstieg in die Spiele-Entwicklung</b> . . . . .	<b>253</b>
	Das Flame-Paket zu Flutter hinzufügen . . . . .	254
	Ein Flame-Rahmenprogramm erstellen . . . . .	255
	Ein Sprite hinzufügen . . . . .	256
	Ein Sprite manuell horizontal bewegen . . . . .	258
	Ein Sprite automatisch vertikal bewegen . . . . .	261
	Kollisionserkennung hinzufügen . . . . .	264
	Text-Rendering hinzufügen . . . . .	268
	Grafik-Primitive hinzufügen . . . . .	272
	Soundeffekte hinzufügen . . . . .	277

<b>Anhang</b>	<b>Richten Sie Ihre Umgebung ein</b>	<b>285</b>
	Herausfinden, welche Dart-Installation die passende ist.	285
	Dart in DartPad ausführen	286
	Das Flutter-Framework installieren.	287
	Flutter Doctor verwenden	287
	Das Dart-SDK installieren	289
	Mit VS Code entwickeln	289
	Android Studio für die Arbeit mit Dart erweitern	290
	Einen Release Channel auswählen	291
	Mit Flutter Config die Zielplattform festlegen.	291
<b>Index</b>		<b>293</b>