

# Auf einen Blick

## TEIL I Einführung

1	Über dieses Buch .....	21
2	Installation und die Wahl der IDE .....	25
3	Warum man mit Dart voll ins Schwarze trifft .....	37
4	Pubs: Abhängigkeiten komfortabel verwalten .....	171

## TEIL II Bedienoberflächen

5	Widgets .....	185
6	Layouting .....	229
7	Navigation .....	319
8	Früchte am Baum: InheritedWidgets .....	357
9	BloCs: View Logic aus dem Weg räumen .....	363
10	Mit dem Nutzer interagieren .....	383
11	Themes und Styles .....	419
12	Animationen .....	435
13	Arbeiten mit Canvas und Paint .....	453

## TEIL III Inhalte und Accessibility

14	Assets und Medien .....	465
15	Den Nutzerkreis vergrößern: Localization und Accessibility .....	479
16	Die Cloud: Unendliche Weiten .....	513
17	Persistieren: Caches und Datenbanken .....	565

## TEIL IV Sensoren und Hardware

18	Mit der Außenwelt kommunizieren .....	591
19	Den Kanal wechseln: Channels zur Plattform .....	601

## TEIL V Endgame

20	Analytics, Tests und Releases .....	623
----	-------------------------------------	-----

# Inhalt

Materialien zum Buch .....	18
----------------------------	----

## TEIL I Einführung

---

<b>1 Über dieses Buch</b>	21
---------------------------	----

<b>1.1 Was Sie in diesem Buch lernen werden</b> .....	22
<b>1.2 Null-Safety als Default-Einstellung ab Flutter 2.2</b> .....	23
<b>1.3 Danksagung</b> .....	24

---

<b>2 Installation und die Wahl der IDE</b>	25
--	----

<b>2.1 Flutter herunterladen und installieren</b> .....	25
<b>2.2 Die Wahl der IDE</b> .....	29
2.2.1 IntelliJ und Android Studio .....	29
2.2.2 Visual Studio Code .....	33
<b>2.3 Die Flutter-Installation voll im Griff</b> .....	35

---

<b>3 Warum man mit Dart voll ins Schwarze trifft</b>	37
--	----

<b>3.1 Der Ursprung von Dart</b> .....	37
3.1.1 Die Köpfe hinter der Sprache .....	38
3.1.2 Die Kernideen .....	38
3.1.3 Eine Sprache des 21. Jahrhunderts .....	39
<b>3.2 Ein erstes Dart-Programm</b> .....	39
3.2.1 Dateinamen .....	41
3.2.2 Kommentare .....	41
<b>3.3 Wichtige Begleiter</b> .....	41
<b>3.4 Variablen</b> .....	42
3.4.1 Wiederholungstäter: Variablen mit var .....	42
3.4.2 Konstanten mit final und const .....	43
3.4.3 Warum Typinferenz? .....	45

<b>3.5 Datentypen</b> .....	47
3.5.1 Numerische Datentypen .....	47
3.5.2 Boolesche Werte .....	49
3.5.3 Strings .....	51
3.5.4 Datum, Zeit und Dauer .....	60
3.5.5 dynamic .....	62
3.5.6 Den Typ identifizieren .....	63
<b>3.6 Funktionen</b> .....	65
3.6.1 Bezeichner .....	66
3.6.2 Parameter .....	66
3.6.3 Rumpf .....	70
3.6.4 Synchrone und asynchrone Ausführung .....	72
3.6.5 Rückgabetyp .....	72
3.6.6 Der Typ Function .....	74
3.6.7 Anonyme Funktionen .....	75
3.6.8 Getter und Setter .....	76
<b>3.7 Klassen, Mixins und Enums</b> .....	79
3.7.1 Eine Klasse .....	79
3.7.2 Object: Der Bottom-Type .....	89
3.7.3 Vererbung und Interfaces .....	93
3.7.4 Statische Elemente .....	99
3.7.5 Mixins .....	100
3.7.6 enums .....	102
3.7.7 typedef .....	105
3.7.8 Extension Methods .....	106
<b>3.8 Generika</b> .....	112
3.8.1 Generische Klassen .....	112
3.8.2 Generische Funktionen .....	114
3.8.3 Den Typparameter einschränken .....	114
3.8.4 Mehrere Typparameter .....	115
<b>3.9 Collections</b> .....	116
3.9.1 List .....	116
3.9.2 Set .....	122
3.9.3 Map .....	124
3.9.4 Iterable .....	126
<b>3.10 Kontrollstrukturen</b> .....	133
3.10.1 if-else .....	133
3.10.2 Die for-Schleife .....	137

3.10.3	for-in .....	137
3.10.4	while und do-while .....	138
3.10.5	Die Schlüsselwörter continue und break .....	139
3.10.6	switch-case .....	139
3.10.7	Assert .....	140
<b>3.11</b>	<b>Exceptions und Errors .....</b>	<b>141</b>
3.11.1	Exceptions .....	141
3.11.2	Errors .....	141
3.11.3	Fehler werfen: throw .....	143
3.11.4	Der zweite Parameter von catch .....	143
3.11.5	Spezifische Exceptions oder Errors fangen .....	144
3.11.6	Das Schlüsselwort finally .....	144
<b>3.12</b>	<b>Asynchrone Programmierung .....</b>	<b>145</b>
3.12.1	Event Loops .....	146
3.12.2	Future .....	146
3.12.3	async und await .....	151
3.12.4	Streams .....	153
3.12.5	Isolates .....	154
<b>3.13</b>	<b>Die Library .....</b>	<b>155</b>
3.13.1	Mini-Library .....	155
3.13.2	Eine Library importieren .....	156
<b>3.14</b>	<b>Struktur eines Projekts .....</b>	<b>157</b>
<b>3.15</b>	<b>Dokumentation des Codes .....</b>	<b>159</b>
<b>3.16</b>	<b>Null-Safety .....</b>	<b>160</b>
3.16.1	Nullability als Opt-In .....	160
3.16.2	var und final .....	161
3.16.3	Das Schlüsselwort required und Parameter .....	162
3.16.4	if-Abfragen beeinflussen den Typ .....	164
3.16.5	Die Migration eines Projekts .....	165
<b>3.17</b>	<b>Zusammenfassung .....</b>	<b>169</b>

---

<b>4</b>	<b>Pubs: Abhängigkeiten komfortabel verwalten .....</b>	<b>171</b>
<b>4.1</b>	<b>Von der Library zum Package .....</b>	<b>171</b>
<b>4.2</b>	<b>Ein Meer aus pubs .....</b>	<b>172</b>
<b>4.3</b>	<b>Wo Licht ist, da fällt auch Schatten .....</b>	<b>172</b>

<b>4.4</b>	<b>Packages verwenden</b>	173
4.4.1	pubspec.yaml	174
4.4.2	Ein Package integrieren	175
<b>4.5</b>	<b>Versionen</b>	177
4.5.1	Caret-Syntax	177
4.5.2	Versionsrahmen	178
4.5.3	Feste Versionen	178
4.5.4	Keine Version angeben	179
4.5.5	Packages aktualisieren	180
4.5.6	Versionen erzwingen	180
<b>4.6</b>	<b>Der System Cache</b>	180
<b>4.7</b>	<b>Der Build Runner</b>	181
<b>4.8</b>	<b>Zusammenfassung</b>	181

## TEIL II Bedienoberflächen

<b>5</b>	<b>Widgets</b>	185
<b>5.1</b>	<b>Ein erstes Widget</b>	186
5.1.1	Stateless Widgets	196
<b>5.2</b>	<b>Stateful Widgets</b>	197
5.2.1	Die Ausgangssituation	197
5.2.2	Ein neues (Stateful)Widget erstellen	198
5.2.3	Das Widget verschieben	204
5.2.4	Das State-Objekt	205
5.2.5	Die LoginMask mit State anreichern	209
<b>5.3</b>	<b>Material Design und Cupertino Design</b>	214
<b>5.4</b>	<b>Vom Widget zum Rendering Object</b>	218
5.4.1	Widget-Tree	218
5.4.2	Element-Tree	219
5.4.3	RenderObject-Tree	220
<b>5.5</b>	<b>Keys</b>	221
5.5.1	Widgets referenzieren	221
5.5.2	Widgets markieren	225
<b>5.6</b>	<b>Zusammenfassung</b>	227

---

<b>6</b>	<b>Layouting</b>	229
<b>6.1</b>	<b>Positionierung</b>	229
6.1.1	Constraints	229
6.1.2	Zentrieren	230
6.1.3	Align	231
6.1.4	Padding	232
6.1.5	Baseline	233
6.1.6	SafeArea	234
6.1.7	Column	234
6.1.8	Row	238
6.1.9	Overflows bekämpfen mit Expanded und Flexible	239
6.1.10	Spacer	243
6.1.11	Tabellen	244
6.1.12	Stacks	247
6.1.13	Wrap	252
<b>6.2</b>	<b>Scroll-Widgets</b>	254
6.2.1	ListView	254
6.2.2	GridView	260
6.2.3	Der Konstruktor für Spezialfälle	264
6.2.4	Vertikal oder horizontal: ScrollDirection	264
6.2.5	Der ScrollController	265
6.2.6	Die Eigenschaft shrinkWrap	268
6.2.7	Ein einfaches Widget scrollen	269
6.2.8	Die Scrollleiste	271
6.2.9	ScrollNotifications	271
6.2.10	Wie im Karussel: Die PageView	273
6.2.11	Slivers	274
6.2.12	Caching von Items	280
<b>6.3</b>	<b>Constraints, Höhen und Breiten</b>	283
6.3.1	AspectRatio	283
6.3.2	ConstrainedBox	284
6.3.3	UnconstrainedBox	284
6.3.4	SizedBox	285
6.3.5	FittedBox	285
6.3.6	FractionallySizedBox	286
6.3.7	LimitedBox	286
6.3.8	OverflowBox	287
6.3.9	SizedOverflowBox	289

6.3.10	IntrinsicHeight und IntrinsicWeight .....	289
6.3.11	LayoutBuilder und SliverLayoutBuilder .....	290
<b>6.4</b>	<b>MediaQueryData</b> .....	292
6.4.1	Drei verschiedene Arten von Padding .....	294
6.4.2	Ein eigenes MediaQueryData-Objekt einfügen .....	296
6.4.3	Bedienungshilfen .....	297
<b>6.5</b>	<b>Overlay</b> .....	297
6.5.1	OverlayEntry .....	297
6.5.2	Overlay auflösen .....	298
6.5.3	OverlayEntry auf dem Overlay positionieren .....	299
6.5.4	Mit Overlay einen Hinweis anzeigen .....	299
<b>6.6</b>	<b>Die Gerüste</b> .....	307
6.6.1	Das Scaffold .....	308
6.6.2	CupertinoTabScaffold .....	312
6.6.3	CupertinoPageScaffold .....	313
6.6.4	Slivers in Scaffolds .....	315
6.6.5	Automatisches Scrollen .....	317
<b>6.7</b>	<b>Zusammenfassung</b> .....	318
<b>7</b>	<b>Navigation</b>	319
<b>7.1</b>	<b>Ein erster Sprung</b> .....	319
<b>7.2</b>	<b>Wenn Flutter nach dem Weg fragt: Named Routes</b> .....	323
7.2.1	InitialRoute und Routes .....	323
7.2.2	PushNamed und Pop .....	324
<b>7.3</b>	<b>Eine Route austauschen</b> .....	325
7.3.1	pushReplacementNamed und pushReplacement .....	325
7.3.2	popAndPushNamed .....	327
7.3.3	replace .....	327
<b>7.4</b>	<b>Routen entfernen</b> .....	327
<b>7.5</b>	<b>Großflächig aufräumen mit push und pop</b> .....	328
<b>7.6</b>	<b>NavigatorObserver</b> .....	328
<b>7.7</b>	<b>Daten austauschen</b> .....	330
7.7.1	Daten über den Konstruktor .....	330

7.7.2	Daten über RouteSettings .....	331
7.7.3	Wenn Routen antworten .....	331
<b>7.8</b>	<b>Den Android-Back-Button zähmen .....</b>	<b>333</b>
<b>7.9</b>	<b>Routen als URIs .....</b>	<b>334</b>
<b>7.10</b>	<b>Verschachtelte Navigator-Objekte .....</b>	<b>334</b>
<b>7.11</b>	<b>Navigator 2.0 .....</b>	<b>336</b>
7.11.1	Wann brauchen Sie den Navigator 2.0? .....	336
7.11.2	Die Bestandsaufnahme .....	336
7.11.3	Von der RouteInformation zum Navigator .....	340
7.11.4	Die Startseite anzeigen .....	342
7.11.5	Die Navigation History ersetzen .....	347
7.11.6	Navigationsziele im Router mit onGenerateRoute .....	352
7.11.7	Zurücknavigation durch Plattformereignisse .....	354
<b>7.12</b>	<b>Zusammenfassung .....</b>	<b>355</b>
<b>8</b>	<b>Früchte am Baum: InheritedWidgets .....</b>	<b>357</b>
<b>8.1</b>	<b>Ein InheritedWidget .....</b>	<b>357</b>
<b>8.2</b>	<b>Das eigene InheritedWidget: TimeInfo .....</b>	<b>358</b>
<b>8.3</b>	<b>Zusammenfassung .....</b>	<b>361</b>
<b>9</b>	<b>BloCs: View Logic aus dem Weg räumen .....</b>	<b>363</b>
<b>9.1</b>	<b>Der BloC .....</b>	<b>363</b>
<b>9.2</b>	<b>BloC und ValueListenable .....</b>	<b>364</b>
<b>9.3</b>	<b>ChangeNotifier .....</b>	<b>368</b>
<b>9.4</b>	<b>Streams .....</b>	<b>371</b>
9.4.1	StreamController .....	372
9.4.2	Broadcasts .....	375
9.4.3	Fehler in Streams .....	377
<b>9.5</b>	<b>Futures .....</b>	<b>379</b>

<b>10 Mit dem Nutzer interagieren</b>	383
<b>10.1 Button, TextField und Co</b>	383
10.1.1 TextFields	386
10.1.2 Den Fokus steuern	395
10.1.3 GestureDetector	398
10.1.4 Buttons	401
10.1.5 Checkbox und Switch	403
<b>10.2 Wischen, Schieben und Ziehen</b>	405
10.2.1 Dismissible	405
10.2.2 Draggable	408
10.2.3 Slider	410
<b>10.3 Dialogs und die Snackbar</b>	413
10.3.1 Dialogs	413
10.3.2 Snackbar	417
<b>11 Themes und Styles</b>	419
<b>11.1 BoxDecoration</b>	419
11.1.1 Ein Kreis mit Container	420
11.1.2 Spezifische Radien mit Container	421
11.1.3 Gradient in einer BoxDecoration	422
11.1.4 Schatten	422
<b>11.2 Clipping</b>	423
11.2.1 ClipRect und ClipPath	423
11.2.2 ClipRRect	425
11.2.3 ClipOval	426
11.2.4 ClipBehavior	426
<b>11.3 Text</b>	427
11.3.1 TextStyle	427
11.3.2 RichText	428
11.3.3 DefaultTextStyle	429
<b>11.4 ThemeData und CupertinoThemeData</b>	430
<b>11.5 Effekte durch Widgets</b>	431
11.5.1 Rotation	431
11.5.2 Opacity	432
11.5.3 Transform	432
11.5.4 BackdropFilter	433

<b>12 Animationen</b>	435
<b>12.1 Implizite Animationen</b>	435
12.1.1 Die Animated-Widgets	435
12.1.2 Überflieger: Hero in der Navigation	438
<b>12.2 Tween-Animationen</b>	440
12.2.1 TweenAnimationBuilder	440
12.2.2 Ein eigenes Tweenable	441
<b>12.3 Explizite Animationen</b>	443
12.3.1 AnimatedIcon	443
12.3.2 Transitions	445
<b>12.4 AnimatedWidget und der AnimatedBuilder</b>	448
12.4.1 AnimatedWidget	448
12.4.2 AnimatedBuilder	450
12.4.3 Widget oder Builder?	451
<b>13 Arbeiten mit Canvas und Paint</b>	453
<b>13.1 CustomPaint und CustomPainter</b>	453
13.1.1 CustomPaint integrieren	453
13.1.2 Formen und Farbe	454
13.1.3 Bilder	458
13.1.4 Text	458
13.1.5 Clipping	460
13.1.6 Weitere Operationen	460
<b>13.2 Zusammenfassung</b>	461
<b>TEIL III Inhalte und Accessibility</b>	
<b>14 Assets und Medien</b>	465
<b>14.1 Bilder</b>	465
14.1.1 Bilder als Assets einbinden	466
14.1.2 Ein Image über eine URL anzeigen	466
14.1.3 Wenn ein Bild im Speicher liegt	467
14.1.4 Ein Bild vom Dateisystem auslesen	467

14.1.5	Caching .....	467
14.1.6	Eigenschaften von Image .....	468
14.1.7	SGVs anzeigen .....	470
<b>14.2</b>	<b>Icons .....</b>	<b>471</b>
14.2.1	IconData .....	471
14.2.2	IconButton .....	473
14.2.3	ImageIcon .....	473
14.2.4	IconTheme .....	474
<b>14.3</b>	<b>Videos .....</b>	<b>474</b>
<b>14.4</b>	<b>Schriften .....</b>	<b>474</b>
14.4.1	Eine Schrift hinzufügen .....	475
14.4.2	Eine Schrift benutzen .....	476
<b>14.5</b>	<b>Textressourcen .....</b>	<b>476</b>

---

## **15 Den Nutzerkreis vergrößern: Localization und Accessibility**

---

<b>15.1</b>	<b>Localization .....</b>	<b>479</b>
15.1.1	Die Lokalisierung in das Projekt einbinden .....	480
15.1.2	Eigene Texte und Übersetzungen .....	481
15.1.3	Datum und Zahlen .....	487
15.1.4	Integration von intl in der IDE .....	492
15.1.5	Im Betrieb die Sprache ändern .....	497
15.1.6	Werte an die lokalisierten Texte übergeben .....	500
<b>15.2</b>	<b>Accessibility .....</b>	<b>500</b>
15.2.1	Schriftgröße .....	500
15.2.2	Screenreader .....	503
<b>15.3</b>	<b>Zusammenfassung .....</b>	<b>511</b>

---

## **16 Die Cloud: Unendliche Weiten**

---

<b>16.1</b>	<b>Daten mit GET beziehen .....</b>	<b>514</b>
<b>16.2</b>	<b>Ein eigener (HTTP-)Client .....</b>	<b>515</b>
16.2.1	Ein AppRequest .....	516
16.2.2	AppResponse .....	516

16.2.3	Der AppHttpClient .....	517
16.2.4	Die Methode get .....	517
<b>16.3</b>	<b>POST, PUT und PATCH .....</b>	<b>519</b>
16.3.1	http.post .....	519
16.3.2	POST im AppHttpClient .....	520
16.3.3	http.put und http.patch .....	523
16.3.4	PUT und PATCH im AppHttpClient .....	524
<b>16.4</b>	<b>Delete .....</b>	<b>527</b>
<b>16.5</b>	<b>Mehr zu Requests .....</b>	<b>528</b>
16.5.1	Header .....	529
16.5.2	Body .....	529
16.5.3	Query Parameters .....	529
16.5.4	Einen Proxy in Verbindung mit einem Client einsetzen .....	530
<b>16.6</b>	<b>Das Package dio .....</b>	<b>531</b>
16.6.1	Integration und erste Schritte .....	531
16.6.2	dio.get .....	533
16.6.3	BaseOptions .....	534
16.6.4	Query Parameters in dio .....	535
16.6.5	dio.post .....	536
16.6.6	Options .....	536
16.6.7	Das CancellationToken .....	537
16.6.8	Den Fortschritt eines Requests anzeigen .....	537
16.6.9	Interceptors .....	538
<b>16.7</b>	<b>JSON-Serialisierung und -Deserialisierung .....</b>	<b>539</b>
16.7.1	json_serializable integrieren .....	540
16.7.2	Die Klasse TodoDto .....	541
16.7.3	Den build_runner hinzufügen .....	542
16.7.4	fromJson und toJson .....	543
16.7.5	TodoDto.fromJson einsetzen .....	544
<b>16.8</b>	<b>Die Webview .....</b>	<b>544</b>
16.8.1	Die Standard-Webview .....	545
16.8.2	Die InAppWebView .....	559
<b>16.9</b>	<b>Zusammenfassung .....</b>	<b>563</b>

<b>17 Persistieren: Caches und Datenbanken</b>	565
<b>17.1 shared_preferences</b>	565
17.1.1 Das Package integrieren	566
17.1.2 In den Cache schreiben	566
17.1.3 Aus dem Cache lesen	567
17.1.4 Weitere nützliche Methoden	568
17.1.5 Einträge entfernen	569
<b>17.2 flutter_cache_manager</b>	569
17.2.1 Integration und Konfiguration	570
17.2.2 Dateien herunterladen	571
17.2.3 Ablegen, löschen und das direkte Lesen aus dem Cache	572
<b>17.3 Datenbanken</b>	574
17.3.1 Die Basis: sqflite	575
17.3.2 Mehr Abstraktion durch moor	581
<b>17.4 Zusammenfassung</b>	587

## TEIL IV Sensoren und Hardware

<b>18 Mit der Außenwelt kommunizieren</b>	591
<b>18.1 Geräteinformationen</b>	591
<b>18.2 Bewegungssensoren und Batterie</b>	593
<b>18.3 Verbindungsdaten</b>	595
<b>18.4 Berechtigungen</b>	595
<b>18.5 Kamera und Fotobibliothek</b>	597
<b>18.6 Standortbestimmung</b>	598
<b>18.7 Zusammenfassung</b>	599

<b>19 Den Kanal wechseln: Channels zur Plattform</b>	601
<b>19.1 Method Channels</b>	602
19.1.1 Ein Flutter-Plugin erstellen	603
19.1.2 Die Anatomie eines Plugins	604
19.1.3 Zusammenfassung	607

<b>19.2 Foreign Function Interface (FFI) .....</b>	608
19.2.1 Überblick .....	608
19.2.2 Die »C-Library« .....	608
19.2.3 Verwendung der Schnittstellen in Dart .....	610
<b>19.3 Hybride Apps .....</b>	612
19.3.1 Native Views in Flutter .....	612
19.3.2 Eine Android-View einbinden .....	613
19.3.3 Eine iOS-View einbinden .....	616

## TEIL V Endgame

<b>20 Analytics, Tests und Releases</b>	623
<b>20.1 Logging .....</b>	623
<b>20.2 Analytics .....</b>	625
<b>20.3 Debugging .....</b>	625
20.3.1 Hot-Reload, Hot-Restart und Full Restart .....	625
20.3.2 Debug-Tools .....	626
<b>20.4 Profiling .....</b>	628
<b>20.5 Testing .....</b>	629
20.5.1 Unit-Testing .....	630
20.5.2 Widget-Testing .....	637
<b>20.6 Releases .....</b>	640
20.6.1 Ein eigenes Pub erstellen .....	641
20.6.2 Android Releases .....	643
20.6.3 iOS-Releases .....	644
 Index .....	645