



# Pragmatic Flutter

Building Cross-Platform Mobile Apps  
for Android, iOS, Web & Desktop

**Priyanka Tyagi**



**CRC Press**  
Taylor & Francis Group

# Pragmatic Flutter



**Taylor & Francis**

Taylor & Francis Group

<http://taylorandfrancis.com>

# Pragmatic Flutter

Building Cross-Platform Mobile Apps  
for Android, iOS, Web, & Desktop

Priyanka Tyagi



**CRC Press**

Taylor & Francis Group

Boca Raton London New York

---

CRC Press is an imprint of the  
Taylor & Francis Group, an **informa** business

First edition published 2022

by CRC Press

6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742

and by CRC Press

2 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

© 2022 Taylor & Francis Group, LLC

CRC Press is an imprint of Taylor & Francis Group, LLC

The right of Priyanka Tyagi to be identified as author of this work has been asserted by her in accordance with sections 77 and 78 of the Copyright, Designs and Patents Act 1988.

Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, access [www.copyright.com](http://www.copyright.com) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. For works that are not available on CCC please contact [mpkbookspermissions@tandf.co.uk](mailto:mpkbookspermissions@tandf.co.uk)

*Trademark notice:* Product or corporate names may be trademarks or registered trademarks and are used only for identification and explanation without intent to infringe.

ISBN: 978-0-367-61209-2 (hbk)

ISBN: 978-1-032-05565-7 (pbk)

ISBN: 978-1-003-10463-6 (ebk)

Typeset in Times LT Std

by KnowledgeWorks Global Ltd.

*To my husband Krishna  
And my children Kalp and Krisha  
Who inspired me to start writing this book and finish it!*



**Taylor & Francis**

Taylor & Francis Group

<http://taylorandfrancis.com>

---

# Contents

|  |      |
|--|------|
| Preface.....   | xiii |
| Author .....   | xv   |
| <b>Chapter 1</b> Dart Fundamentals: A Quick Reference to Dart 2..... | 1    |
| The main Function.....   | 1    |
| Variables & Data Types.....  | 2    |
| Collections.....   | 2    |
| Spread Operator .....  | 3    |
| Transform List Items .....   | 3    |
| Filtering .....  | 3    |
| Adding Item.....   | 4    |
| Removing Item .....  | 4    |
| Adding Multiple Items .....  | 4    |
| Intersection of Two Set(s).....                                      | 5    |
| Union of Two Set(s) .....  | 5    |
| Checking for Key .....   | 6    |
| Checking for Value.....  | 7    |
| Accessing all Values.....  | 7    |
| Iterating Key/Value Pairs .....                                      | 7    |
| Functions .....  | 8    |
| Classes.....   | 10   |
| Conclusion .....   | 12   |
| References .....   | 12   |
| <b>Chapter 2</b> Introduction to Flutter .....                       | 15   |
| Cross-Platform Solutions.....  | 15   |
| Why Flutter .....  | 18   |
| Conclusion .....   | 20   |
| References .....   | 20   |
| <b>Chapter 3</b> Setting Up Environment.....                         | 23   |
| System Requirements for macOS.....                                   | 23   |
| Setting Up Flutter SDK.....  | 23   |
| Setting Up for the Android Platform.....                             | 25   |
| Setting Up for the iOS Platform.....                                 | 27   |
| Setting Up for Web.....  | 28   |
| Setting Up for Desktop.....  | 29   |
| Source Code Online .....   | 31   |



|  |     |
|--|-----|
| Setting Up Editor.....                                   | 32  |
| Conclusion.....  | 32  |
| References.....  | 32  |
| <b>Chapter 4</b> Flutter Project Structure.....          | 35  |
| Choosing Flutter Channel.....                            | 35  |
| Creating Flutter Project.....                            | 36  |
| Cross-Platform Flutter Project Structure.....            | 39  |
| Running Default App: Android, iOS, Web, and Desktop..... | 40  |
| Running Code Samples.....                                | 41  |
| Useful Commands.....                                     | 45  |
| Conclusion.....  | 46  |
| References.....  | 46  |
| <b>Chapter 5</b> Flutter App Structure.....              | 47  |
| Flutter Widgets.....                                     | 48  |
| Display ‘Hello Books’ Text.....                          | 49  |
| Add Cushion Around the Text.....                         | 51  |
| Center the Text.....                                     | 53  |
| App Anatomy #1.....                                      | 53  |
| App Anatomy #2.....                                      | 53  |
| Conclusion.....  | 61  |
| References.....  | 61  |
| <b>Chapter 6</b> Flutter Widgets.....                    | 63  |
| Image Widget.....  | 63  |
| ToggleButtons Widget.....                                | 65  |
| TextField Widget.....                                    | 68  |
| FutureBuilder Async Widget.....                          | 71  |
| Placeholder Widget.....                                  | 74  |
| StreamBuilder Async Widget.....                          | 76  |
| AlertDialog Widget.....                                  | 78  |
| Conclusion.....  | 83  |
| References.....  | 84  |
| <b>Chapter 7</b> Building Layouts.....                   | 87  |
| Revisiting <i>HelloBooksApp</i> Layout.....              | 87  |
| Types of Layout Widgets.....                             | 89  |
| Container Widget.....                                    | 90  |
| Padding Widget.....                                      | 97  |
| ConstrainedBox Widget.....                               | 99  |
| SizedBox Widget.....                                     | 101 |
| Row Widget.....  | 103 |

|                   |  |            |
|-------------------|--|------------|
|                   | IntrinsicHeight Widget.....                            | 105        |
|                   | Column Widget.....                                     | 110        |
|                   | IntrinsicWidth Widget.....                             | 112        |
|                   | ListView Widget.....                                   | 115        |
|                   | GridView Widget.....                                   | 117        |
|                   | Table Widget.....                                      | 117        |
|                   | Stack Widget.....                                      | 119        |
|                   | IndexedStack Widget.....                               | 121        |
|                   | Conclusion .....                                       | 123        |
|                   | References .....                                       | 124        |
| <b>Chapter 8</b>  | <b>Responsive Interfaces.....</b>                      | <b>127</b> |
|                   | FittedBox Widget.....                                  | 127        |
|                   | Expanded Widget.....                                   | 128        |
|                   | Flexible Widget.....                                   | 132        |
|                   | FractionallySizedBox Widget.....                       | 134        |
|                   | LayoutBuilder Widget.....                              | 137        |
|                   | Wrap Widget.....                                       | 137        |
|                   | Conclusion .....                                       | 139        |
|                   | References .....                                       | 141        |
| <b>Chapter 9</b>  | <b>Building User Interface for BooksApp .....</b>      | <b>143</b> |
|                   | The <i>BooksApp</i> Interface.....                     | 143        |
|                   | The <i>BooksApp</i> Anatomy .....                      | 143        |
|                   | Implementing User Interface.....                       | 144        |
|                   | Custom Widget: BooksListing.....                       | 148        |
|                   | Conclusion .....                                       | 151        |
|                   | References .....                                       | 151        |
| <b>Chapter 10</b> | <b>Flutter Themes .....</b>                            | <b>153</b> |
|                   | Global Theme.....                                      | 153        |
|                   | Modularizing Themes .....                              | 157        |
|                   | Using Custom Fonts .....                               | 159        |
|                   | Local Theme .....                                      | 160        |
|                   | Switching Themes .....                                 | 163        |
|                   | Conclusion .....                                       | 167        |
|                   | References .....                                       | 168        |
| <b>Chapter 11</b> | <b>Persisting Data.....</b>                            | <b>169</b> |
|                   | Light Theme (Default).....                             | 169        |
|                   | Key/Value Data Store (Shared Preferences Plugin) ..... | 169        |

|   |     |
|---|-----|
| Local Database (Moor Library) .....             | 173 |
| Light Theme on Multiple Platforms .....         | 180 |
| Dark Theme on Multiple Platforms .....          | 182 |
| Conclusion .....                                | 186 |
| References .....                                | 187 |
| <b>Chapter 12</b> Integrating REST API.....     | 189 |
| What Is an API? .....                           | 189 |
| Flutter Configuration .....                     | 190 |
| API Key .....                                   | 190 |
| API Endpoint .....                              | 192 |
| Building Simple Interface .....                 | 193 |
| Running Code.....                               | 196 |
| Adding Entitlement .....                        | 198 |
| AndroidManifest.xml.....                        | 199 |
| Conclusion .....                                | 200 |
| References .....                                | 201 |
| <b>Chapter 13</b> Data Modeling.....            | 203 |
| Parsing JSON .....                              | 203 |
| ListView Widget: Listing Entries .....          | 207 |
| Custom Widget: BookTile .....                   | 209 |
| Constructing Data Model .....                   | 217 |
| Converting API Response to BookModel List ..... | 220 |
| Run the Code.....                               | 221 |
| Conclusion .....                                | 227 |
| References .....                                | 228 |
| <b>Chapter 14</b> Navigation and Routing.....   | 229 |
| Simple BookDetailsPage Screen.....              | 229 |
| Navigator Widget.....                           | 232 |
| Direct Navigation .....                         | 233 |
| Detecting Gesture.....                          | 233 |
| Passing Data .....                              | 234 |
| Static Navigation .....                         | 235 |
| Detecting Gesture.....                          | 236 |
| Passing Data .....                              | 236 |
| Dynamic Navigation.....                         | 237 |
| The generateRoute() Function.....               | 237 |
| Detecting Gesture.....                          | 238 |
| Passing Data .....                              | 238 |
| Conclusion .....                                | 239 |
| References .....                                | 239 |

|                   |   |     |
|-------------------|---|-----|
| <b>Chapter 15</b> | The Second Page – BookDetailsPage Widget..... | 241 |
|                   | Anatomy of BookDetailsPage Widget.....        | 241 |
|                   | BookDetailsPage Screen’s Layout .....         | 242 |
|                   | Implementing BookDetailsPage Widget.....      | 243 |
|                   | Conclusion .....                              | 249 |
|                   | References .....                              | 249 |
| <br>              |   |     |
| <b>Chapter 16</b> | Introduction to State Management .....        | 251 |
|                   | Revisiting Default <i>CounterApp</i> .....    | 251 |
|                   | Vanilla Pattern.....                          | 252 |
|                   | Conclusion .....                              | 255 |
|                   | References .....                              | 256 |
| <br>              |   |     |
| <b>Chapter 17</b> | ValueNotifier .....                           | 257 |
|                   | Using <i>ValueNotifier</i> Approach .....     | 257 |
|                   | Conclusion .....                              | 262 |
|                   | References .....                              | 262 |
| <br>              |   |     |
| <b>Chapter 18</b> | Provider and ChangeNotifier.....              | 265 |
|                   | What Is Provider .....                        | 265 |
|                   | What Is ChangeNotifier .....                  | 266 |
|                   | Anatomy of <i>CounterApp</i> .....            | 266 |
|                   | Increasing Counter .....                      | 267 |
|                   | Custom Widget: CountWidget .....              | 267 |
|                   | Finished Implementation.....                  | 268 |
|                   | Conclusion .....                              | 270 |
|                   | References .....                              | 270 |
| <br>              |   |     |
| <b>Chapter 19</b> | BLoC Design Pattern .....                     | 271 |
|                   | What Is BLoC Pattern? .....                   | 271 |
|                   | Anatomy of <i>CounterApp</i> .....            | 271 |
|                   | Basic BLoC Pattern Implementation.....        | 273 |
|                   | Event StreamController.....                   | 276 |
|                   | Event Sink .....                              | 276 |
|                   | Event Stream/CounterBloc Constructor .....    | 276 |
|                   | State StreamController.....                   | 277 |
|                   | State Sink .....                              | 277 |
|                   | State Stream.....                             | 277 |
|                   | Improvised BLoC Pattern Implementation .....  | 279 |
|                   | State StreamController.....                   | 280 |
|                   | State Sink .....                              | 280 |

|   |     |
|---|-----|
| State Stream.....                             | 281 |
| Implementing BLoC Pattern Using Library.....  | 282 |
| Conclusion.....                               | 285 |
| References .....                              | 285 |
| <b>Chapter 20</b> Unit Testing .....          | 287 |
| Package Dependencies .....                    | 287 |
| The <i>CounterApp</i> .....                   | 287 |
| Test File .....                               | 292 |
| Running Unit Tests.....                       | 294 |
| Conclusion.....                               | 295 |
| References .....                              | 295 |
| <b>Chapter 21</b> Widget Testing.....         | 297 |
| Package Dependency.....                       | 297 |
| Testing Widgets.....                          | 298 |
| Running Widget Tests .....                    | 305 |
| Conclusion.....                               | 305 |
| References .....                              | 305 |
| <b>Chapter 22</b> Integration Testing.....    | 307 |
| Package Dependency.....                       | 307 |
| Preview <i>CounterApp</i> Code.....           | 307 |
| Test Pair Files .....                         | 310 |
| Writing Integration Tests.....                | 310 |
| Running Integration Tests .....               | 313 |
| Conclusion.....                               | 314 |
| References .....                              | 314 |
| <b>Chapter 23</b> Rolling into the World..... | 315 |
| Adding Launcher Icon.....                     | 315 |
| Releasing Android Apps .....                  | 316 |
| Releasing iOS Apps.....                       | 322 |
| Releasing Web Apps.....                       | 327 |
| Releasing Desktop (macOS) Apps .....          | 330 |
| Conclusion.....                               | 331 |
| References .....                              | 331 |
| <b>Index</b> .....                            | 333 |

---

# Preface

Have you ever thought of creating beautiful blazing-fast native apps for iOS and Android from a single codebase? Have you dreamt of taking your native apps to the web and desktop without costing a fortune? If so, this book is the right place to start your journey of developing cross-platform apps.

Google's Flutter software development kit (SDK) is the latest way of developing beautiful, fluid cross-platform apps for Android, iOS, Web, and Desktops (macOS, Linux, Windows). Google's new Fuchsia OS user interface (UI) is implemented using Flutter as well. Learning to develop mobile apps with Flutter opens the door to multiple devices, form-factors, and platforms from a single codebase.

You don't need any prior experience using Dart to follow along with this book. However, it's recommended to have some familiarity with writing code in one of the object-oriented programming languages like Java and Python. You will pick up the fundamentals of Dart 2 language in the first chapter. We will learn to structure and organize the multi-platform Flutter project and test the setup by running the same code on Android, iOS, web, and desktop platforms. Next, we will explore basic Flutter widgets along with layout widgets while building a non-trivial UI. Later on, we will continue learning to build responsive layouts for different screen sizes and form-factors.

We will be organizing and applying themes and styles, handling user input, and gestures. Then we will move on to fetching data over the network, integrating, and consuming REST API in app. We will build a *BooksApp* to display books listing from Google Books API and integrate this API to fetch book data. Different types of testing strategies will be introduced to develop solid and high-quality apps. You will get hands-on experience using state management solutions, data modeling, routing, and navigation for multi-screen apps. You will learn to use Flutter plugins to persist data in the app. This book concludes by giving the pointers to deploy and distribute your Flutter app across all four platforms.

When you finish this book, you will have a solid foundational knowledge of Flutter SDK that will help you move forward in your journey of building great and successful mobile apps that can be deployed to Android, iOS, web, and desktop (macOS) from a single code base.

Throughout the book, *italic* is used to render file, folder, and Flutter application names. The monospace font is used to represent code snippets, variable names, and data structures. All examples used in this book are available online in the GitHub repository: [https://github.com/ptyagicodecamp/pragmatic\\_flutter](https://github.com/ptyagicodecamp/pragmatic_flutter).

Last but definitely not least, I want to acknowledge the input of my fellow developers Rajalakshmi Balaji, Preetika Tyagi, and Snehal Patil without whom this book wouldn't be possible. I am also thankful to the Flutter community worldwide for sharing their expertise and knowledge for developing great cross-platform applications.



**Taylor & Francis**

Taylor & Francis Group

<http://taylorandfrancis.com>

---

# Author

**Priyanka Tyagi** is a Software Engineering Manager at Willow Innovations, Inc., which builds products that improve the lives and health of women. At Willow Innovations, Inc., she is helping to build a better wearables experience by using Flutter and Bluetooth Low Energy (BLE) technologies.

Priyanka has many years of experience designing and developing software, web, and mobile systems for a diverse range of industries from automobile and e-commerce to entertainment and EdTech to health and wellness. Her expertise lies in Flutter, Android, Firebase, Mobile SDKs, AWS/Google cloud-based solutions, cross-platform apps, and game-based learning. In her previous roles, she has worked with Disney Interactive as Lead Android Engineer. She has also helped various start-ups with her software engineering consulting services.

Priyanka loves to share her tech explorations around mobile apps development and other software engineering topics in her tech blog: <https://priyankatyagi.dev/>.

She is an Internet of Things (IOT) enthusiast and volunteers her time at local public schools to introduce computer science to young minds. She volunteers for the Hour of Code initiative to inspire young developers in the tech world for many years. Priyanka is passionate about mentoring aspiring developers to get started in the tech industry. She earned an MS in computer science from Illinois Institute of Technology, Chicago.

Priyanka lives with her husband and two kids in beautiful California. She loves to read, bake, and hike in her free time.





**Taylor & Francis**

Taylor & Francis Group

<http://taylorandfrancis.com>

---

# 1 Dart Fundamentals

## *A Quick Reference to Dart 2*

Dart programming language is developed by Google. It has been around since 2011. However, it has gained popularity recently since Google announced Flutter software development kit (SDK) for developing cross-platform-applications. Dart aims to help developers build web and mobile applications effectively. It works for building production-ready solutions for client applications as well as for the server-side. Dart has an Ahead-of-Time (AOT) compiler that compiles predictable native code quickly for the target platform. It is optimized to build customized user interfaces natively for multiple platforms. The Dart is a developers-friendly programming language. It is easy for developers coming from different programming language backgrounds to learn Dart without much effort.

The Dart 2 (Announcing Dart 2 Stable and the Dart Web Platform) is the latest version of Dart language including the rewrite of many features, improved performance and productivity. This chapter covers the basics of Dart 2 language syntaxes (Google, 2020) to get started with the journey of building applications using Flutter. A prior understanding of object-oriented programming is needed to follow along with the material. Dart 2 and Dart will be used to infer the Dart 2 programming language in this book. In the upcoming topics, we'll review some of the language features with the help of examples.

### **THE main FUNCTION**

The `main()` function is the entry point for any Dart program. The following code snippet will print "Hello Dart" on the console. Open a 'Terminal' at MacOS or its Windows or Linux equivalent. Create a file say `hello.dart`, and copy the following code snippet in it.

```
---
void main() {
    print("Hello Dart");
}
---
```

### **RUNNING DART PROGRAM**

Go back to the 'Terminal' and execute the file using `dart hello.dart`. The "Hello Dart" is printed on the console.

```
---
$ dart hello.dart
Hello Dart
---
```

## VARIABLES & DATA TYPES

In Dart, you can use the `var` keyword to infer the underlying data type. The following code snippet shows declaring variable `data` to hold the numeric value `1`.

```

```
var data = 1;
print(data);
```

```

The above code snippet will print the `1` on the console. The `runtimeType` (`runtimeType` property) property gives the runtime type of the object it's called on. The following code snippet will print the data type of the `data` variable as `int` on the console.

```

```
var data = 1;
print(data.runtimeType);
```

```

It's valid for the `data` variable to get reassigned to a value from a different data type.

There's another keyword, `dynamic`, to assign a value to a variable similar to the `var` keyword. The difference between the two is that for the `dynamic` keyword, the same variable can be reassigned to a different data type, as shown in the code snippet below:

```

```
// Assigning int to dynamicData
dynamic dynamicData = 1;
// Prints `int` as data type on console
print(dynamicData.runtimeType);

//Re-assigning dynamicData to String data type
dynamicData = "I'm a string now";
// Prints `String` as data type on console
print(dynamicData.runtimeType);
```

```

### SOURCE CODE ONLINE

The source code for this example ([Chapter01: Quick Reference to Dart2 \(Variables & Data Types\)](#)) is available at GitHub.

## COLLECTIONS

### LIST

Dart's `List` (`List<E>` class) is a collection that holds indexable objects. An empty list can be declared using two square brackets `[]`. List items can be put inside these brackets separated by commas.

```

```
List emptyList = [];
```

```

```
//Checking if List is empty
var result = emptyList.isEmpty;
print(result);
```
```

## SPREAD OPERATOR

Dart 2.3 introduced the spread operator (Spread Operator) and null aware spread operator to combine multiple lists into one. Let's create a list `theList` with two items and a null list `theNullList`. If you want to merge these two lists into another list, say `anotherList`, it can be done using spread (...) and null aware spread (?...) operators.

```
```
List theList = ['Dart', 'Kotlin'];
//null list
List theNullList;

// Spread operator ... - Flatten the list and to be merged
with another list
// Null spread operator ...? - Helps to avoid runtime
exceptions - iterator called on null.
List anotherList = ['Java', ...theList, ...?theNullList];

// Printing the merged list created using spread operators
print(anotherList);

// Output
[Java, Dart, Kotlin]
```
```

## TRANSFORM LIST ITEMS

The `map` on the `List` can be called to transform the list items. Let's append the word 'Language' to objects of the `theList` list.

```
```
List theList = ['Dart', 'Kotlin'];
result = theList.map((e) => "$e Language").toList();
print(result);

// Output
[Dart Language, Kotlin Language]
```
```

## FILTERING

The `where` method on `List` can help filter items meeting specific criteria. In the following code snippet, we are filtering the words containing the letter 'a'. Since only 'Dart' matches these criteria, `result` will hold only the word 'Dart'.

```

    ...
    List theList = ['Dart', 'Kotlin'];
    result = theList.where((element) =>
    element.toString().contains('a'));
    print(result); // Dart contains letter 'a'

    // Output
    (Dart)
    ...
  
```

## SET

The Set (`Set<E>` class) data structure holds a collection of objects only once. The duplicates are not allowed when storing data in `Set`. Let's create two sets to learn the usage.

```

    ...
    Set langSet = {'Dart', 'Kotlin', 'Swift'};
    Set sdkSet = {'Flutter', 'Android', 'iOS'};
    ...
  
```

## ADDING ITEM

Let's add a new item 'Java' in the `langSet` and print the final set.

```

    ...
    // Adding 'Java' to Set
    langSet.add('Java');
    print(langSet);

    // Output
    {Dart, Kotlin, Swift, Java}
    ...
  
```

## REMOVING ITEM

Let's remove the newly added item 'Java' from `langSet` and print the final set.

```

    ...
    // Remove Java from Set
    langSet.remove('Java');
    print(langSet);

    // Output
    {Dart, Kotlin, Swift}
    ...
  
```

## ADDING MULTIPLE ITEMS

More than one item can be added all at once in `Set` using the `addAll()` method. It accepts the list of items. Let's add multiple items to both sets.

```

...
// Adding multiple items to each set
langSet.addAll(['C#', 'Java']);
sdkSet.addAll(['C#', 'Xamarin']);
print(langSet);
print(sdkSet);

// Output
{Dart, Kotlin, Swift, C#, Java}
{Flutter, Android, iOS, C#, Xamarin}
...

```

## INTERSECTION OF TWO SET(S)

The `intersection` method on `Set` returns the shared items from both sets. The `langSet` and `sdkSet` both contain the `'C#'` item in them. The intersection on those sets will return the `'C#'`.

```

...
// Find Intersection of two sets (common items)
result = langSet.intersection(sdkSet);
print(result); // C#

// Output
{C#}
...

```

## UNION OF TWO SET(S)

The `union` returns items combined from `langSet` and `sdkSet` without any duplicates. Both sets contain `'C#'`, and it'll be added in the `result` only once.

```

...
// Find a Union of two sets. No duplicates.
result = langSet.union(sdkSet);
print(result);

// Output
{Dart, Kotlin, Swift, C#, Java, Flutter, Android, iOS, Xamarin}
...

```

## MAP

The `Map` (`Map<K, V>` class) data structure is a collection that contains key/value pairs. A value is accessed using the key for that entry. Let's create a `Map` with a key of `int` type, and value is of `String` type.

One way to create a `Map` `intToStringMap` is as below:

```

...
var intToStringMap = Map<int, String>();
...

```

The new key/value pair can be added as shown in the code snippet below:

```

---
intToStringMap[1] = '1';
intToStringMap[2] = '2';
---

```

The first or last entry of the map can be accessed using `first` and `last` properties of `entries` (`entries` property) iterable.

```

---
// first Map entry
result = intToStringMap.entries.first;
print(result);
// Output
MapEntry(1: 1)

// last Map entry
result = intToStringMap.entries.last;
print(result);
// Output
MapEntry(2: 2)
---

```

Let's create a `Map` with the key and value of type `String` as below:

```

---
var techMap = {
  'Flutter': 'Dart',
  'Android': 'Kotlin',
  'iOS': 'Swift',
};
---

```

## CHECKING FOR KEY

The `containsKey(String key)` method on the map returns a boolean true or false depending on whether the given `key` exists in the Map or not.

```

---
// Returns boolean. true if key is found, otherwise false
result = techMap.containsKey('Flutter');
print(result);

// Output
true
---

```

## CHECKING FOR VALUE

The `containsValue(String value)` method returns the boolean value as `true` if the given value exists in the Map.

```

---
// Checking if value is present in the Map
result = techMap.containsValue('Dart');
print(result);

// Output
true
---
```

## ACCESSING ALL VALUES

All values of the key/value pairs in Map collection can be accessed calling `foreach` on `values` as below:

```

---
// Prints all values
techMap.values.forEach((element) {
    print("$element");
});

// Output
Dart
Kotlin
Swift
---
```

## ITERATING KEY/VALUE PAIRS

All key/value pairs in Map collection are iterated, as shown in the code below:

```

---
// Iterates over all key-value pairs and prints them
techMap.entries.forEach((element) {
    print("${element.value} is used for developing
    ${element.key} applications.");
});

// Output
Dart is used for developing Flutter applications.
Kotlin is used for creating Android applications.
Swift is used for developing iOS applications.
---
```

## SOURCE CODE ONLINE

The source code for this example ([Chapter01: Quick Reference to Dart2 \(Collections\)](#)) is available at [GitHub](#).



## FUNCTIONS

Let's create a function that checks if the passed argument is 'Flutter' or not and returns a boolean value. It returns `true` if it's precise 'Flutter' or `false` otherwise. Such functions are known as 'Named Function' because the function's name describes what they are intended to.

```

---
bool isFlutter(String str) {
    return str == 'Flutter';
}
// Using Named Function
dynamic result = isFlutter("Flutter");
print(result);

//Output
true
---
```

## FUNCTION WITH OPTIONAL PARAMETERS

Let's create a function `concat(...)` that joins two strings together when the second string is available. In such a case, the second string can be passed as optional using square brackets `[]`.

```

---
String concat(String str1, [String str2]) {
    return str2 != null ? "$str1 $str2" : str1;
}

// Usage
result = concat("Priyanka", "Tyagi");
print(result);

//Output
Priyanka Tyagi
---
```

## NAMED PARAMETERS

The other way to provide optional parameters is to use named parameters. They can be passed using curly braces `{}`.

```

---
// Named Parameters: Function with optional parameters in
// curly braces
String concat2(String str1, {String str2}) {
    return str2 != null ? "$str1 $str2" : str1;
}

```

```
// Using function with optional params with curly braces{}
result = concat2("Priyanka", str2: "Tyagi");
print(result);
// Output
Priyanka Tyagi
```
```

## PASSING FUNCTION AS PARAMETER

The Dart 2 programming language allows passing a function as a parameter to another function. Let's create a function `int subtract(int a, int b)` to find the difference between the two values 'a' and 'b'. The method `calculate(...)` takes two numbers and a function as a parameter to operate on the values passed.

```
```
int subtract(int a, int b) {
    return a > b ? a - b : b - a;
}
// Passing Function as parameter
int calculate(int value1, int value2, Function(int, int)
function) {
    return function(value1, value2);
}
//Passing function 'subtract' as parameter
result = calculate(5, 4, subtract);
print(result);

// Output
1
```
```

## ARROW SYNTAX

The arrow syntax can be used to write functions in one line. Let's write the function to add two numbers using arrow syntax as below:

```
```
// Arrow Syntax
int add(int a, int b) => a + b;

// Using Arrow Syntax
result = add(5, 4);
print(result);

//Output
9
```
```

## ANONYMOUS FUNCTION

Anonymous functions don't have a name and can be assigned to a variable either using the keyword `var` or `Function`.

```

...
// Anonymous Function
Function anonymousAdd = (int a, int b) {
    return a + b;
};

// Using anonymous functions.
// Calling function variable `anonymousAdd`
result = anonymousAdd(4, 5);
print(result);

//Output
9
...

```

## SOURCE CODE ONLINE

The source code for this example ([Chapter01: Quick Reference to Dart2 \(Functions\)](#)) is available at [GitHub](#).

## CLASSES

Dart classes are created using the keyword `class`. Let's define a class `Person` to represent a person in the real world. This person has a name, an age, and the food it eats.

```

...
class Person {
    String name;
    int age;
    String food;
}
...

```

## CONSTRUCTOR

Dart supports easy to use constructors. Let's see two types of constructors. The short-form constructor looks like below. The first part is required. The parameters inside `[]` are optional.

```

...
Person(this.name, [this.age]);

//Usage
Person person = Person("Priyanka");
...

```

Another type of constructor is the named constructor. All parameters are enclosed in the curly braces '{}'.  
 Dart

```

  ...
  Person.basicInfo({this.name, this.age});

  //Usage
  Person child = Person.basicInfo(name: "Krisha", age: 6);
  ...

```

## GETTERS

The getters in Dart classes are defined using the ``get`` keyword. Let's create a getter ``personName`` to get the name.

```

  ...
  String get personName => this.name;
  ...

```

## SETTERS

The setters in the Dart classes are defined using the ``set`` keyword. Let's create a setter ``personName`` to set the name as below:

```

  ...
  set personName(String value) => this.name = value;
  ...

```

## METHOD

Let's add a method to the class ``Person`` to define the eating behavior. The method ``eats(String food)`` takes the food as ``String`` and assigns it to the class ``food`` property.

```

  ...
  void eats(String food) {
    this.food = food;
  }
  // Usage
  Person child = Person.basicInfo(name: "Krisha", age: 6);
  child.eats("Pizza");
  ...

```

Let's override the method ``toString()`` from the `Object` (`Object` class) class to print a custom message. Every class in Dart extends from the base `Object` class.

```

  ...
  String toString() {
    return "My name is $name, and I like to eat $food";
  }

```

```
//Usage
print(child.toString());

//Output on console
My name is Krisha, and I like to eat Pizza
```
```

## CASCADING SYNTAX

Dart supports cascading syntaxes. It's useful in assigning the values to properties and methods at once using two dots.

```
```
child
  . .name = 'Kalp'
  . .eats("Pasta");
```
```

The setters can also be called using cascaded syntax as shown in the code snippet below:

```
```
child
  . .personName = 'Tanmay'
  . .eats("Pesto");
```
```

## SOURCE CODE ONLINE

The source code for this example ([Chapter01: Quick Reference to Dart2 \(Classes\)](#)) is available at GitHub.

## CONCLUSION

In this chapter, we reviewed the basic concepts and syntaxes useful to getting started with developing Flutter application quickly and efficiently. You were introduced to the structure of the Dart program. You developed your understanding of declaring and using variables in Dart language. We explored methods to store data in collections using various data structures like List, Map, and Set. You also learned to write reusable code using functions. Finally, you learned to architect and structure the code base using Classes and methods.

## REFERENCES

- Dart Dev. (2020, 12 14). *entries property*. Retrieved from api.dart.dev: <https://api.dart.dev/stable/2.10.3/dart-core/Map/entries.html>
- Google. (2020, 11 13). *A tour of the Dart language*. Retrieved from Official Dart Documentation: <https://dart.dev/guides/language/language-tour>

- Google. (2020, 11 13). *List<E> class*. Retrieved from Official Dart Language Documentation: <https://api.dart.dev/stable/2.10.3/dart-core/List-class.html>
- Google. (2020, 11 13). *Map<K, V> class*. Retrieved from Official Dart Language Documentation: <https://api.dart.dev/stable/2.10.3/dart-core/Map-class.html>
- Google. (2020, 11 13). *Object class*. Retrieved from Official Dart Language Documentation: <https://api.dart.dev/stable/2.10.3/dart-core/Object-class.html>
- Google. (2020, 11 13). *runtimeType property*. Retrieved from Official Dart Documentation: <https://api.dart.dev/stable/2.10.2/dart-core/Object/runtimeType.html>
- Google. (2020, 11 13). *Set<E> class*. Retrieved from Official Dart Language Documentation: <https://api.dart.dev/stable/2.10.3/dart-core/Set-class.html>
- Google. (2020, 11 13). *Spread Operator*. Retrieved from Official Dart Language Documentation: <https://dart.dev/guides/language/language-tour#spread-operator>
- Moore, K. (2018, 08 07). *Announcing Dart 2 Stable and the Dart Web Platform*. Retrieved from Medium: <https://medium.com/dartlang/dart-2-stable-and-the-dart-web-platform-3775d5f8eac7>
- Tyagi, P. (2020, 11 13). *Chapter01: Quick Reference to Dart2 (Classes)*. Retrieved from Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter01/classes.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter01/classes.dart)
- Tyagi, P. (2020, 11 13). *Chapter01: Quick Reference to Dart2 (Collections)*. Retrieved from Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter01/collections.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter01/collections.dart)
- Tyagi, P. (2020, 11 13). *Chapter01: Quick Reference to Dart2 (Functions)*. Retrieved from Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter01/functions.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter01/functions.dart)
- Tyagi, P. (2020, 11 13). *Chapter01: Quick Reference to Dart2 (Variables & Data Types)*. Retrieved from Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter01/variables.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter01/variables.dart)



**Taylor & Francis**

Taylor & Francis Group

<http://taylorandfrancis.com>

## Dart Fundamentals

Dart Dev . (2020, 12 14). *entries property*. Retrieved from [api.dart.dev](https://api.dart.dev/stable/2.10.3/dart-core/Map/entries.html): <https://api.dart.dev/stable/2.10.3/dart-core/Map/entries.html>

Google . (2020, 11 13). *A tour of the Dart language*. Retrieved from Official Dart Documentation: <https://dart.dev/guides/language/language-tour>

Google . (2020, 11 13). *List<E> class*. Retrieved from Official Dart Language Documentation: <https://api.dart.dev/stable/2.10.3/dart-core/List-class.html>

Google . (2020, 11 13). *Map<K, V> class*. Retrieved from Official Dart Language Documentation: <https://api.dart.dev/stable/2.10.3/dart-core/Map-class.html>

Google . (2020, 11 13). *Object class*. Retrieved from Official Dart Language Documentation: <https://api.dart.dev/stable/2.10.3/dart-core/Object-class.html>

Google . (2020, 11 13). *runtimeType property*. Retrieved from Official Dart Documentation: <https://api.dart.dev/stable/2.10.2/dart-core/Object/runtimeType.html>

Google . (2020, 11 13). *Set<E> class*. Retrieved from Official Dart Language Documentation: <https://api.dart.dev/stable/2.10.3/dart-core/Set-class.html>

Google . (2020, 11 13). *Spread Operator*. Retrieved from Official Dart Language Documentation: <https://dart.dev/guides/language/language-tour#spread-operator>

Moore, K. (2018, 08 07). *Announcing Dart 2 Stable and the Dart Web Platform*. Retrieved from Medium: <https://medium.com/dartlang/dart-2-stable-and-the-dart-web-platform-3775d5f8eac7>

Tyagi, P. (2020, 11 13). *Chapter01: Quick Reference to Dart2 (Classes)*. Retrieved from Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter01/classes.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter01/classes.dart)

Tyagi, P. (2020, 11 13). *Chapter01: Quick Reference to Dart2 (Collections)*. Retrieved from Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter01/collections.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter01/collections.dart)

Tyagi, P. (2020, 11 13). *Chapter01: Quick Reference to Dart2 (Functions)*. Retrieved from Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter01/functions.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter01/functions.dart)

Tyagi, P. (2020, 11 13). *Chapter01: Quick Reference to Dart2 (Variables & Data Types)*. Retrieved from Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter01/variables.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter01/variables.dart)

## Introduction to Flutter

Adobe . (2020, 11 17). *Adobe PhoneGap*. Retrieved from PhoneGap: <https://phonegap.com/>

Apple . (2020, 11 16). *Develop iOS Applications*. Retrieved from Apple Developer: <https://developer.apple.com/develop/>

Apple . (2020, 11 16). *Swift*. Retrieved from Apple Developer Documentation: <https://developer.apple.com/swift/>

Dart Team . (2020, 11 17). *Dart*. Retrieved from Dart Website: <https://dart.dev/>

Develop Android Applications . (2020, 11 16). Retrieved from Android Developers: <https://developer.android.com/>

Flutter Dev . (2020, 12 15). *Hot reload*. Retrieved from flutter.dev: <https://flutter.dev/docs/development/tools/hot-reload>

Flutter Team . (2020, 11 17). *Cupertino (iOS-style) widgets*. Retrieved from Flutter Documentation: <https://flutter.dev/docs/development/ui/widgets/cupertino>

Flutter Team . (2020, 11 17). *Flutter*. Retrieved from GitHub repository for Flutter SDK: <https://github.com/flutter>

Google . (2020, 11 16). *android studio*. Retrieved from Android Developers: <https://developer.android.com/studio>

Google . (2020, 11 17). *Material Foundation*. Retrieved from Material Design: <https://material.io/design>

Google . (2020, 12 15). *Firebase Platform*. Retrieved from Firebase: <https://firebase.google.com/>



JetBrains & Open-source Contributors . (2020, 11 16). *A modern programming language that makes developers happier*. Retrieved from Kotlin Language: <https://kotlinlang.org/>

Jetbrains . (2020, 11 16). *IntelliJ IDEA*. Retrieved from Jetbrains: <https://www.jetbrains.com/idea/>

Leler, W. (2017, 08 20). *What's Revolutionary about Flutter*. Retrieved from hackernoon: <https://hackernoon.com/whats-revolutionary-about-flutter-946915b09514>

Microsoft . (2020, 11 16). *Visual Studio*. Retrieved from code.visualstudio.com: <https://code.visualstudio.com/>

Ravn, M. (2018, 08 28). *Flutter Platform Channels*. Retrieved from Medium: <https://medium.com/flutter/flutter-platform-channels-ce7f540a104e>

Wikipedia . (2020, 11 16). *Java*. Retrieved from Wikipedia: [https://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))

Wikipedia . (2020, 11 16). *Objective-C*. Retrieved from Wikipedia: <https://en.wikipedia.org/wiki/Objective-C>

Wikipedia . (2020, 11 16). *Reactive Programming*. Retrieved from Wikipedia: [https://en.wikipedia.org/wiki/Reactive\\_programming](https://en.wikipedia.org/wiki/Reactive_programming)

Wikipedia . (2020, 11 16). *Software development kit*. Retrieved from Wikipedia: [https://en.wikipedia.org/wiki/Software\\_development\\_kit](https://en.wikipedia.org/wiki/Software_development_kit)

## Setting Up Environment

Android Developers . (2020, 12 14). *Configure on-device developer options*. Retrieved from developer.android.com: <https://developer.android.com/studio/debug/dev-options>

Apple . (2020, 11 17). *Xcode*. Retrieved from Apple Apps: <https://apps.apple.com/us/app/xcode/id497799835>

Apple . (2020, 11 17). *Xcode*. Retrieved from Apple Developer: <https://developer.apple.com/xcode/>

Apple . (2020, 12 15). *Apple Developer Program*. Retrieved from developer.apple.com: <https://developer.apple.com/programs/>

Cocoapods.org . (2020, 12 15). *CocoaPods*. Retrieved from cocoapods.org: <https://cocoapods.org/>

Everything curl . (2020, 11 17). *command line tool and library*. Retrieved from Curl Website: <https://curl.se/>

Flutter Dev . (2020, 11 17). *Deploy to iOS devices*. Retrieved from Flutter Dev: <https://flutter.dev/docs/get-started/install/macos#deploy-to-ios-devices>

Flutter Dev . (2020, 11 17). *Desktop support for Flutter*. Retrieved from Flutter Development: <https://flutter.dev/desktop>

Flutter Team . (2020, 11 17). *Building a web application with Flutter*. Retrieved from Flutter Web: <https://flutter.dev/docs/get-started/web>

Flutter Team . (2020, 11 17). *Flutter Stable 1.22.4*. Retrieved from Download Flutter: [https://storage.googleapis.com/flutter\\_infra/releases/stable/macos/flutter\\_macos\\_1.22.4-stable.zip](https://storage.googleapis.com/flutter_infra/releases/stable/macos/flutter_macos_1.22.4-stable.zip)

Flutter Team . (2020, 11 17). *Get the Flutter SDK*. Retrieved from Flutter Dev: <https://flutter.dev/docs/get-started/install/macos#get-sdk>

Flutter Team . (2020, 11 17). *Install*. Retrieved from Flutter Dev: <https://flutter.dev/docs/get-started/install>

Flutter Team . (2020, 11 17). *Install Xcode*. Retrieved from Flutter Dev: <https://flutter.dev/docs/get-started/install/macos#install-xcode>

Flutter Team . (2020, 11 17). *iOS setup*. Retrieved from Flutter Dev: <https://flutter.dev/docs/get-started/install/macos#ios-setup>

Flutter Team . (2020, 11 17). *Set up the iOS simulator*. Retrieved from Flutter Dev: <https://flutter.dev/docs/get-started/install/macos#set-up-the-ios-simulator>

Flutter Team . (2020, 11 17). *Update your path*. Retrieved from Flutter Dev: <https://flutter.dev/docs/get-started/install/macos#update-your-path>

Git . (2020, 11 17). *Download for macOS*. Retrieved from Install Git: <https://git-scm.com/download/mac>

Google . (2020, 11 17). *Android setup*. Retrieved from Flutter Dev: <https://flutter.dev/docs/get-started/install/macos#android-setup>

Google . (2020, 11 17). *Install Android Studio*. Retrieved from Flutter Dev: <https://flutter.dev/docs/get-started/install/macos#install-android-studio>

Priyanka Tyagi . (2020, 11 17). *testapp*. Retrieved from GitHub: <https://github.com/ptyagicodecamp/testapp>

Wikipedia . (2020, 11 17). *iPhone SE (2nd generation)*. Retrieved from Wikipedia: [https://en.wikipedia.org/wiki/IPhone\\_SE\\_\(2nd\\_generation\)](https://en.wikipedia.org/wiki/IPhone_SE_(2nd_generation))

## Flutter Project Structure

Dart Team . (2020, 11 17). *List<E> class*. Retrieved from Dart Dev: <https://api.dart.dev/stable/2.8.4/dart-core/List-class.html>

Flutter Dev . (2020, 11 17). *BottomAppBar class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/material/BottomAppBar-class.html>

Flutter Dev . (2020, 11 17). *FloatingActionButton class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/material/FloatingActionButton-class.html>

Tyagi, P. (2020, 11 17). *Chapter04: Flutter Project Structure*. Retrieved from Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/app\\_scaffold\\_demo](https://github.com/ptyagicodecamp/app_scaffold_demo)

Tyagi, P. (2020, 11 17). *Chapter04: Flutter Project Structure (hello\_books)*. Retrieved from Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/hello\\_books](https://github.com/ptyagicodecamp/hello_books)

Tyagi, P. (2021). Chapter 05: Flutter App Structure. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

## Flutter App Structure

Dart Dev . (2020, 11 17). *List<E> class*. Retrieved from Dart Dev: <https://api.dart.dev/stable/2.8.4/dart-core/List-class.html>

Flutter Dev . (2020, 11 17). *AppBar class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/material/AppBar-class.html>

Flutter Dev . (2020, 11 17). *Card Widget*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/material/Card-class.html>

Flutter Dev . (2020, 11 17). *Center class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/Center-class.html>

Flutter Dev . (2020, 11 17). *Components*. Retrieved from Flutter Dev: <https://material.io/components>

Flutter Dev . (2020, 11 17). *Container Widget*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/Container-class.html>

Flutter Dev . (2020, 11 17). *FloatingActionButton*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/material/FloatingActionButton-class.html>

Flutter Dev . (2020, 11 17). *FloatingActionButton class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/material/FloatingActionButton-class.html>

Flutter Dev . (2020, 11 17). *Icon Widget*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/Icon-class.html>

Flutter Dev . (2020, 11 17). *MaterialApp class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/material/MaterialApp-class.html>

Flutter Dev . (2020, 11 17). *Scaffold class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/material/Scaffold-class.html>

Flutter Dev . (2020, 11 17). *setState method*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/State/setState.html>

Flutter Dev . (2020, 11 17). *StatefulWidget*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/StatefulWidget-class.html>

Flutter Dev . (2020, 11 17). *StatelessWidget class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/StatelessWidget-class.html>

Flutter Dev . (2020, 11 17). *Text class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/Text-class.html>

Flutter Dev . (2020, 11 17). *Text Widget*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/Text-class.html>

Google . (2020, 11 17). *Checkbox class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/material/Checkbox-class.html>

Google . (2020, 11 17). *Material Design*. Retrieved from material.io: <https://material.io/design>

Google . (2020, 11 17). *Radio<T> class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/material/Radio-class.html>

Google . (2020, 11 17). *SafeArea class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/SafeArea-class.html>

Google . (2020, 11 17). *Text class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/Text-class.html>

Google . (2020, 11 17). *TextField class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/material/TextField-class.html>

Tyagi, P. (2020, 11 17). *Flutter App Structure: Add cushion around text*. Retrieved from Chapter05: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter05/main\\_05\\_1.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter05/main_05_1.dart)

Tyagi, P. (2020, 11 17). *Flutter App Structure: App Anatomy#2*. Retrieved from Chapter05: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter05/main\\_05\\_3.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter05/main_05_3.dart)

Tyagi, P. (2020, 11 17). *Flutter App Structure: Center the text*. Retrieved from Chapter05: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter05/main\\_05\\_2.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter05/main_05_2.dart)

Tyagi, P. (2020, 11 17). *Flutter App Structure: Display 'Hello Books' text*. Retrieved from Chapter05: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter05/main\\_05\\_0.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter05/main_05_0.dart)

Tyagi, P. (2020, 11 17). *Flutter App Structure: HelloBooksApp*. Retrieved from Chapter05: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter05/main\\_05\\_4.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter05/main_05_4.dart)

Tyagi, P. (2021). Chapter 04: Flutter Project Structure. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

Tyagi, P. (2021). Chapter 10: Flutter Themes. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

## Flutter Widgets

Flutter Dev . (2020, 11 18). *AlertDialog class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/material/AlertDialog-class.html>

Flutter Dev . (2020, 11 18). *Asynchronous programming: streams*. Retrieved from Dart Dev: <https://dart.dev/tutorials/language/streams>

Flutter Dev . (2020, 11 18). *ElevatedButton class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/material/ElevatedButton-class.html>

Flutter Dev . (2020, 11 18). *Future<T> class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/dart-async/Future-class.html>

Flutter Dev . (2020, 11 18). *FutureBuilder<T> class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/FutureBuilder-class.html>

Flutter Dev . (2020, 11 18). *Image class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/Image-class.html>

Flutter Dev . (2020, 11 18). *Placeholder class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/Placeholder-class.html>

Flutter Dev . (2020, 11 18). *Stream<T> class*. Retrieved from Dart Dev: <https://api.dart.dev/stable/2.10.3/dart-async/Stream-class.html>

Flutter Dev . (2020, 11 18). *StreamBuilder class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/StreamBuilder-class.html>

Flutter Dev . (2020, 11 18). *TextEditingController class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/TextEditingController-class.html>

Flutter Dev . (2020, 11 18). *TextField class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/material/TextField-class.html>

Flutter Dev . (2020, 11 18). *ToggleButtons class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/material/ToggleButtons-class.html>

Tyagi, P. (2020, 11 18). *Flutter Widgets: AlertDialog*. Retrieved from Chapter06: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter06/widgets/alert\\_dialog.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter06/widgets/alert_dialog.dart)

Tyagi, P. (2020, 11 18). *Flutter Widgets: FutureBuilder Async Widget*. Retrieved from Chapter06: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter06/widgets/futurebuilder\\_widget.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter06/widgets/futurebuilder_widget.dart)

Tyagi, P. (2020, 11 18). *Flutter Widgets: Image*. Retrieved from Chapter06: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter06/widgets/image\\_widget.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter06/widgets/image_widget.dart)

Tyagi, P. (2020, 11 18). *Flutter Widgets: Placeholder*. Retrieved from Chapter06: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter06/widgets/placeholder\\_widget.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter06/widgets/placeholder_widget.dart)

Tyagi, P. (2020, 11 18). *Flutter Widgets: StreamBuilder Async Widget*. Retrieved from Chapter06: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter06/widgets/streambuilder\\_widget.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter06/widgets/streambuilder_widget.dart)

Tyagi, P. (2020, 11 18). *Flutter Widgets: TextField*. Retrieved from Chapter06: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter06/widgets/textfield.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter06/widgets/textfield.dart)

Tyagi, P. (2020, 11 18). *Flutter Widgets: ToggleButtons*. Retrieved from Chapter06: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter06/widgets/togglebuttons\\_widget.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter06/widgets/togglebuttons_widget.dart)

Tyagi, P. (2021). Chapter 05: Flutter App Structure. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

## Building Layouts

Android Developer . (2020, 12 20). *Density-independent Pixels*. Retrieved from developer.android.com: <https://developer.android.com/guide/topics/resources/more-resources.html#Dimension>

Dart Dev . (2020, 11 18). *clamp method*. Retrieved from Dart Dev: <https://api.dart.dev/stable/2.10.2/dart-core/num/clamp.html>

Flutter Dev . (2020, 11 18). *alignment property*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/widgets/Container/alignment.html>

Flutter Dev . (2020, 11 18). *BoxConstraints class*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/rendering/BoxConstraints-class.html>

Flutter Dev . (2020, 11 18). *BoxDecoration class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/painting/BoxDecoration-class.html>

Flutter Dev . (2020, 11 18). *Center class*. Retrieved from Flutter API Dev: <https://api.flutter.dev/flutter/widgets/Center-class.html>

Flutter Dev . (2020, 11 18). *color property*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/widgets/Container/color.html>

Flutter Dev . (2020, 11 18). *Column class*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/widgets/Column-class.html>

Flutter Dev . (2020, 11 18). *ConstrainedBox class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/ConstrainedBox-class.html>

Flutter Dev . (2020, 11 18). *Container class*. Retrieved from Flutter Api Dev: <https://api.flutter.dev/flutter/widgets/Container-class.html>

Flutter Dev . (2020, 11 18). *devicePixelRatio property*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/dart-ui/Window/devicePixelRatio.html>

Flutter Dev . (2020, 11 18). *EdgeInsets class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/painting/EdgeInsets-class.html>

Flutter Dev . (2020, 11 18). *EdgeInsetsGeometry class*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/painting/EdgeInsetsGeometry-class.html>

Flutter Dev . (2020, 11 18). *Expanded class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/Expanded-class.html>

Flutter Dev . (2020, 11 18). *FittedBox class*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/widgets/FittedBox-class.html>

Flutter Dev . (2020, 11 18). *Flex class*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/widgets/Flex-class.html>

Flutter Dev . (2020, 11 18). *Flexible class*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/widgets/Flexible-class.html>

Flutter Dev . (2020, 11 18). *Flow class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/Flow-class.html>

Flutter Dev . (2020, 11 18). *FractionallySizedBox class*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/widgets/FractionallySizedBox-class.html>

Flutter Dev . (2020, 11 18). *FractionColumnWidth class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/rendering/FractionColumnWidth-class.html>

Flutter Dev . (2020, 11 18). *GestureDetector class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/GestureDetector-class.html>

Flutter Dev . (2020, 11 18). *GridView class*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/widgets/GridView-class.html>

Flutter Dev . (2020, 11 18). *GridView.count constructor*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/GridView/GridView.count.html>

Flutter Dev . (2020, 11 18). *IndexedStack class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/IndexedStack-class.html>

Flutter Dev . (2020, 11 18). *IntrinsicHeight class*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/widgets/IntrinsicHeight-class.html>

Flutter Dev . (2020, 11 18). *IntrinsicWidth class*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/widgets/IntrinsicWidth-class.html>

Flutter Dev . (2020, 11 18). *Layout Widgets*. Retrieved from Flutter Dev: <https://flutter.dev/docs/development/ui/widgets/layout>

Flutter Dev . (2020, 11 18). *LayoutBuilder class*. Retrieved from Flutter Team: <https://api.flutter.dev/flutter/widgets/LayoutBuilder-class.html>

Flutter Dev . (2020, 11 18). *LimitedBox class*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/widgets/LimitedBox-class.html>

Flutter Dev . (2020, 11 18). *ListBody class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/ListBody-class.html>

Flutter Dev . (2020, 11 18). *ListView class*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/widgets/ListView-class.html>

Flutter Dev . (2020, 11 18). *margin property*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/widgets/Container/margin.html>

Flutter Dev . (2020, 12 18). *Material Components widgets*. Retrieved from Flutter Dev: <https://flutter.dev/docs/development/ui/widgets/material>

Flutter Dev . (2020, 11 18). *OverflowBox class*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/widgets/OverflowBox-class.html>

Flutter Dev . (2020, 11 18). *Padding class*. Retrieved from Flutter Api Dev: <https://api.flutter.dev/flutter/widgets/Padding-class.html>

Flutter Dev . (2020, 11 18). *padding property*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/Container/padding.html>

Flutter Dev . (2020, 11 18). *Row class*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/widgets/Row-class.html>

Flutter Dev . (2020, 11 18). *SizedBox class*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/widgets/SizedBox-class.html>

Flutter Dev . (2020, 11 18). *SizedBox.expand constructor*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/SizedBox/SizedBox.expand.html>

Flutter Dev . (2020, 11 18). *SizedOverflowBox class*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/widgets/SizedOverflowBox-class.html>

Flutter Dev . (2020, 11 18). *Stack class*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/widgets/Stack-class.html>

Flutter Dev . (2020, 11 18). *Table class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/Table-class.html>

Flutter Dev . (2020, 11 18). *Transform class*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/widgets/Transform-class.html>

Flutter Dev . (2020, 11 18). *transform property*. Retrieved from Api Flutter Dev: <https://api.flutter.dev/flutter/widgets/Container/transform.html>

Flutter Dev . (2020, 11 18). *Wrap class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/Wrap-class.html>

Flutter Team . (2020, 11 18). *SingleChildScrollView class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/SingleChildScrollView-class.html>

Flutter Team . (2020, 11 18). *TextStyle class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/painting/TextStyle-class.html>

Google . (2020, 11 18). *Layout widgets*. Retrieved from Flutter Dev: <https://flutter.dev/docs/development/ui/widgets/layout>

Google . (2020, 11 18). *Layouts in Flutter*. Retrieved from Flutter Dev: <https://flutter.dev/docs/development/ui/layout>

Tyagi, P. (2020, 11 18). *Building Layouts: Column Widget*. Retrieved from Chapter07: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter07/layouts/multi/column.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter07/layouts/multi/column.dart)

Tyagi, P. (2020, 11 18). *Building Layouts: ConstrainedBox Widget*. Retrieved from Chapter07: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter07/layouts/single/constrained\\_box.dart#L109:L113](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter07/layouts/single/constrained_box.dart#L109:L113)

Tyagi, P. (2020, 11 18). *Building Layouts: Container Widget*. Retrieved from Chapter07: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter07/layouts/single/container.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter07/layouts/single/container.dart)

Tyagi, P. (2020, 11 18). *Building Layouts: GridView Widget*. Retrieved from Chapter07: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter07/layouts/multi/grid\\_view.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter07/layouts/multi/grid_view.dart)

Tyagi, P. (2020, 11 18). *Building Layouts: IndexedStack Widget*. Retrieved from Chapter07: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter07/layouts/multi/indexed\\_stack.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter07/layouts/multi/indexed_stack.dart)

Tyagi, P. (2020, 11 18). *Building Layouts: IntrinsicHeight Widget*. Retrieved from Chapter07: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter07/layouts/single/intrinsic\\_height.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter07/layouts/single/intrinsic_height.dart)

Tyagi, P. (2020, 11 18). *Building Layouts: IntrinsicWidth Widget*. Retrieved from Chapter07: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter07/layouts/single/intrinsic\\_width.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter07/layouts/single/intrinsic_width.dart)

Tyagi, P. (2020, 11 18). *Building Layouts: ListView Widget*. Retrieved from Chapter07: Pragmatic Flutter GitHub Repo:

[https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter07/layouts/multi/listview.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter07/layouts/multi/listview.dart)

Tyagi, P. (2020, 11 18). *Building Layouts: Padding Widget*. Retrieved from Chapter07: Pragmatic Flutter GitHub Repo:

[https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter07/layouts/single/padding.dart#L92:L98](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter07/layouts/single/padding.dart#L92:L98)

Tyagi, P. (2020, 11 18). *Building Layouts: Row Widget*. Retrieved from Chapter07: Pragmatic Flutter GitHub Repo:

[https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter07/layouts/multi/row.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter07/layouts/multi/row.dart)

Tyagi, P. (2020, 11 18). *Building Layouts: SizedBox Widget*. Retrieved from Chapter07: Pragmatic Flutter GitHub Repo:

[https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter07/layouts/single/sized\\_box.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter07/layouts/single/sized_box.dart)

Tyagi, P. (2020, 11 18). *Building Layouts: Stack Widget*. Retrieved from Chapter07: Pragmatic Flutter GitHub Repo:

[https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter07/layouts/multi/stack.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter07/layouts/multi/stack.dart)

Tyagi, P. (2020, 11 18). *Building Layouts: Table Widget*. Retrieved from Chapter07: Pragmatic Flutter GitHub Repo:

[https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter07/layouts/multi/table.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter07/layouts/multi/table.dart)

Tyagi, P. (2020, 11 18). *Revisiting HelloBooksApp source code*. Retrieved from Chapter05: Pragmatic Flutter GitHub Repo:

[https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter05/main\\_05\\_3.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter05/main_05_3.dart)

Tyagi, P. (2021). Chapter 05: Flutter App Structure. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

Tyagi, P. (2021). Chapter 08: Responsive Interfaces. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

## Responsive Interfaces

Flutter Team . (2020, 11 19). *Align class*. Retrieved from Flutter Dev:

<https://api.flutter.dev/flutter/widgets/Align-class.html>

Flutter Team . (2020, 11 19). *ElevatedButton class*. Retrieved from Flutter Dev:

<https://api.flutter.dev/flutter/material/ElevatedButton-class.html>

Flutter Team . (2020, 11 19). *Expanded class*. Retrieved from Flutter Dev:

<https://api.flutter.dev/flutter/widgets/Expanded-class.html>

Flutter Team . (2020, 11 19). *FittedBox class*. Retrieved from Api Flutter Dev:

<https://api.flutter.dev/flutter/widgets/FittedBox-class.html>

Flutter Team . (2020, 11 19). *Flexible class*. Retrieved from Api Flutter Dev:

<https://api.flutter.dev/flutter/widgets/Flexible-class.html>

Flutter Team . (2020, 11 19). *FractionallySizedBox class*. Retrieved from Api Flutter Dev:

<https://api.flutter.dev/flutter/widgets/FractionallySizedBox-class.html>

Flutter Team . (2020, 11 19). *LayoutBuilder class*. Retrieved from Api Flutter Dev:

<https://api.flutter.dev/flutter/widgets/LayoutBuilder-class.html>

Flutter Team . (2020, 11 19). *OverflowBox class*. Retrieved from Flutter Dev:

<https://api.flutter.dev/flutter/widgets/OverflowBox-class.html>

Flutter Team . (2020, 11 19). *SingleChildScrollView class*. Retrieved from Flutter Dev:

<https://api.flutter.dev/flutter/widgets/SingleChildScrollView-class.html>

Flutter Team . (2020, 11 19). *Widget Catalog*. Retrieved from Flutter Api Dev:

<https://flutter.dev/docs/development/ui/widgets>

Flutter Team . (2020, 11 19). *Wrap class*. Retrieved from Api Flutter Dev:

<https://api.flutter.dev/flutter/widgets/Wrap-class.html>

Tyagi, P. (2020, 11 19). *Responsive Interfaces: Expanded Widget*. Retrieved from Chapter08: Pragmatic Flutter GitHub Repo:

[https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter08/layouts/expanded.dart#L84](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter08/layouts/expanded.dart#L84)

Tyagi, P. (2020, 11 19). *Responsive Interfaces: FittedBox Widget*. Retrieved from Chapter08: Pragmatic Flutter GitHub Repo:

[https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter08/layouts/fitted\\_box.dart#L80:L84](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter08/layouts/fitted_box.dart#L80:L84)

Tyagi, P. (2020, 11 19). *Responsive Interfaces: Flexible Widget*. Retrieved from Chapter08: Pragmatic Flutter GitHub Repo:

[https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter08/layouts/flexible.dart#L77:L96](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter08/layouts/flexible.dart#L77:L96)

Tyagi, P. (2020, 11 19). *Responsive Interfaces: FractionallySizedBox Widget*. Retrieved from Chapter08: Pragmatic Flutter GitHub Repo:

[https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter08/layouts/fractionally\\_sized\\_box.dart#L27:L49](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter08/layouts/fractionally_sized_box.dart#L27:L49)

Tyagi, P. (2020, 11 19). *Responsive Interfaces: LayoutBuilder Widget*. Retrieved from Chapter08: Pragmatic Flutter GitHub Repo:

[https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter08/layouts/layoutbuilder.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter08/layouts/layoutbuilder.dart)

Tyagi, P. (2020, 11 19). *Responsive Interfaces: Wrap Widget*. Retrieved from Chapter08: Pragmatic Flutter GitHub Repo:

[https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter08/layouts/wrap.dart#L27:L39](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter08/layouts/wrap.dart#L27:L39)

## Building User Interface for BooksApp

Flutter Dev . (2020, 12 21). *Card class*. Retrieved from Flutter Dev:

<https://api.flutter.dev/flutter/material/Card-class.html>

Google . (2020, 11 20). *Card class*. Retrieved from Flutter Dev:

<https://api.flutter.dev/flutter/material/Card-class.html>

Google . (2020, 11 20). *Padding class*. Retrieved from Api Flutter Dev:

<https://api.flutter.dev/flutter/widgets/Padding-class.html>

json.org . (2020, 11 20). *Introducing JSON*. Retrieved from json.org: <https://www.json.org/>

Tyagi, P. (2020, 11 20). *Building User Interface for BooksApp*. Retrieved from Chapter09: Pragmatic Flutter GitHub Repo:

[https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter09/main\\_09.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter09/main_09.dart)

Tyagi, P. (2021). Chapter 05: Flutter App Structure. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

Tyagi, P. (2021). Chapter 12: Integrating REST API. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

Tyagi, P. (2021). Chapter 13: Data Modeling. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

## Flutter Themes

Burke, K. (2020, 11 20). *Pangolin*. Retrieved from Google Fonts:

<https://fonts.google.com/specimen/Pangolin#standard-styles>

Dart Team . (2020, 11 20). *avoid\_classes\_with\_only\_static\_members*. Retrieved from Dart

language: [https://dart-lang.github.io/linter/lints/avoid\\_classes\\_with\\_only\\_static\\_members.html](https://dart-lang.github.io/linter/lints/avoid_classes_with_only_static_members.html)

Google . (2020, 11 20). *AppBar class*. Retrieved from Flutter Dev:

<https://api.flutter.dev/flutter/material/AppBar-class.html>

Google . (2020, 11 20). *AppBarTheme property*. Retrieved from Flutter Dev:

<https://api.flutter.dev/flutter/material/ThemeData/appBarTheme.html>



Google . (2020, 11 20). *brightness property*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/material/AppBar/brightness.html>

Google . (2020, 11 20). *copyWith method*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/material/TextTheme/copyWith.html>

Google . (2020, 11 20). *IconButton class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/material/IconButton-class.html>

Google . (2020, 11 20). *iconTheme property*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/material/AppBar/iconTheme.html>

Google . (2020, 11 20). *IconThemeData class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/IconThemeData-class.html>

Google . (2020, 11 20). *TextTheme class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/material/TextTheme-class.html>

Google . (2020, 11 20). *Theme class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/material/Theme-class.html>

Google . (2020, 11 20). *ThemeData class*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/material/ThemeData-class.html>

Tyagi, P. (2020, 11 20). *Flutter Themes*. Retrieved from Chapter10: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter10/main\\_10\\_global.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter10/main_10_global.dart)

Tyagi, P. (2020, 11 20). *Flutter Themes: Local Theme*. Retrieved from Chapter10: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter10/main\\_10\\_local.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter10/main_10_local.dart)

Tyagi, P. (2020, 11 20). *Flutter Themes: Modularizing Themes*. Retrieved from Chapter10: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter10/main\\_10\\_global\\_modularize.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter10/main_10_global_modularize.dart)

Tyagi, P. (2020, 11 20). *Flutter Themes: Switching Themes*. Retrieved from Chapter10: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter10/main\\_10\\_switchingThemes.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter10/main_10_switchingThemes.dart)

Tyagi, P. (2020, 11 20). *Flutter Themes: Using Custom Fonts*. Retrieved from Chapter10: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter10/main\\_10\\_global\\_customfont.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter10/main_10_global_customfont.dart)

Tyagi, P. (2021). Chapter 09: Building User Interface for BooksApp. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

## Persisting Data

Apple . (2020, 11 22). *NSUserDefaults*. Retrieved from developer.apple.com: <https://developer.apple.com/documentation/foundation/nsuserdefaults>

Binder, S. (2020, 07 23). *sqlite3\_flutter\_libs*. Retrieved from Pub.dev: [https://pub.dev/packages/sqlite3\\_flutter\\_libs](https://pub.dev/packages/sqlite3_flutter_libs)

Binder, S. (2020, 11 20). *Moor: Persistence library for Dart*. Retrieved from <https://moor.simonbinder.eu/>

Binder, S. (2020, 11 22). *moor plugin*. Retrieved from [simonbinder.eu: https://pub.dev/packages/moor](https://pub.dev/packages/moor)

Binder, S. (2020, 11 22). *moor\_ffi plugin*. Retrieved from [pub.dev: https://pub.dev/packages/moor\\_ffi](https://pub.dev/packages/moor_ffi)

Binder, S. (2020, 11 22). *moor\_generator*. Retrieved from [pub.dev: https://pub.dev/packages/moor\\_generator](https://pub.dev/packages/moor_generator)

Binder, S. (2020, 11 22). *Phasing out the moor\_ffi package*. Retrieved from GitHub Issues: <https://github.com/simolus3/moor/issues/691>

dart.dev . (2020, 11 22). *build\_runner*. Retrieved from pub.dev: [https://pub.dev/packages/build\\_runner](https://pub.dev/packages/build_runner)

dart.dev . (2020, 11 22). *path*. Retrieved from pub.dev: <https://pub.dev/packages/path>

flutter.dev . (2020, 11 22). *path\_provider*. Retrieved from Pub.Dev: [https://pub.dev/packages/path\\_provider](https://pub.dev/packages/path_provider)

Google . (2020, 11 22). *dart:ffi library*. Retrieved from Dart Documentation: <https://api.dart.dev/stable/2.7.0/dart-ffi/dart-ffi-library.html>

Google . (2020, 11 22). *SharedPreferences*. Retrieved from developer.android.com: <https://developer.android.com/reference/android/content/SharedPreferences>

pub.dev . (2020, 11 03). *Shared preferences plugin*. Retrieved from pub.dev: [https://pub.dev/packages/shared\\_preferences](https://pub.dev/packages/shared_preferences)

Tyagi, P. (2020, 11 22). *Persisting Data: Moor Library*. Retrieved from Chapter11: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/tree/master/lib/chapter11/db](https://github.com/ptyagicodecamp/pragmatic_flutter/tree/master/lib/chapter11/db)

Tyagi, P. (2020, 11 22). *Persisting Data: Shared preferences*. Retrieved from Chapter11: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter11/main\\_11\\_shared\\_prefs.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter11/main_11_shared_prefs.dart)

Tyagi, P. (2021). Chapter 10: Flutter Themes. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

## Integrating REST API

Apple . (2020, 11 21). *App Sandbox*. Retrieved from developer.apple.com: [https://developer.apple.com/documentation/security/app\\_sandbox](https://developer.apple.com/documentation/security/app_sandbox)

Apple . (2020, 11 21). *com.apple.security.network.client*. Retrieved from developer.apple.com: [https://developer.apple.com/documentation/bundleresources/entitlements/com\\_apple\\_security\\_network\\_client](https://developer.apple.com/documentation/bundleresources/entitlements/com_apple_security_network_client)

Dart . (2020, 11 21). *Effective Dart: Style*. Retrieved from dart.dev: [https://dart.dev/guides/language/effective-dart/style#do-name-import-prefixes-using-lowercase\\_with\\_underscores](https://dart.dev/guides/language/effective-dart/style#do-name-import-prefixes-using-lowercase_with_underscores)

Dart Team . (2020, 12 22). *HttpResponse class*. Retrieved from Dart Dev: <https://api.dart.dev/stable/2.7.1/dart-io/HttpResponse-class.html>

Flutter Team . (2020, 11 21). *http*. Retrieved from pub.dev: <https://pub.dev/packages/http>

Google . (2020, 11 21). *Acquiring and using an API key*. Retrieved from Google Books API: <https://developers.google.com/books/docs/v1/using#APIKey>

Google . (2020, 11 21). *Books API v1 (Experimental)*. Retrieved from Google Books APIs: [https://developers.google.com/books/docs/overview#books\\_api\\_v1](https://developers.google.com/books/docs/overview#books_api_v1)

Google . (2020, 11 21). *Credentials*. Retrieved from Google APIs: <https://console.developers.google.com/apis/credentials>

Google . (2020, 11 21). *Google Books*. Retrieved from books.google.com: <https://books.google.com/>

Google . (2020, 11 21). *Using API keys*. Retrieved from Google Cloud: [https://cloud.google.com/docs/authentication/api-keys?visit\\_id=637269820995156061-804882371&rd=1](https://cloud.google.com/docs/authentication/api-keys?visit_id=637269820995156061-804882371&rd=1)

JSON Representation . (2020, 11 21). Retrieved from json.org: <https://www.json.org/json-en.html>

Tyagi, P. (2020, 11 21). *Calling REST API*. Retrieved from Chapter12: Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter12/main\\_12.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter12/main_12.dart)

Tyagi, P. (2021). Chapter 13: Data Modeling. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

Wikipedia . (2020, 11 21). *API*. Retrieved from Wikipedia: <https://en.wikipedia.org/wiki/API>

## Data Modeling

- Dart Dev . (2020, 12 22). *LinkedHashMap<K, V> class*. Retrieved from Dart API: <https://api.dart.dev/stable/2.8.4/dart-collection/LinkedHashMap-class.html>
- Dart Team. ( 2020, 11 24). *dart:convert library*. Retrieved from api.dart.dev: <https://api.dart.dev/stable/2.7.1/dart-convert/dart-convert-library.html>
- Dart Team . (2020, 11 24). *HttpResponse class*. Retrieved from api.dart.dev: <https://api.dart.dev/stable/2.7.1/dart-io/HttpResponse-class.html>
- Dart Team . (2020, 11 24). *Map<K, V> class*. Retrieved from api.dart.dev: <https://api.dart.dev/stable/2.8.4/dart-core/Map-class.html>
- Flutter Team . (2020, 11 24). *BuildContext class*. Retrieved from api.flutter.dev: <https://api.flutter.dev/flutter/widgets/BuildContext-class.html>
- Flutter Team . (2020, 11 24). *ListView class*. Retrieved from api.dart.dev: <https://api.flutter.dev/flutter/widgets/ListView-class.html>
- Google . (2020, 11 24). *Card class*. Retrieved from api.flutter.dev: <https://api.flutter.dev/flutter/material/Card-class.html>
- Google . (2020, 11 24). *Cards*. Retrieved from material.io: <https://material.io/components/cards>
- Google . (2020, 11 24). *Column class*. Retrieved from api.flutter.dev: <https://api.flutter.dev/flutter/widgets/Column-class.html>
- Google . (2020, 11 24). *Flexible class*. Retrieved from api.flutter.dev: <https://api.flutter.dev/flutter/widgets/Flexible-class.html>
- Google . (2020, 11 24). *ListView class*. Retrieved from api.flutter.dev: <https://api.flutter.dev/flutter/widgets/ListView-class.html>
- Google . (2020, 11 24). *Row class*. Retrieved from api.flutter.dev: <https://api.flutter.dev/flutter/widgets/Row-class.html>
- Google . (2020, 11 24). *Text class*. Retrieved from api.flutter.dev: <https://api.flutter.dev/flutter/widgets/Text-class.html>
- Tyagi, P. (2020, 11 24). *Chapter 13: BookModel (Part2)*. Retrieved from Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter13/part2/book.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter13/part2/book.dart)
- Tyagi, P. (2020, 11 24). *Chapter 13: BookTile (Part 1)*. Retrieved from Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter13/part1/booktile.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter13/part1/booktile.dart)
- Tyagi, P. (2020, 11 24). *Chapter 13: BookTile (Part 2)*. Retrieved from Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter13/part2/booktile.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter13/part2/booktile.dart)
- Tyagi, P. (2020, 11 24). *Chapter 13: Custom Widget (Part 1)*. Retrieved from Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter13/part1/main\\_13.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter13/part1/main_13.dart)
- Tyagi, P. (2020, 11 24). *Chapter 13: Data Modeling (main method - Part 2)*. Retrieved from Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter13/part2/main\\_13.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter13/part2/main_13.dart)
- Tyagi, P. (2020, 12 22). *Chapter 13*. Retrieved from Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/tree/master/lib/chapter13](https://github.com/ptyagicodecamp/pragmatic_flutter/tree/master/lib/chapter13)
- Tyagi, P. (2021). Chapter 12: Integrating REST API. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

## Navigation and Routing

Chapter 14: Dynamic Navigation . (2020, 11 25). Retrieved from Pragmatic Flutter GitHub Repo:

[https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter14/main\\_14\\_dynamic.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter14/main_14_dynamic.dart)

Dart Team . (2020, 11 25). *Map<K, V> class*. Retrieved from api.dart.dev:

<https://api.dart.dev/stable/2.8.4/dart-core/Map-class.html>

Flutter Team . (2020, 11 25). *MaterialPageRoute<T> class*. Retrieved from api.flutter.dev:

<https://api.flutter.dev/flutter/material/MaterialPageRoute-class.html>

Flutter Team . (2020, 11 25). *pushNamed<T extends Object> method*. Retrieved from Flutter Dev: <https://api.flutter.dev/flutter/widgets/Navigator/pushNamed.html>

Flutter Team . (2020, 11 25). *routes property*. Retrieved from api.flutter.dev:

<https://api.flutter.dev/flutter/material/MaterialApp/routes.html>

GestureDetector class . (2020, 11 24). Retrieved from Flutter Dev:

<https://api.flutter.dev/flutter/widgets/GestureDetector-class.html>

Google . (2020, 11 25). *Navigator class*. Retrieved from api.flutter.dev:

<https://api.flutter.dev/flutter/widgets/Navigator-class.html>

Google . (2020, 11 25). *onGenerateRoute property*. Retrieved from api.flutter.dev:

<https://api.flutter.dev/flutter/widgets/Navigator/onGenerateRoute.html>

Google . (2020, 11 25). *onGenerateRoute property*. Retrieved from Flutter Dev:

<https://api.flutter.dev/flutter/material/MaterialApp/onGenerateRoute.html>

Google . (2020, 11 25). *push<T extends Object> method*. Retrieved from Flutter Dev:

<https://api.flutter.dev/flutter/widgets/Navigator/push.html>

Google . (2020, 11 25). *Route class*. Retrieved from Flutter Dev:

<https://api.flutter.dev/flutter/widgets/Route-class.html>

Google . (2020, 11 25). *RouteSettings class*. Retrieved from Flutter Dev:

<https://api.flutter.dev/flutter/widgets/RouteSettings-class.html>

Tyagi, P. (2020, 11 25). *Chapter 14: Direct Navigation*. Retrieved from Pragmatic Flutter GitHub Repo:

[https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter14/main\\_14\\_direct.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter14/main_14_direct.dart)

Tyagi, P. (2020, 11 25). *Chapter 14: Static Navigation*. Retrieved from Pragmatic Flutter GitHub Repo:

[https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter14/main\\_14\\_static.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter14/main_14_static.dart)

Tyagi, P. (2021). Chapter 12: Integrating REST API. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

Tyagi, P. (2021). Chapter 15: The Second Page: BookDetails. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

Tyagi, P. (n.d.). Chapter 13: Data Modeling. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

## The Second Page – BookDetailsPage Widget

Flutter Team . (2020, 11 25). *url\_launcher plugin*. Retrieved from pub.dev:

[https://pub.dev/packages/url\\_launcher](https://pub.dev/packages/url_launcher)

Google . (2021, 11 25). *FloatingActionButton class*. Retrieved from api.flutter.dev:

<https://api.flutter.dev/flutter/material/FloatingActionButton-class.html>

Google . (2021, 11 25). *SingleChildScrollView class*. Retrieved from api.flutter.dev:

<https://api.flutter.dev/flutter/widgets/SingleChildScrollView-class.html>

Tyagi, P. (2020, 11 25). *Chapter 15: BookDetailsPage*. Retrieved from Pragmatic Flutter GitHub Repo:

[https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter15/book\\_details\\_page.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter15/book_details_page.dart)

Tyagi, P. (2021). Chapter 12: Integrating REST API. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

Tyagi, P. (2021). Chapter 13: Data Modeling. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

Tyagi, P. (2021). Chapter 14: Navigation & Routing. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

## Introduction to State Management

Google . (2020, 11 26). *Differentiate between ephemeral state and app state*. Retrieved from flutter.dev: <https://flutter.dev/docs/development/data-and-backend/state-mgmt/ephemeral-vs-app>

Google . (2020, 11 26). *setState method*. Retrieved from api.flutter.dev: <https://api.flutter.dev/flutter/widgets/State/setState.html>

Google . (2020, 11 26). *StatefulWidget class*. Retrieved from api.flutter.dev: <https://api.flutter.dev/flutter/widgets/StatefulWidget-class.html>

## ValueNotifier

Google . (2020, 11 26). *ChangeNotifier class*. Retrieved from api.flutter.dev: <https://api.flutter.dev/flutter/foundation/ChangeNotifier-class.html>

Google . (2020, 11 26). *foundation library*. Retrieved from api.flutter.dev: <https://api.flutter.dev/flutter/foundation/foundation-library.html>

Google . (2020, 11 26). *ValueListenableBuilder<T> class*. Retrieved from api.flutter.dev: <https://api.flutter.dev/flutter/widgets/ValueListenableBuilder-class.html>

Google . (2020, 11 26). *ValueNotifier<T> class*. Retrieved from api.flutter.dev: <https://api.flutter.dev/flutter/foundation/ValueNotifier-class.html>

Tyagi, P. (2020, 11 26). *Chapter 17: ValueNotifier*. Retrieved from Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter17/valuenotifier\\_co\\_unterapp.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter17/valuenotifier_co_unterapp.dart)

Tyagi, P. (2020, 11 26). *ValueListenable<T> class*. Retrieved from api.flutter.dev: <https://api.flutter.dev/flutter/foundation/ValueListenable-class.html>

Tyagi, P. (2021). Chapter 16: Introduction to State Management. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

Tyagi, P. (2021). Chapter 18: Provider & ChangeNotifier. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

## Provider and ChangeNotifier

Flutter Team . (2020, 11 26). *provider*. Retrieved from pub.dev: <https://pub.dev/packages/provider>

Google . (2020, 11 26). *ChangeNotifier class*. Retrieved from api.flutter.dev: <https://api.flutter.dev/flutter/foundation/ChangeNotifier-class.html>

Google . (2020, 11 26). *foundation library*. Retrieved from api.flutter.dev: <https://api.flutter.dev/flutter/foundation/foundation-library.html>

Google . (2020, 11 26). *InheritedWidget class*. Retrieved from api.flutter.dev: <https://api.flutter.dev/flutter/widgets/InheritedWidget-class.html>

Tyagi, P. (2020, 11 26). *Chapter 18: Provider & ChangeNotifier*. Retrieved from Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter18/provider\\_change](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter18/provider_change)

notifier.dart

Tyagi, P. (2020, 11 26). *mixins*. Retrieved from dart.dev:

<https://dart.dev/guides/language/language-tour#adding-features-to-a-class-mixins>

## BLoC Design Pattern

Dart Team . (2020, 11 26). *Sink<T> class*. Retrieved from api.dart.dev:

<https://api.dart.dev/stable/2.8.4/dart-core/Sink-class.html>

Dart Team . (2020, 11 26). *Stream<T> class*. Retrieved from api.dart.dev:

<https://api.dart.dev/stable/2.8.4/dart-async/Stream-class.html>

Dart Team . (2020, 11 26). *StreamController<T> class*. Retrieved from api.dart.dev:

<https://api.dart.dev/stable/2.8.4/dart-async/StreamController-class.html>

Flutter & Dart Team . (2020, 11 26). *Flutter & Dart Packages*. Retrieved from pub.dev:

<https://pub.dev/>

Flutter Team . (2020, 11 26). *flutter\_bloc*. Retrieved from pub.dev:

[https://pub.dev/packages/flutter\\_bloc](https://pub.dev/packages/flutter_bloc)

Tyagi, P. (2020, 11 26). *Chapter 19: Basic BLoC*. Retrieved from Pragmatic Flutter GitHub

Repo:

[https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter19/bloc\\_pattern1.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter19/bloc_pattern1.dart)

Tyagi, P. (2020, 11 26). *Chapter 19: Improvised BLoC*. Retrieved from Pragmatic Flutter GitHub

Repo:

[https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter19/bloc\\_pattern2.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter19/bloc_pattern2.dart)

Tyagi, P. (2020, 11 26). *Chapter 19: Using BLoC Library*. Retrieved from Pragmatic Flutter

GitHub Repo:

[https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/lib/chapter19/flutter\\_bloc.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/lib/chapter19/flutter_bloc.dart)

Tyagi, P. (2021). Chapter 18: Provider & ChangeNotifier. In P. Tyagi , Pragmatic Flutter:

Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

YouTube . (2020, 11 26). *Build reactive mobile apps with Flutter (Google I/O '18)*. Retrieved

from YouTube: <https://www.youtube.com/watch?v=RS36gBEp8OI>

## Unit Testing

dart.dev . (2020, 11 27). *test package*. Retrieved from pub.dev: <https://pub.dev/packages/test>

Flutter Community . (2020, 11 27). *font\_awesome\_flutter*. Retrieved from pub.dev:

[https://pub.dev/packages/font\\_awesome\\_flutter](https://pub.dev/packages/font_awesome_flutter)

Flutter Team . (2020, 11 27). *flutter\_test*. Retrieved from flutter.dev:

[https://flutter.dev/docs/cookbook/testing/widget/introduction#1-add-the-flutter\\_test-dependency](https://flutter.dev/docs/cookbook/testing/widget/introduction#1-add-the-flutter_test-dependency)

Google . (2020, 11 27). *TextDirection enum*. Retrieved from flutter.dev:

<https://api.flutter.dev/flutter/dart-ui/TextDirection-class.html>

Google . (2020, 11 30). *ElevatedButton class*. Retrieved from api.flutter.dev:

<https://api.flutter.dev/flutter/material/ElevatedButton-class.html>

Google . (2020, 11 30). *Unit Testing*. Retrieved from flutter.dev:

<https://flutter.dev/docs/cookbook/testing/unit>

Tyagi, P. (2020, 11 30). *Chapter20: Unit Testing*. Retrieved from Pragmatic Flutter GitHub

Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/test/main\\_20\\_test.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/test/main_20_test.dart)

Tyagi, P. (2021). Chapter 21: Widget Testing. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

## Widget Testing

Dart Dev . (2020, 11 30). *test*. Retrieved from pub.dev: <https://pub.dev/packages/test>

Flutter Dev . (2020, 11 30). *Finder class*. Retrieved from api.flutter.dev: [https://api.flutter.dev/flutter/flutter\\_test/Finder-class.html](https://api.flutter.dev/flutter/flutter_test/Finder-class.html)

Flutter Dev . (2020, 11 30). *Key class*. Retrieved from api.flutter.dev: <https://api.flutter.dev/flutter/foundation/Key-class.html>

Flutter Dev . (2020, 11 30). *Matcher class*. Retrieved from api.flutter.dev: [https://api.flutter.dev/flutter/package-matcher\\_matcher/Matcher-class.html](https://api.flutter.dev/flutter/package-matcher_matcher/Matcher-class.html)

Flutter Dev . (2020, 11 30). *MediaQuery class*. Retrieved from api.flutter.dev: <https://api.flutter.dev/flutter/widgets/MediaQuery-class.html>

Flutter Dev . (2020, 11 30). *pump method*. Retrieved from api.flutter.dev: [https://api.flutter.dev/flutter/flutter\\_test/TestWidgetsFlutterBinding/pump.html](https://api.flutter.dev/flutter/flutter_test/TestWidgetsFlutterBinding/pump.html)

Flutter Dev . (2020, 11 30). *pumpWidget method*. Retrieved from flutter.dev: [https://api.flutter.dev/flutter/flutter\\_test/WidgetTester/pumpWidget.html](https://api.flutter.dev/flutter/flutter_test/WidgetTester/pumpWidget.html)

Flutter Dev . (2020, 11 30). *tap method*. Retrieved from api.flutter.dev: [https://api.flutter.dev/flutter/flutter\\_test/WidgetController/tap.html](https://api.flutter.dev/flutter/flutter_test/WidgetController/tap.html)

Flutter Dev . (2020, 11 30). *testWidgets function*. Retrieved from api.flutter.dev: [https://api.flutter.dev/flutter/flutter\\_test/testWidgets.html](https://api.flutter.dev/flutter/flutter_test/testWidgets.html)

Flutter Dev . (2020, 11 30). *WidgetTester class*. Retrieved from api.flutter.dev: [https://api.flutter.dev/flutter/flutter\\_test/WidgetTester-class.html](https://api.flutter.dev/flutter/flutter_test/WidgetTester-class.html)

Flutter Team . (2020, 11 27). *flutter\_test*. Retrieved from flutter.dev: [https://flutter.dev/docs/cookbook/testing/widget/introduction#1-add-the-flutter\\_test-dependency](https://flutter.dev/docs/cookbook/testing/widget/introduction#1-add-the-flutter_test-dependency)

Google . (2020, 11 30). *Widget Testing*. Retrieved from flutter.dev: <https://flutter.dev/docs/cookbook/testing/widget/introduction>

Tyagi, P. (2020, 11 30). Chapter 21: *Widget Testing*. Retrieved from Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/blob/master/test/main\\_21\\_test.dart](https://github.com/ptyagicodecamp/pragmatic_flutter/blob/master/test/main_21_test.dart)

Tyagi, P. (2021). Chapter 21: Unit Testing. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

## Integration Testing

Flutter Dev . (2020, 11 30). *flutter\_driver library*. Retrieved from api.flutter.dev: [https://api.flutter.dev/flutter/flutter\\_driver/flutter\\_driver-library.html](https://api.flutter.dev/flutter/flutter_driver/flutter_driver-library.html)

Flutter Dev . (2020, 11 30). *Instrument the app*. Retrieved from Flutter Dev: <https://flutter.dev/docs/cookbook/testing/integration/introduction#4-instrument-the-app>

Tyagi, P. (2020, 11 30). *Chapter 22: Integration Testing*. Retrieved from Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/tree/master/test\\_driver](https://github.com/ptyagicodecamp/pragmatic_flutter/tree/master/test_driver)

Tyagi, P. (2020, 11 30). *Chapter 22: Integration Testing*. Retrieved from Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/tree/master/lib/chapter22](https://github.com/ptyagicodecamp/pragmatic_flutter/tree/master/lib/chapter22)

Tyagi, P. (2021). Chapter 20: Unit Testing. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

Tyagi, P. (2021). Chapter 21: Widget Testing. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

## Rolling into the World

Android Developer . (2020, 12 09). *Google Play*. Retrieved from developer.android.com: <https://developer.android.com/distribute/best-practices/launch>

Android Developer . (2020, 12 09). *Shrink, obfuscate, and optimize*. Retrieved from developer.android.com: <https://developer.android.com/studio/build/shrink-code>

Apple Inc . (2020, 12 07). *Apps*. Retrieved from App Store Connect: <https://appstoreconnect.apple.com/apps>

Apple Inc . (2020, 12 08). *Distributing software on macOS*. Retrieved from developer.apple.com: <https://developer.apple.com/macos/distribution/>

Apple Inc . (2020, 12 09). *Add a new app*. Retrieved from help.apple.com: <https://help.apple.com/app-store-connect/#/dev2cd126805>

Apple Inc . (2020, 12 09). *App Review*. Retrieved from developer.apple.com: <https://developer.apple.com/app-store/review/>

Apple Inc . (2020, 12 09). *App Store Connect*. Retrieved from My Apps: <https://appstoreconnect.apple.com/>

Apple Inc . (2020, 12 09). *Choosing a Membership*. Retrieved from developer.apple.com: <https://developer.apple.com/support/compare-memberships/>

Apple Inc . (2020, 12 09). *Develop*. Retrieved from developer.apple.com: <https://developer.apple.com/develop/>

Apple Inc . (2020, 12 09). *Register a new identifier*. Retrieved from developer.apple.com: <https://developer.apple.com/account/resources/identifiers/add/appid/bundle>

Apple Inc . (2020, 12 09). *Upload an app to App Store Connect*. Retrieved from help.apple.com: <https://help.apple.com/xcode/mac/current/#/dev442d7f2ca>

Dart Dev . (2020, 12 08). *dart2js: Dart-to-JavaScript compiler*. Retrieved from dart.dev: <https://dart.dev/tools/dart2js>

Firebase . (2020, 12 08). *Get started with Firebase Hosting*. Retrieved from firebase.google.com: <https://firebase.google.com/docs/hosting/quickstart>

Firebase Hosting . (2020, 12 08). Retrieved from console.firebase.google.com: <https://firebase.google.com/products/hosting>

Flutter Community . (2020, 12 07). *flutter\_launcher\_icons*. Retrieved from pub.dev: [https://pub.dev/packages/flutter\\_launcher\\_icons](https://pub.dev/packages/flutter_launcher_icons)

Flutter Dev . (2020, 12 08). *Distribution*. Retrieved from Desktop support for Flutter: <https://flutter.dev/desktop#distribution>

Flutter Dev . (2020, 12 08). *macOS-specific support*. Retrieved from Setting up entitlements: <https://flutter.dev/desktop#setting-up-entitlements>

Flutter Dev . (2020, 12 09). *Build and release an iOS app*. Retrieved from flutter.dev: <https://flutter.dev/docs/deployment/ios>

Flutter Dev . (2020, 12 09). *Create a keystore*. Retrieved from flutter.dev: <https://flutter.dev/docs/deployment/android#create-a-keystore>

Flutter Dev . (2020, 12 09). *Review Xcode project settings*. Retrieved from flutter.dev: <https://flutter.dev/docs/deployment/ios#review-xcode-project-settings>

Flutter's channels - dev . (2020, 12 08). Retrieved from Flutter build release channels: <https://github.com/flutter/flutter/wiki/Flutter-build-release-channels#dev>

Google . (2020, 12 07). *Adaptive icons*. Retrieved from developer.android.com: [https://developer.android.com/guide/practices/ui\\_guidelines/icon\\_design\\_adaptive](https://developer.android.com/guide/practices/ui_guidelines/icon_design_adaptive)

Google . (2020, 12 07). *Chromium OS*. Retrieved from The Chromium Projects: <https://www.chromium.org/chromium-os>

Google . (2020, 12 09). *Create a new developer account*. Retrieved from play.google.com: <https://play.google.com/console/signup>

Google . (2020, 12 09). *Google Play Console*. Retrieved from play.google.com: <https://play.google.com/apps/publish>

Google . (2020, 12 09). *Play Store*. Retrieved from play.google.com: [https://play.google.com/store?hl=en\\_US&gl=US](https://play.google.com/store?hl=en_US&gl=US)

Google Play Store . (2020, 12 07). *Google Play Store*. Retrieved from Google Play Store: <https://play.google.com/store>

Launcher icon generator . (2020, 12 09). Retrieved from romannurik.github.io: <https://romannurik.github.io/AndroidAssetStudio/icons->



launcher.html#foreground.type=clipart&foreground.clipart=android&foreground.space.trim=1&foreground.space.pad=0.25&foreColor=rgba(96%2C%20125%2C%20139%2C%200)&backColor=rgb(68%2C%20138%2C%20255)&crop=0

Tyagi, P. (2020, 12 08). *BooksApp Flutter Web App*. Retrieved from Firebase Hosting:  
[https://pragmatic-flutter-booksapp.web.app/#!/](https://pragmatic-flutter-booksapp.web.app/)

Tyagi, P. (2020, 12 08). *Chapter 23: Rolling into the World*. Retrieved from Pragmatic Flutter GitHub Repo: [https://github.com/ptyagicodecamp/pragmatic\\_flutter/tree/master/deploy](https://github.com/ptyagicodecamp/pragmatic_flutter/tree/master/deploy)

Tyagi, P. (2021). Chapter 15: The Second Page: BookDetails Widget. In P. Tyagi , Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop. CRC Press.

Wikipedia . (2020, 12 07). *Chrome OS*. Retrieved from Wikipedia:  
[https://en.wikipedia.org/wiki/Chrome\\_OS](https://en.wikipedia.org/wiki/Chrome_OS)

Wikipedia . (2020, 12 07). *Google Chrome*. Retrieved from Wikipedia:  
[https://en.wikipedia.org/wiki/Google\\_Chrome](https://en.wikipedia.org/wiki/Google_Chrome)