



Flutter App Development

How to Write for iOS and
Android at Once

—
Rap Payne

Apress®

Flutter App Development

How to Write for iOS
and Android at Once

Rap Payne

Apress®

Flutter App Development: How to Write for iOS and Android at Once

Rap Payne
Dallas, TX, USA

ISBN-13 (pbk): 979-8-8688-0484-7
<https://doi.org/10.1007/979-8-8688-0485-4>

ISBN-13 (electronic): 979-8-8688-0485-4

Copyright © 2024 by Rap Payne

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Miriam Haidara
Development Editor: James Markham
Coordinating Editor: Jessica Vakili

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at <https://www.apress.com/gp/services/source-code>.

If disposing of this product, please recycle the paper

This book is dedicated to the men and women of the Flutter Community. I've never seen a group more devoted to the success of others. You're an inspiration and example to me.

Particular thanks to these members of the community who've helped me with Flutter issues. This Texan owes y'all!

Andrew "Red" Brogdon (Columbus, Ohio), Brian Egan (Montana), Frederik Schweiger (Düsseldorf, Germany), Jeroen "Jay" Meijer (Rotterdam, Netherlands), Jochum van der Ploeg (Zwolle, Netherlands), Martin Rybak (New York), Martin Jeret (Estonia), Nash Ramdial (Trinidad), Nilay Yenner (San Francisco), Norbert Kozsir (Karlsruhe, Germany), Pooja Bhaumik (Bengaluru, India), Randal Schwartz (Portland, Oregon), Raouf Rahiche (Casablanca by way of Algeria), Remi Rousselet (Paris), Rohan Tanaja (Berlin), and Scott Stoll (Cleveland, Ohio).

But especially Simon Lightfoot (London), who we all call "The Flutter Whisperer." He taught me much of what I know about Flutter.

Worldwide Praise for *Flutter App Development: How to Write for iOS and Android at Once*

“Rap has written a great starting guide full of information for those who are new to developing multi-platform apps with Flutter.”

—Frederik Schweiger (Düsseldorf, Germany), organizer of the International Flutter Hackathon and creator of the Flutter School

“A great read! This covers everything a beginner might want to know, and more. It explains not only what Flutter is but why it exists works the way it does. It also provides great tips for common pitfalls along the way. Definitely recommended.”

—Jeroen “Jay” Meijer (Rotterdam, Netherlands), leader of Flutter Community GitHub

WORLDWIDE PRAISE FOR FLUTTER APP DEVELOPMENT: HOW TO WRITE FOR IOS AND ANDROID AT ONCE

“Rap’s book is a great book to get started with Flutter. It covers every important topic to write your very first app but also contains valuable information for more seasoned developers.”

—Norbert Kozsir (Karlsruhe, Germany), Flutter
Community editor

“As a non-native English speaker, I’m totally impressed by the simplicity of this book and how much I can read and understand without getting bored.”

—Raouf Rahiche (Algeria), Flutter speaker, developer, and
instructor

“As an early adopter and one of the original members of the Flutter Community, Rap is one of the world’s foremost authorities on Flutter. Where documentation is written for Engineers, by Engineers, Rap is a human who (thankfully!) writes in an enjoyable style that can easily be understood by other humans.”

—Scott Stoll (Cleveland, Ohio), contributor to the Flutter
codebase and cofounder of the Flutter Study Group

Table of Contents

About the Author	xvii
About the Technical Reviewer	xix
Who Is This Book For?	xxi
Foreword	xxvii
Preface	xxix
Chapter 1: Hello Flutter.....	1
What Is Flutter?.....	2
Why Flutter?.....	3
The Other Options	4
Native Solutions	5
Conclusion	6
Chapter 2: Developing in Flutter.....	7
The Flutter Toolchain.....	8
The Flutter SDK.....	8
IDEs	8
IDE DevTools	10
Emulators	11
Keeping the Tools Up to Date.....	13
The Flutter Development Process	16
Scaffolding the App and Files.....	17
Anatomy of a Flutter Project.....	17

TABLE OF CONTENTS

- Running Your App 19
- Running It As a Web App..... 20
- Running It on a Tethered Device 21
- Conclusion 23
- Chapter 3: Everything Is Widgets 25**
- UI As Code 28
- Built-In Flutter Widgets 29
 - Value Widgets 30
 - Layout Widgets 31
 - Navigation Widgets..... 31
 - Other Widgets 32
- How to Create Your Own Stateless Widgets..... 32
- Widgets Have Keys 36
- Passing a Value into Your Widget 37
- Stateless and Stateful Widgets 39
 - So Which One Should I Create? 39
- Conclusion 40
- Chapter 4: Value Widgets..... 41**
- The Text Widget..... 41
- The Icon Widget 42
- The Image Widget 44
 - Embedded Images..... 45
 - Network Images 46
 - Sizing an Image..... 46
- SnackBar Widget..... 49

Input Widgets	51
Text Fields.....	52
Checkboxes	56
Radio Buttons	57
Sliders	59
Dropdowns	60
Putting the Form Widgets Together	62
One Big Form Example	68
Conclusion	68
Chapter 5: Responding to Gestures	69
Meet the Button Family.....	70
ElevatedButton	72
TextButton and IconButton	72
FloatingActionButton (FAB).....	72
SegmentedButton.....	73
CupertinoButton.....	74
Dismissible.....	74
Custom Gestures for Your Custom Widgets.....	75
Step 1: Decide on Your Gestures and Behaviors.....	76
Step 2: Create Your Custom Widget	76
Step 3: Add a GestureDetector widget.....	77
Step 4: Associate Your Gesture with Its Behavior	78
Example 1: Reacting to a Long Press	78
Example 2: Pinching to Add a New Item	80
Example 3: Swiping Left or Right	81
Conclusion	83

TABLE OF CONTENTS

Chapter 6: Navigation and Routing85

- Stack Navigation 86
 - Navigating Forward and Back 87
 - Using Anonymous Routes 88
 - Get Result After a Scene Is Closed 89
- Drawer Navigation 90
 - The Drawer Widget 91
 - Filling the Drawer 93
- Tab Navigation 95
 - DefaultTabController 96
 - TabBarView 97
 - TabBar and Tabs 97
 - TabBar at the Bottom 98
- The Dialog Widget 99
 - showDialog() and AlertDialog 100
 - Responses with a Dialog 101
- Navigation Methods Can Be Combined 102

Chapter 7: Managing State 105

- What Is State? 105
- The Shape of a StatefulWidget 107
- The Most Important Rule About State! 108
- Passing State Down 109
- Lifting State Back Up 110
- An Example of State Management 112
- When Should We Use State? 116
- Conclusion 118

Chapter 8: State Management Libraries	119
The InheritedWidget.....	119
The BLoC Pattern	120
Some Libraries.....	121
ScopedModel.....	121
Redux and Hooks.....	122
Provider	122
Whoa! That’s a Lot of Packages!	122
Raw State.....	123
Riverpod.....	124
1. Install flutter_riverpod.....	125
2. Wrap Your App with a ProviderScope	125
3. Write a Provider	126
4. Inherit from ConsumerStateWidget.....	126
5. Read Data with ref.watch()	127
6. Write Data with ref.read().....	128
Conclusion	128
Chapter 9: Making RESTful API Calls with HTTP	129
What Is an <i>API</i> Call?	130
The Flavors of API Requests.....	131
Making an HTTP GET or DELETE Request	132
Making an HTTP PUT, POST, or PATCH Request.....	134
HTTP Responses to Widgets.....	135
Brute Force – The Easy Way	135
FutureBuilder – The Clean Way	136
StreamBuilder.....	138

TABLE OF CONTENTS

Strongly Typed Classes	140
Create a Business Class	140
Write a .fromJson() Method.....	141
Use .fromJson() to Hydrate the Object	141
One Big Example.....	142
Overview of the App	143
Create the Flutter App.....	145
Making a Strongly Typed Business Class	146
list_people.dart.....	147
A GET Request in Flutter.....	150
A DELETE Request in Flutter.....	150
upsert_person.dart.....	151
A POST and PUT Request in Flutter	152
Conclusion	153
Chapter 10: Styling with Themes.....	155
Thinking in Flutter Styles	155
Individual Styles.....	156
Styling Text	156
Mass-Changing Values.....	158
Themes	159
1. ColorScheme	160
2. ThemeData	164
3. Widget-Specific Themes.....	166
4. Put Them Together in a Theme	167
5. Override the Styles on Individual Widgets	168

Final Styling Thoughts.....	169
A Word About Colors	169
Custom Fonts.....	171
Conclusion	174
Chapter 11: Laying Out Your Widgets	177
Our Approach	177
1. Layout the Entire Screen (a.k.a. Scene)	178
2. Position Widgets Above and Below Each Other or Side by Side	178
3. Handle Situations When We Run Out of Space and Overflow the Scene	180
4. Handle Extra Space in the Scene	180
5. Make Finer Adjustments in Positioning	181
Laying Out the Whole Scene	181
MaterialApp Widget	182
The Scaffold Widget.....	183
The AppBar Widget	185
SafeArea Widget	186
Flutter’s Layout Algorithm.....	187
The Dreaded “Unbounded Height” Error.....	187
The Algorithm Itself.....	191
Conclusion	194
Chapter 12: Layout – Positioning Widgets	195
Putting Widgets Next to or Below Others	195
Responsive Design.....	197
Responsive Design in Flutter	197
Flex and MediaQuery.....	198
Conclusion	199

TABLE OF CONTENTS

Chapter 13: Layout – Fixing Overflows201

- Overflow Warning Bars 201
- Our Options to Correct Overflows..... 202
- A Sample Problem to Solve..... 203
- 1. Allow the Children to Wrap..... 205
- 2. Squeeze the Children Until They Fit 206
- 3. Allow the User to Scroll..... 208
 - SingleChildScrollView Widget..... 209
 - The ListView Widget 211
 - GridView Widget 214
- Conclusion 218

Chapter 14: Layout – Filling Extra Space.....219

- What if There’s Extra Space Left Over?..... 220
- mainAxisAlignment 220
 - crossAxisAlignment..... 222
 - IntrinsicWidth 223
- Expanded Widget 224
 - Open Space with Expandeds 227
- Conclusion 228

Chapter 15: Layout – Fine-Tuning Positioning229

- Container Widget and the Box Model..... 230
- EdgeInsets for Padding and Margin 231
- Alignment and Positioning Within a Container 232
- Container Sizes Are Not Obvious..... 234
- Container Decorations..... 236
 - Border 239

BorderRadius	240
BoxShape	242
Conclusion	244
Chapter 16: Layout – Special Presentation Widgets.....	245
Slivers	245
Stack Widget.....	246
Positioned Widget.....	248
Card Widget.....	250
The Table Widget.....	251
Conclusion	254
Appendix A: Dart Language Overview	255
Appendix B: Futures, Async, and Await.....	271
Appendix C: Including Packages in Your Flutter App.....	277
Appendix D: How to Work with Files.....	287
Appendix E: How to Debug Your Layout.....	297
Index.....	301

About the Author



Rap Payne started Agile Gadgets LLC, a mobile app development company, in 2003. Through it, Rap is a consultant, trainer, and entrepreneur who has written apps, mentored developers, and taught software development classes for government agencies like the NSA, FBI, US Air Force, Navy, Army, NASA, and Britain’s GCHQ and for Fortune 500 companies like Boeing, Walmart, Coca-Cola, Wells Fargo, CVS, Chase, HP, Lockheed Martin, Exxon-Mobil, Lowe’s, Nike, USAA, and Raytheon, to name a few.

Rap and Becky have been married for over 30 years. In addition to their five home-educated children, Rap is a spiritual father to many younger men whom he has mentored over the years. Rap is a middle-of-the-pack marathoner, triathlete, and power lifter.

As a professional mentor and trainer, Rap has mastered teaching highly complex ideas in easy-to-understand ways. And as a real-world developer, he understands the need to teach these topics using practical and realistic examples and exercises.

About the Technical Reviewer



Massimo Nardone has more than 22 years of experience in security, web/mobile development, cloud, and IT architecture. His true IT passions are security and Android.

He has been programming and teaching how to program with Android, Perl, PHP, Java, VB, Python, C/C++, and MySQL for more than 20 years.

He holds a Master of Science in Computing Science from the University of Salerno, Italy.

He has worked as a project manager, software engineer, research engineer, chief security architect, information security manager, PCI/SCADA auditor, and senior lead IT security/cloud/SCADA architect for many years.

His technical skills include security, Android, cloud, Java, MySQL, Drupal, Cobol, Perl, web/mobile development, MongoDB, D3, Joomla, Couchbase, C/C++, WebGL, Python, Pro Rails, django CMS, Jekyll, Scratch, and so on.

He works as Chief Information Security Officer (CISO) for Cargotec Oyj.

He worked as a visiting lecturer and supervisor for exercises at the Networking Laboratory of the Helsinki University of Technology (Aalto University). He holds four international patents (PKI, SIP, SAML, and proxy areas).

Who Is This Book For?

If you're a developer with experience in some object-oriented language like Java, C#, C++, or Objective-C and you want to create Android apps, iOS apps, or web apps with Flutter, this book is for you. It is especially important for you if you want to create an app that runs on multiple platforms and if you are new to Flutter.

If you've got some experience already with Flutter, you'll undoubtedly learn something, but we're not expecting that you have any prerequisite knowledge or experience with Flutter. All of our chapters are written with the assumption that everything in Flutter is completely new to you.

If you know anything about iOS development, Android development, or web development, that will certainly help with understanding the topics because there are lots of analogies in them for Flutter. The more you know about those things, the better, especially JavaScript and React. But if you know none of them, don't fret. They're by no means necessary.

Knowledge of the Dart language also will help. There are some unique but very cool Dart features that we consider best practices. We could have "simplified" the code by not using these best practices, but in the long run, that's not doing you any favors. Instead, we go ahead and use them, but we do explain those things in Appendix A, "Dart Language Overview." In there, we give you a cheat sheet with just enough detail to write code, followed by a more in-depth explanation of the features that will be unexpected by developers of other languages. Pay special attention to the section called "Unexpected Things About Dart."

What Is Covered?

This book teaches you how to create fully functioning and feature-rich apps that run on iOS, Android, desktops, and the Web.

1. Hello Flutter – Welcome to Flutter! We’re giving you a feel for why you’re here. What problems does Flutter solve? Why the boss would choose Flutter vs. some other solution.
2. Developing in Flutter – Flutter’s set of tools aren’t always straightforward. We explain what each tool does and how to use it. This chapter guides you through the process of write-debug-test-run. We get an understanding of the tooling including installation and maintenance.
3. Everything Is Widgets – Widgets are super important to Flutter since they’re the building blocks of every Flutter app. We show why and provide the motivation and how-to to create widgets. Topics include composition, UI as code, widget types, keys, and stateless vs. stateful widgets.
4. Value Widgets – A deep dive into widgets that hold a value, especially user-input fields. Topics include Text, Image, and Icon widgets and how to create forms in Flutter.
5. Responding to Gestures – How to make your program do things in response to user actions like taps, swiping, pinching, and the like. We’ll show you the button family and the GestureDetector widget.

6. Navigation and Routing – Navigation is making the app hide one widget and show another in response to user actions. This makes them feel like they’re moving from one scene to another. We’ll cover stack navigation, tab navigation, and drawer navigation.
7. Managing State – How to get data from one widget to another and how to change that data. We cover how to create StatefulWidgets and design them in the best way.
8. State Management Libraries – An overview of several libraries and a how-to for a super-simple library called Raw State and the most popular one – Riverpod.
9. Making RESTful API Calls with HTTP – How to read from and write to an HTTP API server. This is where we show how to make GET, POST, PUT, DELETE, and PATCH requests.
10. Styling with Themes – This is where we answer all the questions needed to get a real-world app looking good and staying consistent throughout with styles and themes.
11. Laying Out Your Widgets – The beginning of the final section, this chapter introduces the idea of layouts and steps to control layouts and exposes the Flutter layout algorithm.
12. Layout – Positioning Widgets – How to control how widgets are placed side by side and/or above and below.

WHO IS THIS BOOK FOR?

13. Layout – Fixing Overflows – What to do when you’re trying to draw *more than* will fit on a screen.
14. Layout – Filling Extra Space – What to do when you’re trying to draw *less than* will fit. What do you do with that extra space to make it look nice?
15. Layout – Fine-Tuning Positioning – How to adjust the last bits using borders, padding, and margins. How to make nonrectangular shapes.
16. Layout – Special Presentation Widgets – Widgets for when a simple layout won’t do the trick – slivers, stack, card, positioned, and table.

And we have five appendixes.

- A. Dart Language Overview – An easy-to-parse cheat sheet for Dart itself broken into expected features and pleasant surprises.
- B. Futures, Async, and Await – Handling asynchronous activities in Flutter.
- C. Including Packages in Your Flutter App – How to find and include the wealth of third-party, publicly available, and free packages. Also how to write and publish your own.
- D. How to Work with Files – Using libraries. Futures, async, and await. Bundling files with your app. Reading and writing a file. JSON serialization.
- E. How to Debug Your Layout – Interpreting what you see in the visual debugger in both VS Code and Android Studio.

What Is Not Covered and Where Can I Find It?

As importantly, you should know what not to expect in the book. We will not give you a primer on the Dart programming language beyond the aforementioned appendix. We simply didn't think it was the best use of your time and wanted to dive right into Flutter. If you feel you need a primer later on, go here: <https://dart.dev/guides/language/language-tour> followed by <https://dart.dev/tutorials>. We chose not to discuss deploying to the app stores. The stores already do a fine job of explaining how to submit an app. That, and the process, changes so frequently that your definitive resource ought to be the stores themselves. You'll find instructions at <https://developer.apple.com/ios/submit/> and here: <https://play.google.com/apps/publish>. And we aren't going to cover certain advanced topics like device-specific development in iOS and Android or adding Flutter to an existing iOS/Android project. These and so many other topics can be found on the Web by searching.

Foreword

The world of mobile development has shifted. Gone are the days of separate codebases for iOS and Android. Developers today crave efficiency, speed, and the power to reach a wider audience with a single codebase. This is where Flutter shines.

As a Google Developer Expert, I've witnessed firsthand the transformative potential of Flutter. Its elegant architecture, combined with a robust widget library, allows developers to create beautiful and high-performance apps with unmatched speed. The magic of hot reload makes experimentation a breeze, drastically reducing development time and accelerating the iterative process.

But don't let the power of Flutter fool you – it's not without its challenges. Learning a new language and navigating a vast ecosystem of tools and libraries can be daunting. This is where this book comes in.

This book provides a comprehensive guide to Flutter, taking you from the very basics of getting started to more advanced topics like state management and API integration. It is meticulously crafted with clear explanations, practical examples, and hands-on exercises that will solidify your understanding and empower you to build amazing apps.

Whether you're a seasoned developer looking to expand your skillset or a curious beginner eager to enter the world of mobile development, this book is your perfect companion. It's a testament to the dedication and expertise of my friend Rap Payne, and I am confident that it will equip you with the knowledge and confidence to become a successful Flutter developer.

FOREWORD

So, embrace the future of cross-platform development and dive into the pages of this invaluable resource. Get ready to build beautiful, performant apps with Flutter!

Randal L. Schwartz
May 2024

Preface

Welcome to *Flutter App Development: How to Write for iOS and Android at Once!* If you're familiar with my earlier work, *Beginning App Development with Flutter*, you might recognize some echoes here and there. That's because this book expands on and renews the topics covered in the first, incorporating the core concepts that made it valuable.

However, consider this a brand-new journey into the ever-evolving world of Flutter development. Since the publication of my first book, Flutter has undergone significant advancements, offering exciting new features and improved workflows. This book reflects those changes, providing a comprehensive and up-to-date exploration of the framework.

While some sections may contain familiar content, you'll find a wealth of new material within these pages. In the first book, we tried to cover layouts in one chapter. Big mistake. In this book, we expanded that to *six* chapters. In the first book, we talked about state management libraries but didn't explain how they worked. Thanks to critique by Remi Rousselet, we moved those chapters much sooner and wrote a how-to for his Riverpod library. And thanks to certain of my clients like Disney and the US State Department, I expanded on existing concepts, drilled deeper into specific functionalities, introduced entirely new sections that address the latest developments, and rearranged the flow to make it easier to learn Flutter.

Whether you're an intermediate Flutter developer building on the basics or a newcomer embarking on your first project, this book aims to be your trusted companion. We'll guide you through the intricacies of building beautiful and performant iOS and Android applications with Flutter, equipping you with the knowledge and skills to navigate the ever-evolving landscape.

PREFACE

So, even if you've traveled this path before, get ready to embark on a fresh adventure with *Flutter App Development: How to Write for iOS and Android at Once!*

Rap Payne
September 2024

CHAPTER 1

Hello Flutter

Picture this in your mind's eye. You are the CEO of a new business. Obviously, your mission is to maximize sales while minimizing expenses. "Hmmm," you think. "I can really increase sales if I make our products available on the Web." So you ask your friends how to create a web app and they say ...

"You need to hire a web developer. They should know HTML, CSS, JavaScript, and probably some framework like React, Vue, or Angular."

It's expensive but you do it and your gamble pays off. Sales increase markedly. Trying to keep on top of demand, you monitor social media and engage your customers. You hear them say that this web app is great and all but "We'd have been here earlier if you had an app in the App Store." So you talk to your team who, while being experts in the Web, are not iOS developers. They tell you ...

"You need to hire an iOS expert. They should know iOS, Swift or Objective-C, Xcode, macOS, and CocoaPods for development."

Your research shows that this person is *even more* specialized and therefore expensive than your web devs. But again, it seems to be the right thing to do, so you bite the bullet and hire them. But even while this app is being developed, you see that the feedback was not isolated to iOS apps, but instead was looking at all mobile devices. And – oh, snap! – 85% of devices worldwide run Android, not iOS. You bury your head in your hands as you ponder whether or not you can afford to ignore 85% of your potential customers. Your advisors tell you ...

“You need to hire an Android expert. They should know the Android OS, Gradle, Android SDK, XML, Android Studio, and Java or Kotlin.”

“Really?!? Another developer?”, you say. “Yes. And one just as expensive as your iOS developer,” they respond.

Isn't there one person who can do all three things? Some way to share the code between all those environments? Then you could hire just one person. In fact, they could write the code one time and deploy it to the Web, to the App Store, and to the Google Play Store. One codebase to maintain. One place to make improvements and upgrades. One place to squash bugs.

Ladies and gentlemen, allow me to introduce you to Flutter!

What Is Flutter?

Flutter is a set of tooling that allows us to create beautiful apps that run on iOS, Android, the Web, Windows, MacOS, and Linux desktops.

Flutter is ...

- Free (as in free beer; no cost)
- Open source (that's the other sense of the word “free”)
- Backed by and originated at Google
- Being enhanced and maintained by a team of developers at Google and hundreds of non-Google contributors around the globe
- Currently being used by thousands of developers in organizations across the world for production apps
- Fast because it compiles to truly native apps that don't use crutches like WebViews and JavaScript bridges

- Written one place and compiled to a web app for billions of browsers, an iOS app for iPhones and iPads, and an Android app for all of the rest of the phones and tablets out there

Why Flutter?

Google's mission with Flutter is ...

To build a better way to develop for mobile

Notice what is *not* in that mission. There's no mention of Android (which is also owned by Google) nor of iOS nor of the Web. Flutter's goal is to create a better way to develop for all devices. In other words, Flutter should be better to create iOS apps than Swift. It should be better to create Android apps than Kotlin. It should be better to create web apps than HTML/JavaScript. And if you get all of those things simultaneously with one codebase, all the better.

The Flutter team has succeeded spectacularly with this mission.

As proof, Eric Seidel offers this example.¹ The Google CRM team used Flutter to build an internal Android app and did it **three times** faster than with their traditional Android toolchain!

Note Flutter has delivered on their promise to create web apps and desktop apps. But we must note that relatively few people are choosing Flutter for those apps. The real success story of Flutter has been iOS and Android apps. Consider the other platforms icing on the cake, not the main course.

¹http://bit.ly/eric_seidel_flutter_keynote_video at the 21:47 mark

But it turns out that Flutter isn't the only game in town for cross-platform. You have other options.

The Other Options

Cross-platform development comes in three general flavors listed in Table 1-1.

Table 1-1. *Cross-platform development categories*

	Some technologies	Cons	Pros
Progressive web apps (PWA)	HTML/CSS, React, Vue, Angular	Not a real app. Runs in a web browser. Not available in app stores. Hard to create a desktop shortcut. Cannot access many of the device's resources like background processing, compass, proximity sensor, Bluetooth, NFC, and more	Easy to write
Hybrid	PhoneGap, Cordova, Sencha, Capacitor, Ionic	Runs in a WebView so it can be slow. Nearly impossible to share code with the web app	Easier for web devs to learn because it uses HTML and JavaScript as its language and structure
Compile-to-native solutions	React Native, NativeScript, Flutter, Maui	Learning a framework may be difficult. Mastering the toolchain definitely is	Real apps that can be found in the stores and run fast