

Extracted from:

# iOS 8 SDK Development

Creating iPhone and iPad Apps with Swift

This PDF file contains pages extracted from *iOS 8 SDK Development*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

Copyright © 2015 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina

The  
Pragmatic  
Programmers

# iOS 8 SDK Development

Creating iPhone  
and iPad Apps  
with Swift

Chris Adamson  
Janie Clayton

*edited by Rebecca Gulick*



# iOS 8 SDK Development

Creating iPhone and iPad Apps with Swift

Chris Adamson

Janie Clayton

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina



Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic courses, workshops, and other products can help you and your team create better software and have more fun. For more information, as well as the latest Pragmatic titles, please visit us at <https://pragprog.com>.

The team that produced this book includes:

Rebecca Gulick (editor)  
Potomac Indexing, LLC (indexer)  
Liz Welch (copyeditor)  
Dave Thomas (typesetter)  
Janet Furlow (producer)  
Ellie Callahan (support)

For international rights, please contact [rights@pragprog.com](mailto:rights@pragprog.com).

Copyright © 2015 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

Printed in the United States of America.

ISBN-13: 978-1-941222-64-5

Encoded using the finest acid-free high-entropy binary digits.

Book version: P1.0—March 2015

# Introduction

---

Sometimes it starts with an idea: “*I’m going to write an app to organize my music,*” “*I’m going to write a kick-ass game,*” “*I’m going to start a website, and I’d better have a native app too.*” Other times, it’s more of an inkling: “*I bet I could figure out iOS apps,*” “*I wonder how different that is,*” “*If I have to write another freaking login web page, I’m going to scream.*” And sometimes, like Everest, it’s worth conquering just because it’s there.

Whatever your reason for wanting to learn iOS development, it’s a good one. The platform is well established as the leader of the smartphone and tablet revolution, and its thoughtful and comprehensive tools and frameworks empower us to make great applications for iPhone, iPad, and iPod touch users.

It’s also an exciting time to get into iOS development, or even to come back if you dipped a toe in the water a while back but didn’t take the plunge. With Apple’s announcements at WWDC 2014, development for the platform just got a *lot* more interesting, thanks to a much greater level of interaction between apps, between iOS and other devices, and even between developers thanks to new testing and collaboration tools. To top it off, Apple introduced an entirely new programming language, Swift, which it intends as a safer, simpler, and faster alternative to Objective-C, which has served iOS and Mac OS X well throughout their entire lifetimes.

## About This Edition

This is our third time around with an introductory iOS book for Pragmatic Programmers. *iPhone SDK Development* came out in 2009 and covered iPhone OS 3, then was replaced by a 100% rewrite, *iOS SDK Development* in 2012, covering iOS 6. The book you’re reading started out as an update to that, but with all the changes in iOS 7 and 8 (including a change of programming language), it is overwhelmingly new material. If you do catch a bogus copy-and-paste from old material—a reference to an `.h` or `.m` file would be a dead giveaway—we hope you’ll let us know on the errata page.

In preparing this edition, we looked for a balance of topics appropriate to our intended audience: developers already comfortable with object-oriented programming, who are new to the iOS platform. We always look for a mix of the new and the fundamental, the shiny and the solid. We're also interested in spending time on things we've learned from our own experience, both good and bad. You'll see we spend a lot of time on “soft” topics like refactoring, debugging, project organization, and testing, and less time on just touring our favorite frameworks (sorry, AV Foundation). We think it's important to think of iOS development not just as a grab-bag of APIs, but as a process for turning time and enthusiasm into apps that work well, that can be maintained, and that delight and enlighten.

New hotness sometimes means clearing out the old and busted. In the previous edition, we bet pretty heavily on document-based apps and the ability to store them in iCloud. Two years later, we're not seeing much interest in document-driven apps on the App Store, and as for iCloud, as a certain Dark Lord of the Sith said, “you have failed me for the last time.” In their place, we hope you'll enjoy some of the neat new features of iOS 8, including the ability to expose your app's functionality to other apps with extensions, and a variety of tools for making apps look good and work well on both iPhone and iPad. On this latter point, our previous edition's sample projects were iPhone-only, but this time, we are all-in on Universal apps that support the iPhone and iPad form-factors, as well as the extra screen space afforded by the iPhone 6 models.

You may also notice a major change on the cover from the previous edition. The aforementioned books were written by Chris Adamson and Bill Dudney, the second coming after Bill did a two-year stint at Apple as a developer evangelist. In 2013, Bill returned to Apple to work on UIKit, so he's unable to contribute to this title, because of corporate policy and because he's really busy bringing us the very software this book is all about. So for this edition, please welcome new coauthor Janie Clayton. Janie is an iOS developer and self-described “human vertex shader” from Madison, Wisconsin, who's previously served as a technical reviewer for other Pragmatic Programmer iOS titles, and is a speaker at iOS conferences like CocoaConf and 360iDev. She also has uncanny nerd compatibility with Chris, the first person at CocoaConf Chicago to correctly identify the etymology of her youngest pug's name.<sup>1</sup> The first in-person editorial meeting for this book took place at Anime Central in Chicago, probably marking the first time a Prags book has kicked off in cos-

---

1. Janie's dog is named “Delia Derbyshire,” after the unjustly uncredited co-composer of the *Doctor Who* theme music.

play...although considering *Build iOS Games with Sprite Kit* authors Jonathan Penn and Josh Smith, we can't be 100% sure of that.

## So Here's the Plan

Our goal for this book is to get readers fully grounded in the essentials of iOS development, comfortable with the tools, the techniques, and the conventions they'll use to make iOS apps whether by themselves or as part of a team. We don't try to cover everything in the iOS SDK in part because doing so made for a 600-page book in 2009, and would be well into the thousands today. Instead, this book should serve as a prerequisite to all the other topic-specific iOS books published by Pragmatic Programmers, as well as those from other publishers.

In this edition, we have chosen to develop a single sample app through the course of the entire book. The advantages of this approach include less time spent starting projects and wiring up user interfaces, and a greater sense of how to manage an app as it evolves and adds features. This gives us a more reality-based approach to app development than we could get with a series of trivial projects, and more naturally leads to the payoff in the last chapter of submitting an app to the App Store and managing its ongoing development life cycle.

Speaking of payoff, here's the course we're going to take through the book:

- [Chapter 1, \*Playing Around with Xcode\*, on page ?](#),
- [Chapter 2, \*Building Adaptive User Interfaces\*, on page ?](#),
- [Chapter 3, \*Programming in Swift for iOS\*, on page ?](#),
- [Chapter 4, \*Testing Apps\*, on page ?](#),
- [Chapter 5, \*Presenting Data in Table Views\*, on page ?](#),
- [Chapter 6, \*Waiting for Things to Happen with Closures\*, on page ?](#),
- [Chapter 7, \*Doing Two Things at Once with Closures\*, on page ?](#),
- [Chapter 8, \*Growing Our Application\*, on page ?](#),
- [Chapter 9, \*Navigating Between View Controllers\*, on page ?](#),
- [Chapter 10, \*Taking Advantage of Large Screens\*, on page ?](#),
- [Chapter 11, \*Recognizing Gestures\*, on page ?](#),
- [Chapter 12, \*Working with Photos\*, on page ?](#),
- [Chapter 13, \*Launching, Backgrounding, and Extensions\*, on page ?](#),
- [Chapter 14, \*Debugging Apps\*, on page ?](#),
- [Chapter 15, \*Publishing to the App Store\*, on page ?](#).

## Expectations and Technical Requirements

The technical requirements for iOS development are pretty simple: the latest version of Xcode, and a Mac OS X computer that can run it. As of our publication in late 2014, that means Xcode 6.1.1, and a Mac running either OS X 10.9 (“Mavericks”) or 10.10 (“Yosemite”).

All code in this book uses the Swift programming language, which is new in Xcode 6, making this book one of the first to offer a thorough guide to using Swift with the iOS frameworks. Since there are no experts on Swift today other than the people who created it, we expect it will be as new to you as it was to us over the summer while we wrote the book (compounded by the fact that we had to reset when each beta of Xcode 6 changed the language). Fortunately, it’s a neat language that cleans out a lot of cruft from C and Objective-C, and we think you’ll be able to pick it up quickly, provided you’re a proficient programmer in at least one object-oriented language. That can be one of the many curly-brace descendants of C (C++, C#, or Java), or an OO scripting language like Ruby or Python.

## Online Resources

This book isn’t just about static words on a page or screen. It comes with a web page, <http://www.pragprog.com/titles/adios2>, where you can learn more and access useful resources:

- Download the complete source code for all the code examples in the book as ready-to-build Xcode projects.
- Participate in a discussion forum with other readers, fellow developers, and the authors.
- Help improve the book by reporting errata, such as content suggestions and typos.

If you’re reading the ebook, you can also access the source file for any code listing by clicking on the gray-green rectangle before the listing.

As we build our sample projects in this book, we will often write simple code, only to rewrite it with more ambitious code later as our knowledge increases.

All the different versions would be hard to put in one source file. So in the downloadable book code, we often have multiple copies of each project, each representing a different stage of its development. The different stages use numbered folders, like PragmaticTweets-1-1, PragmaticTweets-2-1, and so on, with the first number representing the chapter number and the second being a



revision within that chapter. These folder names also appear in the captions for each code example in the text. You can either code along for the entire book from scratch, or copy over one of these “stages” and pick up from there.

## And So It Begins

So, new platform, new programming language, new opportunities for new ideas and new apps. We can’t wait to see what you come up with. Let’s get started.

---

### Swift is a moving target

---

The Swift programming language is in a period of rapid development and improvement, and that puts early adopters like us on the proverbial “bleeding edge.” And we’ve done a little bleeding already: Changes to the language have broken the code in this book several times during its development, forcing us to issue quick updates.

We expect this is going to continue for a while: Swift language changes we’ve seen in the Xcode 6.3 betas will break the code we wrote for Xcode 6.1. It’s not major—when you open our older code, Xcode’s tooltips usually offer to make the needed change for you—but it’s still something to watch for.



For the time being, we will “lock” each release of the book and its sample code to one version of Xcode. For example, this release works with Xcode 6.1: The code won’t build in Xcode 6.0, nor do we expect it to build for 6.3. When the next version of Xcode comes out with code-breaking changes, we will update the book’s contents and sample code for that version of Swift. If you bought this as an ebook from [pragprog.com](http://www.pragprog.com), come back to the book’s home page at <http://www.pragprog.com/titles/adios2> to generate an update, or set up your [pragprog.com](http://www.pragprog.com) account to automatically send updates to your Dropbox. For store-bought paper copies, come to the home page for updated sample code and, if you like, a discount on adding the ebook version.

---