



Beginning Xcode

Swift 3 Edition

—

Matthew Knott

Apress®

Beginning Xcode

Swift 3 Edition



Matthew Knott

Apress®

Beginning Xcode

Matthew Knott
Ammanford, United Kingdom

ISBN-13 (pbk): 978-1-4302-5004-3
DOI 10.1007/978-1-4302-5005-0

ISBN-13 (electronic): 978-1-4302-5005-0

Library of Congress Control Number: 2016957886

Copyright © 2016 by Matthew Knott

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. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

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: Welmoed Spahr

Lead Editor: Aaron Black

Editorial Board: Steve Anglin, Pramila Balen, Louise Corrigan, James DeWolf, Jonathan Gennick,
Robert Hutchinson, Celestin Suresh John, Nikhil Karkal, Michelle Lowman, James Markham,
Susan McDermott, Matthew Moodie, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke,
Gwenan Spearing

Coordinating Editor: Jessica Vakili

Copy Editor: Kezia Endsley

Compositor: SPi Global

Indexer: SPi Global

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.springer.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 rights@apress.com, or visit www.apress.com.

Apress and friends of ED books 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 Special Bulk Sales–eBook Licensing web page at www.apress.com/bulk-sales.

Any source code or other supplementary materials referenced by the author in this text is available to readers at www.apress.com. For detailed information about how to locate your book's source code, go to www.apress.com/source-code/.

Printed on acid-free paper

Dedicated to all those who keep going no matter what obstacles life throws at them.

Contents at a Glance

About the Author	xv
Acknowledgments	xvii
Introduction	xix
■ Part I: Getting Acquainted	1
■ Chapter 1: Welcome to Xcode.....	3
■ Chapter 2: Diving Right In	13
■ Chapter 3: Project Templates and Getting Around.....	37
■ Chapter 4: Building Interfaces.....	65
■ Chapter 5: Getting Help and Code Completion.....	103
■ Chapter 6: Constraints.....	125
■ Part II: Diving Deeper.....	159
■ Chapter 7: Storyboards.....	161
■ Chapter 8: Table and Collection Views	201
■ Chapter 9: Frameworks, Libraries, and Targets.....	257
■ Chapter 10: Advanced Editing	281
■ Chapter 11: Debugging and Analysis	307

■ Part III: Final Preparations and Releasing	337
■ Chapter 12: Version Control with Git	339
■ Chapter 13: Localization	369
■ Chapter 14: Devices and the Organizer	405
■ Chapter 15: Building, Sharing, and Distributing Applications	423
Index	455

Contents

About the Author	xv
Acknowledgments	xvii
Introduction	xix
■ Part I: Getting Acquainted	1
■ Chapter 1: Welcome to Xcode.....	3
What Is Xcode?.....	3
What Is Swift?	3
Why Choose Xcode?.....	4
Getting and Installing Xcode.....	4
Firing Up Xcode	6
Apple’s Resources for Developers.....	7
The Dev Center	8
Your Developer Account	9
Source Code	10
Additional Resources.....	11
Summary.....	12
■ Chapter 2: Diving Right In	13
Creating Your First Xcode Project.....	14
The Project	19
Designing the Interface	20
Making Connections.....	24

Running and Testing Your Application	27
Adding Files.....	28
Organizing Files in Xcode.....	31
Summary.....	36
■ Chapter 3: Project Templates and Getting Around.....	37
Project Templates.....	37
Master-Detail Applications.....	38
Page-Based Applications.....	39
Single View Applications.....	40
Tabbed Applications.....	41
Games.....	42
New for iOS 10 and Xcode 8.....	42
Template Selection.....	43
Getting Around.....	45
Navigators	45
Toolbar.....	47
Editor	48
Utilities.....	50
Configuring the Second View Controller	59
Debugging Area	63
Summary.....	64
■ Chapter 4: Building Interfaces.....	65
Getting Ready	66
Adding Tab Bar Icons to an Asset Catalog	69
Before You Start	71
Building the Interface.....	72
Setting the Tab Icons	76
Tracking Location with the Track It Tab	78
Mixing Colors with the Slide It Tab	87

Adding “Off the Menu” Controls	92
Changing the Interface with Code	99
Summary	101
■ Chapter 5: Getting Help and Code Completion	103
Getting Help	103
Creating the Project	104
Downloading Additional Documentation	104
Quick Help	105
Documentation Viewer	107
Apple’s Web Site	113
Code Completion	114
Opening Web Sites in Safari	117
Sending an E-Mail with MFMailComposeViewController	117
Sending a Text Message	118
Building the Interface	119
Running the Application	122
Summary	123
■ Chapter 6: Constraints	125
Understanding Auto Layout	125
Building an Authentication View	126
Design Considerations	127
Debugging Views in Xcode	132
Manually Adding Constraints	133
Automatically Adding Constraints	139
Previewing Your Layout	143
Size Classes	144
Finishing Touches	151
Customizing Text Fields	151
Summary	158

- **Part II: Diving Deeper..... 159**
- **Chapter 7: Storyboards..... 161**
 - A Brief History of Storyboards..... 162
 - Creating a New Project Called SocialApp..... 164
 - Creating View Controllers 168
 - Pairing the View to the Controller 176
 - Building Up the Storyboard 178
 - Linking Scenes and Building Interfaces 181
 - What Are Segues? 181
 - Adding a Navigation Controller 186
 - Creating an Interface for the Feed View Controller..... 188
 - Creating an Interface for Tweet View Controller 193
 - Creating an Interface for the Compose View Controller..... 196
 - Summary 199
- **Chapter 8: Table and Collection Views 201**
 - What Is a Table View?..... 201
 - Table View Composition..... 202
 - Table View Styles..... 203
 - Configuring the Accounts View..... 203
 - The Key Attributes of Table Views..... 205
 - Manipulating Static Table Views 206
 - The Accounts and Social Framework 209
 - Retrieving and Displaying Twitter Accounts 210
 - Configuring the Feed View 219
 - Configuring the Tweet View..... 234
 - Configuring the Compose View 236
 - Discovering the Collection View 240
 - Embedding a Tab Bar Controller 240
 - Persisting User Preferences with UserDefaults..... 242

Adding a Collection View Controller	246
Configuring a Collection View	249
Displaying Items in a Collection View	251
Summary	256
■ Chapter 9: Frameworks, Libraries, and Targets.....	257
Understanding Frameworks	257
Creating the Project.....	258
Static Libraries, Frameworks, and Swift	273
Working with Multiple Targets.....	273
Summary.....	279
■ Chapter 10: Advanced Editing	281
Getting Started	281
Efficient Editing	284
Changing Color Schemes.....	284
Organizing and Navigating Code	289
Building the Scene.....	293
Folding Code.....	297
The Code Snippet Library.....	300
Summary.....	306
■ Chapter 11: Debugging and Analysis	307
Building the Application	308
Using Breakpoints to Resolve Logical Errors.....	313
Setting a Breakpoint.....	314
The Debug Navigator	315
The Debug Area	316
The Breakpoint Navigator	317
Runtime Errors	321
Using Exception Breakpoints.....	322

Compile-Time Errors	324
The Issue Navigator	325
Tools to Help with Debugging.....	326
Debugging a Location.....	328
Print Debugging with the Printer Simulator.....	329
Playground	332
Summary.....	335
■ Part III: Final Preparations and Releasing	337
■ Chapter 12: Version Control with Git	339
Why Use Version Control?	339
What Is Git?	339
Creating the Project.....	340
The AVFoundation Framework.....	343
Creating the Interface.....	344
Committing Changes	351
Examining Changes with the Version Editor	353
Branching in a Repository	356
Merging Branches	358
Removing a Branch	361
Using a Remote Repository	362
Registering for GitHub and Creating a Repository.....	362
Adding a GitHub Repository to Xcode.....	364
Pushing to a Remote Repository	366
Updating the Remote Repository.....	367
Summary.....	368
■ Chapter 13: Localization.....	369
Creating the SayMyName Application	370
Adding the Resources.....	372

Creating the Application Interface	373
Laying Out the Views	373
Building the Interface	376
Writing the Code for the SayMyName Application.....	380
Retrieving a Contact	380
Converting Text to Speech	384
Localizing the Application	387
Enabling Localization.....	387
Adding Another Language	389
Storyboards and Localization	390
Localizing Images.....	392
Localizing Code with Localizable.strings.....	395
Testing Localizations	398
Testing Localization with Xcode 8	399
Setting the Application Language in the Scheme.....	402
Summary.....	404
■ Chapter 14: Devices and the Organizer	405
The Role of the Organizer in Xcode 8	405
Preparing Xcode for Deploying to a Device	406
Enrolling in the Apple Developer Program	406
Adding Your Developer Account to Xcode.....	411
Preparing a Device for Deployment.....	414
Managing Devices in Xcode	416
Capturing a Screenshot from a Running Application.....	418
The Archives Organizer	418
Archiving Your Application	419
Summary.....	422

- Chapter 15: Building, Sharing, and Distributing Applications 423**
 - Final Checks Before Publishing Your Application 423
 - Building a Launch Screen 426
 - Discovering iTunes Connect 429
 - Creating an App ID 433
 - Creating an Application Profile 435
 - Uploading an Application to iTunes Connect 441
 - Creating a Distribution Certificate and Profile 442
 - Validating Your Application 443
 - Submitting Your Application to the App Store..... 444
 - Submitting Applications Using the Archives Organizer..... 444
 - Submitting Applications Using the Application Loader..... 445
 - Changing Build Numbers..... 445
 - Creating an .ipa File..... 446
 - Submitting an Application for Approval 449
 - Using the iTunes Connect App 451
 - Summary 453
 - Next Steps 454
- Index..... 455**

About the Author

Matthew Knott has been writing code for as long as he can remember, from marveling at moving pixels on a BBC Micro to writing ridiculous text adventures for his mother on an overheating ZX Spectrum 48k. Knott has been a professional software developer for the past 14 years and now leads a software section in Swansea, Wales. Matthew's work and hobby are basically the same things, but when he's not working, he loves spending time with his wife, Lisa, their two kids, Mikey and Charlotte, and Eva the dog.

Acknowledgments

Writing a book for the Swift programming language proved to be a blessing and a curse. The rapidly evolving ecosystem meant many a weekend producing code updates to stay relevant. For this book, I'd like to acknowledge all the great people who bought the last book and who sent kind messages and stories of how they're learning and developing their own apps.

Thanks to my family for the encouragement to put in another long shift at the keyboard and for keeping the tea flowing.

Introduction

Welcome to *Beginning Xcode*, the book that aims to give you all of the knowledge to start writing applications using what is probably the most powerful integrated development environment (IDE) ever, and it's free.

As with many Apple products, Xcode has simplicity and ease of use in abundance, but don't be fooled; the shiny exterior masks a workhorse of a tool, incredibly powerful with an extensive set of integrated tools for every eventuality. Xcode is the development environment that all other IDEs want to be when they grow up.

Xcode hasn't always been this shining Rock God of awesomeness; it used to be a sorry band of ragtag applications. When I first picked up Xcode 3 in 2007, I remember switching through an array of different applications to create an app, such as the very basic Interface Builder, and finding out how to adapt my knowledge of C into Objective-C. Back then what I really wanted was something that would show me how to get the most out of Xcode and would give me the understanding I needed to get going with the hundreds of app ideas I had in my head.

Fast-forward nine years. Xcode and I have both come a long way. I feel as if I've gone from a kid, bumping my leaky paddleboat aimlessly around a boating lake, to a handsome sea captain at the prow of my vessel, gazing forth as I slice through choppy waves with grace and ease. Well, aside from the handsome part, the analogy is a good one. Xcode 7 is more complete and powerful than ever before; an integrated product that puts the same power in your hands as the developers at Apple who write the apps found in iOS and macOS as well as WatchOS and tvOS to boot.

I've always had a mixed experience of development books in the past and was often sick of building the same old apps again and again. What I've aimed to do in this book is take you through almost every facet of Xcode, helping you to understand the capabilities of each of the key areas as you build a number of cool and exciting projects along the way, and begin to get to grips with the new Swift programming language. By the end of the book, you should be ready to turn the ideas in your head into reality, and I can't wait to see what that looks like.

Prior Assumptions

Before you dive in and start reading this book, it's assumed that you have at least some familiarity with developing for Cocoa Touch and are familiar with the concepts of object-oriented programming. This book is geared toward those developing for iOS; however, it's possible to get a lot out of this book if you're developing macOS applications, because many of the principles presented can be applied to either platform.

It's assumed that you are using a Mac and are preferably running the latest version of macOS. Unlike the Objective-C based equivalent of this title, it's absolutely necessary that you run the latest version of Xcode. There is a common misconception that you need the greatest and latest "souped-up" Mac, but many previous-generation iMacs, MacBooks, Mac Minis, and Mac Pros will work just fine.

It's also assumed that you know how to operate your Mac and how to use macOS. For example, you need to know how to use the Finder, save files, and so forth—all the basics. Finally, a couple of the chapters present scenarios in which an active Internet connection is required, and some features of Xcode perform better when you're connected. Additionally, some later chapters require a physical device and a paid developer account to complete, but for the most part you can use an iOS simulator to run your apps.

It's also worth mentioning that the purpose of this book is not to teach you how to create applications for iOS or teach you how to program in Swift or Objective-C; the purpose of this book is to get you up and running with Xcode so you can apply your current knowledge of Swift and macOS/iOS development and use the latest version of Xcode to its full potential to enable you to work more productively and create fantastic applications.

What's Covered in This Book

Part 1: Getting Acquainted

- *Chapter 1:* This chapter starts you on your journey into the world of Xcode and explains how to get Xcode onto your machine and prepare it for first use. You are shown how to sign up as an Apple developer, and you get a look at the wealth of resources provided by Apple to iOS and macOS developers.
- *Chapter 2:* Here, you start a project and get the ball rolling in terms of becoming familiar with Xcode. You learn the basics of how to create projects and build applications, along with how to get around in Xcode.
- *Chapter 3:* Next, the focus shifts to how to choose from Xcode's different project templates. You also get a guided tour of Xcode's interface along with an introduction to many of the menus, inspectors, and panels you should use to work efficiently.
- *Chapter 4:* This chapter focuses solely on how to design your interfaces using Xcode's built-in interface editor, Interface Builder. It gives you an in-depth look at the libraries and inspectors available.
- *Chapter 5:* Next, you're shown how to access the invaluable help resources that are built right in to Xcode and also how to make the most of its intelligent code-completion feature.
- *Chapter 6:* Building on Chapter 4, you see the Auto Layout system and learn how it works with constraints and size classes to create a single layout for any device.

Part 2: Diving Deeper

- *Chapter 7:* This chapter shows you how to use a key feature for rapid development in Xcode: Storyboards. You see how Storyboards can add a certain degree of logic to how you display and push views in your application.
- *Chapter 8:* This chapter explains how Xcode makes it easy to populate and create table and collection views, with the addition of how to customize their appearance and functionality.
- *Chapter 9:* Here you learn how to add features to your application by adding frameworks and libraries. You also learn how to create a different version of your application in the same project with targets.
- *Chapter 10:* This chapter shows you how to add your own personal touches to Xcode in terms of editing code. In particular, the code editor is the focus of this chapter, and you see how to work more productively and how to customize its appearance and behavior to suit your tastes and requirements.

- *Chapter 11:* This chapter presents the idea of making your application run more efficiently and faster. This is done by looking at the range of different tools and methods included in Xcode. For example, using breakpoints to step through your code systematically. You also learn about the Swift Playground for prototyping and testing your code.

Part 3: Final Preparations and Releasing

- *Chapter 12:* Here you learn how you can protect your code and work effectively as a team by using Git, Xcode's integrated version control software.
- *Chapter 13:* This chapter examines the idea of localization and how to use Xcode to accurately support multiple languages in your app.
- *Chapter 14:* This chapter looks at the Organizer, what it's for, how to navigate around in it, and how to keep your developer assets in good standing order.
- *Chapter 15:* To conclude, you make final touches to your application, build it for release, and then share it either as an IPA file or via the App Store using either Application Loader or the Organizer.

PART I



Getting Acquainted

CHAPTER 1



Welcome to Xcode

Apple provides Xcode to developers to help them create applications for Macs, iPhones, and iPads (macOS and iOS). Xcode was used to create many of your favorite iOS and macOS applications. Arguably, without such a powerful, refined, integrated development environment (IDE), the thriving ecosystem that is the App Store would not exist as it does today.

What Is Xcode?

All developers, regardless of the platform for which they're developing, use an array of tools to take an application from an idea to something that is readily available to millions of people. Apple is no exception, and it provides a cultured, powerful, and polished set of development tools. These tools come together in one application called Xcode. Xcode provides everything you need to create, test, deploy, and distribute applications for iOS and macOS. With the release of iOS 10 and Xcode 8, Apple has continued to build on the radical changes introduced in the previous version of both the operating system and in Xcode. One of the most significant developments in the last two years was that Apple, for the first time, introduced its own proprietary language called Swift.

The purpose of this book is to guide you in becoming familiar with Xcode 8, in the hope that you'll become more confident and embrace it as a means to create amazing, innovative new applications for iOS and macOS. As in many other technical books, as you progress through each chapter, you build on your knowledge and will create a variety of iOS applications that challenge the knowledge you gain through the course of the book.

Although Xcode was created primarily for developers working on iOS and macOS applications, it's also great if you work with other languages such as C, Java, and C++, among others. Xcode has a long, interesting history of releases, some having a very good reception and some less so. First released in 2003, Xcode has had six major releases and has seen a couple of major interface overhauls. After more than 10 years of active development, it's safe to say that Xcode is incredibly powerful and a leading professional set of development tools. What's more, Xcode is available to developers at absolutely no cost; all you need is an iTunes account and you're good to go.

What Is Swift?

Before WWDC 2014, Apple's World Wide Developer conference, if you created a project in Xcode, the code that was added behind the scenes was written in a programming language called Objective-C. At WWDC14, Apple unveiled a new and highly streamlined programming language called Swift. This new language greatly reduced the amount of code needed to achieve equivalent outcomes in Objective-C. Swift is far less rigid than Objective-C and is very similar in structure to Python, a move that will no doubt encourage more developers into what is already one of the most accessible development ecosystems in existence today.

Electronic supplementary material The online version of this chapter (doi:[10.1007/978-1-4302-5005-0_1](https://doi.org/10.1007/978-1-4302-5005-0_1)) contains supplementary material, which is available to authorized users.

To make Swift even more appealing, Apple also introduced a new piece of software called Swift Playground with Xcode 6. Playground gives you the ability to experiment with snippets of code to see the result without putting them into your application, meaning you have a code sandbox in which to try things such as loops or regular expressions and instantly see the result.

Swift has developed very quickly, yet the early days of Swift were difficult as Apple regularly changed how the language and the compiler worked. Code that was valid in one version of Xcode became invalid in the next—not a great situation for authors looking to write a print book about the language. Thankfully, now in version 3.0 at the time of this writing, Swift has continued to develop but also stabilized a great deal. In 2015, Apple made the Swift programming language and it is now available for Linux as well. Head over to <https://swift.org> to find out more about the language and to connect with the development community. Going open source has expanded and accelerated the development of Swift, with large organizations such as IBM becoming its biggest contributors.

The purpose of this book is to teach you how to create apps using Swift in Xcode 8—it isn't a definitive Swift language guide. Apple, however, has released a free book that *is* a definitive guide to the Swift programming language and has made it available via iBooks. Search iBooks for Swift programming or visit <https://itunes.apple.com/us/book/swift-programming-language/id881256329?mt=11>. Although you'll be using Swift version 3, I'll just refer to it as Swift for the course of the book.

Why Choose Xcode?

If you have experience developing for other platforms, then you probably want to know what makes Xcode so great. The main thing is that there's simply no other IDE like it. It's unique in the sense that Apple has created it to be simple, yet at the same time it masks a powerful interior. You have the ability to work with a range of technologies, and you also have a phenomenal developer toolkit at your disposal. Xcode contains everything you could need: an intuitive code editor, advanced debugging, seamless interface-editing features, and the benefit of being constantly updated and maintained by Apple.

In addition, using Xcode is arguably the only practical way to develop applications for iOS and macOS that can truly be called native. Xcode is what Apple itself uses to produce its own innovative software, which is used by millions of people.

Aside from Xcode, it's hard to find a commendable alternative if you want to develop native iOS or macOS applications. Of course, there are third-party services and tools, but when using these, you may often find yourself battling inconsistencies and a lack of compatibility rather than focusing on what's really important: creating great apps (and enjoying doing so). The purpose of Xcode isn't to simply be an IDE: it also helps and guides you on your quest to create something that has the potential to reach a staggeringly large audience. For that reason, Xcode is a fantastic choice.

Getting and Installing Xcode

Before you can download Xcode, there are a couple of things you need to do. You need an iTunes account (or an Apple ID) that allows you to download content from the Mac App Store. Then you're good to go. If you don't have an Apple ID, you can sign up for one at no cost at <http://appleid.apple.com>. This book is written for Xcode 8, and to run it you also need a Mac that's running the latest version of macOS or at least macOS 10.11.5.

Once you're equipped with an Apple ID and a Mac running macOS 10.11.5+, you can begin downloading Xcode. As with many other Mac apps, you simply download it from the Mac App Store at no additional cost. Open the App Store on your Mac, select Categories from the top bar of the window, and then click the Developer Tools category. Usually, you can find Xcode right away, either at the top of the window or in the sidebar on the right, displaying the top free apps. Alternatively, you can use the Search bar at top right and enter xcode. Xcode's icon is a hammer over an "A" blueprint, as shown in Figure 1-1.



Figure 1-1. Xcode in App Store search results

Note If you don't have access to the latest version of macOS or are running an older version that isn't supported, you can download previous versions of Xcode from the iOS Dev Center, but for this you need to have a registered Apple developer account. This is explained later. However, this book covers the latest version of Xcode (which is 8.0 at the time of this writing).

Select the icon and you're taken to Xcode's App Store page. Here you can view all the features of Xcode along with the latest additions to the current version of Xcode (at the time of this writing, this is 8.0) and preview some screenshots of Xcode. To download Xcode, click the gray Free button and enter your Apple ID e-mail address and password. Your download will commence. Xcode is about 4.4 GB, so you can go and make some coffee while you wait for the download to finish, as shown in Figure 1-2.



Figure 1-2. Xcode in the Mac App Store

With Xcode downloaded, open it from your Applications folder. You're prompted to install some additional packages. Click Install and enter your user password. This installation should take a matter of seconds, as shown in Figure 1-3.

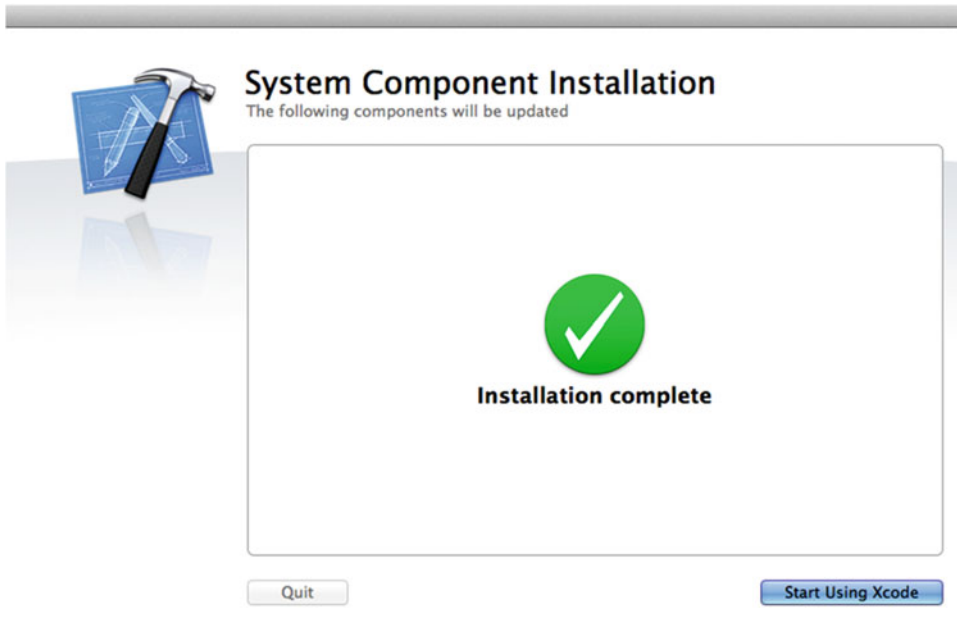


Figure 1-3. Installing additional tools required by Xcode

Firing Up Xcode

Once you’ve successfully downloaded and installed the additional components, you can begin using Xcode. In Chapter 2, you create your first project and become familiar with the basic areas of Xcode; but for now, just make sure everything is in good order so you don’t encounter any problems later.

When you first launch Xcode, you’re presented with a Welcome splash screen. From there, you can create a new project, connect to an external repository, open documentation, visit Apple’s developer portal, and browse a list of your recent projects. For some, this screen causes irritation—you can prevent it from appearing each time you open Xcode by simply checking or unchecking the Show This Window When Xcode Launches box, as shown in Figure 1-4.

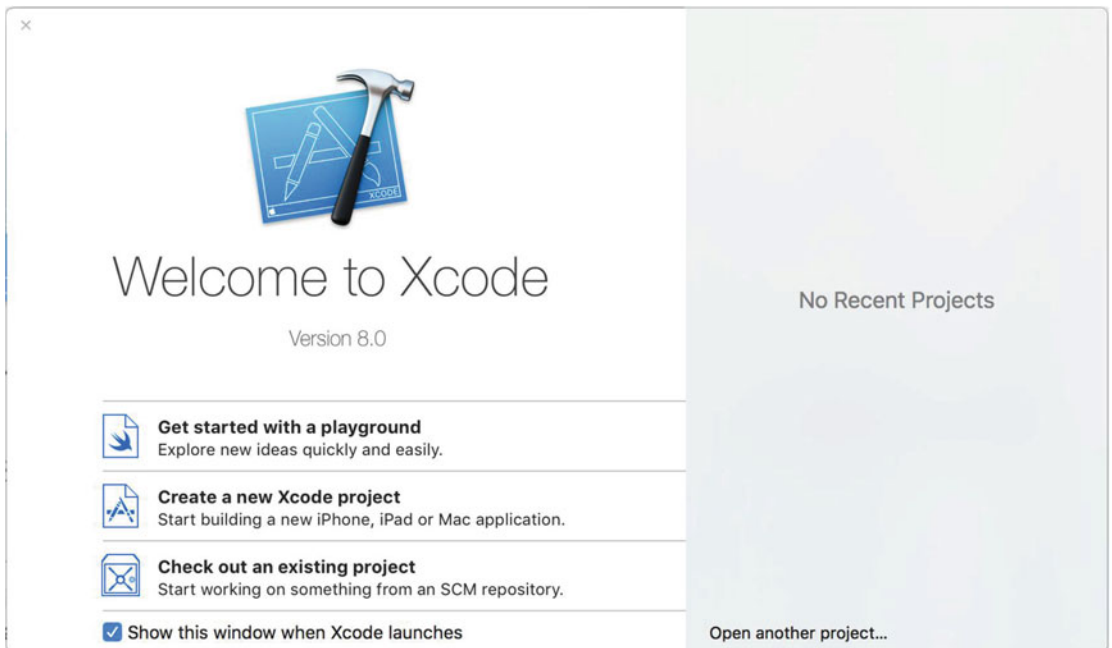


Figure 1-4. Xcode’s Welcome window, which is displayed optionally each time you open Xcode

To create a new project, click the Create A New Xcode Project button on the Welcome screen or navigate to File ► New ► Project, where you’re presented with a range of templates provided by default by Xcode.

If you have gotten to this point, it’s safe to assume that you’ve successfully installed Xcode and that you’re ready to start creating projects. However, let’s save this for a deeper explanation in Chapter 2 and for now look at the variety of resources provided to developers by Apple.

Apple’s Resources for Developers

At this point, you have Xcode downloaded to your machine, and you’ve fired it up to make sure it runs. If there’s one thing that makes Apple stand out from its competitors, it’s the wealth of knowledge, resources, and tools that are made just for developers. There are thousands of documents, thousands of samples to download, and dozens upon dozens of videos you can watch. Currently you have Xcode installed, but that alone isn’t going to make you a great developer of iOS and macOS applications. You also need to use the vast library provided by Apple. To gain access to Apple’s resources, I urge you to sign up as a registered Apple developer. To do this, all you need is an Apple ID; you can create a new one or use the same ID you use to download content from iTunes or the App Store.

First, head over to <http://developer.apple.com>. This is the central web site for Apple developers. Until recently, Apple had a dedicated area called the iOS Dev Center. This has now been replaced with a one-stop-shop solution in line with the new holistic approach Apple has taken toward its paid-for developer accounts. The key areas for you as a developer looking for support and guidance is the Resources page shown in Figure 1-5 and the Support page.

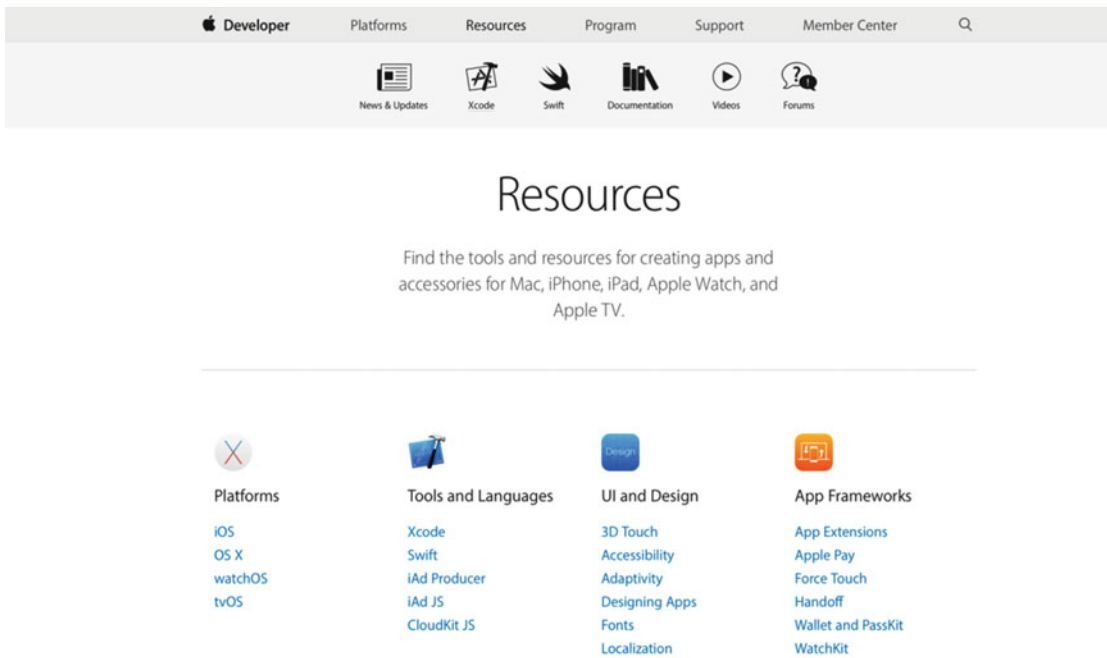


Figure 1-5. The Apple Developer Resources Page

You don't need to be subscribed to the Apple Developer Program in order to gain access to these resources, which includes the Developer Library, a wealth of example applications and their source code, release notes, and much more besides. To access the developer libraries for any of the core Apple technologies, go to Resources ► iOS/macOS/watchOS/tvOS ► Documentation.

The Dev Center

As mentioned previously, Apple really does like to take care of its developers. As a developer, your first port of call is the Developer Library, because it houses most of the resources provided by Apple. If you select the iOS Developer Library link under Documentation and Videos, you're taken to an invaluable section of Apple's developer web site. The Developer Library is a simple and straightforward site: simply use the links on the left to navigate around and to filter the results. You can search for specific keywords or sort the results using one of the column titles, as shown in Figure 1-6.

Learn About iOS 9.2

Explore new technologies and features in
What's New in iOS 9.2.

Title	Resource Type	Topic	Framework	Date
▶ iTunes Connect Developer Guide	Guides	Languages & Utilities		2016-03-01 Minor Change
▶ ThreadedCoreData: Concurrent operations using Core Data	Sample Code	User Experience Tables	CoreData	2016-02-25 Content Update
▶ iAd Advertising Guide for News Publishers	Guides	General		2016-02-19 Minor Change
▶ Migrating to CloudKit	Technical Notes	Data Management	CloudKit	2016-02-18 First Version
▶ How to reproduce bugs reported against App Store submissions	Technical Q&As	Xcode Compiler Tools		2016-02-18 Content Update
▶ MetalDeferredLighting	Sample Code	Graphics & Animation 3D Drawing	Metal	2016-02-18 Minor Change

Figure 1-6. The iOS Developer Library

In addition to the iOS Developer Library, you also have access to an array of getting-started videos that explain core Swift, Objective-C, and Cocoa Touch concepts. You can also access a direct link to the latest version of Xcode on the Mac App Store and can download previous versions of Xcode if you're not running the latest version of macOS or would like to target older versions of iOS.

Your Developer Account

Currently, your Apple ID gives you access to Xcode and all the resources you need to get started as an Xcode developer, but if you're planning to release applications to the App Store or test your applications on your own device, you're going to need to enroll in the Apple Developer Program. Although this isn't necessary at this point in the book, it's a good idea to sign up as a paid developer, because doing so gives you access to the Apple developer forums, prerelease versions of iOS before they're available to the public, prerelease versions of Xcode, the ability to test your applications on your iOS devices, and, of course, the ability to submit applications to the iOS, macOS, or tvOS App Stores.

The cost of signing up at the time of the writing of this book was \$99 per year, but it's required only when you want to publish an application to the App Store or need early access to Xcode or operating system betas. Apple has done a couple of neat things with the developer program in recent years. First, Apple combined all of its developer programs into one, meaning you only need to pay a single fee to develop for all of the available platforms and App Stores. Secondly, they removed the requirement to have a paid up developer account for testing on physical hardware, so you can take your apps on the go with you and show off your progress as you go through the book.

As mentioned, it isn't necessary to sign up this instant, but it's recommended that you do so at some point. To sign up for a paid account, visit <https://developer.apple.com/programs/> and click the Enroll button. You're then guided through the process of signing up; it's straightforward if you follow the steps onscreen, as shown in Figure 1-7.



Figure 1-7. *Choosing between an individual or a company developer account*

It's useful to note that when you're prompted to choose between an individual or company account, if you're planning to operate under a name other than your own, you have to register as an official company (this is verified by Apple) and then acquire what's called a Data Universal Numbering System (DUNS) number that uniquely identifies your company; this takes around seven days to process, so plan ahead. If selling applications under your own name suffices, then go for the simpler option of signing up as an individual. Both accounts are essentially equal in terms of the resources you're able to access. This choice mainly determines the name with which you operate under on the App Store.

Don't worry if you're not ready to do this right now—it's covered in detail later in the book when it becomes essential if you're planning to release applications on the App Store (free or paid). Chapter 14 takes you through enrollment and looks at using Provisioning Profiles and the deployment of an app onto an actual iOS device as opposed to the virtual iOS Simulator.

If you've had an Apple Developer account in the past, you may not be able to enroll. Instead, head over to <https://developer.apple.com/program/renew/index.action> and you will be able to renew your membership and regain access to the program.

Source Code

I strongly recommend that one of the first things you do is to go to the Apress web site for this book and download the entire source code. Either search for the book at www.apress.com or go directly to www.apress.com/9781430250043. When you get to the page for this book, scroll down until you see the section of the web site with four tabs, the third of which is Source Code/Downloads, as shown in Figure 1-8.

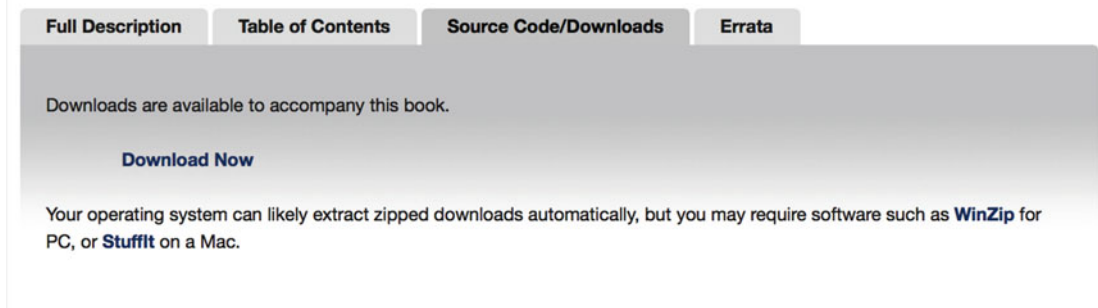


Figure 1-8. The source code download page for this book

Additional Resources

In addition to Apple’s own resources, an extensive amount of third-party resources are available if you ever have a burning question or get stuck somewhere:

Forums: Forums are a great way to ask questions, learn from other people’s questions, and help other people. In particular, Stack Overflow (<http://stackoverflow.com/>) has been invaluable to the entire developer spectrum for years and has a vibrant, active collection of iOS developers. There are also the Apple developer forums, which are available to those with a paid Apple developer account.

Mailing lists: There’s a handy Xcode users mailing list that I recommend you subscribe to and periodically check. Many other developers, including myself, participate in answering questions relating to Xcode. You can subscribe at <https://lists.apple.com/mailman/listinfo/xcode-users>.

Xcode overview: Apple provides a handy user’s guide that’s always being updated to accompany the latest release of Xcode, so it’s a good idea to refer to it when there’s a new update or if you want to follow up on something. It’s available at https://developer.apple.com/library/ios/documentation/ToolsLanguages/Conceptual/Xcode_Overview. Similarly, it’s handy to glance over the latest release notes when Xcode is updated. These are available at <https://developer.apple.com/library/ios/releasenotes/DeveloperTools/RN-Xcode>.

Search engines: It’s easy to underestimate the power of a simple Google search (and it’s apparent many people on online forums don’t have access to them). It can save you a lot of time, because someone, somewhere, at some point has undoubtedly had the same question you have—all you need to do is find out where they asked it!

Videos: If you type “Xcode” into iTunes U search, you’ll find a couple of good university courses that focus not only on Xcode but also on iOS development in general. Similarly, type “Xcode” into a YouTube search, and you’ll be amazed at what you can learn from the short screencasts that have been uploaded.

Contact me: I am happy to field questions via e-mail at matthewknott@me.com or via the Facebook page for this book at <https://www.facebook.com/BeginningXcode>.

Summary

In this chapter, you:

- Successfully downloaded and installed Xcode
- Looked around the iOS Dev Center and at the resources provided by Apple to aid developers
- Signed up and registered as an Apple developer and became aware of the option of signing up for a paid developer account

Chapter 2 explains how to create your first project and helps you become more familiar with Xcode's interface and basic concepts.