Apple made their iOS devices easy to use and now they have extended that to their development tools such as Xcode. In Xcode, you can create native applications in the easiest way. Apple's new Xcode technology is making the development curve smoother with its easy-to-develop features and enhancements.

Xcode can now write code with the performance-upgraded, brand new, innovative language called Swift, so you no longer need to rely on third-party frameworks to create applications.

The book gives you a tour of the new features of Xcode 6. It introduces some important aspects such as the Swift language and its Playgrounds with visual live coding, creating interfaces, storyboards, controllers, frameworks, and live previews. Diving more into the subject, this book shows you how to debug your code, and how to build and test the application on a device or the simulator.

Who this book is written for
This book is aimed at developers who want to make applications for Apple devices with Xcode. iOS developers who have experience using other frameworks and languages can now migrate to Xcode and build creative, native apps.
In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 'Introduction to Xcode'
- A synopsis of the book’s content
- More information on Xcode 6 Essentials

About the Author

Jayant Varma is a technophile with a career spanning over 2 decades and was introduced to computing in the days of 8-bit computers and Z80 chips. While managing the IT and telecom department at the BMW dealerships in India and Oman, and Nissan in Qatar, he worked extensively on Windows, AS/400, and Unix. His love for travelling inspired him to work and travel to several countries. He is currently based in Australia.

His technological journey began as a Microsoft Technologies developer and has diversified; his focus is now on Apple and mobile technologies. He has a masters degree in Business Administration and IT from James Cook University, Australia. He also lectured at James Cook University and co-ordinated the on-shore and off-shore teaching for the Linux/Unix Administration subject. He worked closely with the Australian Computer Society (ACS) and Apple University Consortium (AUC) on workshops and projects.

He authored the book Learn Lua for iOS Game Development, Apress, and has also been a technical reviewer on several titles.

As a founder, consultant, and developer at OZApps (www.oz-apps.com), he helps organizations and individuals integrate technology into their business and strategies. He also conducts trainings and workshops, and writes blogs to share his knowledge with the community.
Xcode 6 Essentials

Welcome to Xcode 6 Essentials. Development has moved on from the realm of computer science students to practically everyone. A couple of decades ago, computers were expensive and hard to use and now everyone has one in their pockets. Similarly, developmental tools have also changed and made life easier and allowed people from many other aspects to get with developing. With the popularity of iOS devices and Macs, the lure of being the App Store millionaire appeals to one and all. With a growing number of developers using Xcode, Apple has been adding a lot of features to it. In 2014, Apple released Xcode 6 and Swift. This book will cover using Xcode, and the language of choice is Swift. This book will take the reader through the various aspects of Xcode that might be important to use. The book employs a combination of a step-by-step approach and a theory approach to help the reader develop.

What This Book Covers

Chapter 1, Introduction to Xcode, introduces what Xcode is all about. It helps you install Xcode, if you do not have it on your system, and set it up for use.

Chapter 2, Tour of Xcode, offers a quick tour of the Xcode UI, the various panels, windows, and settings. This is what you will interact with on a regular basis when developing. It is helpful to know the shortcuts, what each displays, and how to activate or deactivate the windows.

Chapter 3, Playgrounds, takes you through a quick journey of Swift, allowing you to learn Swift interactively using the new feature called Playgrounds. This allows you to see the results of your commands and variables; your code is run in the background with no need to compile and run it.

Chapter 4, Interface Builder, introduces Interface Builder and Storyboards. It further helps non-developers or beginners to create applications with little to no code. This can also be used to create functional mock-ups as required.

Chapter 5, Custom Controls, introduces another new feature added to Xcode: the inclusion of custom controls and live previews. This chapter helps you understand the stub of a custom control and then add code. It further explains how to create this as a framework.

Chapter 6, Debugging, helps you as you progress and write your code, given the possibility of errors creeping in. Powerful debugging tools are introduced; they will help you understand where to look for basic help and will set up breakpoints, monitor variables, and quick look values.
Chapter 7, Building and Running, helps you understand the next step after creating your application: packaging it for distribution to the App Store. After going through this chapter, you will have run a full life cycle introduction to creating an application for iOS devices using Xcode.

Appendix, Conditional Execution and Interface Designing, has some interesting gotchas and tips for using Xcode more efficiently and also learning how the current versions differ from previous versions of Xcode and SDK.
Introduction to Xcode

A carpenter needs his tools to build furniture; similarly, as a developer, you need to have a set of tools that allow you to work in a better and more efficient manner. However, in the analogy of the carpenter, the tools could be used to make different types of furniture. With software, it is a little different. There are plenty of tools and they are dependent on many factors, such as the operating system (Windows, Mac OS X, or Linux) and the language of choice (for example, C, C++, Python, Ruby, JavaScript, and Objective-C). This book is about Xcode, the IDE that Apple makes available to developers who want to develop for Apple devices (iPhone, iPad, Mac, and so on). In this book, we will explore working with Xcode 6.x and all the features that it offers to write code, build UI, and debug, build, and distribute your apps.

Xcode helps you to build native applications that run on and use the features of the current iOS and Mac OS. The current one as of Xcode 6.x is iOS 8 for mobile devices and Mac OS X 10.10 Yosemite for desktops. Xcode 6.x has backward compatibility for iOS 7.0 and OS X 10.9. While there is backward compatibility, new features are available mostly with the newer version or later.

Requirements and installing Xcode

At first, Xcode was available with all Mac OS X installation disks, but then Apple made it available as a purchase from the App Store. Now it is available on the App Store for free. The minimum requirement for Xcode 6 is that you need to have Mac OS X 10.9.x installed with at least 10 GB of disk space. To run the simulator with newer, larger resolutions, it is advisable that you have a large screen display. The retina MacBook Pro or the newer iMac 5K retina would be a very good suggestion to help you work and design at 1:1 resolution.

The method to install Xcode via the App Store eliminates all the setting up and creation of the correct directories, which are generally associated with the installation. This also ensures that all of the upgrades are applied appropriately. If there is an issue, performing a simple reinstallation is as simple as downloading it again from the App Store.
Introduction to Xcode

If you like to work with beta software, you could also consider installing beta versions of Xcode from your Apple Developer Portal. Generally, since Apple does not allow you to create apps with beta versions, unless you are an advanced user, it is advisable to steer clear and wait for the release versions.

Here is a screenshot of the Xcode installation via App Store:
Upon first run, Xcode downloads some of the required components and utilities. These can be found under the **Developer** tools menu. Other components, utilities, and documentation can be downloaded via preferences. After these are installed, the API documentation and iOS simulators need to be downloaded:

You can build and test your iOS applications on the simulator from within Xcode and run OS X applications. To test your iOS app on a device or distribute the iOS or OS X app, you will have to enroll in the Apple Developer program for either iOS or Mac OS X, as appropriate, or both (this is a paid option).

The advantage of joining a developer program is that it provides access to pre-release software. It also offers access to the Apple forums, including two incident supports with Apple (per year).
Introduction to Xcode

Features of Xcode
Xcode is an all-inclusive IDE, and it offers a variety of tools required by a developer. IDE stands for Integrated Development Environment, which basically means that most of the tools required for a developer are available in this environment. The developer will not have to go outside of this environment to perform other tasks. Some basic features of an IDE include the ability to edit, write, and browse code and be able to run and debug the code from the program and return to the code editor when done. The Xcode environment is made up of the following components:

• **Editor**: Xcode offers a code editor with syntax highlighting for some languages, such as C, C++, Objective-C, JavaScript, Ruby, AppleScript, and Java. With third-party add-ins, support has been added for other languages as well, such as Pascal and C#.

• **File View**: Xcode also features file viewers for code, images, media, and data models.

• **Interface Builder**: Xcode features an Interface Builder that allows you to create interfaces for your apps, mobiles, or desktops. This provides a drag-and-drop interface to lay out the components of your app.

• **Debugger**: Xcode also features a debugger; earlier versions of Xcode used GDB, which has been replaced by the LLDB debugger since 4.3.

• **Versioning**: Xcode has Git integration that allows for versioning and integrated source control.

What's new in Xcode 6?
Xcode has been around for a while, and it was at WWDC 2014 that Xcode 6 was introduced. There were a couple of new features that were introduced in Xcode, and we will explore some of them in this book along with the standard features.

The first important thing that Apple introduced is a new Language called **Swift** (detailed in Chapter 3, Playgrounds). Since the days of NeXT, when Steve Jobs returned to Apple, Macs has been using the Objective-C language to write applications for the Mac environment. With the introduction of iPhone, Apple introduced their PhoneSDK, which allowed developers to write apps for their phones using Objective-C. This was later named iOS, as the devices were not just phones. Like any language, Objective-C has its own set of followers and haters. With the uptake of iOS devices, many users wanted to try their hand at developing apps. However, Objective-C largely remained a hurdle so they considered other options. At WWDC 2014, Apple unveiled a new language they had been working on, called Swift. Swift is an easy-to-learn-and-use language that looked more like a scripting language than a structured C variant. The code looked smaller and more manageable.
Chapter 1

The most important bit was that there was no more memory management in the form of alloc and retain statements. To add to this, the introduction to Swift by Apple mentioned, "Semi-colons are optional".

Interactive coding using Playgrounds
Apple also introduced a feature called Playgrounds. This allows you as the developer to quickly type in your commands in an editor, like you would write in a text editor or Microsoft Word. The Playground compiles, runs, and displays the results of each line of code that is typed. This topic is further detailed in Chapter 3, Playgrounds.

There is an open source project that runs your Objective-C code like Playgrounds. So you can simply alter your code in the editor and the running application is altered on the fly. This is not part of this book but could be interesting for some advanced users.

Mac OS X storyboards
Before storyboards, there were XIB files that described the user interface or the form in XML format. Apple introduced the storyboard, which allowed users to define the relationship between the elements, mainly to direct the workflow. This was originally introduced only for iOS projects, but with Xcode 6, storyboards are now available for OS X projects too. This is detailed in Chapter 4, Interface Builder.

Live design and responsive UI
In earlier versions of Xcode, there was little control over debugging and testing controls you created. Now, Apple offers a way to interact with your controls while in design mode in Interface Builder.

Xcode now offers new functionality to allow you to create a single storyboard that would adapt to different screen sizes with the help of size classes and autolayout. They are detailed in Chapter 4, Interface Builder.

Visual debugging
When debugging your code, Xcode can now display variables and objects visually. This helps make debugging much simpler and easy to use. It also offers functionality to provide custom previews for your own classes/objects while debugging. They are detailed in Chapter 5, Custom Controls.
Improved debugger

Even the debugger has some new features added to it. Some of these allow you to test and optimize your code and eliminate errors, including visual debugging with a hierarchical view of the views and elements on the screen. This is detailed in Chapter 6, Debugging.

In addition to these, we will also cover some of the other features that were present in earlier versions, which are equally important in getting things done.

Summary

This short chapter talks about how Xcode fits in with your development requirements. Every year, Apple adds new features to Xcode (and hopefully continues). This time around, out of the many features added, one of the major new additions that has everyone talking is a language called Swift and Playgrounds. In our next chapter, we will have a detailed look at Xcode, its components, and UI.
Where to buy this book

You can buy Xcode 6 Essentials from the Packt Publishing website.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.