# Contents

## Preface

- About this book
- What’s it like to build for watchOS?
- A note for iOS and macOS developers

## Introduction: Swift for Complete Beginners

- How to install Xcode and create a playground
- Variables and constants
- Types of Data
- Operators
- String Interpolation
- Arrays
- Dictionaries
- Conditional statements
- Loops
- Switch case
- Functions
- Optionals
- Optional chaining
- Enumerations
- Structs
- Classes
- Properties
- Static properties and methods
- Access control
- Polymorphism and typecasting
- Closures
- Protocols
- Extensions
- Protocol extensions
Wrap up

**Project 1: NoteDictate**  
- NoteDictate: Setting up  
- Defining our data  
- Working with views  
- Showing a list of notes  
- Reading user input  
- Selecting and deleting notes  
- NoteDictate: Wrap up

**Project 2: Rock, Paper, Scissors**  
- Rock, Paper, Scissors: Setting up  
- Proportional layout  
- Picking a move  
- Adding some pressure  
- Winning the game  
- Rock, Paper, Scissors: Wrap up

**Project 3: Storing data**  
- Storing Data: Setting up  
- Upgrading project 1  
- Reading and writing user settings  
- Storing Data: Wrap up

**Project 4: WatchFX**  
- WatchFX: Setting up  
- Picking from an array  
- Cleaning up our layout  
- Selecting favorite currencies  
- Making network requests  
- Supporting the Digital Crown  
- WatchFX: Wrap up
<table>
<thead>
<tr>
<th>Project 5: Hue Knows</th>
<th>254</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hue Knows: Setting up</td>
<td></td>
</tr>
<tr>
<td>Building the core game</td>
<td></td>
</tr>
<tr>
<td>Ending and restarting</td>
<td></td>
</tr>
<tr>
<td>Timing notifications for later</td>
<td></td>
</tr>
<tr>
<td>Hue Knows: Wrap up</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project 6: Views and Modifiers</th>
<th>275</th>
</tr>
</thead>
<tbody>
<tr>
<td>Views and modifiers: Introduction</td>
<td></td>
</tr>
<tr>
<td>Why does SwiftUI use structs for views?</td>
<td></td>
</tr>
<tr>
<td>Why modifier order matters</td>
<td></td>
</tr>
<tr>
<td>Why does SwiftUI use “some View” for its view type?</td>
<td></td>
</tr>
<tr>
<td>Conditional modifiers</td>
<td></td>
</tr>
<tr>
<td>Environment modifiers</td>
<td></td>
</tr>
<tr>
<td>Views as properties</td>
<td></td>
</tr>
<tr>
<td>View composition</td>
<td></td>
</tr>
<tr>
<td>Custom modifiers</td>
<td></td>
</tr>
<tr>
<td>Views and modifiers: Wrap up</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project 7: Magic 8-Ball</th>
<th>300</th>
</tr>
</thead>
<tbody>
<tr>
<td>Magic 8-Ball: Setting up</td>
<td></td>
</tr>
<tr>
<td>The basics of complications</td>
<td></td>
</tr>
<tr>
<td>Seeing into the future</td>
<td></td>
</tr>
<tr>
<td>Magic 8-Ball: Wrap up</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project 8: Safe Crack</th>
<th>316</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safe Crack: Setting up</td>
<td></td>
</tr>
<tr>
<td>Building the basic UI</td>
<td></td>
</tr>
<tr>
<td>Twist to win</td>
<td></td>
</tr>
<tr>
<td>Ending the game</td>
<td></td>
</tr>
<tr>
<td>Safe Crack: Wrap up</td>
<td></td>
</tr>
</tbody>
</table>
Project 9: Animation
  Animation: Setting up
  Creating implicit animations
  Customizing animations in SwiftUI
  Animating bindings
  Creating explicit animations
  Showing and hiding views with transitions
  Animation: Wrap up

Project 10: iMove
  iMove: Setting up
  Building the user interface
  Setting up permissions
  Reading live data from HealthKit
  Updating our user interface
  Handling state changes
  iMove: Wrap up

Project 11: ColorSpin
  ColorSpin: Setting up
  Showing a SpriteKit scene
  Controlling the player
  Creating warning edges
  Launching balls
  Spin to win!
  ColorSpin: Wrap up

Project 12: WatchConnectivity
  WatchConnectivity: Setting up
  A tale of two interfaces
  Establishing communication
  Sending messages
  Sending files
  WatchConnectivity: Wrap up