

핵심만 골라 배우는

SwiftUI 기반의 iOS 프로그래밍

SwiftUI Essentials: iOS Edition

by Neil Smyth

Copyright © 2019 Neil Smyth / Payload Media, Inc. All Rights Reserved.

Korean translation copyright © 2020 by J-Pub Co., Ltd.

This Korean edition was published by arrangement with Neil Smyth through Agency-One, Seoul.

이 책의 한국어판 저작권은 에이전시원을 통한 저작권사와의 독점 계약으로 제이펍에 있습니다.
저작권법에 의해 한국 내에서 보호를 받는 저작물이므로 무단 전재와 무단 복제를 금합니다.

핵심만 골라 배우는

SwiftUI 기반의 iOS 프로그래밍

1쇄 발행 2020년 3월 16일

지은이 닐 스미스

옮긴이 황반석

펴낸이 장성두

펴낸곳 주식회사 제이펍

출판신고 2009년 11월 10일 제406-2009-000087호

주소 경기도 파주시 회동길 159 3층 3-B호

전화 070-8201-9010 / 팩스 02-6280-0405

홈페이지 www.jpub.kr / [원고투고 jeipub@gmail.com](mailto:jeipub@gmail.com)

독자문의 readers.jpub@gmail.com / [교재문의 jeipubmarketer@gmail.com](mailto:jeipubmarketer@gmail.com)

편집팀 이종무, 이민숙, 최병찬, 이주원 / 소통·기획팀 민지환, 송찬수 / 회계팀 김유미

진행 및 교정·교열 장성두 / 내지디자인 최병찬 / 표지디자인 최병찬

용지 에스에이치페이퍼 / 인쇄 한승인쇄 / 제본 광우제책사

ISBN 979-11-90665-02-5 (93000)

값 26,000원

- ※ 이 책은 저작권법에 따라 보호를 받는 저작물이므로 무단 전재와 무단 복제를 금지하며,
이 책 내용의 전부 또는 일부를 이용하려면 반드시 저작권자와 제이펍의 서면 동의를 받아야 합니다.
- ※ 잘못된 책은 구입하신 서점에서 바꾸어 드립니다.

제이펍은 독자 여러분의 아이디어와 원고 투고를 기다리고 있습니다. 책으로 펴내고자 하는 아이디어나 원고가 있는 분께서는
책의 간단한 개요와 차례, 구성과 저(역)자 약력 등을 메일로 보내주세요. (보내실 곳: jeipub@gmail.com)

핵심만 골라 배우는

SwiftUI 기반의 iOS 프로그래밍

SwiftUI Essentials: iOS Edition

닐 스미스 지음 / 황반석 옮김



Jpub
제이퍼블

※ 드리는 말씀

- 이 책에 기재된 내용을 기반으로 한 운용 결과에 대해 저/역자, 소프트웨어 개발자 및 제공자, 제이펍 출판사는 일체의 책임을 지지 않음으로 양해 바랍니다.
- 이 책에 등장하는 각 회사명, 제품명은 일반적으로 각 회사의 등록 상표 또는 상표입니다. 본문 중에는 ™, ©, ® 마크 등이 표시되어 있지 않습니다.
- 이 책에서 사용하고 있는 제품 버전이나 URL 등은 독자의 학습 시점이나 환경에 따라 책의 내용과 다를 수 있습니다.
- 본문에서 중요 단어, 파일명, 버튼명, 탭명, 메뉴명, 프레임워크명, 폴더명, 템플릿명, 프로젝트명, 프레임워크명 등은 고딕체로 표기하였으며, 함수명, 변수명, 상수명, 뷰명, 메서드명, 컴포넌트명, 수정자명, 객체명, 인스턴스명, 프로토킴명, 뷰명, 컨테이너명, 클래스명, 프로퍼티명, 타입명, 모듈명, 매개변수명, 키워드명, 구문 이름, 절 이름, 블록 이름 등은 소스 코드 서체로 표기하였습니다.
- 책의 내용과 관련된 문의 사항은 옮긴이나 출판사로 연락해 주시기 바랍니다.
 - 옮긴이: <http://cafe.naver.com/petersbook> 또는 naya.peter@gmail.com
 - 출판사: readers.jpub@gmail.com

차례

옮긴이 머리말	xvi
베타리더 후기	xviii

CHAPTER 1

시작하기_1

1.1	스위프트 프로그래머에게	2
1.2	스위프트가 처음인 프로그래머에게	2
1.3	소스 코드 다운로드	3
1.4	피드백	3
1.5	오타자	3

CHAPTER 2

애플 개발자 프로그램 가입하기_4

2.1	Xcode 11과 iOS 13 SDK 다운로드하기	4
2.2	애플 개발자 프로그램	4
2.3	애플 개발자 프로그램에 등록할 시점	5
2.4	애플 개발자 프로그램에 등록하기	6
2.5	요약	7

CHAPTER 3

Xcode 11과 iOS 13 SDK 설치하기_8

3.1	macOS 버전 확인하기	8
3.2	Xcode 11과 iOS 13 SDK 설치하기	9
3.3	Xcode 시작하기	9
3.4	애플 아이디를 Xcode 설정에 추가하기	10
3.5	개발자 인증서와 배포 인증서	11

CHAPTER 4

Xcode 11 플레이그라운드_12

4.1	플레이그라운드란?	12
4.2	새로운 플레이그라운드 생성하기	13
4.3	스위프트 플레이그라운드 예제 기초	14

4.4	결과 보기	17
4.5	서식 있는 텍스트 주석 추가하기	18
4.6	여러 개의 플레이그라운드 페이지로 작업하기	20
4.7	플레이그라운드에서 UIKit으로 작업하기	21
4.8	플레이그라운드에 리소스 추가하기	22
4.9	향상된 라이브 뷰로 작업하기	24
4.10	요약	26

CHAPTER 5

스위프트 데이터 타입, 상수, 그리고 변수_27

5.1	스위프트 플레이그라운드 사용하기	28
5.2	스위프트 데이터 타입	28
5.3	스위프트 변수	33
5.4	스위프트 상수	34
5.5	상수와 변수 선언하기	34
5.6	타입 선언과 타입 추론	35
5.7	스위프트 튜플	36
5.8	스위프트 옵셔널 타입	37
5.9	타입 캐스팅과 타입 검사	41
5.10	요약	44

CHAPTER 6

스위프트 연산자와 표현식_45

6.1	스위프트 표현식 구문	45
6.2	기본 할당 연산자	46
6.3	스위프트 산술 연산자	46
6.4	복합 할당 연산자	47
6.5	비교 연산자	48
6.6	불리언 논리 연산자	49
6.7	범위 연산자	49
6.8	삼항 연산자	51
6.9	비트 연산자	51
6.10	복합 비트 연산자	55
6.11	요약	56

CHAPTER 7

스위프트의 제어 흐름_57

7.1	반복 제어 흐름	57
7.2	스위프트 for-in 구문	58
7.3	repeat ... while 반복문	60

7.4	반복문에서 빠져나오기	60
7.5	continue 구문	61
7.6	조건부 흐름 제어	62
7.7	if 구문 사용하기	62
7.8	if ... else ... 구문 사용하기	63
7.9	if ... else if ... 구문 사용하기	63
7.10	guard 구문	64
7.11	요약	65

CHAPTER 8

스위프트의 switch 구문_66

8.1	왜 switch 구문을 사용하나?	66
8.2	switch 구문 사용하기	66
8.3	스위프트의 switch 구문 예제	67
8.4	case 구문 결합하기	68
8.5	switch 구문에서 범위 매칭하기	69
8.6	where 구문 사용하기	69
8.7	fallthrough	70
8.8	요약	71

CHAPTER 9

스위프트 5의 함수, 메서드, 클로저 개요_72

9.1	함수란 무엇인가?	72
9.2	메서드란 무엇인가?	73
9.3	스위프트 함수를 선언하는 방법	73
9.4	단일 표현식에서의 암묵적 반환	74
9.5	스위프트 함수 호출하기	75
9.6	반환값 처리하기	75
9.7	지역 매개변수명과 외부 매개변수명	75
9.8	함수에 디폴트 매개변수 선언하기	77
9.9	여러 결괏값 반환하기	77
9.10	가변개수 매개변수	78
9.11	변수인 매개변수	79
9.12	입출력 매개변수로 작업하기	79
9.13	매개변수인 함수	81
9.14	클로저 표현식	83
9.15	스위프트의 클로저	85
9.16	요약	85

CHAPTER 10**스위프트의 객체지향 프로그래밍 기초_86**

10.1 객체란 무엇인가?	86
10.2 클래스란 무엇인가?	87
10.3 스위프트 클래스 선언하기	87
10.4 클래스에 인스턴스 프로퍼티 추가하기	88
10.5 메서드 정의하기	88
10.6 클래스 인스턴스 선언하기와 초기화하기	90
10.7 클래스 인스턴스 초기화하기와 소멸하기	90
10.8 메서드 호출하기와 프로퍼티 접근하기	91
10.9 저장 프로퍼티와 연산 프로퍼티	92
10.10 지연 저장 프로퍼티	94
10.11 스위프트에서 self 사용하기	95
10.12 스위프트 프로토콜 이해하기	96
10.13 불투명 반환 타입	98
10.14 요약	100

CHAPTER 11**스위프트의 서브클래싱과 익스텐션의 개요_101**

11.1 상속, 클래스, 그리고 하위 클래스	101
11.2 스위프트 상속 예제	102
11.3 하위 클래스의 기능 확장하기	103
11.4 상속받은 메서드 오버라이딩하기	104
11.5 하위 클래스 초기화하기	105
11.6 SavingsAccount 클래스 사용하기	106
11.7 스위프트 클래스 익스텐션	106
11.8 요약	107

CHAPTER 12**스위프트 구조체_108**

12.1 스위프트 구조체 개요	108
12.2 값 타입 vs. 참조 타입	109
12.3 구조체와 클래스는 언제 사용하는가?	112
12.4 요약	112

CHAPTER 13**스위프트 프로퍼티 래퍼_113**

13.1 프로퍼티 래퍼 이해하기	113
13.2 간단한 프로퍼티 래퍼 예제	114
13.3 여러 변수와 타입 지원하기	116
13.4 요약	119

CHAPTER 14

스위프트의 배열과 딕셔너리 컬렉션으로 작업하기_120

- 14.1 가변형 컬렉션과 불변형 컬렉션 120
- 14.2 스위프트 배열 초기화 120
- 14.3 스위프트 배열로 작업하기 122
- 14.4 배열 항목 섞기와 무작위로 가져오기 123
- 14.5 배열에 항목 추가하기 124
- 14.6 배열 반복하기 124
- 14.7 타입이 혼합된 배열 생성하기 125
- 14.8 스위프트 딕셔너리 컬렉션 126
- 14.9 스위프트 딕셔너리 초기화 127
- 14.10 시퀀스 기반의 딕셔너리 초기화 128
- 14.11 딕셔너리 항목 개수 128
- 14.12 딕셔너리 항목 접근하기와 갱신하기 129
- 14.13 딕셔너리 항목 추가하기와 제거하기 129
- 14.14 딕셔너리 반복 130
- 14.15 요약 130

CHAPTER 15

스위프트의 에러 핸들링 이해하기_131

- 15.1 에러 핸들링 이해하기 131
- 15.2 에러 타입 선언하기 132
- 15.3 에러 던지기 132
- 15.4 에러를 던지는 메서드와 함수 호출하기 134
- 15.5 에러 객체에 접근하기 135
- 15.6 에러 캐칭 비활성화하기 135
- 15.7 defer 구문 사용하기 135
- 15.8 요약 136

CHAPTER 16

SwiftUI 개요_137

- 16.1 UIKit과 인터페이스 빌더 138
- 16.2 SwiftUI의 선언적 구문 138
- 16.3 SwiftUI는 데이터 주도적이다 139
- 16.4 SwiftUI vs. UIKit 140
- 16.4 요약 142

CHAPTER 17**SwiftUI 모드로 Xcode 이용하기_143**

17.1 Xcode 11 시작하기	143
17.2 SwiftUI 프로젝트 생성하기	144
17.3 SwiftUI 모드의 Xcode	145
17.4 프리뷰 캔버스	147
17.5 프리뷰 고정하기	148
17.6 화면 설계 수정하기	149
17.7 에디터 콘텍스트 메뉴	152
17.8 여러 디바이스에서 미리보기	153
17.9 시뮬레이터에서 앱 실행하기	156
17.10 실제 iOS 디바이스에서 앱 실행하기	156
17.11 디바이스와 시뮬레이터 관리하기	159
17.12 네트워크 테스트 활성화하기	159
17.13 빌드 에러 처리하기	160
17.14 애플리케이션 성능 모니터링	160
17.15 사용자 인터페이스 레이아웃 계층구조 살펴보기	161
17.16 요약	164

CHAPTER 18**기본 SwiftUI 프로젝트 분석_165**

18.1 예제 프로젝트 생성하기	165
18.2 UIKit과 SwiftUI	166
18.3 AppDelegate.swift 파일	166
18.4 SceneDelegate.swift 파일	167
18.5 ContentView.swift 파일	168
18.6 Assets.xcassets	169
18.7 Info.plist	169
18.8 LaunchScreen.storyboard	169
18.9 요약	169

CHAPTER 19**SwiftUI로 커스텀 뷰 생성하기_170**

19.1 SwiftUI 뷰	170
19.2 기본 뷰 생성하기	171
19.3 뷰 추가하기	172
19.4 하위 뷰로 작업하기	174
19.5 프로퍼티로서의 뷰	175
19.6 뷰 변경하기	176
19.7 텍스트 스타일로 작업하기	176

19.8 수정자 순서	178
19.9 커스텀 수정자	179
19.10 기본적인 이벤트 처리	180
19.11 onAppear 메서드와 onDisappear 메서드	181
19.12 커스텀 컨테이너 뷰 만들기	181
19.13 요약	183

CHAPTER 20

SwiftUI 스택과 프레임_184

20.1 SwiftUI 스택	184
20.2 Spacer, alignment, 그리고 padding	187
20.3 컨테이너의 자식 뷰 제한	189
20.4 텍스트 줄 제한과 레이아웃 우선순위	190
20.5 SwiftUI 프레임	192
20.6 frame과 GeometryReader	194
20.7 요약	195

CHAPTER 21

상태, Observable 객체, 그리고 Environment 객체로 작업하기_196

21.1 상태 프로퍼티	197
21.2 상태 바인딩	199
21.3 Observable 객체	200
21.4 Environment 객체	202
21.5 요약	204

CHAPTER 22

SwiftUI 예제 튜토리얼_205

22.1 예제 프로젝트 생성하기	205
22.2 프로젝트 살펴보기	207
22.3 레이아웃에 VStack 추가하기	208
22.4 스택에 슬라이더 뷰 추가하기	209
22.5 상태 프로퍼티 추가하기	210
22.6 Text 뷰에 수정자 추가하기	211
22.7 회전과 애니메이션 추가하기	212
22.8 스택에 TextField 추가하기	214
22.9 색상 피커 추가하기	215
22.10 레이아웃 정리하기	216
22.11 요약	219

CHAPTER 23**Observable 객체와 Environment 객체 튜토리얼_220**

23.1 ObservableDemo 프로젝트에 대하여	220
23.2 프로젝트 생성하기	221
23.3 Observable 객체 추가하기	221
23.4 ContentView 레이아웃 설계하기	222
23.5 두 번째 뷰 추가하기	224
23.6 내비게이션 추가하기	225
23.7 Environment 객체 사용하기	226
23.8 요약	228

CHAPTER 24**스택 정렬과 정렬 가이드_229**

24.1 컨테이너 정렬	229
24.2 정렬 가이드	232
24.3 정렬 가이드 도구 사용하기	235
24.4 커스텀 정렬 타입	236
24.5 스택 정렬 교차하기	239
24.6 ZStack 커스텀 정렬	241
24.7 요약	245

CHAPTER 25**SwiftUI 리스트와 내비게이션_246**

25.1 SwiftUI 리스트	246
25.2 SwiftUI 동적 리스트	248
25.3 NavigationView와 NavigationLink	252
25.4 편집 가능하게 만들기	253
25.5 요약	256

CHAPTER 26**SwiftUI 리스트와 내비게이션 튜토리얼_257**

26.1 ListNavDemo 프로젝트에 대하여	257
26.2 ListNavDemo 프로젝트 생성하기	258
26.3 프로젝트 준비하기	258
26.4 Car 구조체 추가하기	259
26.5 JSON 데이터 로딩하기	260
26.6 데이터 저장소 추가하기	261
26.7 콘텐츠 뷰 설계하기	261
26.8 상세 뷰 설계하기	264

26.9	리스트에 내비게이션 추가하기	266
26.10	자동차 추가하는 뷰 설계하기	267
26.11	Add 버튼과 Edit 버튼 구현하기	270
26.12	Edit 버튼 메서드 추가하기	272
26.13	요약	273

CHAPTER 27

SwiftUI에서 탭 뷰 만들기_274

27.1	SwiftUI 탭 뷰 개요	274
27.2	TabViewDemo 앱 생성하기	275
27.3	TabView 컨테이너 추가하기	275
27.4	콘텐츠 뷰 추가하기	276
27.5	탭 아이템 추가하기	276
27.6	탭 아이템 태그 추가하기	277
27.7	요약	278

CHAPTER 28

SwiftUI에서 콘텍스트 메뉴 바인딩하기_279

28.1	ContextMenuDemo 프로젝트 생성하기	279
28.2	콘텐츠 뷰 준비하기	279
28.3	콘텍스트 메뉴 추가하기	280
28.4	콘텍스트 메뉴 테스트하기	281
28.5	요약	282

CHAPTER 29

SwiftUI 그래픽 드로잉 기초_283

29.1	DrawDemo 프로젝트 생성하기	283
29.2	SwiftUI 도형	283
29.3	오버레이 사용하기	286
29.4	커스텀 경로와 도형 그리기	286
29.5	그래디언트 그리기	289
29.6	요약	292

CHAPTER 30

SwiftUI 애니메이션과 전환_293

30.1	AnimationDemo 예제 프로젝트 생성하기	293
30.2	암묵적 애니메이션	293
30.3	애니메이션 반복하기	296
30.4	명시적 애니메이션	297
30.5	애니메이션과 상태 바인딩	298

30.6	자동으로 애니메이션 시작하기	299
30.7	SwiftUI 전환	302
30.8	전환 결합하기	303
30.9	비대칭 전환	304
30.10	요약	304

CHAPTER 31

SwiftUI에서 제스처 작업하기_305

31.1	GestureDemo 예제 프로젝트 생성하기	305
31.2	기본 제스처	305
31.3	onChanged 액션 콜백	308
31.4	updating 콜백 액션	309
31.5	제스처 구성하기	311
31.6	요약	313

CHAPTER 32

UIView와 SwiftUI 통합하기_314

32.1	SwiftUI와 UIKit의 통합	314
32.2	UIView를 SwiftUI와 통합하기	315
32.3	Coordinator 추가하기	317
32.4	UIKit 델리게이션과 데이터 소스 처리하기	319
32.5	예제 프로젝트	320
32.6	UIScrollView 래핑하기	320
32.7	코디네이터 구현하기	322
32.8	MyScrollView 사용하기	323
32.9	요약	324

CHAPTER 33

UIViewController를 SwiftUI와 통합하기_325

33.1	UIViewController와 SwiftUI	325
33.2	ViewControllerDemo 프로젝트 생성하기	326
33.3	UIImagePickerController 래핑하기	326
33.4	콘텐츠 뷰 설계하기	327
33.5	MyImagePicker 완성하기	329
33.6	콘텐츠 뷰 완성하기	331
33.7	앱 테스트하기	332
33.8	요약	333

CHAPTER 34

SwiftUI와 UIKit 통합하기_334

- 34.1 호스팅 컨트롤러의 개요 334
- 34.2 UIHostingController 예제 프로젝트 335
- 34.3 SwiftUI 콘텐츠 뷰 추가하기 335
- 34.4 스토리보드 준비하기 336
- 34.5 호스팅 컨트롤러 추가하기 339
- 34.6 Segue 액션 구성하기 340
- 34.7 컨테이너 뷰 포함하기 343
- 34.8 코드로 SwiftUI 포함하기 346
- 34.9 요약 348

CHAPTER 35

앱 스토어에 iOS 13 애플리케이션 등록을 위한 준비와 제출하기_349

- 35.1 iOS 배포 인증서 검증하기 349
- 35.2 앱 아이콘 추가하기 352
- 35.3 시작 화면 설계하기 353
- 35.4 프로젝트를 팀에 할당하기 354
- 35.5 배포를 위해 애플리케이션 아카이브하기 354
- 35.6 iTunes Connect에서 애플리케이션 설정하기 355
- 35.7 애플리케이션 검증하기와 제출하기 357
- 35.8 검수를 위해 앱 구성하고 제출하기 360

찾아보기 363

‘아이폰 애플리케이션’이라는 이름으로 대표되는 아이폰·아이패드·애플워치·애플티비·맥 애플리케이션 개발 방법은 끊임없이 개선되며 발전해오고 있습니다. 이제는 옛날얘기가 되어버렸지만, 초기에는 인터페이스 빌더라는 것이 따로 있었고 xib 파일로 각각의 화면을 표현하였습니다. 그러던 어느 날, 애플은 인터페이스 빌더를 Xcode 안에 포함시켰고 스토리보드로 모든 화면의 구축과 흐름을 만들게 해주었죠. 이와 함께 auto layout이 생기면서 다양한 화면 크기에 따라 적합한 화면을 만드는 방법을 제공했습니다. 그러다가 새로운 언어인 Swift가 발표되었고, 이제는 SwiftUI가 찾아왔네요. 애플은 Swift 언어를 발표하면서 애플리케이션 개발을 주도할 언어가 될 것이라고 말하면서도 Objective-C에 대한 지원을 계속하고 있습니다. 2014년에 출시된 Swift 언어가 벌써 6년이 되었지만, 아마 지금도 Objective-C로 애플리케이션을 구현하는 개발자가 있을 것이라 생각합니다. 이번에 애플은 SwiftUI를 발표했습니다. 지금부터 미래의 애플리케이션 개발을 이끌게 될 것임을 느낄 수 있어서 관심과 기대가 커집니다.

2019년 6월 3일, 애플이 WWDC 2019에서 발표한 SwiftUI는 스위프트의 성능을 바탕으로 사용자 인터페이스를 구축할 수 있는 혁신적이면서도 간소한 방법입니다. SwiftUI는 선언적 구문을 사용합니다. 역자와 같이 이전 방식에 익숙한 개발자들에게는 매우 당혹스러울 수 있습니다. 복잡하게 구성된 UI와 그 안에서 동작하는 로직을 담고 있는 코드가 (이전 방식에 비해) 너무 단순하기 때문입니다. 이전에 아이폰 애플리케이션 개발 경험이 있는 개발자라면 SwiftUI 덕분에 얼마나 단순해졌는지를 쉽게 알 수 있을 겁니다.

이 책은 SwiftUI에 관한 내용뿐만 아니라 애플리케이션 개발과 출시를 위한 모든 과정을 담고 있습니다. 먼저, 애플 개발자 프로그램(Apple Developer Program) 가입에 대한 내용과 Xcode 설치에 대한 내용으로 시작합니다. 아이폰 애플리케이션 개발을 해본 독자에게는 너무나 당연한 내용이겠지만, 이제 막 입문한(또는 입문하려는) 독자에게는 너무나 중요한 내용이

기 때문입니다. Xcode가 없다면 개발을 못 하고, 애플 개발자 프로그램에 가입하지 않는다면 여러분의 환상적인 앱을 앱스토어에 출시할 수 없겠지요. 스위프트 언어에 대한 내용도 포함 되어 있습니다. 이 책에서 설명하는 것이 스위프트 언어에 대한 전부는 아니지만 핵심적이자 기본적으로 알아야 할 내용임은 틀림없습니다. 사용자 인터페이스에 대해 배운다는 것은 내장된 컴포넌트(뷰)를 어떻게 화면에 배치하는지와 함께, 애플리케이션 내에서 계산되고 처리된 데이터를 화면에 표현하는 방법을 배운다는 것입니다. 스택, 리스트, 이미지, 텍스트 등의 기본 컴포넌트에 대해 배우다 보면 여러분이 만들려는 애플리케이션을 어떻게 구성해야 하는지를 알게 될 것입니다. 이와 함께, SwiftUI에서의 그래픽 드로잉과 애니메이션, 그리고 화면 전환에 대해서도 설명합니다. 이 책의 후반부에는 기존의 UIKit으로 구현한(또는 구현하고 있는) 프로젝트에 SwiftUI를 통합하는 방법을 설명합니다. 아마도 이 부분은 애플리케이션을 개발하고 있는 경험 있는 개발자들에게 매우 중요한 부분이 될 것입니다.

이 책의 중반부까지 번역하면서 개인적으로 머리에 맴도는 질문은 ‘그래서 어찌라고?’였습니다. SwiftUI가 좋다는 것은 점점 명확해지는데 기존에 개발하던 프로젝트를 생각하면 너무 난감했거든요. 말 그대로 ‘그래서 어찌라는 거지? 기존의 프로젝트를 다시 처음부터 SwiftUI로 구현해야 한다는 건가? UIKit으로 개발된 프로젝트는 새롭게 바닥부터 SwiftUI로 다시 해야 하나?’라는 생각뿐이었습니다. 하지만, 이 책의 후반부에 설명하는 통합 부분은 어떻게 합칠 수 있는지를 알려줍니다. 반대로, SwiftUI 기반의 프로젝트에 UIKit을 통합하는 방법도 설명하고 있습니다.

SwiftUI는 이제 막 시작되었습니다. 이제 계속 발전하고 개선되면서 아이폰·아이패드·애플워치·애플티비·맥 애플리케이션 개발 방법을 새롭게 주도할 것입니다. 이 새로운 흐름과 함께 하는 여러분에게 이 책이 훌륭한 가이드가 되길 바랍니다.

옮긴이 황반석

 **노수진**

SwiftUI를 한 번도 안 써본 사람이라도 예제를 보고 코드를 따라서 작성해 볼 수 있어서 좋았습니다. SwiftUI를 처음 사용하려 할 때 필요한 지식을 담고 있어서 좋은 시작점이 된 것 같습니다. 예제 일부 코드에서 인텐트와 줄바꿈 등에 일관성이 결여된 곳이 보이던데 출간 전에 해결하기를 바랍니다.

 **박명훈(카닥)**

SwiftUI에 대한 책이라 혹시라도 iOS 개발 기초가 없는 이들이 읽기 어려운 내용들만 있진 않을까 하였으나, 처음 iOS 개발을 시작하는 사람들도 읽기 쉽게 차근차근 설명되어 있어서 이 책 한 권만 읽어도 Swift의 최신 기술을 바로 습득할 수 있을 것이라 생각합니다.

 **박홍욱(연세대학교 하이퍼커넥트)**

이 책의 가장 큰 장점은 현재 최신 경향을 반영한 한글로 된 책이라는 점입니다. Swift 5와 SwiftUI가 도입되면서 나온 새로운 개념을 잘 녹여내고 있습니다. 물론, 기본서인 만큼 매우 자세한 설명이 부족하여 핵심 용어나 개념에 대해서는 추가적인 공부가 필요할 수 있습니다. 따라서 애플 공식 홈페이지에서 제공하는 기본적인 SwiftUI 튜토리얼을 마치고 해당 도서를 본다면 SwiftUI를 통한 애플리케이션 개발에 큰 도움이 될 것입니다.

 **이동건(카카오페이)**

스위프트를 처음 접하면서 최근에 나온 SwiftUI를 철저히 다뤄보고 싶은 독자라면 처음부터 차근차근 학습하기를 권합니다. 그리고 Swift 문법을 알고 있으면서 기존에 UIKit 기반의 프

로그래밍을 해본 분들이라면 앞부분의 문법 설명은 건너뛰고 보시기를 권합니다. 새롭게 나온 SwiftUI를 빠르게 배우실 수 있을 겁니다.

이석곤(엔컴)

iOS 13에서 가장 큰 변화는 SwiftUI라고 생각합니다. 애플의 모든 플랫폼에서 기본 지원을 하고 더 적은 코드로 더 우수한 앱을 만들 수 있습니다. 예전에 iOS를 오브젝티브-C로 개발 할 때는 코드 양이 많고 문법이 어려웠는데, 그에 비해 Swift는 읽기 쉽고 작성하기 편했습니다. 그리고 Xcode 디자인 도구와 코드가 연동이 되어 작성한 코드 결과를 실시간 미리보기를 할 수 있습니다. 이제는 여기에 더하여 SwiftUI로 강력한 UI 코드를 작성할 수 있습니다. 책은 전반적으로 입문자에 맞게 잘 되어 있어 어렵지 않게 볼 수 있는 수준이었습니다.

이원우(SK Planet)

책 전체에서 Swift 기본 문법이 차지하는 비중이 많아서 초반에 흥미를 떨어뜨렸습니다. 본격적인 SwiftUI를 설명하는 장에서는 겉가지 설명도 많았던 점도 아쉬웠습니다.

이현승(미국 씨티뱅크)

국내에 SwiftUI에 관한 책이 많지 않은 상황에서 SwiftUI에 대하여 요점을 쉽고 정확히 알려주는 책이라서 반가웠습니다. 특히, SwiftUI는 iOS 13부터 지원을 하기 때문에 대부분의 애플리케이션은 기존의 UIKit과 병행해서 사용해야 합니다. 기존 UIKit과 SwiftUI를 동시에 사용하는 방법을 구체적으로 설명하고 있어서 현업에서 당장 적용시키는 데에도 무리가 없었습니다.

전찬주(원티랩)

정말 재미있게 SwiftUI를 살펴볼 수 있었습니다. 아직 한글로 된 자료가 많지 않아서 영문으로 천천히 공부하고 있었는데, 베타리딩하면서 많은 도움이 되었습니다. 이미 iOS 개발 경험이 있다면 SwiftUI에 대한 핵심을 정말 빠르게 살펴볼 수 있는 책이었습니다. 예제 코드도 전반적으로 괜찮고 SwiftUI가 나오면서 바뀐 Swift의 변경사항에 대해서도 알 수 있어서 좋았습니다.

정욱재(스캐터랩)

불과 얼마 전에 나온 SwiftUI를 쉽고 명확하게 설명해주는 책입니다. Swift 문법까지 설명하고 있어서 다소 양이 많아 보일 순 있지만, 입문자들에게는 도움이 될 수도 있을 것 같습니다. 기존에 Swift 5를 알고 있는 사람이나 모르는 사람이나 쉽게 SwiftUI를 익힐 수 있는 책입니다.



주영호(연세대학교 일반대학원)

글자는 Text, 이미지는 Image. SwiftUI는 참으로 애플답다는 생각이 듭니다. 이 책에서는 SwiftUI의 기초부터 응용까지를 무려 한글로 볼 수 있습니다. 이 책 한 권으로 어떤 앱이든 만들 수 있게 되는 건 아니지만, 앞으로 꾸준히 업데이트될 SwiftUI의 다양한 기능과 매력을 느끼기에 충분합니다.



제이펍은 책에 대한 애정과 기술에 대한 열정이 뜨거운 베타리더들로 하여금
출간되는 모든 서적에 사전 검증을 시행하고 있습니다.

CHAPTER

1

시작하기

이 책의 목표는 SwiftUI와 Xcode 11 그리고 Swift 5 프로그래밍 언어로 iOS 13 애플리케이션을 만드는 데 필요한 핵심 기술을 알려주는 것이다.

이 책은 기초적인 내용부터 시작한다. iOS 개발 환경을 설정하는 데 필요한 과정부터 스위프트 언어를 배우고 테스트하기 위하여 스위프트 플레이그라운드(Swift Playground)를 어떻게 사용하는지에 대한 내용을 담고 있다.

또한, 프로그래밍 언어인 스위프트 5의 데이터 타입, 제어문, 함수, 객체지향 프로그래밍, 프로퍼티 래퍼(property wrapper), 그리고 에러 처리 등에 대한 내용을 다룰 것이다.

Xcode의 SwiftUI 개발 모드를 둘러보면서 SwiftUI와 프로젝트 구조에 대한 핵심 개념을 소개할 것이며, 커스텀 SwiftUI 뷰를 생성하는 방법과 스택, 프레임, 폼 등의 사용자 인터페이스 레이아웃을 생성하기 위한 SwiftUI 뷰의 연결 방법도 설명할 것이다.

Observable 객체와 Environment 객체 모두를 state 프로퍼티와 함께 사용하여 데이터를 처리하는 방법도 다루며, 주요 사용자 인터페이스 디자인 개념인 수정자(modifier), 리스트, 탭 뷰, 컨텍스트 메뉴(context menu), 사용자 인터페이스 내비게이션 등에 대한 주제도 다룬다.

그래픽 그리기와 사용자 인터페이스 애니메이션, 그리고 뷰 전환과 제스처 처리에 대한 내용도 있다. 또한, 기존의 UIKit 기반의 프로젝트에 SwiftUI 뷰를 포함시키는 방법을 설명하며, SwiftUI에 UIKit 코드를 넣는 방법도 설명한다.

마지막 장에서는 완성된 앱을 패키징하여 앱 스토어에 올리는 방법에 대해 설명한다.

이 책에서 다루는 주제들에 대해서는 상세한 튜토리얼을 통해 연습하게 될 것이며, 다운로드 받을 수 있는 소스 코드도 제공될 것이다.

다시 한번 말하지만, 이 책의 목표는 SwiftUI를 이용하여 여러분만의 앱을 만드는 데 필요한 기술을 가르치는 것이다. 필자는 여러분이 인텔 기반의 맥을 가지고 있으며 iOS 13 SDK와 Xcode 11을 다운로드받을 준비가 되어 있음을 가정하고 설명할 것이다.

1.1 스위프트 프로그래머에게

이 책은 스위프트를 이미 알고 있는 프로그래머뿐만 아니라 스위프트와 iOS 개발이 처음인 사람들 모두를 위해 쓰여진 책이다. 만약 여러분이 스위프트 5.1 프로그래밍 언어에 익숙하다면 스위프트에 대한 내용이 있는 장은 건너뛰어도 괜찮다. 하지만, 스위프트 5.1의 새로운 기능에 익숙하지 않다면 9.4 ‘단일 표현식에서의 암묵적 반환’, 10.13 ‘불투명 반환 타입’, 그리고 13장 ‘프로퍼티 래퍼’에 대한 내용 정도는 읽어보길 권장한다. SwiftUI를 구현하고 이해하는데 이들 기능이 핵심이기 때문이다.

1.2 스위프트가 처음인 프로그래머에게

만약 여러분이 스위프트로 프로그래밍하는 것이 (또는 프로그래밍 자체가) 처음이라면 이 책 전체의 내용을 처음부터 차근차근 읽고 따라오면 된다.

1.3 소스 코드 다운로드

이 책에 포함된 예제(소스 코드와 Xcode 프로젝트)는 다음의 URL에서 다운로드받을 수 있다.

URL <https://github.com/Jpub/SwiftUI>

1.4 피드백

이 책에 대해 만족하길 바라며, 이 책에 대한 오류나 질문 등의 내용은 역자 이메일 주소 (naya.peter@gmail.com)로 보내주기 바란다.

1.5 오타자

이 책 내용의 정확성을 위하여 모든 노력을 다하지만, 이 책에서 다루는 주제들의 범위와 난이도로 인해 약간의 오류나 오타자가 있을 수 있다. 발견된 오류나 오타자에 대해서는 제이펍 홈페이지(<https://www.jpub.kr>)의 이 책 소개 페이지에서 찾을 수 있다.

홈페이지에 게시되지 않은 오류나 오타자를 발견했다면 제이펍 출판사(readers.jpub@gmail.com)나 역자(naya.peter@gmail.com)에게 알려주기 바란다.

2

애플 개발자 프로그램 가입하기

iOS 13 기반의 애플리케이션 개발에 대해 공부하는 첫 단계는 애플 개발자 프로그램에 가입할 때 얻게 되는 혜택이 무엇인지 이해하는 것과 어느 타이밍에 가입하는 게 좋은지를 결정하는 일일 것이다. 이번 장에서는 개발자 프로그램의 장점과 비용에 대해 설명할 것이며, 가입 절차에 대해서도 알아볼 것이다.

2.1 Xcode 11과 iOS 13 SDK 다운로드하기

최신 버전의 iOS SDK와 Xcode는 맥 앱 스토어에서 무료로 다운로드할 수 있다. 여기서 생기는 질문은 ‘iOS SDK와 Xcode를 다운로드했는데도 애플 개발자 프로그램에 가입할 것인가? 아니면 앱 개발에 필요한 지식을 쌓을 때까지 기다릴 것인가?’이다.

2.2 애플 개발자 프로그램

애플 개발자 프로그램 멤버십은 현재 개인 개발자 등록 비용이 1년에 99달러이며, 기관(기업) 멤버십도 있다.

iOS 9 그리고 Xcode 7 이전에는 개발 중인 앱을 실제 iOS 기기에 설치해서 테스트하려면 개발자 프로그램에 가입해야만 했다. 하지만, 이제는 애플 아이디만 있으면 개발자 프로그램에 등록하지 않아도 실제 iOS 기기에서 앱을 설치하고 테스트할 수 있다.

애플 개발자 프로그램에 가입하지 않고도 할 수 있는 것이 많아지긴 했지만, 몇몇 기능은 애플 개발자 프로그램 멤버십 없이는 온전한 테스트를 할 수 없다. 특히, iCloud 접근, 애플 페이, 게임 센터, 앱 내 결제(In-App Purchasing)는 애플 개발자 프로그램 멤버십이 있어야만 기능을 사용할 수 있고 테스트도 할 수 있다.

애플 개발자 프로그램에 등록하면 애플의 iOS 엔지니어로부터 기술 지원을 받을 수 있다. 애플 개발자 프로그램에 등록(또는 갱신)하면 2회의 무료 지원이 제공되며, 그 이상의 지원을 받으려면 비용을 지불해야 한다. 또한, 다른 iOS 개발자들로부터 도움과 안내를 받거나 다른 개발자들이 발견한 문제와 해결책을 알 수 있는 애플 개발자 포럼의 멤버십을 받게 된다.

애플 개발자 프로그램 멤버십은 앞으로 출시될 Xcode와 iOS의 베타 버전을 미리 받아 볼 수 있는 혜택도 있다.

무엇보다 가장 중요한 애플 개발자 프로그램의 혜택은 애플리케이션을 앱 스토어에 등록하여 판매할 수 있게 해준다는 점이다.

분명한 것은 여러분의 애플리케이션이 앱 스토어에 등록되기 전에 애플 개발자 프로그램 멤버십이 필요하다는 것이다. 그렇다면 언제 가입하는 것이 좋을까?

2.3 애플 개발자 프로그램에 등록할 시점

애플 개발자 프로그램 멤버십이 많은 혜택을 주는 것은 확실하며, 애플리케이션을 앱 스토어에 등록하려면 결국에는 멤버십이 필요하다. 애플 개발자 프로그램에 지금 가입할지 아니면 나중에 가입할지는 경우에 따라 다르다. 만약 여러분이 iOS 애플리케이션 개발을 처음 배우는 단계이거나 개발할 애플리케이션에 대한 아이디어가 없다면 프로그램 멤버십이 지금 당장 필요하지는 않을 것이다. 여러분의 개발 기술이 향상되고 개발할 애플리케이션에 대한 아이디어가 견고해진 이후에 가입해도 늦지 않다.

하지만 반대로, 애플리케이션을 출시할 준비가 되었거나 iCloud, 앱 내 결제, 애플 페이 등과 같은 고급 기능이 필요하다면 차라리 일찌감치 개발자 프로그램에 등록하는 게 좋다.

2.4 애플 개발자 프로그램에 등록하기

만약 여러분이 회사를 위해 iOS 애플리케이션을 개발하려 한다면 회사가 멤버십을 가지고 있는지 먼저 확인하는 게 좋다. 회사에 멤버십을 관리하는 담당자를 찾아가서 개발팀에 합류할 수 있도록 Apple Developer Program Member Center에서 초대해 달라고 요청하도록 하자. 담당자가 초대를 하게 되면, 여러분의 멤버십이 활성화될 수 있는 링크가 담긴 'You Have Been Invited to Join an Apple Developer Program'이라는 제목의 이메일을 애플에서 받을 것이다. 만약 여러분이나 여러분의 회사에서 아직 프로그램 멤버가 아니라면 다음의 주소에서 등록할 수 있다.

URL <https://developer.apple.com/programs/enroll/>

애플은 회사 또는 개인의 등록 옵션을 제공한다. 개인으로 등록하려면 여러분의 신원을 증명하기 위하여 신용 카드 정보를 입력해야 한다. 회사로 등록하려면 법적 서명 권한을 가지고 (또는 그런 권한을 가진 담당자에게 찾아가 권한을 위임받고) D-U-N-S number¹와 법인 상태를 확인하는 문서² 등을 제출해야 한다.

개인 멤버로 개발자 프로그램에 등록하면 통상 24시간 이내에 승인되며, 애플로부터 활성화 양식이 이메일로 도착한다. 회사 멤버로 등록하면 추가로 확인을 요구하는 게 많아서 훨씬 더 오래 걸릴 수 있다.

활성화를 기다리는 동안 여러분의 애플 아이디와 비밀번호로 다음의 URL에 접근하면 몇몇 메뉴에 대해 제한된 상태로 멤버 센터(Member Center)에 로그인할 수 있을 것이다.

URL <https://developer.apple.com/membercenter>

로그인한 후에 페이지 상단에 있는 **Your Account** 탭을 클릭하면 개발자 프로그램 가입 상태가 **Enrollment Pending**으로 표시된다. 활성화 이메일을 받은 후에 멤버 센터에 다시 로그인하면 이제는 모든 메뉴와 리소스에 접근할 수 있다(그림 2-1 참고).

1 **음긴이** Data Universal Numbering System, D&B라는 업체가 개별업체에 부여하는 고유의 9자리 번호로, 전 세계 기업들의 인식번호로 인정받고 있다.

2 **음긴이** 예를 들면, 사업자등록증