Easy Mobile #1

도와주세요! 이제 막 앱 개발을 시작했어요!

Xcode4로 시작하는
아이폰 프로그래밍

Yoshinao Mori 지음
김태현 옮김

로드북
RoadBook

Xcode4로 시작하는 아이폰 프로그래밍

지은이 _ Yoshinao Mori
옮긴이 _ 김태현

1판 1쇄 발행일 _ 2012년 1월 13일

펴낸이 _ 장미경
펴낸곳 _ 로드북
편집 _ 임성춘
디자인 _ 이호용(표지), 박진희(본문)

주소 _ 서울시 관악구 신림동 1451-15 101호
출판 등록 _ 제 2011-21호(2011년 3월 22일)
전화 _ 02)874-7883
팩스 _ 02)6280-6901
정가 _ 25,000원
ISBN _ 978-89-966598-7-7 93560

Yoku Wakaru iPhone Apuri Kaihatsu no Kyokasho by Yoshinao Mori
Copyright © 2011 by Yoshinao Mori

All rights reserved.
Original Japanese edition published by Mynavi Corporation.
This Korean edition is published by arrangement with Mynavi Corporation,
Tokyo in care of Tuttle-Mori Agency, Inc., Tokyo through EntersKorea Co., Ltd., Seoul.

이 책의 한국어판 저작권은 (주)엔터스코리아를 통한 일본의 Mynavi Corporation과의
독점 계약으로 로드북이 소유합니다. 신 저작권법에 의하여 한국 내에서 보호를 받는
저작물이므로 무단전재와 무단복제를 금합니다.

책 내용에 대한 의견이나 문의는 출판사 이메일이나 블로그로 연락해 주십시오.
잘못 만들어진 책은 서점에서 교환해 드립니다.

이메일 chief@roadbook.co.kr
블로그 www.roadbook.co.kr
Q&A roadbook.zerois.net/qna

Xcode4로 시작하는
아이폰 프로그래밍

Yoshinao Mori 지음
김태현 옮김

지은이 글

이 책은「아이폰 앱을 만들어보고 싶다」고 생각하는 분들을 위한 입문서입니다.
아이폰 앱 개발은 소스 코드 작성법이나 화면(사용자 인터페이스)을 만드는 방법이 조금은 특이합니다. 그래서「아주 간단한 앱을 만들어보고 싶을 뿐」이지만 시작하는 방법을 몰라서 시작도 못해보고 좌절해버리는 경우가 많습니다. 필자도 처음엔 고민을 많이 했습니다.
아이폰에서 앱을 사용할 때는 별 어려움 없이 간단하고 쉽게 했으니, 앱을 개발할 때도 쉽고 간단한 방법은 없을까 생각하곤 했는데, 알고 보니 그러한 방법이 있었습니다. 아이폰 앱을 개발하는 데 필요한 소프트웨어인 Xcode에는「인터페이스 빌더」라는 알기 쉬운 도구가 준비되어 있습니다. 이 도구를 사용하면 아이폰 앱을「보이는 그대로 디자인해서」만들 수 있습니다.
Xcode 3까지는 인터페이스 빌더가 별도의 독립된 소프트웨어였지만, Xcode 4로 넘어오면서 Xcode 본체와 통합되었습니다. 앱 화면을 디자인할 때 자연스럽게 표시되므로 더 사용하기 쉽습니다.
이 책에서는 이 Xcode 4용 인터페이스 빌더의 사용법을 중심으로, 아이폰 앱 개발 방법을 설명합니다.
아이폰 앱 개발에 익숙한 분들 중에는 인터페이스 빌더를 사용하지 않는 분도 있습니다. 프로그래밍에 정통해 있어서 도구를 사용해 만드는 것보다 세밀하고 빠르게 설정할 수 있기 때문이라고 생각해봅니다.
그러나 프로그래밍에 정통하지 않아도 인터페이스 빌더를 사용하면 직관적인 디자인이 가능하며, 앱의 형태와 내부 구조를 분리해서 생각할 수도 있게 됩니다. 여기에 무엇보다도 앱 개발이 간단해집니다.
그러니 일단은 자신의 손으로 앱을 만들 수 있다는 것을 체험해주시기 바랍니다.
이 책은「아이폰 앱 개발을 위한 출발점」에 지나지 않습니다. 공부하고 연구해나가면 그 앞에 무한히 펼쳐진 앱의 세계를 자신의 것으로 만들 수 있을 것입니다. 꼭 자신의 손으로 만든 기발하고 매력적인 앱을 전 세계를 향해 릴리즈해주시기 바랍니다.

2011년 7월
모리 요시나오

옮긴이의 글

아이폰 앱 개발 관련 서적은 서가에 범람하고 있지만, 실제로 도움이 되는 책은 그리 많지 않습니다. 이 책은 그 중 몇 안 되는 정말 도움이 되는 책 중 하나라고 자신합니다. 역자 또한 아이폰 앱 개발 능력을 손에 넣고 싶어 여러 책을 섭렵했지만 제대로 설명하는 책이 별로 없었습니다. 이 책을 접하고 나서야 비로소 앱을 만드는 방법을 이해할 수 있었습니다(그만큼 알기 쉽게 설명하고 있다는 의미입니다). 그래서 이 책은 지금까지 앱 개발을 시작했지만 도중에 좌절했던 분들에게 더욱 추천합니다.

물론 이 책만으로 아이폰 앱 개발 능력을 숙달하기는 어렵습니다. 초심자들을 대상으로 하고 있기 때문입니다. 또한 상당히 많은 항목을 설명하느라 설명이 깊지 않습니다. 그러니 이 책을 통해 아이폰 앱 개발 방법을 개략적으로 이해한 후에는 아이폰 앱의 개발 언어인 오브젝티브 C에 대한 공부도 게을리하지 말기를 당부합니다. 저자도 말하고 있듯이 오브젝티브 C에 대한 숙달 정도가 참신하고 유용한 앱을 개발할 수 있는 기본이 됩니다.

이 책은 총 12장으로 구성되어 있습니다. 1~4장은 아이폰 앱 개발 환경 꾸미기와 오브젝티브 C에 대한 간략한 설명, 앱 개발 맛보기로 구성되어 있습니다. 5~10장은 아이폰 앱에서 사용하는 여러 기능을 어떻게 구현하는지 하나하나 예를 들어가며 설명하고 있습니다. 11~12장에서는 개발한 앱을 세상에 내놓기 위해 잘 마무리하는 방법과 실제 아이폰에서 테스트하는 방법을 설명하고 있습니다.

이 책은 두 번 번역이라는 우여곡절 끝에 빛을 보게 되었습니다(번역 도중 Xcode 버전이 바뀌는 바람에 처음부터 다시 번역해야 했습니다). 무한한 인내심을 발휘해주신 임성춘 편집장님과 조악한 문장이 글이 되게 해주신 편집자님, 맛깔스런 책으로 만들어주신 박진희님께 감사의 말씀을 전하며, 사랑하는 아내 윤주와 아빠의 배경이 되어준 멋진 딸 이현이에게도 고맙다는 말을 전합니다.

앱 스토어에 등록된 아이폰 앱도 수십 만개지만 비슷비슷한 앱이 거의 대부분입니다. 문제는 아이디어입니다. 앱들의 바다에서 마치 등대처럼 유용하게 반짝이는 앱 개발자가 되기를 기원합니다.

2011년 겨울을 맞이하는 동경 한편에서

김태현

목차

1장 애플리케이션 개발 환경 만들기

1-1 아이폰 앱 개발 준비물 16
Lecture _ 개발에 필요한 환경 16
1-2 iOS SDK 다운로드와 인스톨 20

2장 iOS SDK 기본

2-1 iOS SDK의 정체 30
2-2 Xcode 31
2-3 인터페이스 빌더 34
2-4 iOS 시뮬레이터 37

3장 오브젝티브 C 기본

3-1 오브젝티브 C 기본 40
Lecture _ 오브젝티브 C란 무엇인가? 40
Lecture _ 오브젝티브 C의 역사 41
Lecture _ 객체지향이란? 41
Lecture _ 기본 문법 43
Lecture _ 변수 44
3-2 포인터 변수 46
Lecture _ 포인터 변수란? 46
Lecture _ 문자열 데이터 처리 49
Lecture _ 많은 데이터를 일괄 처리하기 – 배열 53
Lecture _ 다량의 데이터를 키워드로 처리하기 55
3-3 제어문 58
Lecture _ if 문 58
if else 문 59
조건식 59

Lecture _ switch 문 60
Lecture _ while 문 62
Lecture _ for 문 62
Lecture _ 주석 63
3-4 클래스 64
Lecture _ 오브젝트란? 클래스란? 64
Lecture _ 오브젝트 사용 방법 65
오브젝트 만들기 65
젝트 해제하기 66
속성에 액세스하기 66
메소드 실행하기 66

4장 애플리케이션 개발 기본

4-1 프로젝트 작성 70
프로젝트 만들기 70
템플릿 선택 71
4-2 프로젝트 구조와 작성 순서 74
Lecture _ 자주 사용하는 파일 75
Lecture _ 화면 디자인과 프로그램 76
Lecture _ 배치한 컨트롤과 프로그램을 연결하는 방법 77
IBOutlet 연결 78
IBAction 연결 78
여러 가지 IBOutlet과 IBAction 연결 방법 79
Lecture _ 애플리케이션 전체에 관한 프로그램 82
Lecture _ 델리게이트(Delegate) 83
Lecture _ 용어 검색 84
4-3 첫 번째 앱 제작 85
Lecture _ 앱을 만드는 순서 85
Practice _ 「"안녕하세요"라고 표시합니다」 86

5장 기본 컨트롤을 사용하여 만들기

5-1 UILabel : 문자열 표시 92
Lecture _ UILabel로 할 수 있는 일 92
Practice _「레이블에 문자열을 표시합니다」 96
5-2 UIButton : 버튼을 눌러서 무언가를 합니다. 101
Lecture _ UIButton으로 할 수 있는 일 101
Practice _「버튼을 탭하면 문자열을 표시합니다」 104
5-3 UITextField : 텍스트 입력 109
Lecture _ UITextField로 할 수 있는 일 109
Practice _「문자열을 입력하면, 해당 문자열을 표시합니다」 116
5-4 UITextView : 긴 문자열 표시 121
Lecture _ UITextView로 할 수 있는 일 121
Practice _「긴 문자열을 표시합니다」 125
5-5 UISwitch : On/Off 스위치 130
Lecture _ UISwitch로 할 수 있는 일 130
Practice _「스위치를 전환하면, 문자열을 표시합니다」 132
5-6 UISlider : 슬라이더를 이용한 값 설정 137
Lecture _ UISlider로 할 수 있는 일 137
Practice _「슬라이더를 움직이면 그 값을 표시합니다」 140
5-7 UIDatePicker : 날짜 설정 145
Lecture _ UIDatePicker로 할 수 있는 일 145
Practice _「날짜를 선택하면 해당 날짜를 표시합니다」 149
5-8 UIAlertView : 경고창 표시 155
Lecture _ UIAlertView로 할 수 있는 일 155
Practice _「버튼으로 경고창을 표시합니다」 159
5-9 UIActionSheet : 액션 시트 표시 164
Lecture _ UIActionSheet로 할 수 있는 일 164
Practice _「버튼으로 액션 시트를 표시합니다」 168

6장 그림과 애니메이션 처리

6-1 UIImageView : 그림 표시 176
Lecture _ UIImageView로 할 수 있는 일 176
인터페이스 빌더로 레이아웃할 때 설정할 수 있는 것 176
그림 등록 방법 178
속성에서 설정하거나 확인할 수 있는 것 179
Practice _「리소스 그림을 표시합니다(소스 코드 미사용)」 181
Practice _「리소스 그림을 표시합니다(소스 코드 사용)」 183
6-2 책갈피 애니메이션 : UIImageView의 기능을 이용한 애니메이션 188
Lecture _ 책갈피 애니메이션을 만드는 방법 188
Practice _「버튼을 탭하면 애니메이션을 재생합니다」 192
6-3 간이 애니메이션 : UIView의 기능을 이용한 애니메이션 198
Lecture _ 간이 애니메이션을 만드는 방법 198
Practice _「버튼을 탭하면 애니메이션을 재생합니다」 203
6-4 타이머 애니메이션 : NSTimer를 이용한 애니메이션 208
Lecture _ NSTimer로 애니메이션을 만드는 방법 208
Practice _「레이블이 계속 움직이는 애니메이션」 211

7장 아이폰에 걸맞은 기능 구현

7-1 아이폰의 회전에 대응하기 : 아이폰을 옆으로 눕혔을 경우의 처리 218
Lecture _ 아이폰의 회전에 대응하는 방법 218
화면의 회전에 대응한 레이아웃 만들기 218
아이폰의 회전에 맞춰 화면을 자동으로 옆으로 눕히기 222
Practice _「아이폰의 회전에 맞춰 화면 회전시키기」 223
7-2 UIAccelerometer : 아이폰의 기울기 조사하기 227
Lecture _ 가속도 센서 사용법 227
Practice _「아이폰의 기울기를 수치로 표시합니다」 232
Practice _「기울이면 볼이 굴러가는 장남감」 237

7-3 CoreLocation.framework : 현재 위치와 방향 조사 243
Lecture _ CoreLocation을 사용한 현재 위치와 방향 조사 방법 243
Practice _「현재 위치의 위도와 경도, 방향을 알아냅니다」 249
7-4 MapKit.framework : 지도 표시 255
Lecture _ MKMapView를 이용한 지도 사용법 255
Practice _「현재 위치의 지도를 표시합니다」 262

8장 데이터 읽기와 쓰기

8-1 UIWebView : 웹 페이지 표시 270
Lecture _ 웹뷰를 이용한 웹 페이지 표시 방법 270
Practice _「앱 안에서 웹 페이지를 표시합니다」 273
8-2 UIImage : 웹 그림 표시 278
Lecture _ 이미지 뷰를 이용해 웹에 있는 그림을 표시하는 방법 278
Practice _「웹에 있는 그림을 표시합니다」 281
8-3 NSXMLParser : 웹에 있는 XML 읽어들이기 285
Lecture _ XML 해석 방법 286
Practice _「XML을 읽어들이고 표시합니다」 293
8-4 NSUserDefault : 일시적으로 데이터 저장 299
Lecture _ 유저 디폴트를 사용한 데이터 읽기/쓰기 방법 299
Practice _「문자열을 입력하고, 다음 번 실행 시에 표시합니다」 303

9장 멀티 뷰 앱 만들기

9-1 Utility Application : 앞뒤가 전환되는 앱 310
Lecture _ 유틸리티 애플리케이션 작성 방법 310
Practice _「뒤쪽 화면에서 스위치를 ON/OFF하면 앞쪽 화면이 변하는 앱」 317

9-2 Tab Bar Application : 여러 화면이 전환되는 앱 325
Lecture _ 탭바 애플리케이션 작성 방법 325
Practice _「두 개의 탭으로 화면을 전환할 때마다 카운트를 증가시킵니다」 336
Practice _「세 개의 탭으로 화면을 전환할 때마다 카운트를 증가시킵니다」 343

10장 테이블 표시

10-1 UITableView : 복수의 데이터를 리스트로 표시하기 350
Lecture _ 테이블 뷰의 구조 350
Lecture _ 테이블 뷰 사용 방법 351
Practice _「테이블에 같은 문자열을 20개 표시합니다」 360
Practice _「테이블에 설정한 문자열을 2x3행으로 표시합니다」 364
10-2 내비게이션 기반 : 화면을 슬라이드하면서 계층별로 화면 표시하기 369
Lecture _ 리스트를 선택(탭)하면 화면이 슬라이드되는 앱 작성 방법 369
Practice _「행을 선택하면 화면이 슬라이드되고 하위 계층 화면이 표시됩니다」 377
10-3 셀 사용자 지정 : 셀의 표시 사용자 지정하기 386
Lecture _ 셀을 조금만 변경하는 방법 386
테이블 뷰 전체의 배경색 설정 386
셀의 높이 설정 387
문자열 내용 설정 387
문자열의 색 설정 387
폰트나 크기 설정 388
셀의 액세서리 설정 388
셀 스타일 설정 389
Lecture _ 셀을 자유롭게 배치하는 방법 389
Practice _「테이블 뷰의 셀을 직접 작성합니다」 395

11장 앱 완성하기

11-1 아이콘 404
Lecture _ 아이콘 설정 방법 404
Practice_「앱에 아이콘을 설정합니다」 406
11-2 초기 화면 409
Lecture _ 초기 화면 작성 방법 409
Practice _「앱에 초기 화면을 설정합니다」 411
11-3 지역화 414
Lecture _ 지역화 방법 414
앱 이름 전환 방법 414
사용하는 문자열을 전환하는 방법 417
사용하는 그림을 전환하는 방법 419
Practice _「앱을 지역화합니다」 421

12장 실제 기기 테스트

12-1 실제 기기에서 테스트하기 426
Lecture _ iOS Developer Program에 유료 회원 등록하기 426
Lecture _「Organizer」에서 Provisioning Profile 만들기 427
Organizer 427
Lecture _ 앱을 실제 아이폰에 설치해서 실행하기 430

Practice

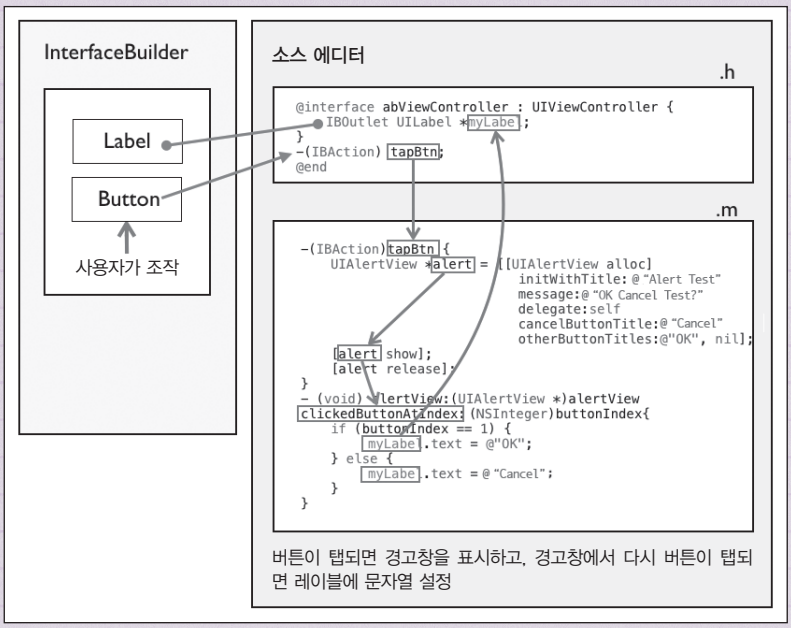
「버튼으로 경고창을 표시합니다」

난이도 : ★☆☆☆☆
제작 시간 : 5분

인터페이스 빌더에서 버튼을 배치하고, 소스 에디터에서 「버튼을 탭하면 경고창을 표시하고 OK와 Cancel 중 어느 버튼을 탭했는지 레이블에 표시하는 프로그램」을 만듭니다.

【개요】

먼저, 인터페이스 빌더에서 레이블과 버튼을 만듭니다.
다음으로, 이 레이블과 버튼을 프로그램에서 처리할 수 있게 준비합니다. 「레이블명(변수)」과 「버튼을 탭했을 때 실행할 기능명(메소드명)」을 헤더 파일(.h)에 만들고, 인터페이스 빌더에서 만든 레이블과 버튼을 「연결」합니다.
마지막으로, 프로그램 부분을 구현 파일(.m)에 작성합니다. 버튼을 탭했을 때 실행할 기능(메소드)을 만들고, 이 메소드에서 경고창을 만들어 표시하고 해제합니다. 또, clickedButtonAtIndex 메소드를 준비해서 어느 버튼을 탭했는지 레이블에 표시합니다.



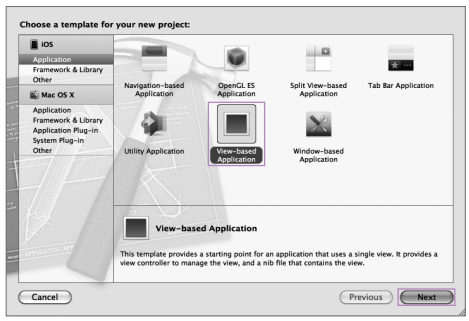
무슨 의미인가요?

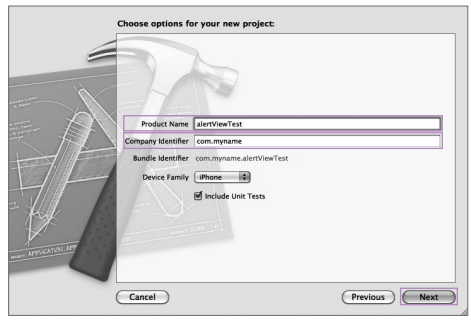
「버튼이 탭되어 tapBtn이 호출되면, 경고창을 만들어 표시합니다(확인합니다. OK입니까? Cancel, OK). 버튼을 탭하면 해당 버튼의 번호를 확인해서 OK인지 Cancel인지를 myLabel에 표시합니다.」

1 새 프로젝트를 만듭니다.

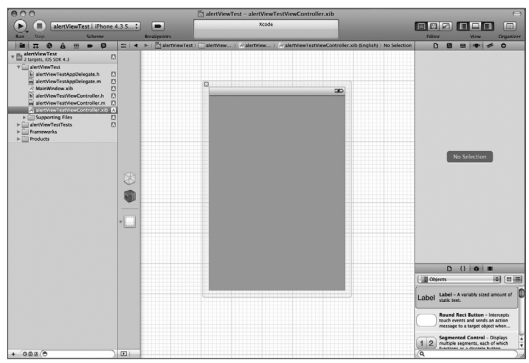
Xcode를 실행하고 「Create a new Xcode project」를 선택한 후, 「View-based Application」을 선택합니다.

[Next] 버튼을 클릭한 후 「Project Name」과 「Company Identifier」를 설정하고 다시 [Next]를 클릭하여 저장합니다. 여기서는 프로젝트명을 「alertViewTest」로, 회사 식별자를 「com.myname」으로 지정했습니다.



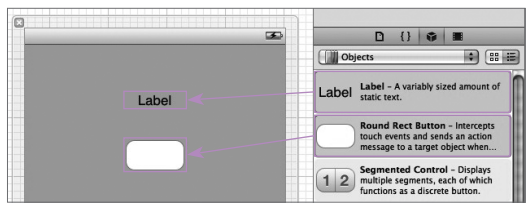


2 여러 파일이 생성되어 있는데, 그 중에서 「alertViewTestViewController.xib」를 클릭하여 인터페이스 빌더를 표시하고 메뉴에서 [View] – [Utilities] – [Object Library]를 선택합니다.



3 인터페이스 빌더에서 레이블과 버튼을 만듭니다.

인터페이스 빌더의 [Object Library]에서 「Label」과 「Round Rect Button」을 「View」로 드래그합니다. 레이블은 문자열을 표시할 수 있을 정도로 크기를 설정해둡니다.

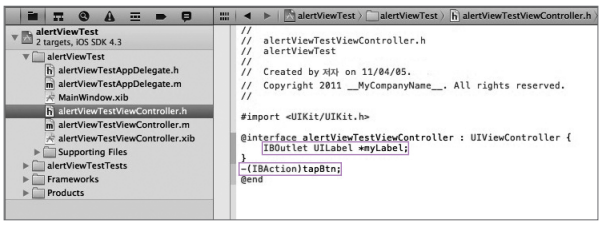


4 소스 에디터로 레이블명과 버튼을 탭했을 때 실행할 메소드명을 만듭니다.

Xcode로 돌아가서 .h 파일(alertViewTestViewController.h)을 선택합니다. .h 파일에는 이미 여러 가지 소스 코드가 입력되어 있습니다. @interface…{ ~ } 사이의 행과 } 다음 행에 각각 다음과 같은 코드를 추가합니다. 추가한 후에는 파일을 저장합니다.

```
01  IBOutlet UILabel *myLabel;
```

```
01  -(IBAction) tapBtn;
```

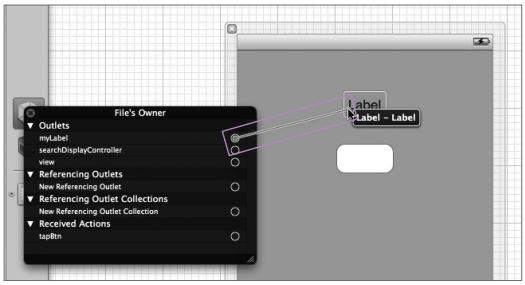


alertViewTestViewController.h

5 「레이블명과 인터페이스 빌더에서 만든 레이블을 연결」합니다.

「alertViewTestViewController.xib」를 클릭해서 인터페이스 빌더를 표시하고, Dock의 「File's Owner」를 오른쪽 버튼으로 클릭합니다.

「myLabel」 오른쪽에 있는 「○」를 드래그해서 선을 잡아늘려 레이블에 연결합니다. 이렇게 하면 「레이블명과 인터페이스 빌더에서 만든 레이블이 연결」됩니다.

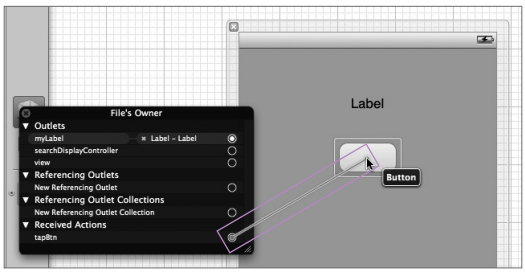


레이블명 오른쪽에 있는 ○를 드래그해서 레이블에 연결

6 「메소드명과 인터페이스 빌더에서 만든 버튼을 연결」합니다.

Dock의 「File's Owner」를 오른쪽 버튼으로 클릭합니다. 「Received Action」 항목에서 「tapBtn」 오른쪽에 있는 「○」를 드래그해서 선을 잡아늘려 버튼에 연결합니다.

이때 표시되는 액션 리스트에서 「Touch Up Inside(탭)」를 선택합니다. 이렇게 하면 「Xcode에서 만든 메소드명과 인터페이스 빌더에서 만든 버튼이 연결」됩니다.



메소드명 오른쪽에 있는 ○를 드래그해서 버튼에 연결

Touch Up Inside 선택

---

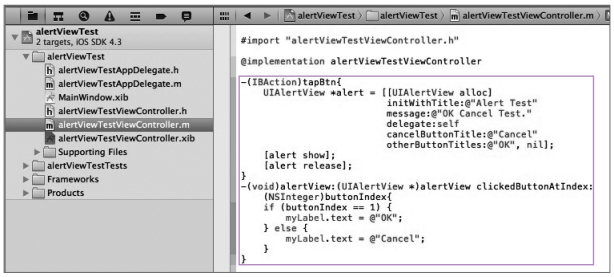
7 소스 에디터로 버튼을 탭하면 경고창을 표시하는 프로그램을 만듭니다.

.m 파일(alertViewdatePickerTestViewController.m)을 선택합니다.

@implementation 다음 행에 다음과 같은 코드를 추가합니다.

또, 어느 버튼을 탭했는지 확인할 수 있게 clickedButtonAtIndex 메소드를 만들고 그 안에서 레이블에 문자열을 표시합니다.

```
01  - (IBAction) tapBtn {
02        UIAlertView *alert = [[UIAlertView alloc]
02           initWithTitle:@"Alert Test"
03           message:@"OK Cancel Test?"
04           delegate:self
05           cancelButtonTitle:@"Cancel"
06           otherButtonTitles:@"OK", nil];
08        [alert show];
09        [alert release];
10  }
11  -(void)alertView : (UIAlertView *)alertView clickedButtonAtIndex: (NSInteger)buttonIndex {
12     if (buttonIndex == 1) {
13        myLabel.text = @"OK";
14     } else {
15        myLabel.text = @"Cancel";
16     }
17  }
```



alertViewTestViewController.m

8 시뮬레이터에서 실행합니다.

[Scheme]의 풀다운 메뉴에서 「Simulator」를 선택하고, [Run] 버튼을 눌러서 실행합니다.

iOS 시뮬레이터가 시작되고 프로그램(앱)이 실행됩니다.

버튼을 탭하면 경고창이 표시되고, [OK] 버튼을 선택하면 "OK", [Cancel] 버튼을 선택하면 "Cancel"이 표시됩니다.