

# iOS 9 핵심 노트 <sup>ଜন</sup>





# iOS 9 핵심 노트 <sup>☞</sup>™

**바 한빛미디이** 



표지 사진 **변진남** 이 책의 표지는 변진남 님이 보내 주신 풍경사진을 담았습니다. 리얼타임은 독자의 시선을 담은 풍경사진을 책 표지로 보여주고자 합니다. 사진 보내기 ebookwriter@hanbit.co.kr

#### iOS 9 핵심 노트

**초판발행** 2015년 11월 16일

지은이 야곰 / 펴낸이 김태현 펴낸곳 한빛미디어(주) / 주소 서울시 마포구 양화로 7길 83 한빛미디어(주) IT출판부 전화 02-325-5544 / 팩스 02-336-7124 등록 1999년 6월 24일 제10-1779호 ISBN 978-89-6848-754-5 15000 / 정가 12,000원

총괄 배용석 / 책임편집 김창수 / 기획·편집 정지연 / 교정 이미연 디자인 표지/내지 여동일 마케팅 송경석 / 영업 김형진, 김진불, 조유미

이 책에 대한 의견이나 오탈자 및 잘못된 내용에 대한 수정 정보는 한빛미디어(주)의 홈페이지나 아래 이메일로 알려주십시오. 한빛미디어 홈페이지 www.hanbit.co.kr / 이메일 ask@hanbit.co.kr

Published by HANBIT Media, Inc. Printed in Korea Copyright ⓒ 2015 야곰 & HANBIT Media, Inc. 이 책의 저작권은 야곰과 한빛미디어(주)에 있습니다. 저작권법에 의해 보호를 받는 저작물이므로 무단 복제 및 무단 전재를 금합니다.

지금 하지 않으면 할 수 없는 일이 있습니다. 책으로 펴내고 싶은 아이디어나 원고를 메일(ebookwriter@hanbit.co.kr)로 보내주세요. 한빛미디어(주)는 여러분의 소중한 경험과 지식을 기다리고 있습니다.

#### 지은이\_ **야곰**

yagom's blog(http://blog.yagom.net/)를 운영하는 iOS 개발 블로거이자 iOS 프 로그래머로, 아이폰 개발자 커뮤니티인 맥부기(http://cafe.naver.com/mcbugi)에서 강좌를 연재 중이다. 컴퓨터 교육을 전공했으며 비전공자와 학생들에게 컴퓨터 지식 을 더 쉽고 재미있게 알리는 데 관심이 많다. 2010년부터 iOS 개발을 시작해 현재까 지도 계속 iOS 관련 개발에 열정을 쏟고 있다.

내일 걱정은 모레 하는 것이 좋다고 생각하며 스스로 긍정적인 마음가짐을 빼면 시체 라고 말한다. 스스로 개발자라고 생각하지 않는 것을 보면 괴짜임이 틀림없다. 무엇보 다 여행과 요리를 좋아한다. 스마트폰의 수요가 늘어남에 따라 사용자가 필요한 기능도 많아졌습니다. 그에 발맞 추어 모바일 운영체제도 제각기 발전을 거듭하고 있고 iOS도 사용자의 편의를 위해 점점 더 빠르게 OS를 진화해 가고 있습니다. 이 책에 iOS 9에서 추가된 모든 기능을 담지는 못했지만, 공통으로 사용할 수 있는 굵직굵직한 신규 기능 위주로 정리해 보았 습니다.<sup>01</sup> 재빠르게 변하는 OS에 발맞추어 나가는 데 작지만 도움이 되면 좋겠습니다.

책을 읽다가 부족한 부분은 언제든지 블로그(http://blog.yagom.net) 또는 이메일 (yagomsoft@me.com)로 연락해 주세요.

항상 믿고 응원해 주시며 사랑을 아끼지 않는 가족과 동료, 이 책의 편집을 멋지게 해 주신 한빛미디어 정지연 님, 함께 응원해 주신 조희진 님, 이중민 님께 무한한 감사의 말씀을 전합니다. 사랑합니다.

<sup>01</sup> iOS 9에 관한 더 많은 정보를 얻으려면 https://developer.apple.com/ios/를 방문해 보세요.



이 책은 이미 iOS 개발 경험이 있는 개발자 중에서 빠르게 iOS 9의 기능과 예제를 훑 어보고자 하는 독자를 대상으로 합니다. Xcode 사용법과 Cocoa Touch 프레임워 크를 이해하고 있으며 관련 키워드들을 이해할 수 있어야 합니다. 또한, Swift 기본 문 법을 알고 있어야 합니다. iOS를 처음 접하거나 Cocoa Touch 프레임워크에 관한 전반적인 이해가 없는 독자는 소화하기 어려울 수 있습니다.

장마다 해당 기능을 간단히 설명하고 예제를 함께 수록하였습니다. 이 책의 예제는 Xcode 7.0, iOS SDK 9.0, Swift 2.0 버전 기준으로 작성되었으며, 언어와 iOS SDK, Xcode 버전에 따라 코드가 동작하지 않을 수도 있습니다. 예제는 다음 주소에서 다운 로드하세요. 예제에 사용한 이미지 파일은 다음 주소의 images 폴더에 있습니다.

https://bitbucket.org/yagom/ios9\_yagom

한빛 리얼타임은 IT 개발자를 위한 전자책입니다.

요즘 IT 업계에는 하루가 멀다 하고 수많은 기술이 나타나고 사라져 갑니다. 인 터넷을 아무리 뒤져도 조금이나마 정리된 정보를 찾기도 쉽지 않습니다. 또한, 잘 정리되어 책으로 나오기까지는 오랜 시간이 걸립니다. 어떻게 하면 조금이라 도 더 유용한 정보를 빠르게 얻을 수 있을까요? 어떻게 하면 남보다 조금 더 빨 리 경험하고 습득한 지식을 공유하고 발전시켜 나갈 수 있을까요? 세상에는 수 많은 종이책이 있습니다. 그리고 그 종이책을 그대로 옮긴 전자책도 많습니다. 전자책에는 전자책에 적합한 콘텐츠와 전자책의 특성을 살린 형식이 있다고 생 각합니다.

한빛이 지금 생각하고 추구하는, 개발자를 위한 리얼타임 전자책은 이렇습니다.

#### eBook First -빠르게 변화하는 IT 기술에 대해 핵심적인 정보를 신속하게 제공합니다

500페이지 가까운 분량의 잘 정리된 도서(종이책)가 아니라, 핵심적인 내용을 빠르게 전달하기 위해 조금은 거칠지만 100페이지 내외의 전자책 전용으로 개발한 서비스입 니다. 독자에게는 새로운 정보를 빨리 얻을 기회가 되고, 자신이 먼저 경험한 지식과 정보를 책으로 펴내고 싶지만 너무 바빠서 엄두를 못 내는 선배, 전문가, 고수 분에게 는 좀 더 쉽게 집필할 수 있는 기회가 될 수 있으리라 생각합니다. 또한, 새로운 정보 와 지식을 빠르게 전달하기 위해 O'Reilly의 전자책 번역 서비스도 하고 있습니다.

#### 무료로 업데이트되는 전자책 전용 서비스입니다

종이책으로는 기술의 변화 속도를 따라잡기가 쉽지 않습니다. 책이 일정 분량 이상으 로 집필되고 정리되어 나오는 동안 기술은 이미 변해 있습니다. 전자책으로 출간된 이 후에도 버전 업을 통해 중요한 기술적 변화가 있거나 저자(역자)와 독자가 소통하면서 보완하여 발전된 노하우가 정리되면 구매하신 분께 무료로 업데이트해 드립니다.

### ➔ 독자의 편의를 위해 DRM-Free로 제공합니다

구매한 전자책을 다양한 IT 기기에서 자유롭게 활용할 수 있도록 DRM-Free PDF 포맷으로 제공합니다. 이는 독자 여러분과 한빛이 생각하고 추구하는 전자책을 만들 어 나가기 위해 독자 여러분이 언제 어디서 어떤 기기를 사용하더라도 편리하게 전자 책을 볼 수 있도록 하기 위함입니다.

# ▲ 전자책 환경을 고려한 최적의 형태와 디자인에 담고자 노력했습니다

종이책을 그대로 옮겨 놓아 가독성이 떨어지고 읽기 어려운 전자책이 아니라, 전자책 의 환경에 가능한 한 최적화하여 쾌적한 경험을 드리고자 합니다. 링크 등의 기능을 적극적으로 이용할 수 있음은 물론이고 글자 크기나 행간, 여백 등을 전자책에 가장 최적화된 형태로 새롭게 디자인하였습니다.

앞으로도 독자 여러분의 충고에 귀 기울이며 지속해서 발전시켜 나가겠습니다.

지금 보시는 전자책에 소유 권한을 표시한 문구가 없거나 타인의 소유권한을 표시한 문구가 있다면 위법하게 사용하고 있을 가능성이 큽니다. 이 경우 저작권법에 따라 불이익을 받으실 수 있습니다.

다양한 기기에 사용할 수 있습니다. 또한, 한빛미디어 사이트에서 구매하신 후에는 횟수와 관계없이 내려받으 실 수 있습니다.

한빛미디어 전자책은 인쇄, 검색, 복사하여 붙이기가 가능합니다.

전자책은 오탈자 교정이나 내용의 수정·보완이 이뤄지면 업데이트 관련 공지를 이메일로 알려 드리며, 구매하 신 전자책의 수정본은 무료로 내려받으실 수 있습니다.

이런 특별한 권한은 한빛미디어 사이트에서 구매하신 독자에게만 제공되며, 다른 사람에게 양도나 이전은 허 락되지 않습니다.



chapter <b>1</b>	새로운 연락처 프레임워크 011
	1.1 Contacts와 ContactsUI 011
	1.2 돌아보기 021
chapter 2	SafariServices 프레임워크 023
	2.1 SECoferil/iou/Controller 022
	2.1 SESTIMATIVEWCONTROLLER 023
chapter 3	UIKit 프레임워크 027
	3.1 스텍 뉴 027
	3.2 너국 위원진 오토레이아웃 세약 수정 036
	3.3 도티피케이션 액션 043
	3.4 3D Touch 050
	3.5 돌아보기 071
chapter <b>4</b>	iPad 멀티태스킹 073
	4.1 iPad 멀티태스킹 지원 073
	4.2 참고사항 084
	4.3 돌아보기 086
chapter 5	스포트라이트 시원
	5.1 스포트라이트 지원 087
	5.2 돌아보기 <b>104</b>

chapter 6	Watch — 107
	6.1 iOS의 WatchKit과 watchOS의 WatchKit <b>107</b> 6.2 iOS의 WatchKit 애플리케이션 <b>110</b>
	6.3 watchOS의 WatchKit 애플리케이션 137
	6.4 돌아보기 167
chapter 7	앱 씨닝 169
	7 1 얘 씨님이라 169
	7.2 슬라이싱 169
	7.3 HERE 171
	7.0
chapter 8	Photos와 PhotosUI 프레임워크 175
	8.1 Live Photos 175
	부록 191
	▲ 1 Xcode 프로제트에 버들 리소스 초가하기 101
	A 2 App Transport Security 101
	A.5 watch05 2에서 105 에글니케이션과 중간이기 194

# <sup>chapter</sup> ] 새로운 연락처 프레임워크

## 1.1 Contacts와 ContactsUI

애플은 iOS 9 발표와 함께 새로운 연락처 프레임워크를 선보였는데, Contacts 와 ContactsUI 프레임워크가 그 주인공입니다.

Contacts 프레임워크는 모든 애플 플랫폼에서 사용할 수 있으며 iOS 9에서 사 라진 기존의 Address Book 프레임워크를 대체하게 됩니다. 따라서 Contacts 프레임워크로 iOS 또는 매킨토시 기기에 저장된 사용자의 연락처 정보에 접근할 수 있습니다. 애플리케이션 대부분은 연락처를 수정하기보다는 읽기만 하므로 이 프레임워크는 스레드에서 사용해도 안전하며 읽기 전용으로 최적화되어 있습니 다(물론 수정과추가 기능도 있습니다).

Contacts 프레임워크는 연락처를 관리하기 위해 기본으로 'CNContact'라는 클 래스를 이용합니다. CNContact 클래스의 객체는 변경할 수 없는 객체로, 모양새 와 쓰임이 NSDictionary와 유사합니다. 물론 CNMutableContact 클래스의 객 체를 이용하면 연락처의 속성을 변경할 수 있습니다. 연락처 속성 중에 전화번호, 이메일과 같이 여러 항목이 있으면 CNLabeledValue 객체의 배열이 그 값을 나 타내게 됩니다.

CNLabeledValue 객체 또한 스레드에서 사용할 수 있으며, 변경할 수 없는 레이 블과 값의 튜플<sup>Tuple</sup>로 이루어져 있습니다. 레이블은 집전화, 직장전화 같은 설명 값을 표현합니다. Contacts 프레임워크는 미리 지정된 레이블도 제공하지만, 직 접 사용자 정의 레이블도 생성할 수 있습니다.

그림 1-1 CNContact 객체와 주변 객체의 상관도<sup>01</sup>



ContactsUI 프레임워크는 기기 연락처 정보의 열람, 수정, 선택, 생성 등을 편리 하게 도와주는 컨트롤러를 제공하는데, CNContactPickerViewController 와 CNContactViewController 등을 포함합니다.

그럼 새로운 연락처 프레임워크를 사용하여 새로운 연락처를 생성하고, 수정해 보는 예제를 진행하겠습니다. 우선 Xcode를 실행하여 새로운 [Single View Application] 프로젝트를 생성합니다.

C Loading					💼 📀 🤿
hoose a template for your nr IOS Application Framework & Library Watch OS Application Framework & Library OS X Application Framework & Library System Plug-in Other	w project: Master-Detail Application Game Single View Applic This template provid	Page-Based Application	I) Bingle View Application	Tabbed Application	D (
Cancel	a view controller to n	nanage the view, and :	e storyboard or hib hie tr	avious Next	

프로젝트 이름은 'BasicContacts'라고 짓습니다(절대 'Contacts'라고 이름 짓지 마

그림 1-2 Single View Application 프로젝트 생성

<sup>01</sup> 참고 : https://goo.gl/0hZFZb

세요. 프레임워크와 동일한 이름을 사용하는 프로젝트는 모듈명이 기본으로 프로젝트명과 같아

지기 때문에 프레임워크를 import하기 골치 아파집니다).

그림 1-3 프로젝트 이름 지정

Choose options for your new project:			
Product Name:	RasicContacts		
Organization Name:	yagom		
Organization Identifier:	net.yagom		
Bundle Identifier:	net.yagom.BasicContacts		
Language:	Swift	0	
Devices:	iPhone	0	
	Use Core Data		
	Include Unit Tests		
	Include UI Tests		
Cancel		Previous	Next
Cancel		Previous	

프로젝트 생성이 완료되면 ViewController.swift 파일로 이동하여 다음 코드 를 작성합니다. 소스 코드 중 clickShowContactsPickerController 메서드 는 주소록 선택 화면을 보여주며, clickShowContactsController 메서드는 선 택된 연락처를 보여줍니다. 그리고 clickEditContacts 메서드는 선택된 연락 처 정보를 코드로 수정하여 업데이트하는 메서드입니다. 그 외 내용은 코드에 주 석을 달아 설명하였으니 코드를 참고하시기 바랍니다.

[코드 1-1] BasicContacts 프로젝트의 ViewController.swift 파일 전체 코드

```
import UIKit
// Contacts 프레임워크와 ContactsUI 프레임워크를 import합니다.
import Contacts
import ContactsUI
// CNContactPickerDelegate 역할을 수행할 것을 명시합니다.
class ViewController: UIViewController, CNContactPickerDelegate {
    // MARK: - Properties
    // 선택된 연락처 정보를 가질 객체입니다.
    var selectedContact: CNContact?
    // MARK: - Methods
```

```
func dismissModalViewController() {
       self.dismissViewControllerAnimated(true, completion: nil)
   }
   // MARK: - IBActions
   @IBAction func clickShowContactsPickerController() {
       // 연락처 선택 뷰 컨트롤러를 생성하여 화면에 보여줍니다.
       let contactsPicker: CNContactPickerViewController = CNContactPicker
ViewController()
       contactsPicker.delegate = self
       self.presentViewController(contactsPicker, animated: true, completion: nil)
   }
   @IBAction func clickShowContactsController() {
       // 선택하여 가져온 연락처 정보가 있다면 연락처 정보를 보여줍니다.
       if let contact = selfselectedContact {
          let contactViewController: CNContactViewController = CNContactView
Controller(forContact: contact)
          let navigation = UINavigationController(rootViewController:
contactViewController)
          let closeButton = UIBarButtonItem(title: "Close", style:
UIBarButtonItemStyle.Done, target: self, action: "dismissModalViewController")
          contactViewController.navigationItem.rightBarButtonItem = closeButton
          self.presentViewController(navigation, animated: true, completion: nil)
      }
   }
   @IBAction func clickEditContacts() {
       // 선택된 연락처 정보를 수정하는 한 Mutable 객체로 복사해옵니다.
       if let mutableContact = self.selectedContact?_mutableCopy() as?
CNMutableContact {
          // 이름을 변경할 수 있습니다.
          mutableContact.givenName = "yagom"
          mutableContact.familyName = "Jo"
          // 이메일 정보를 수정할 수 있습니다.
          let email = CNLabeledValue(label:CNLabelWork, value: "yagomsoft@
gmail.com ")
          let iCloudMail = CNLabeledValue(label:CNLabelEmailiCloud,
value: "yagomsoft@me.com")
```

```
mutableContact.emailAddresses = [email, iCloudMail]
mutableContact.phoneNumbers = [CNLabeledValue(
```

label:CNLabelPhoneNumberiPhone,

value:CNPhoneNumber(stringValue: "010-0000-0000 "))]

```
// 홈페이지나 블로그 등 URL 정보를 변경할 수 있습니다.
let blogURLString: String = "http://blog.yagom.net"
mutableContact.urlAddresses = [CNLabeledValue(label:
CNLabelURLAddressHomePage, value: blogURLString)]
```

// 주소를 변경할 수 있습니다. let homeAddress = CNMutablePostalAddress() homeAddress.street = "Sang-su"

homeAddress.city = "Mapo"

homeAddress.state = "Seoul"

homeAddress.postalCode = "00000"

```
mutableContact.postalAddresses = [CNLabeledValue(label:CNLabelHom
```

e, value:homeAddress)]

```
// 생임을 변경할 수 있습니다
let birthday = NSDateComponents()
birthday.day = 11
birthday.month = 7
// 년도는 비밀이니까 넣지 않겠습니다
// birthday.year = ???
mutableContact.birthday = birthday
// 연락처 정보를 저장하거나 가져올 수 있는 CNContactStore 객체입니다.
let store = CNContactStore()
// 저장 또는 변경을 위한 요청 객체입니다.
let updateRequest = CNSaveRequest()
updateReguest.updateContact(mutableContact)
// 저장(기존 정보 업데이트)을 시도해보고 실패하면 로그를 남깁니다.
do {
   try store.executeSaveRequest(updateRequest)
   self.selectedContact = mutableContact
} catch let error {
   print(error)
   return
```

```
}
}
}
```

```
// MARK: - ContactsPickerViewController Delegate
```

// 주소록에서 한 개의 연락처 정보를 선택했을 때 연락처 객체가 넘어옵니다.

```
// contactPicker:didSelectContacts 메서드와 동시에 구현할 경우
didSelectContacts 메서드가 우선됩니다.
```

func contactPicker(picker: CNContactPickerViewController, didSelectContact contact: CNContact) {

self.selectedContact = contact

}

```
/*
// 예제에서 다루는 것 외에 다양하 델리게이트 메서드를 소개하기 위해 주석 처리합니다.
```

// 주소록에서 여러 개의 연락처 정보를 선택했을 때 연락처 객체의 배열이 넘어옵니다. func contactPicker(picker: CNContactPickerViewController, didSelectContacts contacts: [CNContact]) {

}

// 주소록의 연락처에서 하나의 속성을 선택했을 때 속성 객체가 넘어옵니다.

```
// contactPicker:didSelectContactProperties 메서드와 동시에 구현할 경우
didSelectContactProperties 메서드가 우선됩니다.
```

```
func contactPicker(picker: CNContactPickerViewController,
didSelectContactProperty contactProperty: CNContactProperty) {
```

}

```
// 주소록의 연락처에서 여러 개의 속성을 선택했을 때 속성 객체의 배열이 넘어옵니다.
func contactPicker(picker: CNContactPickerViewController,
didSelectContactProperties contactProperties: [CNContactProperty]) {
```

} \*/ }

코드 작성을 마치면 Main.storyboard 파일로 이동하여 버튼 세 개를 올려주고 타이틀을 [그림 1-4]처럼 변경한 후 액션을 알맞게 연결합니다.

#### 그림 1-4 버튼의 액션 연결상태

V BasicContacts	View Controller Scene					
BasicContacts	View Controller					
AppDelegate.swift	Top Layout Guide					
ViewController.swift	Bottom Lavout					
Main.storyboard	View					
Accete vogesete	B Show Contac			O Mou	Controllor	
	B Show Contac			Triggered Seques	Controller	
LaunchScreen.storyboard	B Edit Contact			manual		
info.plist	Einet Beeneerder			▼ Outlets		
Products	(i) First Responder		Show Contacts Picker View Controller	searchDisplayController		0
	Charles and Dates Dates			view		۲
			Show Contacts View Controller	<ul> <li>Presenting Segues</li> </ul>		
				relationship		
				show		0
			Edit Contact Information	show detail		ō.
				present modally		ŏ.
				popover presentation		No.
				embed		Ň.
				pour (deprecated)		×.
				custom		×
				Referencing Outlets		
				New Beferencing Outlet		
				▼ Beferencing Outlet Collect	lons	
				New Referencing Outlet Collect	ion	
		~		▼ Received Actions		
		$\rightarrow$		clickEditContacts	* Edit Contact Information	$\odot$
					Touch Up Inside	
				clickShowContactsController	* Show Contacts View C Touch Up Inside	۲
				clickShowContactsPickerCo	* Show Contacts Picker Touch Up Inside	۲
						<u> </u>

그리고 단축키 [command + R] 또는 Xcode 상단의 [실행] 버튼을 눌러 애플 리케이션을 실행한 후, 화면 가장 위의 버튼인 [Show Contacts Picker View Controller]를 선택합니다.

그림 1-5 Show Contacts Picker View Controller 버튼 선택



그러면 다음과 같이 기기에 저장된 연락처의 주소록이 화면에 표시되는데 이중 주 소록에서 가져오려는 연락처를 선택합니다. 여기서는 'Anna Haro'라는 이름으 로 저장된 연락처를 선택했습니다.

#### 그림 1-6 저장된 연락처 선택

Carrier 穼	6:07 PM	-
Groups	All Contacts	Cancel
	Q Search	
Kate <b>Bell</b>		
Anna <b>Haro</b>		
Daniel <b>Higg</b>	g <b>ins</b> Jr.	
David <b>Taylo</b>	or	
Hank M. Za	akroff	

연락처를 선택하면 다시 처음 화면으로 돌아갑니다. 이번에는 두 번째 [Show Contacts View Controller] 버튼을 선택합니다.

그림 1-7 Show Contacts View Controller 버튼 선택



연락처 선택 화면에서 선택한 연락처 정보가 화면에 표시됩니다.

그림 1-8 Contacts View Controller에 보이는 연락처 정보

Carrier 穼	6:14 PM	Close
AH Anna "Anni	a <b>Haro</b> e″	
home (555) 522-824	3	0 C
home anna-haro@ma	ac.com	$\bigcirc$
home 1001 Leavenwo Sausalito CA 9 USA	orth Street 4965	
birthday August 29, 198	5	
anniversary February 15, 20	002	
Notes		

이번에는 앞에서 코드로 작성한 정보로 연락처 정보를 변경해 봅니다. [Close] 버 튼을 눌러 화면을 닫고 첫 화면 하단의 [Edit Contact Information] 버튼을 선 택합니다.

그림 1-9 Edit Contact Information 버튼 선택



이 애플리케이션에서 연락처 정보에 코드로 접근하는 것이 처음이라면 다음과 같 은 경고가 발생합니다. 연락처 정보에 접근하는 것을 허용하기 위해 [OK]를 선택 해 주세요.

그림 1-10 연락처 정보 접근 허용 여부를 묻는 경고창



이후 특별한 오류 로그가 없다면 정상적으로 편집된 것입니다. [Show Contacts View Controller] 버튼을 선택하여 변경된 연락처 정보를 확인해 봅니다. 미리 코드로 변경하려고 지정해 둔 정보들이 편집된 것을 확인할 수 있습니다.

그림 1-11 수정된 연락처 정보

yagom Jo "Annie"	
iPhone 01000000000	Q &
work yagomsoft@gmail.com	$\bigcirc$
iCloud yagomsoft@me.com	$\bigcirc$
homepage http://blog.yagom.net	
home Sang-su Mapo Seoul 00000	
birthday July 11	
anniversary February 15, 2002	

이처럼 Contacts 프레임워크와 ContactsUI 프레임워크를 사용하면 기존의 Address Book보다 더 편리하고 쉽게 사용자의 연락처에 접근하고 수정할 수 있 습니다. 또한, 기본 UI로 깔끔하게 연락처 정보를 표현할 수 있습니다. 사용자가 미리 지정해둔 문구들을 연락처에 삽입하는 등의 기능을 담은 애플리케이션을 만 들어보면 좋을 것 같은데요?

### 1.2 돌아보기

1장에 사용한 클래스를 간단히 정리해 보겠습니다. 각 항목에 관한 자세한 내용은 링크를 참고해 주시기 바랍니다.

구분	설명	참고 링크
CNContacts	Contacts와 ContactsUI 프레임워 크에서 연락처 정보를 담는 역할을 합니다.	https://goo.gl/J6qedD
CNMutableContacts	CNContacts 객체를 수정할 수 있 도록 확장한 클래스입니다.	https://goo.gl/6Caaew
CNLabeledValue	CNContacts 객체에서 연락처 속성 을 표현할 때 사용하는 클래스입니다.	https://goo.gl/deywdq
CNContactStore	연락처 정보 저장, 불러오기 등을 할 때 사용되는 클래스입니다.	https://goo.gl/ID4CTH
CNSaveRequest	CNContactStore에 연락처 정보를 저장하거나 수정하기를 요청하는 데 필요한 클래스입니다.	https://goo.gl/R6JY1h
CNContactPickerView Controller	연락처 정보를 선택하여 정보를 가져 오는 데 사용하는 뷰 컨트롤러 클래스 입니다.	https://goo.gl/zTTgqH
CNContactViewController	연락처 정보를 보여주기 위해 사용되 는 뷰 컨트롤러 클래스입니다.	https://goo.gl/oWWDrM

# <sup>chapter</sup> 2 SafariServices 프레임워크

## 2.1 SFSafariViewController

iOS 9 전에는 웹 페이지를 제대로 탐색할 수 있는 화면을 구현하려면 웹뷰 (WebView)와 [앞으로 가기], [뒤로 가기] 버튼을 포함한 툴바를 함께 구현해야 했 습니다. 심지어는 URL을 입력하는 필드도 없었습니다.

iOS 9부터 기존의 SafariServices 프레임워크에 사파리 뷰 컨트롤러가 추가되 어 이 귀찮음을 단방에 해결해 줄 수 있습니다. 이제 간편하게 웹뷰를 볼 수 있는 컨트롤러를 코드 몇 줄로 바로 보여줄 수 있다는 뜻입니다! 별다른 설명이 필요 없 죠! 신나게 만들어 봅시다.

먼저 'EasySafari'라는 이름으로 새로운 Single View Application 프로젝트 를 생성합니다(새로운 프로젝트를 생성하는 방법은 [그림 1-2]를 참고하세요). 새로운 프로 젝트를 생성하면 코드를 작성합니다. 코드는 정말 간단합니다! 몇 줄 쓰지 않아도 되니 얼른 작성해 봅시다.

ViewController.swift 파일로 이동하여 다음 코드를 작성합니다. 자세한 내용 은 코드 주석을 참고하시기 바랍니다.

```
[코드 2-1] EasySafari 프로젝트의 ViewController.swift 파일 전체 코드
```

import UIKit

// 더욱 강력해진 SafariServices 프레임워크를 import합니다.

import SafariServices

// ViewController 객체가 SafariViewControllerDelegate의 역할을 수행할 것을 명시합니다.

```
class ViewController: UIViewController, SFSafariViewControllerDelegate {
```

```
// MARK: - IBActions
```

```
@IBAction func clickShowSafariViewController() {
```

```
// NSURL 객체 생성에 실패할 확률이 있으므로 if-let 구문을 사용하여 nil이 반
환되지 않는지 확인합니다.
```

```
if let url = NSURL(string: "http://blog.yagom.net") {
```

```
// 사파리 뷰 컨트롤러를 원하는 URL을 사용하여 생성하고 화면에 보여줍니다.
let safariViewController = SFSafariViewController(URL: url)
safariViewController.delegate = self
```

```
self.presentViewController(safariViewController, animated: true,
completion: nil)
```

```
}
   }
   // MARK: - SFSafariViewControllerDelegate
   // 사파리 뷰 컨트롤러의 Done(완료) 버튼을 선택했을 때 호출되는 델리게이트 메서드
입니다.
   func safariViewControllerDidFinish(controller: SFSafariViewController) {
      // 모달로 올라온 화면을 내려줍니다.
      self.dismissViewControllerAnimated(true, completion: nil)
   }
   /*
   // 또다른 SFSafariViewControllerDelegate 메서드입니다.
   // 특정 Activity를 특정 URL에 맞게 제공할 수 있습니다.
   func safariViewController(controller: SFSafariViewController,
activityItemsForURL URL: NSURL, title: String?) → [UIActivity] {
   }
   */
```

```
}
```

코드를 작성한 후에는 Main.storyboard 파일로 이동하여 버튼 하나를 올려주

#### 고 액션도 연결합니다.

#### 그림 2-1 스토리보드에서 액션 연결

1 10						
🔻 📋 EasySafari	View Controller Scene					
The EasySafari	View Controller					
AppDelegate.swift	Top Layout Guide					
ViewController.swift	Bottom Lavout					
Main storyboard	View					
Assate vesseste	B Show Safari					
Assets.xcassets	First Besponder			L		
LaunchScreen.storyboard	E Evit				View Controller	
Info.plist	Characterized Fater Dalat				<ul> <li>Triggered Segues</li> </ul>	Q
Products	-> Storyboard Entry Point		show Safari View Co	ontrolle	manual	0
					▼ Outlets	
					searchDisplayController	0
					view Xiew	
					Presenting Segues	
					relationship	, o i
					show	<u> </u>
					show detail	<u> </u>
					present modally	<u> </u>
					popover presentation	<u> </u>
					embed	<u> </u>
					push (deprecated)	×
					modal (deprecated)	× ×
					Referencing Outlets	
					New Referencing Outlet	
					▼ Beferencing Outlet Collections	Ŭ Ŭ
					New Referencing Outlet Collection	0
		/			▼ Beceived Actions	
					clickShowSafariViewController × Show Safari View Co	ntr 🔍 🔿

이제 준비 끝! 바로 실행해 봅시다! 단축키 [command + R] 또는 Xcode 상단의 [실행] 버튼을 눌러 애플리케이션을 실행한 후 스토리보드에서 액션을 연결한 버 튼을 선택해 주세요. 여기서는 [Show Safari View Controller]를 선택합니다.

그림 2-2 EasySafari 애플리케이션 첫 실행 화면

С	arrier ᅙ	8:06 PM
	Show Safari V	iew Controller
_		

완벽한 사파리 화면이 반겨줄 거예요! 버튼도 다양하게 있습니다. 오른쪽 아래 사 파리 아이콘 버튼을 선택하면 현재 보고 있는 페이지가 사파리 애플리케이션에서 보이게 됩니다.

#### 그림 2-3 사파리 뷰 컨트롤러의 모습



SFSafariViewController에서 보는 페이지들은 기본 내장 사파리 애플리케이 션과 세션, 쿠키 모두 연동되니 참고하시기 바랍니다. 지난 몇 년간의 제바람이 이 제야 실현되어 개인적으로는 너무나 기쁘네요. :)

# 2.2 돌아보기

2장에 사용한 컨트롤러를 간단히 정리했습니다. 항목에 관한 자세한 내용은 링크 를 참고해 주시기 바랍니다.

구분	설명	참고 링크
SFSafariViewController	실제 사파리 애플리케이션과 유사한 화 면 구조로 웹 페이지를 서핑할 수 있는 뷰를 제공하는 뷰 컨트롤러입니다.	https://goo.gl/t7kzMm