

[핵심만 골라 배우는 iOS 8 프로그래밍] 코드 수정사항

작성자: 황반석
작성일: 2015.04.17

아래는 iOS 8.3 업데이트로 인해 책의 코드에서 수정해야 할 부분을 정리한 것들입니다. 수정해야 할 부분은 붉은색으로 표기해 두었으며, iOS 8.3 에 맞춰 테스트한 책 소스 전체는 아래의 링크에서 받으실 수 있습니다. (※ 아래 편집본은 코드 들여쓰기가 무시된 상태임을 알려드립니다)

소스 코드 다운로드: <https://github.com/Jpub/iOS8>

테스트 환경

OS: Mac OS X Yosemite 10.10.3

iOS: SDK 8.3

Xcode: 6.3

132 페이지 - 첫 번째 코드 블록

```
override func touchesBegan(touches: Set<NSObject>, withEvent event: UIEvent) {
```

210 페이지 - 두 번째 코드 블록

```
override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {  
    let destination = segue.destinationViewController as! Scene2ViewController  
    destination.labelText = "Arrived from Scene 1"  
}
```

236 페이지 - 두 번째 코드 블록

```
override func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath:  
    NSIndexPath) -> UITableViewCell {  
    let cell = self.tableView.dequeueReusableCellWithIdentifier(  
        "AttractionTableViewCell", forIndexPath: indexPath) as! AttractionTableViewCell  
    let row = indexPath.row  
    cell.attractionLabel.font = UIFont.preferredFontForTextStyle(UIFontTextStyleHeadline)  
    cell.attractionLabel.text = attractionNames[row]  
    cell.attractionImage.image = UIImage(named: attractionImages[row])  
    return cell  
}
```

237 페이지 - 첫 번째 코드 블록

```
let cell = self.tableView.dequeueReusableCellWithIdentifier(  
    "AttractionTableViewCell", forIndexPath: indexPath) as! AttractionTableViewCell
```

245 페이지 - 두 번째 코드 블록

```

override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {
if segue.identifier == "ShowAttractionDetails" {
let detailViewController = segue.destinationViewController
as! AttractionDetailViewController
let myIndexPath = self.tableView.indexPathForSelectedRow()
let row = myIndexPath?.row
detailViewController.webSite = webAddresses[row!]
}
}

```

251 페이지 - 두 번째 코드 블록

```

override func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath:
NSIndexPath) -> UITableViewCell {
let cell = tableView.dequeueReusableCellWithIdentifier("Cell", forIndexPath: indexPath) as!
UITableViewCell
UITableViewCell
cell.textLabel?.text = siteNames![indexPath.row]
return cell
}

```

253 페이지 - 첫 번째 코드 블록

```

let controller = (segue.destinationViewController as!
UINavigationController).topViewController
as! DetailViewController
controller.detailItem = urlString
controller.navigationItem.leftBarButtonItem =
self.splitViewController?.displayModeButtonItem()
controller.navigationItem.leftItemsSupplementBackButton = true
}
}}

```

253 페이지 - 두 번째 코드 블록

```

func configureView() {
// Update the user interface for the detail item.
if let detail: AnyObject = self.detailItem {
if let myWebview = self.webView {
let url = NSURL(string: detailItem as! String)
let request = NSURLRequest(URL: url!)
myWebview.scalesPageToFit = true
myWebview.loadRequest(request)
}
}
}}

```

263 페이지 - 첫 번째 코드 블록

```

import UIKit

```

```

class ContentViewController: UIViewController {
var dataObject: AnyObject?
@IBOutlet weak var webView: UIWebView!
.
.
.
override func viewWillAppear(animated: Bool) {
super.viewWillAppear(animated)
webView.loadHTMLString(dataObject as! String, baseURL: NSURL(string: ""))
}
...

```

265 페이지 - 첫 번째 코드 블록

```

func viewControllerAtIndex(index: Int) -> ContentViewController? {
if (self.pageContent.count == 0) || (index >= self.pageContent.count) {
return nil
}
let storyboard = UIStoryboard(name: "Main", bundle: NSBundle.mainBundle())
let dataViewController = storyboard.instantiateViewControllerWithIdentifier("contentView")
as! ContentViewController
dataViewController.dataObject = pageContent[index]
return dataViewController
}
func indexOfViewController(viewController: ContentViewController) -> Int {
if let dataObject: AnyObject = viewController.dataObject { return
self.pageContent.indexOfObject(dataObject)
} else {
return NSNotFound
}}

```

265 페이지 - 두 번째 코드 블록

```

let storyboard = UIStoryboard(name: "Main", bundle: NSBundle.mainBundle())
let dataViewController = storyboard.instantiateViewControllerWithIdentifier("contentView")
as! ContentViewController

```

266 페이지 - 첫 번째 코드 블록

```

func pageViewController(pageViewController: UIPageViewController,
viewControllerBeforeViewController viewController: UIViewController) -> UIViewController? {
var index = self.indexOfViewController(viewController as! ContentViewController)
if (index == 0) || (index == NSNotFound) { return nil
}
index—
return self.viewControllerAtIndex(index) }

```

```

func pageViewController(pageViewController: UIPageViewController,
viewControllerAfterViewController viewController: UIViewController) -> UIViewController? {
var index = self.indexOfViewController(viewController as! ContentViewController)
if index == NSNotFound { return nil
}
index++
if index == self.pageContent.count {
return nil }
return self.viewControllerAtIndex(index) }

```

267 페이지 - 첫 번째 코드 블록

```

override func viewDidLoad() {
super.viewDidLoad()
self.createContentPages()
self.pageController = UIPageViewController( transitionStyle: .PageCurl,
navigationOrientation: .Horizontal, options: nil)
self.pageController?.delegate = self self.pageController?.dataSource = self
let startingViewController: ContentViewController = self.viewControllerAtIndex(0)!
let viewControllers: NSArray = [startingViewController]
self.pageController!.setViewControllers(viewControllers as <AnyObject>,
direction: .Forward, animated: false, completion: nil)
self.addChildViewController(self.pageController!)
self.view.addSubview(self.pageController!.view)
var pageViewRect = self.view.bounds self.pageController!.view.frame = pageViewRect
self.pageController!.didMoveToParentViewController(self)
}

```

268 페이지 - 두 번째 코드 블록

```

self.pageController!.setViewControllers(viewControllers as <AnyObject>,
direction: .Forward,
animated: false,
completion: nil)

```

288 페이지 - 첫 번째 코드 블록

```

import UIKit
class ViewController: UIViewController {
@IBOutlet weak var textBox: UITextField!
var fileMgr: NSFileManager = NSFileManager.defaultManager() var docsDir: String?
var dataFile: String?
override func viewDidLoad() { super.viewDidLoad()
let dirPaths = NSSearchPathForDirectoriesInDomains( .DocumentDirectory, .UserDomainMask,
true)
docsDir = dirPaths[0] as? String dataFile =
docsDir?.stringByAppendingPathComponent("datafile.dat")
}
}

```

```

if fileMgr.fileExistsAtPath(dataFile!) {
let databuffer = fileMgr.contentsAtPath(dataFile!) var datastring = NSString(data: databuffer!,
encoding: NSUTF8StringEncoding)
textBox.text = datastring as! String
}
}
.
.
.

```

301 페이지 - 두 번째 코드 블록

```

override func loadFromContents(contents: AnyObject,
ofType typeName: String, error outError: NSErrorPointer) -> Bool {
if let userContent = contents as? NSData { userText = NSString(bytes: contents.bytes,
length: userContent.length,
encoding: NSUTF8StringEncoding) as? String
}
return true
}

```

302 페이지 - 두 번째 코드 블록

```

let dirPaths = NSSearchPathForDirectoriesInDomains(.DocumentDirectory, .UserDomainMask,
true)
let docsDir = dirPaths[0] as! String
let dataFile = docsDir.stringByAppendingPathComponent("savefile.txt")
documentURL = NSURL(fileURLWithPath: dataFile)
document = MyDocument(fileURL:documentURL!)
document!.userText = ""

```

303 페이지 - 두 번째 코드 블록

```

override func viewDidLoad() {
super.viewDidLoad()
let dirPaths = NSSearchPathForDirectoriesInDomains(.DocumentDirectory,
.UserDomainMask, true)
let docsDir = dirPaths[0] as! String
let dataFile = docsDir.stringByAppendingPathComponent("savefile.txt")
documentURL = NSURL(fileURLWithPath: dataFile)
document = MyDocument(fileURL: documentURL!)
document!.userText = ""

```

310 페이지 - 두 번째 코드 블록

```

let query: NSMetadataQuery = notification.object as! NSMetadataQuery
query.disableUpdates()
NSNotificationCenter.defaultCenter().removeObserver(self,

```

```
name: NSMetadataQueryDidFinishGatheringNotification,  
object: query)  
query.stopQuery()
```

311 페이지 - 두 번째 코드 블록

```
if query.resultCount == 1 {  
let resultURL = results[0].valueForAttribute(NSMetadataItemURLKey) as! NSURL  
document = MyDocument(fileURL: resultURL)  
document?.openWithCompletionHandler({(success: Bool) -> Void in if success {  
println("iCloud file open OK")  
self.textView.text = self.document?.userText  
self.ubiquityURL = resultURL  
} else {  
println("iCloud file open failed")  
}})  
} else {}
```

312 페이지 - 두 번째 코드 블록

```
func metadataQueryDidFinishGathering(notification: NSNotification) -> Void  
{  
let query: NSMetadataQuery = notification.object as! NSMetadataQuery  
query.disableUpdates()  
NSNotificationCenter.defaultCenter().removeObserver(self, name:  
NSMetadataQueryDidFinishGatheringNotification, object: query)  
query.stopQuery()  
let results = query.results  
if query.resultCount == 1 {  
let resultURL = results[0].valueForAttribute(NSMetadataItemURLKey) as! NSURL  
document = MyDocument(fileURL: resultURL)  
document?.openWithCompletionHandler({(success: Bool) -> Void in if success {  
println("iCloud file open OK")  
self.textView.text = self.document?.userText  
self.ubiquityURL = resultURL  
} else {  
println("iCloud file open failed") }  
})  
} else {  
document = MyDocument(fileURL: ubiquityURL!)  
document?.saveToURL(ubiquityURL!, forSaveOperation: .ForCreating,
```

328 페이지 - 두 번째 코드 블록

```
override func viewDidLoad() { super.viewDidLoad()  
let filemgr = NSFileManager.defaultManager()  
let dirPaths = NSSearchPathForDirectoriesInDomains(.DocumentDirectory,
```

```

.UserDomainMask, true)
let docsDir = dirPaths[0] as! String
dataFilePath = docsDir.stringByAppendingPathComponent("data.archive")
if filemgr.fileExistsAtPath(dataFilePath!) {
let dataArray = NSKeyedUnarchiver.unarchiveObjectWithFile(dataFilePath!)
as! [String]
name.text = dataArray[0]
address.text = dataArray[1]
phone.text = dataArray[2] }
}

```

343 페이지 - 첫 번째 코드 블록

```

override func viewDidLoad() { super.viewDidLoad()
let filemgr = NSFileManager.defaultManager()
let dirPaths = NSSearchPathForDirectoriesInDomains(.DocumentDirectory, .UserDomainMask,
true) let docsDir = dirPaths[0] as! String
databasePath = docsDir.stringByAppendingPathComponent("contacts.db")
if !filemgr.fileExistsAtPath(databasePath as String) {
let contactDB = FMDatabase(path: databasePath as String)
if contactDB == nil {
println("Error: ₩(contactDB.lastErrorMessage())") }
if contactDB.open() {
let sql_stmt = "CREATE TABLE IF NOT EXISTS CONTACTS (ID
INTEGER PRIMARY KEY AUTOINCREMENT, NAME TEXT, ADDRESS TEXT, PHONE TEXT)" if
contactDB.executeStatements(sql_stmt) {
println("Error: ₩(contactDB.lastErrorMessage())") }
contactDB.close() } else {
println("Error: ₩(contactDB.lastErrorMessage())") }
}
}

```

344 페이지 - 첫 번째 코드 블록

```

@IBAction func saveData(sender: AnyObject) {
let contactDB = FMDatabase(path: databasePath as String)
if contactDB.open() {
let insertSQL = "INSERT INTO CONTACTS (name, address, phone)
VALUES ('₩(name.text)', '₩(address.text)', '₩(phone.text)')"
let result = contactDB.executeUpdate(insertSQL, withArgumentsInArray: nil)
if !result {
status.text = "Failed to add contact" println("Error: ₩(contactDB.lastErrorMessage())")
} else {
status.text = "Contact Added" name.text = ""
address.text = ""
phone.text = ""
}
}
}

```

```

}
} else {
println("Error: ₩(contactDB.lastErrorMessage())") }
}

```

345 페이지 - 첫 번째 코드 블록

```

@IBAction func findContact(sender: AnyObject) {
let contactDB = FMDatabase(path: databasePath as String)
if contactDB.open() {
let querySQL = "SELECT address, phone FROM CONTACTS WHERE name = '₩(name.text)'"
let results:FMResultSet? = contactDB.executeQuery(querySQL, withArgumentsInArray: nil)
if results?.next() == true {
address.text = results?.stringForColumn("address") phone.text =
results?.stringForColumn("phone") status.text = "Record Found"
} else {
status.text = "Record not found" address.text = ""
phone.text = ""
}
contactDB.close() } else {
println("Error: ₩(contactDB.lastErrorMessage())") }
}

```

360 페이지 - 첫 번째 코드 블록

```

import UIKit
import CoreData
class ViewController: UIViewController {
let managedObjectContext = (UIApplication.sharedApplication().delegate as!
AppDelegate).managedObjectContext
@IBOutlet weak var name: UITextField!
@IBOutlet weak var address: UITextField!
@IBOutlet weak var phone: UITextField!
@IBOutlet weak var status: UILabel! 첫

```

362 페이지 - 첫 번째 코드 블록

```

@IBAction func findContact(sender: AnyObject) {
let entityDescription = NSEntityDescription.entityForName("Contacts",
inManagedObjectContext: managedObjectContext!)
let request = NSFetchRequest() request.entity = entityDescription
let pred = NSPredicate(format: "(name = %@)", name.text) request.predicate = pred
var error: NSError?
var objects = managedObjectContext?.executeFetchRequest(request, error: &error)
if let results = objects {

```



```

if results.count > 0 {
let match = results[0] as! NSManagedObject
name.text = match.valueForKey("name") as! String
address.text = match.valueForKey("address") as! String
phone.text = match.valueForKey("phone") as! String
status.text = "Matches found: ₩(results.count)"
} else {
status.text = "No Match"
}}
}

```

379 페이지 - 첫 번째 코드 블록

```

override func touchesBegan(touches: Set<NSObject>, withEvent event: UIEvent)
{ addressField.endEditing(true)
commentsField.endEditing(true)
}

```

380 페이지 - 첫 번째 코드 블록

```

func imagePickerController(picker: UIImagePickerController, didFinishPickingMediaWithInfo
info: [NSObject : AnyObject]) {
self.dismissViewControllerAnimated(true, completion: nil)
let image = info[UIImagePickerControllerOriginalImage] as! UIImage
imageView.image = image
photoURL = saveImageToFile(image);
}

```

```

func imagePickerControllerDidCancel(picker: UIImagePickerController)
{
self.dismissViewControllerAnimated(true, completion: nil)
}

```

384 페이지 - 첫 번째 코드 블록

```

} else {
if results.count > 0 {
var record = results[0] as! CKRecord
self.currentRecord = record dispatch_async(dispatch_get_main_queue()) {
self.commentsField.text = record.objectForKey("comment") as! String
let photo = record.objectForKey("photo") as! CKAsset
let image = UIImage(contentsOfFile: photo.fileURL.path!)
self.imageView.image = image
self.photoURL = self.saveImageToFile(image!) }
} else { dispatch_async(dispatch_get_main_queue()) {
self.notifyUser("No Match Found",

```

```

message: "No record matching the address was
found") }
}}
)))
}

```

392 페이지 - 첫 번째 코드 블록

```

func application(application: UIApplication, didReceiveRemoteNotification userInfo:
[NSObject : AnyObject]) {
let viewController: ViewController = self.window?.rootViewController as! ViewController
let notification: CKNotification = CKNotification(fromRemoteNotificationDictionary: userInfo)
if (notification.notificationType == CKNotificationType.Query) {
    let queryNotification = notification as! CKQueryNotification
    let recordID = queryNotification.recordID

```

393 페이지 - 두 번째 코드 블록

```

func fetchRecord(recordID: CKRecordID) -> Void {
publicDatabase = container.publicCloudDatabase
publicDatabase?.fetchRecordWithID(recordID, completionHandler: ({record, error in
if let err = error { dispatch_async(dispatch_get_main_queue()) {
self.notifyUser("Fetch Error", message: err.localizedDescription)
}
} else {
dispatch_async(dispatch_get_main_queue()) {
self.currentRecord = record
self.addressField.text = record.objectForKey("address") as! String
self.commentsField.text = record.objectForKey("comment") as!
String let photo = record.objectForKey("photo") as! CKAsset
let image = UIImage(contentsOfFile: photo.fileURL.path!) self.imageView.image = image
self.photoURL = self.saveImageToFile(image!)
}}
})) }

```

394 페이지 - 첫 번째 코드 블록

```

func application(application: UIApplication!, didFinishLaunchingWithOptions launchOptions:
NSDictionary!) -> Bool {
let settings = UIUserNotificationSettings(forTypes: .Alert | .Badge | .Sound, categories: nil)
application.registerUserNotificationSettings(settings)
application.registerForRemoteNotifications()
if let options: NSDictionary = launchOptions {
let remoteNotification = options.objectForKey(
UIApplicationLaunchOptionsRemoteNotificationKey) as? NSDictionary
if let notification = remoteNotification { self.application(application,
didReceiveRemoteNotification: notification as [NSObject : AnyObject]) }

```

```
}  
return true }
```

402 페이지 - 첫 번째 코드 블록

```
override func touchesBegan(touches: Set<NSObject>, withEvent event: UIEvent) {  
    let touchCount = touches.count  
    let touch = touches.first as! UITouch  
    let tapCount = touch.tapCount  
    methodStatus.text = "touchesBegan"  
    touchStatus.text = "₩(touchCount) touches"  
    tapStatus.text = "₩(tapCount) taps"  
}
```

403 페이지 - 첫 번째 코드 블록

```
override func touchesMoved(touches: Set<NSObject>, withEvent event: UIEvent) {  
    let touchCount = touches.count  
    let touch = touches.first as! UITouch  
    let tapCount = touch.tapCount  
    methodStatus.text = "touchesMoved"  
    touchStatus.text = "₩(touchCount) touches"  
    tapStatus.text = "₩(tapCount) taps"  
}
```

403 페이지 - 두 번째 코드 블록

```
override func touchesEnded(touches: Set<NSObject>, withEvent event: UIEvent) {  
    let touchCount = touches.count  
    let touch = touches.first as! UITouch  
    let tapCount = touch.tapCount  
    methodStatus.text = "touchesEnded"  
    touchStatus.text = "₩(touchCount) touches"  
    tapStatus.text = "₩(tapCount) taps"  
}
```

404 페이지 - 첫 번째 코드 블록

```
let touch = touches.first as! UITouch  
let point = touch.locationInView(self.view)
```

408 페이지 - 첫 번째 코드 블록

```
override func touchesBegan(touches: Set<NSObject>, withEvent event: UIEvent) {  
    let theTouch = touches.first as! UITouch  
    startPoint = theTouch.locationInView(self.view)  
    let x = startPoint!.x  
    let y = startPoint!.y  
    xCoord.text = ("x = ₩(x)")  
}
```

```
yCoord.text = ("y = ₩(y)")
}
```

408 페이지 - 두 번째 코드 블록

```
override func touchesEnd(touches: Set<NSObject>, withEvent event: UIEvent) {
    let theTouch = touches.first as! UITouch
    var endLocation = theTouch.locationInView(self.view)
    let x = touchLocation.x
    let y = touchLocation.y
    xCoord.text = ("x = ₩(x)")
    yCoord.text = ("y = ₩(y)")
}
```

409 페이지 - 첫 번째 코드 블록

```
override func touchesBegan(touches: Set<NSObject>, withEvent event: UIEvent) {
    let theTouch = touches.first as! UITouch
    startPoint = theTouch.locationInView(self.view)
    let x = startPoint!.x
    let y = startPoint!.y
    xCoord.text = ("x = ₩(x)")
    yCoord.text = ("y = ₩(y)")
}
```

445 페이지 - 두 번째 코드 블록

```
override func collectionView(collectionView: UICollectionView!, cellForItemAtIndexPath
indexPath: NSIndexPath!) ->
UICollectionViewCell! {
    let cell = collectionView.dequeueReusableCellWithReuseIdentifier(reuseIdentifier,
forIndexPath: indexPath) as! MyCollectionViewCell

    // Configure the cell
    let image = UIImage(named: carImages[indexPath.row])
    cell.imageView.image = image
    return cell
}
```

448 페이지 - 첫 번째 코드 블록

```
// MARK: UICollectionViewDelegate
override func collectionView(collectionView: UICollectionView, didSelectItemAtIndexPath
indexPath: NSIndexPath) {
    let myLayout = UICollectionViewFlowLayout()
    myLayout.scrollDirection = UICollectionViewScrollDirection.Horizontal
    self.collectionView!.setCollectionViewLayout(myLayout, animated: true)
}
```

455 페이지 - 세 번째 코드 블록

```
override func layoutAttributesForElementsInRect(rect: CGRect) -> [AnyObject] {
    let allAttributesInRect = super.layoutAttributesForElementsInRect(rect)
    for cellAttributes in allAttributesInRect! { self.modifyLayoutAttributes(cellAttributes as
    UICollectionViewLayoutAttributes)
    }
    return allAttributesInRect!
}
```

456 페이지 - 첫 번째 코드 블록

```
func modifyLayoutAttributes(layoutAttributes: UICollectionViewLayoutAttributes) {
    if layoutAttributes.indexPath == currentCellPath {
        layoutAttributes.transform3D = CATransform3DMakeScale(currentCellScale!,
        currentCellScale!, 1.0) layoutAttributes.center = currentCellCenter!
        layoutAttributes.zIndex = 1
    }
}
```

457 페이지 - 첫 번째 코드 블록

```
override func viewDidLoad() { super.viewDidLoad()
    let myLayout = MyFlowLayout()
    self.collectionView!.setCollectionViewLayout(myLayout, animated: true)
    let pinchRecognizer = UIPinchGestureRecognizer(target: self, action: "handlePinch:")
    self.collectionView!.addGestureRecognizer(pinchRecognizer)
    // Uncomment the following line to preserve selection between presentations
    // self.clearsSelectionOnViewWillAppear = false
    // Register cell classes
    // self.collectionView.registerClass(UICollectionViewCell.self, forCellWithReuseIdentifier:
    reuseIdentifier)
```

458 페이지 - 첫 번째 코드 블록

```
func handlePinch(gesture: UIPinchGestureRecognizer) {
    let layout = self.collectionView!.collectionViewLayout as! MyFlowLayout
    if gesture.state == UIGestureRecognizerState.Began {
        // 핀치의 초기 위치 얻기
        let initialPinchPoint = gesture.locationInView(self.collectionView)
        // 핀치 위치를 특정 셀로 변환하기 let pinchedCellPath =
        self.collectionView!.indexPathForItemAtPoint(initialPinchPoint)
        // indexPath 를 셀에 저장하기
        layout.currentCellPath = pinchedCellPath }
    else if gesture.state == UIGestureRecognizerState.Changed {
        // 선택된 셀의 새로운 중앙 위치 저장하기
        layout.currentCellCenter = gesture.locationInView(self.collectionView) // 스케일 값 저장하기
        layout.setCurrentCellScale(gesture.scale)
```

```

}
else {
self.collectionView!.performBatchUpdates({ layout.currentCellPath = nil layout.currentCellScale
= 1.0},
completion:nil)
}
}

```

502 페이지 - 첫 번째 코드 블록

```

override func touchesEnded(touches: Set<NSObject>, withEvent event: UIEvent) {
let touch = touches.first as? UITouch
let location = touch.locationInView(self.view)
UIView.animateWithDuration(2.0, delay: 0.0,
options: UIViewAnimationOptions.CurveEaseInOut, animations: {
let scaleTrans = CGAffineTransformMakeScale(self.scaleFactor,
self.scaleFactor) let rotateTrans = CGAffineTransformMakeRotation(
CGFloat(self.angle * M_PI / 180))
self.boxView!.transform = CGAffineTransformConcat(scaleTrans, rotateTrans)
self.angle = (self.angle == 180 ? 360 : 180) self.scaleFactor = (self.scaleFactor == 2 ? 1 : 2)
self.boxView?.center = location
}, completion: nil) }

```

502 페이지 - 두 번째 코드 블록

```

let touch = touches.first as? UITouch
let location = touch.locationInView(self.view)

```

522 페이지 - 두 번째 코드 블록

```

override func touchesBegan(touches: Set<NSObject>, withEvent event: UIEvent) {
let theTouch = touches.first as! UITouch
currentLocation = theTouch.locationInView(self.view)
attachment = UIAttachmentBehavior(item: blueBoxView!, attachedToAnchor:
currentLocation!)
animator?.addBehavior(attachment)
}

```

523 페이지 - 첫 번째 코드 블록

```

override func touchesMoved(touches: Set<NSObject>, withEvent event: UIEvent) {
let theTouch = touches.first as! UITouch
currentLocation = theTouch.locationInView(self.view)
attachment?.anchorPoint = currentLocation!
}

```

523 페이지 - 두 번째 코드 블록

```

override func touchesEnded(touches: Set<NSObject>, withEvent event: UIEvent)

```

```
{ animator?.removeBehavior(attachment)
}
```

524 페이지 - 첫 번째 코드 블록

```
override func touchesBegan(touches: Set<NSObject>, withEvent event: UIEvent) {
    let theTouch = touches.first as! UITouch
    currentLocation = theTouch.locationInView(self.view)
    let offset = UIOffsetMake(20, 20)
    attachment = UIAttachmentBehavior(item: blueBoxView!, attachedToAnchor:
    currentLocation!)
    animator?.addBehavior(attachment)
}
```

537 페이지 - 첫 번째 코드 블록

```
override func touchesBegan(touches: Set<NSObject>, withEvent event: UIEvent) {
    /* Called when a touch begins */
    for touch: AnyObject in touches {
        let location = touch.locationInView(self)
        let sprite = SKSpriteNode(imageNamed:"Spaceship")
        sprite.xScale = 0.5 sprite.yScale = 0.5 sprite.position = location
        let action = SKAction.rotateByAngle(CGFloat(M_PI), duration:1)
        sprite.runAction(SKAction.repeatActionForever(action))
        self.addChild(sprite)
    }
    .
    .
}
```

541 페이지 - 첫 번째 코드 블록

```
override func touchesBegan(touches: Set<NSObject>, withEvent event: UIEvent) {
    /* Called when a touch begins */
    let welcomeNode = childNodeWithName("welcomeNode")
    if (welcomeNode != nil) {
        let fadeAway = SKAction.fadeOutWithDuration(1.0)
        welcomeNode?.runAction(fadeAway, completion: {
            let doors = SKTransition.doorwayWithDuration(1.0)
            let archeryScene = ArcheryScene(fileName: "ArcheryScene")
            self.view?.presentScene(archeryScene, transition: doors)
        })
    }
}
```

547 페이지 - 첫 번째 코드 블록

```
override func touchesBegan(touches: Set<NSObject>, withEvent event: UIEvent) {
    let archerNode = self.childNodeWithName("archerNode")
    if archerNode != nil {
        let animate = SKAction.animateWithTextures(archerAnimation, timePerFrame: 0.05)
        archerNode?.runAction(animate)
    }
}
```

555 페이지 - 세 번째 코드 블록

```
func didBeginContact(contact: SKPhysicsContact) {
    let firstNode = contact.bodyA.node as! SKSpriteNode
    let secondNode = contact.bodyB.node as! SKSpriteNode
    if (contact.bodyA.categoryBitMask == arrowCategory) && (contact.bodyB.categoryBitMask ==
    ballCategory) {
        let contactPoint = contact.contactPoint
        let contact_y = contactPoint.y
        let target_y = secondNode.position.y
        let margin = secondNode.frame.size.height/2 - 25
        if (contact_y > (target_y - margin)) && (contact_y < (target_y + margin)) { println("Hit")
        score++ }
    }
}
```

556 페이지 - 첫 번째 코드 블록

```
func didBeginContact(contact: SKPhysicsContact) {
    let firstNode = contact.bodyA.node as! SKSpriteNode
    let secondNode = contact.bodyB.node as! SKSpriteNode
    if (contact.bodyA.categoryBitMask == arrowCategory) && (contact.bodyB.categoryBitMask ==
    ballCategory) {
        let contactPoint = contact.contactPoint
        let contact_x = contactPoint.x
        let contact_y = contactPoint.y
        let target_y = secondNode.position.y
```

569 페이지 - 두 번째 코드 블록

```
func didBeginContact(contact: SKPhysicsContact) {
    let firstNode = contact.bodyA.node as! SKSpriteNode
    let secondNode = contact.bodyB.node as! SKSpriteNode
    if (contact.bodyA.categoryBitMask == arrowCategory) && (contact.bodyB.categoryBitMask ==
    ballCategory) {
        let contactPoint = contact.contactPoint
        let contact_y = contactPoint.y
        let target_x = secondNode.position.x
        let target_y = secondNode.position.y
        let margin = secondNode.frame.size.height/2 - 25
```


570 페이지 - 첫 번째 코드 블록

```
if (contact_y > (target_y - margin))
&& (contact_y < (target_y + margin)) {
let burstPath = NSBundle.mainBundle().pathForResource("BurstParticle", ofType: "sks")
if burstPath != nil { let burstNode =
NSKeyedUnarchiver.unarchiveObjectWithFile(burstPath!) as! SKEmitterNode
burstNode.position = CGPointMake(target_x, target_y) secondNode.removeFromParent()
self.addChild(burstNode)
}
score++ }
}}
```

580 페이지 - 두 번째 코드 블록

```
override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {
let destination = segue.destinationViewController as! UIViewController
destination.interstitialPresentationPolicy = ADInterstitialPresentationPolicy.Automatic
}
```

601 페이지 - 첫 번째 코드 블록

```
override func encodeRestorableStateWithCoder(coder: NSCoder) {
coder.encodeObject(myTextView.text, forKey:"UnsavedText")
super.encodeRestorableStateWithCoder(coder)
}
override func decodeRestorableStateWithCoder(coder: NSCoder) {
myTextView.text = coder.decodeObjectForKey("UnsavedText") as! String
super.decodeRestorableStateWithCoder(coder)
}
```

609 페이지 - 두 번째 코드 블록

```
override func decodeRestorableStateWithCoder(coder: NSCoder) {
myTextView.text = coder.decodeObjectForKey("UnsavedText") as! String
super.decodeRestorableStateWithCoder(coder)
}
```

625 페이지 - 첫 번째 코드 블록

```
@IBAction func getDirections(sender: AnyObject) {
let geoCoder = CLGeocoder()
let addressString = "₩(address.text) ₩(city.text) ₩(state.text) ₩(zip.text)"
geoCoder.geocodeAddressString(addressString, completionHandler: {(placemarks:
[AnyObject]!, error: NSError!) in
if error != nil {
println("Geocode failed with error: ₩(error.localizedDescription)")
} else if placemarks.count > 0 {
```

```

let placemark = placemarks[0] as! CLPlacemark
let location = placemark.location
self.coords = location.coordinate self.showMap()
}}
}

```

638 페이지 - 첫 번째 코드 블록

```

func locationManager(manager: CLLocationManager!, didUpdateLocations locations:
[NSObject]!)
{
var latestLocation: NSObject = locations[locations.count - 1]
latitude.text = String(format: "%.4f", latestLocation.coordinate.latitude)
longitude.text = String(format: "%.4f", latestLocation.coordinate.longitude)
horizontalAccuracy.text = String(format: "%.4f", latestLocation.horizontalAccuracy)
altitude.text = String(format: "%.4f", latestLocation.altitude)
verticalAccuracy.text = String(format: "%.4f", latestLocation.verticalAccuracy)
if startLocation == nil {
startLocation = latestLocation as! CLLocation
}
var distanceBetween: CLLocationDistance =
latestLocation.distanceFromLocation(startLocation)
distance.text = String(format: "%.2f", distanceBetween)
}

```

656 페이지 - 첫 번째 코드 블록

```

let search = MKLocalSearch(request: request)
search.startWithCompletionHandler({(response: MKLocalSearchResponse!,
error: NSError!) in
if error != nil {
println("Error occurred in search: ₩(error.localizedDescription)")
} else if response.mapItems.count == 0 {
println("No matches found")
} else {
println("Matches found")
for item in response.mapItems as! [MKMapItem] {
println("Name = ₩(item.name)")
println("Phone = ₩(item.phoneNumber)")
self.matchingItems.append(item as MKMapItem)
println("Matching items = ₩(self.matchingItems.count)")
var annotation = MKPointAnnotation()
annotation.coordinate = item.placemark.coordinate
annotation.title = item.name
self.mapView.addAnnotation(annotation)
}}
}

```

```
}} }
```

656 페이지 - 두 번째 코드 블록

```
func showRoute(response: MKDirectionsResponse) {  
  for route in response.routes as! [MKRoute] {  
    routeMap.addOverlay(route.polyline, level: MKOverlayLevel.AboveRoads)  
  }  
}
```

663 페이지 - 두 번째 코드 블록

```
override func numberOfSectionsInTableView(tableView: UITableView) -> Int {  
  // Return the number of sections.  
  return 1  
}
```

```
override func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int {  
  // Return the number of rows in the section.  
  return mapItems.count  
}
```

```
override func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath:  
  NSIndexPath) -> UITableViewCell {  
  let cell = tableView.dequeueReusableCellWithIdentifier( "resultCell", forIndexPath: indexPath)  
  as! ResultsTableViewCell  
  // Configure the cell...  
  let row = indexPath.row  
  let item = mapItems[row] cell.nameLabel.text = item.name  
}
```

664 페이지 - 두 번째 코드 블록

```
override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {  
  let destination = segue.destinationViewController as! ResultsTableViewController  
  destination.mapItems = self.matchingItems  
}
```

667 페이지 - 두 번째 코드 블록

```
func showRoute(response: MKDirectionsResponse) {  
  for route in response.routes as! [MKRoute] {  
    routeMap.addOverlay(route.polyline, level: MKOverlayLevel.AboveRoads)  
    for step in route.steps {  
      println(step.instructions)  
    }  
  }  
  let userLocation = routeMap.userLocation
```

```

let region = MKCoordinateRegionMakeWithDistance(userLocation.location.coordinate, 2000,
2000)
routeMap.setRegion(region, animated: true)
}
func mapView(mapView: MKMapView!, rendererForOverlay

```

668 페이지 - 두 번째 코드 블록

```

override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {
let routeViewController = segue.destinationViewController as! RouteViewController
let indexPath = self.tableView.indexPathForSelectedRow()
let row = indexPath?.row
routeViewController.destination = mapItems![row!]
}

```

711 페이지 - 첫 번째 코드 블록

```

let textItem: NSAttributedString = extensionContext?.inputItems[0] as! NSAttributedString

```

711 페이지 - 두 번째 코드 블록

```

let textItem: NSAttributedString = extensionContext?.inputItems[0] as! NSAttributedString
let textItemProvider = textItem.attachments![0] as! NSAttributedStringProvider
if textItemProvider.hasItemConformingToTypeID(kUTTypeText as String) {
}

```

712 페이지 - 첫 번째 코드 블록

```

override func viewDidLoad() { super.viewDidLoad()
let textItem: NSAttributedString = extensionContext?.inputItems[0] as! NSAttributedString
let textItemProvider = textItem.attachments![0] as! NSAttributedStringProvider
if textItemProvider.hasItemConformingToTypeID(kUTTypeText as String) {
textItemProvider.loadItemForTypeID(kUTTypeText as String,
options: nil,
completionHandler: handleCompletion)
}}

```

713 페이지 - 첫 번째 코드 블록

```

@IBAction func done() {
var returnProvider = NSAttributedStringProvider(item: convertedString, typeId: kUTTypeText as
String)
var returnItem = NSAttributedString()
returnItem.attachments = [returnProvider]
extensionContext!.completeRequestReturningItems(
[returnItem], completionHandler: nil)
}

```

718 페이지 - 83.1 절 첫 번째 문단 마지막 줄

Product Name 은 **ActionHostApp** 이라고 하자.

722 페이지 - 첫 번째 코드 블록

```
func myHandler(activityType:String!, completed: Bool, returnedItems: [AnyObject]!, error:
NSError!) {
if returnedItems.count > 0 {
let textItem: NSEXTensionItem = returnedItems[0] as! NSEXTensionItem
let textItemProvider: AnyObject = textItem.attachments![0]
if textItemProvider.hasItemConformingToTypeIdentifier( kUTTypeText as String) {
textItemProvider.loadItemForTypeIdentifier( kUTTypeText as String,
options: nil,
completionHandler: {(string:NSSecureCoding!,
error: NSError!) -> Void in let newText = string as! String
self.myTextView.text = newText
}}
}
}
..}
```

749 페이지 - 첫 번째 코드 블록

```
func imagePickerController(picker: UIImagePickerController, didFinishPickingMediaWithInfo
info: [NSObject : AnyObject]) {
let mediaType = info[UIImagePickerControllerMediaType] as! NSString
self.dismissViewControllerAnimated(true, completion: nil)
if mediaType.isEqualToString(kUTTypeImage as String) {
let image = info[UIImagePickerControllerOriginalImage] as! UIImage
imageView.image = image
if (newMedia == true) { UIImagePickerControllerWriteToSavedPhotosAlbum(image, self,
"image:didFinishSavingWithError:contextInfo:", nil)
} else if mediaType.isEqualToString(kUTTypeMovie as String) {
// 비디오를 지원하기 위한 코드가 옴
}}
}}
```

754 페이지 - 첫 번째 코드 블록

```
import UIKit
import AVKit
import AVFoundation
class ViewController: UIViewController {
.
.
override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {
let destination = segue.destinationViewController as! AVPlayerViewController
let url = NSURL(string: "http://www.ebookfrenzy.com/ios_book/movie/movie.mov")
destination.player = AVPlayer(URL: url)
```

```
}  
.  
.
```

766 페이지 - 두 번째 코드 블록

```
override func viewDidLoad() {  
    super.viewDidLoad() playButton.enabled = false stopButton.enabled = false  
    let dirPaths = NSSearchPathForDirectoriesInDomains(.DocumentDirectory,  
    .UserDomainMask, true)  
    let docsDir = dirPaths[0] as! String  
    let soundFilePath = docsDir.stringByAppendingPathComponent("sound.caf")  
}
```

767 페이지 - 첫 번째 코드 블록

```
let soundFileURL = NSURL(fileURLWithPath: soundFilePath) let recordSettings =  
[AVEncoderAudioQualityKey: AVAudioQuality.Min.rawValue, AVEncoderBitRateKey: 16,  
AVNumberOfChannelsKey: 2,  
AVSampleRateKey: 44100.0]  
var error: NSError?  
let audioSession = AVAudioSession.sharedInstance()  
audioSession.setCategory(AVAudioSessionCategoryPlayAndRecord, error: &error)  
if let err = error {  
    println("audioSession error: ₩(err.localizedDescription)")  
}  
audioRecorder = AVAudioRecorder(URL: soundFileURL,  
settings: recordSettings as [NSObject : AnyObject] , error: &error)  
if let err = error {  
    println("audioSession error: ₩(err.localizedDescription)")  
} else { audioRecorder?.prepareToRecord()  
}  
}
```

780 페이지 - 첫 번째 코드 블록

```
func imagePickerController(picker: UIImagePickerController, didFinishPickingMediaWithInfo  
info: [NSObject : AnyObject]) {  
    self.dismissViewControllerAnimated(true, completion: nil)  
    let image = info[UIImagePickerControllerOriginalImage] as! UIImage  
    postImage.image = image  
}  
func imagePickerControllerDidCancel(picker: UIImagePickerController) {  
    self.dismissViewControllerAnimated(true, completion: nil)  
}
```

780 페이지 - 두 번째 코드 블록

```
let account = ACAccountStore()
```

```

let accountType =
account.accountTypeWithAccountTypeIdentifier(ACAccountTypeIdentifierTwitter)
account.requestAccessToAccountsWithType(accountType, options: nil, completion: {(success:
Bool, error: NSError!) -> Void in
if success {
let arrayOfAccounts = account.accountsWithAccountType(accountType)
if arrayOfAccounts.count > 0 {
let twitterAccount = arrayOfAccounts.last as! ACAccount
}
}})

```

785 페이지 - 첫 번째 코드 블록

```

override func touchesBegan(touches: Set<NSObject>, withEvent event: UIEvent) {
postText.endEditing(true)
}

```

786 페이지 - 첫 번째 코드 블록

```

completion: {(success: Bool, error: NSError!) -> Void in
if success {
let arrayOfAccounts = account.accountsWithAccountType(accountType)
if arrayOfAccounts.count > 0 {
let twitterAccount = arrayOfAccounts.last as! ACAccount
let message = ["status" : "My first post from iOS 8"] let requestURL = NSURL(string:
"https://api.twitter.com/1.1/statuses/update.json")
let postRequest = SLRequest(forServiceType: SLServiceTypeTwitter,
requestMethod: SLRequestMethod.POST, URL: requestURL,
parameters: message)
}}
})

```

786 페이지 - 두 번째 코드 블록

```

let account = ACAccountStore()
let accountType = account.accountTypeWithAccountTypeIdentifier(
ACAccountTypeIdentifierTwitter)
account.requestAccessToAccountsWithType(accountType, options: nil, completion: {(success:
Bool, error: NSError!) -> Void in
if success {
let arrayOfAccounts = account.accountsWithAccountType(accountType)
if arrayOfAccounts.count > 0 {
let twitterAccount = arrayOfAccounts.last as! ACAccount
let message = ["status" : "My first post from iOS 8"]
let requestURL = NSURL(string: "https://api.twitter.com/1.1/statuses/update.json")
let postRequest = SLRequest(forServiceType:
SLServiceTypeTwitter,

```

```
requestMethod: SLRequestMethod.POST, URL: requestURL,  
parameters: message)  
postRequest.account = twitterAccount  
postRequest.performRequestWithHandler({ (responseData: NSData!,
```

793 페이지 - 두 번째 코드 블록

```
func getTimeLine() {  
let account = ACAccountStore()  
let accountType = account.accountTypeWithAccountTypeIdentifier(  
ACAccountTypeIdentifierTwitter)  
account.requestAccessToAccountsWithType(accountType, options: nil, completion: {(success:  
Bool, error: NSError!) -> Void in  
if success {  
let arrayOfAccounts = account.accountsWithAccountType(accountType)  
if arrayOfAccounts.count > 0 {  
let twitterAccount = arrayOfAccounts.last as! ACAccount  
let requestURL = NSURL(string: "https://api.twitter.com/1.1/statuses/user_timeline.json")  
let parameters = ["screen_name" : "@techotopia", "include_rts" : "0",  
"trim_user" : "1",  
"count" : "20"]  
let postRequest = SLRequest(forServiceType: SLServiceTypeTwitter,  
requestMethod: SLRequestMethod.GET, URL: requestURL,  
parameters: parameters)  
postRequest.account = twitterAccount  
postRequest.performRequestWithHandler( {(responseData: NSData!,
```

794 페이지 - 첫 번째 코드 블록

```
urlResponse: NSHTTPURLResponse!,  
error: NSError!) -> Void in var err: NSError?  
self.dataSource = NSJSONSerialization.JSONObjectWithData(responseData, options:  
NSJSONReadingOptions.MutableLeaves, error: &err) as! [AnyObject]  
if self.dataSource.count != 0 { dispatch_async(dispatch_get_main_queue()) {  
self.tweetTableView.reloadData() }  
}})  
}  
} else {  
println("Failed to access account") }  
}}}
```

795 페이지 - 첫 번째 코드 블록

```
postRequest.performRequestWithHandler( {(responseData: NSData!,  
urlResponse: NSHTTPURLResponse!, error: NSError!) -> Void in  
var err: NSError? self.dataSource =  
NSJSONSerialization.JSONObjectWithData(responseData, options:
```



```

NSJSONReadingOptions.MutableLeaves,
error: &err) as! [AnyObject]
if self.dataSource.count != 0 {
dispatch_async(dispatch_get_main_queue()) { self.tweetTableView.reloadData()
}}
})

```

795 페이지 - 세 번째 코드 블록

```

func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) ->
UITableViewCell {
let cell = self.tweetTableView.dequeueReusableCellWithIdentifier("Cell")
as! UITableViewCell let row = indexPath.row
let tweet = self.dataSource[row] as! NSDictionary
cell.textLabel?.text = tweet objectForKey("text") as? NSString
cell.textLabel?.numberOfLines = 0
return cell
}

```

802 페이지 - 두 번째 코드 블록

```

import UIKit
import StoreKit
class ViewController: UIViewController {
@IBOutlet weak var level2Button: UIButton!
override func viewDidLoad() {
super.viewDidLoad()
let appDelegate = UIApplication.sharedApplication().delegate
as! AppDelegate
appDelegate.homeViewController = self
}
.
.
.

```

809 페이지 - 첫 번째 코드 블록

```

func paymentQueue(queue: SKPaymentQueue!, updatedDownloads downloads:
[AnyObject]!) {
for download in downloads as! [SKDownload] {
switch download.downloadState {
case SKDownloadState.Active:
println("Download progress ₩(download.progress)")
println("Download time = ₩(download.timeRemaining)")

```

```

break
case SKDownloadState.Finished:
// 다운로드 완료. 콘텐츠 파일 URL 은
// download.contentURL 로 참조된 경로에 있다 // 그것을 다른 안전한 곳으로 옮겨서 압축을 풀고
// 사용자가 접근할 수 있도록 하자
break
default: break
}
}
}

```

822 페이지 - 두 번째 코드 블록

```

func getProductInfo()
{
if SKPaymentQueue.canMakePayments() {
let request = SKProductsRequest(productIdentifiers:
NSSet(objects: self.productID) as Set<NSObject>)
request.delegate = self
request.start()
} else {
productDescription.text = "Please enable In App Purchase in Settings"
}
}
}

```

823 페이지 - 세 번째 코드 블록

```

func paymentQueue(queue: SKPaymentQueue!, updatedTransactions transactions:
[NSObject]!) {
for transaction in transactions as! [SKPaymentTransaction] {
switch transaction.transactionState {
case SKPaymentTransactionState.Purchased:
self.unlockFeature()
SKPaymentQueue.defaultQueue().finishTransaction(transaction)
case SKPaymentTransactionState.Failed:
SKPaymentQueue.defaultQueue().finishTransaction(transaction)
default:
break
}
}
}
}

```

824 페이지 - 첫 번째 코드 블록

```

func unlockFeature() {
let appDelegate = UIApplication.sharedApplication().delegate as! AppDelegate

```

```
appdelegate.homeViewController!.enableLevel2()  
buyButton.enabled = false  
productTitle.text = "Item has been purchased"  
}
```