iOS 프로그래밍

파운데이션과 앱 만들기

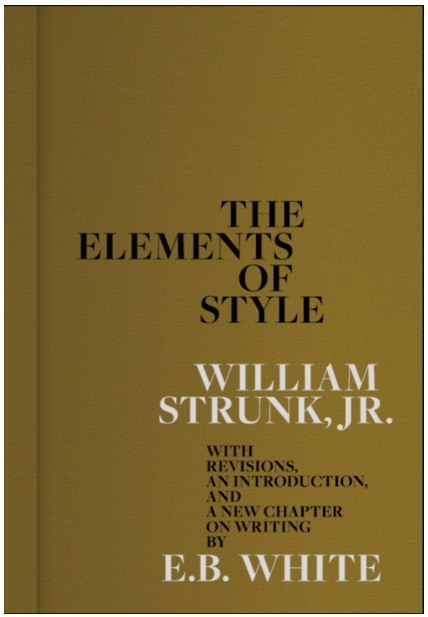
코드스쿼드

학습 목표

* 기본 앱 Outlet/Action 구현하기
* 표준 라이브러리 활용하기
* 파운데이션 프레임워크 활용하기
* iOS 앱 템플릿
* 뷰컨트롤러 단위로 생각하기

Naming Guide

- API 설계할 때 중요한 포인트
- 단지 간단하거나 너저분하지 않으면서
- 명쾌하도록 노력할 것
- 필수적인 정보를 담고
- 불필요하거나 부가적인 정보는 빼자



https://swift.org/documentation/api-design-guidelines/

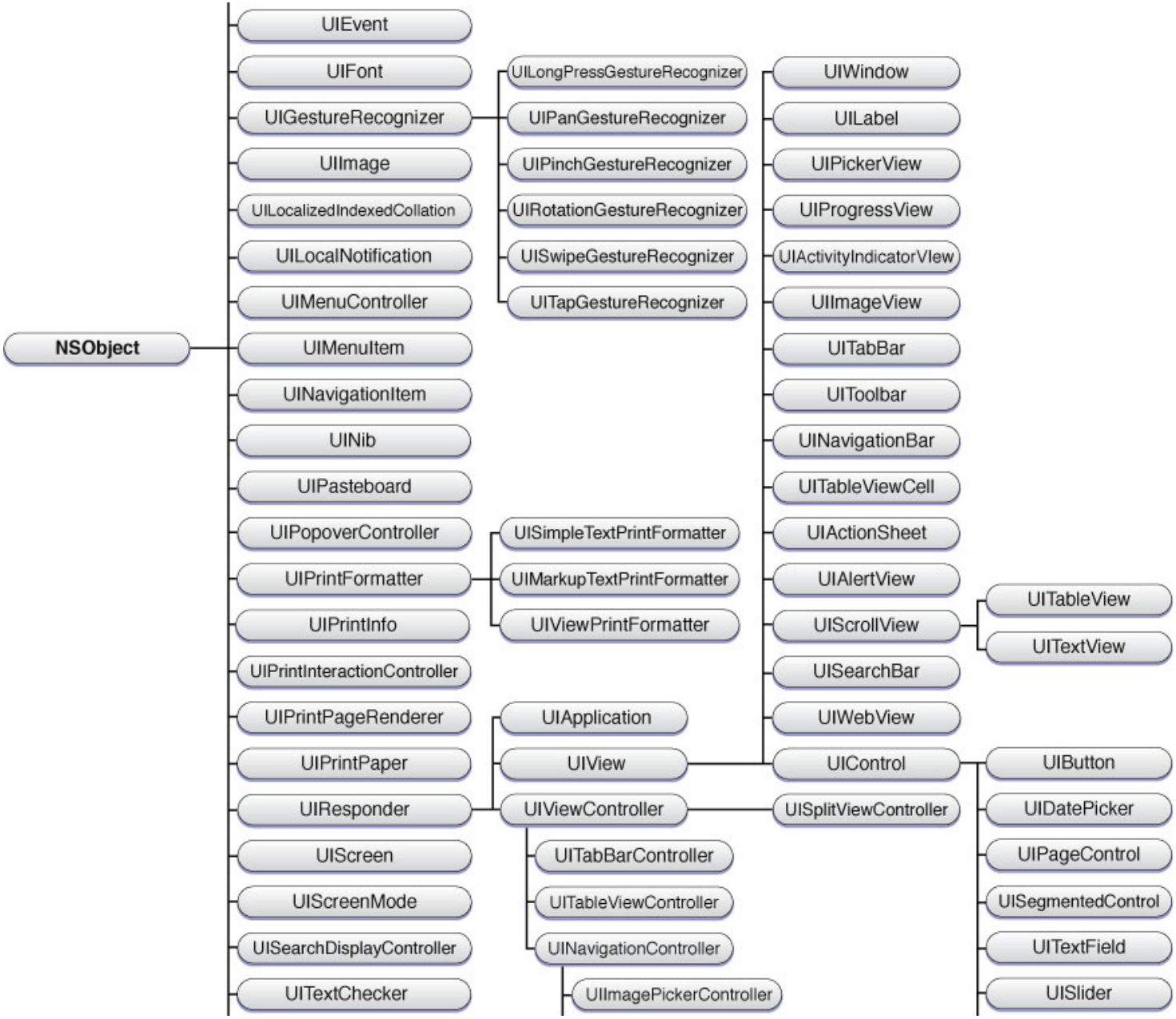
https://gist.github.com/godrm/d07ae33973bf71c5324058406dfe42dd

표준 라이브러리

Standard Library

기능 단위 묶음

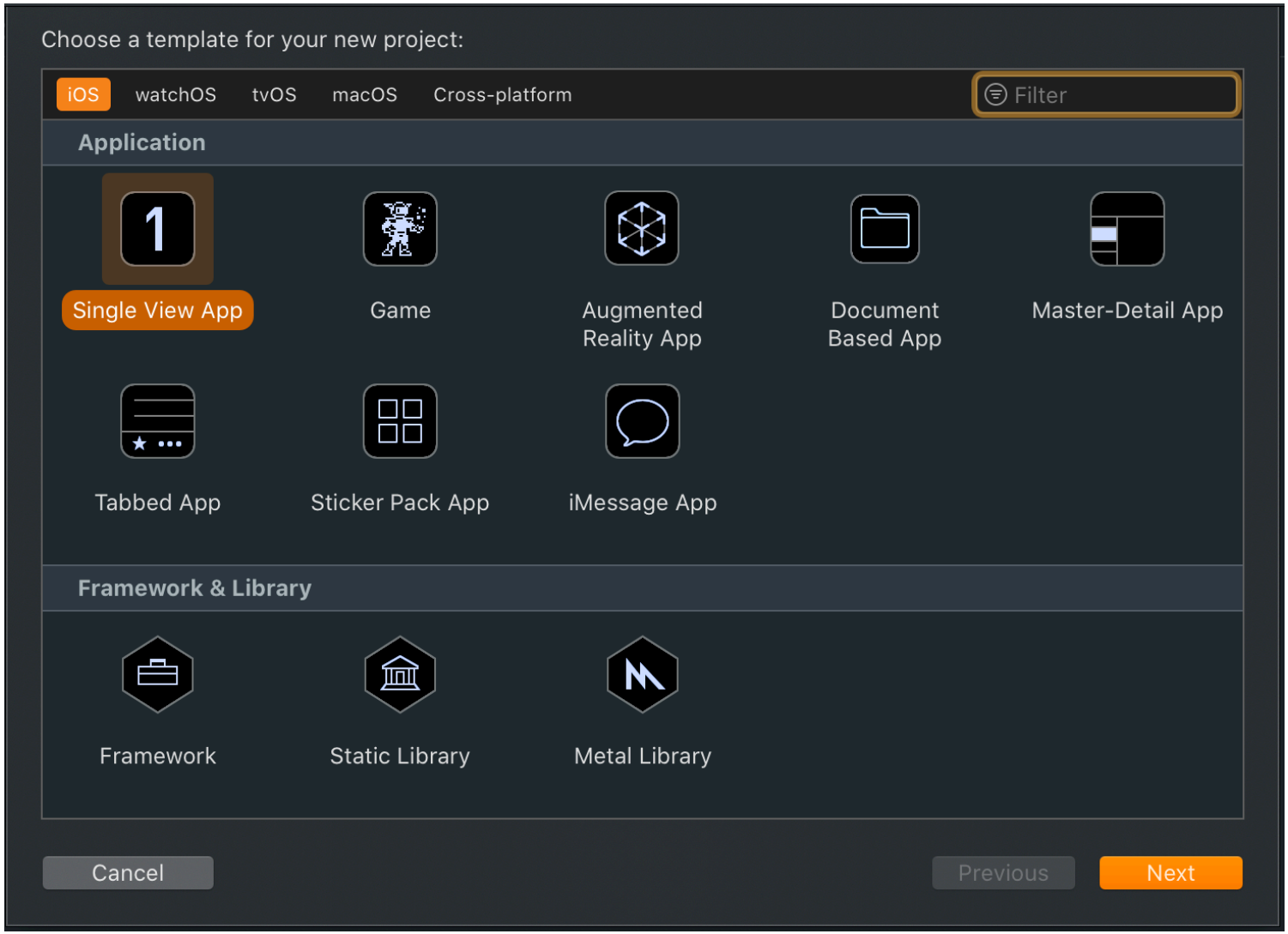
https://developer.apple.com/reference/swift

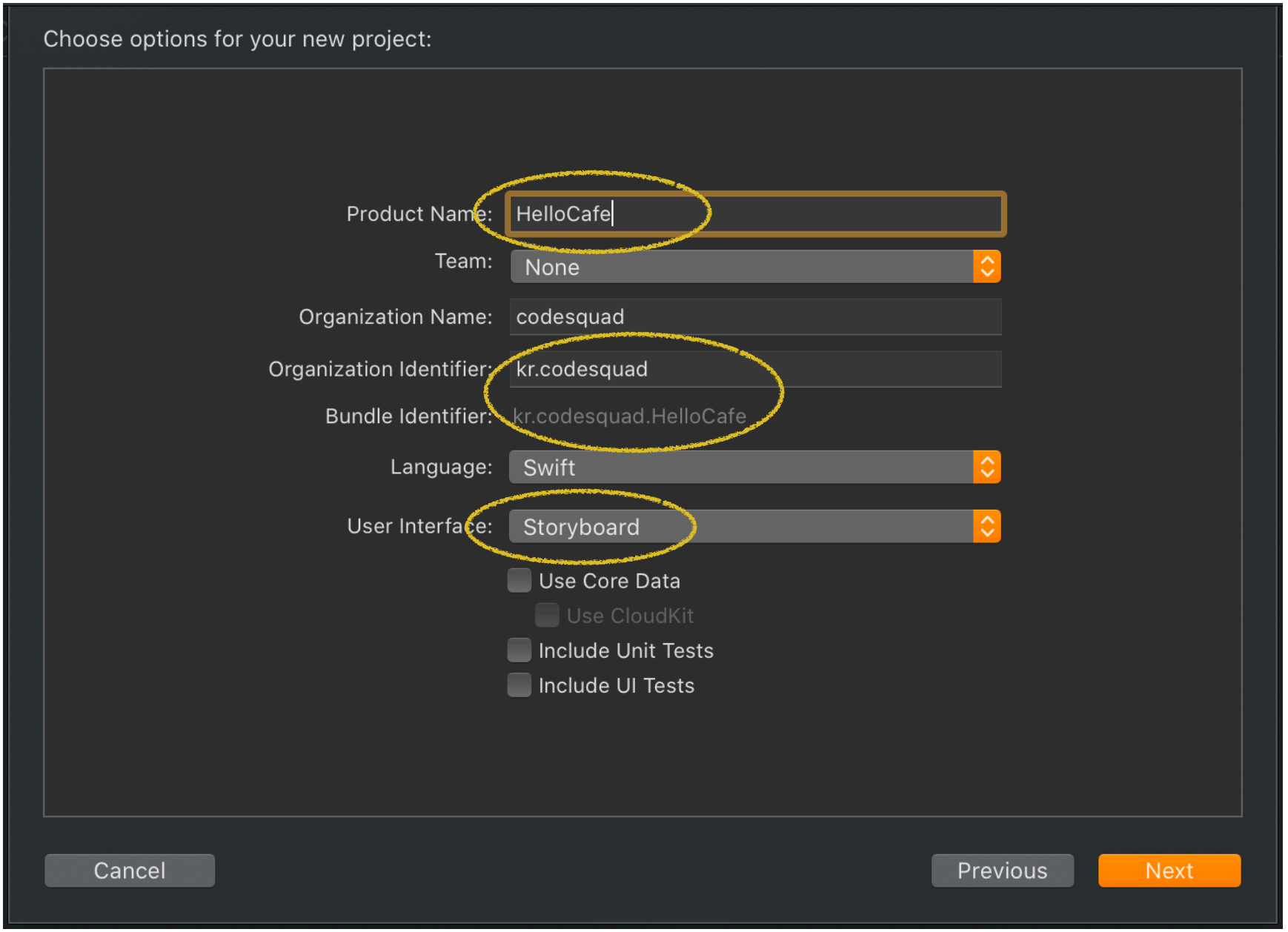

NSObject
UIEvent
UIFont
UIGestureRecognizer
UILongPressGestureRecognizer
UIPanGestureRecognizer
UIPinchGestureRecognizer
UIRotationGestureRecognizer
UISwipeGestureRecognizer
UITapGestureRecognizer
UIImage
UILocalizedIndexedCollation
UILocalNotification
UIMenuController
UIMenuItem
UINavigationItem
UINib
UIPasteboard
UIPopoverController
UIPrintFormatter
UISimpleTextPrintFormatter
UIMarkupTextPrintFormatter
UIViewPrintFormatter
UIPrintInfo
UIPrintInteractionController
UIPrintPageRenderer
UIPrintPaper
UIResponder
UIApplication
UIView
UIViewController
UITabBarController
UITableViewController
UINavigationController
UIImagePickerController
UIScreen
UIScreenMode
UISearchDisplayController
UITextChecker
UIWindow
UILabel
UIPickerView
UIProgressView
UIActivityIndicatorView
UIImageView
UITabBar
UIToolbar
UINavigationBar
UITableViewCell
UIActionSheet
UIAlertView
UIScrollView
UITableView
UITextView
UISearchBar
UIWebView
UIControl
UIButton
UIDatePicker
UIPageControl
UISegmentedControl
UITextField
UISlider
UISplitViewController

일단 만들어봅시다!

드디어 앱

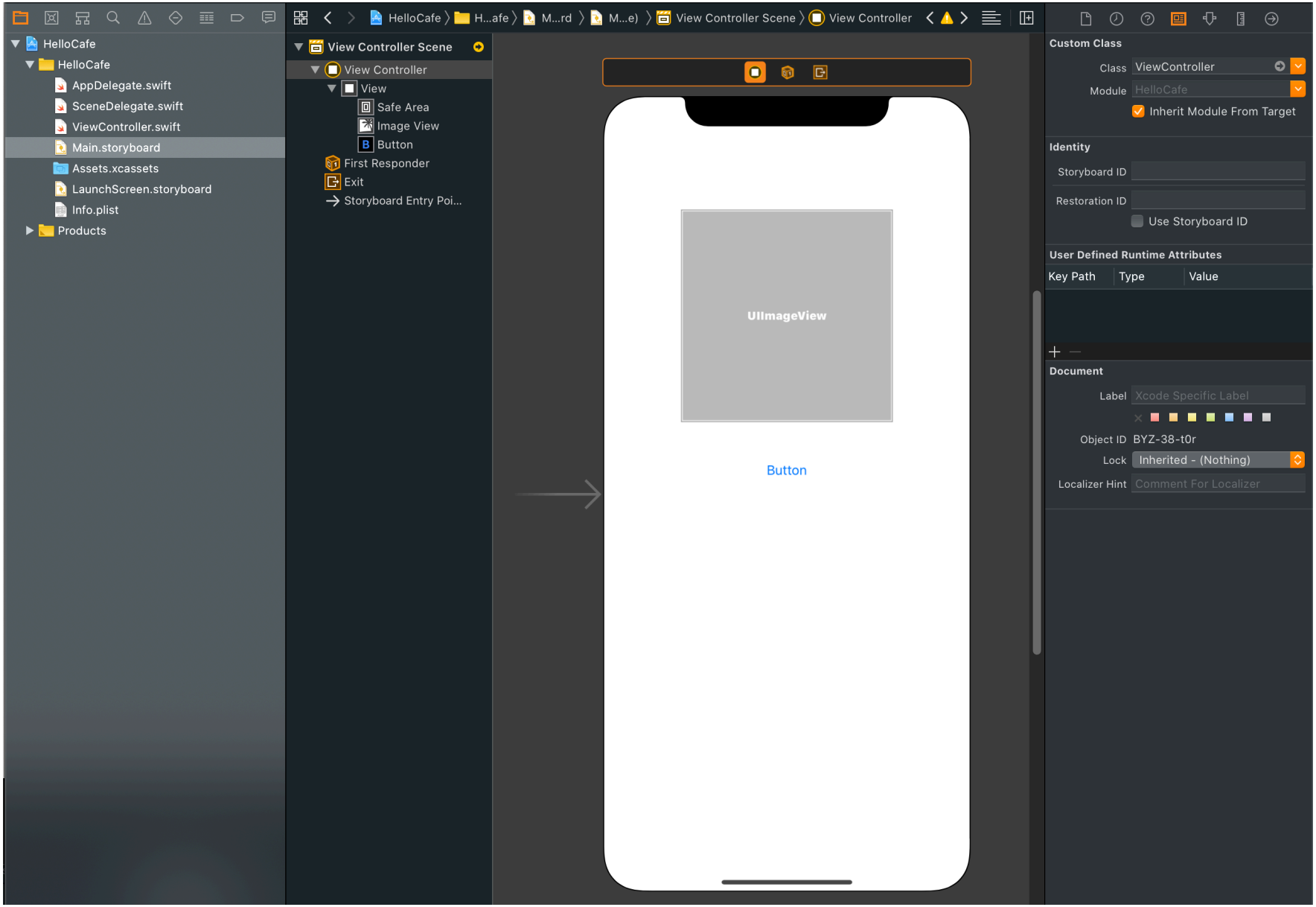
File > New > Project




Choose options for your new project:
Product Name:
HelloCafe
Team:
None
Organization Name:
codesquad
Organization Identifier:
kr.codesquad
Bundle Identifier:
kr.codesquad.HelloCafe
Language:
Swift
User Interface:
Storyboard
Use Core Data
Use CloudKit
Include Unit Tests
Include UI Tests
Cancel
Previous
Next

데모

앱 만들기


HelloCafe
HelloCafe
AppDelegate.swift
SceneDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
HelloCafe 〉 H...afe 〉 M...rd 〉 M...e) 〉 View Controller Scene 〉 View Controller
View Controller Scene
View Controller
View
Safe Area
Image View
Button
First Responder
Exit
Storyboard Entry Poi...
UIImageView
Button
Custom Class
Class ViewController
Module HelloCafe
Inherit Module From Target
Identity
Storyboard ID
Restoration ID
Use Storyboard ID
User Defined Runtime Attributes
Key Path
Type
Value
Document
Label
Xcode Specific Label
Object ID BYZ-38-t0r
Lock Inherited - (Nothing)
Localizer Hint
Comment For Localizer

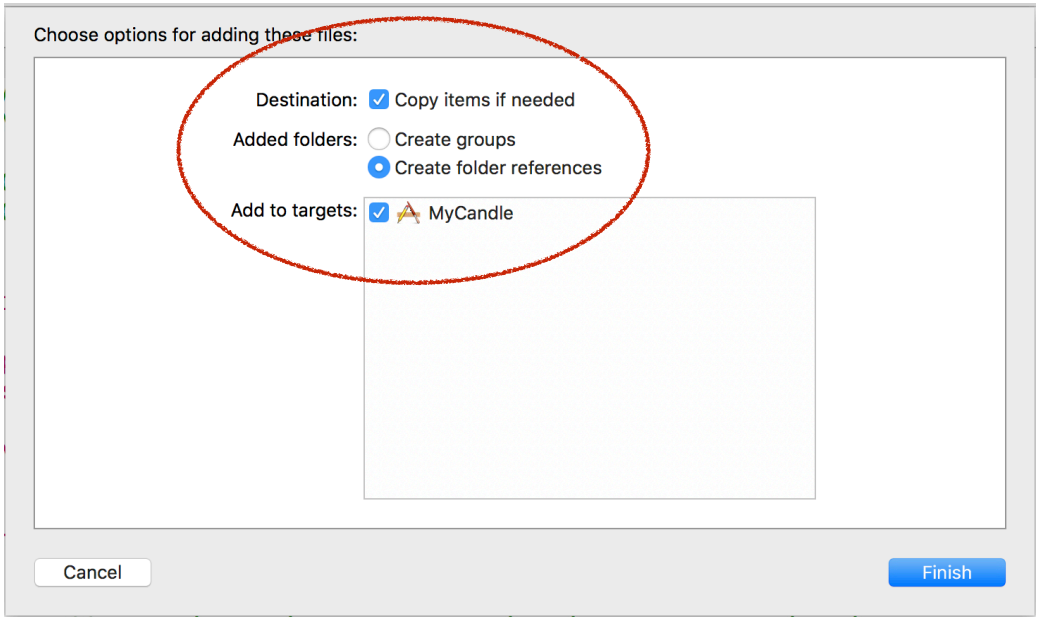
IBOutlet
vs.
IBAction

다음 두 이미지 파일을 받으세요

http://public.codesquad.kr/jk/Swift/cafe-latte.jpg

http://public.codesquad.kr/jk/Swift/green-tea.jpg

파인더에서 두 파일을 선택하고
Xcode 프로젝트 화면으로 드래그 & 드롭 하세요

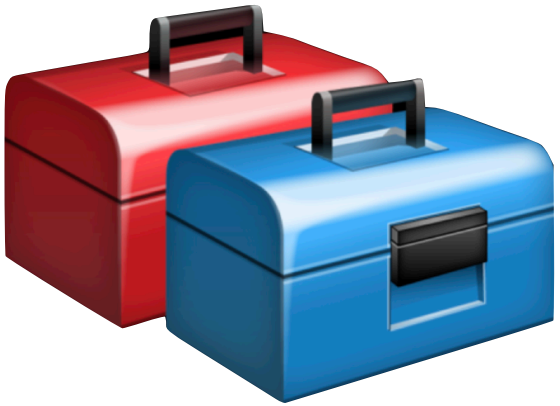


Foundation

Array, String,
Dictionary, Set, Date

프레임워크 Frameworks

Foundation



AppKit/UIKit

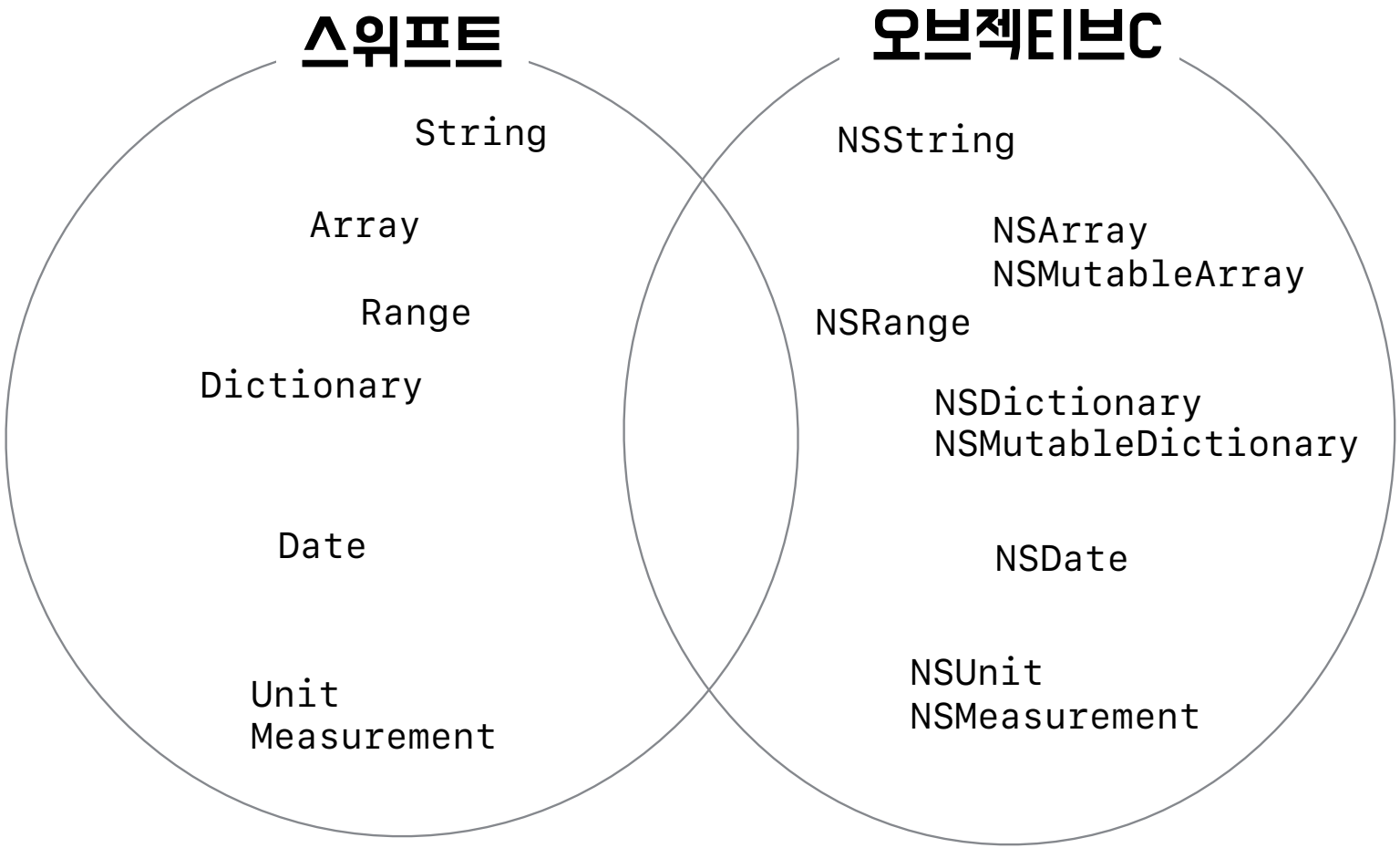
Cocoa Touch

아이폰OS에서 최상위 레벨의 레이어

UIKit Framework 화면표시 담당

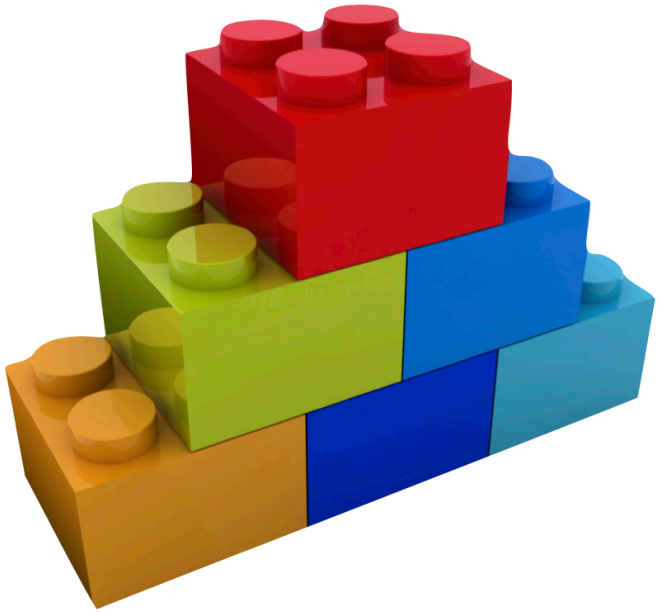
Foundation는 큰 수정 없이 아이폰에 적용

Cocoa	Cocoa Touch
AppKit NSView NSTableView NSButton	→ UIKit UIView UITableView UIButton
Foundation NSString NSThread NSArray	→ Foundation NSString NSThread NSArray



https://developer.apple.com/documentation/foundation

블록 쌓기

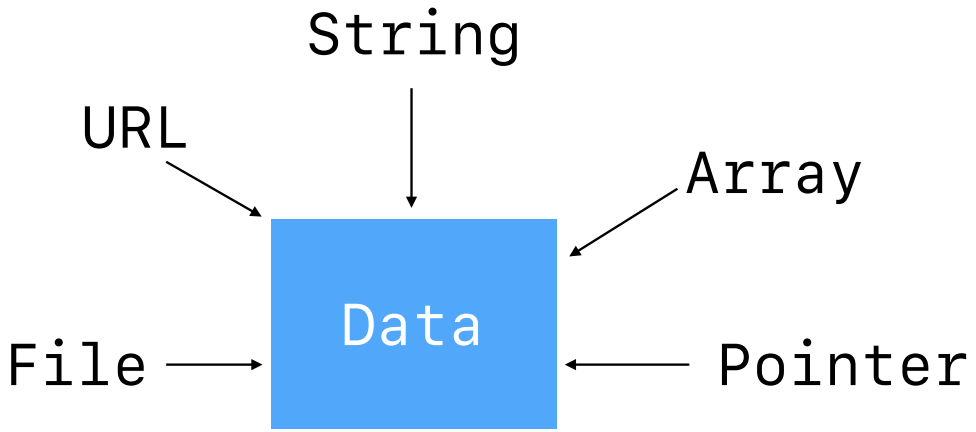


- 문자열
- 콜렉션
- 날짜와 시간
- 영구저장소와 아카이브
- 파일과 URL
- 번들
- OperationQueue

여러 가지 데이터

바이너리 데이터

Data



유용한 타입

- 범위 구조체 : Range

```
let underFive = 0.0..<5.0
```

```
print(underFive.contains(3.14))     // Prints "true"
print(underFive.contains(6.28))     // Prints "false"
print(underFive.contains(5.0))      // Prints "false"
```

```
let empty = 0.0..<0.0
print(empty.contains(0.0))          // Prints "false"
print(empty.isEmpty)                // Prints "true"
```

기하학 관련 타입

- CGPoint, CGSize

```
struct CGPoint {
  var x : CGFloat
  var y : CGFloat
}
```

```
struct CGSize {
  var width : CGFloat
  var height : CGFloat
}
```

```
struct CGRect {
  var origin : CGPoint
  var size : CGSize
}
```

문자열 Strings

Foundation

Strings

```
NSString* myString = @"This is a String";
NSString* myString = [NSString stringWithFormat:"techcamp-%d",2017];
```

```
let myString = "This is a String"
let campString = "techcamp-"+"2007"


let pathString = "Library/Caches/Images/dataCache.temp"
let pathComponents = pathString.components(separatedBy: "/")
let fileName = pathString.components(separatedBy: "/").last

let fileContents = try? String.init(contentsOfFile: pathString)
```

Strings https://developer.apple.com/documentation/swift/string

* 대부분의 텍스트 정보를 보관
  * String / NSString, NSMutableString
* 유니코드 문자열의 배열 형태
* 불투명한 컨테이너 (내부가 보이지 않음)
* 일반적인 목적
  * 비교 compare
  * 검색 range, contain
  * 인코딩 변환 encode

Search Strings

```
let name = "Marie Curie"
let firstSpace = name.index(of: " ") ?? name.endIndex
let firstName = name[..<firstSpace]
// firstName == "Marie"
```

```
let sanjose = "San Jose\u{301}"
let notfound = sanjose.range(of: "Jose")
//not found
let found = sanjose.range(of: "José")
//4..<9
```

Character	S	a	n		J	o	s	é	´
Location	0	1	2	3	4	5	6	7	8

A View of String

```
let cafe = "Cafe\u{301} du 🌍"
print(cafe)
// Prints "Café du 🌍"
print(cafe.count)
// Prints "9"
print(Array(cafe))
// Prints "["C", "a", "f", "é", " ", "d", "u", " ", "🌍"]"
```

Unicode Scalar View

```
print(cafe.unicodeScalars.count)
// Prints "10"
print(Array(cafe.unicodeScalars))
// Prints "["C", "a", "f", "e", "\u{0301}", " ", "d", "u", " ", "\u{0001F30D}"]"
print(cafe.unicodeScalars.map { $0.value })
// Prints "[67, 97, 102, 101, 769, 32, 100, 117, 32, 127757]"
```

UTF-8 View

```
print(cafe.utf8.count)
// Prints "14"
print(Array(cafe.utf8))
// Prints "[67, 97, 102, 101, 204, 129, 32, 100, 117, 32, 240, 159, 140, 141]"
```

콜렉션 Collections

콜랙션

객체들을 보관해 놓는 곳

* NSArray, NSDictionary, NSSet ...
* NSMutableArray, NSMutableDictionary, NSMutableSet ...

일반적으로 많이 사용

* Iterating
* Sorting
* Filtering

Foundation

Arrays

```
NSMutableArray* arr = [NSMutableArray array];
[arr addObject:@"randomString"];
```

```
NSArray* arr = [NSArray arrayWithObjects:@"foo",@"bar",@"codd",nil];
NSString* complexString = [arr componetsJoinedByString:@"--"];
```

```
"foo--bar--codd"
```

```
let arr = ["foo", "bar", "codd"]
let complexString = arr.joined(separator: “--”)

var mutableArray = NSMutableArray()
mutableArray.addObjects(from: ["foo", "bar", "codd", "scott"])
```

Foundation

Dictionaries

```
NSMutableDictionary* md = [NSMutableDictionary dictionary];
[md setObject:@“value" forKey:@“key"];

NSDictionary* dict = @{ @"key" : @"value", @"key2" : @"value2" };
for (NSString* key in dict.allKeys) {
    NSLog(dict[key]);
}
```

```
let dict = [ "key" : "value", "key2" : "value2" ]
for (key, value) in dict {
    print(key,value)
}

var mutableDictionary = NSMutableDictionary()
mutableDictionary.setObject(10, forKey: "age" as NSString)
mutableDictionary.setObject("hana", forKey: "name" as NSString)
```

날짜 Date 시간 Time

날짜와 시간

날짜와 시간 계산의 복잡한 내용을 구현

사용자 설정(타임존)에 맞춰서 자동 조절

활용 예시

* 달력에 날짜를 표시
* 특정 날짜 사이의 시간 계산
* 특정 로케일에서 사용자 설정에 맞는 날짜/숫자 포맷

왜 Date & Time을 쓰나

특정 요일부터 12시간 뒤를 표시하고 싶을 때

오늘날짜	원하는 것	실제 값	차이점
2017년 7월 5일 2:00 PM	2017년 7월 6일 2:00 AM	2017년 7월 5일 2:00 PM	OK
2017년 10월 5일 2:00 PM	2017년 10월 6일 2:00 AM	2017년 10월 6일 1:00 AM	썸머타임 끝났음!
2017년 10월 5일 2:00 PM	2017년 10월 5일 1:00 AM	2017년 10월 5일 2:00 AM	중국 상하이!

Date & Time Classes

NSDate

캘린더와 타임존에 독립적으로

절대 시간을 기준으로 계산

특정 시각을 기준으로 Double 형으로 측정한 시간

Date & Time Classes

NSCalendar

년-단위 시작beginning, 길이length, 경계division를 정의

예) 그레고리안, 히브리 달력

Date & Time Classes

NSDateComponents

특정 캘린더 속성을 반영하는

년도, 달, 일 등을 포함하는

간단한 구조체

NSFormatter

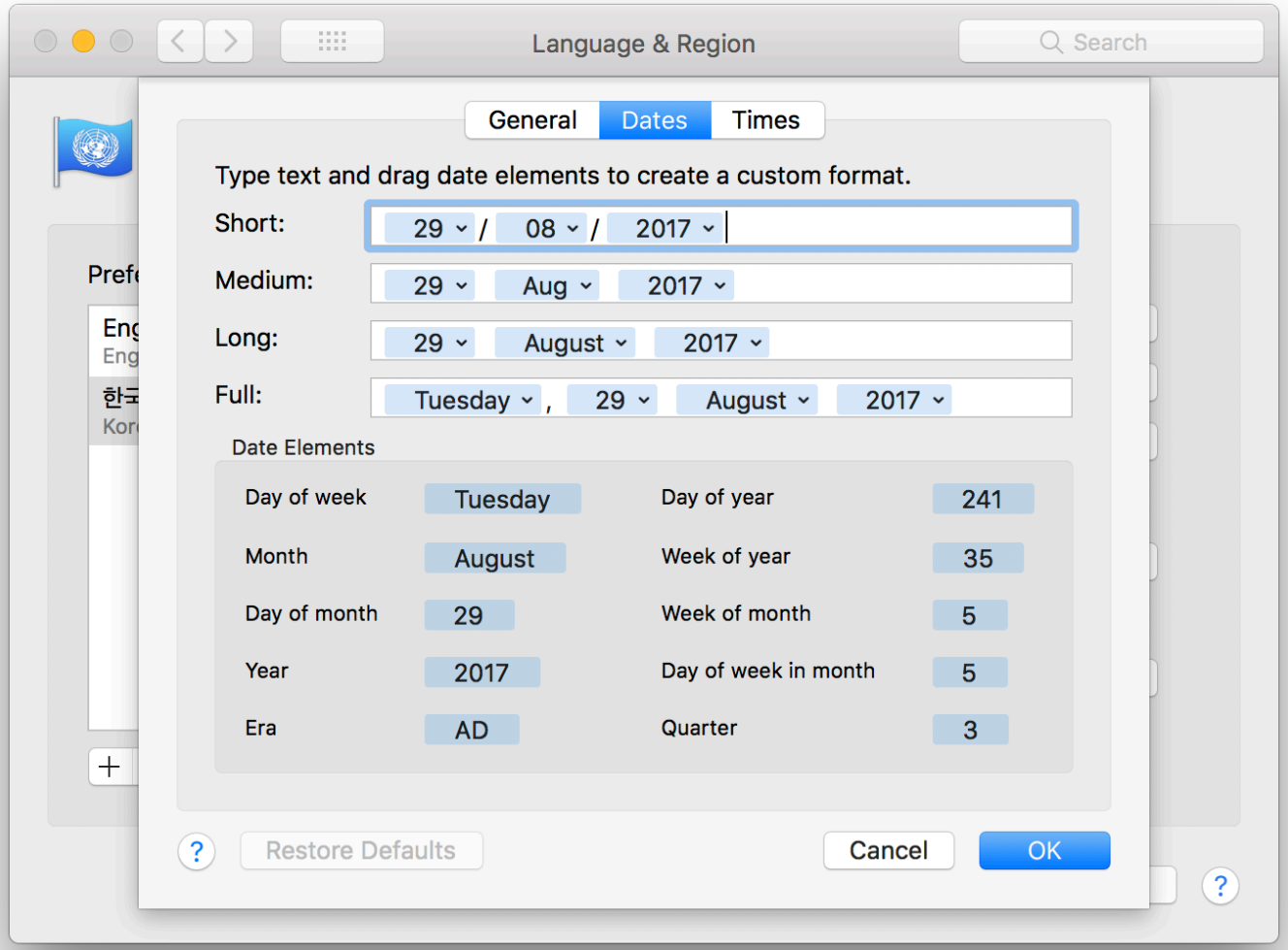
NSDateFormatter

날짜 정보와 문자열 형태를 상호 변환하는 기준

NSNumberFormatter

숫자와 문자열 형태를 상호 변환하는 기준

각 클래스에 미리 정의된 스타일을 쓰거나 설정 가능


Language & Region
Search
General
Dates
Times
Type text and drag date elements to create a custom format.
Short:
29 / 08 / 2017
Medium:
29 Aug 2017
Long:
29 August 2017
Full:
Tuesday , 29 August 2017
Date Elements
Day of week
Tuesday
Day of year
241
Month
August
Week of year
35
Day of month
29
Week of month
5
Year
2017
Day of week in month
5
Era
AD
Quarter
3
Restore Defaults
Cancel
OK

Files & URLs

파일과 URL

NSURL 클래스는 파일과 리소스를 참조하기 위한 방법 제공

NSFileManager를 통해서 파일 시스템에 접근 가능

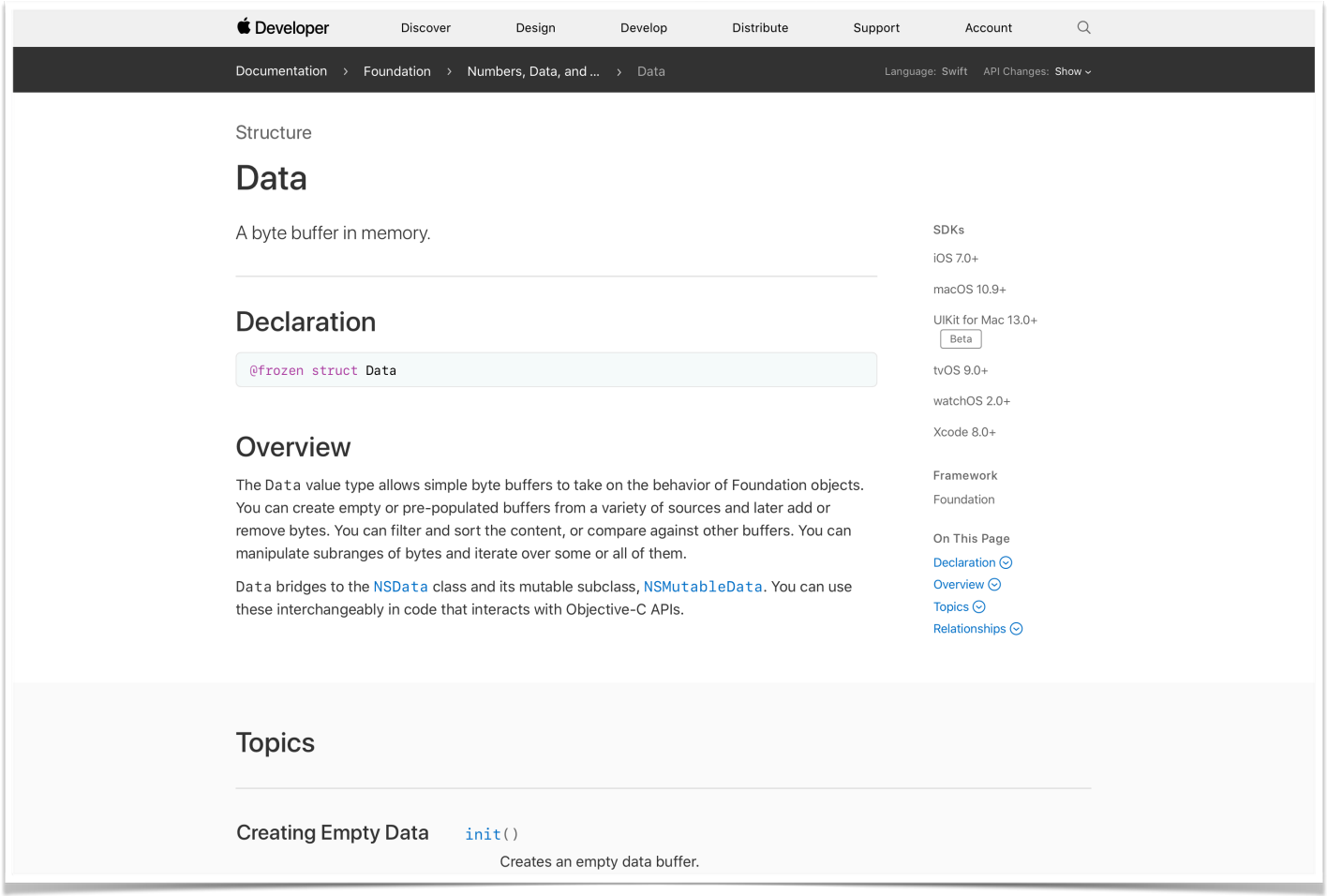
URL 로딩 클래스를 통해서 네트워크 리소스에 접근 가능

Foundation

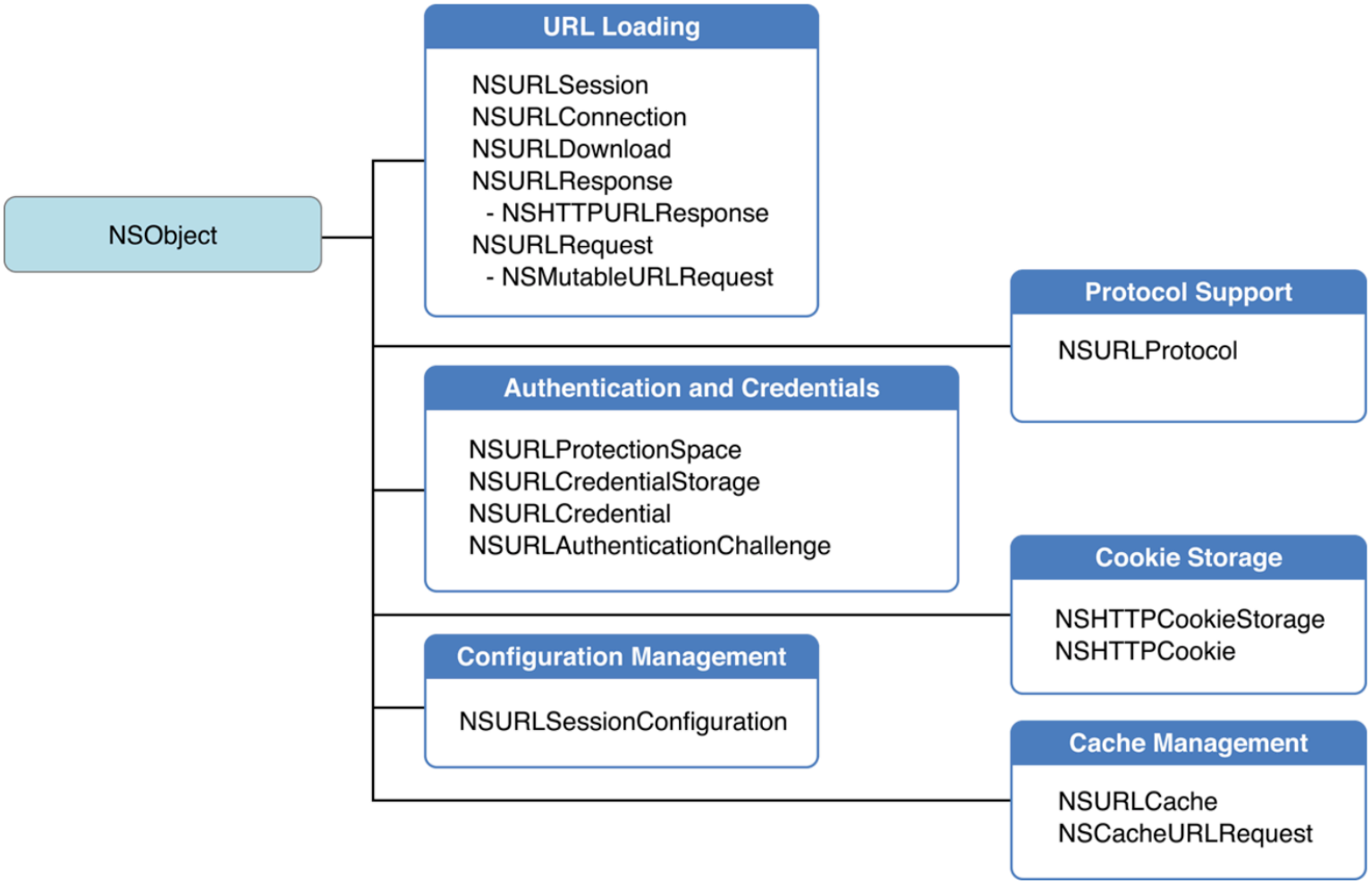
URLComponents

```
var template = URLComponents()
template.scheme = "https"
template.host = "www.apple.com"
template.path = "/shop/buy-iphone"
template.queryItems = [URLQueryItem(name: "step", value: "detail")]

var urls = Array<URLComponents>()
for product in ["iPhone7", "iPhone7-Plus"] {
    var components = template
    components.queryItems!.append(URLQueryItem(name: "product",
                                               value: product))
    urls.append(components)
}
```

Sync Remote Resource


Developer
Discover
Design
Develop
Distribute
Support
Account
Documentation
Foundation
Numbers, Data, and ...
Data
Language: Swift
API Changes: Show
Structure
Data
A byte buffer in memory.
Declaration
@frozen struct Data
Overview
The Data value type allows simple byte buffers to take on the behavior of Foundation objects. You can create empty or pre-populated buffers from a variety of sources and later add or remove bytes. You can filter and sort the content, or compare against other buffers. You can manipulate subranges of bytes and iterate over some or all of them.
Data bridges to the NSData class and its mutable subclass, NSMutableData. You can use these interchangeably in code that interacts with Objective-C APIs.
SDKs
iOS 7.0+
macOS 10.9+
UIKit for Mac 13.0+
Beta
tvOS 9.0+
watchOS 2.0+
Xcode 8.0+
Framework
Foundation
On This Page
Declaration
Overview
Topics
Relationships
Topics
Creating Empty Data
init()
Creates an empty data buffer.

URL Session Classes



Persistence

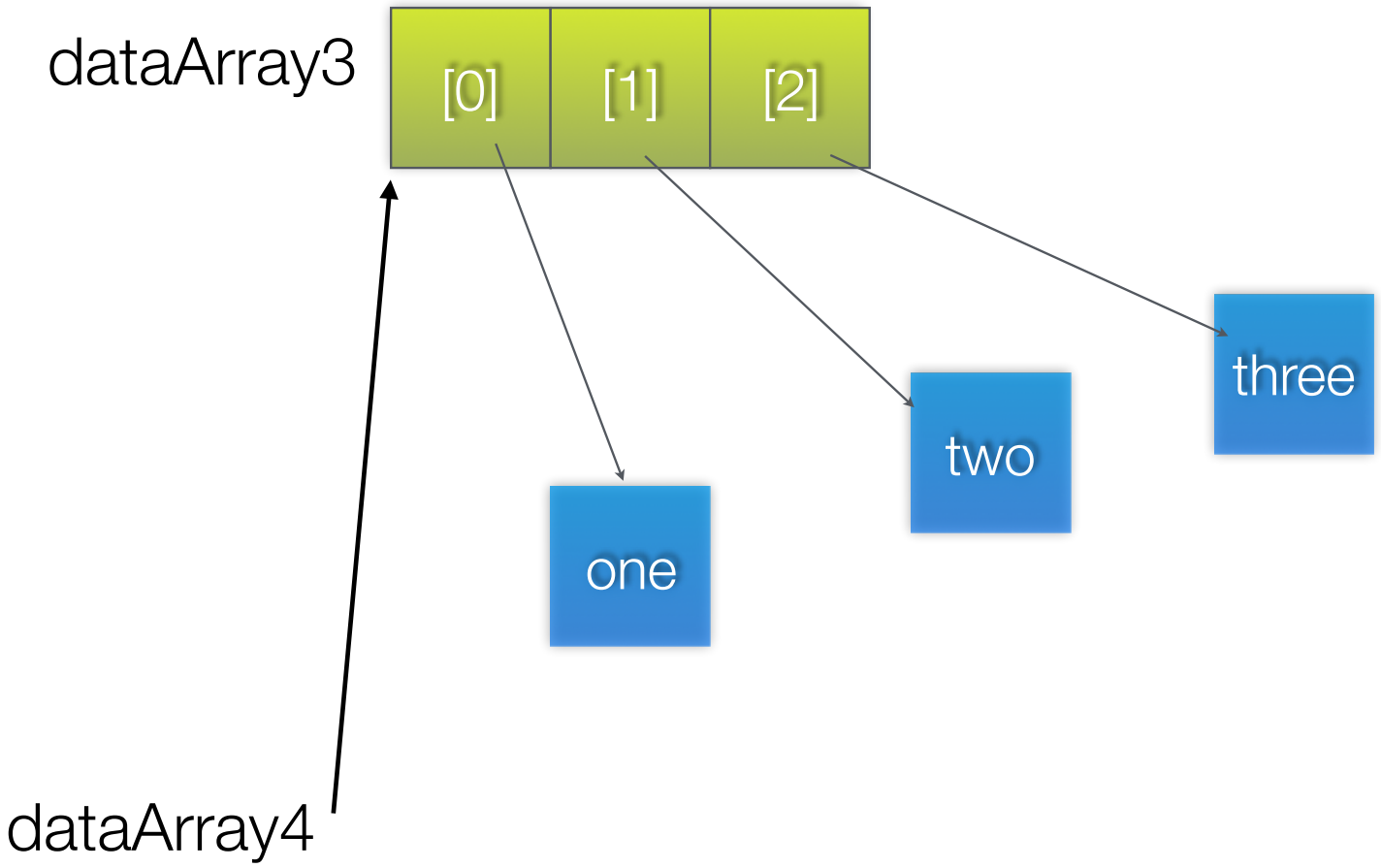
Persistence Cheat Sheet

데이터 타입	추천 방식
사용자 설정	UserDefaults
작은 파일, 크로스 플랫폼	Codable, PropertyListSerialization
객체 참조 그래프와 (non-property list types)	KeyedArchiver
큰 데이터, 객체 참조 데이터베이스	Core Data
특정 형식의 데이터	Custom Format

객체 인스턴스 복사(?)

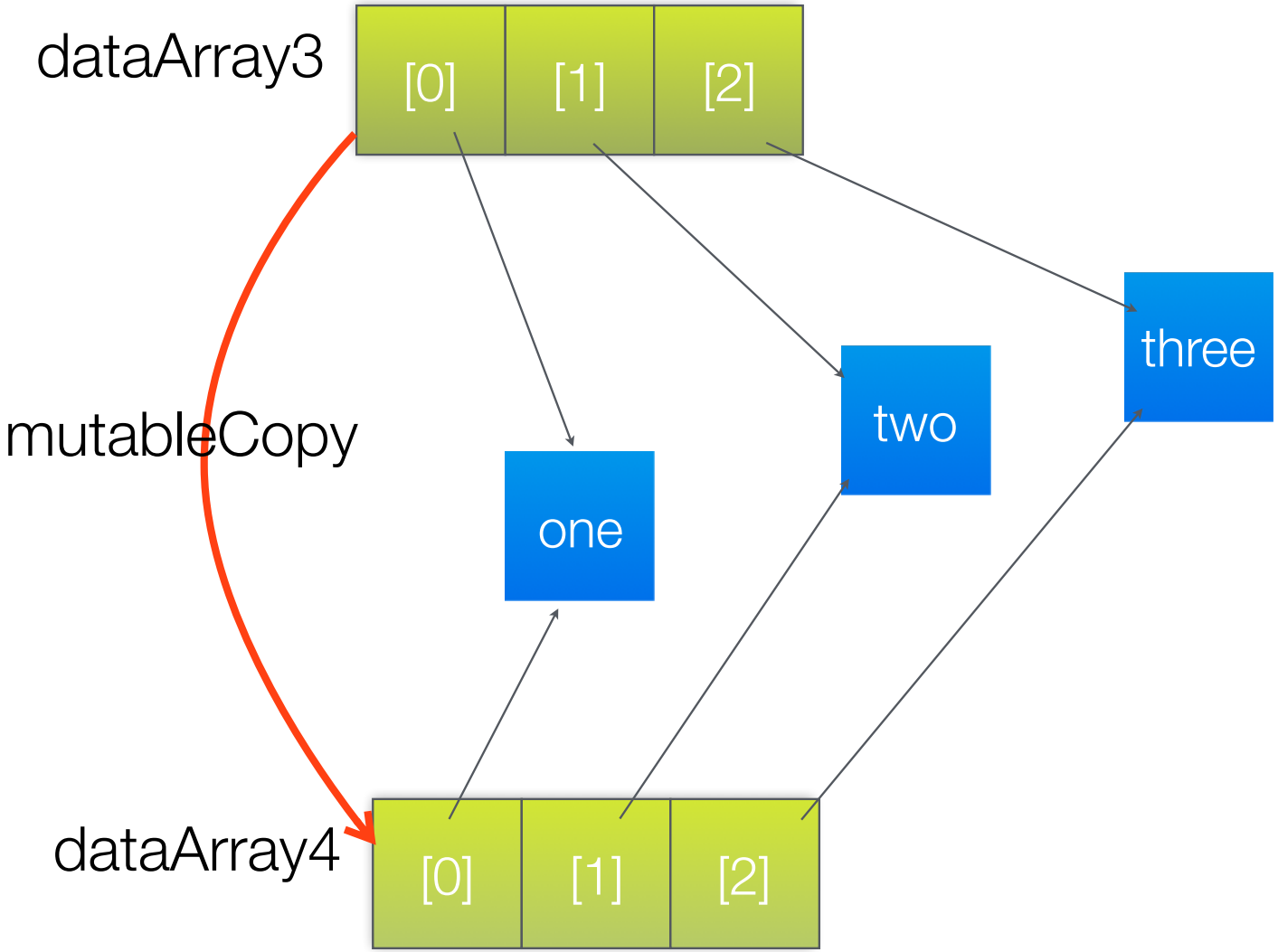
```
var dataArray : [String] = ["one", "two", "three", "four"]

var dataArray2 = dataArray
dataArray2.removeObject(at: 0)
print(dataArray)
print(dataArray2)
```

```
var dataArray3 : NSMutableArray = ["one", "two", "three", "four"]

var dataArray4 = dataArray3
dataArray4.removeObject(at: 0)
print(dataArray3)
print(dataArray4)
```


dataArray3
[0]
[1]
[2]
one
two
three
dataArray4

copy와 mutableCopy 메서드

```
var dataArray3 : NSMutableArray = ["one", "two", "three", "four"]

var dataArray4 = dataArray.mutableCopy() as! NSMutableArray
dataArray4.removeObject(at: 0)
print(dataArray3)
print(dataArray4)
```


dataArray3
[0]
[1]
[2]
mutableCopy
one
two
three
dataArray4
[0]
[1]
[2]

얕은 복사

```
var dataArray : NSMutableArray = [NSMutableString(string: "one"),
NSMutableString(string:"two"), NSMutableString(string:"three"),
NSMutableString(string:"four")
]

dataArray2 = dataArray.mutableCopy() as! NSMutableArray
var stringItem = dataArray[0] as! NSMutableString
stringItem.append("ONE")
print(dataArray)
print(dataArray2)
```

```
var buttonArray : [UIButton] = [UIButton(type: .custom),
             UIButton(type: .infoDark),UIButton(type: .contactAdd)]

var buttonArray2 = buttonArray
buttonArray[0].frame = CGRect(x:100, y:100, width:100, height:100)
print(buttonArray)
print(buttonArray2)
```

NSCopying 프로토콜 구현하기

```
extension Milk : NSCopying {
    public func copy(with zone: NSZone? = nil) -> Any {
        let newMilk = Milk()
        return newMilk
    }
}


var whiteMilk1 = Milk()
whiteMilk1.brand = "seoul"
var whiteMilk2 = whiteMilk1.copy() as! Milk
whiteMilk2.brand = "maeil"

print(whiteMilk1.brand)
print(whiteMilk2.brand)
```

Bundles

번들 구조

코드와 리소스를 묶어놓은 것

플랫폼이나 아키텍처에 따라 여러 바이너리도 가능

로컬라이징 리소스 로딩에 간편한 구조

예) 프레임워크 번들, 파일 패키지, 앱 번들 패키지

macOS 앱 번들 구조

```
MyApplication.app/
    Contents/
        Info.plist
        MacOS/
            MyApplication
        Resources/
            MyApplication.icns
            en.lproj/
                localizedImage.png
                MainMenu.nib
                Localizable.strings
            fr.lproj/
                localizedImage.png
                MainMenu.nib
                Localizable.strings
```

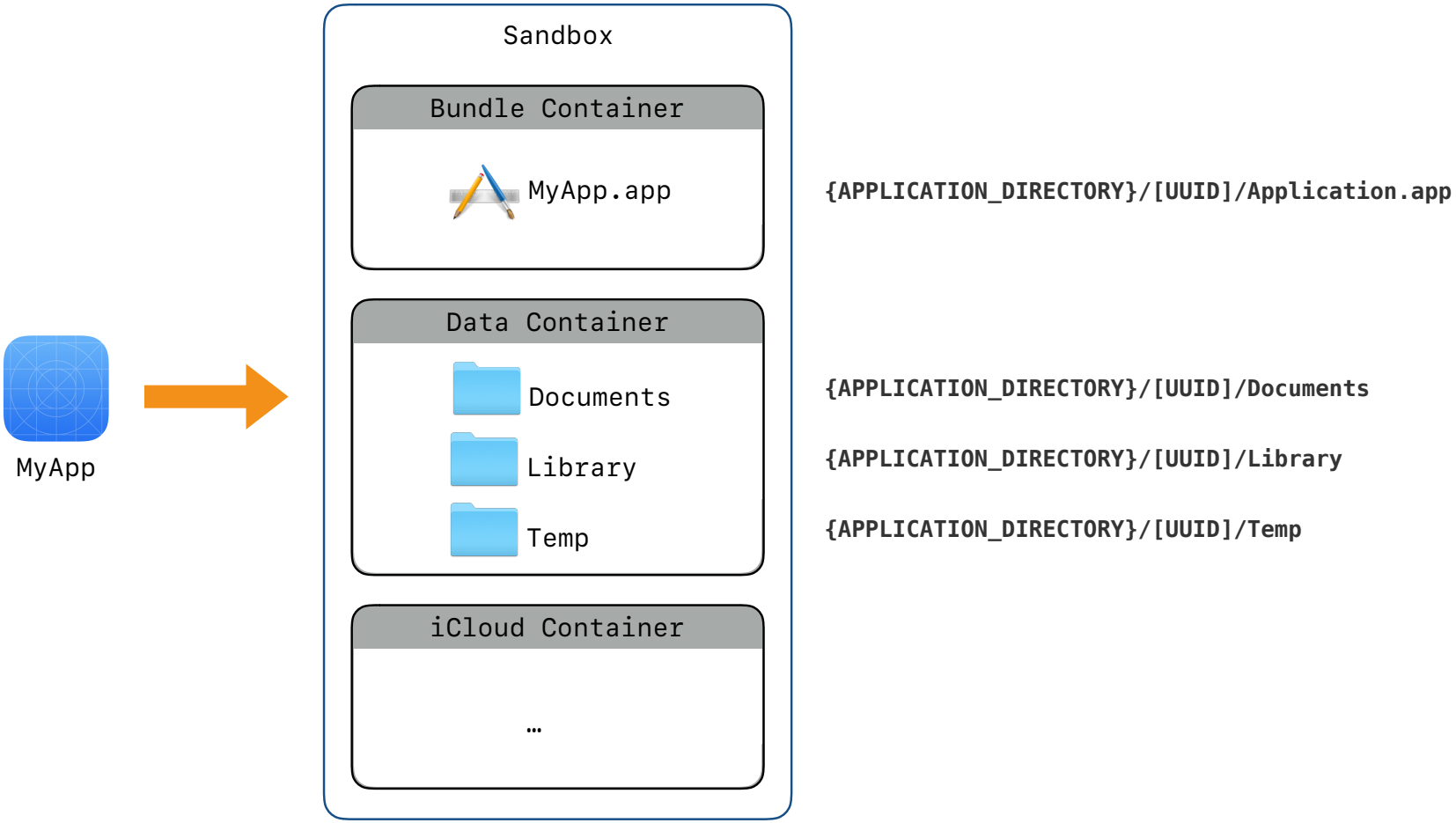
iOS Bundle

~/Library/Developer/CoreSimulator/Devices/…

```
…/Devices/5893E592-C69A-4472-8E38-062C3D1DFC7B/
data/Containers/Bundle/Application/
FB333984-21AC-410C-A8ED-0352BC5F868A/
SwiftExample.app
```

iOS App Sandbox

{APPLICATION_DIRECTORY} =>
/var/mobile/Containers/Bundle/Application



번들 리소스 불러오기

```
let url = Bundle.main.url(forResource: "localizedImage",
                          withExtension: "png")
```

en.lproj/localizedImage.png

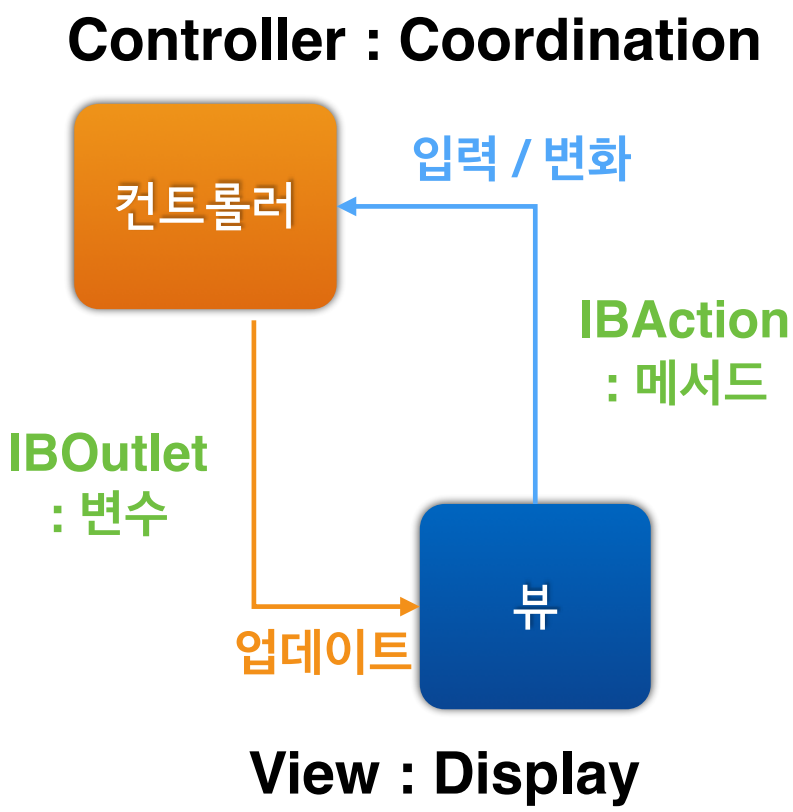
fr.lproj/localizedImage.png

유용한 Value 타입들

AffineTransform
CharacerSet
Data
Date
DateComponents
DateInterval
Decimal
IndexPath
IndexSet
Measurement
Notification
PersonNameComponents
URL
URLComponents
URLRequest
URLQueryItem
UUID

Swift vs Objective-C Runtime

IBOutlet vs. IBAction



코딩미션

Measurement 앱 만들기

코드스쿼드

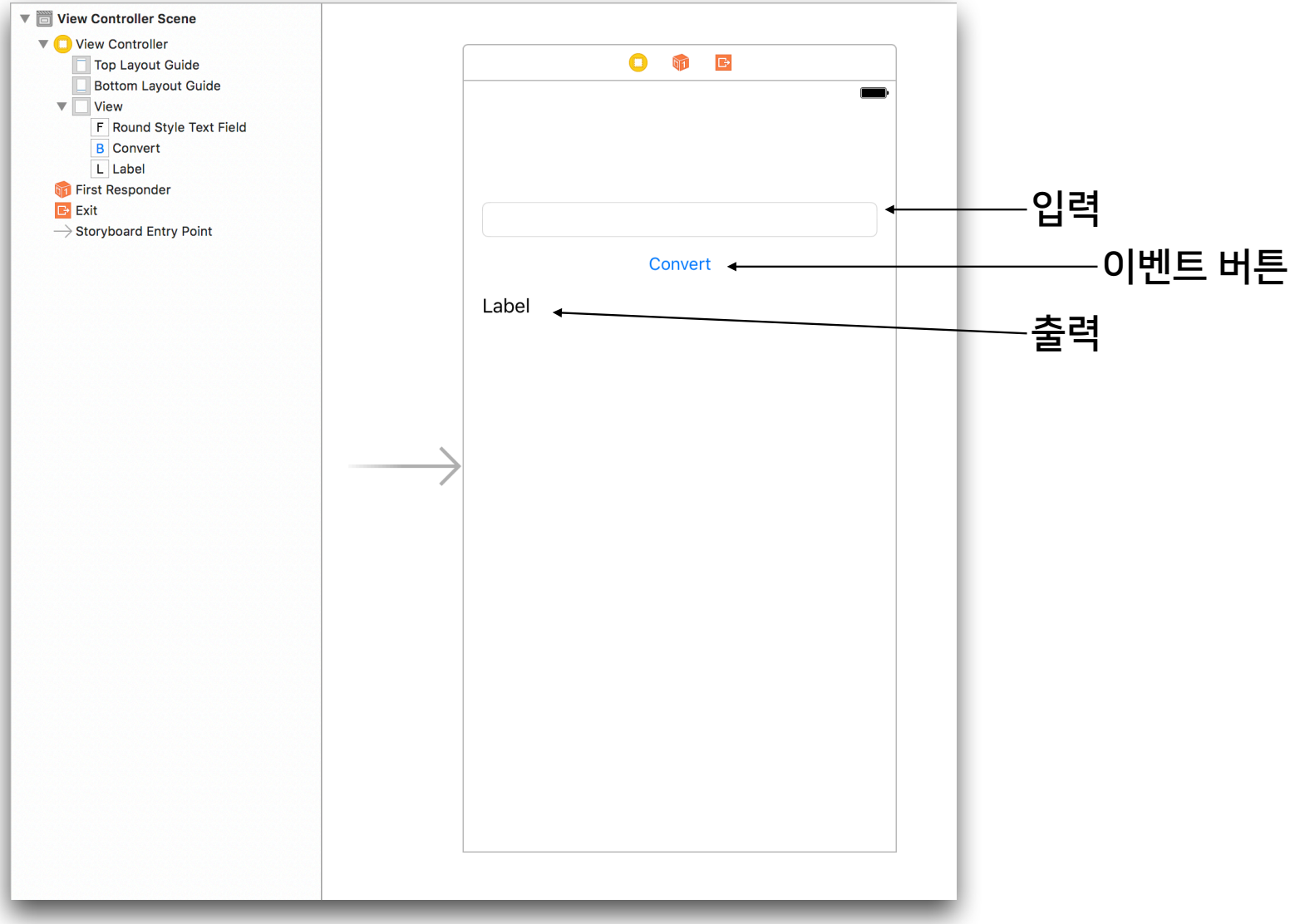
Measurement 활용하기

https://developer.apple.com/reference/foundation/nsmeasurement

```
let street1 = Measurement(value: 73, unit: UnitLength.meters)
let street2 = Measurement(value: 67, unit: UnitLength.meters)
var commuteDistance = street1 + street2
commuteDistance.convert(to: UnitLength.yards)
```

https://developer.apple.com/reference/foundation/dimension

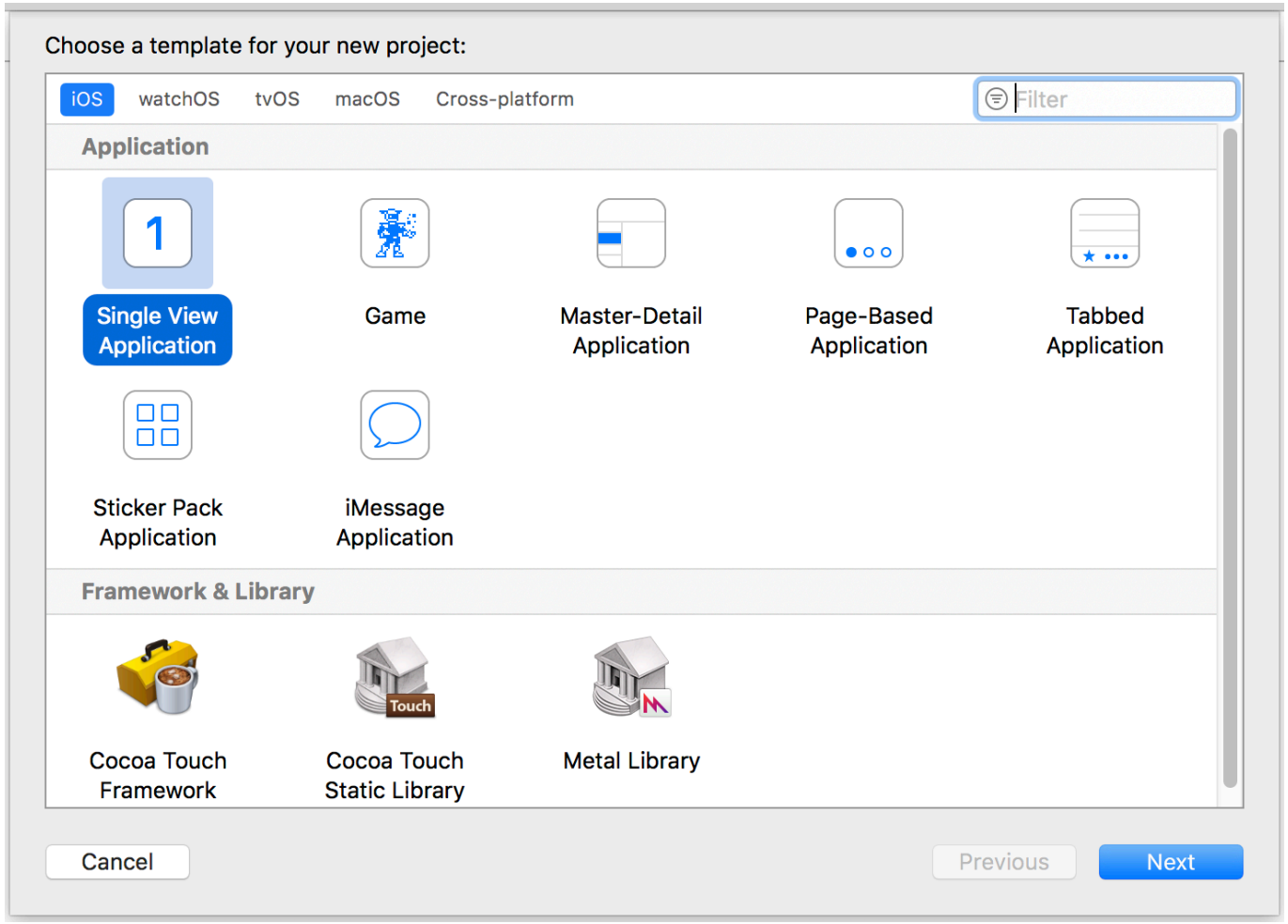
여러 단위가 미리 만들어져 있음
길이 제외하고 단위 중에 하나를 고르세요!


View Controller Scene
View Controller
Top Layout Guide
Bottom Layout Guide
View
Round Style Text Field
Convert
Label
First Responder
Exit
Storyboard Entry Point
Convert
Label
입력
이벤트 버튼
출력

iOS 앱 아키텍처

앱 템플릿

File > New > Project



UIApplication

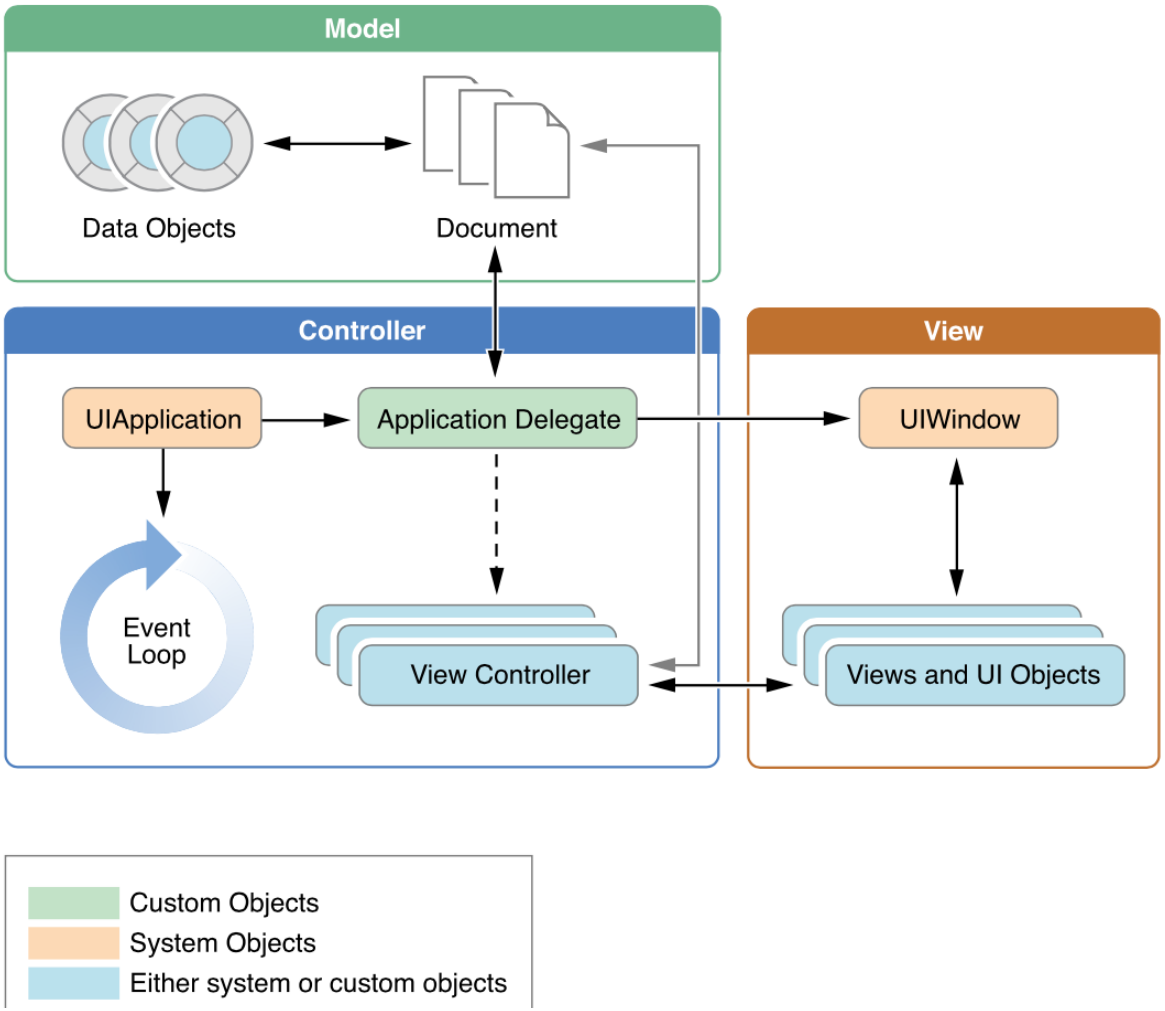
https://developer.apple.com/documentation/uikit/uiapplication

https://developer.apple.com/documentation/uikit/1622933-uiapplicationmain

```
import UIKit

UIApplicationMain(
    CommandLine.argc,
    CommandLine.unsafeArgv,
    NSStringFromClass(UIApplication.self),
    NSStringFromClass(AppDelegate.self)
)
```

Key Objects

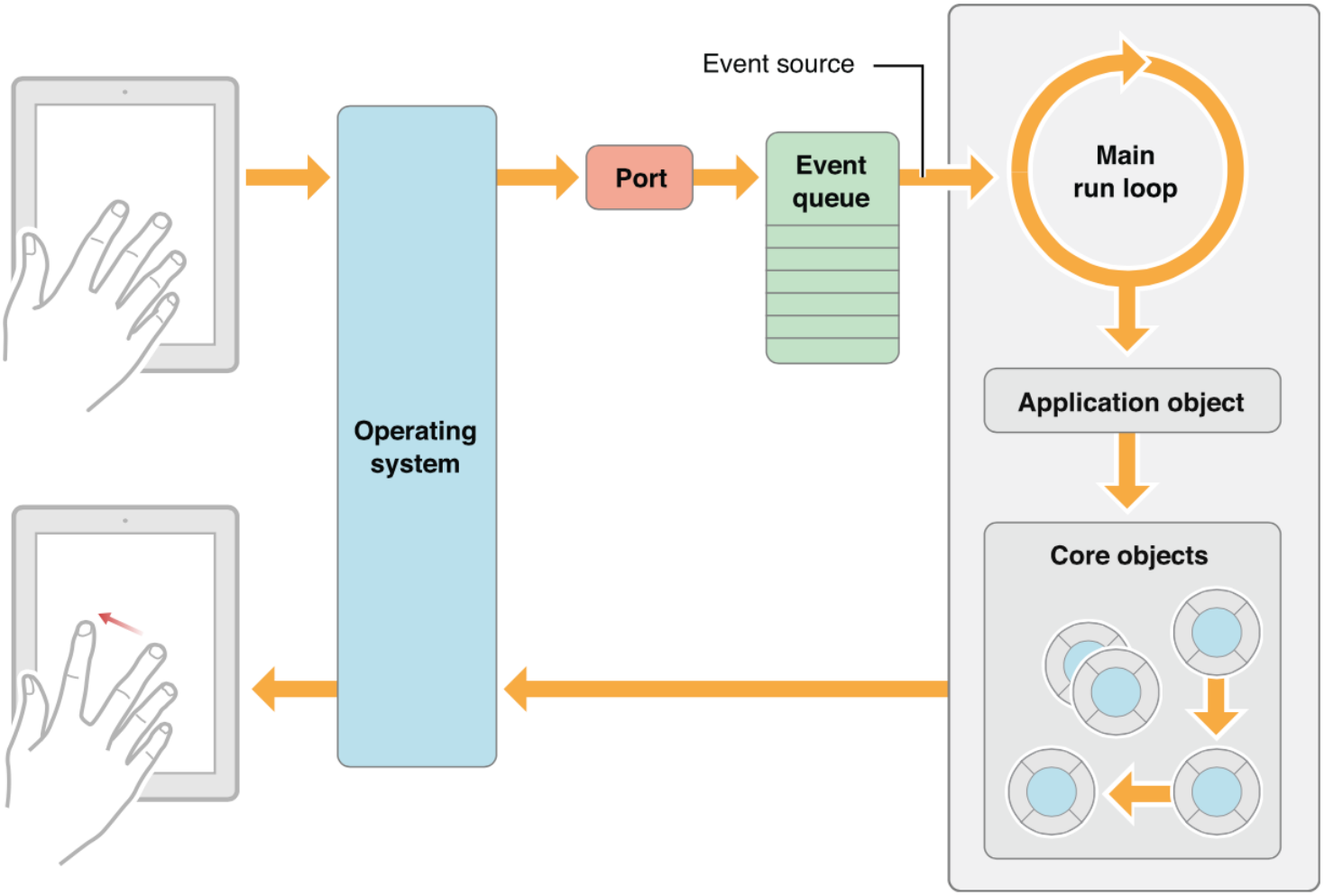
Figure 2-1 Key objects in an iOS app



출처: App Programming Guide

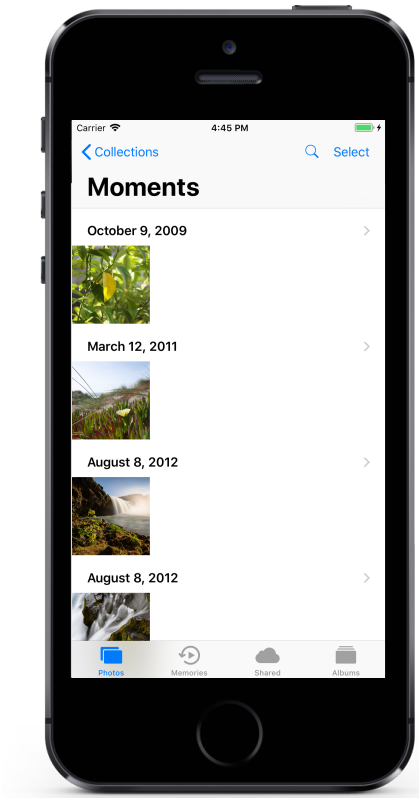
Main run-loop

Figure 3-7 Processing events in the main run loop



출처: App Programming Guide

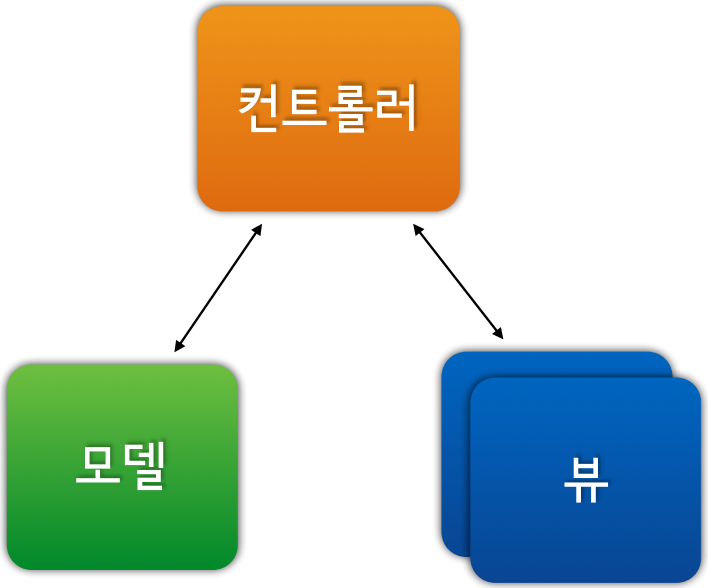
UIViewController



View Controller를 권장하는 이유

* 품질 좋은 앱을 ViewController 단위로 간편하게 만들기 위해서
* 재사용 가능한 단위로 ViewController를 사용함

View Controller 기초



* MVC 패턴에서 C
* 화면 전체 콘텐츠를 담당하거나 특정 영역을 담당
* 모델 객체를 내부 프로퍼티로 포함하기도 함
* 앱은 View Controller 간의 흐름으로 구성됨

UIViewController 주요 콜백

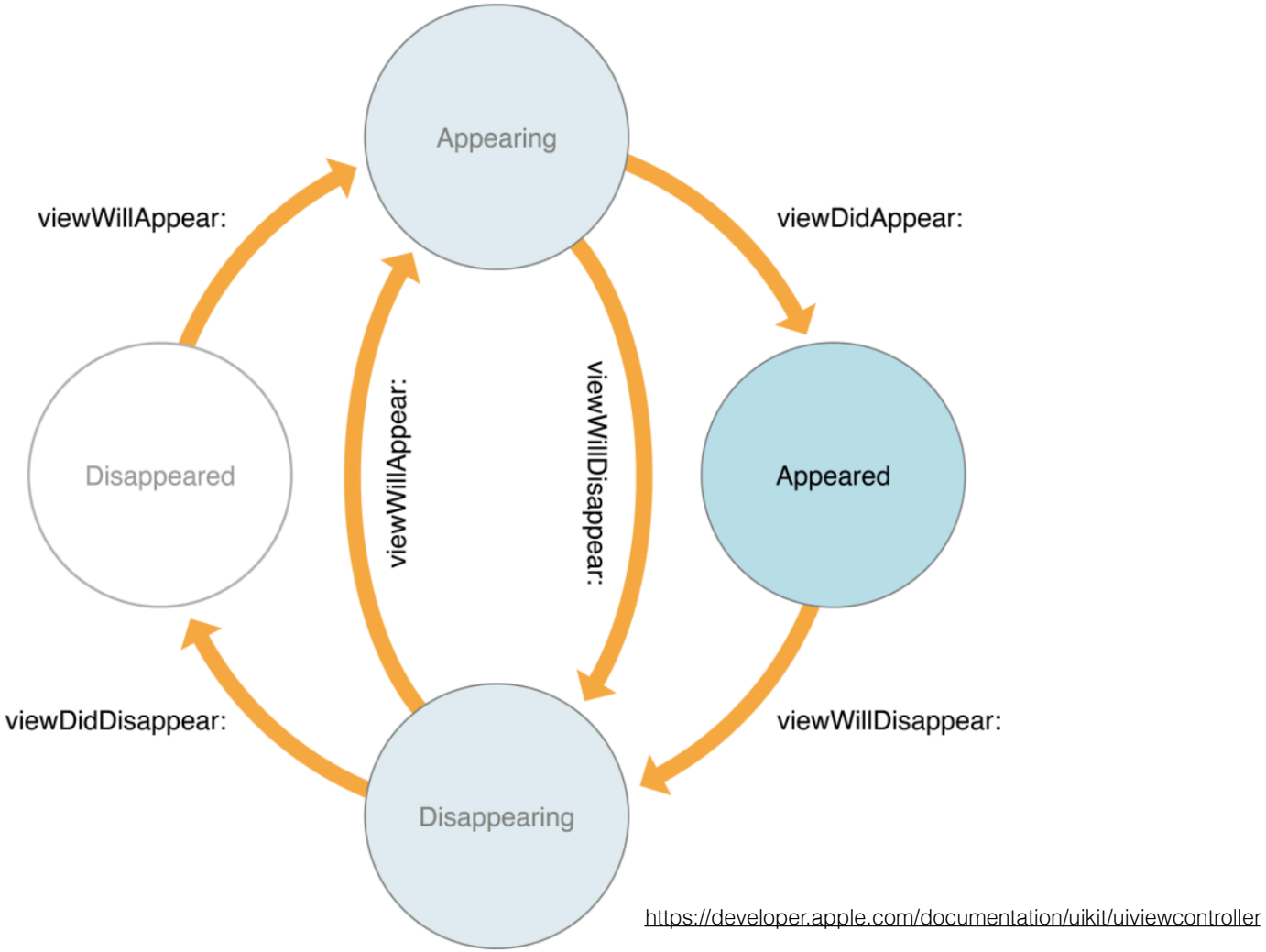
화면 관련

```
-viewDidLoad()
-viewWillAppear(Bool)
-viewDidAppear(Bool)
-viewWillDisappear(Bool)
-viewDidDisappear(Bool)
```

회전 관련

```
var shouldAutorotate : Bool
var supportedInterfaceOrientations : UIInterfaceOrientationMask
var preferredInterfaceOrientationForPresentation : UIInterfaceOrientation
```

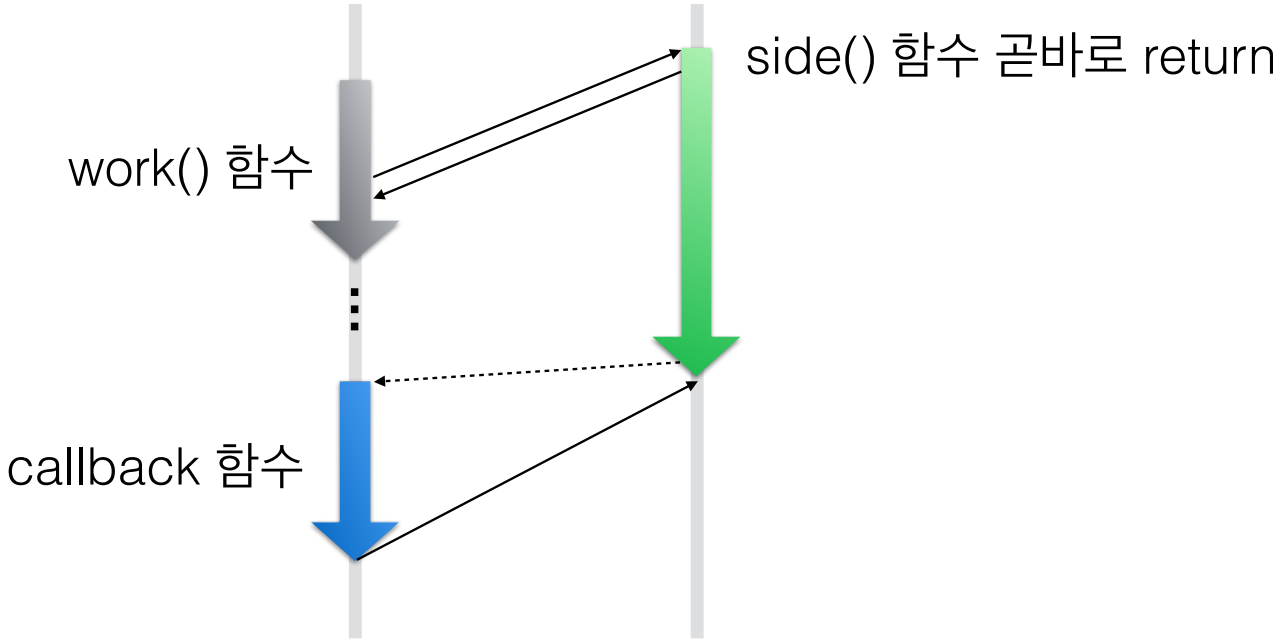
View-related States



https://developer.apple.com/documentation/uikit/uiviewcontroller

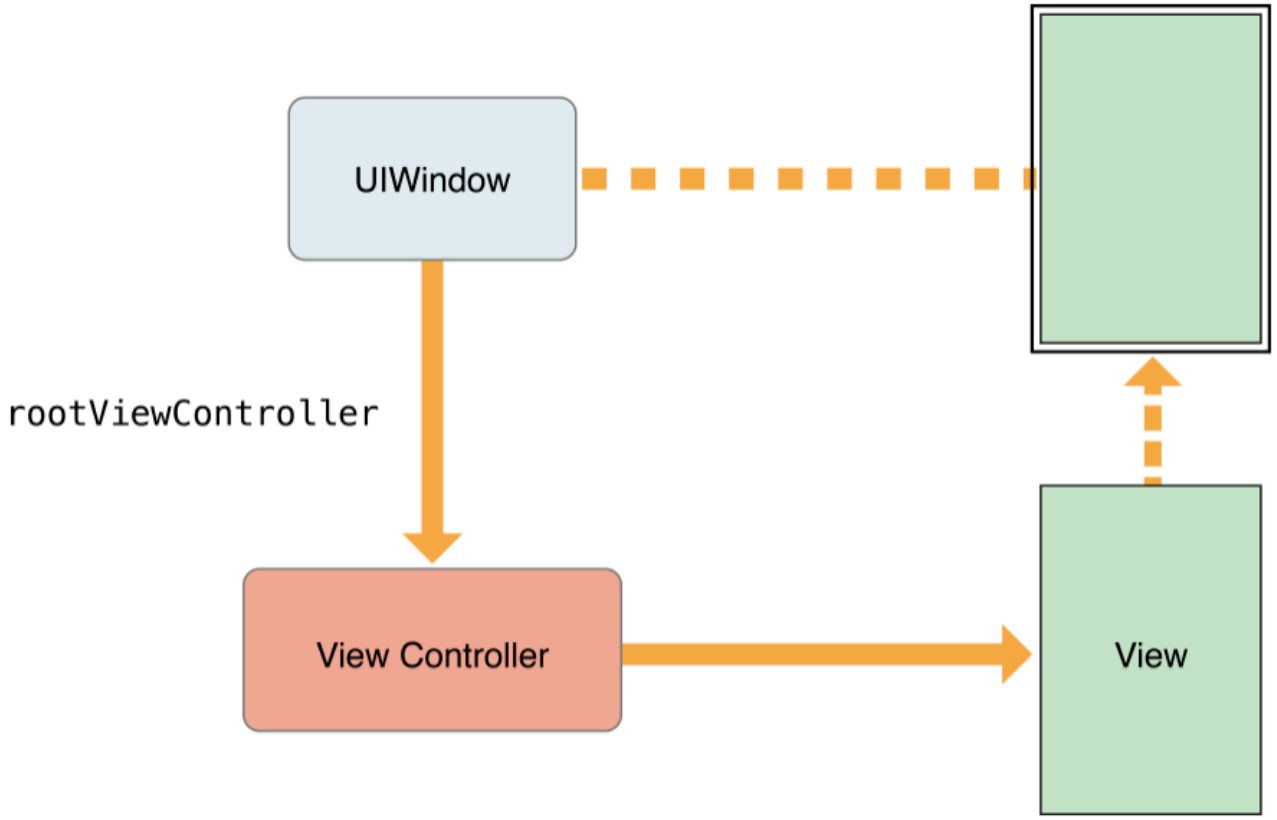
비동기 Asynchronous (콜백)

* 타이밍을 안 맞춘다
* 타이밍이 맞을때까지 기다리지 않는다
* (**누군가** 작업하고) 다했으면 (**어디로**) 알려줘 (콜백!)



root viewcontroller

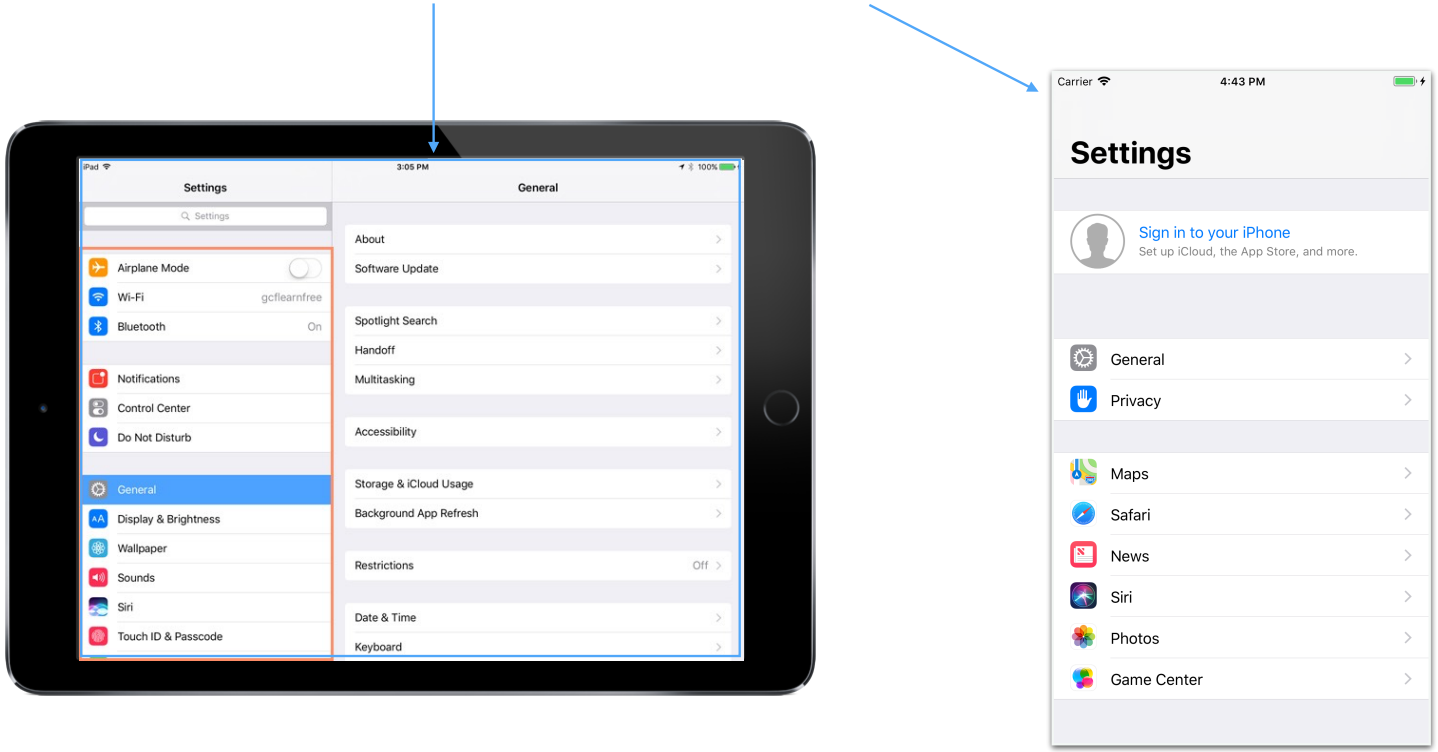
Figure 2-1 The root view controller



View Controller 기초 용어

root view controller 라는 것은 “화면을 가득채우는 것”

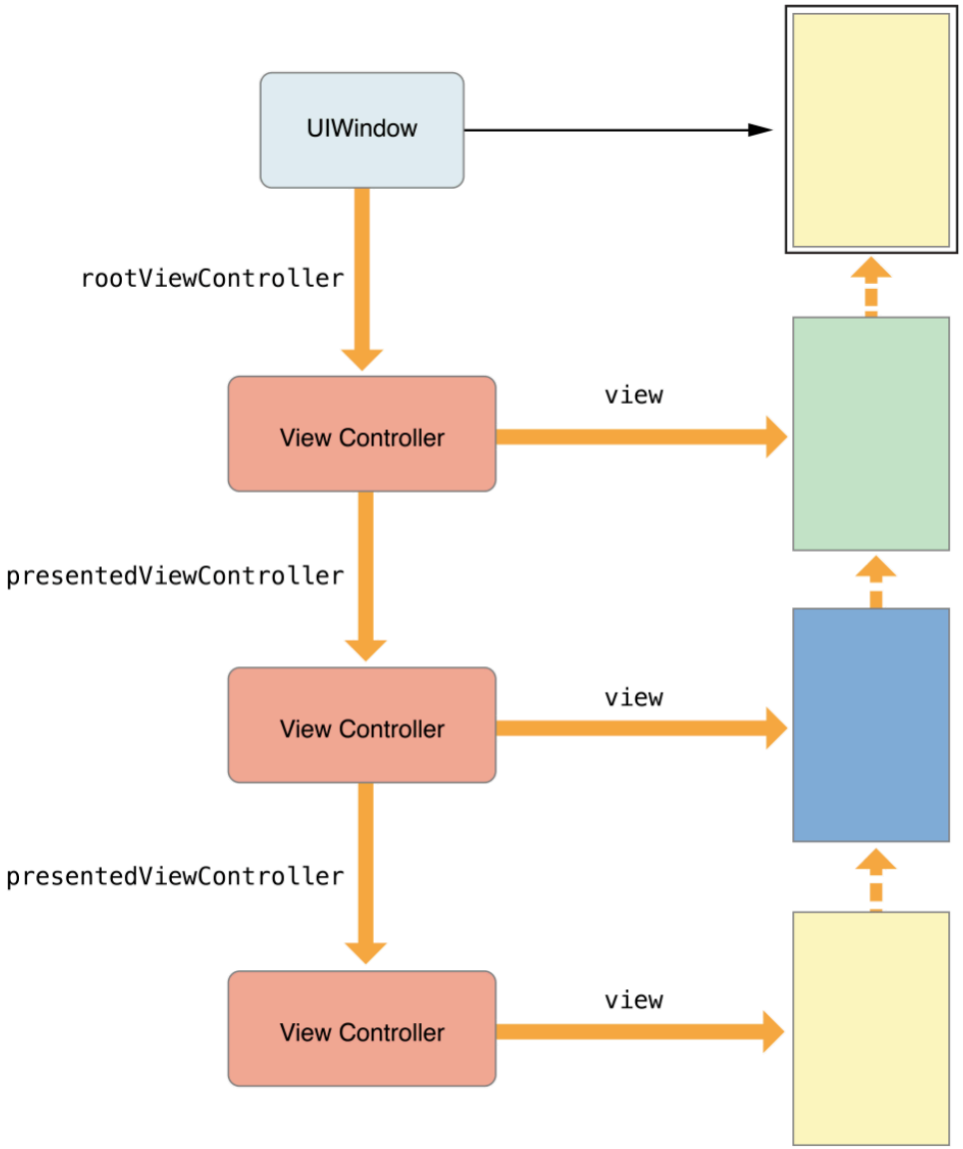
```
window.rootViewController = myViewController
```



Custom View Controller

* UIViewControlller 서브클래스
* View 계층에 맞춰 관련있는 ViewControlller 다루기
* 필요한 API 메서드 오버라이드하기
* 자신만의 비즈니스 로직 추가
* 앱의 동작 흐름에 맞추기

Figure 2-3 Presented view controllers

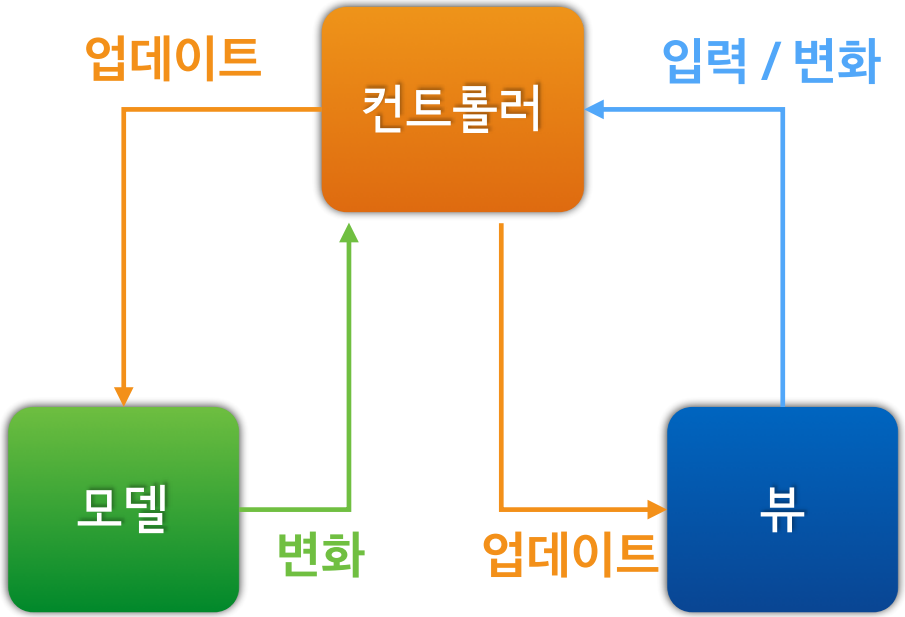


Model - View - Controller

Model - View - Controller

UIViewController

Controller : Coordination



Model : Data

NSObject

View : Display

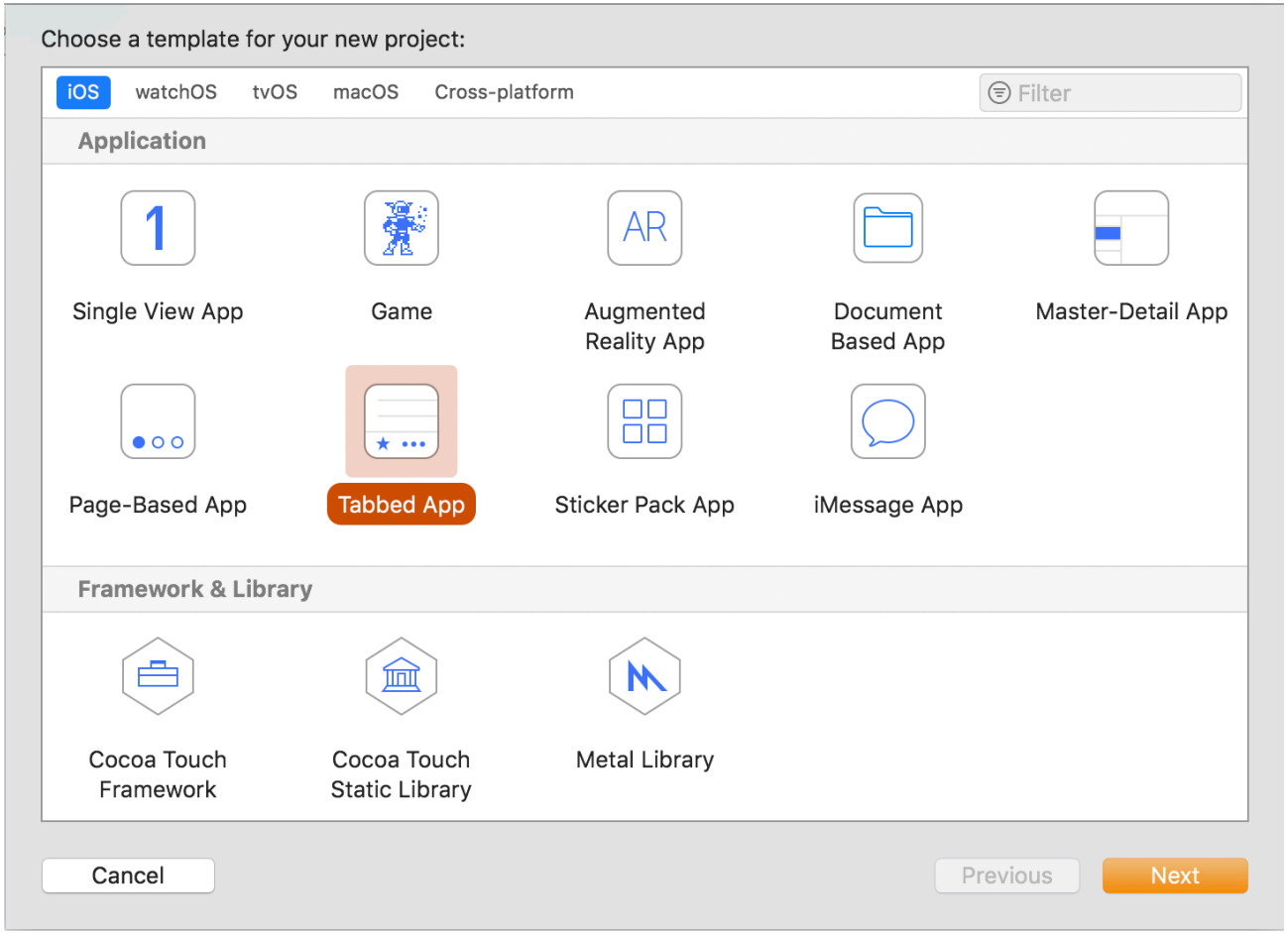
UIView

간단 앱을 만들어봅시다!

미니 프로젝트

개인 미션

- Tabbed Application 프로젝트를 만드세요



Clock app

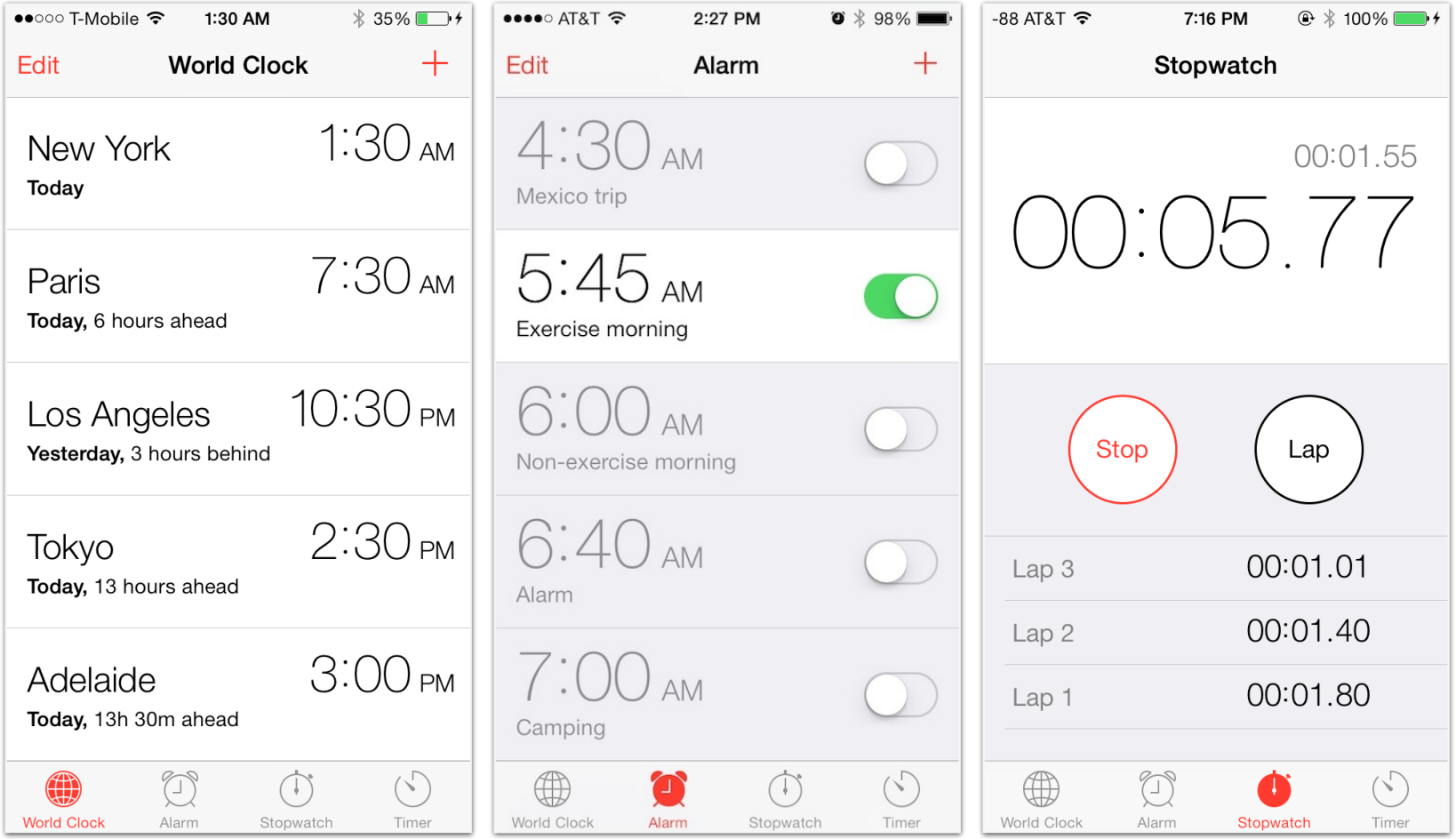
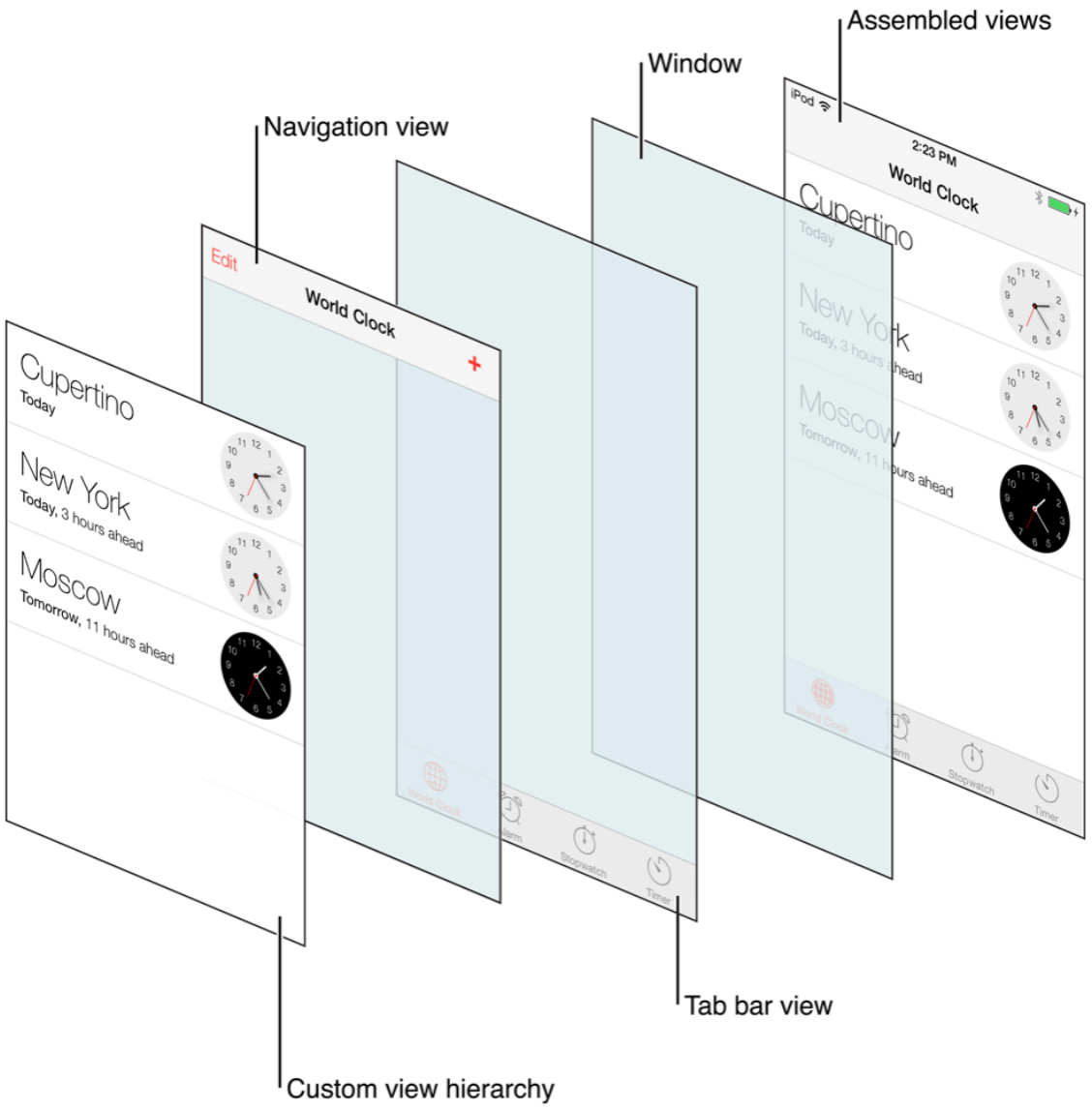
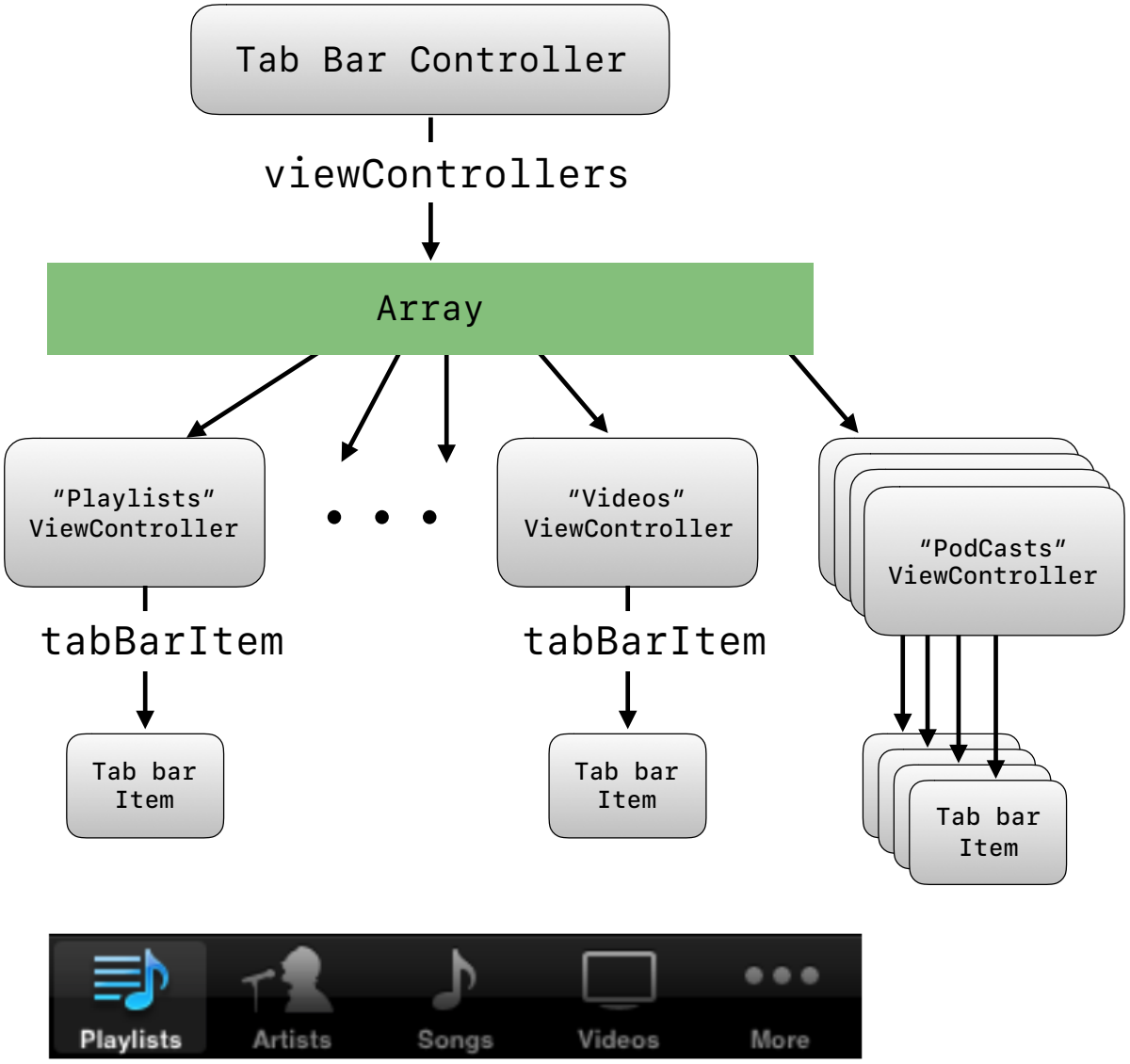
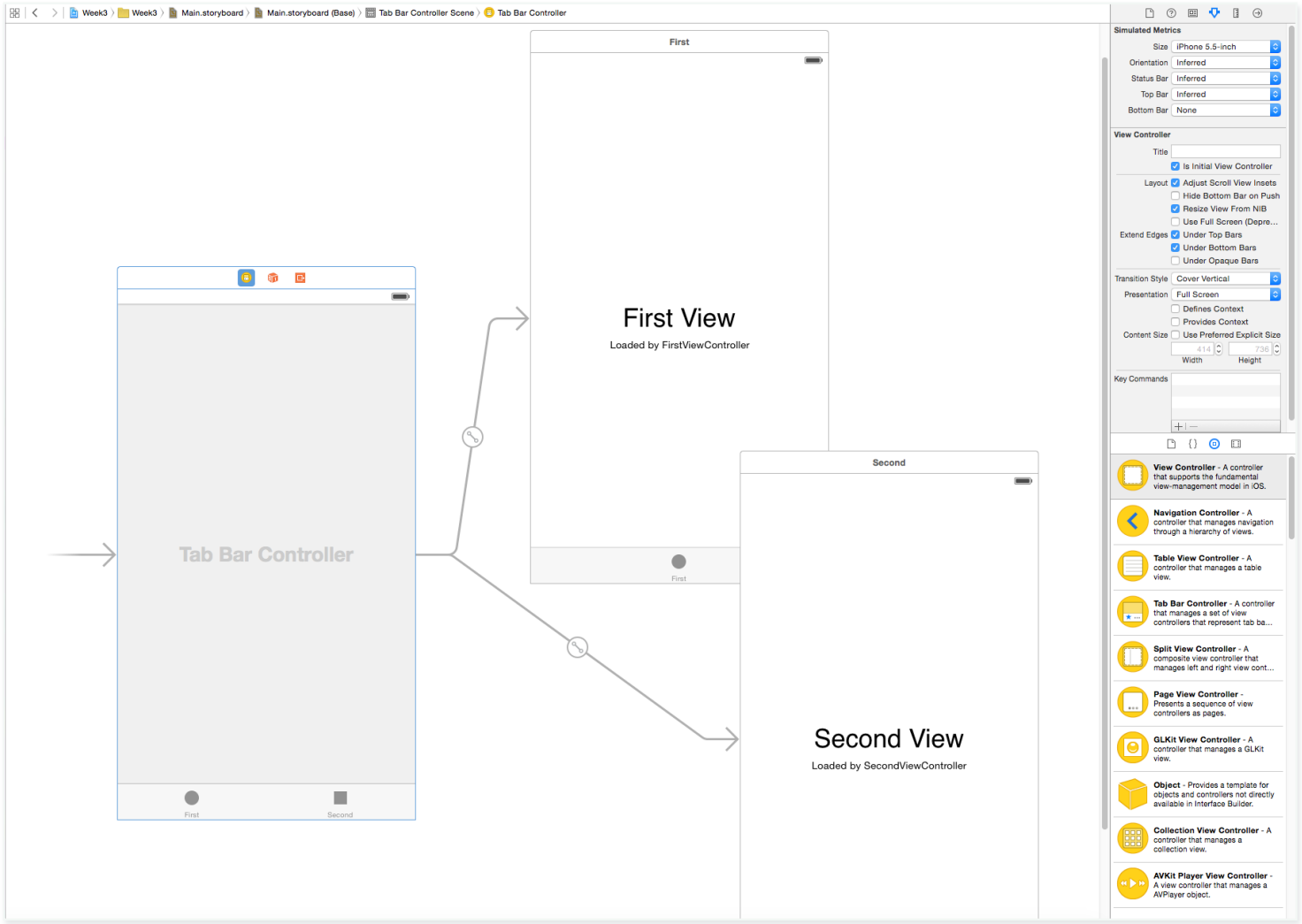

●●○○○ T-Mobile 1:30 AM 35%
Edit
World Clock
+
New York
Today
1:30 AM
Paris
Today, 6 hours ahead
7:30 AM
Los Angeles
Yesterday, 3 hours behind
10:30 PM
Tokyo
Today, 13 hours ahead
2:30 PM
Adelaide
Today, 13h 30m ahead
3:00 PM
World Clock
Alarm
Stopwatch
Timer
●●●●○ AT&T 2:27 PM 98%
Edit
Alarm
+
4:30 AM
Mexico trip
5:45 AM
Exercise morning
6:00 AM
Non-exercise morning
6:40 AM
Alarm
7:00 AM
Camping
World Clock
Alarm
Stopwatch
Timer
-88 AT&T 7:16 PM 100%
Stopwatch
00:01.55
00:05.77
Stop
Lap
Lap 3
00:01.01
Lap 2
00:01.40
Lap 1
00:01.80
World Clock
Alarm
Stopwatch
Timer

Figure 1-1 The views of a navigation interface



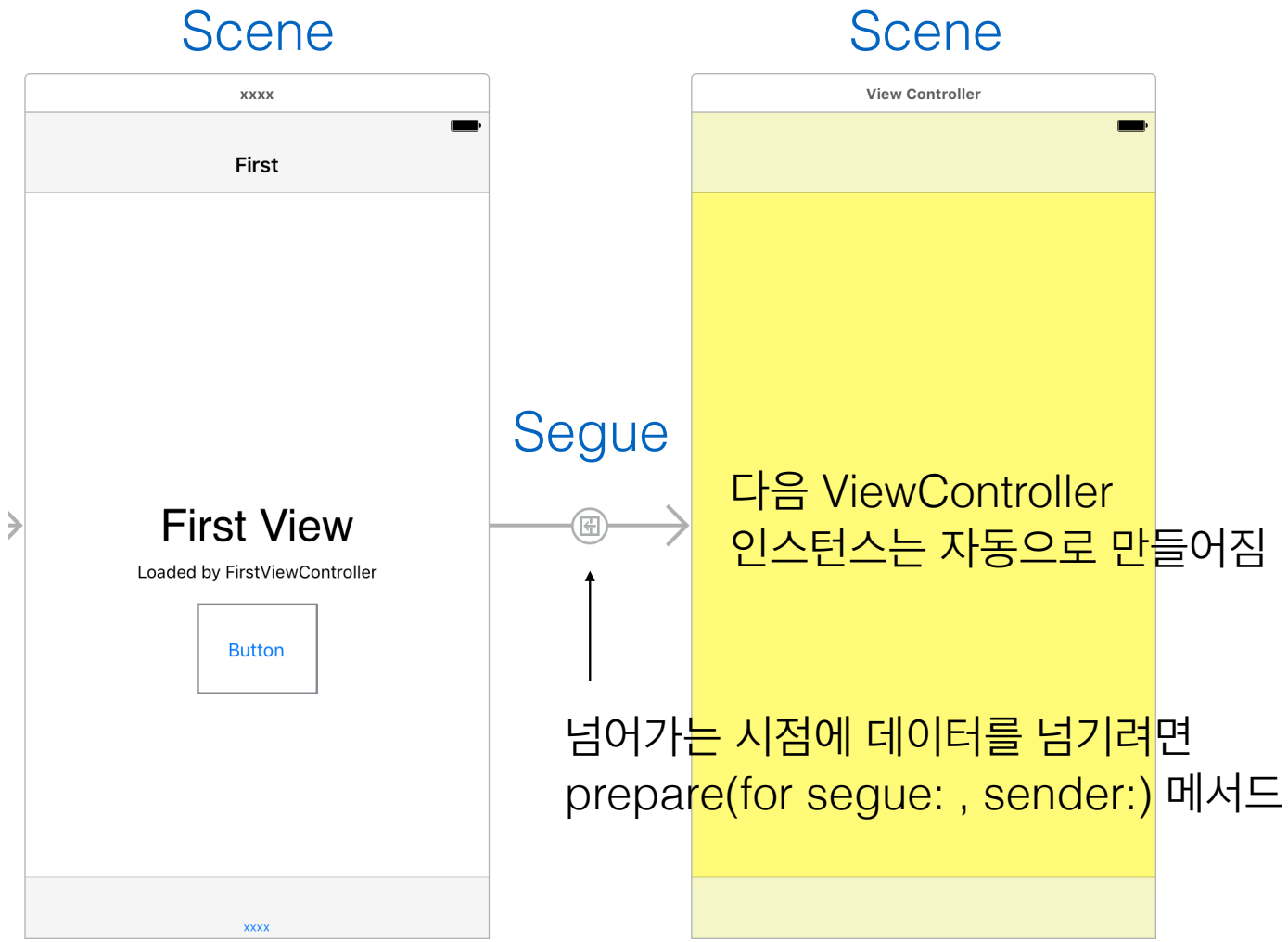

Tab Bar Controller
viewControllers
Array
"Playlists" ViewController
"Videos" ViewController
"PodCasts" ViewController
tabBarItem
tabBarItem
Tab bar Item
Tab bar Item
Tab bar Item
Playlists
Artists
Songs
Videos
More

Storyboard


Week3 〉 Week3 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 Tab Bar Controller Scene 〉 Tab Bar Controller
First
First View
Loaded by FirstViewController
First
Second
Second View
Loaded by SecondViewController
Tab Bar Controller
First
Second
Simulated Metrics
Size iPhone 5.5-inch
Orientation Inferred
Status Bar Inferred
Top Bar Inferred
Bottom Bar None
View Controller
Title
Is Initial View Controller
Layout Adjust Scroll View Insets
Hide Bottom Bar on Push
Resize View From NIB
Use Full Screen (Depre...
Extend Edges Under Top Bars
Under Bottom Bars
Under Opaque Bars
Transition Style Cover Vertical
Presentation Full Screen
Defines Context
Provides Context
Content Size Use Preferred Explicit Size
Width
Height
Key Commands
View Controller - A controller that supports the fundamental view-management model in iOS.
Navigation Controller - A controller that manages navigation through a hierarchy of views.
Table View Controller - A controller that manages a table view.
Tab Bar Controller - A controller that manages a set of view controllers that represent tab ba...
Split View Controller - A composite view controller that manages left and right view cont...
Page View Controller - Presents a sequence of view controllers as pages.
GLKit View Controller - A controller that manages a GLKit view.
Object - Provides a template for objects and controllers not directly available in Interface Builder.
Collection View Controller - A controller that manages a collection view.
AVKit Player View Controller - A view controller that manages a AVPlayer object.

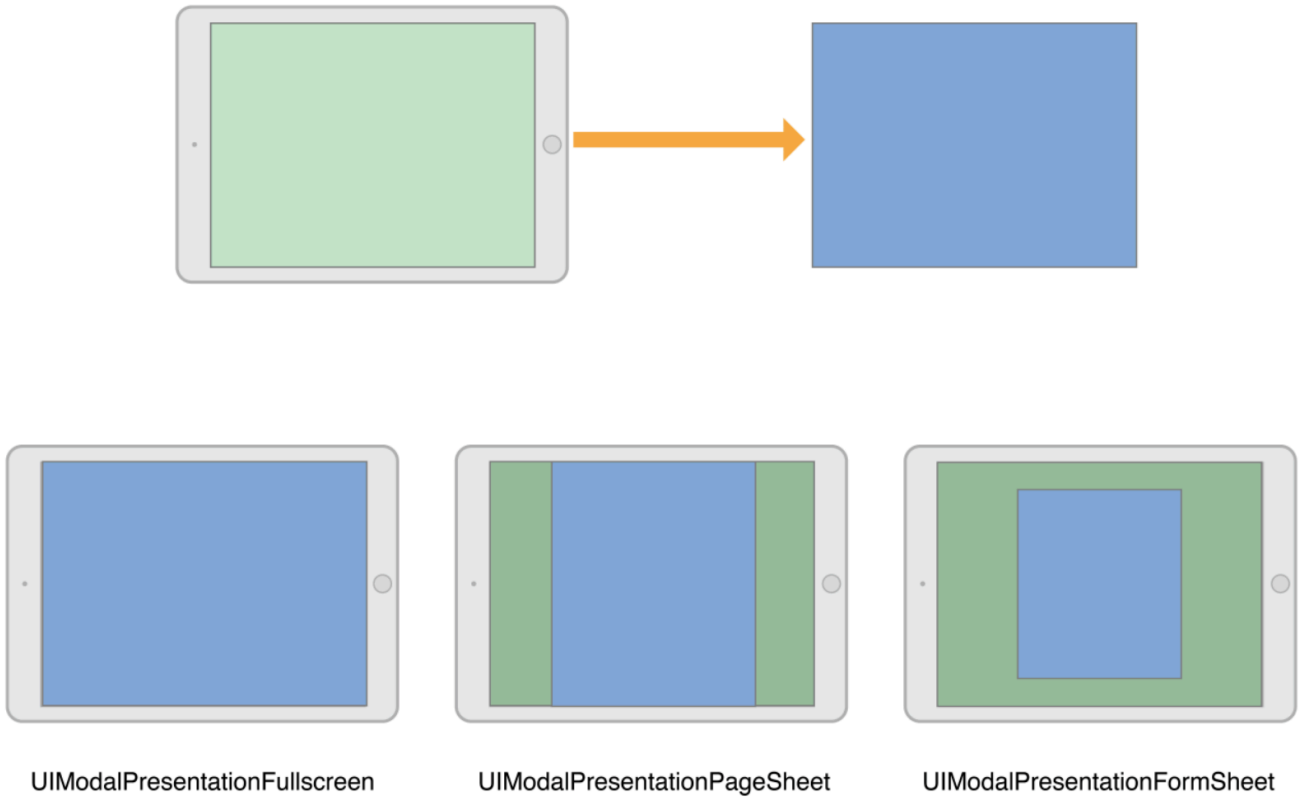
```
<?xml version="1.0" encoding="UTF-8"?>
<document type="com.apple.InterfaceBuilder3.CocoaTouch.Storyboard.XIB" version="3.0" toolsVersion="10116" systemVersion="16A201w"
    targetRuntime="iOS.CocoaTouch" propertyAccessControl="none" useAutolayout="YES" useTraitCollections="YES"
    initialViewController="Xcv-7W-Mgc">
    <dependencies>
        <deployment identifier="iOS"/>
        <plugIn identifier="com.apple.InterfaceBuilder.IBCocoaTouchPlugin" version="10085"/>
    </dependencies>
    <scenes>
        <!--UserInfo-->
        <scene sceneID="tne-QT-ifu">
            <objects>
                <viewController id="BYZ-38-t0r" userLabel="UserInfo" customClass="ViewController" sceneMemberID="viewController">
                    <layoutGuides>
                        <viewControllerLayoutGuide type="top" id="y3c-jy-aDJ"/>
                        <viewControllerLayoutGuide type="bottom" id="wfy-db-euE"/>
                    </layoutGuides>
                    <view key="view" contentMode="scaleToFill" id="8bC-Xf-vdC" customClass="NXMyView">
                        <rect key="frame" x="0.0" y="0.0" width="375" height="667"/>
                        <autoresizingMask key="autoresizingMask" widthSizable="YES" heightSizable="YES"/>
                        <color key="backgroundColor" white="1" alpha="1" colorSpace="custom" customColorSpace="calibratedWhite"/>
                    </view>
                    <tabBarItem key="tabBarItem" title="UserInfo" id="udo-Ue-52b"/>
                    <simulatedScreenMetrics key="simulatedDestinationMetrics" type="retina47"/>
                </viewController>
                <placeholder placeholderIdentifier="IBFirstResponder" id="dkx-z0-nzr" sceneMemberID="firstResponder"/>
            </objects>
            <point key="canvasLocation" x="803.5" y="161.5"/>
        </scene>
        <!--MyLibrary-->
        <scene sceneID="C0T-FK-E3t">
            <objects>
                <tableViewController id="0BG-yq-ofP" userLabel="MyLibrary" sceneMemberID="viewController">
                    <tableView key="view" clipsSubviews="YES" contentMode="scaleToFill" alwaysBounceVertical="YES" dataMode=
                        "prototypes" style="plain" separatorStyle="default" rowHeight="90" sectionHeaderHeight="28"
                        sectionFooterHeight="28" id="Lb5-LA-n7z">
                        <rect key="frame" x="0.0" y="0.0" width="375" height="667"/>
                        <autoresizingMask key="autoresizingMask" widthSizable="YES" heightSizable="YES"/>
                        <color key="backgroundColor" white="1" alpha="1" colorSpace="calibratedWhite"/>
                        <prototypes>
                            <tableViewCell clipsSubviews="YES" contentMode="scaleToFill" selectionStyle="default"
                                indentationWidth="10" rowHeight="90" id="e8b-bE-qGJ">
                                <rect key="frame" x="0.0" y="28" width="375" height="90"/>
                                <autoresizingMask key="autoresizingMask"/>
                                <tableViewCellContentView key="contentView" opaque="NO" clipsSubviews="YES" multipleTouchEnabled=
                                    "YES" contentMode="center" tableViewCell="e8b-bE-qGJ" id="Fa7-q0-DRe">
```

Scene과 Segue



Presentation Style

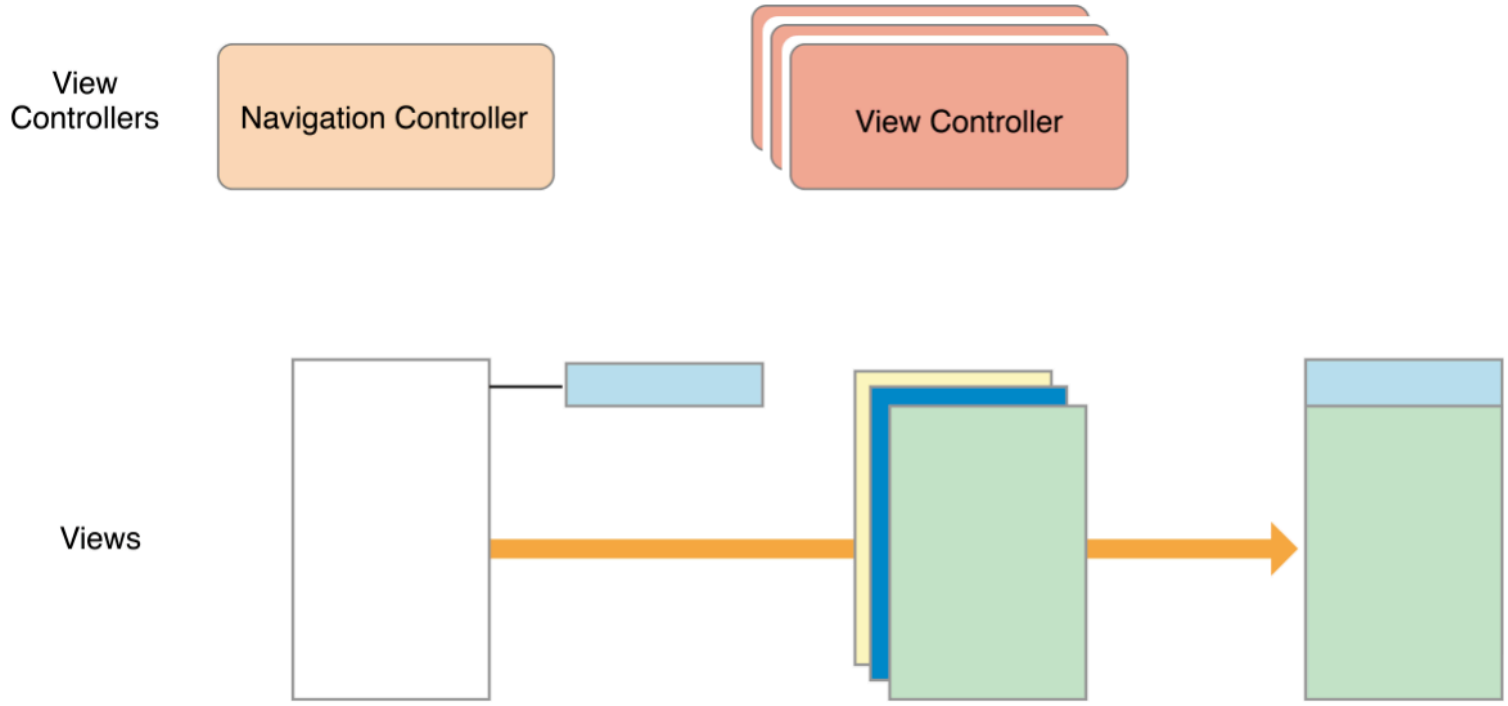
Figure 8-1 The full screen presentation styles



https://developer.apple.com/library/content/featuredarticles/ViewControllerPGforiPhoneOS/PresentingaViewController.html

내비게이션 컨트롤러

Figure 5-1 Structure of a navigation interface



How to use UIStoryboad

```
//번들에서 특정 스토리보드 파일을 불러오는 경우
+ (UIStoryboard *)storyboardWithName:(NSString *)name
                              bundle:(NSBundle *)storyboardBundleOrNil


//스토리보드 객체에서 시작 지점의 뷰 컨트롤러 객체 만들기
- (__kindofUIViewController *)instantiateInitialViewController


//스토리보드 객체에서 특정 이름의 뷰 컨트롤러 객체 만들기
- (__kindofUIViewController *)instantiateViewControllerWithIdentifier:
(NSString *)identifier
```

미션

First Tab을 다음과 같이 완성하세요

FirstViewController를 UINavigationViewController에 embedded 시키세요

FirstViewController에 버튼을 하나 추가하고 DetailViewController를 push하세요

DetailViewController는 배경색을 노란색으로 칠하세요

DetailViewController에도 버튼을 하나 추가하고 Detail2ViewController를 push하세요

Detail2ViewController는 배경색을 파란색으로 칠하세요

View Controller Containers

Parent ViewController

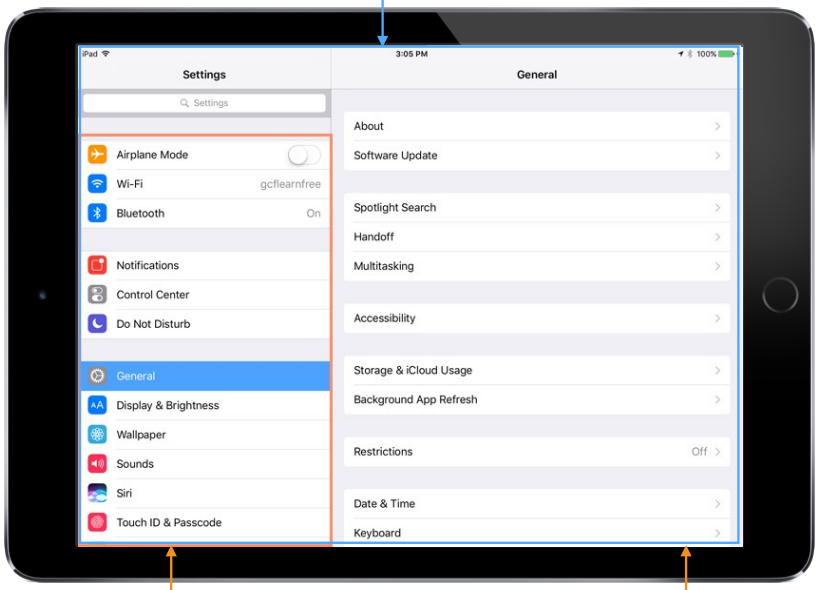
꼭 필요한가?

- What is the role of the container and what role do its children play?
- How many children are displayed simultaneously?
- What is the relationship (if any) between sibling view controllers?
- How are child view controllers added to or removed from the container?
- Can the size or position of the children change? Under what conditions do those changes occur?
- Does the container provide any decorative or navigation-related views of its own?
- What kind of communication is required between the container and its children? Does the container need to report specific events to its children other than the standard ones defined by the UIViewController class?
- Can the appearance of the container be configured in different ways? If so, how?

View Controller 기초 용어

root view controller 와 Master/Detail ViewController

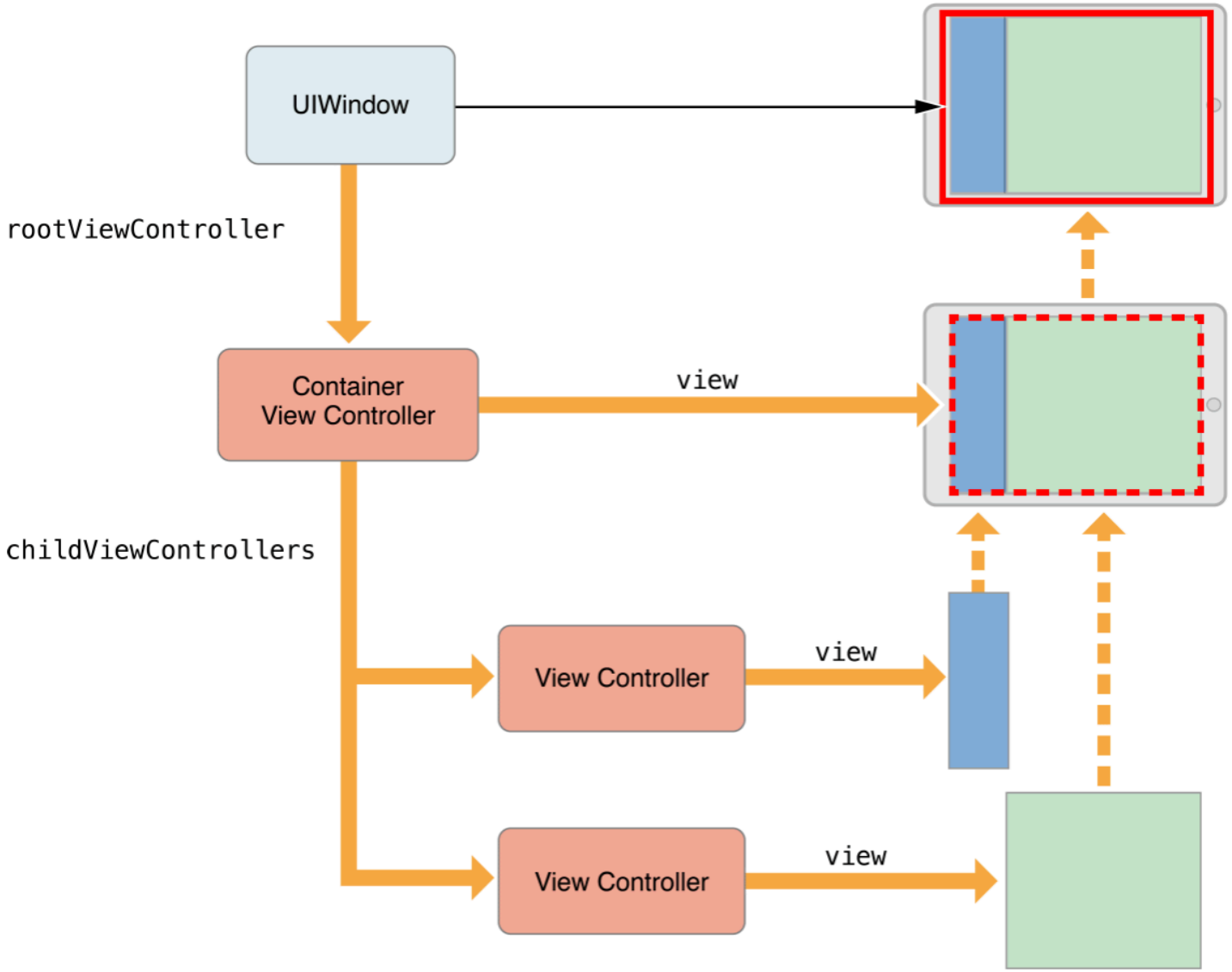
`window.rootViewController`



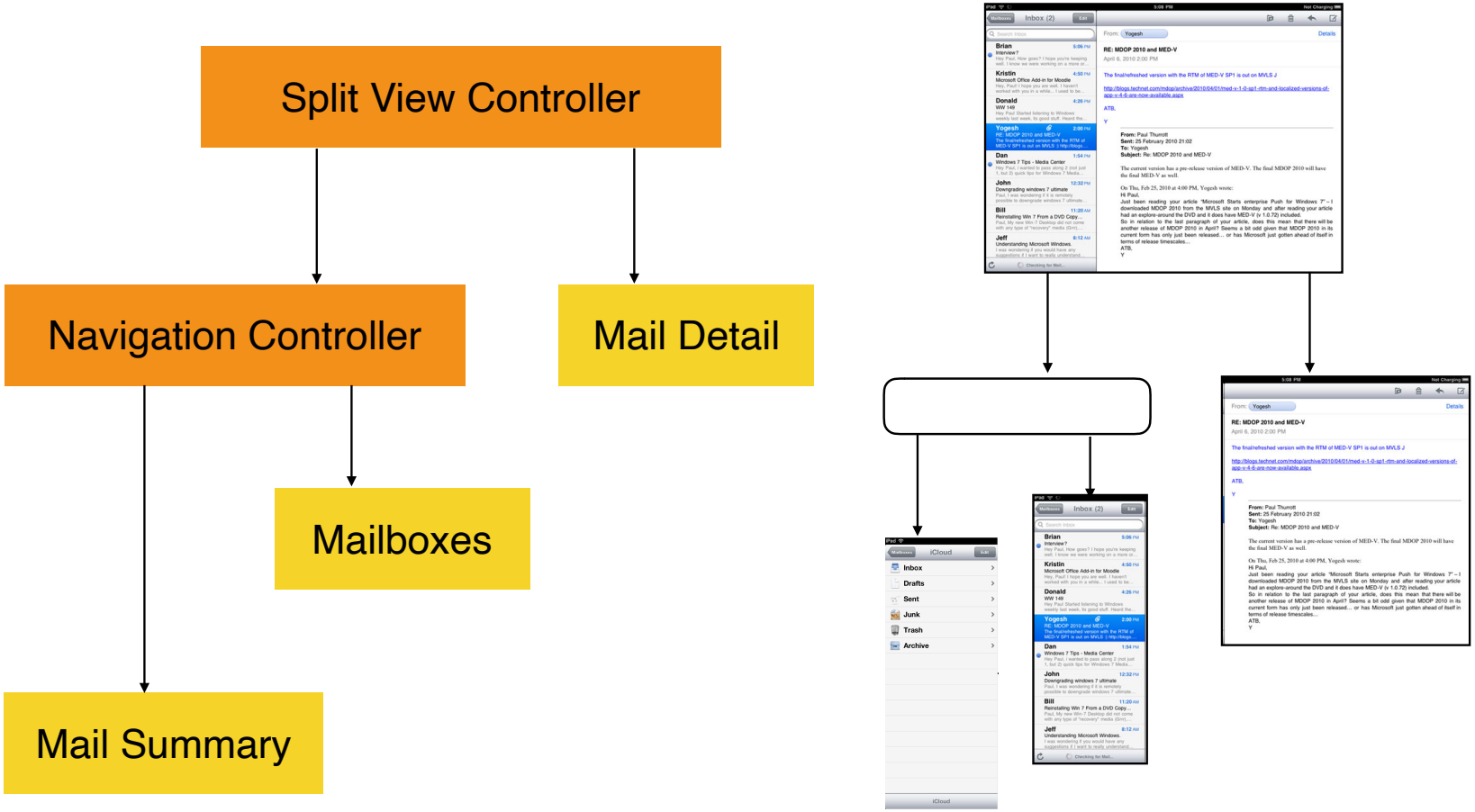
Master TableViewController

Detail TableViewController

Figure 2-2 A container acting as the root view controller



View Controller Container



View Controller Containers

컨트롤러 계층관련 API

하위 컨트롤러 추가/삭제

```
addChildViewController(_:)
removeFromParentViewController()
```

하위 컨트롤러 접근

```
var childViewControllers: [UIViewController]
```

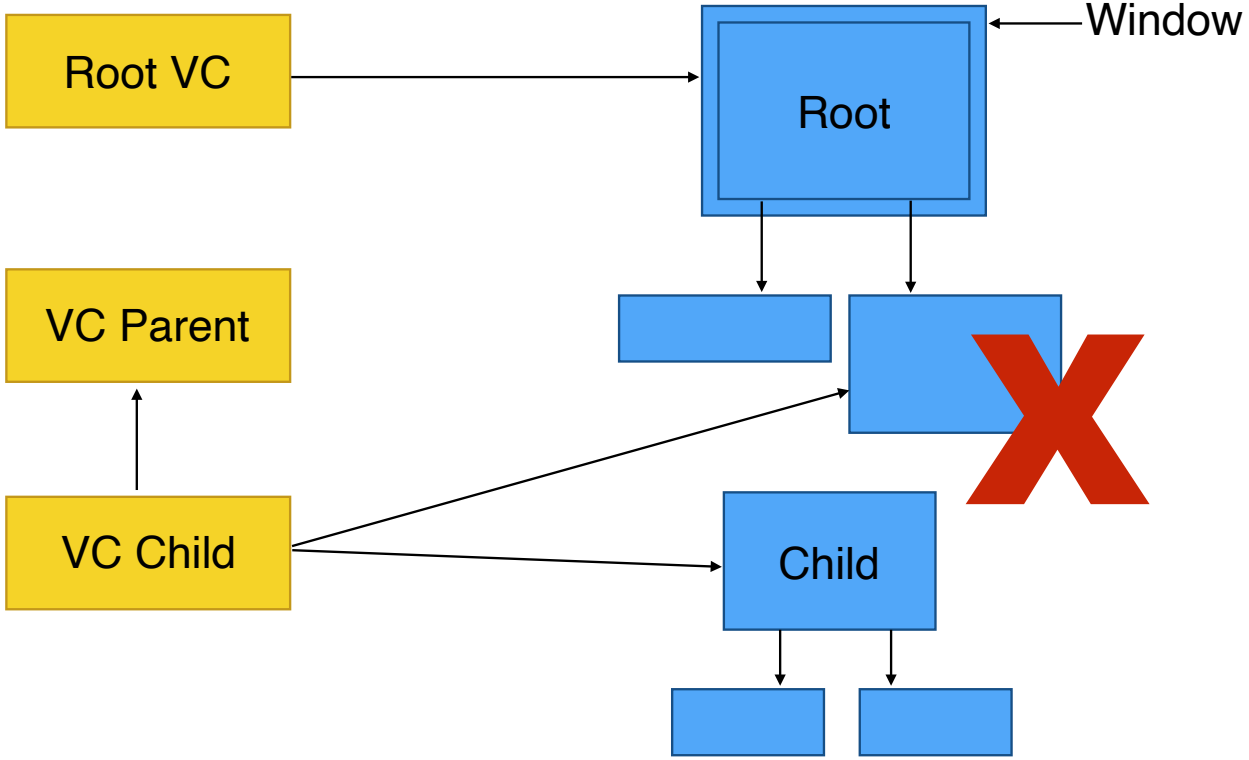
하위 콜백

```
willMove(toParentViewController:)
didMove(toParentViewController:)
```

View Controller Containers

VC 계층과 View 계층

```
root.view.addSubview(child.view)
```



View Controller Containers

VC 계층과 View 계층 추가할 때

1. 부모 컨트롤러에서 `addChildViewController(_:)` 호출
2. 자식 컨트롤러 root view를 컨테이너에 추가
3. 자식 view에 대해 레이아웃, 크기, 제약 변경
4. 자식 컨트롤러에서 `didMove(toParentViewController:)` 호출

```
- (void) displayContentController: (UIViewController*) content {
    [self addChildViewController:content];
    content.view.frame = [self frameForContentController];
    [self.view addSubview:self.currentClientView];
    [content didMoveToParentViewController:self];
}
```

View Controller Containers

VC 계층과 View 계층 제거할 때

1. 자식 컨트롤러에서 `willMove(toParentViewController:)` 호출
2. 자식 view에 대한 제약 제거
3. 자식 컨트롤러 root view를 컨테이너에서 제거
4. 자식 컨트롤러에서 `removeFromParenetViewController()` 호출

```
- (void) hideContentController: (UIViewController*) content {
    [content willMoveToParentViewController:nil];
    [content.view removeFromSuperview];
    [content removeFromParentViewController];
}
```

View Controller Containers

자식 컨트롤러 사이 전환

```
self.addChildViewController(note)

self.transition(from: recipe,
                to: note,
                duration: 3,
                options: .transitionFlipFromRight,
                animations: nil) { (finished) in
                    note.didMove(toParentViewController: self)
}
```

View Controller Containers

Appearance 관련 콜백

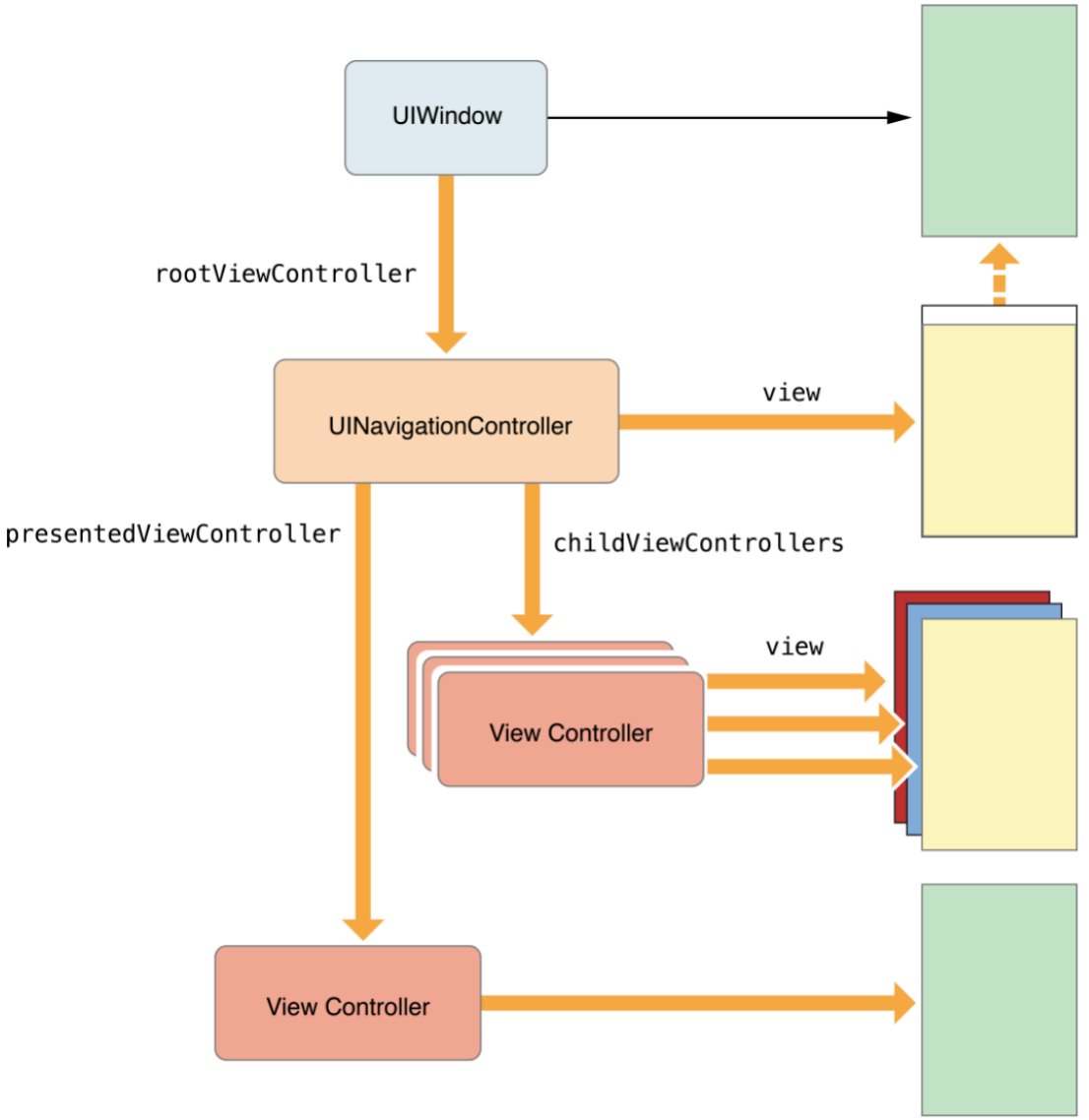
`viewWillAppear:`
- 윈도우의 뷰 계층에 새로운 뷰를 추가하기 직전
- (필요하면) vc.view.layoutSubviews() 호출하기 직전

`viewDidAppear:`
- 윈도우의 뷰 계층에 새로운 뷰를 추가한 직후
- (필요하면) vc.view.layoutSubviews() 호출한 직후

`viewWillDisappear:`
- 윈도우의 뷰 계층에서 특정 뷰를 제거하기 직전

`viewDidDisappear:`
- 윈도우의 뷰 계층에 특정 뷰를 제거한 직후

Figure 2-4 A container and a presented view controller



System View Controllers

Built-in UIKit View Controllers

System View Controllers

* Image Picker
* Video Editor
* Document Browser & Previewing
* iCloud Sharing
* Shared Activities
* Printer Picker
* Word Lookup
* Extra…

뷰컨트롤러 추가 미션

UIImagePickerController

미션 내용

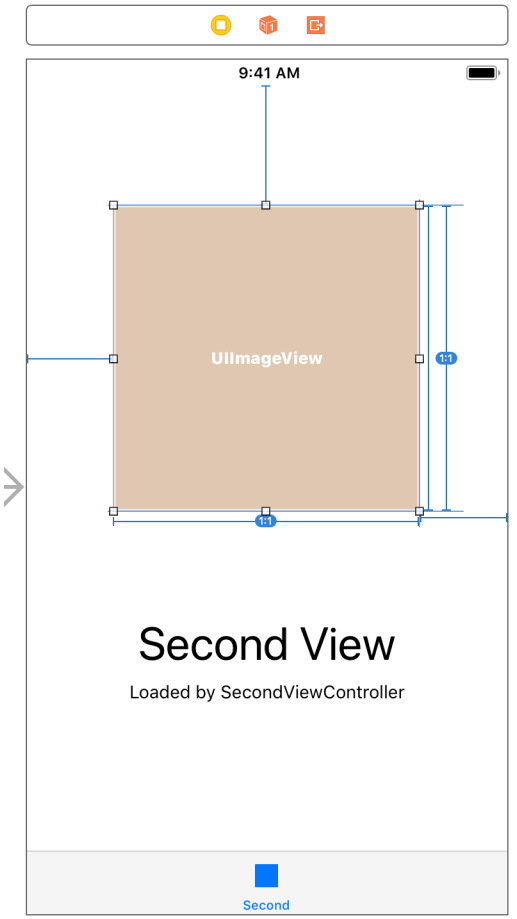
Second Tab을 <프로파일 화면>역할로 다음과 같이 완성하세요

SecondViewController에 [profileImageView] 이미지뷰를 추가하고,

터치하면 UIImagePickerController를 띄워보세요.

SourceType은 photoLibrary로 지정하고, 사진을 선택하면 해당 사진으로 [profileImageView]를 바꿔보세요.

https://developer.apple.com/documentation/uikit/uiimagepickercontroller



미션 내용

Second Tab을 <프로파일 화면>역할로 다음과 같이 완성하세요

[ProfileImageView] 모서리를 둥글게 혹은 아예 동그랗게 만들어보세요.

생각꺼리

- 지정한 이미지를 삭제하려면 어떤 작업이 필요할까요?
- 선택한 이미지를 전달 받으려면 어떤 작업이 필요할까요?
- 취소하고 그냥 빠져나오고 싶으면?

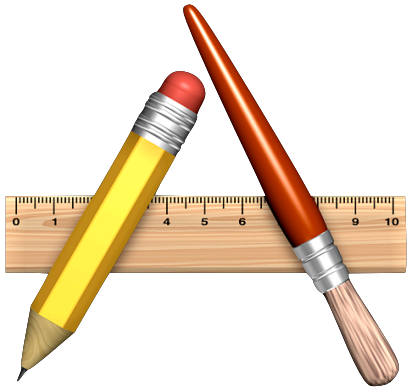
델리게이션
Delegate

* 서브 클래싱 없이 나만의 동작을 구현
* 데이터 제공
* 사용자 인터페이스 조작 가능
* 라이프사이클 노티피케이션

cf. 단순 Callback 함수와의 차이점

앱 델리게이션

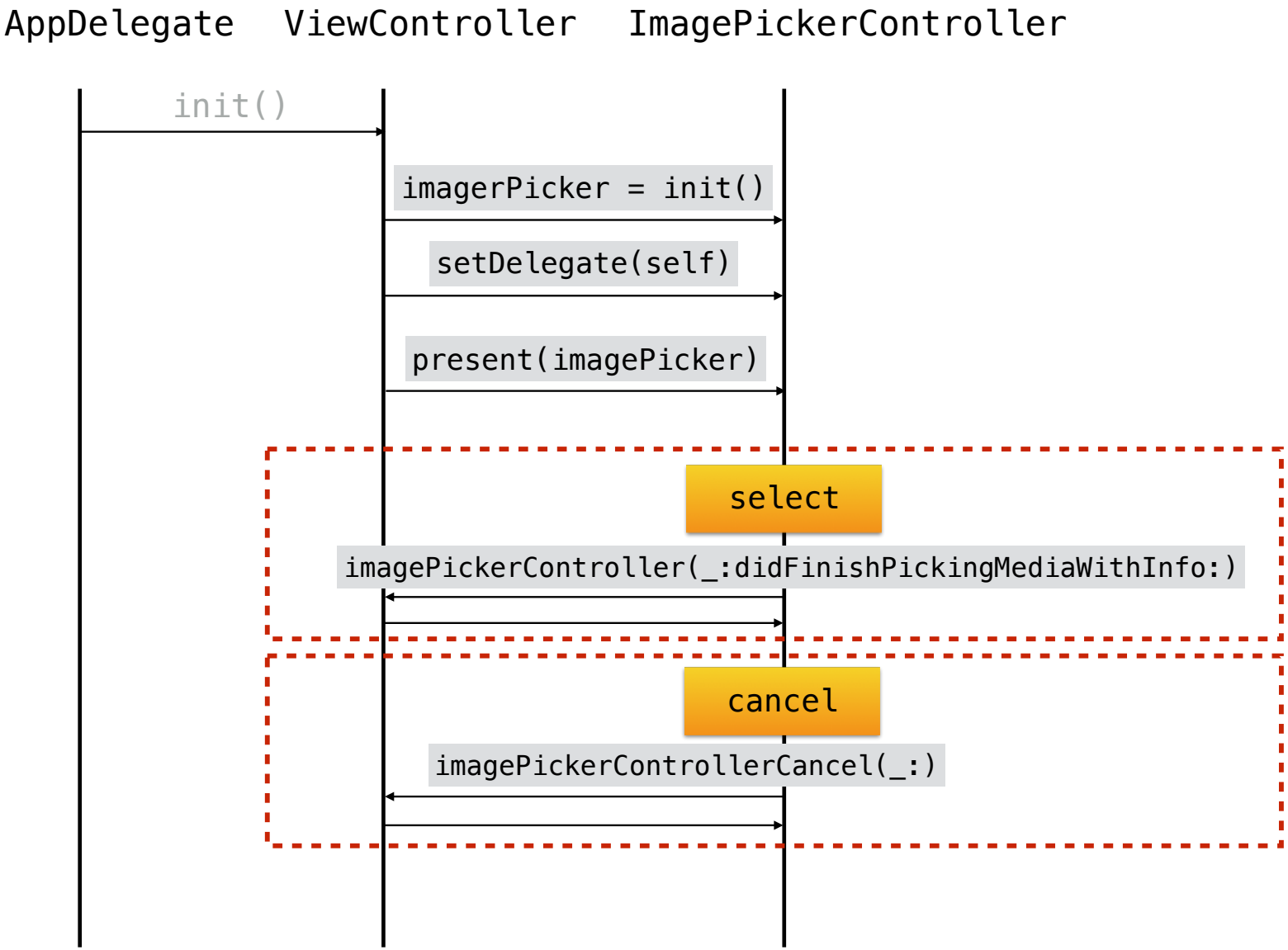
AppDelegate



`UIApplicationDelegate`

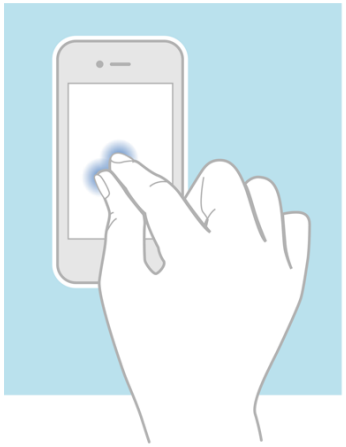
앱 상태 변화에 대한 callback 함수들 모음

```
application(_:didFinishLaunchingWithOptions:)
applicationDidEnterBackground(_:)
applicationWillTeminate(_:)
application(_:shouldSaveApplicationState:)
```

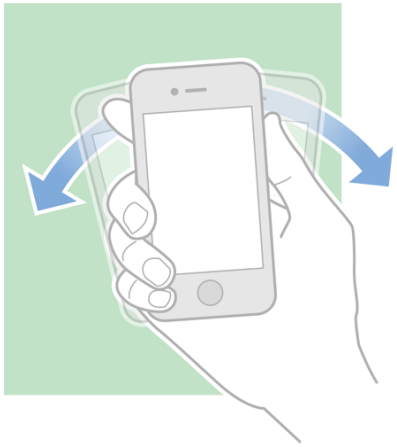

AppDelegate
ViewController
ImagePickerController
init()
imagerPicker = init()
setDelegate(self)
present(imagePicker)
select
imagePickerController(_:didFinishPickingMediaWithInfo:)
cancel
imagePickerControllerCancel(_:)

Multi-Touch Event

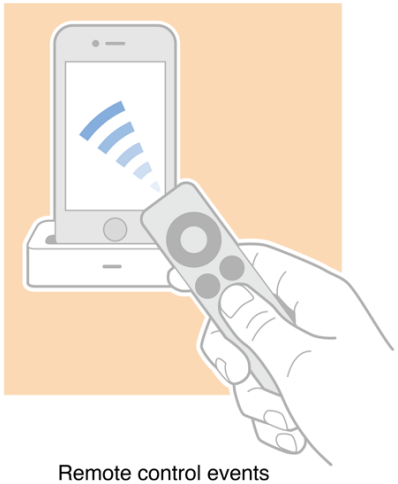
UIEvent



Multitouch events



Accelerometer events

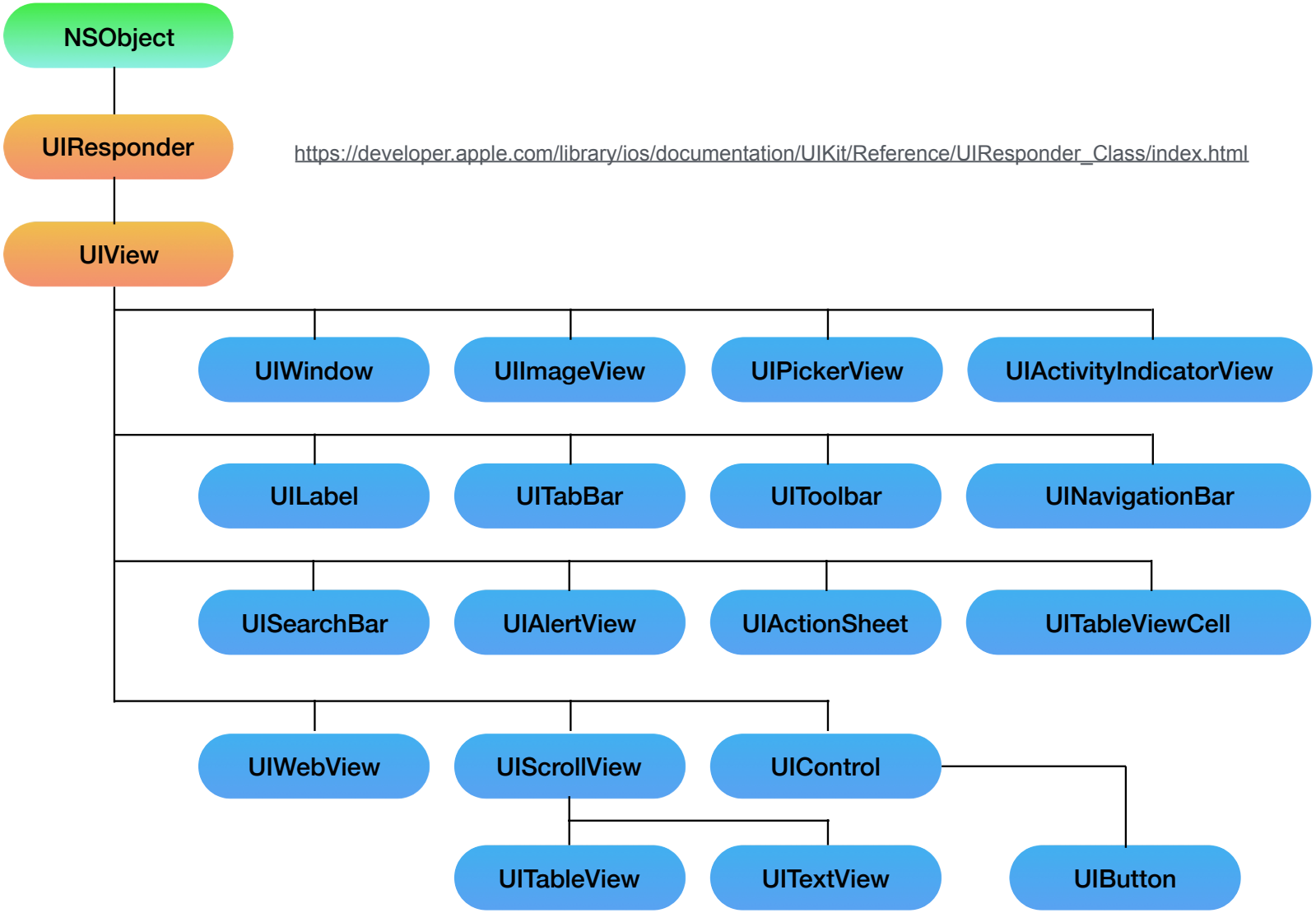


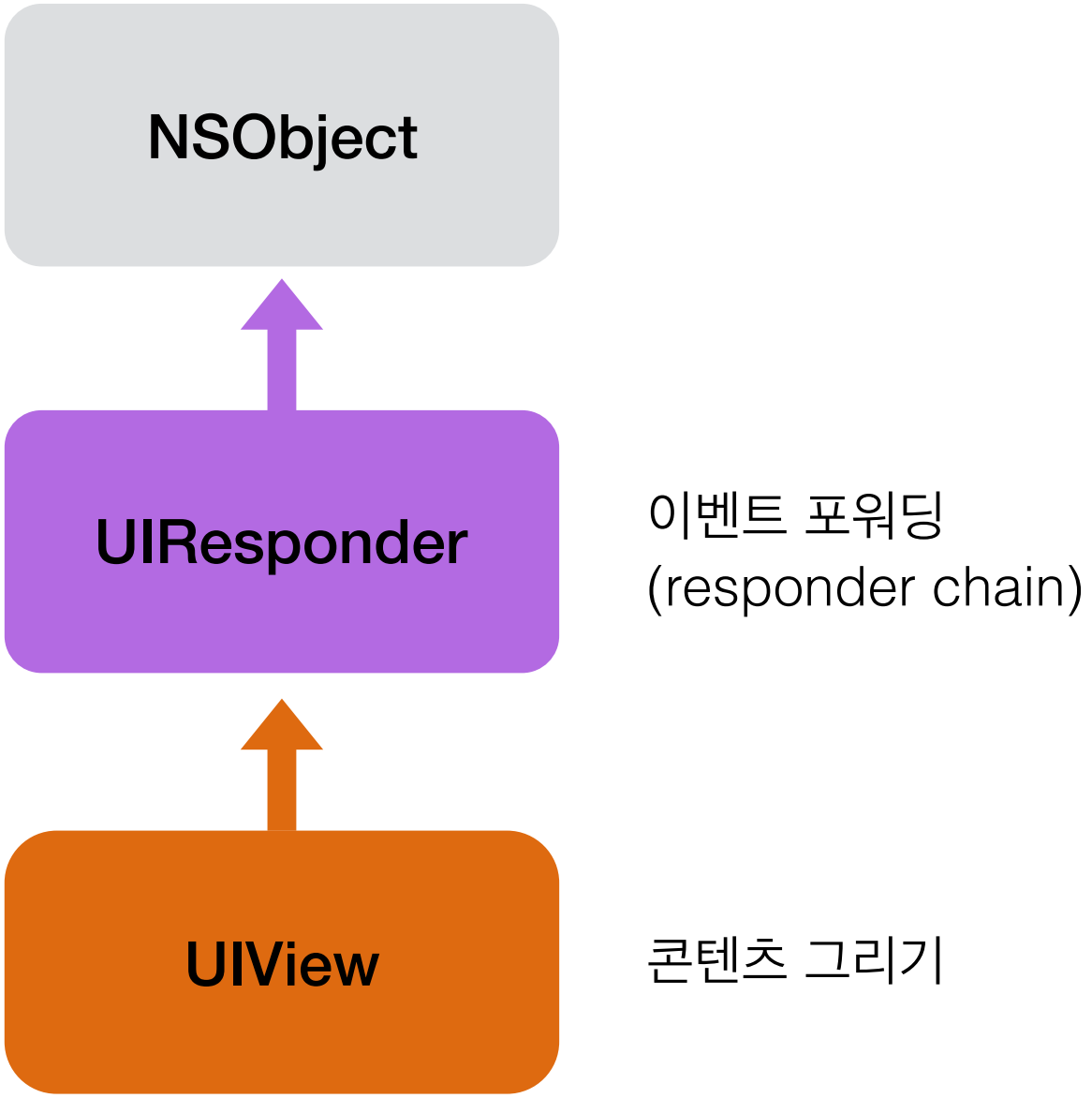
Remote control events

- Event Type, sub-type, timestamp…
- UIEventType

  UIEventTypeTouches, UIEventTypeMotion, UIEventTypeRemoteControl

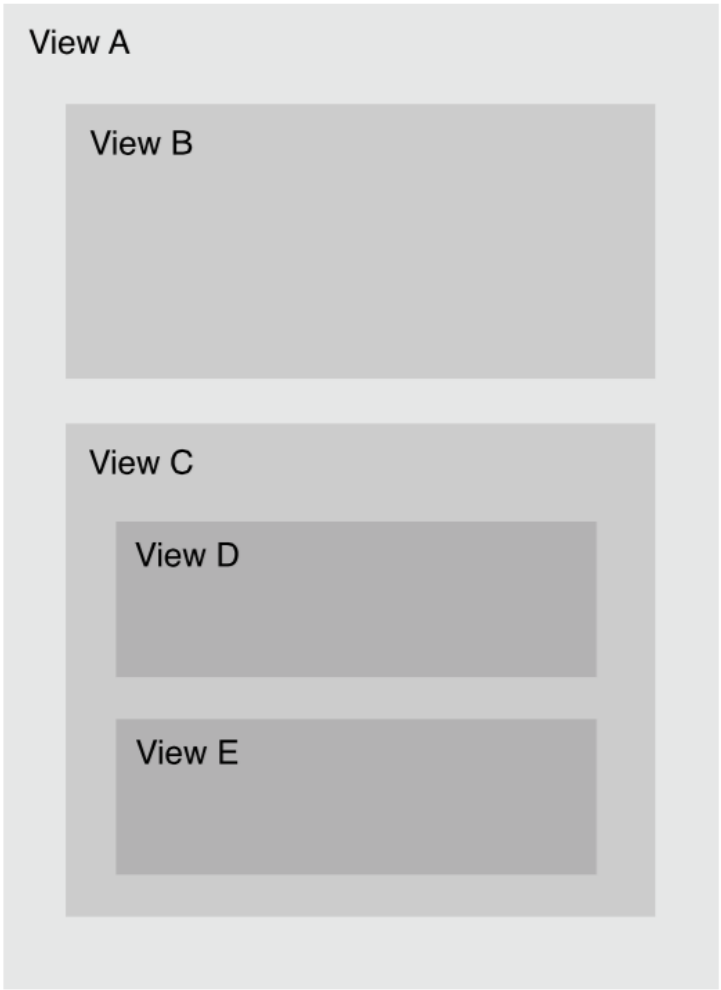
View Class Hierarchy




NSObject
UIResponder
이벤트 포워딩
(responder chain)
UIView
콘텐츠 그리기

Hit-Test

https://developer.apple.com/documentation/uikit/uiview/1622469-hittest



Single Touch 처리하기

터치가 눌리면...

UIApplication 객체가 하는 특별한 동작

- 뷰 계층 중에서 가장 깊이
- 터치가 눌린 뷰를 찾음
- First Responder를 사용안함

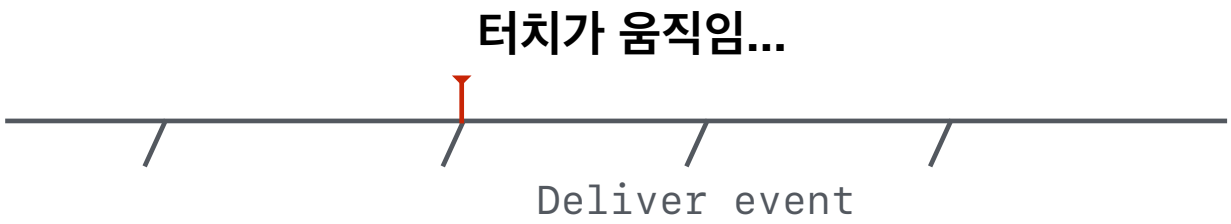
터치 이벤트와 뷰는 연결됨 (다시 만들수 없음)

Single Touch 처리하기

이벤트 전달

- UIApplication/UIWindow `-sendEvent:`
- hit-test view가 이벤트 받음
- `touchesBegan(_:with:)` 메서드

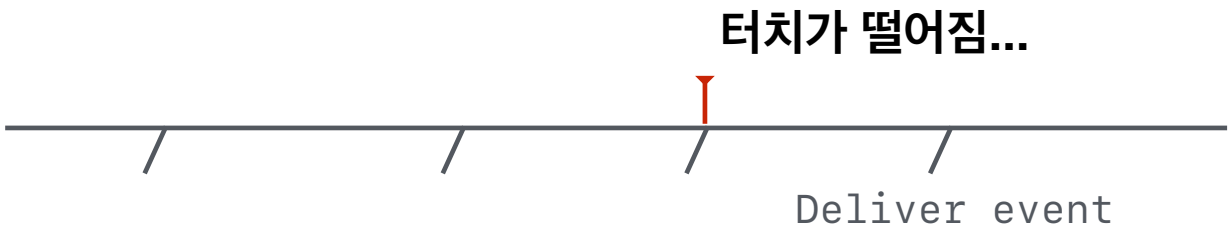
Single Touch 처리하기



동일한 방식으로 이벤트 전달

- `-sendEvent:` 실행
- hit-test view가 이벤트 받음
- `touchesMoved(_:with:)` 메서드

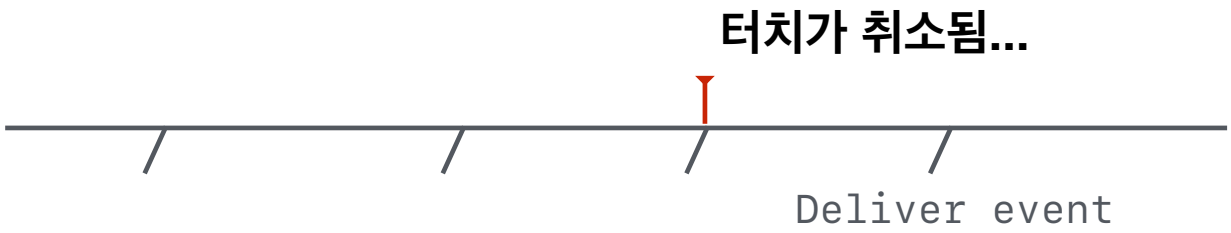
Single Touch 처리하기



동일한 방식으로 이벤트 전달

- `-sendEvent:` 실행
- hit-test view가 이벤트 받음
- `touchesEnded(_:with:)` 메서드

Single Touch 처리하기



동일한 방식으로 이벤트 전달

- `-sendEvent:` 실행
- hit-test view가 이벤트 받음
- `touchesCancelled(_:with:)` 메서드

Single Touch 처리하기

요약

- 터치가 시작할 때 hit-test view를 찾음
- hit-test view가 모든 메서드를 받음

Touch Event Handler

- (void)touchesBegan:(NSSet *)touches withEvent(UIEvent *)event;

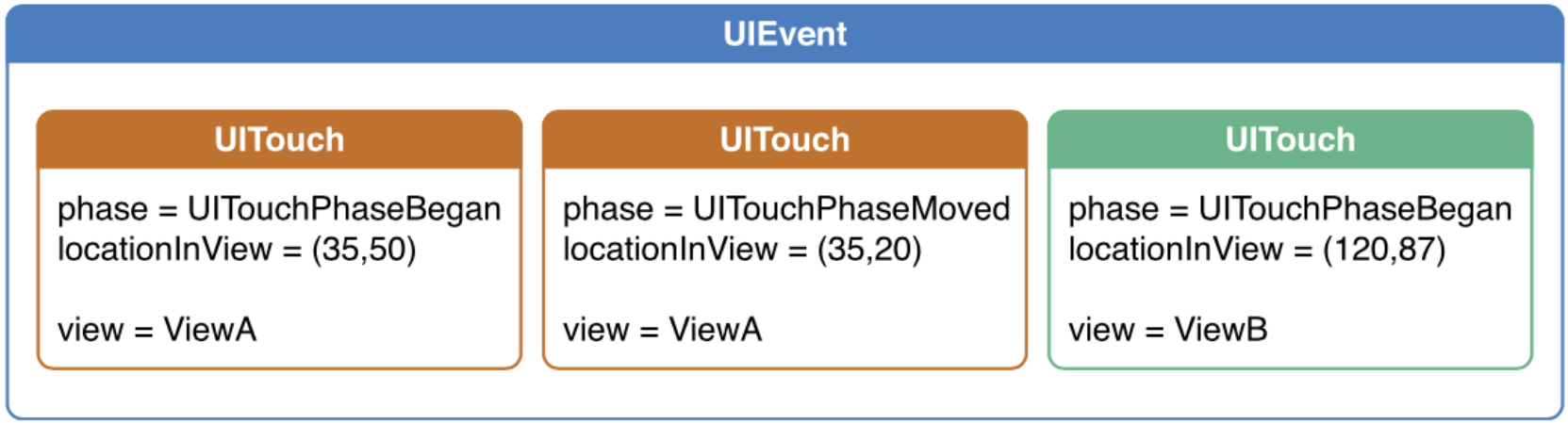
- (void)touchesMoved:(NSSet *)touches withEvent(UIEvent *)event;

- (void)touchesEnded:(NSSet *)touches withEvent(UIEvent *)event;

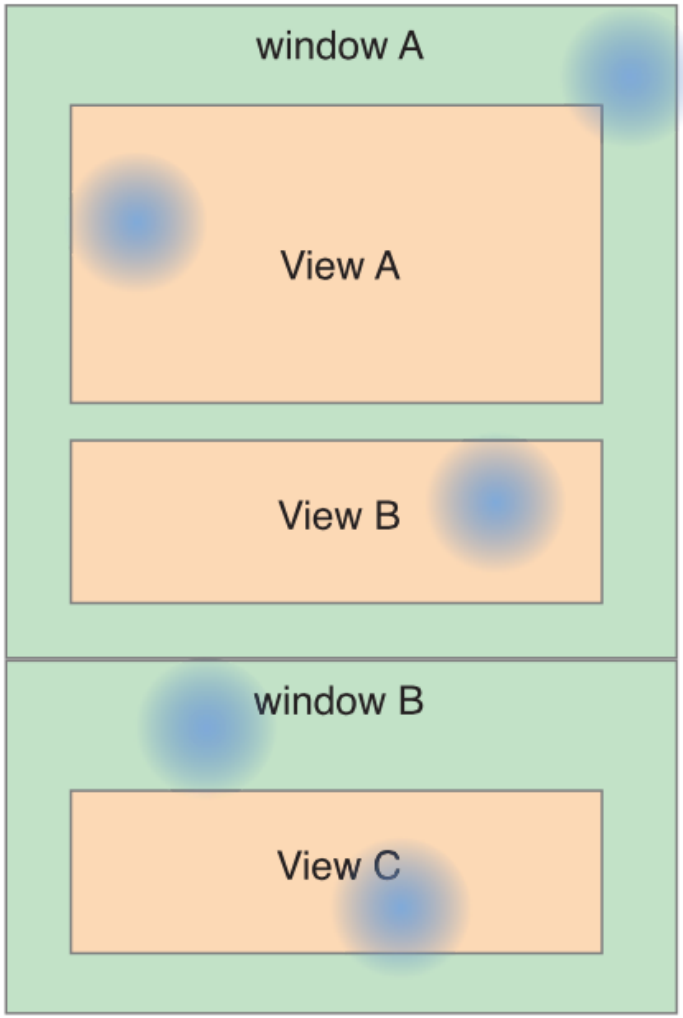
- (void)touchesCancelled:(NSSet *)touches withEvent(UIEvent *)event;

UITouch objects

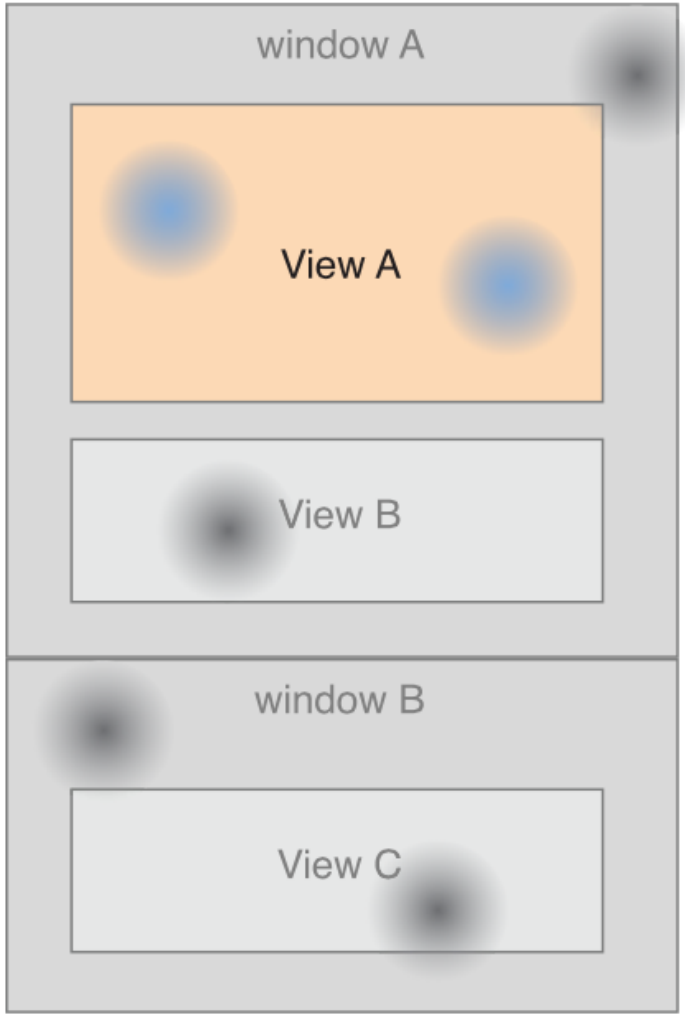
Relationship of a UIEvent object and its UITouch objects



-allTouches



-touchesForView:



나만의 뷰 추가하기

커스컴 UIView 만들기

코드스쿼드

미션 내용

Third Tab을 추가하고 다음과 같이 완성하세요

새로운 뷰 컨트롤러를 추가하고 DrawViewController로 지정하세요.

UIView를 상속받는 DrawView 클래스를 생성하세요.

Scene의 기본 View를 선택하고, Custom Class로 DrawView를 지정하세요.

힌트: UIView 상속받아 DrawView 커스텀 뷰 클래스를 만드세요. (ViewController와 상관없이 뷰를 확장)

touchBegan 좌표부터 touchMoved 좌표들을 Array에 기록하세요. touchEnded 에서 기록한 좌표들을 차례로 이어서 선을 그리세요.

선의 굵기와 선 색상을 지정하세요.

-drawRect: 메서드에서 한꺼번에 그립니다.

다시 touchBegan이 시작되면 이건 그림은 지우세요.