iOS 프로그래밍

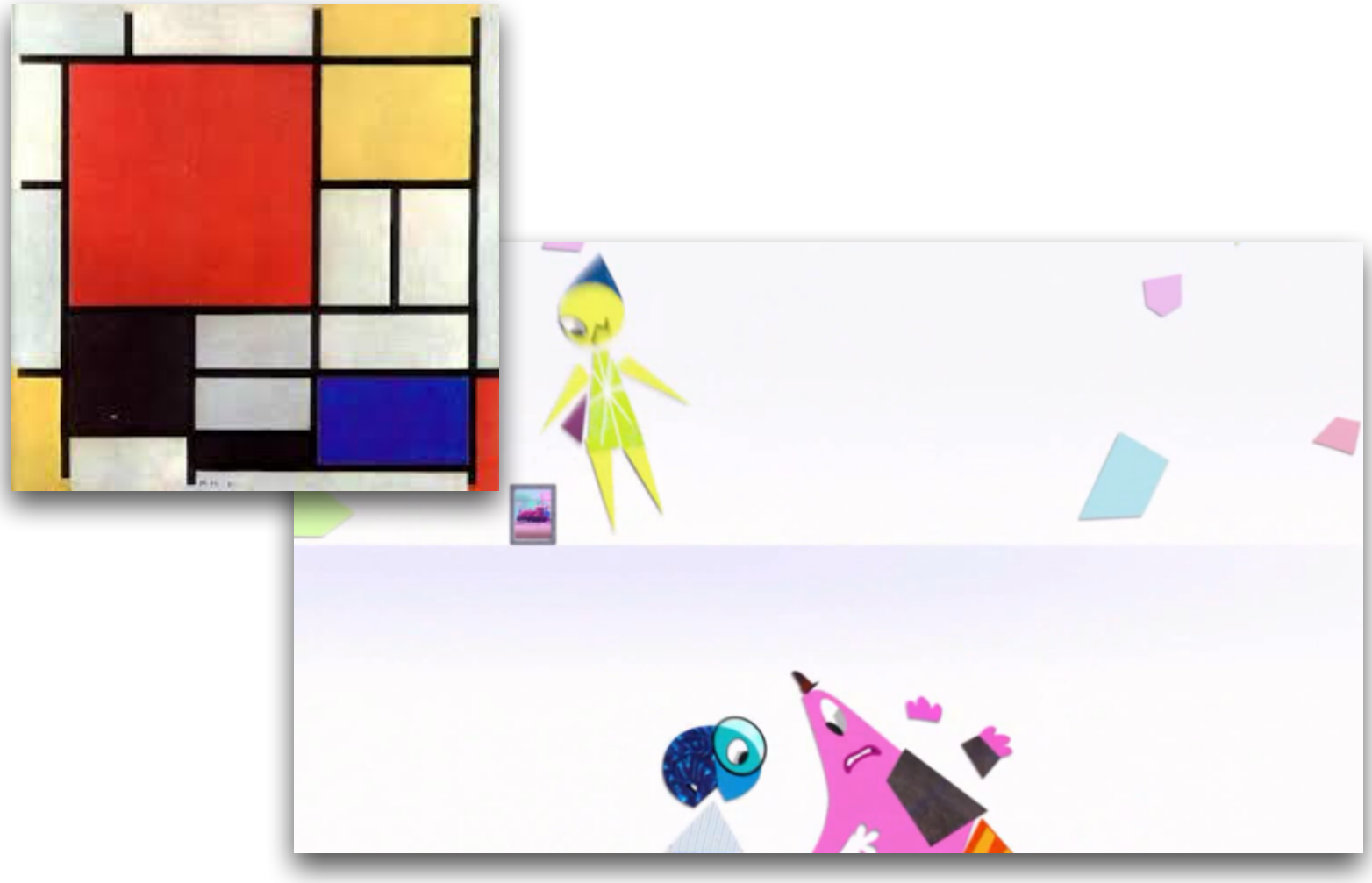
View Programming + AutoLayout

코드스쿼드

학습 목표

* 복습하기
* iOS를 위한 MVC
* Xib 와 Storyboard
* 뷰 객체와 좌표 시스템
* 뷰 레이아웃
* 오토레이아웃 이해하기

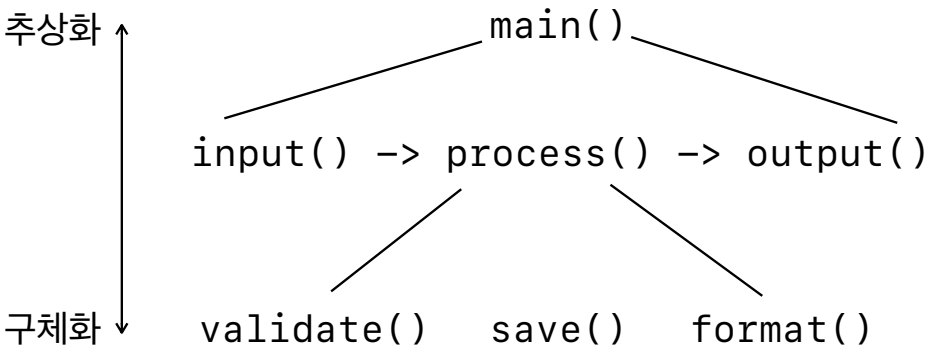
추상화 Abstraction



추상화 메커니즘

프로시저 추상화 (procedure abstraction)
- 무엇을 해야 하는지 추상화

기능 분해(Functional decomposition)
알고리즘 분해 (Algorithmic decomposition)



top-down 하향식 분해는
설계가 어느 정도 안정화된 이후에
설계를 설명하고 문서화하기 편리함

바바라 리스코프

데이터 추상화 (data abstraction)
- 무엇을 알아야 하는지 추상화

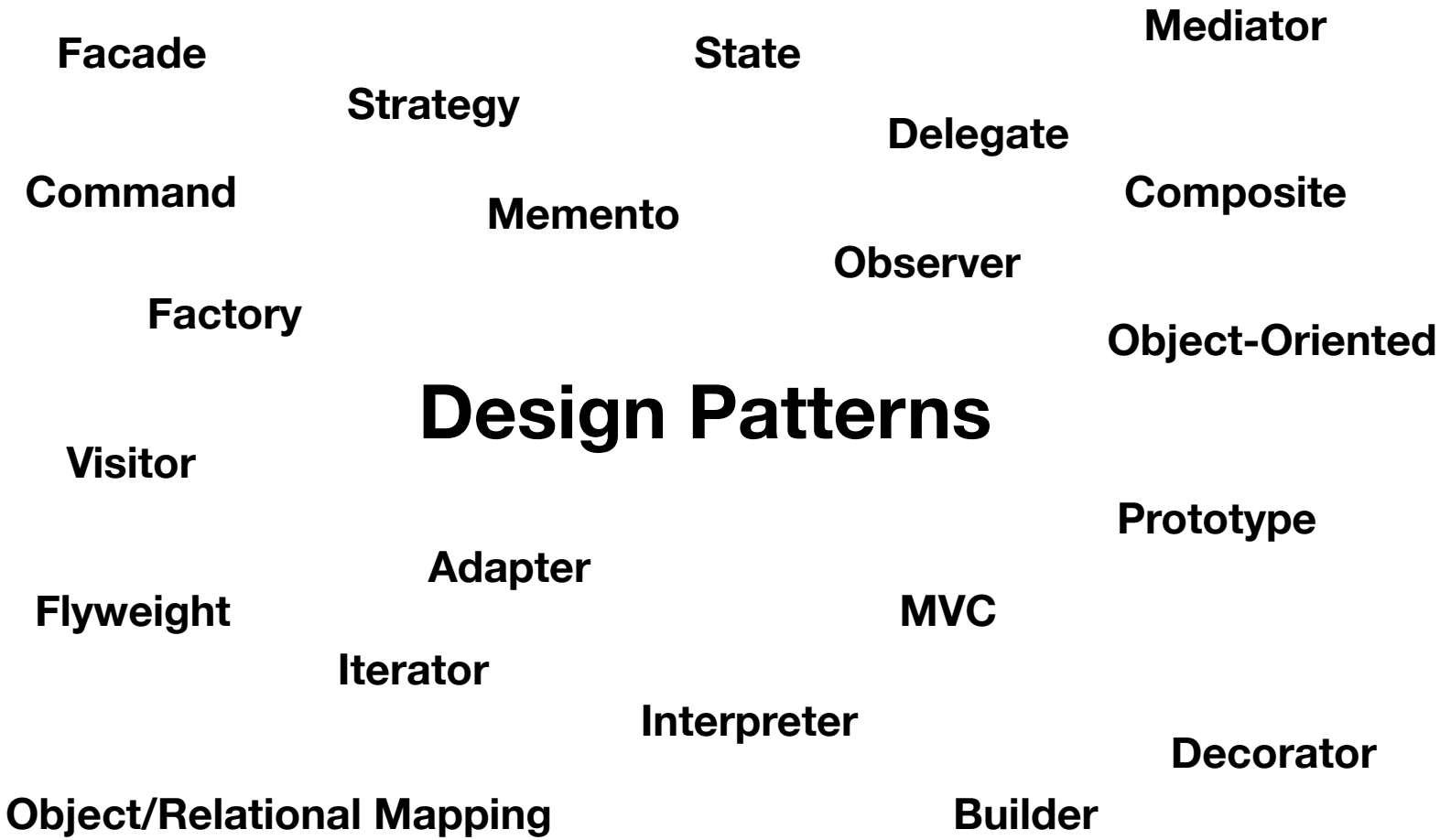
타입을 추상화(type abstraction) ➔ Abstract Data Type

- 타입 정의를 선언할 수 있어야 함
- 타입 인스턴스를 다루는 오퍼레이션 집합을 정의할 수 있어야 함
- 제공하는 오퍼레이션으로 조작하도록 보호할 수 있어야 함
- 타입에 대한 여러 인스턴스를 생성할 수 있어야 함

데이터 중심 프로시저 추상화 ➔ Object-oriented

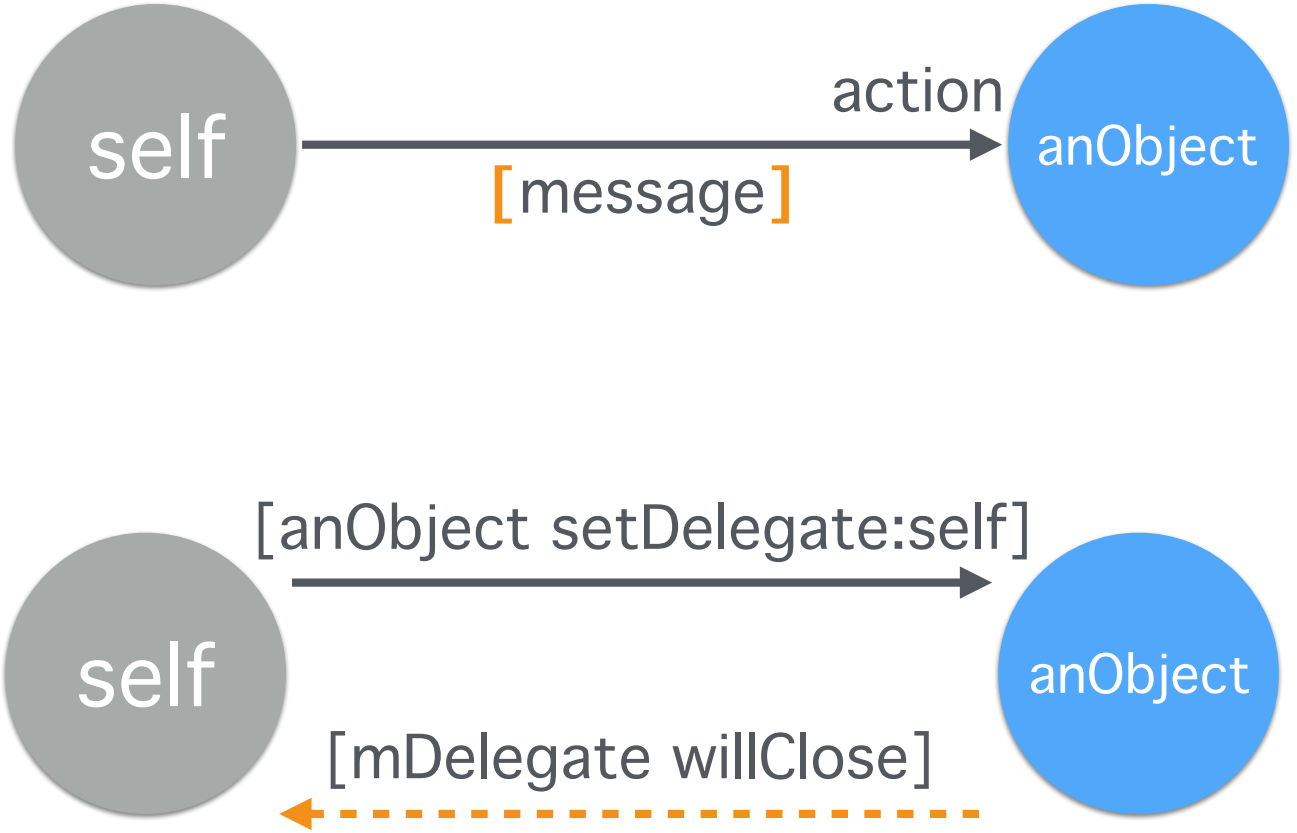
- 상속과 다형성 지원

앨런 케이


Facade
State
Mediator
Strategy
Delegate
Command
Composite
Memento
Observer
Factory
Object-Oriented
Design Patterns
Visitor
Prototype
Adapter
Flyweight
MVC
Iterator
Interpreter
Decorator
Object/Relational Mapping
Builder

Delegate

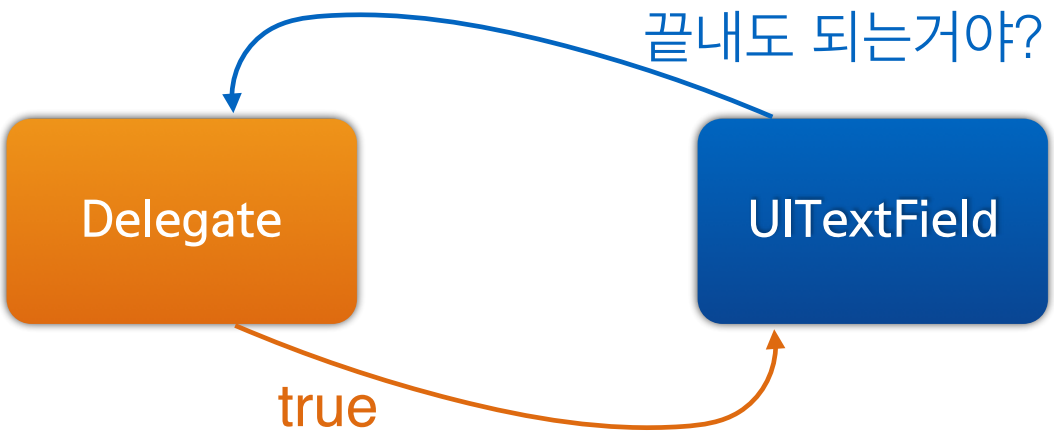
Don’t call me, I’ll call you



Delegation

Reusing controls without subclassing

사용자가 return 키를 누른 경우



`UITextFieldDelegate`

Delegation in UIKit

여러 클래스가 델리게이트를 갖고 있음

will / did / should

Delegation in UIKit

UIApplicationDelegate

“ 이제 곧 active가 아닐꺼야 ”

```
- (void)applicationWillResignActive:
```

Delegation in UIKit

UIScrollViewDelegate

“ 직전에 줌이 됐어 ”

```
- (void)scrollview_Did_Zoom:
```

Delegation in UIKit

UITextFieldDelegate

“ 콘텐츠를 깨끗하게 지워도 되? ”

```
- (BOOL)textField*Should*Clear:
```

iOS를 위한 MVC

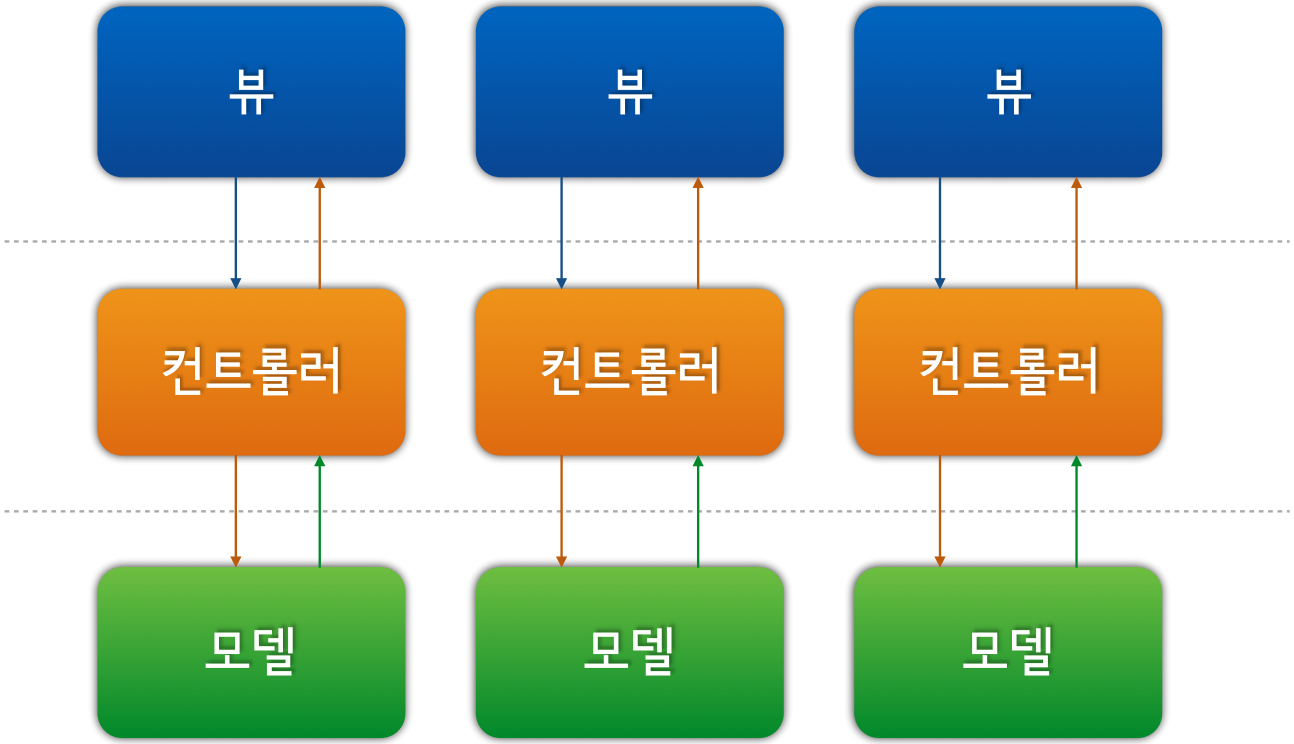
앱과 컨트롤러

화면당 ViewController 하나일까?

모든걸 다하는 컨드롤러는 만들지마라

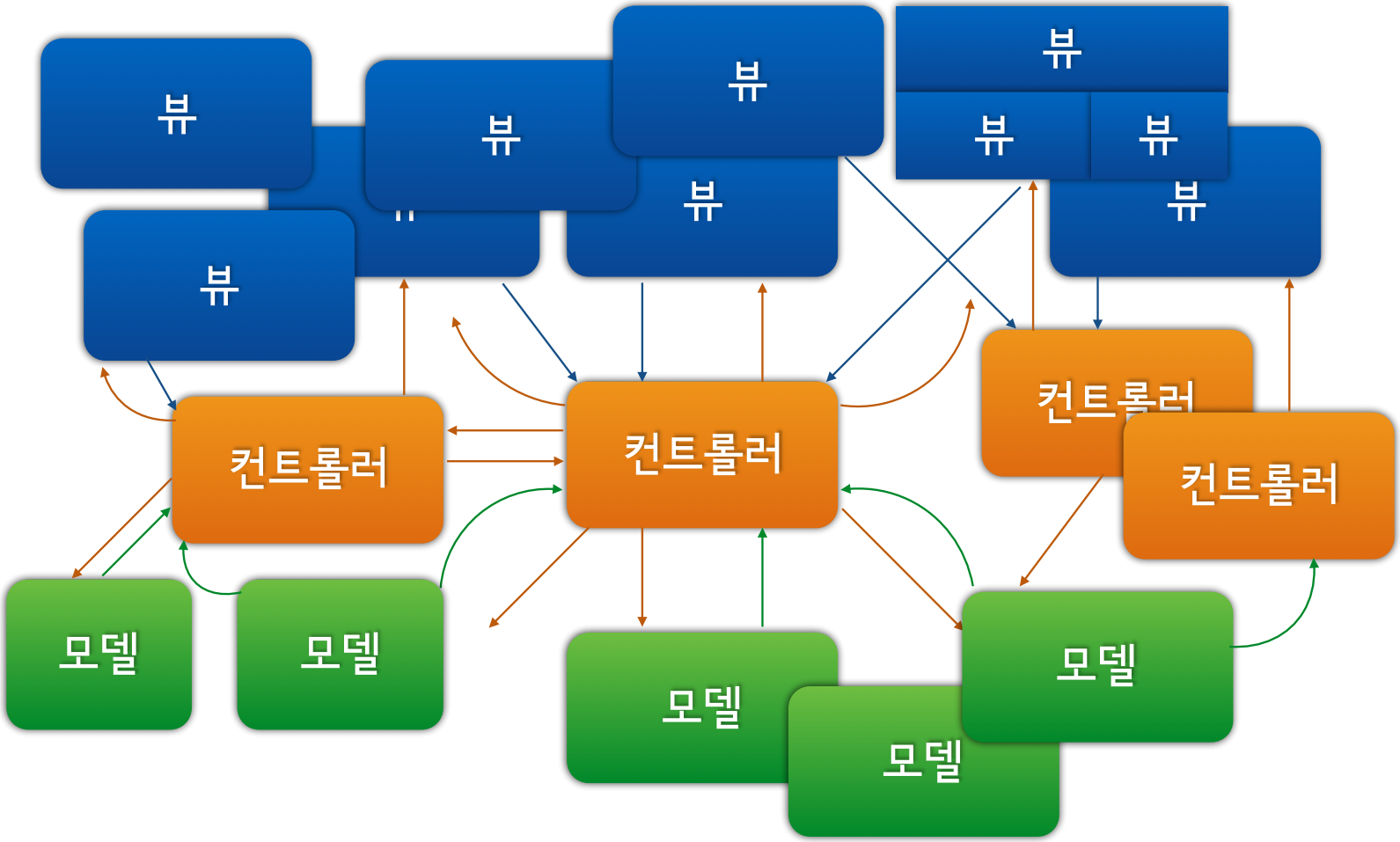
화면이 큰 아이패드에서도

ViewController는 역할이 나뉘어져야 한다



MVC 아키텍처

실제 상황에서는?



느슨한 연결(loose Coupling) 지향

유연성을 확보하라

X 메시지을 보낼 때 MVC 계층을 건너띄지마
- 컨트롤러로 메시지를 조정할 수 있음

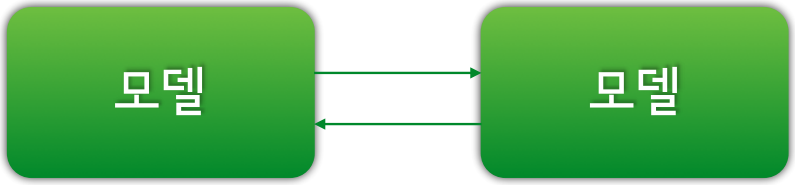
X 하나의 오브젝트에 MVC 역할을 섞지마
- 하나의 객체가 너무 많을 일을 하도록 하지말 것

X 뷰 클래스에서 모델 데이터를 선언하지마
- 중복 데이터는 피해야 함

메시지 관리

유연성 확보

X 모델끼리 직접적인 양방향
데이터 전송은 하지마라

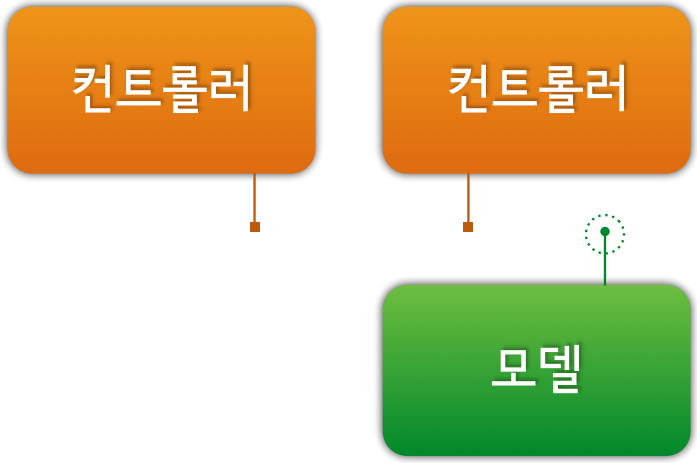


메시지 관리

유연성 확보

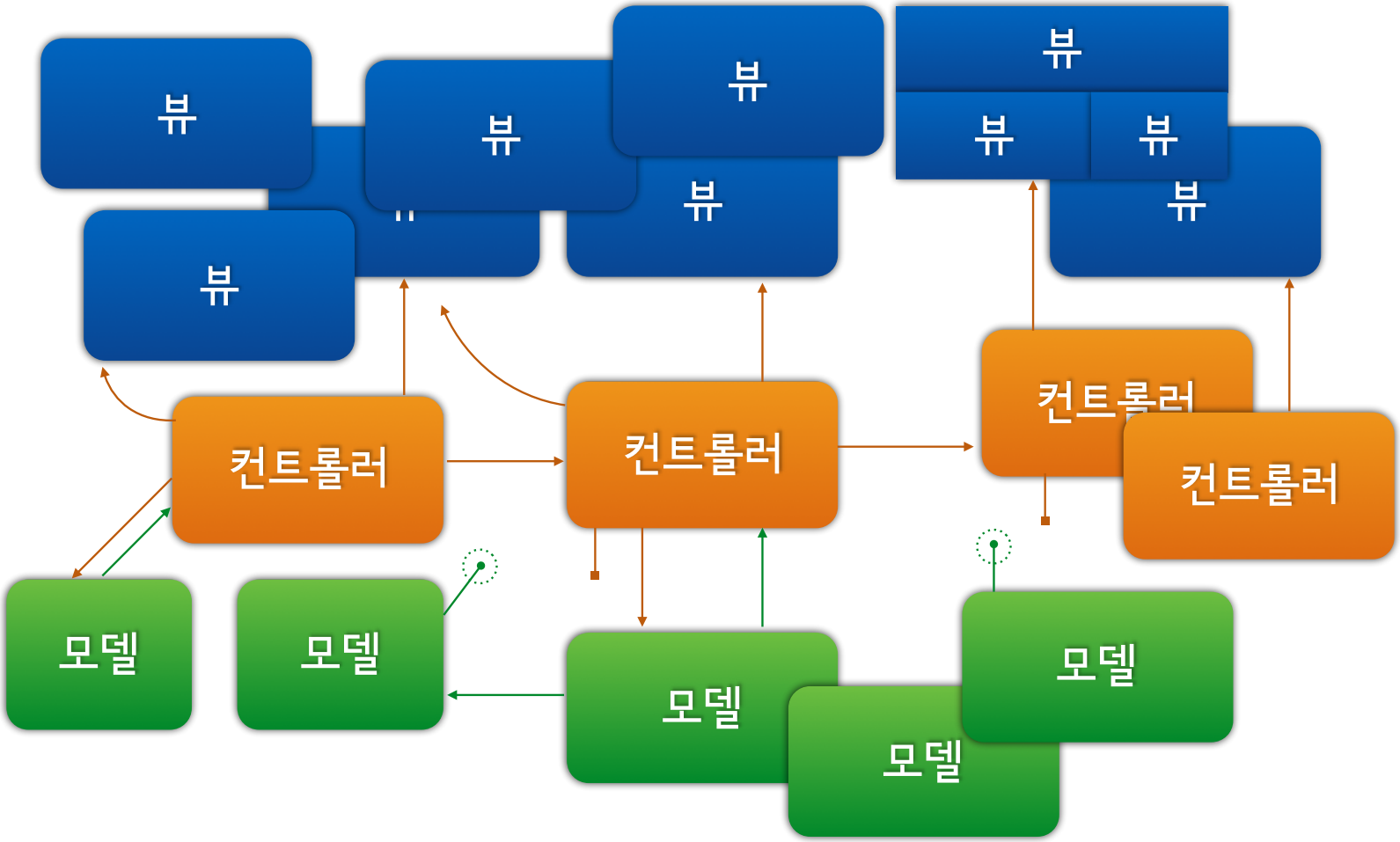
동시에 여러 화면을 업데이트할 때

Observer Pattern



MVC 아키텍처

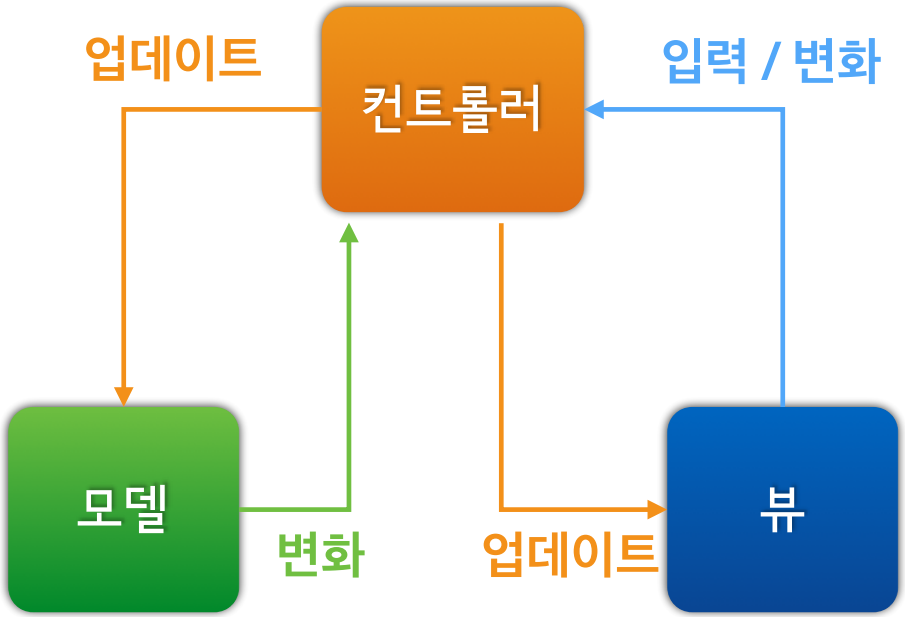
실제 상황에서는?



Model - View - Controller

업데이트 관리!

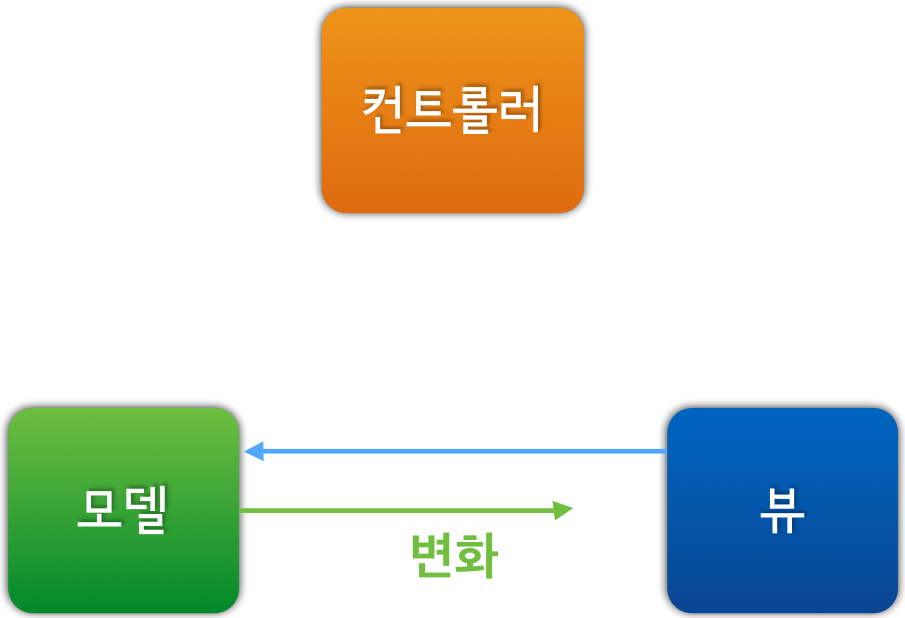
Controller : Coordination



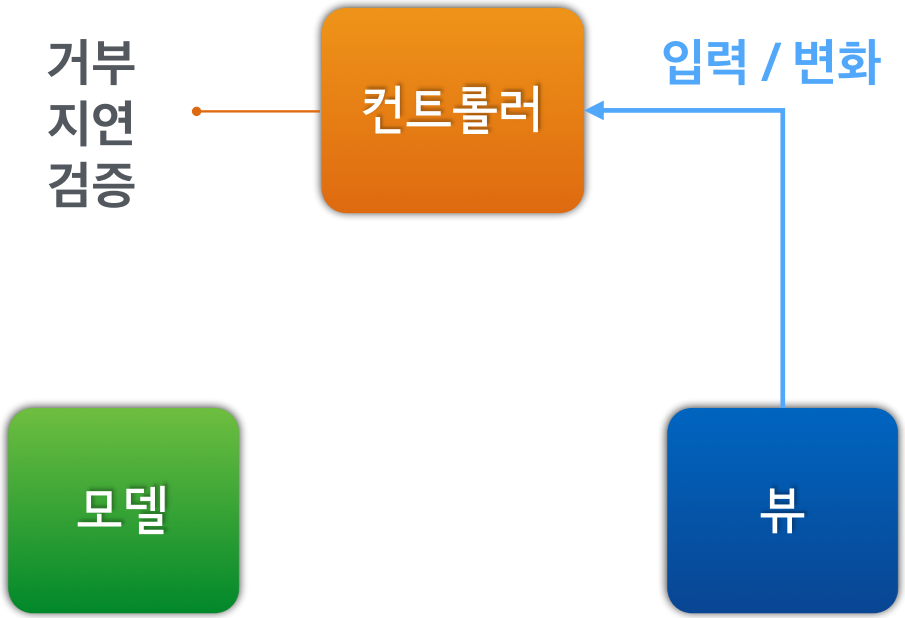
Model : Data

View : Display

X 컨트롤러를 건너띄지 마라



변화를 대처하는 올바른 방법

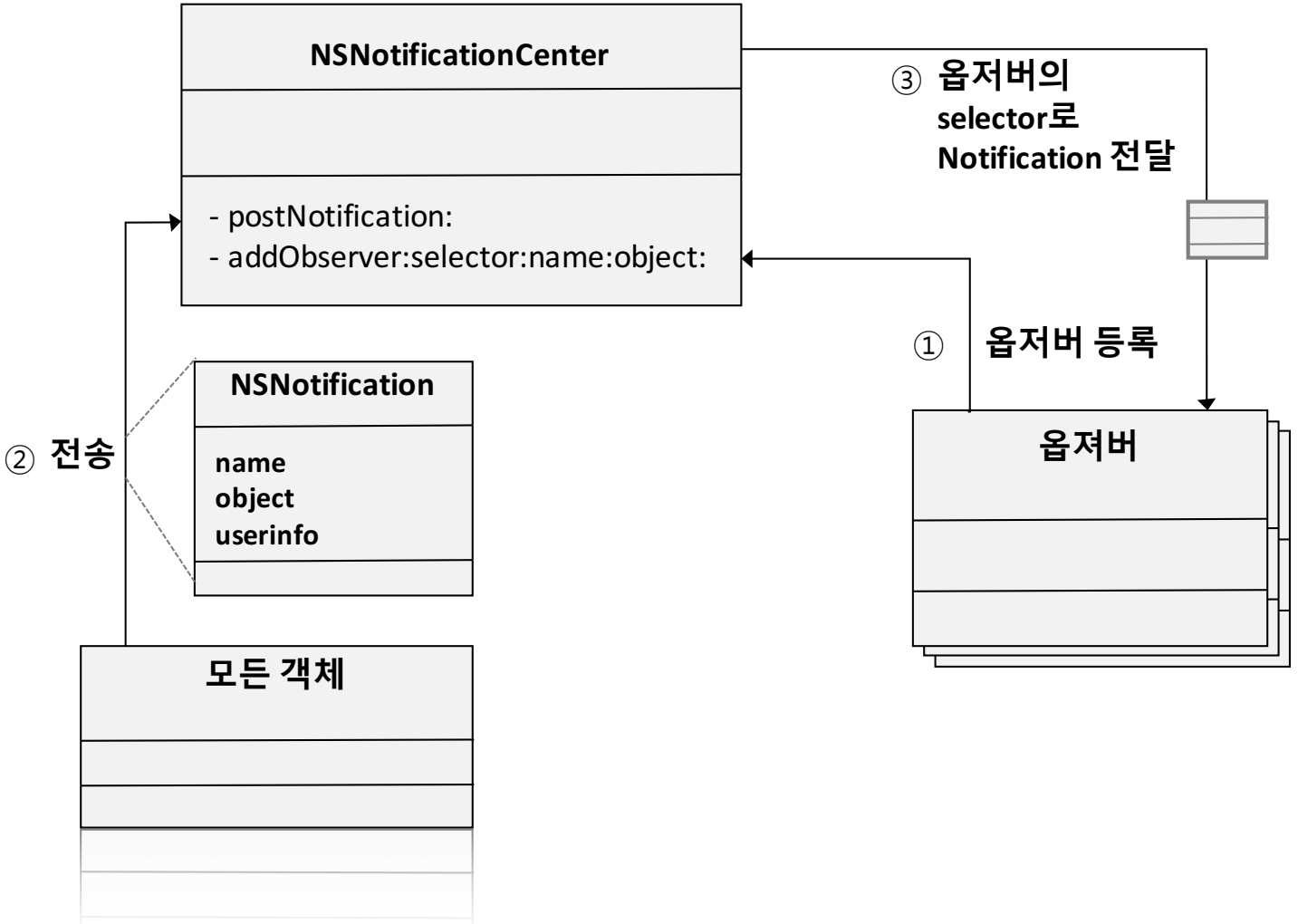


✓ 변화를 대처하는 올바른 방법

NotificationCenter

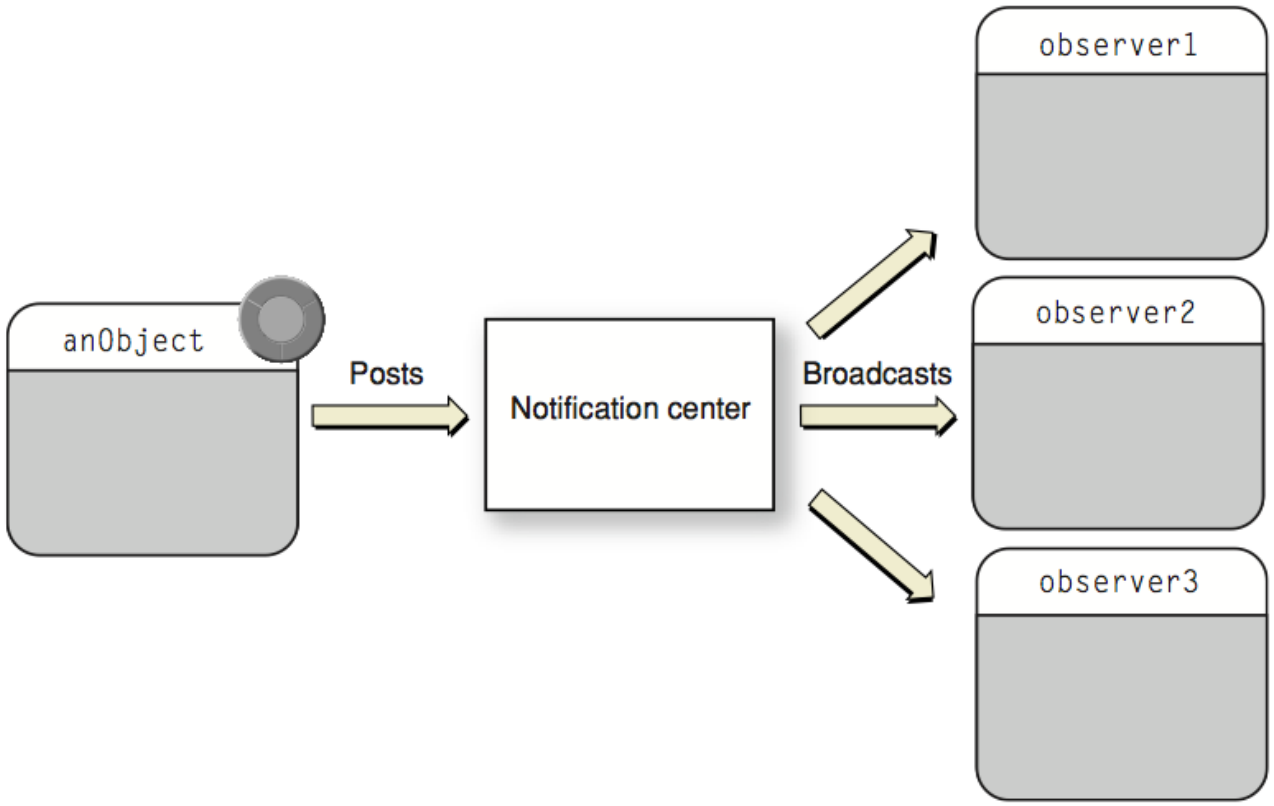
Observer Pattern

NSNotificationCenter



NSNotificationCenter

Figure 5-6 Posting and broadcasting a notification



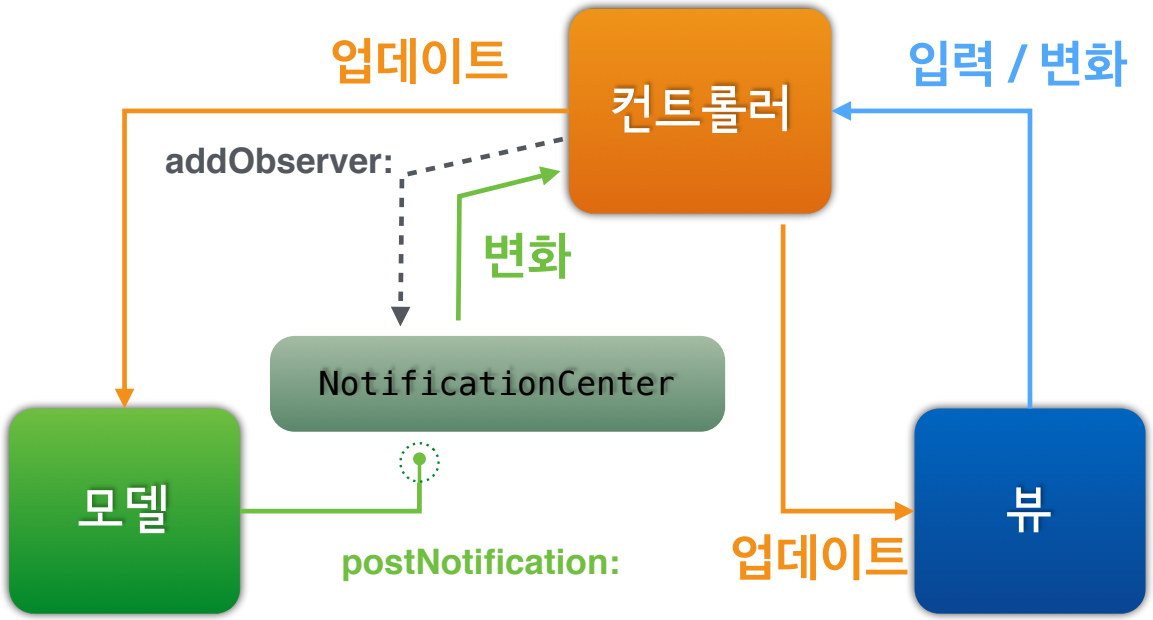
NSNotificationCenter

Observer	Notification Name	Sender
observerA	ModelDataChanged	albumModel
observerB	nil	albumTableView
observerC	DidShakeMotion	albumViewController
observerC	AllDataChanged	nil
observerD	nil	nil

Model - View - Controller

UIViewController

Controller : Coordination



Model : Data

NSObject

View : Display

UIView

XIB

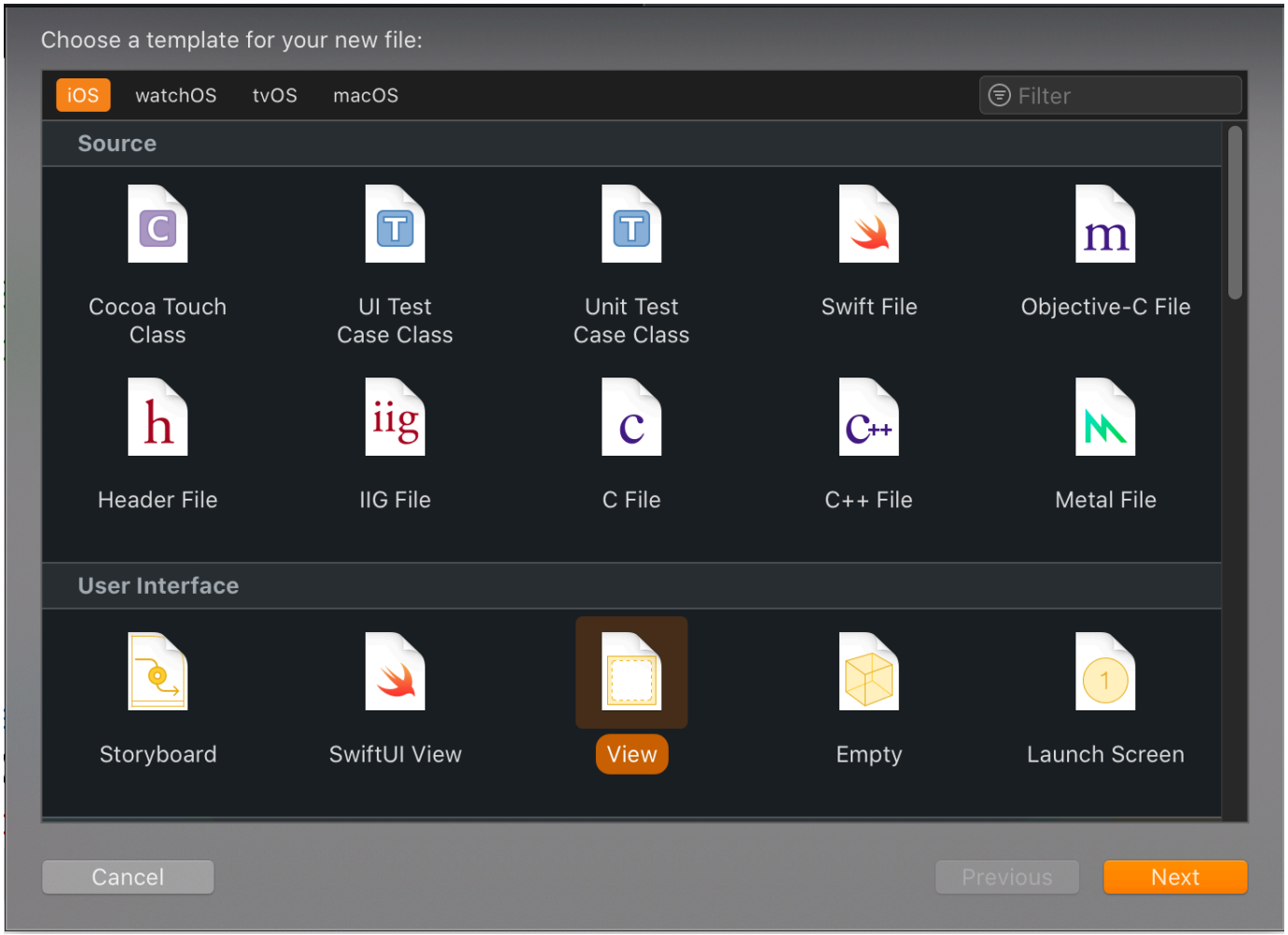
- NIB 파일 (인터페이스 빌더 리소스 파일)
  - Next Interface Builder
  - XML Interface Builder

Binary

XML

- 화면 UI를 기록하는 XML
- 안드로이드의 layout XML과 비슷
- XIB에는 하나 이상의 뷰 객체가 존재할 수 있음
- 소스버전관리 사용을 위해 nib 대신 xib 사용
- MVC 구조에서 뷰와 컨트롤러를 분리하기 위함

How to make


Choose a template for your new file:
iOS
watchOS
tvOS
macOS
Filter
Source
Cocoa Touch Class
UI Test Case Class
Unit Test Case Class
Swift File
Objective-C File
Header File
IIG File
C File
C++ File
Metal File
User Interface
Storyboard
SwiftUI View
View
Empty
Launch Screen
Cancel
Previous
Next

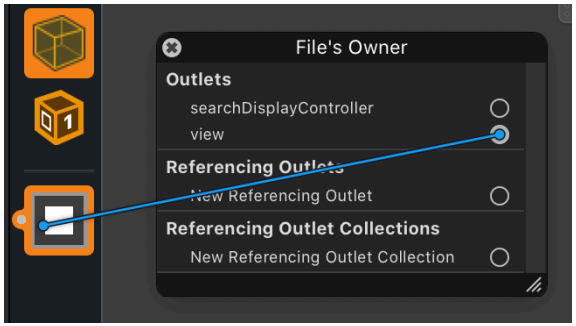
How to use

1) NSBundle

```
let nibObjects = Bundle.main.loadNibNamed(“View", owner: self, options: nil)

let nibView = nibObjects?.first as? UIView
```

2) UIViewController

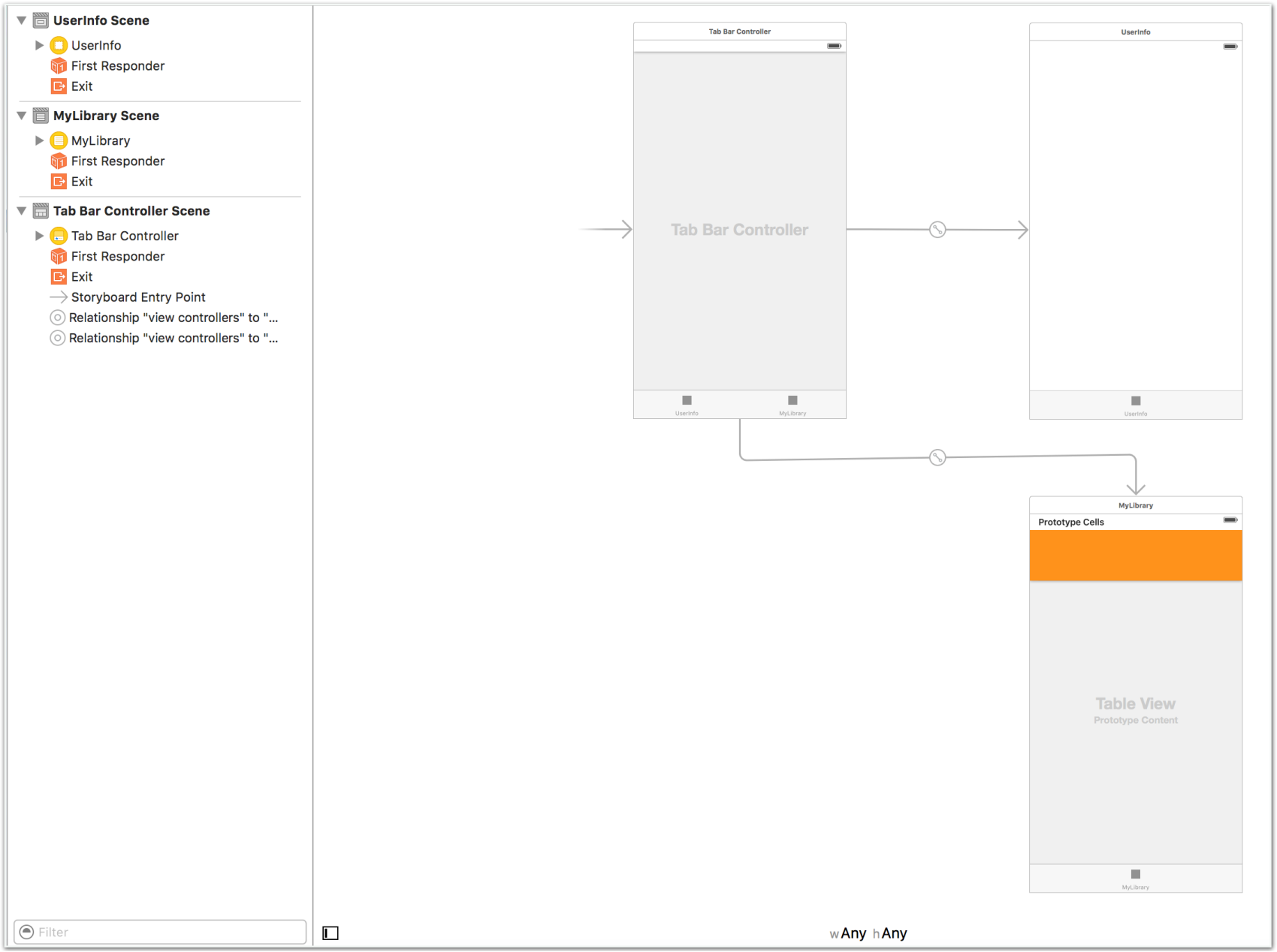
```
//ViewController.view outlet to View
let viewController = UIViewController.init(nibName: "View", bundle: nil)

self.view.addSubview(viewController.view)
```



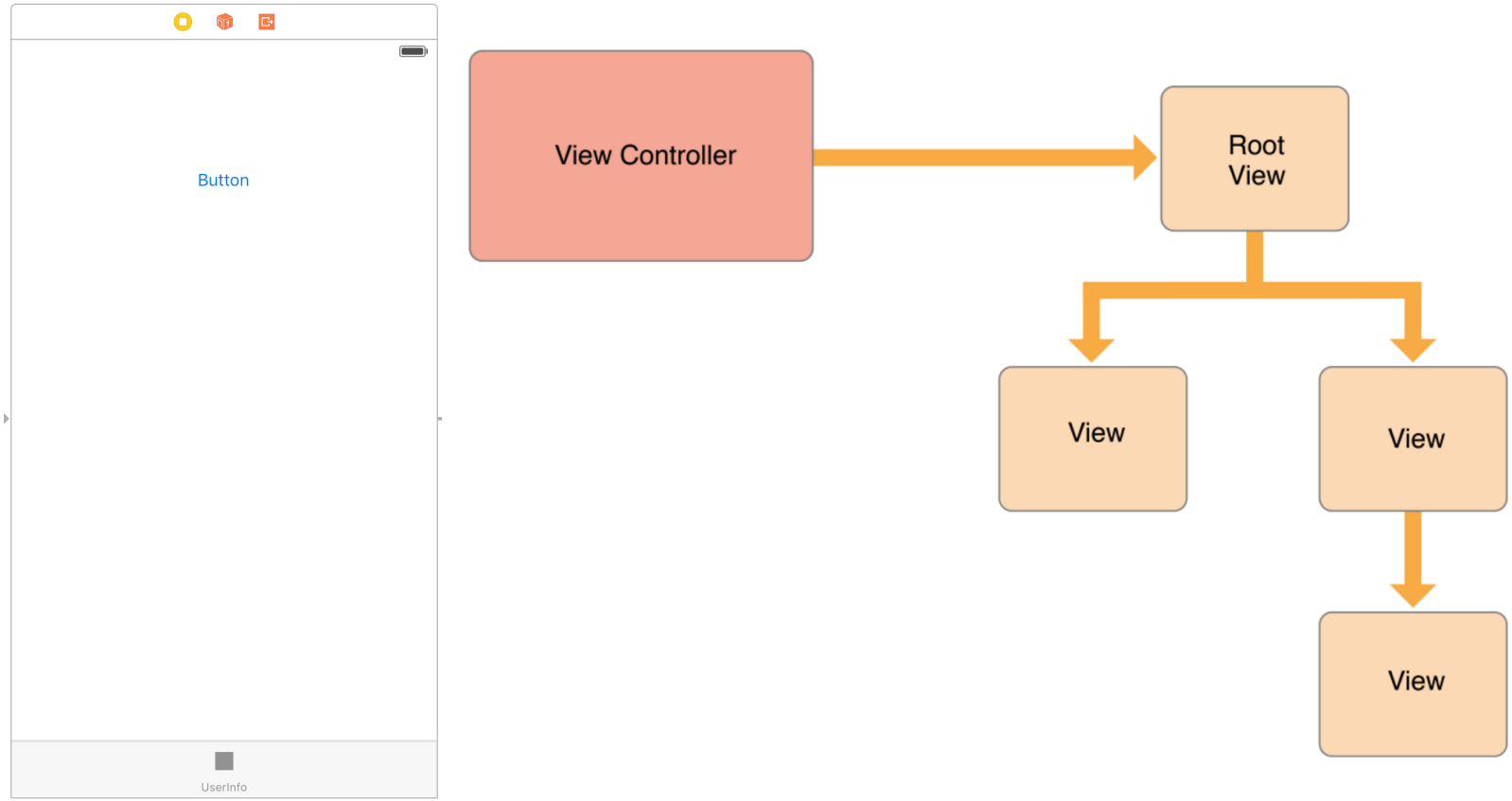
장단점

- 독립적인 뷰 디자인
- 뷰단위 재사용성 높음
- 화면사이즈 / 로컬라이즈 리소스 미리 보기
- 동적으로 로딩 가능 (Lazy Loading)
- 적용하기 어려운 경우
  - 콘텐츠 구성이나 레이아웃이 동적으로 바뀌는 경우
  - 인터페이스 빌더에서 바꿀 수 없는 속성
  - 뷰 컨트롤러 사이의 연결 (Transition Animation)

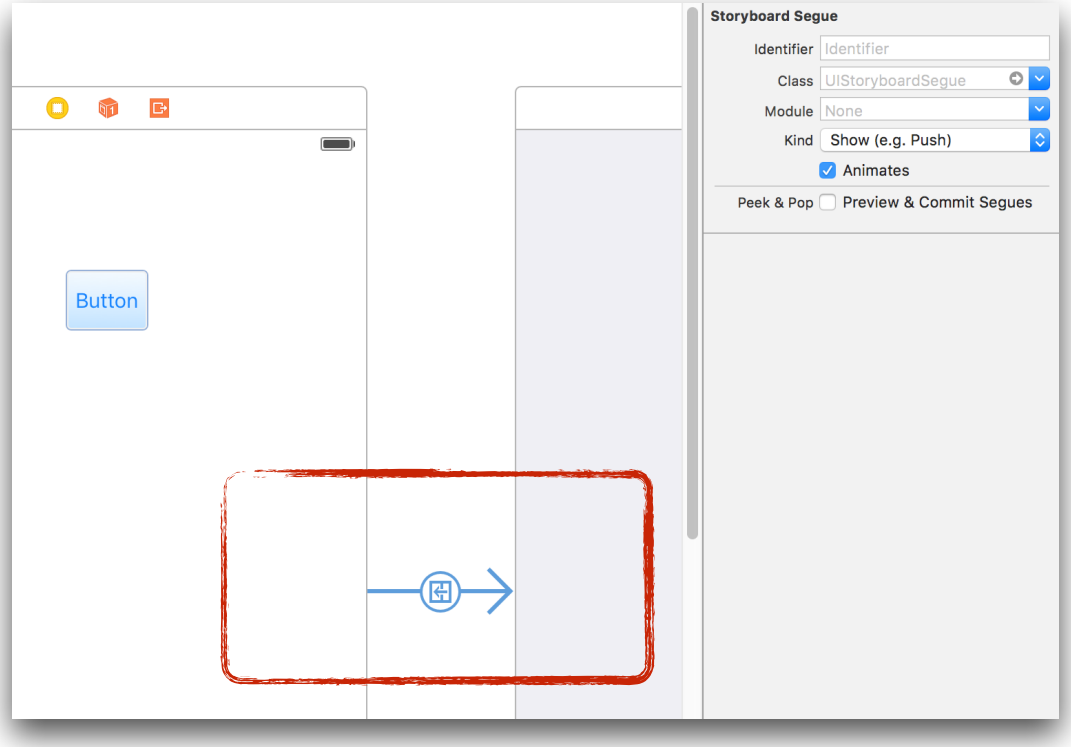
StoryBoard


UserInfo Scene
UserInfo
First Responder
Exit
MyLibrary Scene
MyLibrary
First Responder
Exit
Tab Bar Controller Scene
Tab Bar Controller
First Responder
Exit
Storyboard Entry Point
Relationship "view controllers" to "...
Relationship "view controllers" to "...
Filter
Tab Bar Controller
Tab Bar Controller
UserInfo
MyLibrary
UserInfo
UserInfo
MyLibrary
Prototype Cells
Table View
Prototype Content
MyLibrary
wAny hAny

화면(Scene)



화면 전환(Segue)



- 화면과 화면 사이의 전환에 대한 정의
- Transition Animation 효과도 지정 가능
- 다음 화면 객체가 자동으로 생성

```
<?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">
```

App. Bundle

- Resources 디렉토리
- MainStoryboard.storyboardc
- Archived View-Object
- Unarchive (to UIStoryboard)
- Outlet connection to instance variable

Archived View-Object

```
override class func awakeFromNib() {
    super.awakeFromNib()
    print("awakeFromNib")
}
```

뷰 객체를 모두 unarchive 처리한 직후

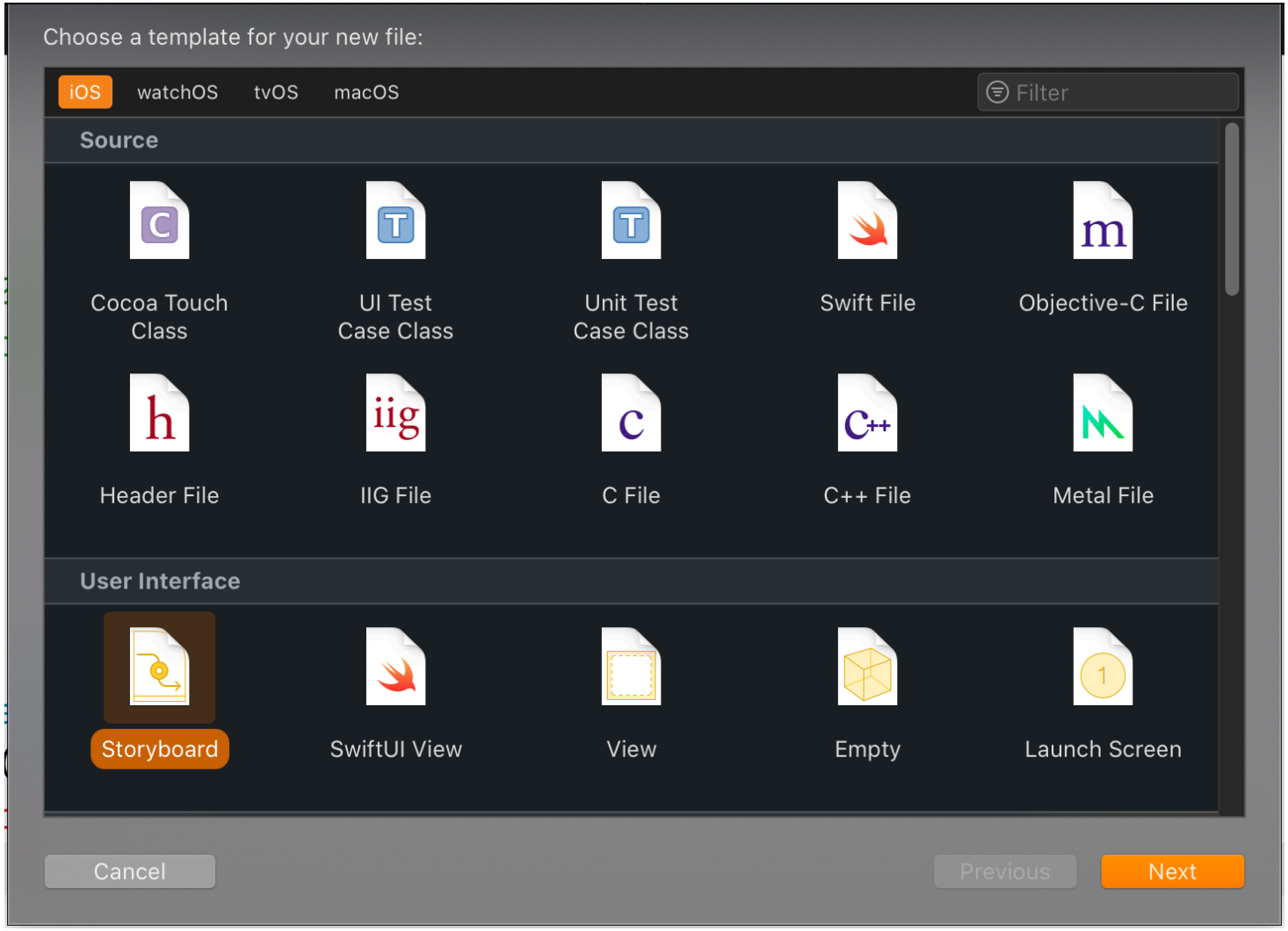
```
required init?(coder: NSCoder) {
    super.init(coder: coder)
    print("initWithCoder")
}
```

뷰 객체를 unarchive 할 때

```
override init(frame: CGRect) {
    super.init(frame: frame)
    print("initWithFrame")
}
```

뷰 객체를 직접 생성할 때

How to make


Choose a template for your new file:
iOS
watchOS
tvOS
macOS
Filter
Source
Cocoa Touch Class
UI Test Case Class
Unit Test Case Class
Swift File
Objective-C File
Header File
IIG File
C File
C++ File
Metal File
User Interface
Storyboard
SwiftUI View
View
Empty
Launch Screen
Cancel
Previous
Next

How to use UIStoryboad

```
//번들에서 특정 스토리보드 파일을 불러오는 경우
init(name: String, bundle storyboardBundleOrNil: Bundle?)
```

https://developer.apple.com/documentation/uikit/uistoryboard/1616216-init

```
//스토리보드 객체에서 시작 지점의 뷰 컨트롤러 객체 만들기
func instantiateInitialViewController() -> UIViewController?
```

https://developer.apple.com/documentation/uikit/uistoryboard/1616213-instantiateinitialviewcontroller

```
//스토리보드 객체에서 특정 이름의 뷰 컨트롤러 객체 만들기
func instantiateViewController(withIdentifier identifier: String) ->
UIViewController
```

https://developer.apple.com/documentation/uikit/uistoryboard/1616214-instantiateviewcontroller

장단점

- 앱 시작 화면부터 전체 화면의 흐름
- 화면사이즈 / 로컬라이즈 리소스 미리 보기
- 정적 Cell 디자인으로 프로토타이핑 가능
- 화면/파일 단위 동적으로 로딩 가능 (Lazy Loading)
- 화면 사이 전환에 대한 처리
- 적용하기 어려운 경우
  - 콘텐츠 구성이나 레이아웃이 동적으로 바뀌는 경우
  - 인터페이스 빌더에서 바꿀 수 없는 속성
  - 버전관리 도구에서 충돌이 자주 발생

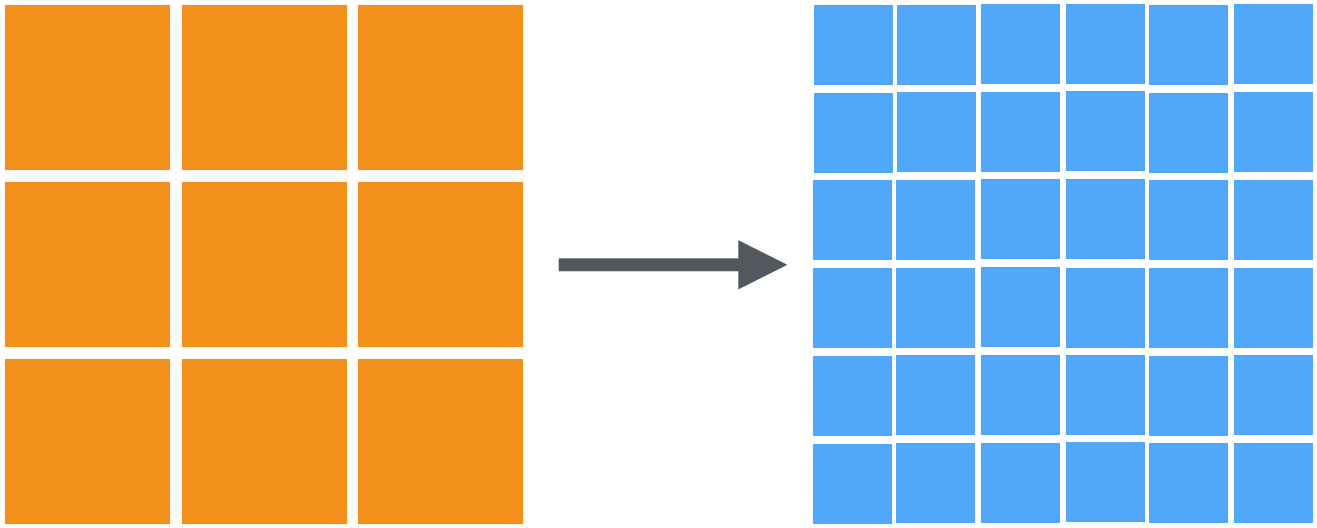
Coordinate System

좌표 시스템

Coordinate System

Retina Displays

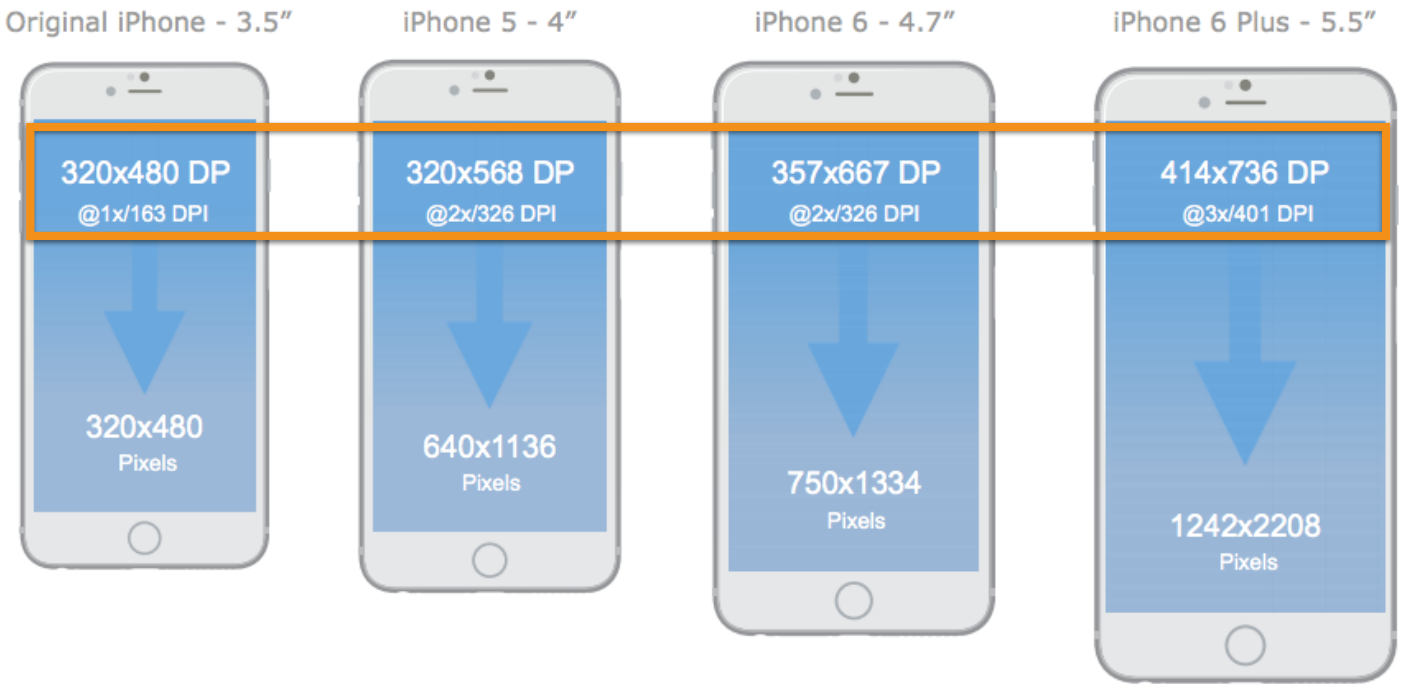
4x pixels of previous displays



UIScreen's bounds

디바이스 화면 크기를 알아야 할 때

`UIScreen.main.bounds`



iPad mini

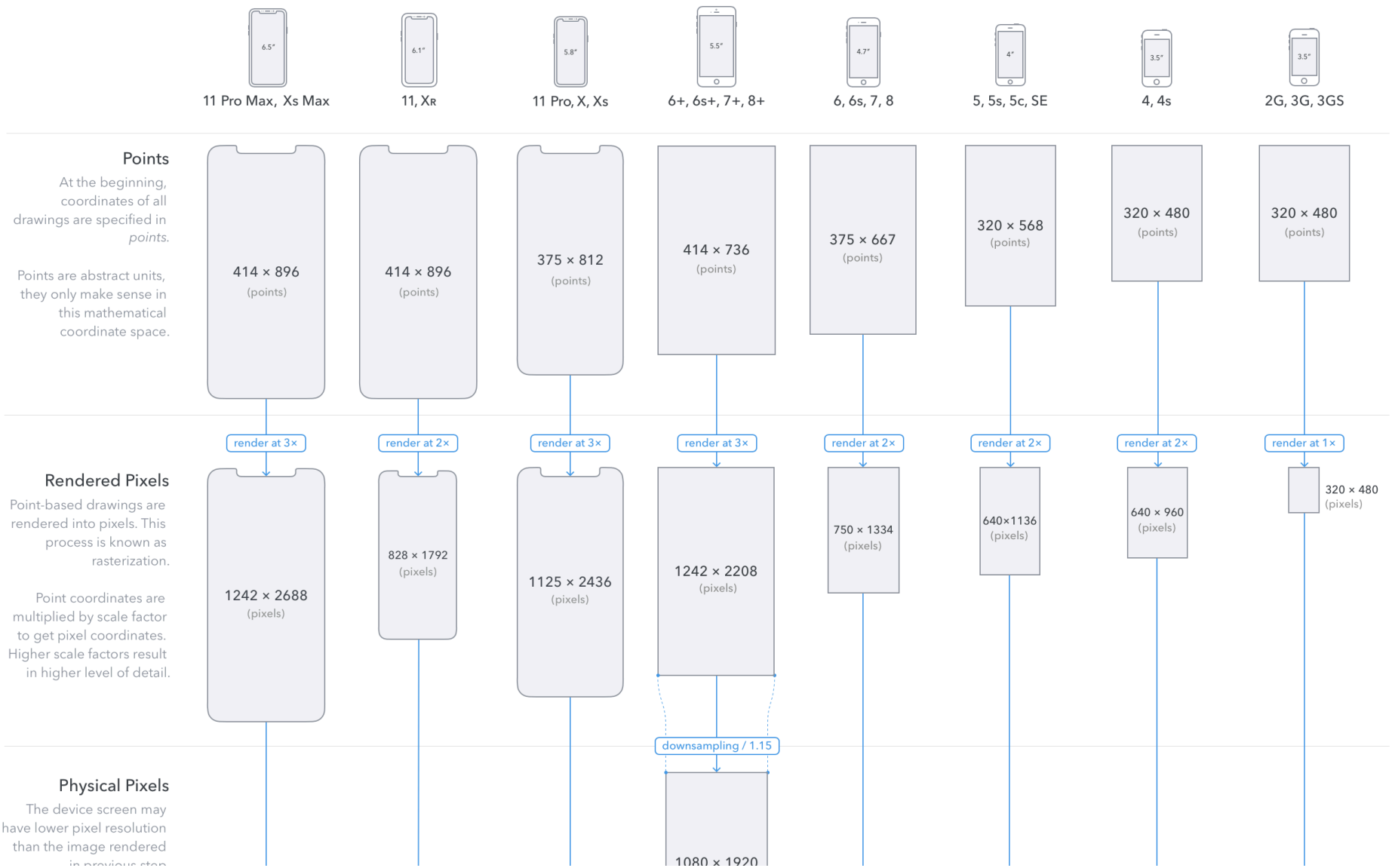
768 x 1024
@2x

iPad Air

768 x 1024
@2x

iPad Pro

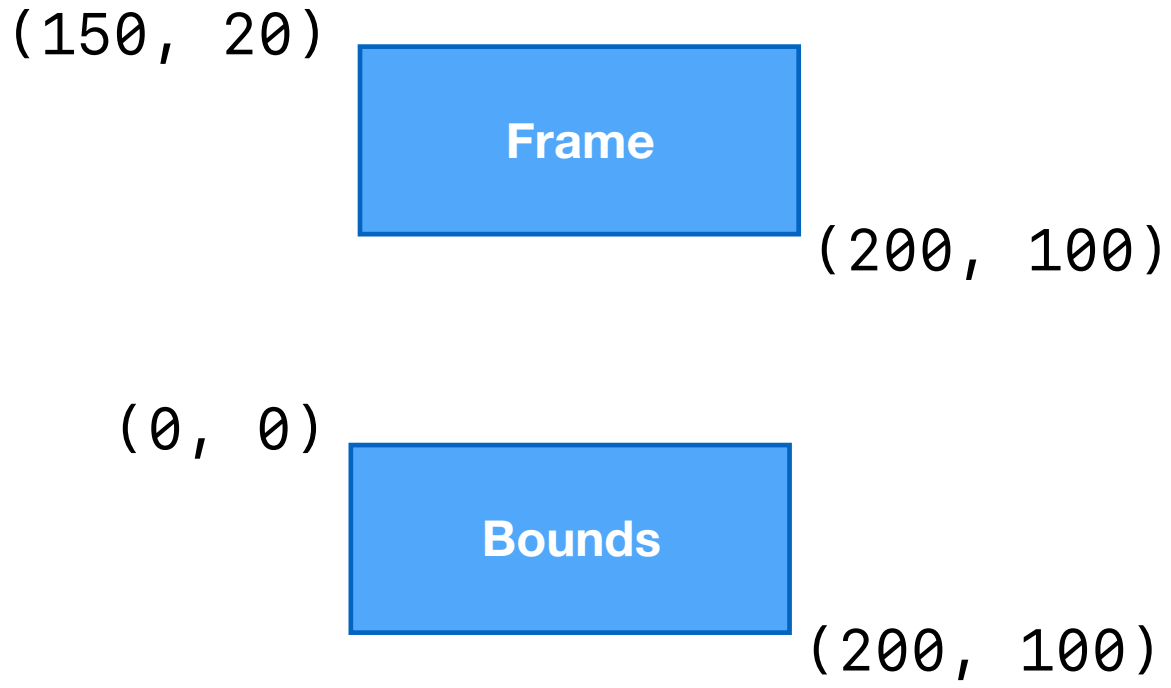
1024 x 1366
@2x



https://www.paintcodeapp.com/news/ultimate-guide-to-iphone-resolutions

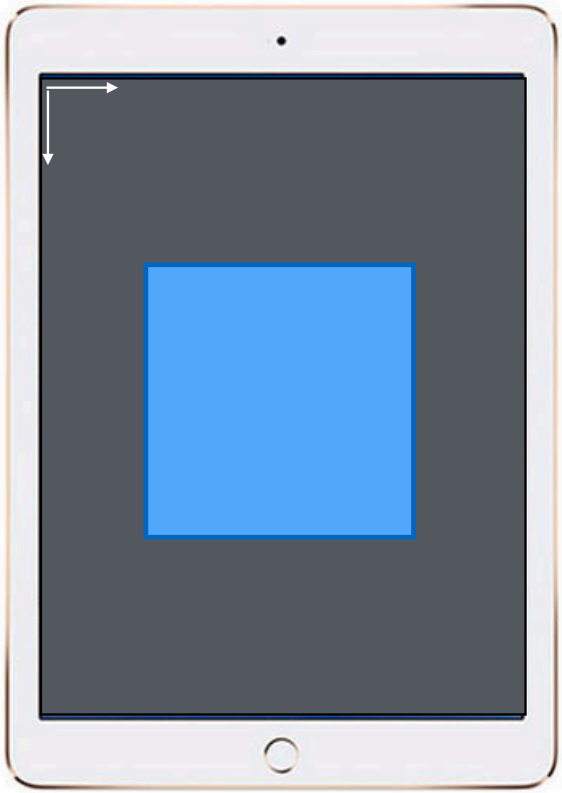
UIView

Frame & Bounds



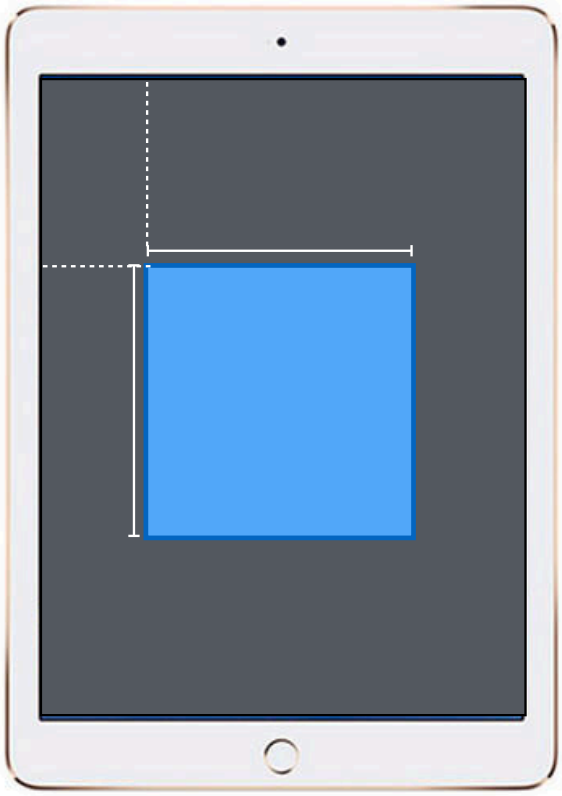
Geometry

Positioning and Sizing a View



Frame

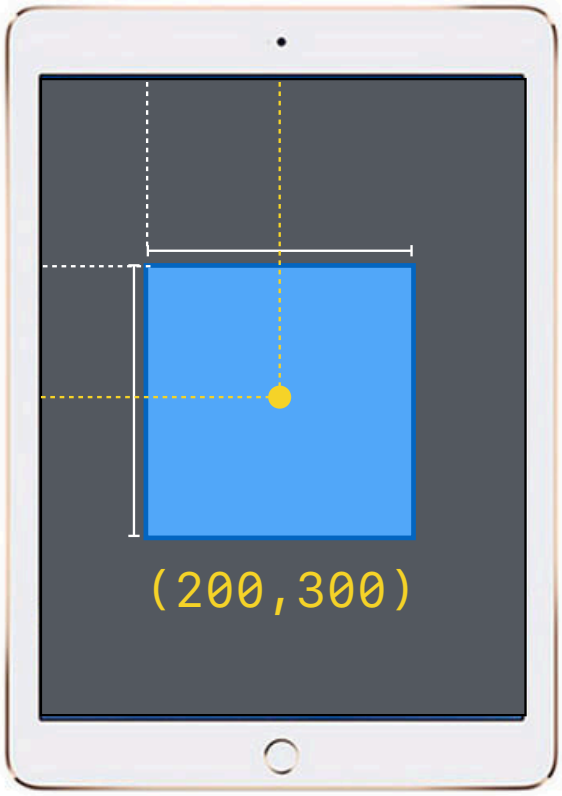
Superview’s coordinate Space



```
{(100,200),(200,200)}
```

Center

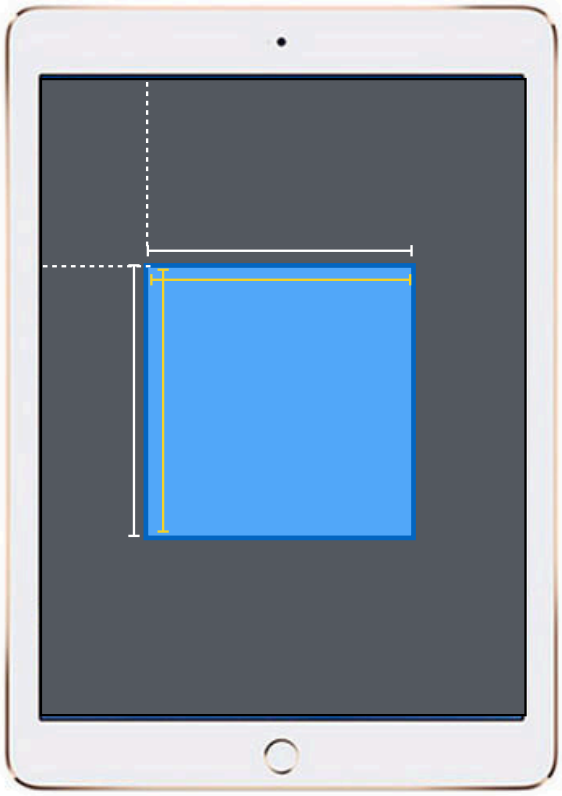
Superview’s coordinate Space




```
{(100,200),(200,200)}
```

Bounds

View’s coordinate Space



`{(0,0),(200,200)}`

`{(100,200),(200,200)}`

Frame vs. Bounds

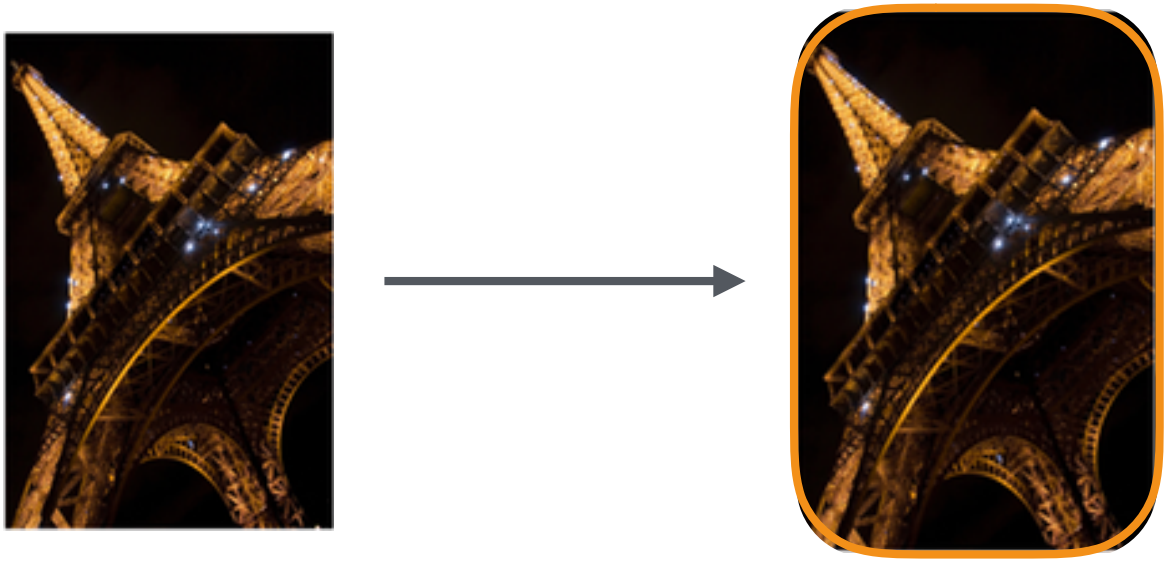
400 x 200 size
UIImageView

Superview's bounds =
{(0,0), (200,200)}

ImageView's frame =
{(0,0), (400,200)}

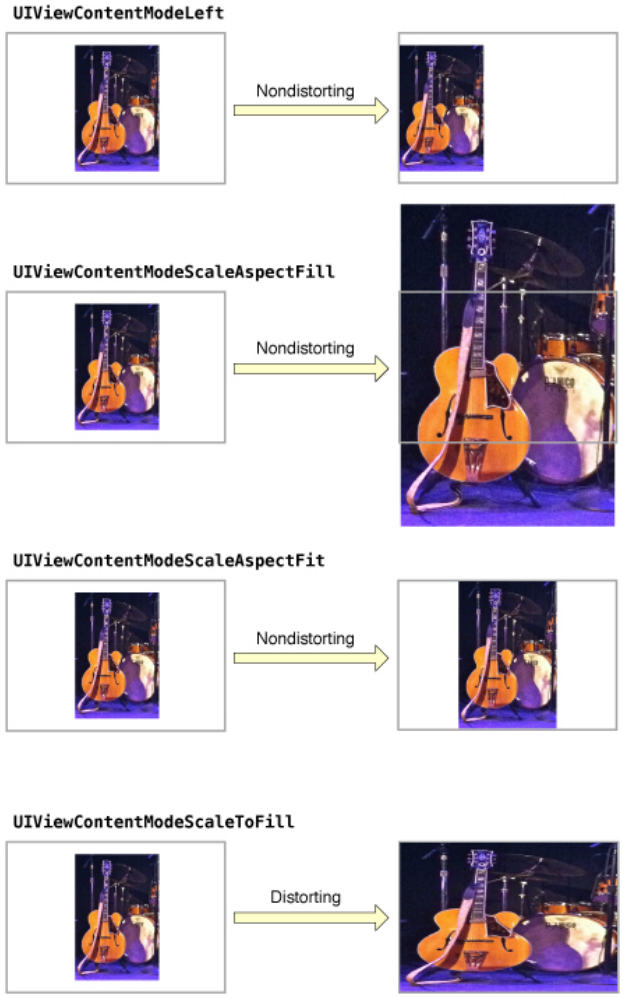
Boundary Clipping

* UIView clipsToBounds property
* CALayer masksToBounds property



View Content Mode

```
enum UIViewContentMode {
    case scaleToFill
    case scaleAspectFit
    case scaleAspectFill
    case redraw
    case center
    case top
    case bottom
    case left
    case right
    case topLeft
    case topRight
    case bottomLeft
    case bottomRight
}
```

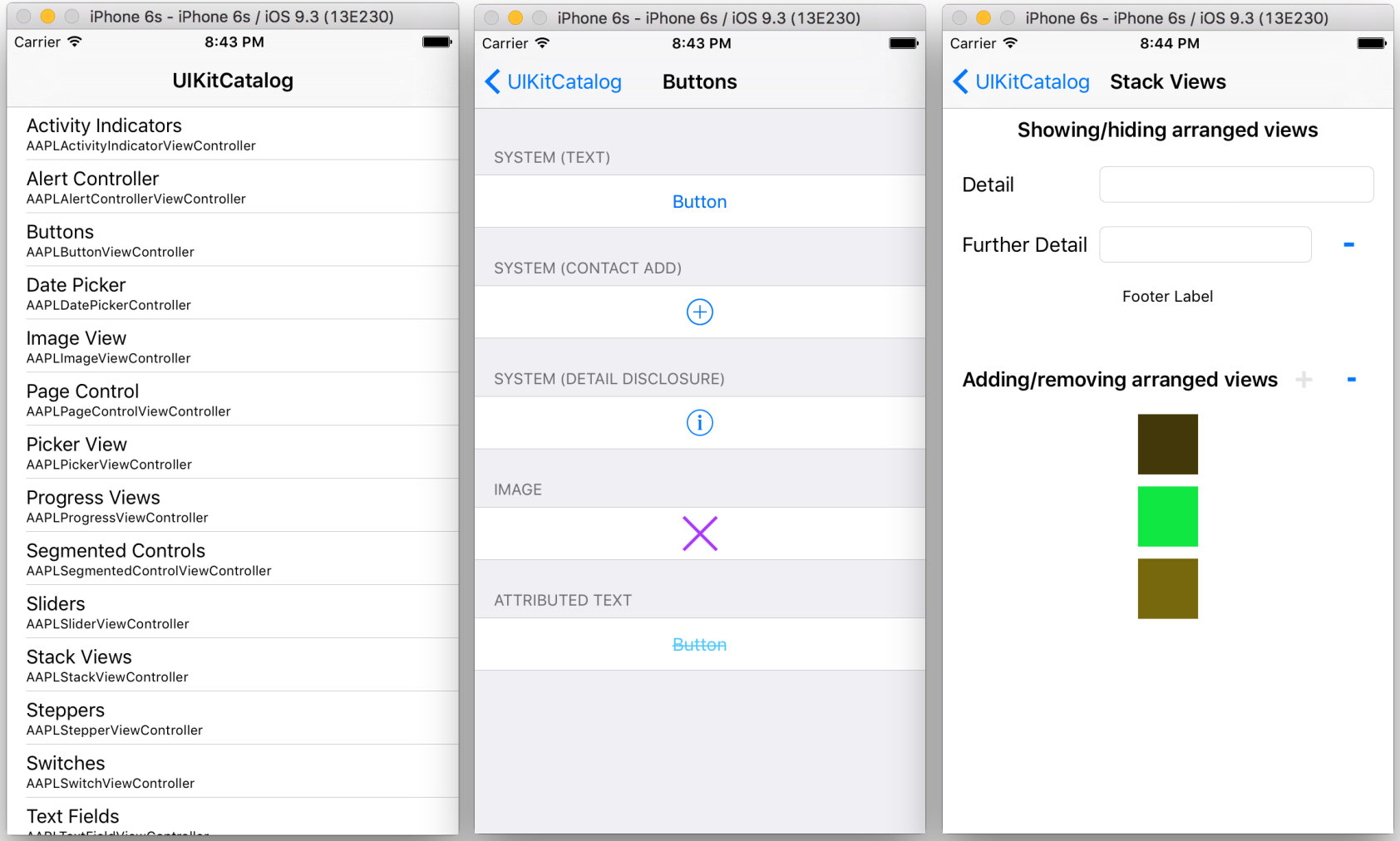


iOS 프로그래밍

미션. 자판기 View Programming

코드스쿼드

UIKit Catalog Example



https://developer.apple.com/documentation/uikit/views_and_controls/uikit_catalog_creating_and_customizing_views_and_controls

XIB vs. Storyboard vs. Code

- prototype을 위한 화면 전체 흐름이 중요한가?
- 데이터 구조에 따라 달라지는 동적 화면인가?
- 반복적으로 재사용하는 단위가 무엇인가?
- 화면 레이아웃 처리를 디자인 타임에 하나? 런타임에 하나?
- 코드로 해야만하는 커스텀 UI 처리가 얼마나 많은가?
- UI 리소스가 얼마나 많은거나 복잡한가?
- 팀내에서 코드가 익숙한가? IB가 익숙한가?


Carrier
12:40 PM
100%
자판기 화면 만들기
재고 추가 버튼
추가
추가
추가
추가
추가
잔액 추가 버튼
+1000
+5000
잔액 : 50000원
잔액 표시 라벨
5개
재고 수량
5개
5개
5개
5개
구매
구매
구매
구매
구매
구매 버튼
메뉴 이미지
140 x 100
Border = 5
Round = 15
(40, 575)
구매 목록
이미지 간격 50pt
아이패드 화면 크기
(1024 x 768) 기준

이미지 파일 링크

https://www.dropbox.com/s/esimudns5r3s3o6/bossam.png

https://www.dropbox.com/s/f0f6gn7kawo9ytc/chicken.png

https://www.dropbox.com/s/1lzf4f6hxa95prj/dduck.png

https://www.dropbox.com/s/exbbboaxn9oud1v/hamburger.png

https://www.dropbox.com/s/sh9kyoqjguqf2q4/pizza.png

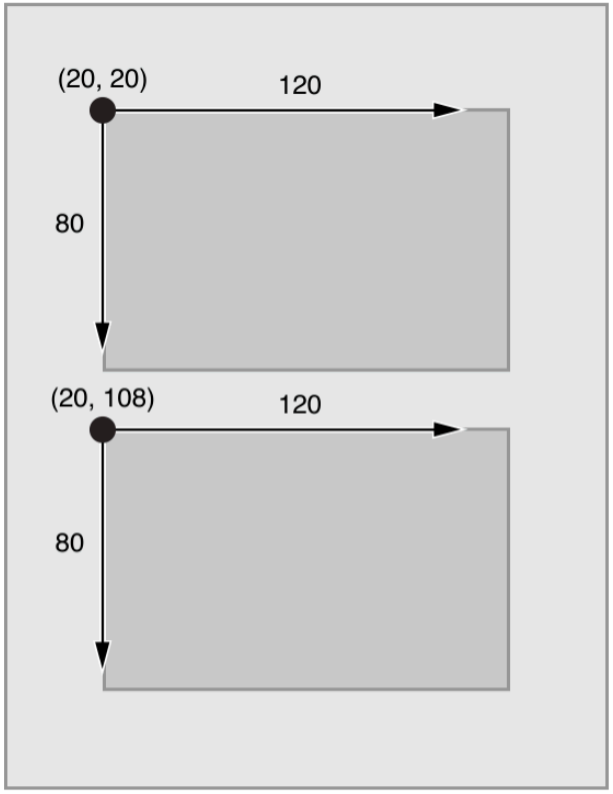
Keywords for View

* IBOutletCollection
* StackView
* Add View by code
* init(frame:)
* Button
* ScrollView

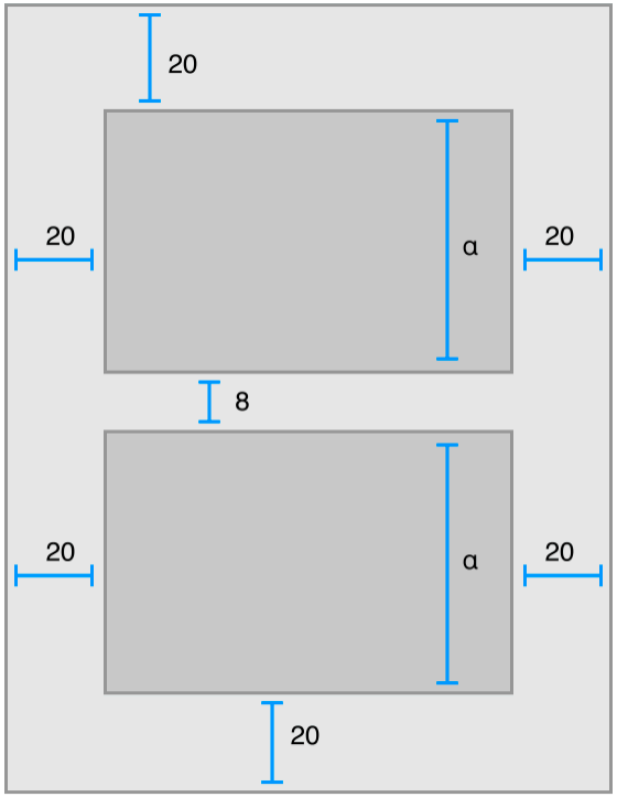
뷰 레이아웃

수동레이아웃 vs. 오토레이아웃

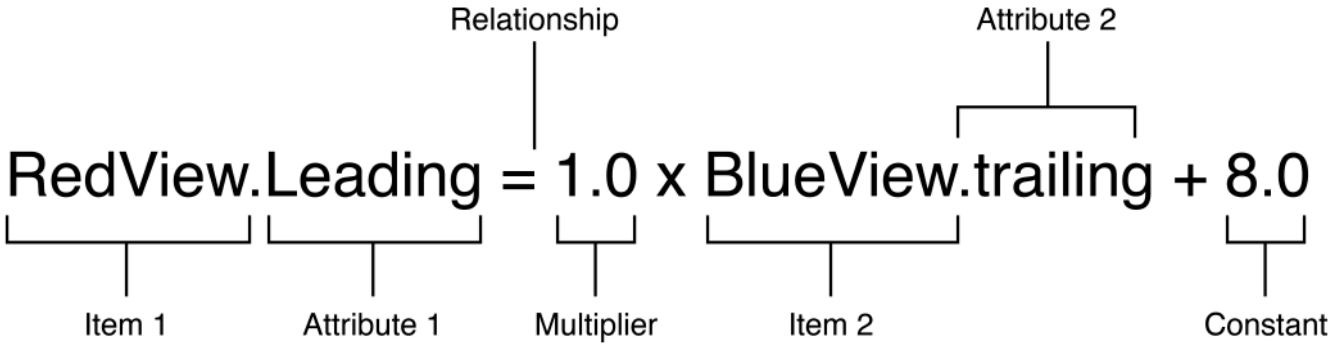
Frame-based



Auto-layout
(constraints)

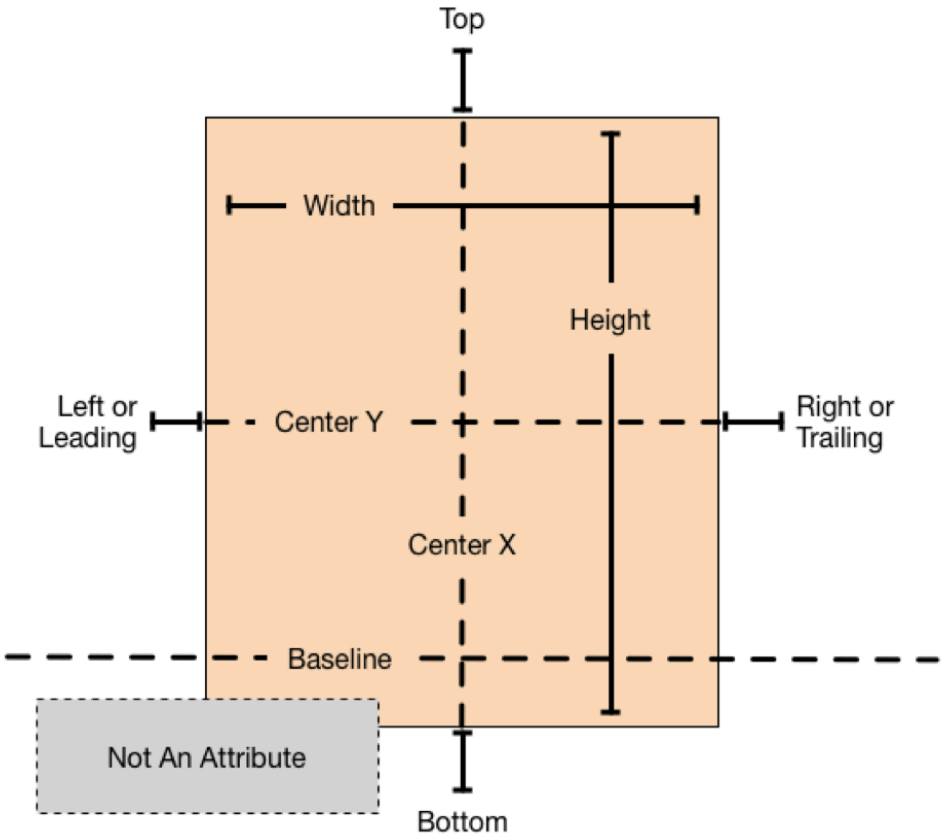


Constraint

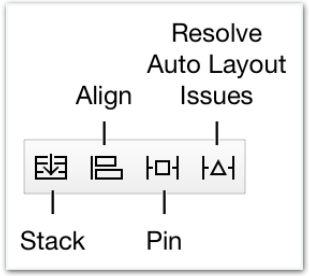


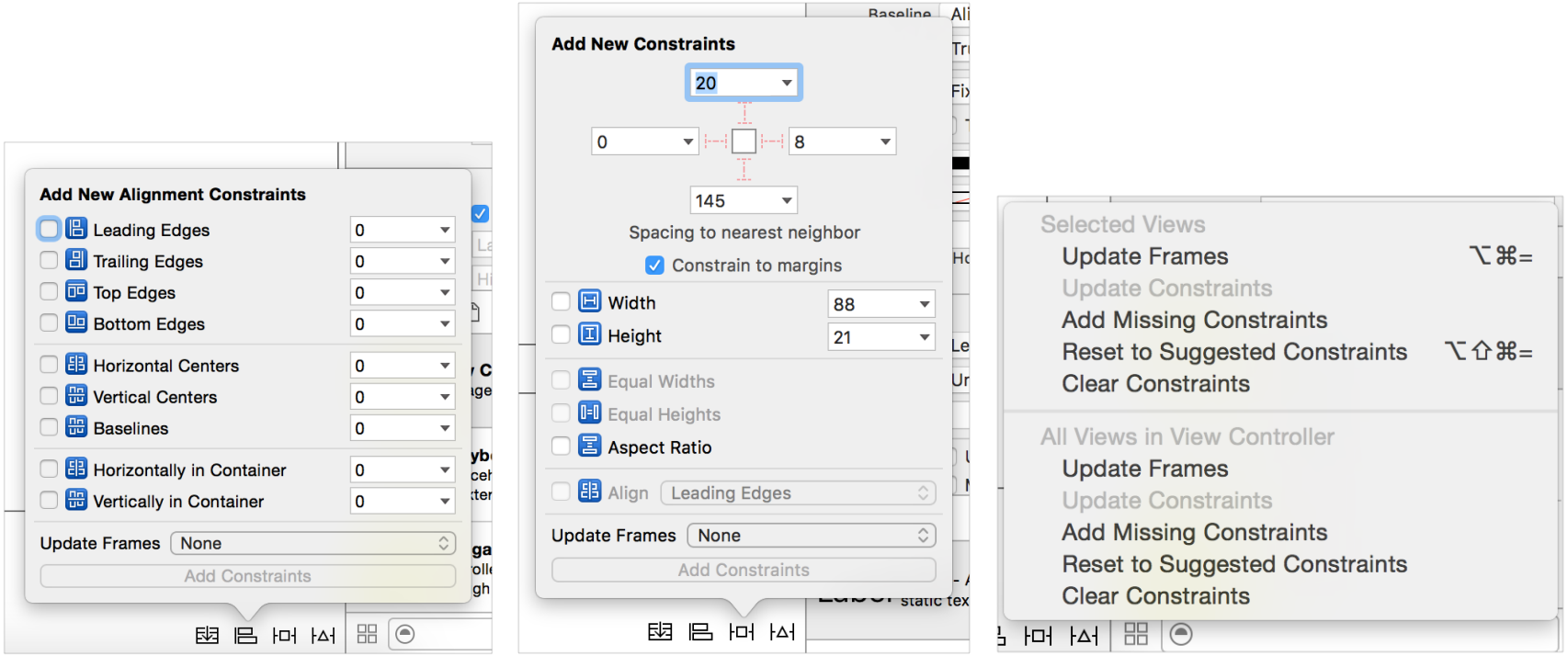
https://developer.apple.com/library/archive/documentation/UserExperience/Conceptual/AutolayoutPG/index.html

Auto Layout Attributes



인터페이스빌더에서 작업하기





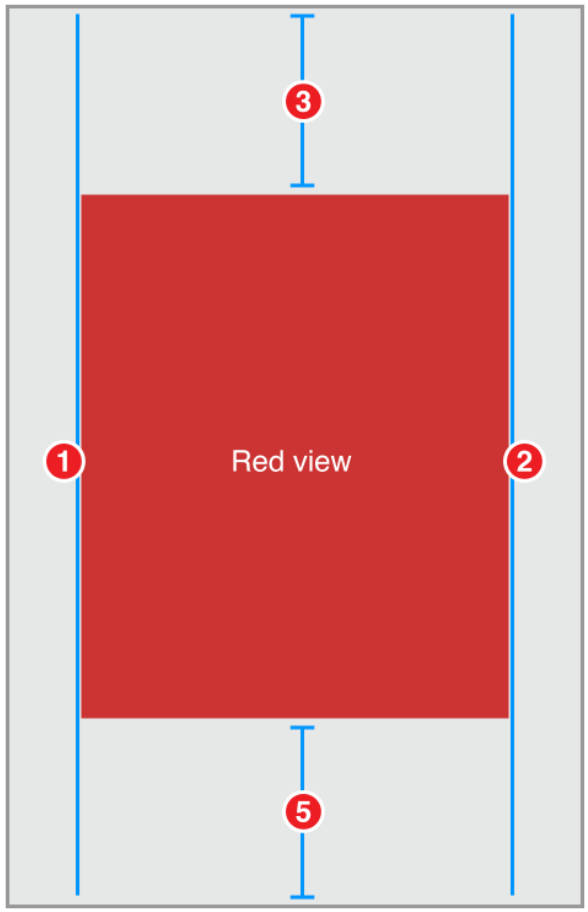
https://developer.apple.com/library/archive/documentation/UserExperience/Conceptual/AutolayoutPG/WorkingwithConstraintsinInterfaceBuidler.html

오토레이아웃 예시

다양한 케이스들

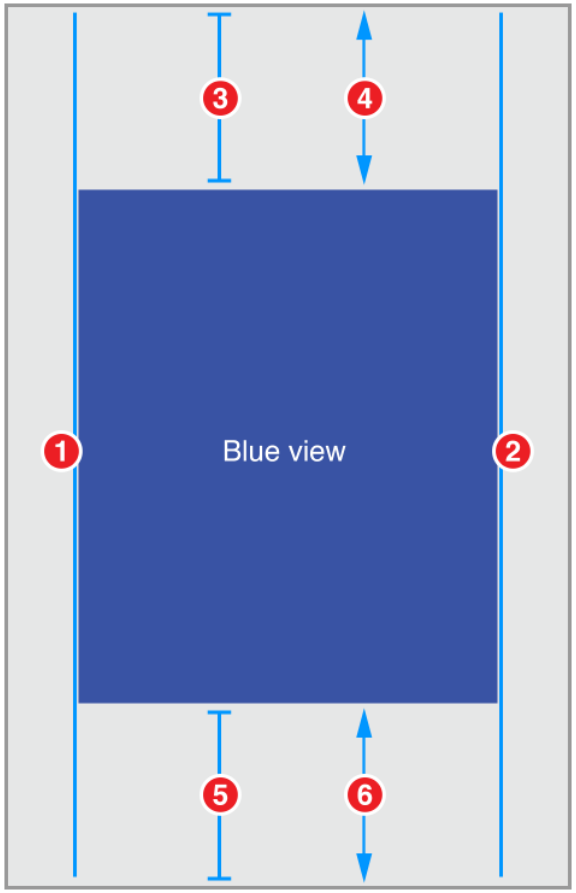
https://developer.apple.com/sample-code/xcode/downloads/Auto-Layout-Cookbook.zip

Simple View



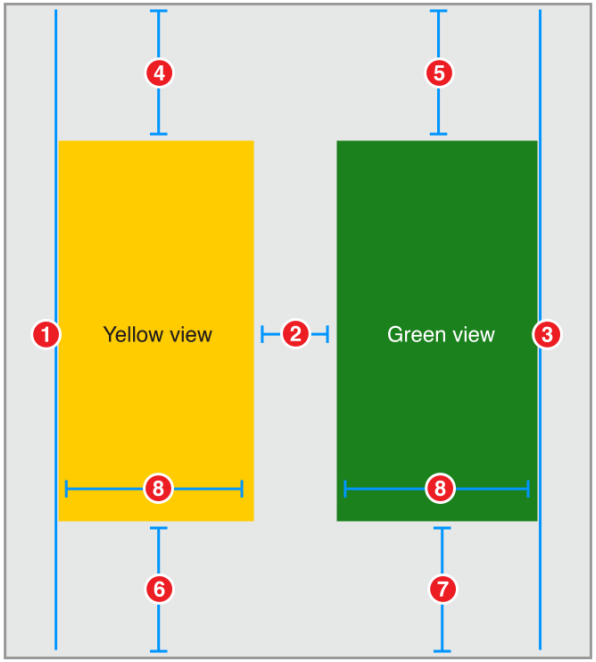

```
Red View.Leading = Superview.LeadingMargin
Red View.Trailing = Superview.TrailingMargin
Red View.Top = Top Layout Guide.Bottom + 20.0
Bottom Layout Guide.Top = Red View.Bottom + 20.0
```

Adaptive View



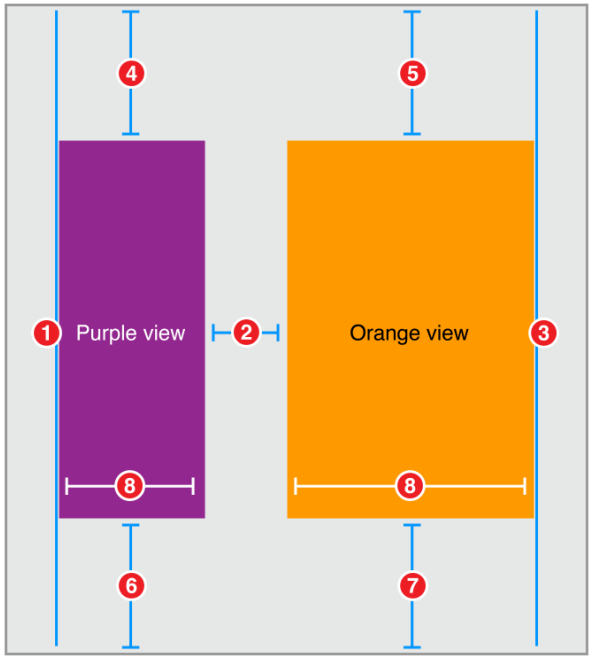

```
Blue View.Leading = Superview.LeadingMargin
Blue View.Trailing = Superview.TrailingMargin
Blue View.Top = Top Layout Guide.Bottom
                          + Standard (Priority 750)
Blue View.Top >= Superview.Top + 20.0
Bottom Layout Guide.Top = Blue View.Bottom
                          + Standard (Priority 750)
Superview.Bottom >= Blue View.Bottom + 20.0
```

Two Equal-width



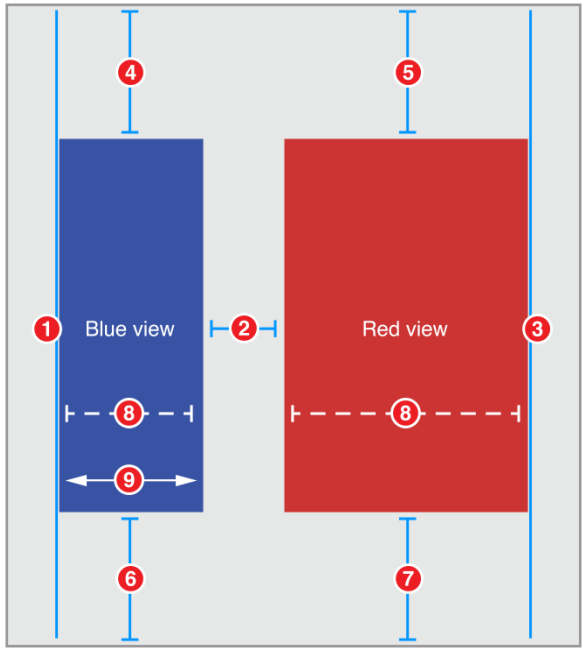

```
Yellow View.Leading = Superview.LeadingMargin
Green View.Leading = Yellow View.Trailing + Standard
Green View.Trailing = Superview.TrailingMargin
Yellow View.Top = Top Layout Guide.Bottom + 20.0
Green View.Top = Top Layout Guide.Bottom + 20.0
Bottom Layout Guide.Top = Yellow View.Bottom + 20.0
Bottom Layout Guide.Top = Green View.Bottom + 20.0
Yellow View.Width = Green View.Width
```

Two Different-width



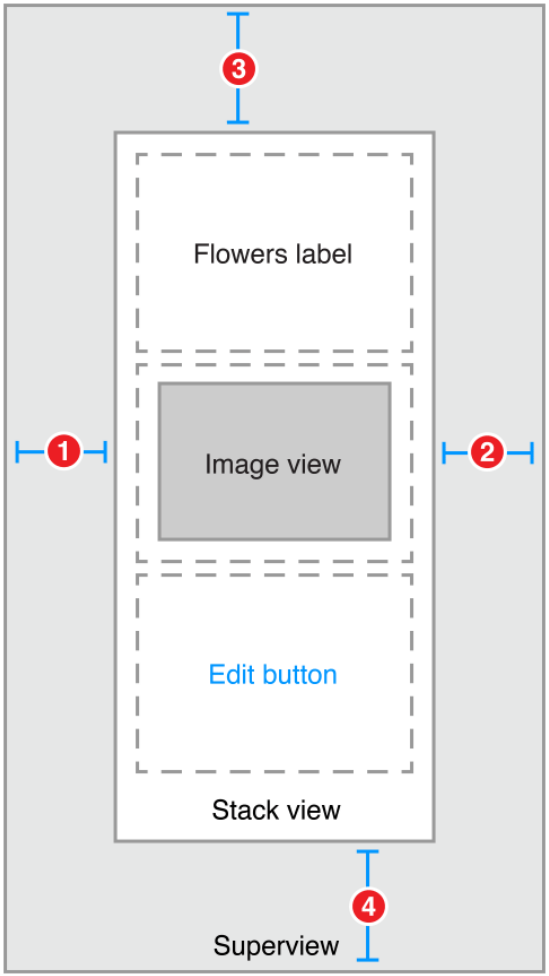

```
Purple View.Leading = Superview.LeadingMargin
Orange View.Leading = Purple View.Trailing + Standard
Orange View.Trailing = Superview.TrailingMargin
Purple View.Top = Top Layout Guide.Bottom + 20.0
Orange View.Top = Top Layout Guide.Bottom + 20.0
Bottom Layout Guide.Top = Purple View.Bottom + 20.0
Bottom Layout Guide.Top = Orange View.Bottom + 20.0
Orange View.Width = 2.0 x Purple View.Width
```

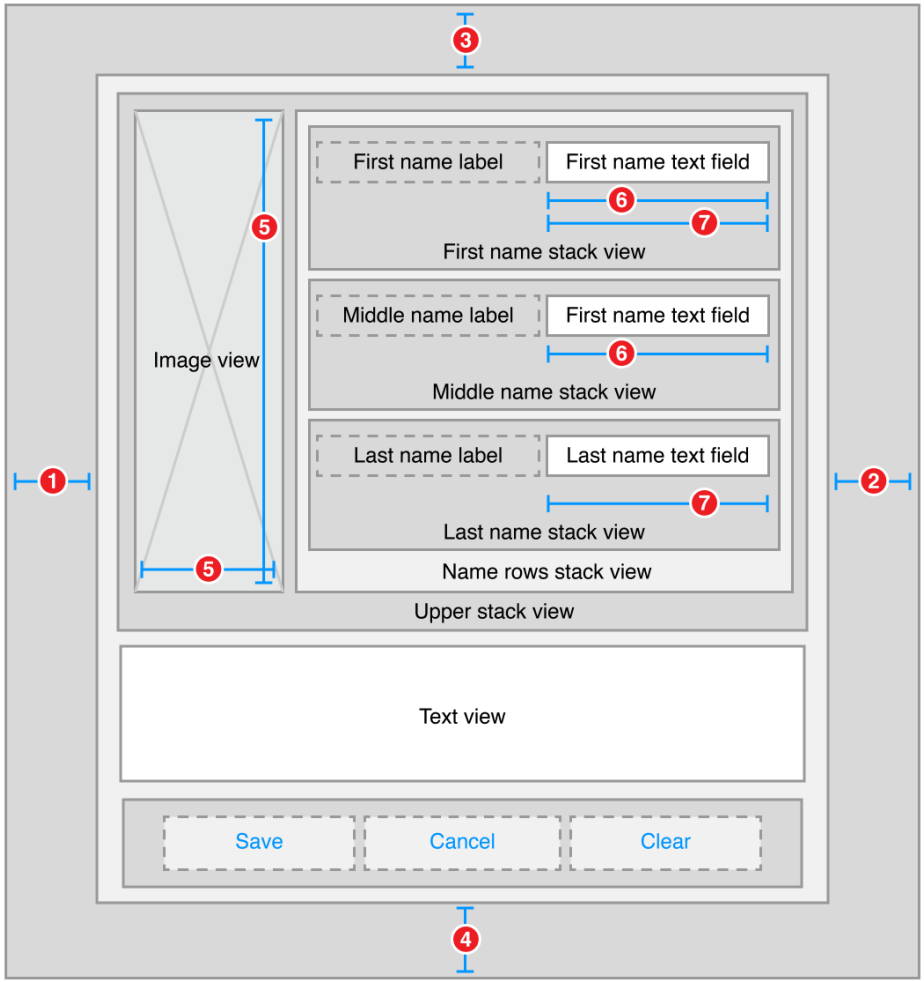
Two Views with complex



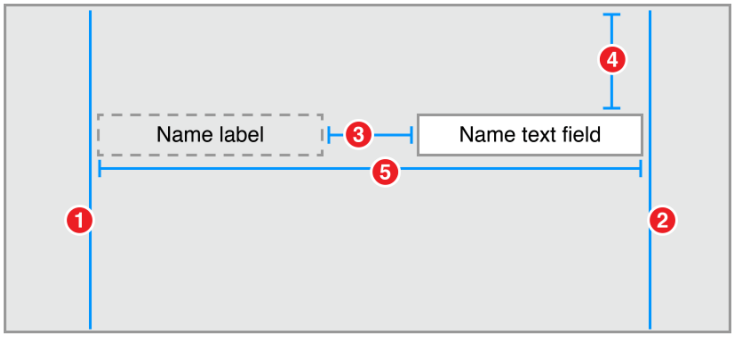

```
Blue View.Leading = Superview.LeadingMargin
Red View.Leading = Blue View.Trailing + Standard
Red View.Trailing = Superview.TrailingMargin
Blue View.Top = Top Layout Guide.Bottom + 20.0
Red View.Top = Top Layout Guide.Bottom + 20.0
Bottom Layout Guide.Top = Blue View.Bottom + 20.0
Bottom Layout Guide.Top = Red View.Bottom + 20.0
Red View.Width = 2.0 x Blue View.Width (Priority 750)
Blue View.Width >= 150.0
```

StackView


3
Flowers label
1
Image view
2
Edit button
Stack view
4
Superview

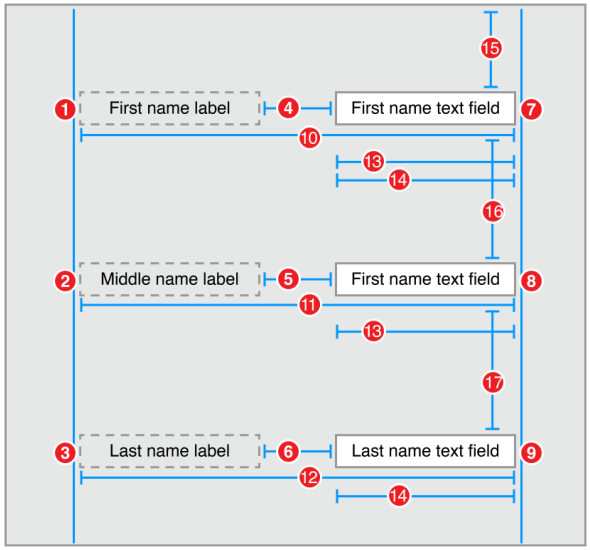

3
First name label
First name text field
6
7
First name stack view
5
Image view
Middle name label
First name text field
6
Middle name stack view
1
2
Last name label
Last name text field
7
Last name stack view
5
Name rows stack view
Upper stack view
Text view
Save
Cancel
Clear
4

Simple Label + TextField



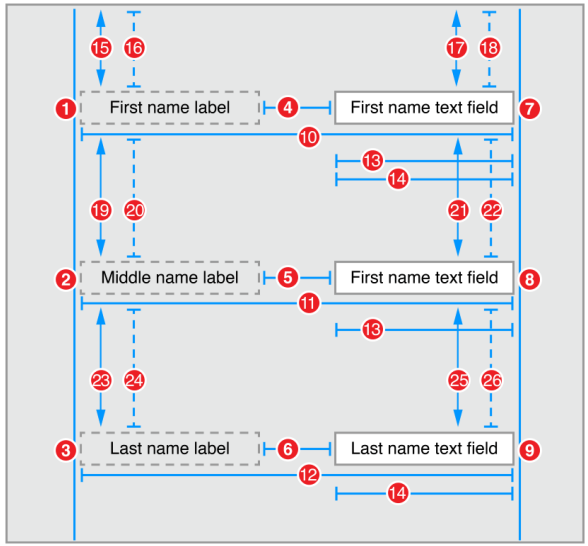

```
Name Label.Leading = Superview.LeadingMargin
Name Text Field.Trailing = Superview.TrailingMargin
Name Text Field.Leading = Name Label.Trailing + Standard
Name Text Field.Top = Top Layout Guide.Bottom + 20.0
Name label.Baseline = Name Text Field.Baseline
```

Fixed Height Columns




```
First Name Label.Leading = Superview.LeadingMargin
Middle Name Label.Leading = Superview.LeadingMargin
Last Name Label.Leading = Superview.LeadingMargin
First Name Text Field.Leading = First Name Label.Trailing
                                    + Standard
Middle Name Text Field.Leading = Middle Name Label.Trailing
                                    + Standard
Last Name Text Field.Leading = Last Name Label.Trailing
                                    + Standard
First Name Text Field.Trailing = Superview.TrailingMargin
Middle Name Text Field.Trailing = Superview.TrailingMargin
Last Name Text Field.Trailing = Superview.TrailingMargin
First Name Label.Baseline = First Name Text Field.Baseline
Middle Name Label.Baseline = Middle Name Text Field.Baseline
Last Name Label.Baseline = Last Name Text Field.Baseline
First Name Text Field.Width = Middle Name Text Field.Width
First Name Text Field.Width = Last Name Text Field.Width
First Name Text Field.Top = Top Layout Guide.Bottom + 20.0
Middle Name Text Field.Top = First Name Text Field.Bottom
                                    + Standard
Last Name Text Field.Top = Middle Name Text Field.Bottom
                                    + Standard
```

Dynamic Height Columns

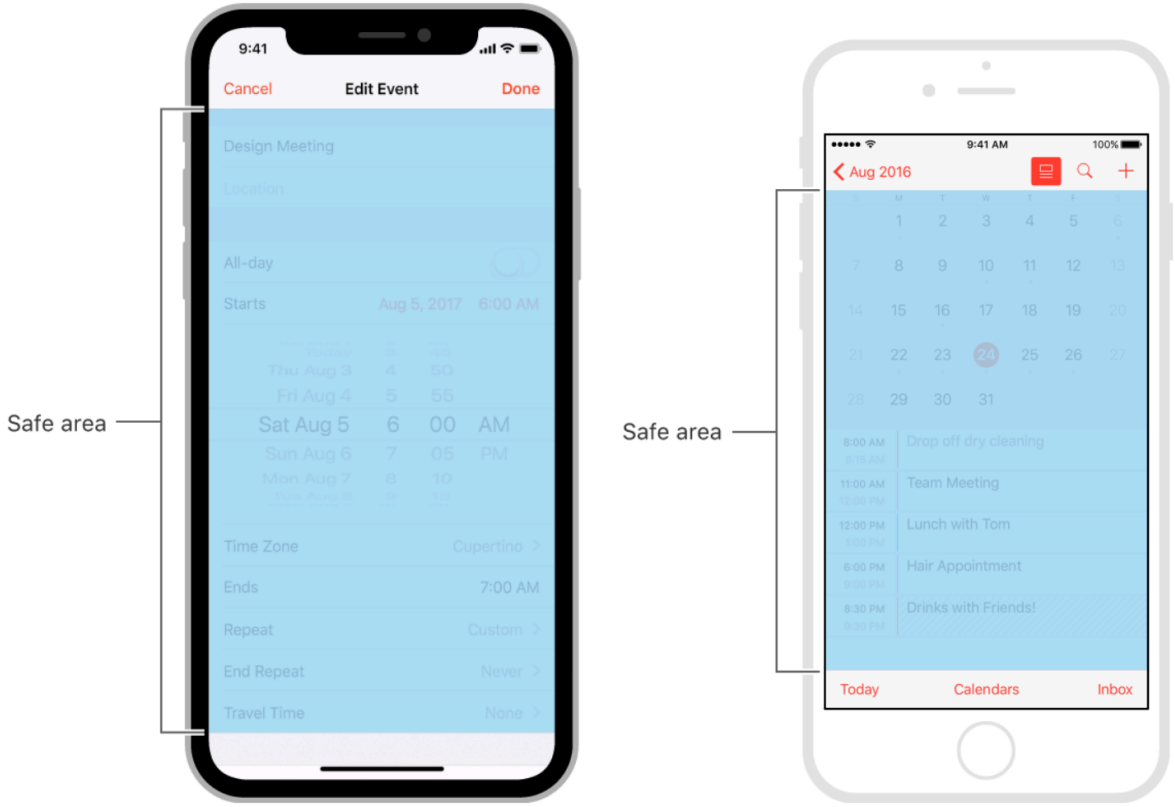



```
First Name Label.Leading = Superview.LeadingMargin
Middle Name Label.Leading = Superview.LeadingMargin
Last Name Label.Leading = Superview.LeadingMargin
First Name Text Field.Leading = First Name Label.Trailing + Standard
Middle Name Text Field.Leading = Middle Name Label.Trailing
                                                + Standard
Last Name Text Field.Leading = Last Name Label.Trailing + Standard
First Name Text Field.Trailing = Superview.TrailingMargin
Middle Name Text Field.Trailing = Superview.TrailingMargin
Last Name Text Field.Trailing = Superview.TrailingMargin
First Name Label.Baseline = First Name Text Field.Baseline
Middle Name Label.Baseline = Middle Name Text Field.Baseline
Last Name Label.Baseline = Last Name Text Field.Baseline
First Name Text Field.Width = Middle Name Text Field.Width
First Name Text Field.Width = Last Name Text Field.Width
First Name Label.Top >= Top Layout Guide.Bottom + 20.0
First Name Label.Top = Top Layout Guide.Bottom
                                          + 20.0 (Priority 249)
First Name Text Field.Top >= Top Layout Guide.Bottom + 20.0
First Name Text Field.Top = Top Layout Guide.Bottom
                                          + 20.0 (Priority 249)
Middle Name Label.Top >= Top Layout Guide.Bottom + Standard
Middle Name Label.Top = Top Layout Guide.Bottom
                                          + Standard (Priority 249)
Middle Name Text Field.Top >= Top Layout Guide.Bottom + Standard
Middle Name Text Field.Top = Top Layout Guide.Bottom
                                          + Standard (Priority 249)
Last Name Label.Top >= Top Layout Guide.Bottom + Standard
Last Name Label.Top = Top Layout Guide.Bottom
                                           + Standard (Priority 249)
Last Name Text Field.Top >= Top Layout Guide.Bottom + Standard
Last Name Text Field.Top = Top Layout Guide.Bottom
                                            + Standard (Priority 249)
```

Safe Area

Relative Content

Figure 1 The safe area of an interface



https://developer.apple.com/documentation/uikit/uiview/positioning_content_relative_to_the_safe_area

View Animation

UIView, Core Animation, UIKit Dynamics

UIView Property

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

- frame
- bounds
- center
- alpha
- background color
- transform

클로저 인자값으로 사용하는 API

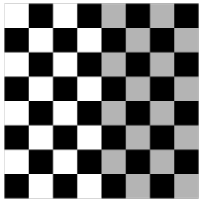
```
UIView.animate(withDuration: TimeInterval, animations: ()->Void)
```

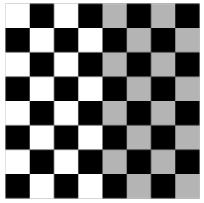
```
var view = UIView(frame: CGRect(x: 10, y: 10, width: 100, height: 100))
UIView.animate(withDuration: 5, animations: {
    view.frame = CGRect(x: 100, y: 100, width: 100, height: 100)
})
```

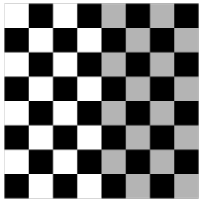
```
UIView.animate(withDuration: TimeInterval,
               animations: ()->Void,
               completion: ((Bool)->Void)?)
```

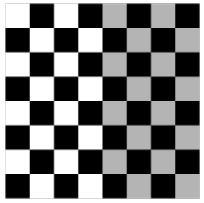
```
var view = UIView(frame: CGRect(x: 10, y: 10, width: 100, height: 100))
UIView.animate(withDuration: 5,
               animations: {
                 view.frame = CGRect(x: 100, y: 100, width: 100, height: 100)
               },
               completion: { (complete:Bool) in
               //maybe other animation
               }
)
```

Affine Transform

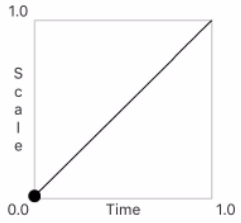
Identity 

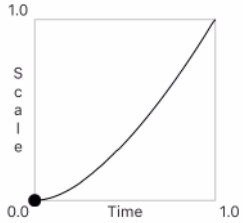
Scale 

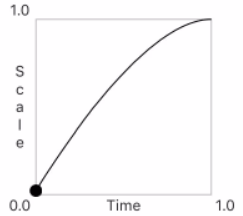
Transition 

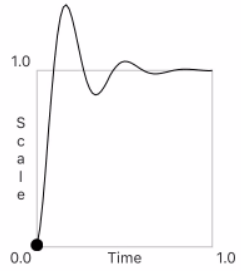
Rotate 

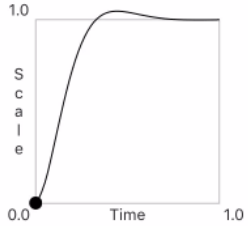
Easing - Timing Function





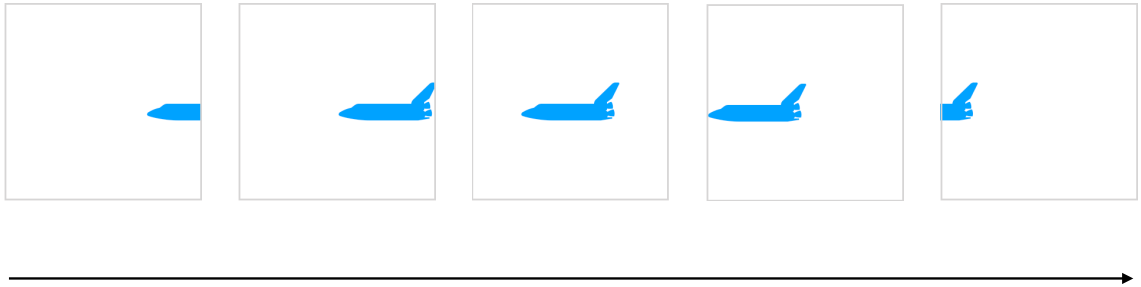




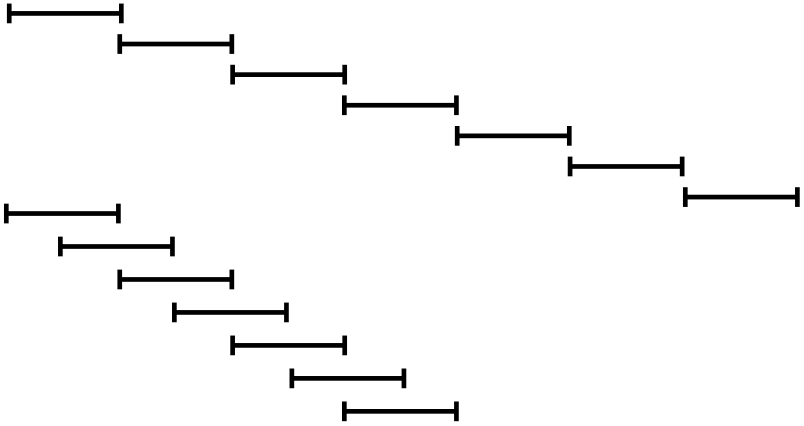


https://medium.com/@RobertGummesson/a-look-at-uiview-animation-curves-part-1-191d9e6de0ab

Keyframe Animation

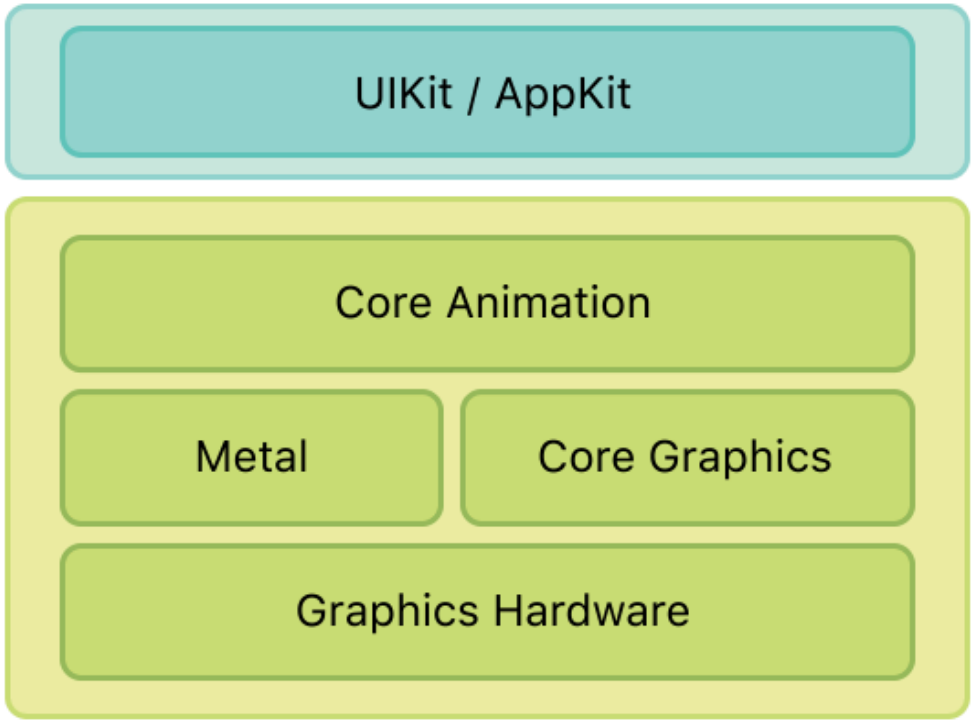


animation timing

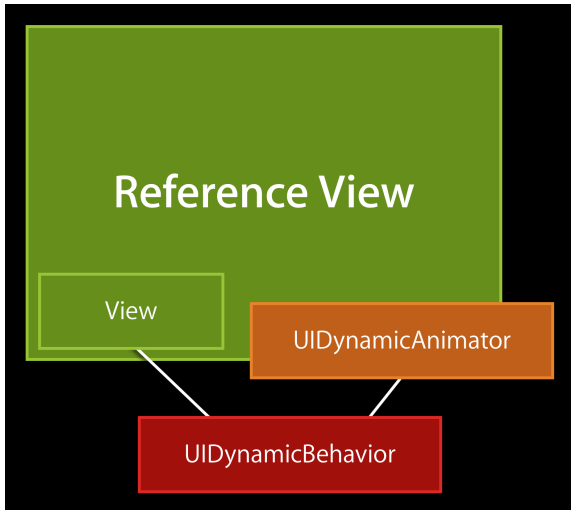


Display Link

https://developer.apple.com/documentation/quartzcore/cadisplaylink



UIDynamicAnimator



https://developer.apple.com/videos/play/wwdc2013/206/

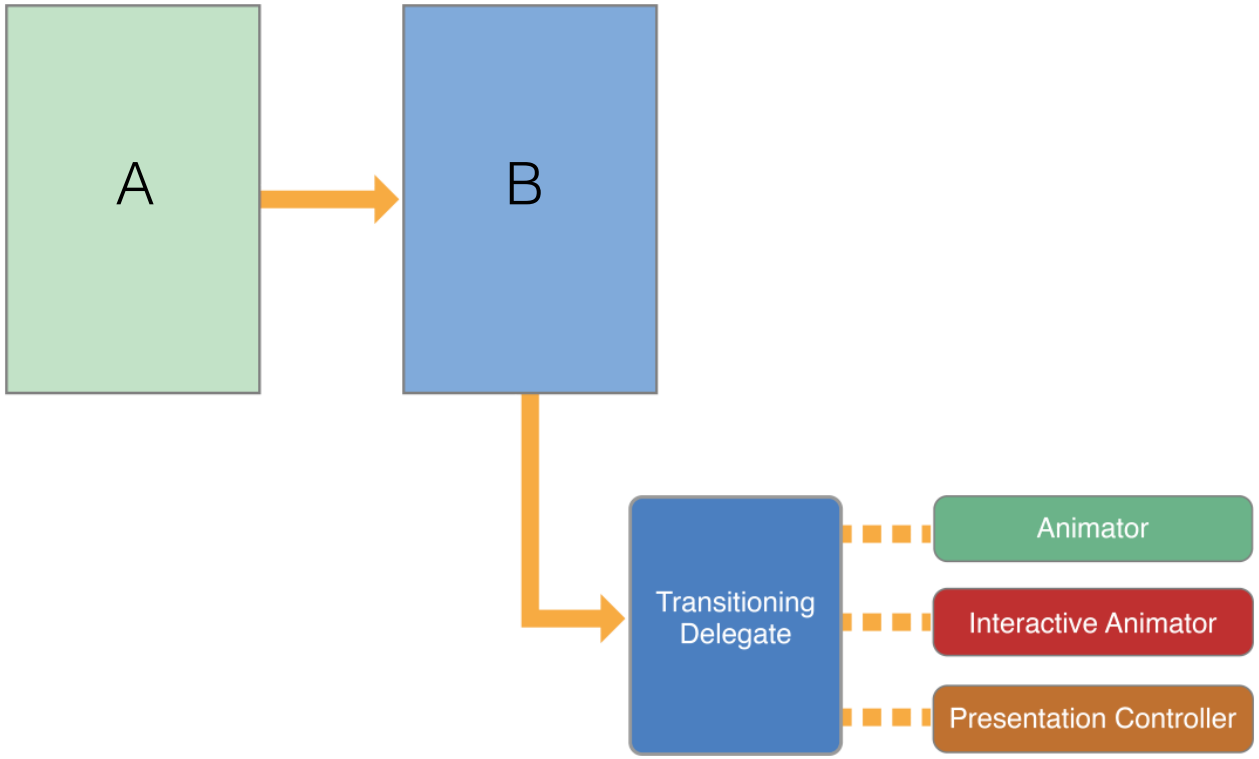
https://developer.apple.com/videos/play/wwdc2013/221/

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

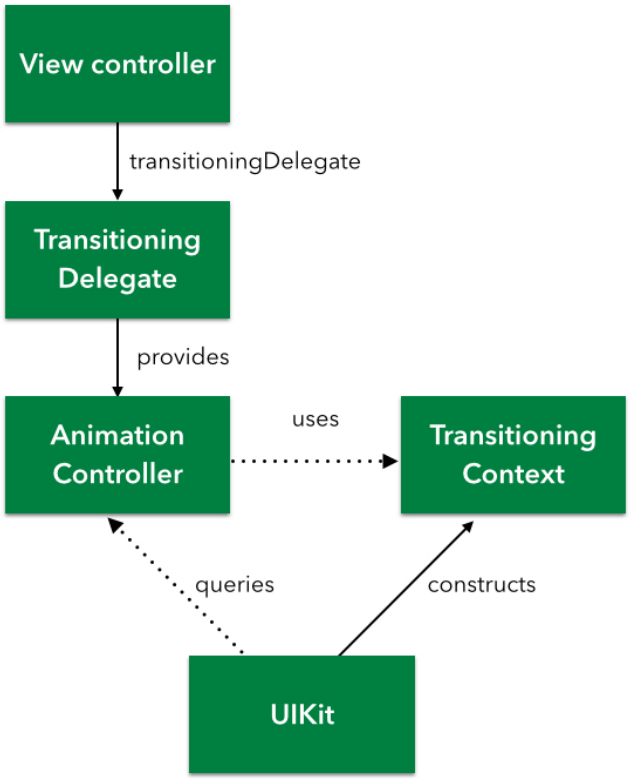
https://www.raywenderlich.com/76147/uikit-dynamics-tutorial-swift

https://developer.apple.com/library/archive/samplecode/DynamicsCatalog/Introduction/Intro.html

ViewController Transition



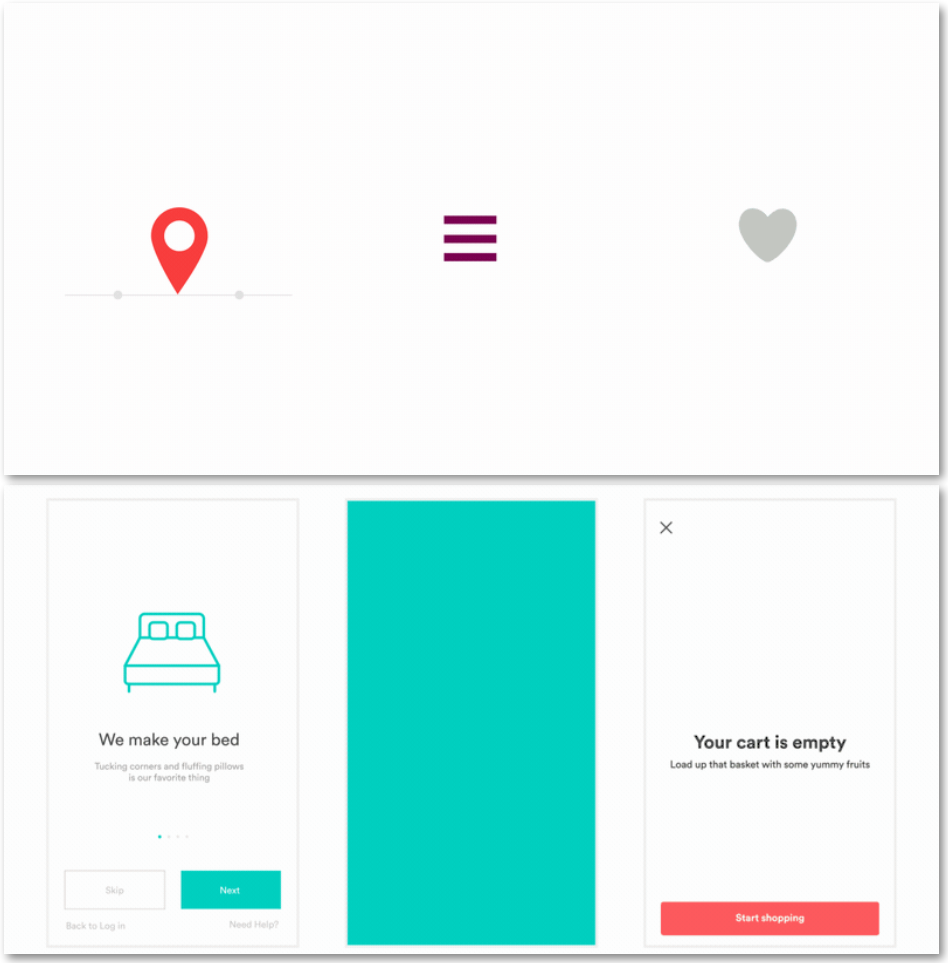
Custom Transition



https://www.raywenderlich.com/170144/custom-uiviewcontroller-transitions-getting-started

Custom Animation - Lottie

https://github.com/airbnb/lottie-ios



Core Graphics APIs

Core Graphics is C API

Quartz 2D Documentation

https://developer.apple.com/library/ios/documentation/GraphicsImaging/Conceptual/drawingwithquartz2d/Introduction/Introduction.html#//apple_ref/doc/uid/TP30001066

Quartz 2D for iOS Sample

https://developer.apple.com/library/ios/samplecode/QuartzDemo/Introduction/Intro.html#//apple_ref/doc/uid/DTS40007531

UIView:UIResponder:NSObject

눈에 보이는 모든 클래스의 슈퍼 클래스

```
CGRect frame , CGRect bounds
UIView *superview
NSArray *subviews
CALayer *layer
```

```
- drawRect:
- addSubview:
- removeFromSuperview
- setNeedsDisplay
- setNeedsLayout
+ beginAnimation: context:
```

Color Fill

UIKit에 준비된 API들

```
class MyView: UIView {
    ...
    override func draw(_ rect: CGRect) {
        UIColor.red.setFill()
        UIRectFill(self.bounds)
    }
    ...
}
```

https://developer.apple.com/library/content/documentation/2DDrawing/Conceptual/DrawingPrintingiOS/Introduction/Introduction.html

UIKit 그래픽 Context

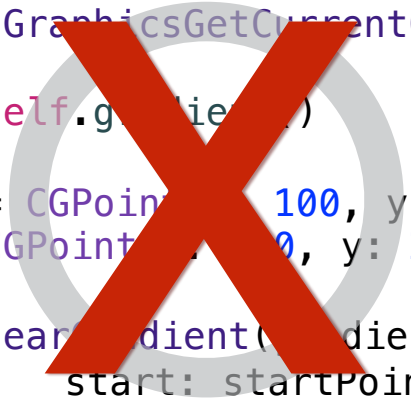
draw(rect:) 함수 내에서 처리

```
override func draw(_ rect: CGRect) {
    let context = UIGraphicsGetCurrentContext()
    context?.flush()

    //drawing code here
}
```

Don’t Call Us, We’ll call you

No context, No Drawing!

```
func buttonTouched(_ sender: UIButton) {
    let context = UIGraphicsGetCurrentContext()

    let gradient = self.g      ie    )

    let startPoint = CGPoin      100, y: 100)
    let endPoint = CGPoint      0, y: 200)

    context?.drawLinear      dient(      dient,
                             start: startPoint,
                             end: endPoint, options: [])
}
```



Don’t Call Us, We’ll call you

State changed, time to redraw

```
func buttonTouched(_ sender: UIButton) {

    myView.setNeedDisplay()
}
```

미션:

UIView에서 상속받는 GradientView 클래스를 생성하고

IB에서 View에 대한 Custom Class로 지정하세요.

GradientView 클래스 -drawRect: 를 새로 구현합니다.

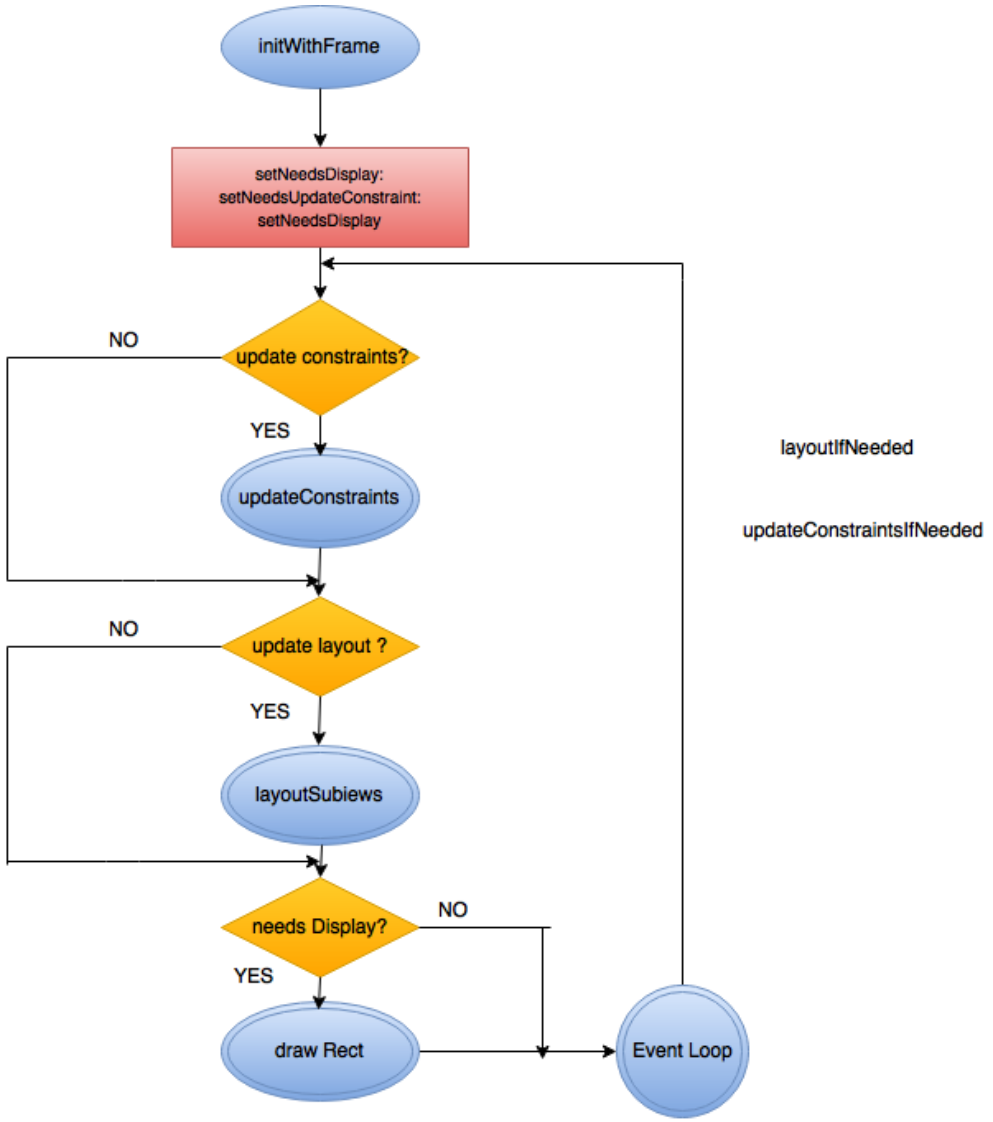
배경을 원하는 시작/종료색 gradient로 색칠해보세요.

Render Loop

Draw or Die

사용 목적	레이아웃	디스플레이	제약사항
세부 업데이트 처리를 구현	layoutSubviews	draw	updateConstraints
다음 업데이트 주기에 다시 그리도록	setNeedsLayout	setNeedsDisplay	setNeedsUpdateContraints invalidateIntrinsicContentSize
dirty 해서 업데이트를 즉시할 때	layoutIfNeeded		updateConstraintsIfNeeded
뷰 업데이트가 되는 경우 액션	addSubview frame/bounds 변경 화면 스크롤 화면 회전	bounds 변경	activate/deactivate constraint's value or priority 뷰 제거

Render



http://tech.gc.com/demystifying-ios-layout/

Render Loop