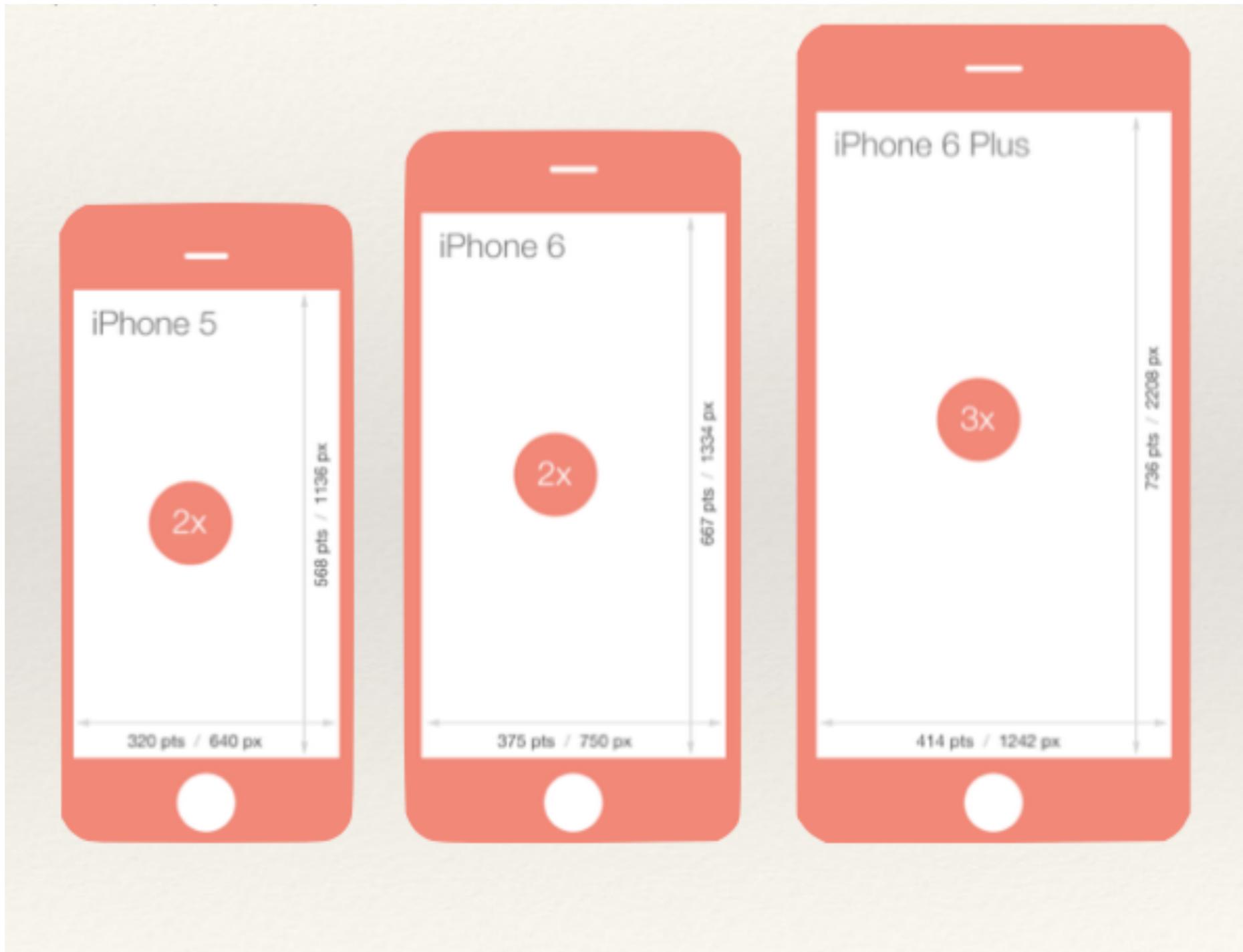

UI Base Guide

강사 주영민

Resolutions

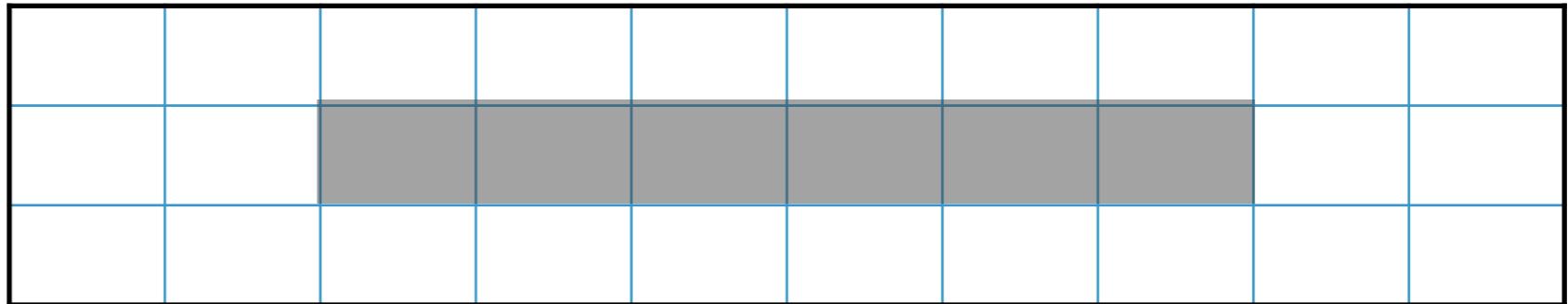
- Pixel : 실제 이미지 사이즈
- Point : 화면에 표시되는 이미지 사이즈

Resolutions

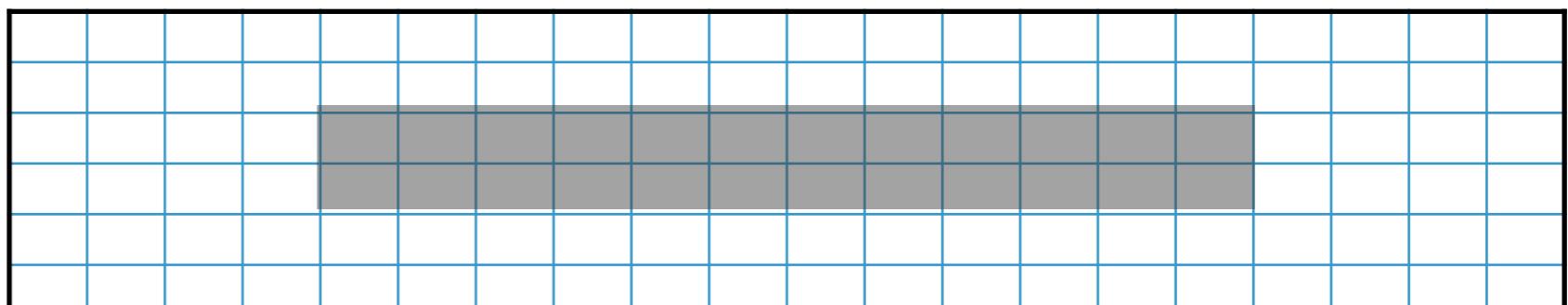


iPhone pixel Size

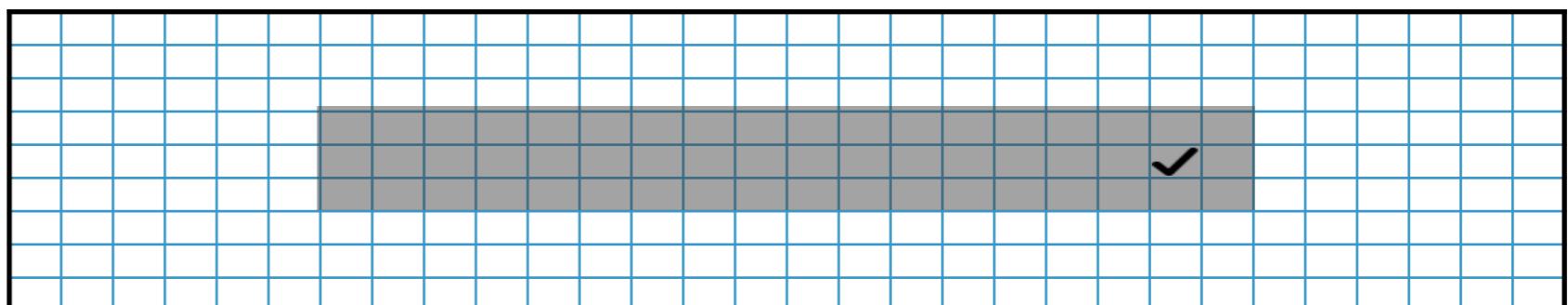
Original iPhone



@2x

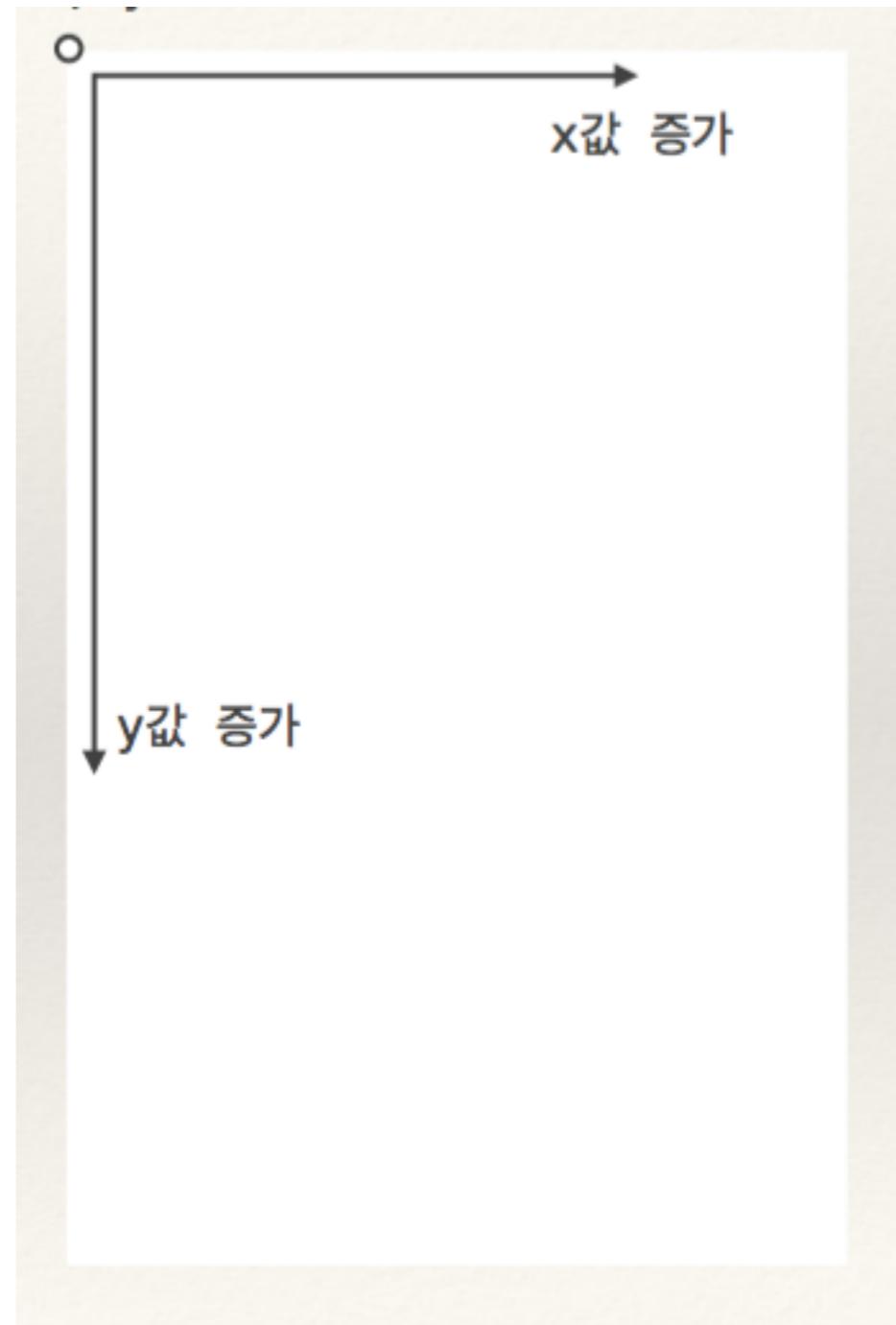


@3x



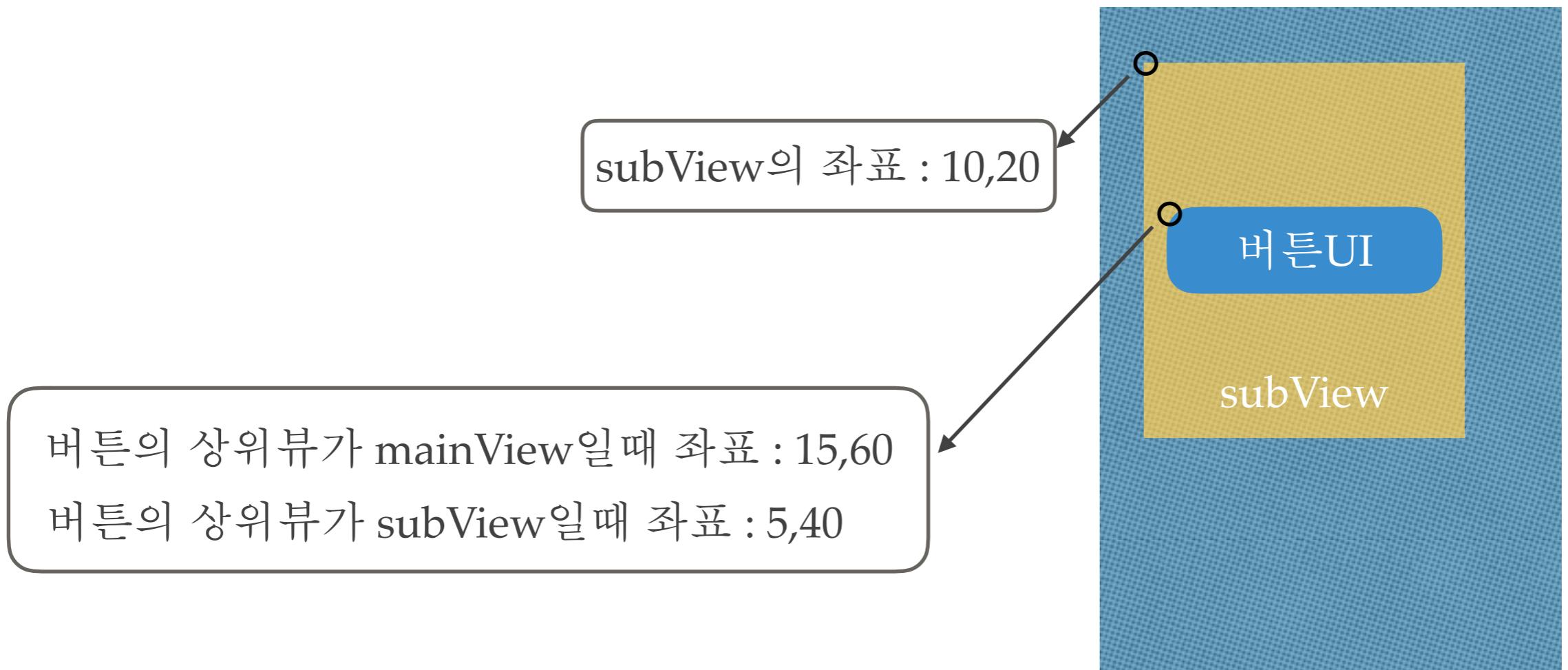
좌표계 (Frame-Base)

- 좌표계 : View기준 좌측 상단이 0, 0이다.



좌표계 (Frame-Based)

- 뷰의 위치는 상대적으로 상위뷰를 기준으로 위치를 잡는다



UI Code

강사 주영민

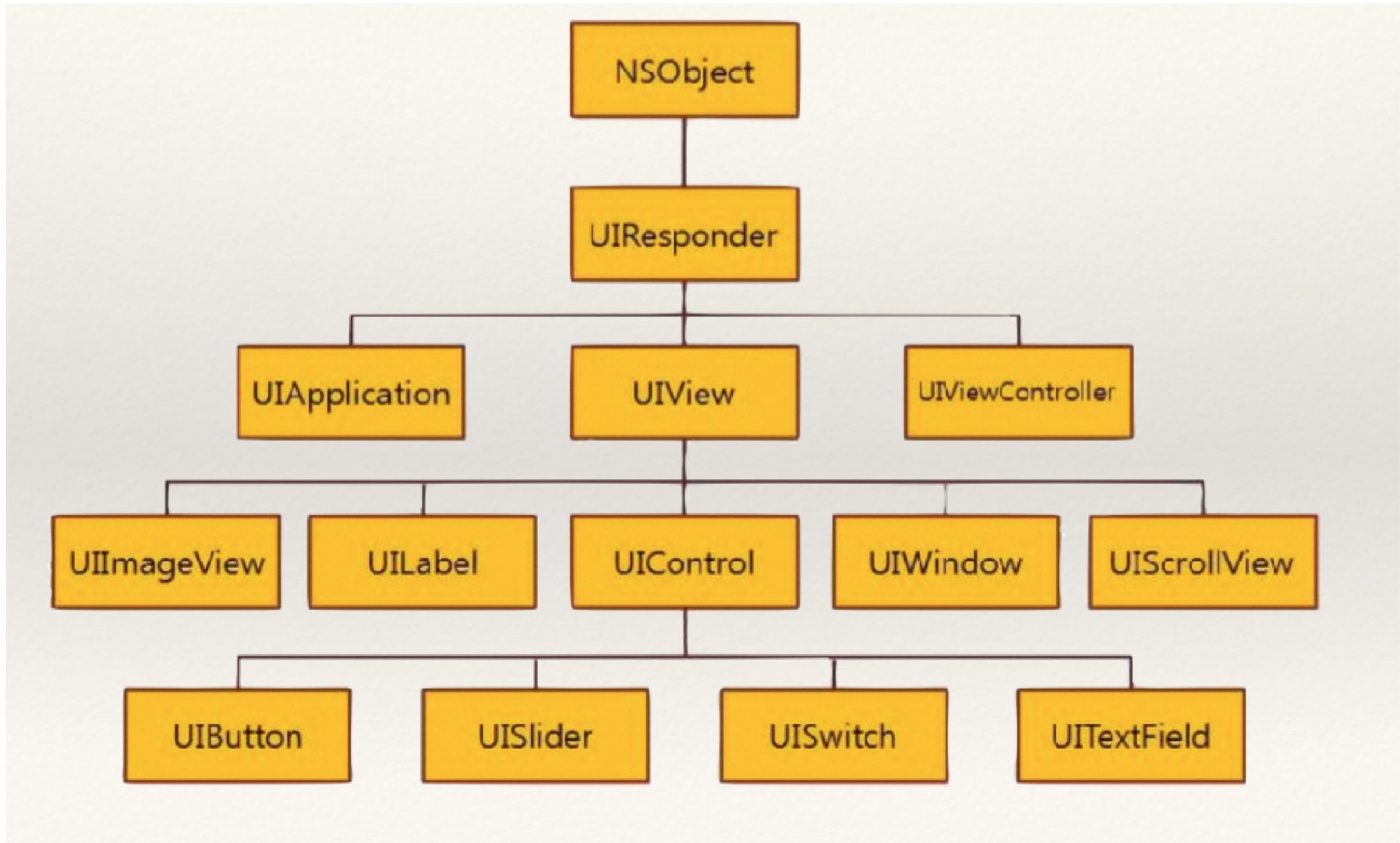
What is Framework

- 애플리케이션 프레임워크(Application Framework)는 프로그래밍에서 특정 운영 체제를 위한 **응용 프로그램 표준 구조를 구현하는 클래스와 라이브러리 모임**이다. 간단하게 프레임워크라고도 부른다.
- 재사용할 수 있는 수많은 코드를 프레임워크로 통합함으로써 개발자가 새로운 애플리케이션을 위한 표준 코드를 다시 작성하지 않아도 같이 사용된다.
- 프레임워크의 구현은 객체 지향 프로그래밍 기법이 사용되고 있는 응용 프로그램 고유의 클래스가 프레임워크의 기존 클래스를 상속할 수 있다.

UIKit Framework

- Cocoa Touch Framework에 추가된 UI관련 기능의 클래스가 모여있는 Framework
- `import UIKit`

UI Class Hierarchy



UIResponder

- The UIResponder class defines an interface for objects that respond to and handle events
 - 주요 항목

```
open func becomeFirstResponder() -> Bool  
open func resignFirstResponder() -> Bool
```

```
open func touchesBegan(_ touches: Set<UITouch>, with event: UIEvent?)
```

```
open func touchesMoved(_ touches: Set<UITouch>, with event: UIEvent?)
```

```
open func touchesEnded(_ touches: Set<UITouch>, with event: UIEvent?)
```

```
open func touchesCancelled(_ touches: Set<UITouch>, with event: UIEvent?)
```

Quick Help - 다시 한번!

- command + shift + O

UIView

- 가장 기본이 되는 View
- UIComponent들의 조합으로 화면이 구성되며 UIView를 상속 받았다. 즉 iOS 화면구성은 UIView의 집합으로 되어 있다.

UIView Class

- 주요 항목 (file 확인 하기)

```
public init(frame: CGRect)
open var tag: Int
open var layer: CALayer { get }
open var frame: CGRect
open var bounds: CGRect
open var isMultipleTouchEnabled: Bool
open var clipsToBounds: Bool
open var backgroundColor: UIColor?
open var isHidden: Bool
open var contentMode: UIViewContentMode

open func addSubview(_ view: UIView)
open func insertSubview(_ view: UIView, at index: Int)
open func removeFromSuperview()
open func layoutIfNeeded()
```

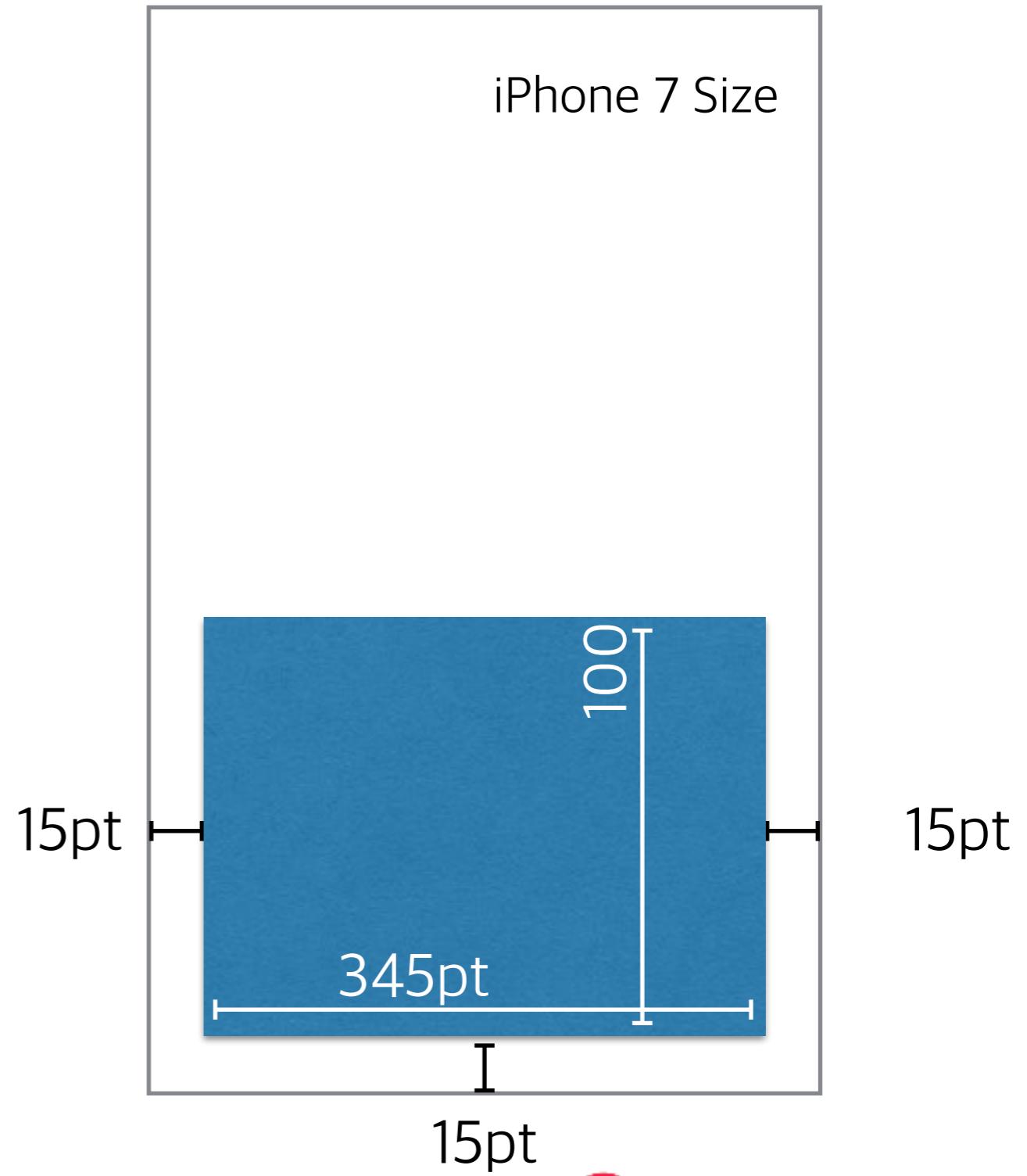
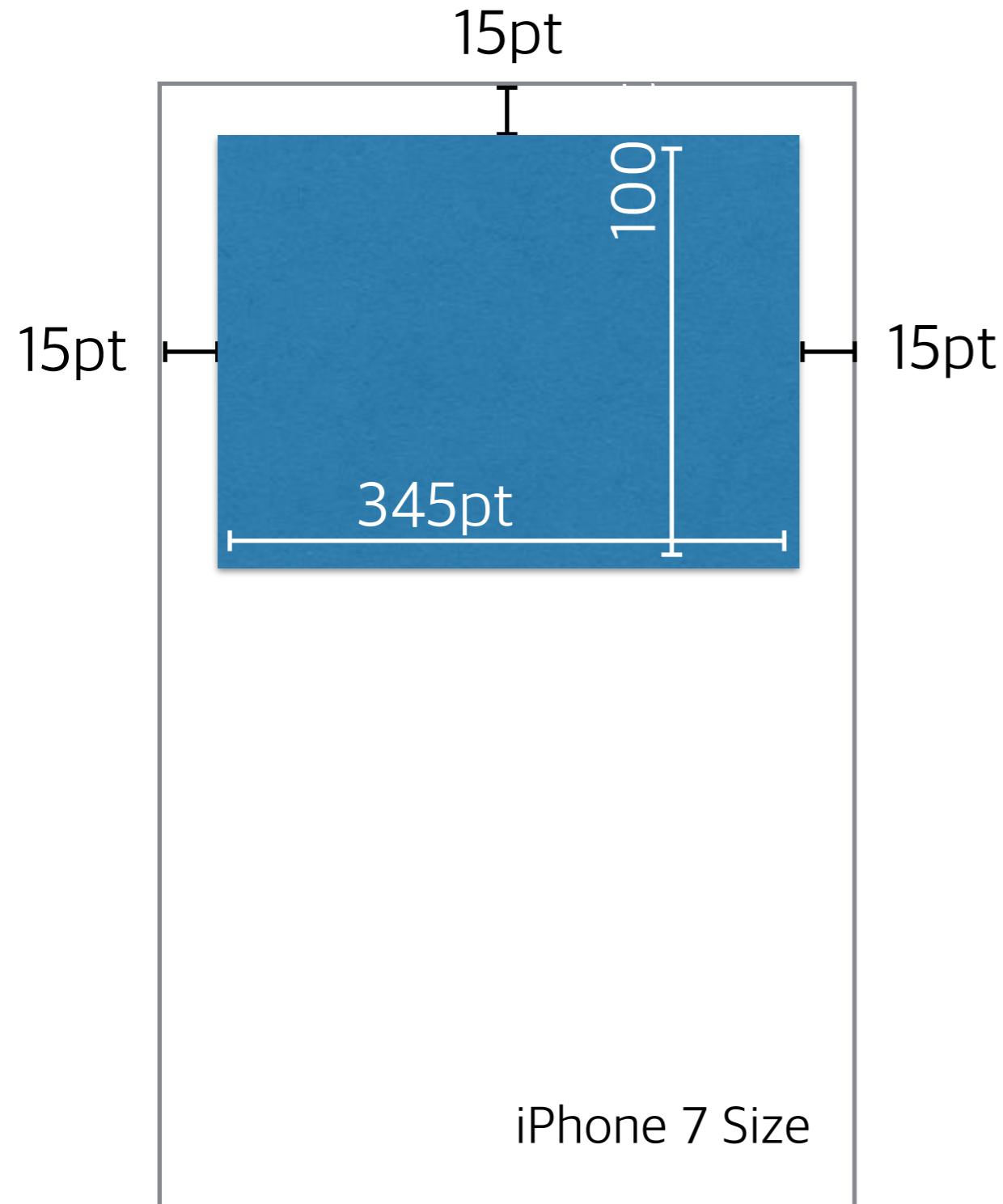
UIView 예제

```
//객체 생성  
let newView = UIView(frame: CGRect(x: 0, y: 0, width: 100,  
height: 100))  
  
//배경 색변경  
newView.backgroundColor = UIColor.green  
  
//뷰 투명도 50%  
newView.alpha = 0.5  
  
//뷰 추가  
self.view.addSubview(newView)
```

View를 만들어 봅시다.

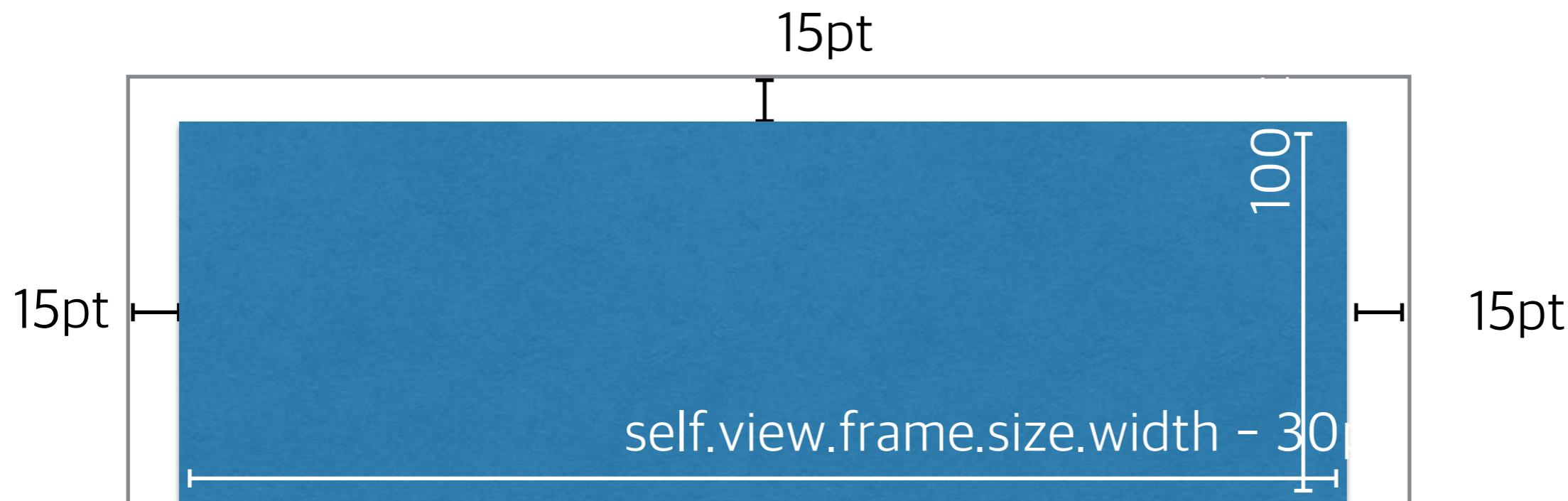
- 코드를 이용해서 View만들기
 - View에 addSubView 하기
- * View 만들기 tip! (인스턴스 생성, 프레임 설정, addSubView)

View 만들기 연습



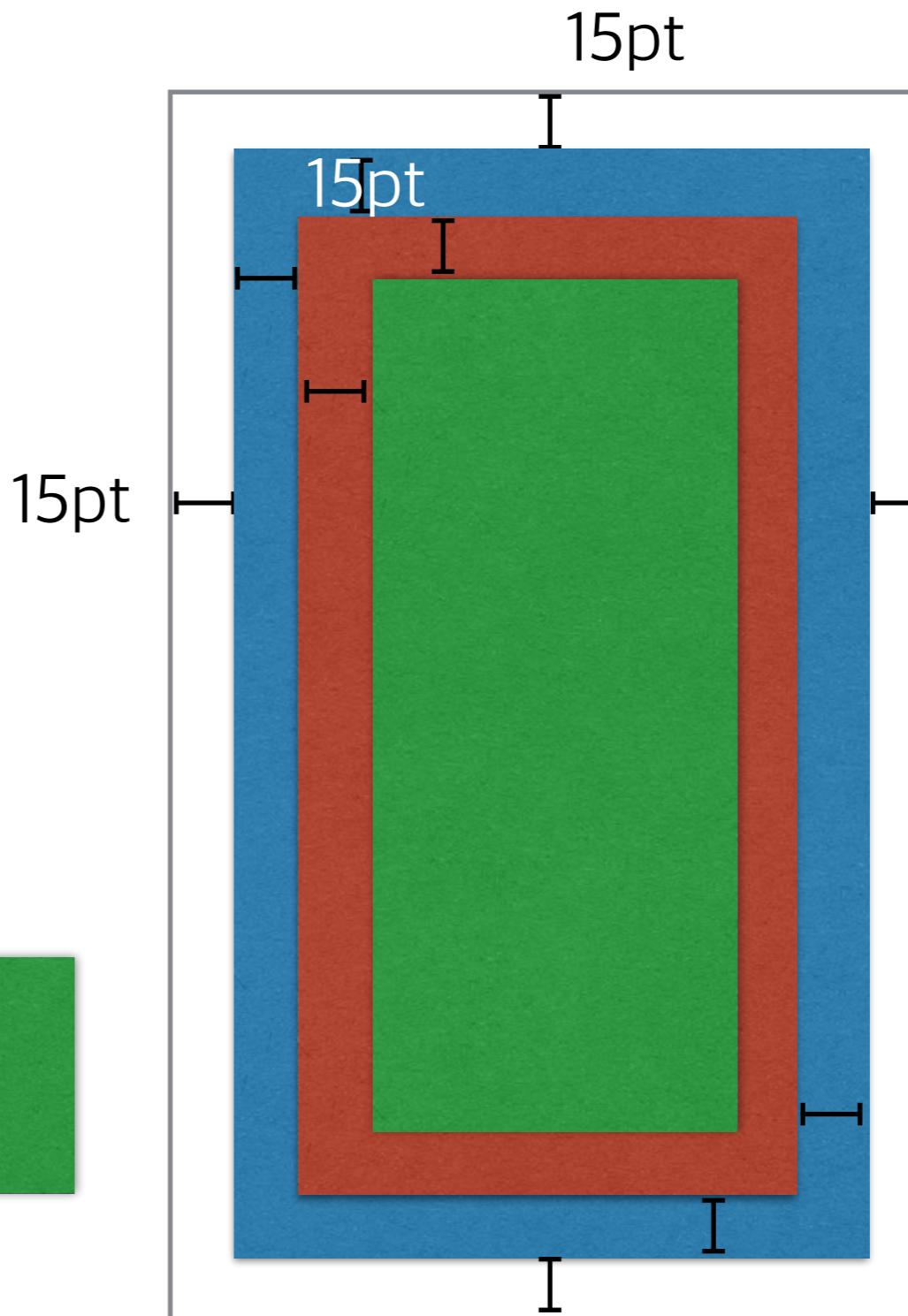
15pt

View 만들기 연습



- SuperView의 size를 참조해서 view의 size를 정한다.
- view의 가로 사이즈가 기기 사이즈에 따라 유동적으로 변한다.

View 만들기 연습



- 각 view의 상하좌우 여백은 15
- 계층 구조로 3개 view 만들기

다른 UI만들기

- UILabel은 어떻게 만들면 될까요?
- UIImageView는 어떻게 만들면 될까요?

UILabel file

- UILabel file을 봅시다.
- 어떤 속성과 어떤 메소드가 숨어 있나요?
- 추가로 봐야될 Type에는 어떤것이 있나요?

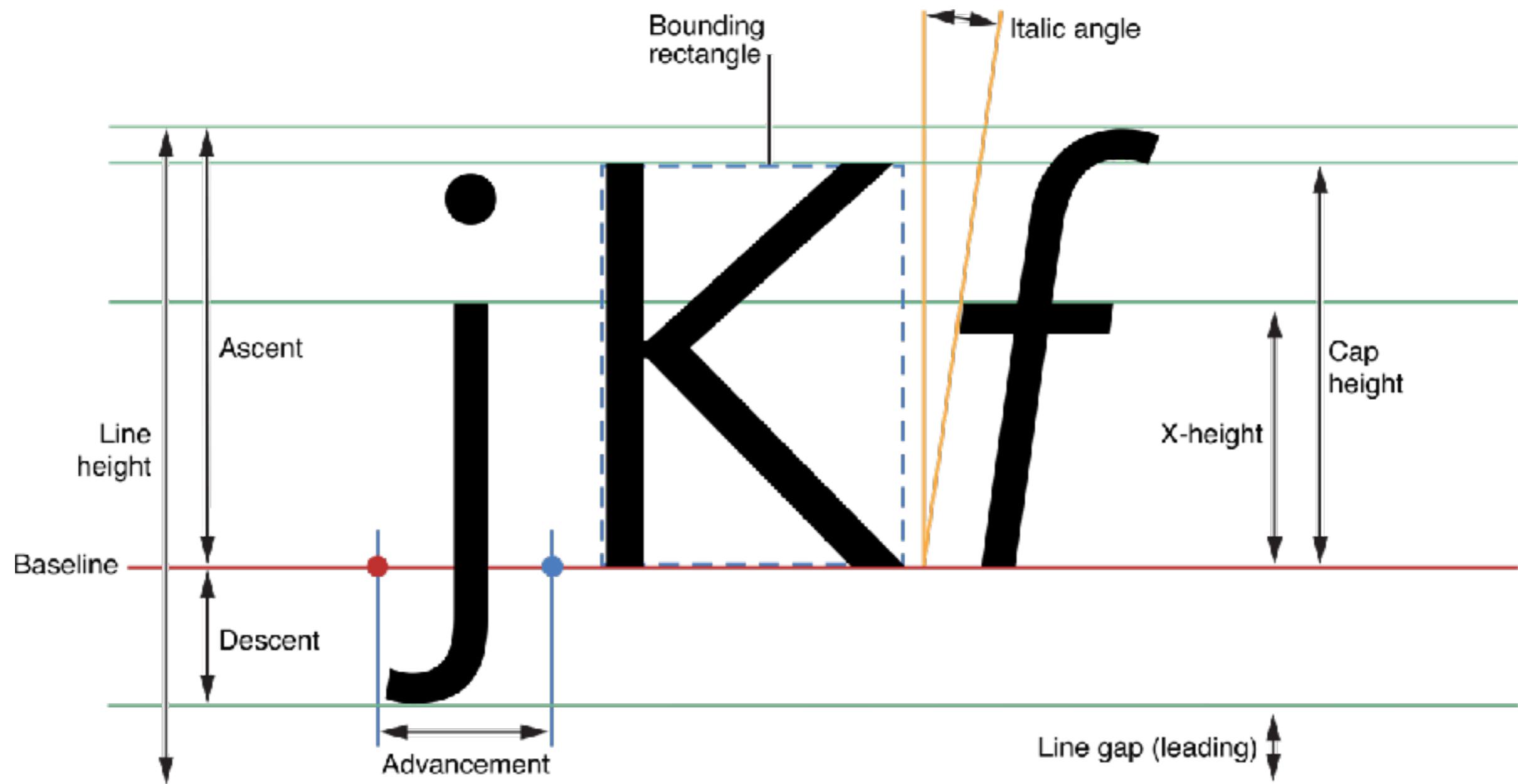
UIImageView file

- UIImageView file을 봅시다.
- 어떤 속성과 어떤 메소드가 숨어 있나요?
- 추가로 봐야될 Class Type에는 어떤것이 있나요?

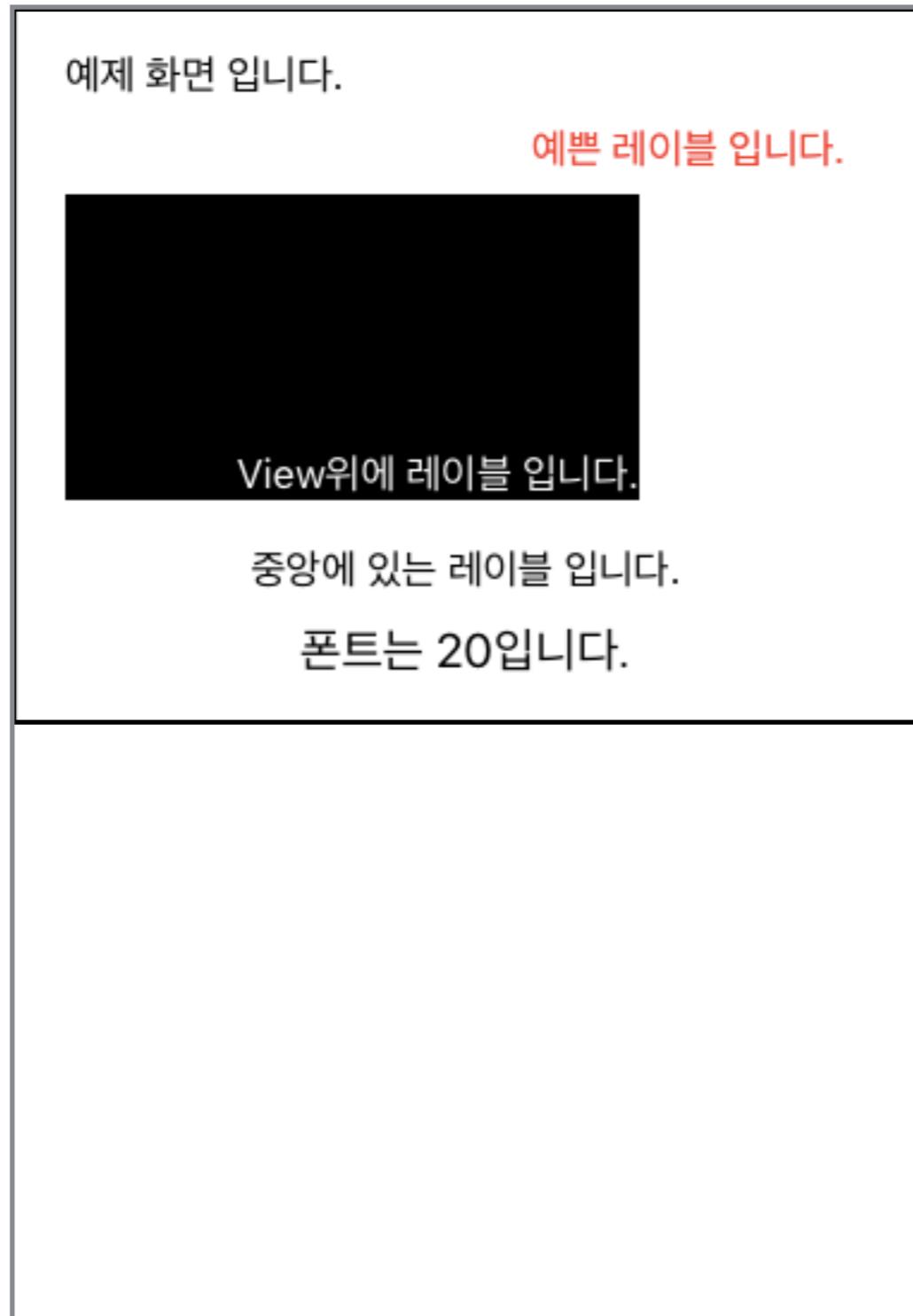
UILabel 예제

```
let newLb = UILabel(frame: CGRect(x: 0, y: 0, width: 100,  
height: 30))  
  
newLb.text = "테스트"  
  
newLb.textColor = UIColor.gray  
  
newLb.textAlignment = NSTextAlignment.center  
  
newLb.font = UIFont.systemFont(ofSize: 10)  
  
newView.addSubview(newLb)
```

UILabel Font



UI실습 다음 화면을 만들어 봅시다.



UIImageView 예제

```
let newImgV = UIImageView(frame: CGRect(x: 5, y: 5, width: 90,  
height: 90))  
newImgV.image = UIImage(named:"img.png")  
newImgV.contentMode = UIViewContentMode.scaleAspectFit  
newView.addSubview(newImgV)
```

Add Image File

- 단일 이미지 리소스 추가
- Assets.xcassets 에 추가

Assets

강사 주영민

Asset Catalogs

- 효율적인 리소스관리를 위해 사용
- 각 다른 디바이스에 따라 다른 리소스를 매칭 해두고 실행시 적절한 리소스가 선택되어 표시된다.
- trait collection의 정보에 따라 적절한 이미지를 찾아준다.

Size Class

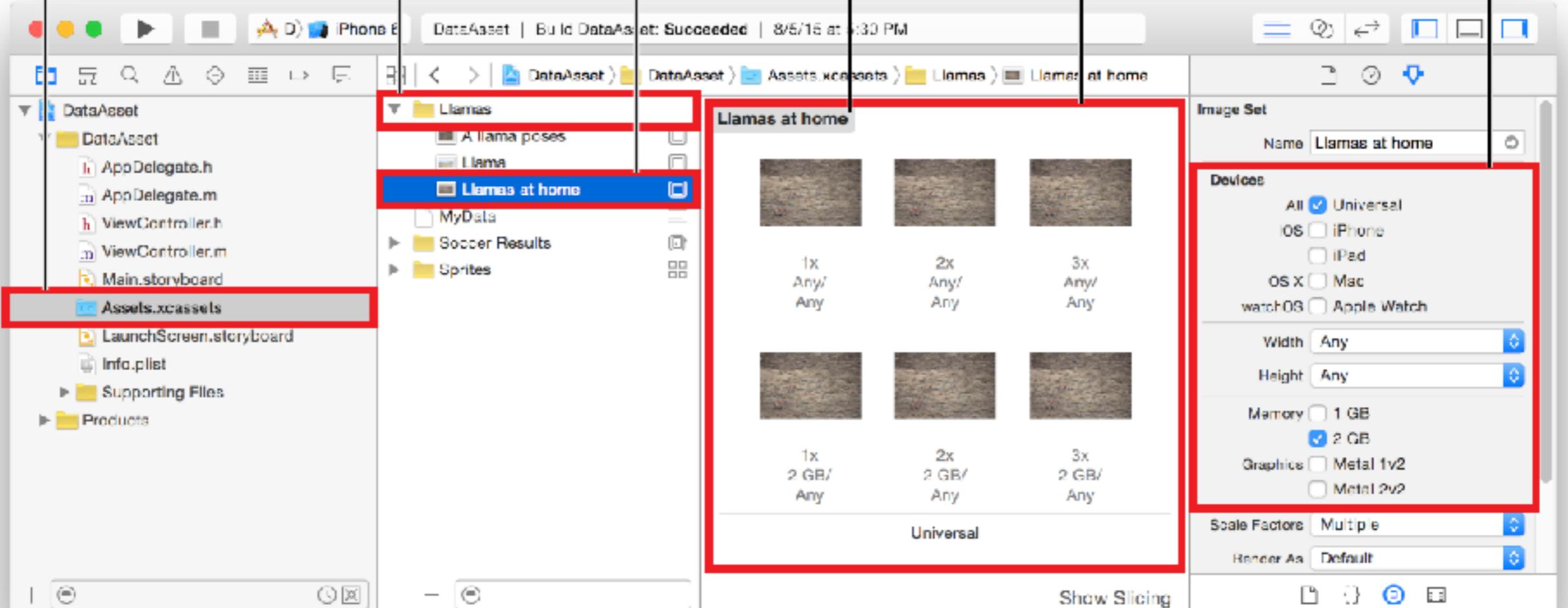
	Regular	Compact
Regular	iPad 가로모드, iPad 세로모드	iPhone 세로모드
Compact	iPhone 6 Plus 가로모드	iPhone 가로모드*
* iPhone 6 Plus 제외		

Trait Collection

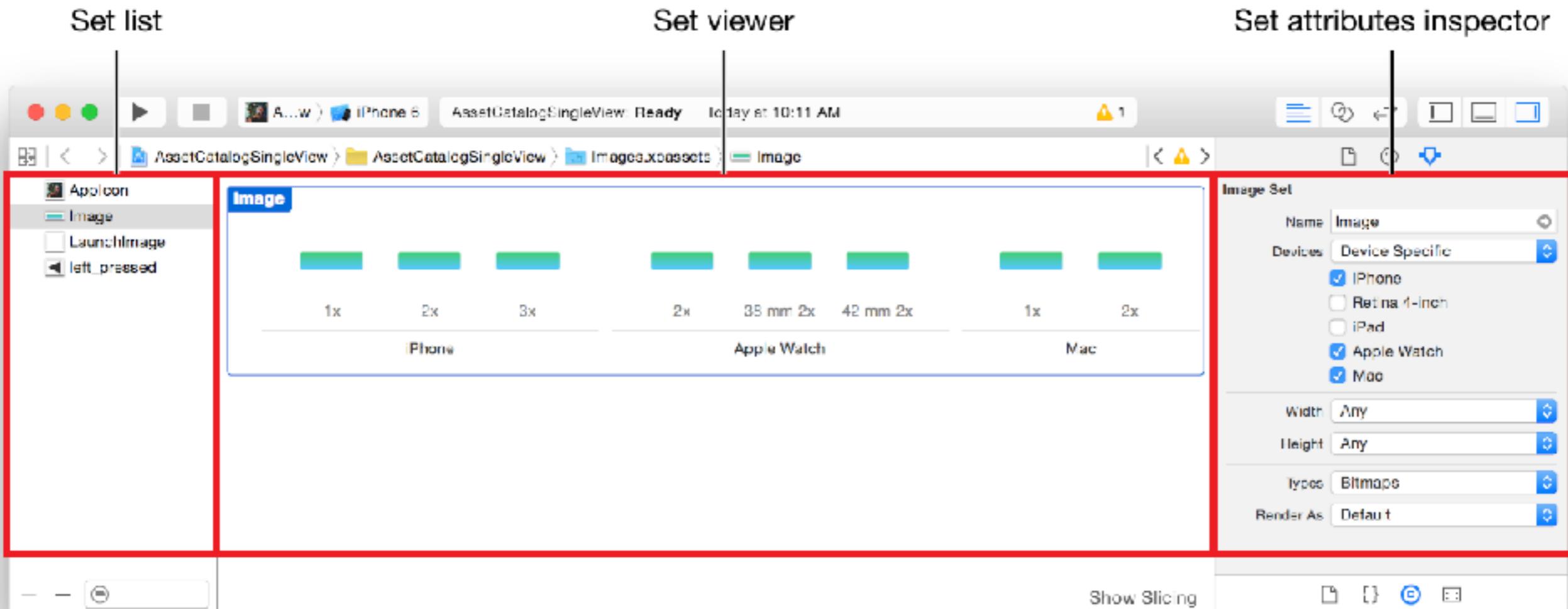
horizontalSizeClass	Compact
verticalSizeClass	Regular
userInterfaceIdiom	Phone
displayScale	2.0

*UIViewController의 traitCollection 프로퍼티로 정보를 구할 수 있다.

Asset Catalogs Contents

Asset catalog	Group	Asset	Asset name	Asset files	Attributes
 A screenshot of the Xcode interface showing the Asset Catalog editor. The left sidebar lists project files: DataAsset, DataAsset.h, DataAsset.m, Viewcontroller.h, Viewcontroller.m, Main.storyboard, Assets.xcassets, LaunchScreen.storyboard, Info.plist, Supporting Files, and Products. The Assets.xcassets file is selected and highlighted with a red box. Inside the Assets.xcassets folder, there is a folder named "Llamas". Within "Llamas", there are three subfolders: "A llama poses", "Llama", and "Llamas at home". The "Llamas at home" folder is also highlighted with a red box. The main canvas displays the contents of "Llamas at home". It shows two rows of images. The top row has three images labeled "1x Any/ Any", "2x Any/ Any", and "3x Any/ Any". The bottom row has three images labeled "1x 2 GB/ Any", "2x 2 GB/ Any", and "3x 2 GB/ Any". Below these rows is a "Universal" section. In the bottom right corner of the main canvas, there is a "Show Slicing" button. To the right of the main canvas is a panel titled "Image Set" which shows the name "Llamas at home" and a preview of the first image. Further down, there is a "Devices" section with checkboxes for "All" (checked), "iPhone" (unchecked), "iPad" (unchecked), "Mac" (unchecked), and "Apple Watch" (unchecked). There are also sections for "Width" (Any), "Height" (Any), "Memory" (1 GB checked, 2 GB checked), "Graphics" (Metal 1v2 unchecked, Metal 2v2 unchecked), "Scale Factors" (Multiple), and "Render As" (Default). DataAsset Build DataAsset: Succeeded 8/5/16 at 4:30 PM DataAsset Assets.xcassets Llamas Llamas at home DataAsset DataAsset.h DataAsset.m Viewcontroller.h Viewcontroller.m Main.storyboard Assets.xcassets LaunchScreen.storyboard Info.plist Supporting Files Products Llamas A llama poses Llama Llamas at home MyData Soccer Results Sprites Llamas at home 1x Any/ Any 2x Any/ Any 3x Any/ Any 1x 2 GB/ Any 2x 2 GB/ Any 3x 2 GB/ Any Universal Show Slicing Image Set Name Llamas at home Devices All <input checked="" type="checkbox"/> Universal OS <input type="checkbox"/> iPhone <input type="checkbox"/> iPad OS X <input type="checkbox"/> Mac watchOS <input type="checkbox"/> Apple Watch Width Any Height Any Memory <input type="checkbox"/> 1 GB <input checked="" type="checkbox"/> 2 GB Graphics <input type="checkbox"/> Metal 1v2 <input type="checkbox"/> Metal 2v2 Scale Factors Multiple Render As Default					

Asset Catalogs Contents



Asset Catalogs can include

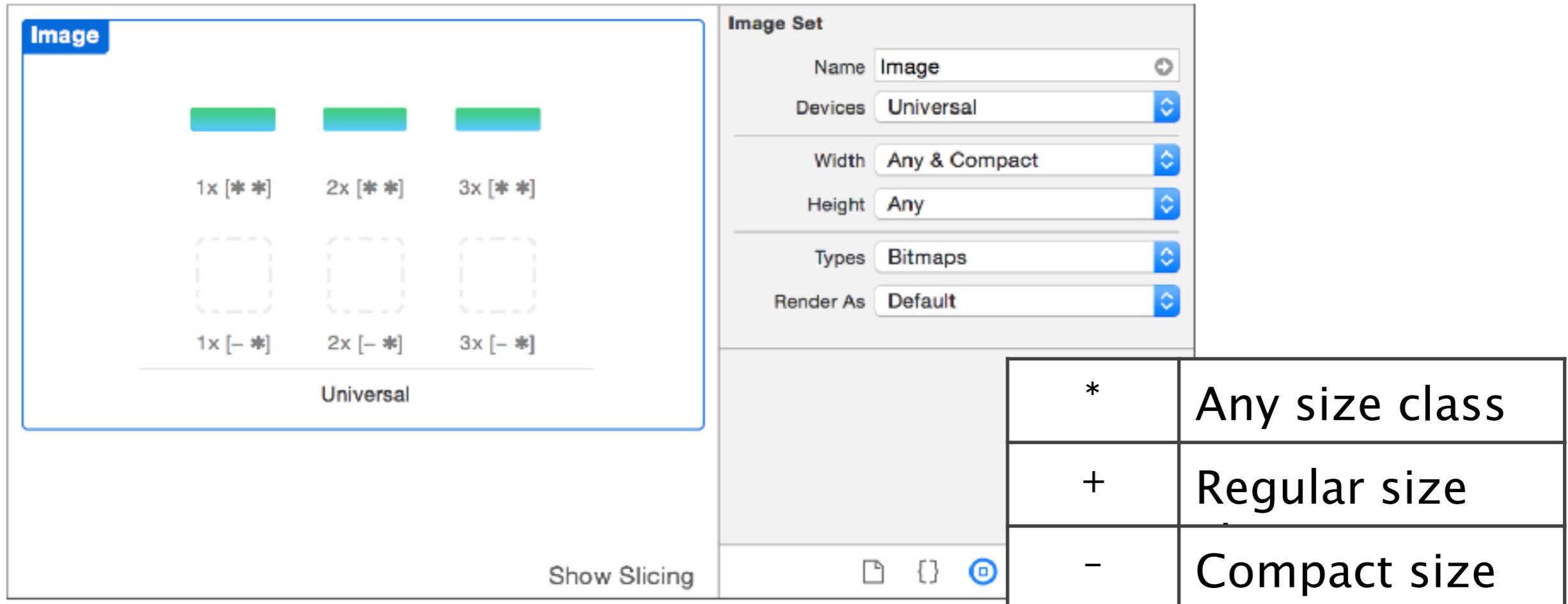
- **Data sets** : Used for any type of date file except binary executable file.
- **Folders** : Used to group assets or other folders.
- **Image sets** : Used for most types of images, an image set contains all the versions, or representations, of an image that are necessary to support various devices and scale factors.
- **iOS App icons** : App icon sets contain all the representations of the icon that iOS apps must provide to be displayed on a device's Home screen and in the App Store.
- **Launch images** : Launch image sets contain all the representations of the static launch image that iOS apps must provide to be temporarily displayed when the app is launched.
- **OS X icons, Sprite Atlas, Watch complications, watchOS App icons.**

Asset Catalogs 사용법

- Create Asset
- Add New Asset
- Grouping assets
- Custom SizeClasses
- Resizable Area Image

Asset Catalogs 사용법

- Custom SizeClasses



해상도 [가로Size, 세로Size]

Ref

- <https://goo.gl/cI94lW> //Asset Catalog Help
- <https://goo.gl/eYA5zs> //Asset Catalog Reference

ContentsMode

UIViewContentModeScaleAspectFill



Nondistorting



UIViewContentModeScaleAspectFit



Nondistorting



UIViewContentModeScaleToFill



Distorting



UI실습 : UIImageView만들기

- 추가할 이미지 다운로드
- UIImageView인스턴스 생성 및 화면에 띄우기
- contentsMode에 따라 보이는 화면 비교하기

UIControl

- UIControl에 사용자 인터렉션에 의한 응답에 대해 특별한 액션을 줄수 있게 설정하는 클래스

UIControl - 주요 항목

```
public static var normal: UIControlState { get }
```

```
public static var highlighted: UIControlState { get }
```

```
public static var disabled: UIControlState { get }
```

```
public static var selected: UIControlState { get }
```

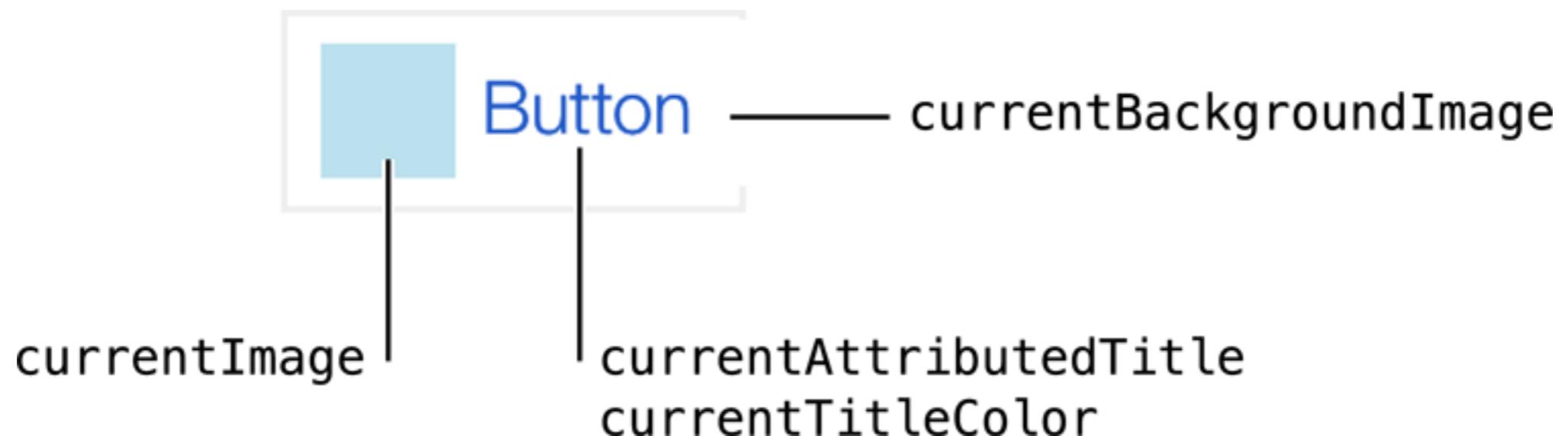
```
open var isEnabled: Bool
```

```
open var isSelected: Bool
```

```
open func addTarget(_ target: Any?,  
                    action: Selector,  
                    for controlEvents: UIControlEvents)
```

UIButton

- 사용자의 이벤트를 받아 처리해주는 UI
- 버튼 구조 : Title, Image, backgroundImage



UIButton file

- UIButton file을 봅시다.
- 어떤 속성과 어떤 메소드가 숨어 있나요?
- 추가로 봐야될 Class Type에는 어떤것이 있나요?

UIButton 예제

```
let btn = UIButton(frame: CGRect(x: 0, y: 0, width: 100,  
height: 100))  
btn.titleLabel?.text = "버튼"  
btn.setBackgroundImage(UIImage(named:"bg1.png"), for:  
UIControlState.normal)  
btn.setTitleColor(.green, for: .normal)  
btn.addTarget(self,  
    action:#selector(ViewController.btnAction(sender:)),  
    for: .touchUpInside)  
self.view.addSubview(btn)
```

ViewController파일에 btnAction 메소드가 존재 해야함

```
func btnAction(sender:UIButton) {  
    //버튼 이벤트시 액션  
}
```

UIButton 실습

1번 버튼

2번 버튼

3번 버튼

4번 버튼

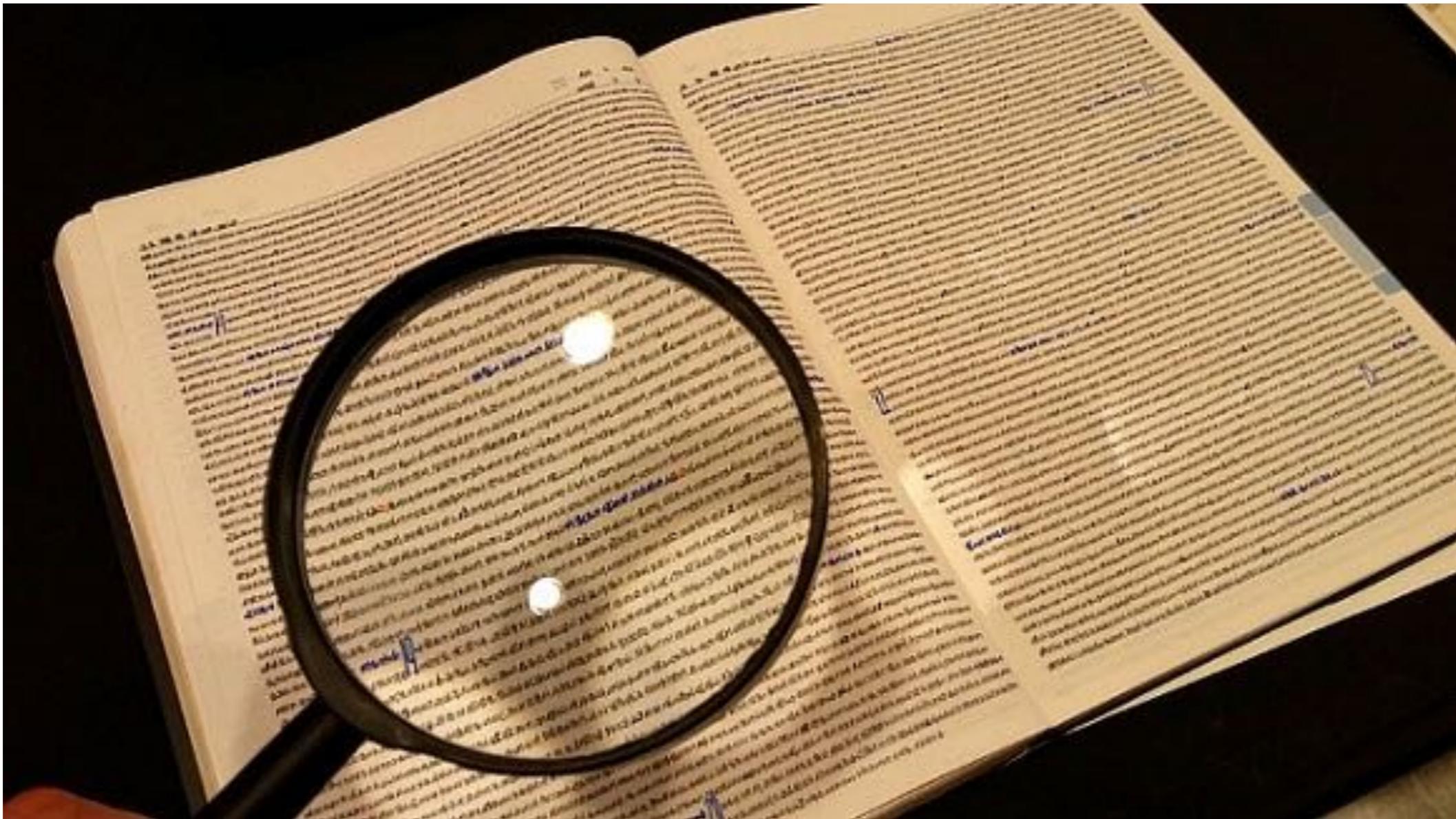
선택된 버튼은 : 2번 버튼

UITextField

- 사용자 텍스트 입력을 위한 UI Component.



UITextField



UITextField - 실습



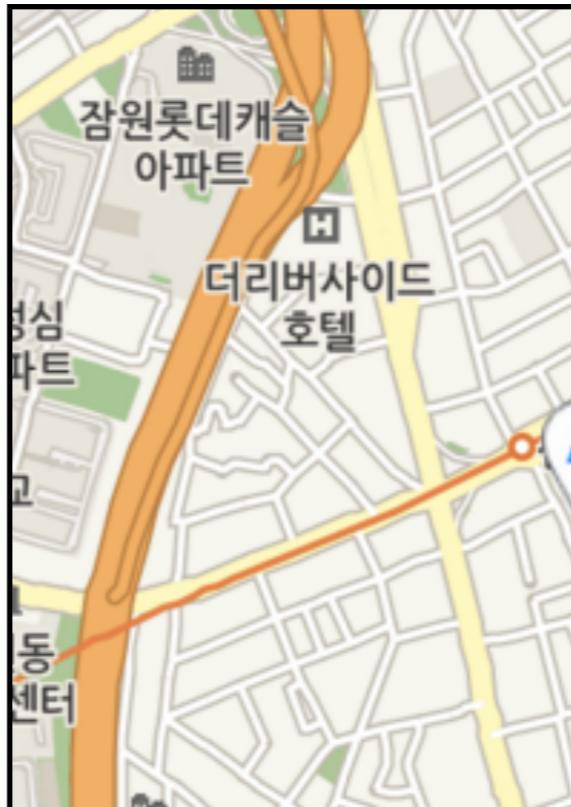
1. 텍스트 필드에서 텍스트 입력
2. 확인 버튼 클릭
3. 아래 레이블에 입력된 텍스트 보여주기

UITextFieldDelegate

- Protocol을 이용한 Delegate Pattern 사용
- 텍스트 필드의 이벤트를 Controller에게 전달하기 위해 사용
- .delegate 프로퍼티에 컨트롤러 인스턴스 할당!!
- 일단은 사용법부터 익히기

UIScrollView

- ViewSize보다 확장된 뷰를 보기위한 View
- UIScrollView에 추가된 View는 ContentView위에 추가 된다.

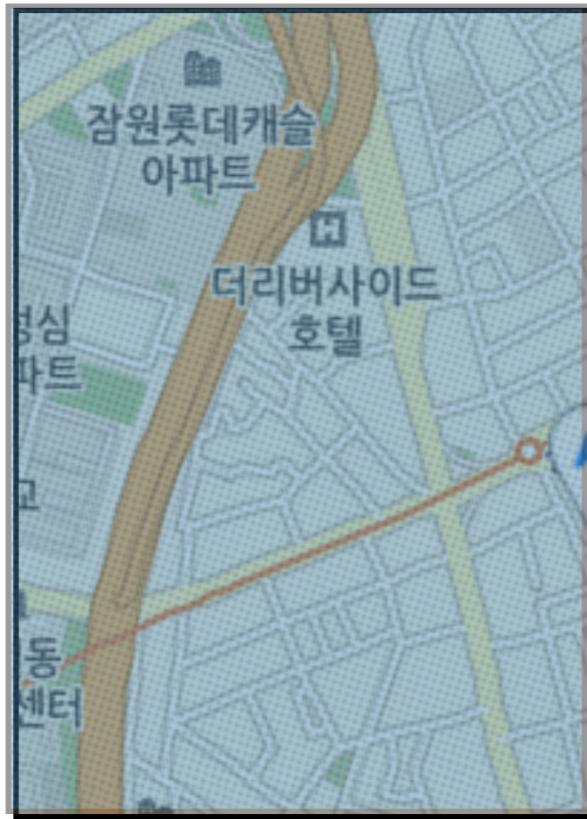


UIScrollView - ContentView



UIScrollView - 스크롤

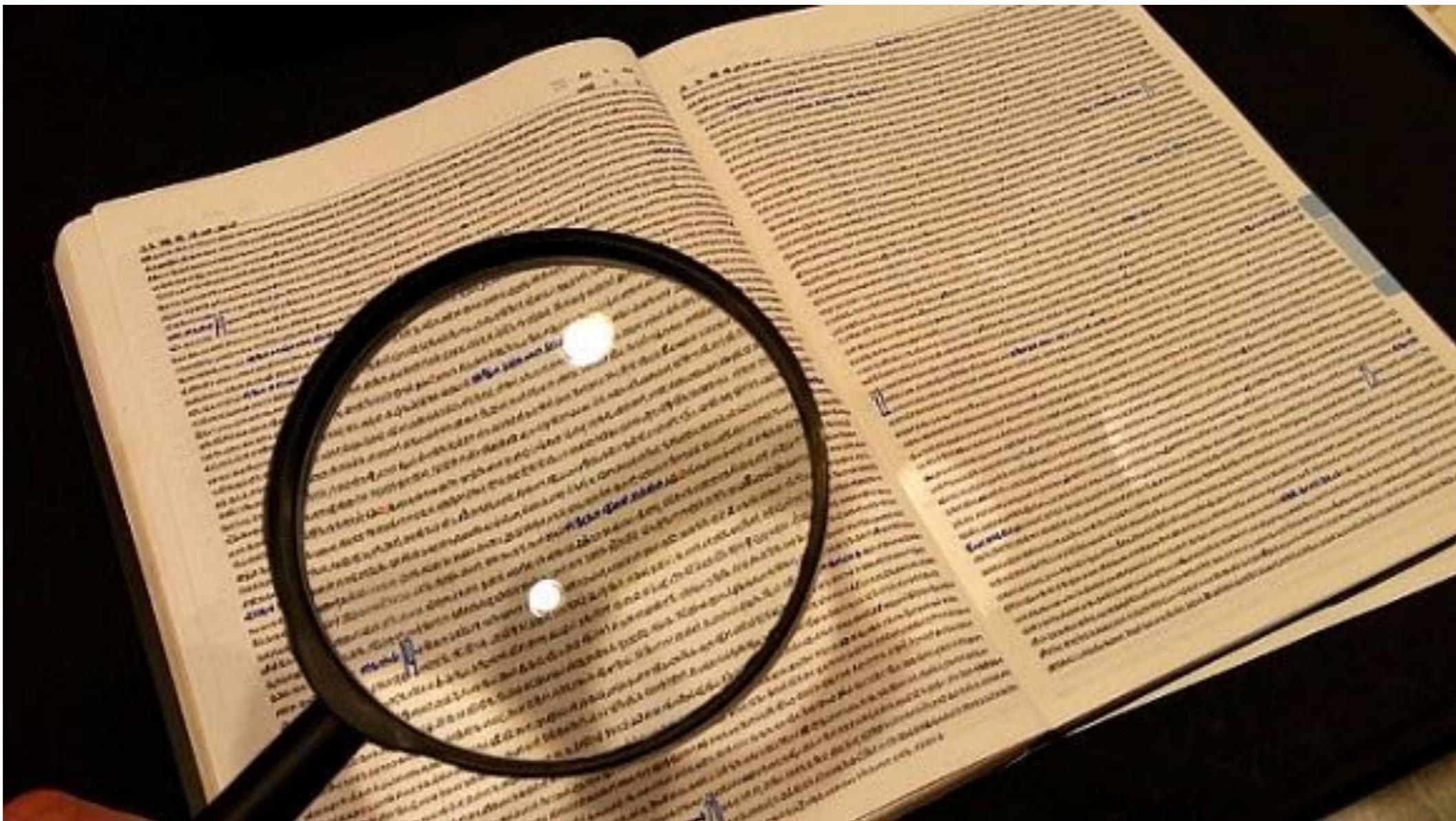
실화면



내부동작



UIScrollView



UIScrollView 만들기

1. 스크롤뷰 추가 (Autolayout 지정)
2. 콘텐츠 뷰 추가하기
 - UIView 추가
 - add top, bottom, leading, and trailing edges
3. (선택 사항) 가로 스크롤을 사용하지 않으려면 콘텐츠 뷰의 너비를 스크롤보기의 너비와 같게 설정합니다.
4. (선택 사항) 수직 스크롤을 사용하지 않으려면 컨텐츠보기의 높이를 스크롤보기의 높이와 같게 설정하십시오.

UIScrollView - 실습

가로 3페이지 스크롤 뷰 만들기(컨텐츠는 이미지뷰)

