

---

# UI Base Guide

---

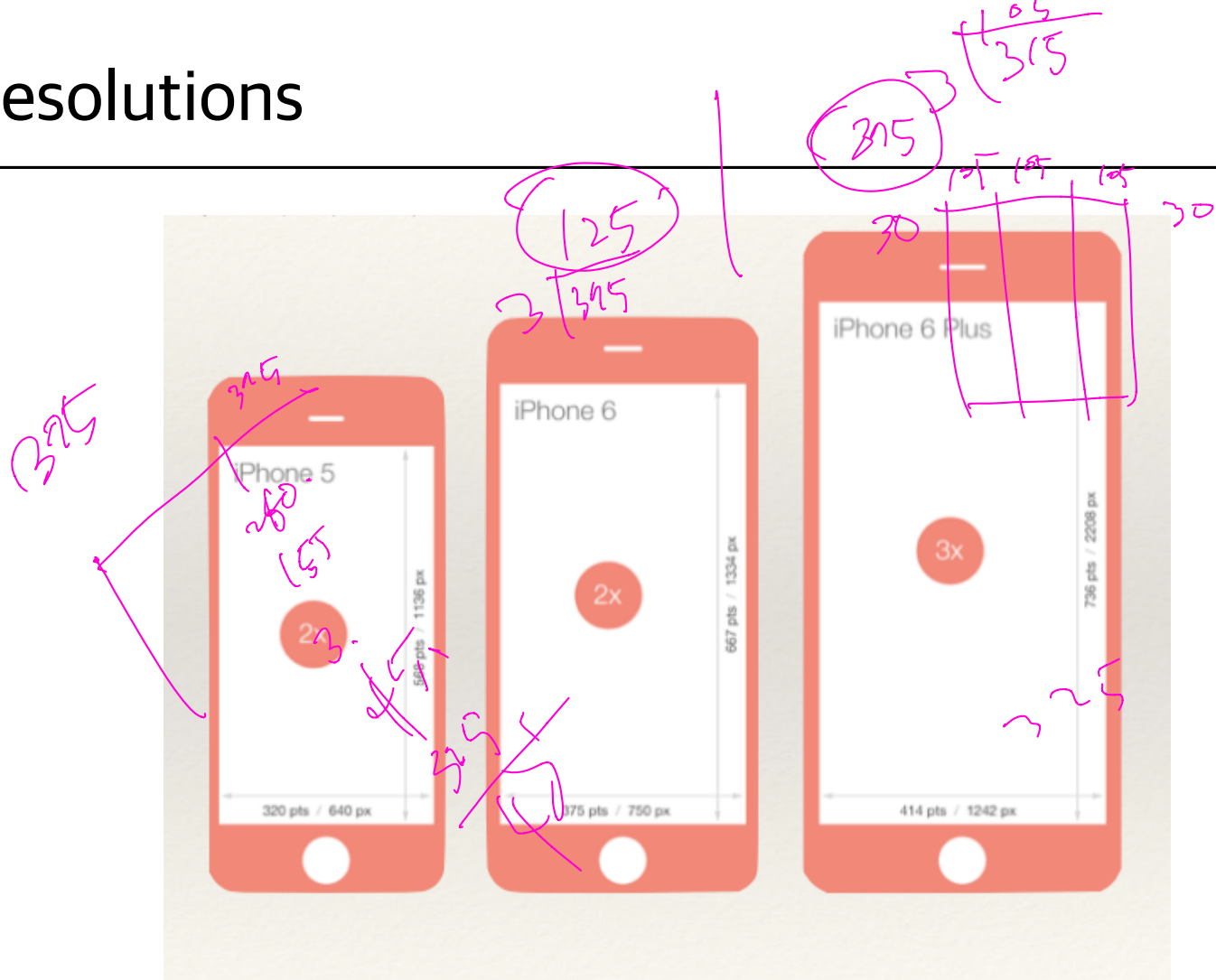
강사 주영민

# Resolutions

---

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

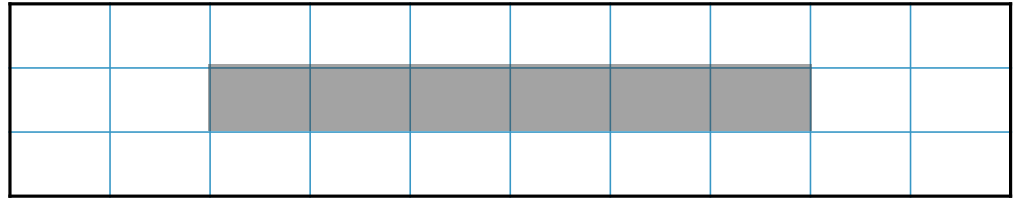
# Resolutions



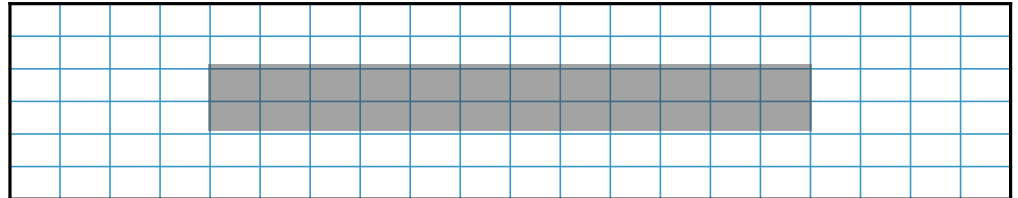
# iPhone pixel Size

---

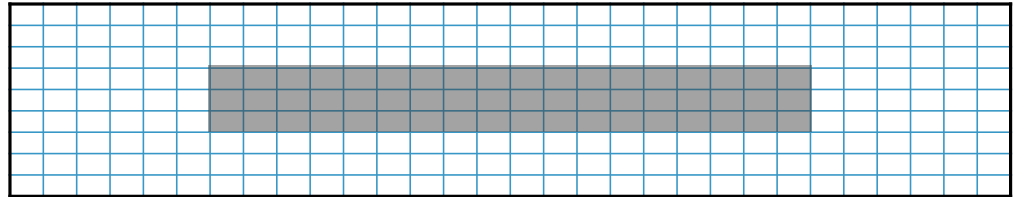
Original iPhone



@2x



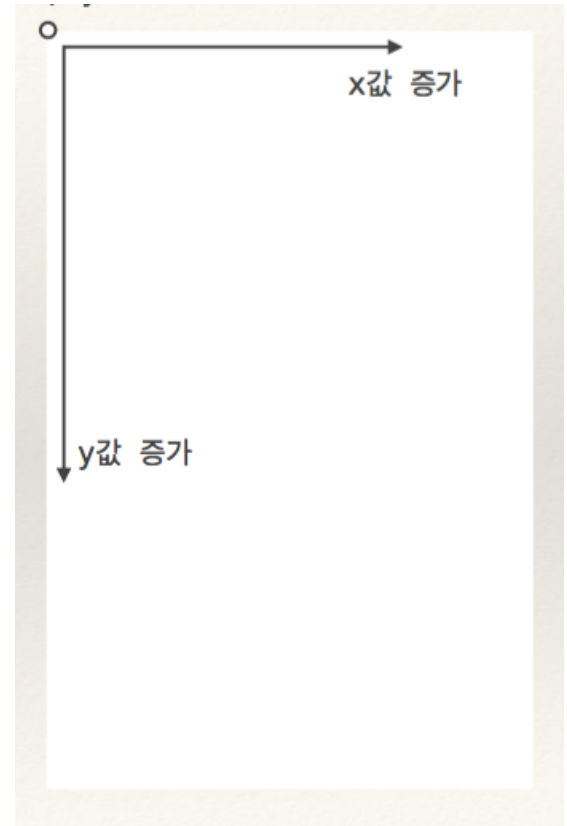
@3x



# 좌표계 (Frame-Base)

---

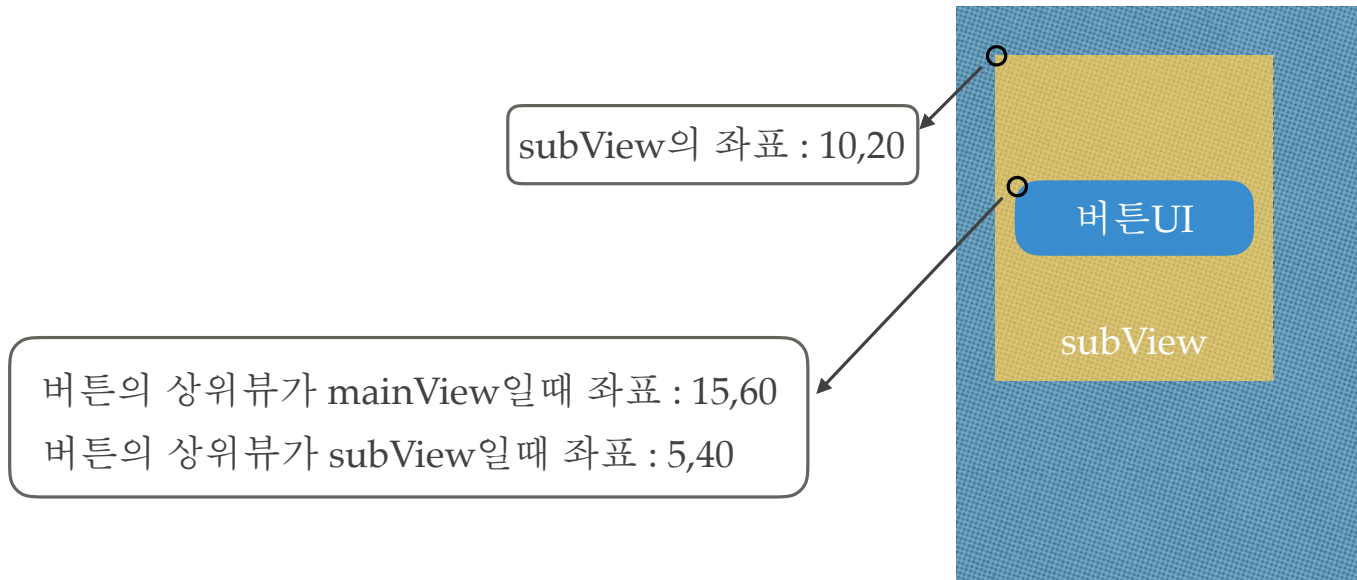
- 좌표계 : View기준 좌측 상단이 0,0이다.



# 좌표계 (Frame-Base)

---

- 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 통합 함으로써 개발자가 새로운 애플리케이션을 위한 표준 코드를 다시 작성하지 않아도 사용된다.



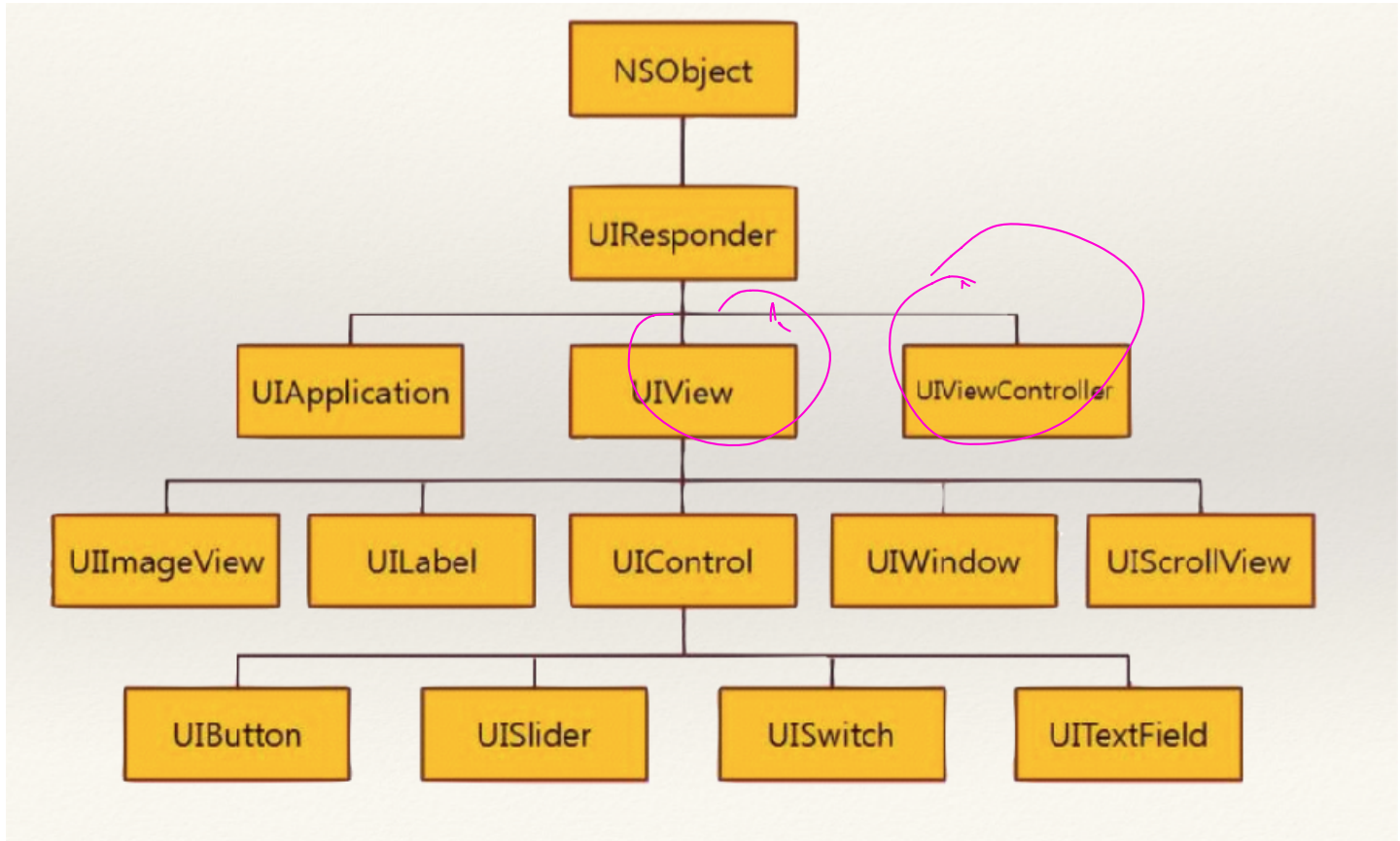
# UIKit Framework

---

- 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`

*focusing on 관건점.*

`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?)`

*UI Gesture 관련 별도.*

# 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.

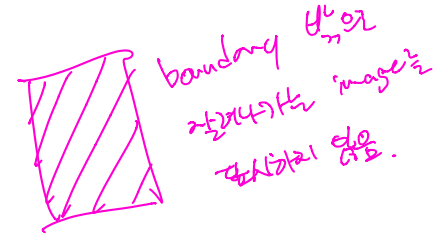
# UIView Class

- 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
```

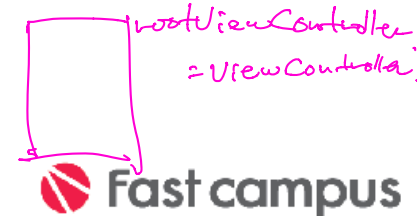
```
open func addSubview(_ view: UIView)
open func insertSubview(_ view: UIView, at index: Int)
open func removeFromSuperview()
open func layoutIfNeeded()
```

frame vs bounds  
x y width height



동작 K 숨김. 차이?  
타이거 도면 상조고.

rootView Controller. 화면의 상위과 동일.



# 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)
```

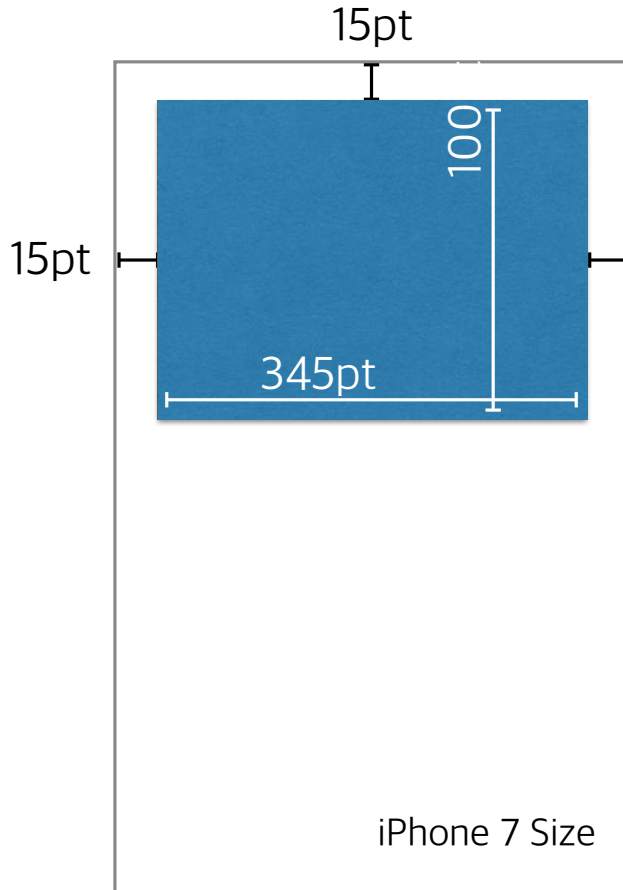
# View를 만들어 봅시다.

---

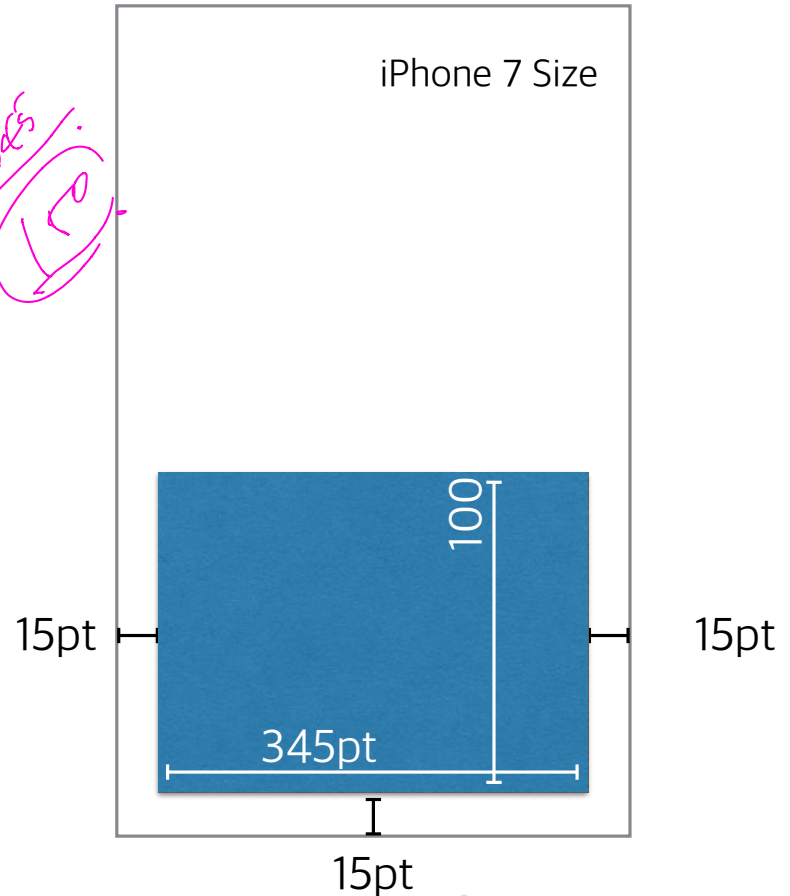
- 코드를 이용해서 View만들기
- View에 addSubview 하기
  - \* View 만들기 tip! (인스턴스 생성, 프레임 설정, addSubview)



# View 만들기 연습

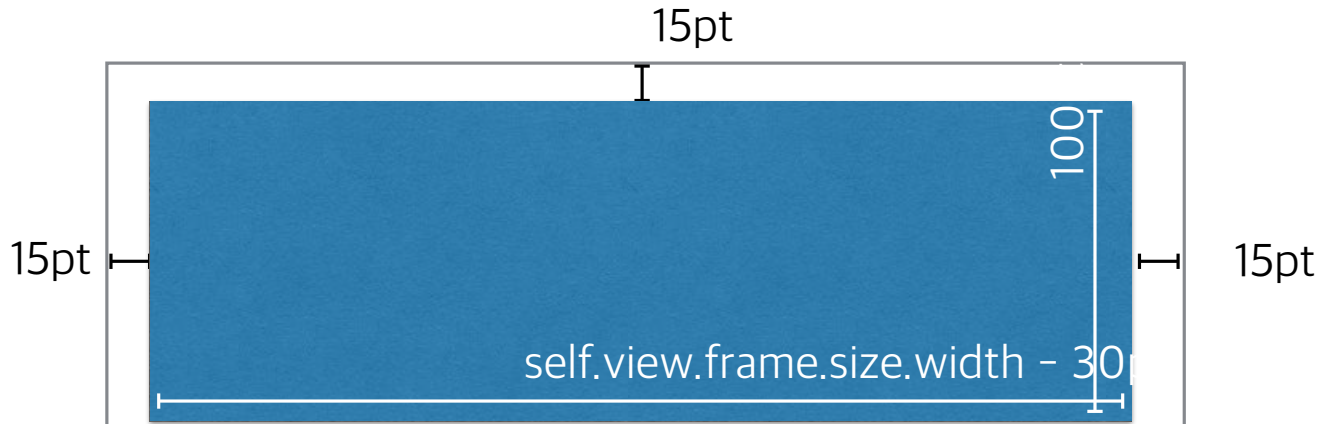


2x5.  
30



# View 만들기 연습

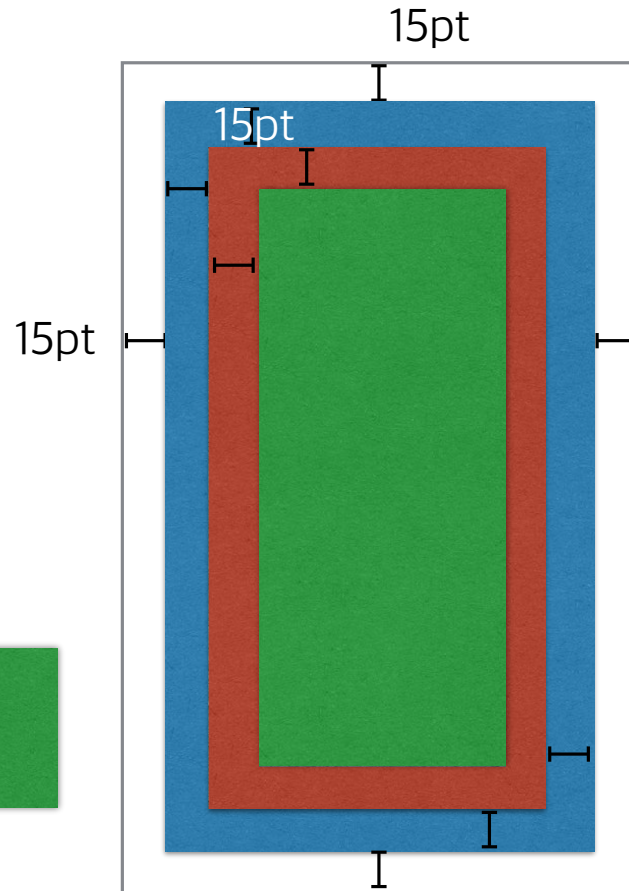
---



- 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
```

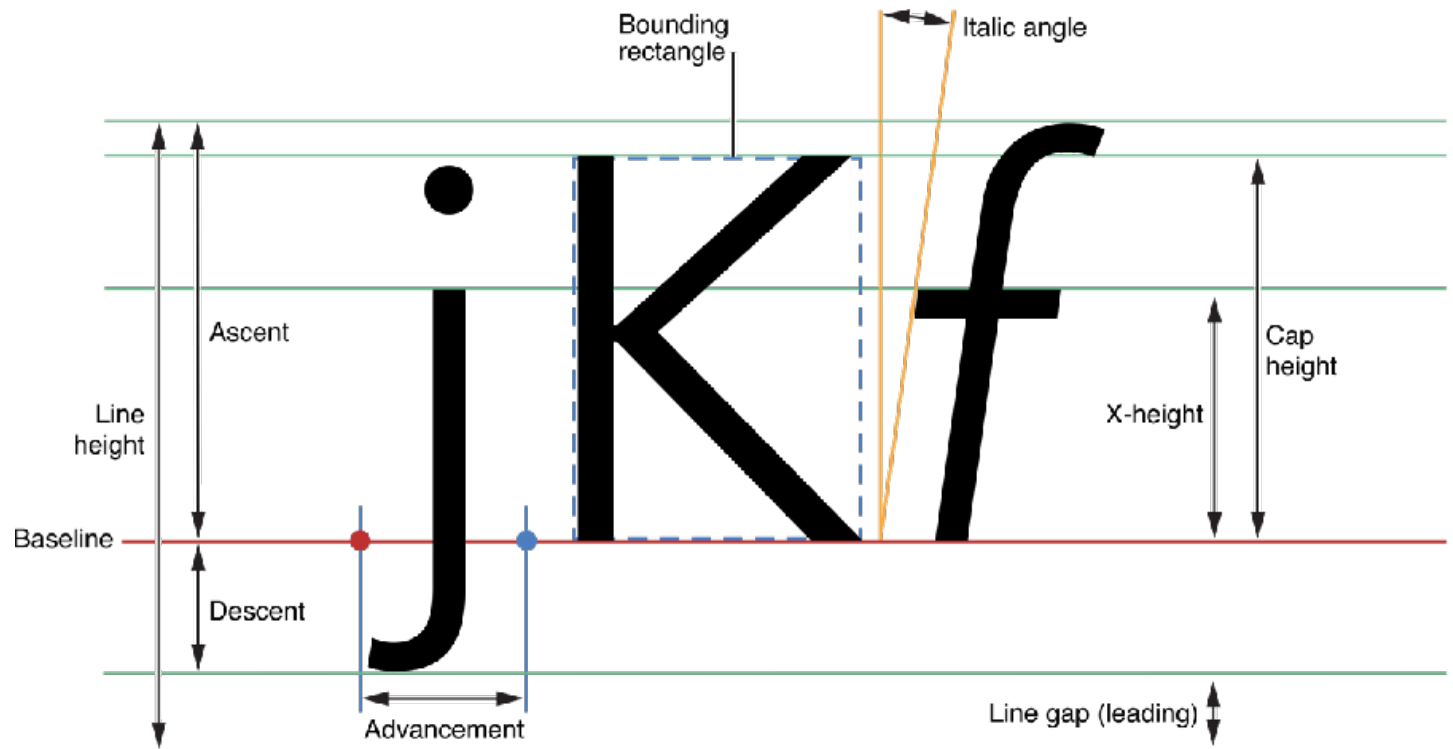
```
newLb.textAlignment = NSTextAlignment.center
```

```
newLb.font = UIFont.systemFont(ofSize: 10)
```

```
newView.addSubview(newLb)
```

*text, color, alignment, font*

# UILabel Font



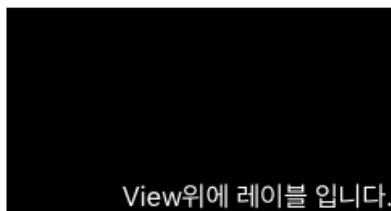


# UI실습 다음 화면을 만들어 봅시다.

---

예제 화면 입니다.

예쁜 레이블 입니다.



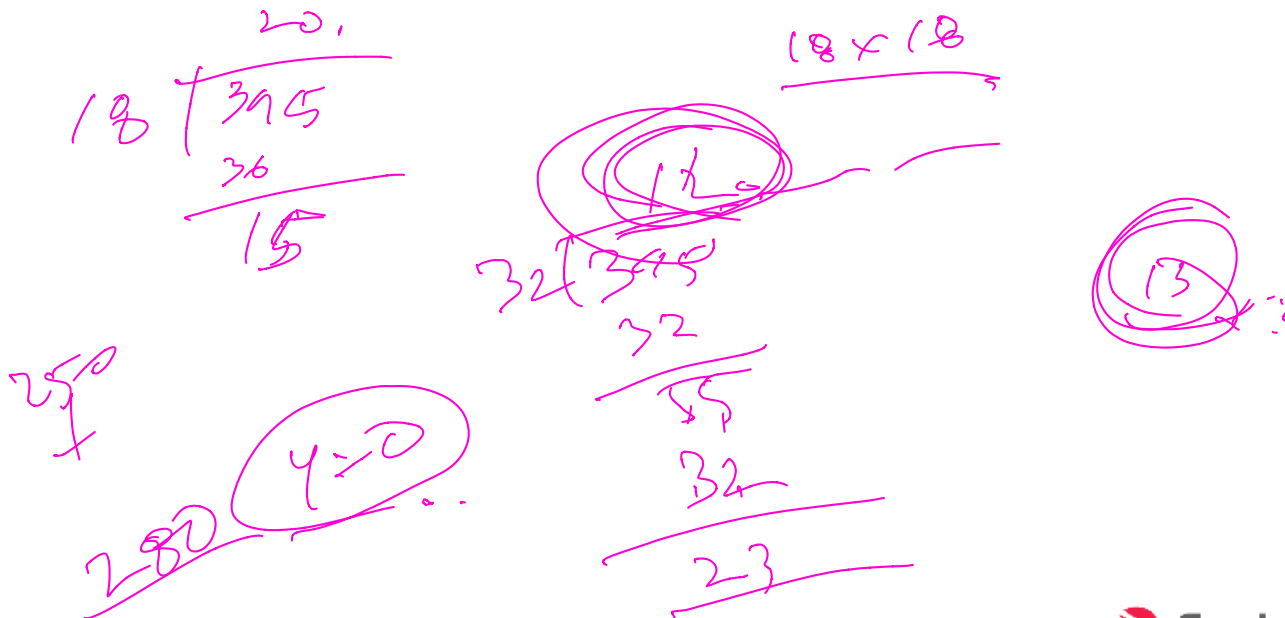
View위에 레이블 입니다.

중앙에 있는 레이블 입니다.

폰트는 20입니다.

# UIImageView 예제

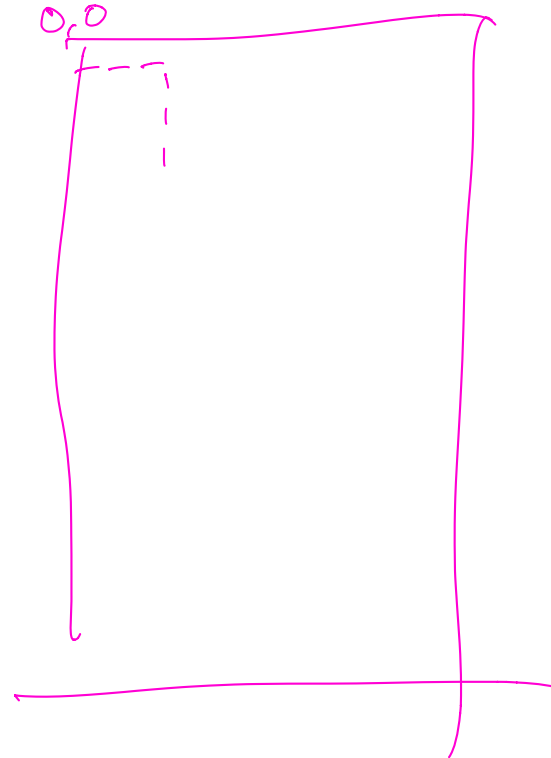
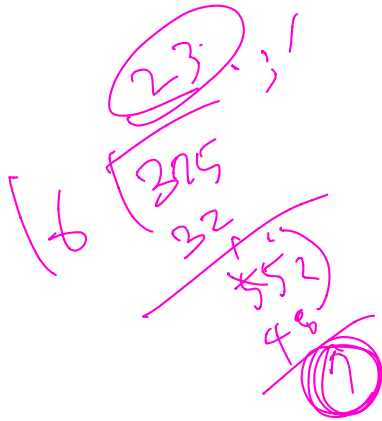
```
let newImgV = UIImageView(frame: CGRect(x: 5, y: 5, width: 90, height: 90))
newImgV.image = UIImage(named: "img.png")
newImgV.contentMode = UIViewContentMode.scaleAspectFit
newView.addSubview(newImgV)
```



# Add Image File

---

- 단일 이미지 리소스 추가
- Assets.xcassets 에 추가



---

# Assets

---

강사 주영민

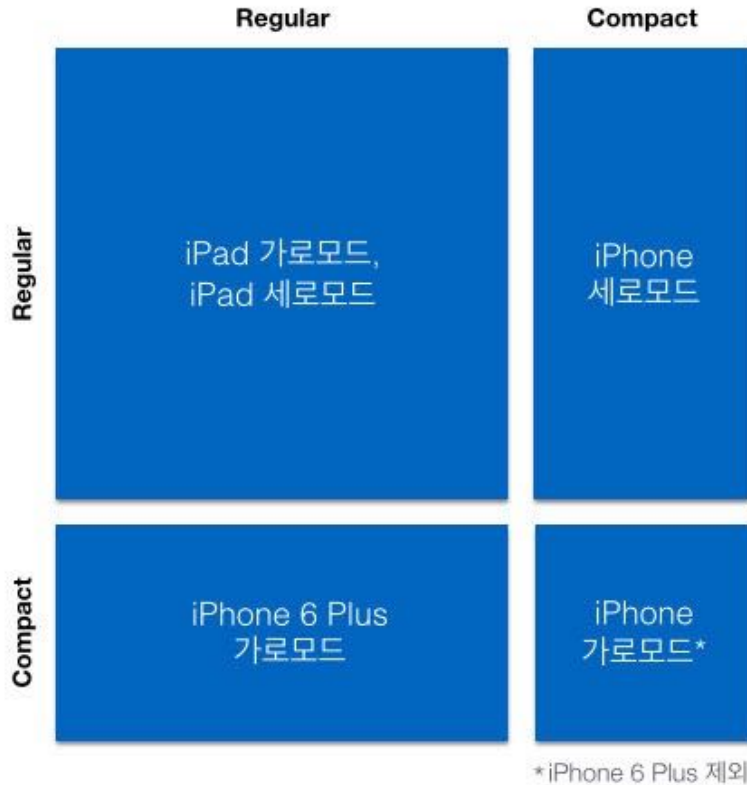
# Asset Catalogs

---

- 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

---



# Trait Collection

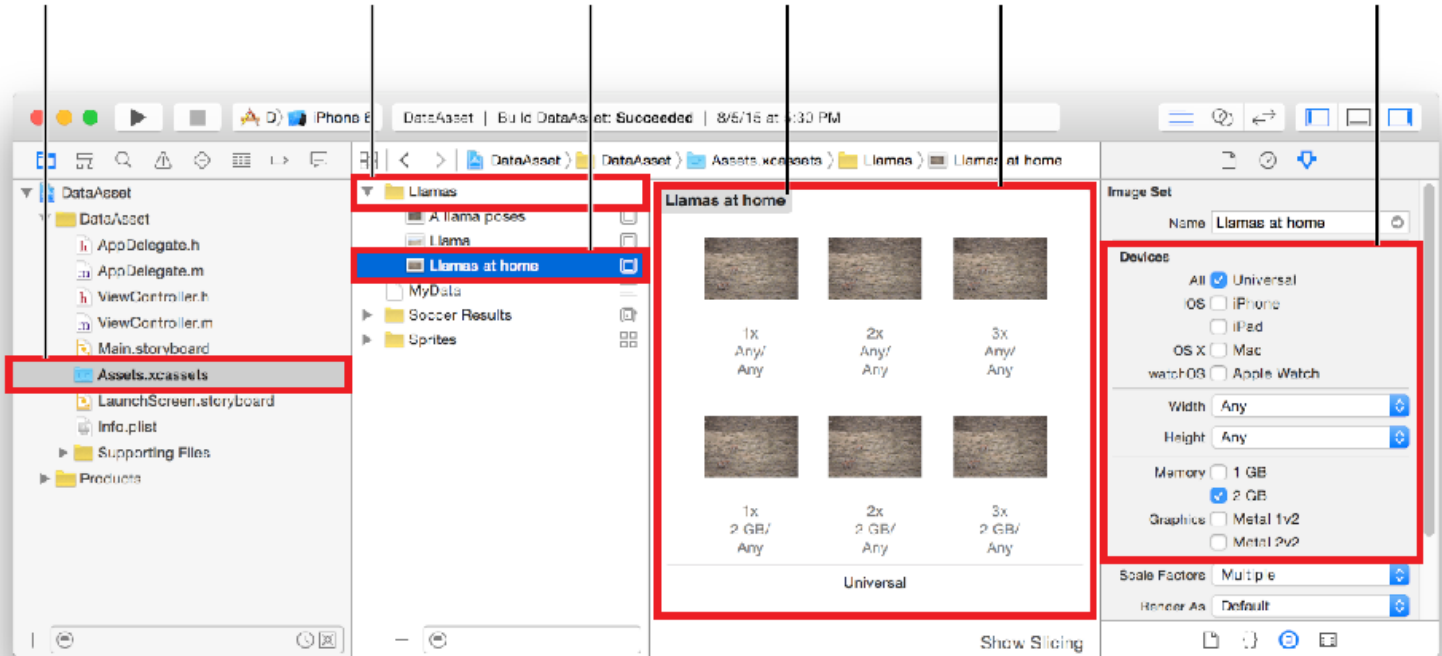
---

horizontalSizeClass	Compact
verticalSizeClass	Regular
userfaceIdiom	Phone
displayScale	2.0

\*UIViewController의 traitCollection프로퍼티로 정보를 구할수 있다.

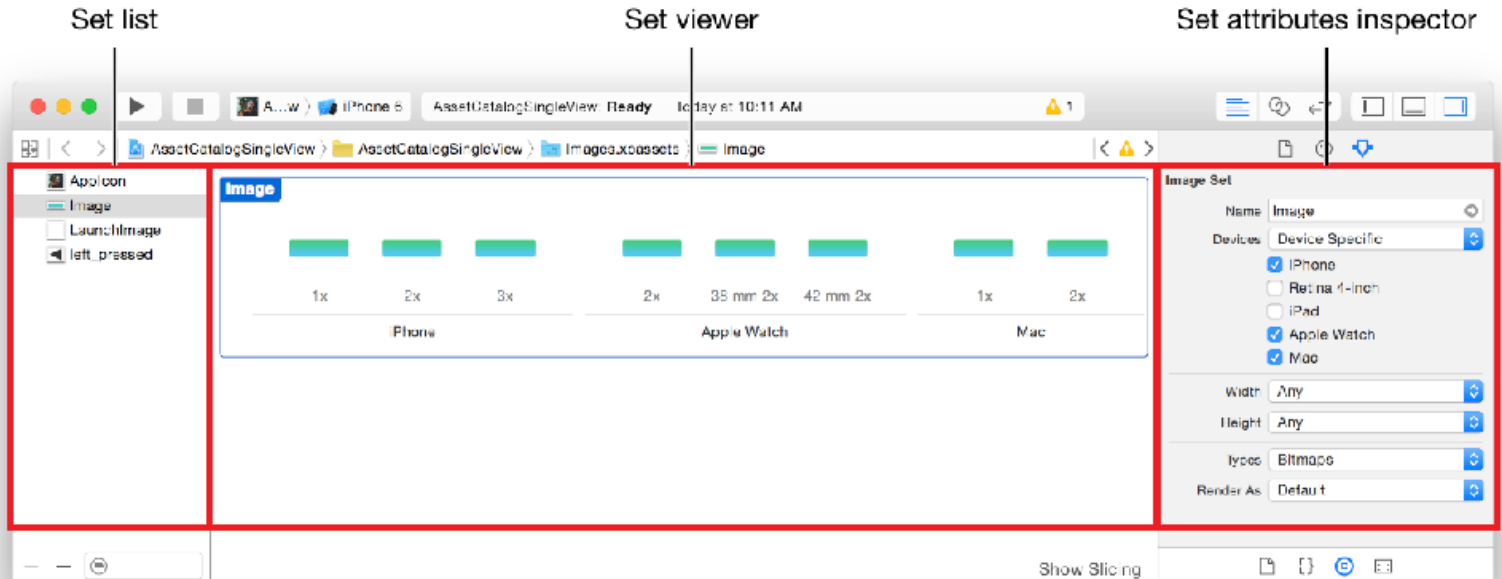
# Asset Catalogs Contents

Asset catalog      Group      Asset      Asset name      Asset files      Attributes





# Asset Catalogs Contents



# Asset Catalogs can inClude

---

- **Data sets** : Used for any type of data 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.**

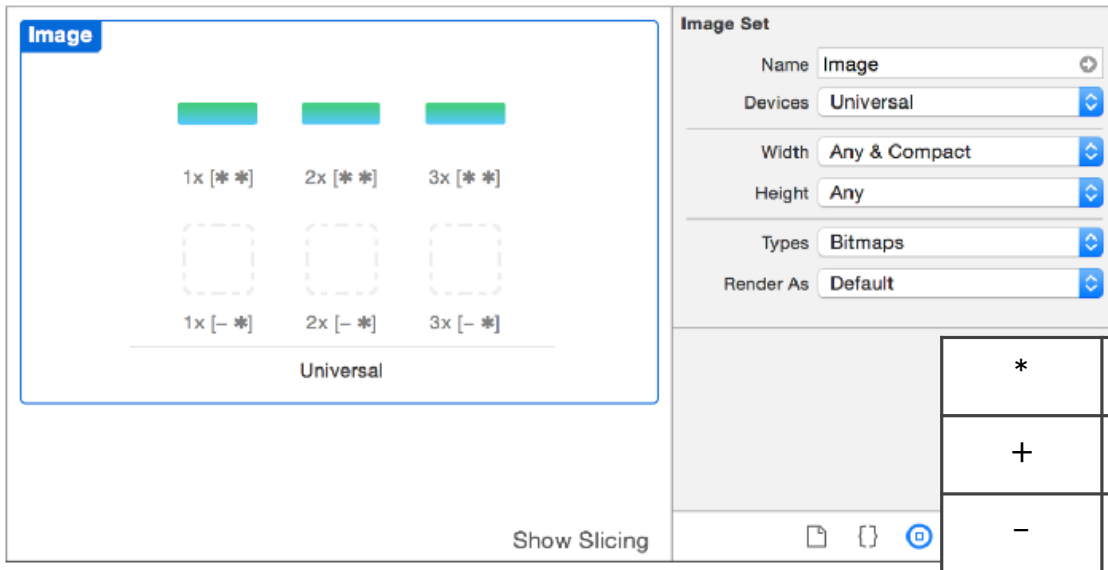
# Asset Catalogs 사용법

---

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

# Asset Catalogs 사용법

- Custom SizeClasses



The screenshot shows the Xcode Asset Catalog interface. On the left, a preview window displays a blue bar with three segments, each labeled with a resolution and size class: 1x [\* \*], 2x [\* \*], and 3x [\* \*]. Below these are three dashed boxes representing the same segments at different resolutions: 1x [- \*], 2x [- \*], and 3x [- \*]. The word "Universal" is centered below the dashed boxes. On the right, the "Image Set" panel shows settings for the "Image" set: Name (Image), Devices (Universal), Width (Any & Compact), Height (Any), Types (Bitmaps), and Render As (Default). At the bottom right, a table explains the symbols used in the size class labels.

*	Any size class
+	Regular size
-	Compact size

해상도 [가로Size, 세로Size]

# ContentsMode

**UIViewContentModeScaleAspectFill**



Nondistorting



**UIViewContentModeScaleAspectFit**



Nondistorting



**UIViewContentModeScaleToFill**



Distorting



# UI실습 : UIImageView만들기

---

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

# UIControl

---

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

# UIControl - 주요 항목

---

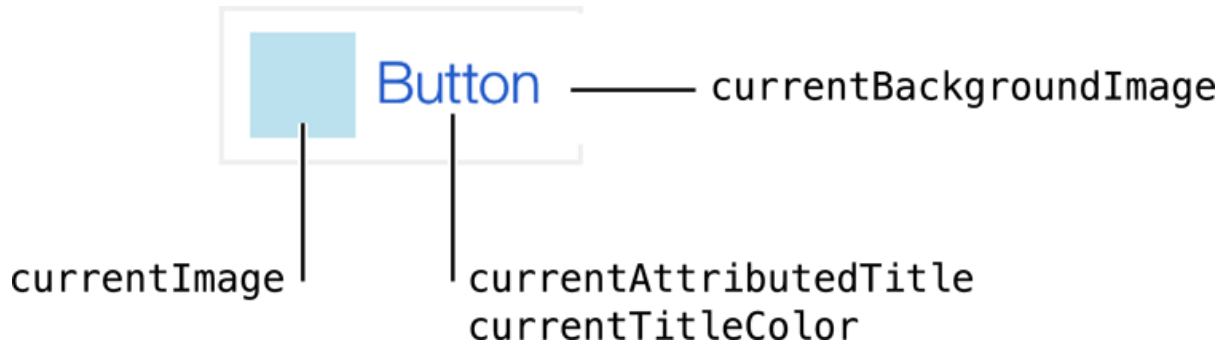
```
public static var normal: UIControlState { get }  
public static var highlighted: UIControlState { get }  
public static var disabled: UIControlState { get }  
public static var selected: UIControlState { get }  
  
open var isEnabled: Bool  
  
open var isSelected: Bool  
  
open func addTarget(_ target: Any?,  
                    action: Selector,  
                    forControlEvents: 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