

Autolayout Anchors

NSLayoutConstraint

```
// Creating constraints using NSLayoutConstraint
NSLayoutConstraint(item: subview,
                  attribute: .leading,
                  relatedBy: .equal,
                  toItem: view,
                  attribute: .leadingMargin,
                  multiplier: 1.0,
                  constant: 0.0).isActive = true

NSLayoutConstraint(item: subview,
                  attribute: .trailing,
                  relatedBy: .equal,
                  toItem: view,
                  attribute: .trailingMargin,
                  multiplier: 1.0,
                  constant: 0.0).isActive = true
```

NSLayoutAnchor

iOS 9.0 이상에서 사용 가능

```
// Creating the same constraints using Layout Anchors
let margins = view.layoutMarginsGuide

subview.leadingAnchor.constraint(equalTo: margins.leadingAnchor).isActive = true
subview.trailingAnchor.constraint(equalTo: margins.trailingAnchor).isActive = true
```

Horizontal Layout Anchors

View 의 수평선, X축에 관련된 제약조건
NSLayoutXAxisAnchor 클래스의 객체

- Leading anchor
- Trailing anchor
- Left anchor
- Right anchor
- Center-X anchor

```
view.leadingAnchor  
    .constraint(equalTo: NSLayoutAnchor<NSLayoutXAxisAnchor>)
```

Vertical Layout Anchors

View 의 수직선, Y축에 관련된 제약조건
NSLayoutYAxisAnchor 클래스의 객체

- Top anchor
- Bottom anchor
- Center-Y anchor
- First baseline anchor
- Last baseline anchor

```
view.topAnchor  
    .constraint(equalTo: NSLayoutAnchor<NSLayoutYAxisAnchor>)
```

Dimension Layout Anchors

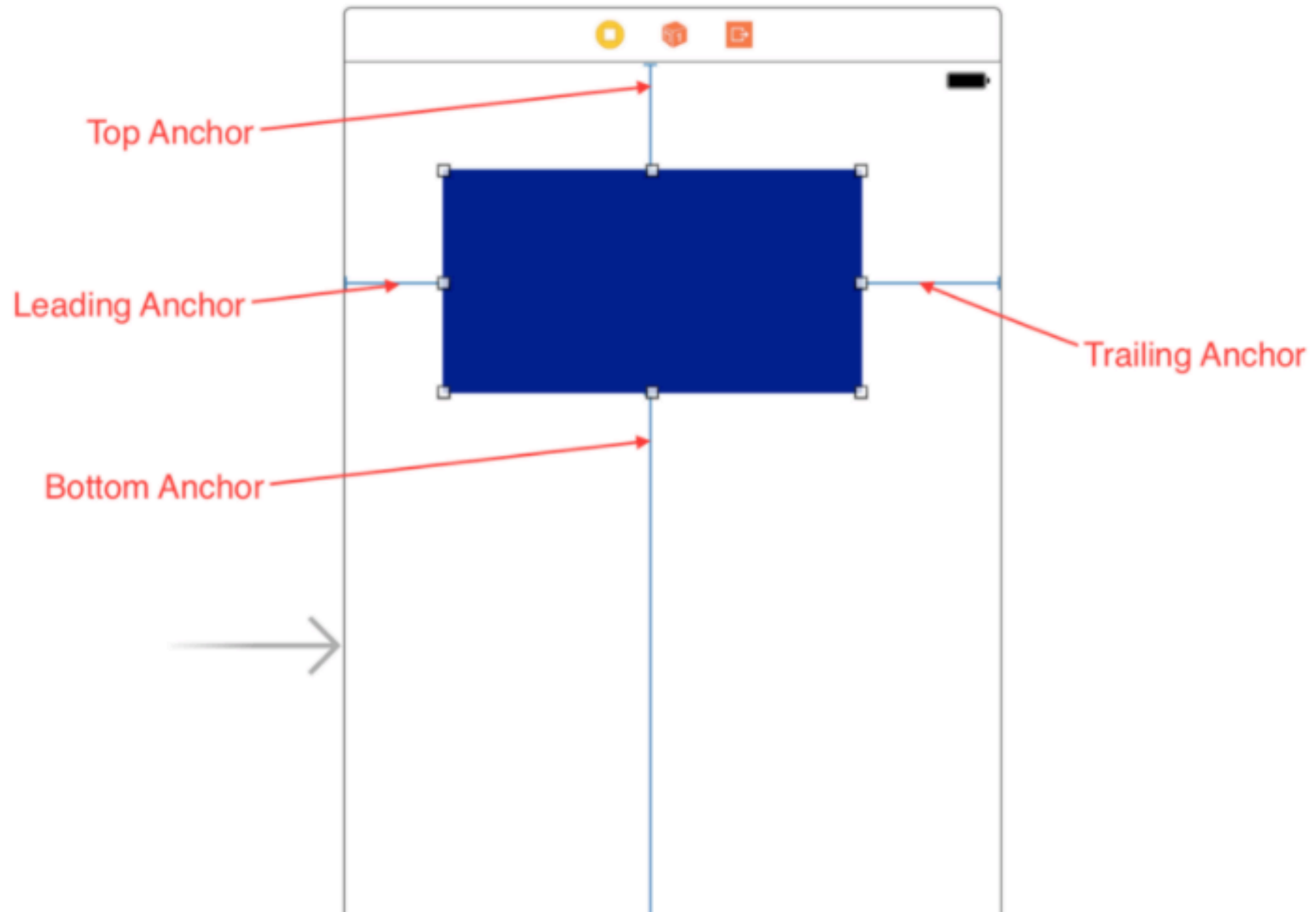
View 의 크기 정의

NSLayoutConstraint 클래스의 객체

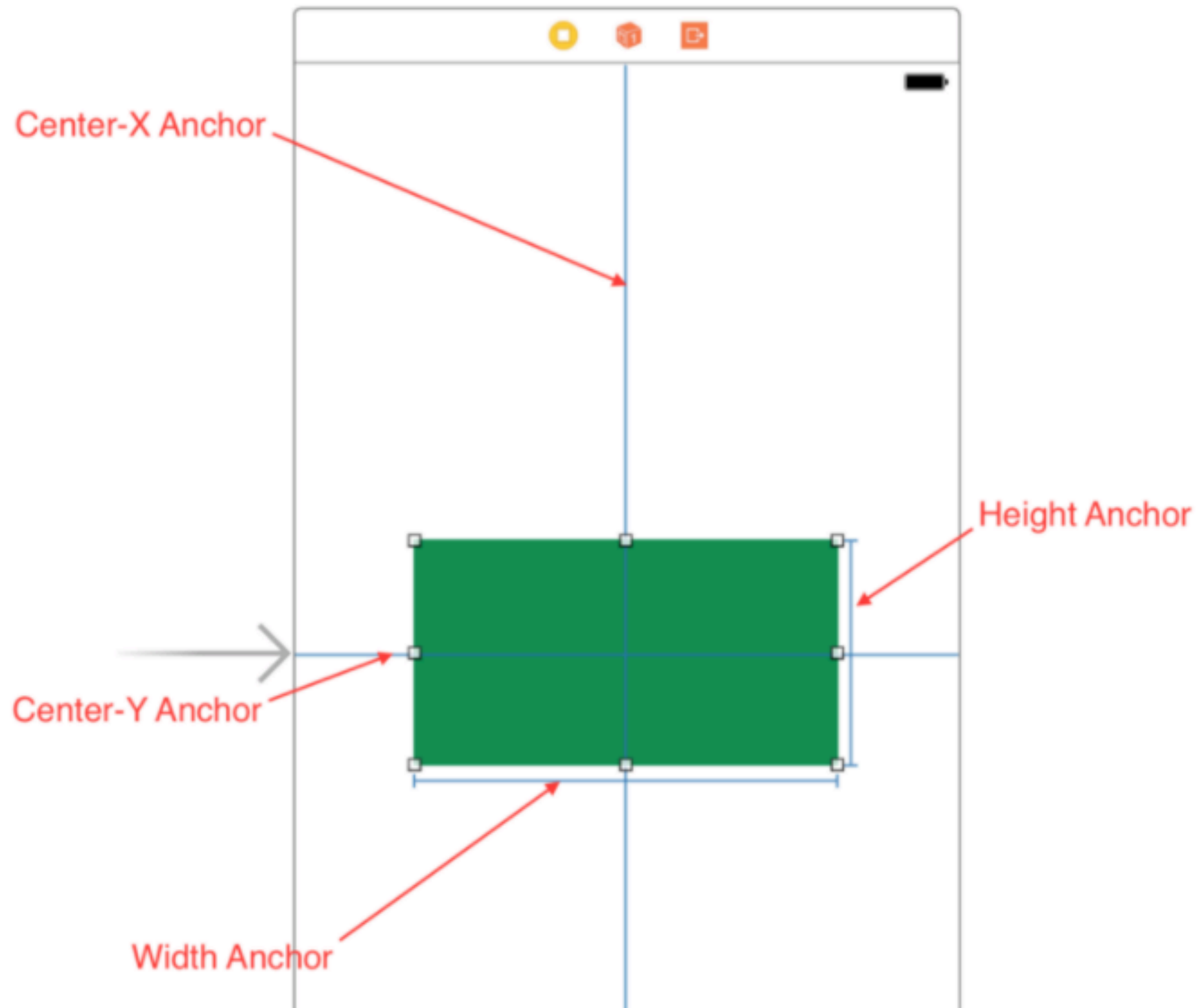
- Width anchor
- Height anchor

```
view.widthAnchor  
    .constraint(equalToConstant: CGFloat)
```

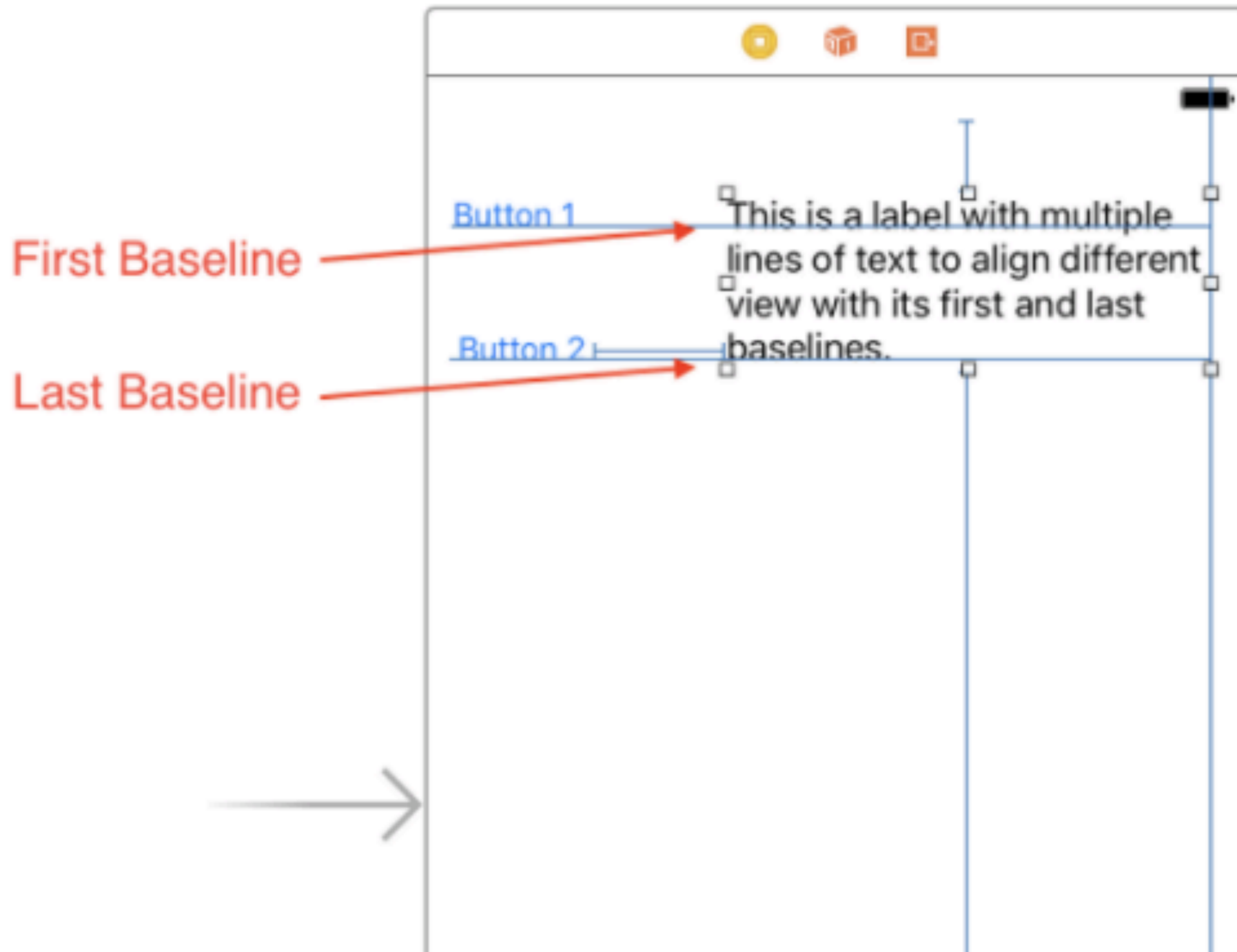
Basic Layout Constraints



Dimension & Center X,Y Layout



First, Last Baseline Constraints



Practice

스토리보드에서 만들었던 이 뷰를 오토레이아웃 앵커를 이용해 동일하게 만들기

- 각 뷰는 가장자리로부터 각각 20 씩 떨어져있고 뷰 사이 거리는 10 이며 두 뷰의 길이는 동일
- iPhone 8과 iPhone X 이상의 모델에서 실행해보고 비교

