
AutoLayout

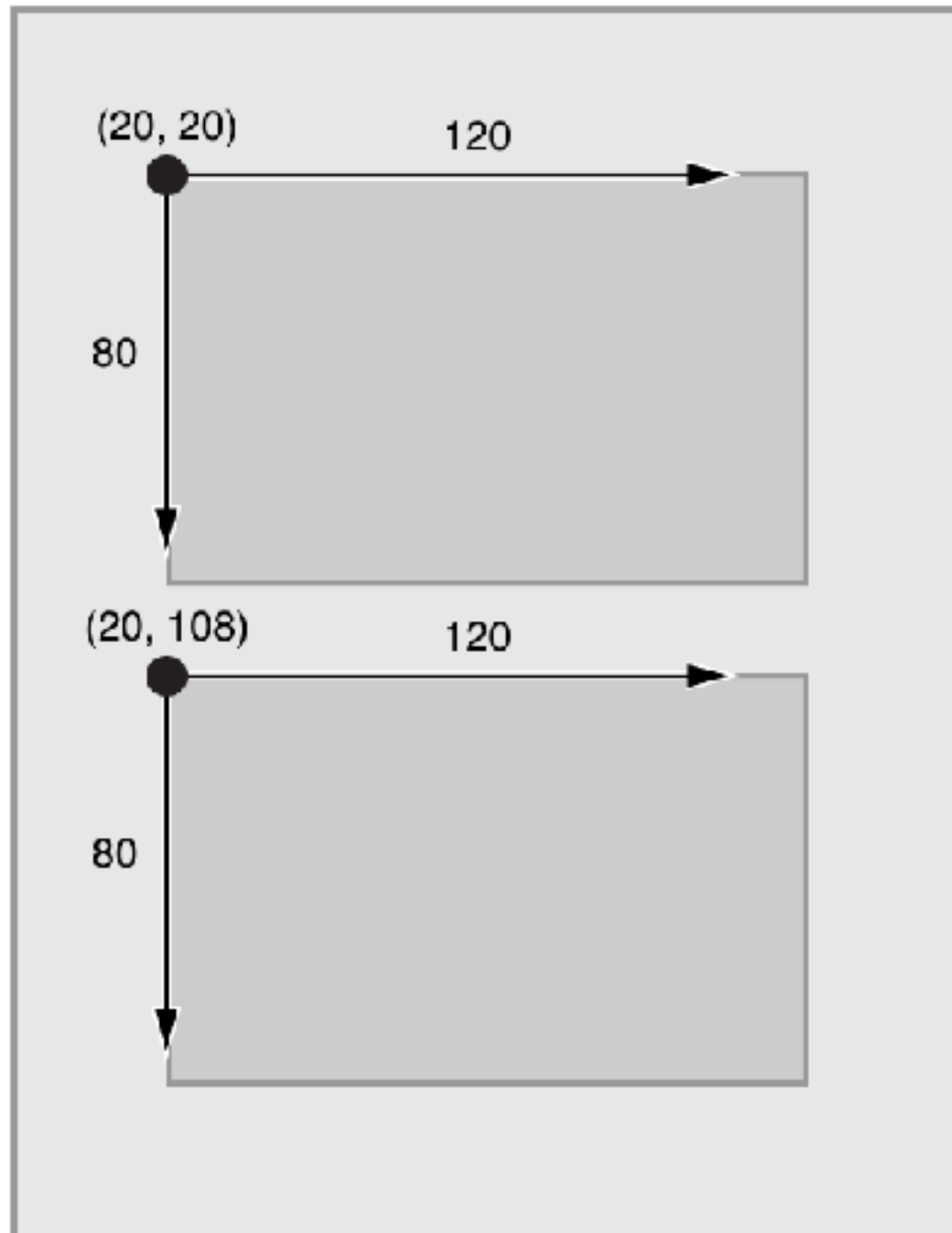
강사 주영민

AutoLayout

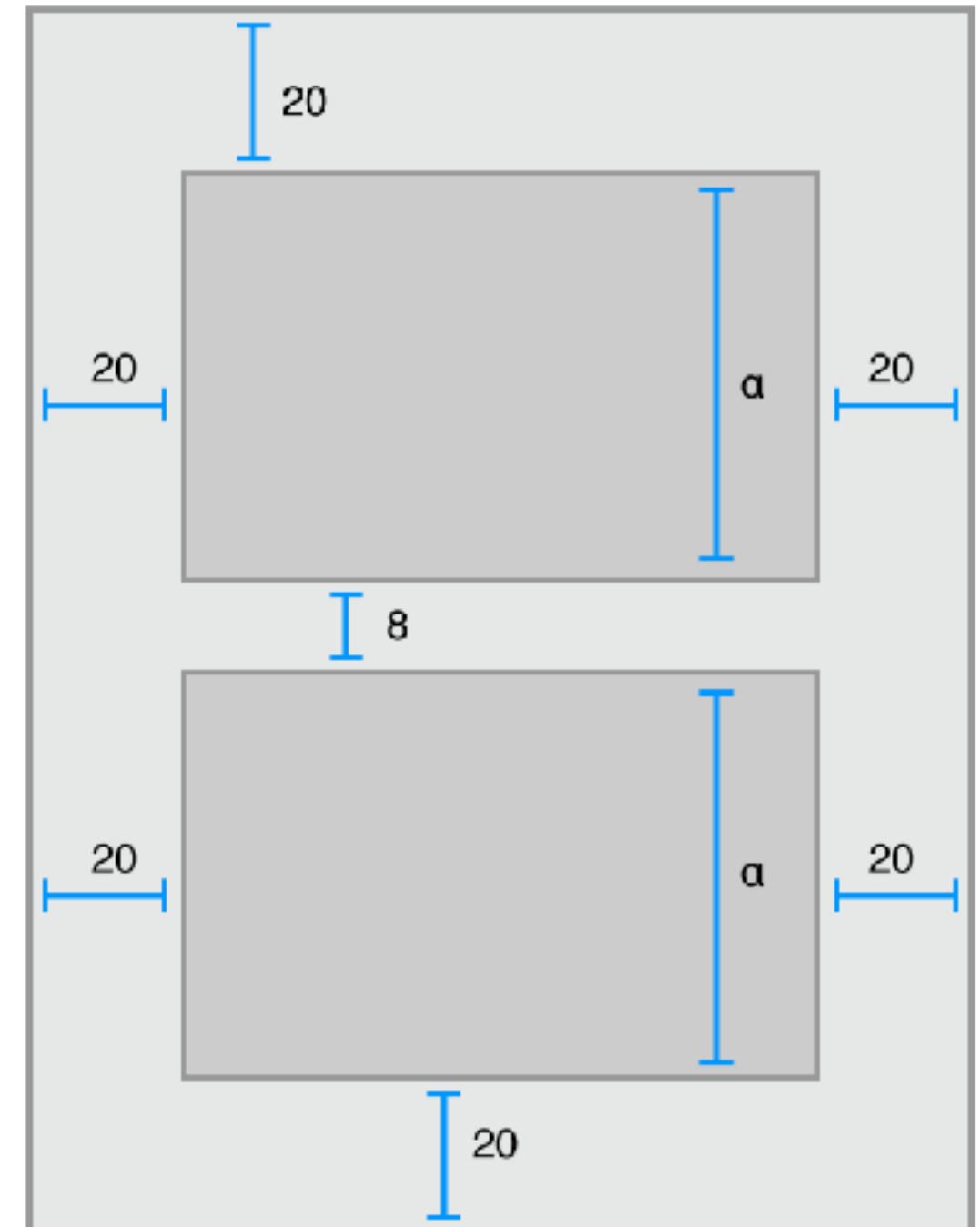
- AutoLayout은 각각의 View의 Size와 위치를 제약사항을 통해서 유동적으로 계산하여 표현하는 방법

Auto Layout VS Frame-Based Layout

Frame-Based Layout



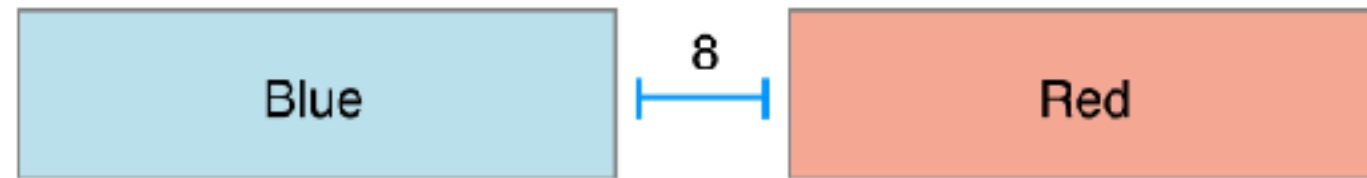
Auto Layout



Constraint(제약)

- 각 View의 거리, 길이, 위치 등을 표현하기 위한 제약

Constraint



$$\underbrace{\text{RedView.Leading}}_{\text{Item 1}} = \underbrace{1.0}_{\text{Multiplier}} \times \underbrace{\text{BlueView.trailing}}_{\text{Item 2}} + \underbrace{8.0}_{\text{Constant}}$$

Relationship

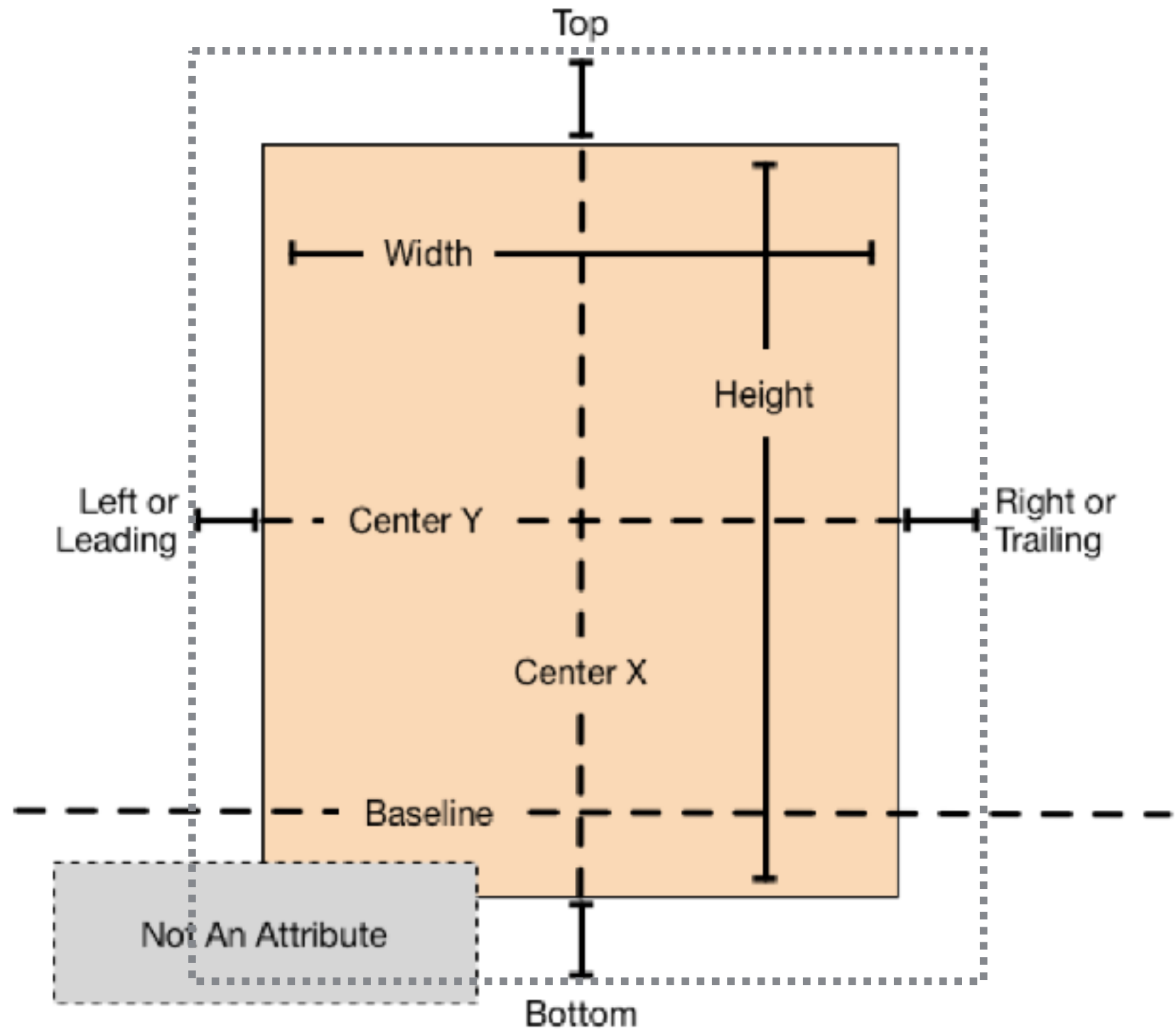
Attribute 2

Attribute 1

- Multiplier : 비율을 통한 레이아웃 설정을 위한 속성
- Constant : 일정한 간격을 유지하기 위한 속성

Attribute(속성)

- Size attributes
 - ✓ width
 - ✓ height
- Location attributes
 - ✓ Leading
 - ✓ Trailing
 - ✓ Top
 - ✓ Bottom
 - ✓ Vertical
 - ✓ Horizontal



Constraint 공식

대상 View의 Attribute는 기준View의 Attribute X 비율 +간격이다.

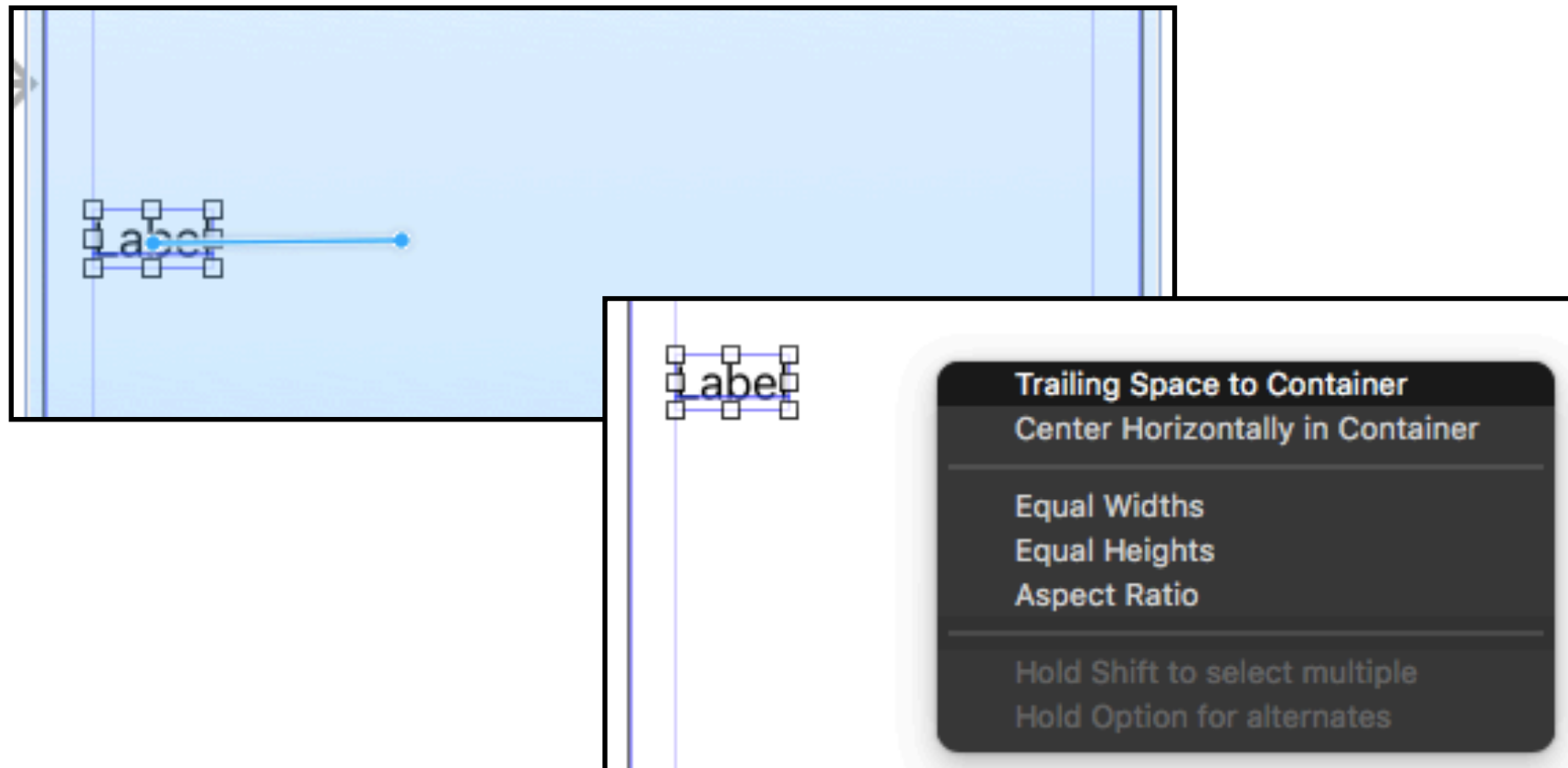
$$\text{Item1.Attribute} = \text{비율} \times \text{Item2.Attribute} + \text{간격}$$

제약사항 만들기

강사 주영민

Using Stroyboard

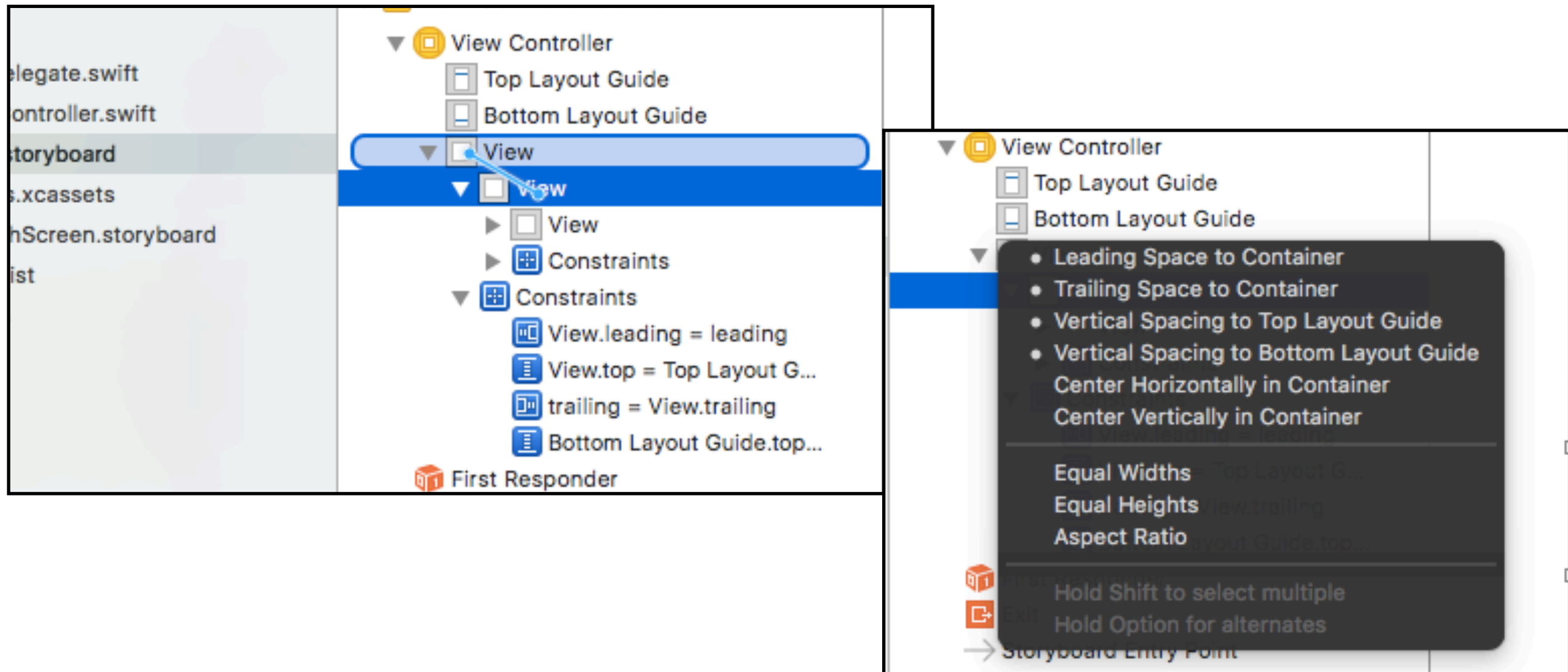
Control + Drag



✓ drag의 위치와 방향에 따라 다른 제약 메뉴가 나타난다.

Using Storyboard

Control + Drag



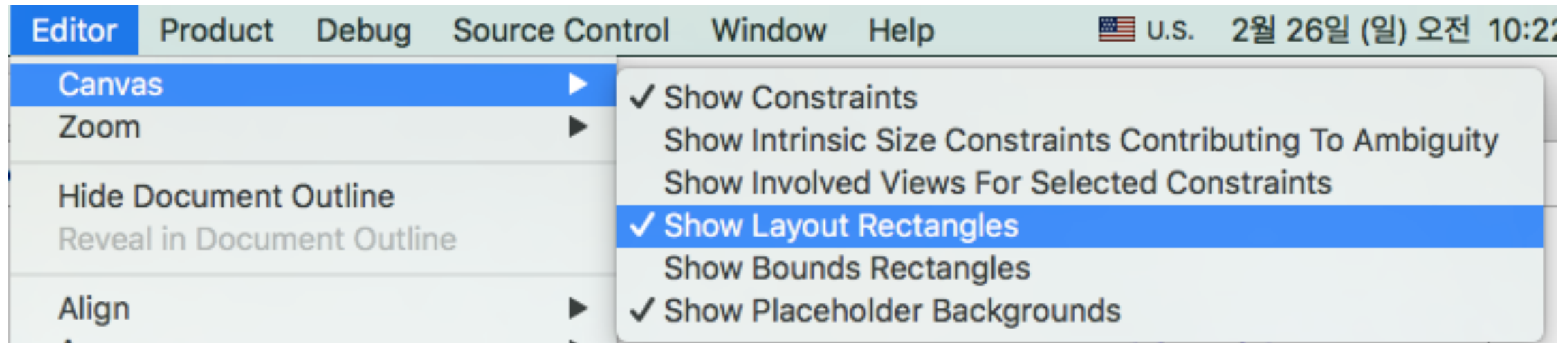
✓ drag의 위치와 방향에 따라 다른 제약 메뉴가 나타난다.

Layout margin

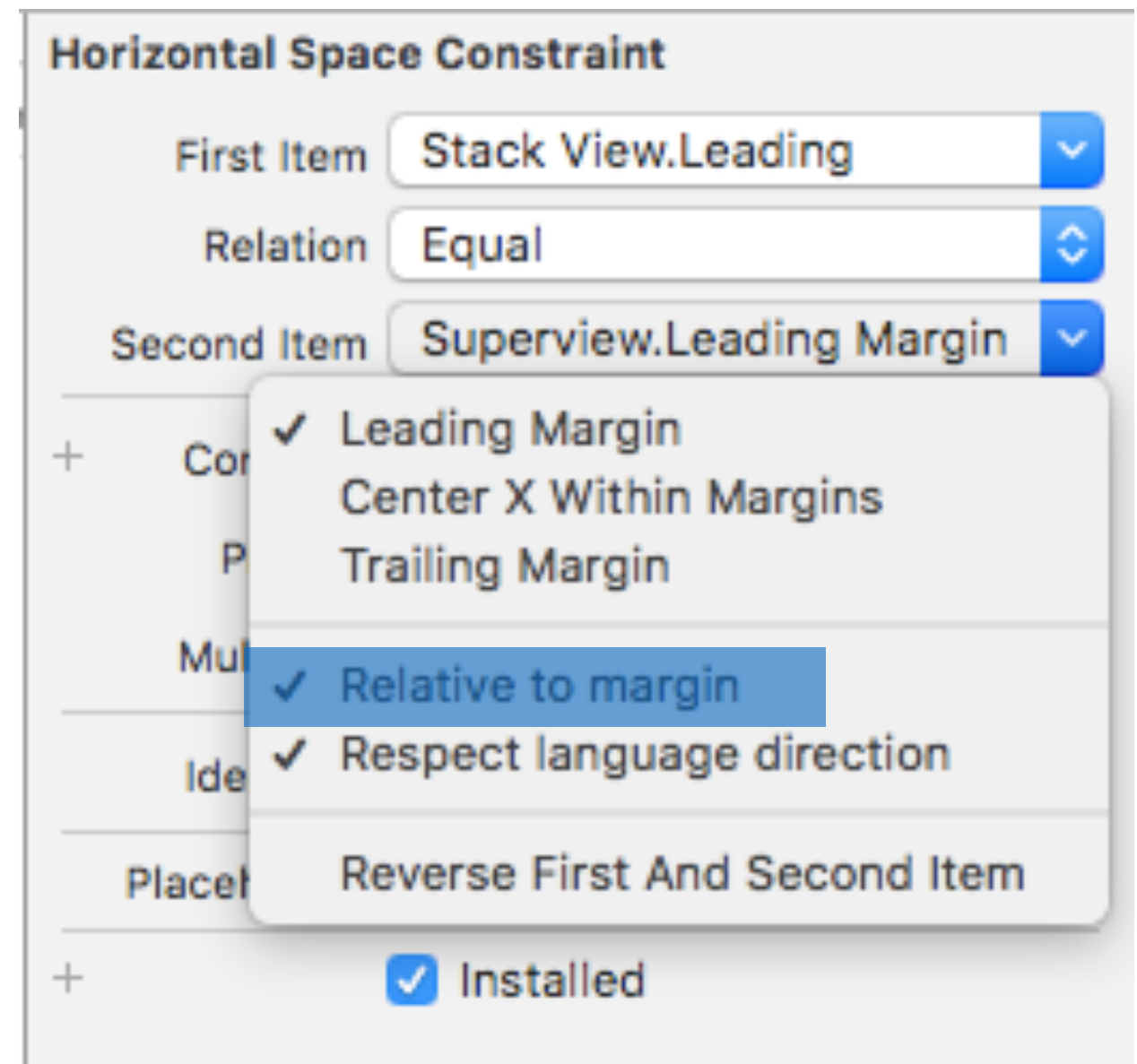
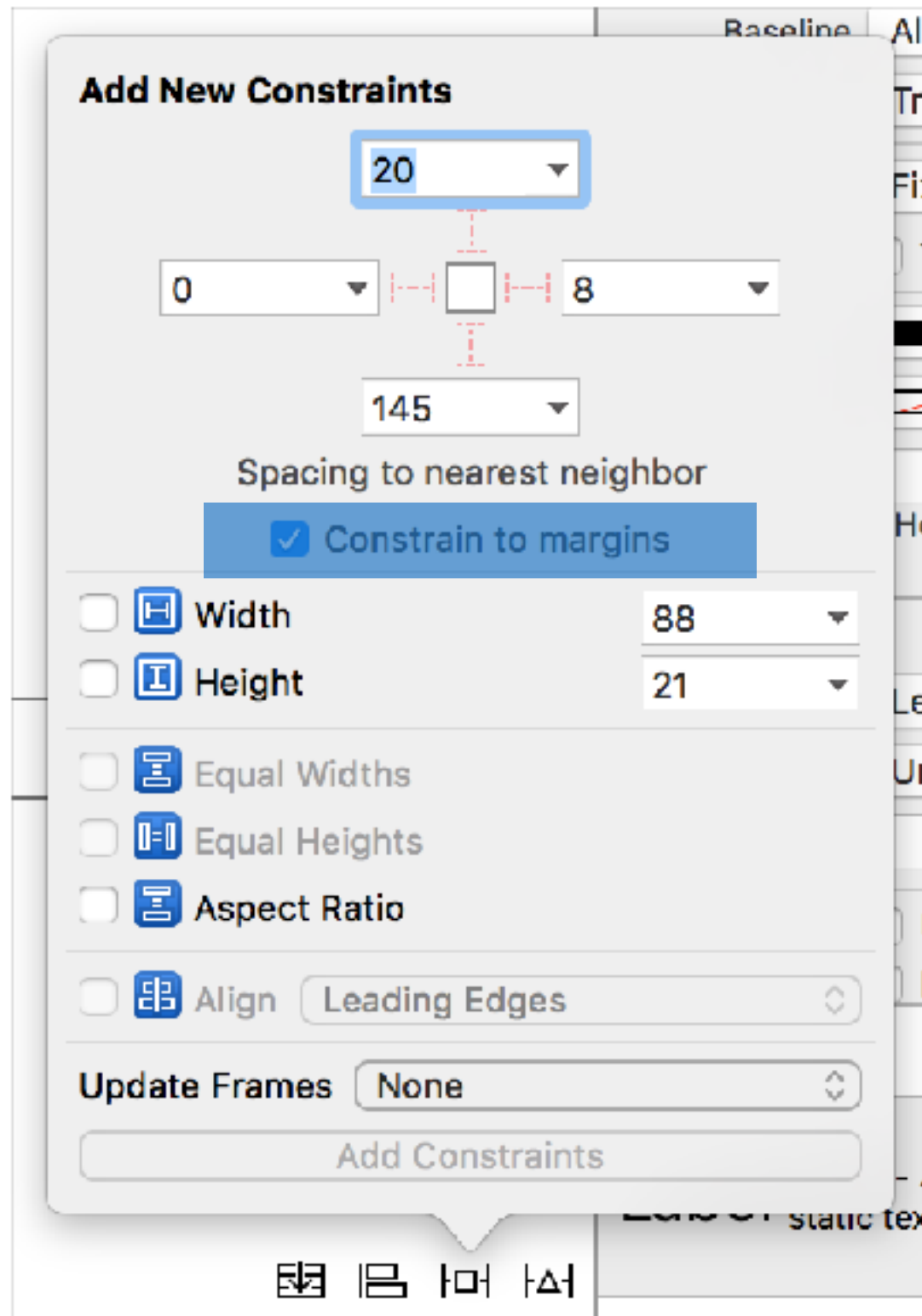
- SuperView의 가장자리와 SubView와의 간격이 유지되도록 설정된 내부 패딩값
- 기본적으로 UIView의 내부에 8Point의 여백을 가지고 있다.
- Interface Builder에서는 변경할수 없으며, UIView.layoutMargins 프로퍼티를 통해 변경 가능하다.



Show Layout Margin



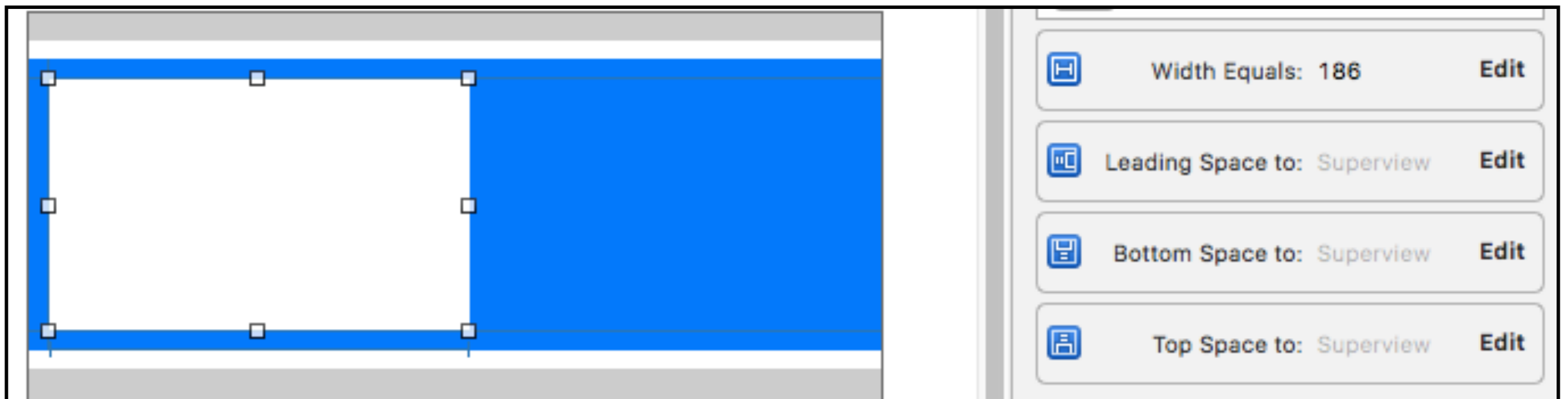
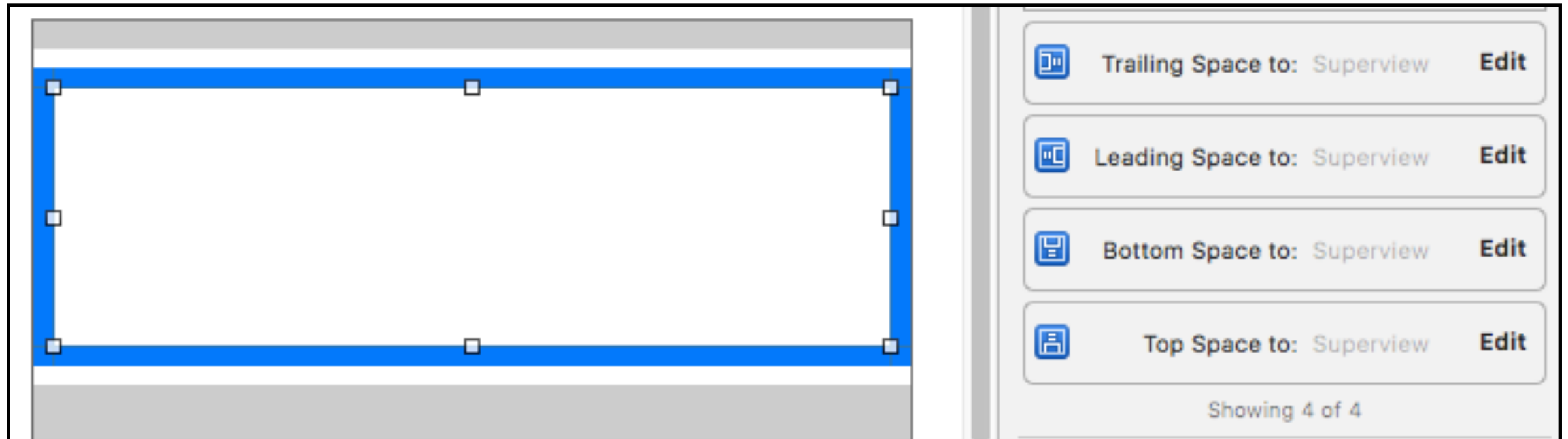
Layout Margin 제거



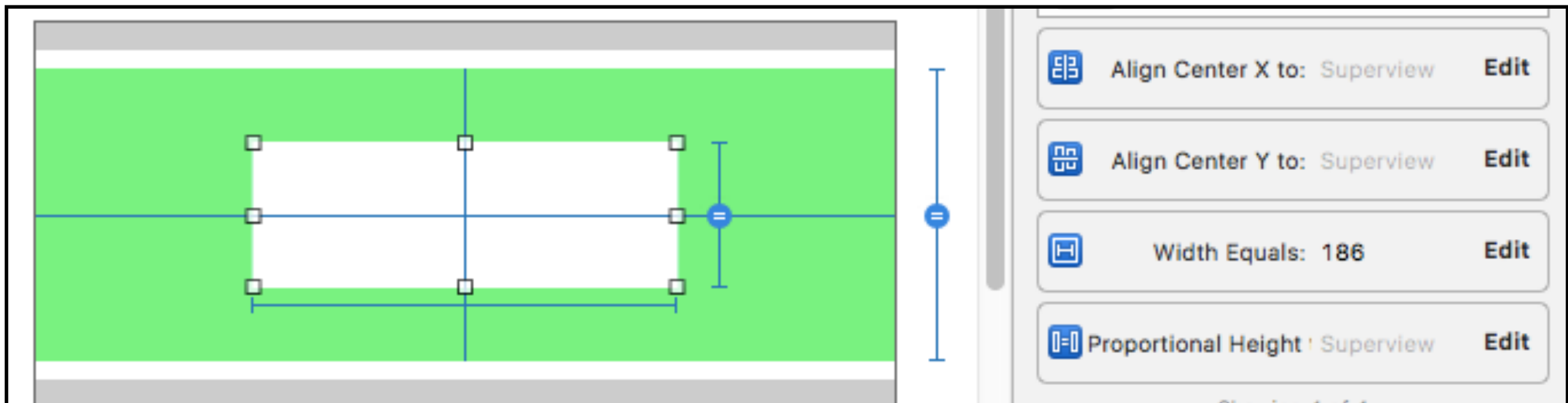
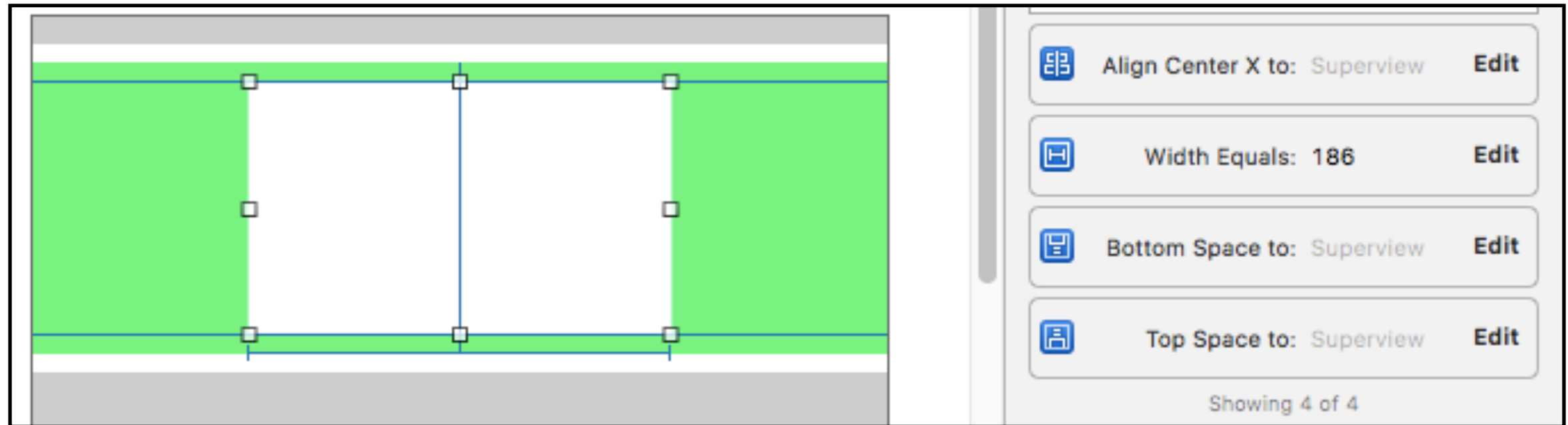
제약사항 만들기 팁

- 각 View의 주변 모든 제약사항이 만족해야된다.
- 가로제약, 세로제약을 확인
- 중복된 제약은 제거한다.
- 화면 배치의 기준 View를 정해서 연관된 제약사항을 만들시 수정에 따른 변경 고려

예제



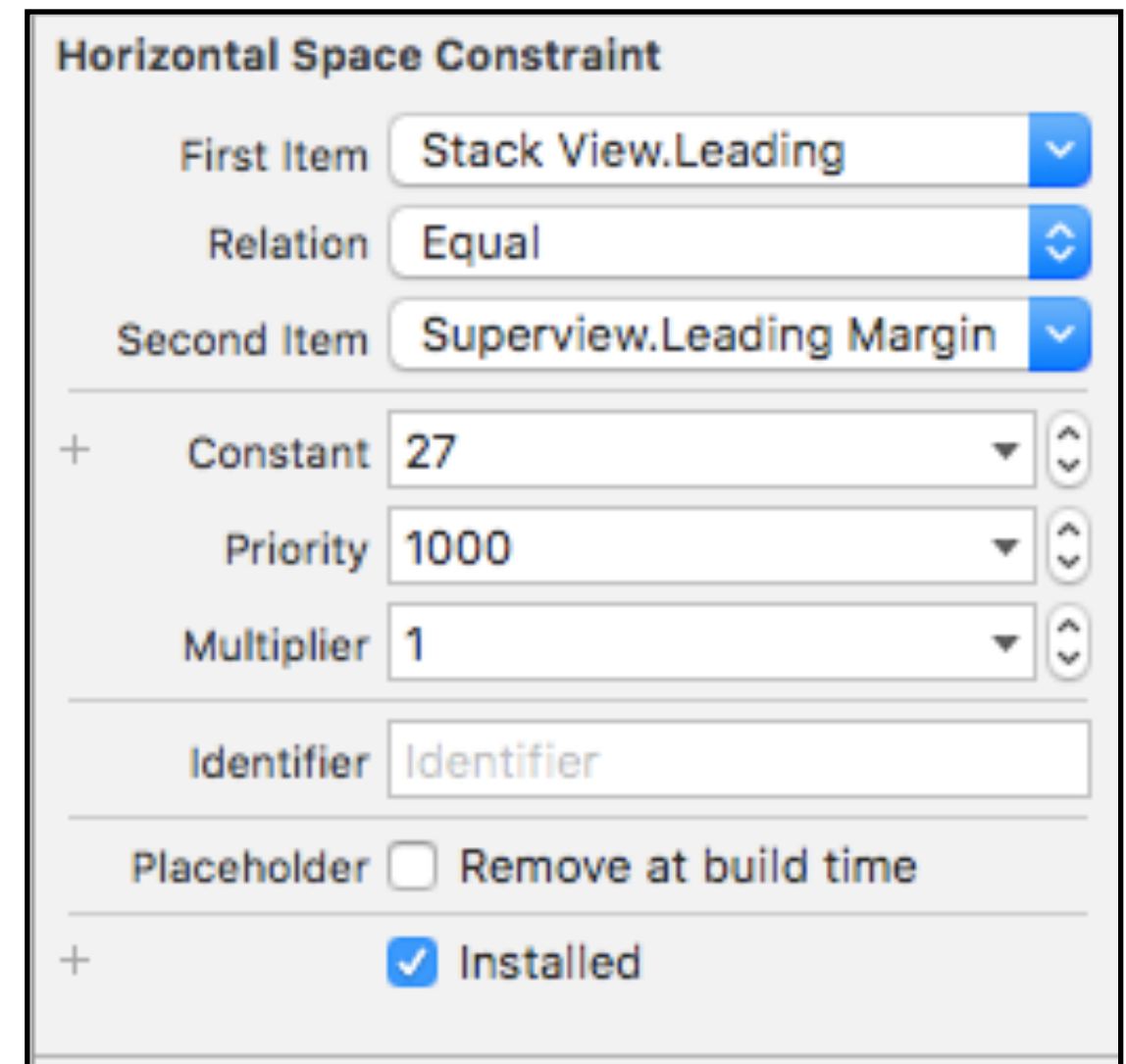
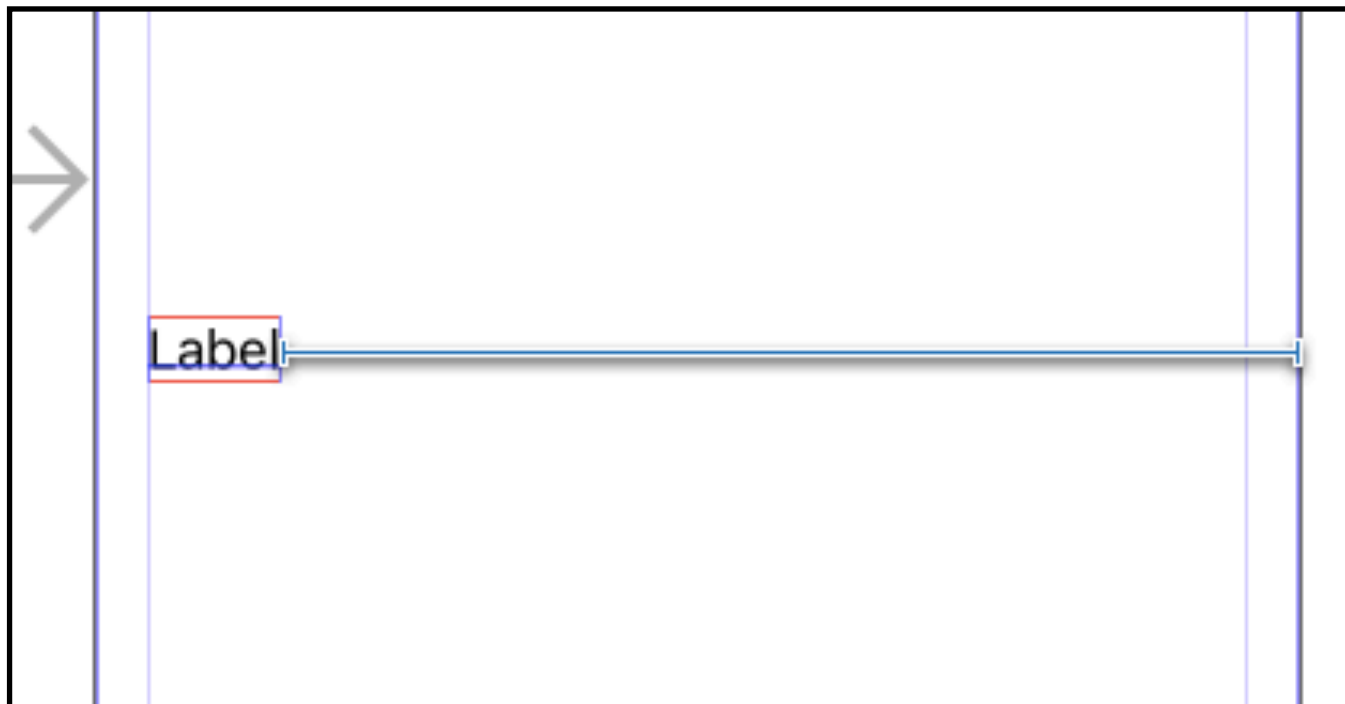
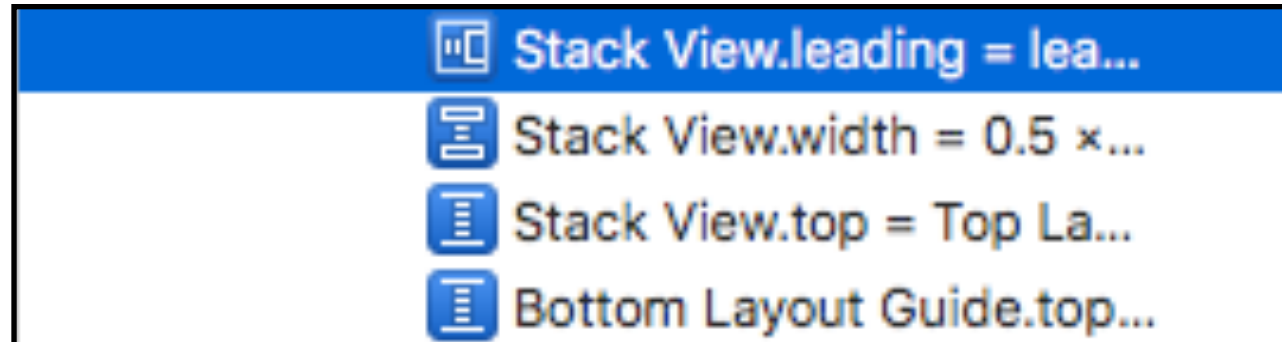
예제



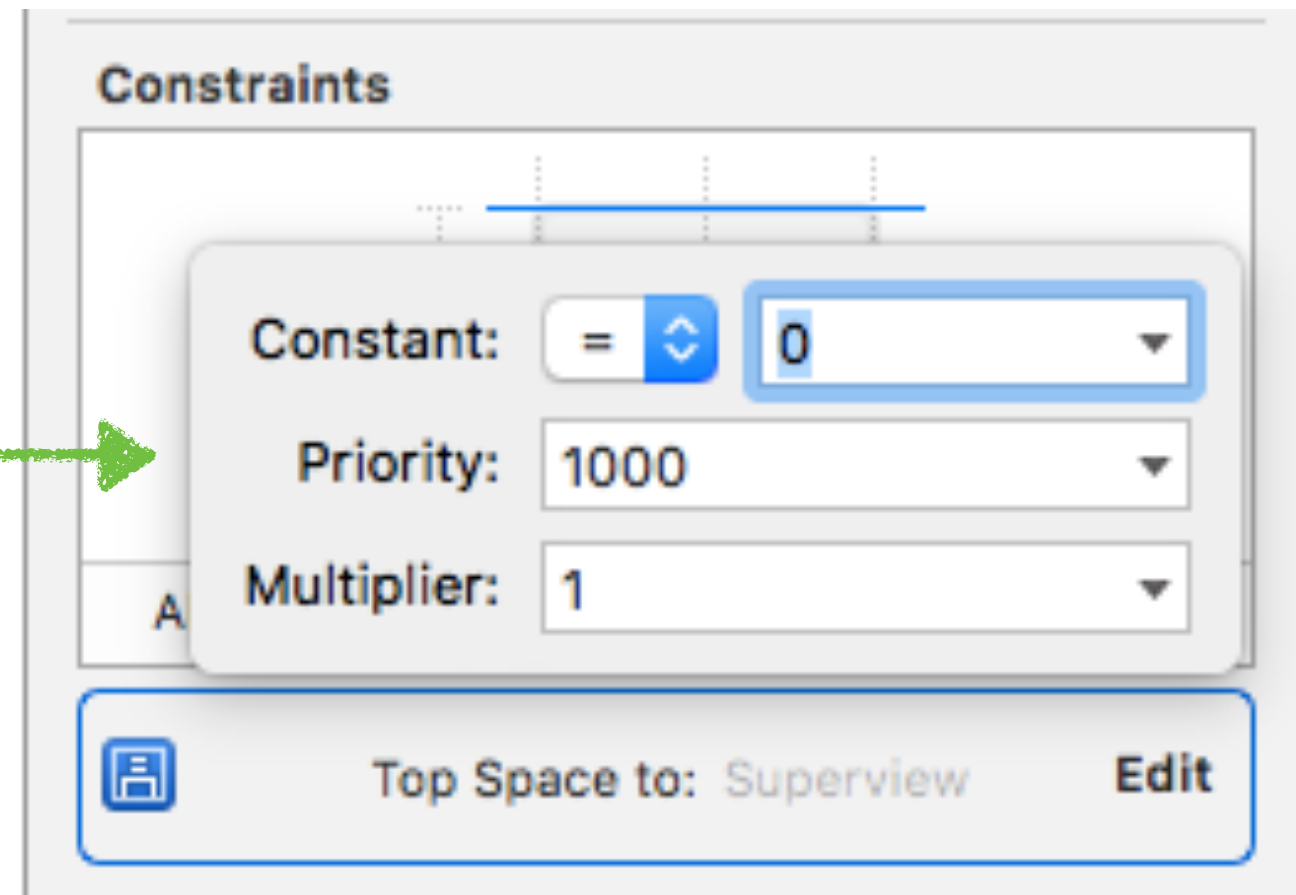
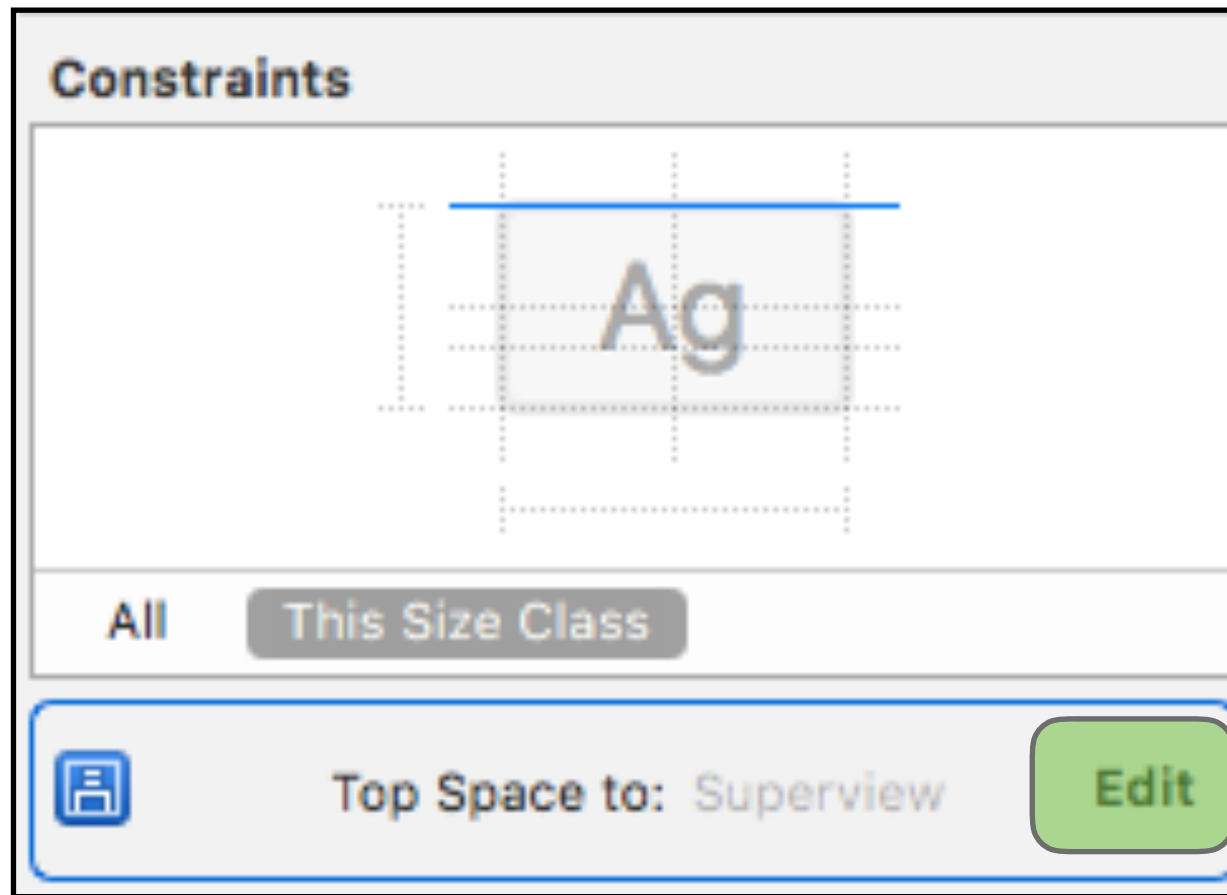
제약 설정

강사 주영민

constraint 선택 설정



constraint Edit 버튼 설정



설정 방법

Horizontal Space Constraint

First Item: Stack View.Leading

Relation: Equal

Second Item: Superview.Leading Margin

+ Constant: 27

Priority: 1000

Multiplier: 1

Identifier: Identifier

Placeholder ☐ Remove at build time

+ ☒ Installed

$\text{Item1.Attribute} = \text{Multiplier} \times \text{Item2.Attribute} + \text{Constrant}$

* 같은 Priority에서 같은 제약이 존재 할수 없다.

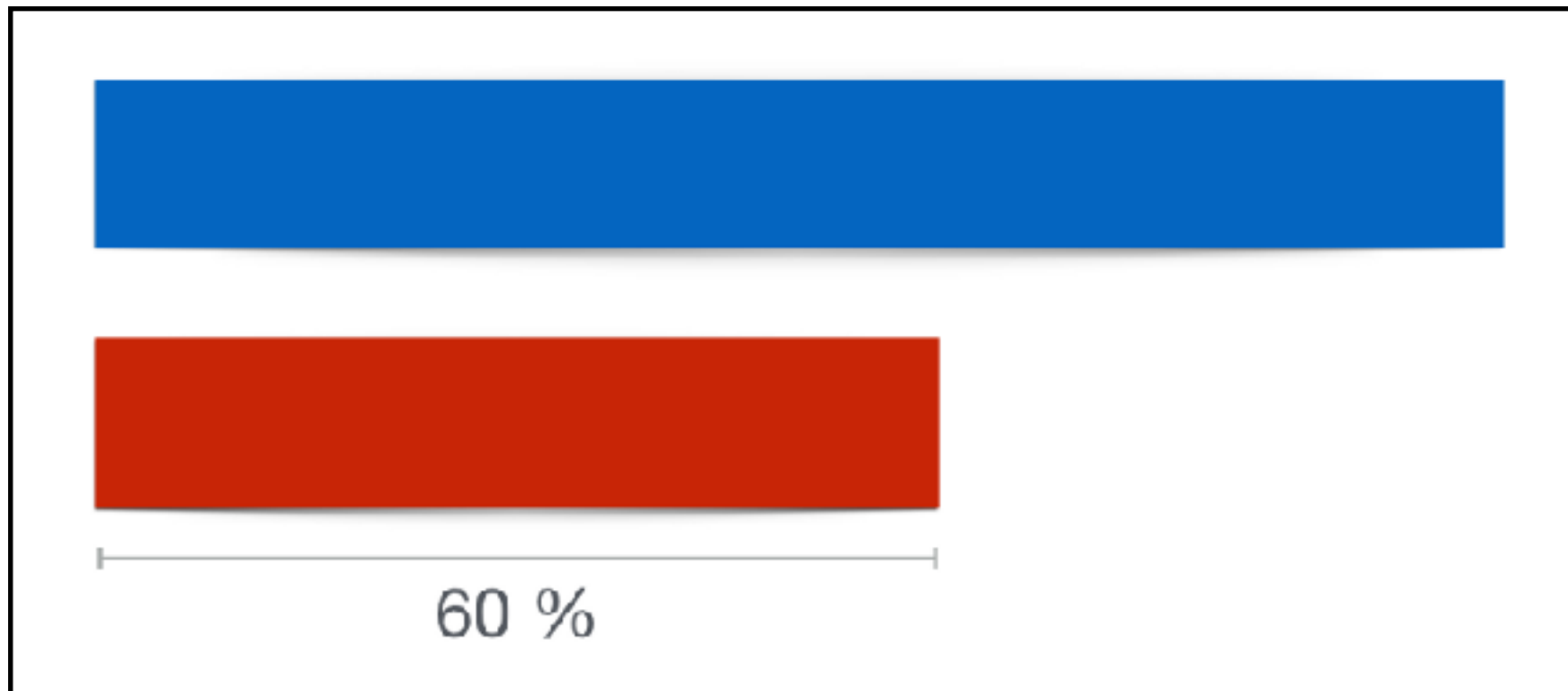
실습



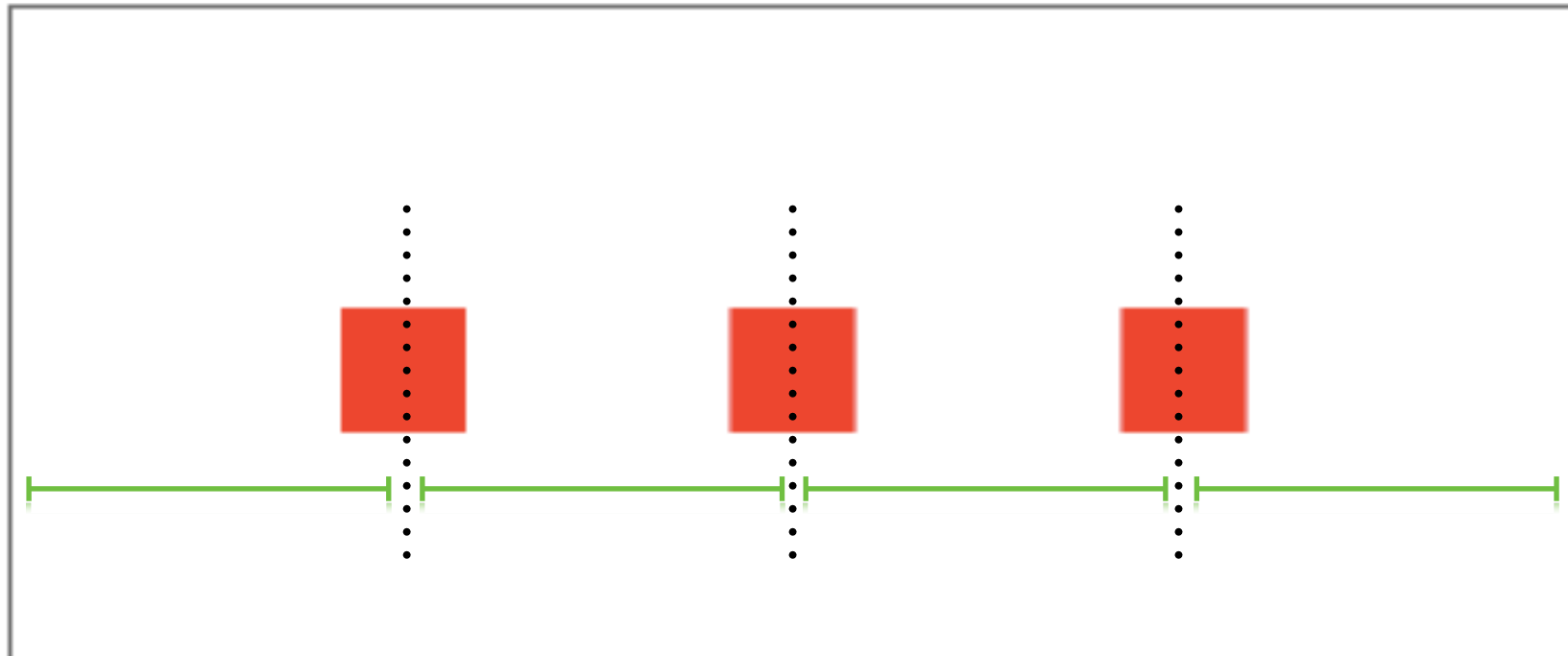
- Red View의 넓이가 Blue View의 넓이의 두 배

실습

- redView의 길이가 BlueView의 60퍼센트 길이



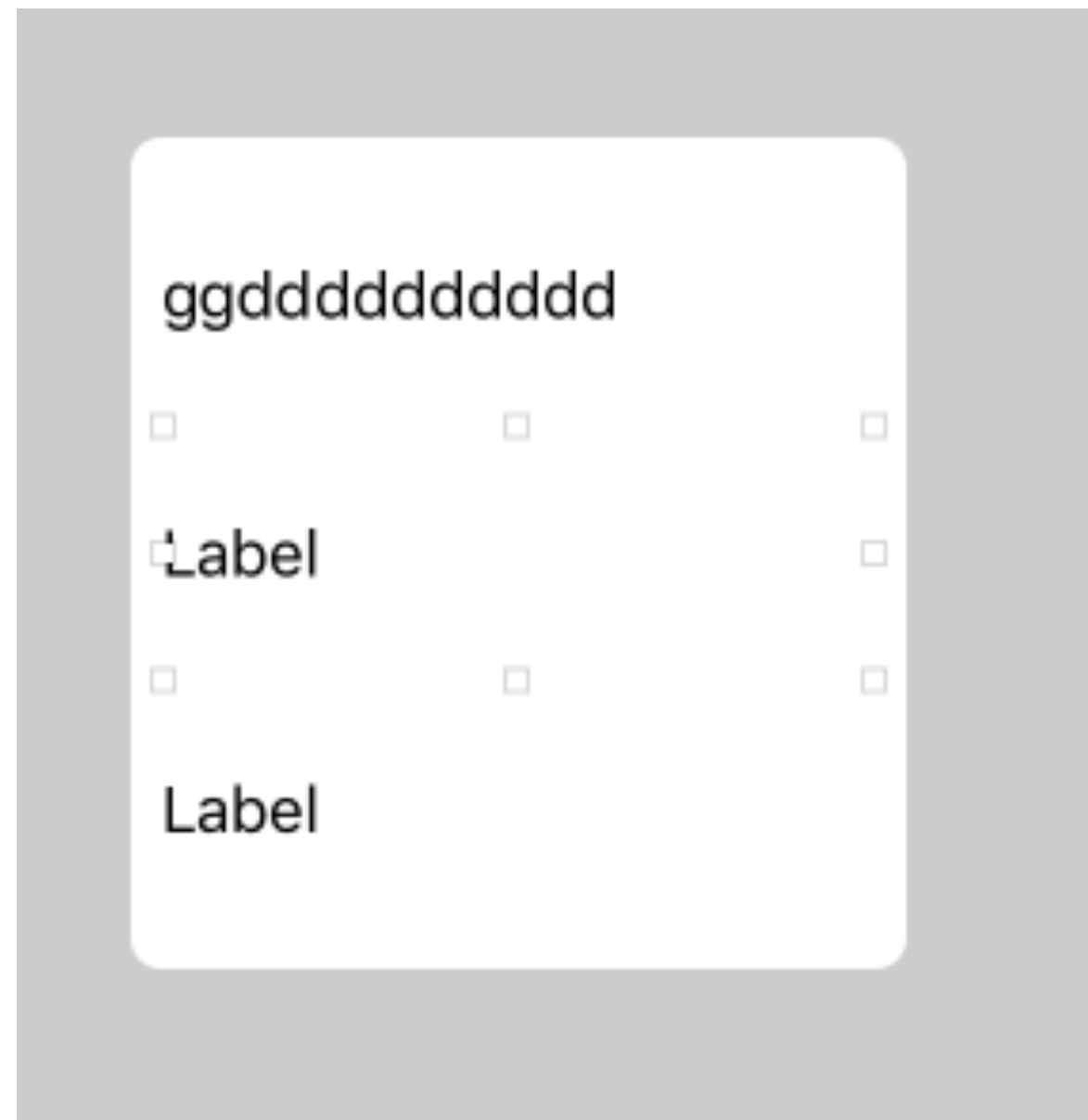
실습 : 동일한 간격 View



- 한변이 30point인 정사각형
- superView의 삼등분 지점에 센터가 위치

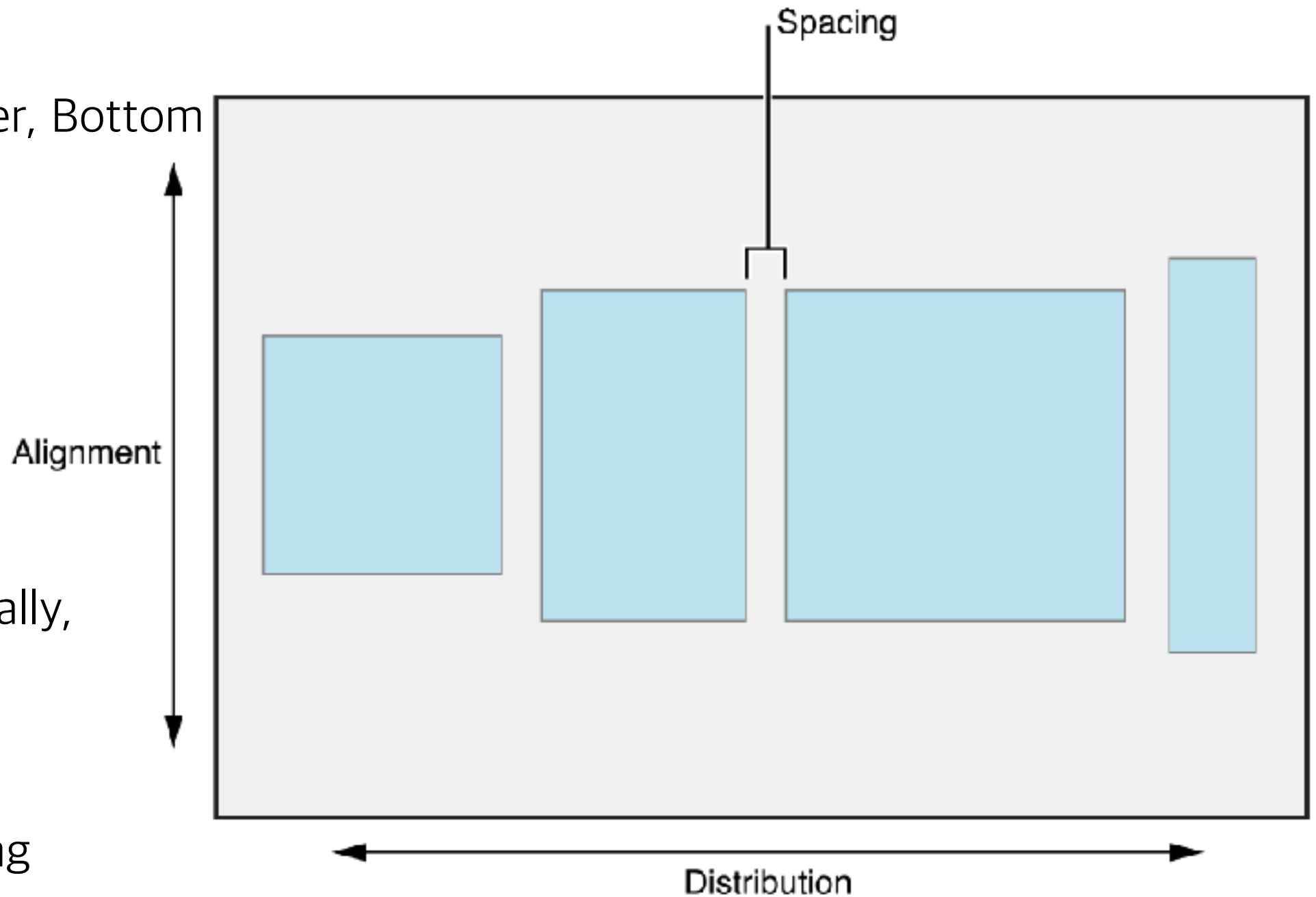
StackView

- 제약없이 View를 자동배치
- StackView의 하위뷰로 추가시 Option에 따라 View가 자동으로 배치
- iOS9이후에 사용 가능
- Horizontal Stack View와 Vertical Stack View로 나뉘어져 있다.



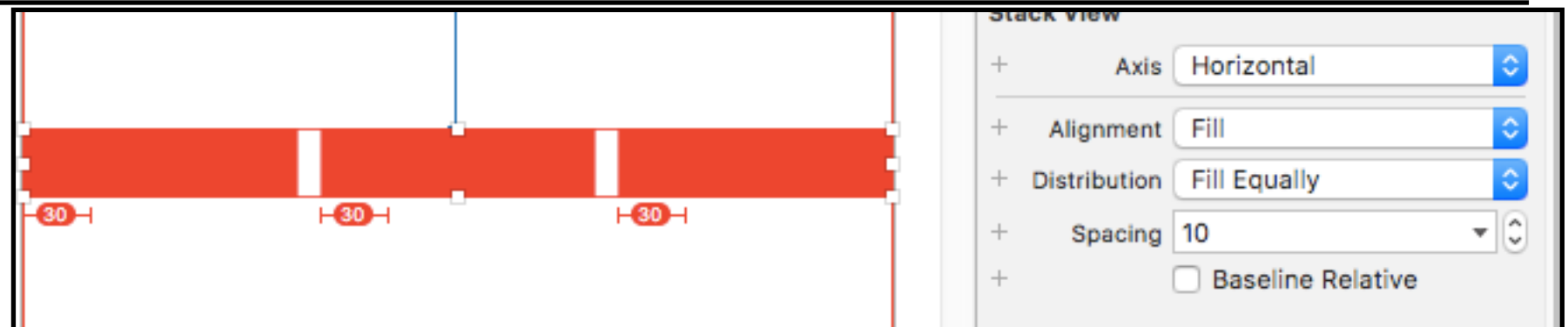
StackView 구조

- Alignment
 - Fill, Top, Center, Bottom
- Distribution
 - Fill,
 - Fill Equally,
 - Fill Proportionally,
 - Equal Spacing,
 - Equal Centering

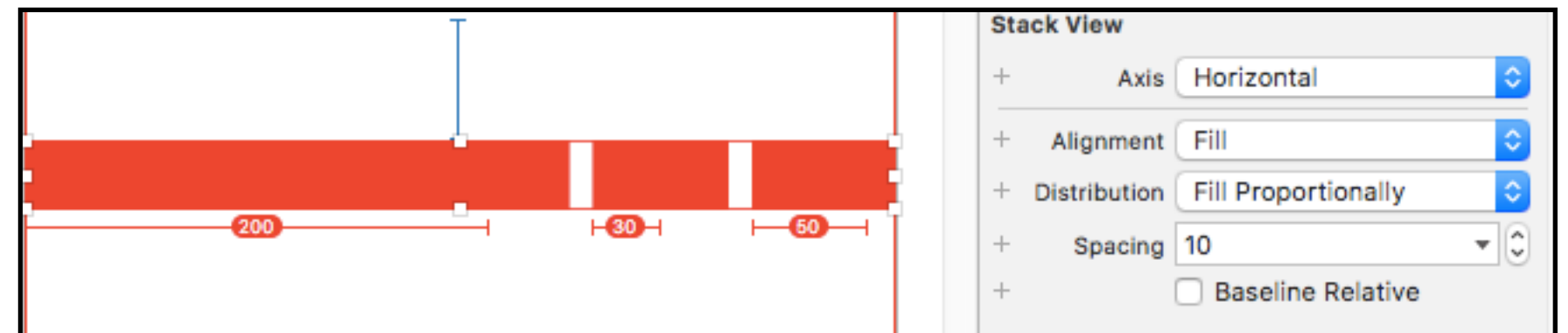


StackView Distribution

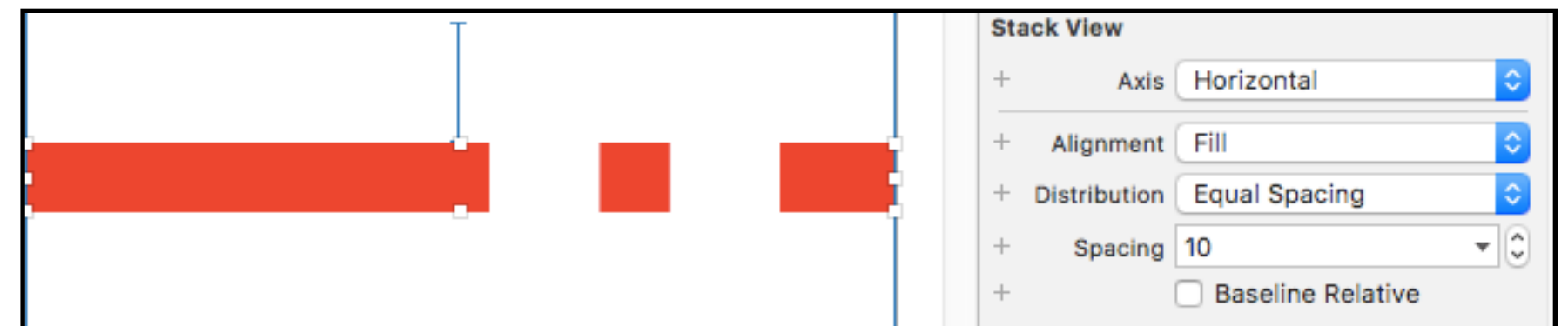
- Fill Equally,



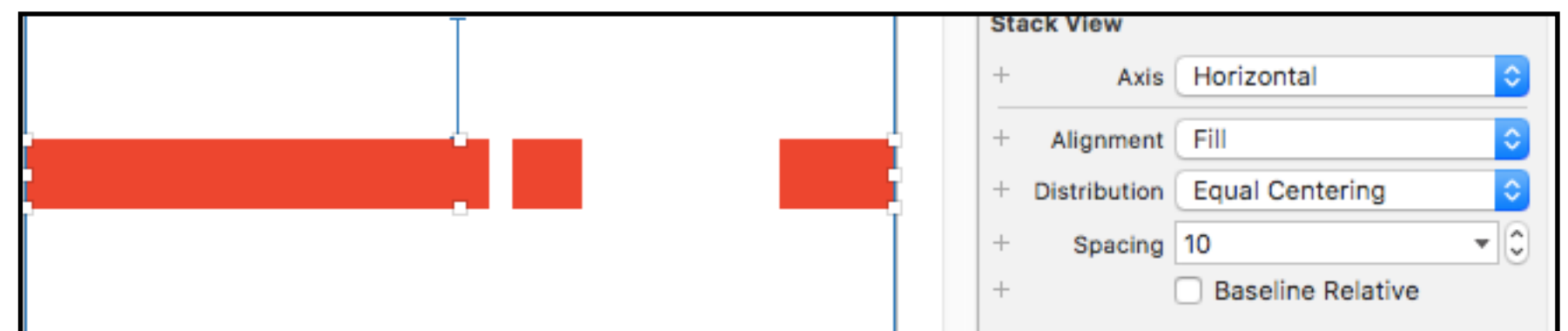
- Fill Proportionally,



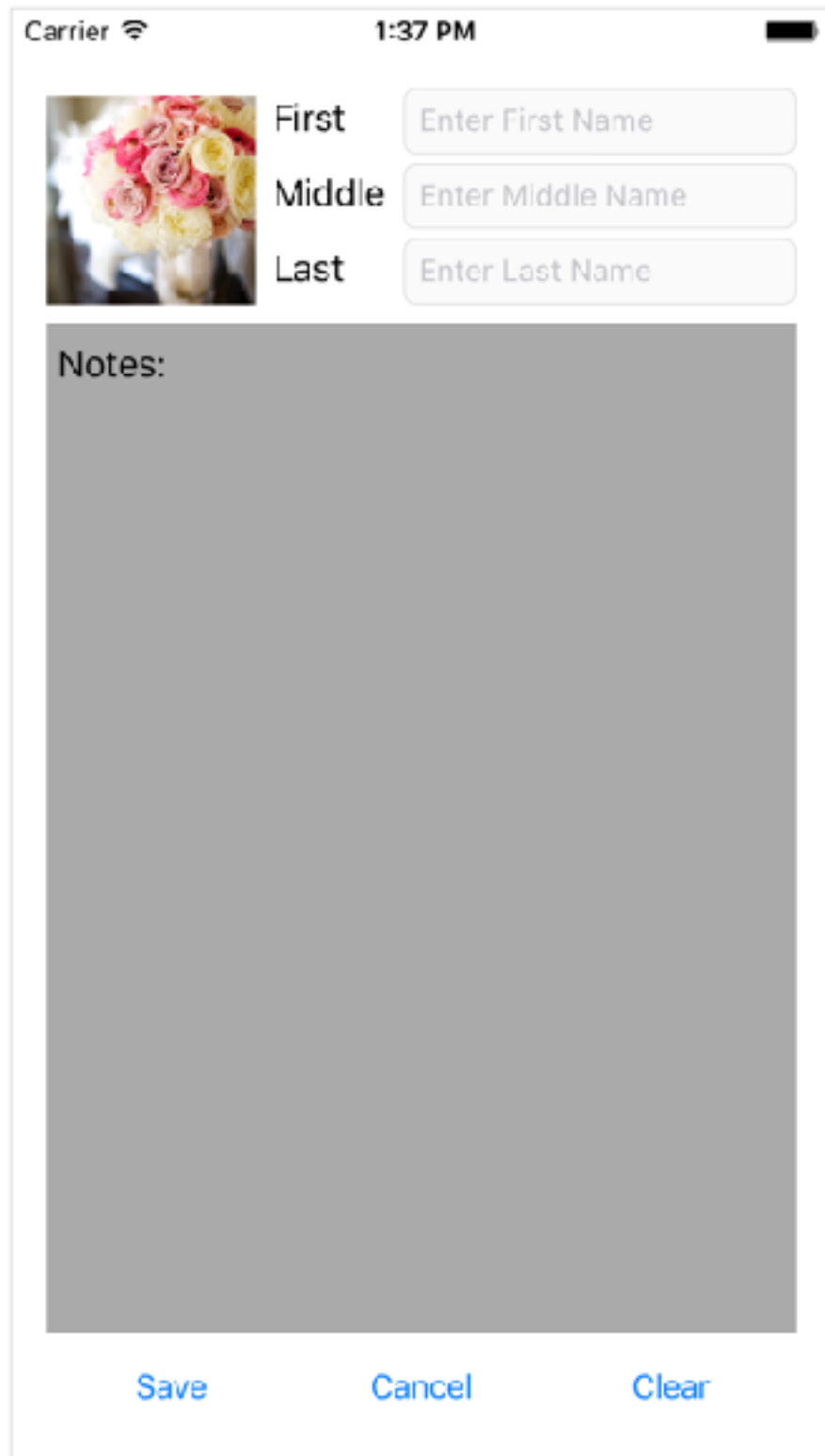
- Equal Spacing,



- Equal Centering



실습



A mobile application interface for entering a name. At the top, the status bar shows 'Carrier', signal strength, '1:37 PM', and battery level. Below the status bar is a header area with a small image of a bouquet of flowers on the left. To the right of the image are three text input fields labeled 'First', 'Middle', and 'Last'. Each field has a placeholder text 'Enter First Name', 'Enter Middle Name', and 'Enter Last Name' respectively. Below the input fields is a large, empty text area labeled 'Notes:'. At the bottom of the screen are three buttons: 'Save', 'Cancel', and 'Clear'.

- 어떻게 만들어야 할까요?