

AutoLayout

뷰에 주어진 제약조건에 따라 뷰의 크기와 위치를 동적으로 계산해 배치하는 것
외부 또는 내부의 변화에 동적으로 반응하여 유저 인터페이스 구성

External / Internal Changes

외적 변화 요소 (External Changes)

- 서로 다른 기기 및 스크린 크기
- 기기 회전
- iPad / iPhone+ 등에서 Split View 로 진입하거나 빠져나올 때
- ...

내적 변화 요소 (Internal Changes)

- 앱에서 보여지는 콘텐츠의 변화
- 국제화 지원 (텍스트, 날짜와 숫자, RTL 등)
- Dynamic Type 지원 (글꼴 크기)
- ...

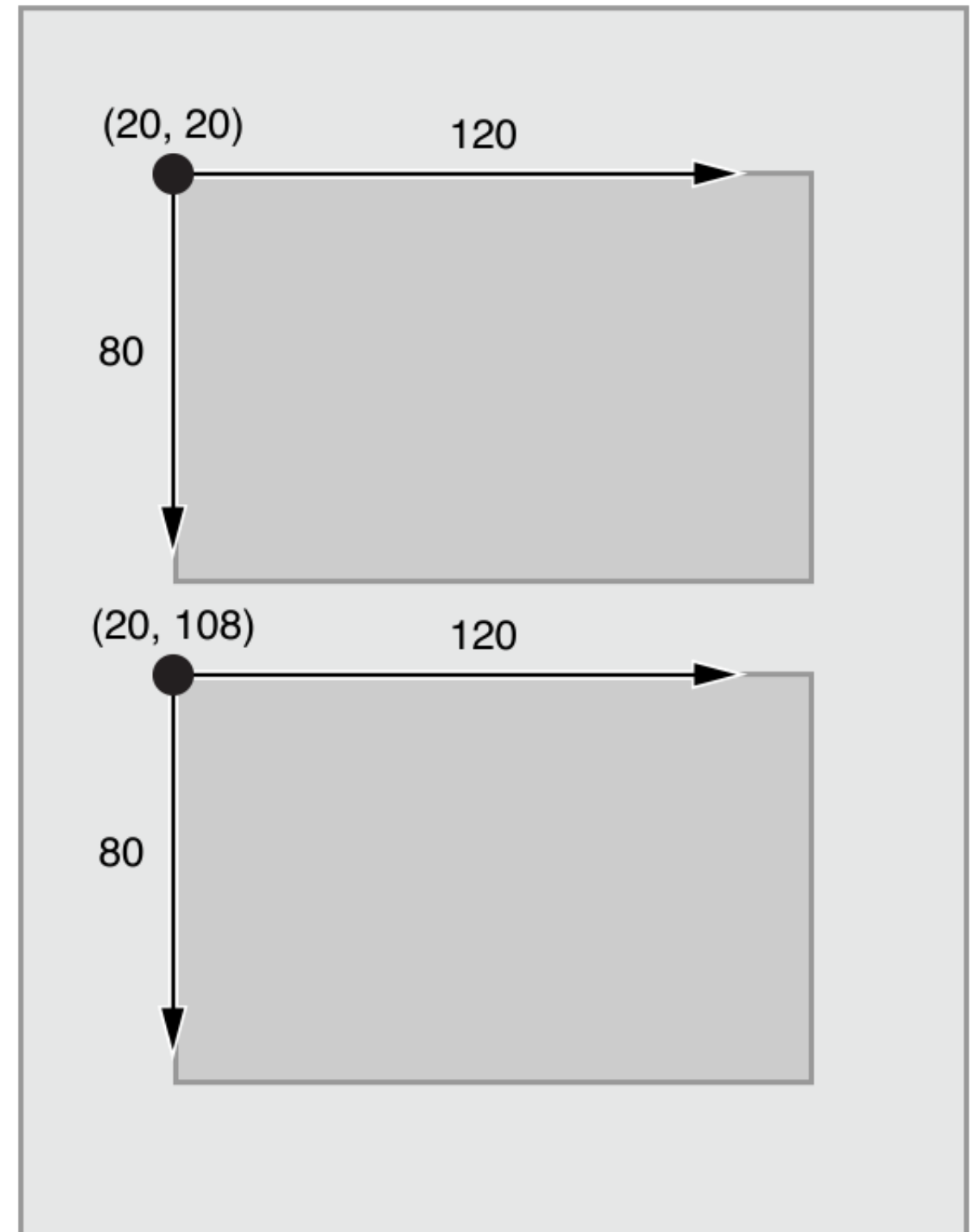
Laying out a user interface

유저 인터페이스 구성을 위한 3가지 주요 접근 방식

- Frame 기반의 프로그래밍 방식
- Autoresizing masks
- Auto Layout

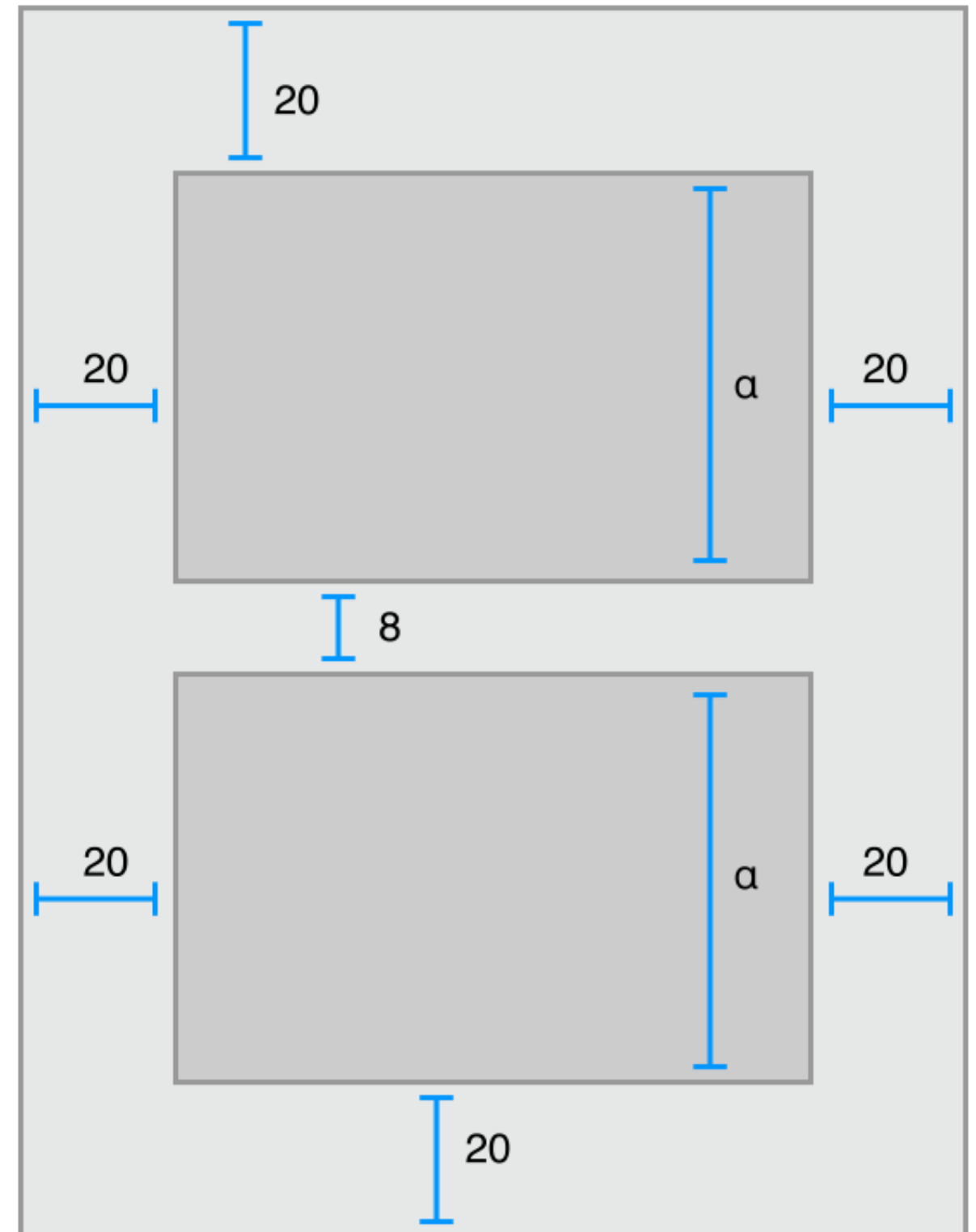
프레임 기반의 프로그래밍 방식

- 원점의 위치와 크기를 통해 그릴 영역 계산
- 가장 유연하며 빠른 성능
- 모든 뷰에 대해 개별적인 설정과 관리
- 동적인 변화에 대한 설계 및 디버그, 유지 관리에 많은 노력 필요

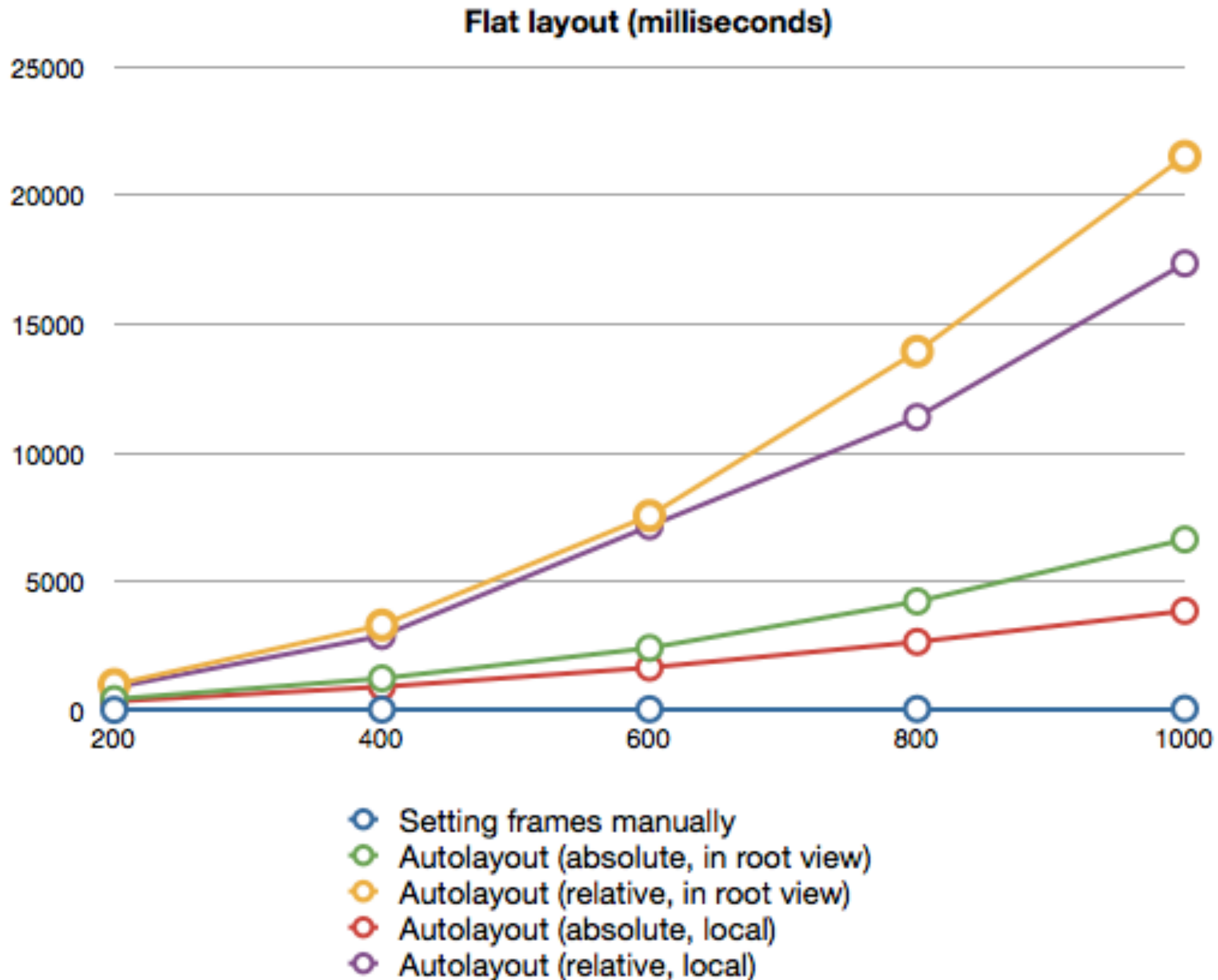


오토레이아웃 방식

- 제약 조건을 이용해 유저 인터페이스 정의
- 뷰간의 관계 설정을 통한 크기와 위치 계산
- 내/외부 변경 사항에 동적으로 반응
- Frame 기반에 비해 느린 성능



Benchmark



Control-Dragging Constraints



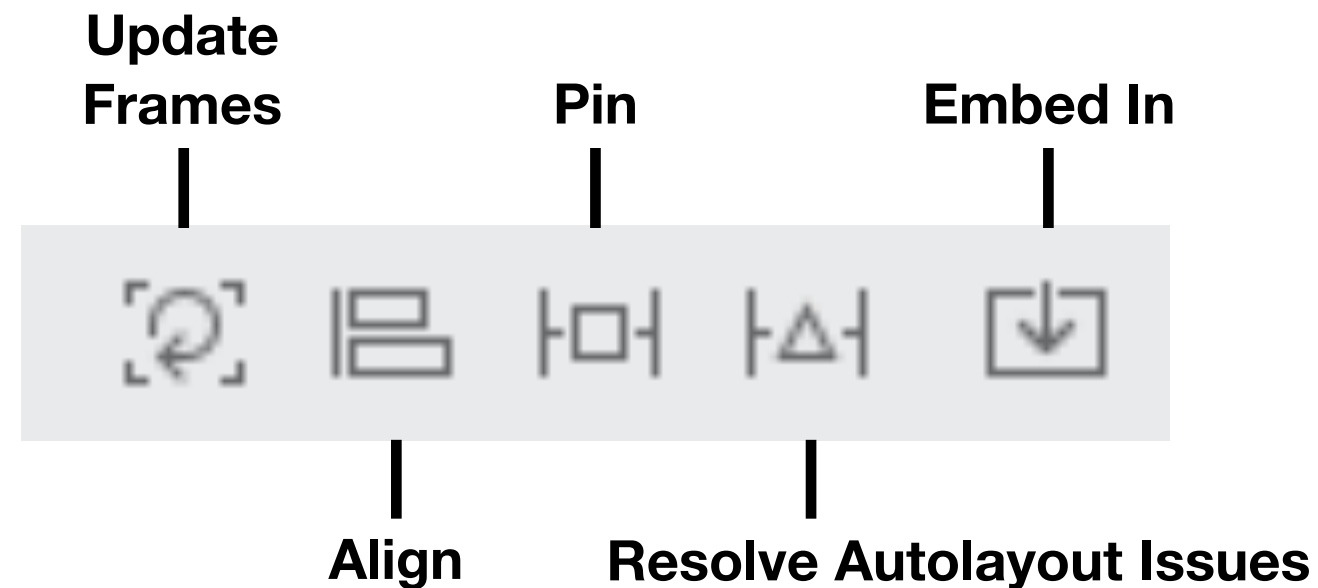
Name Horizontal Spacing

Top
Center Vertically
Baseline
Bottom

Equal Widths
Equal Heights
Aspect Ratio

Hold Shift to select multiple
Hold Option for alternates










Auto Layout tools








- **Update Frames** : 제약조건과 맞지 않는 뷰 위치 갱신
- **Align** : 정렬에 관한 제약사항 설정
- **Pin** : 간격, 크기, 비율 등에 대한 제약 조건 설정
- **Resolve Autolayout Issues** : 오토레이아웃 관련 문제 해결
- **Embed In** : 컨테이너 뷰 / 뷰컨트롤러 추가

Align Tool

Add New Alignment Constraints

<input type="checkbox"/>	 Leading Edges	<input type="text" value="0"/>
<input type="checkbox"/>	 Trailing Edges	<input type="text" value="0"/>
<input type="checkbox"/>	 Top Edges	<input type="text" value="0"/>
<input type="checkbox"/>	 Bottom Edges	<input type="text" value="0"/>
<input type="checkbox"/>	 Horizontal Centers	<input type="text" value="0"/>
<input type="checkbox"/>	 Vertical Centers	<input type="text" value="0"/>
<input type="checkbox"/>	 First Baselines	<input type="text" value="0"/>
<input type="checkbox"/>	 Horizontally in Container	<input type="text" value="0"/>
<input type="checkbox"/>	 Vertically in Container	<input type="text" value="0"/>

Add Constraints

Pin Tool

Add New Constraints

84

169 168

212

Spacing to nearest neighbor

☒ Constrain to margins

☐ Width 77

☐ Height 30

☐ Equal Widths

☐ Equal Heights

☐ Aspect Ratio

Add Constraints

Add New Constraints

84

168

212

Spacing to nearest neighbor

Constrain to margins

Use Standard Value
Use Current Canvas Value

- ☐ View (current distance = 532)
- ☐ Safe Area (current distance = 532)
- ☐ Image View (current distance = 292)
- ☐ Segmented Control (current distance = 241)
- ☒ Stack View (current distance = 84)

Resolve Auto Layout Issues Tool

Selected Views

Update Constraint Constants

Add Missing Constraints

Reset to Suggested Constraints

Clear Constraints

All Views in Predicate View Controller

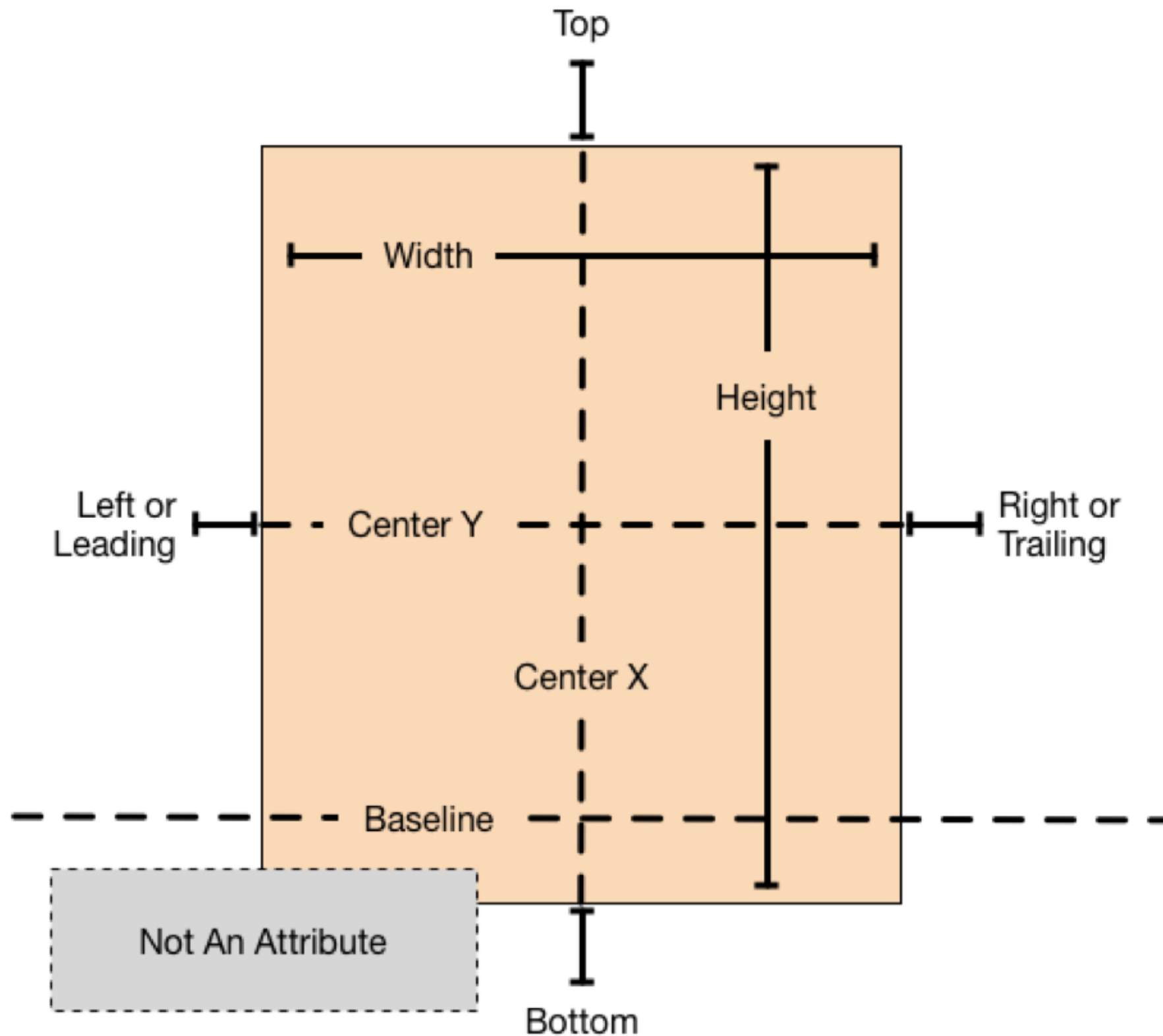
Update Constraint Constants

Add Missing Constraints










Reset to Suggested Constraints

Clear Constraints

Auto Layout Attributes



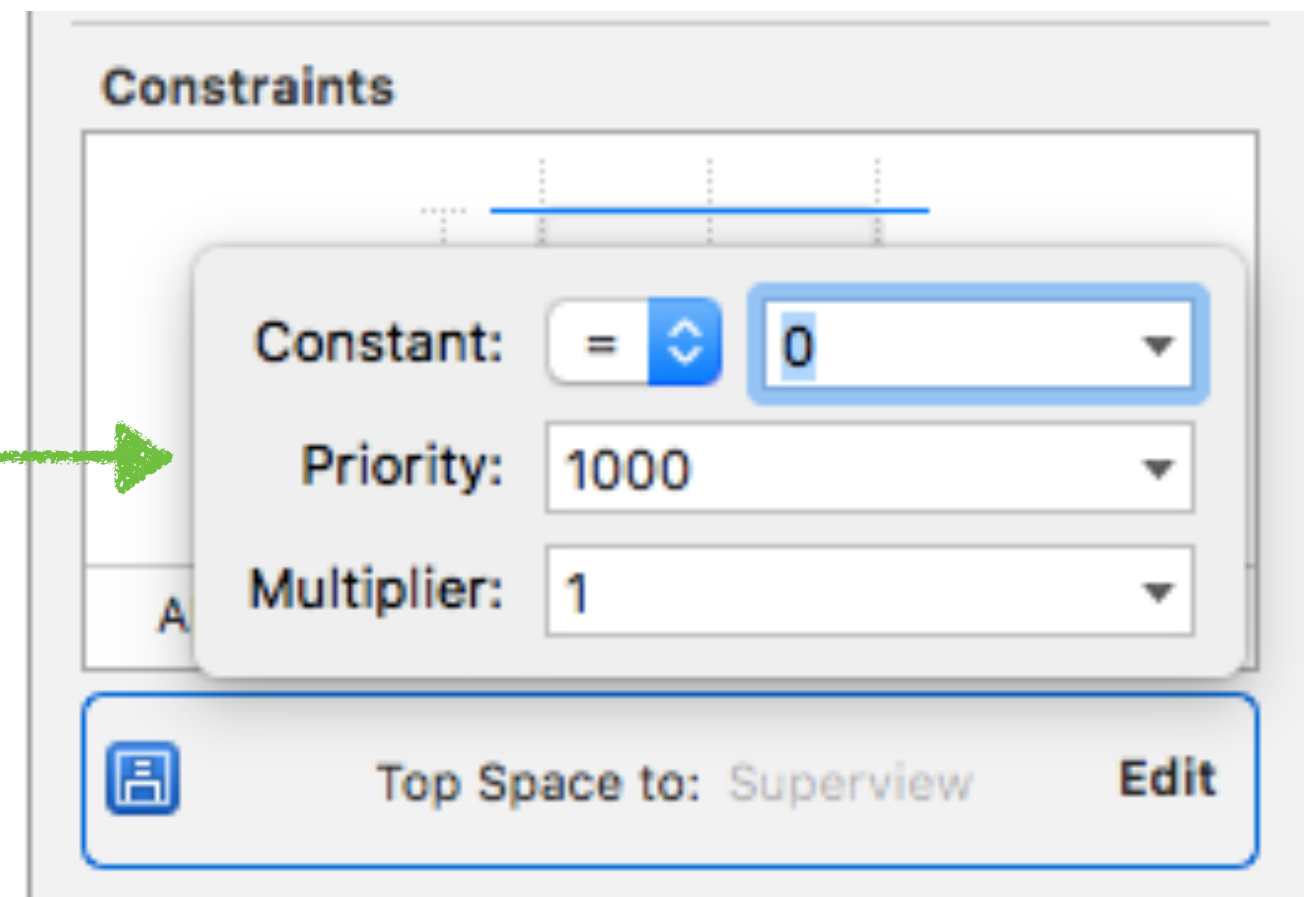
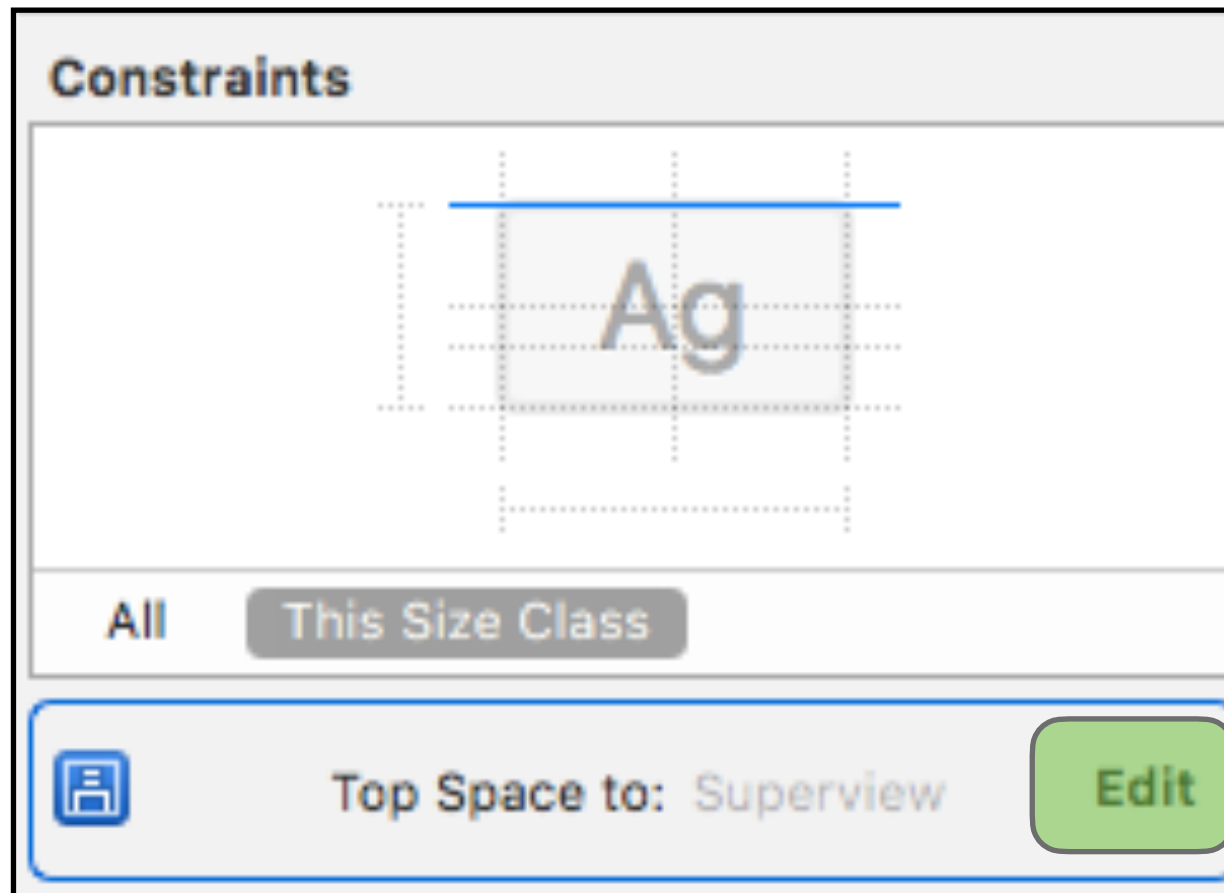
Constraint

- ▼  Constraints
 -  Safe Area.trailing = Imag...
 -  Safe Area.trailing = Stac...
 -  Image View.leading = Saf...
 -  Image View.top = Safe Ar...
 -  Segmented Control.cent...
 -  Segmented Control.top...
 -  Stack View.top = Segme...
 -  Stack View.leading = Saf...

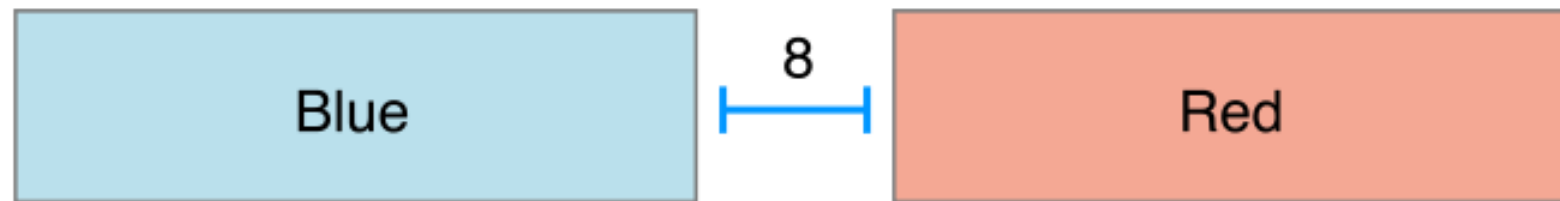
Leading Alignment Constraint

First Item	Image View.Leading	➔	▼
Relation	Equal		⬆ ⬇ ⬆
Second Item	Safe Area.Leading	➔	▼
+ Constant	30	▼	⬆ ⬇ ⬆
Priority	1000	▼	⬆ ⬇ ⬆
Multiplier	1	▼	⬆ ⬇ ⬆
Identifier	Identifier		
Placeholder	<input type="checkbox"/> Remove at build time		
+	<input checked="" type="checkbox"/> Installed		

Constraint



Anatomy of a Constraint




$$\text{RedView.Leading} = 1.0 \times \text{BlueView.trailing} + 8.0$$


Diagram illustrating the components of the constraint equation:


- Item 1:** RedView
- Attribute 1:** Leading
- Relationship:** =
- Multiplier:** 1.0
- Item 2:** BlueView
- Attribute 2:** trailing
- Constant:** 8.0



Anatomy of a Constraint



Horizontal Space Constraint



First Item 

Relation 

Second Item 

+ Constant  

Priority  

Multiplier  

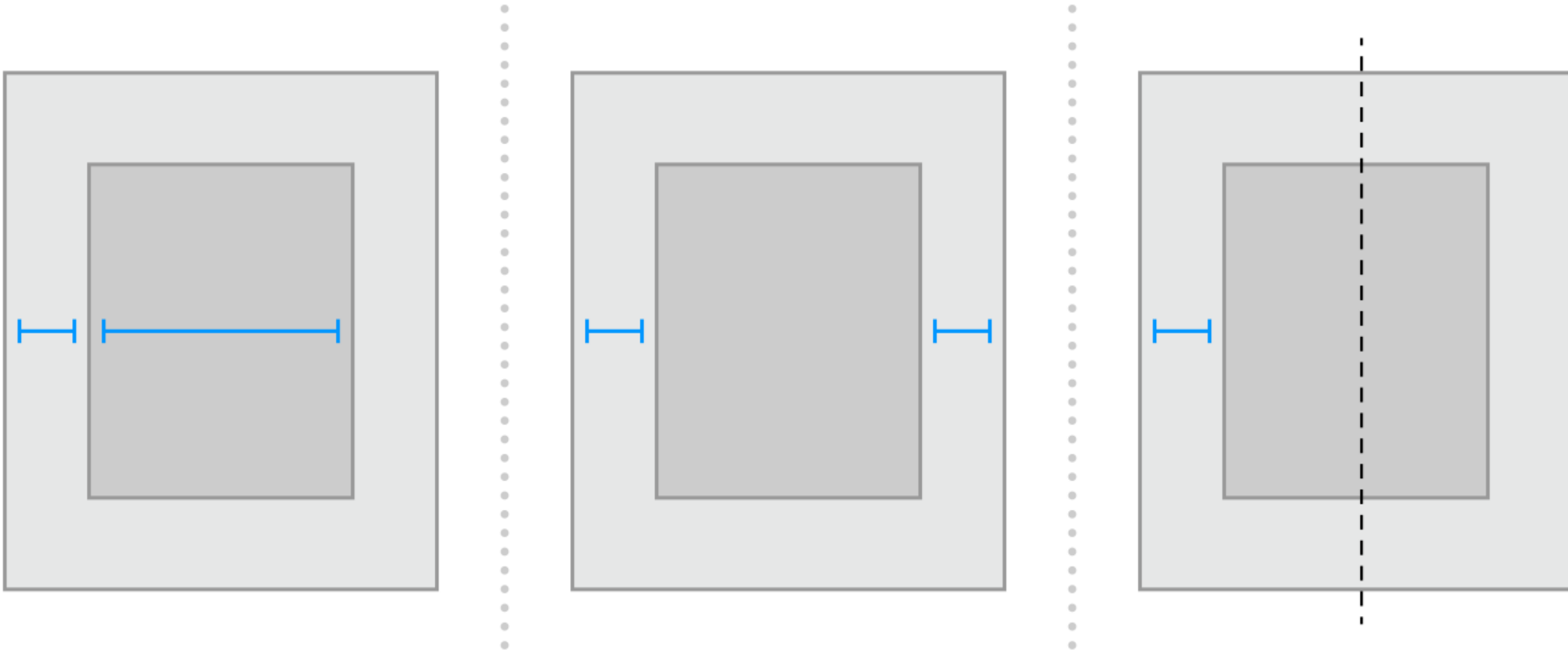
Identifier

Placeholder ☐ Remove at build time

+ ☒ Installed

Item1.Attribute = Multiplier X Item2.Attribute + Constraint

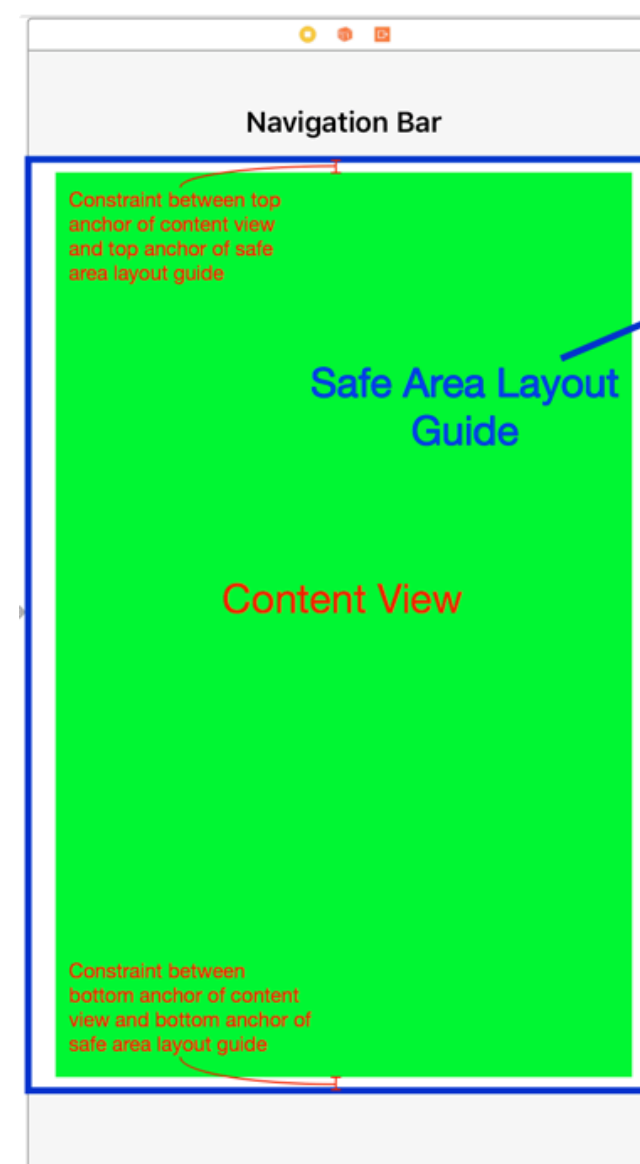
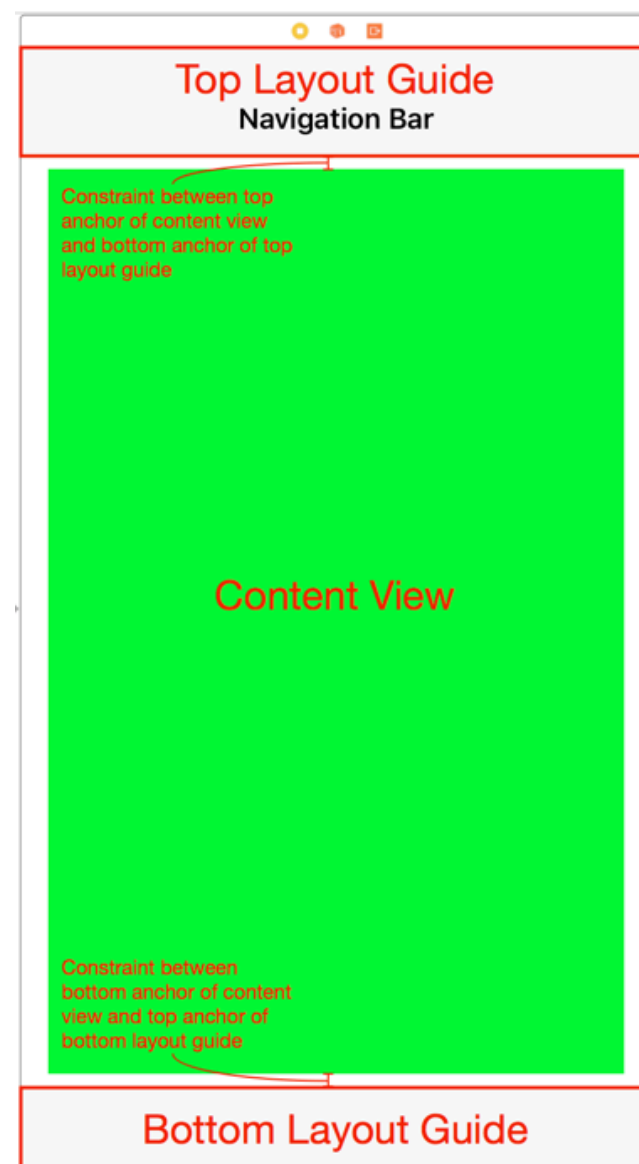
Nonambiguous, Satisfiable Layouts



Layout Guide

iOS 7 - Top / Bottom Layout Guide

iOS 11 - Safe Area Layout Guide



Deprecated Message

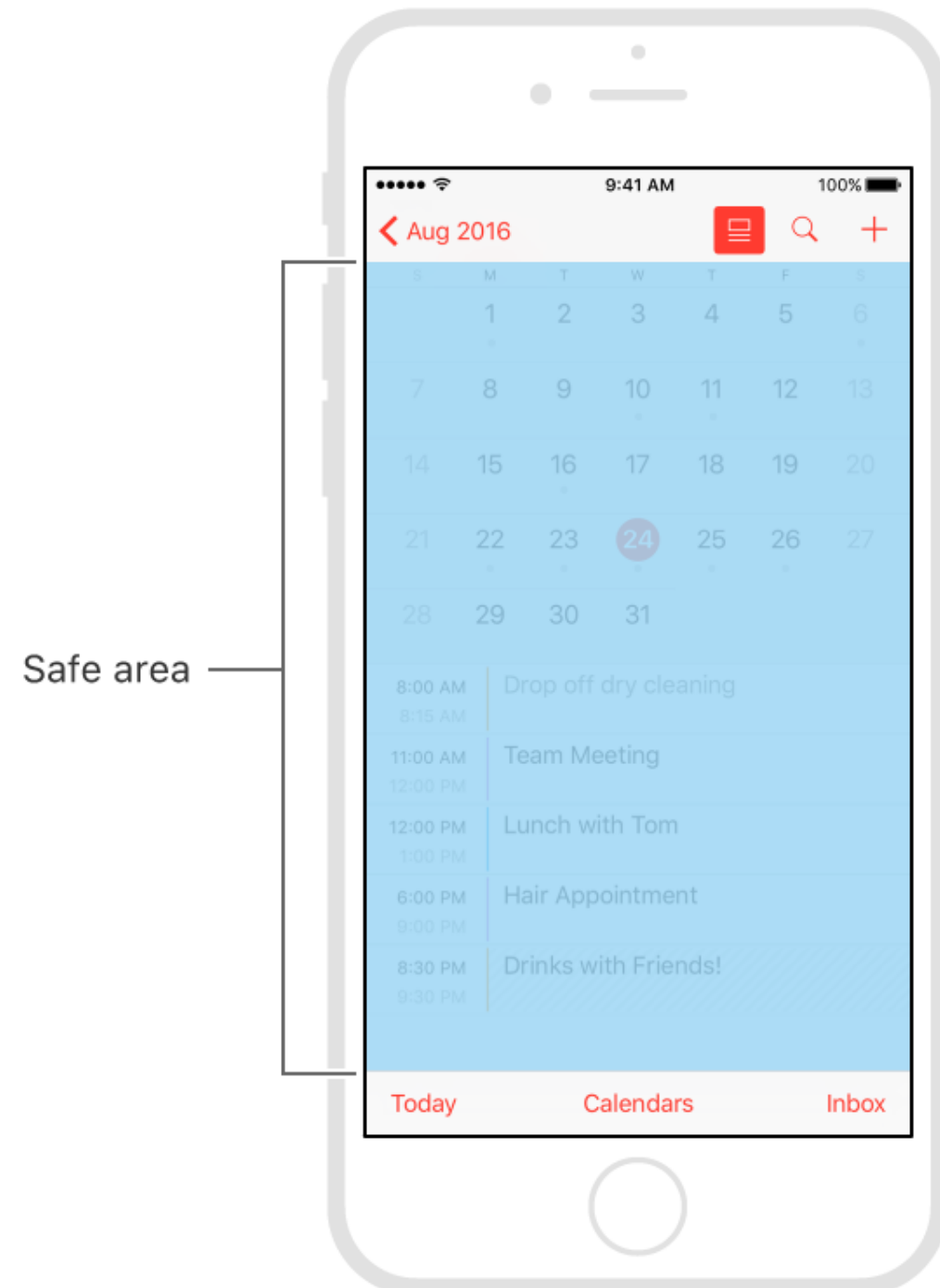
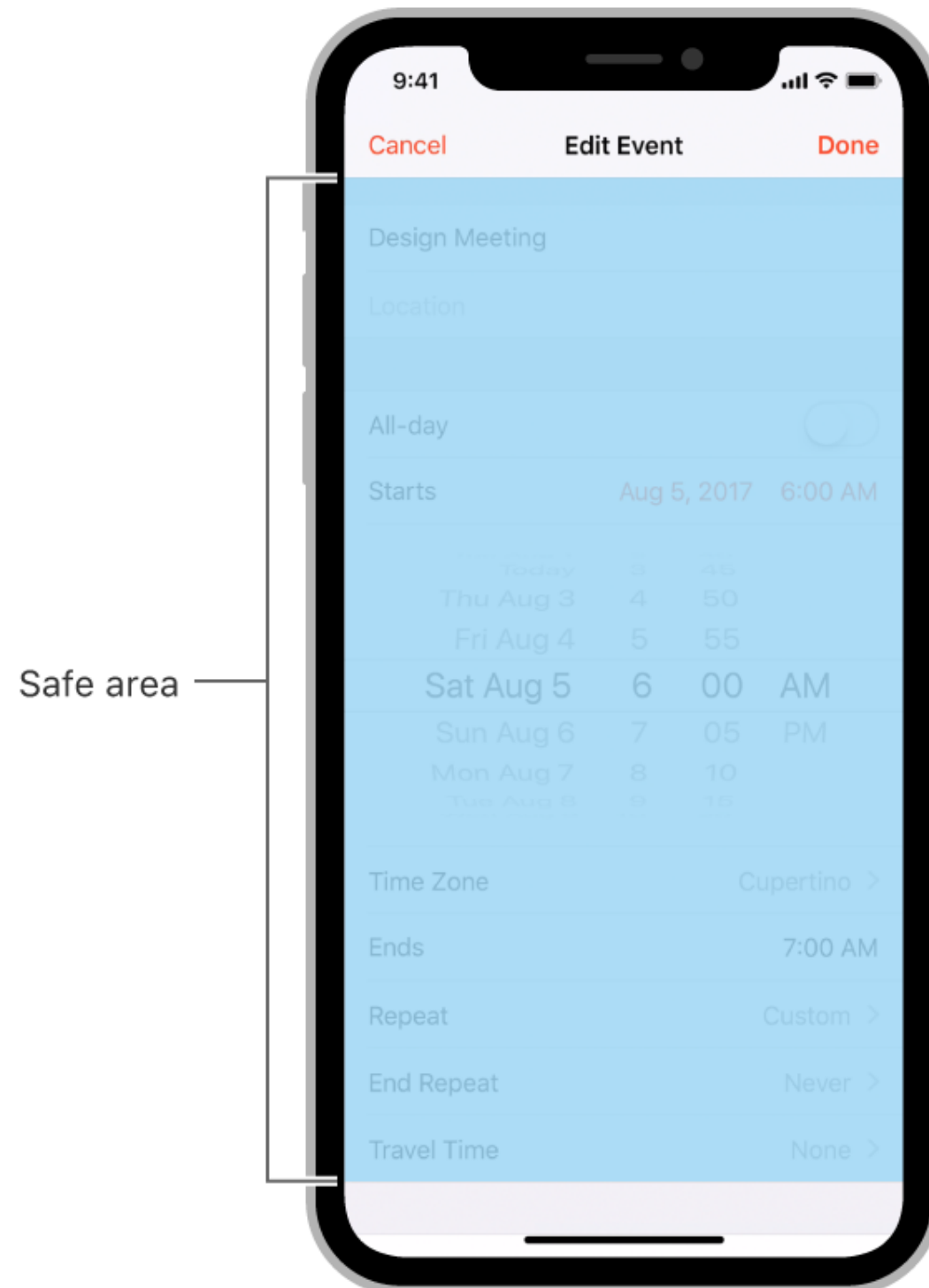
```
@available(iOS, introduced: 7.0, deprecated: 11.0, message: "Use  
view.safeAreaLayoutGuide.topAnchor instead of  
topLayoutGuide.bottomAnchor")
```

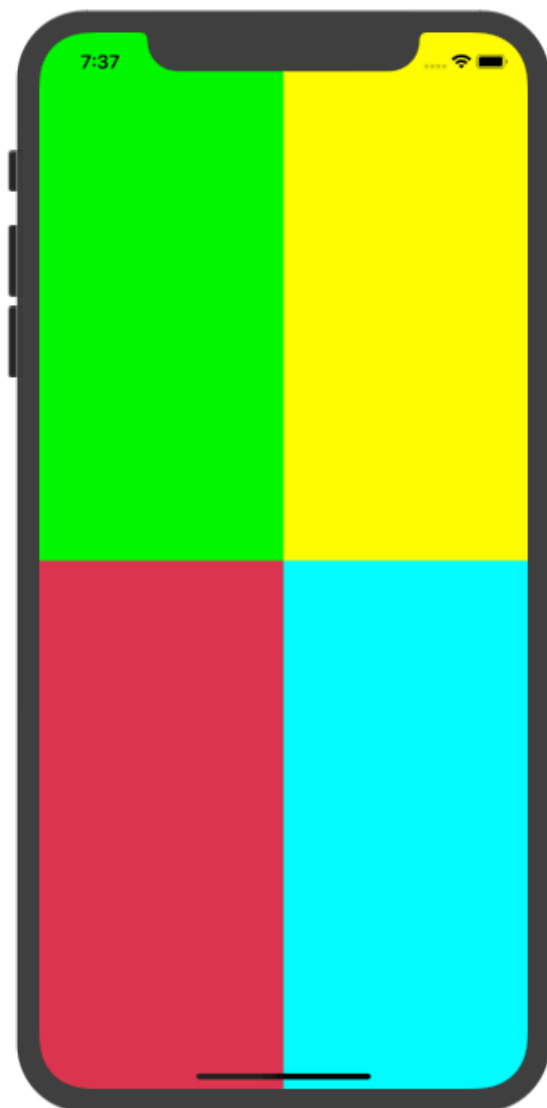
```
open var topLayoutGuide: UILayoutSupport { get }
```

```
@available(iOS, introduced: 7.0, deprecated: 11.0, message: "Use  
view.safeAreaLayoutGuide.bottomAnchor instead of  
bottomLayoutGuide.topAnchor")
```

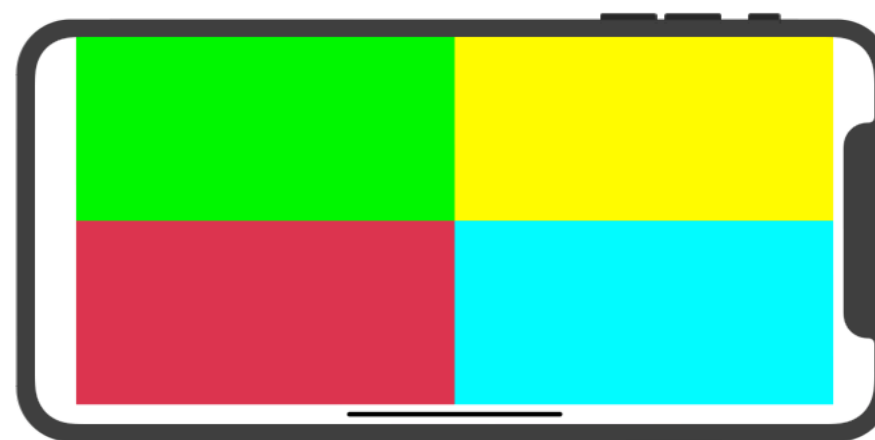
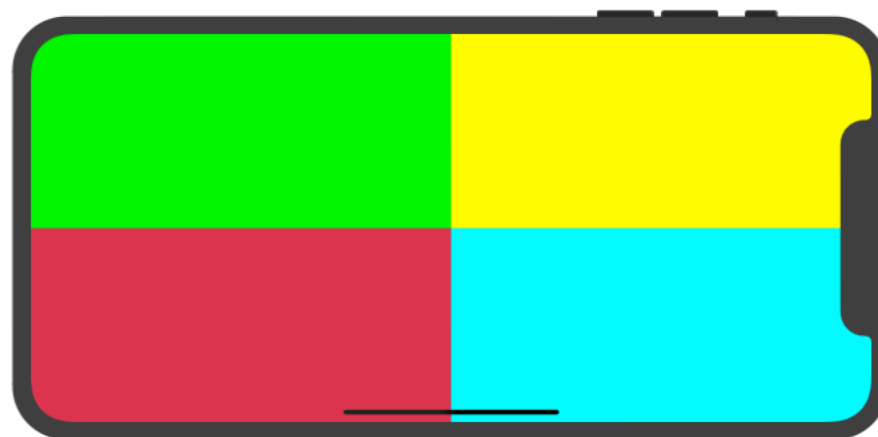
```
open var bottomLayoutGuide: UILayoutSupport { get }
```

The safe area of an interface

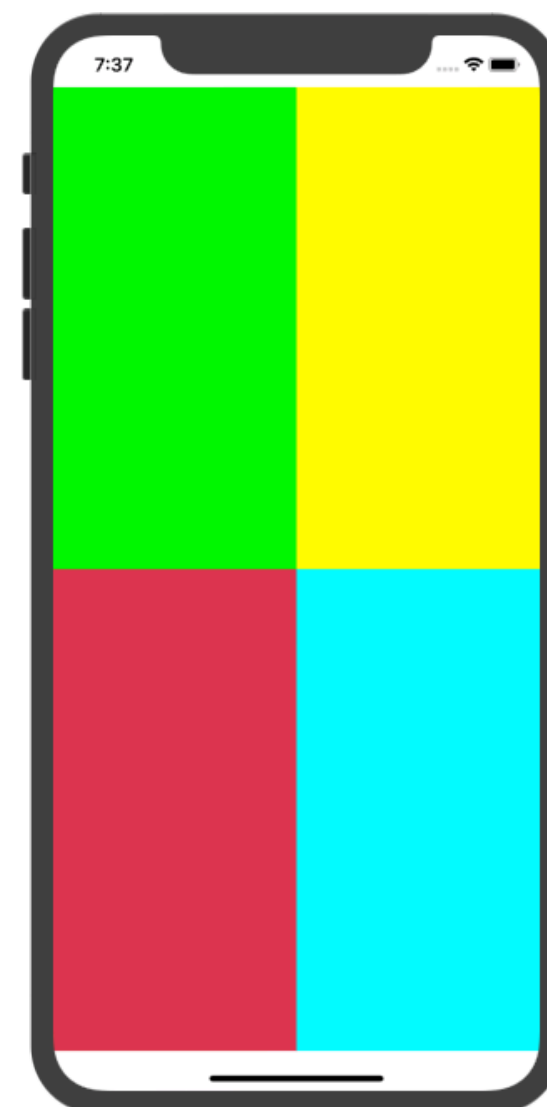




[View]

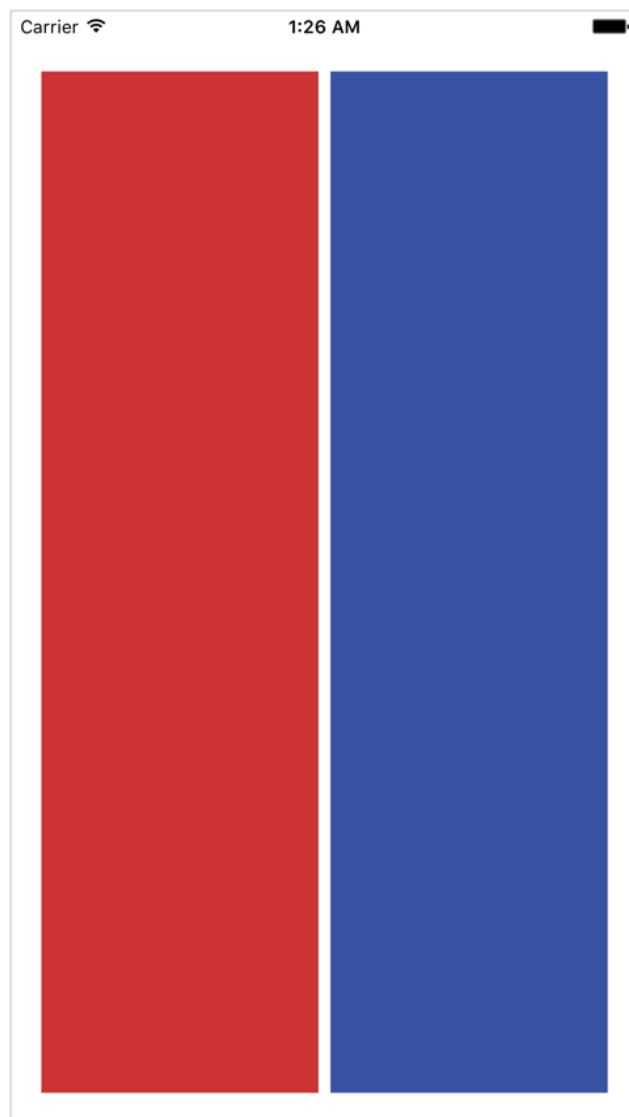


[Safe Area]

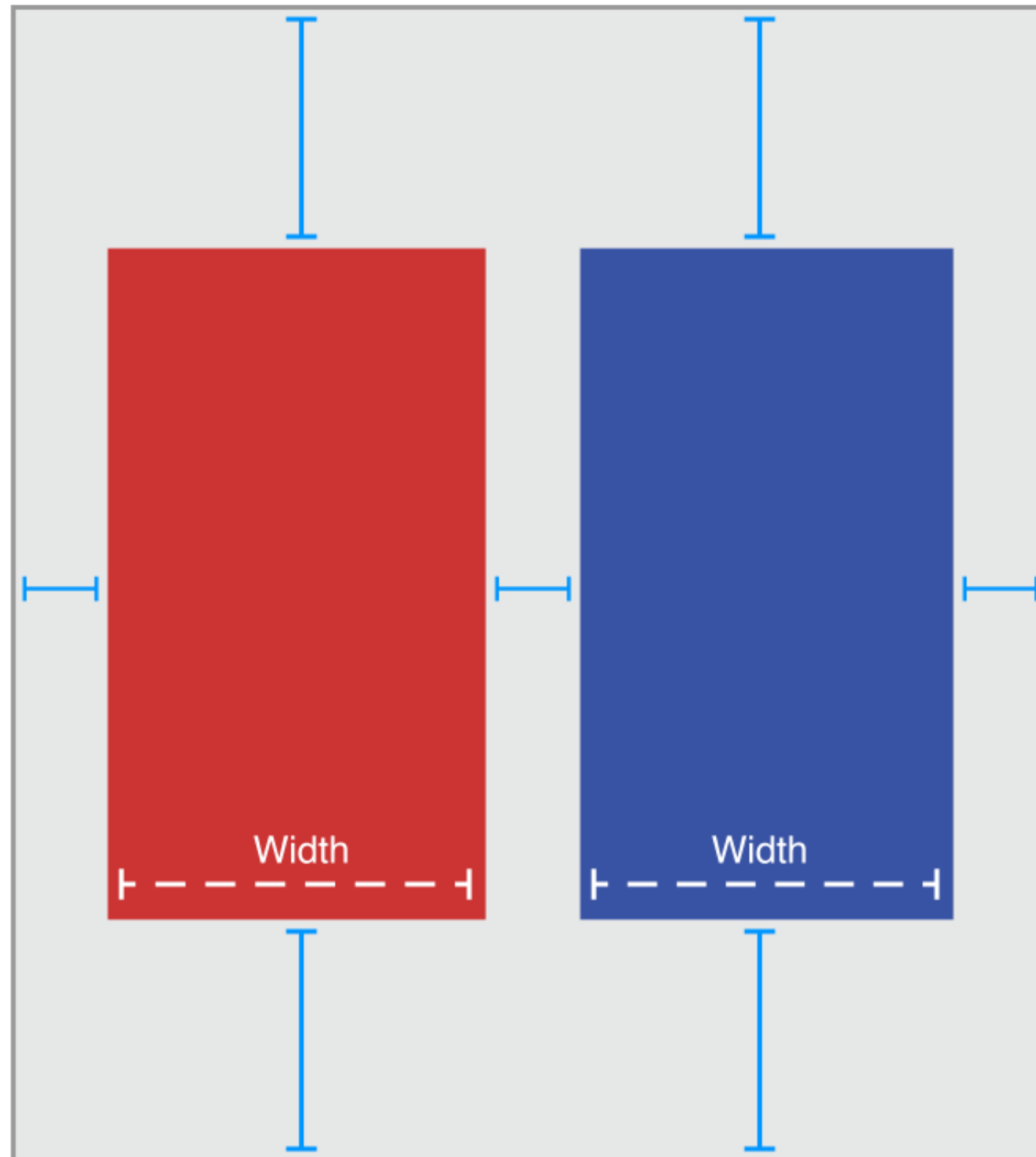


아래 두 개의 뷰를 오토레이아웃을 이용해 배치해보기

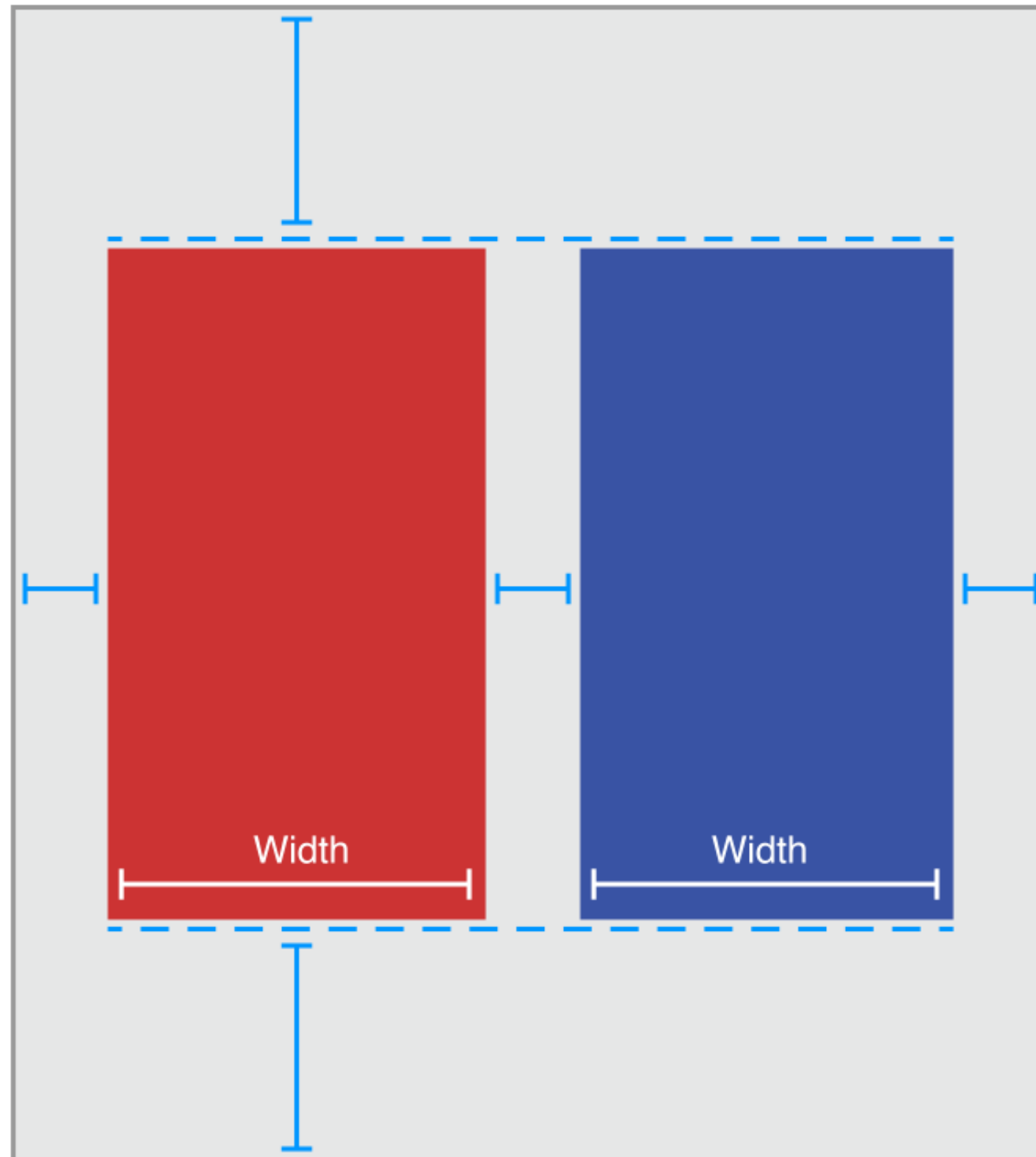
- 각 뷰는 가장자리로부터 각각 20 씩 떨어져있고 뷰 사이 거리는 10 이며 두 뷰의 길이는 동일
- iPhone 8과 iPhone X 이상의 모델에서 실행해보고 비교



Solution 1



Solution 2



아래 두 개의 뷰를 프레임을 이용해 배치해보기

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

