



오토레이아웃





학습목표

- 오토레이아웃의 개념을 이해한다.
 - 오토레이아웃 지원 Tool들의 사용방법을 이해하고 활용할 수 있다.
 - 오토레이아웃에서 크기 제약조건과 여백제약 조건의 적용을 이해하고 앱 개발에 활용할 수 있다.
 - 콘텐츠 고유 크기와 허깅 우선순위, 압축저항 우선순위를 이해하고 활용할 수 있다.
-
- 참고 사이트
 - https://developer.apple.com/library/archive/documentation/UserExperience/Conceptual/AutolayoutPG/index.html#//apple_ref/doc/uid/TP40010853-CH7-SW1



오토레이아웃 개념

■ 동일 소프트웨어 다른 화면

- 하나의 앱이 여러 디바이스에 적절한 화면 크기를 유지하여야 함



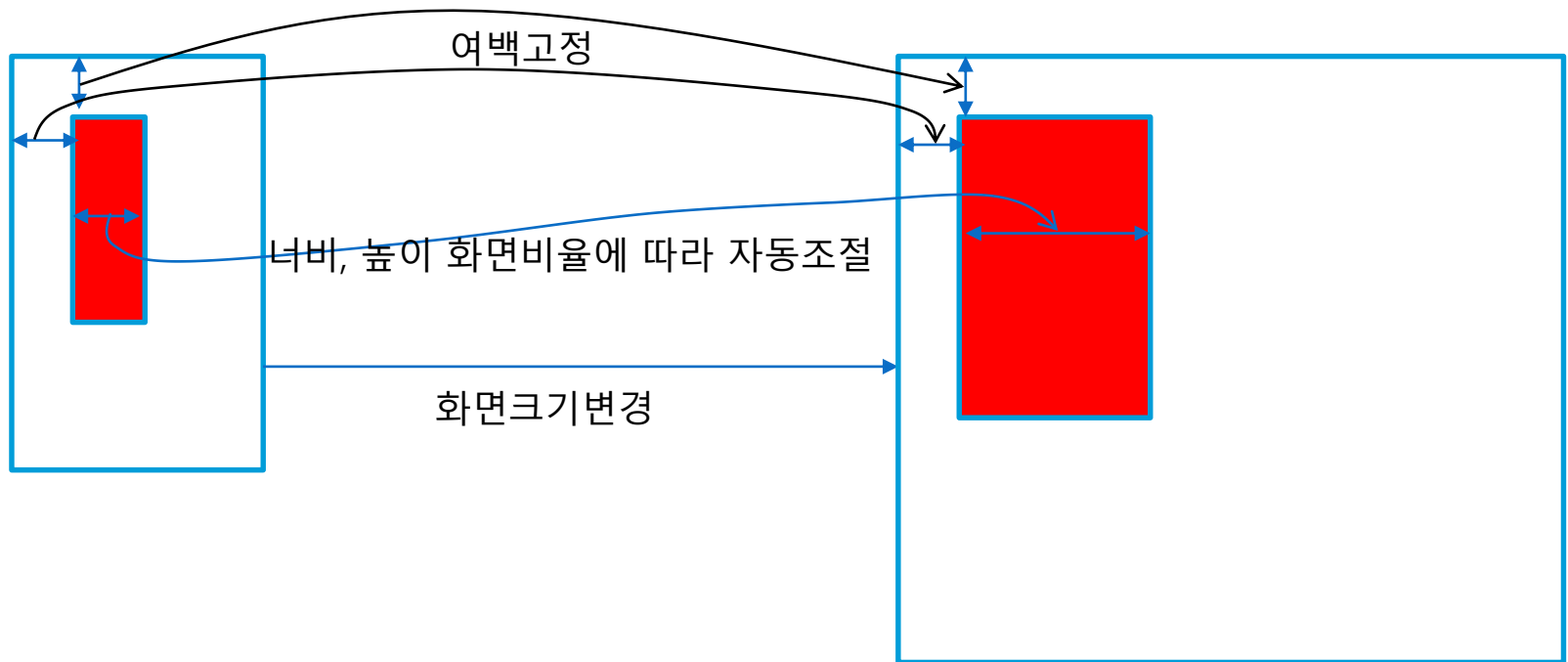
- 오토리사이징과 오토레이아웃이 있음



오토리사이징

■ 오토리사이징(autoResizing)


- 화면의 크기에 따라 View의 크기를 자동 조절
- 기능
 - 여백 고정기능
 - View 크기 조절 기능





오토리사이징

■ 설정 방법

- 각 View 객체별로 설정
- 원하는 View 객체를 선택한 후 크기 속성 inspector()에서 설정

여백의 실선: 화면의 크기에 관계없이 현재 여백으로 고정

View의 실선: 화면의 크기에 따라 View의 크기를 비례적으로 조절



- 상기 설정의 해석

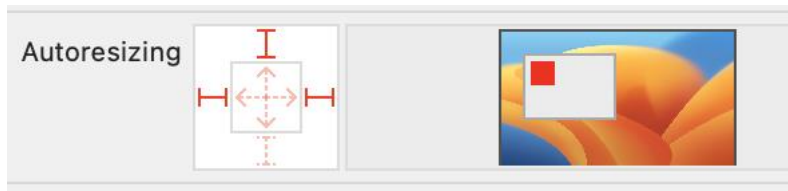
- View의 너비: 화면의 크기에 따라 비례하여 너비가 조절된
- View의 높이: 화면의 크기가 변하여도 높이는 고정
- 왼쪽, 위쪽 여백: 화면의 크기가 변하여도 높이는 고정
- 오른쪽, 아래쪽 여백: View의 너비, View의 높이, 왼쪽, 위쪽 여백이후 남는 값



오토리사이징

■ 설정 방법

- 설정 오류가 있는 경우
 - 아래 설정의 경우 좌.우 여백도 고정이고 View의 너비도 고정이다.
 - 이 경우 View의 너비 고정이 오른쪽 여백 고정보다 우선순이다.



■ iOS 6이후

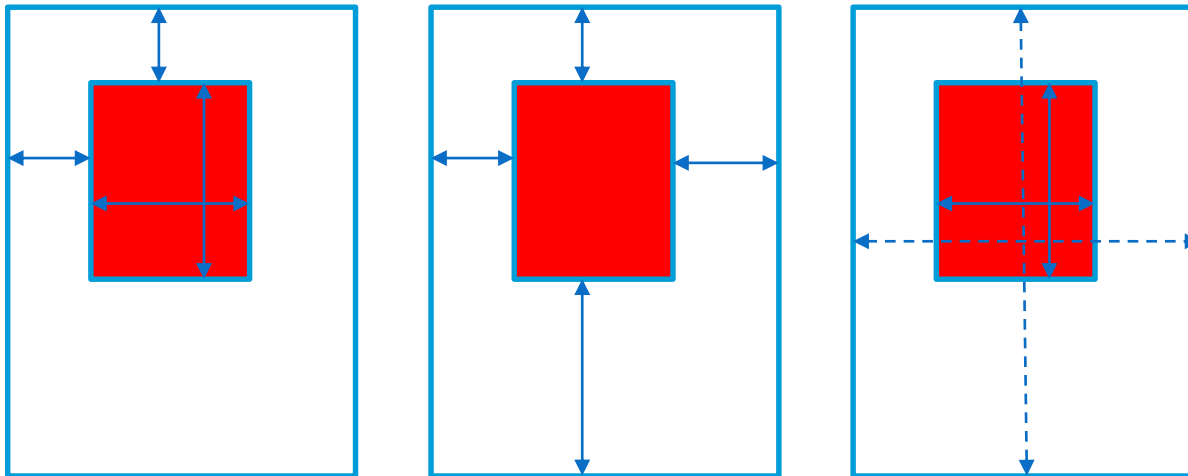
- 오토레이아웃을 사용하면서 오토리사이징은 잘 사용하지 않는다.
- 오토레이아웃과 오토리사이징을 동시에 적용할 수도 있다.



오토레이아웃

■ 오토레이아웃

- 하나의 View에 대하여 어떠한 화면의 크기에 대해서도 좌측상단좌표(left, height)와 크기(width, height)를 계산할 수 있으면 Satisfiable layout라 함
- 화면상의 모든 View가 Satisfiable layout이면 오토레이아웃을 만족한다고 함

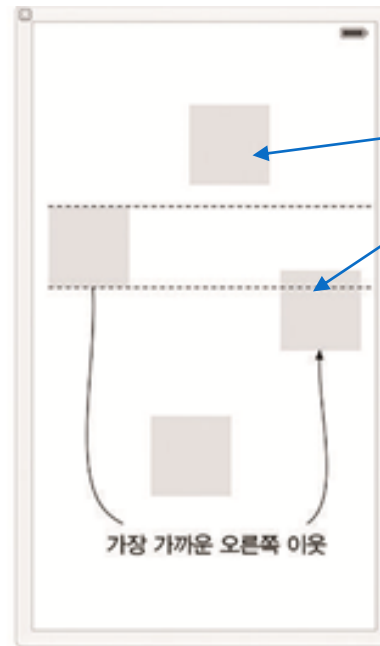
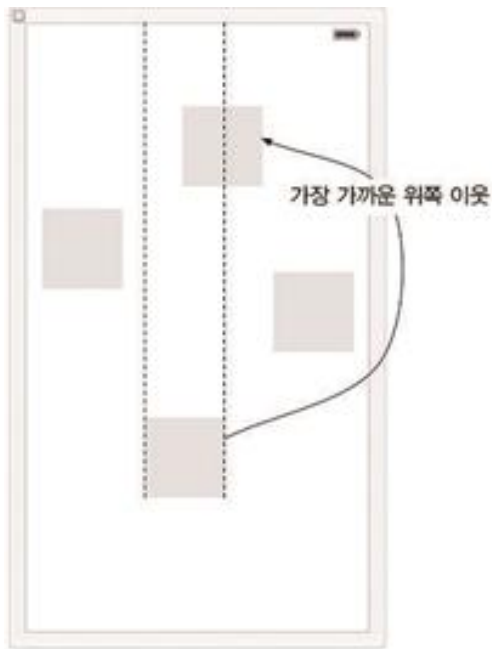




오토레이아웃 개념

■ 오토레이아웃의 개념

- View의 크기(너비, 높이)와 이웃 View와 간격으로 Satisfiable Layout를 이루는 것
- 다른 이웃: 가장 가까운 이웃(Default)



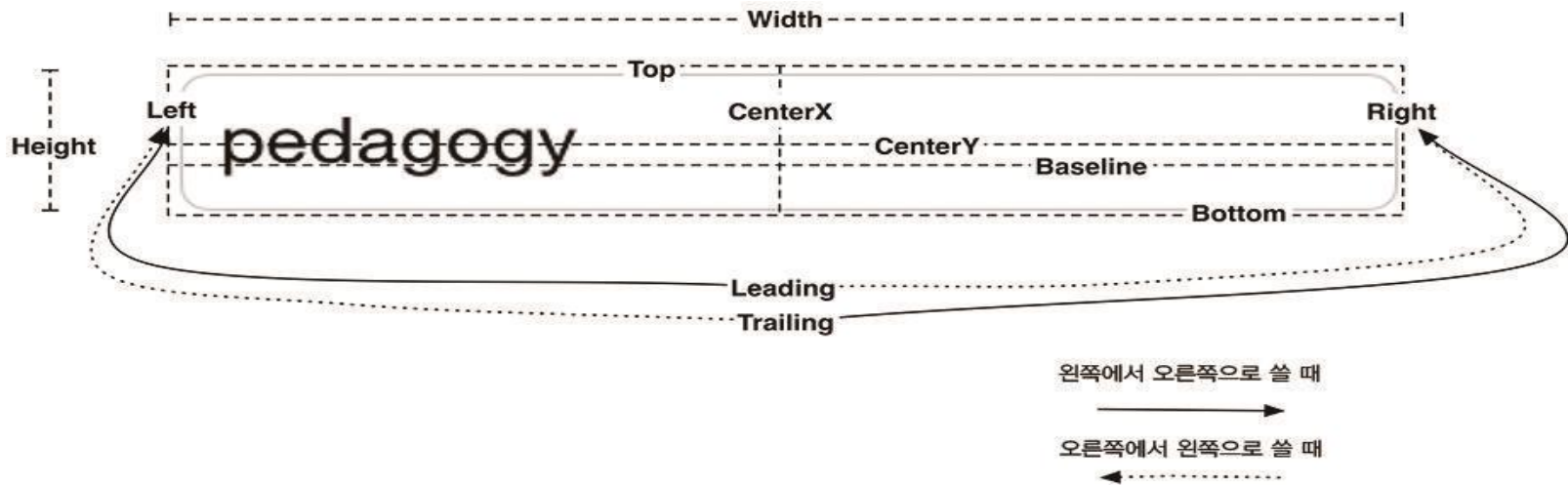
이것의 위쪽 가장 가까운 이웃은 부모 뷰이다

- 가까운 이웃이 없는 경우는 부모



오토레이아웃

■ 오토레이아웃의 정의에 필요한 속성



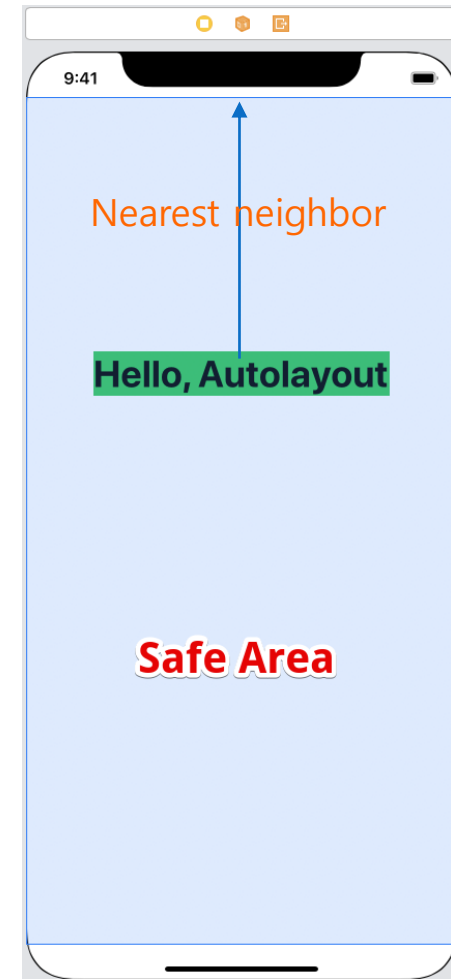
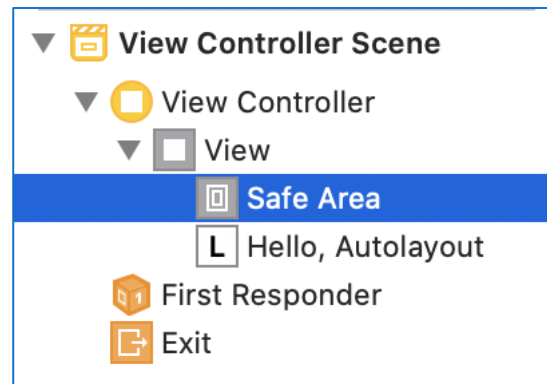
- leading, left, trailing, right
 - 중동국가에서는 leading = right, trailing = left
 - 그 외 국가에서는 leading = left, trailing = right



오토레이아웃

■ Safe Area

- iOS 7에서 애플은 어떤 뷰들(상태 바, 네비게이션 바, 탭 바 등) 이 있는 스크린 영역을 가리지 않도록 `topLayoutGuide`, `bottomLayoutGuide` 지정
- iOS 11에서 이 속성들을 금지 시켰고 `safe area` 를 도입
- Apple은 `safe area` 밖에 View를 배치하지 말 것을 권고





오토레이아웃

■ 여백 계산



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

Relationship: RedView.Leading = 1.0 x BlueView.trailing + 8.0

Attribute 2: BlueView.trailing

이런 것을 **Constraint**라 함
코드에서 runtime에 생성할수도 있음

```
imageView.width = 1.5 * imageView.height + 0.0
```

```
NSLayoutConstraint(item: imageView  
  attribute: .Width  
  relatedBy: .Equal  
  toItem: imageView  
  attribute: .Height  
  multiplier: 1.5  
  constant: 0.0)
```



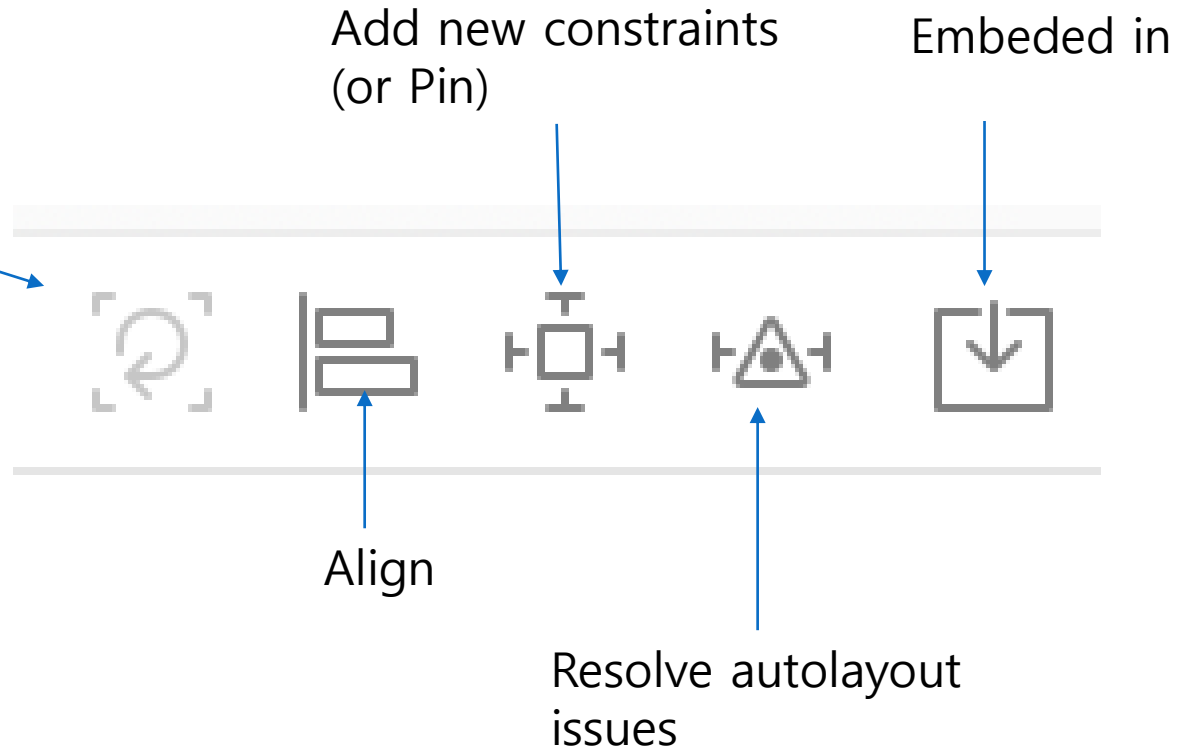
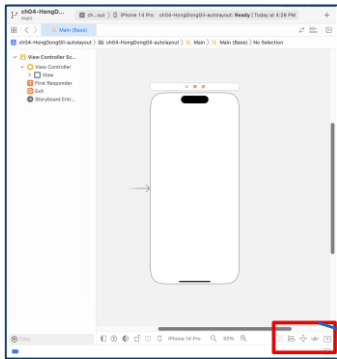
오토레이아웃

- 새 프로젝트 생성
 - 프로젝트명: Autolayout



오토레이아웃 지원 도구

- Main.storyboard를 오픈하라
 - 에디터 창의 우측하단에 다음과 같은 오토레이아웃 툴을 제공

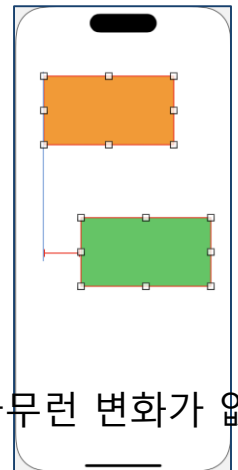
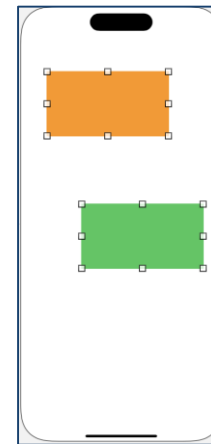
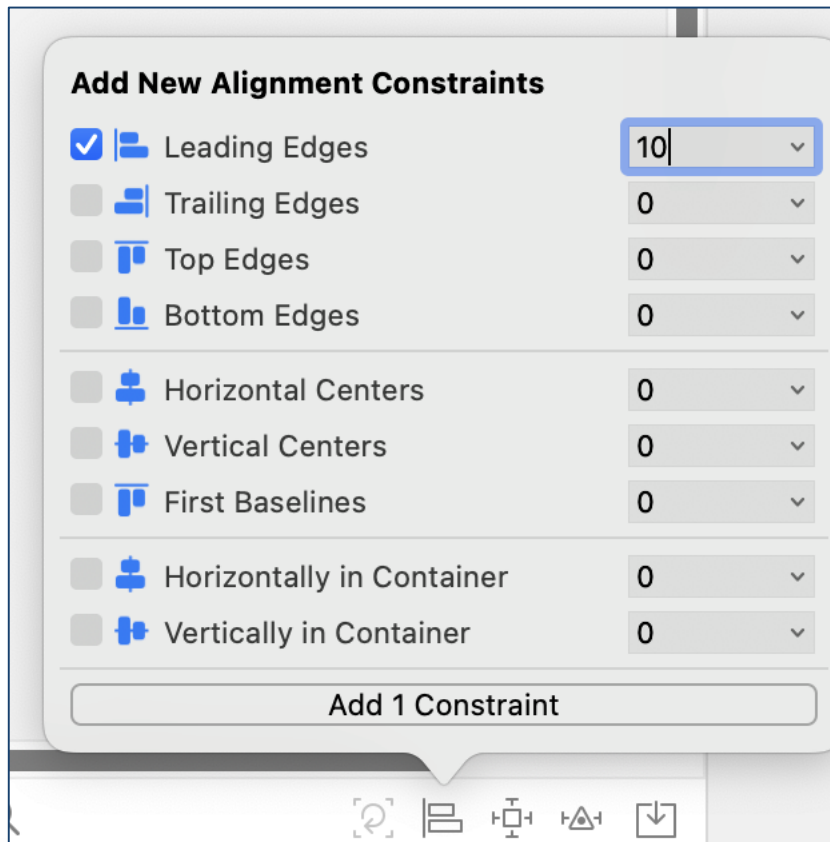




Align Tool

■ Align Tool

- Align Tool은 뷰들을 정렬 시키는데 사용
- 원하는 뷰들을 선택하고 Align Tool을 클릭
- 아래와 같은 팝업에서 적절한 제약조건을 선택 후 더함



아무런 변화가 없다

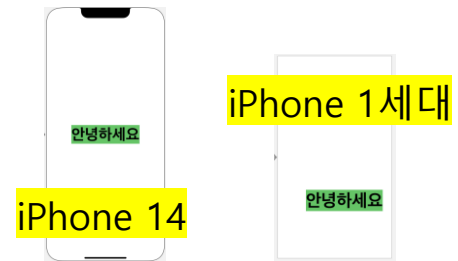
오토레이아웃은 View가 Satisfiable Layout가 아니면 적용되지 않을 뿐더러 붉은색 라인을 보이면서 제한 조건이 부족함을 보인다.



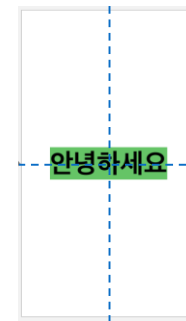
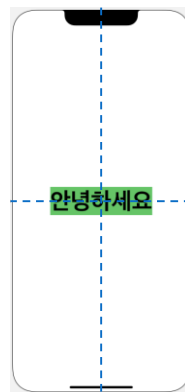
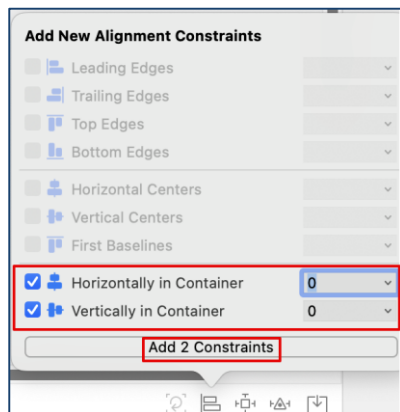
Align Tool

■ UILabel객체 화면 중앙에 배치하기

- 오브젝트 라이브러리에서 UILabel를 끌어다 Viewcontroller에 놓아라
 - 텍스트를 "안녕하세요"로 하고 (배경색:그린색, 폰트크기: 50)으로 하라
- 절대레이아웃 적용
 - 화면의 가이드라인에 따라 정중앙에 오도록 배치하라
 - 아직 Align Tool을 적용하지 마라
 - 각기 다른 두 아이폰에서 실행하여 보라
- 오토레이아웃 적용
 - UIView를 선택한 상태에서 Align Tool을 클릭하라.
 - 다음과 같이 선택하고 각기 다른 두 아이폰에서 실행하여 보라



절대 레이아웃이 적용되었기 때문에 화면이 회전하거나 크기가 변경되면 위치가 정중앙에 있지 않기 때문





Align Tool

■ 앞의 오토레이아웃 결과

The screenshot shows the Xcode Interface Builder for a storyboard named 'Main (Base)'. In the Document Outline on the left, the hierarchy is: View Controller Scene > View Controller > View > Safe Area > 안녕하세요. The 'Constraints' section for '안녕하세요' is expanded, showing two constraints: '안녕하세요.centerX = centerX' (highlighted with a red box) and '안녕하세요.centerY = centerY'. A red arrow points from this constraint to the 'Center X Alignment Constraint' details pane on the right. The details pane shows: First Item: 안녕하세요.Center X, Relation: Equal, Second Item: Superview.Center X, Constant: 0, Priority: 1000, Multiplier: 1, Identifier: Identifier, Placeholder: Remove at build time (unchecked), and Installed: checked (checked). A blue arrow points from the 'Multiplier' field to the text 'y = ax + b' below.

constraint가 적용될때마다 Document Outline에 객체가 하나씩생김 현재 2개가 적용되어 2개가 있음

// View객체를 outlet 변수로 만듦과 같이 // constraint도 outlet 변수로 만들수 있음

Document Outline에서 선택된 constraint가 굵게 표시됨

Document Outline에서 선택된 constraint에 대하여 상세 내역이 나옴

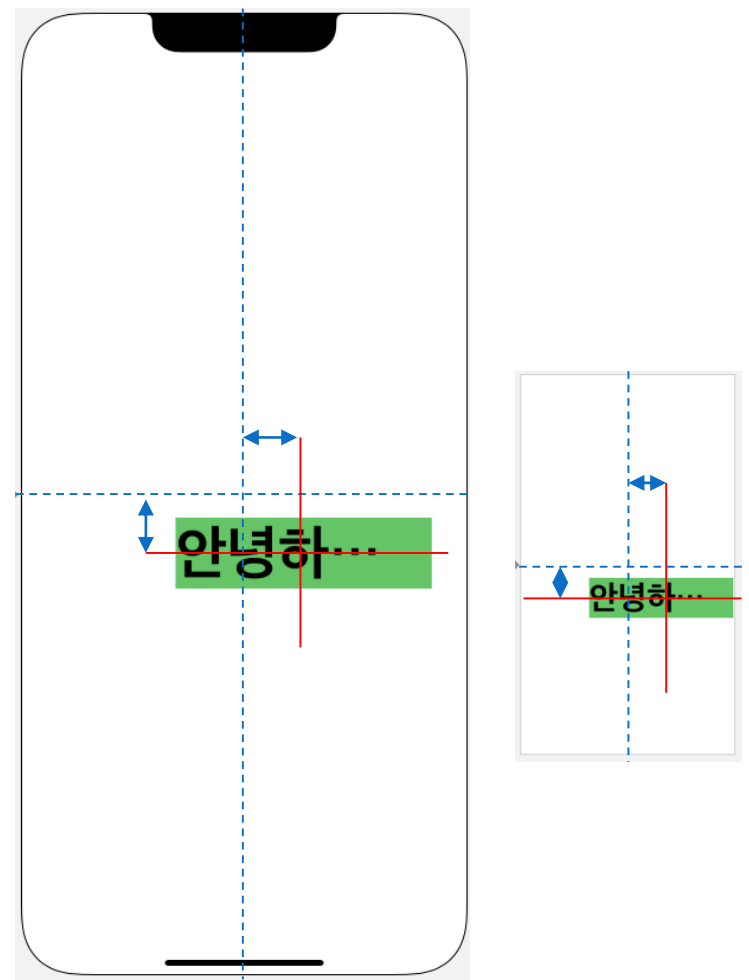
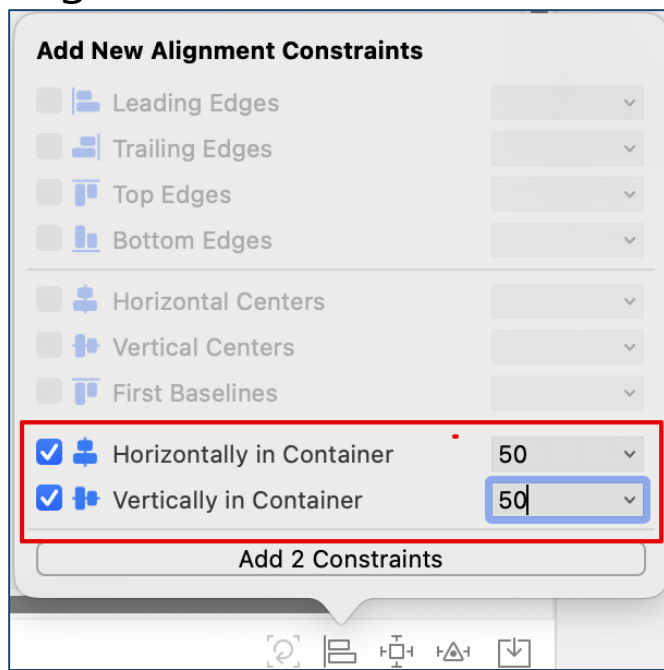
$y = ax + b$

y: first item
a: multiplier
x: second item
b: constant
// 여기서 y, a, x, b를
// 수정할 수 있음



예제

- UILabel객체를 화면 중앙에서 우로 10, 아래로 10되는 지점으로 배치하기
 - 오브젝트 라이브러리에서 UILabel를 끌어다 Viewcontroller에 놓아라
 - 텍스트를 "안녕하세요"로 하고 (배경색:그린색, 폰트크기: 50)으로 하라
 - Align Tool에서 아래와 같이 설정하라

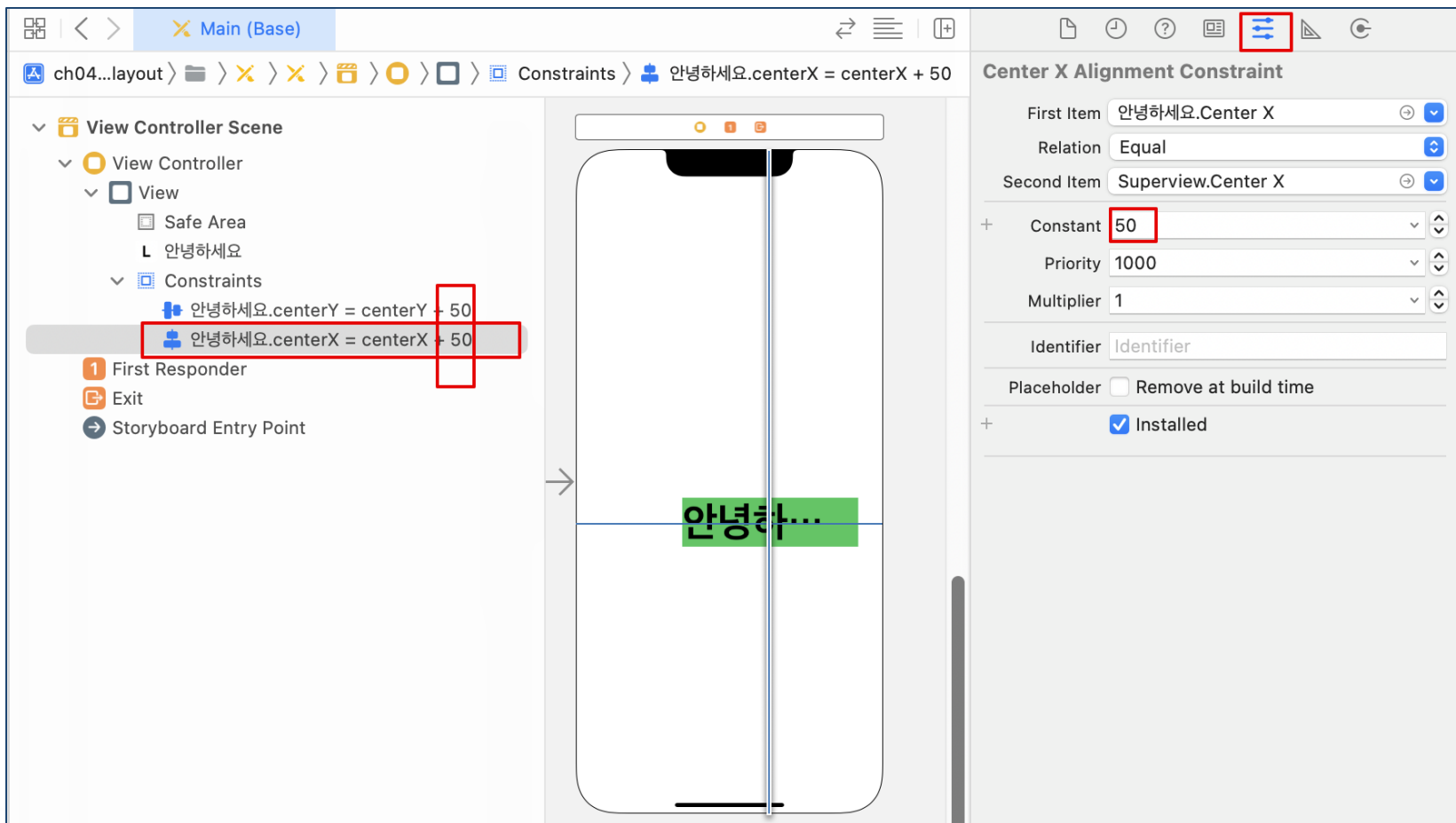


붉은색 라인이 생기는 경우는 앞에 적용된 constraints에 다시 상충되는 constraints를 덧붙였기 때문 => 앞의 constraints를 지운후 다시 적용하면됨
다음 슬라이드 참조



■ 앞의 오토레이아웃 결과

- 앞에서 설정한 50이라는 값이 어디 나타나는지 확인하라
- 오른쪽 속성창에서 50대신에 100을 입력하여보라





Resolve Auto Layout Issues Tool

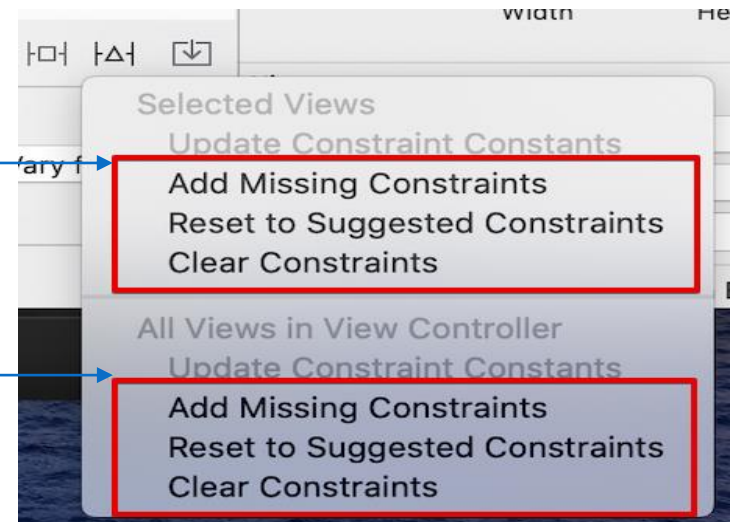
■ Resolve Auto Layout Issues Tool



- 일반적인 자동 레이아웃 문제를 해결하기 위한 여러 가지 옵션을 제공
- 상단은 현재 선택된 뷰, 아래쪽 옵션은 모든 뷰에 영향
- "Update Constraint Constants"는 현재의 Constraint에서 UIView의 이동 후 이동값에 대한 상수를 적용하고자 하는 경우
- "Add Missing Constraints"는 추가적으로 제약 조건들을 자동으로 생성
- "Reset to Suggested Constraints"는 현재 적용된 제약 조건들을 지우고 필요한 제약 조건들을 자동으로 생성하는 방법
- "Clear Constraints"는 해당 제약조건을 모두 지움

선택된 뷰

전체 뷰





예제

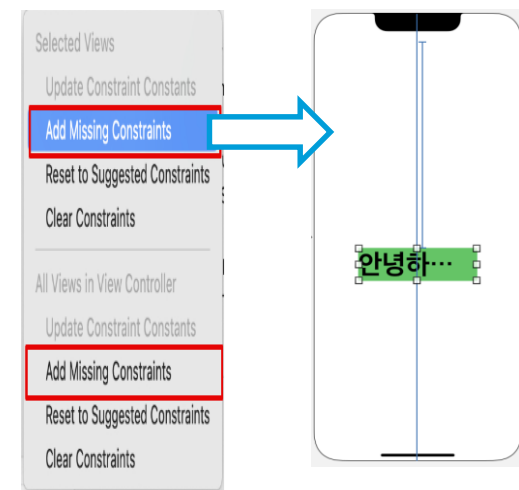
- 앞의 예제에서 현재 모든 제약조건 지우기 및 제약조건 부분 적용후 자동완성하기
 - "Resolve Auto Layout Issues Tool"의 "clear constraints" 이용하여 모든 제약조건을 지워라
 - Align Tool을 이용하여 UILabel을 수평중앙에 오도록만 하는 constraint를 적용하라
 - "Resolve Auto Layout Issues Tool"의 "Add missing constraints " 이용하여 필요한 제약조건을 지워라



UILabel을 선택해도 어떤 라인이 없다는 것은 오토레이아웃이 전혀 적용되지 않았음을 의미함



UILabel을 선택하면 수직중앙라인 파란색이라는 것은 x, width는 알수 있다는 의미 UILabel의 상하부분에 붉은 라인이 있다는 것은 y를 알수 없다는 의미 이것은 추가적인 constraint가 필요함을 의미함



UILabel을 선택하면 새로운 파란 수직라인이 추가됨 이것은 UILabel의 y에 대한 constraint임 더이상 붉은 라인이 없으면 오토레이아웃이 완성되었다는 의미임(물론 우리가 것은 아니지만)



Add new constraints

■ Add new constraints(Pin Tool)

- Pin tool은 뷰의 상대적 위치와 크기에 대한 제약 조건을 적용
- 원하는 뷰를 선택하고 Pin Tool을 클릭
- 팝업 뷰에서 다양한 여백 및 크기 선택

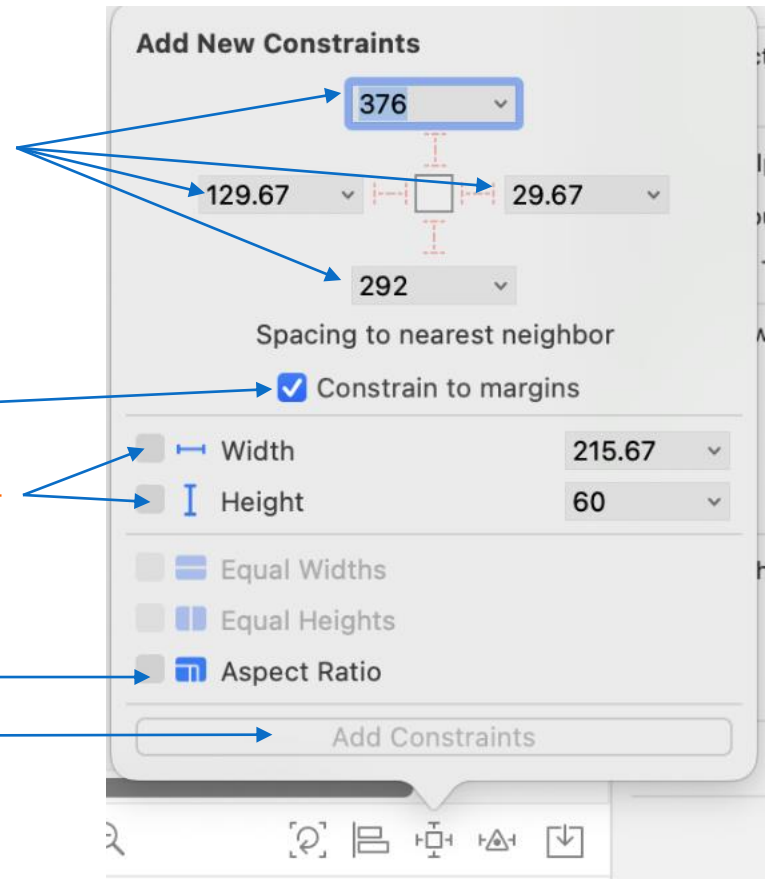
상하좌우 마진(여백)
현재의 여백을 가장가까운 이웃에 대한
값으로 보이고 있음
점선을 클릭하여 실선으로 변경해야 적용됨

부모뷰의 마진(여백)적용
부모뷰가 마진을 가진다면
이를 반영한후 여백 적용
* 부모의 여백은 크기 인스
펙트에서 설정할 수 있음

너비/높이 크기
현재의 크기를 표시하고 있음
체크를 하여야 적용됨

가로/세로 비율반영
가로/세로 비율반영시

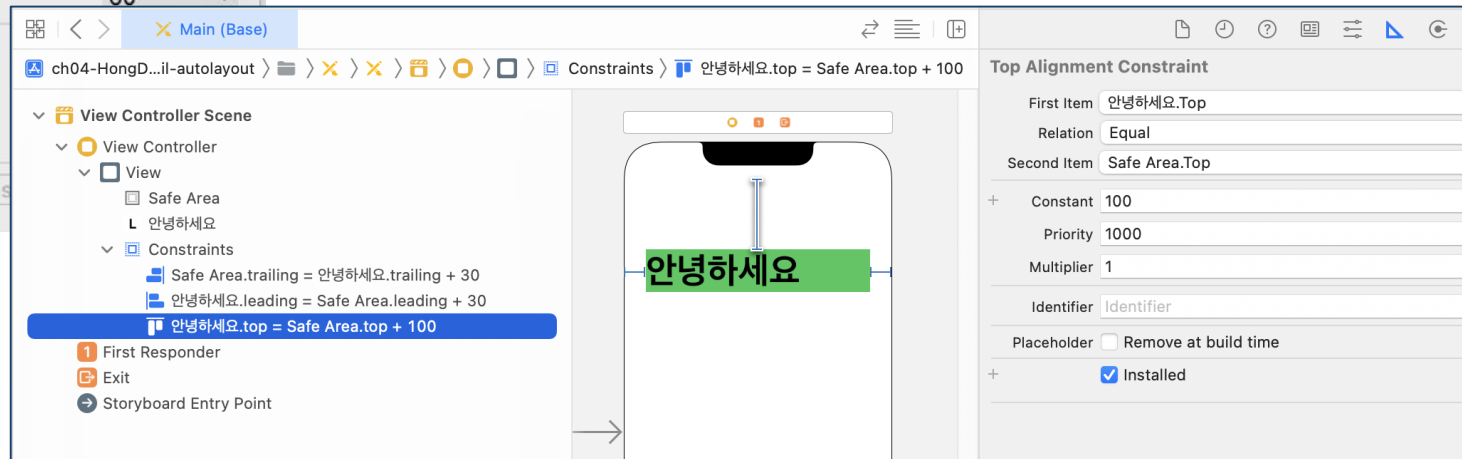
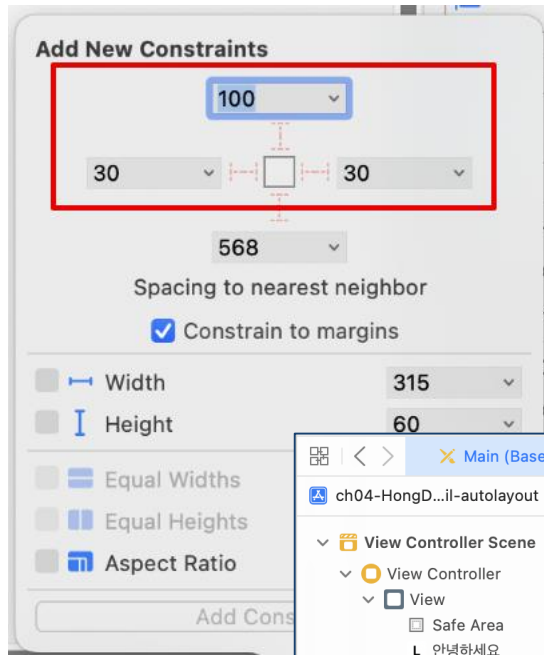
최종적으로
클릭해야 반영됨





Add new constraints

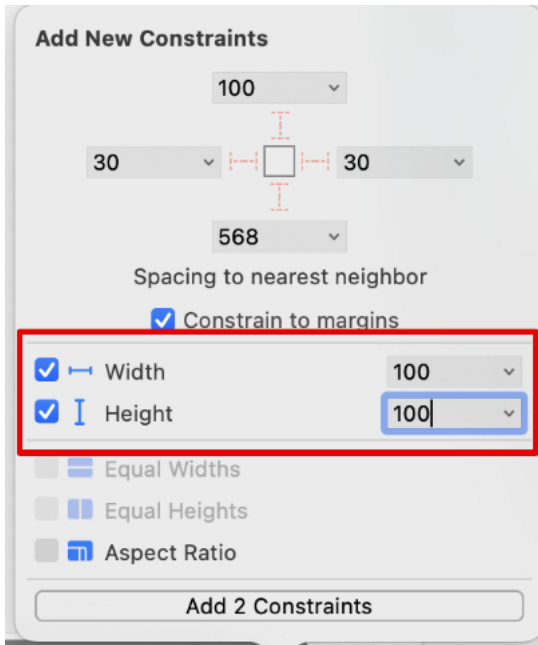
- UILabel객체 상단여백, 왼쪽여백, 오른쪽여백을 각각 100, 30, 30으로 적용하기
 - 현재 "안녕하세요" UILabel의 모든 constraint를 지워라
 - Pin tool을 선택하고 상단여백, 왼쪽여백, 오른쪽여백을 아래와 같이 적용하라.





예제

- 앞의 오토레이아웃에서 UILabel의 너비와 높이를 100, 100으로 하라
 - Pin Tool에서 다음과 같이 적용해보자





상대적 콘텐츠 크기 제어

■ 콘텐츠(뷰 객체) 크기

- 뷰 객체의 너비와 높이
 - 원칙적으로는 Constraint를 적용하여 크기를 설정하여야 함
 - 일부 뷰 객체는 디폴트로 크기가 정해지는 것도 있음

■ 고유 콘텐츠 크기 (intrinsic content size)

- 뷰 객체가 표시할 내용을 기반으로 뷰의 크기가 어느 정도될 지에 대한 정보
 - 라벨의 고유 콘텐츠 크기는 표시할 텍스트 크기에 기반을 둔다.
 - 이미지 뷰의 고유 콘텐츠 크기는 직접 선택한 이미지 크기이다.

■ 상대적 콘텐츠 크기

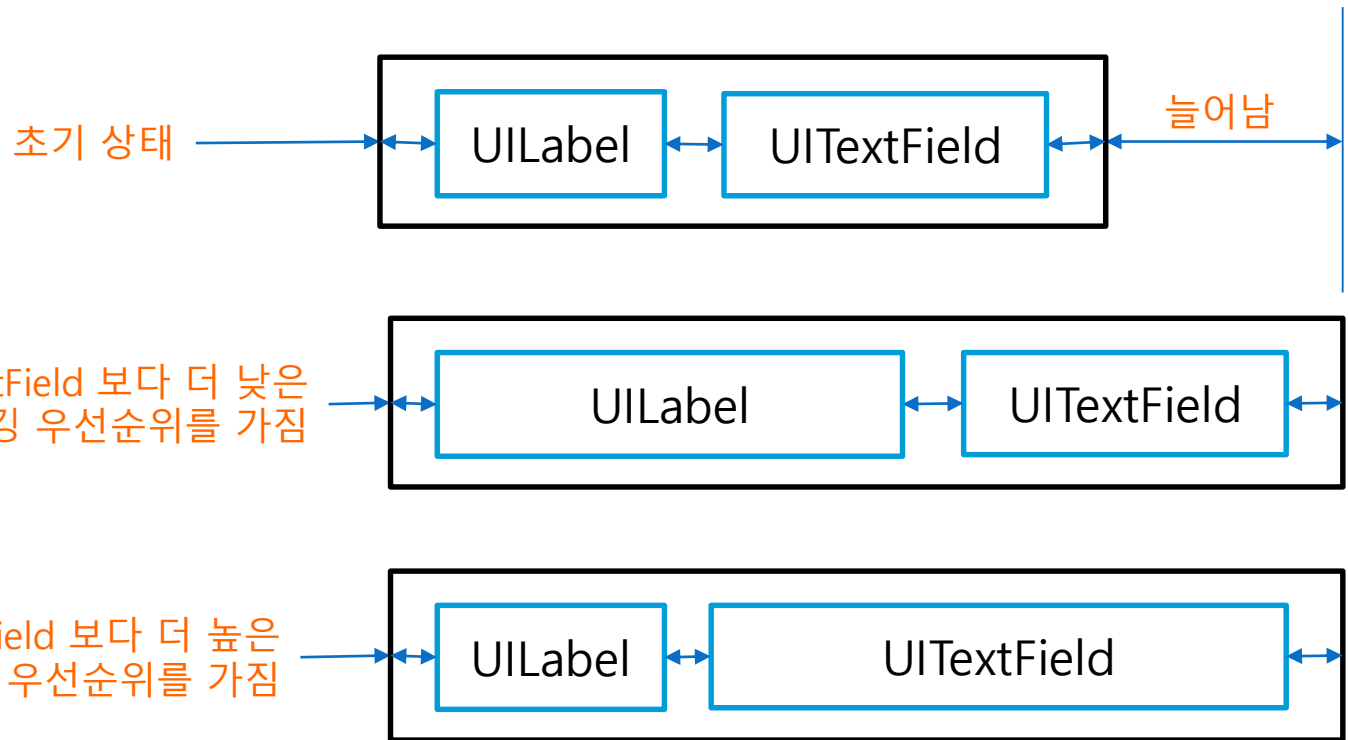
- 여러 개의 뷰 객체가 동시에 크기가 변경되어야 하는 경우 콘텐츠의 크기
- 각 뷰 객체는 크기 변경에 대한 우선 순위가 있어 이에 따라 크기가 변경됨



콘텐츠 크기의 우선 순위

■ 허깅 우선순위(hugging priority)

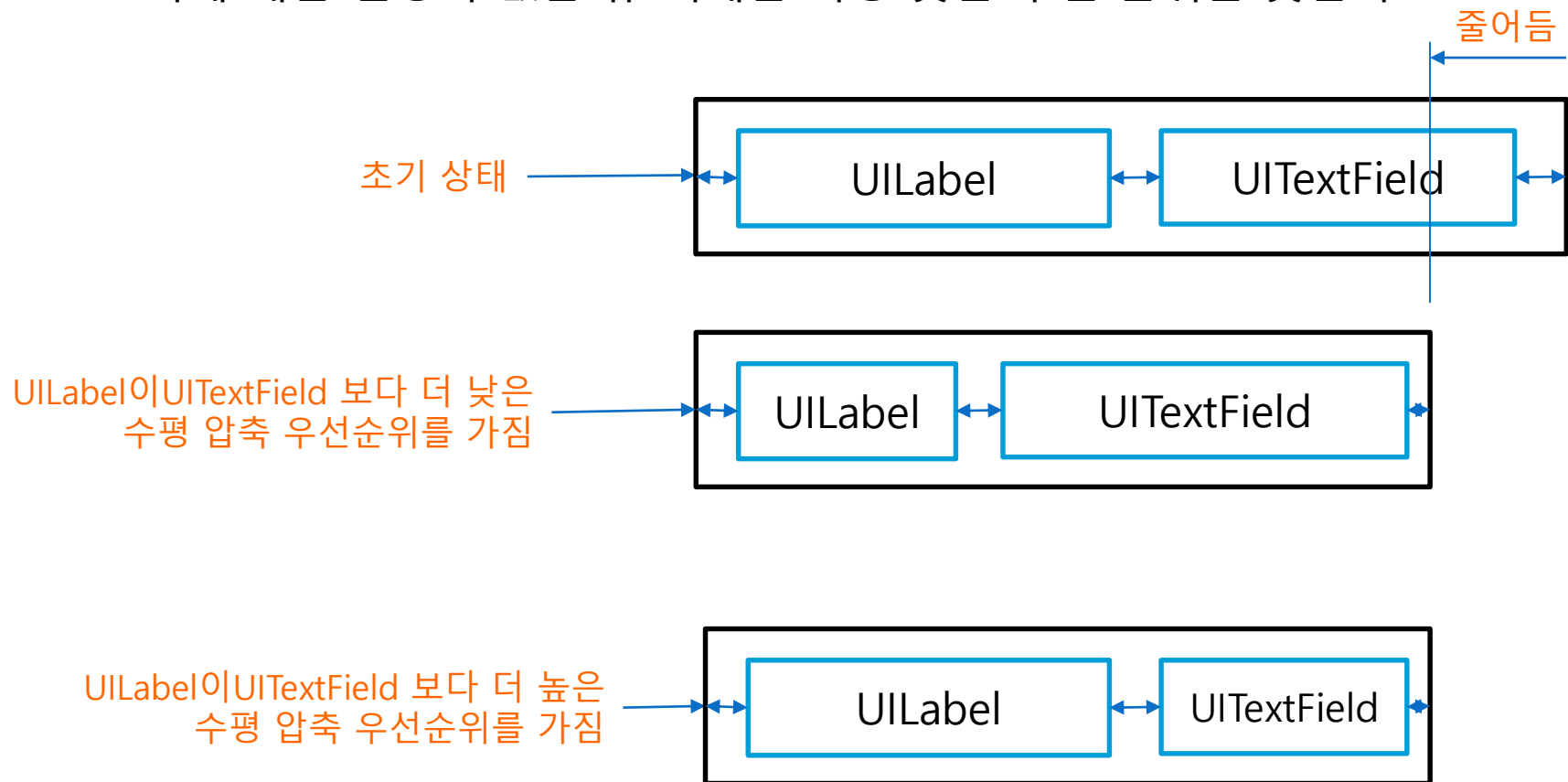
- 늘어나기 싫어하는 우선순위 (낮으면 잘 늘어난다, 고무밴드와 같다)
- 크기에 대한 설정이 없는 뷰 객체는 가장 낮은 우선 순위를 갖는다





콘텐츠 크기의 우선 순위

- 압축저항 우선순위(compression resistance priority)
 - 줄어들기 싫어하는 우선순위 (낮으면 잘 줄어든다, 스프링과 같다)
 - 크기에 대한 설정이 없는 뷰 객체는 가장 낮은 우선 순위를 갖는다

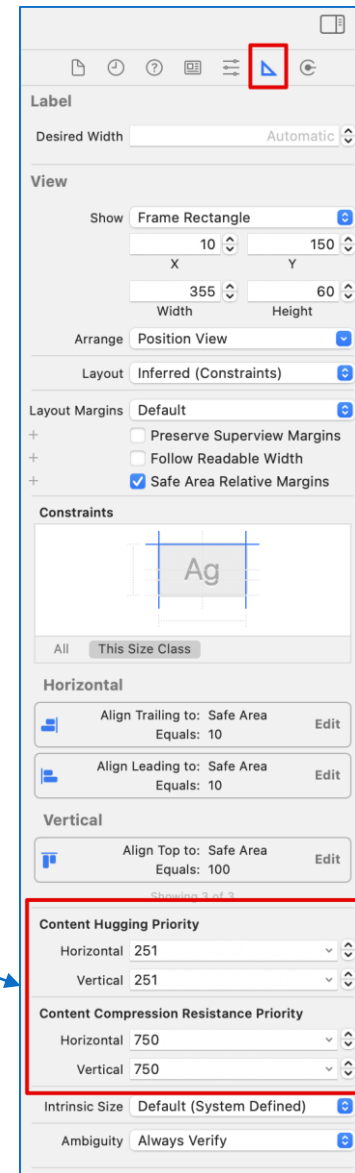
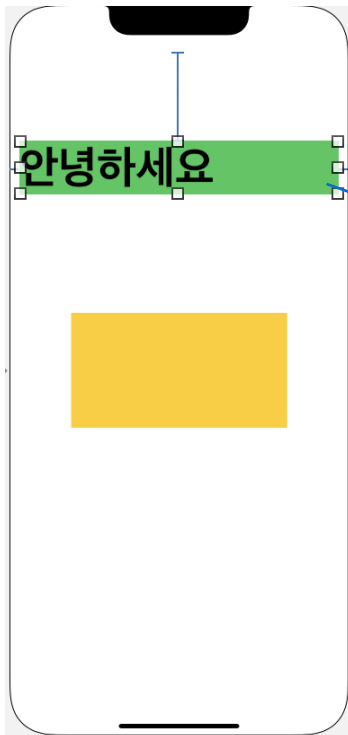




우선순위 적용 방법

■ 크기가 설정된 경우

- UILabel(안녕하세요)를 선택한다.
- 우측상단에 Size inspector를 선택한다.
- 하단에 오른쪽 그림과 같이 두가지 우선순위를 적용할 수 있다





우선순위 적용 방법

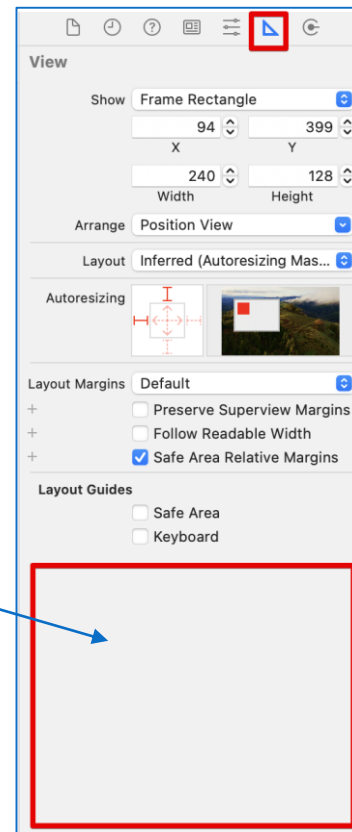
■ 크기가 설정되지 않은 경우

- UIView(아래 노란 박스, 크기 설정이 안된 상태)를 선택한다.
- 우측상단에 Size inspector를 선택한다.
- 우선순위에 대한 내용이 나타나지 않는다.



크기 설정이 안된 상태

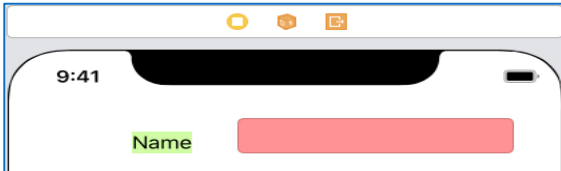
이 상태에서 pin 툴을 이용하여 크기를 설정하면 우선순위 패널이 나타난다





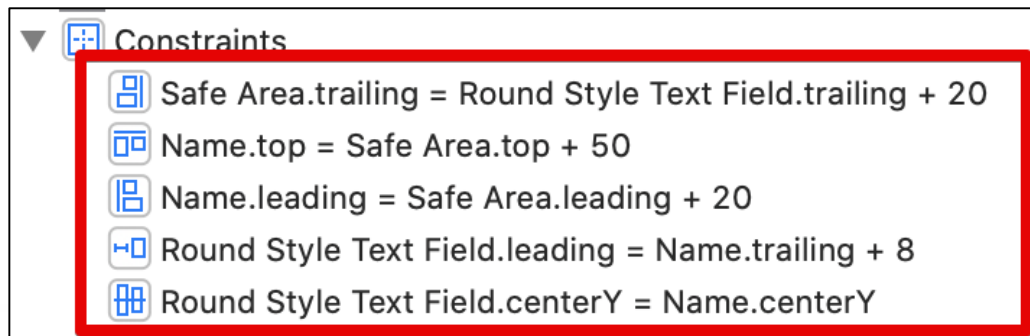
허깅 우선순위 예제

- Name Label(배경색: 그림)과 TextField(배경색: 주황)로 UI를 구성



- 다음과 같은 제약 조건을 적용하라

- Name에 대한 제약조건
 - 왼쪽여백: 20, 오른쪽여백: 8 상단여백:50
- TextField에 대한 제약조건
 - 왼쪽여백: Name의 오른쪽에 있으므로 이미 적용됨
 - 오른쪽여백:20
 - 상단여백: Name의 수평 중앙이 일치하도록
 - Name와 TextField를 동시에 선택후 Align Tool에서 "Vertical Centers" 선택





허깅 우선순위 예제

■ 실행

- Name는 고유 콘텐츠 크기를 유지하나 UITextField는 늘어났다.
 - Name의 수평 허깅 우선순위는 251이나 UITextField는 248이다.
 - 이러한 이유로 UITextField가 늘어났다.



■ 테스트

- UITextField는 260으로 변경하여 보라
 - 또는 Name의 수평 허깅우선순위를 247로 해도 된다

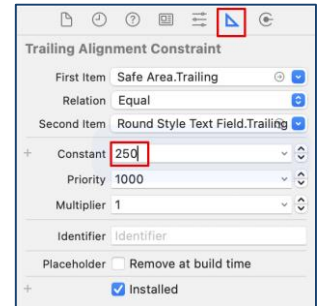




압축저항 우선순위 예제

■ 앞의 예제에서 UITextField의 Trailing 여백을 20에서 250으로 변경

- Document Outline에서 "Safe Area.trailing = Round Style Text Field.trailing + 20"를 선택
- Inspector의 속성창에서 "20" -> "250"으로 수정
 - Document Outline에서 20->250으로 수정하면 안된다.



■ 실행

- UITextField가 줄어들고, 키인을 많이 하여도 스크롤이 일어날뿐 크기는 늘어나지 않는다.
- 이유
 - Name의 압축 우선순위는 750이다.
 - UITextField의 압축 우선순위는 750이다.
 - 둘 다 동일한 경우 iOS가 임의적으로 한다
 - 따라서 UITextField의 압축 우선순위는 749로 변경하는 것과 동일하다(이렇게 UITextField가 줄어들기를 원한다면)

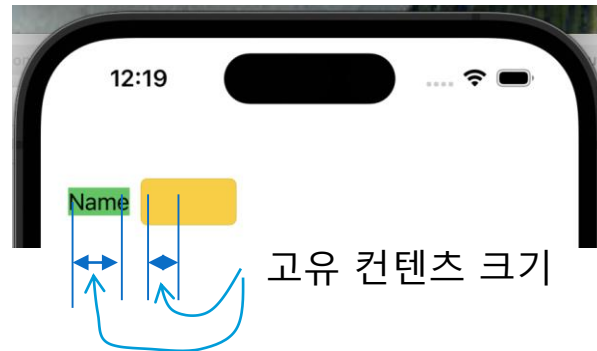




압축저항 우선순위 예제

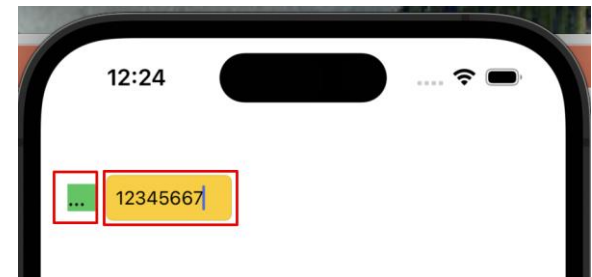
■ 테스트

- Name의 압축 우선순위는 740로 해보라.
- 또는 UITextField의 압축 우선순위는 760으로 해보라
- 앞과 큰 변화가 없다.
 - 왜냐하면 Name은 고유 콘텐츠 크기 만큼 설정되었고, UITextField는 현재 입력된 글자가 없으므로 실제로 고유 콘텐츠 너비가 최소여서 오히려 늘어난 상태이다.



- 데이터 입력

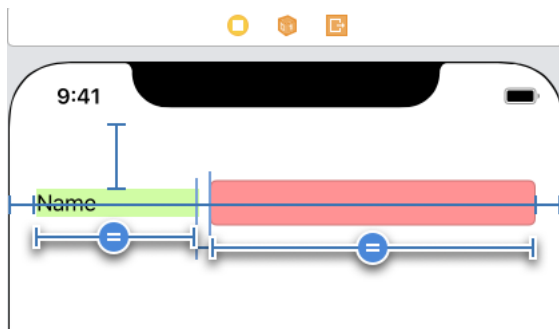
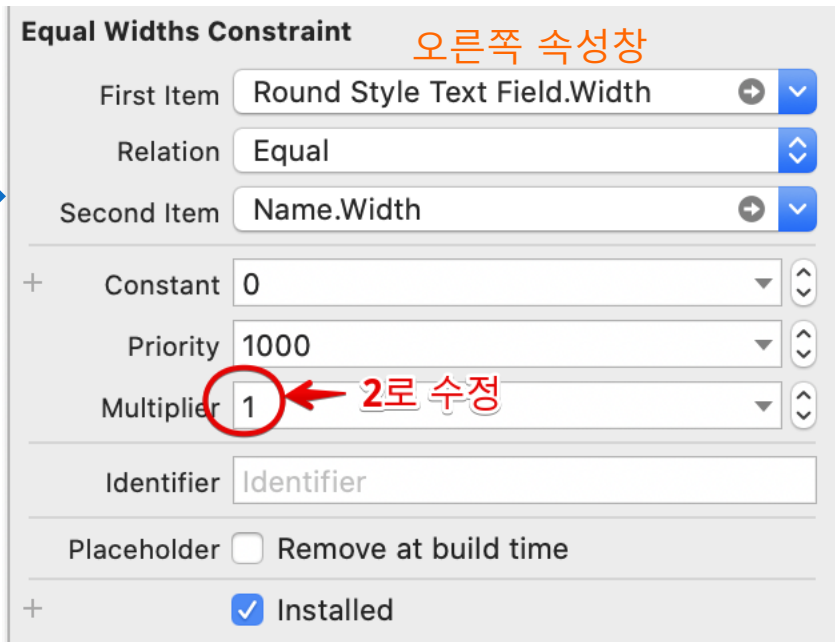
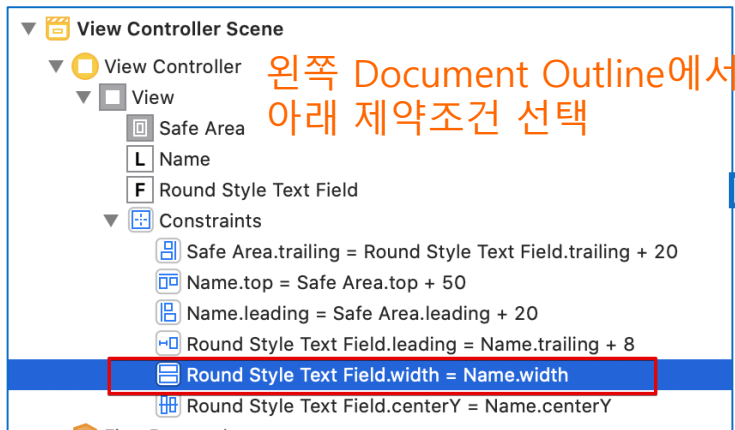
- UITextField에서 12345678을 입력해보라
- Name이 줄어들고 UITextField가 늘어남을 알 수 있다





우선순위 무시

- Name와 TextField의 크기를 1:2로 분배
 - Name, TextField를 동시에 선택
 - Pin Tool에서 "Equal Width" 선택

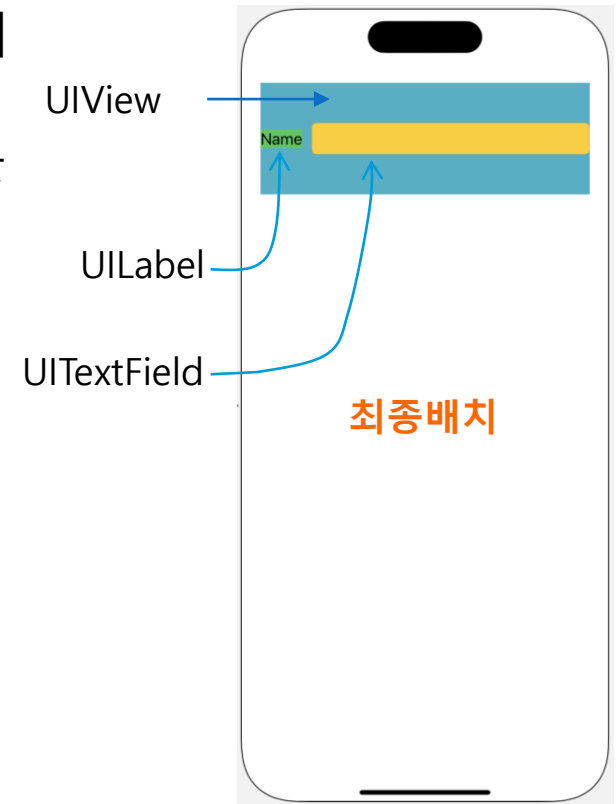
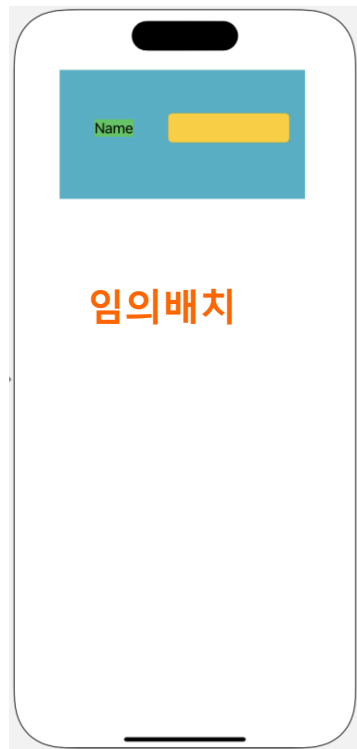




예제

■ 우측과 같은 UI를 만들어라

- UIView, UILabel, UITextField를 맨아래 그림과 같이 놓아라
- 임의로 컨트롤을 배치하되 가능하면 오토레이아웃이 최종적으로 적용되는 것과 비슷하게 놓아라





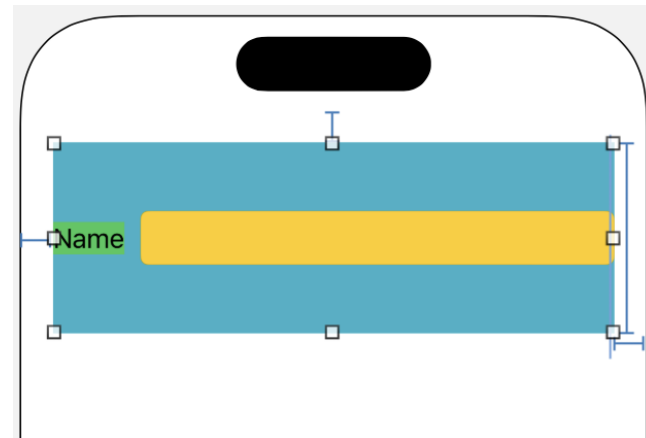
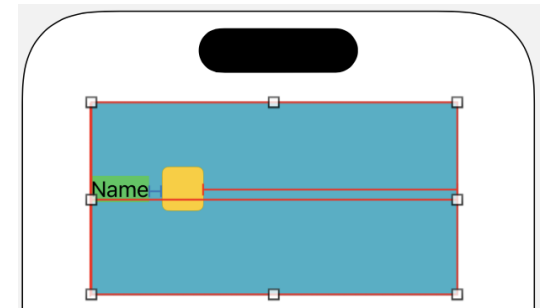
예제

■ 다음과 같은 제약조건 적용하라

- (1) UIView: 상하좌우 여백 20, 20, 20, 높이 120
- (2) Name: 좌여백 0, 우 여백 10, 수평중앙->UIView의 중앙에 정렬
- (3) UITextField: 좌여백 10, 우여백 0, 수평중앙->UIView의 중앙에 정렬

■ 방법1(포함된 자식들부터 제약조건 적용)

- 제약조건 (2), (3)을 순서적으로 적용하라
 - 우측 그림과 같이 대부분의 constraints가 붉은 색이다.
- 제약조건 (1)을 적용하라
 - 아래와 같이 오토레이아웃이 완성된다.

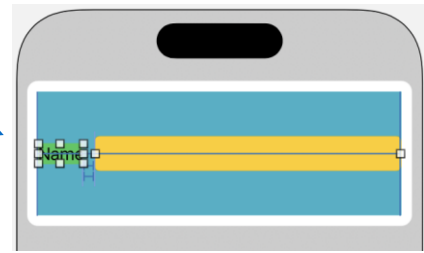
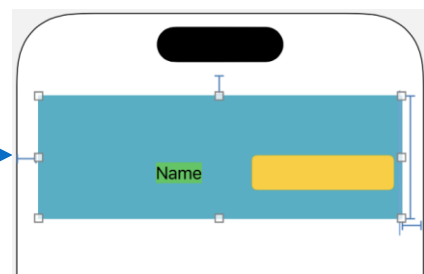




예제

■ 방법2(포함하는 부모부터 제약조건 적용)

- 모든 제약조건을 clear하라
- 그리고 컨트롤의 배치도 처음처럼 하라(임의배치)
- 제약조건 (1)을 적용하라
 - 붉은 라인이 없다
- 제약조건 (2), (3)을 순서대로 적용하라



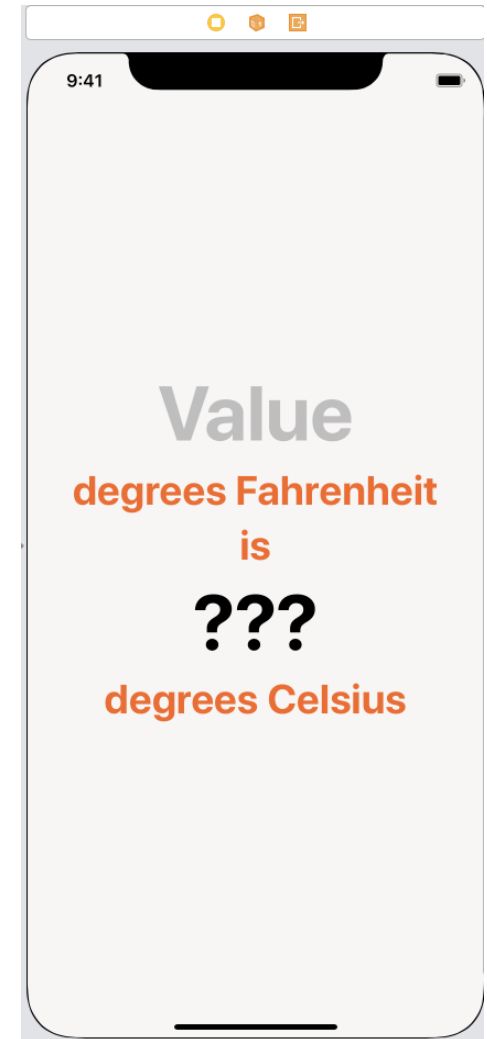
■ 방법1, 방법2

- 최종 결과는 동일하다
- 그러나 방법1은 중간과정에 이해하기 어려운 오류가 발생한다
- 결론
 - 포함관계에 있는 오토레이아웃은 포함하는 부모부터 먼저 오토레이아웃을 적용하는 것이 이해하기 쉽다.



ConversionViewController 개선

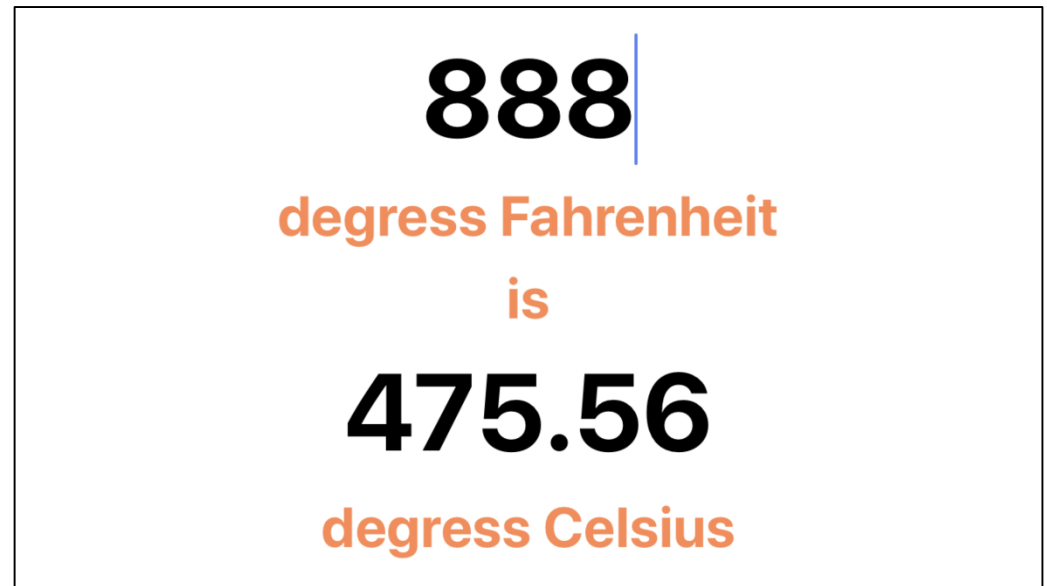
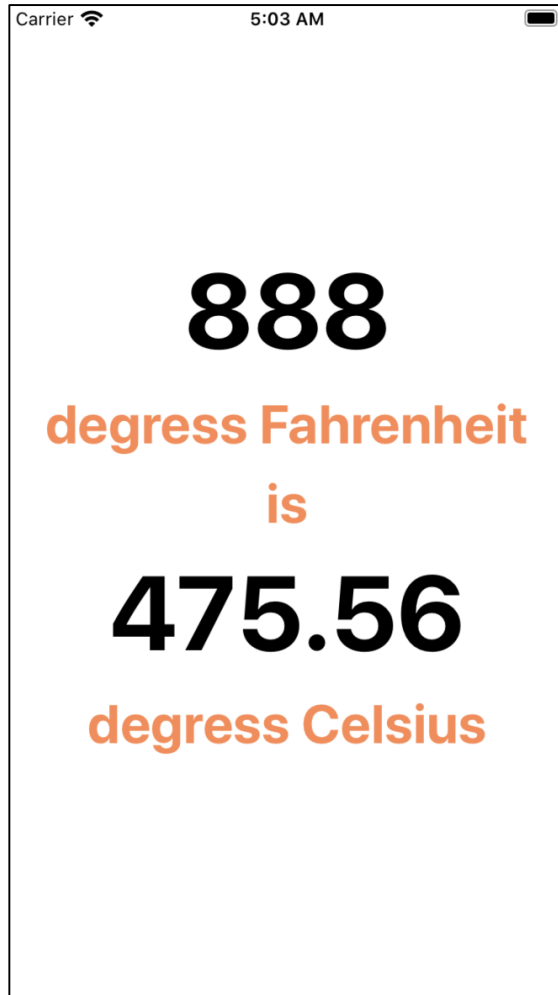
- 3장에 오로레이아웃 적용
- 오토레이아웃 적용
 - 5개의 모든 컨트롤을 선택하라
 - Align Tool을 클릭하여 "Horizontally in Container"를 선택하라(Center Align)
 - Text Field를 제외한 4개의 Label을 선택하라
 - Pin Tool에서 상단 여백을 5로하라(컨트롤 사이의 수직여백을 5로한다)
 - "is" Label을 선택하라
 - Align Tool을 이용하여 이를 "Vertically in Container"를 선택하라(이것은 'Is' Label을 수직적 중앙에 놓고 모두 분산되도록 한다)





ConversionViewController 개선

■ 실행





Auto Layout Without Constraints


■ StackView


- 스택 뷰는 복잡한 제약 조건을 도입하지 않고도 자동 레이아웃의 기능을 쉽게 활용할 수 있는 방법을 제공
- 안드로이드에서 LinearLayout과 매우 유사'


■ StackView의 주요 속성



- axis: (UIStackView only): 오리엔테이션 지정
- distribution: 자식 뷰들을 오리엔테이션 방향에서 따라 분배 방법 지정
- alignment: 오리엔테이션의 수직 방향에서 자식 뷰들의 위치 지정
- spacing: distribution에서 자식 뷰들 사이의 간격.

Stack View

+ Axis Vertical 

+ Alignment Fill 

+ Distribution Fill 

+ Spacing 8  

+ ☐ Baseline Relative

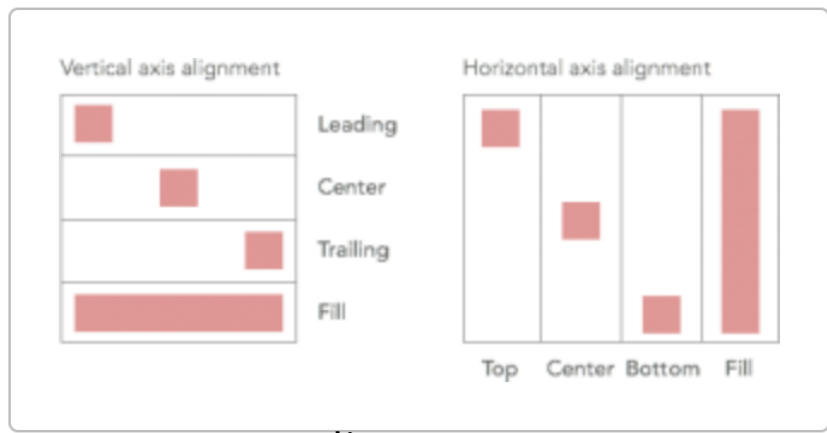


StackView

StackView 속성



axis



alignment




distribution



■ 사용해 보기

– UI 설정

- UILabel(초록색 배경)과 UITextField(파란색 배경)를 적절한 위치에 좌우가 되도록 갖다 놓아라
 - UILabel의 텍스트를 Login ID로 하라
- UILabel과 UITextField를 모두 선택한 상태에서 "메뉴->Editor->Embedded In->StackView를 선택하라."
 - Interface Builder의 오른쪽하단 ()를 클릭하여 StackView를 선택하여도 된다.
- StackView의 배경색을 그림과 같이 적당한 색(노란색)으로 변경하라

✓ View Controller Scene

✓ View Controller

✓ View

Safe Area

Stack View

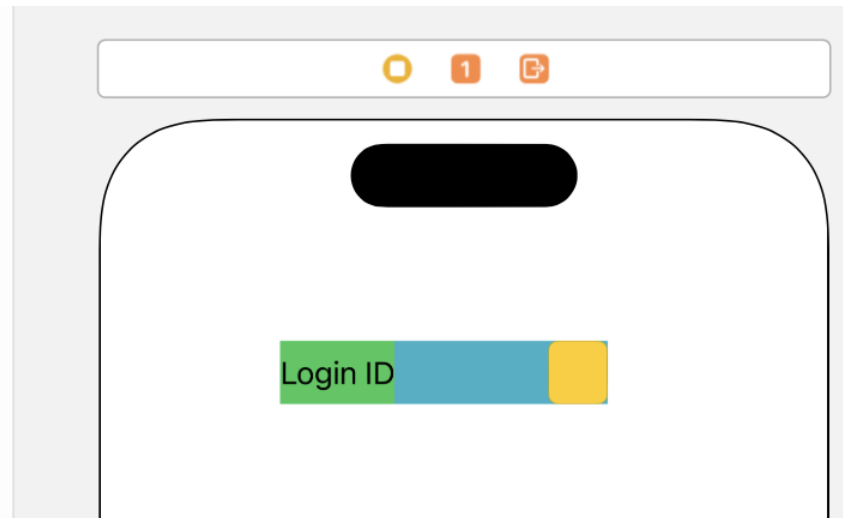
L Login ID

F Round Style Text Field

1 First Responder

Exit

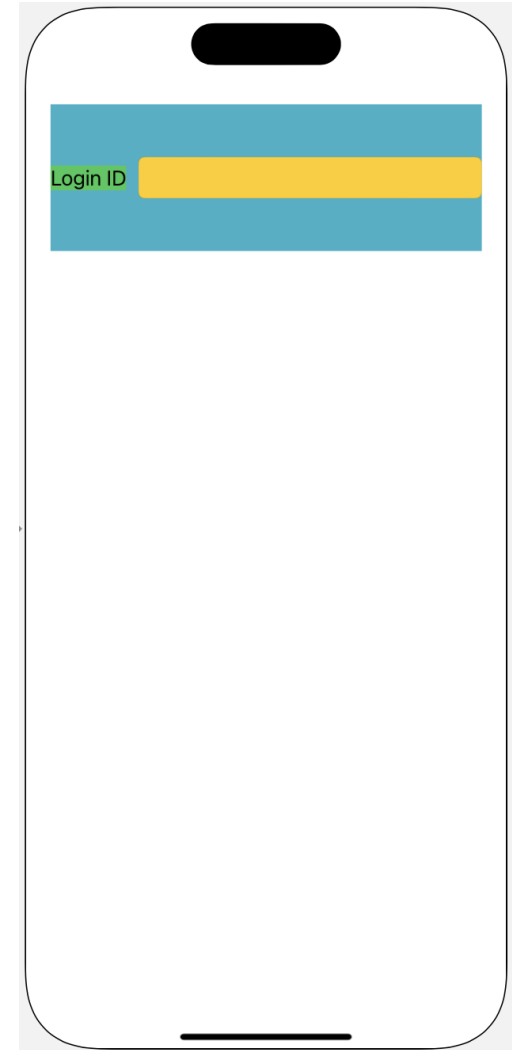
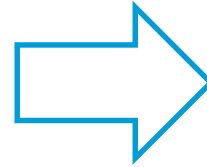
Storyboard Entry Point





StackView 설정

- constraints 설정
 - 상좌우 여백: 20, 높이: 120
- Alignment: Center
- Distribution: Fill
- Space: 10
- 앞의 예제에서 UIView를 사용하는 것보다 UIStackView를 사용하는 것이 훨씬 쉽다.
- 연습
 - UITextField의 허깅 우선순위를 260으로 수정해보라



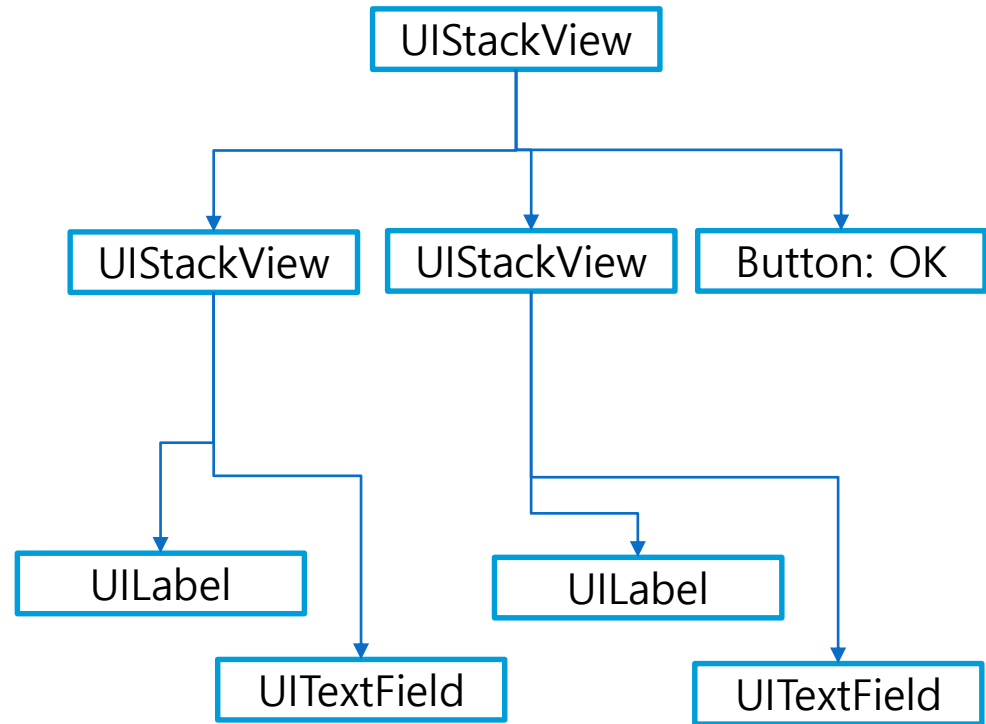


- 다음과 같은 Login 화면을 만들어보자

Login ID

Password

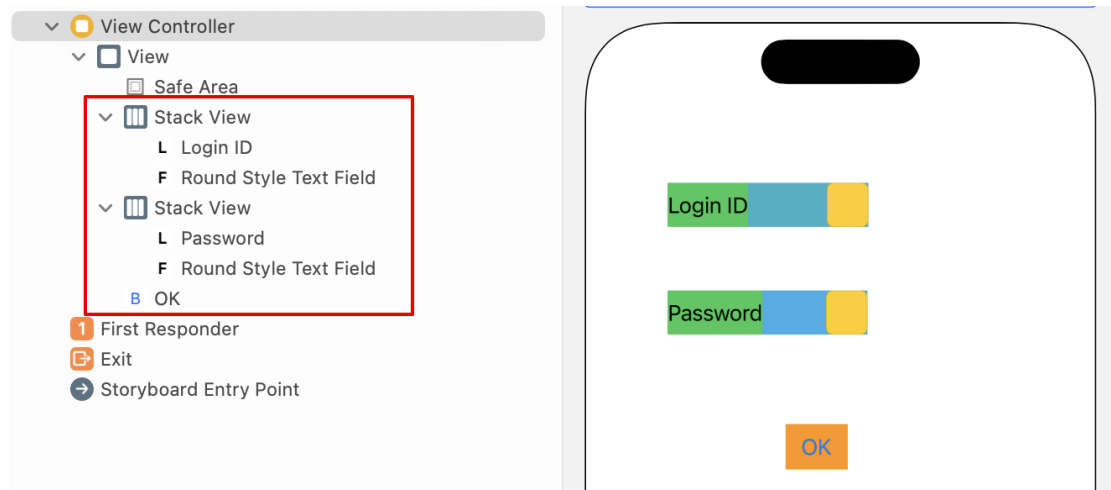
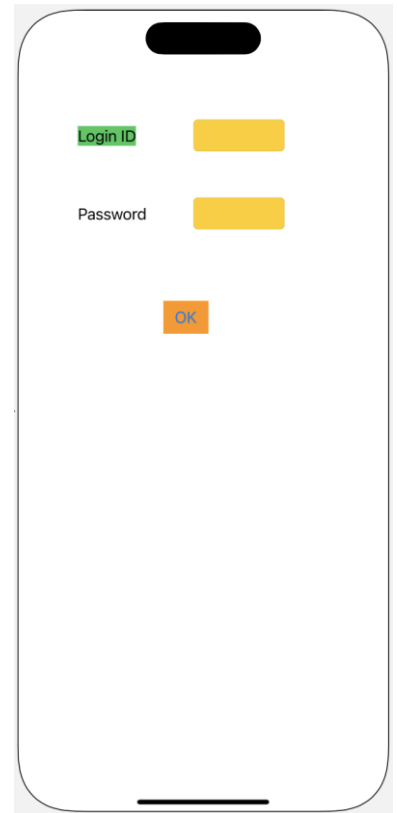
OK





예제

- 다음과 같이 UILabel(Login ID), UITextField, UILabel>Password), UITextField, UIButton(OK)를 적절한 위치에 놓아라
- StackView적용
 - Login ID, UITextField를 동시에 선택한 후 StackView를 적용하라. StackView 배경색을 변경하라
 - Password, UITextField를 동시에 선택한 후 StackView를 적용하라 . StackView 배경색을 변경하라





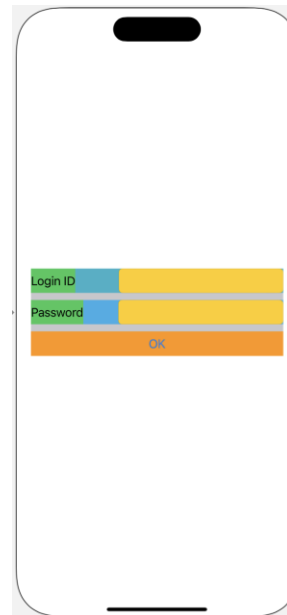
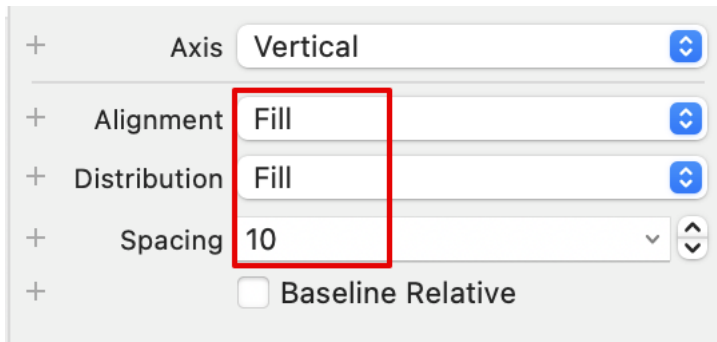
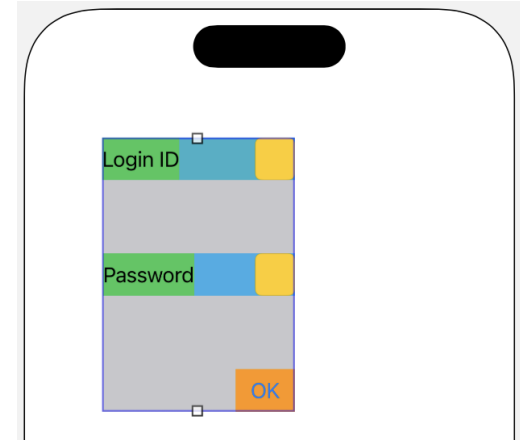
예제

StackView 적용

- 두 개의 StackView와 OK버튼을 동시에 선택한 후 이들에 대하여 StackView를 적용하라.
- 배경색을 적절히 지정하라

바깥 StackView 크기 여백 적용

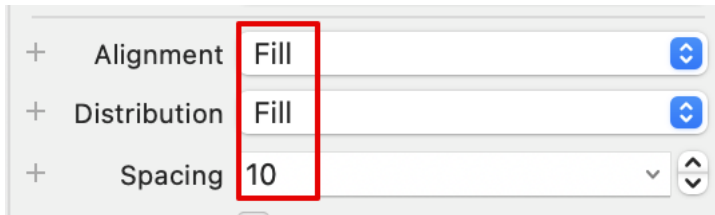
- 좌우여백 20
- 수평 중심
- 다음 속성적용





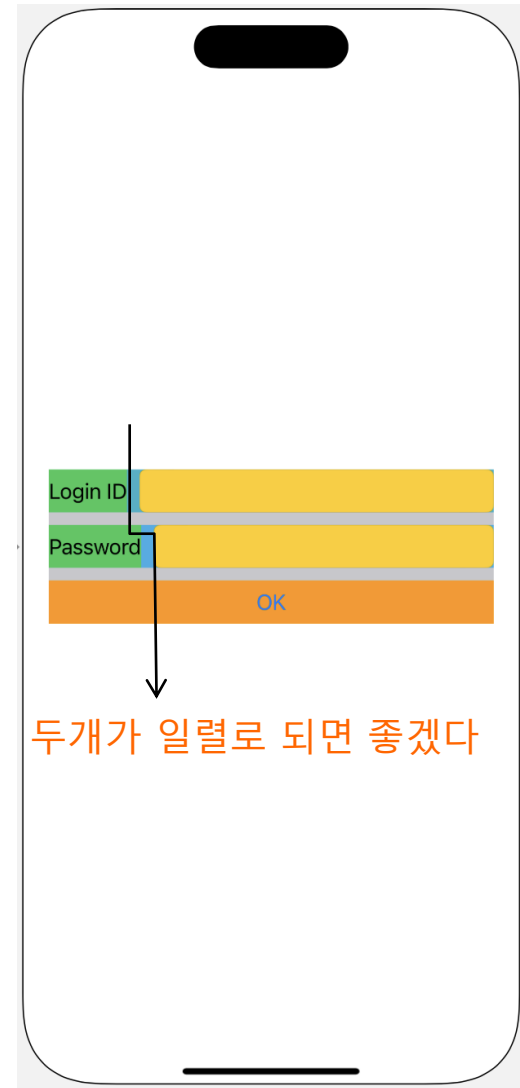
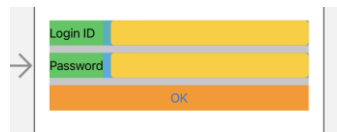
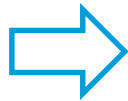
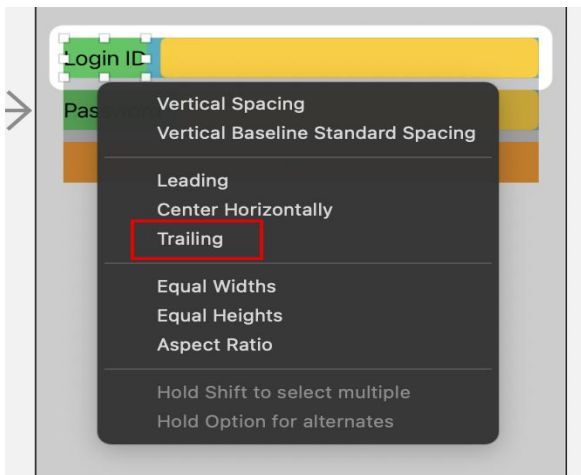
예제

■ 안쪽 2개의 StackView 속성 설정



■ Login ID와 Password를 일렬로 정렬

- Login ID를 Ctrl+Drag -> Password에서 Drop



- Trailing 위에서 Drop



■ 완성

