

Layout : Part I

Mobile Software
2021 Fall

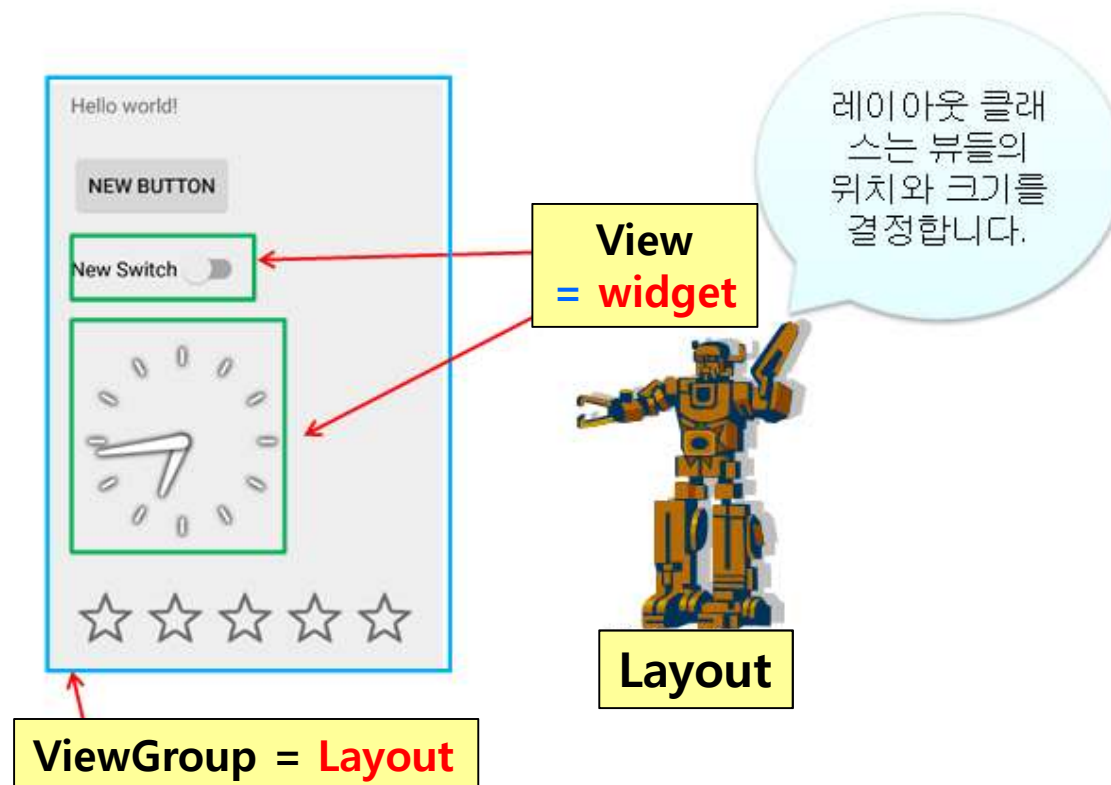
All rights reserved, 2021, Copyright by Youn-Sik Hong (편집, 배포 불허)

What to do next?

- **ViewGroup과 View**
- Layout
 - ConstraintLayout
- 실습

Layout

- Layout은 화면에 View 또는 ViewGroup을 배치
 - View 또는 ViewGroup의 **위치**와 **크기**를 지정



ViewGroup과 View

- **ViewGroup = Layout**

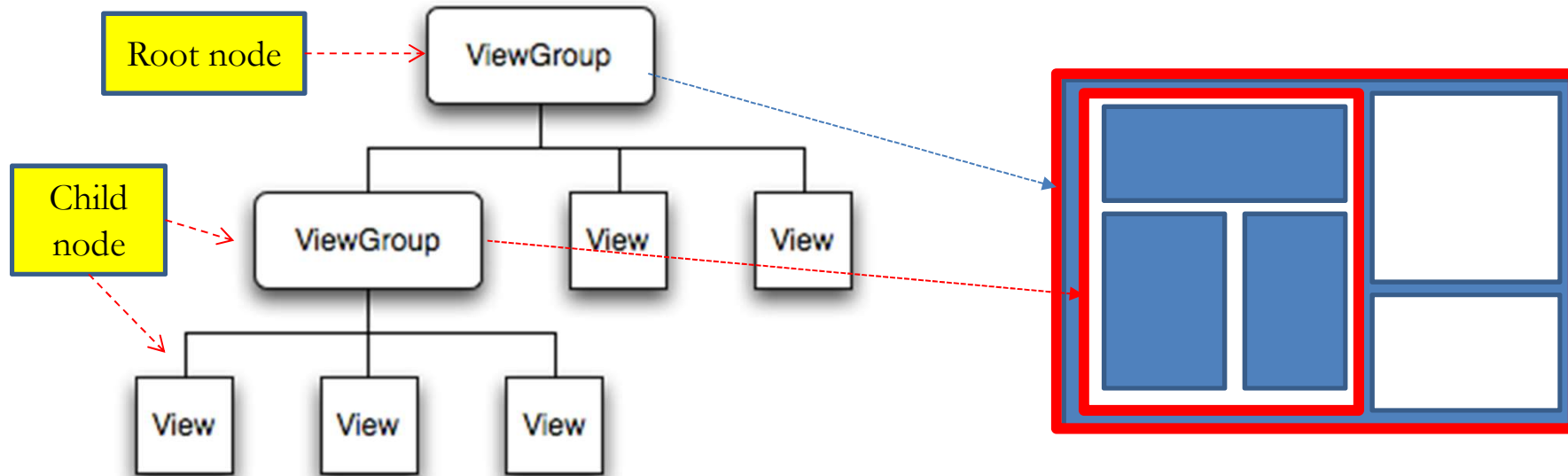
- View 및 ViewGroup을 담을 수 있는 container 역할
 - View 및 ViewGroup을 자신의 공간에 배치
- **ConstraintLayout** (기본 레이아웃)
- **LinearLayout**, **RelativeLayout**
- **FrameLayout**, **GridLayout**, **TableLayout**
- **CoordinatorLayout**

Camel case:
여러 개 단어로 이루어진 경우
각 단어의 첫 번째 문자는
대문자(capital letter)

- **View = Widget**

- UI 기본 구성 요소: **basic building block**
 - 눈으로 볼 수 있으며, 이미 만들어져 있음.
- **Button**, **TextView**, **EditView**, **RadioButton**, **CheckBox**, ...

View는 계층 구조



Android 화면(외부) = UI(User interface)

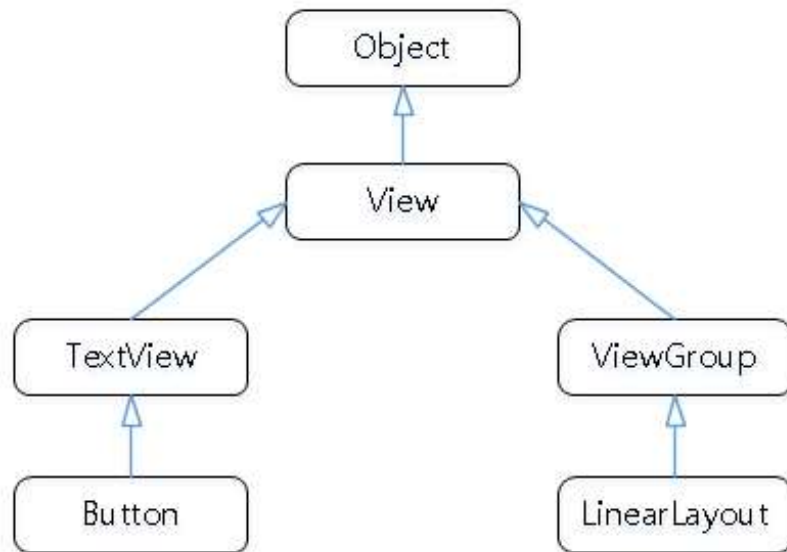
user로부터 입력을 받아 처리한 결과를 보여준다.

Android 화면(내부) = 계층 구조 (hierarchical structure)

view는 자신을 관리하는 ViewGroup에 포함

➔ ViewGroup은 child node에 해당하는 view를 갖고 있다.

ViewGroup과 View는 상속 관계



클래스 상속 (inheritance)
View ← ViewGroup
superclass *subclass*

ViewGroup은 **View**로부터 상속받았다.
→ View 는 ViewGroup을 포함한다.

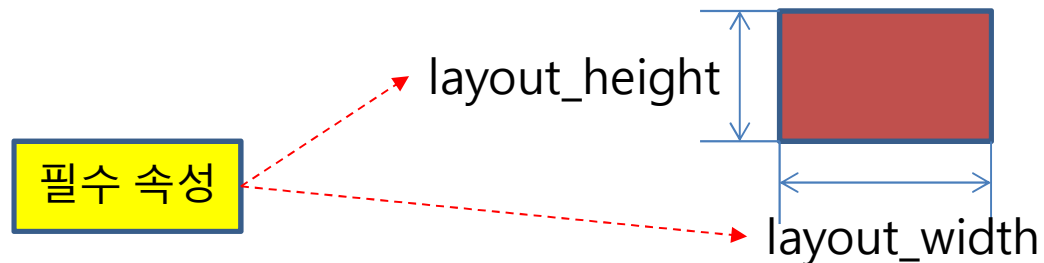
Button은 **TextView**로부터 상속받았다.
→ TextView는 Button을 포함한다.



모든 view의 크기는 사각형이다.

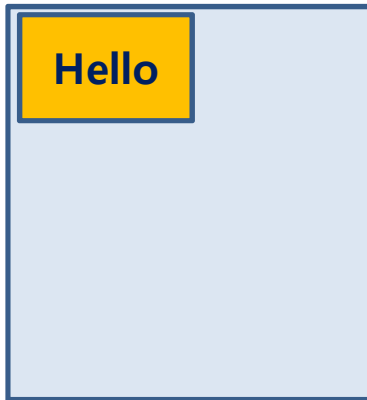
View 의 크기 (1/2)

- view의 크기 : view의 **높이**, **너비**를 지정
 - 높이 : **layout_height**, 너비 : **layout_width**
 - view의 모양에 관계없이 view가 차지하는 공간은 직사각형
- 크기 지정(**필수** – 지정하지 않으면 오류 발생)
 - **match_parent** : view를 포함하는 ViewGroup의 크기와 같음
 - **wrap_content** : view에 포함된 내용 크기와 같음
 - 절대 값 : 숫자
 - 숫자 뒤에 단위 **dp**를 붙여야 함
 - **0dp** → 여유 공간에 따라 크기가 정해짐

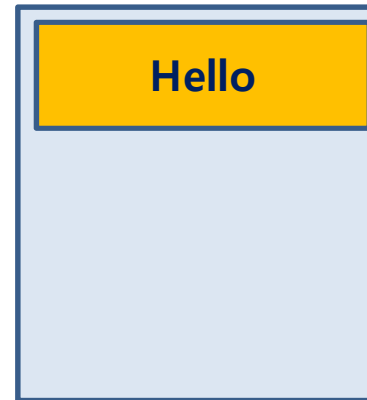


View 의 크기 (2/2)

layout_width="wrap_content"
layout_height="wrap_content"

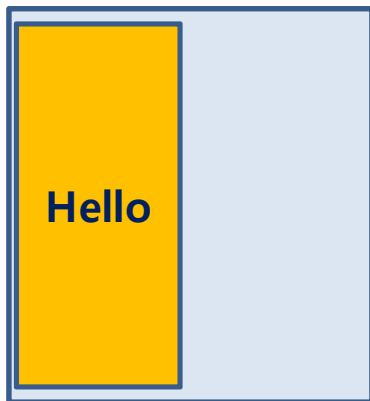


layout_width="match_parent"
layout_height="wrap_content"

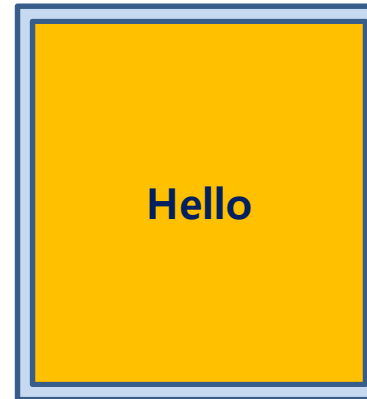


wrap_content나
match_parent 대신
숫자를 사용할 수 있음
(예: 40dp. dp는 단위)

layout_width="wrap_content"
layout_height="match_parent"



layout_width="match_parent"
layout_height="match_parent"



ConstraintLayout에서는
match_parent 속성 대신
match_constraint
속성을 사용.

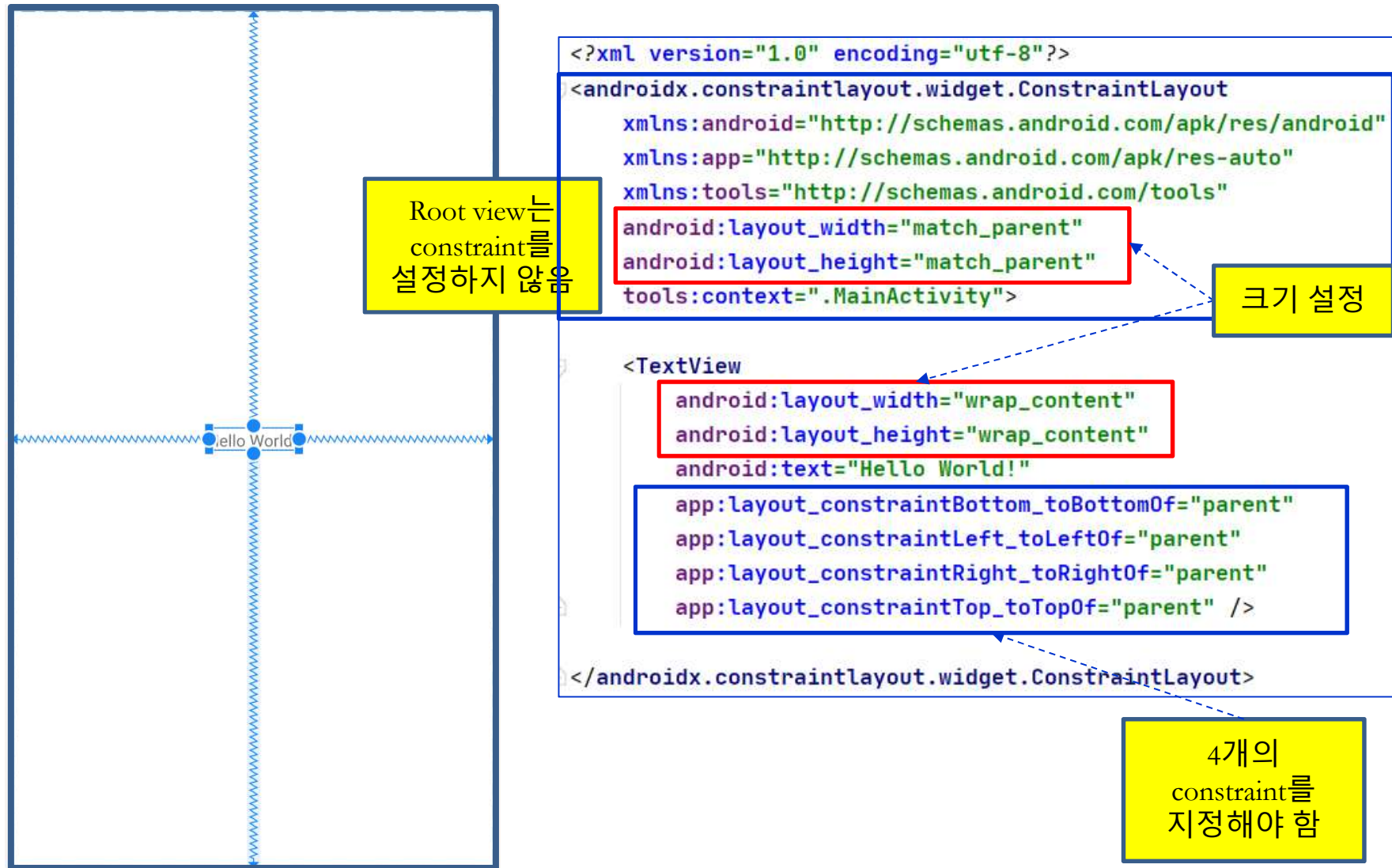
What to do next?

- ViewGroup과 View
- **Layout**
 - **ConstraintLayout**
- 실습
- 참고 사이트
 - <https://developer.android.com/training/constraint-layout>

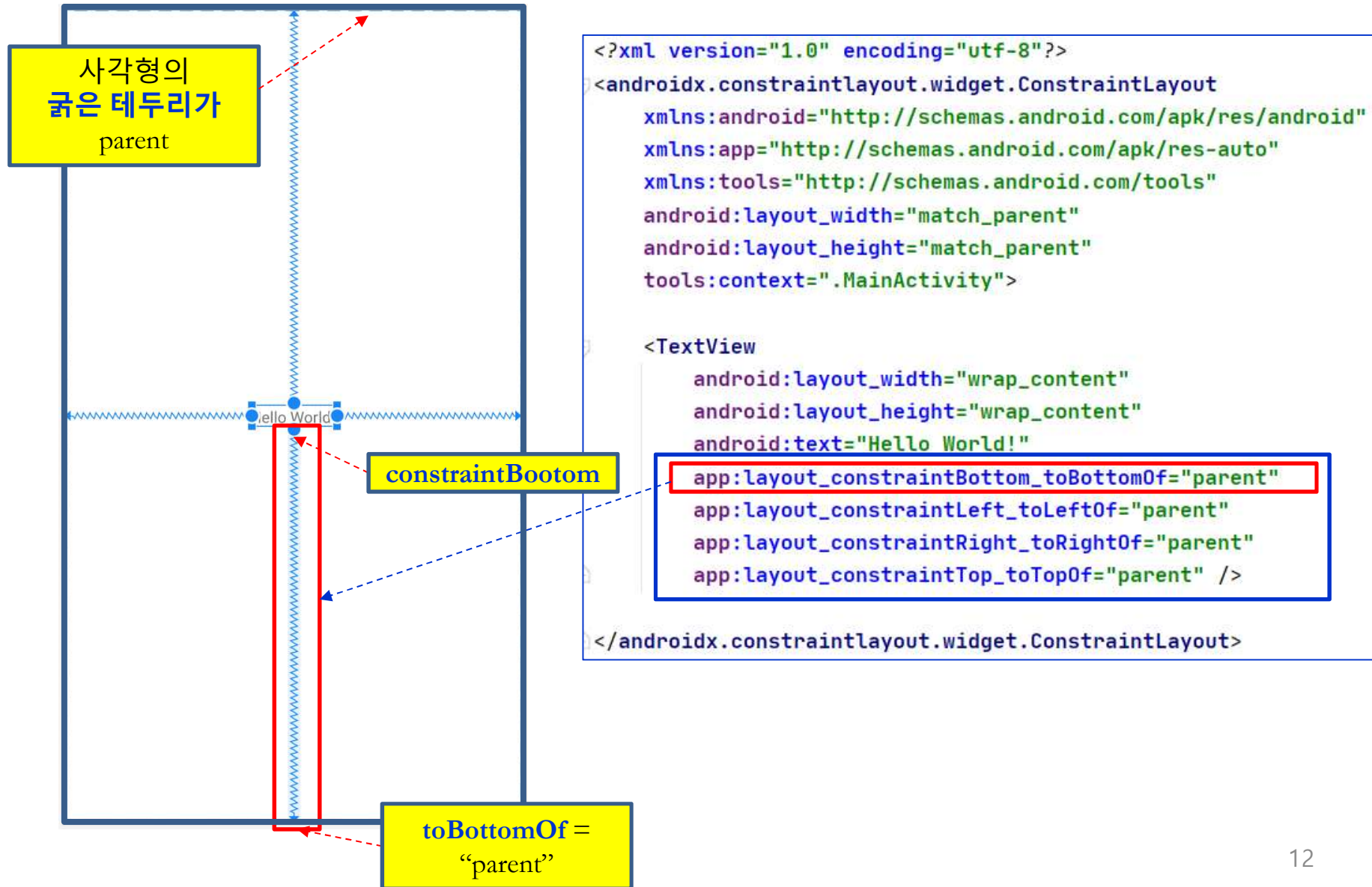
ConstraintLayout 이란?

- Android 7(Nougat) 부터 추가
 - 화면 크기(phone → tablet)가 다르거나, 장치를 회전했을 때(portrait → landscape) 화면 변화에 유연하게 대처
 - Android studio에서는 기본 레이아웃
- It allows you to create large and complex layouts with a **flat view hierarchy**.
- It is **similar** to **RelativeLayout**.
 - But its more **flexible** than RelativeLayout.

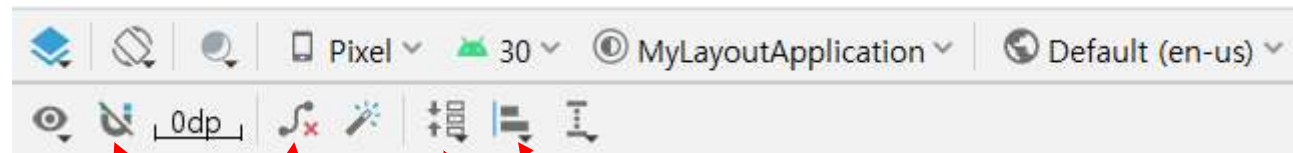
ConstraintLayout 기본 속성 (1/2)



ConstraintLayout 기본 속성 (2/2)



Practice #1: TextView의 constraint을 없앤 후 다시 constraints를 설정해 보자!



Autoconnect
to parent
(toggle)

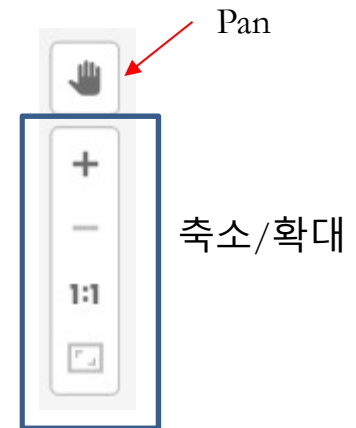
Infer
Constraints

Pack

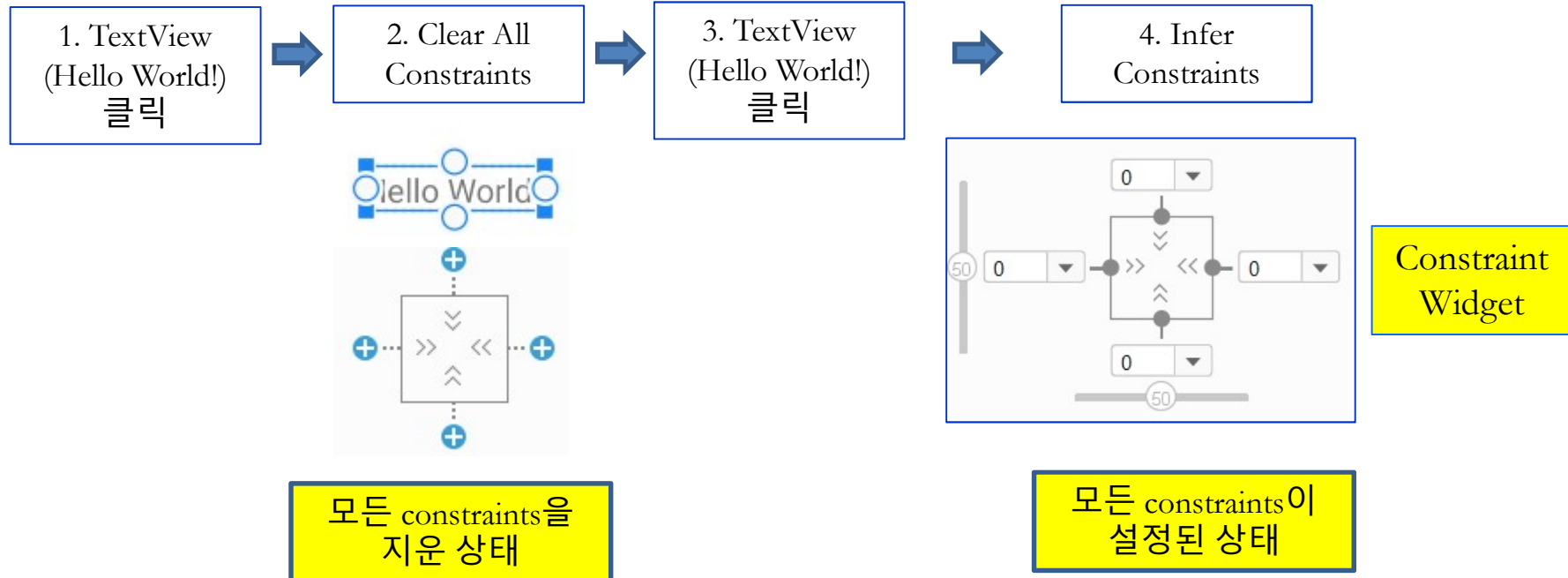
Align

Guidelines

Clear All
Constraints

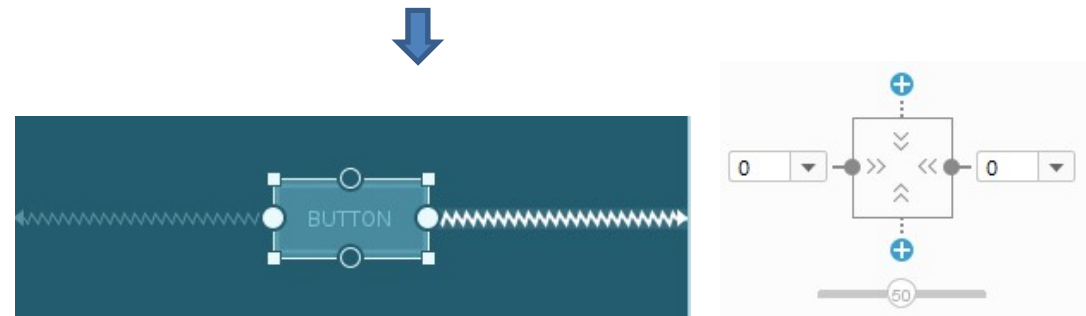
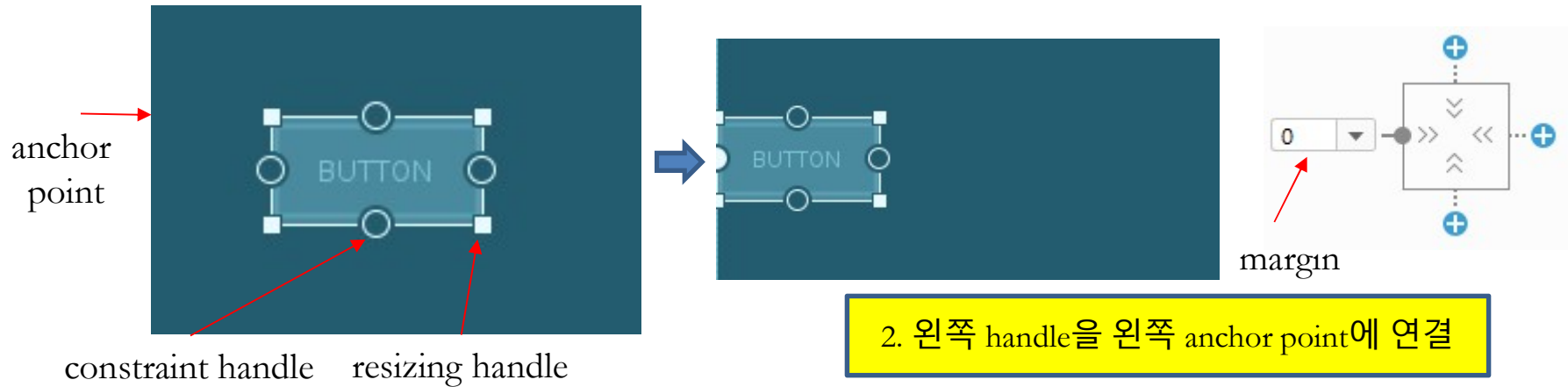


Practice #1: TextView의 constraint를 없앤 후 다시 constraints를 설정해 보자!



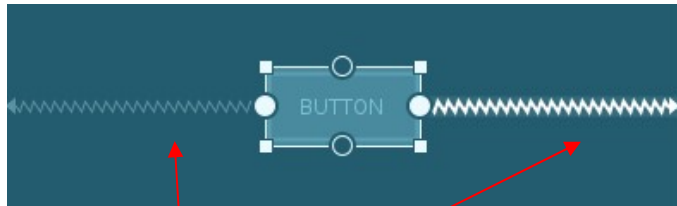
Let's **add constraints** ourselves!

Button 배치 → 수평 constraint 설정



4. XML 파일 확인!

Let's see how the constraint widget changes after setting the horizontal constraints.

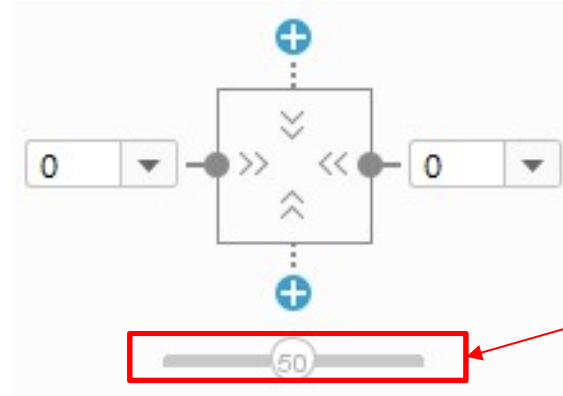


용수철 모양이 나타남

- ➔ 양 끝에서 서로 view를 끌어당긴다.
- ➔ margin : 0 dp

margin = 0dp

margin(간격)이 0이 아니라,
안드로이드 시스템에서
배치 상황에 따라
자율 조정하겠다는 뜻.



Properties window에
수평 bias bar가 나타남

Horizontal bias
현재 값은 50%(=0.5)

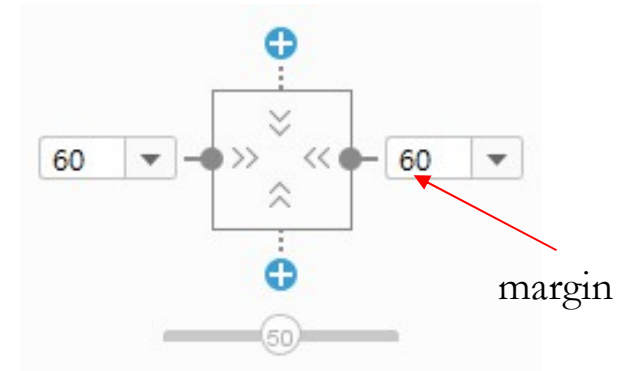
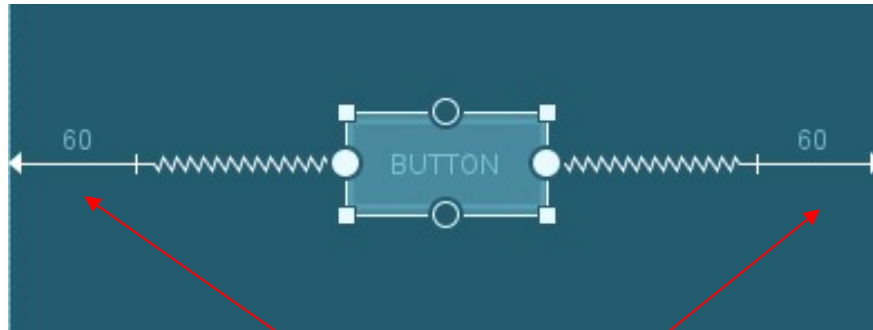
- ➔ Horizontal bias를 30으로 바꾸어 보자.
- ➔ Horizontal bias를 80으로 바꾸어 보자.



다시 화면 중앙에 배치하려면

- ➔ Button 클릭
- ➔ 마우스 오른쪽 버튼
- ➔ Center
- ➔ Horizontally

Let's set the margin of **60dp** on both the **left and right constraint**.



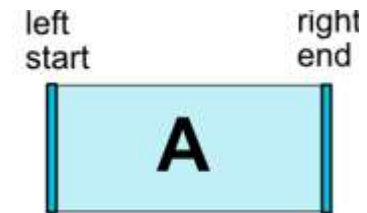
용수철 대신 직선 : 왼쪽과 오른쪽에 각각 최소 60dp 간격은 확보.
→ 용수철 모양은 화면 크기 등에 따라 가변 조절.

margin 속성

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="60dp"
    android:layout_marginLeft="60dp"
    android:layout_marginEnd="60dp"
    android:layout_marginRight="60dp"
    android:text="Button"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    tools:layout_editor_absoluteY="342dp" />
```

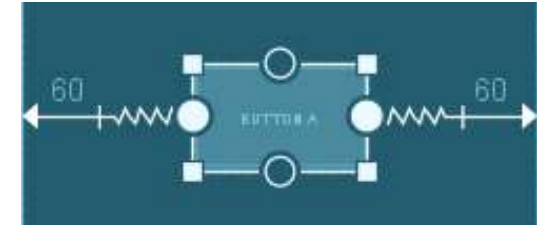
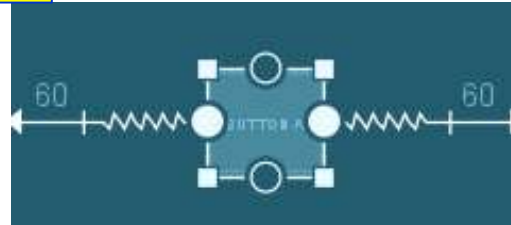
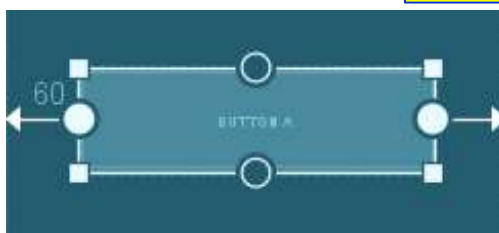
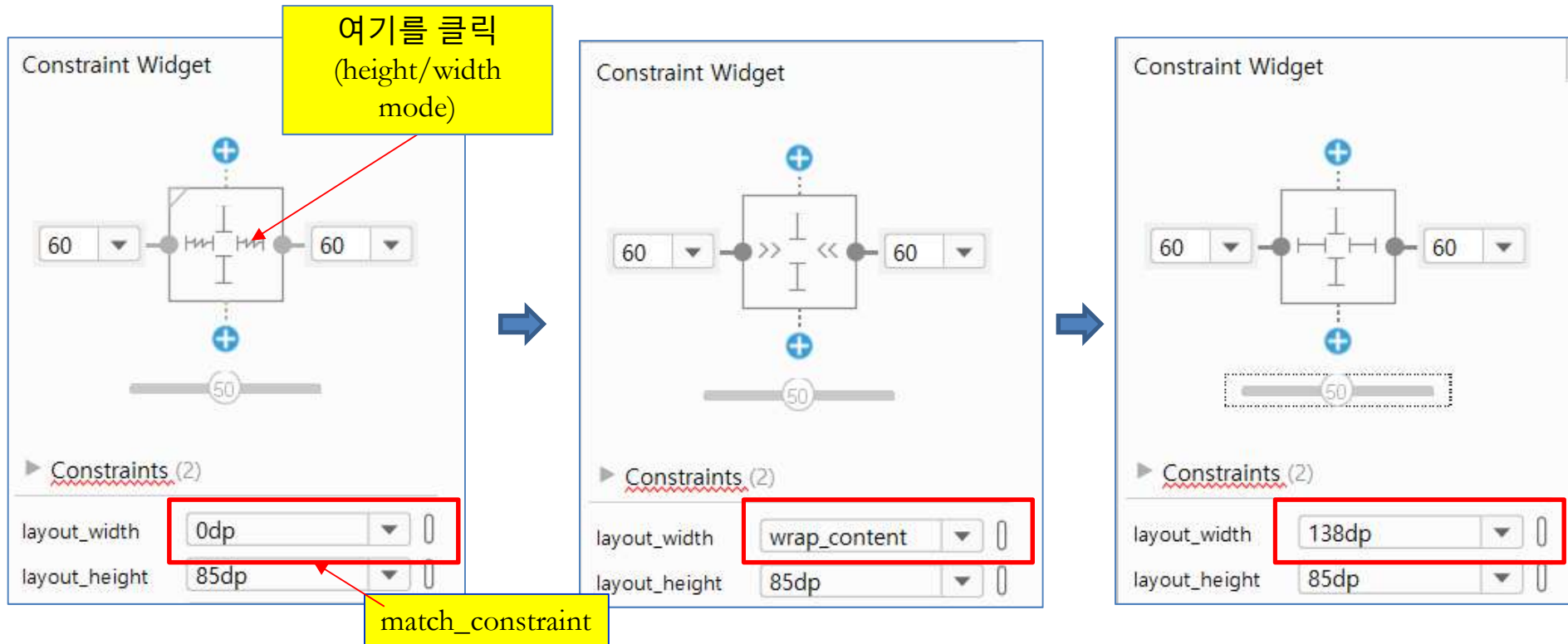
Vertical constraint을 아직
지정하지 않았기 때문에
수직 위치(y)를 절대 값으로 지정

left와 start는 같은 의미
동일한 속성을 2번 정의



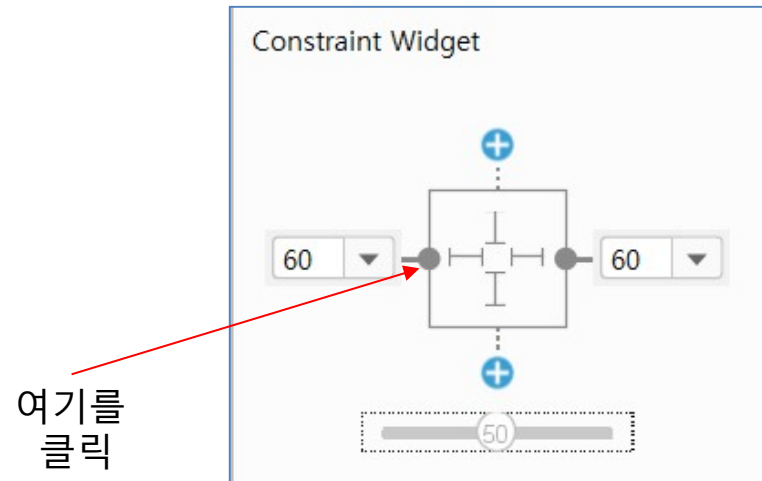
right와 end도 같은 의미
동일한 속성을 2번 정의

Let's play! : Adjust the view size



View의 크기를 최대!
 → margin을 제외하고 남는 공간을
 전부 view에 할당

Delete a constraint



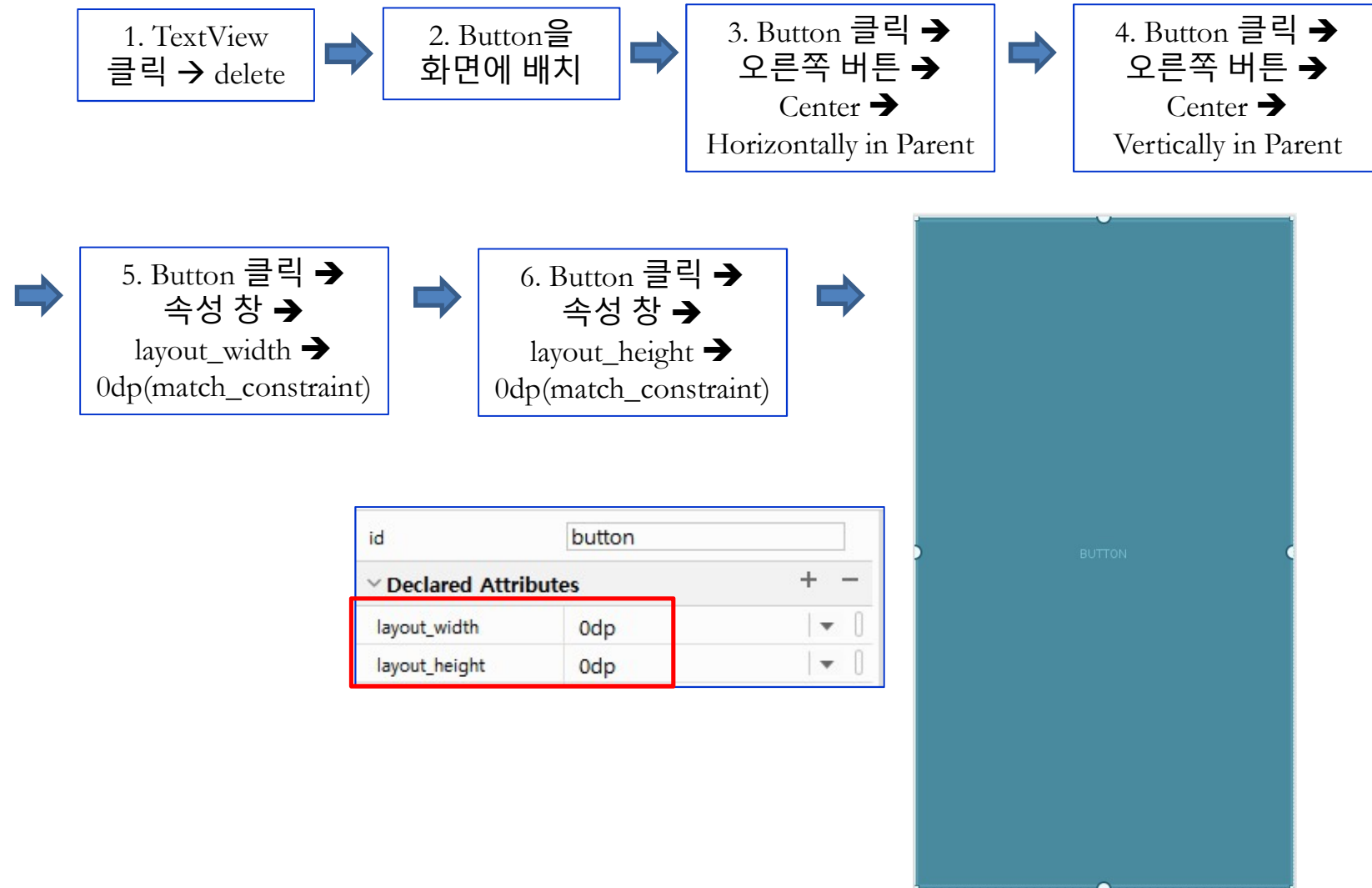
한 번에 한 개의 constraint을 없애려면

→ view 선택 → Properties Window에서 delete constraint을 클릭

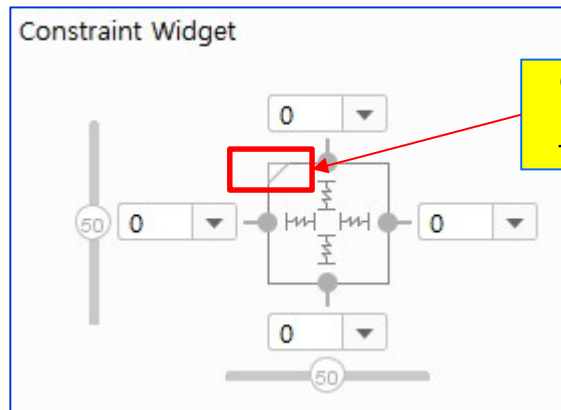
View에 설정된 모든 constraints을 없애려면

→ view 선택 → 오른쪽 버튼 → Clear All Constraints

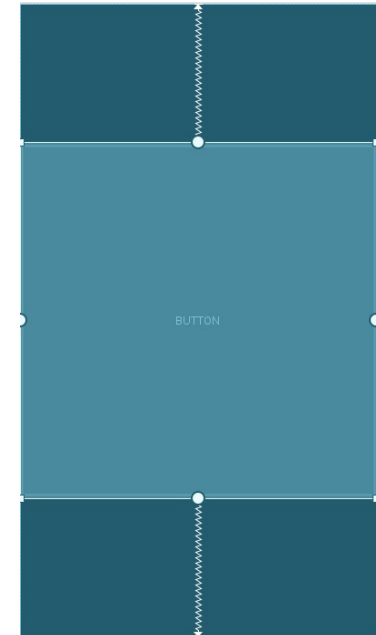
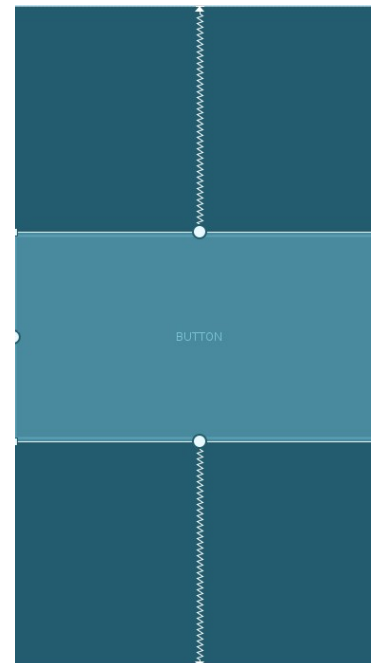
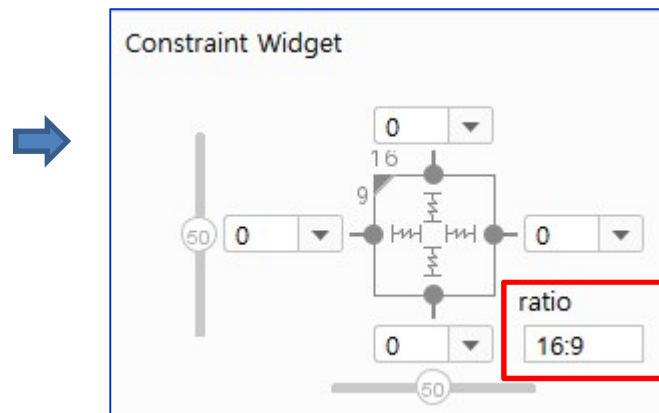
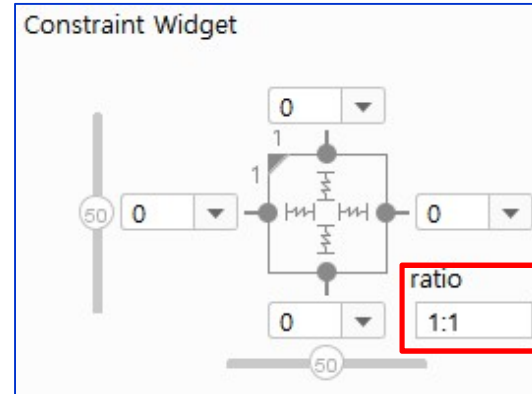
Practice #2: TextView 삭제 → Button을 화면 중앙에 배치



What happens if I change the size of the button to match_constraint?



역 삼각형
모양 클릭

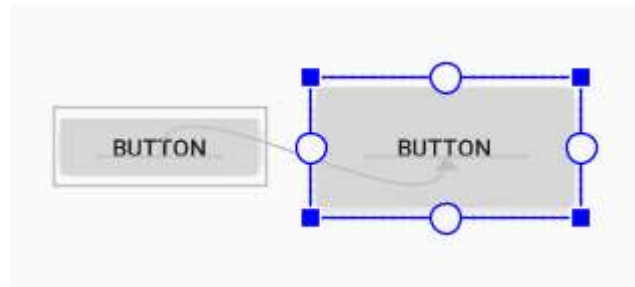
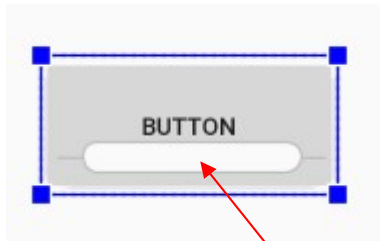


속성 창에서
“Dimension”을 검색

Dimension	
<unnamed>	
layout_constraints	
layout_constraintDimensionRatio	H, 16:9

Baseline Alignment

크기가 서로 다른 2개의 Button의 baseline을 일치시킴.



Baseline :

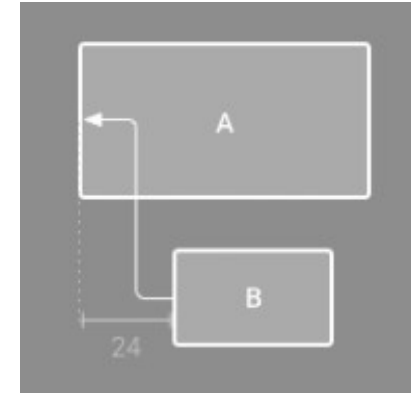
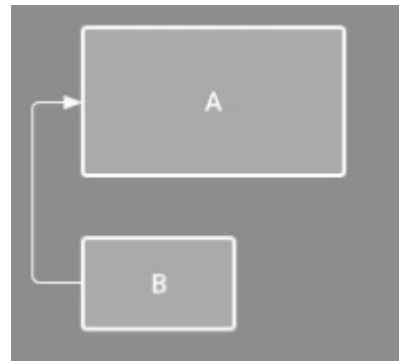
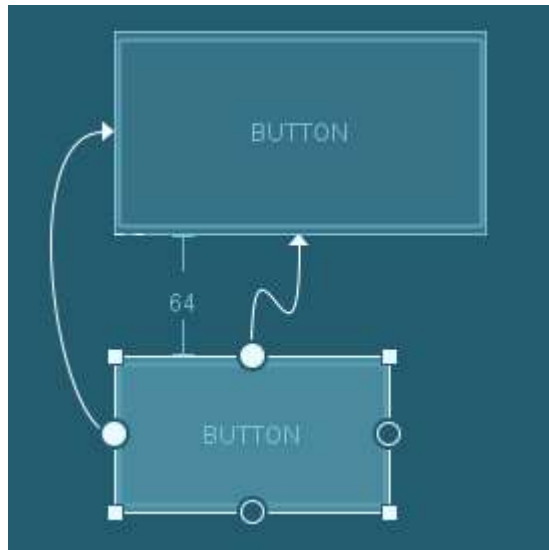
right click → Show Baseline

View Alignment

2개의 버튼의 왼쪽경계선을 일치시킴

➔ Top Edges, Bottom Edges, **Left Edges**, Right Edges

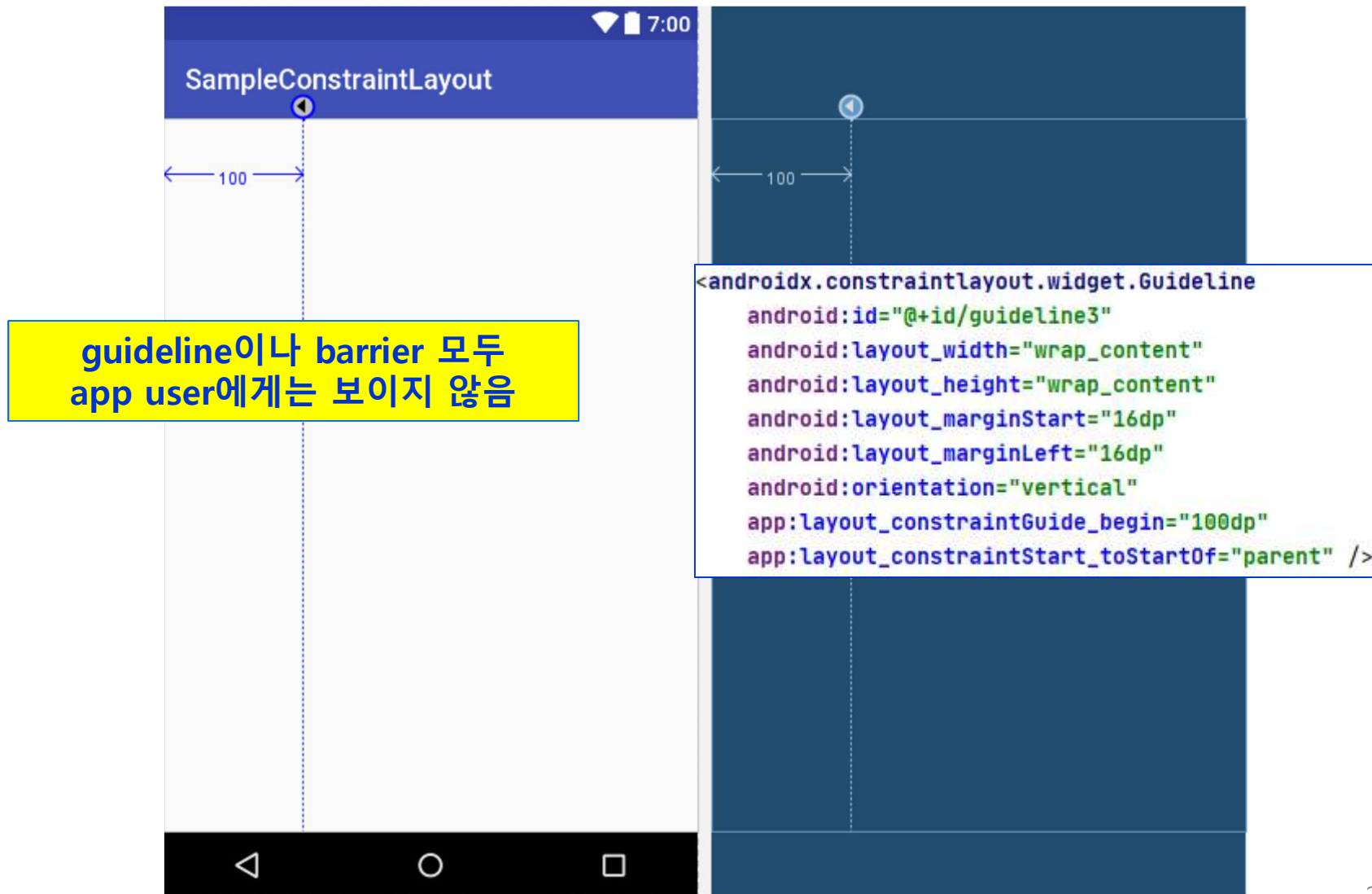
➔ Horizontally Center, Vertical Centers



Ctrl 키를 누른 상태로
2개 버튼을 모두 선택 ➔ 오른쪽 버튼 ➔ Align ➔ Left Edges

*작은 Button 의 위치를 왼쪽으로 옮기면 어떻게 될까?
작은 Button 의 위치를 오른쪽으로 옮기면 어떻게 될까?*

Constrains to a guideline (1/2)



SampleConstraintLayout

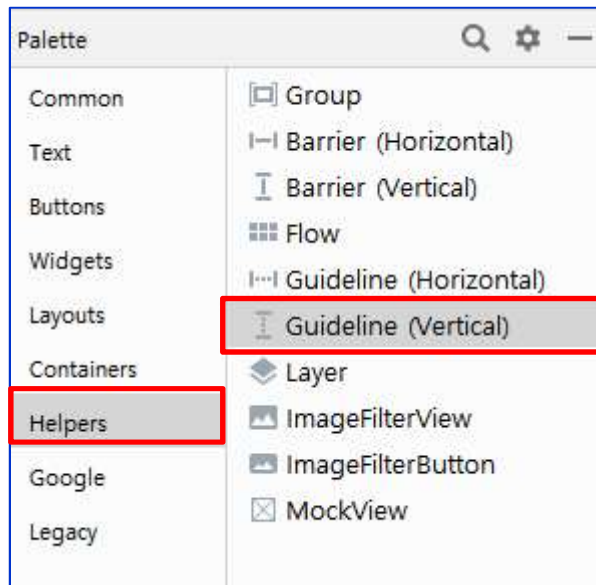
100

guideline이나 barrier 모두
app user에게는 보이지 않음

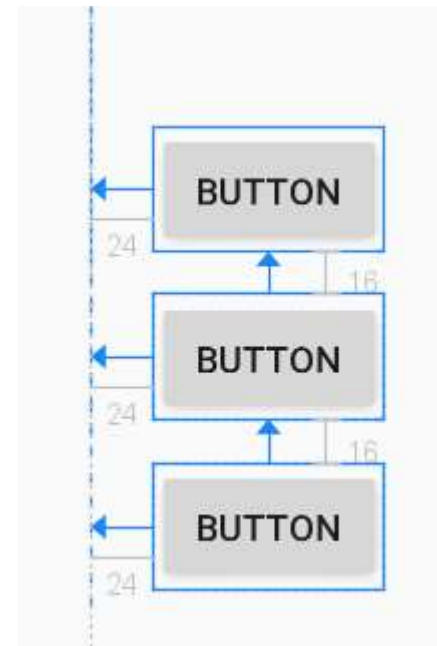
```
<androidx.constraintlayout.widget.Guideline
    android:id="@+id/guideline3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginLeft="16dp"
    android:orientation="vertical"
    app:layout_constraintGuide_begin="100dp"
    app:layout_constraintStart_toStartOf="parent" />
```


Constrain to a guideline (2/2)

Palette > Helpers > Guideline (Vertical) 선택



Guideline 오른쪽에 버튼 3개 배치

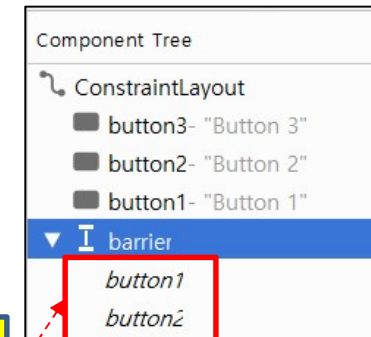
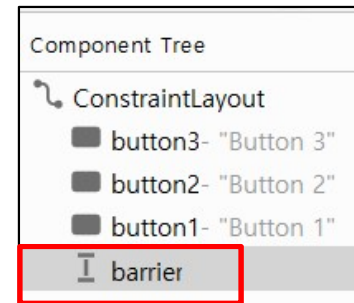
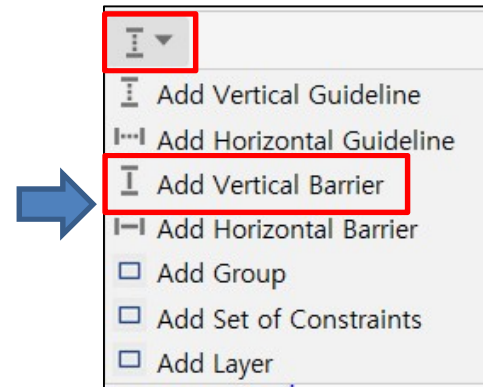
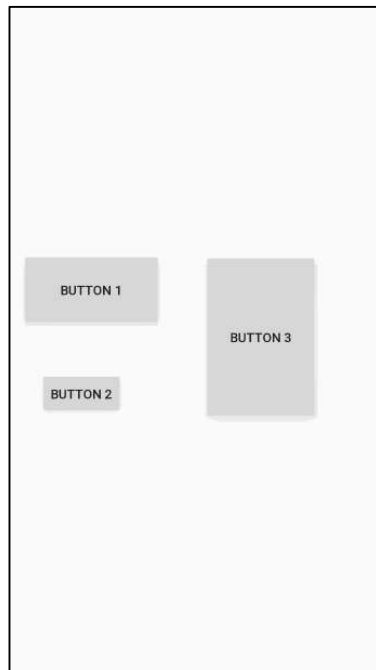


또는 아이콘에서 Horizontal Guidelines 선택



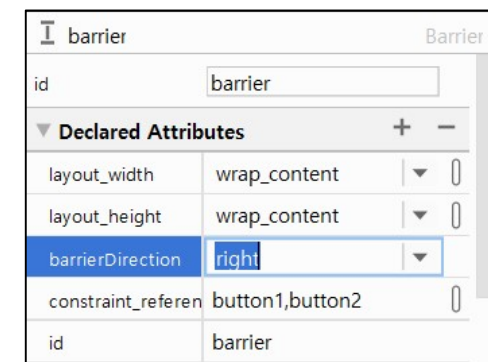
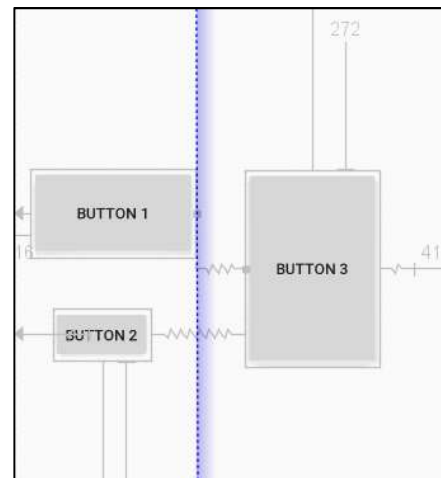
Guidelines

Adding Barriers



button1과 button2를
barrier 아래로 drag & drop

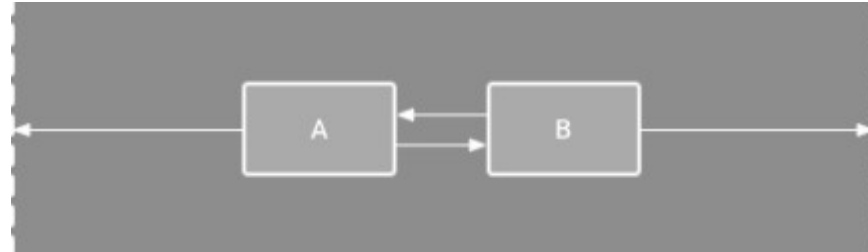
Button 3은 barrier를
넘어 올 수 없음
→ 항상 button1과
→ button 2의
→ 오른쪽에 위치.



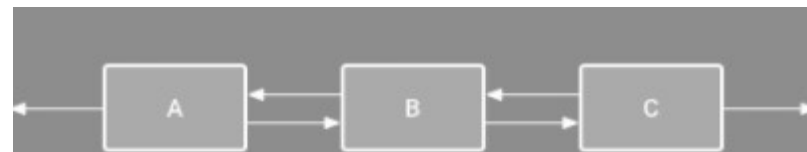
Properties window
barrier direction을
Right 로 설정

Control linear groups with a chain (1/2)

A chain is a group of views that are linked to each other with **bidirectional position constraints**.



Spread

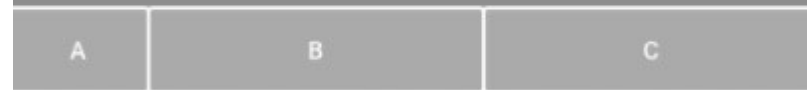


Spread Inside



Constraints >
horizontal_weight,
vertical_weight

Weighted



Spread 체인
스타일 일 때
weight 적용 가능

Constraints >
horizontal_bias,
vertical_bias

Packed



Packed 체인
스타일 일 때
bias 적용 가능

3개 버튼을 수평 체인으로 묶기



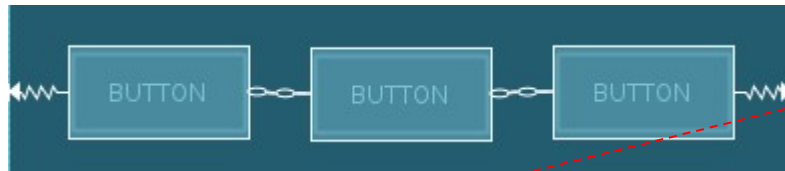
1. Disable Autoconnect to parent



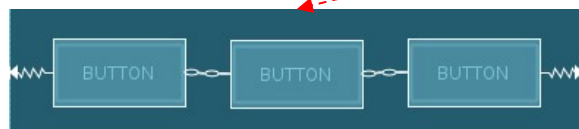
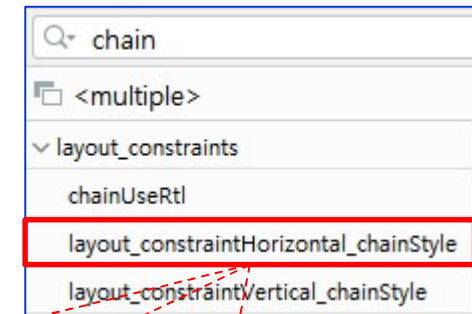
2. 3개 버튼을 수평 방향으로 배치



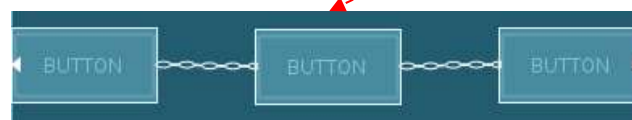
3. 3개 버튼을 모두 선택
(경계선이 흰색으로 표시)



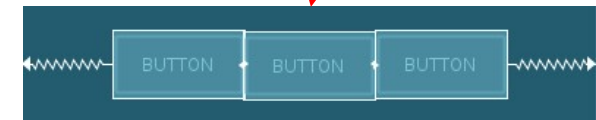
4. 마우스 오른쪽 버튼 → Chains
→ Create Horizontal Chain



Spread



Spread inside

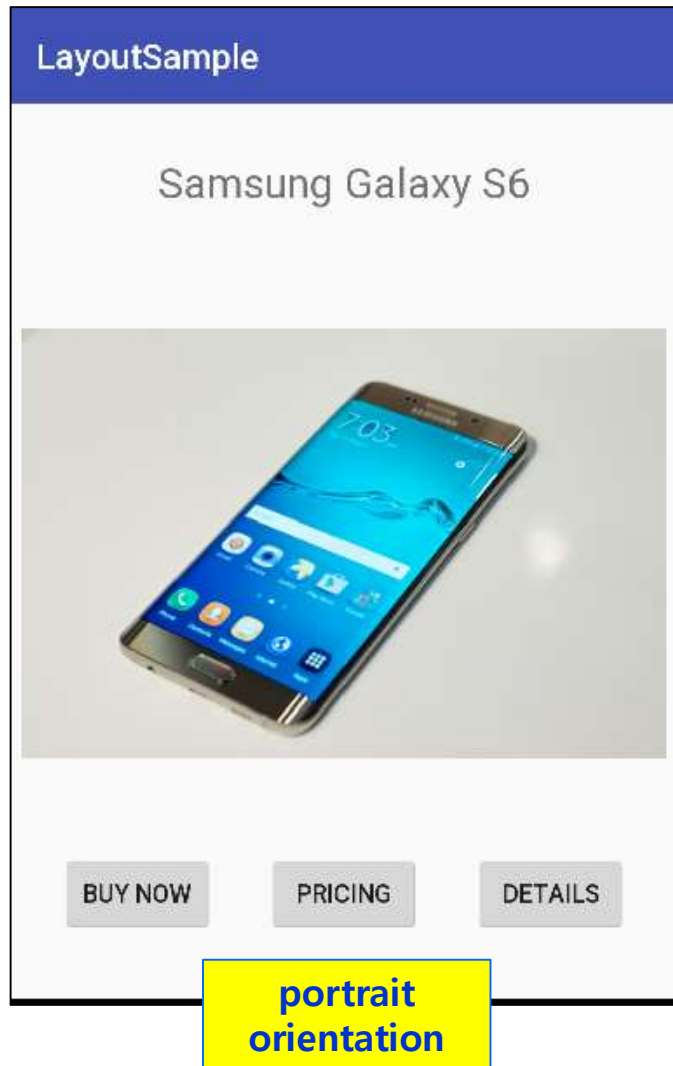


packed

What to do next?

- ViewGroup과 View
- Layout
 - ConstraintLayout
- **실습**

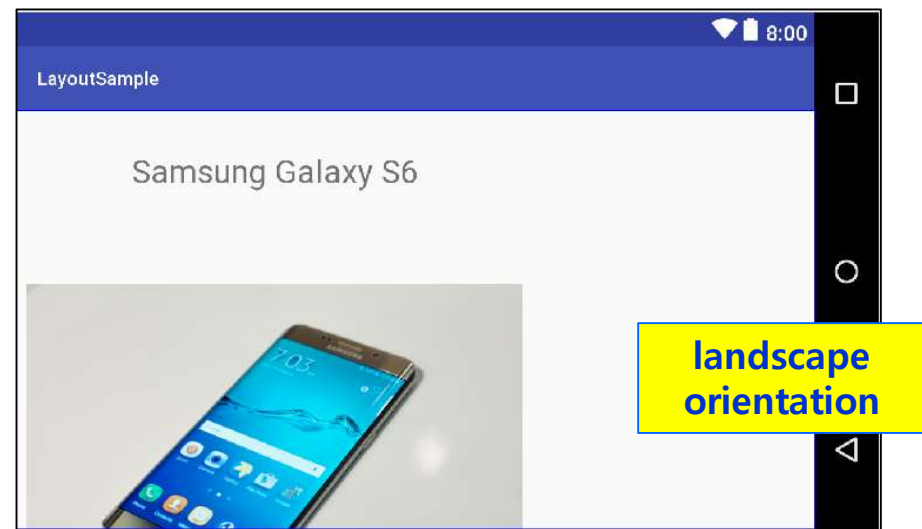
실습 요약



Constraints를 추가하지 않은 상태로
TextView, ImageView, Button(3개)을 배치

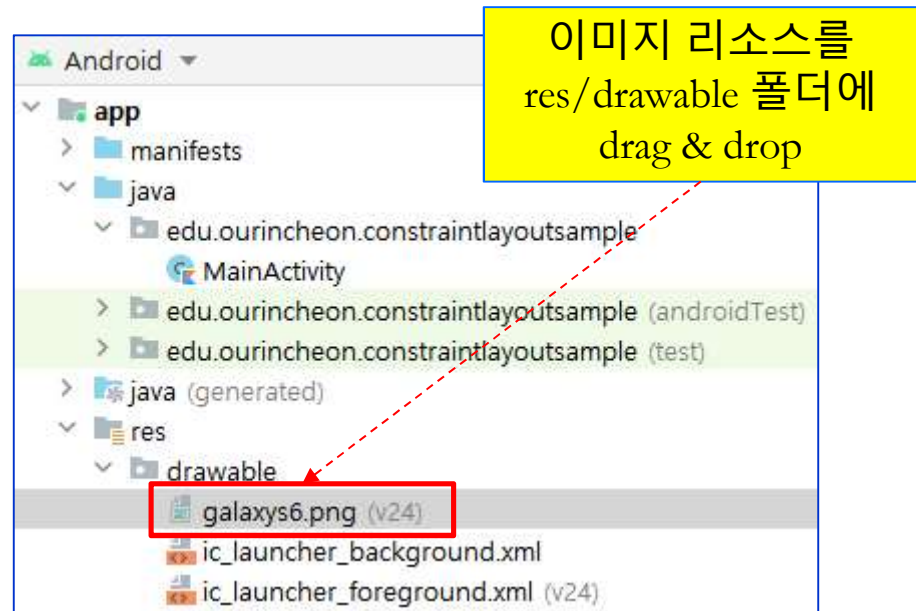


Orientation을 Landscape로
바꾸면 어떤 모양이 될까?



실습 준비

- 새 프로젝트 생성
 - Activity : **Empty Activity**
 - Application name : **Constraint Layout Sample**
- 리소스
 - **galaxy6.png**



실습 (1/4) : Add the widgets

TextView, ImageView,
Button(3개)을
오른쪽 그림처럼 배치



Auto connect → off



TextView 속성 설정

```
android:text="Samsung Galaxy S6"  
android:gravity="center"  
android:textAlignment="center"  
android:textSize="24sp"
```



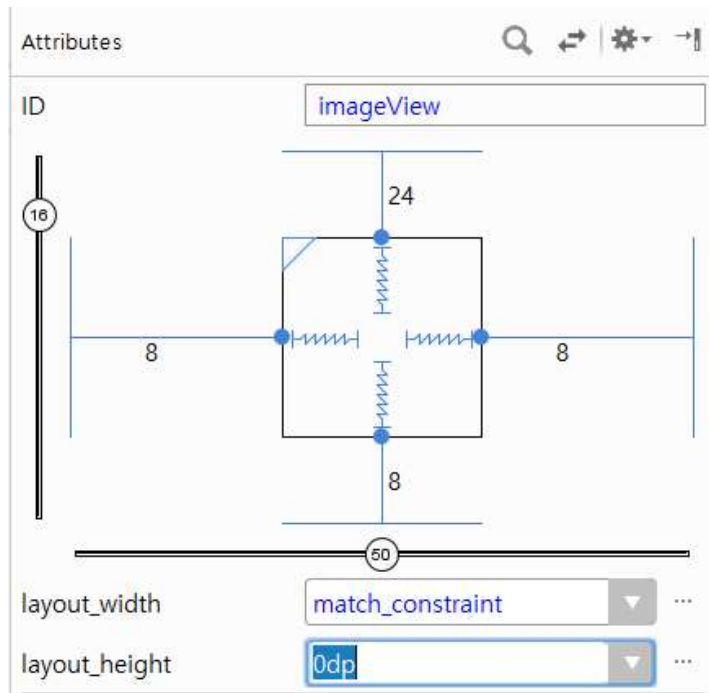
ImageView에 리소스 설정

**View > Tool Windows
> Resource Manager
→ galaxys6.png → 더블 클릭**

실습 (2/4) : Adds the constraints



TextView의 Left (=Start),
Right (=End), Top에 constraint 추가



ImageView의 4곳에
Constraints 추가
Top은 **TextView**의 Bottom에 연결
Bottom은 **Pricing** Button의 Top에 연결

layout_width = **0dp**
layout_height = **0dp**
top margin = 24dp
left, right, bottom margin = 8dp

실습 (3/4) : Create horizontal chain



Auto connect → on

3개 버튼을 모두 선택



마우스 오른쪽 버튼

→ Center Horizontally 선택

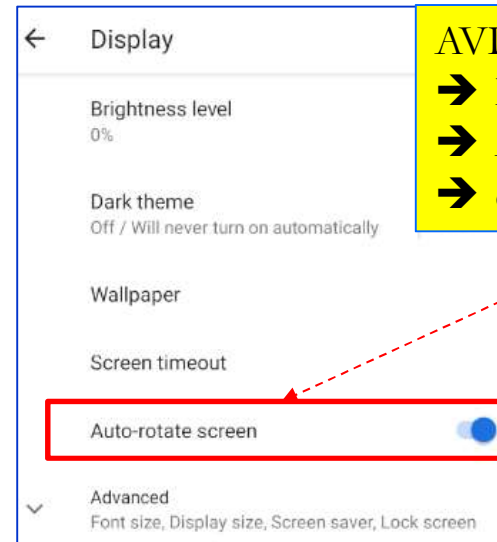
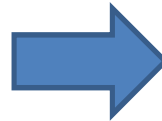
→ Horizontal_chainStyle : spread



3개의 버튼 모두에 대해
Bottom을 parent의 Bottom에 연결
Bottom margin = 16dp

실습 (4/4): Test the layout

portrait
orientation



AVD의 Settings

- Display
- Auto rotate screen
- enable

landscape
orientation

