ConstraintLayout

Mobile Software
2019 Fall

참고 사이트: https://developer.android.com/training/constraint-layout/index.html#adjust-the-view-margins

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

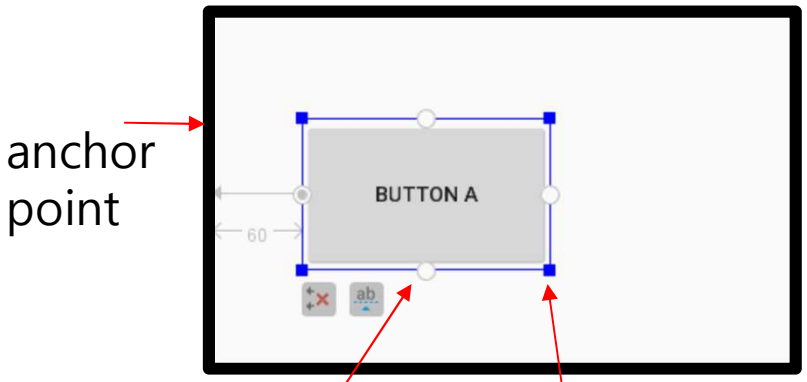
- It allows you to create large and complex layouts with a **flat view hierarchy**.
- It is **similar to RelativeLayout**.
  - But its more **flexible** than RelativeLayout.

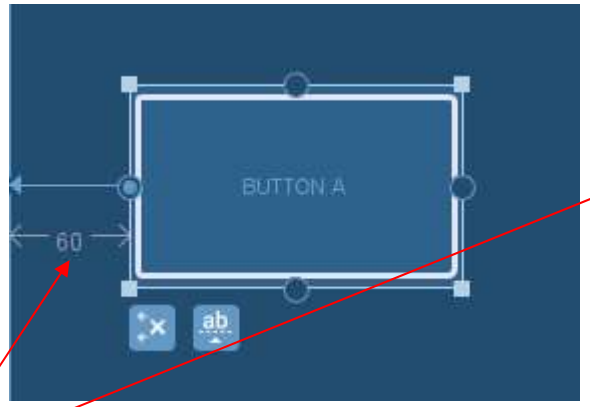
ConstraintLayout 관련 용어

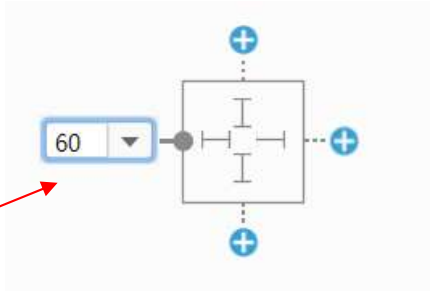
- **Constraint** (제약)
  - 화면 크기나 장치 회전 등에 자동으로 적응하도록 함
- **Margin :** 간격(여백)을 지정
- **Constraint bias :** 0~1 (0%~100%) 사이 값
  - 수직, 수평 상대 비율에 따라 widget 위치가 정해짐
- **Chain** 과 **Chain head**
  - 2개 이상 widget을 하나의 그룹처럼 동작하도록 만듦
- **Chain style**
  - Spread, spread inside, weighted, packed
- **Baseline alignment**
- **Guideline**
- Widget 크기
  - fixed, **MATCH_CONSTRAINT**(0dp), wrap_content

Add a constraint

버튼 배치 ➔ 이 버튼에 수평 constraint을 설정하자.







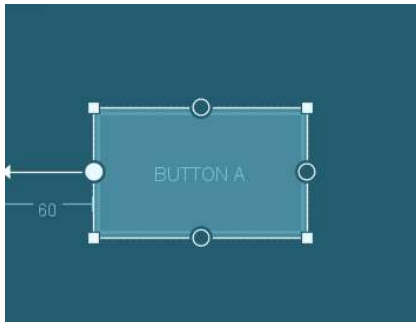
Properties창에서
margin 설정

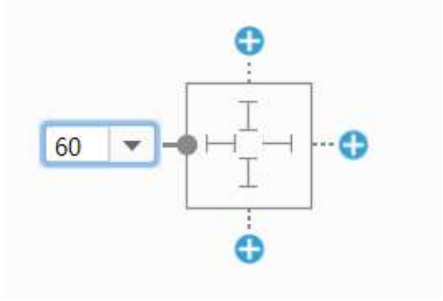
Anchor point
- The edge of another view
- The edge of the layout
- A *guideline*

뷰의 constraint handle을
부모 레이아웃의 anchor point에
연결한다.

최소 2개의 constraint
(수평, 수직)가 필요.

XML layout





Properties window

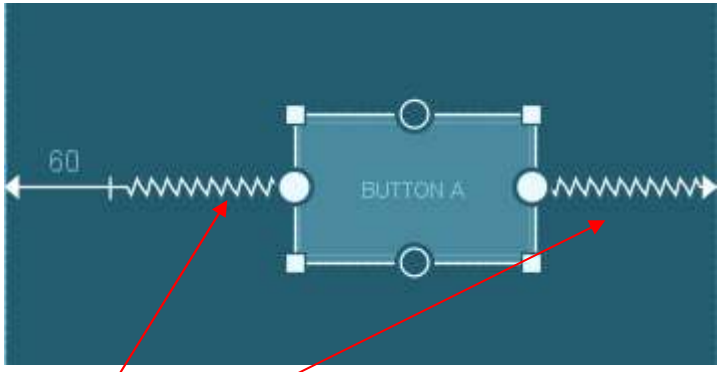
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.ourincheon.sampleconstraintlayout.MainActivity"
    tools:layout_editor_absoluteY="81dp"
    tools:layout_editor_absoluteX="0dp">

    <Button
        android:id="@+id/button17"
        android:layout_width="164dp"
        android:layout_height="100dp"
        android:text="Button A"
        tools:layout_editor_absoluteY="71dp"
        android:layout_marginLeft="60dp"
        app:layout_constraintLeft_toLeftOf="parent" />
</android.support.constraint.ConstraintLayout>
```

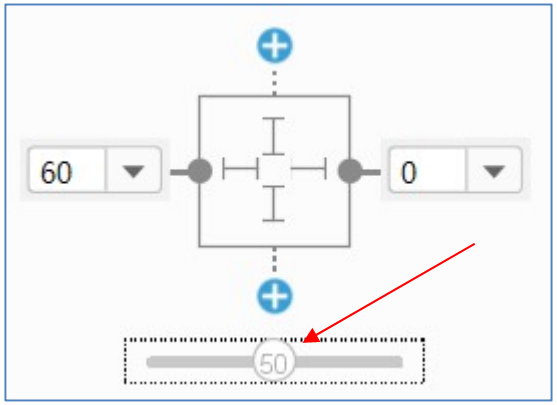
This view is not constrained, it only has designtime positions, so it will jump to (0,0) unless you add constraints more... (Ctrl+F1)

Add another constraint

버튼의 오른쪽에도 수평 constraint을 설정하자.

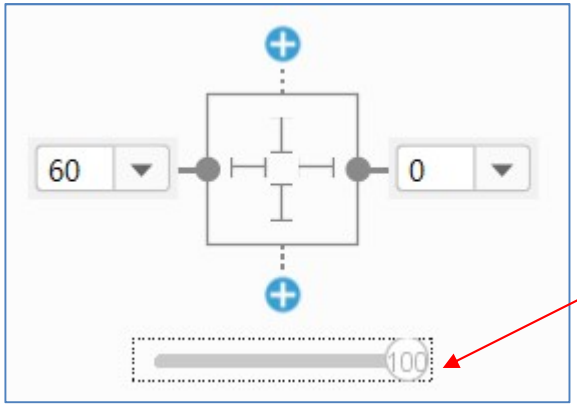


용수철 모양이 나타남
➔ 양 끝에서 서로 view를 끌어당긴다!?



Properties window에 수평 bias bar가 나타남

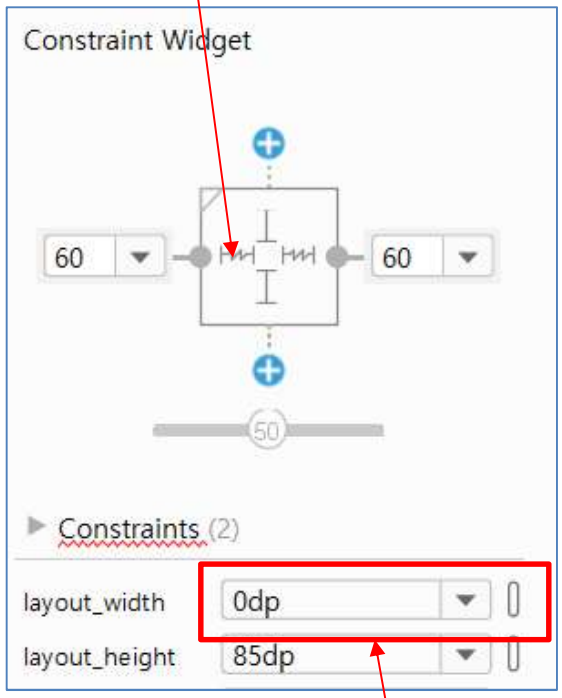
현재 값은 50% (=0.5)

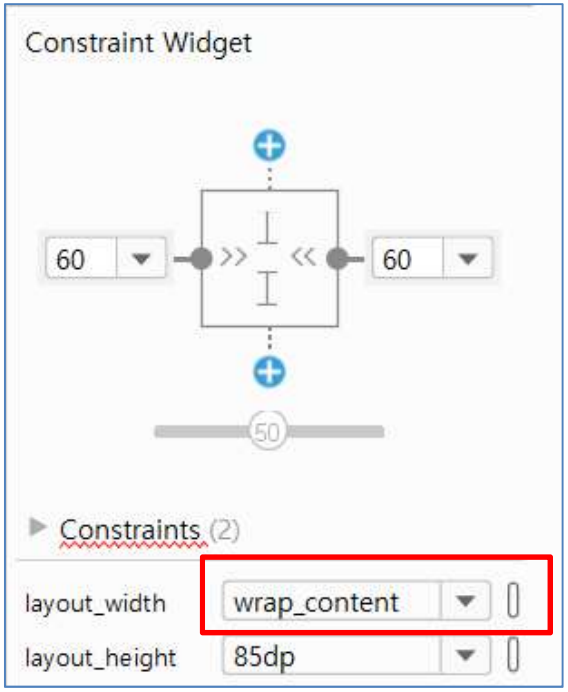


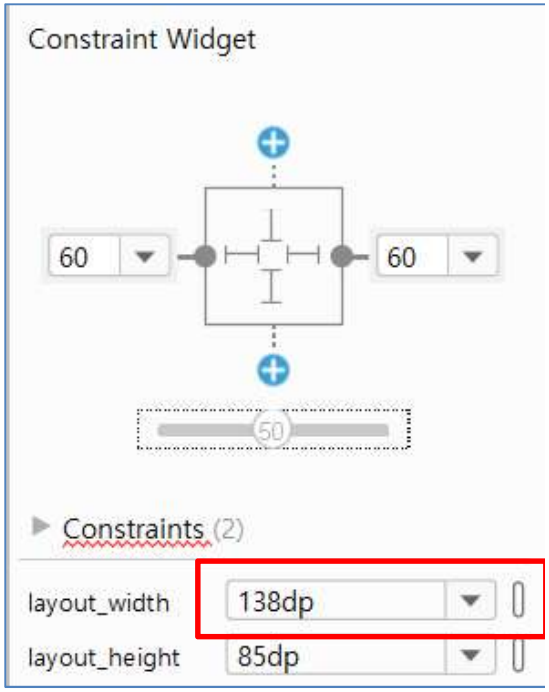
Constraint bias를 100%(=1.0)로 바꾸면 어떻게 될까?

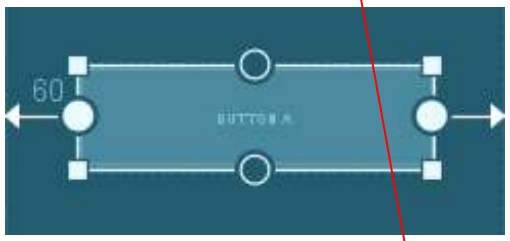
여기를 클릭
(**height/width mode**)

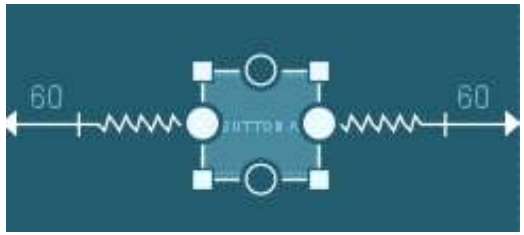
Adjust the view size

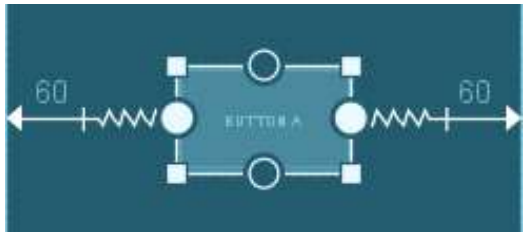






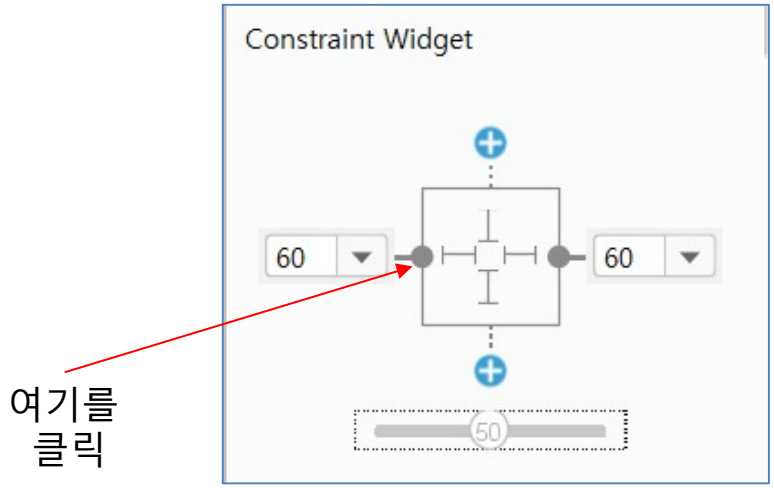






match_constraint 선택

Delete a constraint

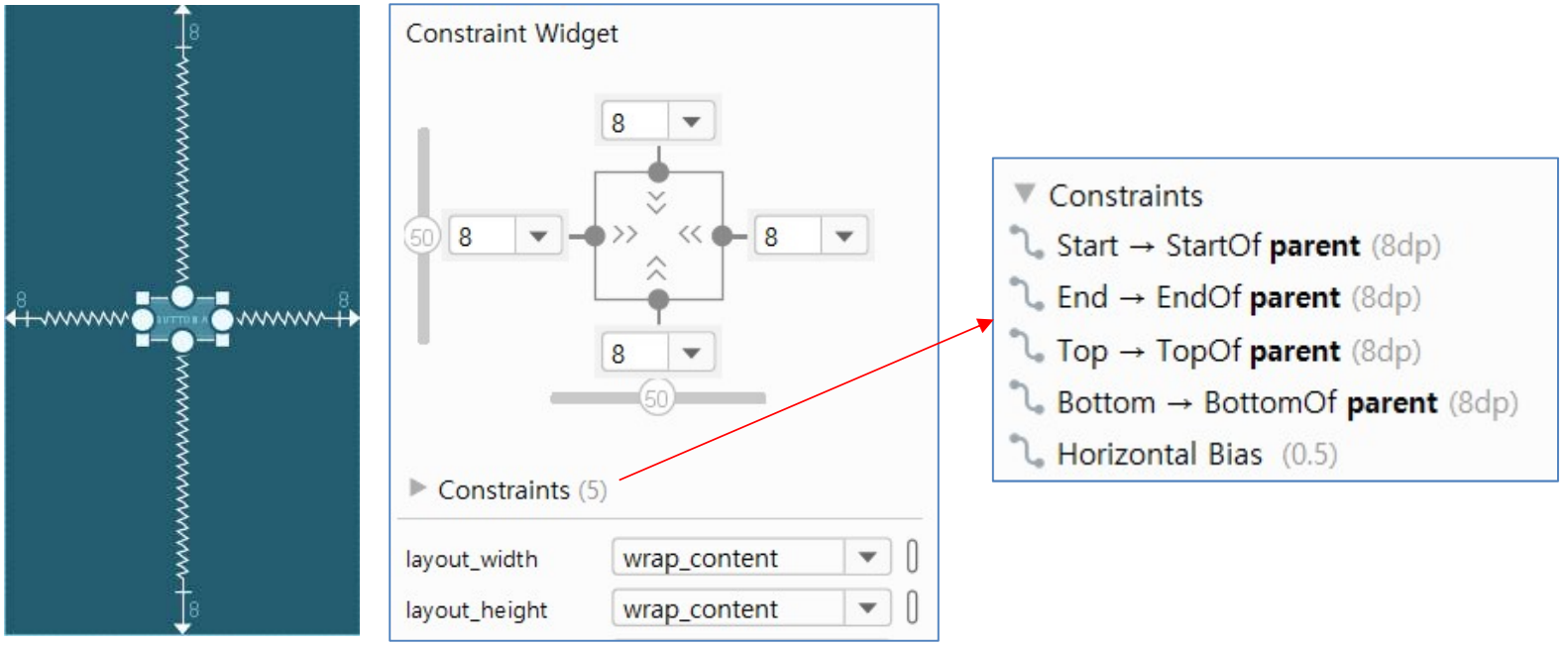


한 번에 한 개의 constraint을 지우려면
Properties 창에서 delete constraint(파란색 원)을 클릭

View에 설정된 모든 constraints을 없애려면
view 선택 ➔ 오른쪽 버튼 ➔ Clear All Constraints

Practice #1

버튼 배치 ➔ XML layout의 에러 표시가 없어지도록
constraint을 설정 하시오. 단, 아래 그림처럼
화면 정 중앙에 버튼을 배치 하시오.



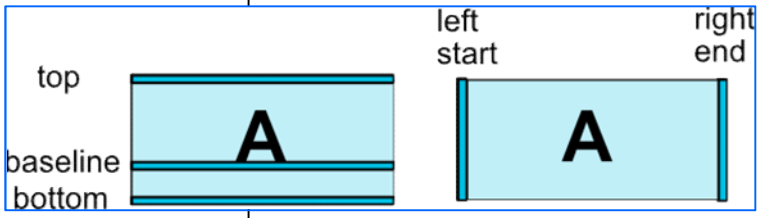
방법 1 : 수평, 수직 bias bar를 각각 50%(=0.5)로 조절
bias bar가 properties 창에 나타나려면 어떻게 하면 될까?

방법 2 : *view 선택 ➔ 오른쪽 버튼 ➔* ***Center Horizontally in Parent***
view 선택 ➔ 오른쪽 버튼 ➔ ***Center Vertically in Parent***

Check the XML layout

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ChainTestActivity"
    tools:layout_editor_absoluteY="81dp">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="Button"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
```



잠깐! Shortcut icons

Auto Connect

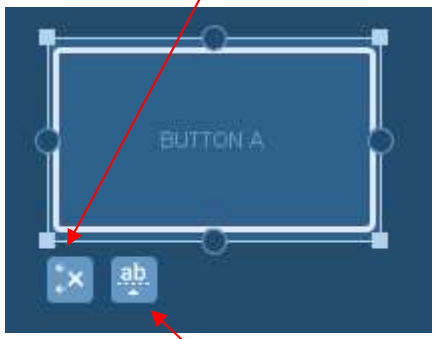
Hide Constraints

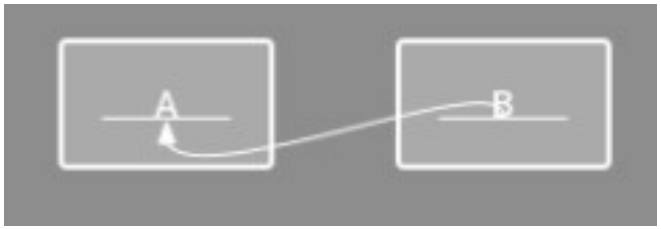
Infer Constraints

Default View Margin (=8)
새 값을 지정하면,
지금 설정하는 view부터 적용

Guidelines
수평, 수직 guideline 중 하나를 선택

Clear All Constraints



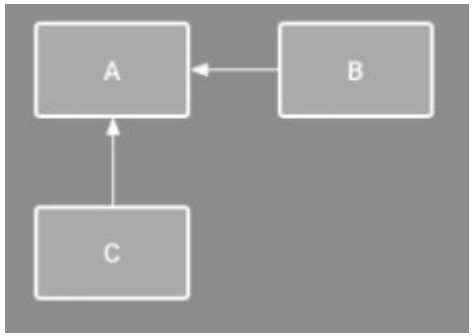


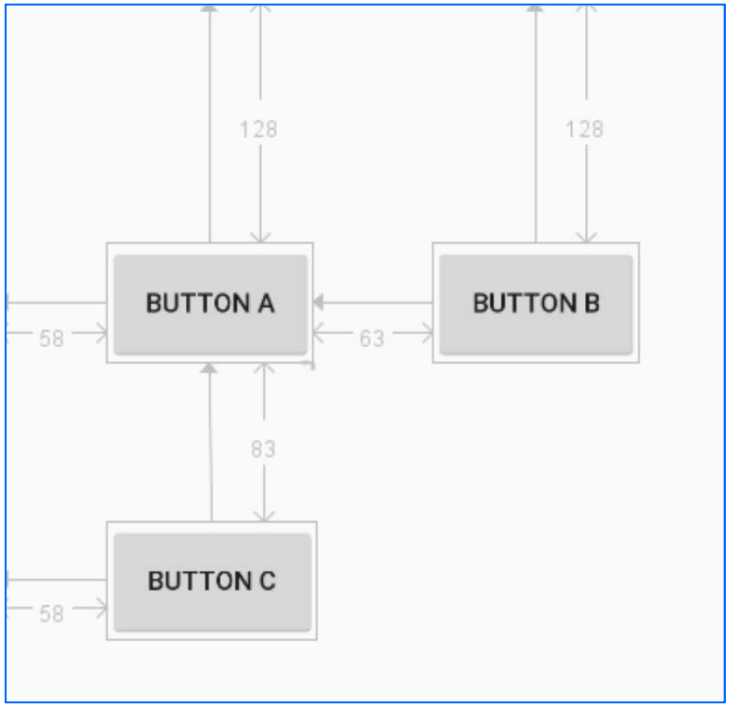
Baseline button : 누르면 baseline이 표시됨.
다른 뷰의 text baseline과 일치시킬 때 사용

Practice #2: Order position

Constraint : Button B는 Button A의 오른쪽에 놓이고,
Button C는 Button A의 밑에 놓여야 한다.

Button C를 Button A 위로 옮기면 어떻게 될까?
Button B를 Button A 왼쪽으로 옮기면 어떻게 될까?



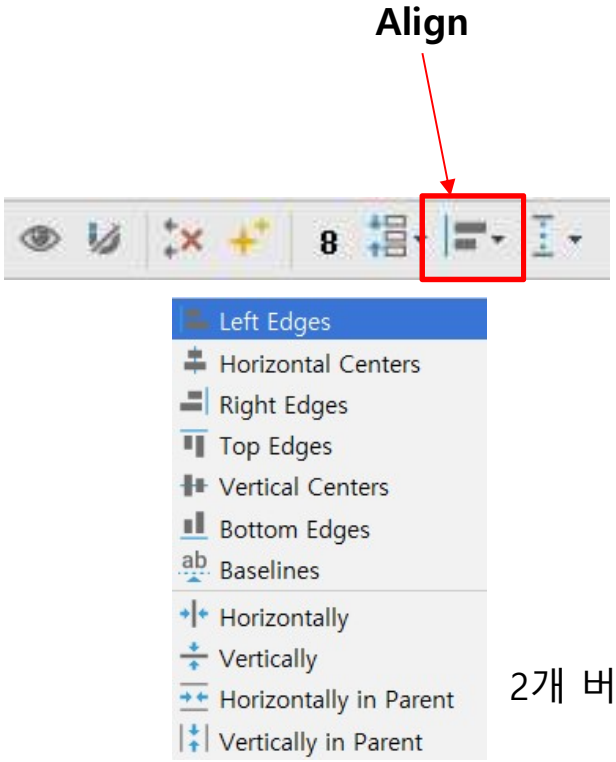


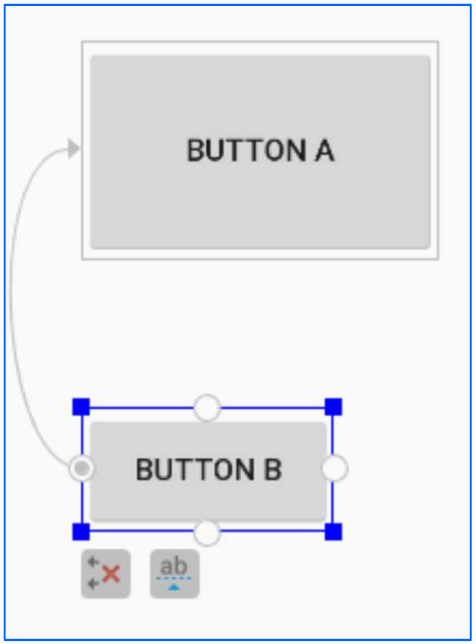
Practice #3: Alignment

Constraint : Button B의 왼쪽 경계선과 Button A의 왼쪽 경계선을 일치시키고 싶다.

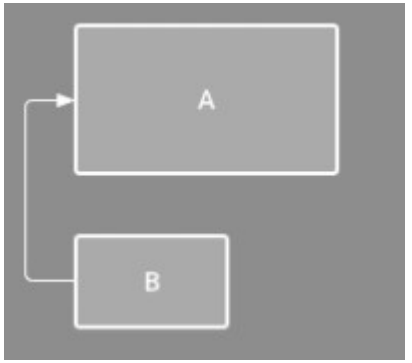
Button B의 위치를 왼쪽으로 옮기면 어떻게 될까?

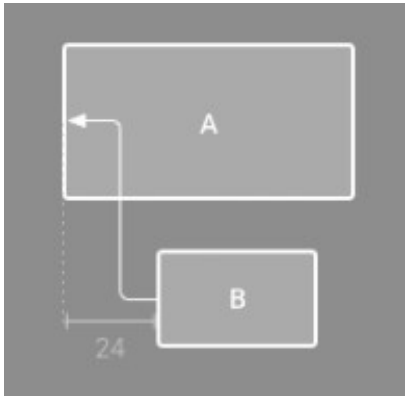
Button B의 위치를 오른쪽으로 옮기면 어떻게 될까?





2개 버튼을 모두 선택 ➔ 오른쪽 버튼
➔ **Align Left Edges**

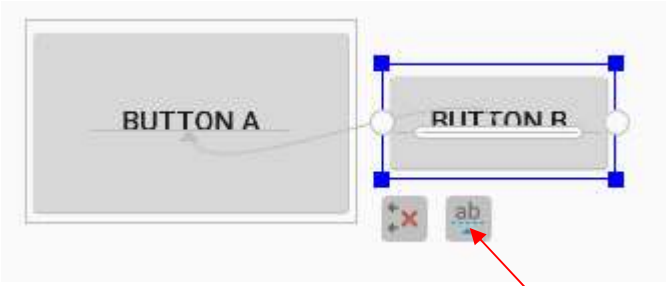


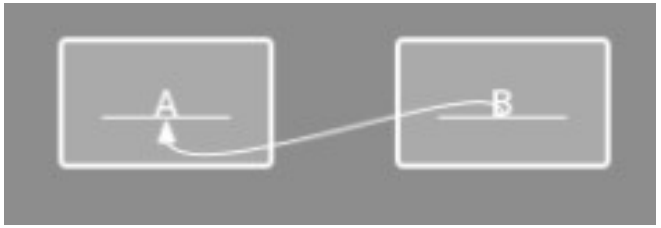


Practice #4: Baseline Alignment

Constraint : 크기가 서로 다른 2개의 Button의 baseline을 일치시키고 싶다.

Button B의 위치를 위로 옮기면 어떻게 될까?
Button B의 위치를 왼쪽, 오른쪽으로 옮기면 어떻게 될까?
Button A의 위치를 위, 아래로 옮기면 어떻게 될까?

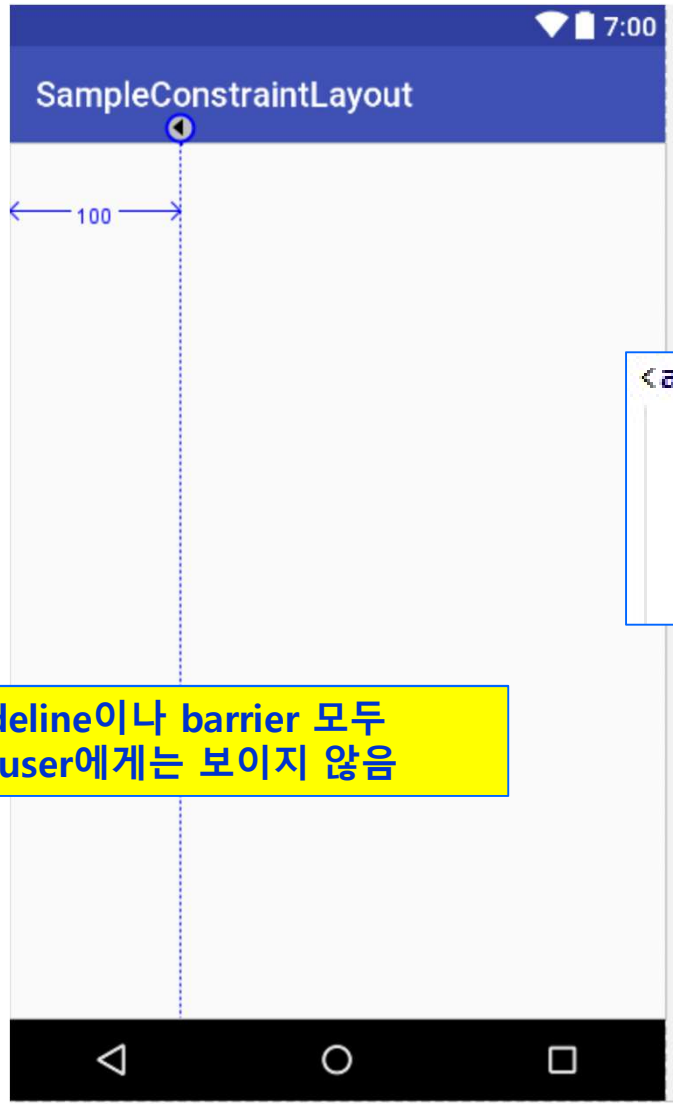


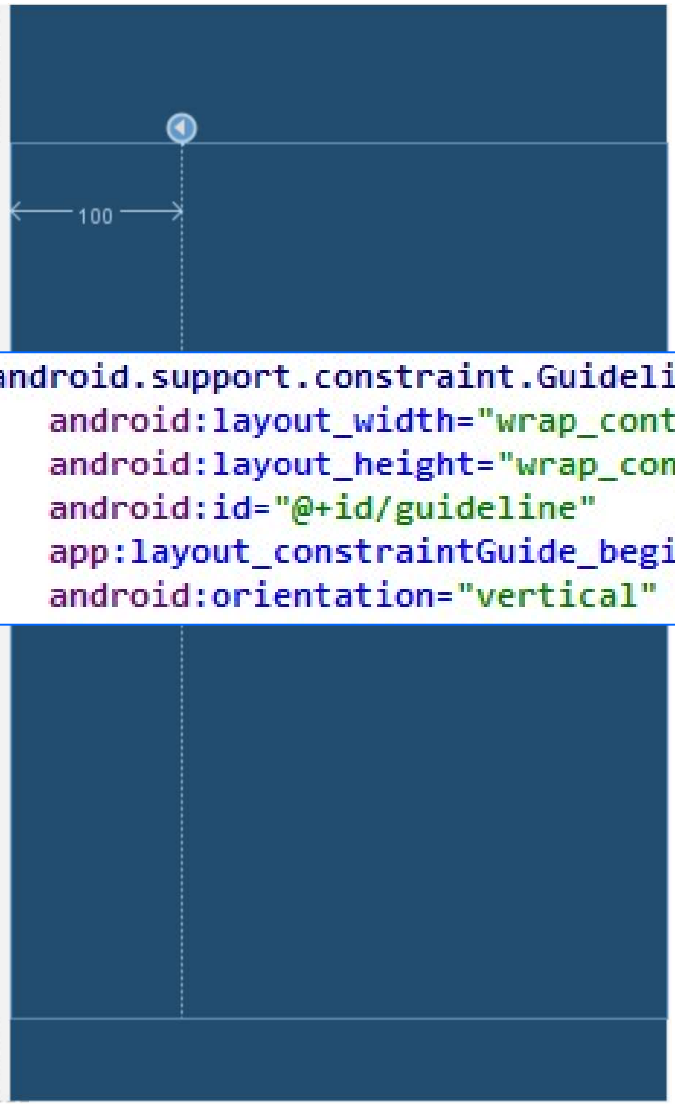


Baseline button을 눌러 baseline을 표시한다.

2개의 baseline을 연결한다.

Practice #5: Constrain to a guideline (1/3)

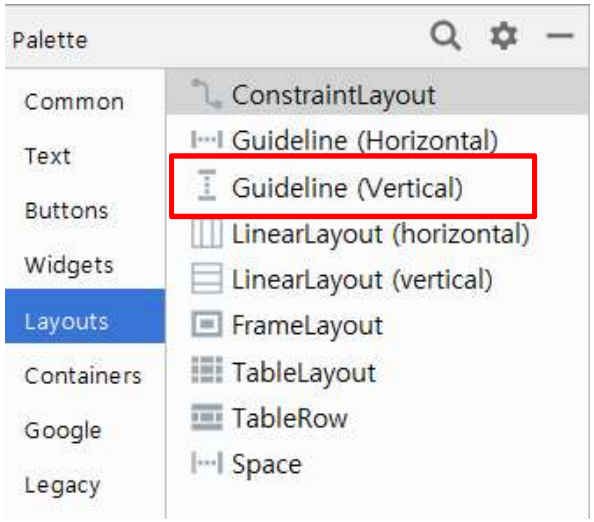




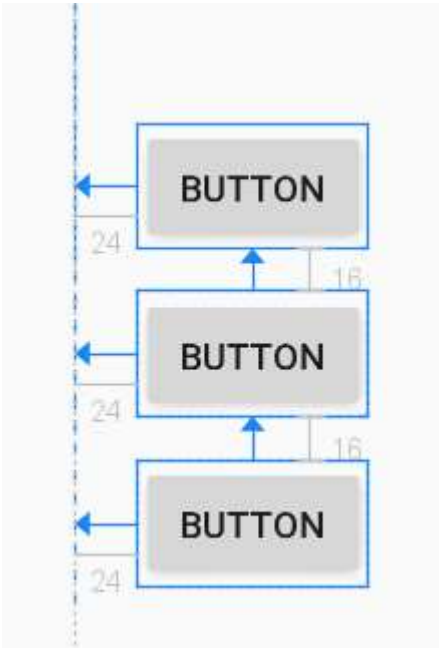

```
<android.support.constraint.Guideline
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/guideline"
    app:layout_constraintGuide_begin="100dp"
    android:orientation="vertical" />
```

guideline이나 barrier 모두
app user에게는 보이지 않음

Practice #5: Constrain to a guideline (2/3)



Layouts에서 Guideline(Vertical) 선택



Guideline 오른쪽에 버튼 3개 배치

Practice #5: Constrain to a guideline (3/3)

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.constraint.Guideline
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/guideline"
        app:layout_constraintGuide_begin="100dp"
        android:orientation="vertical" />

    <Button … />

    <Button … />

    <Button … />

</android.support.constraint.ConstraintLayout>
```

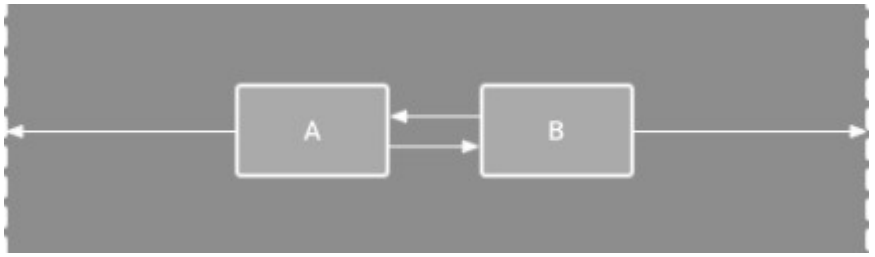
```
<Button
    android:id="@+id/button4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button"
    android:layout_marginLeft="24dp"
    app:layout_constraintLeft_toLeftOf="@+id/guideline"
    app:layout_constraintTop_toTopOf="parent"
    android:layout_marginTop="100dp" />

<Button
    android:id="@+id/button5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button"
    android:layout_marginLeft="24dp"
    app:layout_constraintLeft_toLeftOf="@+id/guideline"
    android:layout_marginTop="16dp"
    app:layout_constraintTop_toBottomOf="@+id/button4" />

<Button
    android:id="@+id/button6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button"
    android:layout_marginLeft="24dp"
    app:layout_constraintLeft_toLeftOf="@+id/guideline"
    android:layout_marginTop="16dp"
    app:layout_constraintTop_toBottomOf="@+id/button5" />
```

Control linear groups with a chain

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



Constraints > horizontal_chainStyle, vertical_chainStyle

Spread

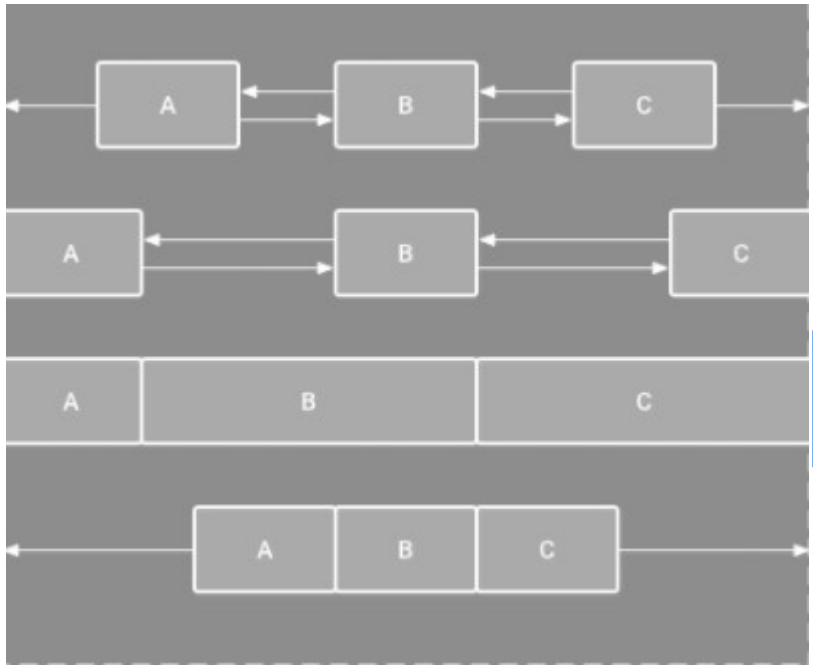
Spread Inside

Constraints > horizontal_weight, vertical_weight

Weighted

Constraints > horizontal_bias, vertical_bias

Packed

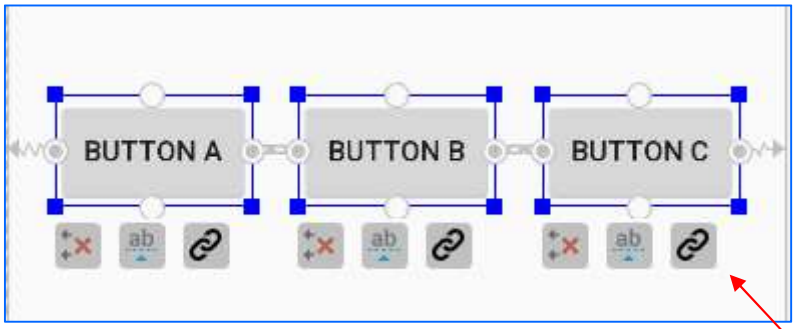


Spread 체인 스타일 일 때 weight 적용 가능

Packed 체인 스타일 일 때 bias 적용 가능

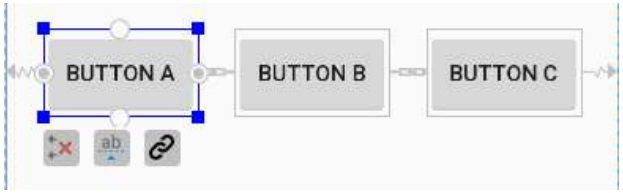
Control linear groups with a chain

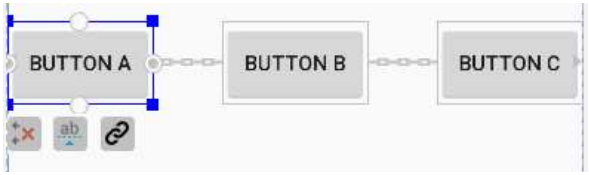
Head widget(BUTTON A) 선택
→ Shift키를 누른 상태로
다른 2개의 버튼 선택



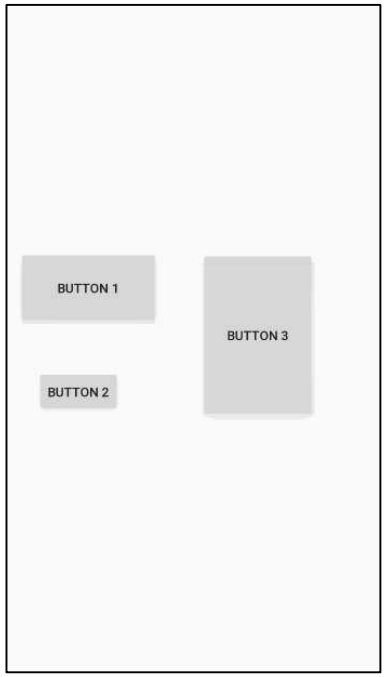
오른쪽 버튼 ➔ **Center Horizontally**

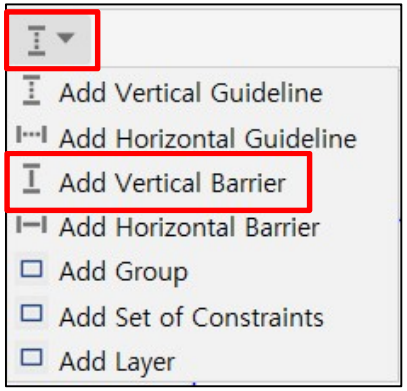
Chain button이 나타남 : 이 버튼을 누를 때마다
Spread ➔ spread inside ➔ packed 상태로 바뀐다.

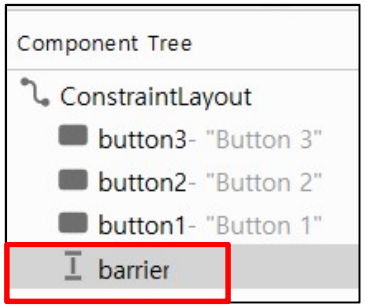


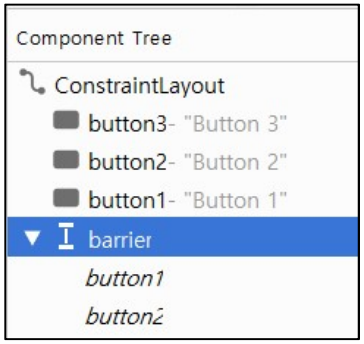


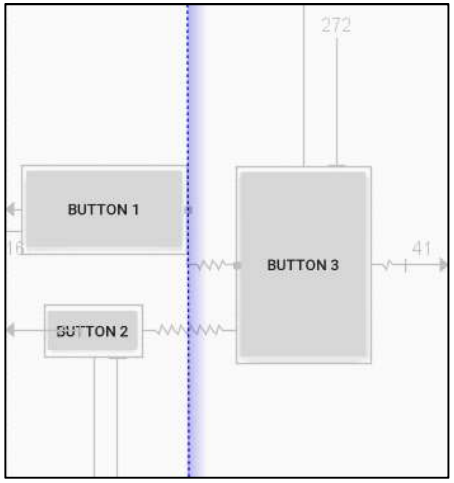
Practice #6: Adding Barriers

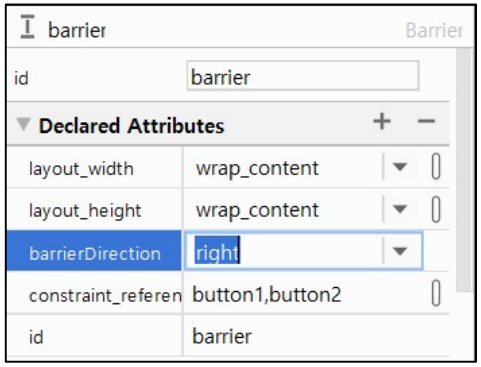








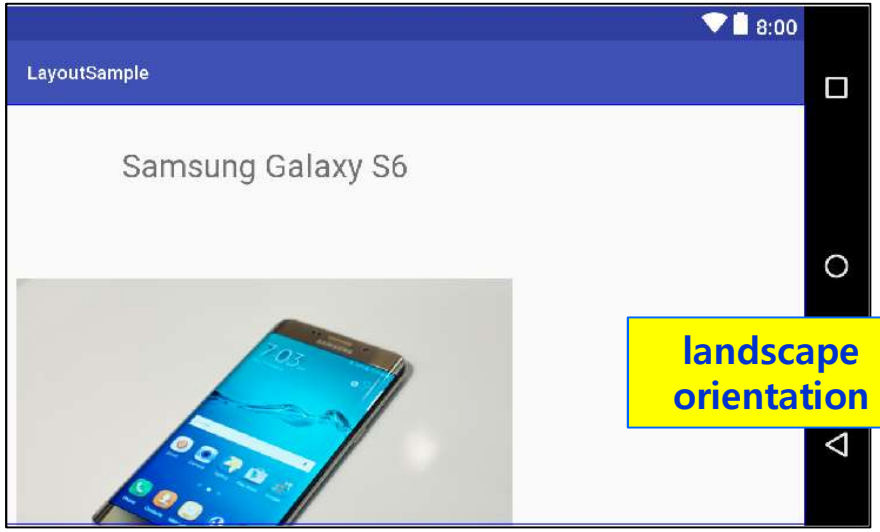




실습 요약

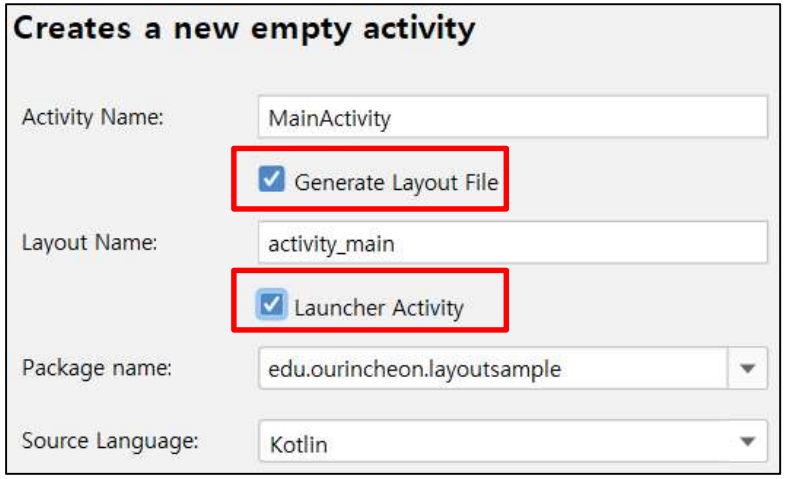
Constraints를 추가하지 않은 상태로
TextView, ImageView, Button(3개)을 배치

Orientation을 Landscape로
바꾸면 어떤 모양이 될까?



실습 준비

- 새 프로젝트 생성
  - **Activity** : **Add No Activity**
  - **Application name** : **Layout Sample**
  - **Minimum API level : API 24** (Nougat)
- Create a New Activity
  - 패키지 > 오른쪽 버튼
    - New > Activity > **Empty Activity**



실습 (1/4) : Add the widgets

Auto connect ➔ off

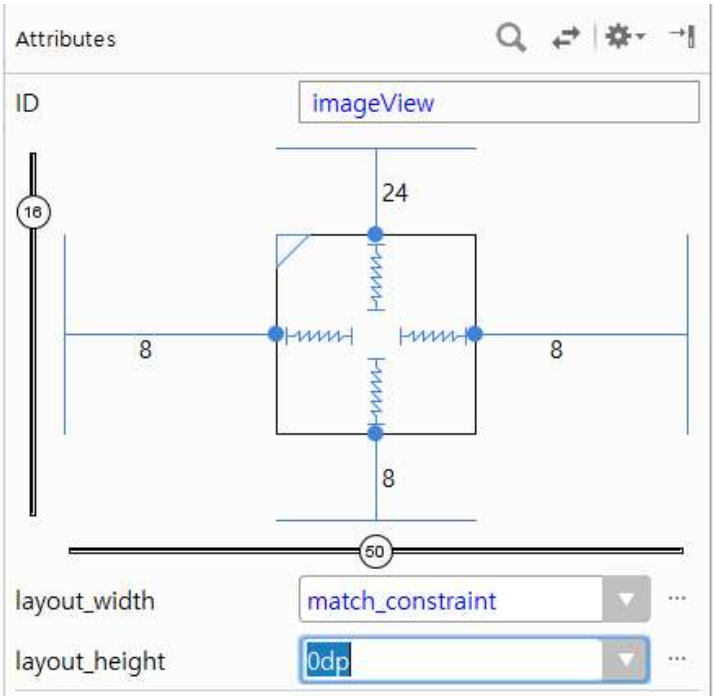
TextView, ImageView,
Button(3개)을
오른쪽 그림처럼 배치

textSize = 24sp
textAlignment = center

View > Tool Windows
> Resource Manager
➔ galaxys6.png ➔ Import

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

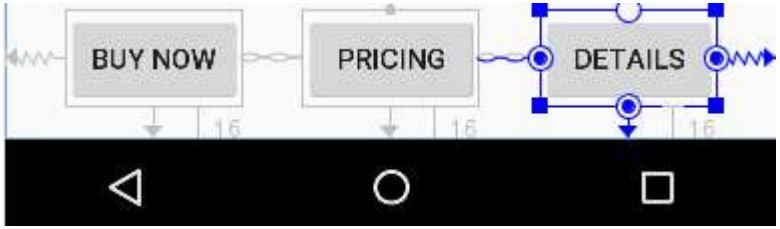
layout_width = **0dp**
layout_height = **0dp**
top margin = 24dp
left, right, bottom margin = 8dp

실습 (3/4) : Create horizontal chain

Auto connect ➔ on

Head widget은 **BUY NOW**
Shift 키를 누른 상태에서
다른 2개 버튼 선택

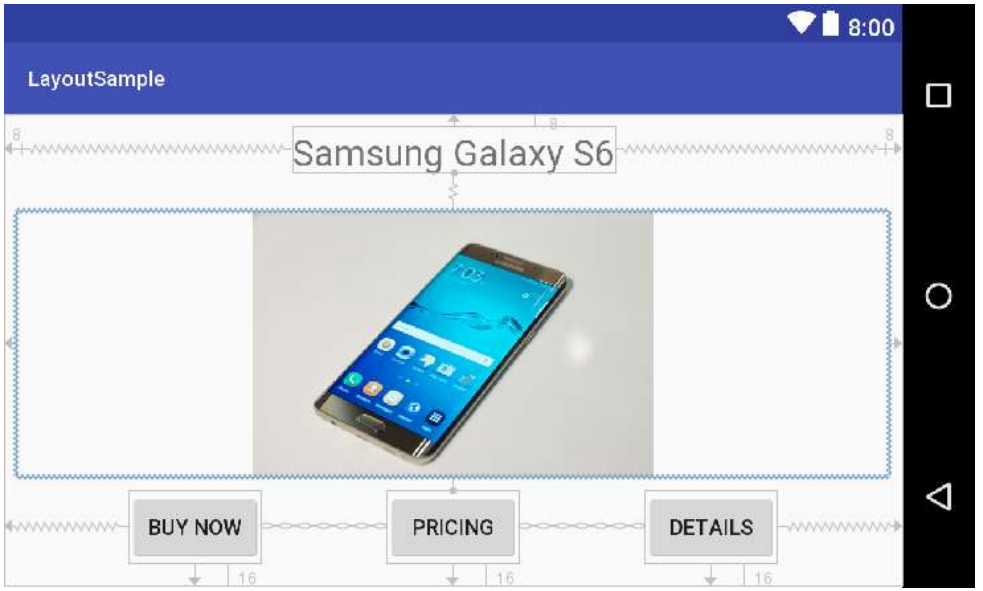
마우스 오른쪽 버튼
➔ Center Horizontally 선택
➔ Horizontal_chainStyle : spread



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

실습 (4/4): Test the layout

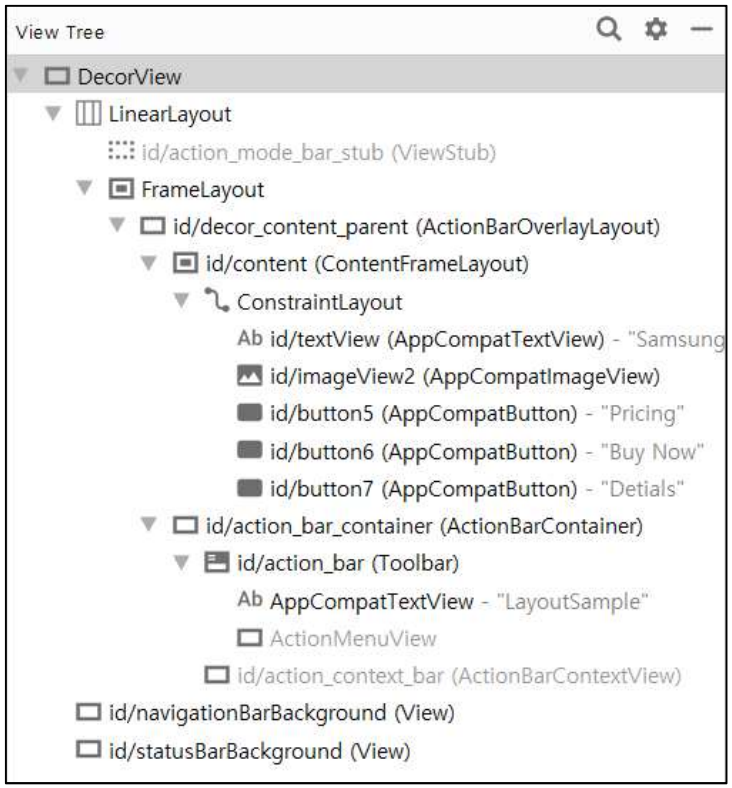
portrait orientation



landscape orientation

Tool > Layout Inspector

에뮬레이터에서 앱이 실행 중이어야 함



Hierarchy of components

Visual representation of the layout design

Property settings