

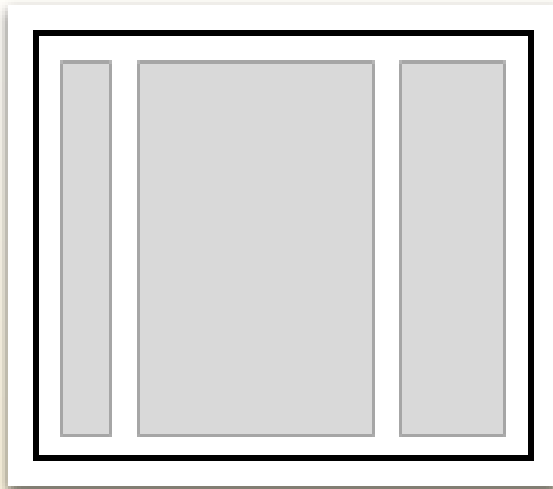


# 01

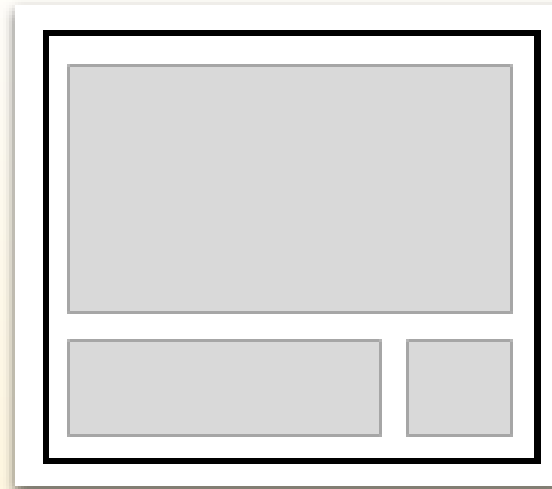
## 레이아웃 – Linear Layout

<https://developer.android.com/guide/topics/ui/layout/linear?hl=ko>

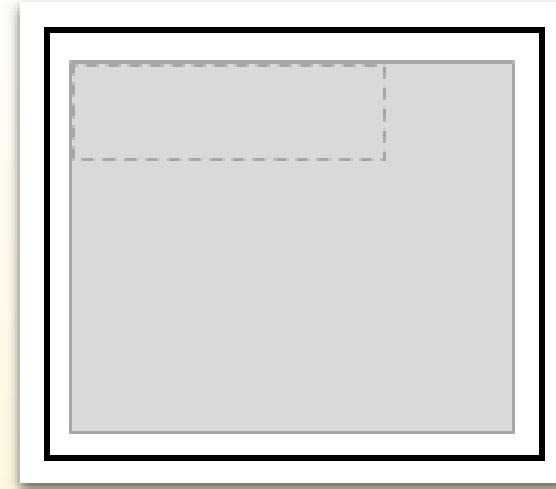
레이아웃 이름	설명
제약 레이아웃 (ConstraintLayout)	제약 조건(Constraint) 기반 모델 제약 조건을 사용해 화면을 구성하는 방법 안드로이드 스튜디오에서 자동으로 설정하는 디폴트 레이아웃
리니어 레이아웃(LinearLayout)	박스(Box) 모델 한 쪽 방향으로 차례대로 뷰를 추가하며 화면을 구성하는 방법 뷰가 차지할 수 있는 사각형 영역을 할당
상대 레이아웃(ReleativeLayout)	규칙(Rule) 기반 모델 부모 컨테이너나 다른 뷰와의 상대적 위치로 화면을 구성하는 방법
프레임 레이아웃(FrameLayout)	싱글(Single) 모델 가장 상위에 있는 하나의 뷰 또는 뷰그룹만 보여주는 방법 여러 개의 뷰가 들어가면 중첩하여 쌓게 됨 가장 단순하지만 여러 개의 뷰를 중첩한 후 각 뷰를 전환하여 보여주는 방식으로 자주 사용함
테이블 레이아웃(TableLayout)	격자(Grid) 모델 격자 모양의 배열을 사용하여 화면을 구성하는 방법 HTML에서 많이 사용하는 정렬 방식과 유사하지만 많이 사용하지는 않음



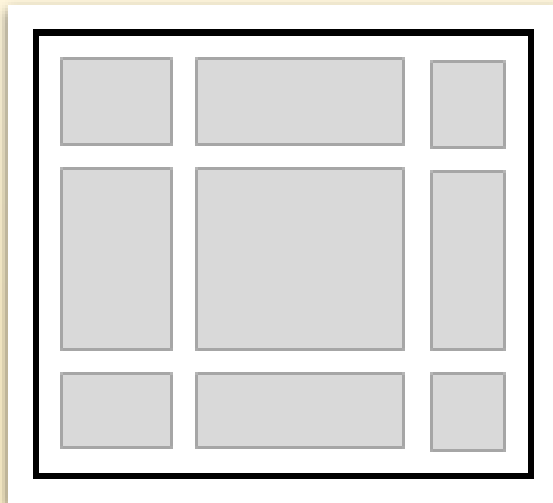
LinearLayout



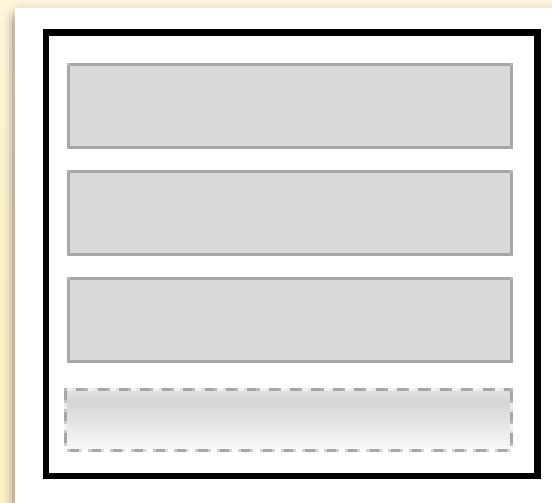
RelativeLayout



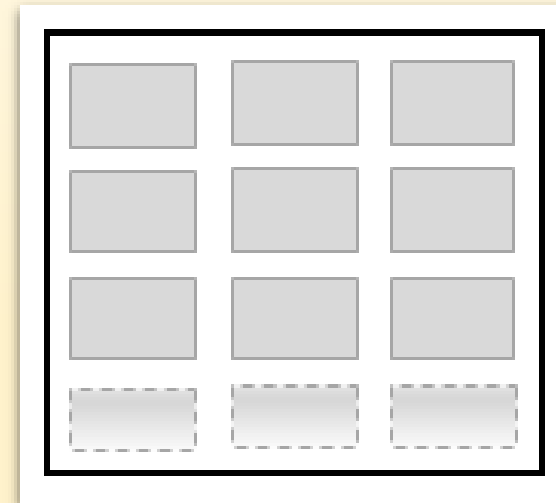
FrameLayout



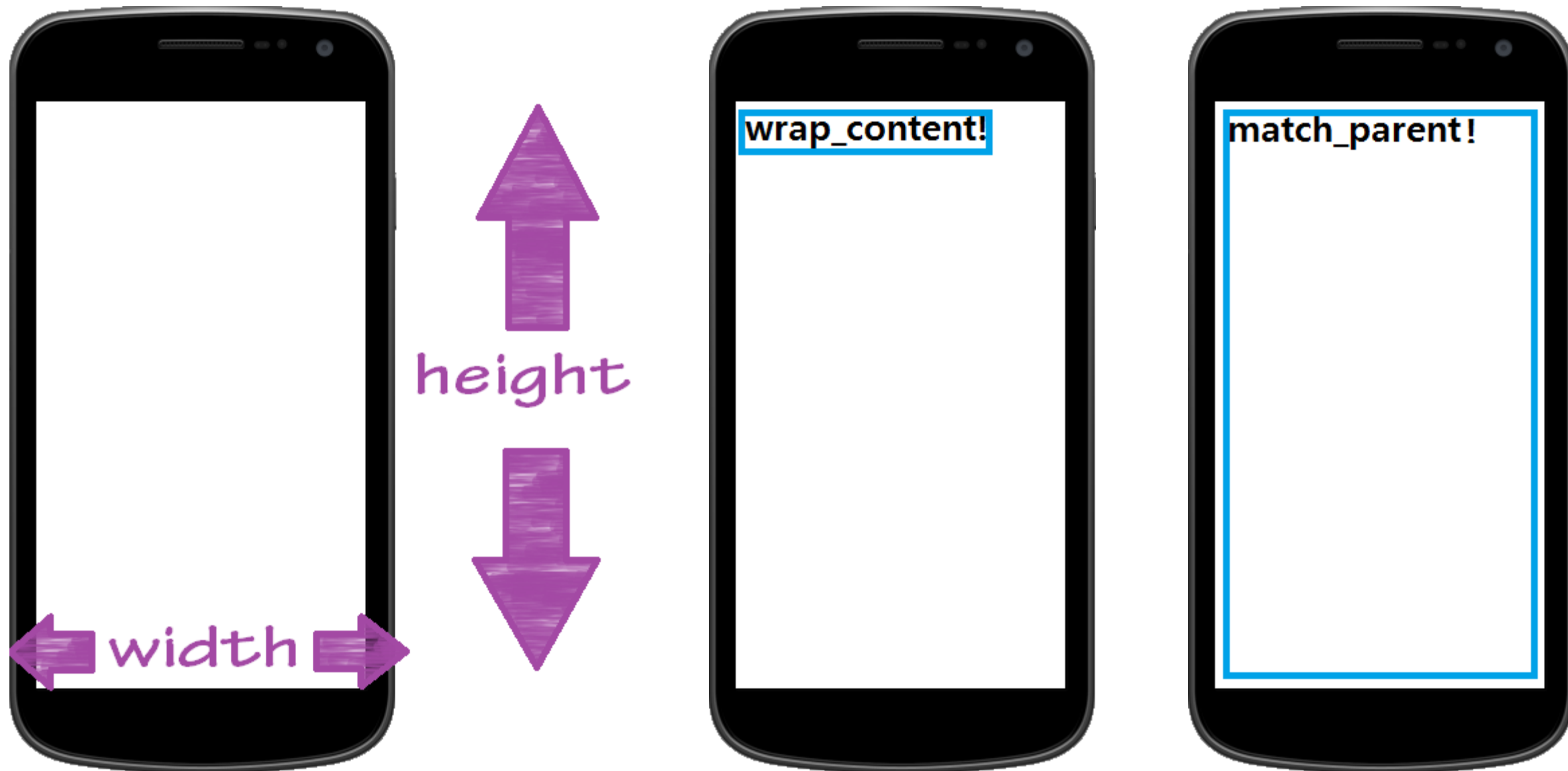
TableLayout

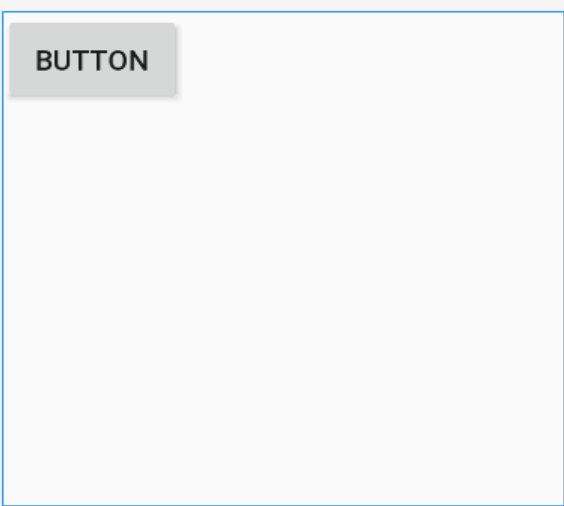
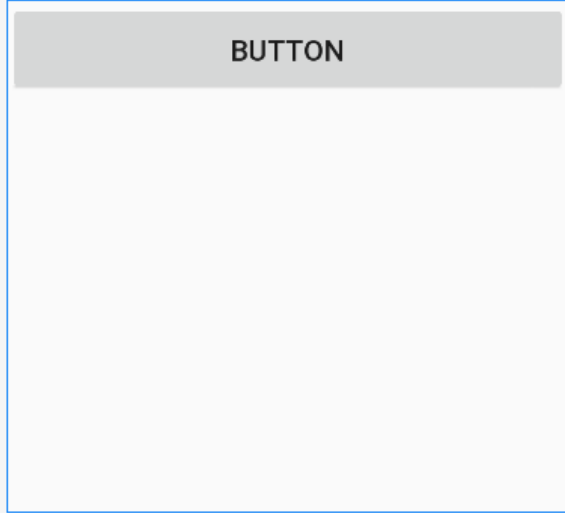
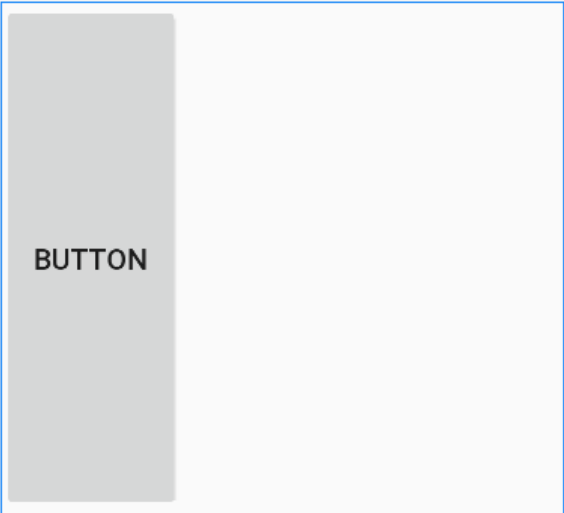
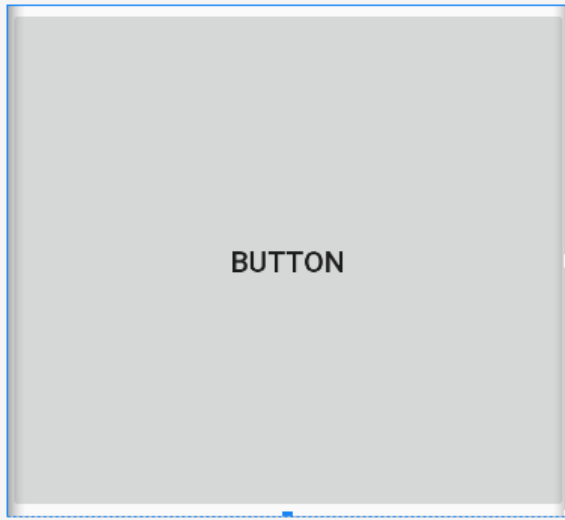


ListView



GridView



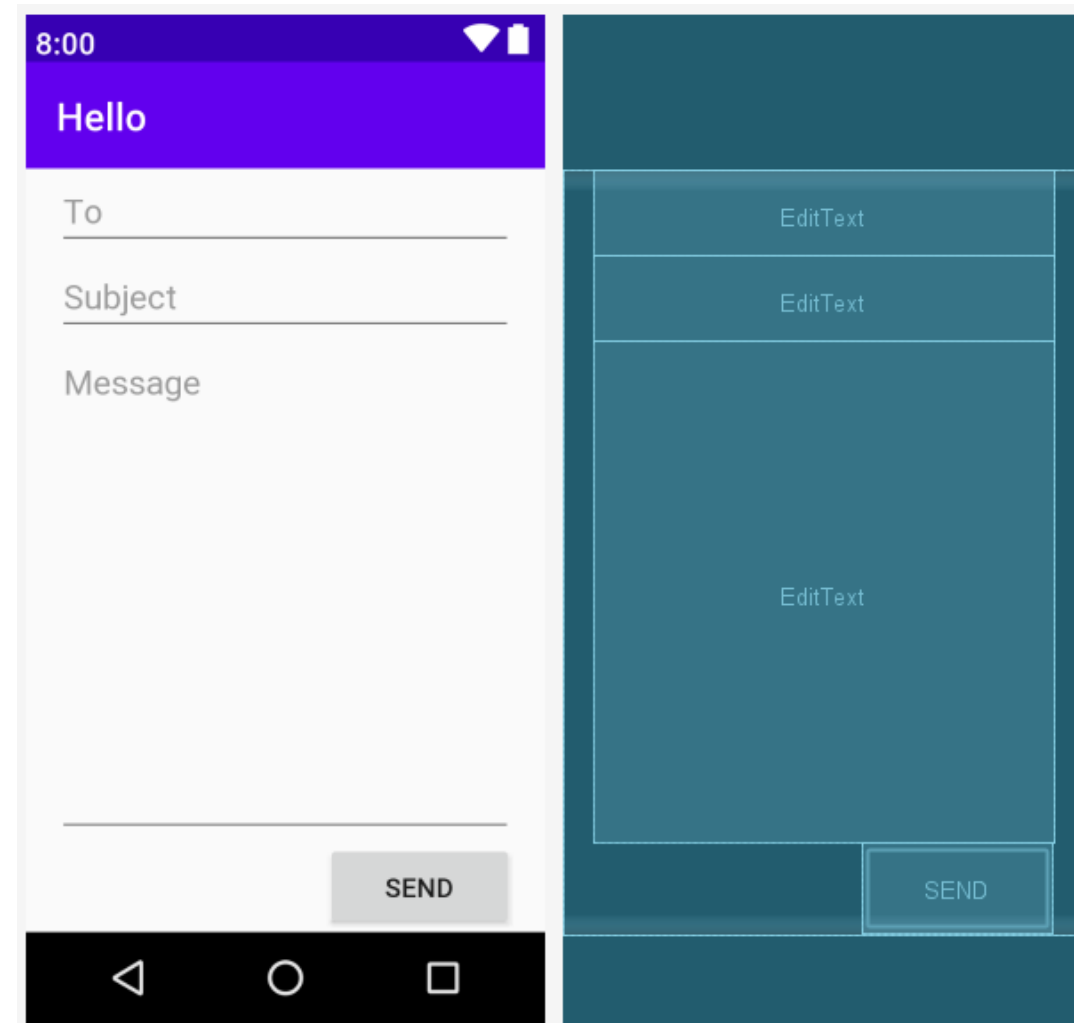
<div>width</div> <div>height</div>	wrap_content	match_parent
wrap_content		
match_parent		

단위	설명
px(pixels)	화면의 실제 픽셀을 나타낸다. 픽셀은 권장되는 단위는 아닌데 그 이유는 장치마다 화면의 밀도가 다르기 때문이다.
dp(density-independent pixels)	dp는 화면의 밀도가 160dpi 화면에서 하나의 물리적인 픽셀을 말한다. 따라서 크기를 160dp로 지정하면 화면의 밀도와는 상관없이 항상 1인치가 된다. dp로 뷰의 크기를 지정하면 화면의 밀도가 다르더라도 항상 동일한 크기로 표시된다.
sp(scale-independent pixels)	화면 밀도와 사용자가 지정한 폰트 크기에 영향을 받아서 변환된다. 이 단위는 폰트 크기를 지정하는 경우에 추천된다.
pt(points)	1/72 인치를 표시한다.
mm(millimeters)	밀리미터를 나타낸다.
in(inches)	인치를 나타낸다.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:orientation="vertical" >
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="To" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Subject" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="top"
        android:hint="Message" />
    <Button
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="right"
        android:text="Send" />
</LinearLayout>

```





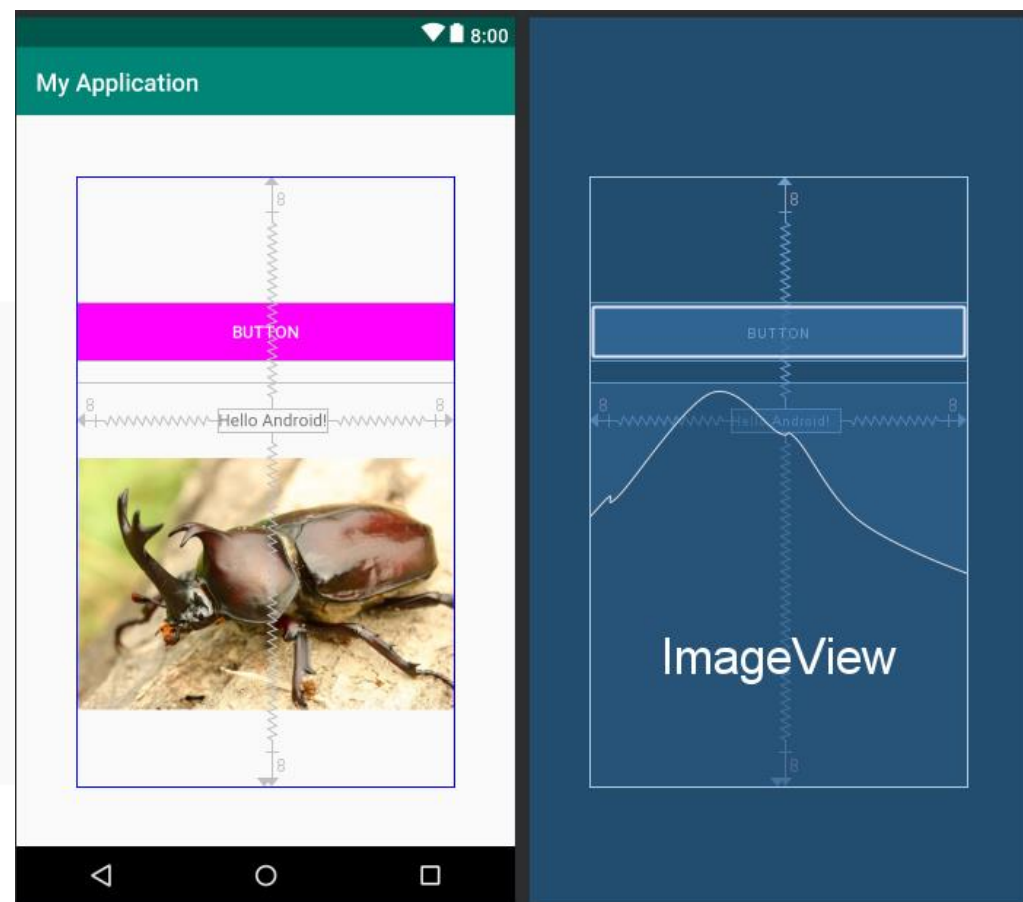
- 뷰가 레이아웃에 추가될 때 보이지 않는 뷰의 테두리(Border)가 생김 -> 뷰의 영역(Box)
- 뷰는 테두리를 기준으로 바깥쪽과 안쪽 공간을 띄움
- 마진(Margin): 테두리 바깥쪽 공간
- 패딩(Padding): 테두리 안쪽 공간

## 마진 값 조정 속성

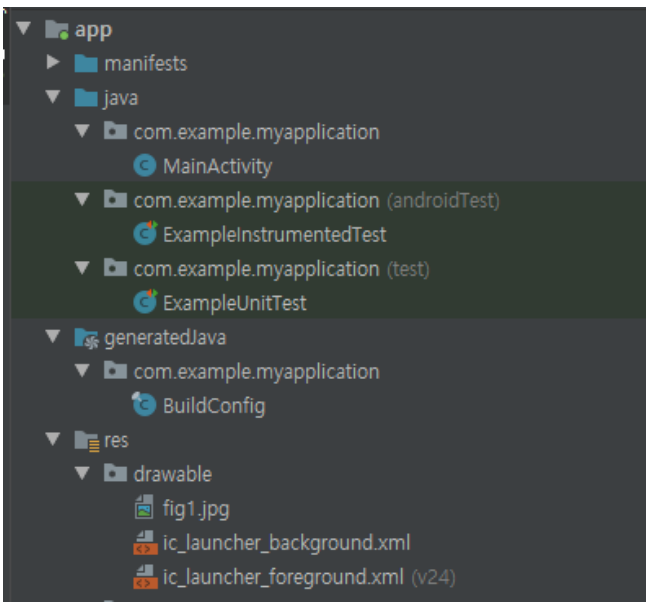
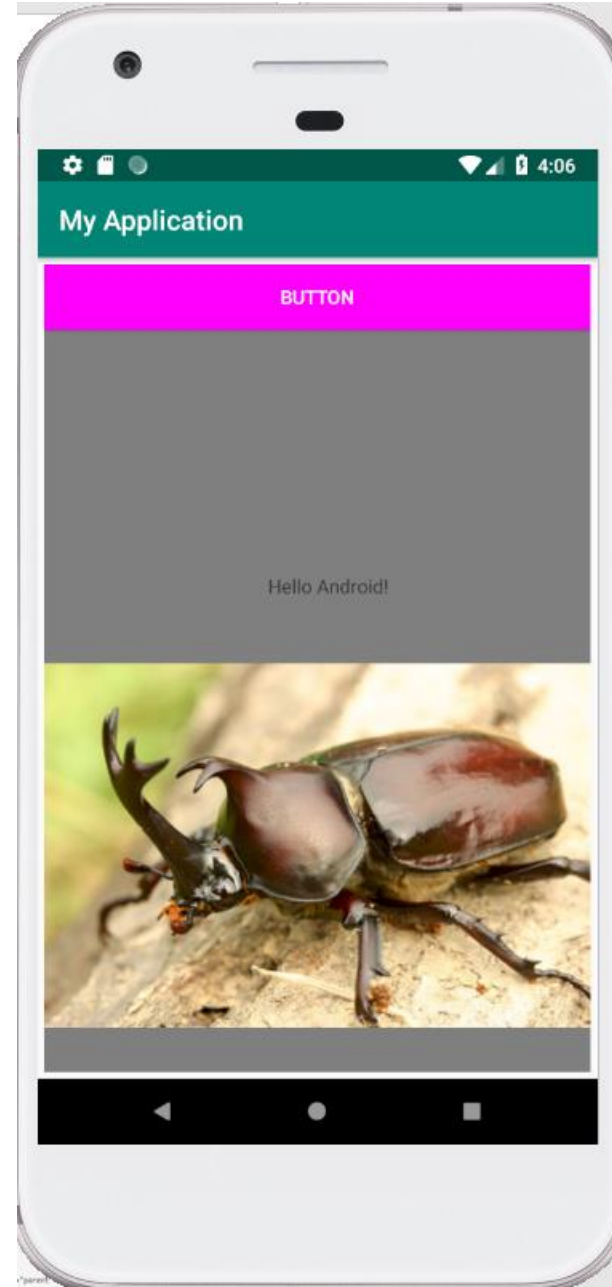
```
layout_margin  
layout_marginTop  
layout_marginBottom  
layout_marginLeft  
layout_marginRight
```

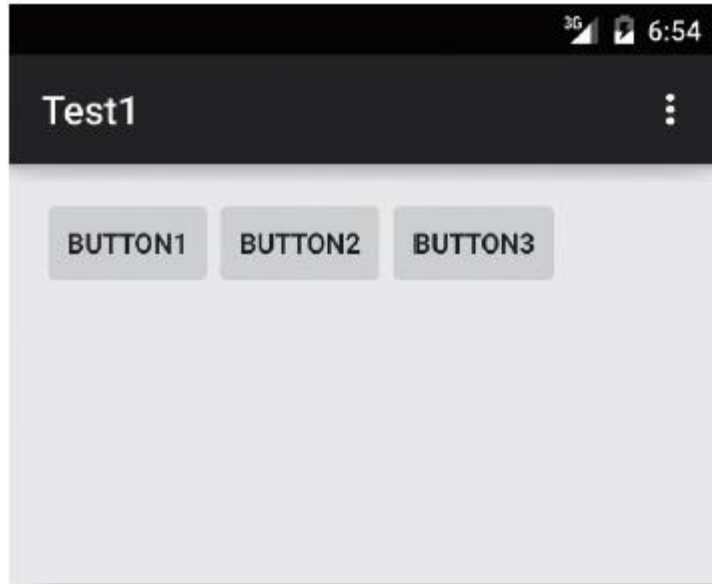
## 패딩 값 조정 속성

```
padding  
paddingTop  
paddingBottom  
paddingLeft  
paddingRight
```

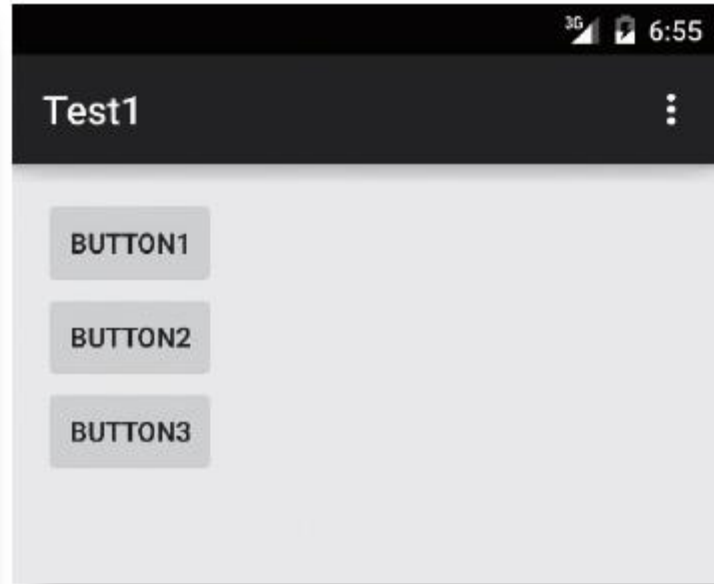


- `background` 속성을 설정하여 배경을 그림
- XML에서 #을 붙이고 ARGB(Alpha, Red, Green, Blue)의 순서대로 색상의 값 기록(16진수)
- `#ff0000`은 빨간색, `#00ff00`은 녹색
- 앞 두 자리 알파를 추가하여 `#ffff0000`은 빨간색, `#88ff0000`은 반투명 빨간색
- 이미지도 배경으로 지정할 수 있음
- 이미지 파일은 `/res/drawable`에 있음 이후 XML에서 `android:background="@drawable/imaegname"`





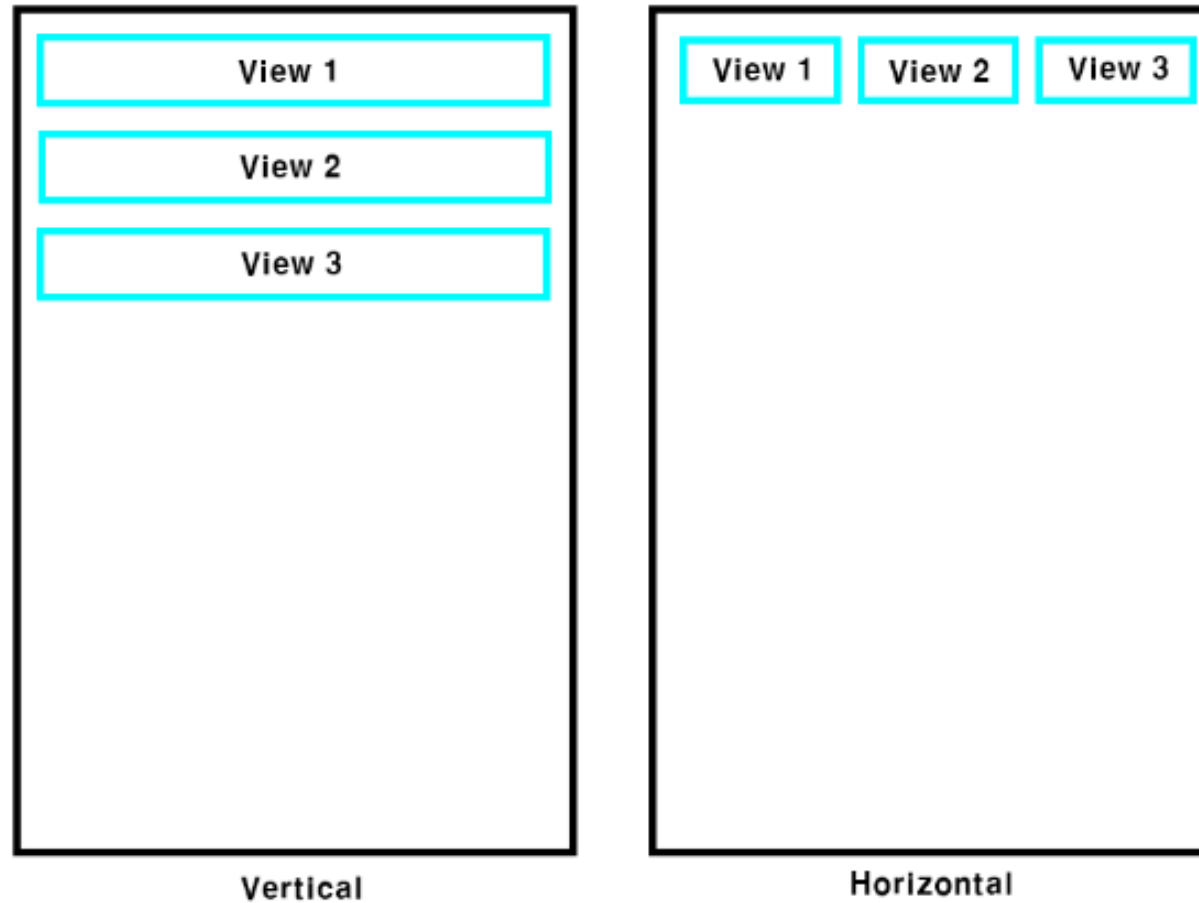
(a) 수평 배치



(b) 수직 배치

**android** : orientation = "**vertical**" : 하위 뷰들을 수직방향으로 배치

**android** : orientation = "**horizontal**" : 하위 뷰들을 수평방향으로 배치

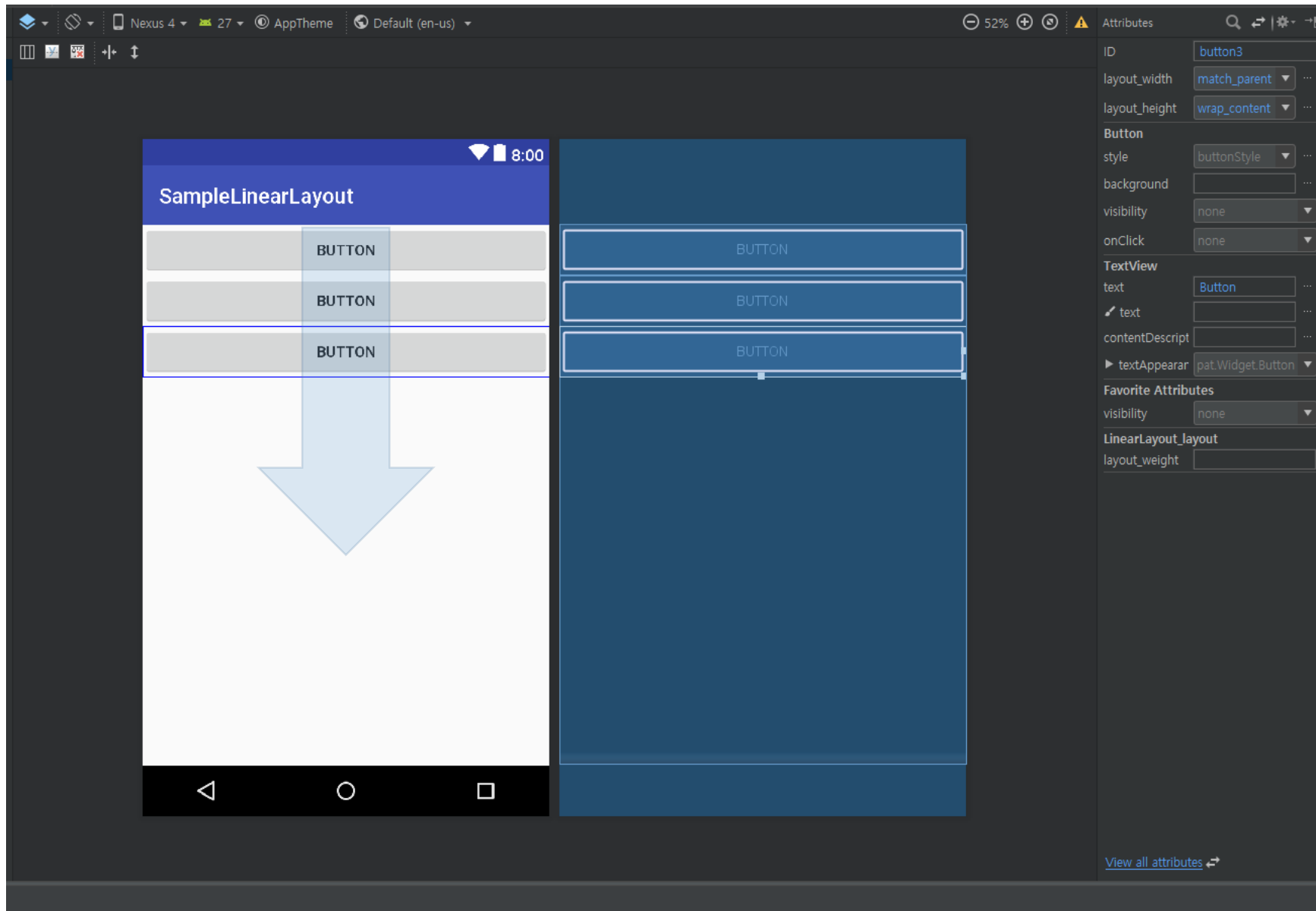


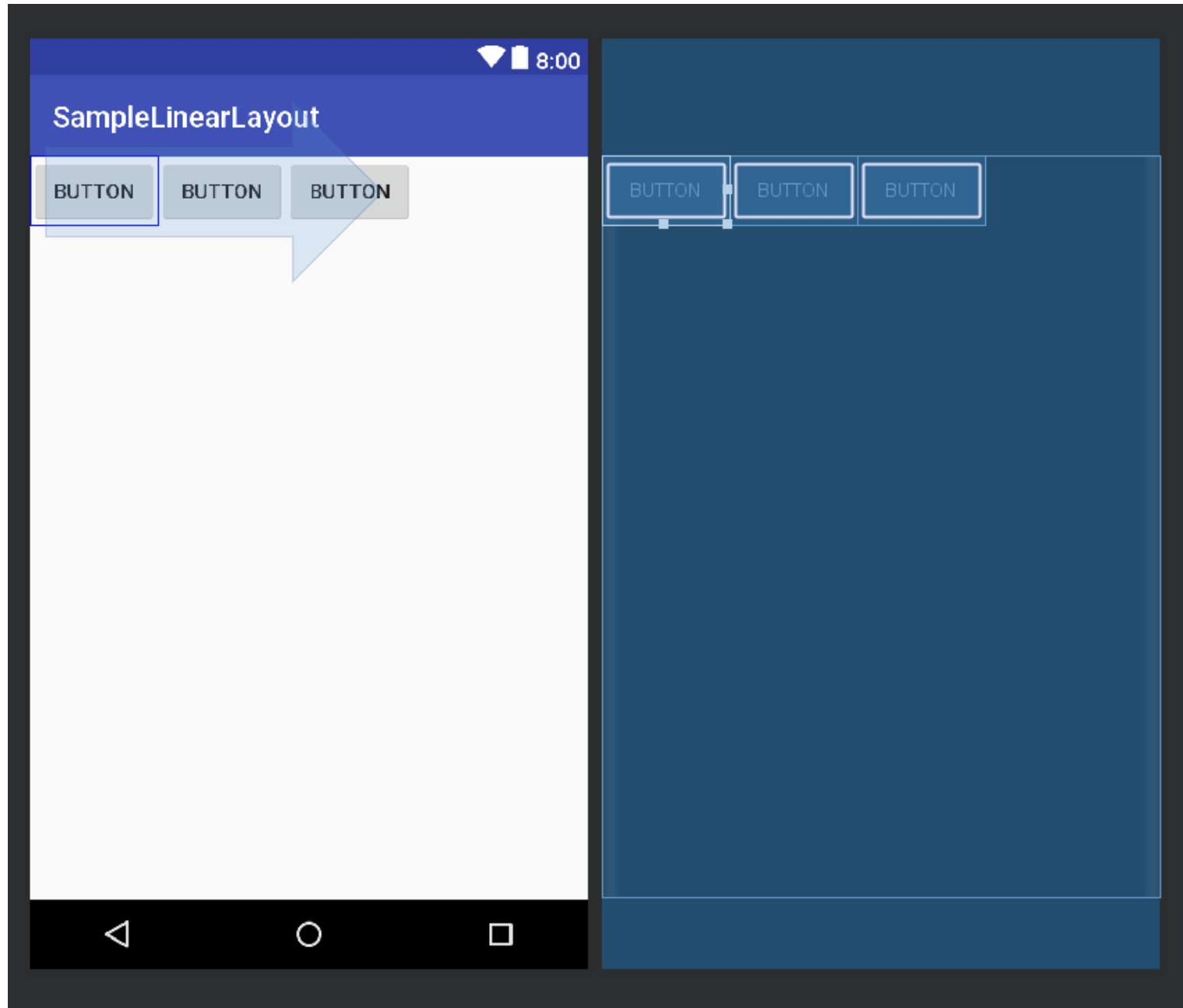
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >

</LinearLayout>
```

Design 탭으로 가서 버튼 세 개 추가, `layout_width` 속성은 `match_parent`, `layout_height` 속성은 `wrap_content`





```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:orientation="horizontal" ←
```

자식을 수평으로 배치

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent">
```

```
    <Button
```

```
        android:id="@+id/button01"
```

```
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
```

```
        android:text="버튼 1"/>
```

```
    <Button
```

```
        android:id="@+id/button02"
```

```
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
```

```
        android:text="버튼 2"/>
```

```
    <Button
```

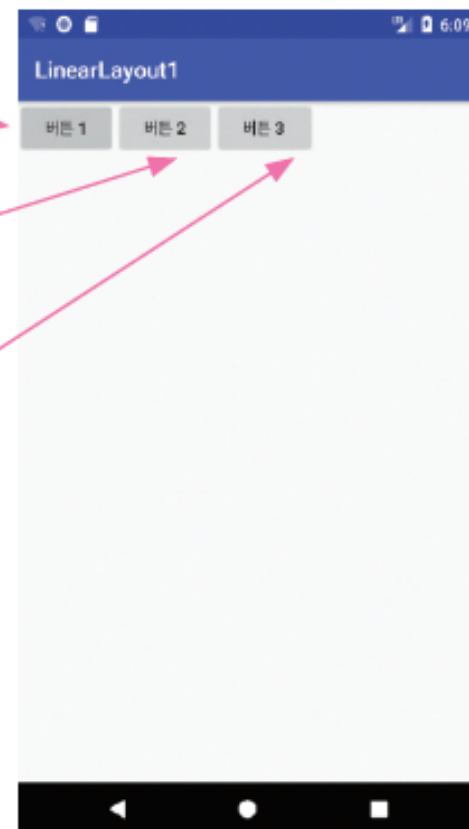
```
        android:id="@+id/button03"
```

```
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
```

```
        android:text="버튼 3"/>
```

```
</LinearLayout>
```





속성	관련 메소드	설명
orientation	setOrientation(int)	“horizontal”은 수평으로, “vertical”은 수직으로 배치한다.
gravity	setGravity(int)	x축과 y축 상에 자식을 어떻게 배치할 것인지를 지정한다.
baselineAligned	setBaselineAligned (boolean)	false로 설정되면 자식뷰들의 기준선을 정렬하지 않는다.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:orientation="vertical"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    android:gravity="center"
```

```
>
```

```
    <Button
```

```
        android:id="@+id/button01"
```

```
        android:layout_width="wrap_content"
```

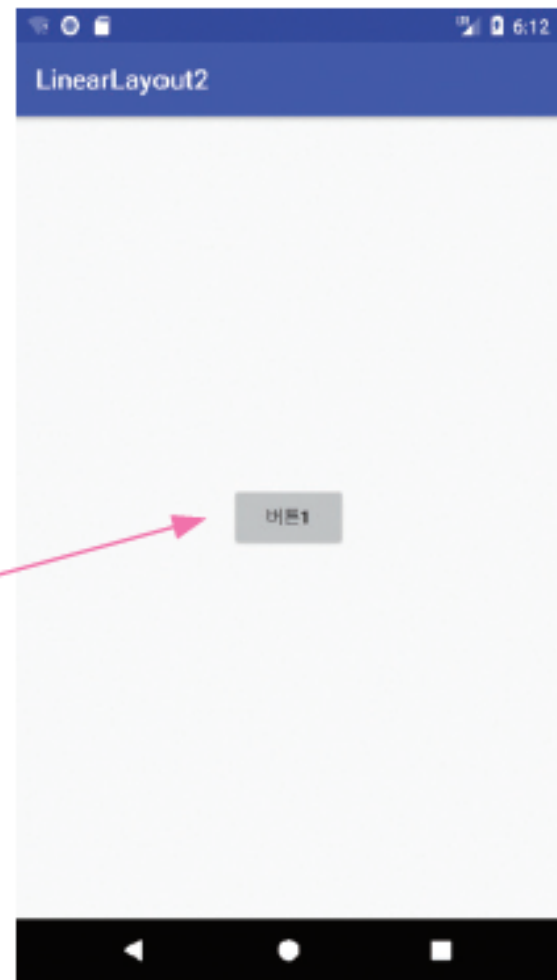
```
        android:layout_height="wrap_content"
```

```
        android:text="버튼1"
```

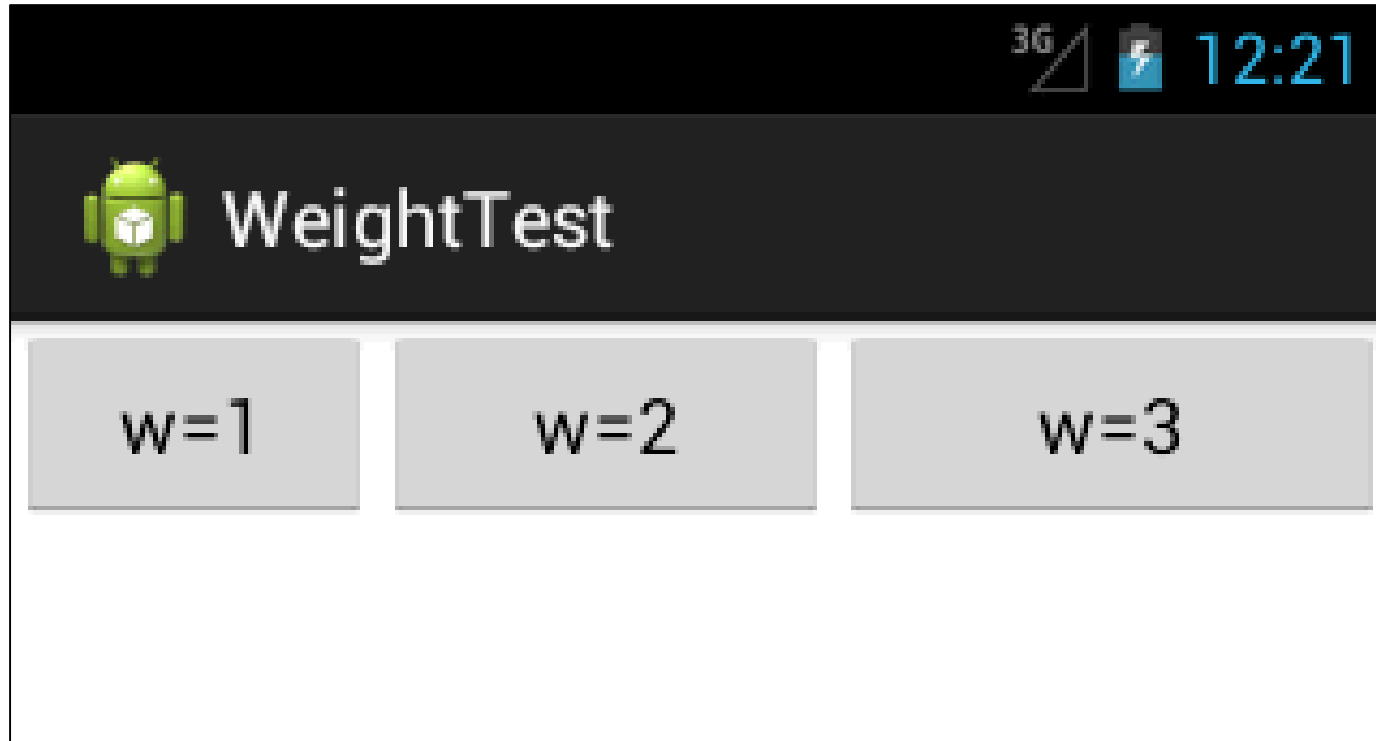
```
    />
```

```
</LinearLayout>
```

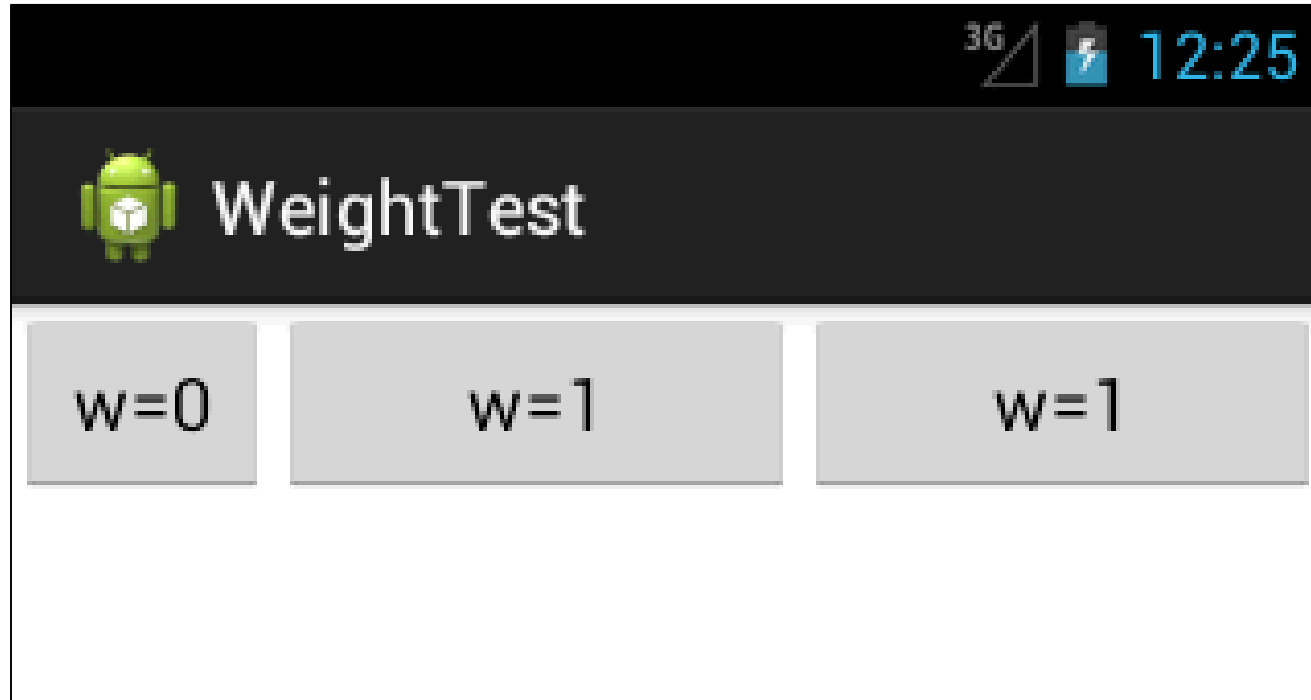
"자식뷰를 중앙에 배치할  
것"이라는 의미이다.



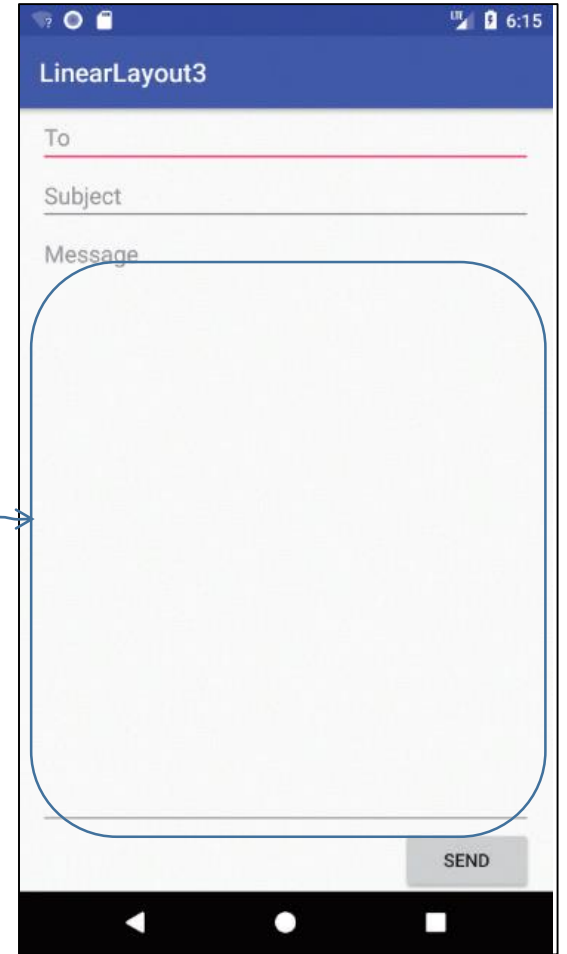
- 선형 레이아웃의 자식 뷰들의 가중치가 각각 1, 2, 3이면, 남아있는 공간의 1/6, 2/6, 3/6을 각각 할당받는다.



- 가중치를 1로 선언한 2개의 텍스트 뷰들은 남아있는 공간을 동일하게 차지할 것이다.



- 버튼, 텍스트 뷰, 에디트 텍스트 등의 뷰들을 가중치를 다르게 하여 배치한 예
- 에디트 텍스트만 가중치가 1이고 나머지는 전부 0



```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:orientation="vertical" >

```

가중치가 디폴트로 0이 된다.

```

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="To" />

```

```

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Subject" />

```

```

<EditText
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:gravity="top"
    android:hint="Message" />

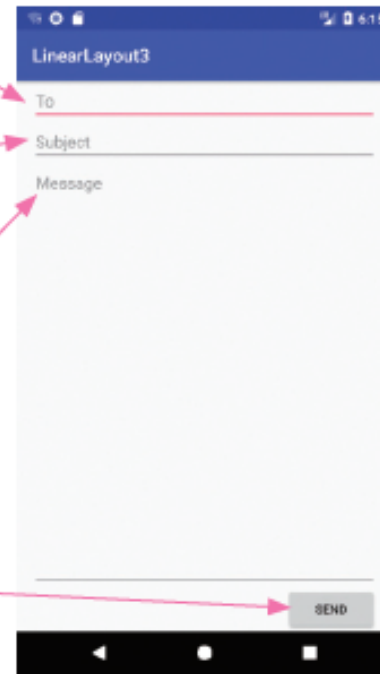
```

가중치를 1로 설정

```

<Button
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:layout_gravity="right"
    android:text="Send" />

```

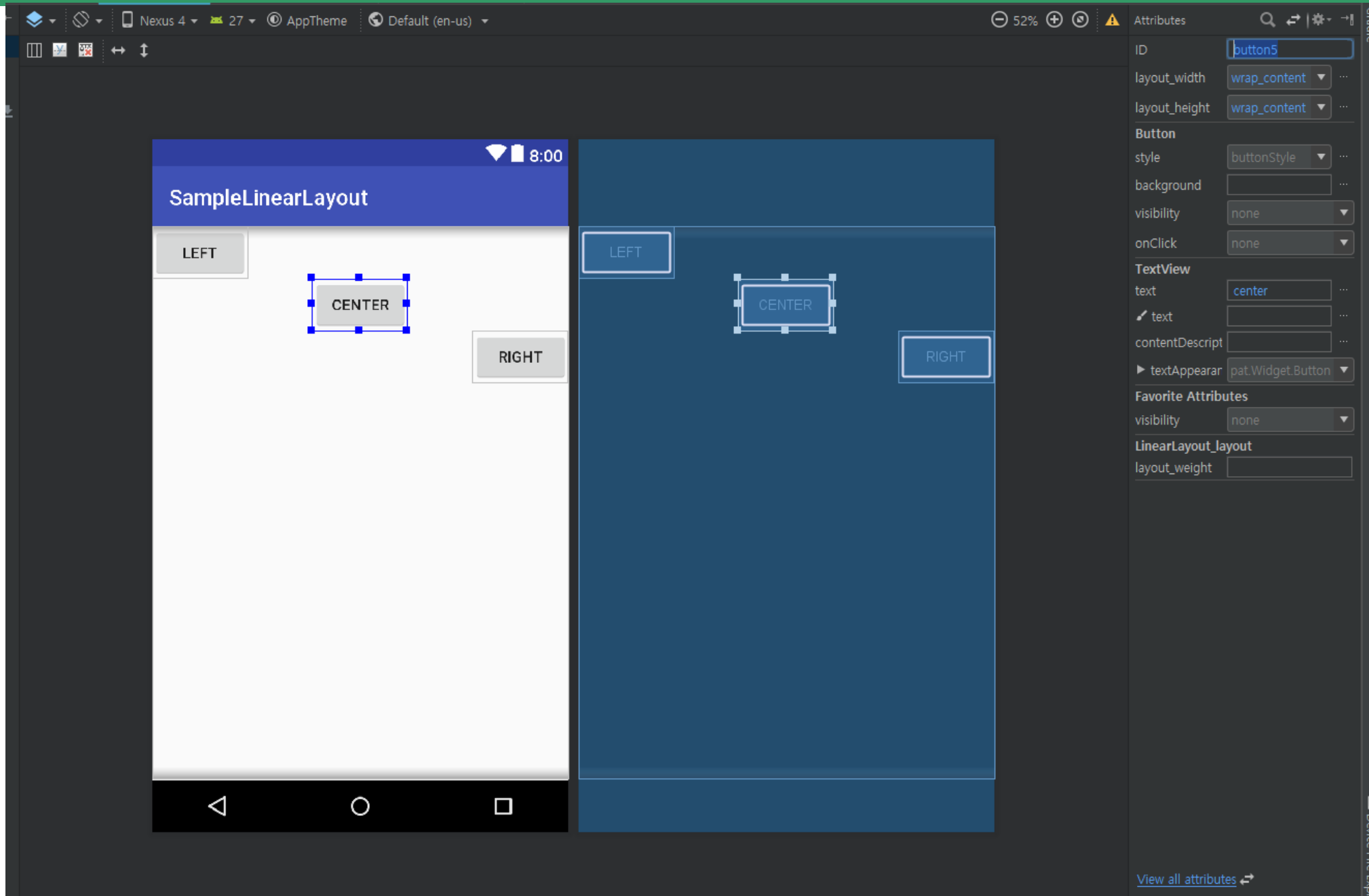


```

</LinearLayout>

```

상수	값	설명
top	0x30	객체를 컨테이너의 상단에 배치, 크기를 변경하지 않음
bottom	0x50	객체를 컨테이너의 하단에 배치, 크기를 변경하지 않음
left	0x03	객체를 컨테이너의 좌측에 배치, 크기를 변경하지 않음
right	0x05	객체를 컨테이너의 우측에 배치, 크기를 변경하지 않음
center_vertical	0x10	객체를 컨테이너의 수직의 중앙에 배치, 크기를 변경하지 않음
fill_vertical	0x70	객체를 컨테이너의 수직을 채우도록 배치
center_horizontal	0x01	객체를 컨테이너의 수평의 중앙에 배치, 크기를 변경하지 않음
fill_horizontal	0x07	객체를 컨테이너의 수평을 채우도록 배치
center	0x11	객체를 컨테이너의 수평, 수직의 중앙에 배치
fill	0x77	객체가 컨테이너를 가득 채우도록 배치





<!-- 텍스트뷰 안의 글자를 왼쪽으로 정렬 -->

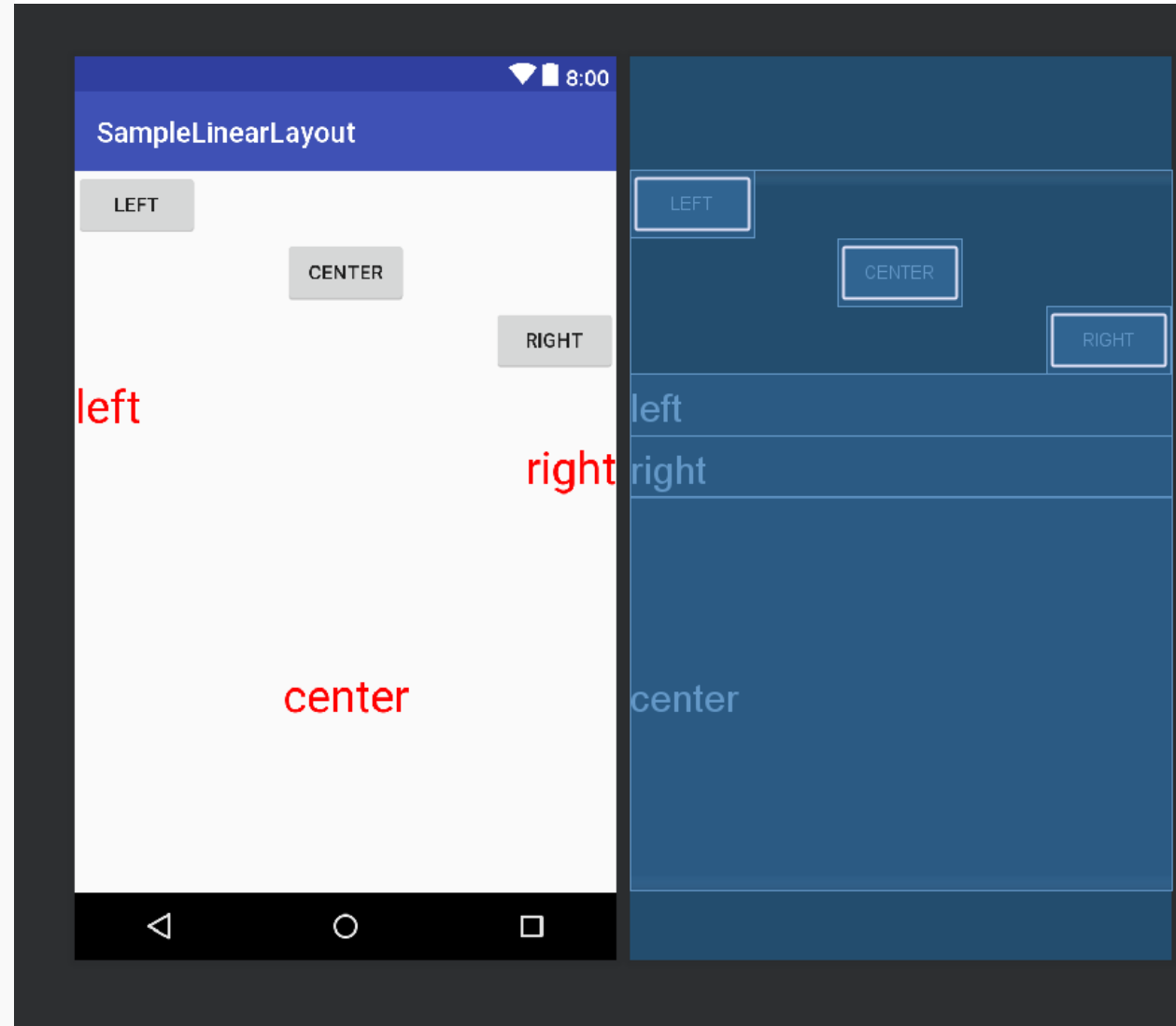
```
<TextView
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="left"
    android:textColor="#ffff0000"
    android:textSize="32dp"
    android:text="left"/>
```

<!-- 텍스트뷰 안의 글자를 오른쪽으로 정렬 -->

```
<TextView
    android:id="@+id/textView2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="right"
    android:textColor="#ffff0000"
    android:textSize="32dp"
    android:text="right"/>
```

<!-- 텍스트뷰 안의 글자를 가로와 세로의 가운데로 정렬 -->

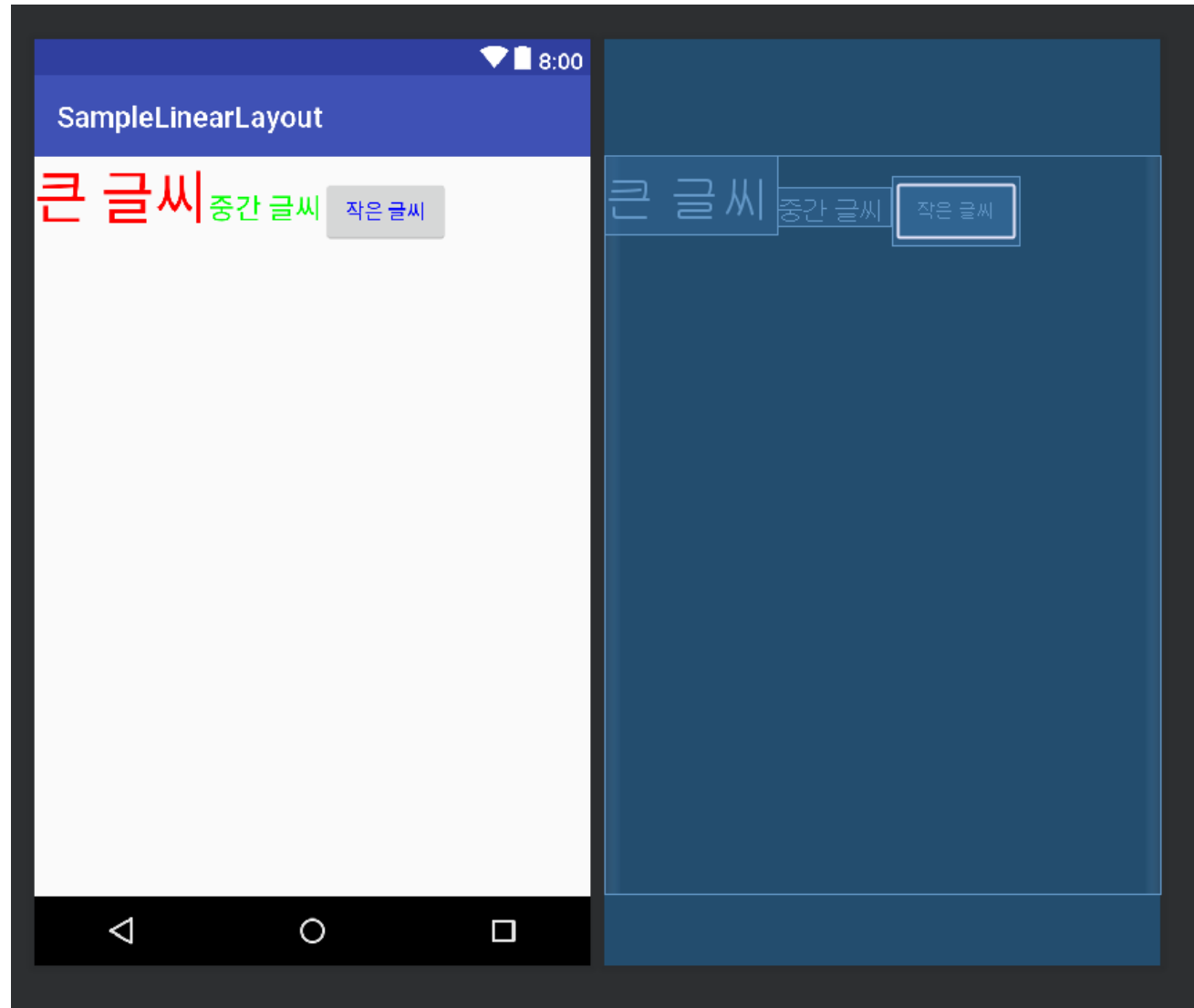
```
<TextView
    android:id="@+id/textView3"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_horizontal|center_vertical"
    android:textColor="#ffff0000"
    android:textSize="32dp"
    android:text="center" />
```

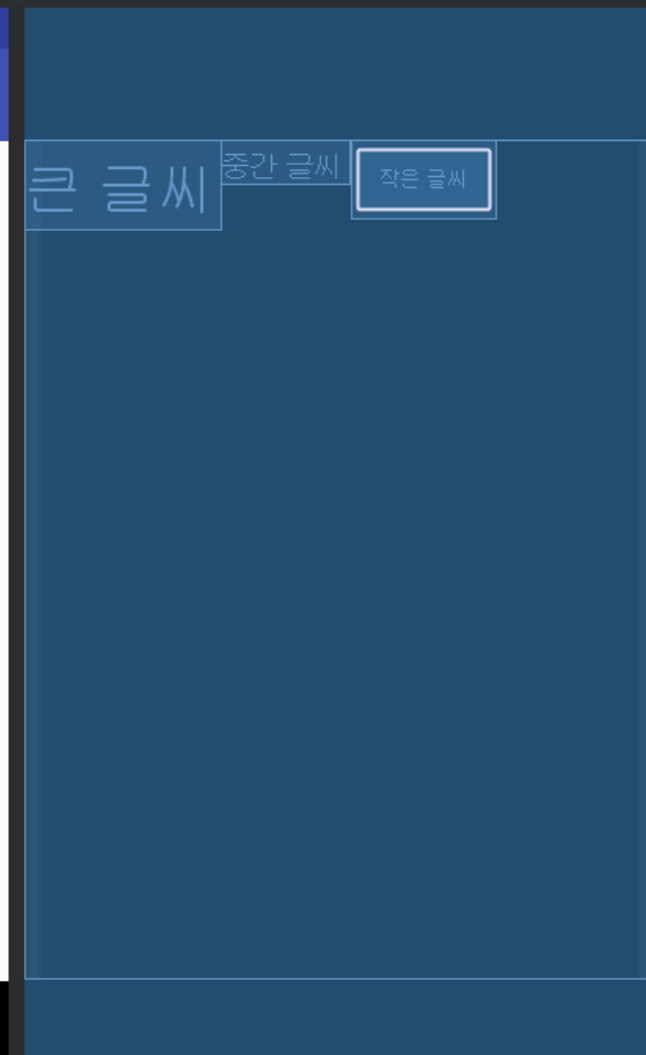
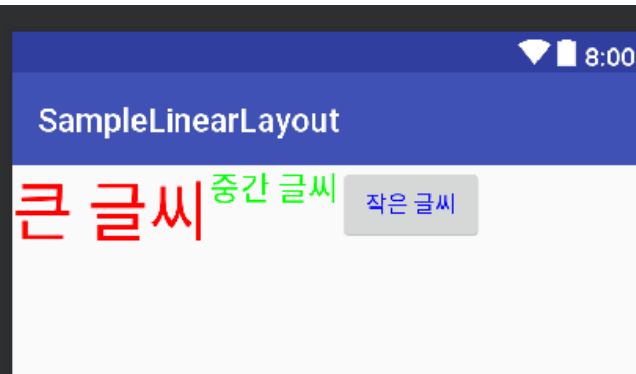


```
<!-- 텍스트뷰의 글자 크기를 크게 -->
<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="큰 글씨"
    android:textColor="#ffff0000"
    android:textSize="40dp"
/>
```

```
<!-- 텍스트뷰의 글자 크기를 작게 -->
<TextView
    android:id="@+id/textView5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="중간 글씨"
    android:textColor="#ff00ff00"
    android:textSize="20dp"/>
```

```
<!-- 버튼의 글자 크기를 좀 더 작게 -->
<Button
    android:id="@+id/button7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="작은 글씨"
    android:textColor="#ff0000ff"
    android:textSize="14dp" />
```





```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:baselineAligned="false"
    android:orientation="vertical">
```



```

<TextView
    android:id="@+id/textView6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="텍스트"
    android:textColor="#ffff0000"
    android:textSize="24dp"
    android:background="#ffffff00"
android:padding="20dp"/>

```

<!-- 텍스트뷰 위젯과 부모 여윌 공간 사이의 여백을 10dp로 설정 -->

```

<TextView
    android:id="@+id/textView7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="텍스트"
    android:textColor="#ffff0000"
    android:textSize="24dp"
    android:background="#ff00ffff"
android:layout_margin="10dp"/>

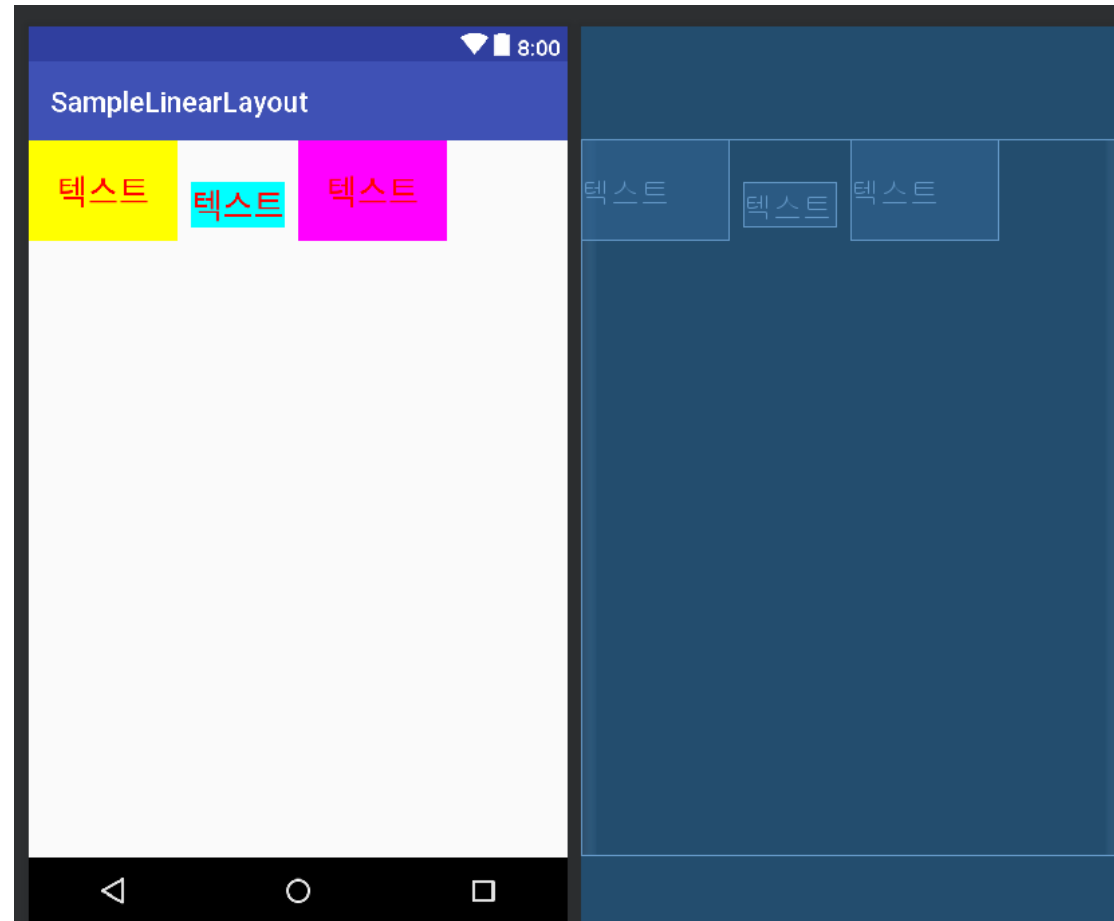
```

<!-- 텍스트뷰 내부의 여백을 20dp로 설정 -->

```

<TextView
    android:id="@+id/textView8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="텍스트"
    android:textColor="#ffff0000"
    android:textSize="24dp"
    android:background="#ffff00ff"
android:padding="20dp"/>

```



```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <!-- 리니어 레이아웃을 수평 방향으로 설정 -->
    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        >

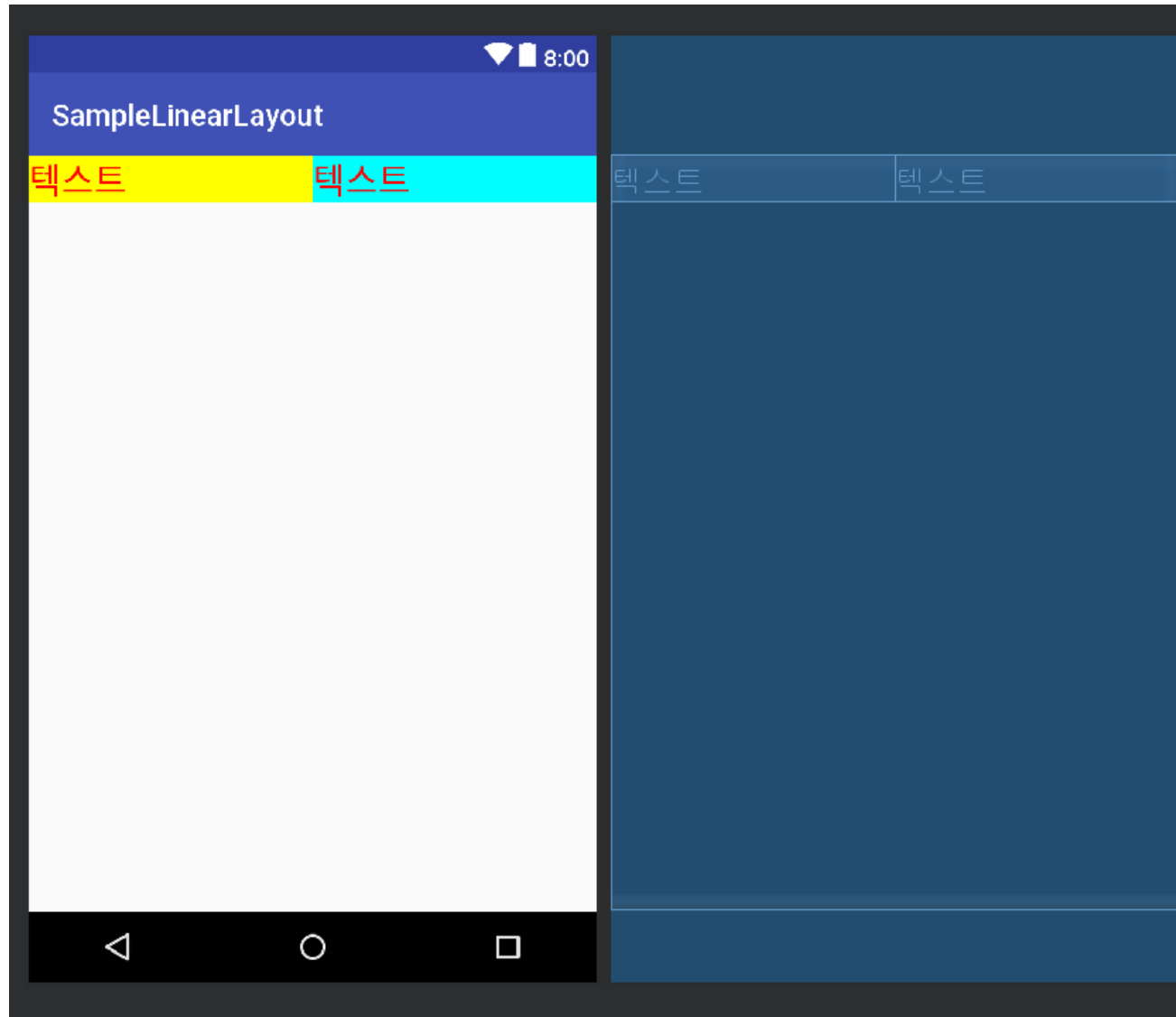
        <!-- layout_weight를 1로 설정 -->
        <TextView
            android:id="@+id/textView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:background="#ffffff00"
            android:text="텍스트"
            android:textColor="#ffff0000"
            android:textSize="24dp"
            android:layout_weight="1"/>

        <!-- layout_weight를 1로 설정 -->
        <TextView
            android:id="@+id/textView2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:background="#ff00ffff"
            android:text="텍스트"
            android:textColor="#ffff0000"
            android:textSize="24dp"
            android:layout_weight="1"/>

    </LinearLayout>

</LinearLayout>

```



```

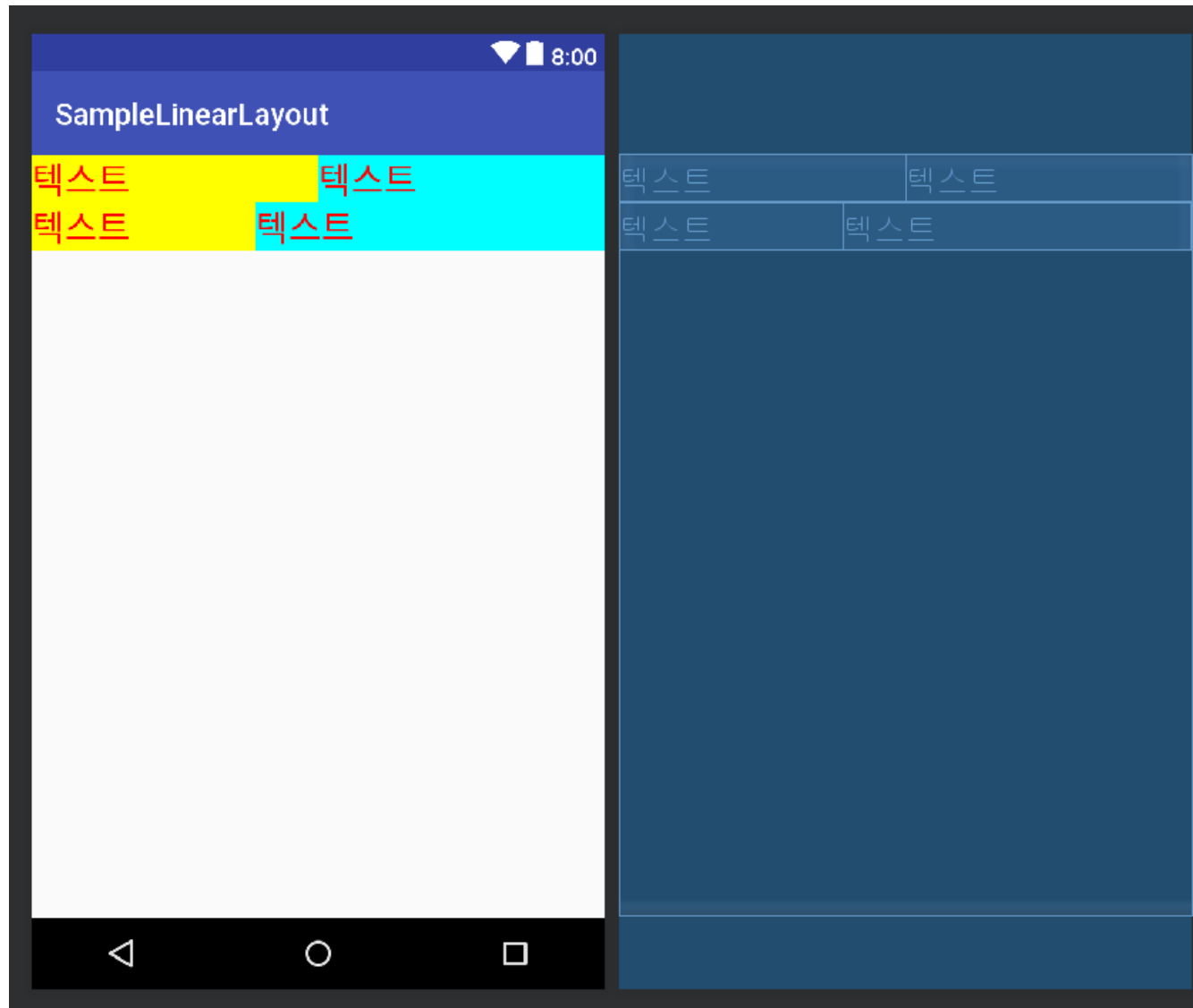
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    >

    <!-- layout_weight를 1로 설정 -->
    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#ffffff00"
        android:text="텍스트"
        android:textColor="#ffff0000"
        android:textSize="24dp"
        android:layout_weight="1"/>

    <!-- layout_weight를 2로 설정 -->
    <TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#ff00ffff"
        android:text="텍스트"
        android:textColor="#ffff0000"
        android:textSize="24dp"
        android:layout_weight="2"/>

</LinearLayout>

```



```
<LinearLayout
```

```
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
>
```

```
<!-- layout_width를 0dp로 한 후 layout_weight를 1로
```

```
<TextView
```

```
    android:id="@+id/textView5"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:background="#ffffff00"
    android:text="텍스트"
    android:textColor="#ffff0000"
    android:textSize="24dp"
    android:layout_weight="1"/>
```

```
<!-- layout_width를 0dp로 한 후 layout_weight를 2로
```

```
<TextView
```

```
    android:id="@+id/textView6"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:background="#ff00ffff"
    android:text="텍스트"
    android:textColor="#ffff0000"
    android:textSize="24dp"
    android:layout_weight="2"/>
```

```
</LinearLayout>
```

## SampleLinearLayout

텍스트 텍스트  
텍스트 텍스트  
텍스트 텍스트

텍스트 텍스트  
텍스트 텍스트  
텍스트 텍스트

## gravity 속성으로 지정할 수 있는 대표적인 값

정렬 속성 값	설명
top	대상 객체를 위쪽 끝에 배치하기
bottom	대상 객체를 아래쪽 끝에 배치하기
left	대상 객체를 왼쪽 끝에 배치하기
right	대상 객체를 오른쪽 끝에 배치하기
center_vertical	대상 객체를 수직 방향의 중앙에 배치하기
center_horizontal	대상 객체를 수평 방향의 중앙에 배치하기
fill_vertical	대상 객체를 수직 방향으로 여유 공간만큼 확대하여 채우기
fill_horizaontal	대상 객체를 수평 방향으로 여유 공간만큼 확대하여 채우기
center	대상 객체를 수직 방향과 수평 방향의 중앙에 배치하기
fill	대상 객체를 수직 방향과 수평 방향으로 여유 공간만큼 확대하여 채우기
clip_vertical	대상 객체의 상하 길이가 여유 공간보다 클 경우에 남는 부분을 잘라내기 top clip_vertical로 설정한 경우 아래쪽에 남는 부분 잘라내기 bottom clip_vertical로 설정한 경우 위쪽에 남는 부분 잘라내기 center_vertical clip_vertical로 설정한 경우 위쪽과 아래쪽에 남는 부분 잘라내기
clip_horizontal	대상 객체의 좌우 길이가 여유 공간보다 클 경우에 남는 부분을 잘라내기 right clip_horizontal로 설정한 경우 왼쪽에 남는 부분 잘라내기 left clip_horizaontal로 설정한 경우 오른쪽에 남는 부분 잘라내기 center_horizontal clip_horizontal로 설정한 경우 왼쪽과 오른쪽에 남는 부분 잘라내기



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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=".MainActivity">

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_above="@+id/button2"
        android:text="Button"
        android:background="#ff0088ff"/>

    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:text="Button" />

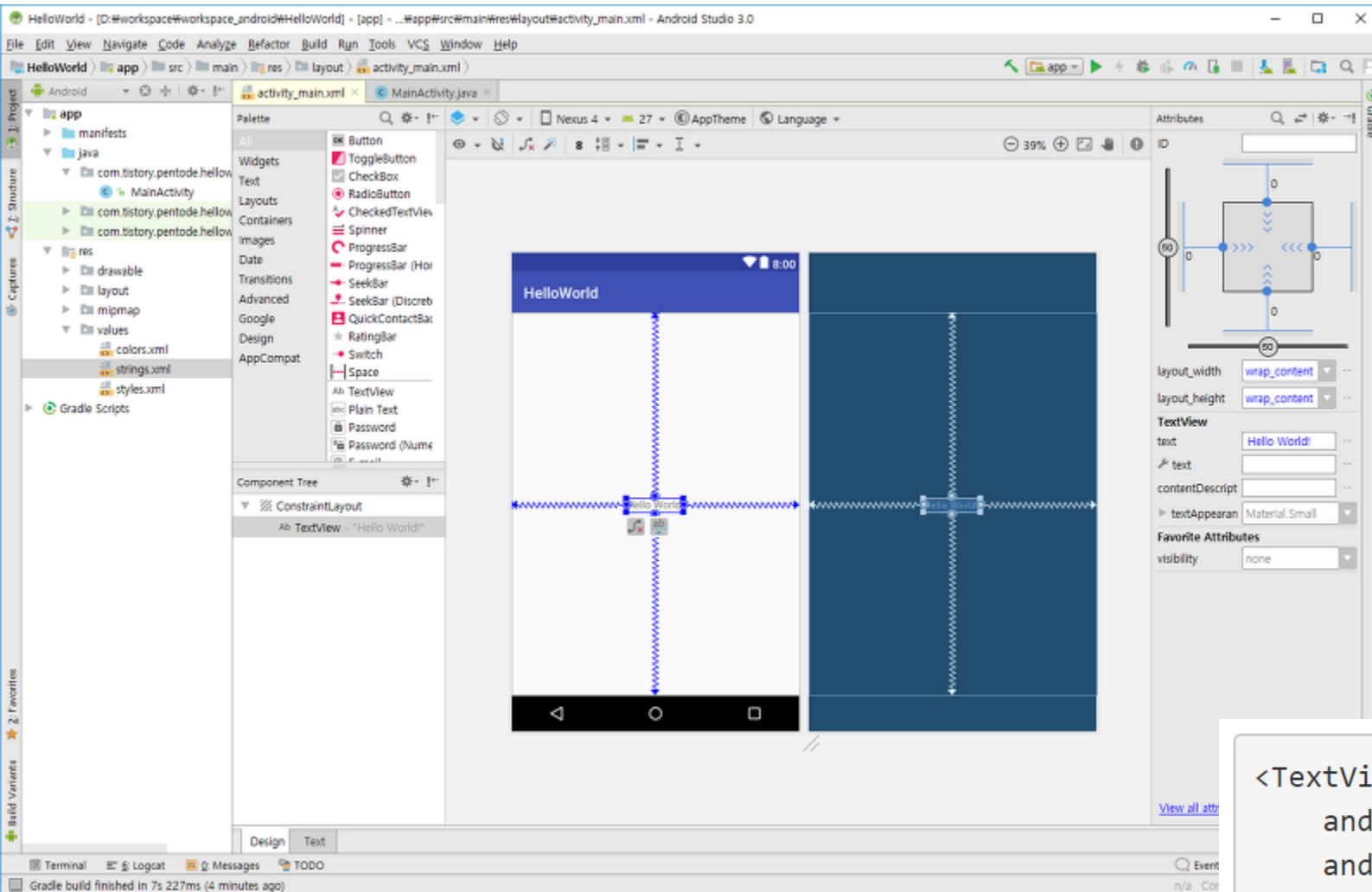
</RelativeLayout>
```

## 상대 레이아웃에서 부모 컨테이너와의 상대 위치를 이용해 뷰를 배치할 수 있는 속성

속성	설명
layout_alignParentTop	부모 컨테이너의 위쪽과 뷰의 위쪽을 맞춤
layout_alignParentBottom	부모 컨테이너의 아래쪽과 뷰의 아래쪽을 맞춤
layout_alignParentLeft	부모 컨테이너의 왼쪽 끝과 뷰의 왼쪽 끝을 맞춤
layout_alignParentRight	부모 컨테이너의 오른쪽 끝과 뷰의 오른쪽 끝을 맞춤
layout_centerHorizontal	부모 컨테이너의 수평 방향 중앙에 배치함
layout_centerVertical	부모 컨테이너의 수직 방향 중앙에 배치함
layout_centerInParent	부모 컨테이너의 수직 방향 중앙에 배치함

## 부모 컨테이너가 아닌 다른 뷰와의 상대적 위치를 이용해 뷰를 배치하는 속성

속성	설명
layout_above	지정한 뷰의 위쪽에 배치함
layout_below	지정한 뷰의 아래쪽에 배치함
layout_toLeftOf	지정한 뷰의 왼쪽에 배치함
layout_toRightOf	지정한 뷰의 오른쪽에 배치함
layout_alignTop	지정한 뷰의 위쪽과 맞춤
layout_alignBottom	지정한 뷰의 아래쪽과 맞춤
layout_alignLeft	지정한 뷰의 왼쪽과 맞춤
layout_alignRight	지정한 뷰의 오른쪽과 맞춤
layout_alignBaseline	지정한 뷰와 내용물의 아래쪽 기준선(baseline)을 맞춤



### <TextView

```

android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Hello World!"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />

```

- strings.xml 파일을 더블 클릭해서 연 다음 다음을 추가합니다.

```
<resources>
    <string name="app_name">HelloWorld</string>
    <string name="greeting">Hello World!</string>
</resources>
```

- activity\_main.xml 파일을 Text 탭에서 다음을 수정합니다.

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/greeting"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-
  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=".MainActivity">

  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="onButton1Clicked"
    android:text="버튼 클릭"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

  <Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="onButton2Clicked"
    android:text="NAVER"
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="0dp" />

  <Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="onButton3Clicked"
    android:text="메뉴화면"
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="48dp" />

</android.support.constraint.ConstraintLayout>

```

activity\_main.xml

## MainActivity.java

```

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

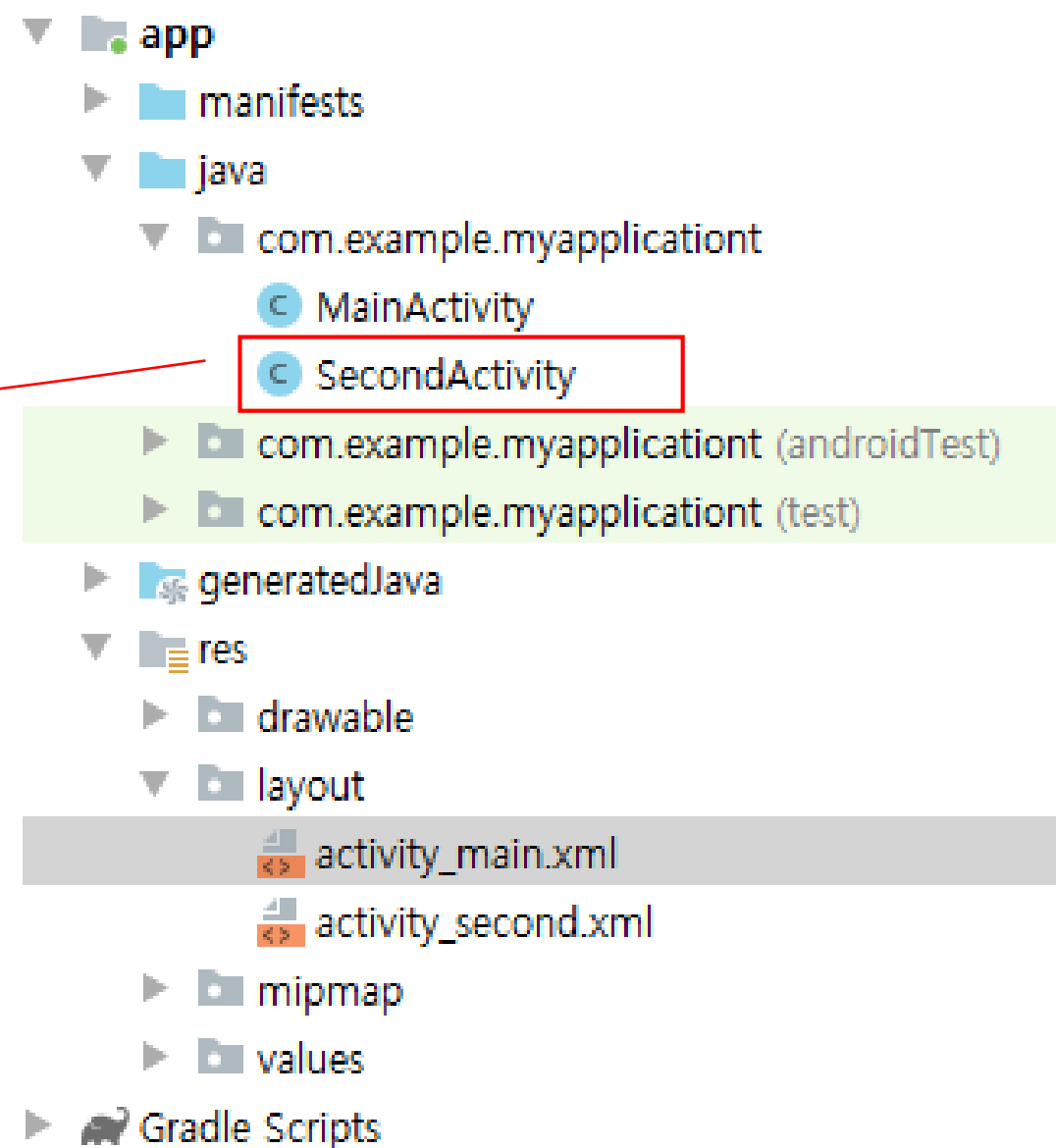
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

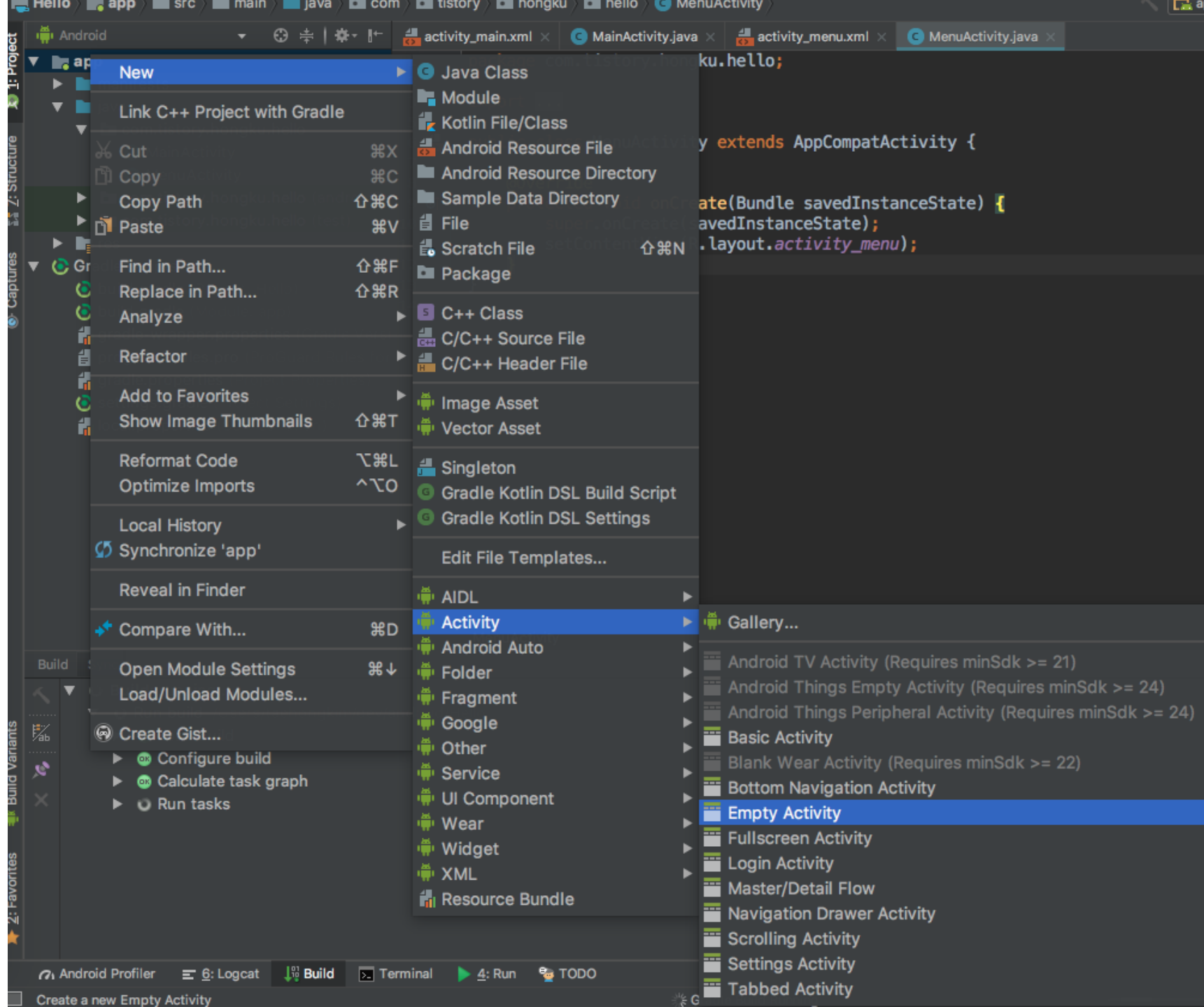
    // 메인 화면 띄우기
    public void onButton3Clicked(View view){
        Intent intent = new Intent(this, MenuActivity.class);
        startActivity(intent);
    }
}

```

```
import android.content.Intent;
```

```
public void onClicked(View view) {  
    Intent intent = new Intent(this, SecondActivity.class);  
    startActivity(intent);  
}
```





New

Link C++ Project with Gradle



Cut



Copy



Copy Path



Paste

Find in Path...

Replace in Path...

Analyze

Refactor

Add to Favorites



Show Image Thumbnails

Reformat Code

Optimize Imports

Local History



Synchronize 'app'

Reveal in Finder



Compare With...

Open Module Settings

Load/Unload Modules...



Create Gist...

Configure build

Calculate task graph

Run tasks

Java Class

Module

Kotlin File/Class

Android Resource File

Android Resource Directory

Sample Data Directory

File

Scratch File

Package

C++ Class

C/C++ Source File

C/C++ Header File

Image Asset

Vector Asset

Singleton

Gradle Kotlin DSL Build Script

Gradle Kotlin DSL Settings

Edit File Templates...

AIDL

Activity

Android Auto

Folder

Fragment

Google

Other

Service

UI Component

Wear

Widget

XML

Resource Bundle

ku.hello;

extends AppCompatActivity {

onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_menu);

Gallery...

Android TV Activity (Requires minSdk >= 21)

Android Things Empty Activity (Requires minSdk >= 24)

Android Things Peripheral Activity (Requires minSdk >= 24)

Basic Activity

Blank Wear Activity (Requires minSdk >= 22)

Bottom Navigation Activity

Empty Activity

Fullscreen Activity

Login Activity

Master/Detail Flow

Navigation Drawer Activity


Scrolling Activity

Settings Activity

Tabbed Activity




New Android Activity




# Configure Activity

Android Studio



Creates a new empty activity



Activity Name:

MenuActivity

☒ Generate Layout File

Layout Name:

activity\_menu

☐ Launcher Activity

☒ Backwards Compatibility (AppCompat)

Package name:

com.tistory.hongku.hello

▼

Source Language:

Java

▼

Target Source Set:

main

▼

The name of the activity class to create

Cancel

Previous

Next

Finish

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    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=".MenuActivity">

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onBackPressed"
        android:text="이전으로"
        tools:layout_editor_absoluteX="148dp"
        tools:layout_editor_absoluteY="231dp" />

</android.support.constraint.ConstraintLayout>

```

activity\_menu.xml

```

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;

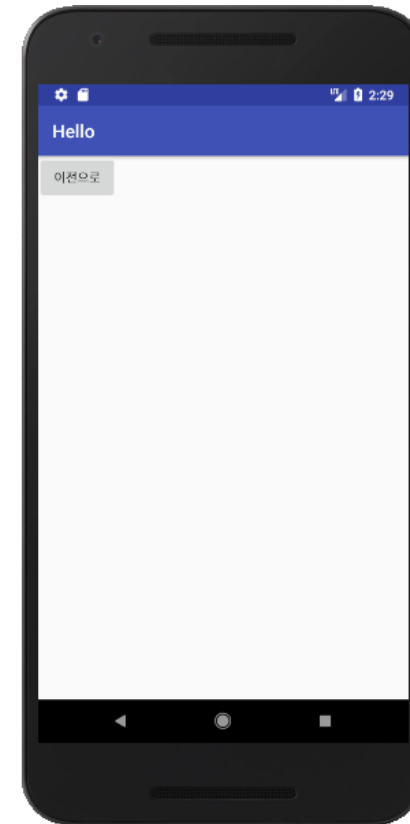
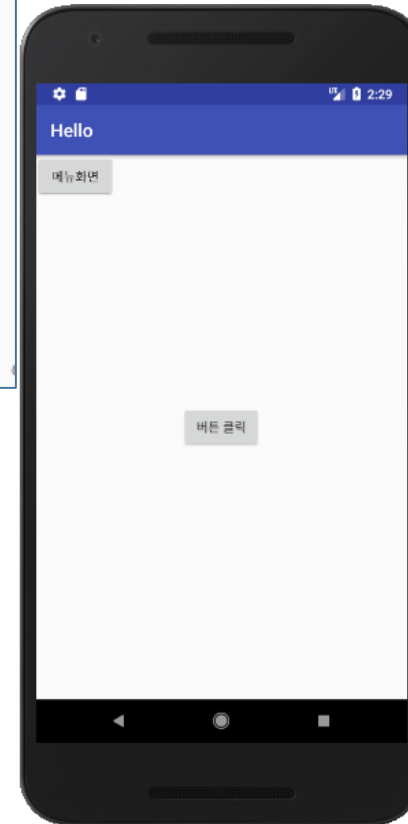
public class MenuActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_menu);
    }

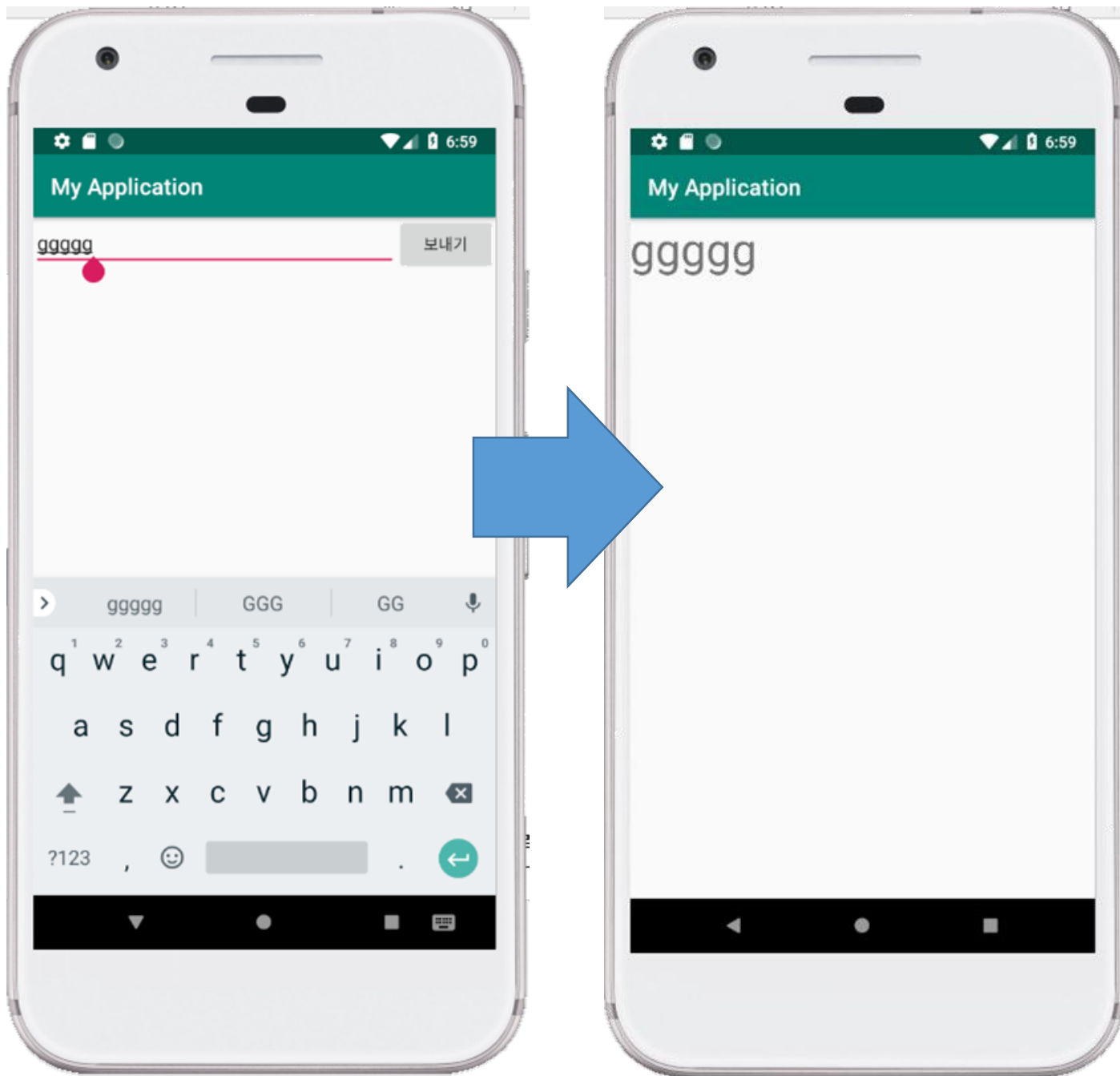
    // 이전으로 돌아가기
    public void onBackPressed(View view){
        finish(); // 현재 이 화면을 없애줌
    }
}

```

MenuActivity.java



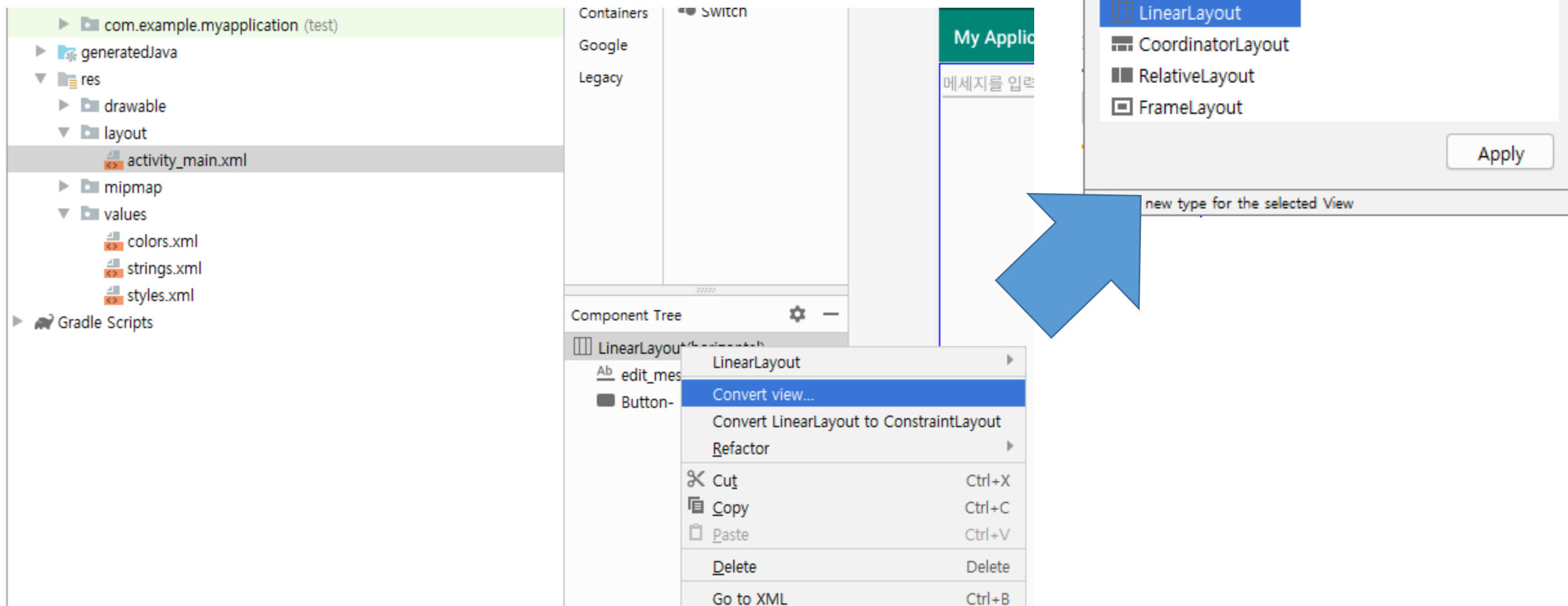
# 앱 만들기 - 1



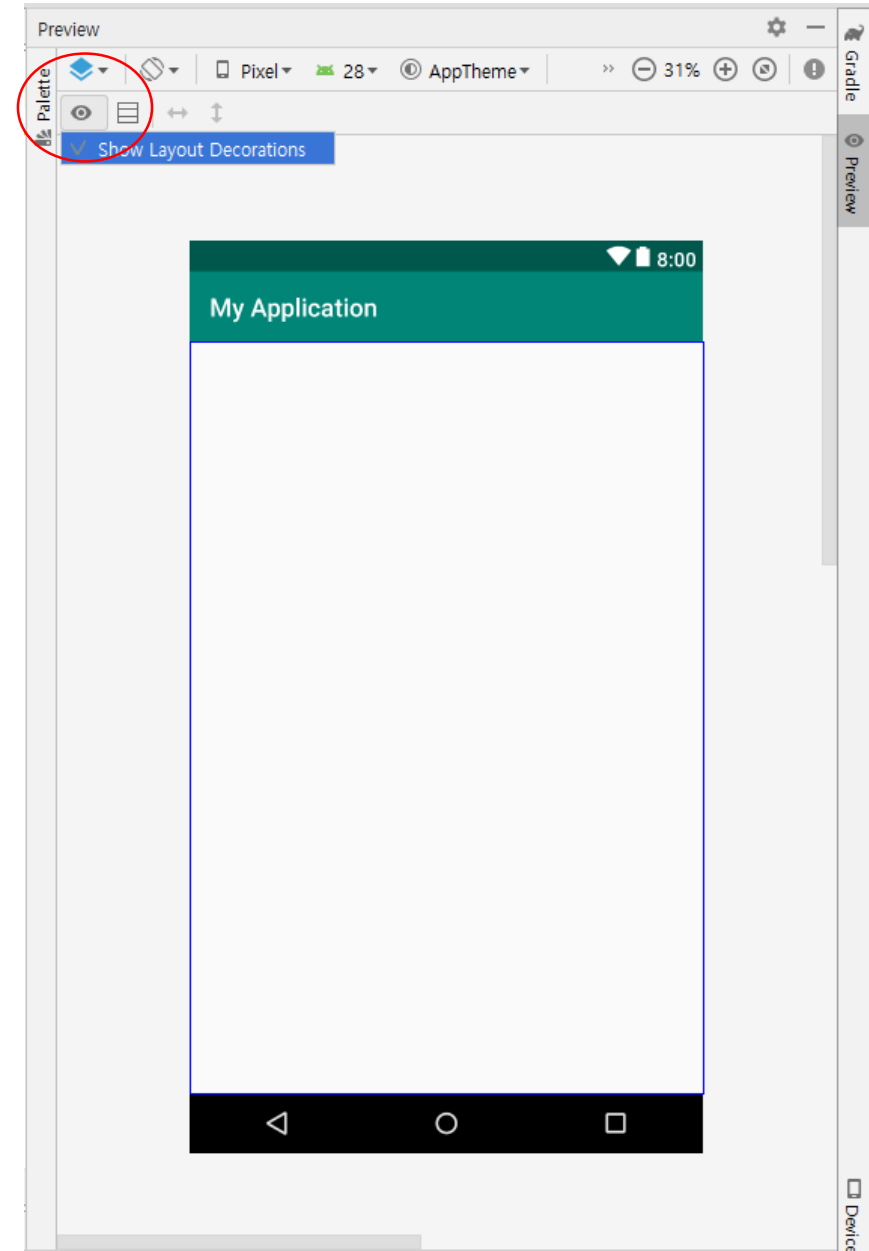
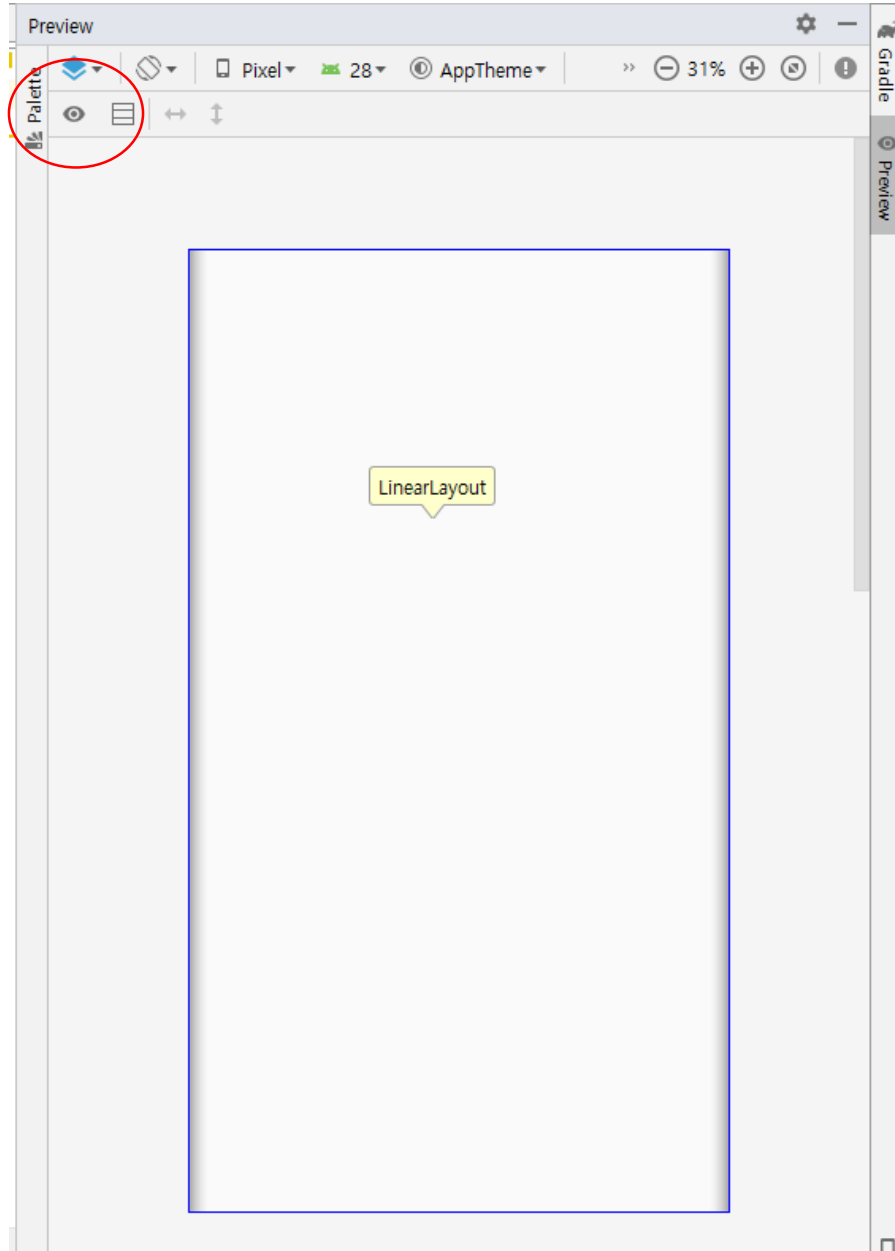
보내기 버튼을 누르면 입력한 Text가 새 화면에 나타난다.

# UI 생성 - Convert view

constraint Layout → Linear Layout



# activity\_main.xml



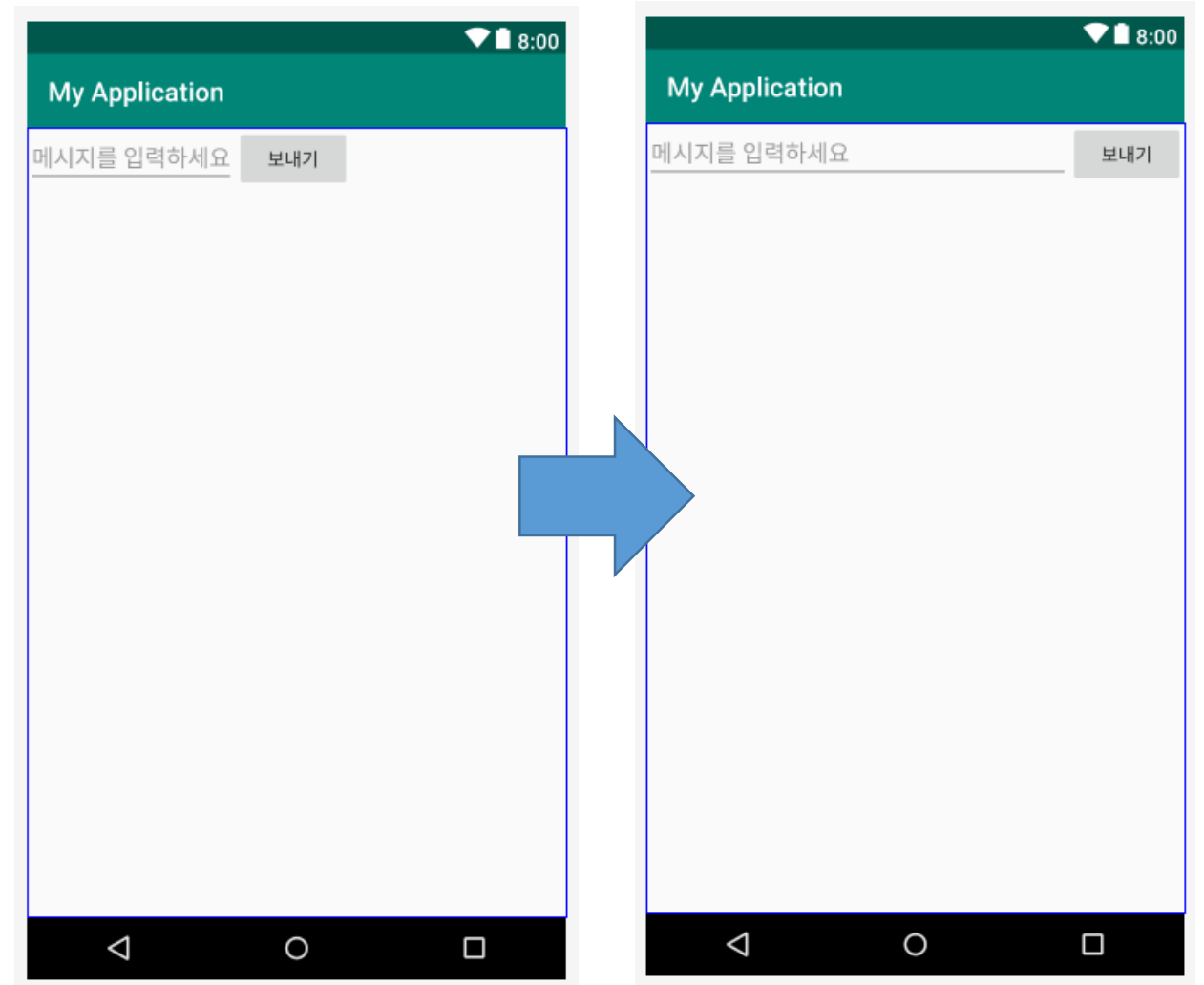
# activity\_main.xml

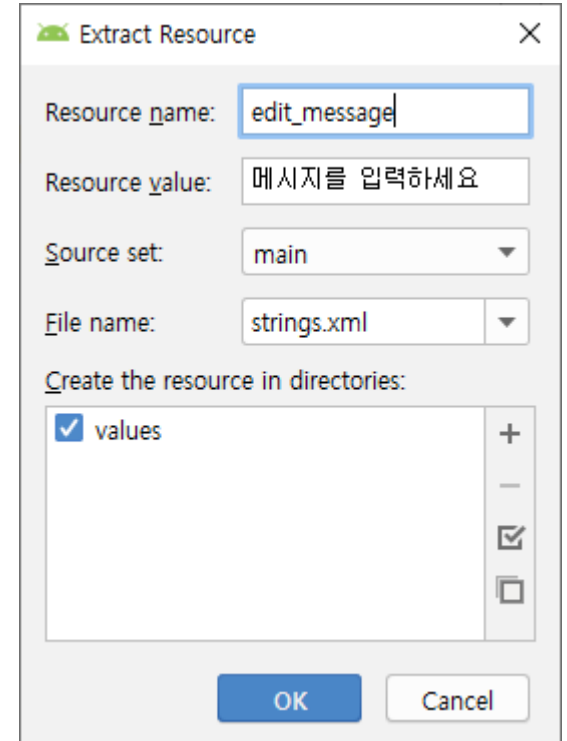
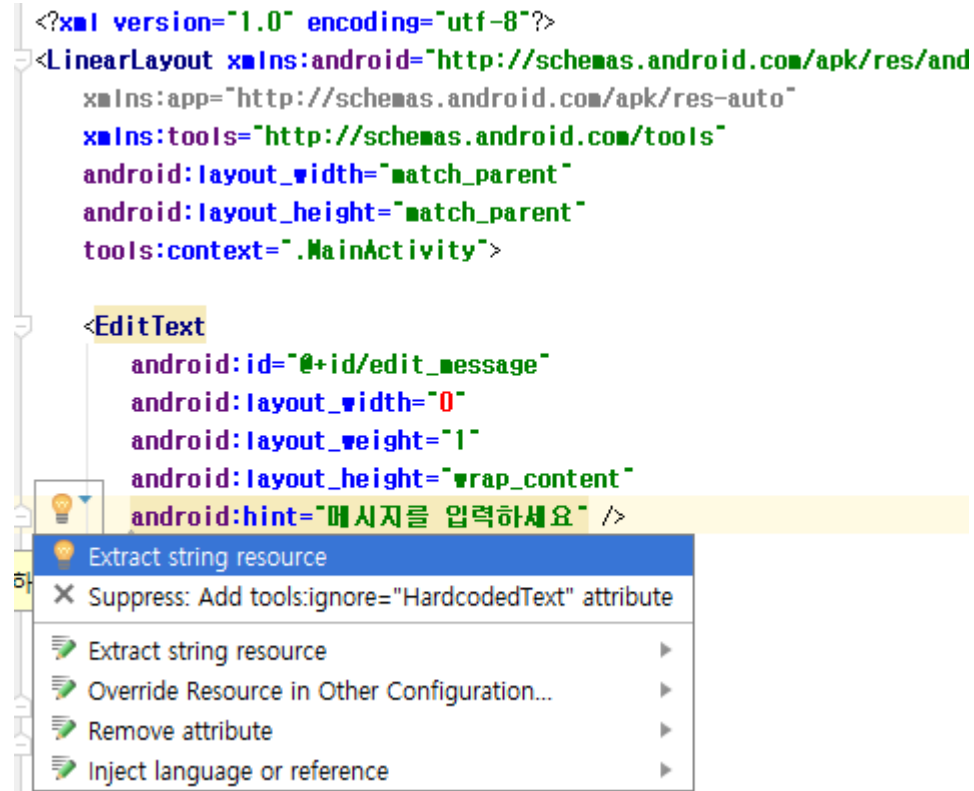
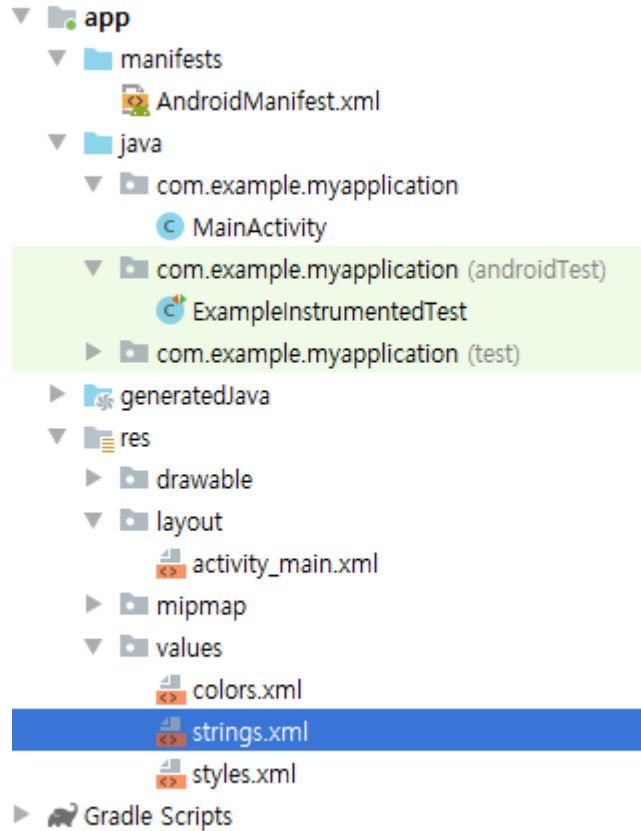
```
<EditText  
    android:id="@+id/edit_message"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:hint="메시지를 입력하세요" />
```

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="보내기" />
```

```
<EditText  
    android:id="@+id/edit_message"  
    android:layout_width="0dp"  
    android:layout_weight="1"  
    android:layout_height="wrap_content"  
    android:hint="메시지를 입력하세요" />
```

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="보내기" />
```





**Extract string resource**



```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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=".MainActivity">

    <EditText
        android:id="@+id/edit_message"
        android:layout_width="0"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        android:hint=" 메시지를 입력하세요" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=" 보내기" />

</LinearLayout>

```

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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=".MainActivity">

    <EditText
        android:id="@+id/edit_message"
        android:layout_width="0"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        android:hint="@string/edit_message" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/send_message" />

</LinearLayout>

```

## activity\_main.xml

```

<resources>
    <string name="app_name">My Application</string>
</resources>

```

```

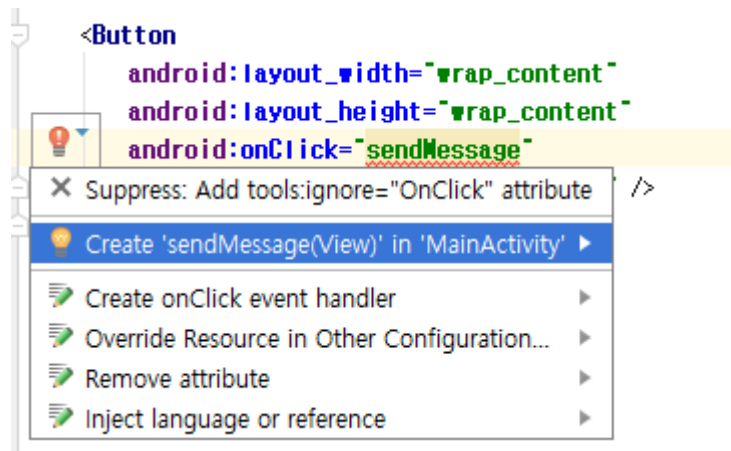
<resources>
    <string name="app_name">My Application</string>
    <string name="edit_message"> 메시지를 입력하세요</string>
    <string name="send_message"> 보내기</string>
</resources>

```

## strings.xml

## <Button

```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:onClick="sendMessage"  
android:text="@string/send_message" />
```



## 버튼을 클릭 했을 때 발생하는 이벤트

```
public class MainActivity extends AppCompatActivity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }
```

```
    public void sendMessage(View view) {  
    }
```



에디트 텍스트를 생성한다.

전체는 상대적 레이아웃

TableLayout으로 버튼들을 정렬  
한다.

버튼을 `android:layout_alignParentBottom="true"`  
로 하여 아래에 붙인다.