Week09.

터치센서

개발환경 구축 절차

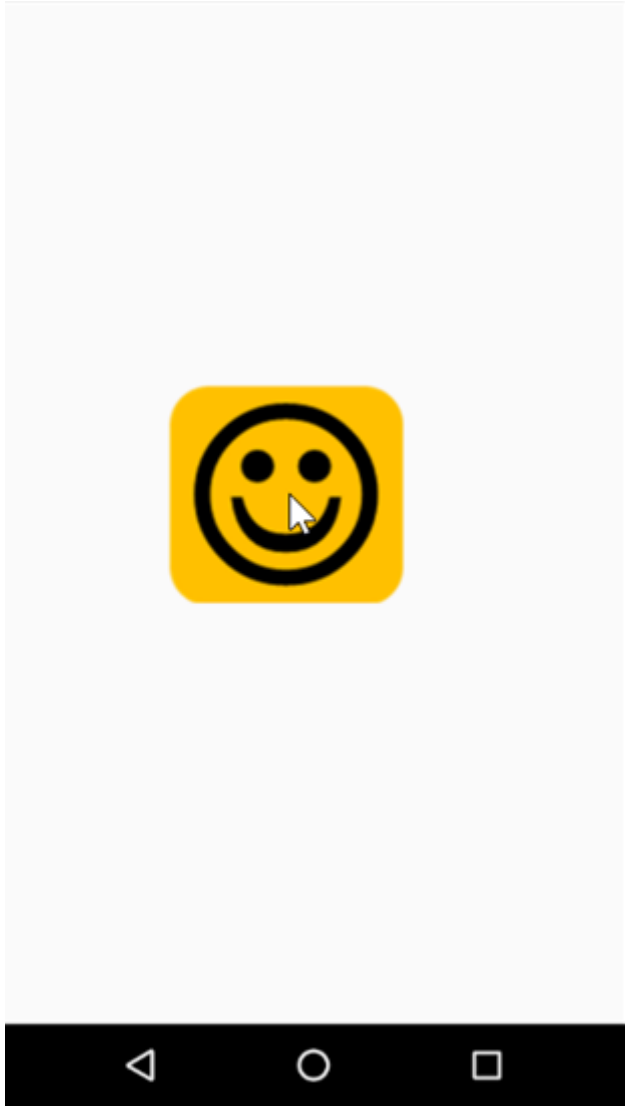
주 차	수 업 내 용
1	수업 소개
2	개발 환경 구축과 맛보기 프로젝트
3	텍스트 출력과 레이아웃
4	이미지의 출력
5	이벤트 처리와 액티비티 간 이동
6	오디오 재생
7	비디오 재생
8	**중간고사**
9	애니메이션
10	**사물인터넷과 센서 – 터치 센서,** 모션 센서
11	사물인터넷과 센서 – 위치 센서, 환경 센서
12	NFC 활용
13	공공 DB 오픈 API 활요
14	구글 맵과 위치 추적
15	**기말 고사**

https://github.com/hopypark

터치 센서를 활용 앱의 예

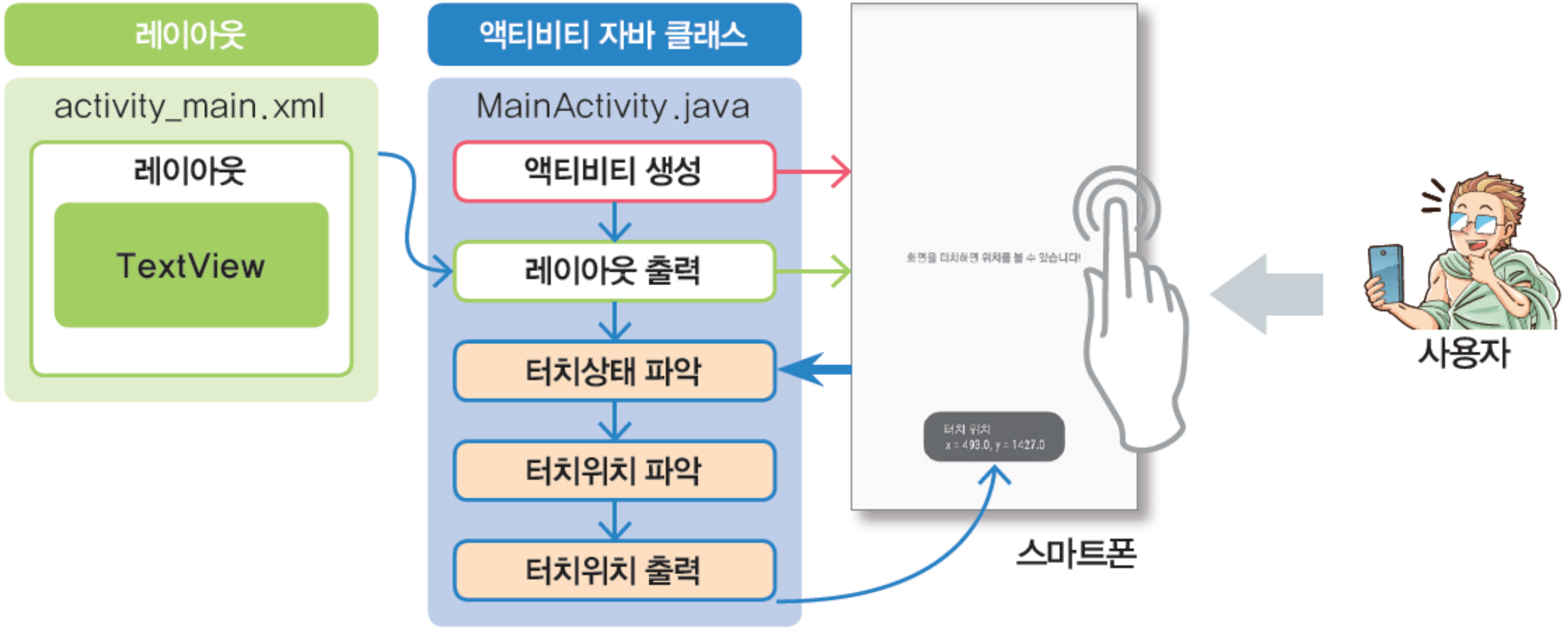
(a) 게임레벨 선택

(b) 카카오프렌즈 일렬 배치하기



Follow Me

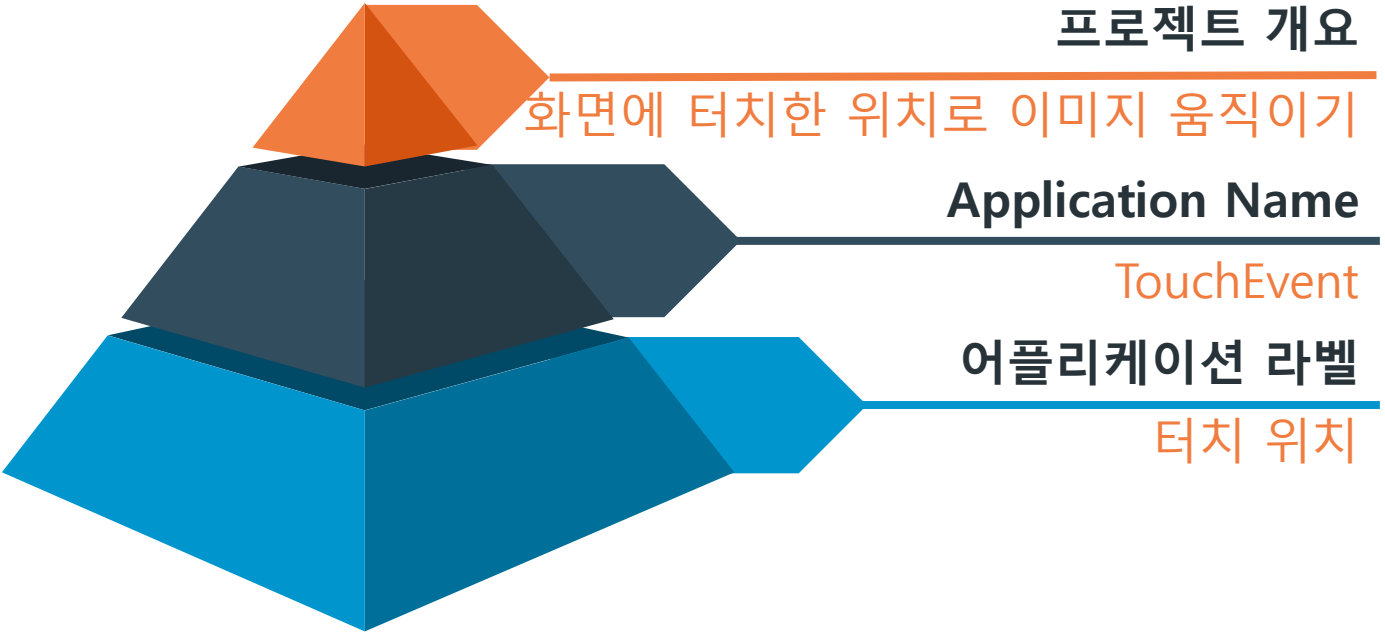
터치 센서 원리


레이아웃
activity_main.xml
레이아웃
TextView
액티비티 자바 클래스
MainActivity.java
액티비티 생성
레이아웃 출력
터치상태 파악
터치위치 파악
터치위치 출력
화면을 터치하면 위치를 볼 수 있습니다!
터치 위치
x = 493.0, y = 1427.0
스마트폰
사용자

onTouchEvent() 메소드의 매개변수

터치 이벤트	내용
ACTION_CANCEL	제스처가 중지될 때
ACTION_DOWN	누르는 제스처가 시작될 때
ACTION_MOVE	누르는 동안 이동 등의 변화가 일어남(ACTION_DOWN과 ACTION_UP 사이)
ACTION_UP	누르는 제스처가 끝남
ACTION_OUTSIDE	제스처가 화면을 벗어남

Step 0.프로젝트 개요



프로젝트 개요

화면에 터치한 위치로 이미지 움직이기

Application Name

TouchEvent

어플리케이션 라벨

터치 위치

Step 1. 프로젝트 생성

절차	내 용
①프로젝트 시작	메뉴에서 'File → New Project' 클릭
②프로젝트 구성	Application Name: **TouchEvent**
	Company Domain: kyungtae.example.com(디폴트 사용)
	Project Location: ~/AndroidStudioProject/ktpark/TouchEvent
③제품형태	Phone and Tablet(사용할 안드로이드 버전 지정: 8.1 Oreo)
④액티비티 유형	Empty Activity
⑤파일 옵션	Activity Name: MainActivity(디폴트 사용)
	Layout Name: activity_main(디폴트 사용)

Step 2. 파일 편집

모듈	폴더	소스 파일	편집 내용
manifests		AndroidManifest.xml	
java	com.example.kyungtae.video1	MainActivity.java	• 터치 위치 출력
res	drawable	smile.png	• 스마일 이미지
	layout	activity_main.xml	• 이미지 리소스 출력
	mipmap	ic_launcher.png	
	values	colors.xml	
		dimens.xml	
		styles.xml	

이미지 리소스

smile.png(drawable)

화면 레이아웃

```
RelativeLayout
    ImageView
        id @+id/smile
        src @drawable/smile
```

activity_main.xml (layout)

앱 라벨

텍스트 자원

```
string
    app_name 나를 따라와~~
```

strings.xml (values)

액티비티 제어

```
onCreate()
    super onCreate()
    setContentView(R.layout.activity_main)
    ImageView iv =
        (ImageView)findViewById(R.id.smile);

onTouchEvent()
    ObjectAnimator anim =
        bjectAnimator.ofFloat
            (iv, "translationX",previous,
x);

            ...
```

MainActivity.java (layout)

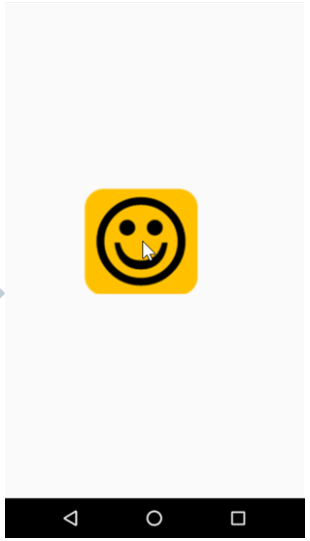
어플리케이션 구성
액티비티의 자바 클래스

어플리케이션 기본 정보

```
application
    icon  @mipmap/ic_launcher
    label @string/app_name
    theme @style/AppTheme
    activity
        name MainActivity
```

AndroidManifest.xml (manifest)

컴파일/빌더



터치 x, y 위치 확인

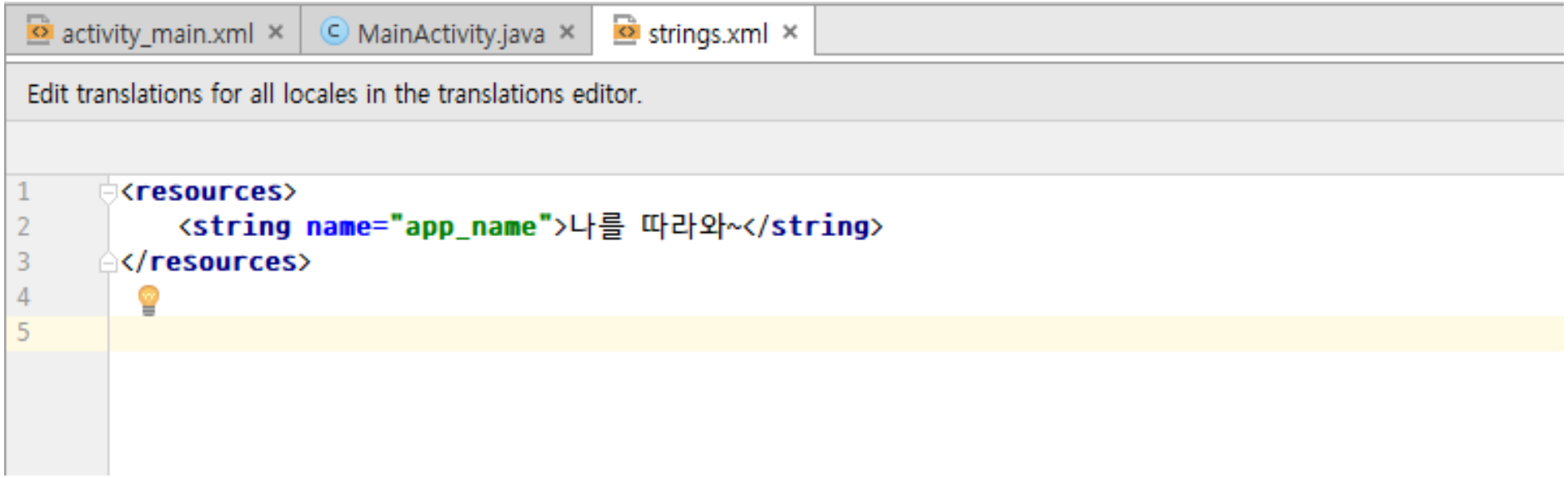
컴파일/빌더 정보

```
build gradle(Project)
build gradle(Module app)
gradle properties
settings gradle
local properties
```

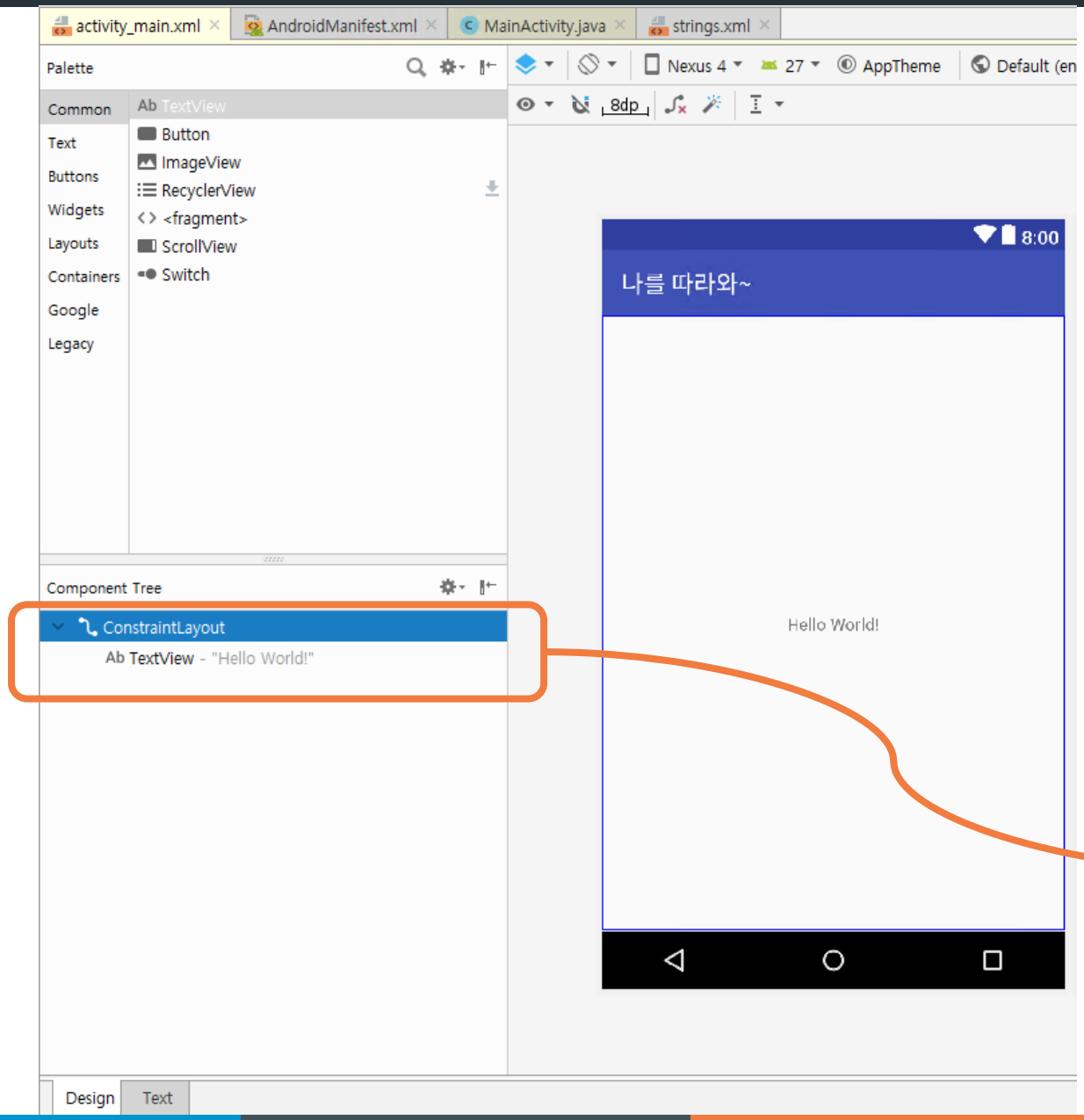
(Gradle Scripts)

Step 2.1 텍스트 자원의 편집

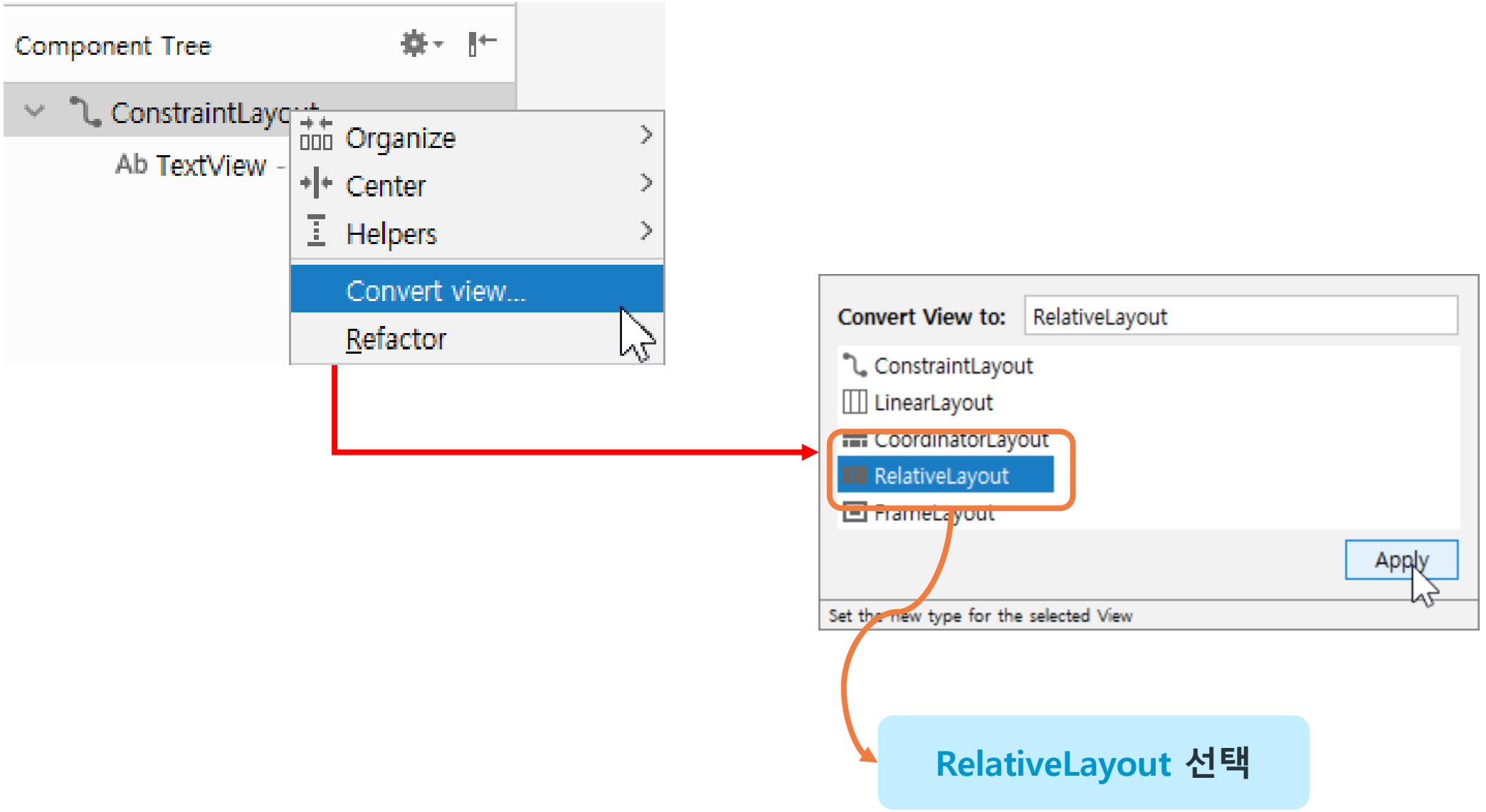
- strings.xml



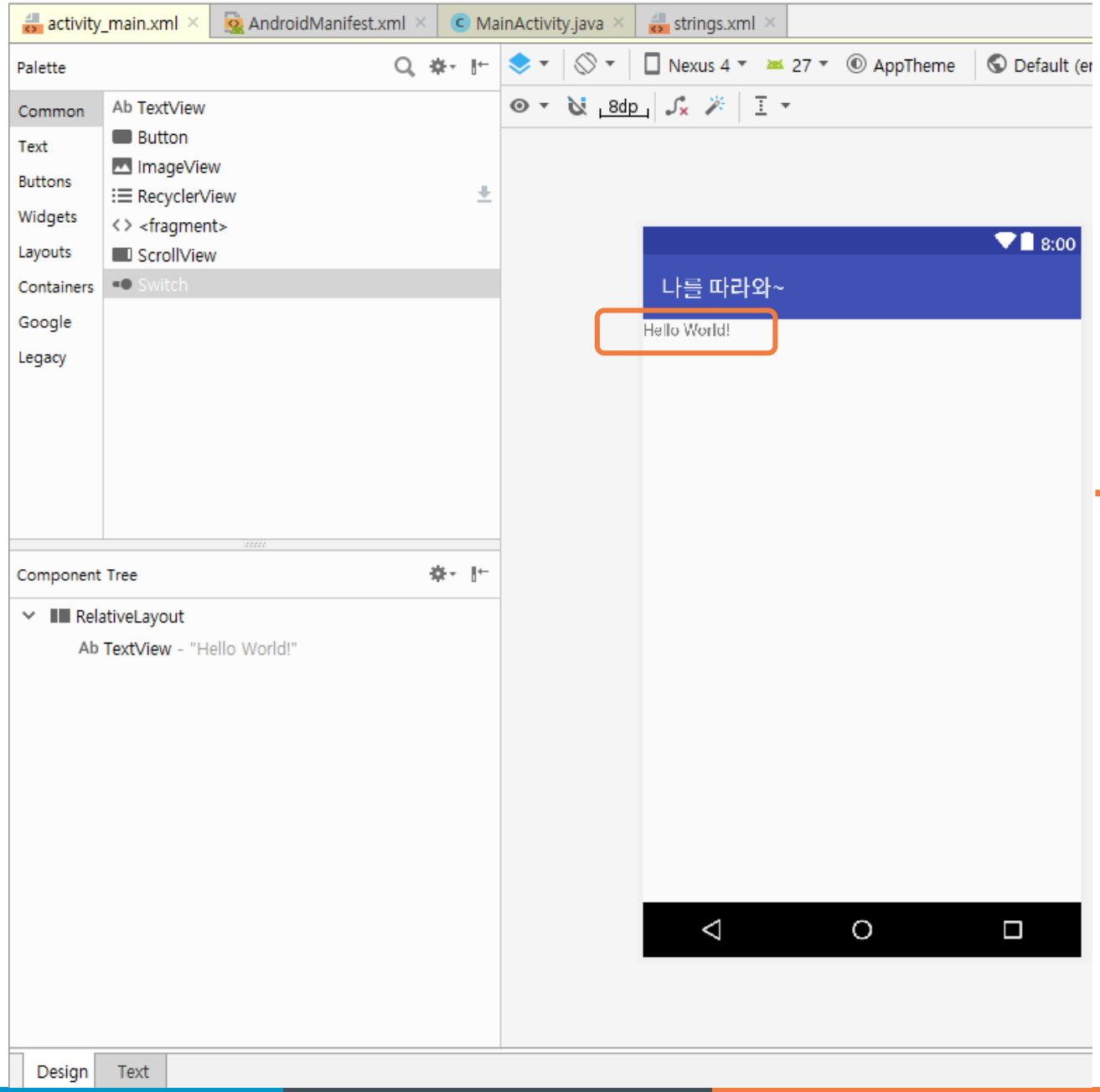
2.2 화면 설계

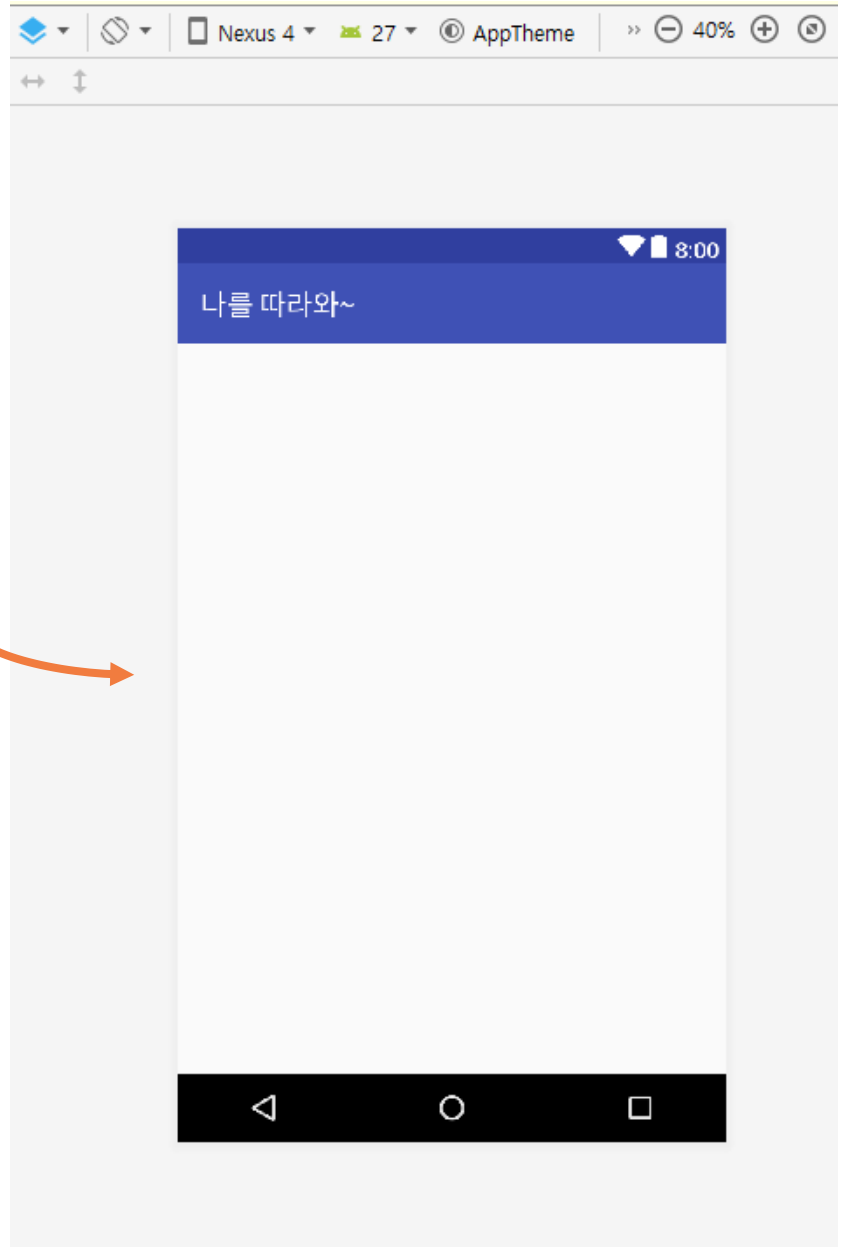


ConstraintLayout → RelativeLayout으로 변경

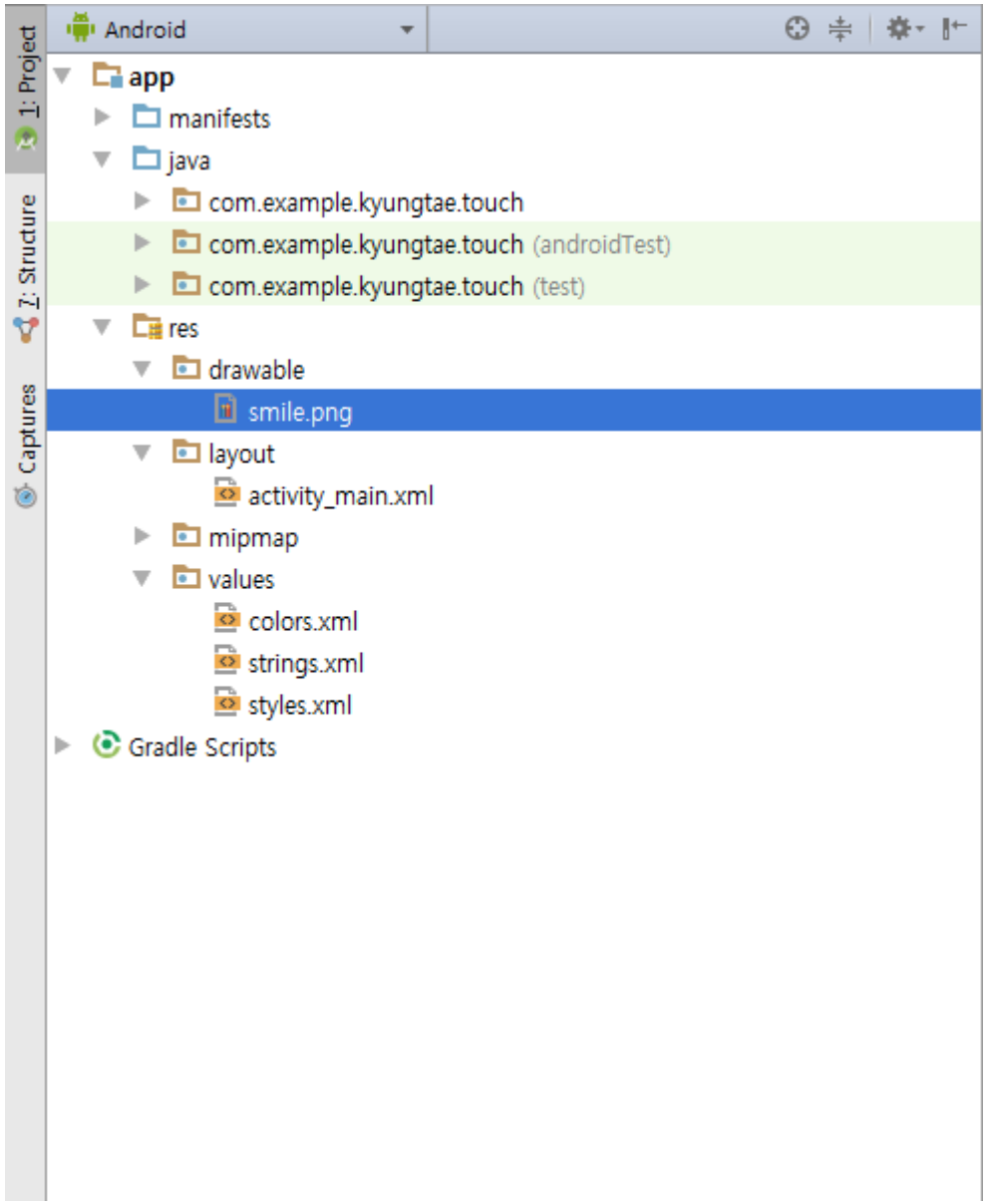

Component Tree
ConstraintLayout
Ab TextView
Organize
Center
Helpers
Convert view...
Refactor
Convert View to:
RelativeLayout
ConstraintLayout
LinearLayout
CoordinatorLayout
RelativeLayout
FrameLayout
Apply
Set the new type for the selected View
RelativeLayout 선택

- Layout 초기화 설정 - 기본 TextView 삭제



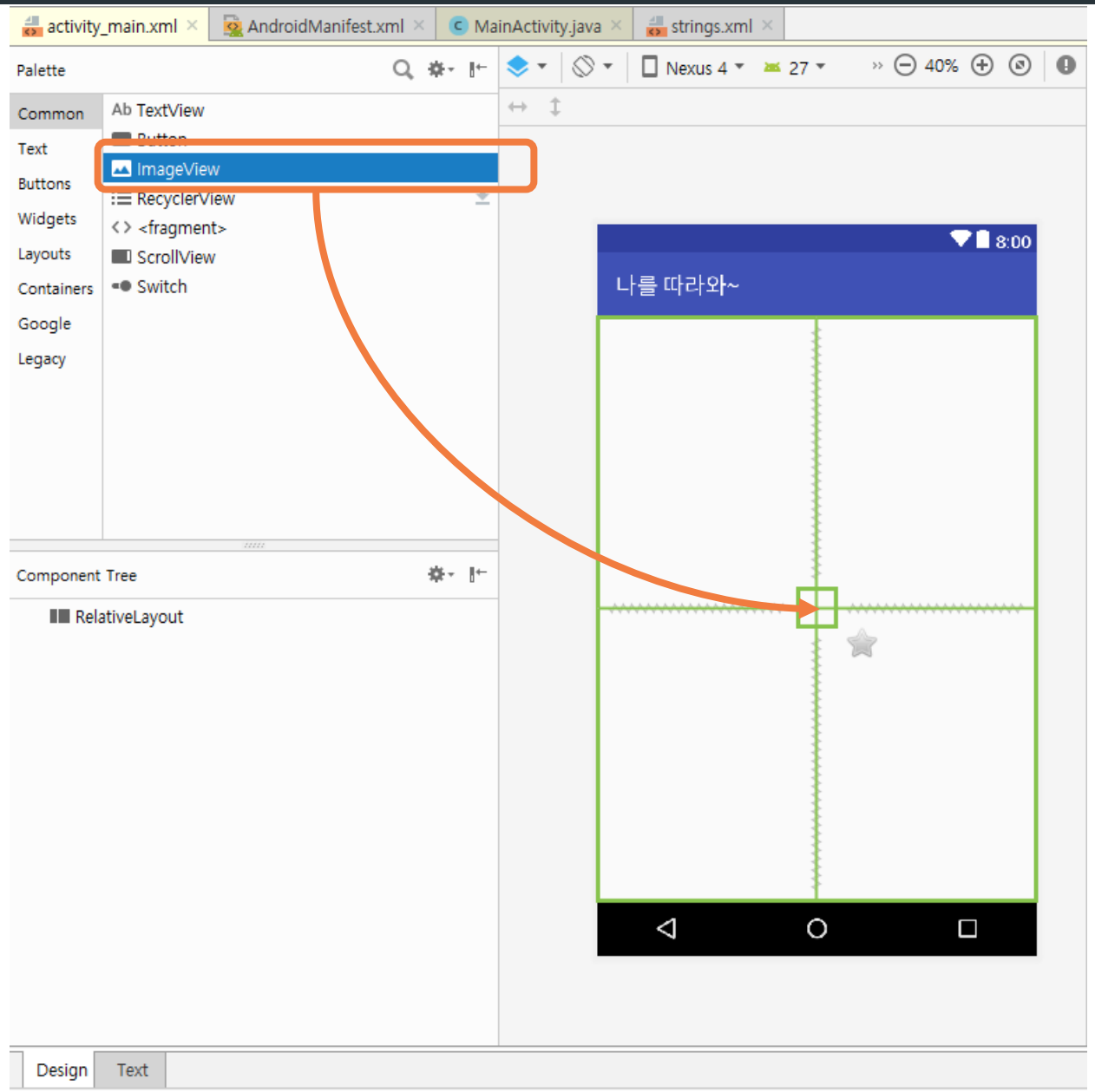


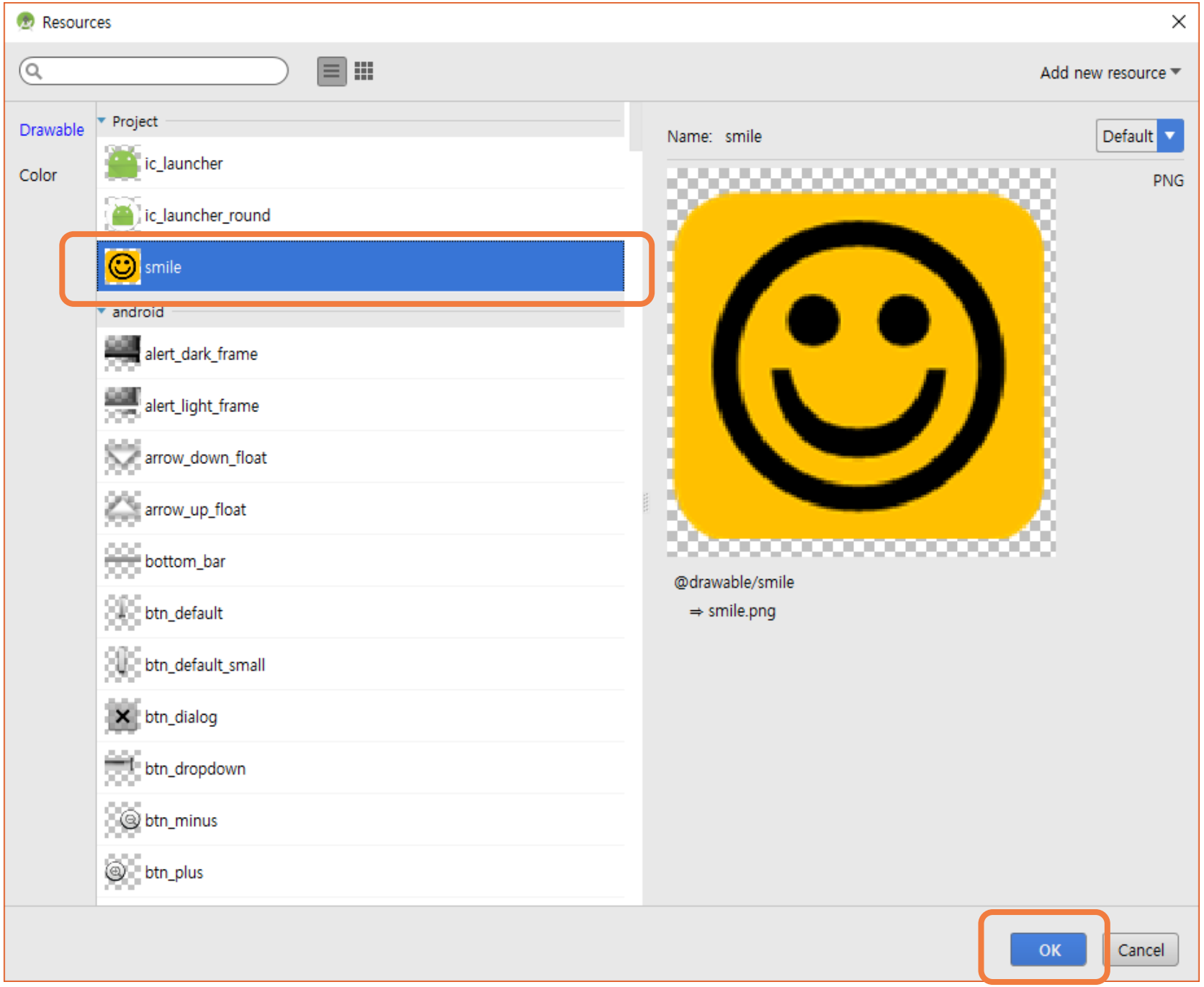
2.3 이미지 리소스 – res/drawable 에 smile.png 추가



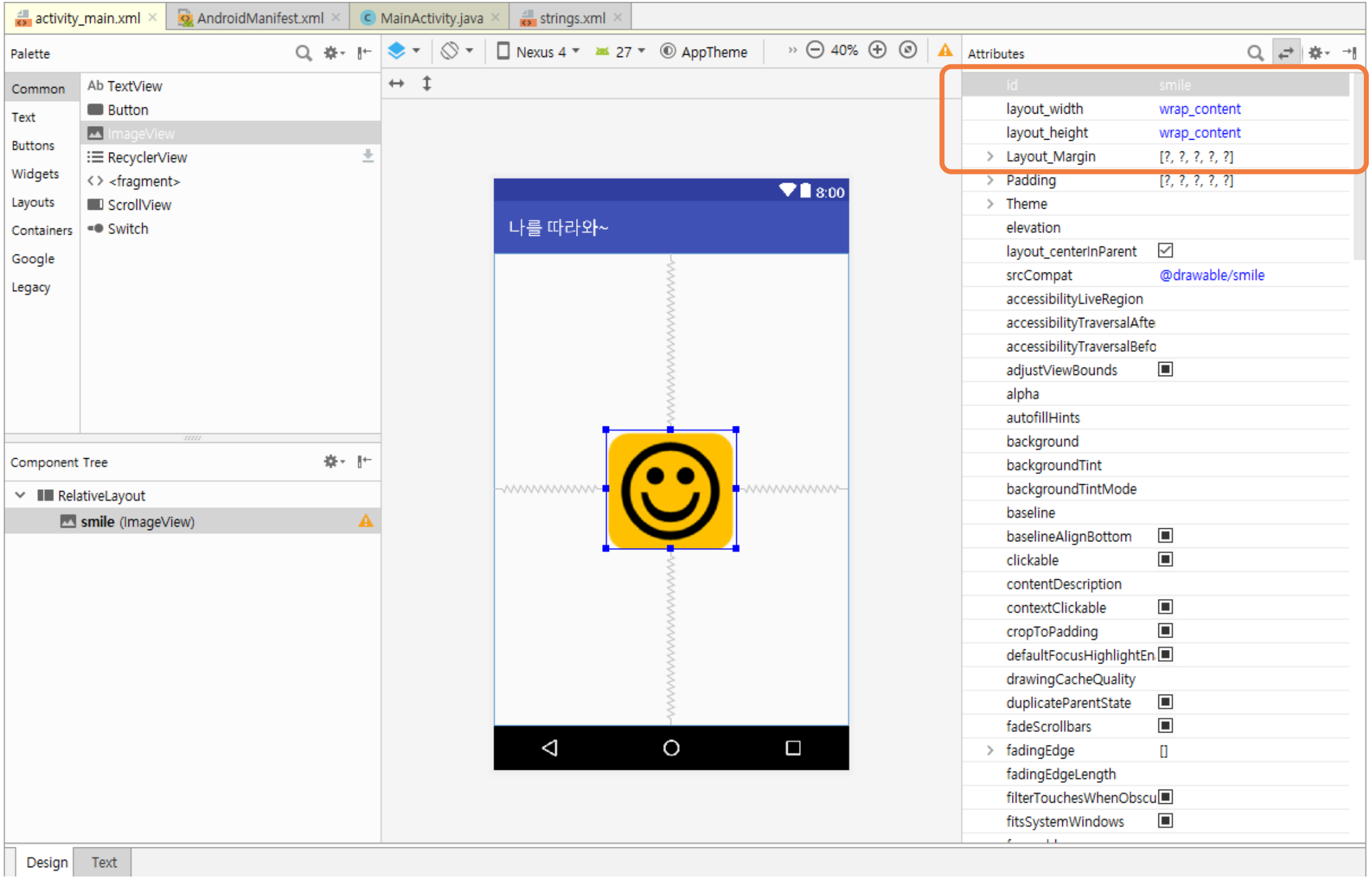
smile.png

2.4 화면 설계


activity_main.xml
AndroidManifest.xml
MainActivity.java
strings.xml
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Google
Legacy
Ab TextView
ImageView
RecyclerView
<fragment>
ScrollView
Switch
Component Tree
RelativeLayout
Nexus 4
27
40%
8:00
나를 따라와~
Design
Text



- Drawable 항목의 smile을 선택


activity_main.xml
AndroidManifest.xml
MainActivity.java
strings.xml
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Google
Legacy
Ab TextView
Button
ImageView
RecyclerView
<fragment>
ScrollView
Switch
Component Tree
RelativeLayout
smile (ImageView)
Nexus 4
27
AppTheme
40%
8:00
나를 따라와~
Attributes
id
smile
layout_width
wrap_content
layout_height
wrap_content
Layout_Margin
[?, ?, ?, ?, ?]
Padding
[?, ?, ?, ?, ?]
Theme
elevation
layout_centerInParent
srcCompat
@drawable/smile
accessibilityLiveRegion
accessibilityTraversalAfte
accessibilityTraversalBefo
adjustViewBounds
alpha
autofillHints
background
backgroundTint
backgroundTintMode
baseline
baselineAlignBottom
clickable
contentDescription
contextClickable
cropToPadding
defaultFocusHighlightEn
drawingCacheQuality
duplicateParentState
fadeScrollbars
fadingEdge
[]
fadingEdgeLength
filterTouchesWhenObscu
fitsSystemWindows
Design
Text

- ImageView의 src 속성 변경

activity_main.xml | AndroidManifest.xml | MainActivity.java | strings.xml

```
<?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">

    <ImageView
        android:id="@+id/smile"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        app:srcCompat="@drawable/smile" />
</RelativeLayout>
```

nActivity.java | strings.xml

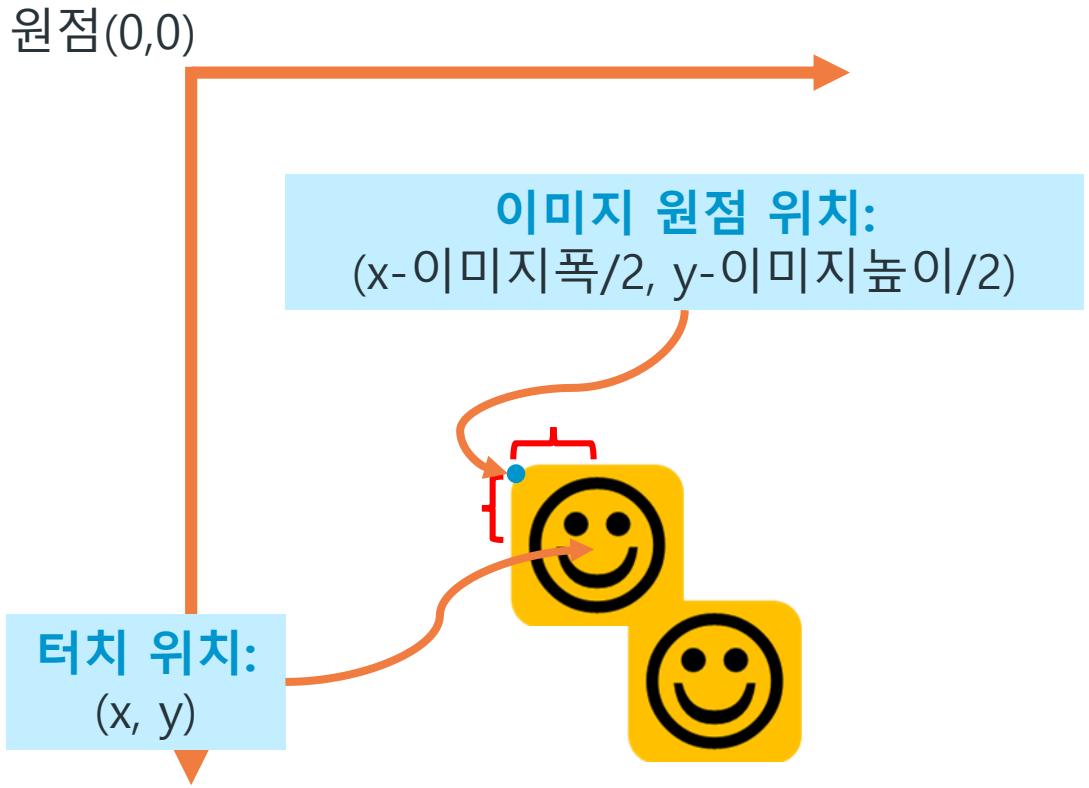
```
f-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/smile"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:src="@drawable/smile" />
</RelativeLayout>
```

터치한 위치에 아이콘 이미지 위치시키기

- 이미지의 원점과 터치 위치 사이의 오차처리

이미지 x = x-이미지 폭/2;

이미지 y = y-이미지 높이/2;



2.5 Activity 제어(MainActivity.java)

- 화면을 전체화면 크기로 만들기 위한 액티비티 상속 클래스 변경

activity_main.xml × | MainActivity.java × | strings.xml ×

```
1  package com.example.kyungtae.touch;
2
3  import android.support.v7.app.AppCompatActivity;
4  import android.os.Bundle;
5
6  public class MainActivity extends AppCompatActivity {
7
8      @Override
9      protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13 }
14
```

activity_main.xml × | MainActivity.java × | strings.xml ×

```
1  package com.example.kyungtae.touch;
2
3  import android.support.v4.app.FragmentActivity;
4  import android.support.v7.app.AppCompatActivity;
5  import android.os.Bundle;
6
7  public class MainActivity extends FragmentActivity {
8
9      @Override
10     protected void onCreate(Bundle savedInstanceState) {
11         super.onCreate(savedInstanceState);
12         setContentView(R.layout.activity_main);
13     }
14 }
15
```

화면 상단의 액션 바(제목창) 나타나지 않는 전체화면 크기로 만들기

activity_main.xml | AndroidManifest.xml | MainActivity.java | strings.xml

```java
package com.example.research.touchevent;

import android.os.Bundle;
import android.support.v4.app.FragmentActivity;
import android.view.MotionEvent;
import android.view.View;
import android.view.WindowManager;
import android.widget.ImageView;

public class MainActivity extends FragmentActivity {

    ImageView iv_Smile;

    float iv_Width = 0f;
    float iv_Height = 0f;

    float x, y;

```

타이틀 – 안테나, 배터리 정보 표시 장 없애기

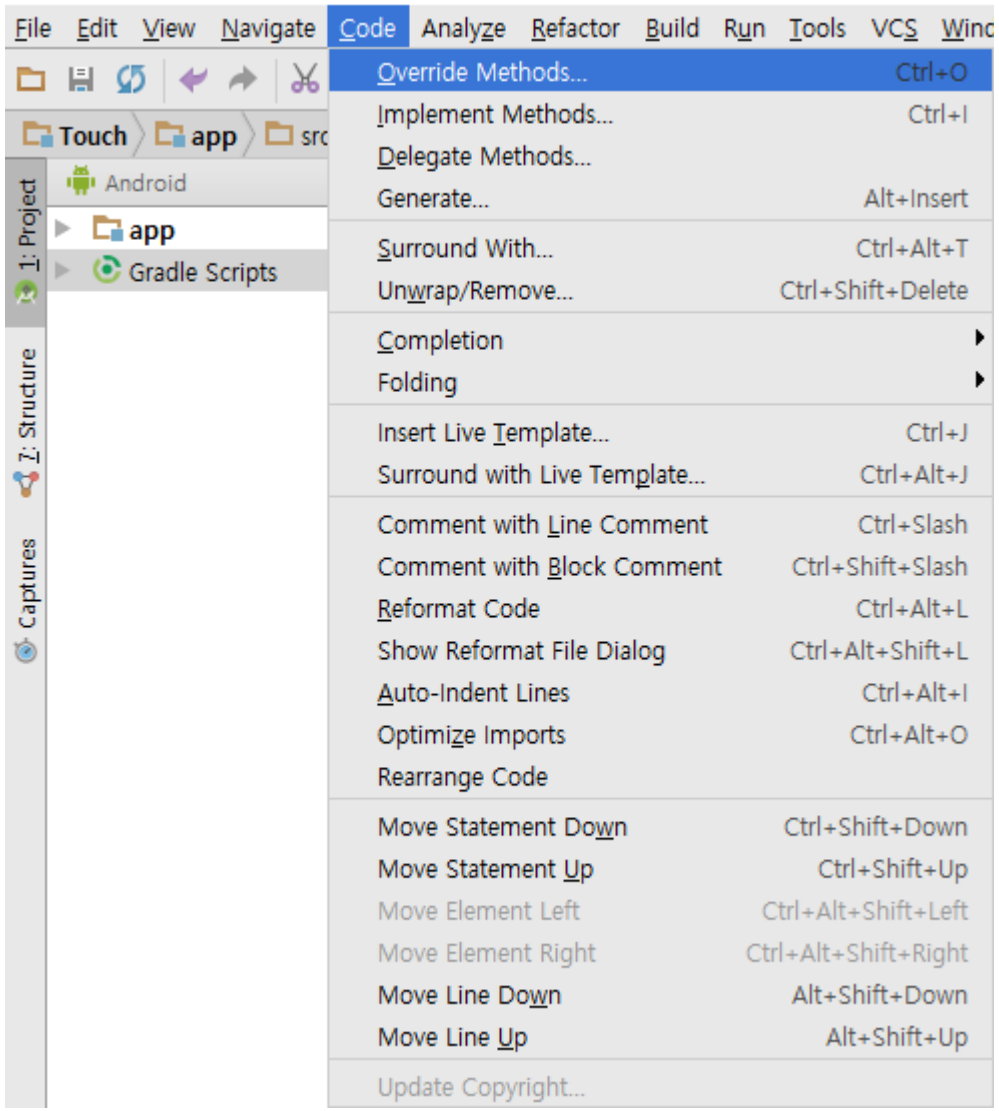
화면 중앙에 이미지가 위치

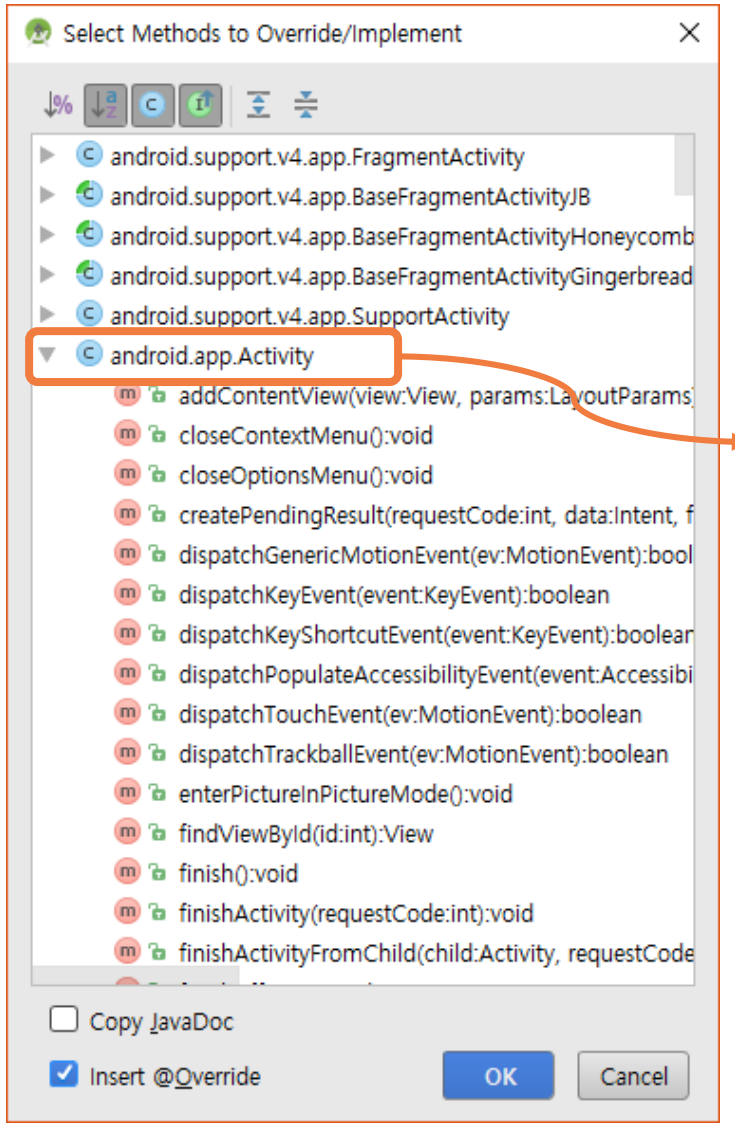
```
19      @Override
20      protected void onCreate(Bundle savedInstanceState) {
21          super.onCreate(savedInstanceState);
22          setContentView(R.layout.activity_main);
23
24          // 윈도우를 전체 크기로 설정-안테나와 배터리 정보창 제거
25          getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
26
27          iv_Smile = (ImageView) findViewById(R.id.smile);
28
29          // 이미지뷰의 크기 얻기
30          iv_Smile.measure(View.MeasureSpec.UNSPECIFIED  , View.MeasureSpec.UNSPECIFIED );
31          iv_Height = iv_Smile.getMeasuredHeight();
32          iv_Width = iv_Smile.getMeasuredWidth();
33
34      }
35
```

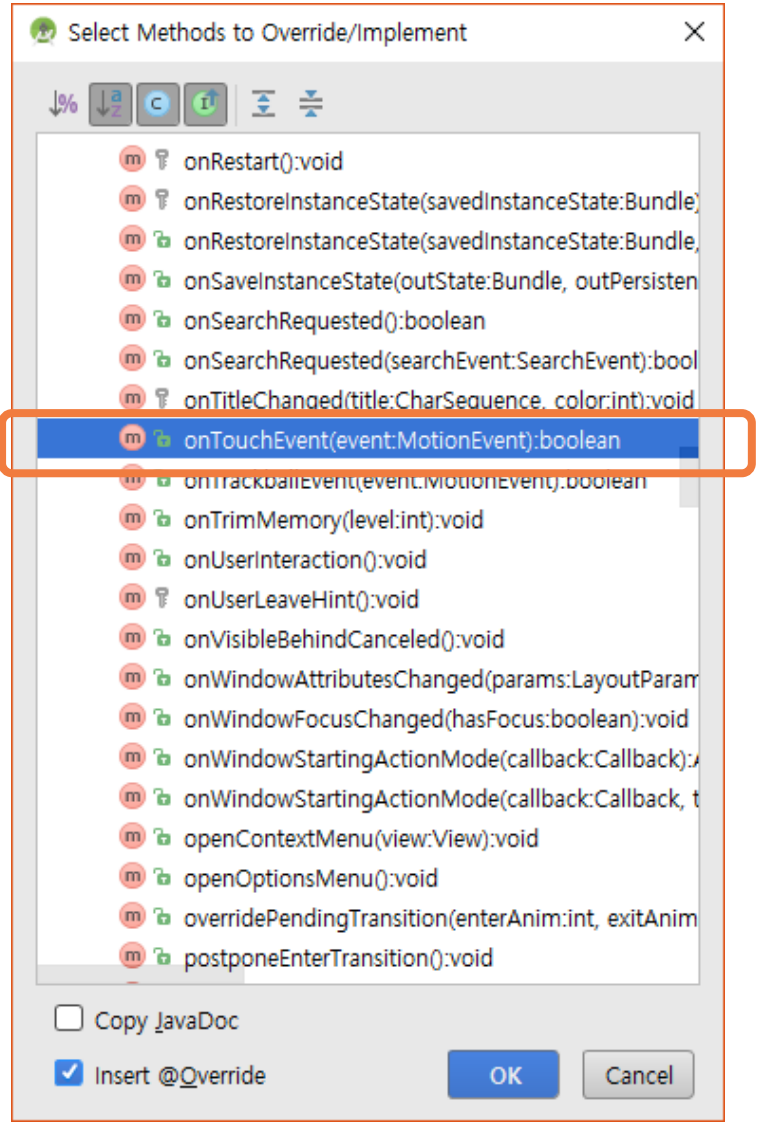
안테나와 배터리 정보창 제거

뷰의 부모에게 위임하여 크기를 얻는다.

onTouchEvent() 메소드 재정의(Override)







- 추가된 onTouchEvent() 매소드

```
@Override
public boolean onTouchEvent(MotionEvent event) {
    return super.onTouchEvent(event);
}
```

➔ 수정은 아래와 같이

```
36      @Override
37      public boolean onTouchEvent(MotionEvent event) {
38          switch (event.getAction()){
39              case MotionEvent.ACTION_DOWN:
40                  break;
41              case MotionEvent.ACTION_MOVE:
42                  int touch_x = (int) event.getX();
43                  int touch_y = (int) event.getY();
44
45                  x = touch_x - iv_Width / 2;
46                  y = touch_y - iv_Height / 2;
47                  // 이미지를 터치한 곳으로 이동 시킨다.
48                  iv_Smile.setX(x);
49                  iv_Smile.setY(y);
50
51                  break;
52              case MotionEvent.ACTION_UP:
53                  break;
54          }
55          return false;
56      }
57  }
58
```

아이콘이 움직일 때 애니메이션을 추가해보자.

- 우선 smile 아이콘을 왼쪽위(원점)에 위치 시킨다.
- 애니메이션을 위해 이전 위치를 저장하기 위한 변수를 추가한다.

```
12  public class MainActivity extends FragmentActivity {
13
14      ImageView iv_Smile;
15
16      float iv_Width = 0f;
17      float iv_Height = 0f;
18
19      float previousX = 0f;
20      float previousY = 0f;
21
22      float x, y;
23
```

- 실제 애니메이션을 구현한다.(다음 페이지)

• 빨간 상자부분을 변경한다.

```
42      @Override
43      public boolean onTouchEvent(MotionEvent event) {
44          switch (event.getAction()){
45              case MotionEvent.ACTION_DOWN:
46                  break;
47              case MotionEvent.ACTION_MOVE:
48                  int touch_x = (int) event.getX();
49                  int touch_y = (int) event.getY();
50
51                  x = touch_x - iv_Width / 2;
52                  y = touch_y - iv_Height / 2;
53                  // 이미지를 터치한 곳으로 이동 시킨다.
54
55                  ObjectAnimator smileX = ObjectAnimator.ofFloat(iv_Smile,  propertyName: "translationX", previousX, x);
56                  ObjectAnimator smileY = ObjectAnimator.ofFloat(iv_Smile,  propertyName: "translationY", previousY, y);
57                  smileX.start();
58                  smileY.start();
59
60                  previousX = x;
61                  previousY = y;
62
63                  break;
64              case MotionEvent.ACTION_UP:
65                  break;
66          }
67          return false;
68      }
69  }
```

변경 전

```
// 이미지를 터치한 곳으로 이동 시킨다.
iv_Smile.setX(x);
iv_Smile.setY(y);
```

변경 후

클래스와 속성/메소드

- 클래스

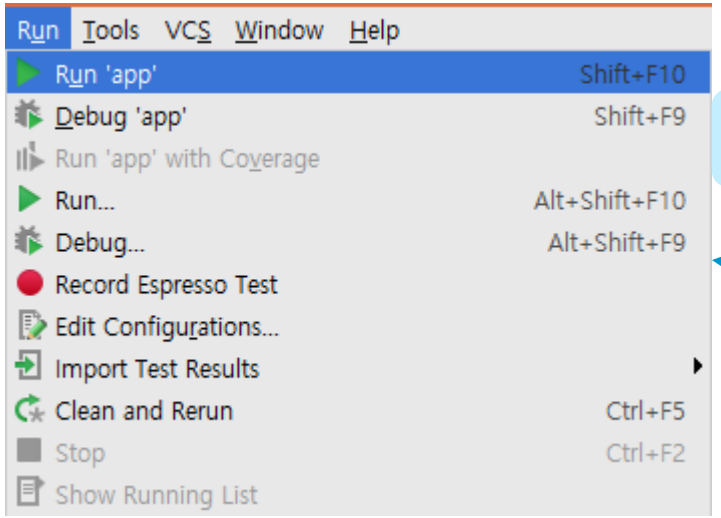
클래스	설명
ObjectAnimator	목표 객체에 대한 애니메이션 특성을 설정

- 메소드

클래스	메소드	설명
ObjectAnimator	static ObjectAnimator **ofFloat**(Object target, String propertyName, float ...values)	values 사이의 애니메이션을 만들고 ObjectAnimator를 반환함
	ObjectAnimator **setDuration**(long duration)	애니메이션 시간 설정, 밀리초 단위이며, 기본값은 300밀리초로 설정됨
	void **start**()	애니메이션 시작

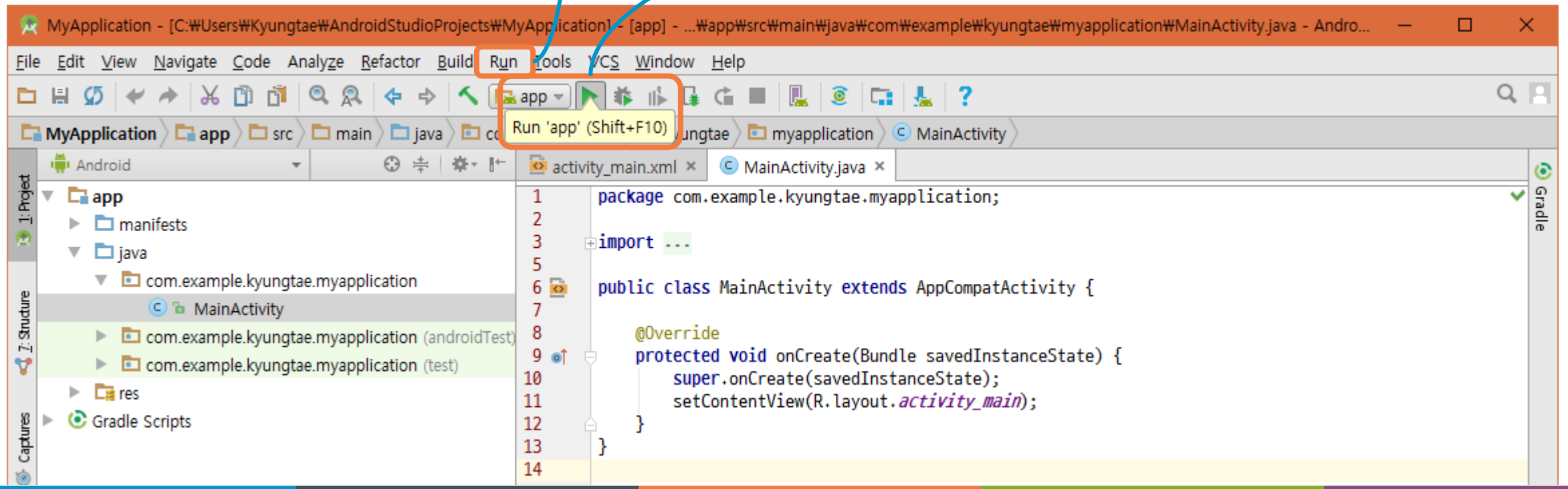
매개변수	설명
target	애니메이션 대상
propertyName	애니메이션 특성 이름
values	시간에 따라 애니메이션 될 값들

Step 3. 프로젝트 실행

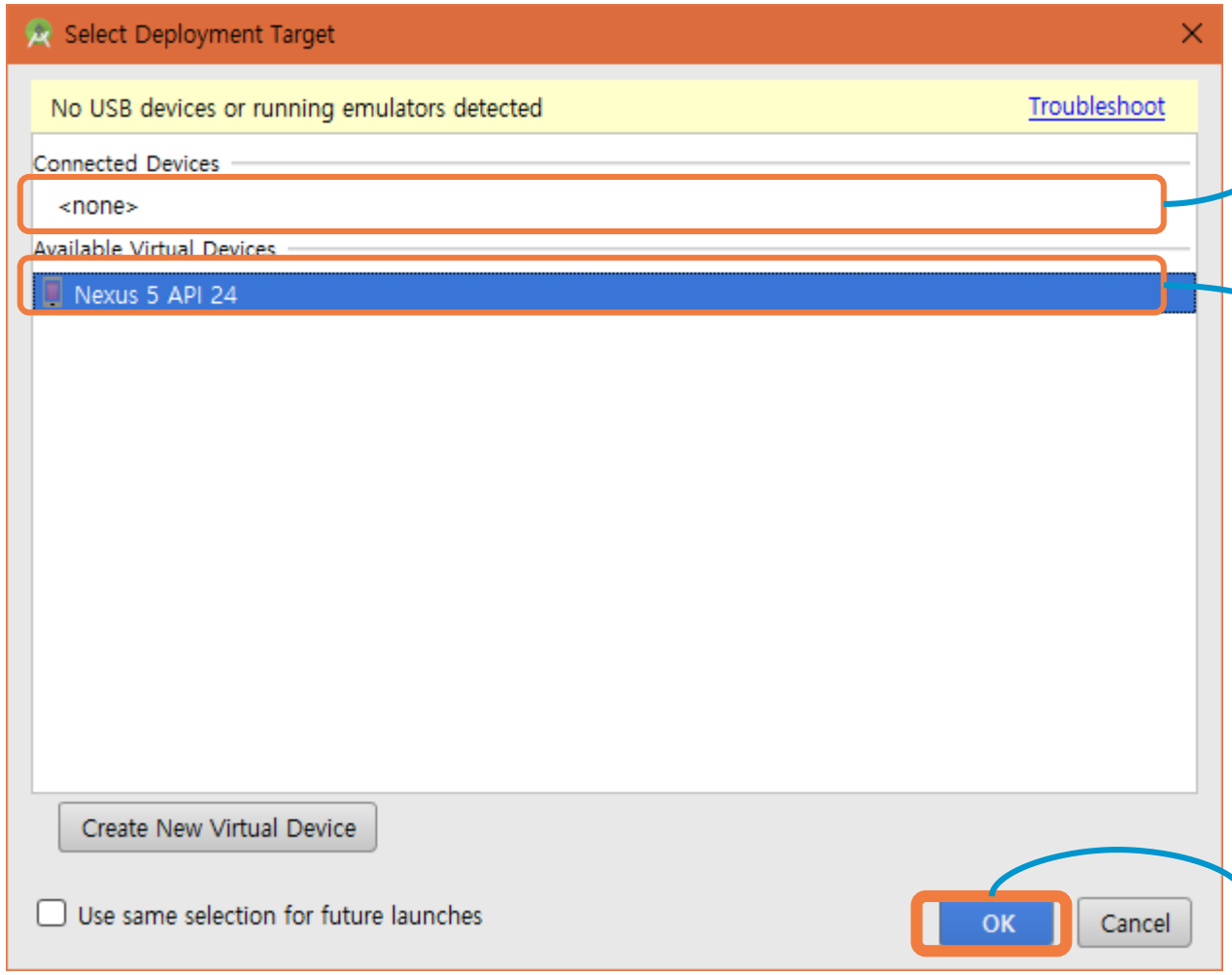


Run → Run 'app' 메뉴 클릭

앱 실행 아이콘 클릭



• AVD 장비 선택하기



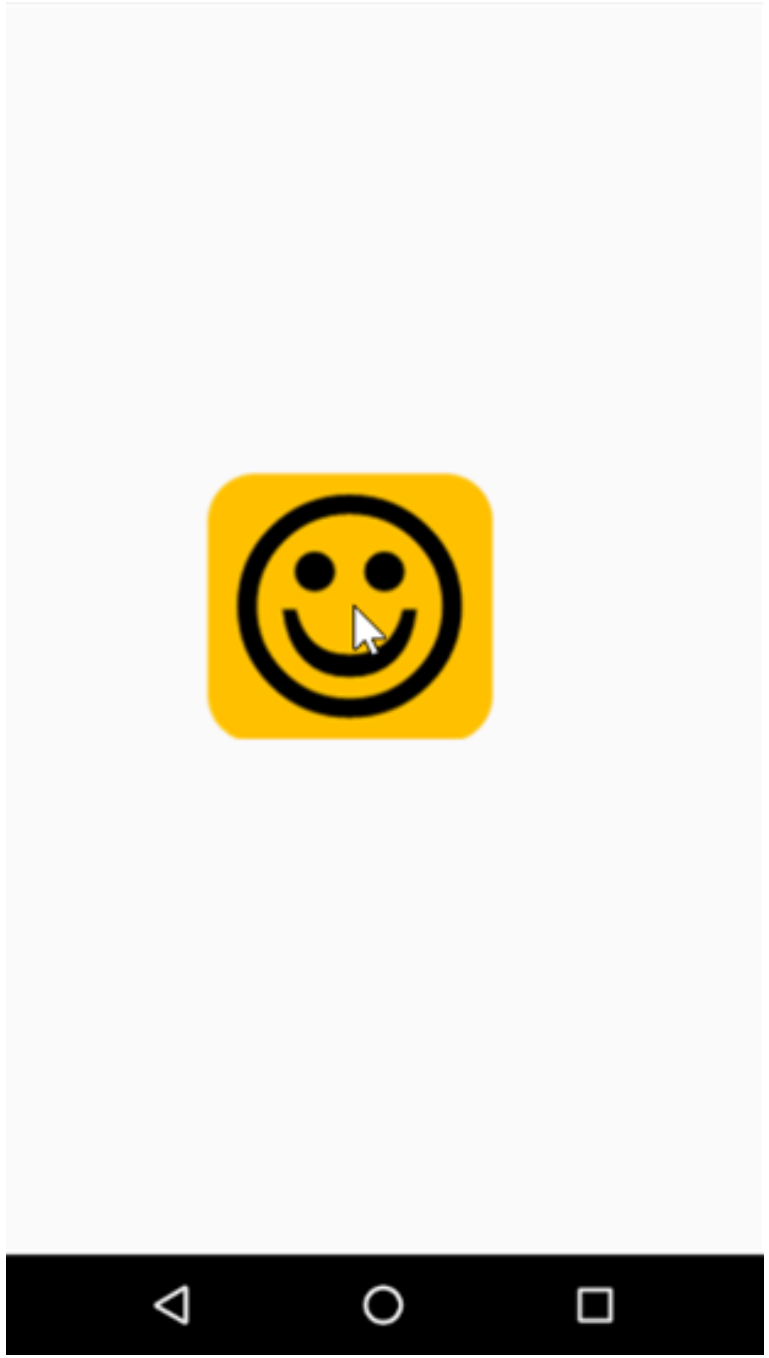
데이터 케이블로 연결된 스마트폰

AVD

스마트폰 또는 AVD를 선택하고 'OK' 버튼을 클릭

- 실행 결과

Outputs



Question & Answer