

Week09.
터치센서



개발환경 구축 절차

2

주 차	수업 내용
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 활용
14	구글 맵과 위치 추적
15	기말 고사



<https://github.com/hopypark>

터치 센서를 활용 앱의 예

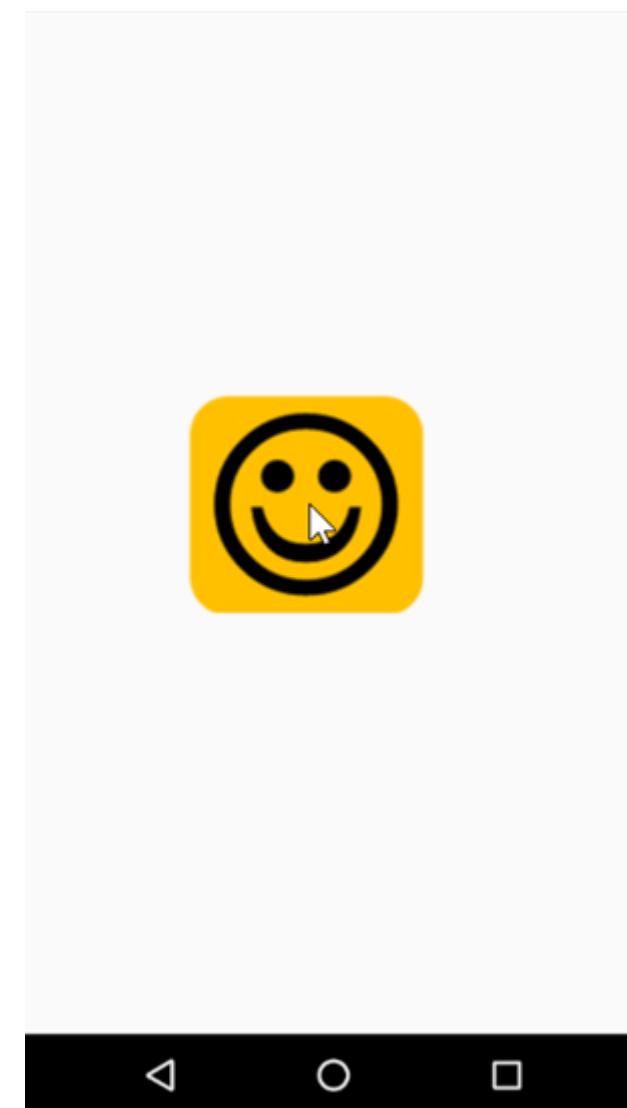
4



(a) 게임레벨 선택



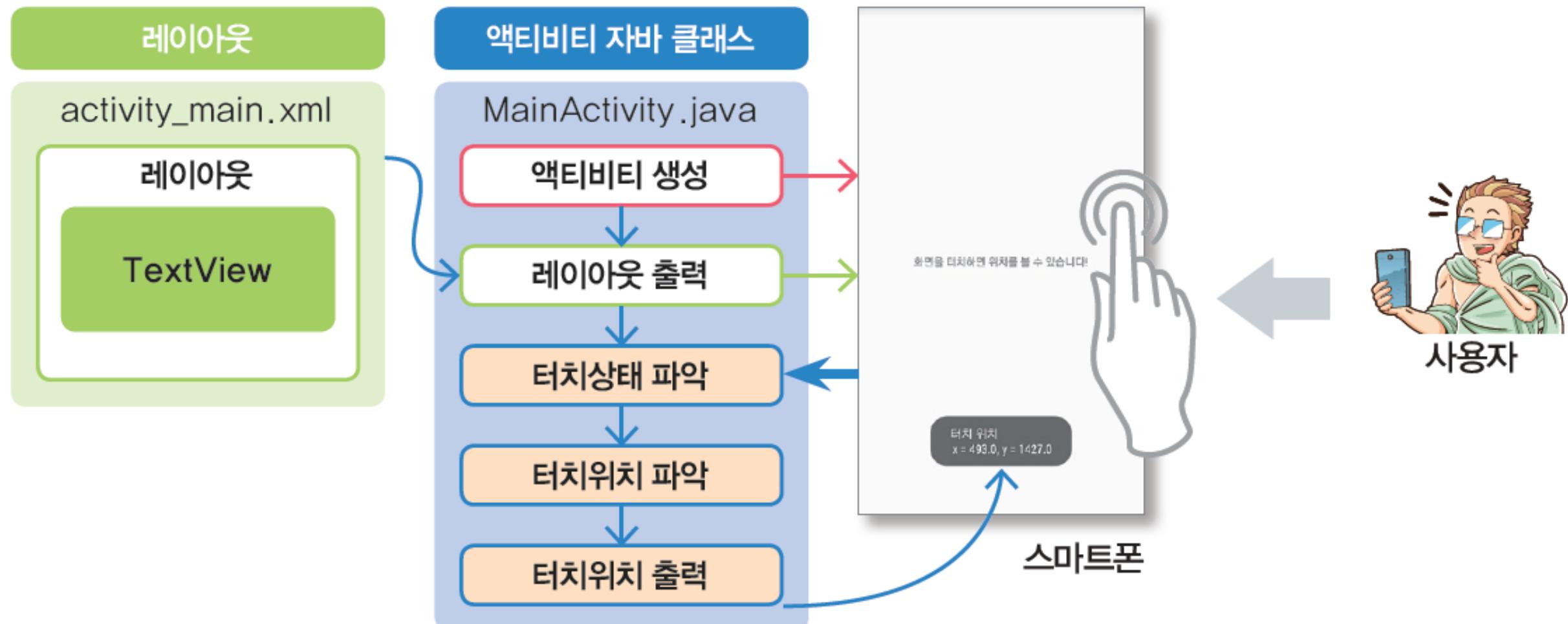
(b) 카카오프렌즈 일렬 배치하기



Follow Me

터치 센서 원리

5

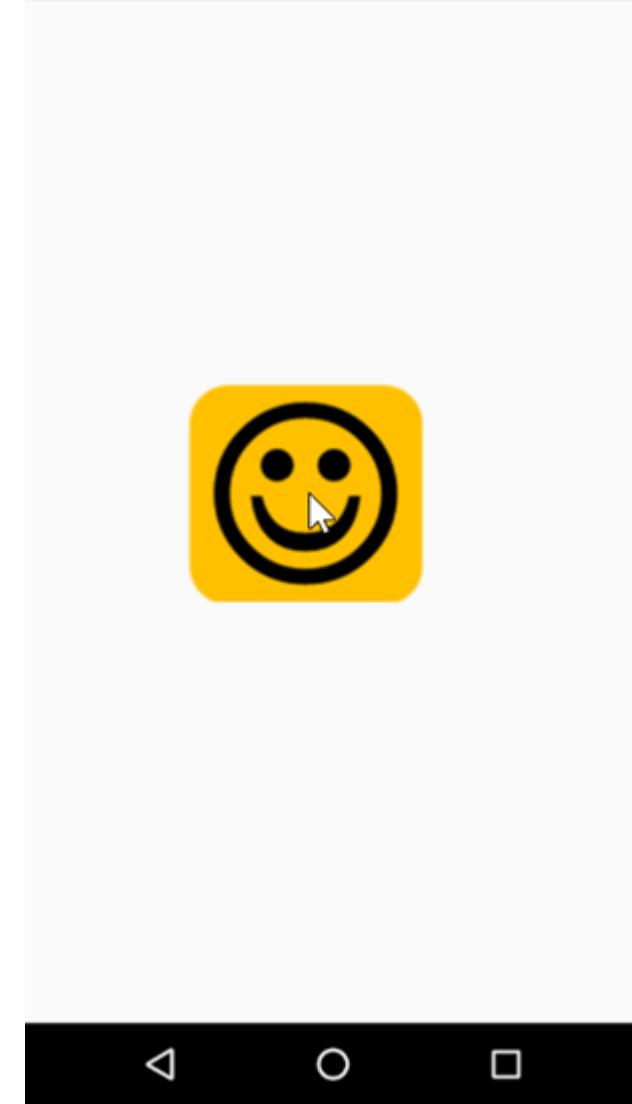


onTouchEvent() 메소드의 매개변수

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

7



Step 1. 프로젝트 생성

8

절차	내용
① 프로젝트 시작	메뉴에서 'File → New Project' 클릭
② 프로젝트 구성	Application Name: TouchEvent
	Company Domain: kyungtae.example.com (디폴트 사용)
	Project Location: ~/AndroidStudioProject/ktpark/TouchEvent
③ 제품형태	Phone and Tablet (사용할 안드로이드 버전 지정: 8.1 Oreo)
④ 액티비티 유형	Empty Activity
⑤ 파일 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	<ul style="list-style-type: none">터치 위치 출력
res	drawable	smile.png	<ul style="list-style-type: none">스마일 이미지
	layout	activity_main.xml	<ul style="list-style-type: none">이미지 리소스 출력
	mipmap	ic_launcher.png	
	values	colors.xml	
		dimens.xml	
		styles.xml	

이미지 리소스



smile.png(drawable)

앱 라벨

텍스트 자원

string

app_name 나를 따라와~~

strings.xml (values)

화면 레이아웃

RelativeLayout

ImageView

id @+id/smile
src @drawable/smile

activity_main.xml (layout)

액티비티 제어

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

    onTouchEvent()
        ObjectAnimator anim =
            ObjectAnimator.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

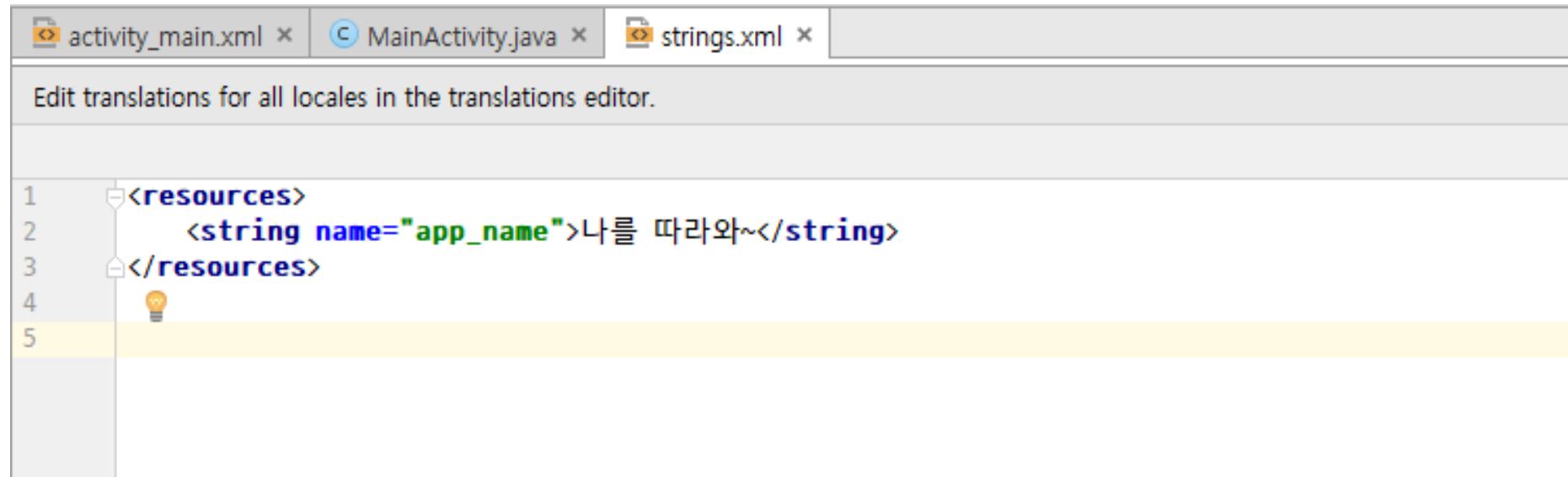
build.gradle(Project)
build.gradle(Module app)
gradle properties
settings.gradle
local.properties

(Gradle Scripts)

Step 2.1 텍스트 자원의 편집

11

- strings.xml

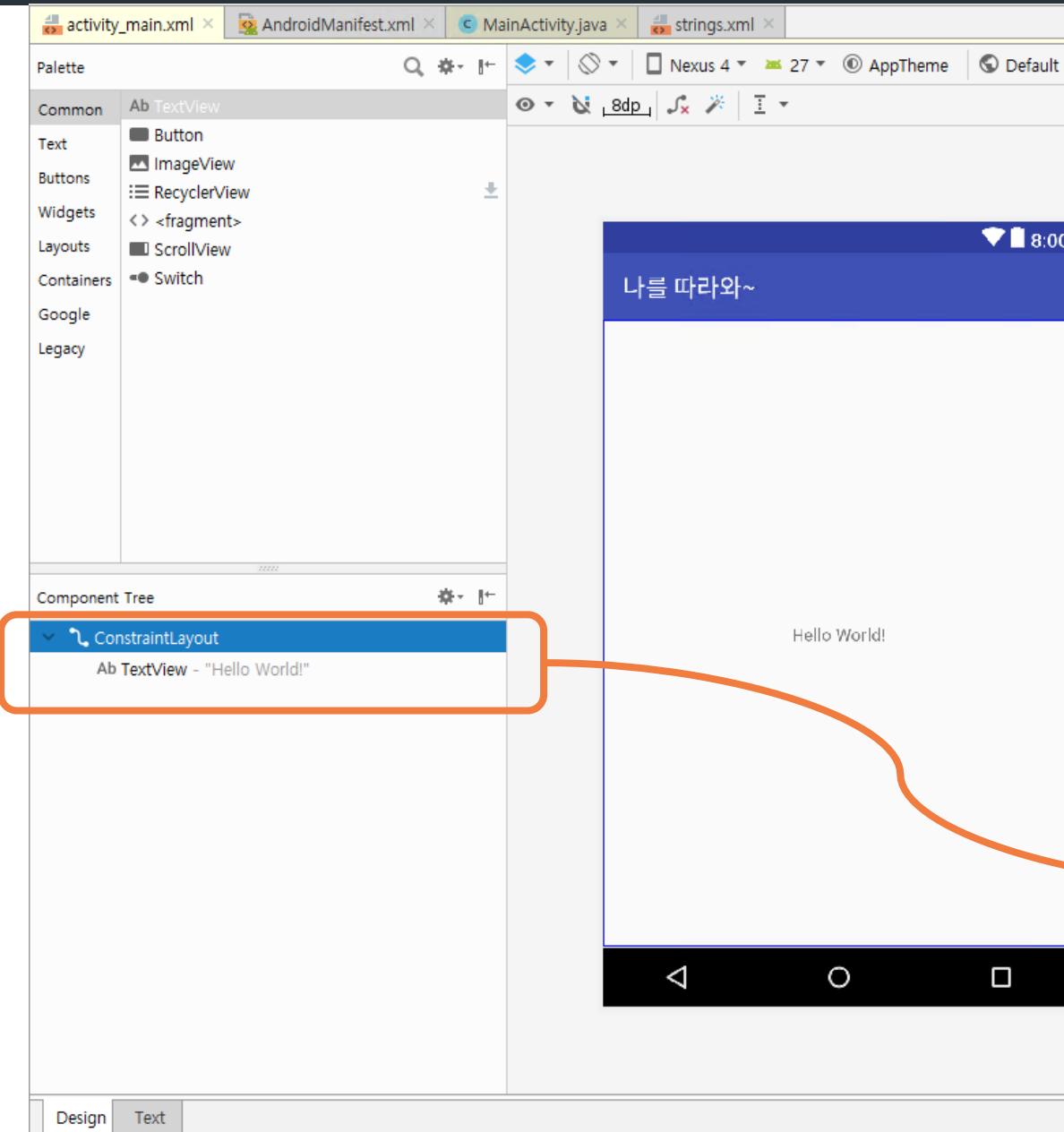


The screenshot shows the Android Studio interface with the tabs "activity_main.xml", "MainActivity.java", and "strings.xml" at the top. The "strings.xml" tab is selected. A message "Edit translations for all locales in the translations editor." is displayed above the code editor. The code editor contains the following XML:

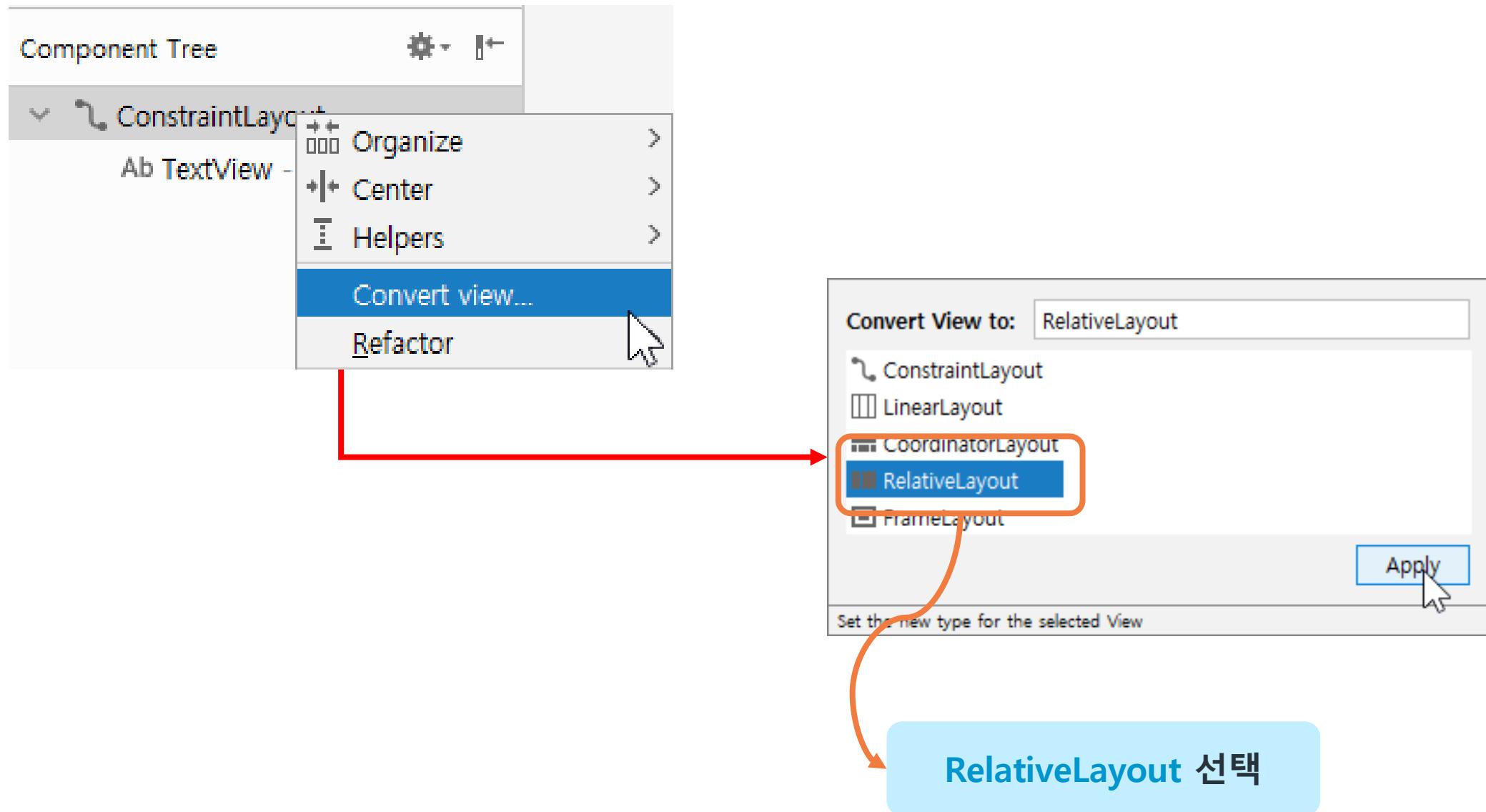
```
1 <resources>
2   <string name="app_name">나를 따라와~</string>
3 </resources>
4
5
```

Line 2 contains a string with the value "나를 따라와~". Line 4 has a yellow lightbulb icon, indicating a warning or suggestion. Line 5 is highlighted with a yellow background.

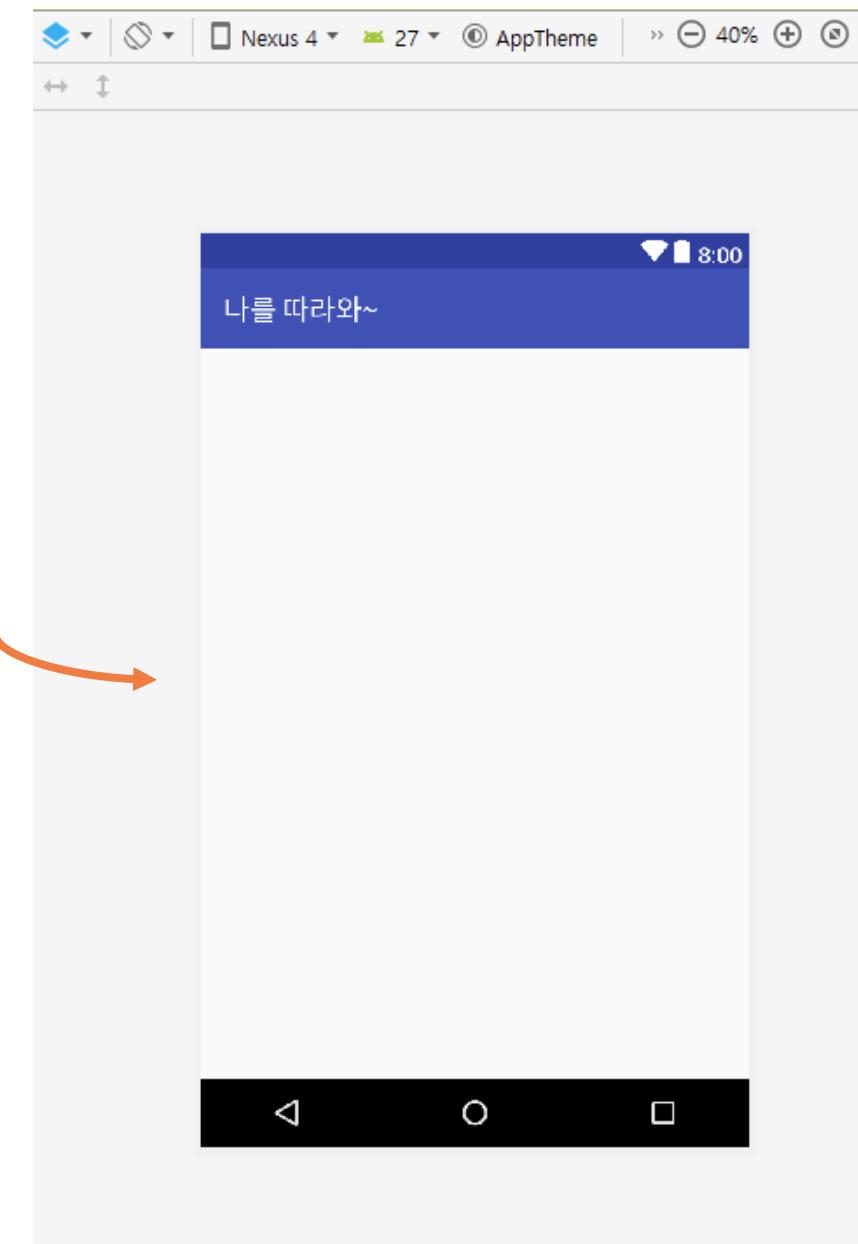
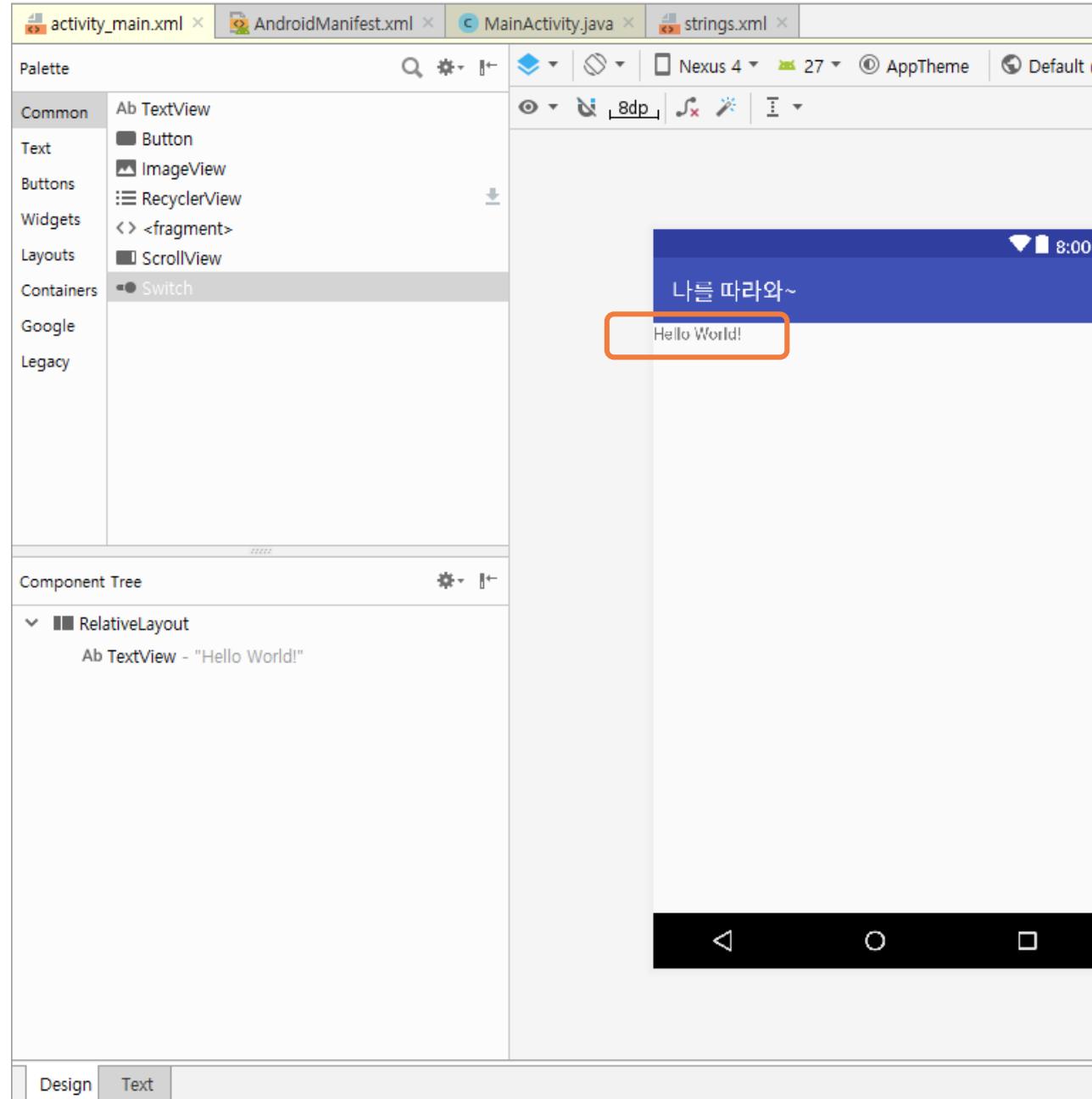
2.2 화면 설계



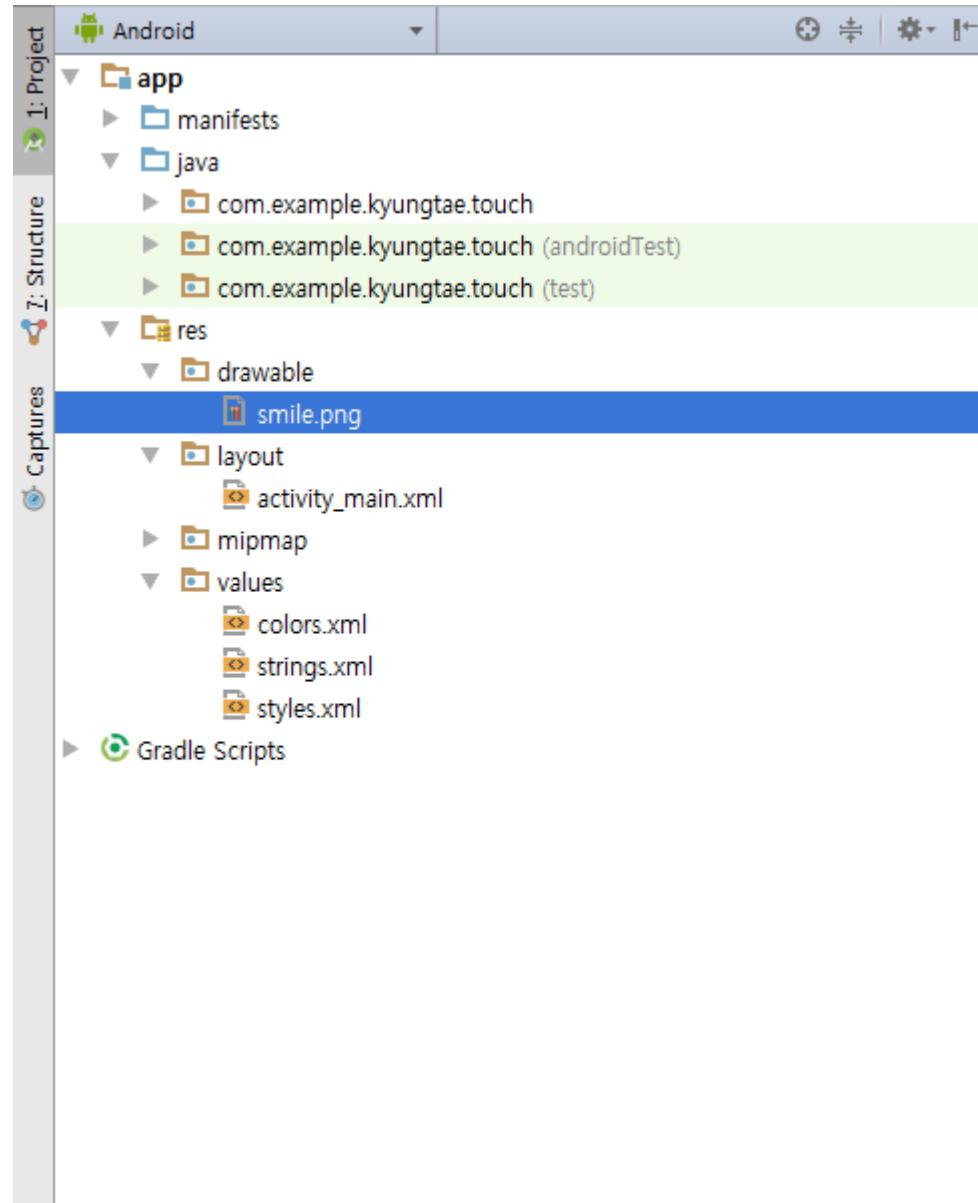
ConstraintLayout →
RelativeLayout으로 변경



• Layout 초기화 설정 - 기본 TextView 삭제

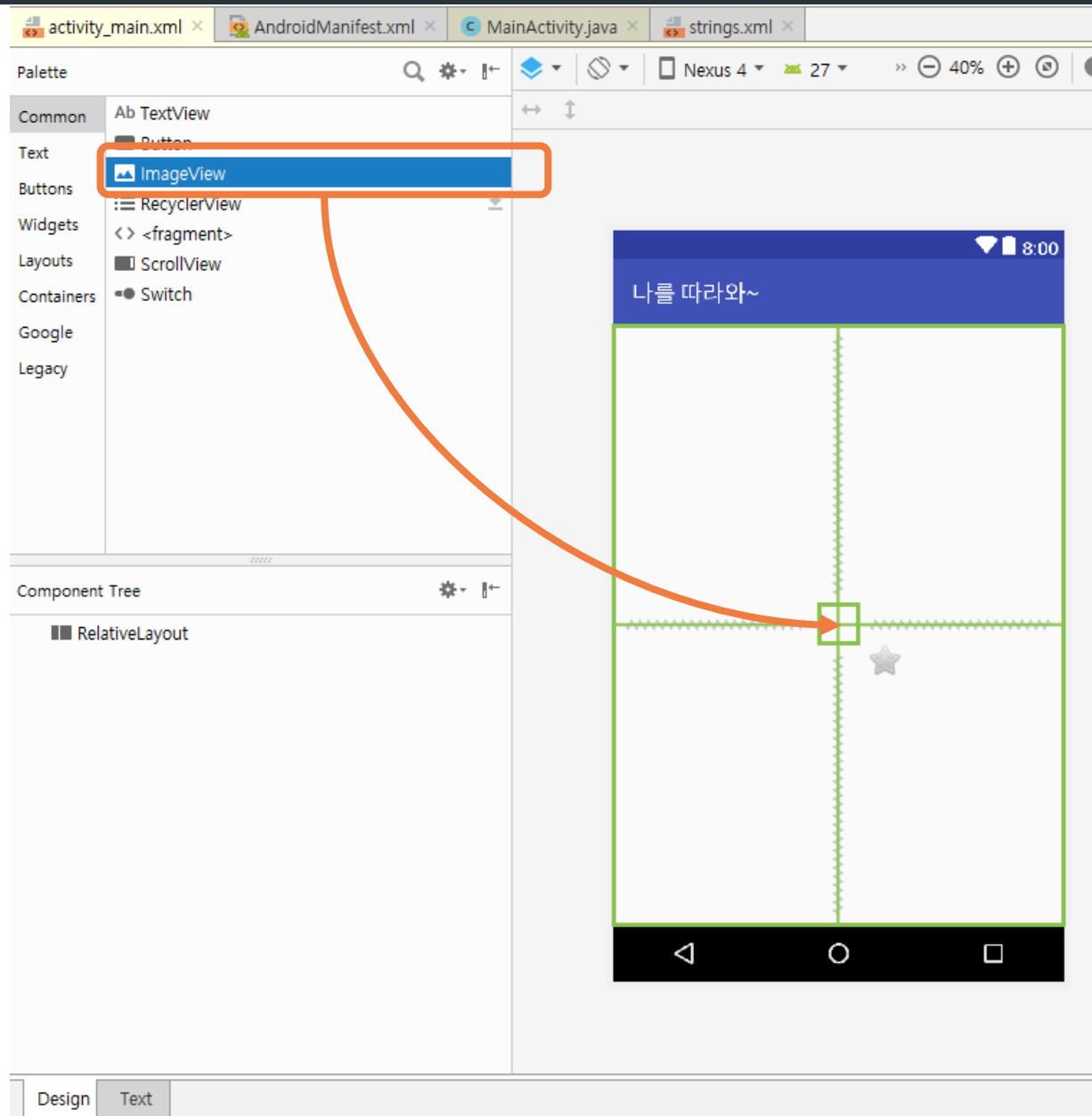


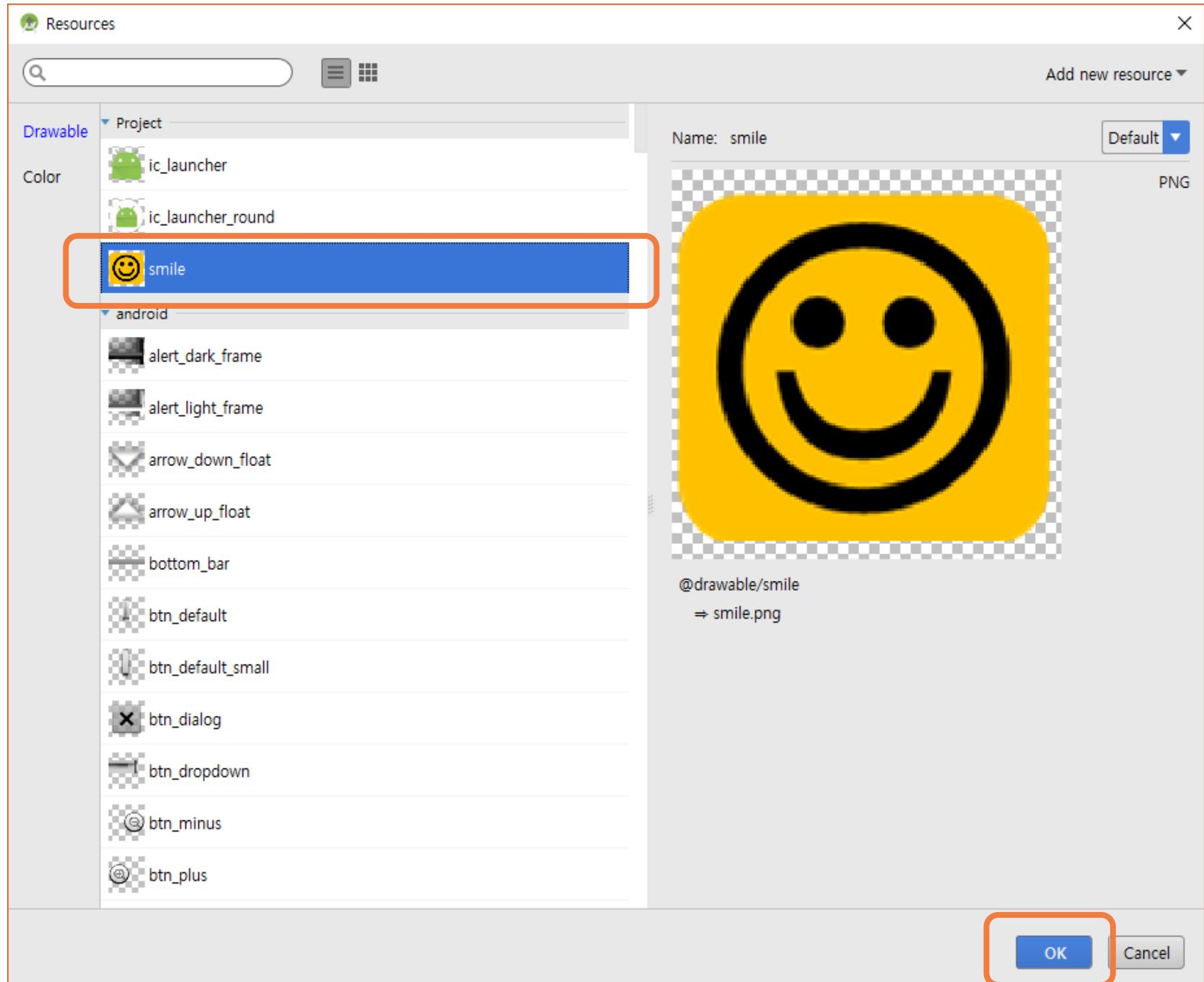
2.3 이미지 리소스 – res/drawable 에 smile.png 추가



smile.png

2.4 화면 설계





- Drawable 항목의 smile을 선택

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

Palette Nexus 4 27 AppTheme 40%

Common Ab TextView
Text Button
Buttons ImageView
Widgets RecyclerView
Layouts <> <fragment>
Containers ScrollView
Google Switch
Legacy

Component Tree
RelativeLayout
smile (ImageView)

Attributes

id	smile
layout_width	wrap_content
layout_height	wrap_content
Layout_Margin	[?, ?, ?, ?, ?]
Padding	[?, ?, ?, ?, ?]
Theme	
elevation	
layout_centerInParent	<input checked="" type="checkbox"/>
srcCompat	@drawable/smile
accessibilityLiveRegion	
accessibilityTraversalAfter	
accessibilityTraversalBefore	
adjustViewBounds	<input type="checkbox"/>
alpha	
autofillHints	
background	
backgroundTint	
backgroundTintMode	
baseline	
baselineAlignBottom	<input type="checkbox"/>
clickable	<input type="checkbox"/>
contentDescription	
contextClickable	<input type="checkbox"/>
cropToPadding	<input type="checkbox"/>
defaultFocusHighlightEnabled	<input type="checkbox"/>
drawingCacheQuality	
duplicateParentState	<input type="checkbox"/>
fadeScrollbars	<input type="checkbox"/>
fadingEdge	
fadingEdgeLength	
filterTouchesWhenObscured	<input type="checkbox"/>
fitsSystemWindows	<input type="checkbox"/>

Design Text

• ImageView의 src 속성 변경

The screenshot shows two instances of the same XML layout file, `activity_main.xml`, side-by-side. A red curved arrow points from the `app:srcCompat` attribute in the top instance to the `android:src` attribute in the bottom instance, indicating a refactoring or migration process.

Top Instance (Original Code):

```
<?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>
```

Bottom Instance (Refactored Code):

```
<?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"
        android:src="@drawable/smile" />
</RelativeLayout>
```

터치한 위치에 아이콘 이미지 위치시키기

- 이미지의 원점과 터치 위치 사이의 오차처리

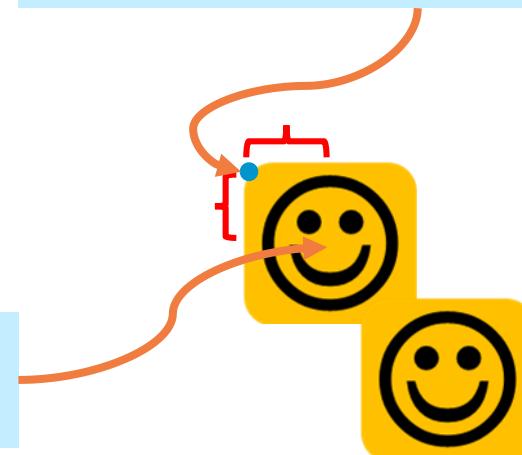
이미지 $x = x - \text{이미지 폭}/2;$

이미지 $y = y - \text{이미지 높이}/2;$

원점(0,0)

터치 위치:
(x, y)

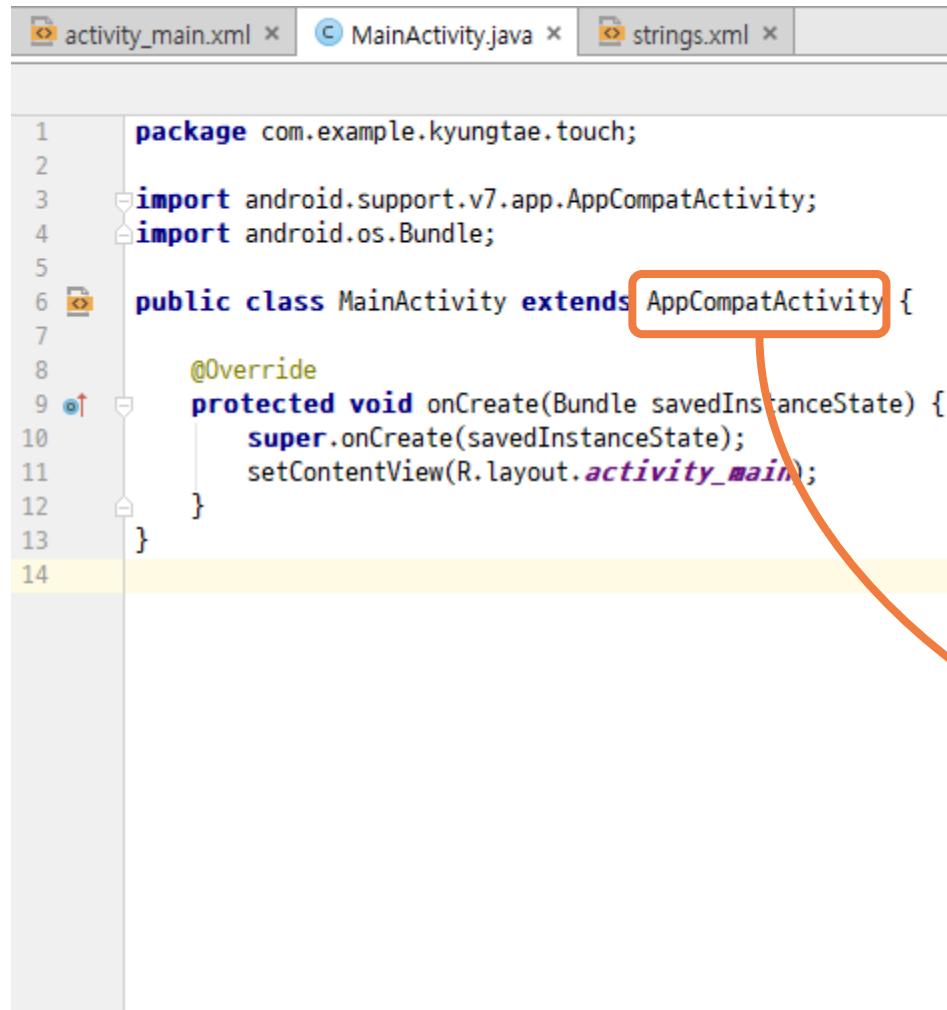
이미지 원점 위치:
($x - \text{이미지 폭}/2, y - \text{이미지 높이}/2$)



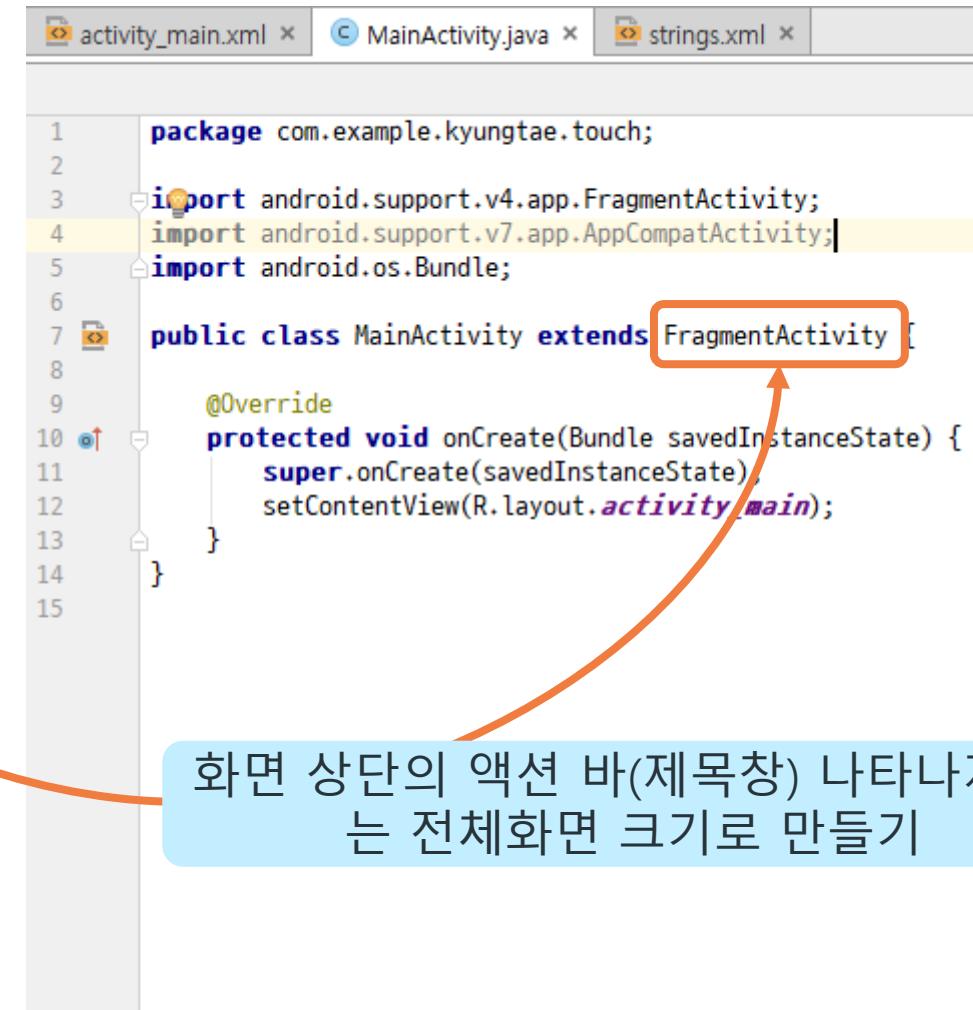
2.5 Activity 제어(MainActivity.java)

21

- 화면을 전체화면 크기로 만들기 위한 액티비티 상속 클래스 변경



```
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
```



```
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

MainActivity.java

```
1 package com.example.user.touchevent;  
2  
3 import android.animation.ObjectAnimator;  
4 import android.support.v4.app.FragmentActivity;  
5 import android.os.Bundle;  
6 import android.util.DisplayMetrics;  
7 import android.view.MotionEvent;  
8 import android.view.WindowManager;  
9 import android.widget.ImageView;  
10  
11 public class MainActivity extends FragmentActivity {  
12  
13     ImageView iv_Smile; // smile 이미지용 객체  
14     float previousX = 0f; // 이전 x 위치  
15     float previousY = 0f; // 이전 y 위치  
16  
17     float screen_Width = 0f; // 화면의 폭  
18     float screen_Height = 0f; // 화면의 높이  
19  
20     float iv_Width = 0f; // 이미지의 폭  
21     float iv_Height = 0f; // 이미지의 높이  
22  
23     float x, y; // 아이콘의 임시 x,y 위치  
24 }
```



10:41

타이틀 – 안테나, 배터리 정보
표시 장 없애기

화면 중앙에 이미지가 위치



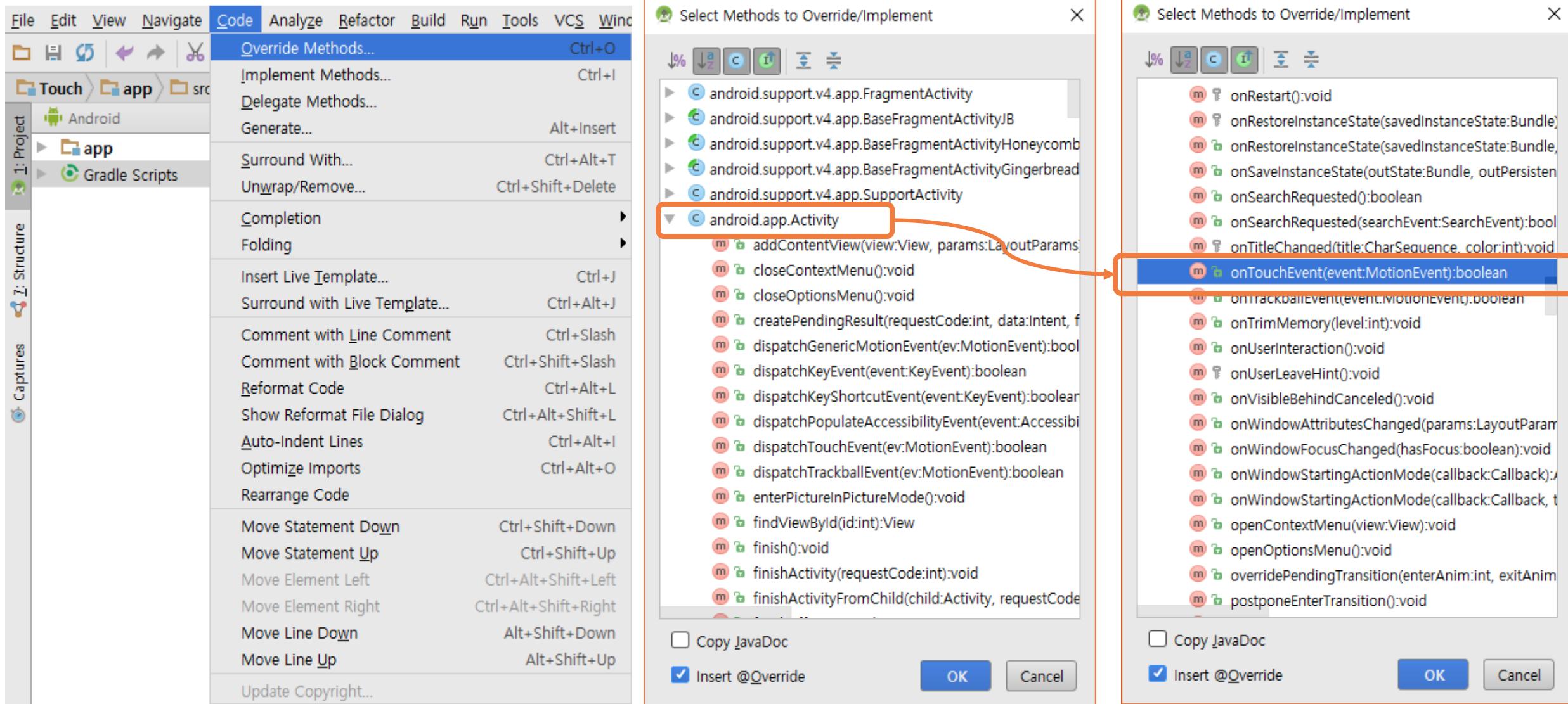
```
25  
26 @Override  
27     protected void onCreate(Bundle savedInstanceState) {  
28         super.onCreate(savedInstanceState);  
29  
30         // 윈도우를 전체 화면 크기로 설정  
31         getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,  
32                         WindowManager.LayoutParams.FLAG_FULLSCREEN);  
33  
34         setContentView(R.layout.activity_main);  
35         iv_Smile = (ImageView) findViewById(R.id.smile);  
36  
37         // 화면 전체 크기 얻기  
38         DisplayMetrics displayMetrics = getApplicationContext().getResources().getDisplayMetrics();  
39         screen_Width = displayMetrics.widthPixels;  
40         screen_Height = displayMetrics.heightPixels;  
41  
42         // 이미지 크기 얻기  
43         iv_Smile.measure(iv_Smile.getMeasuredWidth(), iv_Smile.getMeasuredHeight());  
44         iv_Height = iv_Smile.getMeasuredHeight();  
45         iv_Width = iv_Smile.getMeasuredWidth();  
46  
47         // 화면의 중앙위치 계산  
48         x = screen_Width / 2 - iv_Width / 2;  
49         y = screen_Height / 2 - iv_Height / 2;  
50         // 화면의 중앙으로 이미지를 위치시킴  
51         iv_Smile.setX(x);  
52         iv_Smile.setY(y);  
53         // 이전 위치로 중앙을 설정  
54         previousX = x;  
55         previousY = y;  
56     }  
57 }
```

안테나와 배터리 정보창 제거

이미지가 위치할 중앙 위치

onTouchEvent() 메소드 재정의(Override)

25



- 추가된 onTouchEvent() 매소드

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

→ 수정은 다음 페이지

```
58  
59     @Override  
60     public boolean onTouchEvent(MotionEvent event) {  
61  
62         switch (event.getAction()){  
63             case MotionEvent.ACTION_DOWN:  
64                 break;  
65             case MotionEvent.ACTION_MOVE:  
66                 int touch_x = (int) event.getX();  
67                 int touch_y = (int) event.getY();  
68  
69                 x = touch_x - iv_Width / 2;  
70                 y = touch_y - iv_Height / 2;  
71  
72                 ObjectAnimator smileX = ObjectAnimator.ofFloat(iv_Smile, "translationX", previousX, x);  
73                 smileX.start();  
74                 ObjectAnimator smileY = ObjectAnimator.ofFloat(iv_Smile, "translationY", previousY, y);  
75                 smileY.start();  
76  
77                 previousX = x;  
78                 previousY = y;  
79  
80                 break;  
81             case MotionEvent.ACTION_UP:  
82                 break;  
83         }  
84     return false;  
85 }  
86 }
```

클래스와 속성/메소드

- 클래스

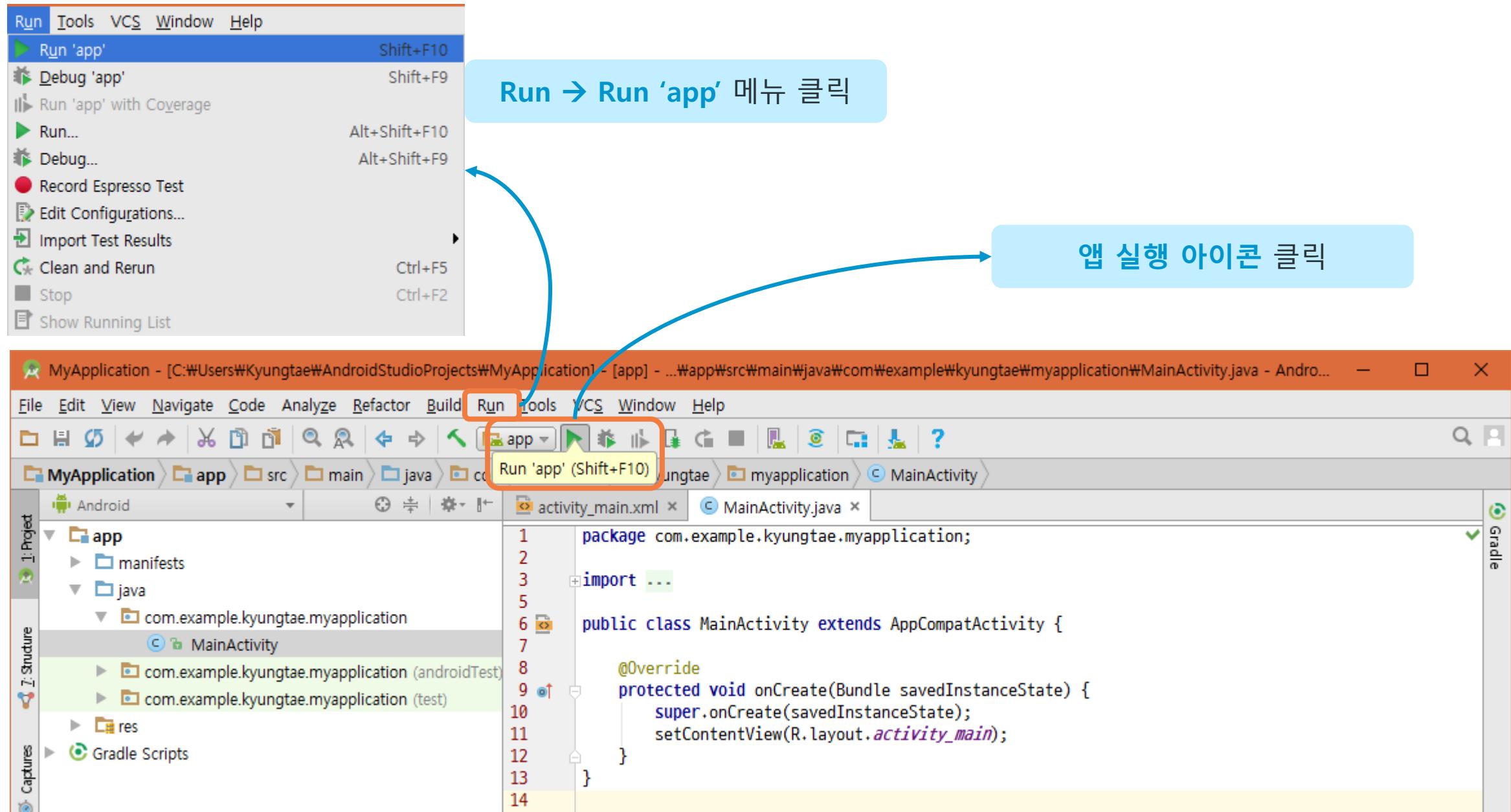
클래스	설명
ObjectAnimator	목표 객체에 대한 애니메이션 특성을 설정

- 메소드

클래스	메소드	설명								
ObjectAnimator	<code>static ObjectAnimator ofFloat(Object target, String propertyName, float ...values)</code>	values 사이의 애니메이션을 만들고 ObjectAnimator를 반환함 <table><thead><tr><th>매개변수</th><th>설명</th></tr></thead><tbody><tr><td>target</td><td>애니메이션 대상</td></tr><tr><td>propertyName</td><td>애니메이션 특성 이름</td></tr><tr><td>values</td><td>시간에 따라 애니메이션 될 값들</td></tr></tbody></table>	매개변수	설명	target	애니메이션 대상	propertyName	애니메이션 특성 이름	values	시간에 따라 애니메이션 될 값들
매개변수	설명									
target	애니메이션 대상									
propertyName	애니메이션 특성 이름									
values	시간에 따라 애니메이션 될 값들									
<code>ObjectAnimator setDuration(long duration)</code>	애니메이션 시간 설정, 밀리초 단위이며, 기본값은 300밀리초로 설정됨									
<code>void start()</code>	애니메이션 시작									

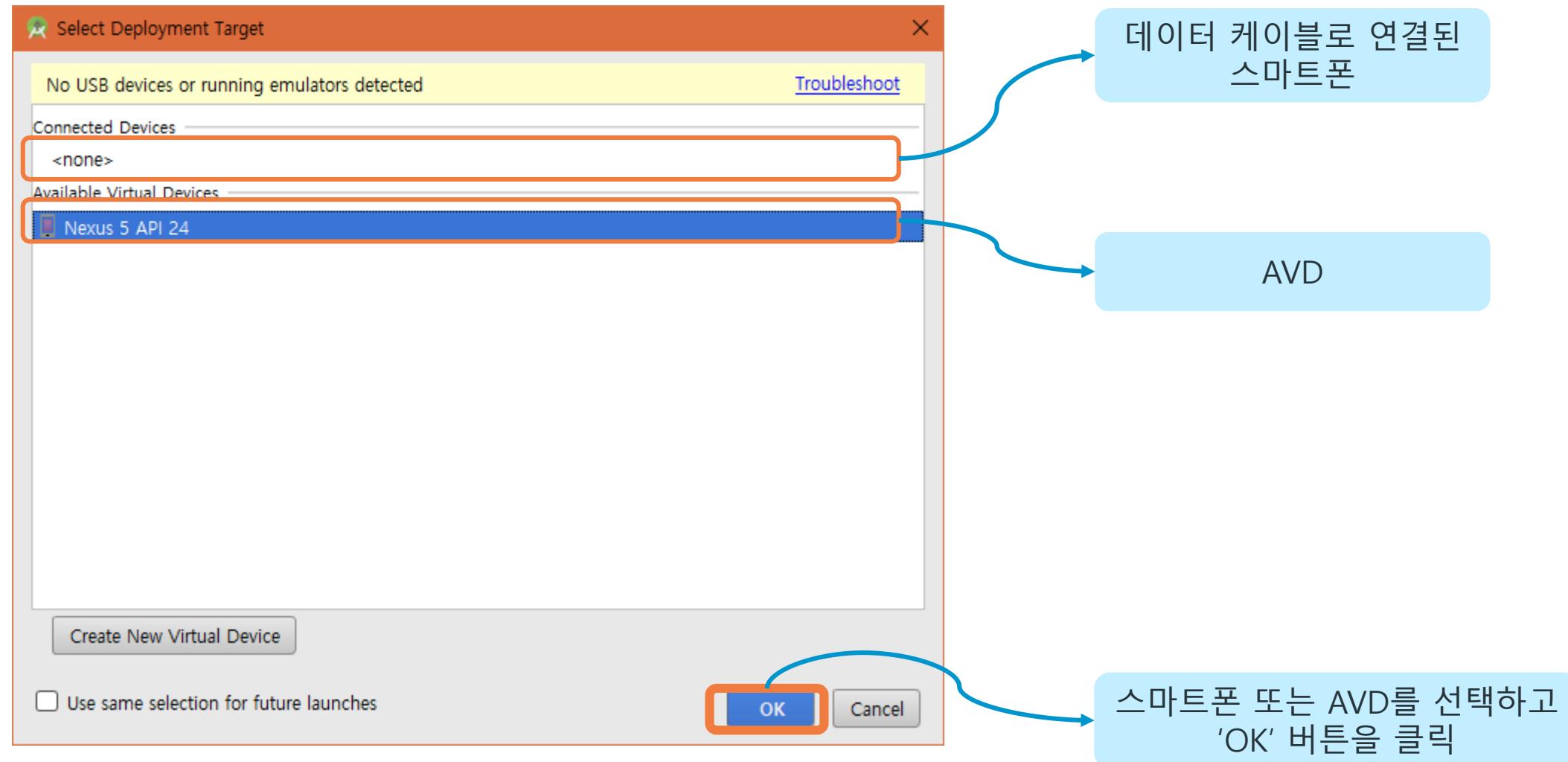
Step 3. 프로젝트 실행

29



• AVD 장비 선택하기

30



- 실행 결과



O outputs





question

&



answer

