

Week05. 이벤트 처리와 액티비티 간 이동



안드로이드 앱

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

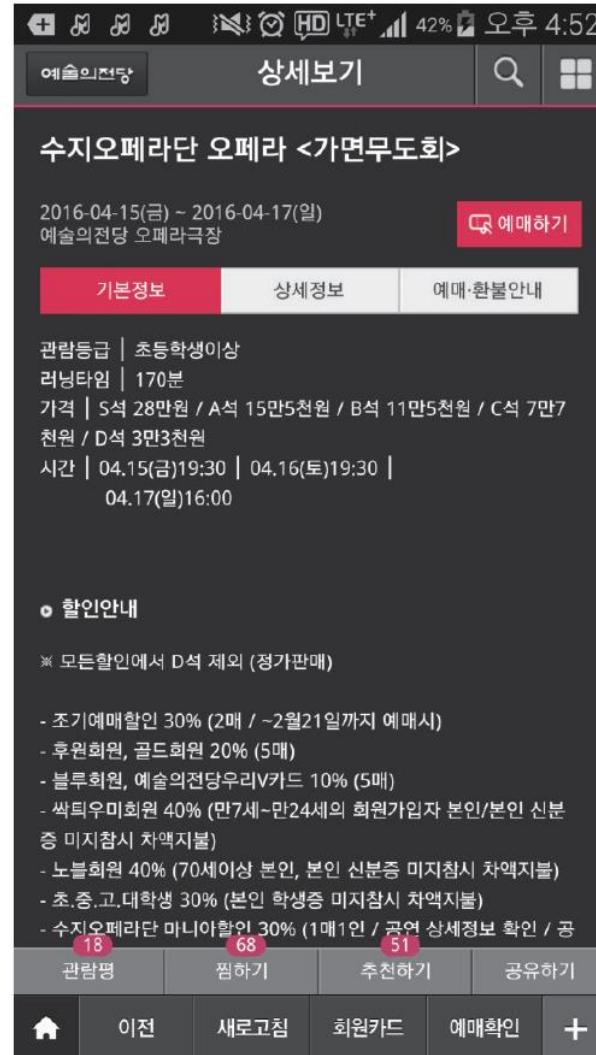


<http://github.com/hopypark/>

사용자 클릭 처리 앱의 예



액티비티 1(초기 화면)

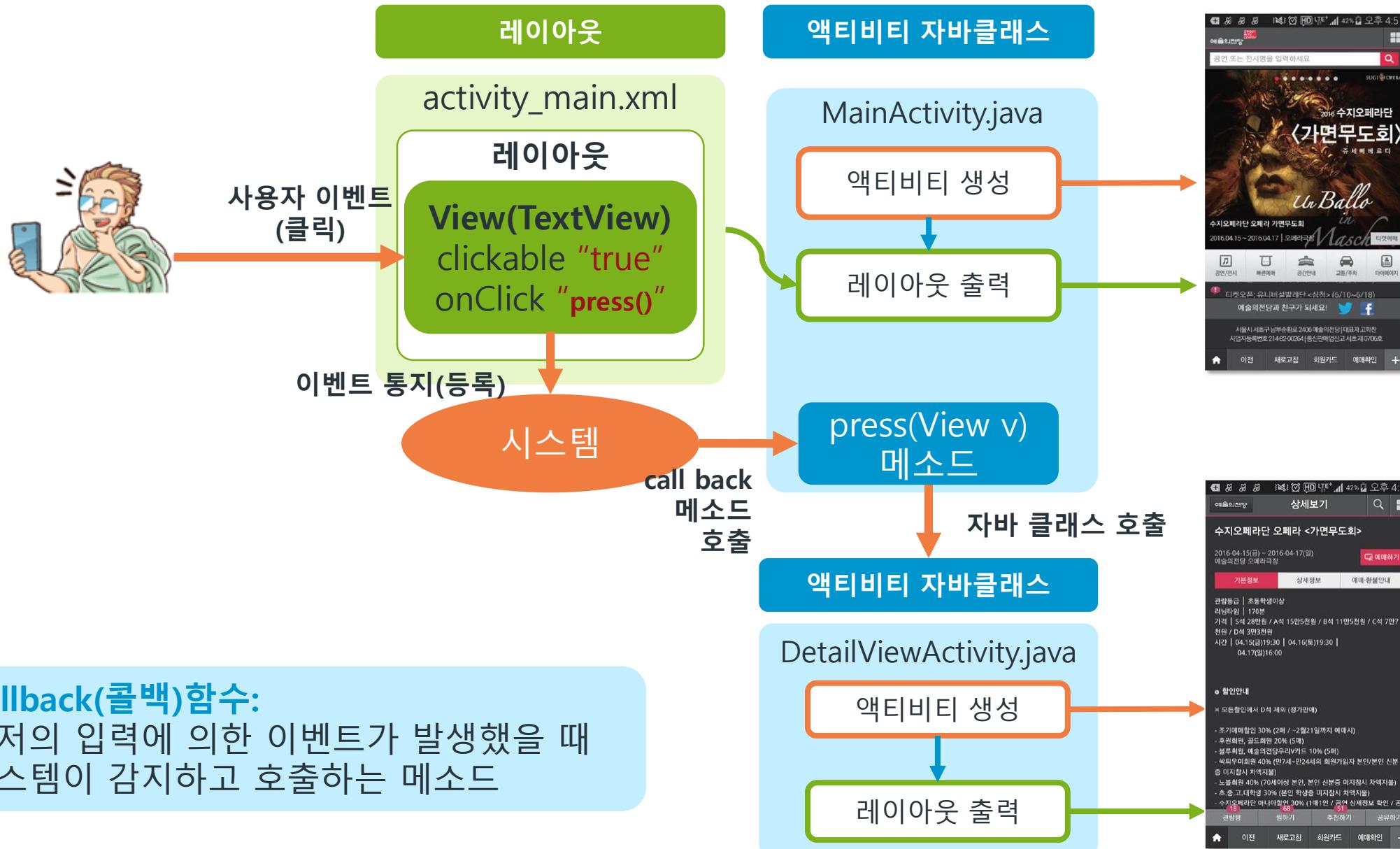


액티비티 2(상세 화면)

● 예술의 전당 앱

사용자 클릭 처리와 콜백 메소드

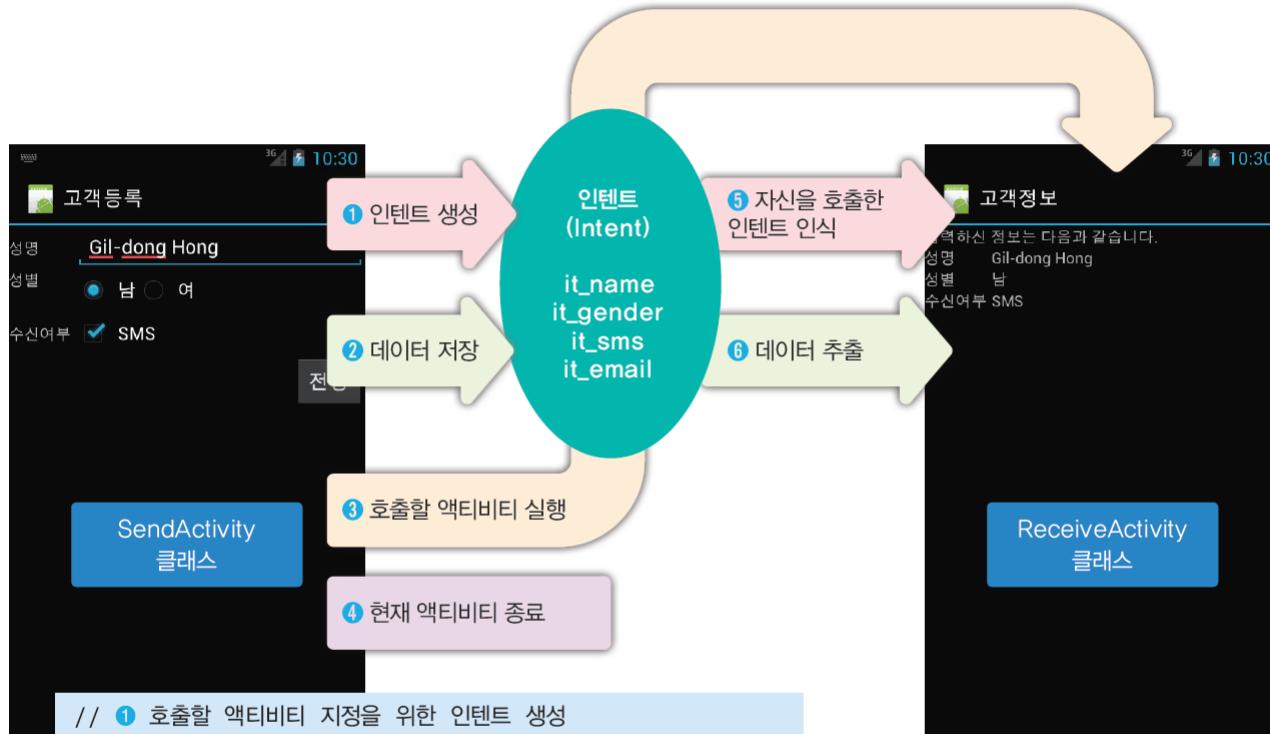
5



액티비티 간 이동과 정보 전달 – 인텐트(Intent)

6

```
// ⑤ 자신을 호출한 인텐트 생성  
Intent it = getIntent();  
// ⑥ 인텐트에서 데이터 추출  
String str_name = it.getStringExtra("it_name");
```



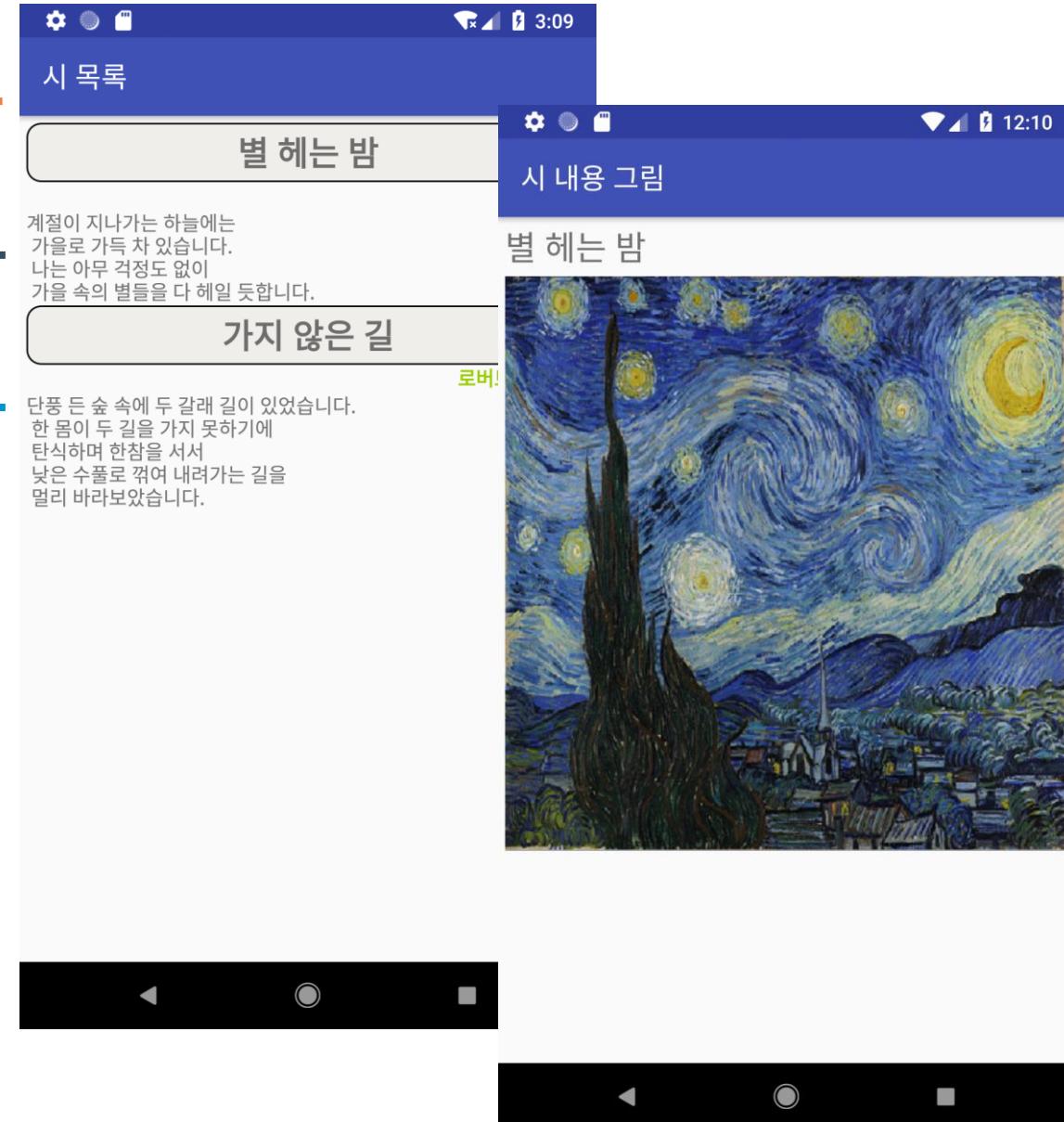
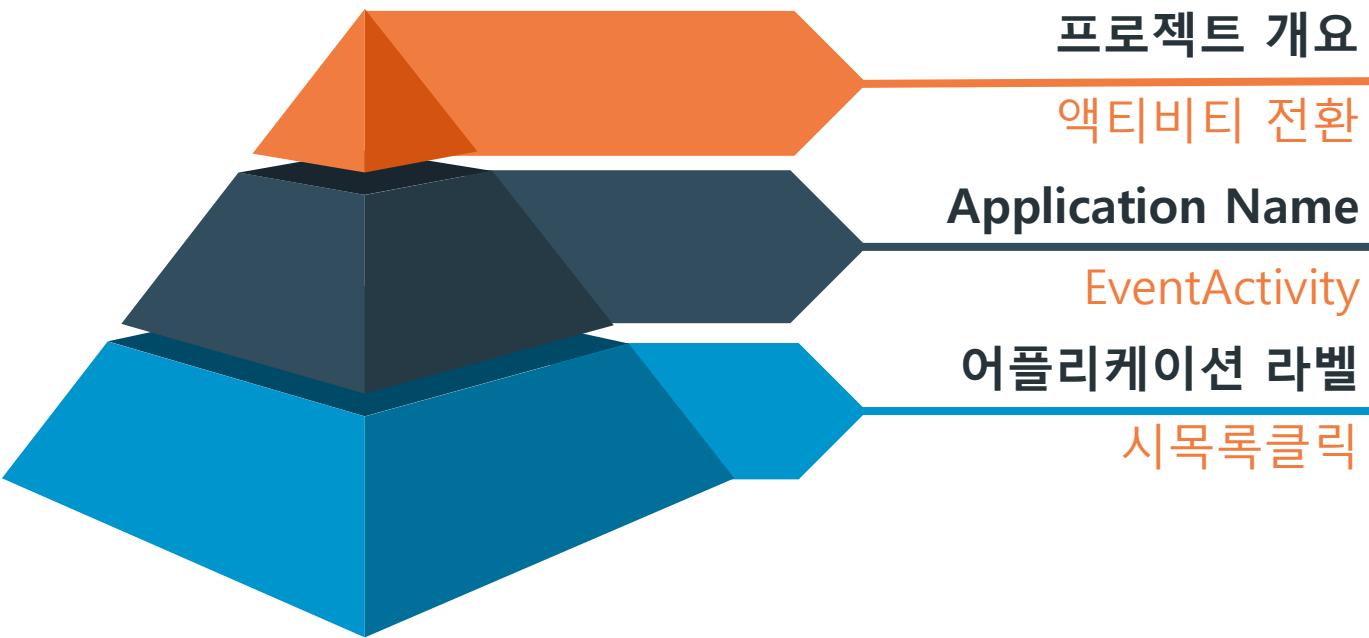
```
// ① 호출할 액티비티 지정을 위한 인텐트 생성  
Intent it= new Intent(this, ReceiveActivity.class);  
// ② 인텐트에 데이터 저장(성명)  
// str_name의 값을 it_name 변수에 저장함  
it.putExtra("it_name", str_name);  
// ③ 인텐트에서 지정한 액티비티 실행  
startActivity(it);  
// ④ 현재 액티비티 종료  
finish();
```

- ① 인텐트 생성
- ② 데이터 저장
- ③ 호출할 액티비티 실행
- ④ 현재 액티비티 종료
- ⑤ 호출된 액티비티에서 자신을 호출한 인텐트 인식
- ⑥ 데이터 추출

인텐트에는
“it_name” = “홍길동”
“it_gender” = “남”
...

Step 0. 프로젝트 개요

15



Step 1. 프로젝트 생성

16

절차	내용
① 프로젝트 시작	메뉴에서 'File → New Project' 클릭
② 프로젝트 구성	Application Name: EventActivity
	Company Domain: 사용자계정.example.com(디폴트 사용)
③ 제품형태	Phone and Tablet(사용할 안드로이드 버전 지정: Android 8.1 Oreo)
④ 액티비티 유형	Empty Activity
⑤ 파일 옵션	Activity Name: MainActivity
	Layout Name: activity_main

Step 2. 파일 편집

17

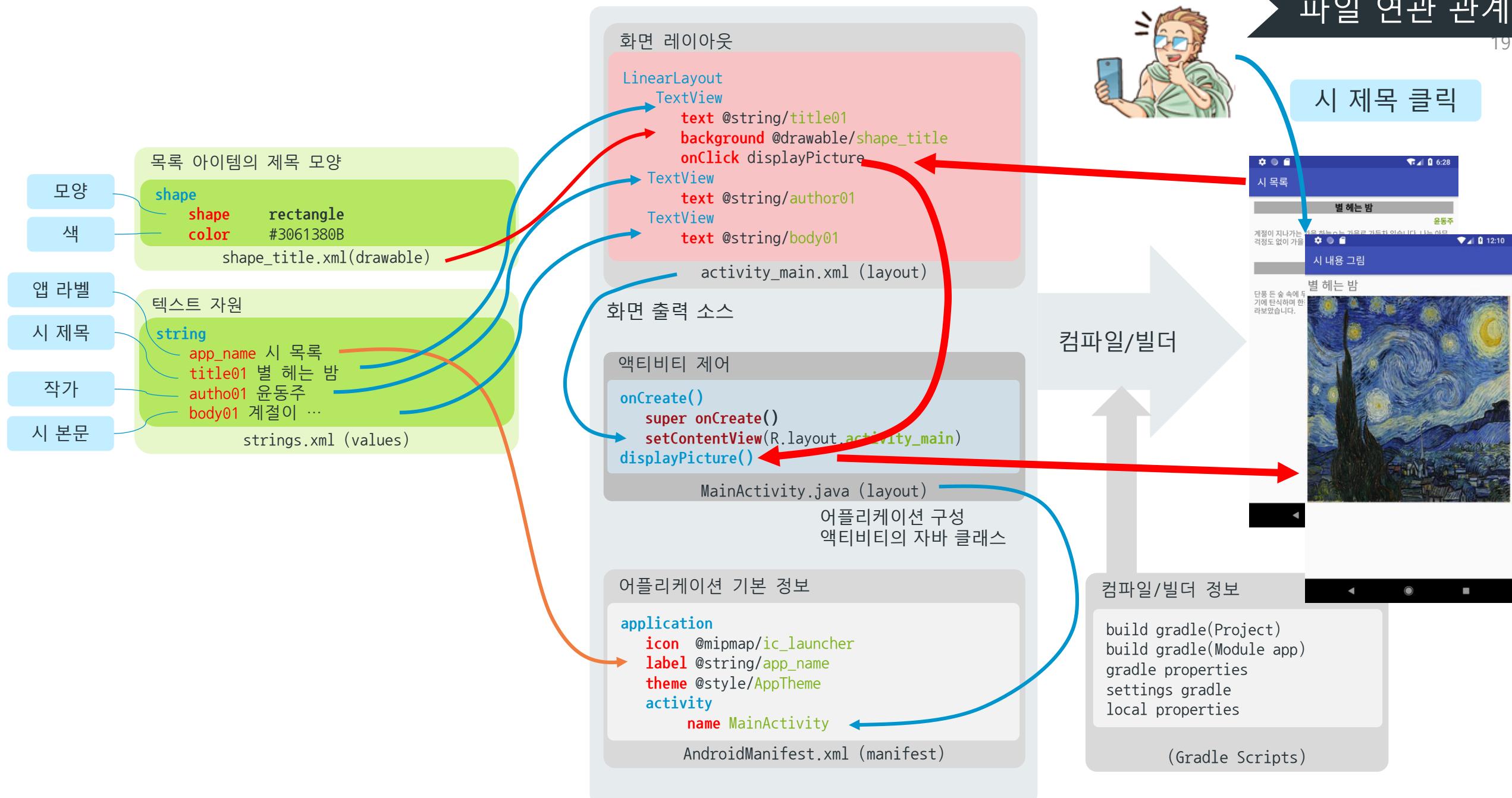
모듈	폴더	소스 파일	편집 내용
manifests		AndroidManifest.xml	
java	com.example.kyungtae.eventActivity	MainActivity.java	첫번째 시작 액티비티(화면)
		DetailViewActivity.java	두번째 액티비티(화면)
res	drawable	starry_night.png	<ul style="list-style-type: none"> 명화 이미지
		still_life_with_kettle.png	<ul style="list-style-type: none"> 명화 이미지
		shape_title.xml	<ul style="list-style-type: none"> 시 제목 출력모양 설계 (배경색, 패딩, 모서리)
	layout	activity_main.xml	<ul style="list-style-type: none"> 시 목록 화면 구성 (제목, 작가, 설명)
		activity_detail_view.xml	<ul style="list-style-type: none"> 두번째 화면 구성 – 명화 출력
	mipmap	ic_launcher.png	
	values	dimens.xml	
		strings.xml	<ul style="list-style-type: none"> 시 목록 아이템 (제목, 작가, 설명)
		styles.xml	

Step 2.1 이미지 파일 복사

18

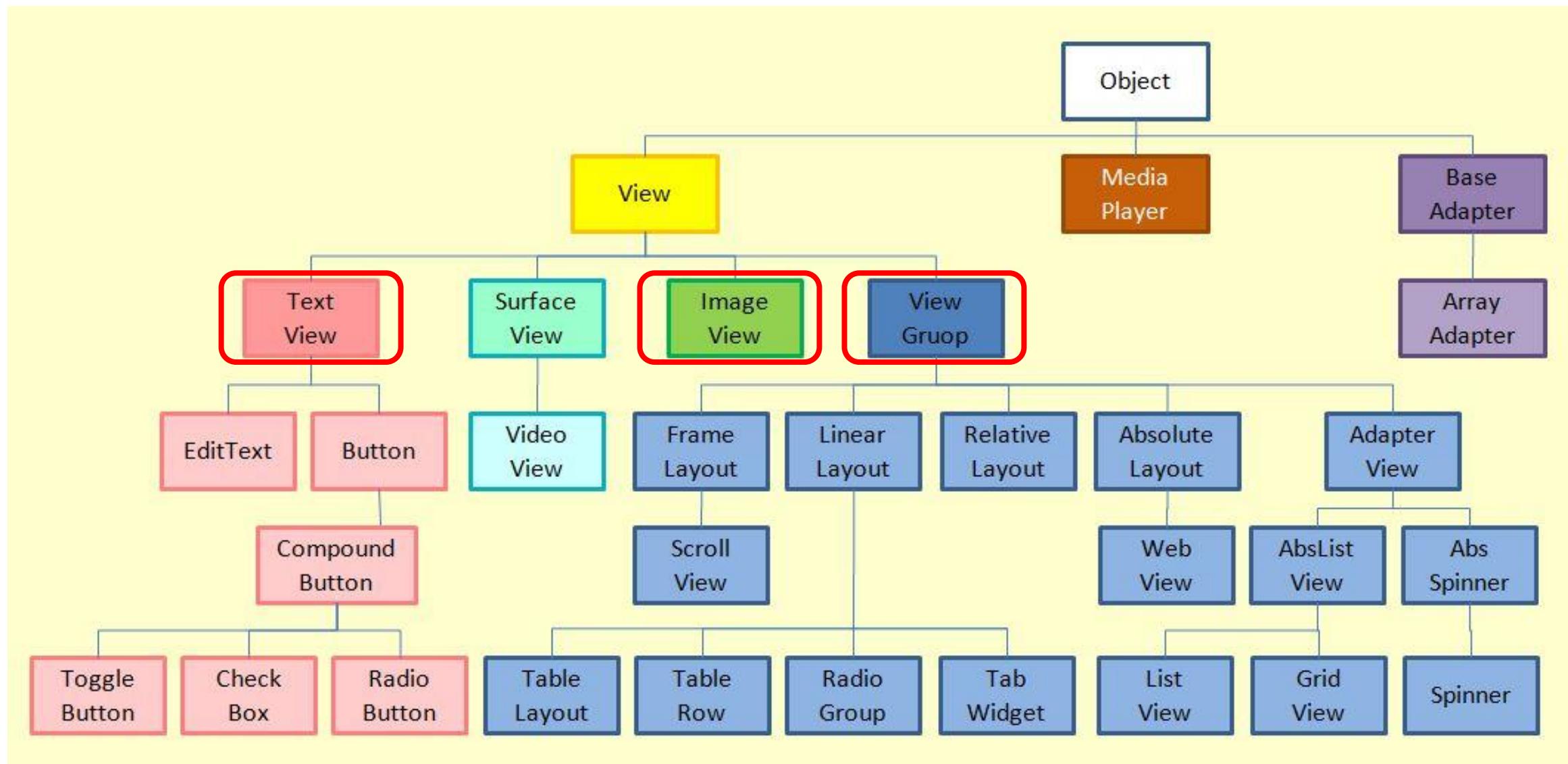
- res 폴더에 있는 **starry_night.png** 파일을 **drawable** 폴더에 저장

모듈	폴더	소스 파일	이미지
res	drawable	starry_night.png	 A reproduction of Vincent van Gogh's painting 'The Starry Night'. It depicts a dark, swirling night sky filled with yellow stars and a crescent moon, with a small town and rolling hills at the bottom.
res	drawable	still_life_with_kettle.png	 A reproduction of Vincent van Gogh's painting 'Still Life with a Coffee Pot'. It shows a still life arrangement on a table, including a green jar, a blue and white ceramic pitcher, a lemon, and a cloth.



안드로이드 자바 클래스 계층도

20



클래스와 속성/메소드 - View 클래스의 속성

- 화면 레이아웃 설정 파일인 activity_main.xml에서 여러 위젯 (TextView, ImageView, … 등의 컴포넌트) 속성 설정

클래스	속성	설명
View	android:clickable	클릭 이벤트에 대한 반응 여부를 정의함(true, false)
	android:id	뷰(or 다른 위젯)를 구별하기 위한 이름 android:id="@+id/아이디 이름" 으로 지정
	android:onClick	뷰가 클릭될 때 실행되는 메소드의 이름 (메소드는 MainActivity.java 파일에 코딩)
	android:tag	문자열로 나타내는 태그 – 명화 그림파일 이름

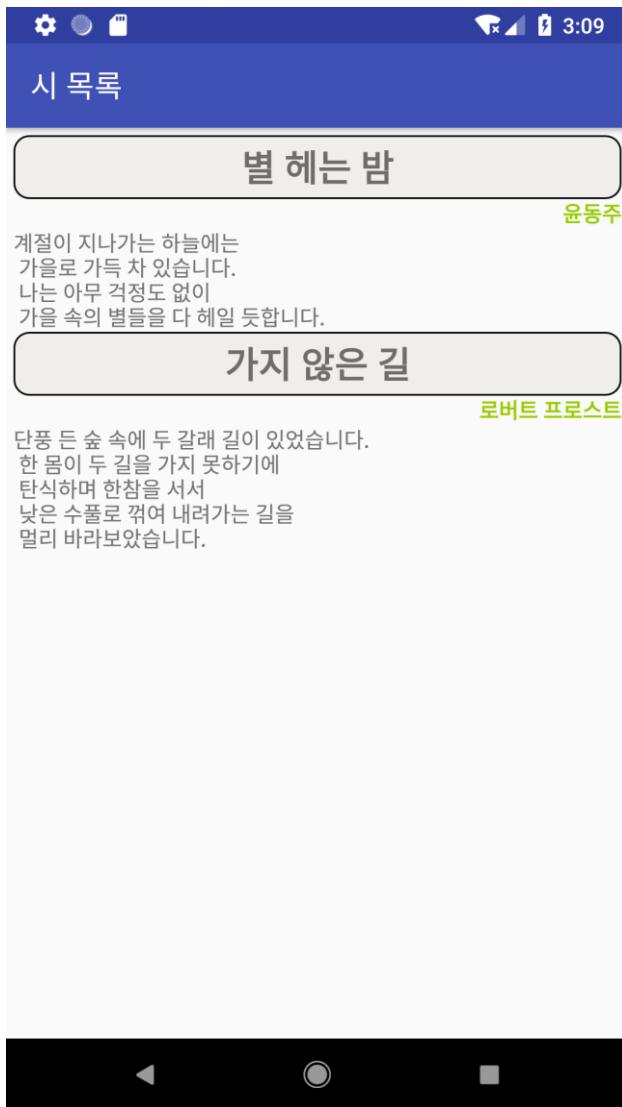
- TextView는 View 클래스를 상속(예약어 **extends**) 받았기 때문에 위의 속성을 그대로 다 가진다.

액티비티 전환



안드로이드 앱

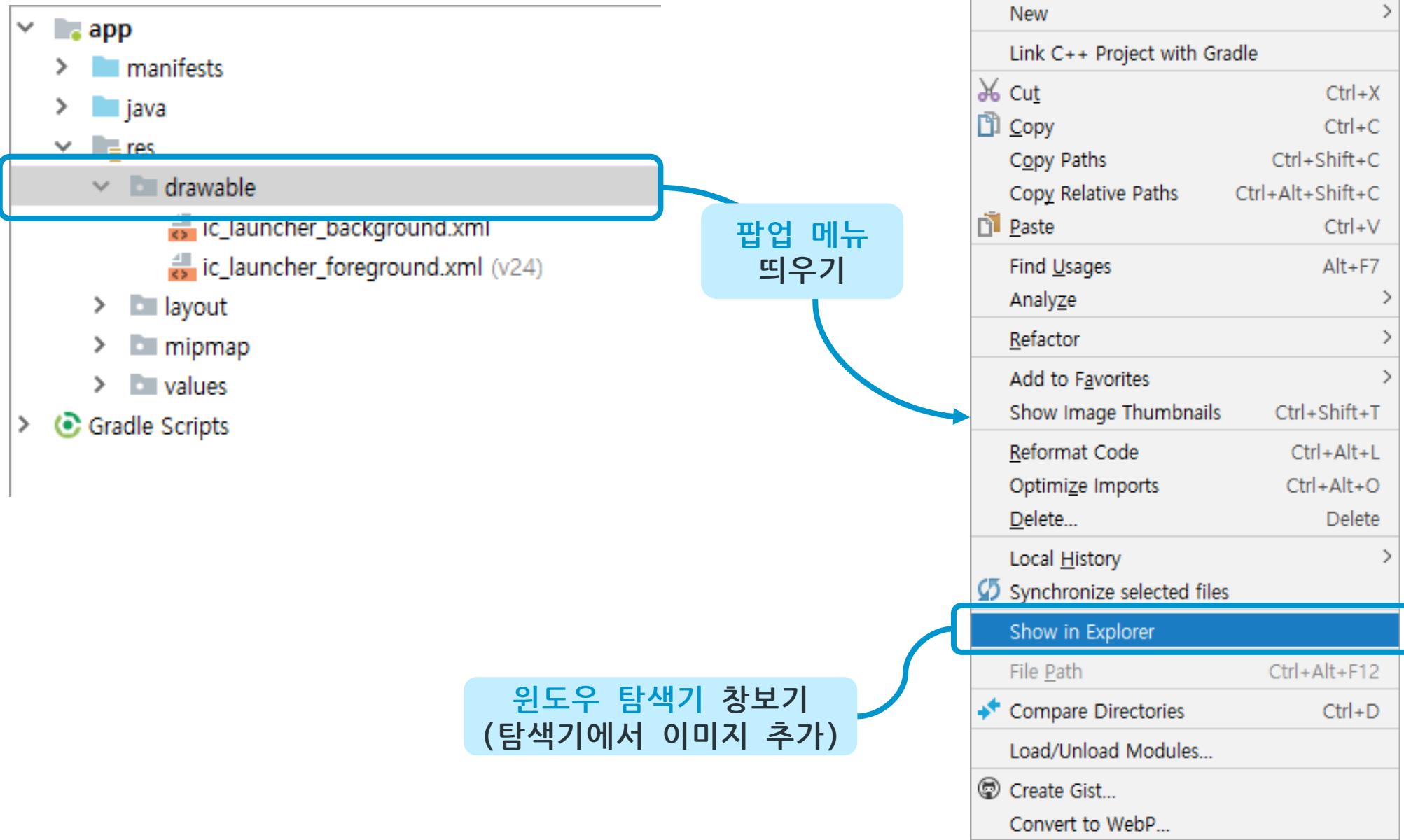
O utputs

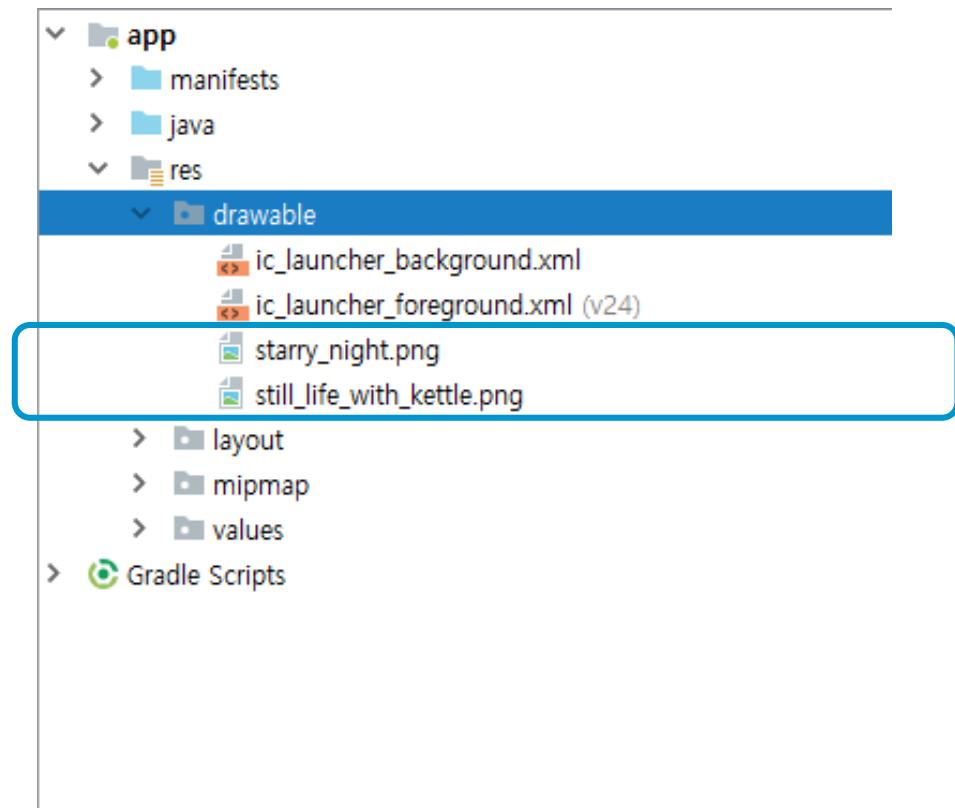
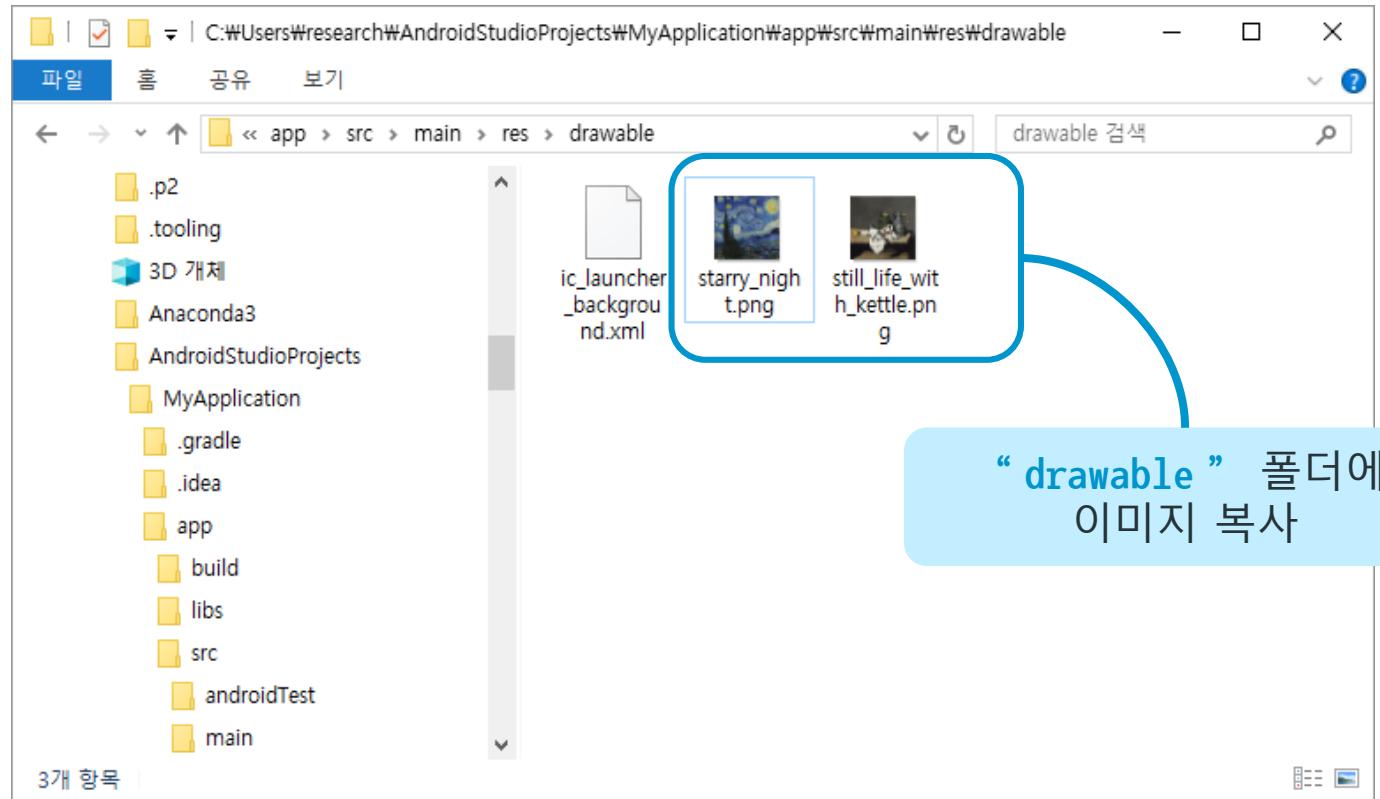


인테트 활용

- 두번째 액티비티(화면)을 띄워서 **시 제목**과 **명화**를 보여준다.
- intent를 사용해 **시제목**과 **명화 파일이름**을 넘겨 준다.

drawable 폴더에 이미지 추가하기



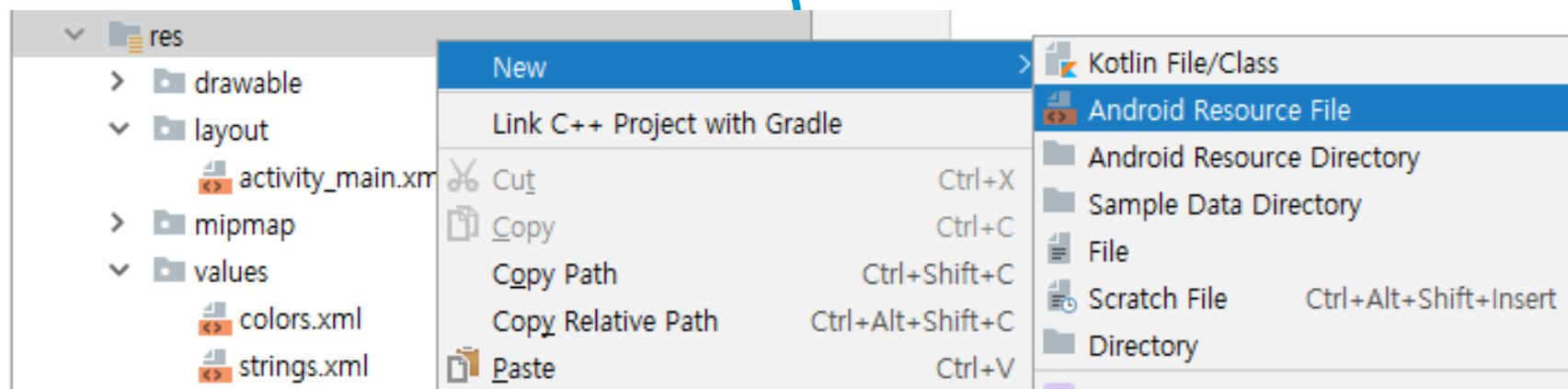


drawable 리소스 – 제목 배경그림

- **shape_title.xml** 생성(res/drawable 폴더)

- drawable resource를 이용한 그림 출력
- 제목(title)에 대한 출력 모양을 지정
- **drawable 폴더에는 화면에 그릴 수 있는 요소(도형)를 XML로 정의**
- **android:shape** 속성을 사용하여 다른 XML 리소스에 적용할 수 있는 그래픽에 대한 일반적인 개념

XML 파일 생성



- Set New Resource File

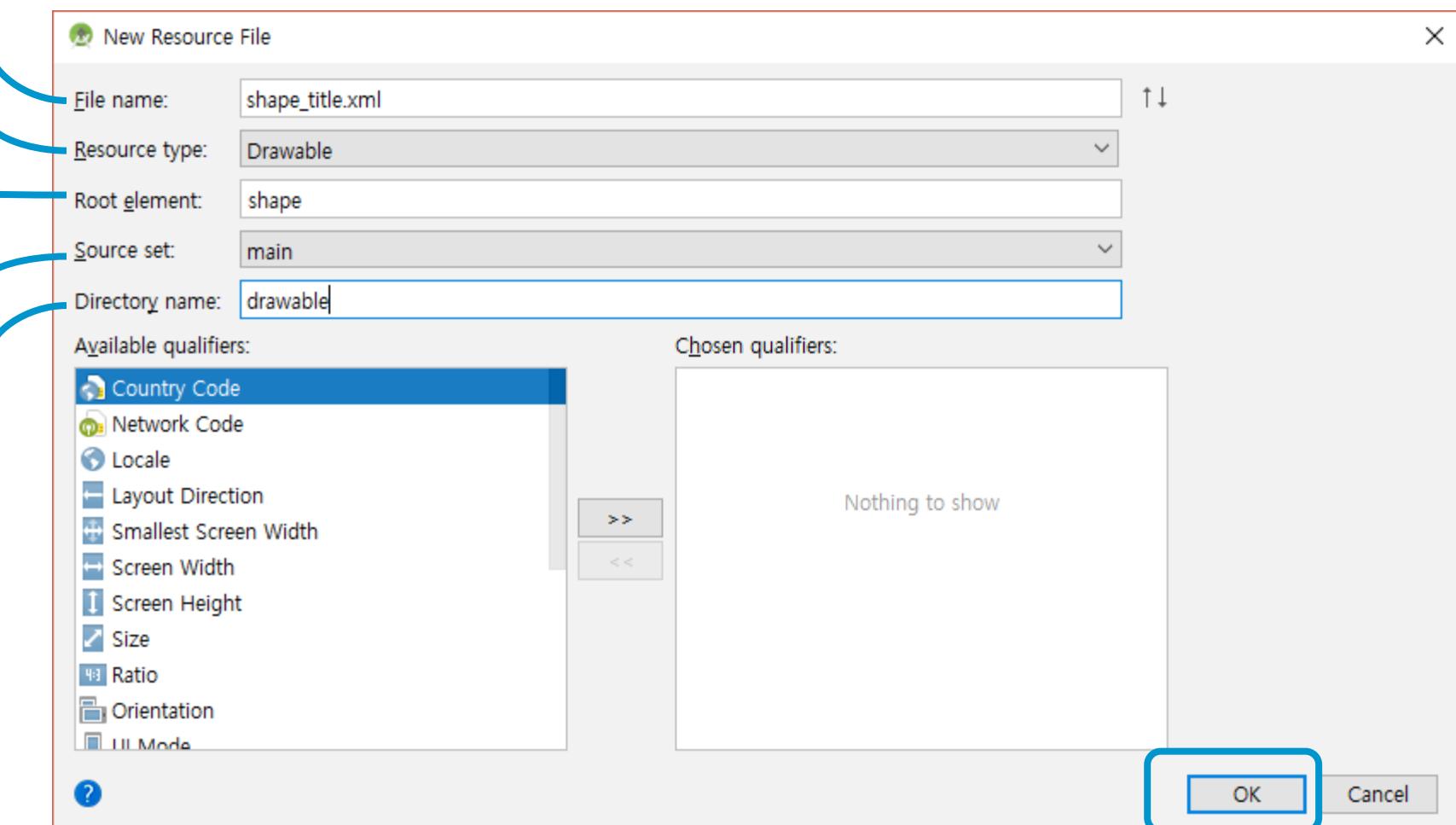
File name: shape_title.xml

Resource type: Drawable

Root element: shape

Source set: main

Directory name: drawable



- 생성된 shape_title.xml

shape_title.xml 파일

The screenshot shows the Android Studio interface. On the left, the project structure is displayed under the 'app' folder. A blue box highlights the 'res' directory, specifically the 'drawable' folder which contains 'shape_title.xml'. The 'shape_title.xml' file is also highlighted with a blue box in the code editor on the right. The code editor shows the XML content:

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

The tabs at the top of the code editor include 'activity_main.xml', 'strings.xml', 'shape_title.xml', and 'MainActivity.java'.

• shape_title.xml 소스

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="rectangle">
    <solid android:color="#1061380B" />
    <stroke
        android:width="1dp"
        android:color="#000000" />
    <padding
        android:left="10dp"
        android:top="5dp"
        android:right="10dp"
        android:bottom="5dp" />
    <corners
        android:radius="10dp" />
</shape>
```

사각형의 배경색 지정

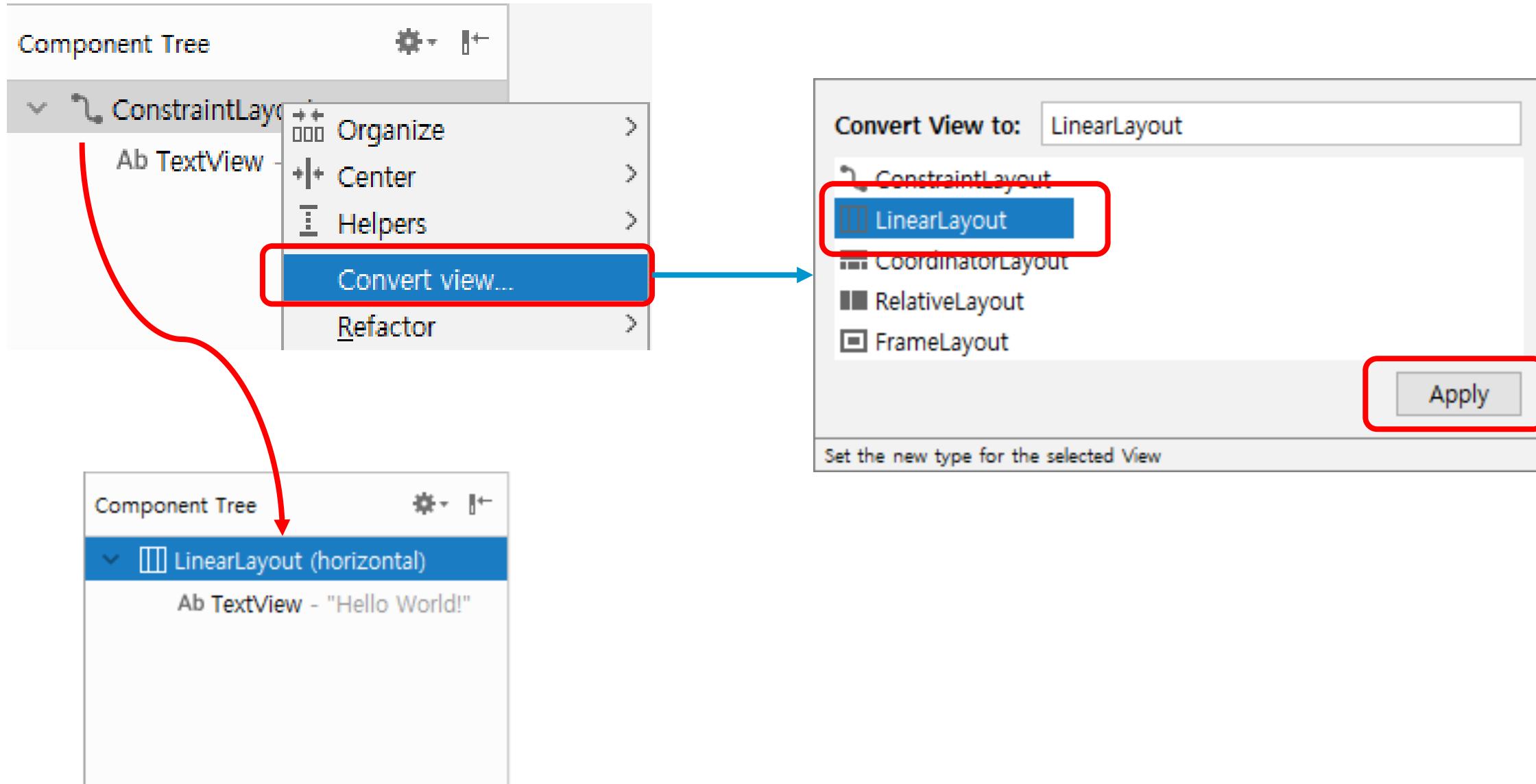
사각형 테두리 색

내부 패딩 정보

출력모양 모서리를 둥근 모양으로 지정(반지름은 5dp)

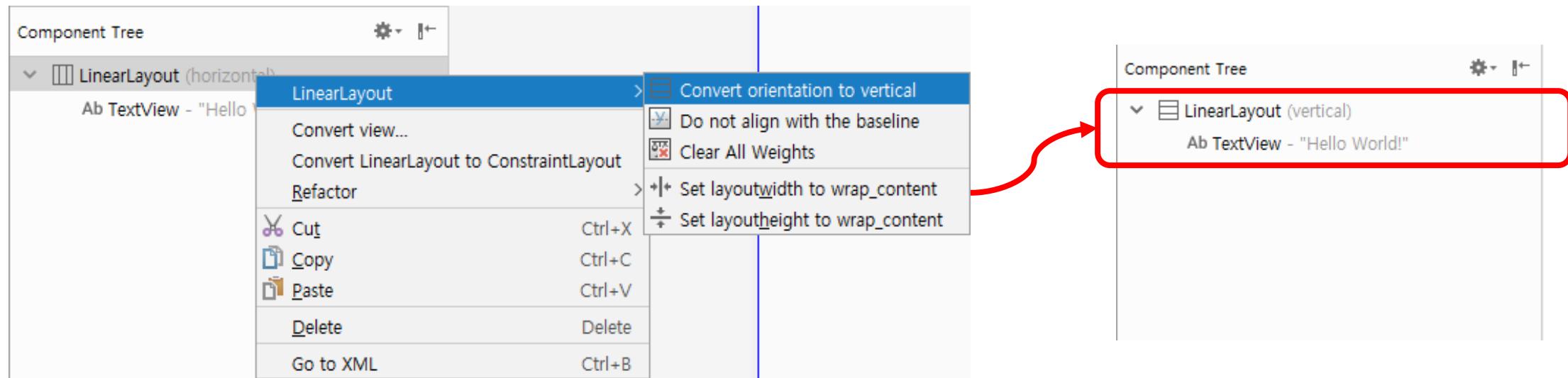
ConstraintLayout을 LinearLayout로 바꾸기

39



LinearLayout의 방향을 Horizontal → Vertical로 변경하기

40



```
activity_main.xml MainActivity.java
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical"
8     tools:context=".MainActivity">
9
10    <TextView
11        android:layout_width="wrap_content"
12        android:layout_height="wrap_content"
13        android:text="Hello World!" />
14
15 </LinearLayout>
```

Strings.xml 파일

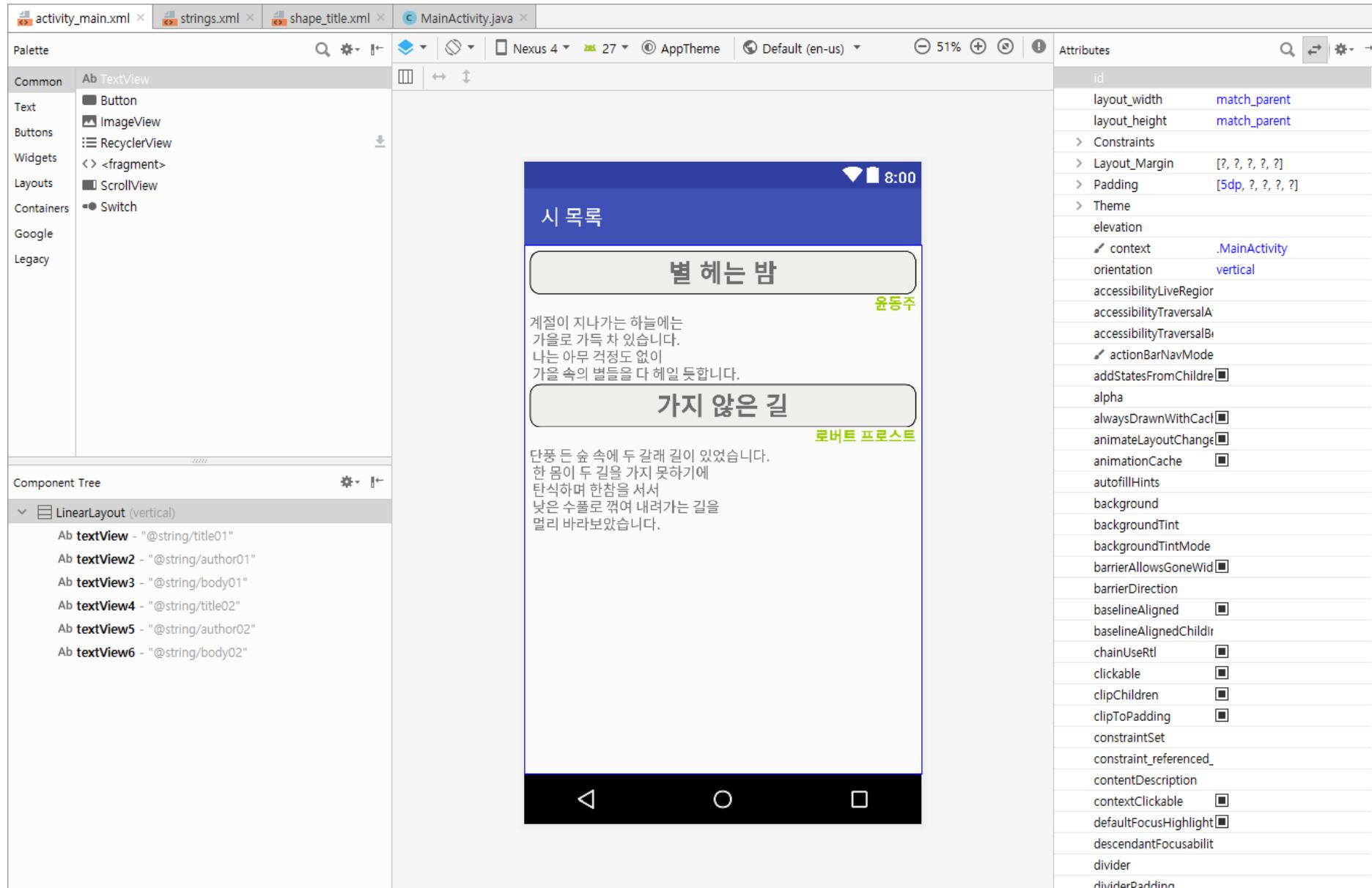
42

Edit translations for all locales in the translations editor.

```
1 <resources>
2     <string name="app_name">시 목록</string> → 첫번째 액티비티의 제목
3     <string name="app_name2">시 내용 그림</string> → 두번째 액티비티의 제목
4
5     <string name="title01">별 헤는 밤</string>
6     <string name="author01">윤동주</string>
7     <string name="body01">
8         계절이 지나가는 하늘에는\n
9         가을로 가득 차 있습니다.\n
10        나는 아무 걱정도 없이\n
11        가을 속의 별들을 다 헤일 듯합니다.
12    </string>
13
14    <string name="title02">가지 않은 길</string>
15    <string name="author02">로버트 프로스트</string>
16    <string name="body02">
17        단풍 든 숲 속에 두 갈래 길이 있었습니다.\n
18        한 몸이 두 길을 가지 못하기에\n
19        탄식하며 한참을 서서\n
20        낮은 수풀로 꺾여 내려가는 길을\n
21        멀리 바라보았습니다.
22    </string>
23 </resources>
24
25
```

TextView에 속성을 적용한 결과 화면

43



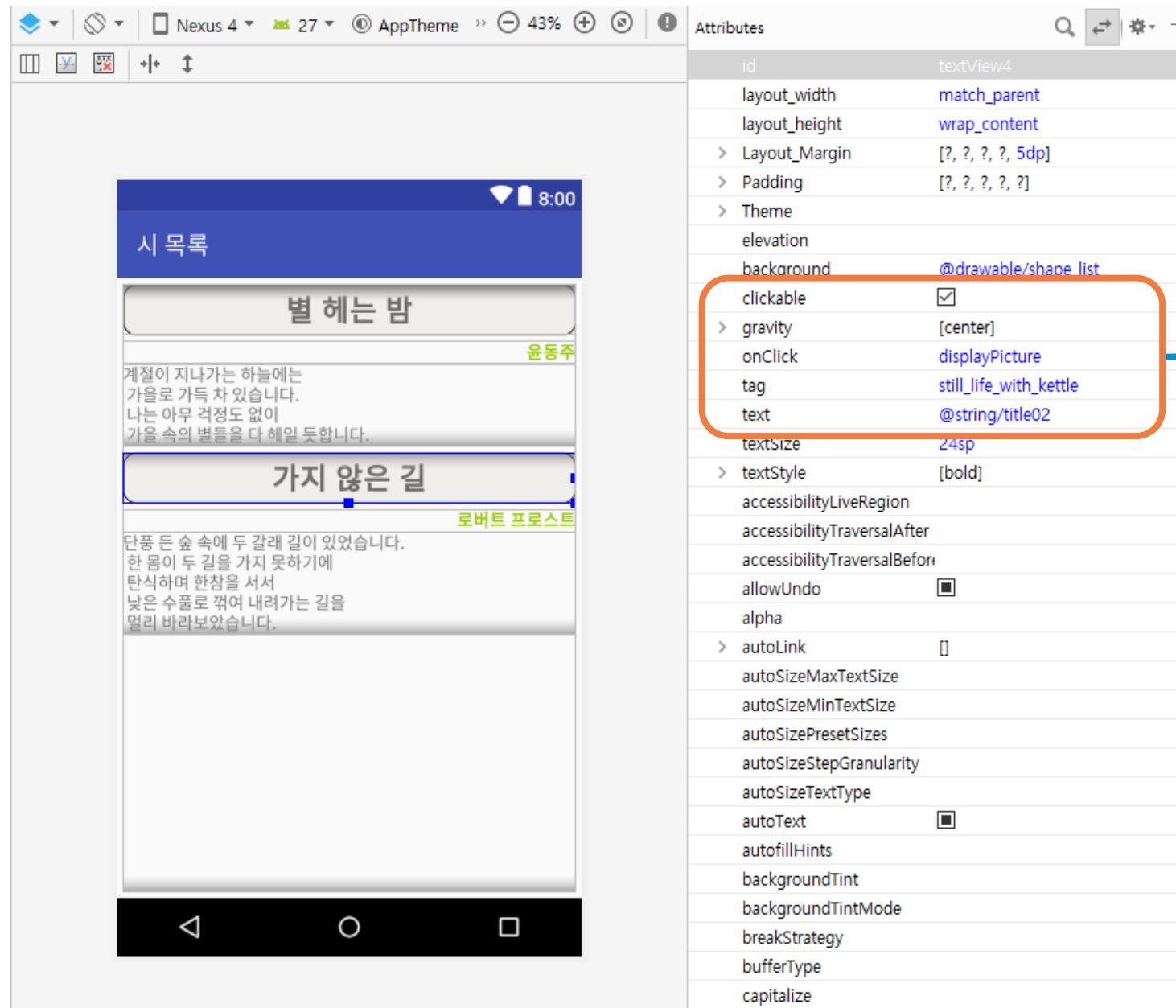
Step 2.2 파일 편집-기존 예제에서 속성 수정

44

The screenshot shows the Android Studio XML editor interface. On the left is the preview of the Android application screen, which contains a title bar with "시 목록" and "8:00", a main content area with two buttons ("별 헤는 밤" and "가지 않은 길"), and some descriptive text. On the right is the XML code editor with the "Attributes" tab selected. A specific section of the code is highlighted with an orange rectangle, and a blue arrow points from this highlighted area to a callout box containing four key attributes and their values.

Attributes	Value
id	textView
layout_width	match_parent
layout_height	wrap_content
Layout_Margin	[?, ?, ?, ?, 5dp]
Padding	[?, ?, ?, ?, ?]
Theme	
elevation	
background	@drawable/shape_list
clickable	<input checked="" type="checkbox"/>
gravity	[center]
onClick	displayPicture
tag	starry_night
text	@string/title01
textSize	24sp
textStyle	[bold]
accessibilityLiveRegion	
accessibilityTraversalAfter	
accessibilityTraversalBefore	
allowUndo	<input type="checkbox"/>
alpha	
autoLink	[]
autoSizeMaxTextSize	
autoSizeMinTextSize	
autoSizePresetSizes	
autoSizeStepGranularity	
autoSizeTextType	
autoText	<input type="checkbox"/>
autofillHints	
backgroundTint	
backgroundTintMode	
breakStrategy	
bufferType	
capitalize	

clickable: true
onClick: displayPicture
tag: starry_night
text: @string/title01



clickable: true
onClick: displayPicture
tag: still_life_with_kettle
text: @string/title02

Step 2.2 파일 편집- 자바 메소드 만들기

46

MainActivity.java

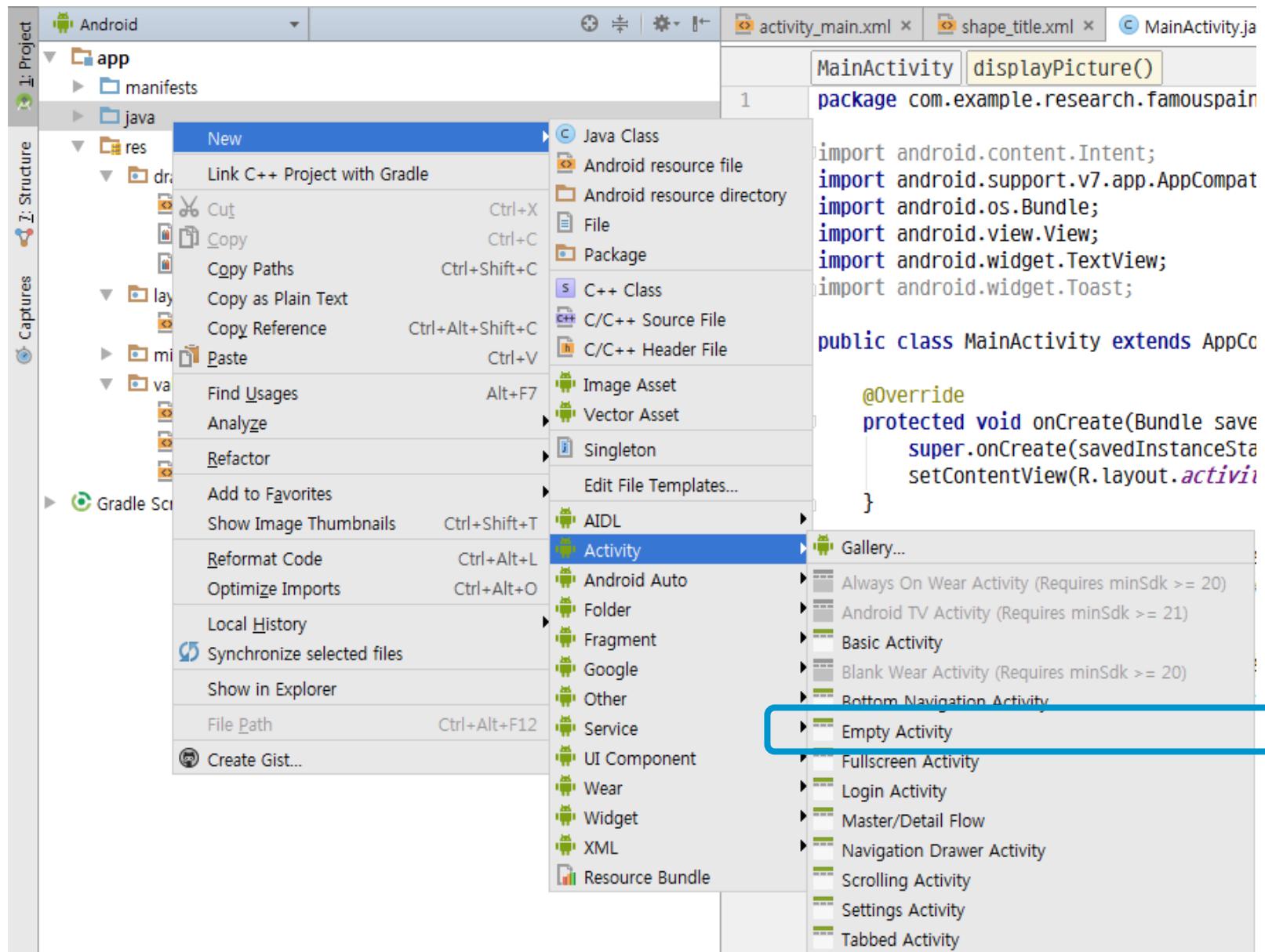
```
activity_main.xml activity_detail_view.xml AndroidManifest.xml DetailViewActivity.java MainActivity.java S

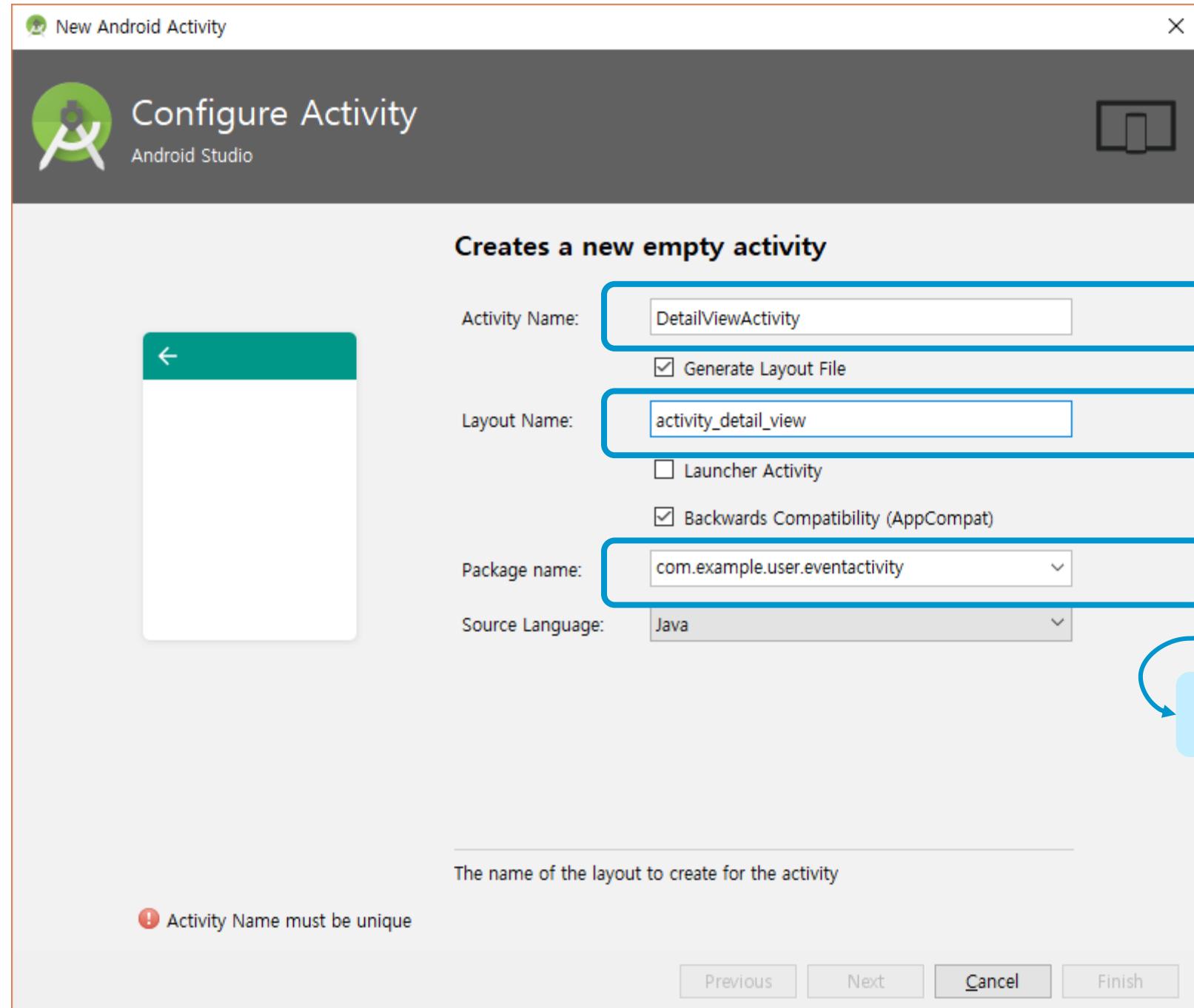
1 package com.example.user.eventactivity;
2
3 import android.content.Intent;
4 import android.support.v7.app.AppCompatActivity;
5 import android.os.Bundle;
6 import android.view.View;
7 import android.widget.TextView;
8
9 public class MainActivity extends AppCompatActivity {
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_main);
15     }
16
17     public void displayPicture(View view){
18         int vID = view.getId(); // 클릭한 객체의 ID를 얻는다.
19
20         // 클릭한 위젯요소(TextView)의 Text, Tag 속성값을 가져온다.
21         TextView textView = (TextView) findViewById(vID); // id가 vID인 위젯요소(TextView)를 가져온다.
22         String title = (String) textView.getText(); // 위젯요소(TextView)의 text 내용을 가져온다(시 제목).
23         String tag = (String) textView.getTag(); // 위젯요소(TextView)의 tag 내용을 가져온다(그림 파일 이름).
24
25         // Intent에 저장하여 보내기
26         // DetailViewActivity로 보낼 Intent 생성
27         Intent it = new Intent( packageContext: this, DetailViewActivity.class );
28         it.putExtra( name: "it_title", title ); // Intent에 "it_title"이라는 이름으로 title 저장
29         it.putExtra( name: "it_tag", tag ); // Intent에 "it_tag"라는 이름으로 tag 저장
30         startActivity(it); // DetailViewActivity 화면 시작
31     }
32 }
```

displayPicture 메소드 추가

Step 2.2 파일 편집 – 두 번째 Activity 추가

47





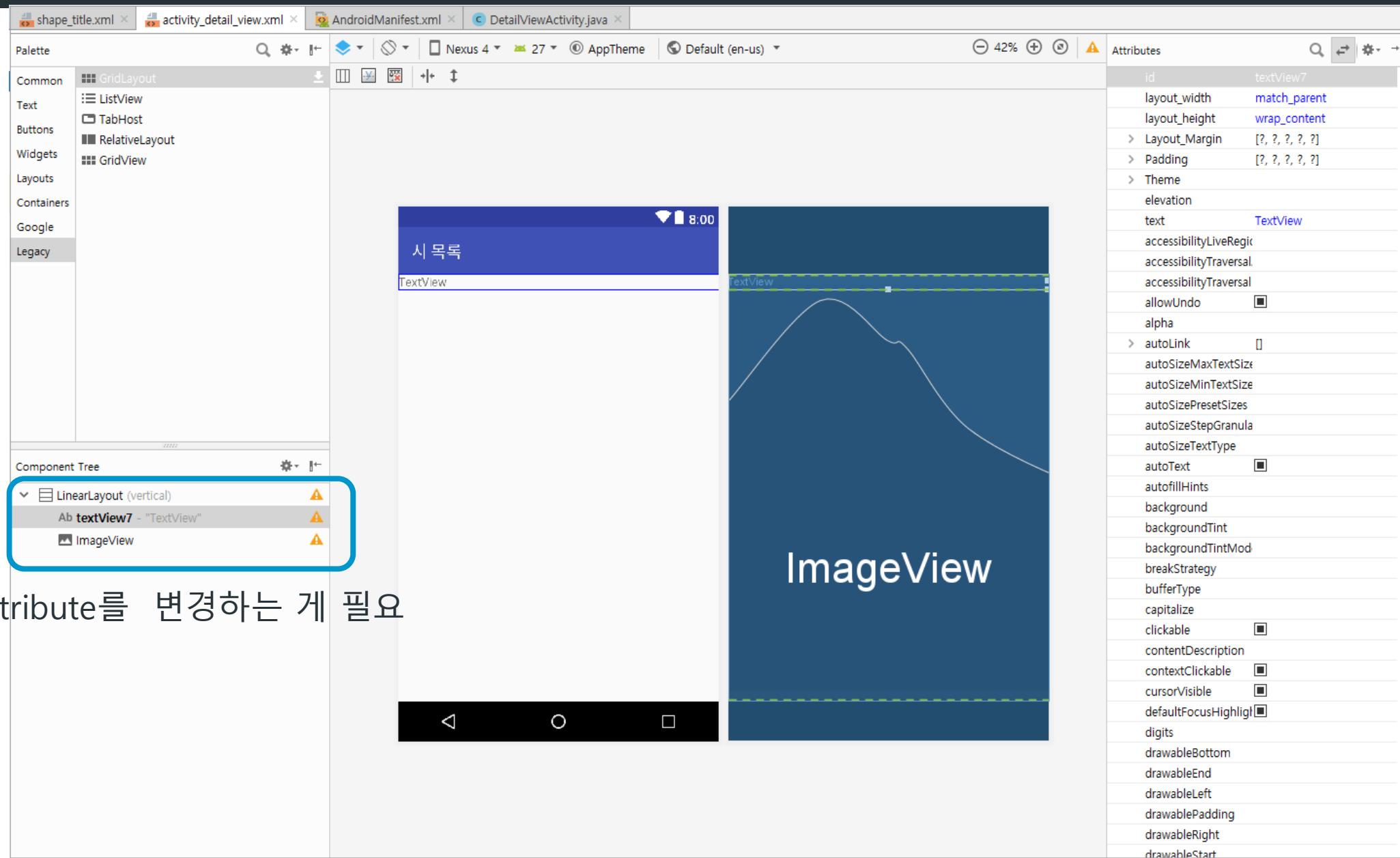
Activity Name:
DetailViewActivity

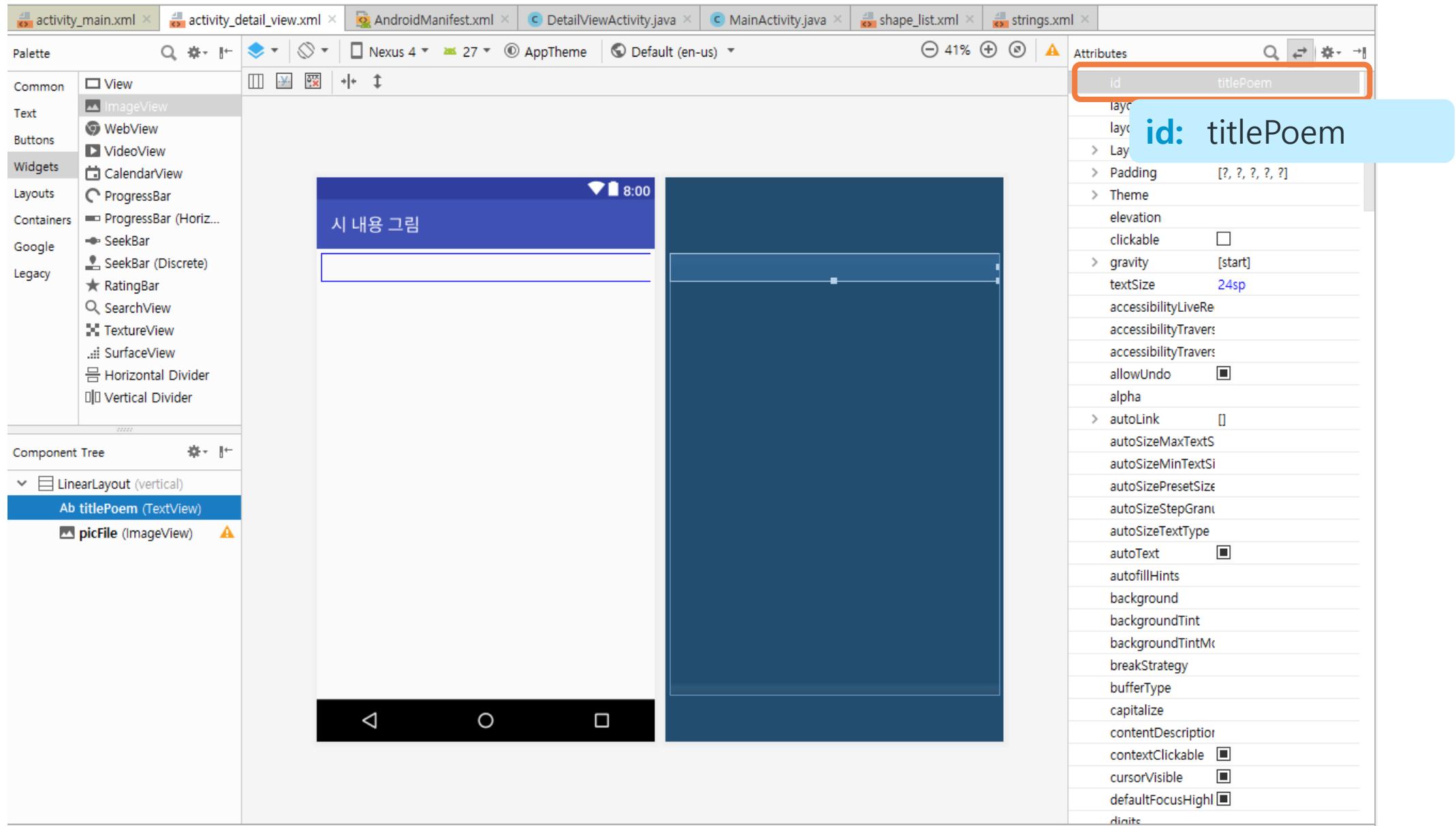
Layout Name:
activity_detail_view

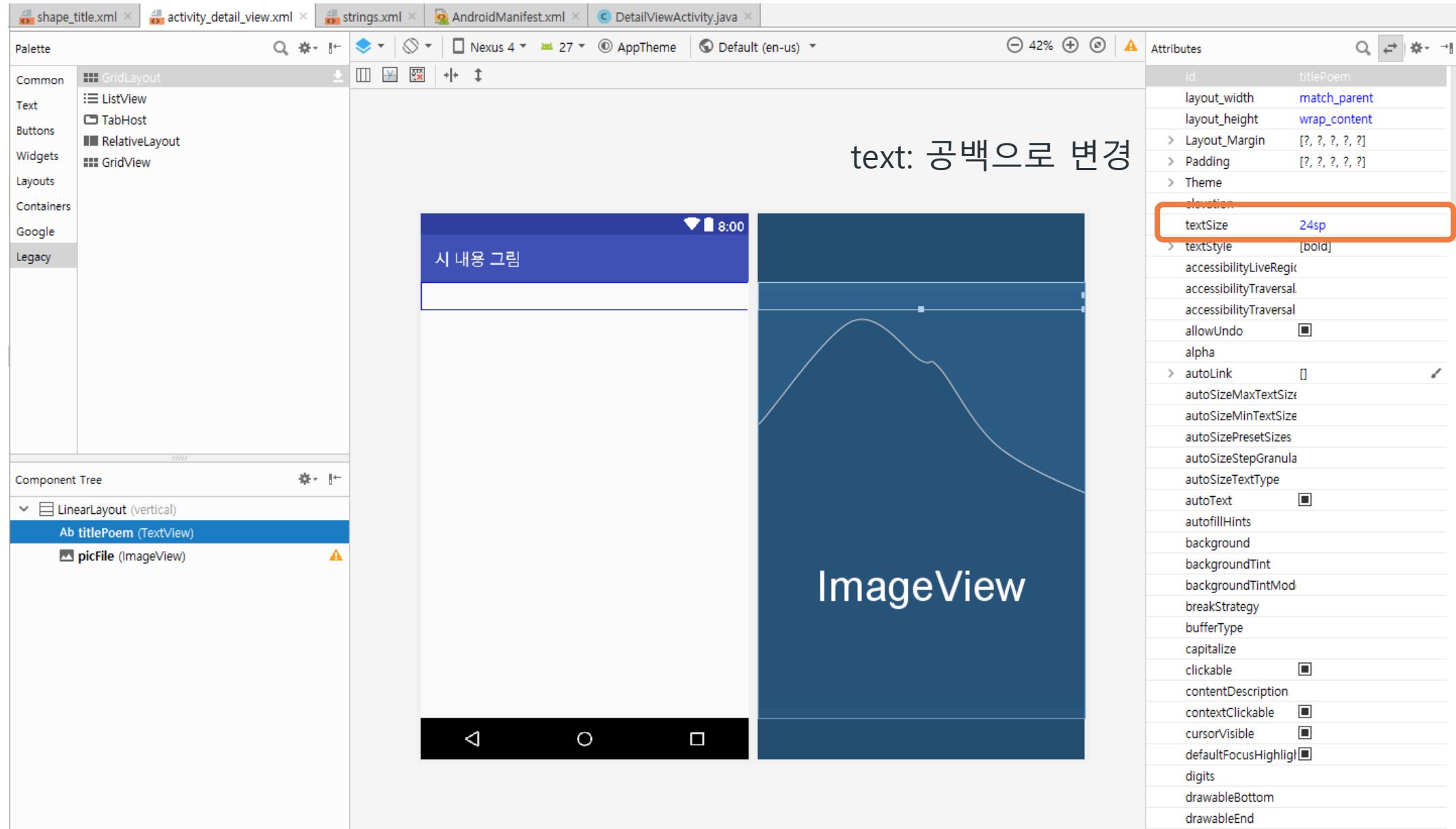
Package name:
com.example.user.eventactivity

Step 2.2 파일 편집-activity_detail_view.xml

49

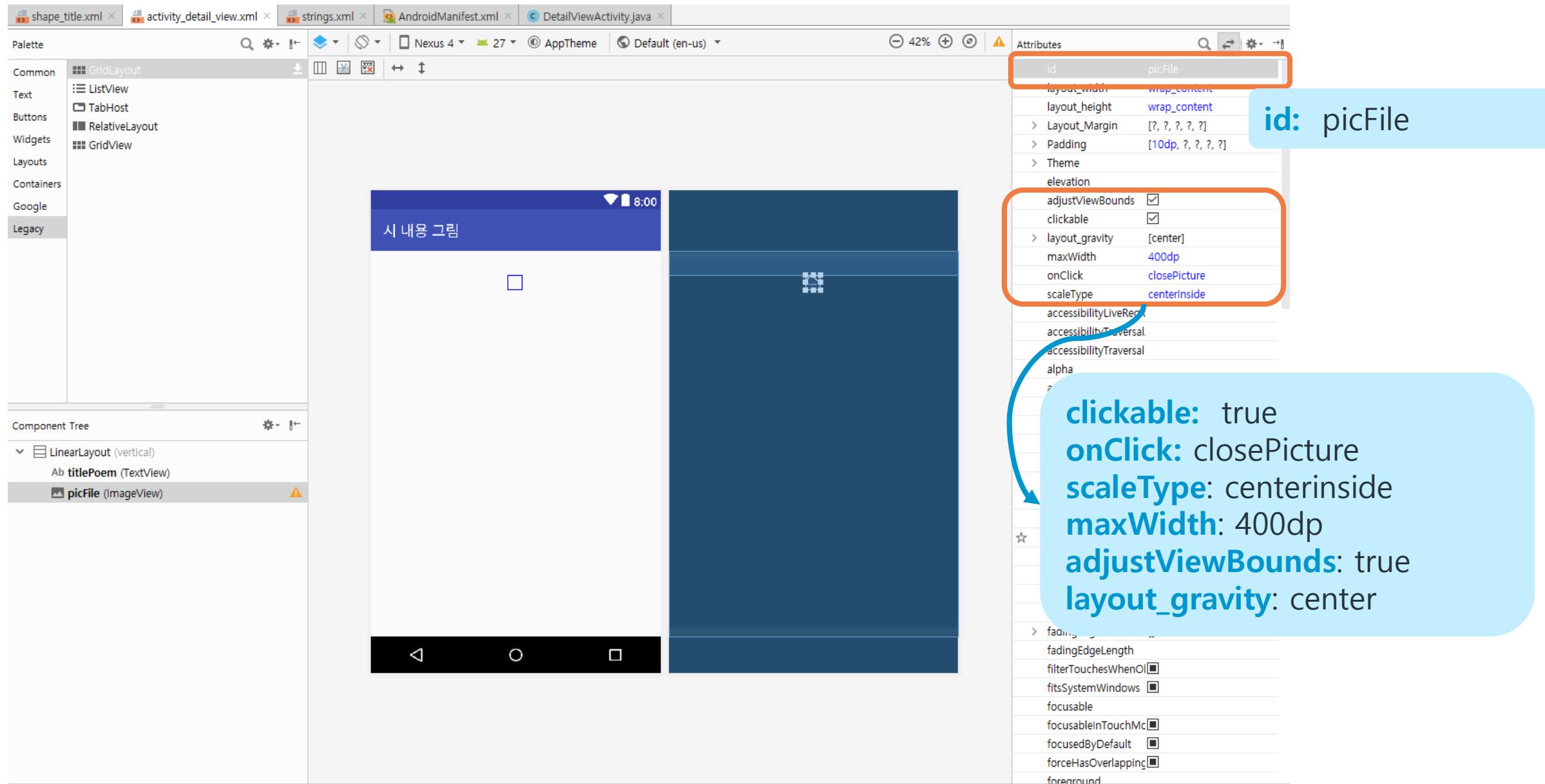






activity_detail_view.xml의 ImageView 속성 설정

54



Step 2.2 파일 편집-AndroidManifest.xml 파일 수정

57

- 두번째 액티비티의 제목 추가(수정)

The screenshot shows the AndroidManifest.xml file in the Android Studio editor. The code is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.research.famouspaintings">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="시 목록"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".DetailViewActivity"></activity>
    </application>
</manifest>
```

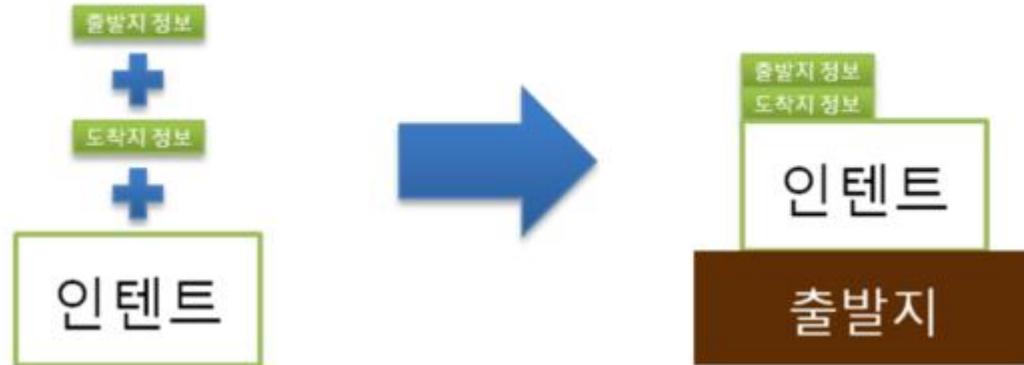
The screenshot shows the AndroidManifest.xml file after modification. The code is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.user.eventactivity">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="시 목록"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".DetailViewActivity"
            android:label="@string/app_name2"></activity>
    </application>
</manifest>
```

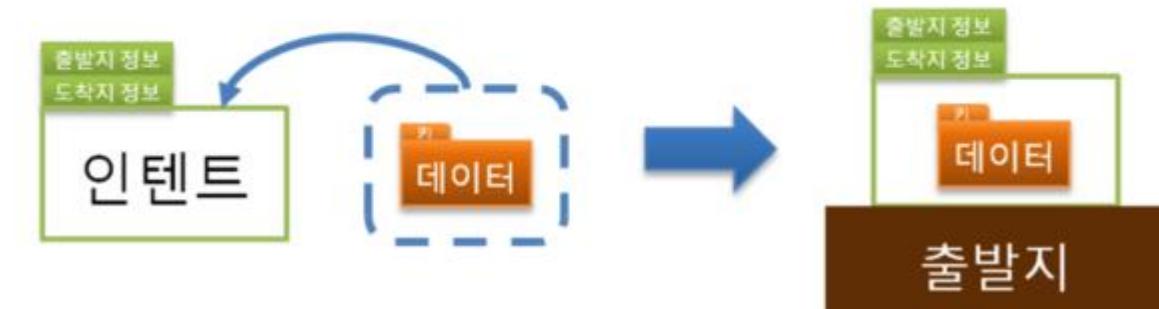
① 출발지 액티비티에서 인텐트를 만든다.

```
Intent intent = new Intent(this, SubActivity.class);
```



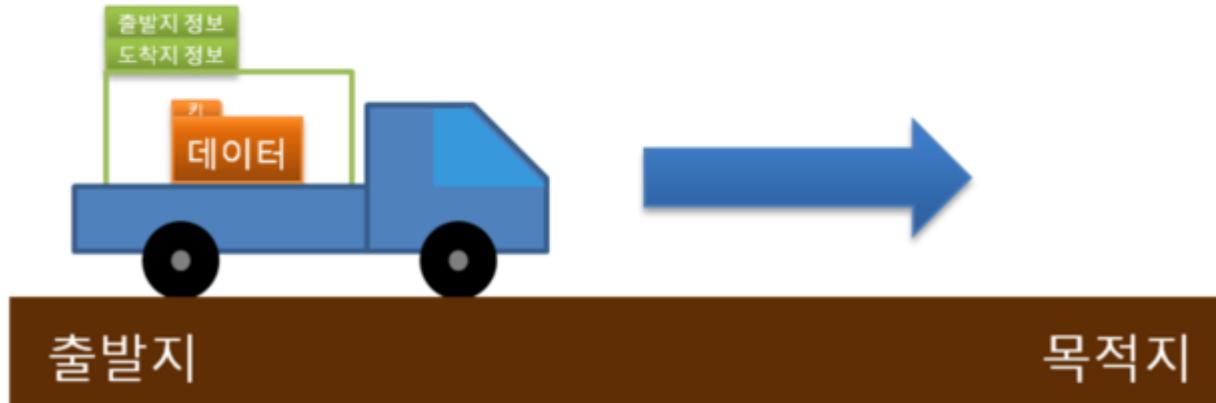
② 인텐트에 데이터를 적재한다.

```
intent.putExtra("value", "문자열");
```



③ 인텐트를 목적지 액티비티로 발송한다.

```
startActivity(intent);
```



④ 인텐트를 트럭에서(?) 내린다.

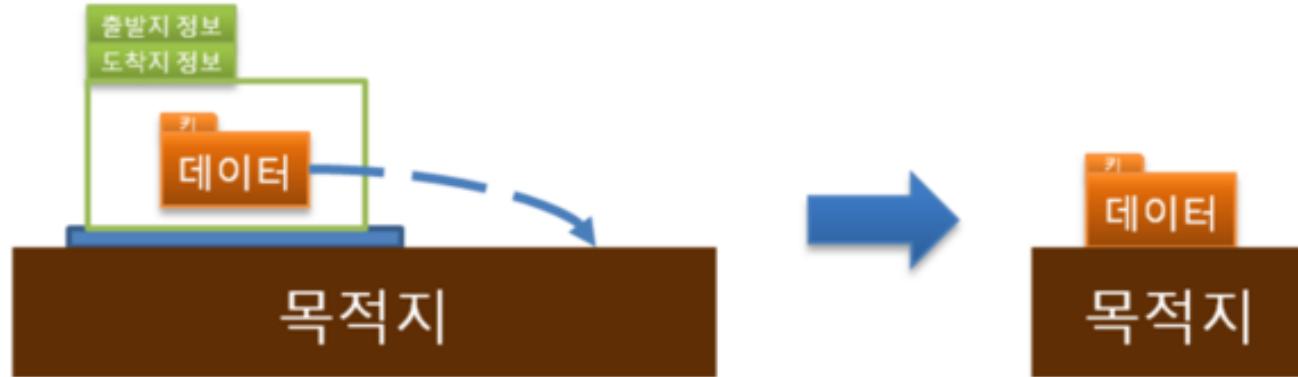
```
Intent intent = getIntent();
```



⑤ 인텐트에서 데이터를 꺼낸다.

61

```
String data = intent.getStringExtra("value");
```



Step 2.2 파일 편집-DetailViewActivity.java

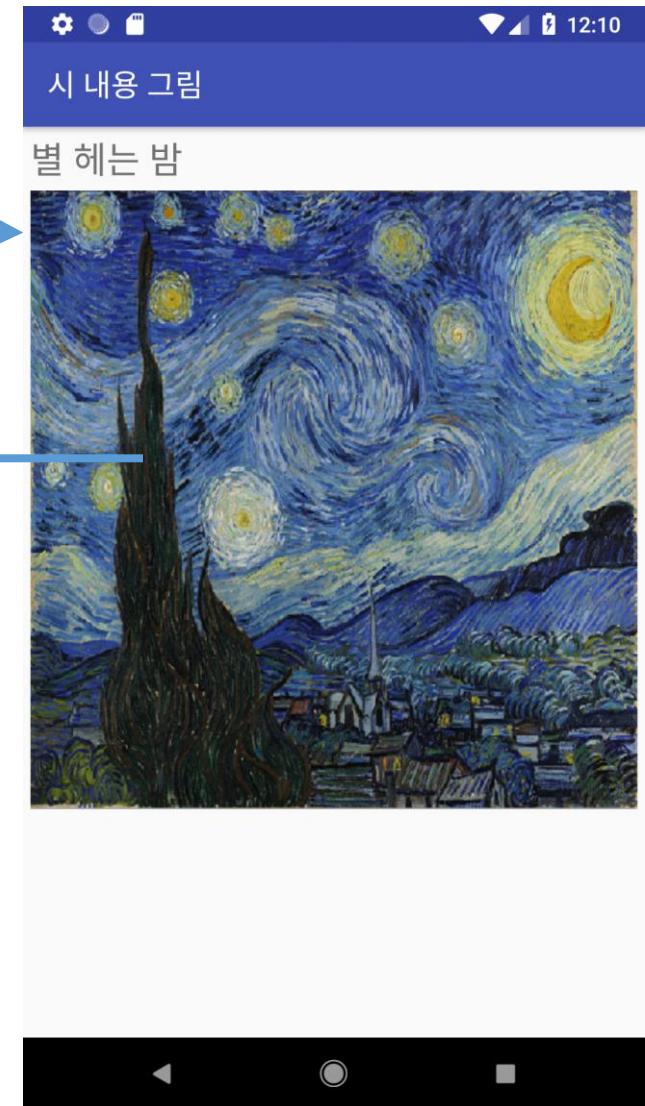
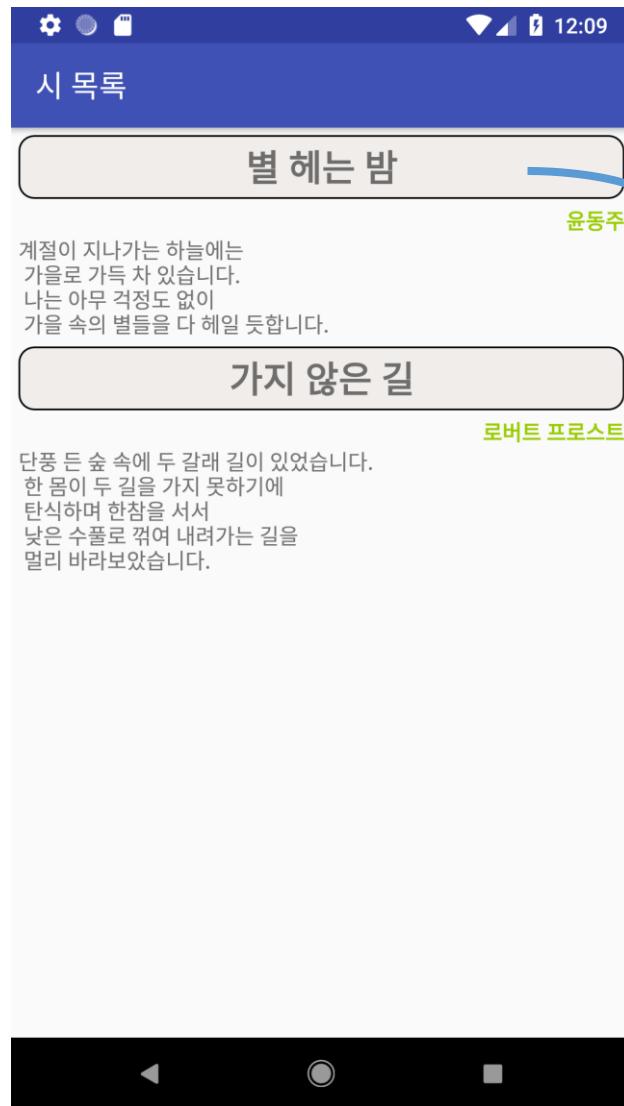
64

```
activity_main.xml x activity_detail_view.xml x AndroidManifest.xml x DetailViewActivity.java x MainActivity.java x sha
1 package com.example.user.eventactivity;
2
3 import ...
4
5 public class DetailViewActivity extends AppCompatActivity {
6
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.activity_detail_view);
11
12        Intent it = getIntent(); // 현재 Activity로 전송된 Intent를 얻는다.
13        String title = it.getStringExtra( name: "it_title"); // "it_title"이름으로 저장된 내용을 가져온다.
14        String tag = it.getStringExtra( name: "it_tag"); // "it_gat"이름으로 저장된 내용을 가져온다.
15
16        // TextView에 내용 입력하기
17        TextView tvTitle = (TextView) findViewById(R.id.titlePoem); // TextView 객체 확인
18        tvTitle.setText(title);
19        // ImageView에 그림 표시하기
20        ImageView ivPicture = (ImageView) findViewById(R.id.picFile); // ImageView 객체 확인
21        Resources res = getResources();
22        int id_img = res.getIdentifier(tag, defType: "drawable", getPackageName());
23        Drawable drawable = res.getDrawable(id_img, theme: null);
24        ivPicture.setBackground(drawable);
25    }
26
27    public void closePicture(View view){
28        finish();
29    }
30}
```

Intent로부터 받아온 내용을
TextView에 표시하고, 그림
을 표시한다.

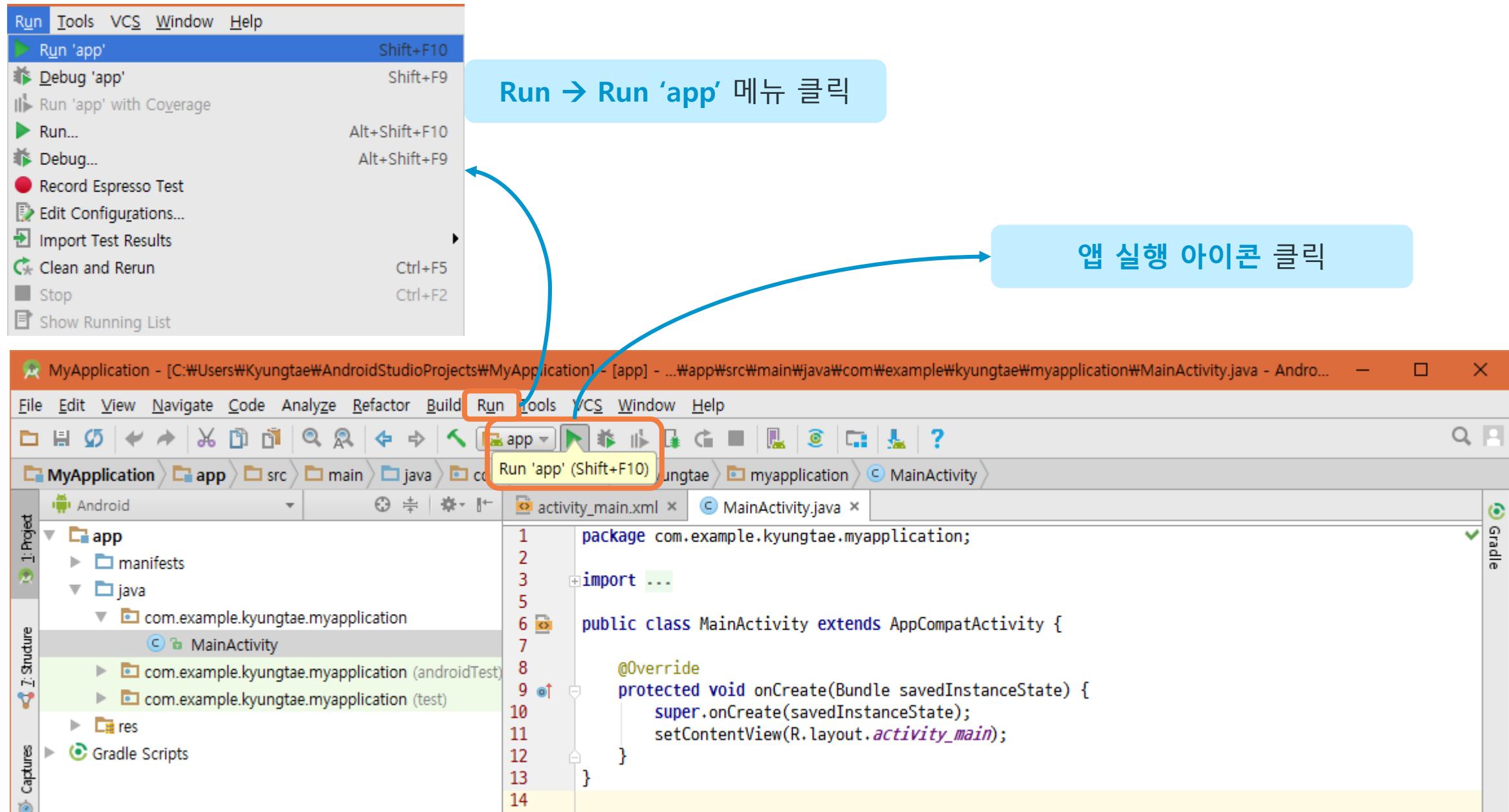
액티비티에 보여진 그림을 클
릭하면 실행됨(화면 닫기).

O utputs



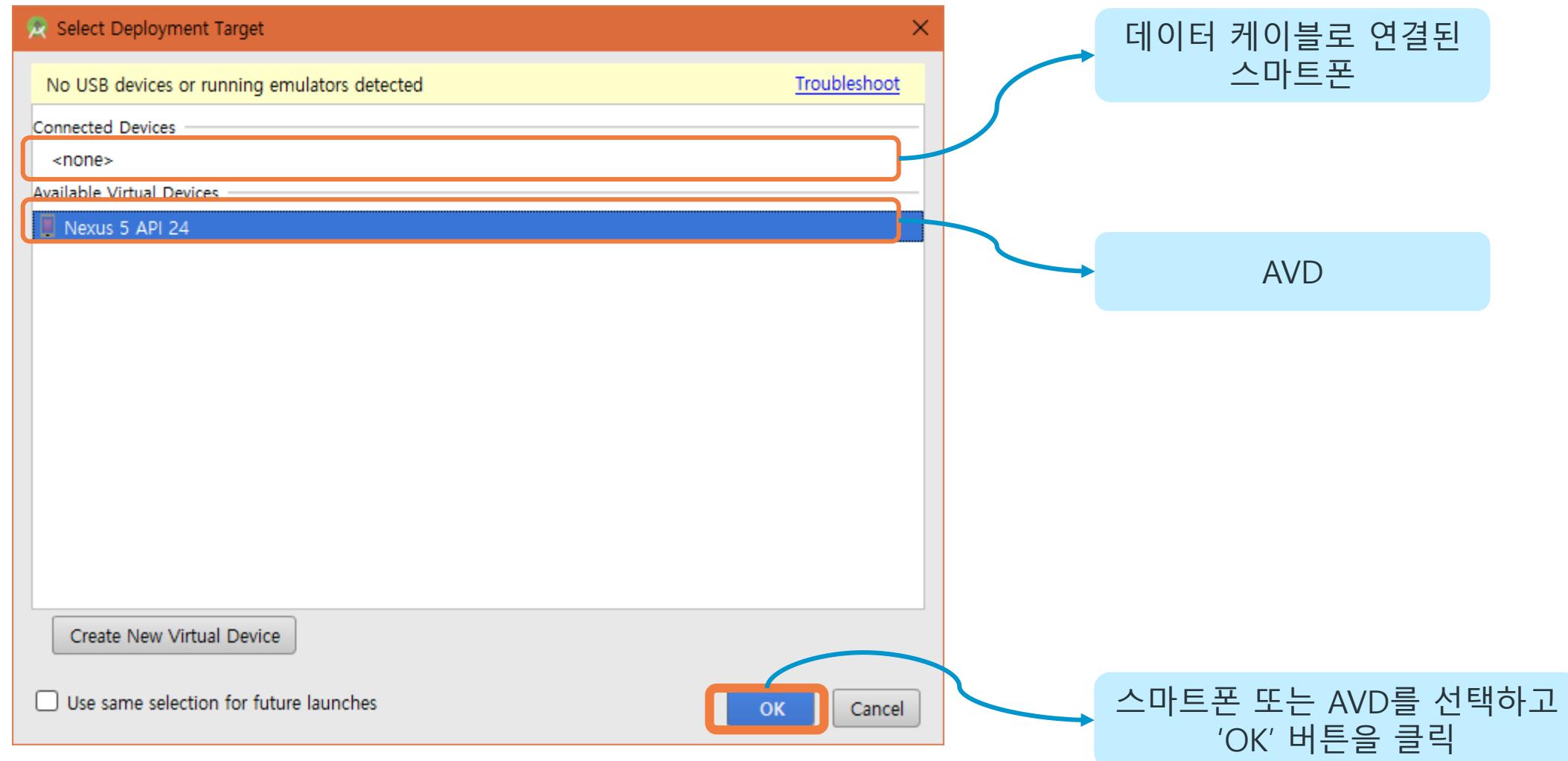
Step 3. 프로젝트 실행

67



• AVD 장비 선택하기

68





question

&



answer

79

