

**Week05.**

# 이벤트 처리와 액티비티 간 이동



# 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	<b>이벤트 처리와 액티비티 간 이동</b>
6	오디오 재생
7	비디오 재생
8	<b>중간고사</b>
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	<b>기말 고사</b>



The screenshot shows a web browser window with multiple tabs. The active tab is 'Lecture2018/AndroidApp at master · hopypark · GitHub'. The address bar shows the URL 'https://github.com/hopypark/Lecture2018/tree/master/AndroidApp'. The repository page for 'hopypark / Lecture2018' is displayed, showing the 'Code' tab selected. The file tree for the 'master' branch is visible, listing files and folders. A red rectangle highlights the file 'Week05.Chap08.이벤트 처리와 액티비티 간 이동.pdf'.

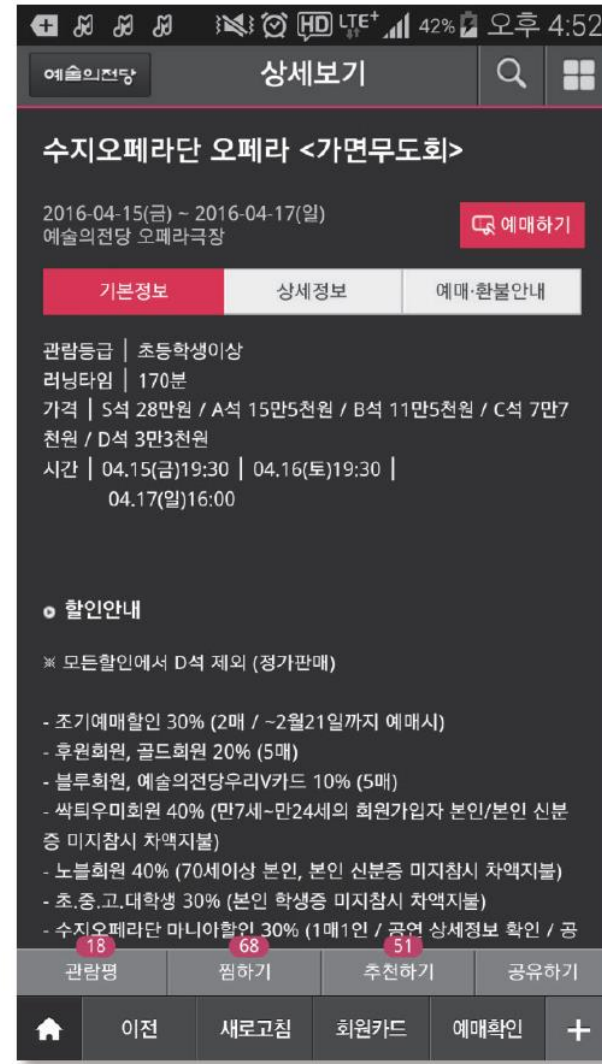
File/Folder	Action	Time
resources	Add files via upload	22 hours ago
sources	Update readme.md	6 minutes ago
Week01.강의 소개.pdf	Add files via upload	28 days ago
Week02.Chap02.앱 개발환경 구축.pdf	Add files via upload	21 days ago
Week02.Chap03.앱 프로젝트 구조와 실행원리.pdf	Add files via upload	14 days ago
Week03.Chap06.텍스트 출력과 레이아웃.pdf	Add files via upload	14 days ago
Week04.Chap07.이미지의 출력.pdf	Add files via upload	8 days ago
Week05.Chap08.이벤트 처리와 액티비티 간 이동.pdf	Add files via upload	13 hours ago
readme.md	Create readme.md	28 days ago

# 사용자 클릭 처리 앱의 예

4



액티비티 1(초기 화면)

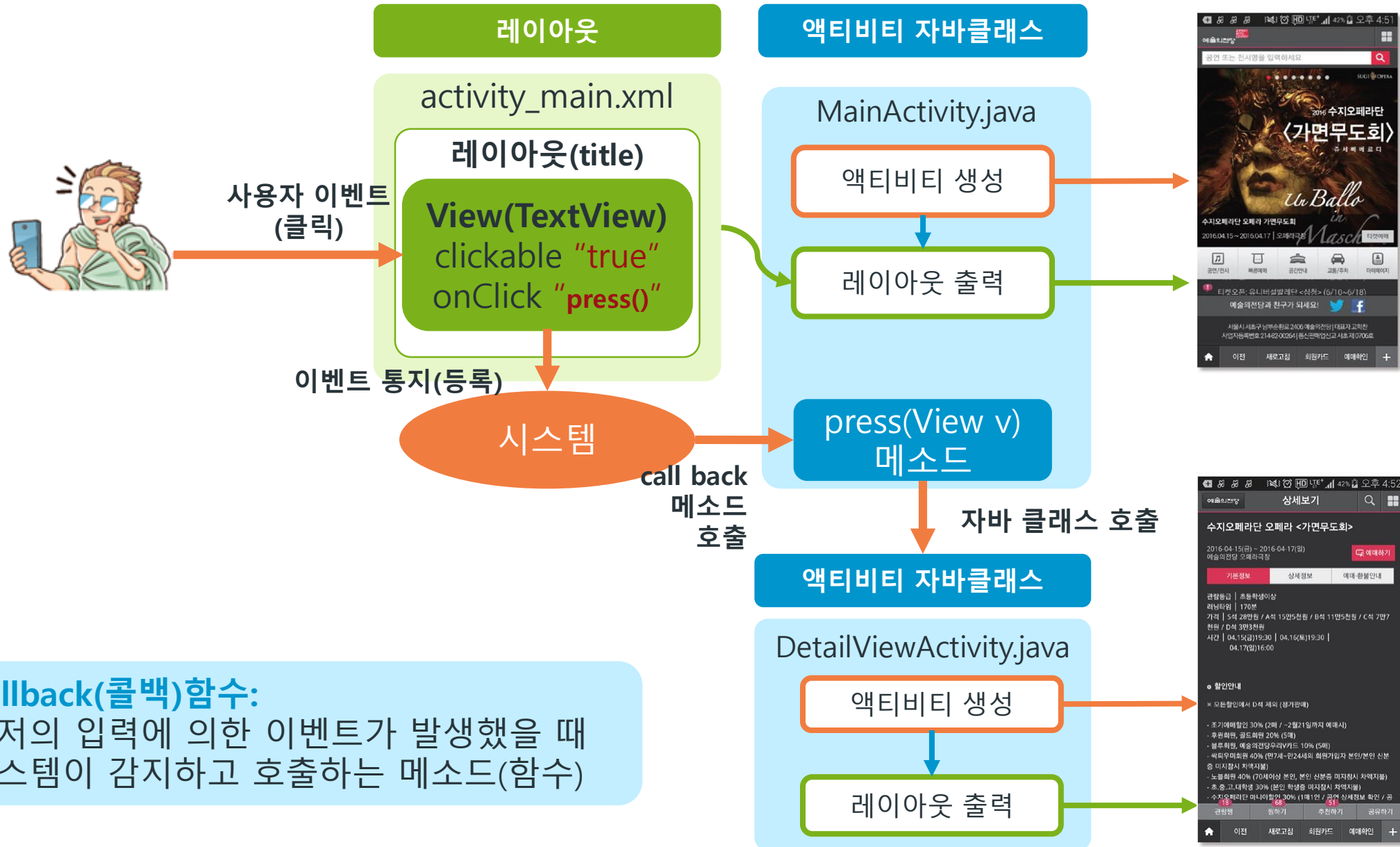


액티비티 2(상세 화면)

- 예술의 전당 앱

# 사용자 클릭 처리와 콜백 메소드(함수)

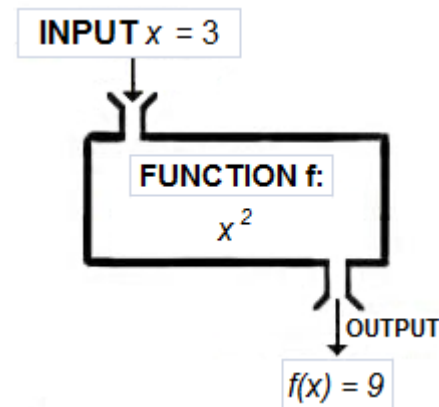
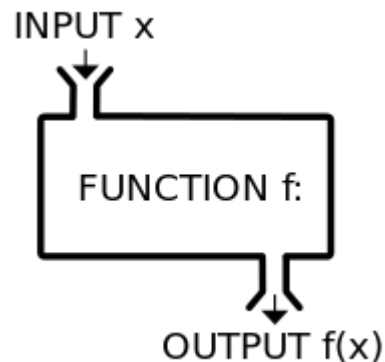
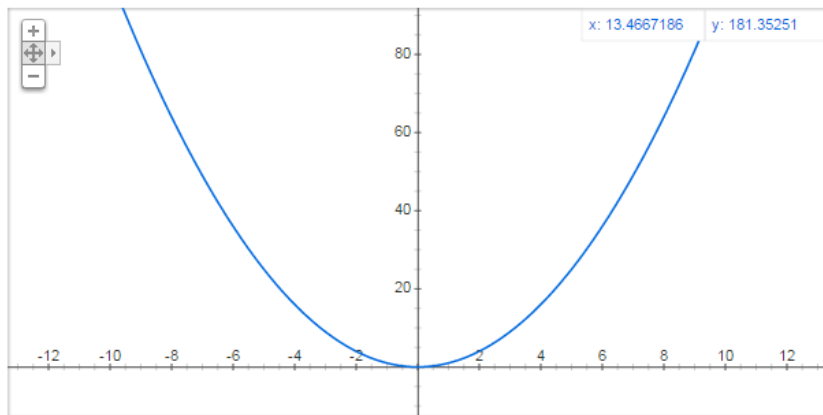
5



# 메소드(함수)란?

6

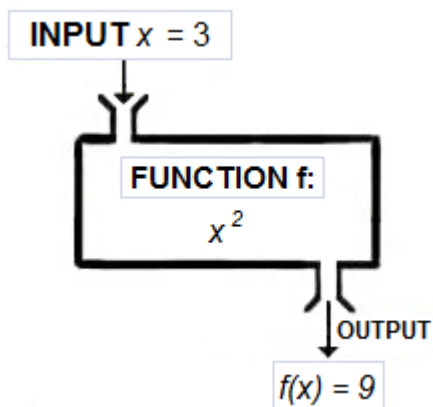
- 함수(Function)  $f(x) \rightarrow x^2$  OR  $f(x) = x^2$



**f:** function name

**int:** return type  
(output)

**int x:**  
parameters or arguments  
(input)



function f(input x)  
{  
  output x \* x;  
}

Java

```
int x = 3;  
int f(int x)  
{  
  return x * x;  
}
```

return x \* x;

function body

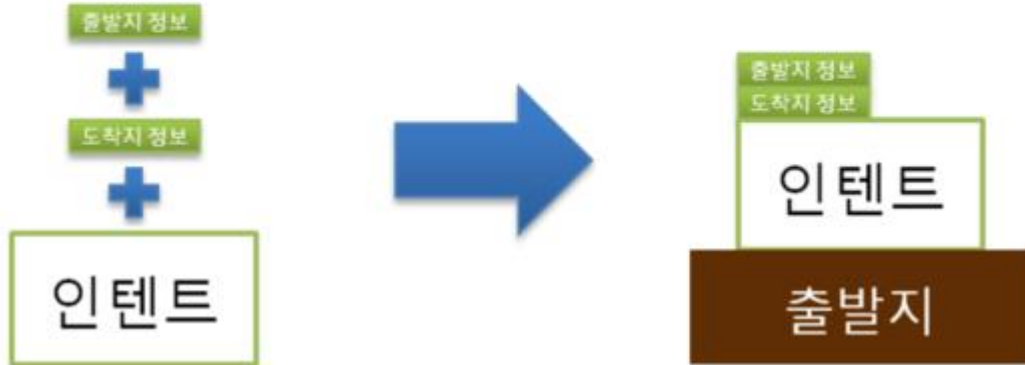


# 인테트 활용(정보 전달)

- 두번째 액티비티(화면)을 띄워서 **시 제목**과 **영화**를 보여준다.
- 이 때, intent를 사용해 **시 제목**과 **영화 파일이름**을 넘겨 준다.

## ① 출발지 액티비티에서 인텐트를 만든다.

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



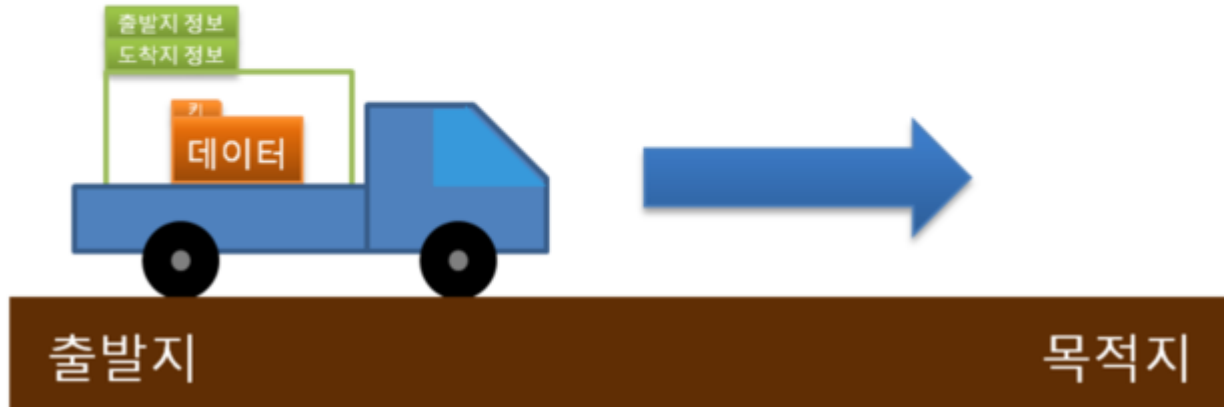
## ② 인텐트에 데이터를 적재한다.

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



## ③ 인텐트를 목적지 액티비티로 발송한다.

```
startActivity(intent);
```



## ④ 인텐트를 트럭에서(?) 내린다.

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

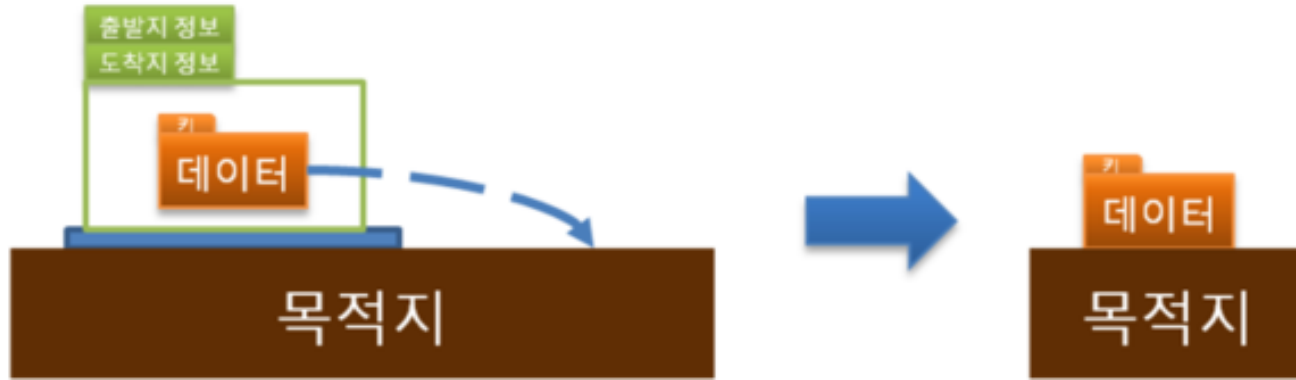




⑤ 인텐트에서 데이터를 꺼낸다.

9

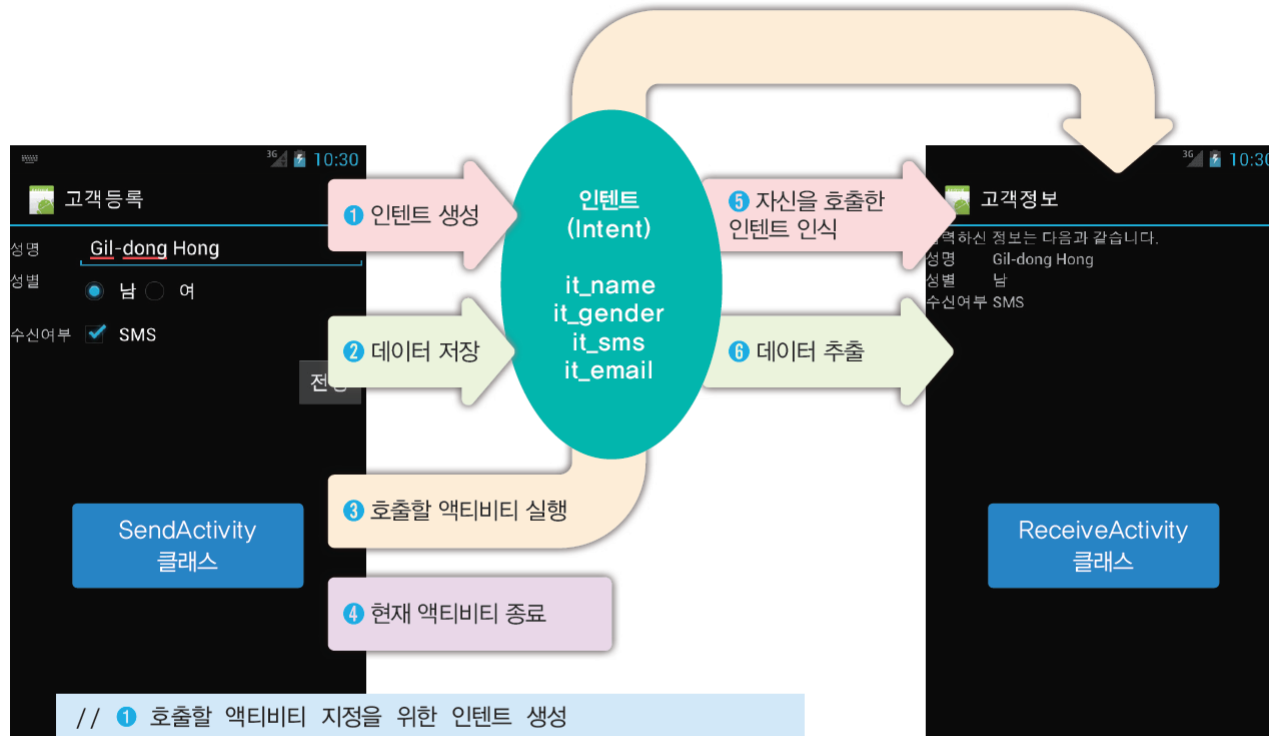
```
String data = intent.getStringExtra("value");
```



# 액티비티 간 이동과 정보 전달 – 인텐트(Intent)

10

```
// 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();
```

인텐트에는  
"it\_name" = "홍길동"  
"it\_gender" = "남"  
...

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

# 기존 프로젝트 열기

Intro\_ML\_withPython - Dropbox x IntroML/ x 02.supervised\_learning x Lecture2018/AndroidApp/sources x +

GitHub, Inc. [US] | https://github.com/hopypark/Lecture2018/tree/master/AndroidApp/sources

Search or jump to... / Pull requests Issues Marketplace Explore

hopypark / Lecture2018

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Branch: master Lecture2018 / AndroidApp / sources / Create new file Upload files Find file History

hopypark Update readme.md Latest commit c0451fe 21 minutes ago

EventActivity.zip Add files via upload 24 minutes ago

readme.md Update readme.md 21 minutes ago

readme.md

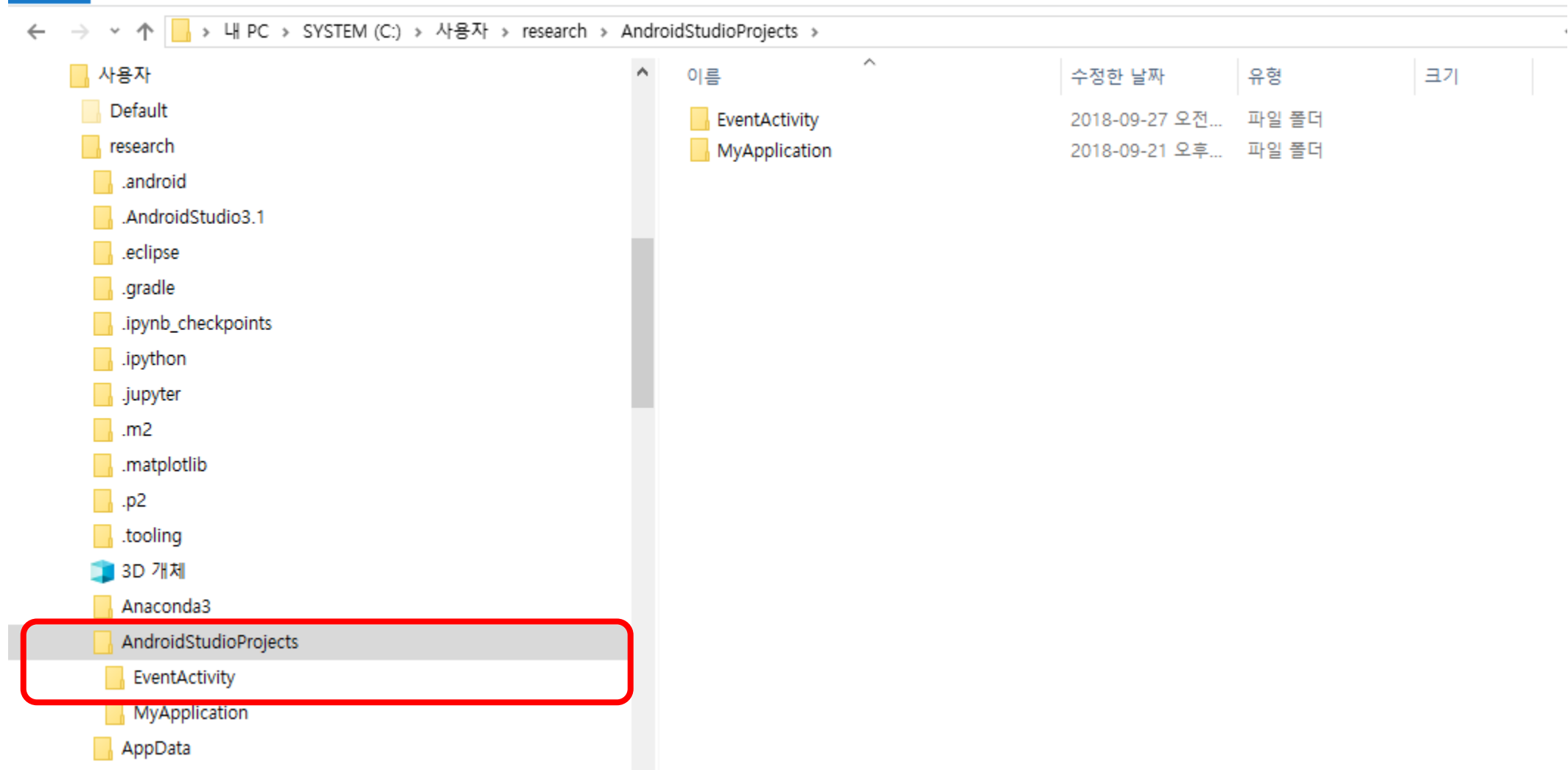
**Zip파일 소스 사용하기**

C:\Users\사용자이름\AndroidStudioProjects 폴더에 압축 파일이름으로 압축 풀기한다.

EventActivity.zip - 이벤트와 액티비티 전환 프로젝트 소스

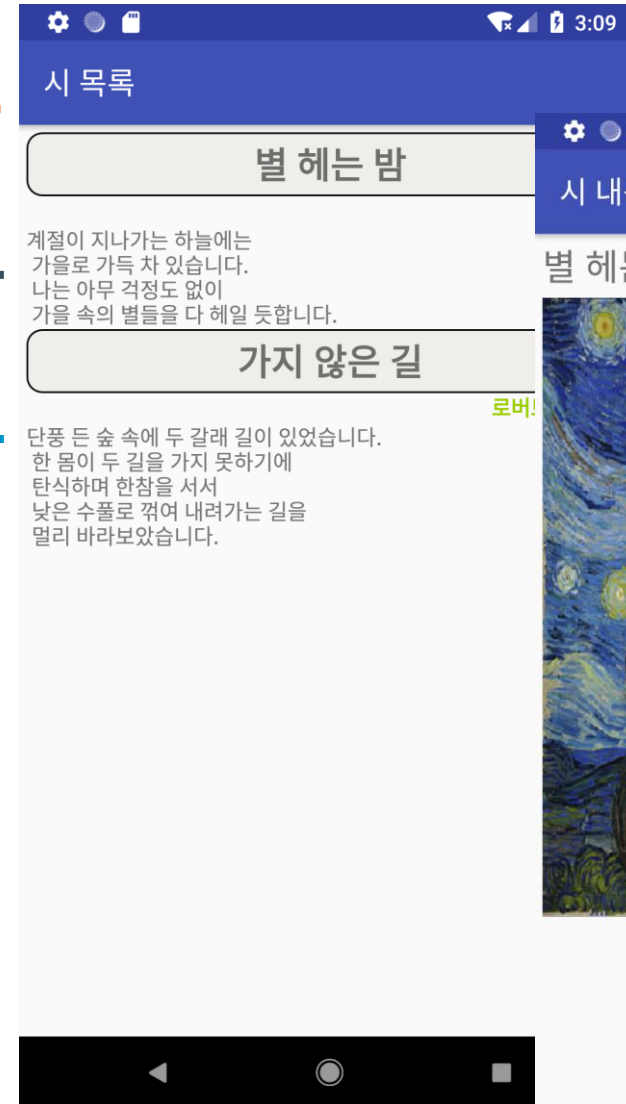
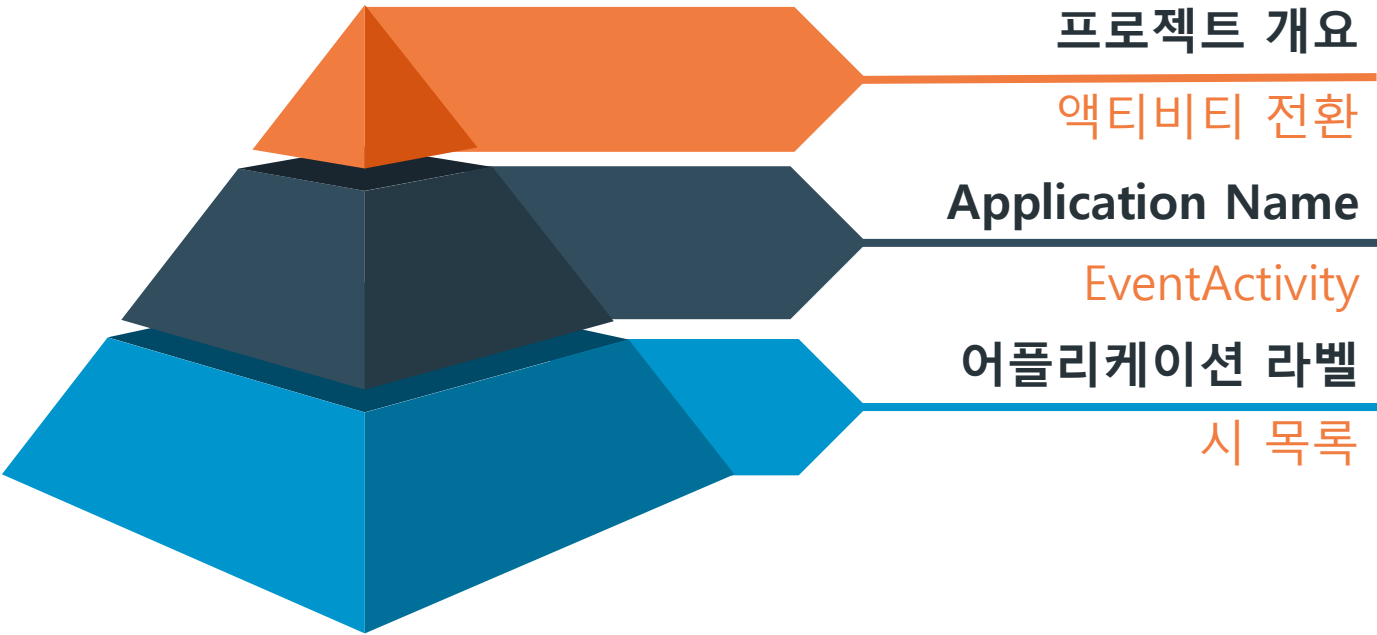
## • Zip파일 소스 사용하기

- C:\Users\사용자이름\AndroidStudioProjects 폴더에 압축 파일이름으로 압축 풀기한다.



# Step 0.프로젝트 개요

21



# Step 1. 프로젝트 생성

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

# Step 2. 파일 편집

23

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	• 영화 이미지
		still_life_with_kettle.png	• 영화 이미지
		shape_title.xml	• 시 제목 출력모양 설계 (배경색, 패딩, 모서리)
	layout	activity_main.xml	• 시 목록 화면 구성 (제목, 작가, 설명)
		activity_detail_view.xml	• 두번째 화면 구성 – 영화 출력
	mipmap	ic_launcher.png	
	values	dimens.xml	
		strings.xml	• 시 목록 아이템 (제목, 작가, 설명)
		styles.xml	

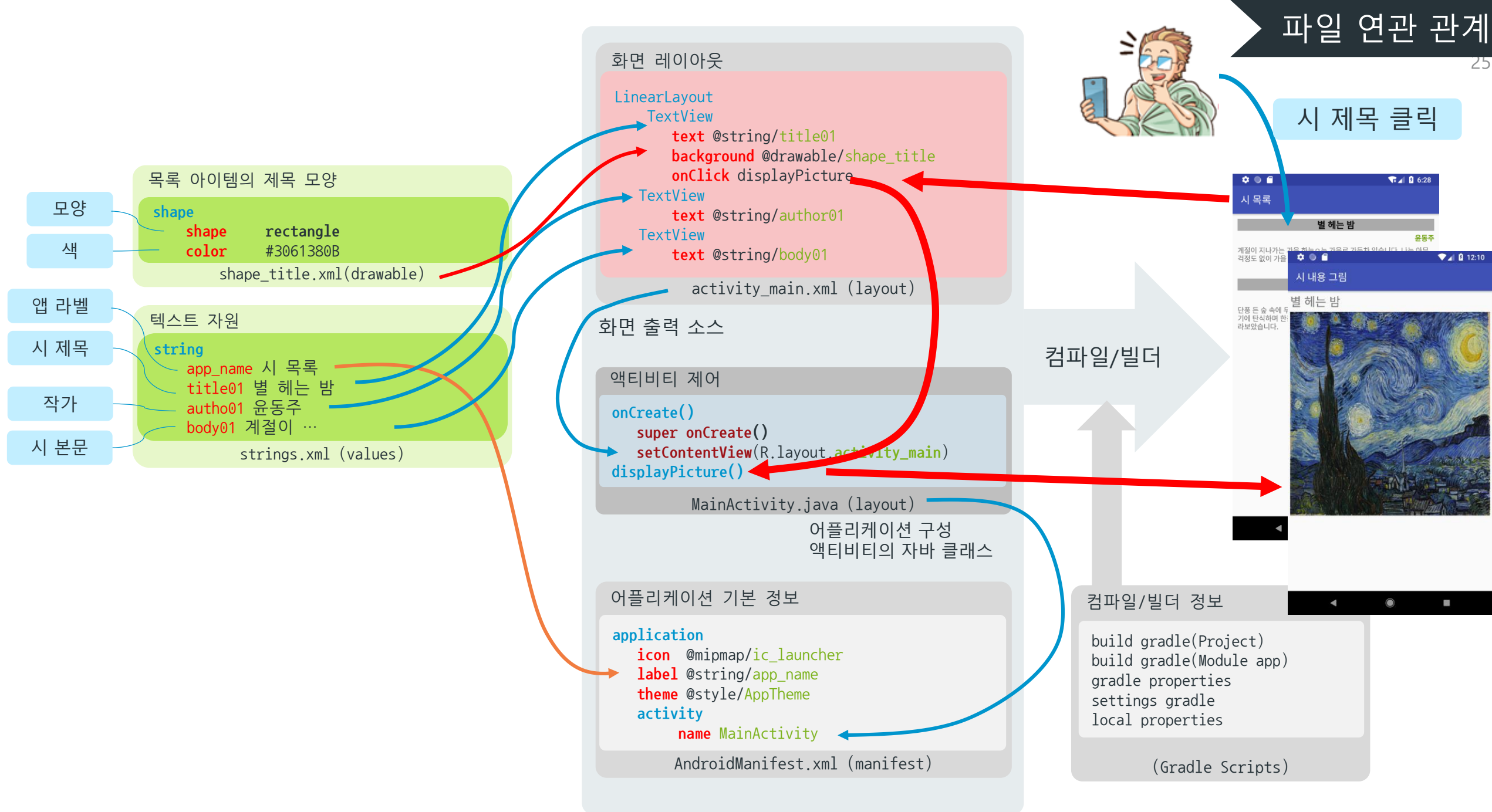
# Step 2.1 이미지 파일 복사

24

- res 폴더에 있는 **starry\_night.png** 파일을 **drawable** 폴더에 저장

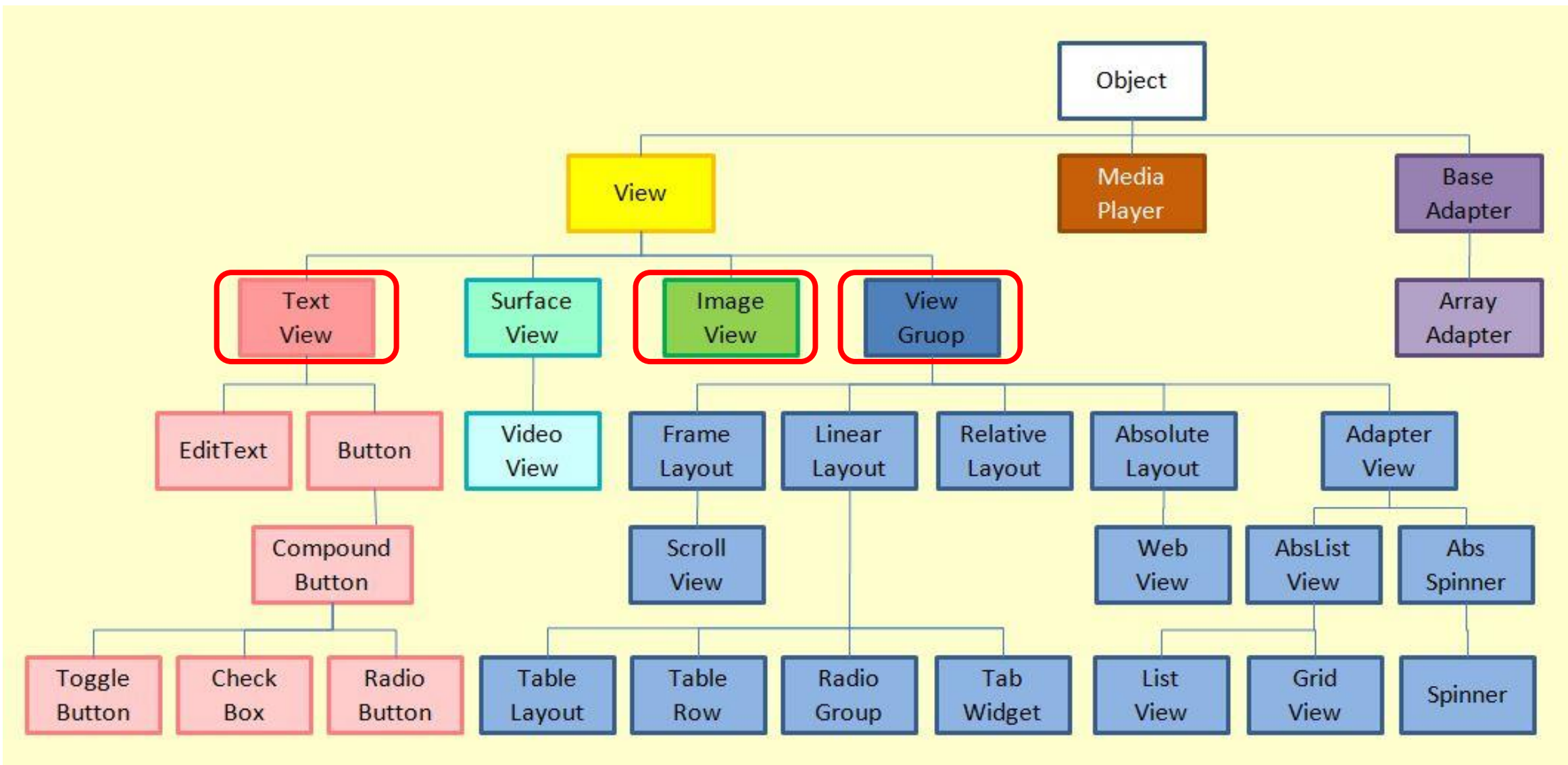
모듈	폴더	소스 파일	이미지
res	drawable	starry_night.png	
res	drawable	still_life_with_kettle.png	





# 안드로이드 자바 클래스 계층도

26



# 클래스와 속성/메소드 - TextView 클래스의 속성

- 화면 레이아웃 설정 파일인 activity\_main.xml에서 여러 위젯 (TextView, ImageView, ... 등의 컴포넌트) 속성 설정

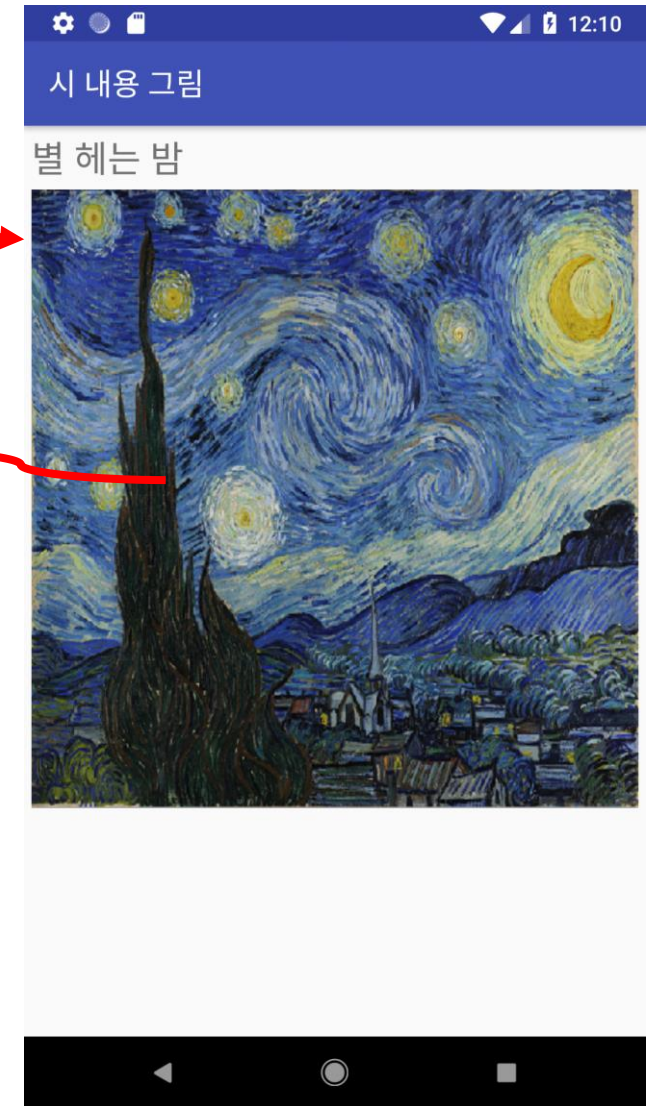
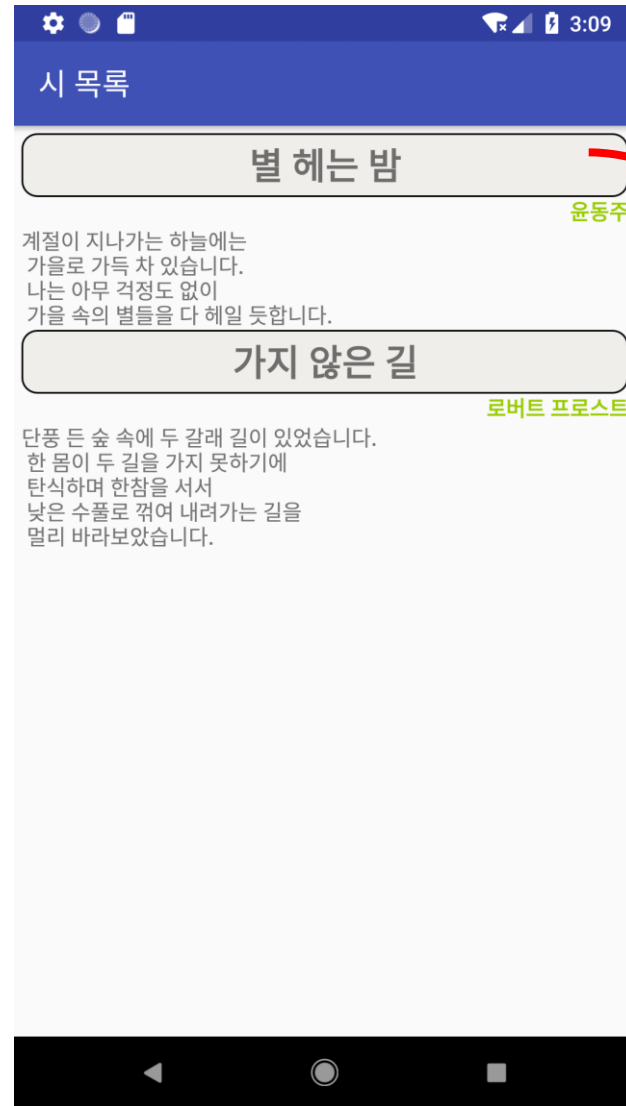
클래스(Class)	속성(Attributes)	설명(Description)
TextView	android:clickable	클릭 이벤트에 대한 반응 여부를 정의함(true, false)
	android:id	뷰(or 다른 위젯)를 구별하기 위한 이름 <b>android:id="@id/아이디 이름" 으로 지정</b>
	android:onClick	뷰가 클릭될 때 실행되는 메소드의 이름 <b>(메소드는 MainActivity.java 파일에 코딩)</b>
	android:tag	문자열로 나타내는 태그 - 명화 그림파일 이름

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

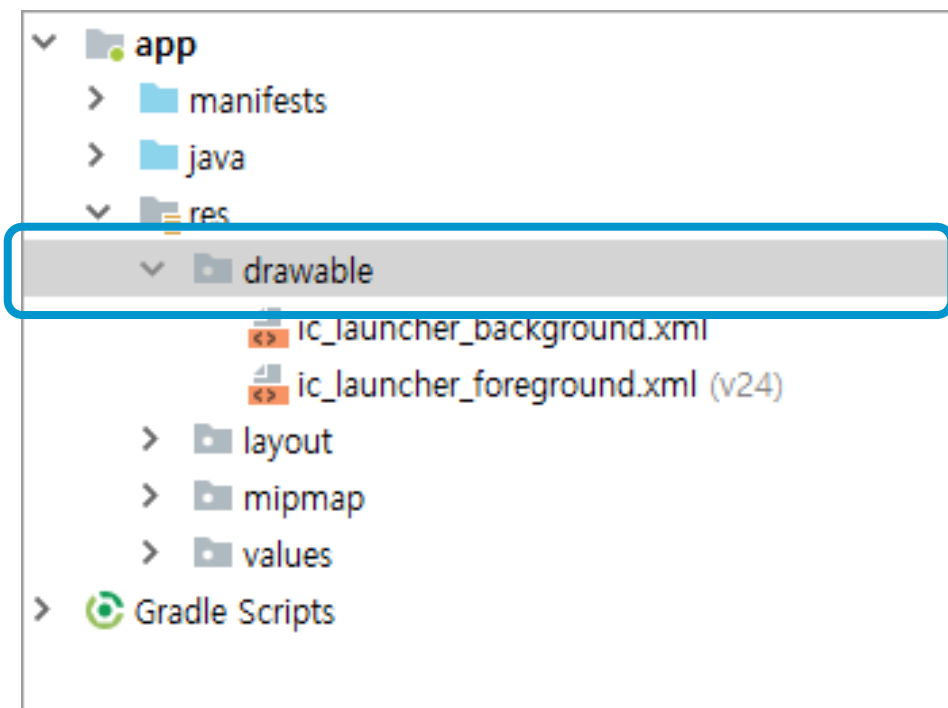
# 이벤트와 액티비티 전환



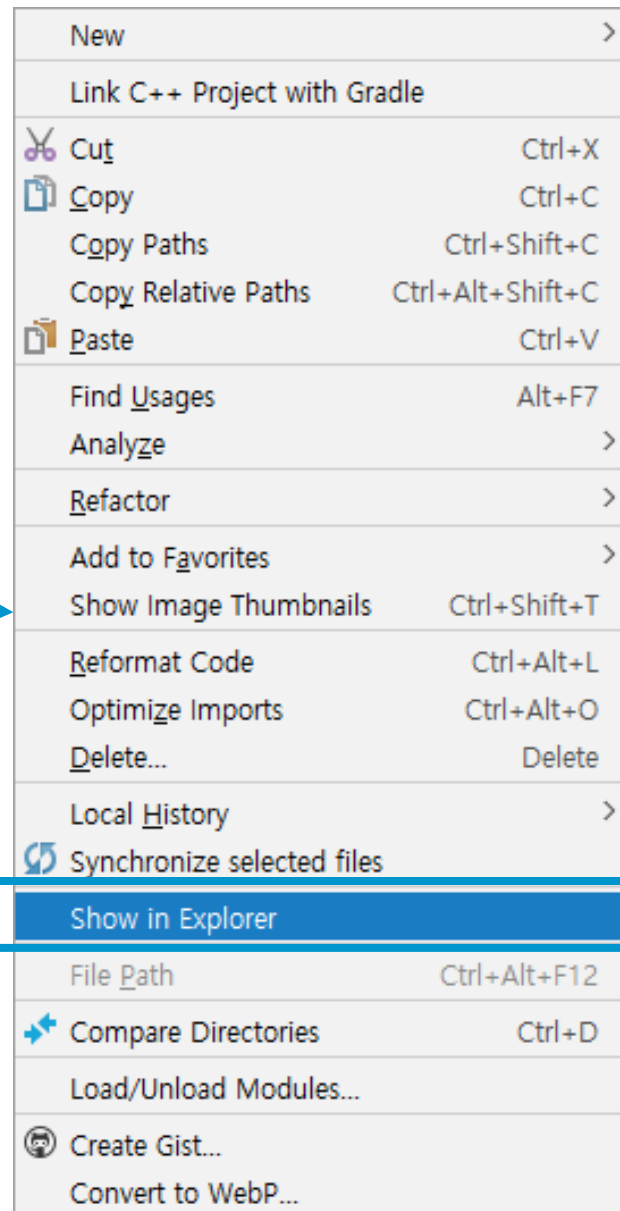
# O utputs



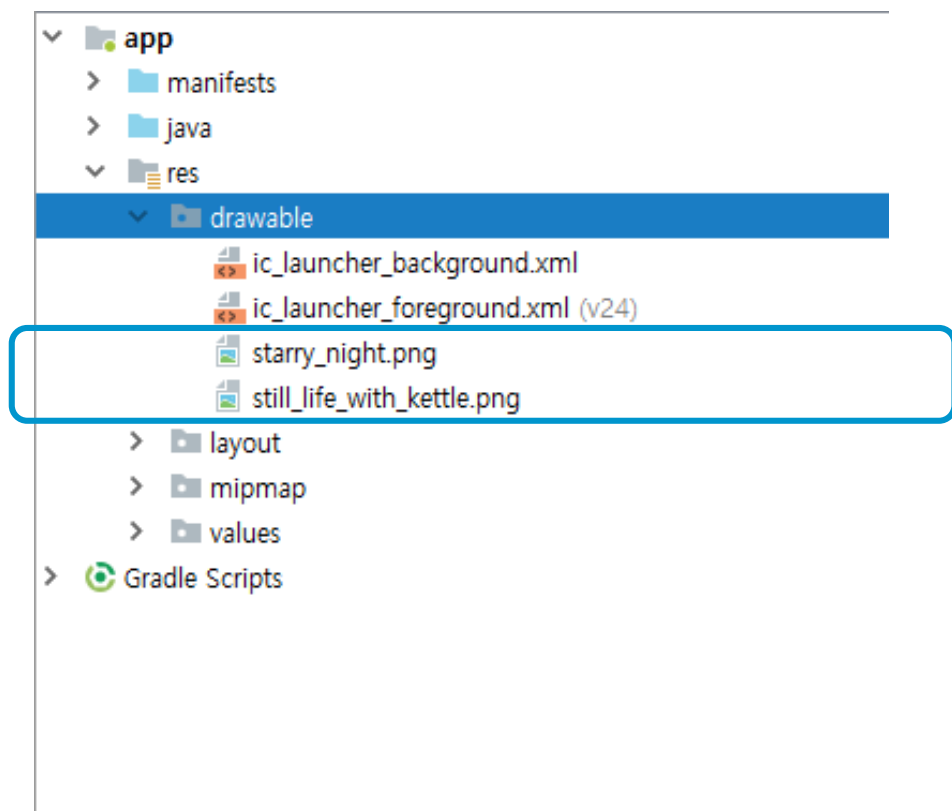
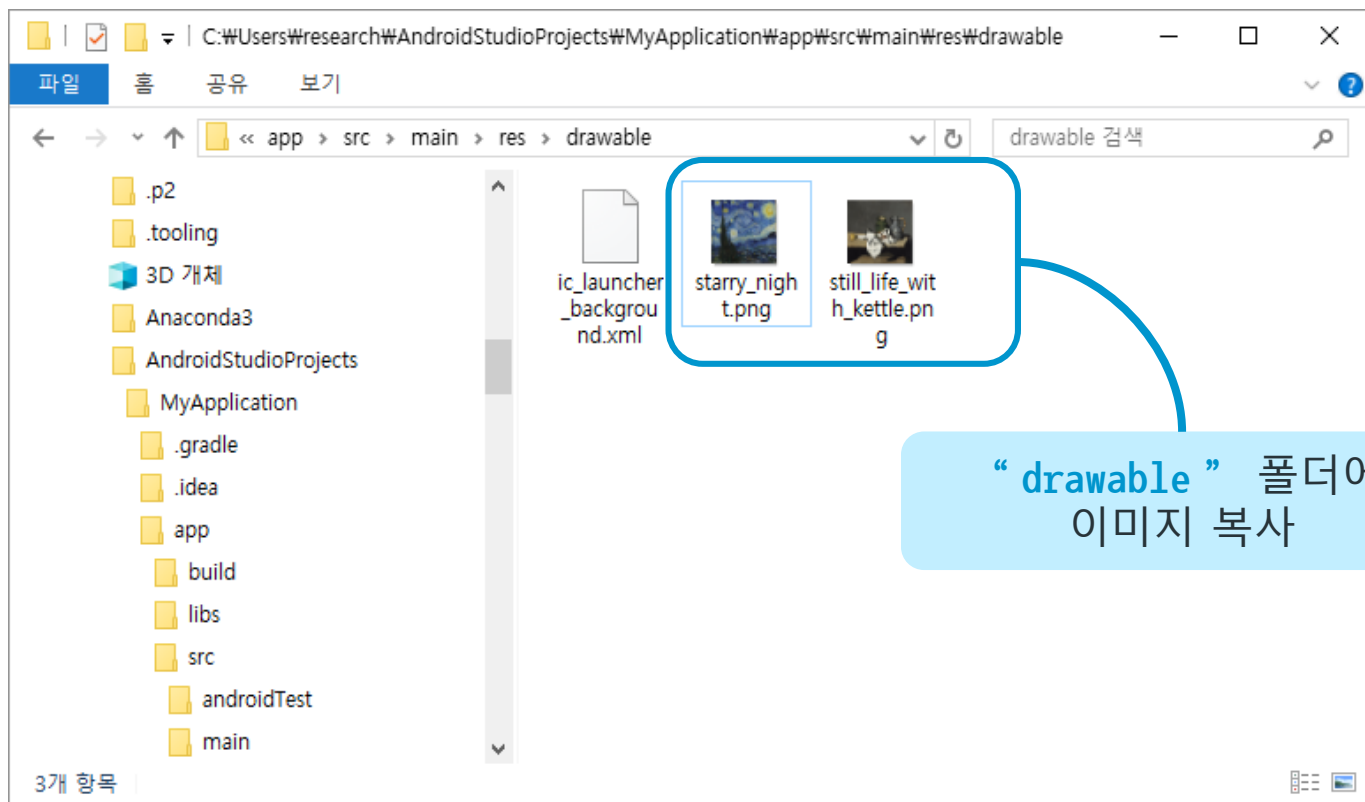
# drawable 폴더에 이미지 추가하기



팝업 메뉴  
띄우기



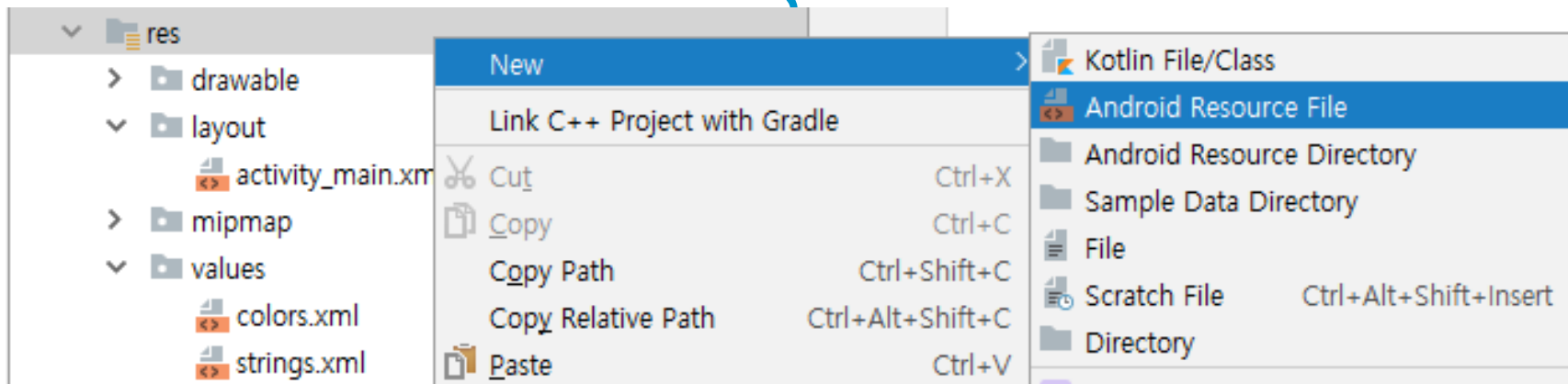
원도우 탐색기 창보기  
(탐색기에서 이미지 추가)



# 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

The screenshot shows the 'New Resource File' dialog box. The fields are filled as follows:

- File name: shape\_title.xml
- Resource type: Drawable
- Root element: shape
- Source set: main
- Directory name: drawable

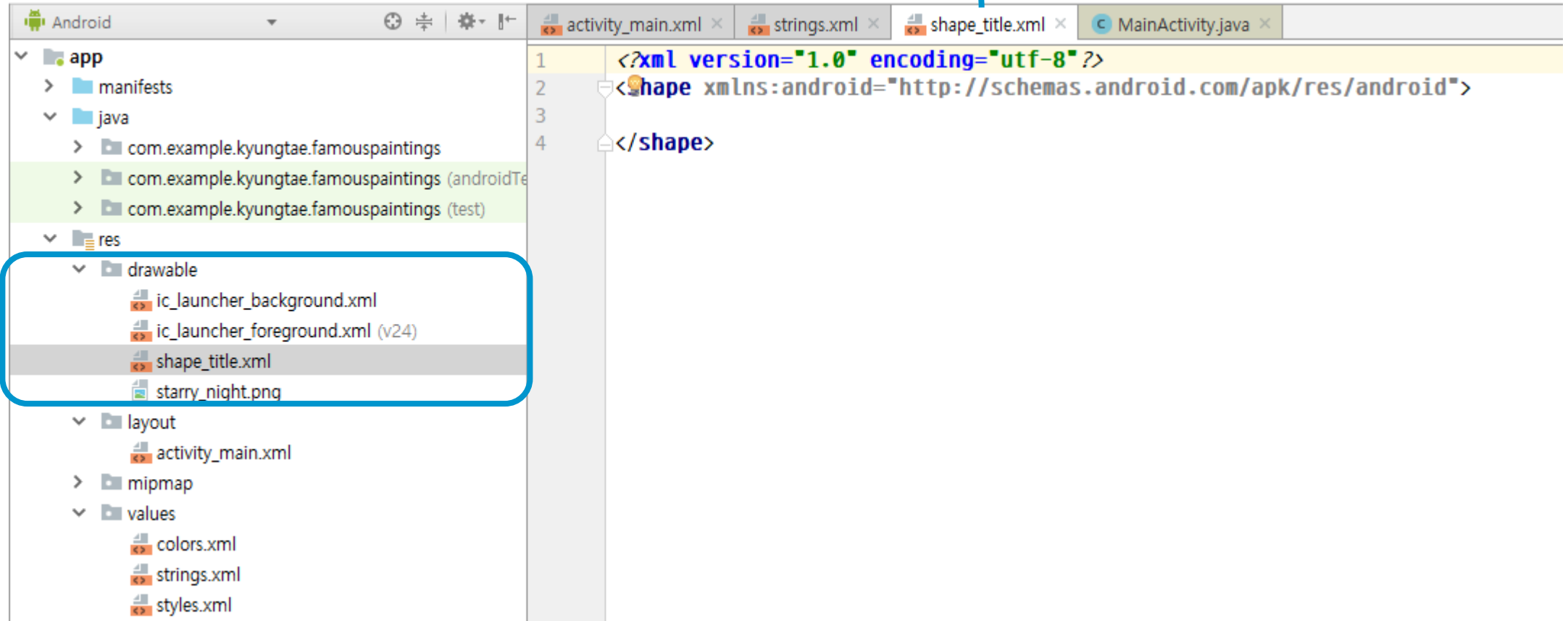
Below the fields, there are two lists:

- Available qualifiers:** Country Code, Network Code, Locale, Layout Direction, Smallest Screen Width, Screen Width, Screen Height, Size, Ratio, Orientation, UI Mode.
- Chosen qualifiers:** Nothing to show

At the bottom right, the 'OK' button is highlighted with a blue box, and the 'Cancel' button is also visible.

- 생성된 shape\_title.xml

shape\_title.xml 파일



## • shape\_title.xml 소스

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="rectangle">
3      <solid android:color="#1061380B" />
4
5      <stroke
6          android:width="1dp"
7          android:color="#000000" >
8      </stroke>
9
10     <padding
11         android:left="10dp"
12         android:top="5dp"
13         android:right="10dp"
14         android:bottom="5dp" >
15     </padding>
16
17     <corners
18         android:radius="10dp" >
19     </corners>
20 </shape>
```

사각형의 배경색 지정

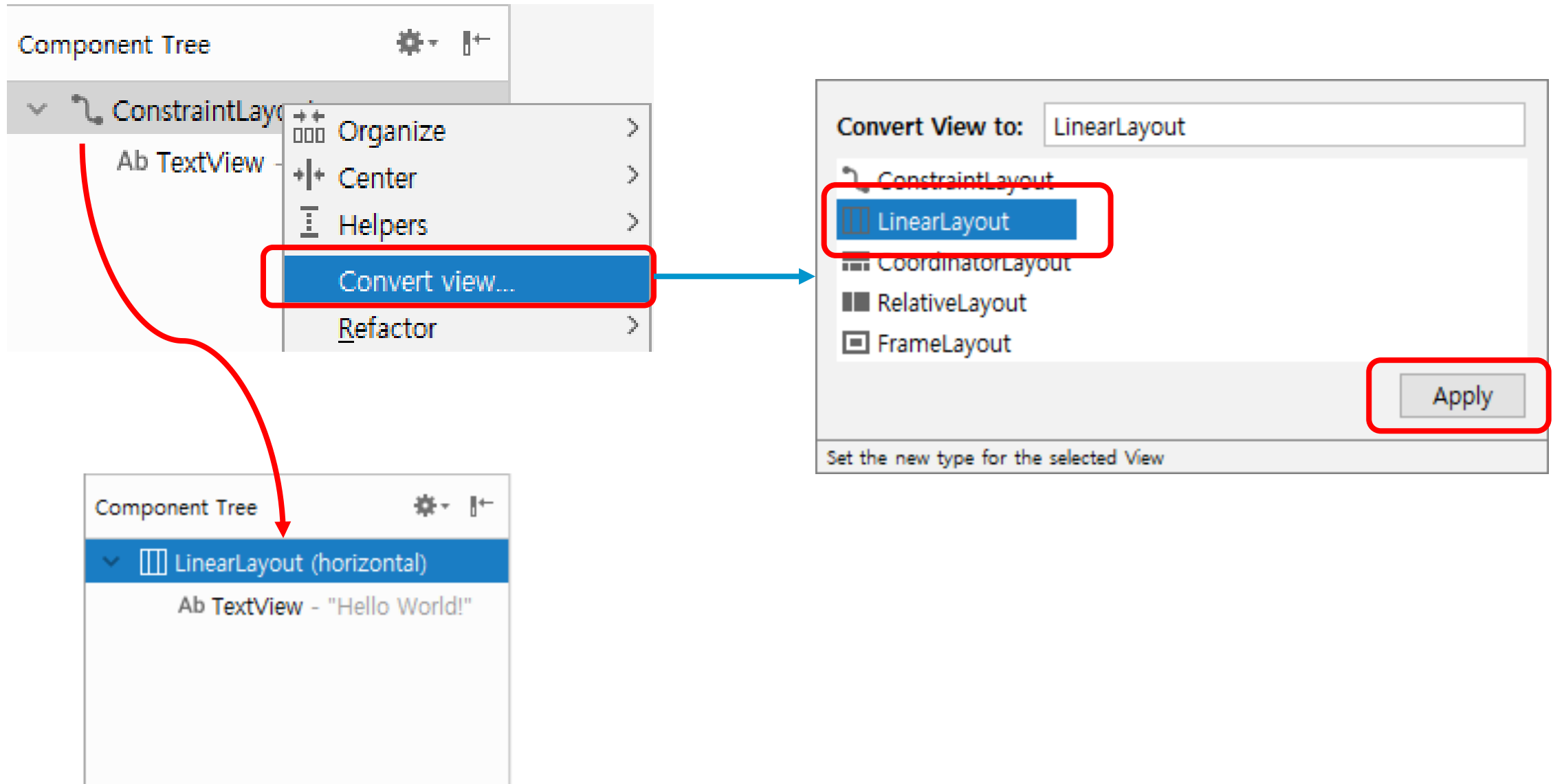
사각형 테두리 색

내부 패딩 정보

출력모양 모서리를 둥근 모양으로 지정(반지름은 5dp)

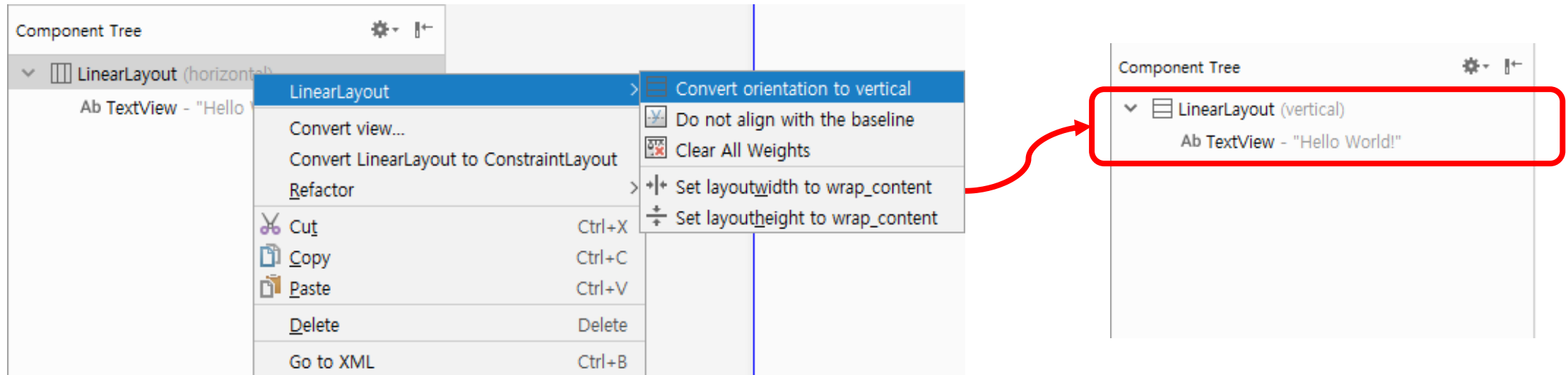
# ConstraintLayout을 LinearLayout로 바꾸기

44



# LinearLayout의 방향을 Horizontal → Vertical로 변경하기

45

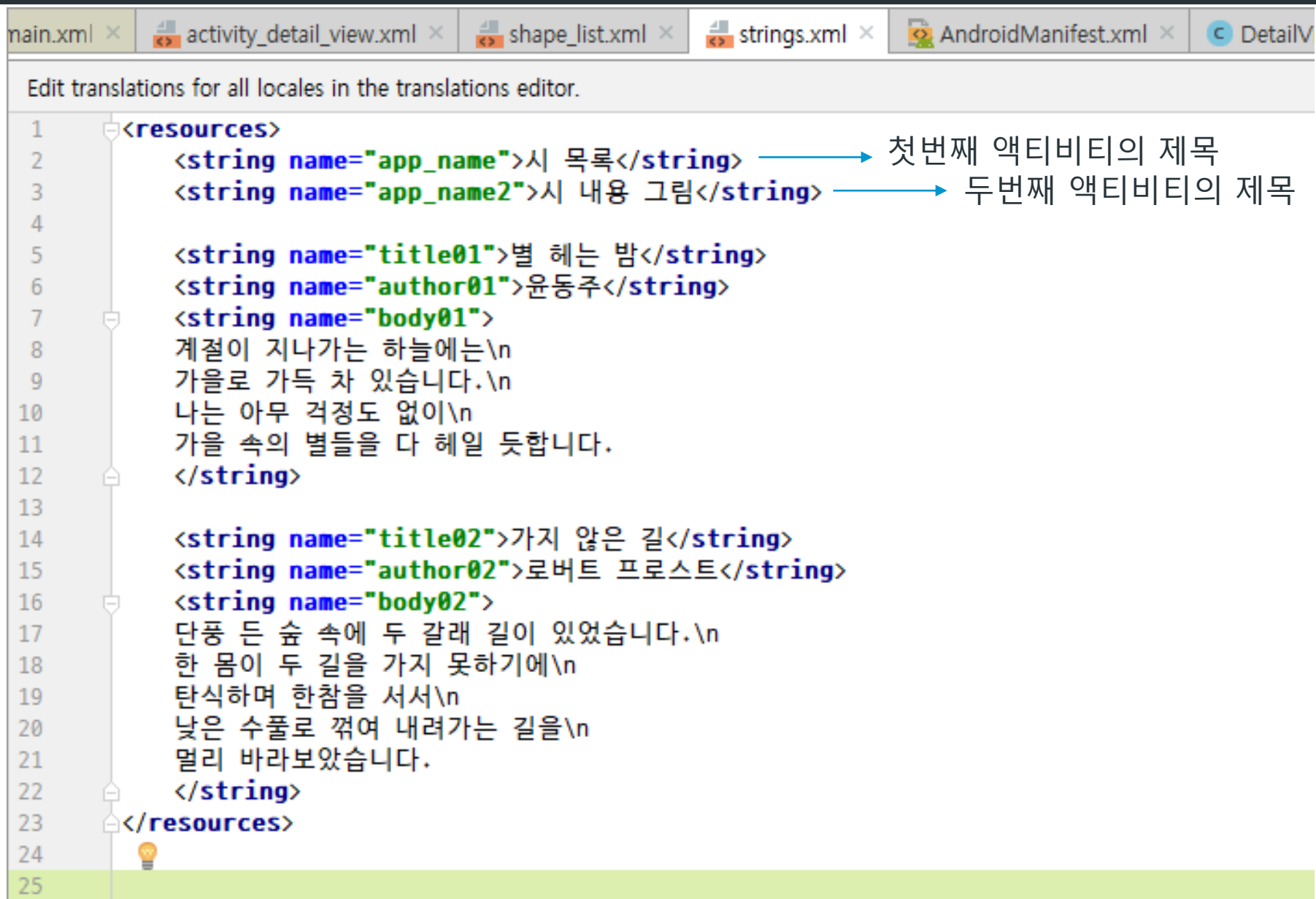




The image shows a code editor with two tabs: 'activity\_main.xml' and 'MainActivity.java'. The 'activity\_main.xml' tab is active, displaying XML code for an Android layout. The code is as follows:

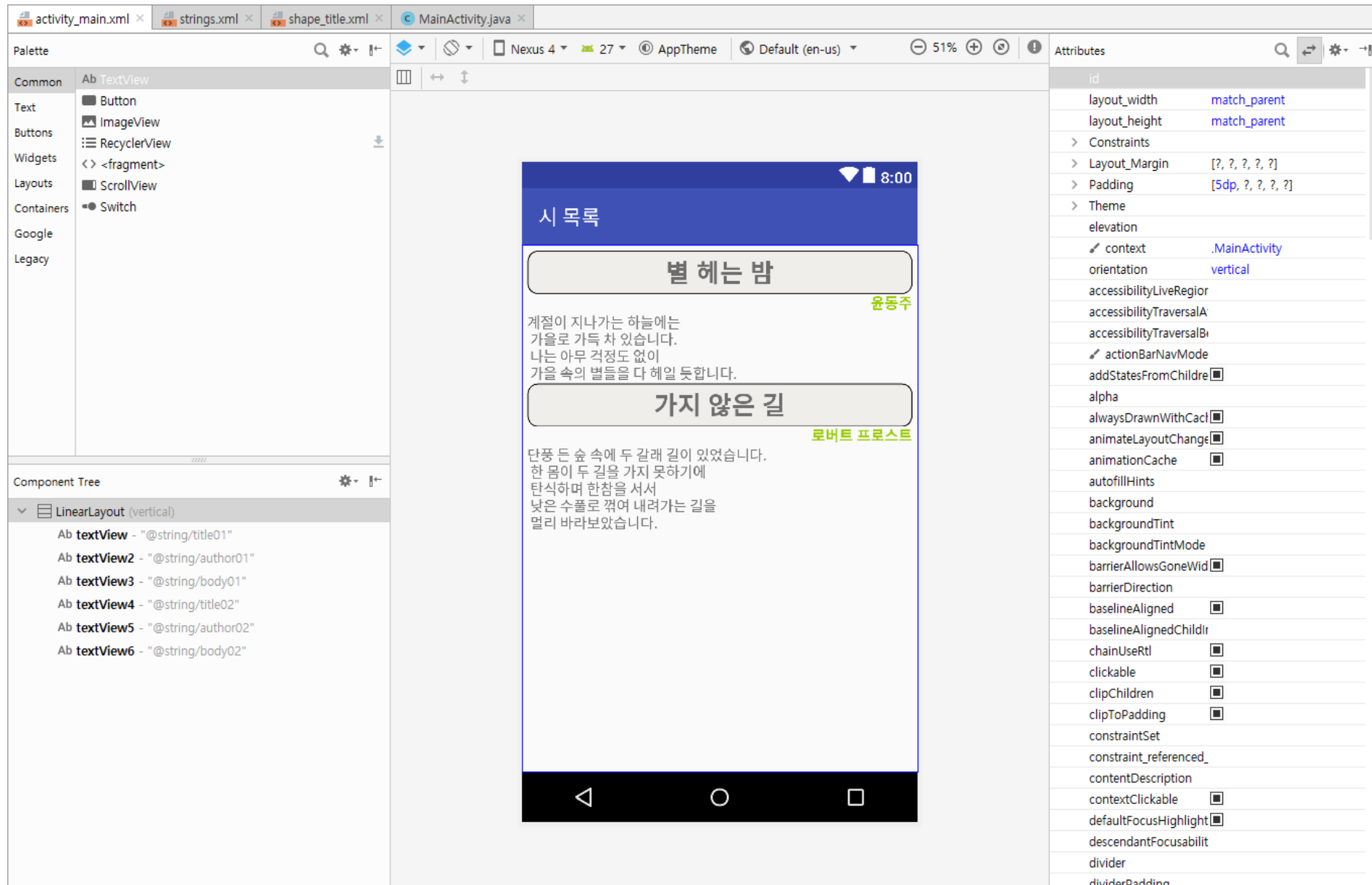
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
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>
```

A red rectangular box highlights the `<TextView>` element on lines 10 through 13. A blue arrow points from the right side of this box to a light blue button labeled '삭제' (Delete) in Korean.



# TextView에 속성을 적용한 결과 화면

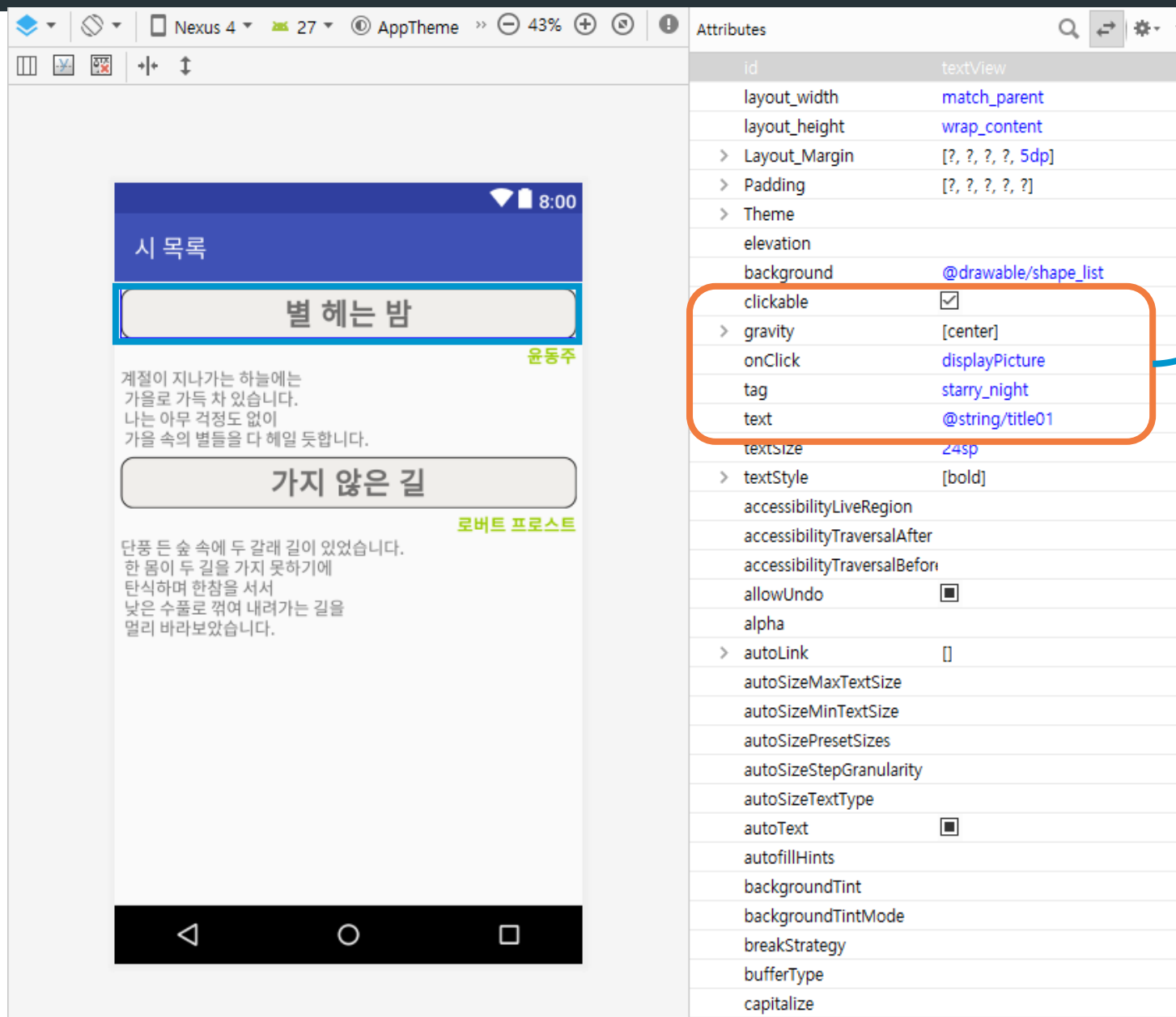
48



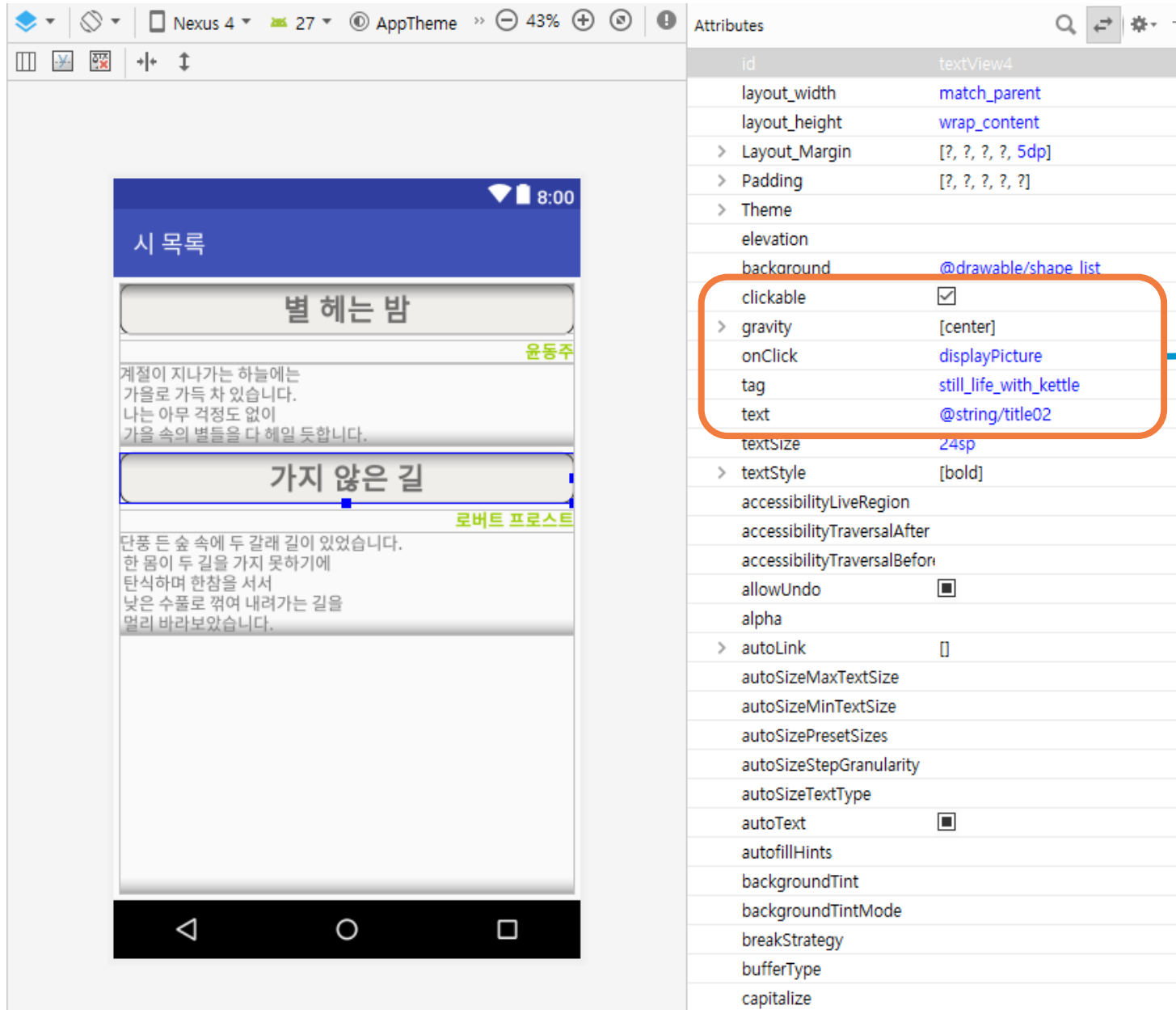


# Step 2.2 파일 편집-기존 예제에서 속성 수정

49



**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

51

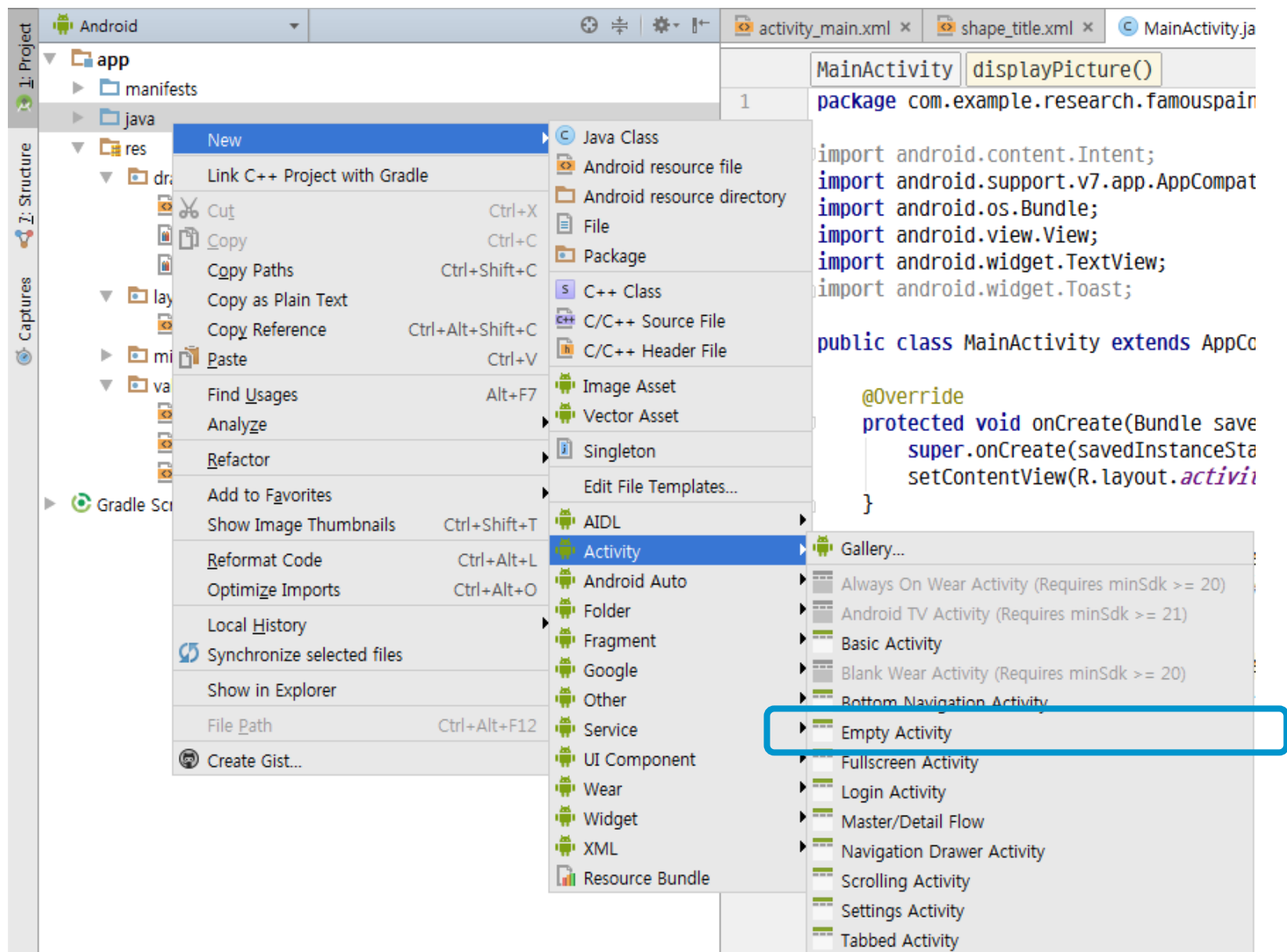
```
activity_main.xml x activity_detail_view.xml x AndroidManifest.xml x C DetailViewActivity.java x C MainActivity.java x S
1 package com.example.user.eventactivity;
2
3 import android.content.Intent;
4 import android.support.v7.app.AppCompatActivity;
5 import android.os.Bundle;
6 import android.view.View;
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 }
```

MainActivity.java

displayPicture 메소드 추가

# Step 2.2 파일 편집 – 두 번째 Activity 추가

52



New Android Activity

## Configure Activity

Android Studio

**Creates a new empty activity**

Activity Name:

☒ Generate Layout File

Layout Name:

☐ Launcher Activity

☒ Backwards Compatibility (AppCompat)

Package name:

Source Language:

The name of the layout to create for the activity

! Activity Name must be unique

Previous Next Cancel Finish

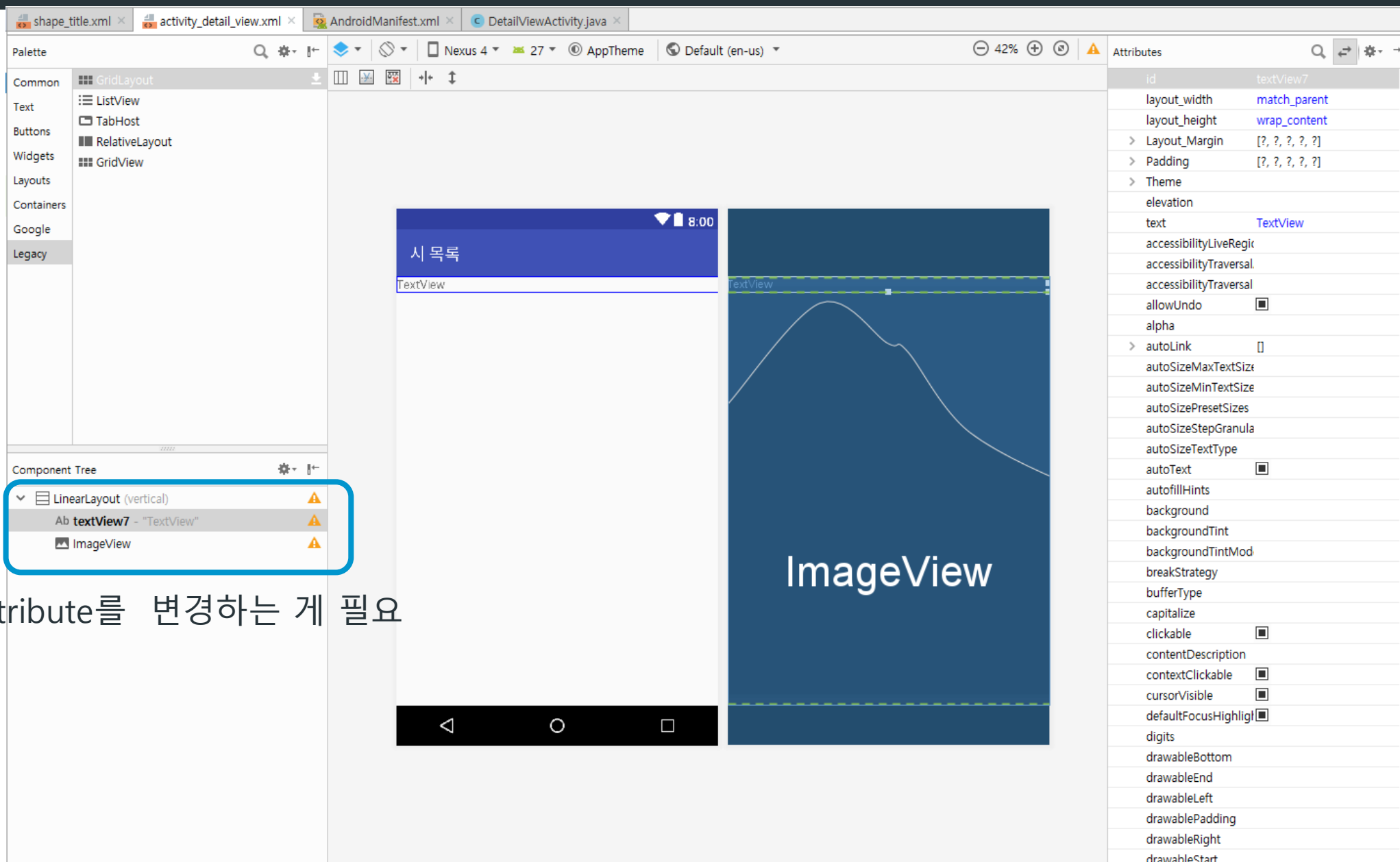
**Activity Name:**  
DetailViewActivity

**Layout Name:**  
activity\_detail\_view

**Package name(개인마다 다름):**  
com.example.user.eventactivity

# Step 2.2 파일 편집-activity\_detail\_view.xml

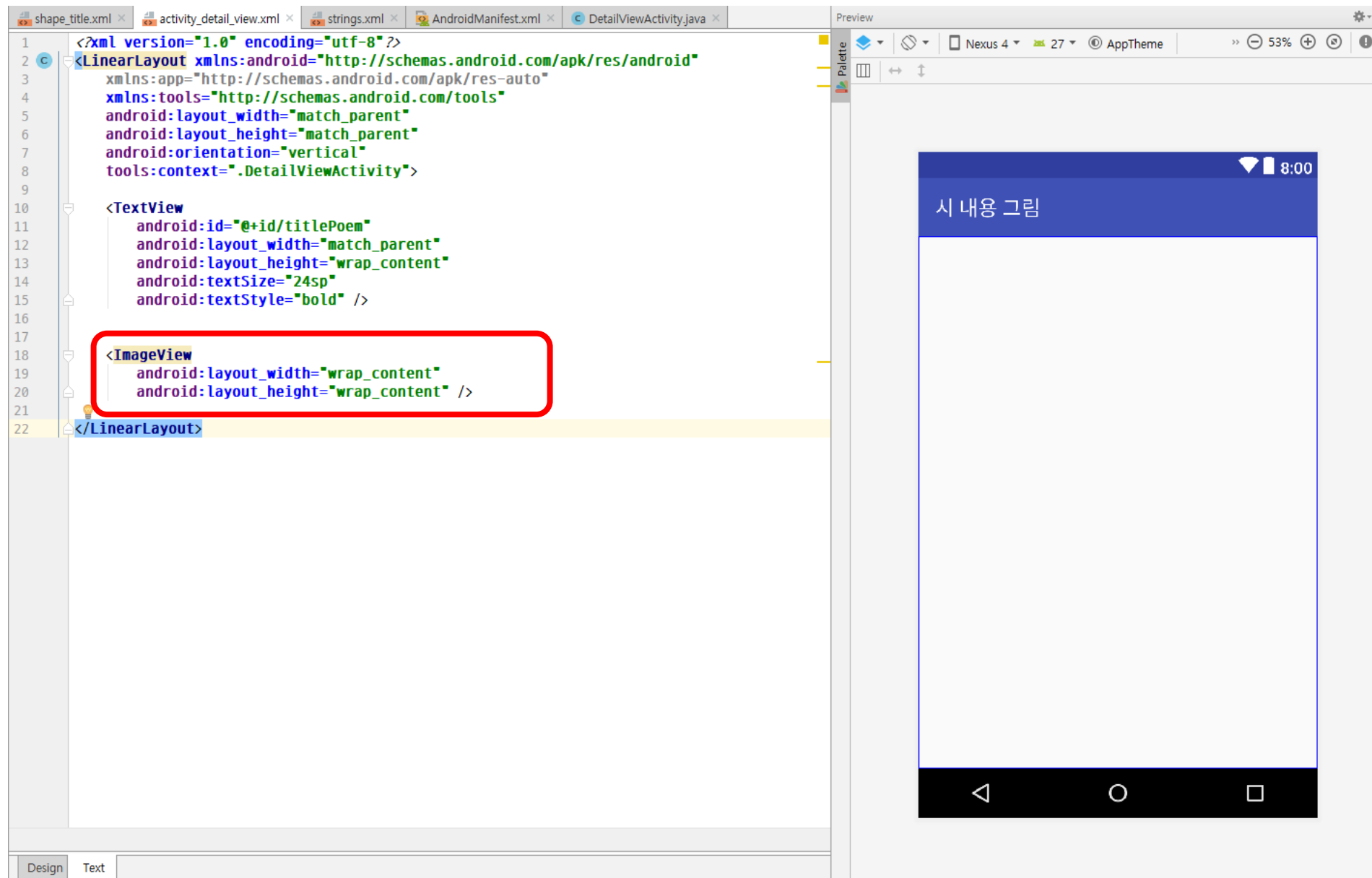
54



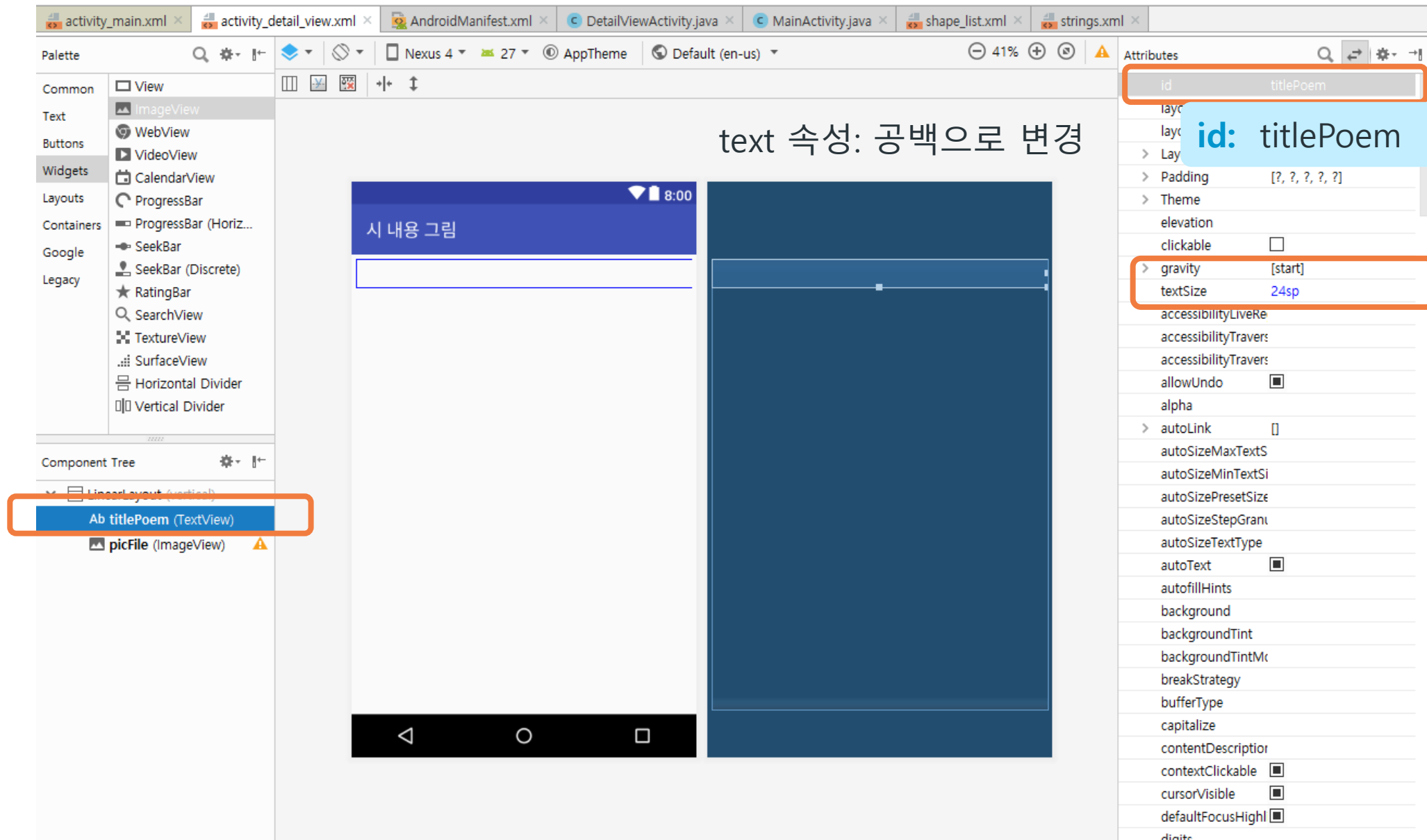
# ImageView 추가하기

55

- 기본 이미지를 선택하지 않음 → **Text 모드에서 작업해야 함.**



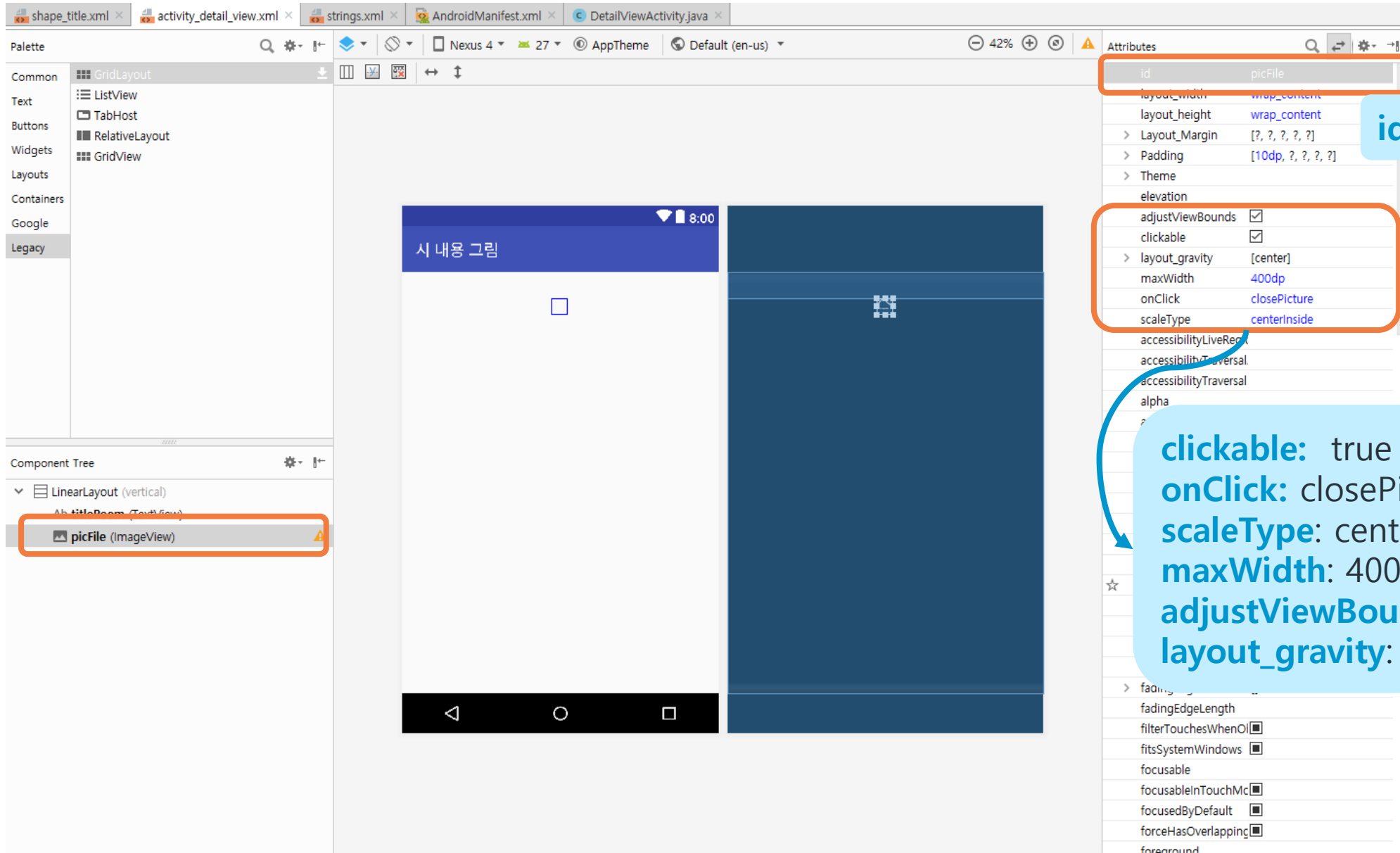
# TextView 속성 변경





# ImageView 속성 설정

60



id: picFile

clickable: true  
onClick: closePicture  
scaleType: centerinside  
maxWidth: 400dp  
adjustViewBounds: true  
layout\_gravity: center

# Step 2.2 파일 편집-AndroidManifest.xml 파일 수정

63

- 두번째 액티비티의 제목 추가(수정)

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.research.famouspaintings">
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="시 목록"
9         android:roundIcon="@mipmap/ic_launcher_round"
10        android:supportRtl="true"
11        android:theme="@style/AppTheme">
12        <activity android:name=".MainActivity">
13            <intent-filter>
14                <action android:name="android.intent.action.MAIN" />
15
16                <category android:name="android.intent.category.LAUNCHER" />
17            </intent-filter>
18        </activity>
19        <activity android:name=".DetailViewActivity"></activity>
20    </application>
21
22 </manifest>
23
24
```

상자 부분만 수정  
(줄바꿈은 안 해도 됨)

```
ity_main.xml x activity_detail_view.xml x AndroidManifest.xml x shape_list.xml x string
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.user.eventactivity">
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="시 목록"
9         android:roundIcon="@mipmap/ic_launcher_round"
10        android:supportRtl="true"
11        android:theme="@style/AppTheme">
12        <activity android:name=".MainActivity">
13            <intent-filter>
14                <action android:name="android.intent.action.MAIN" />
15
16                <category android:name="android.intent.category.LAUNCHER" />
17            </intent-filter>
18        </activity>
19        <activity android:name=".DetailViewActivity"
20            android:label="@string/app_name2">
21            </activity>
22    </application>
23
24 </manifest>
```

# Step 2.2 파일 편집-DetailViewActivity.java

68

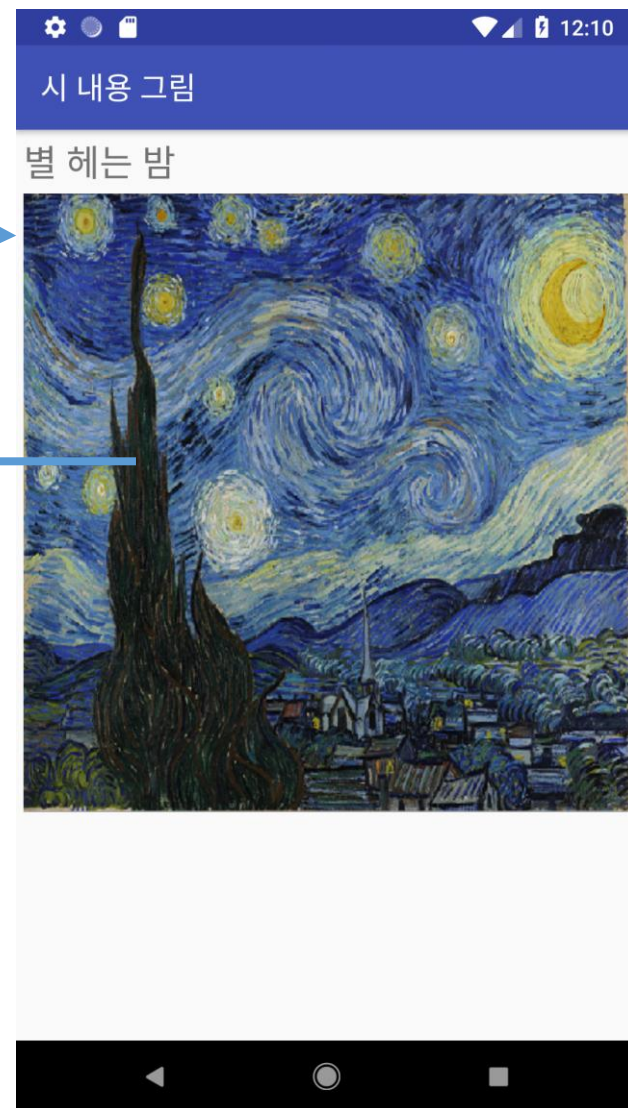
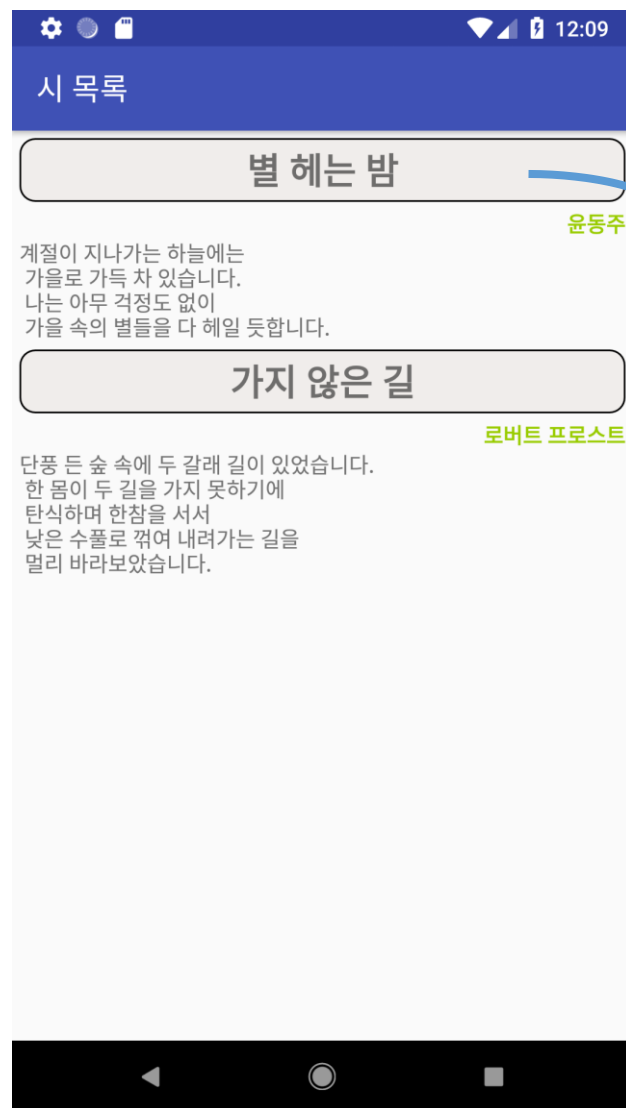
```
activity_main.xml x activity_detail_view.xml x AndroidManifest.xml x DetailViewActivity.java x MainActivity.java x sha
1 package com.example.user.eventactivity;
2
3 import ...
14
15 public class DetailViewActivity extends AppCompatActivity {
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_detail_view);
21
22         Intent it = getIntent(); // 현재 Activity로 전송된 Intent를 얻는다.
23         String title = it.getStringExtra( name: "it_title"); // "it_title"이름으로 저장된 내용을 가져온다.
24         String tag = it.getStringExtra( name: "it_tag"); // "it_gat"이름으로 저장된 내용을 가져온다.
25
26         // TextView에 내용 입력하기
27         TextView tvTitle = (TextView) findViewById(R.id.titlePoem); // TextView 객체 확인
28         tvTitle.setText(title);
29         // ImageView에 그림 표시하기
30         ImageView ivPicture = (ImageView) findViewById(R.id.picFile); // ImageView 객체 확인
31         Resources res = getResources();
32         int id_img = res.getIdentifier(tag, defType: "drawable", getPackageName()); // tag→ 그림파일이름(리소스 이름),
33         Drawable drawable = res.getDrawable(id_img, theme: null); // getPackageName(): 현재 어플리케이션의 패키지 이름
34         ivPicture.setBackground(drawable);
35     }
36
37     public void closePicture(View view){
38         finish();
39     }
40 }
```

Intent로부터 받아온 내용을  
TextView에 표시하고, 그림  
을 표시한다.

// tag→ 그림파일이름(리소스 이름),  
getPackageName(): 현재 어플리케이션의 패키지 이름

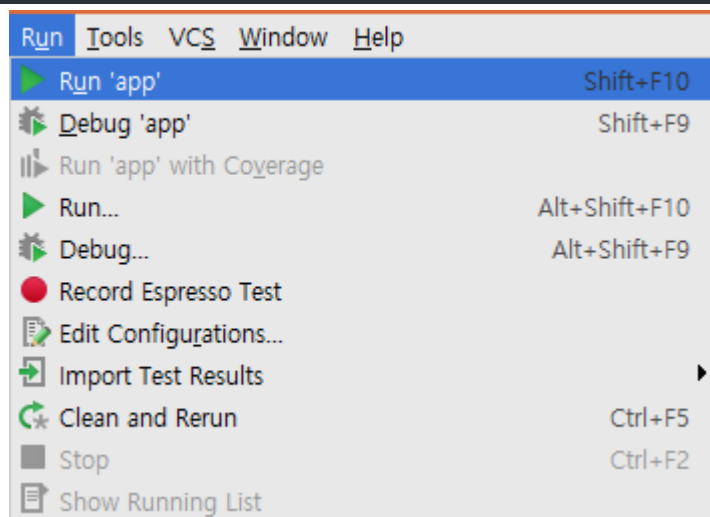
액티비티에 보여진 그림을 클  
릭하면 실행됨(화면 닫기).

# O utputs



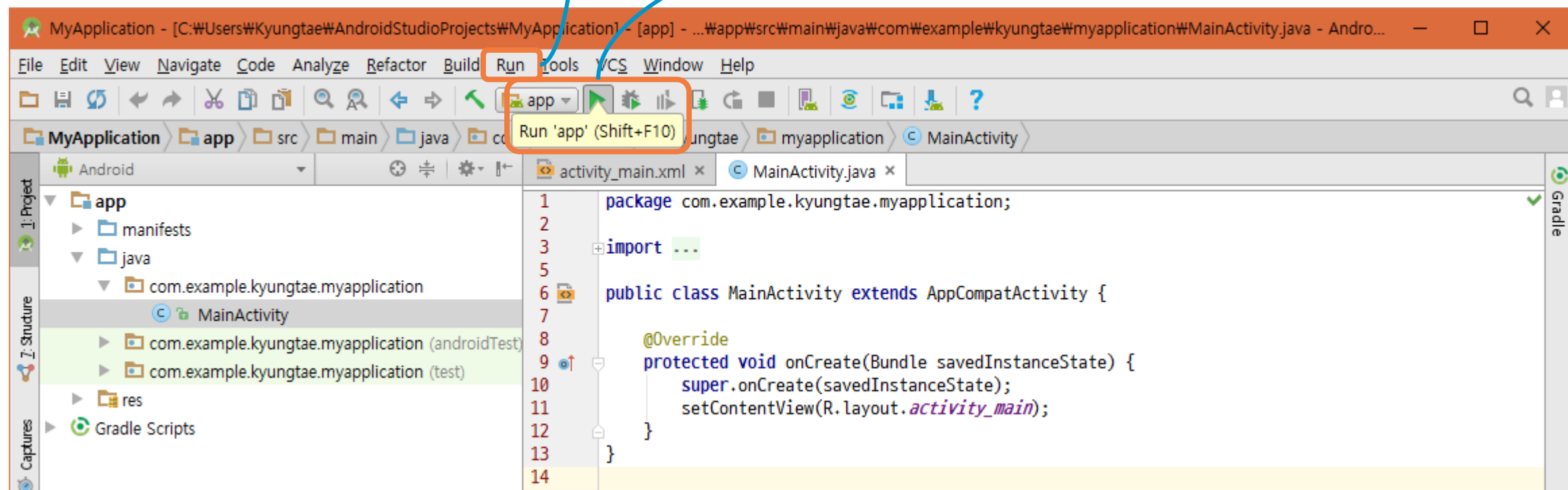
# Step 3. 프로젝트 실행

71



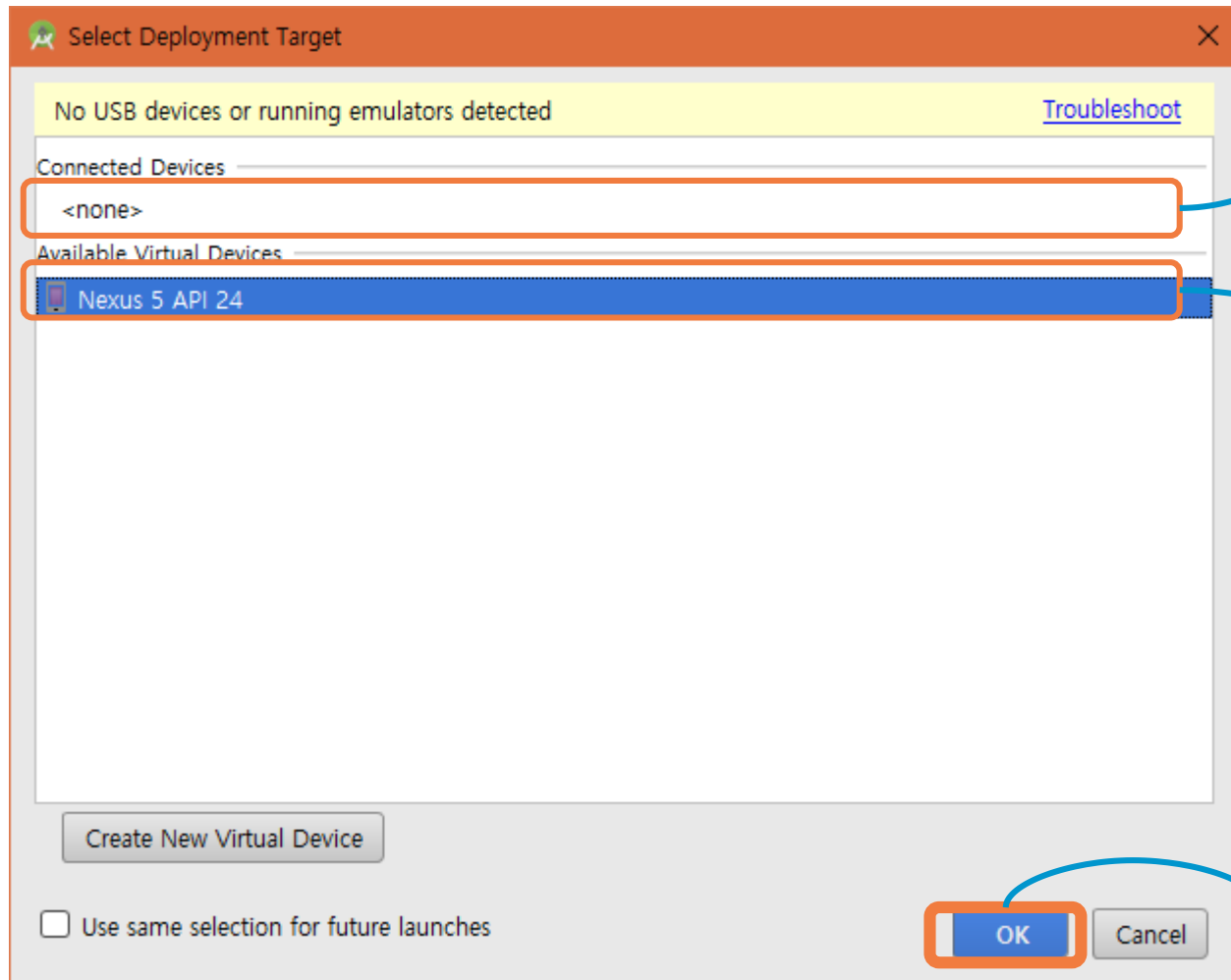
Run → Run 'app' 메뉴 클릭

앱 실행 아이콘 클릭



## • AVD 장비 선택하기

72



데이터 케이블로 연결된  
스마트폰

AVD

스마트폰 또는 AVD를 선택하고  
'OK' 버튼을 클릭

# Q & A

uestion  
nswer

83

