

# Week04. 이미지의 출력



# 개발환경 구축 절차

2

주 차	수업 내용
1	수업 소개
2	개발 환경 구축과 맛보기 프로젝트
3	텍스트 출력과 레이아웃
4	<b>이미지의 출력</b>
5	이벤트 처리와 액티비티 간 이동
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>



강의 자료 - <http://github.com/hopypark>

# 이미지 출력 앱의 예



(a) 사진 목록(인스타그램)

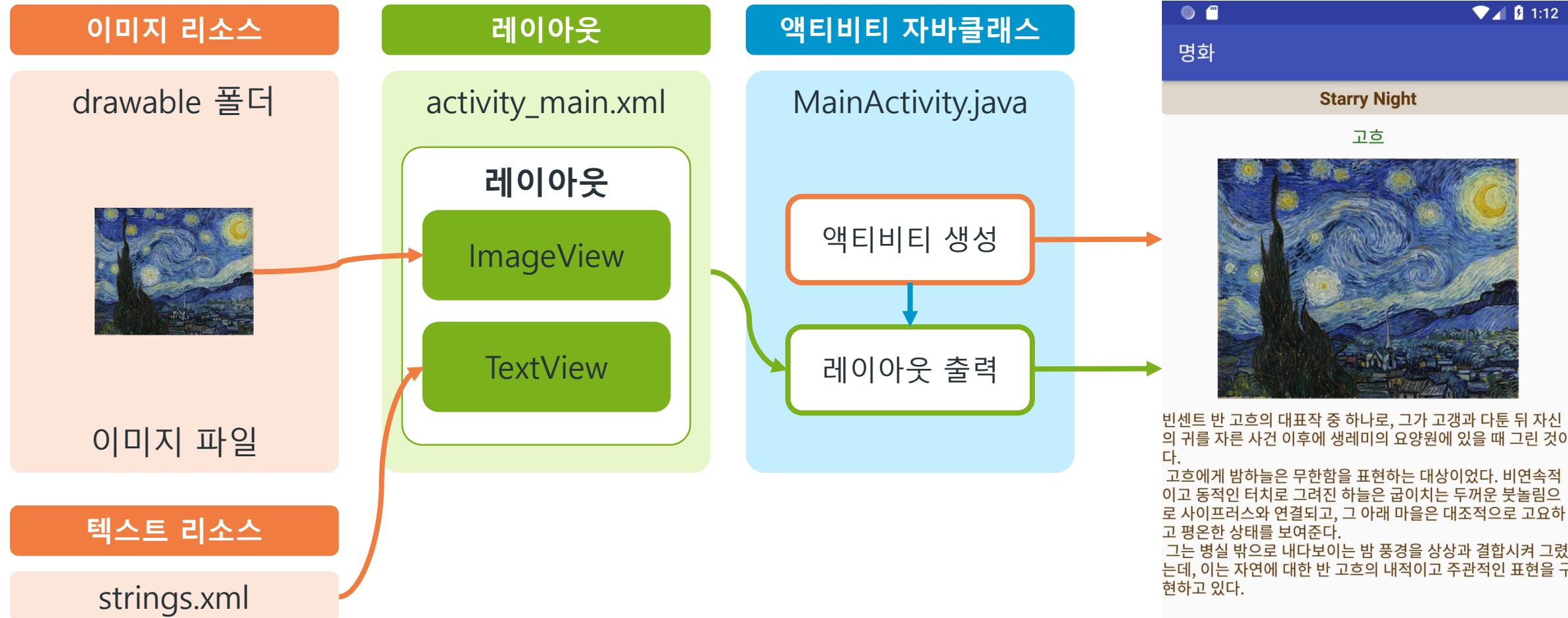


(b) 음식 배달 메뉴(배달의민족)

• 이미지 중심 앱의 예

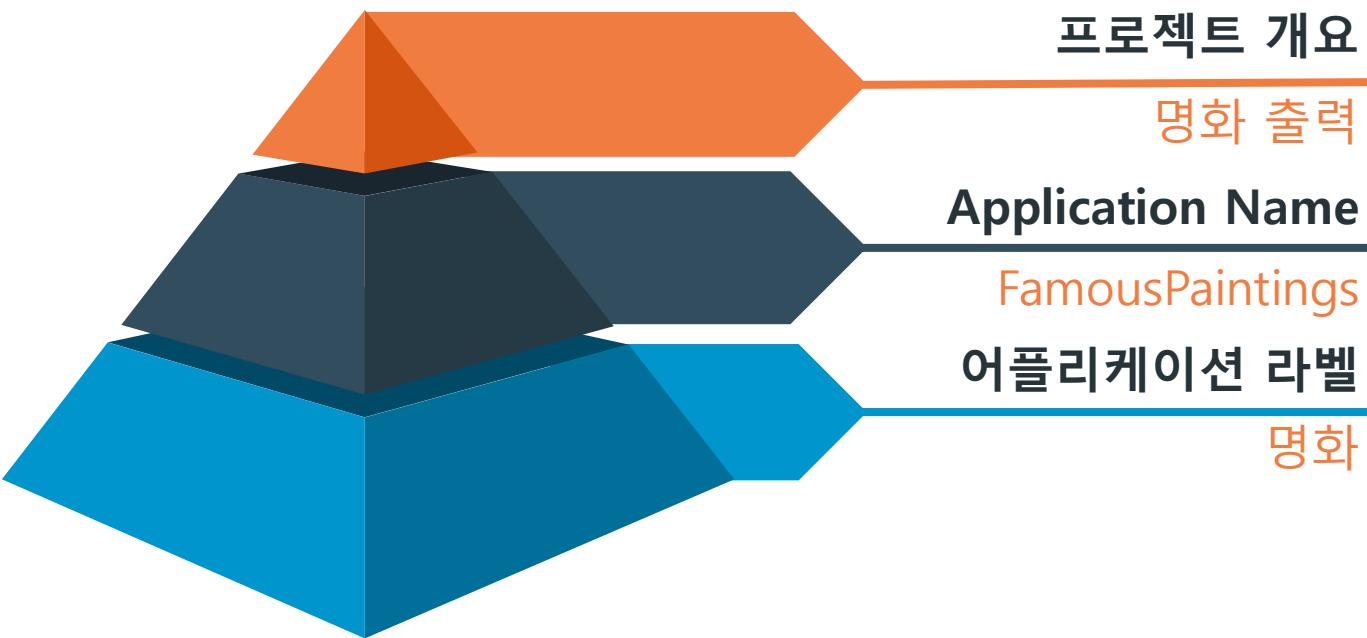
# 이지미 출력 원리

5



# Step 0. 프로젝트 개요

6



빈센트 반 고흐의 대표작 중 하나로, 그가 고갱과 다툼 뒤 자신의 귀를 자른 사건 이후에 생레미의 요양원에 있을 때 그린 것이다.

고흐에게 밤하늘은 무한함을 표현하는 대상이었다. 비연속적이고 동적인 터치로 그려진 하늘은 굽이치는 두꺼운 웃놀림으로 사이프러스와 연결되고, 그 아래 마을은 대조적으로 고요하고 평온한 상태를 보여준다.

그는 병실 밖으로 내다보이는 밤 풍경을 상상과 결합시켜 그렸는데, 이는 자연에 대한 반 고흐의 내적이고 주관적인 표현을 구현하고 있다.

# Step 1. 프로젝트 생성

7

절차	내용
① 프로젝트 시작	메뉴에서 'File → New Project' 클릭
② 프로젝트 구성	Application Name: FamousPaintings Company Domain: 사용자계정.example.com(디폴트 사용)
	Project Location: D:/workspace/ktpark/FamousPaintings
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

8

모듈	폴더	소스 파일	편집 내용
manifests		AndroidManifest.xml	
java	com.example.kyungtae.famouspaintings	MainActivity.java	
	drawable	starry_night.png	<ul style="list-style-type: none"><li>명화 이미지</li></ul>
		shape_title.xml	<ul style="list-style-type: none"><li>명화 제목의 출력모양 설계(배경색, 패딩, 모서리)</li></ul>
	layout	activity_main.xml	<ul style="list-style-type: none"><li>명화 화면 구성(제목, 작가, 이미지, 설명)</li></ul>
res	mipmap	ic_launcher.png	
		dimens.xml	
	values	strings.xml	<ul style="list-style-type: none"><li>명화 목록 아이템(제목, 작가, 설명)</li></ul>
		styles.xml	

## 배경 이미지



starry\_night.png (drawable)

## 아이콘 이미지

모양  
shape  
shape rectangle  
색  
color #3061380B  
shape\_title.xml (drawable)

## 앱 라벨

## 시 제목

## 작가

## 시 본문

## 텍스트 자원

string  
app\_name 명화  
title01 Starry Night  
auth01 고흐  
desc01 빈 센트 반 고흐 ...  
strings.xml (values)

## 화면 테마

style  
AppTheme:  
Theme.AppCompat.Light.DarkActionBar  
item  
colorPrimary  
@color/colorPrimary  
styles.xml (values)

## 화면 테마 구성 색상

color  
colorPrimary #3F51B5  
colors.xml (values)

## 화면 레이아웃

LinearLayout  
TextView  
text @string/title01  
background @drawable/shape\_title  
TextView  
text @string/artist01  
ImageView  
src @string/starry\_night  
TextView  
text @string/desc01  
activity\_main.xml (layout)

## 화면 출력 소스

액티비티 제어  
onCreate  
super onCreate()  
setContentView(R.layout.activity\_main)  
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



## 컴파일/빌더 정보

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

(Gradle Scripts)

# Step 2.1 이미지 파일 복사

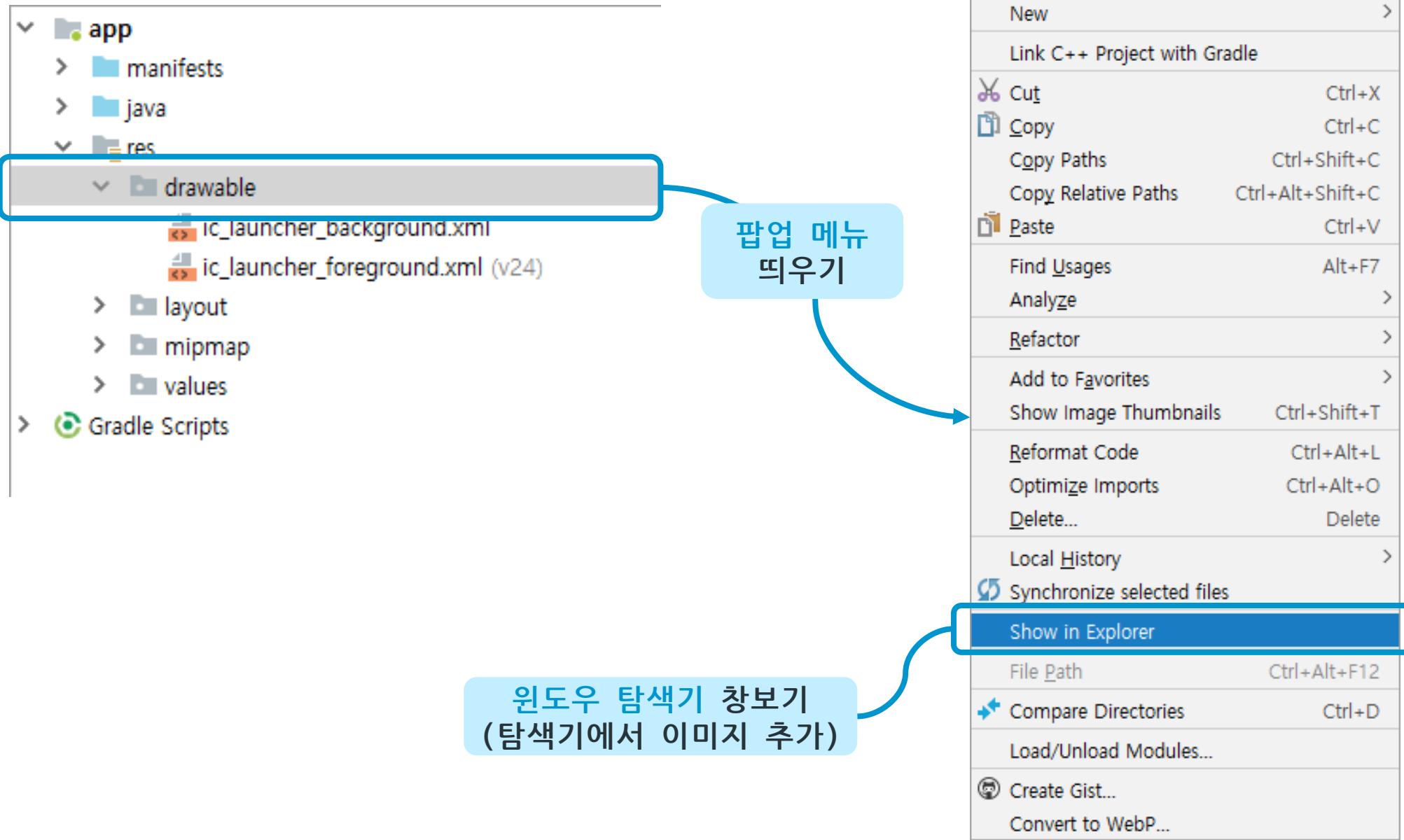
10

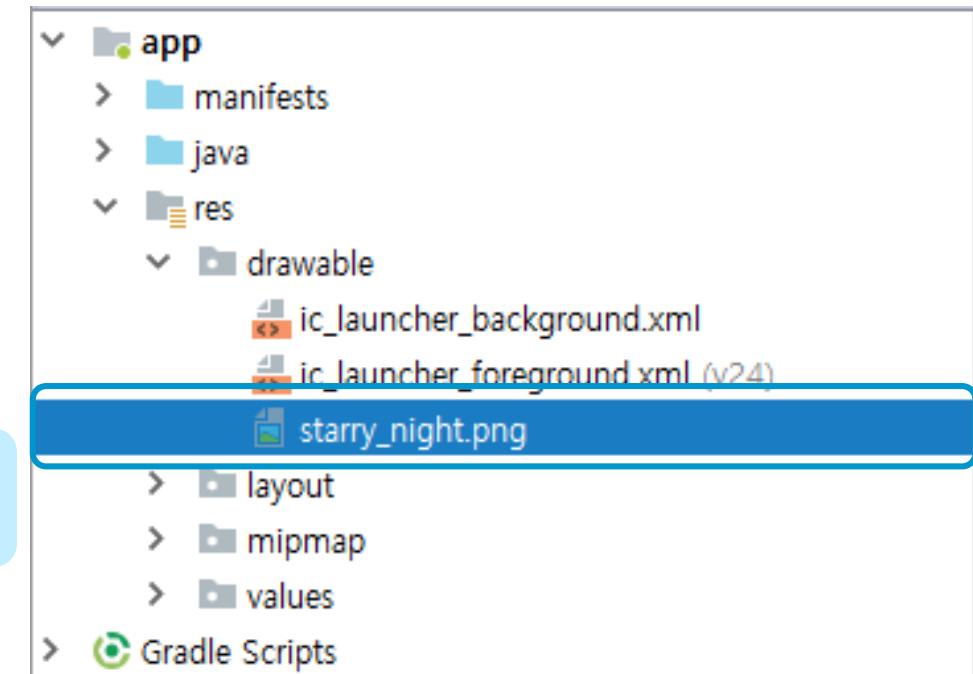
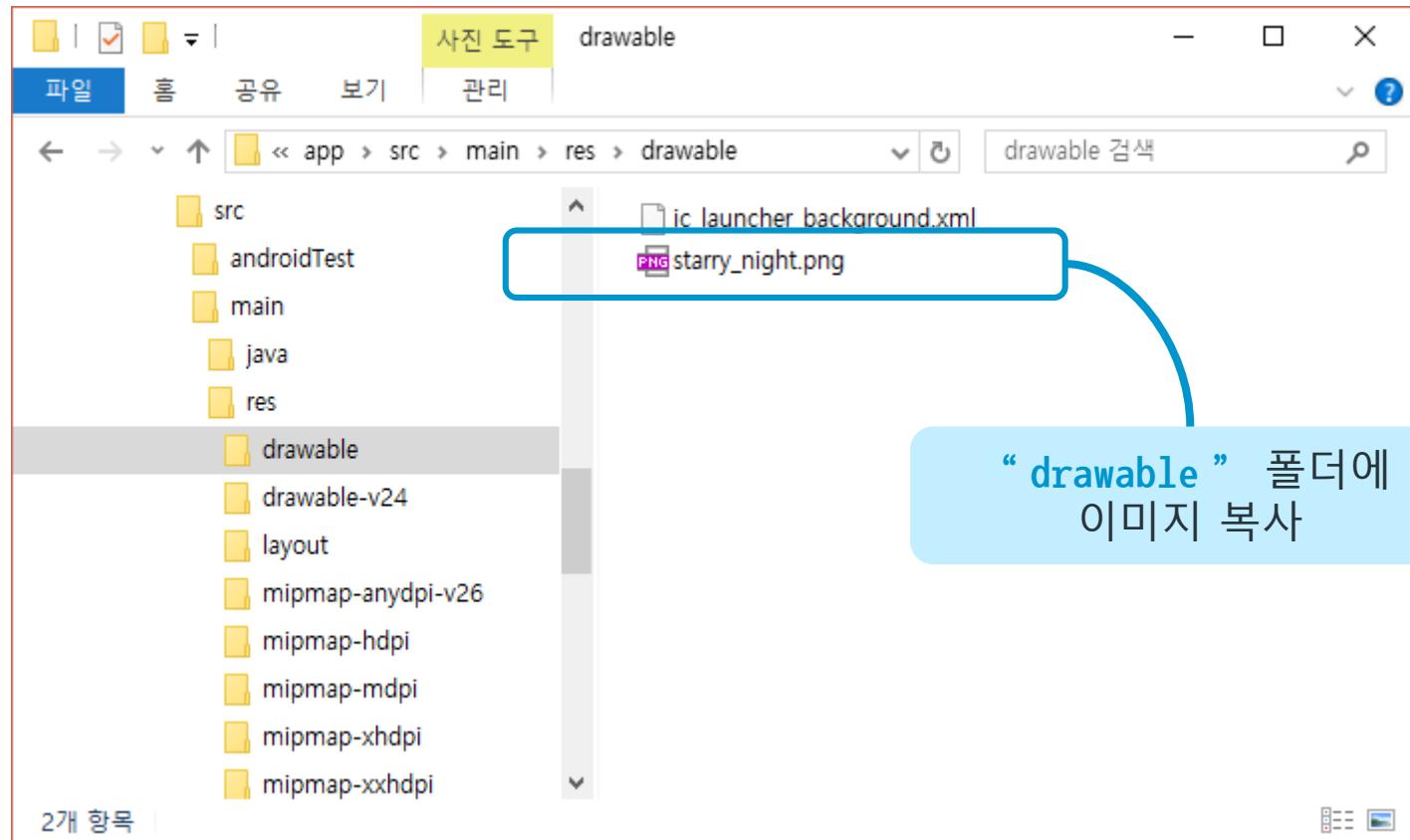
- res 폴더에 있는 **starry\_night.png** 파일을 **drawable** 폴더에 저장

모듈	폴더	소스 파일	이미지
res	drawable	starry_night.png	 A reproduction of Vincent van Gogh's painting "Starry Night". It depicts a dark, swirling night sky filled with numerous small, yellow stars and a single, large, crescent moon. In the foreground, there is a dark, craggy mountain peak on the left and a small town with houses and church spires at the base of the hill on the right.

drawable 폴더에  
이미지 추가하기

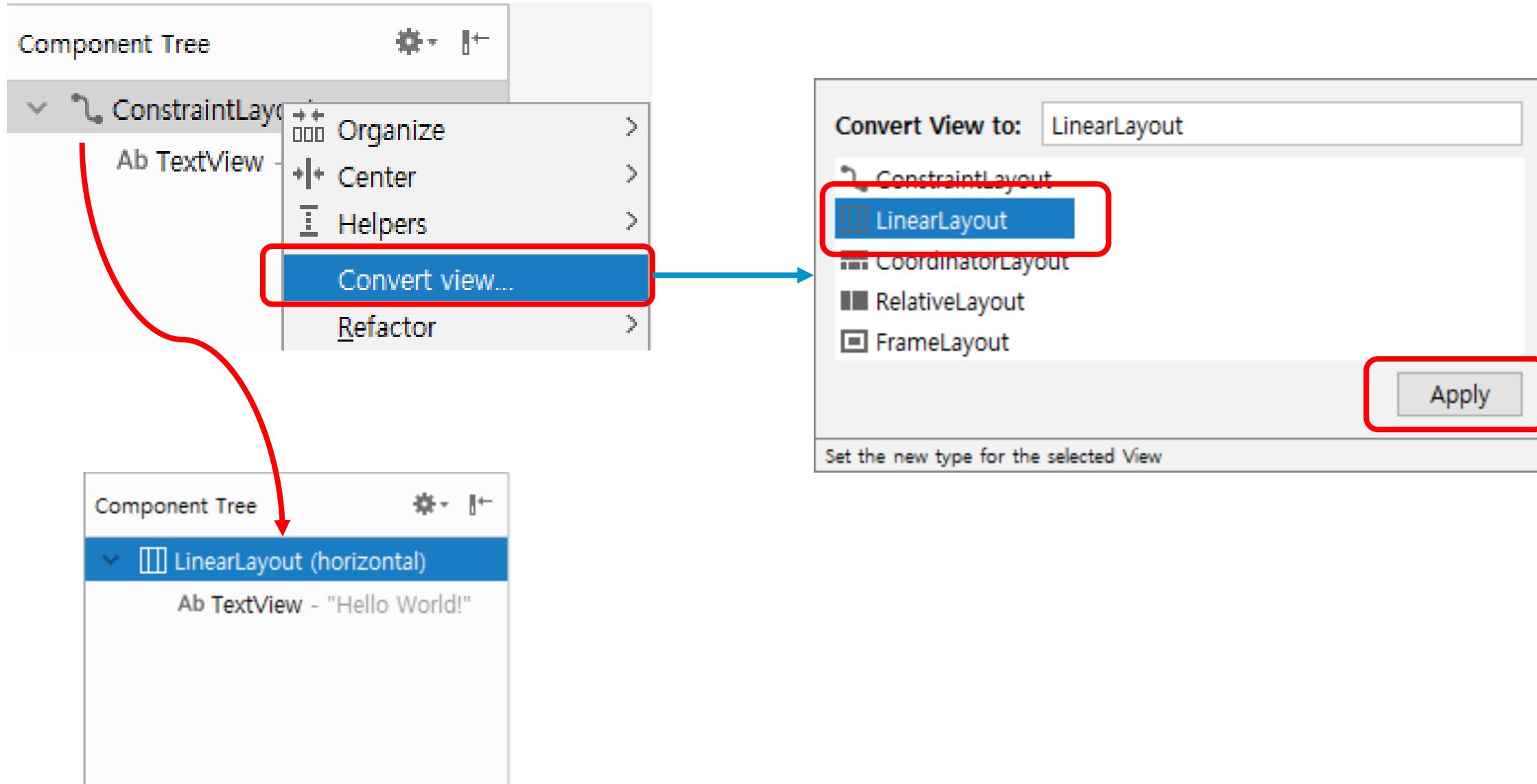
# drawable 폴더에 이미지 추가하기





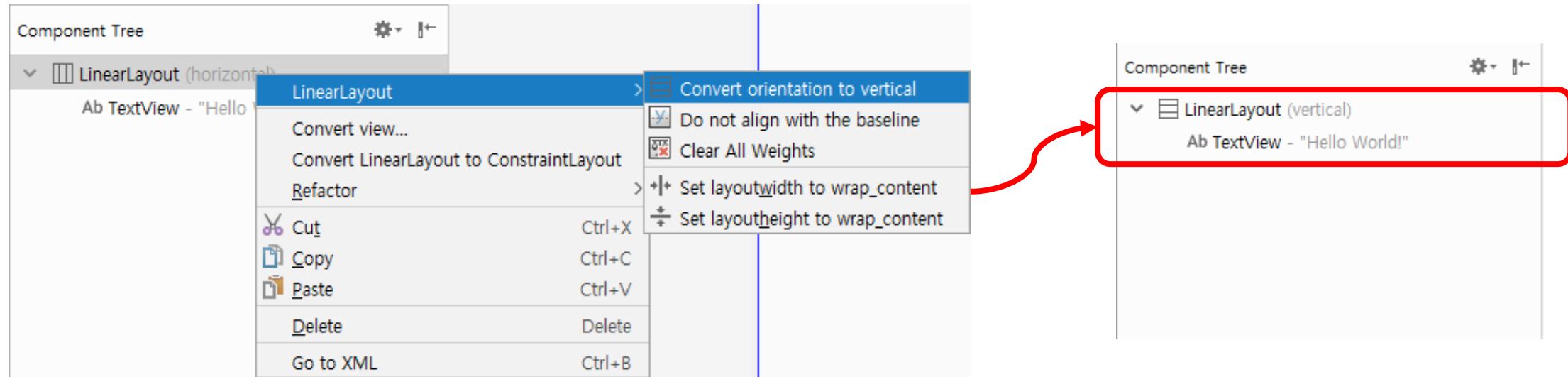
# ConstraintLayout을 LinearLayout로 바꾸기

14



# LinearLayout의 방향을 Horizontal → Vertical로 변경하기

15

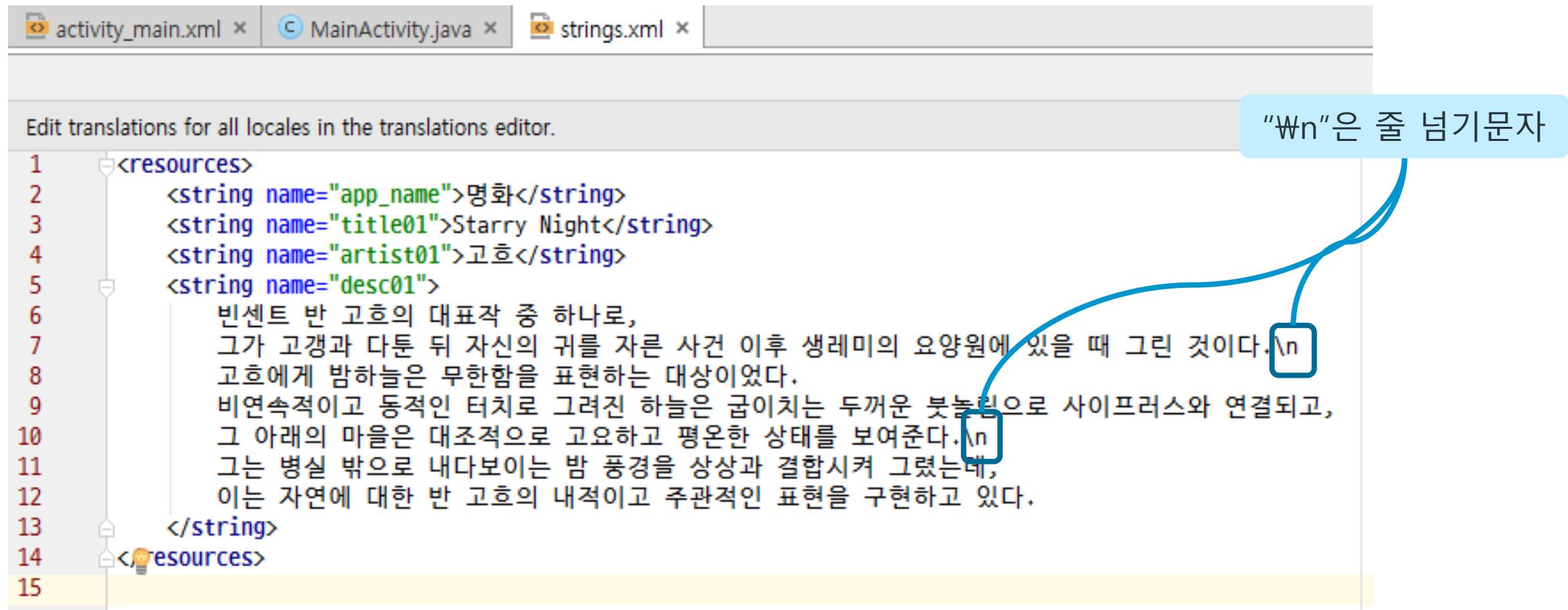


```
activity_main.xml MainActivity.java
```

```
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>
```

# Step 2.2 텍스트 자원의 편집

- strings.xml



activity\_main.xml x MainActivity.java x strings.xml x

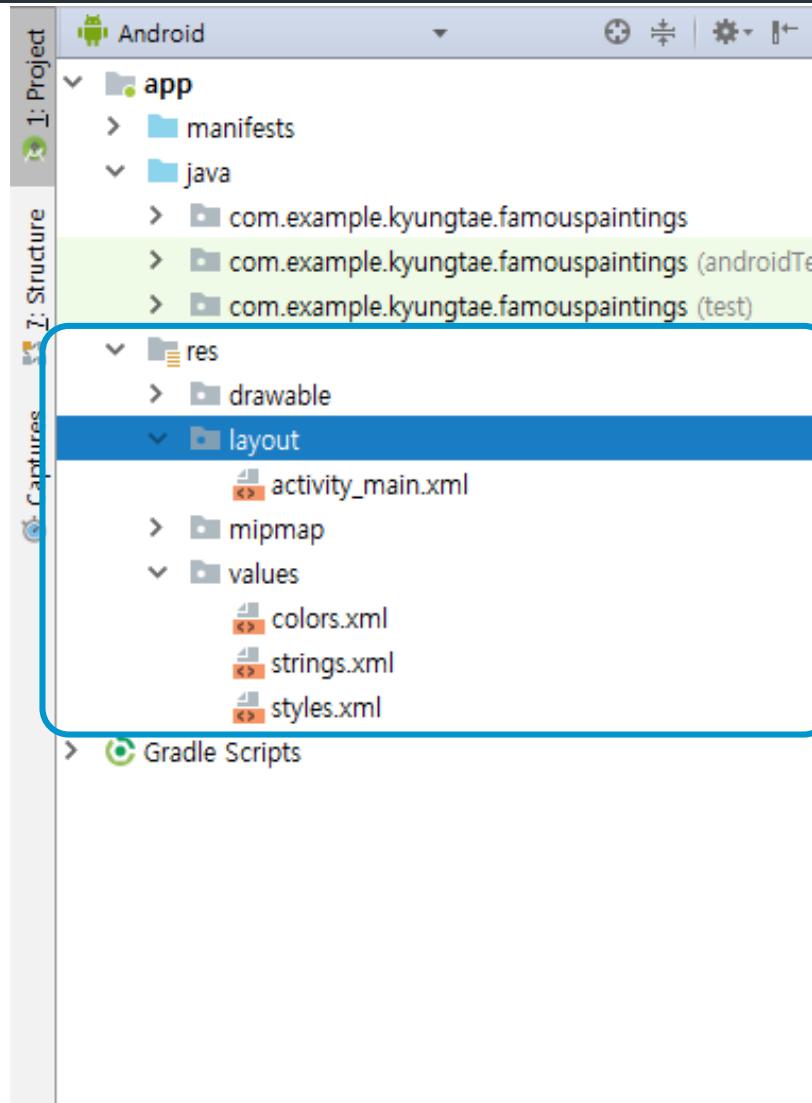
Edit translations for all locales in the translations editor.

```
1 <resources>
2     <string name="app_name">명화</string>
3     <string name="title01">Starry Night</string>
4     <string name="artist01">고흐</string>
5     <string name="desc01">
6         빈센트 반 고흐의 대표작 중 하나로,
7         그가 고갱과 다툼 뒤 자신의 귀를 자른 사건 이후 생레미의 요양원에 있을 때 그린 것이다.\n
8         고흐에게 밤하늘은 무한함을 표현하는 대상이었다.
9         비연속적이고 동적인 터치로 그려진 하늘은 굽이치는 두꺼운 붓놀림으로 사이프러스와 연결되고,
10        그 아래의 마을은 대조적으로 고요하고 평온한 상태를 보여준다.\n
11        그는 병실 밖으로 내다보이는 밤 풍경을 상상과 결합시켜 그렸는데,
12        이는 자연에 대한 반 고흐의 내적이고 주관적인 표현을 구현하고 있다.
13    </string>
14 </resources>
```

"\n"은 줄 넘기문자

# Drawable Resource

18



모듈에 사용되는 리소스 파일 관리  
(화면 레이아웃 설계, 텍스트와 이미지 리소스 등)

# Drawable Resource

19

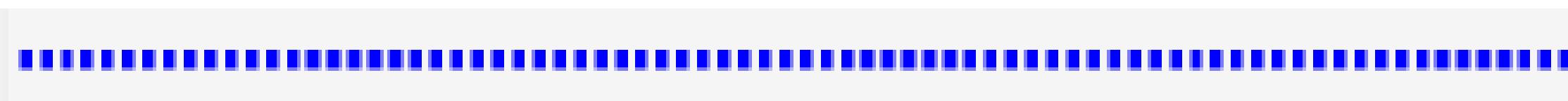
- 화면에 출력하는 그래픽을 위한 일반적인 개념으로 여러 유형이 존재한다.
- **Shape Drawable**은 색상, 테두리, 그라데이션을 넣거나 가장자리를 둘글게 처리하는 등의 수준 높은 버튼이나 배경을 그릴 수 있다.
- Shape Drawable은 다른 drawable과 마찬가지로 `res/drawable/` 내에 XML 파일(`res/drawable/filename.xml`)을 생성하면 된다.
- Java 소스에서는 `R.drawable.filename` 형태로, 액티비티 XML에서는 `@drawable/filename` 형태로 사용

- 예제-선 그리기

20

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

    <stroke
        android:width="5dp"
        android:color="#FF0000FF"
        android:dashWidth="3dp"
        android:dashGap="2dp">
    </stroke>
</shape>
```

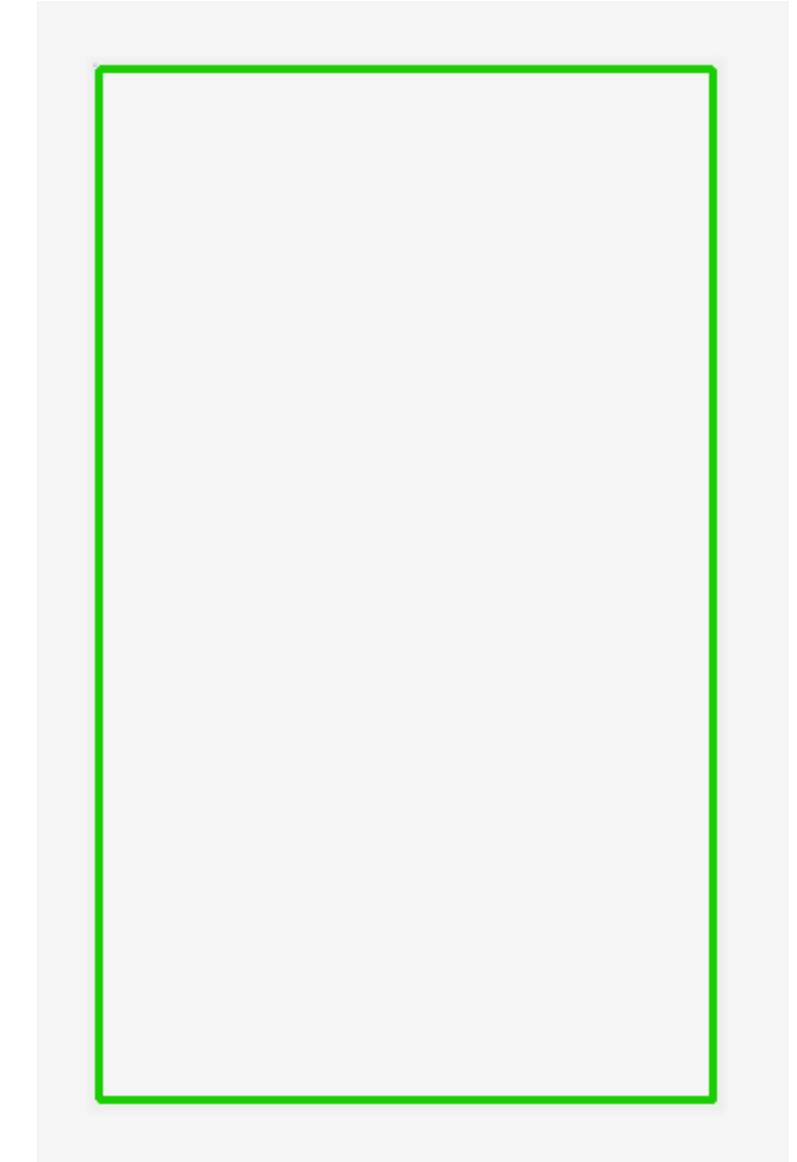


## • 예제-사각형 그리기

21

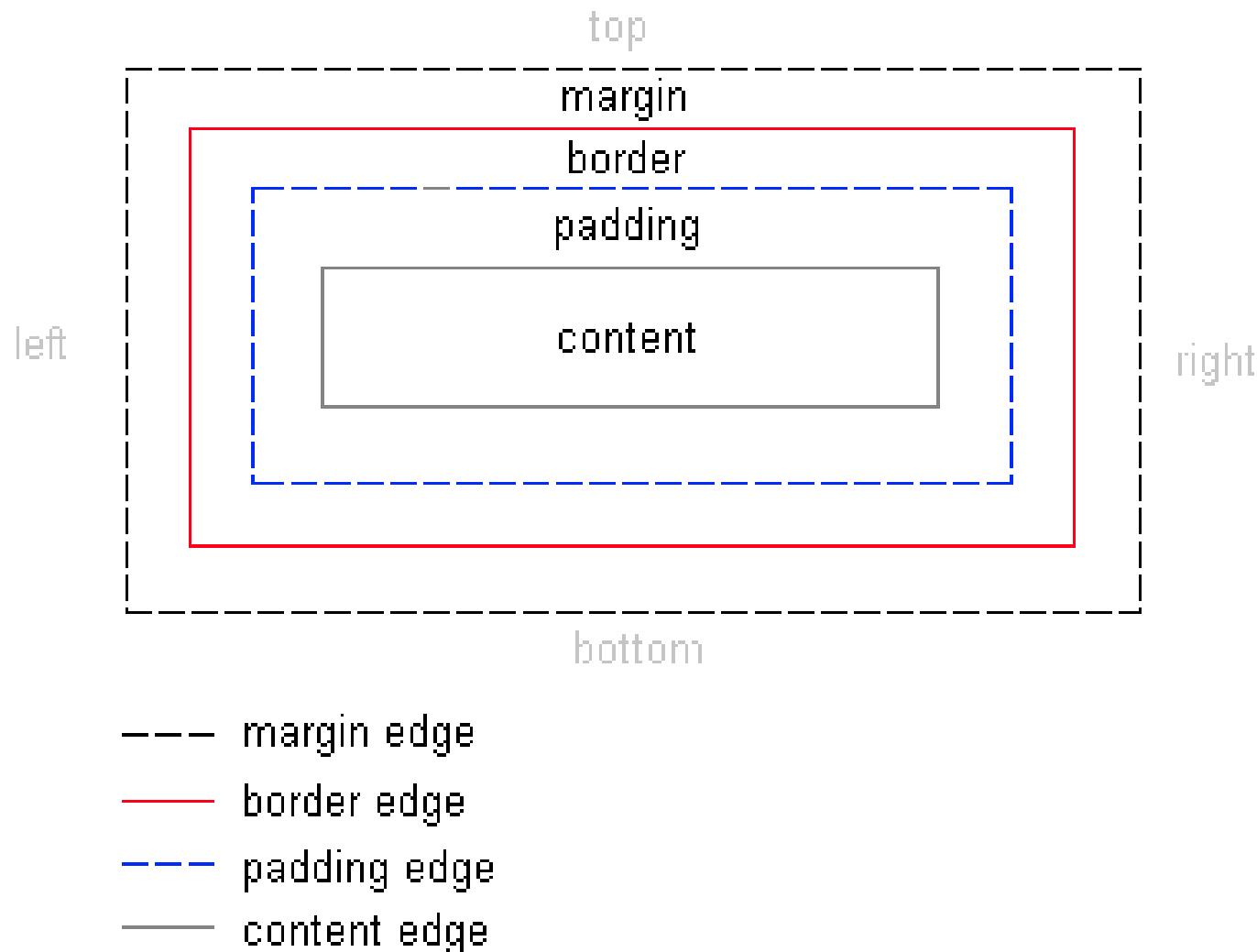
```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle"

    <stroke
        android:width="5dp"
        android:color="#19CD00">
    </stroke>
    <corners
        android:bottomLeftRadius="2dp"
        android:bottomRightRadius="2dp"
        android:topLeftRadius="2dp"
        android:topRightRadius="2dp" />
    <padding android:bottom="5dp"
        android:left="5dp"
        android:right="5dp"
        android:top="5dp"/>
</shape>
```



Padding은 content와 border 사이의 공간

# Padding은 content와 border 사이의 공간

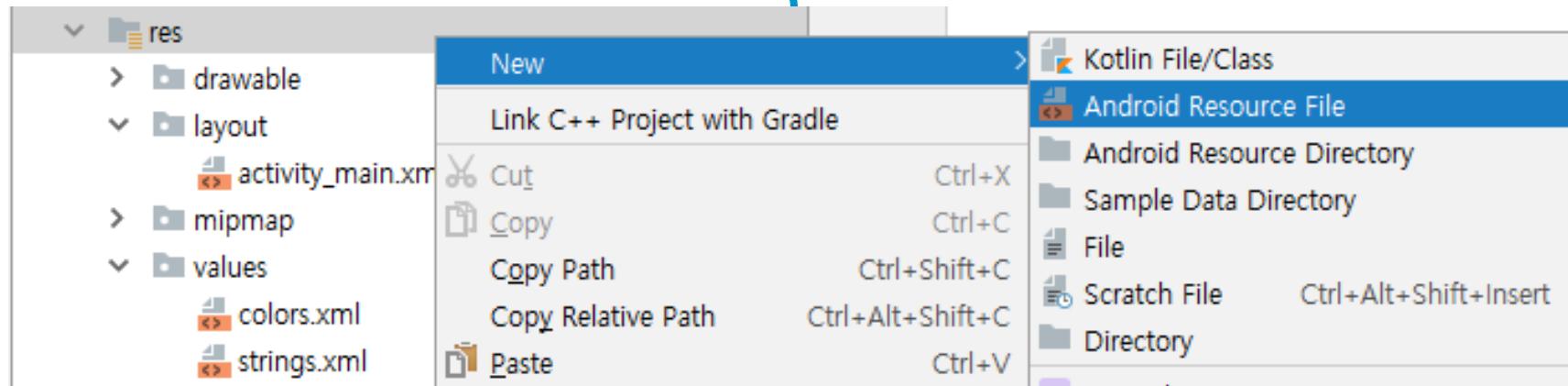


## • shape\_title.xml 생성(res/drawable 폴더)

- 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

**Starry Night**

XML 파일 생성



- Set New Resource File

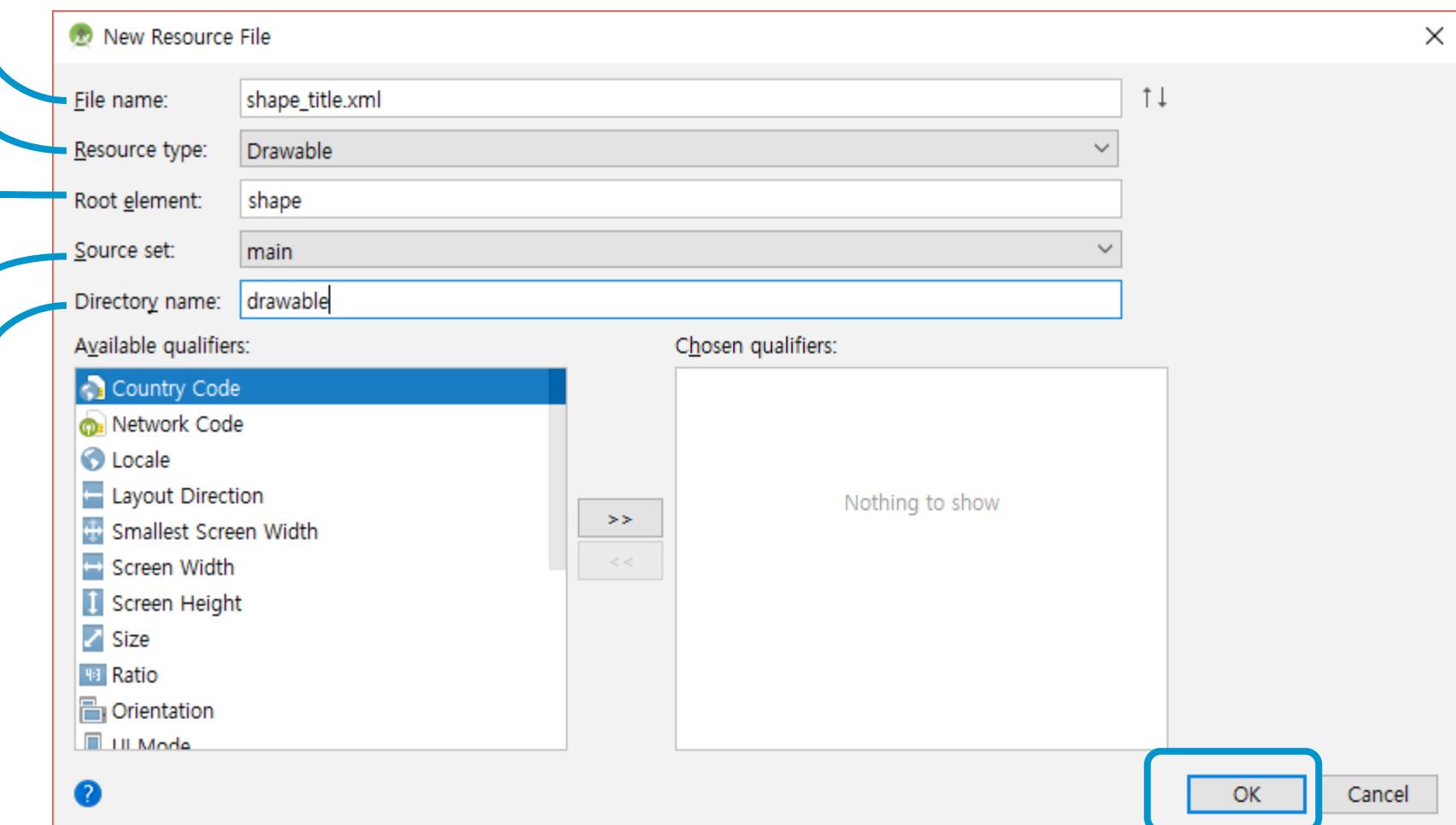
**File name:** shape\_title.xml

**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>
```

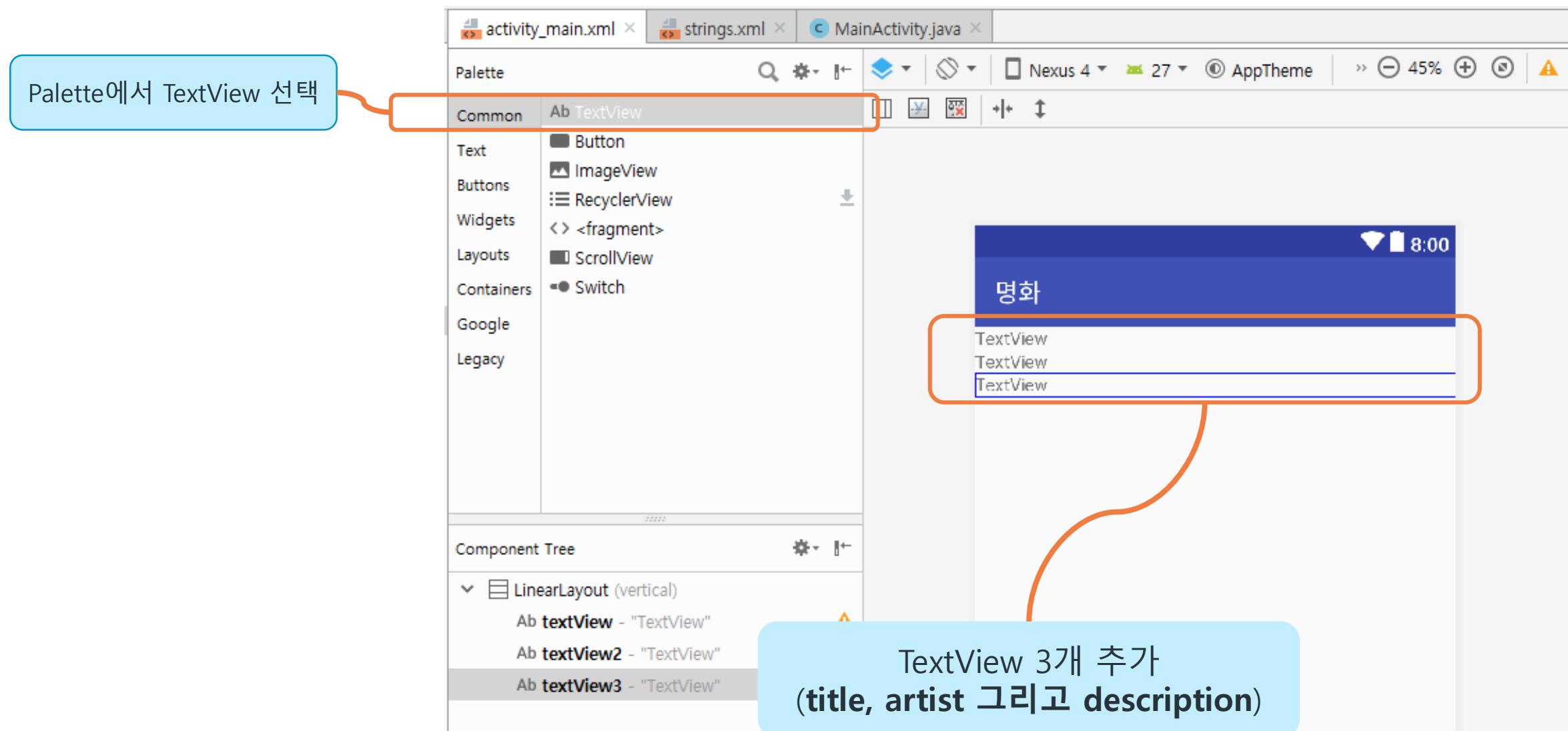
The tabs at the top of the code editor show other files: activity\_main.xml, strings.xml, shape\_title.xml (which is the active tab), and MainActivity.java.

- shape\_title.xml 소스

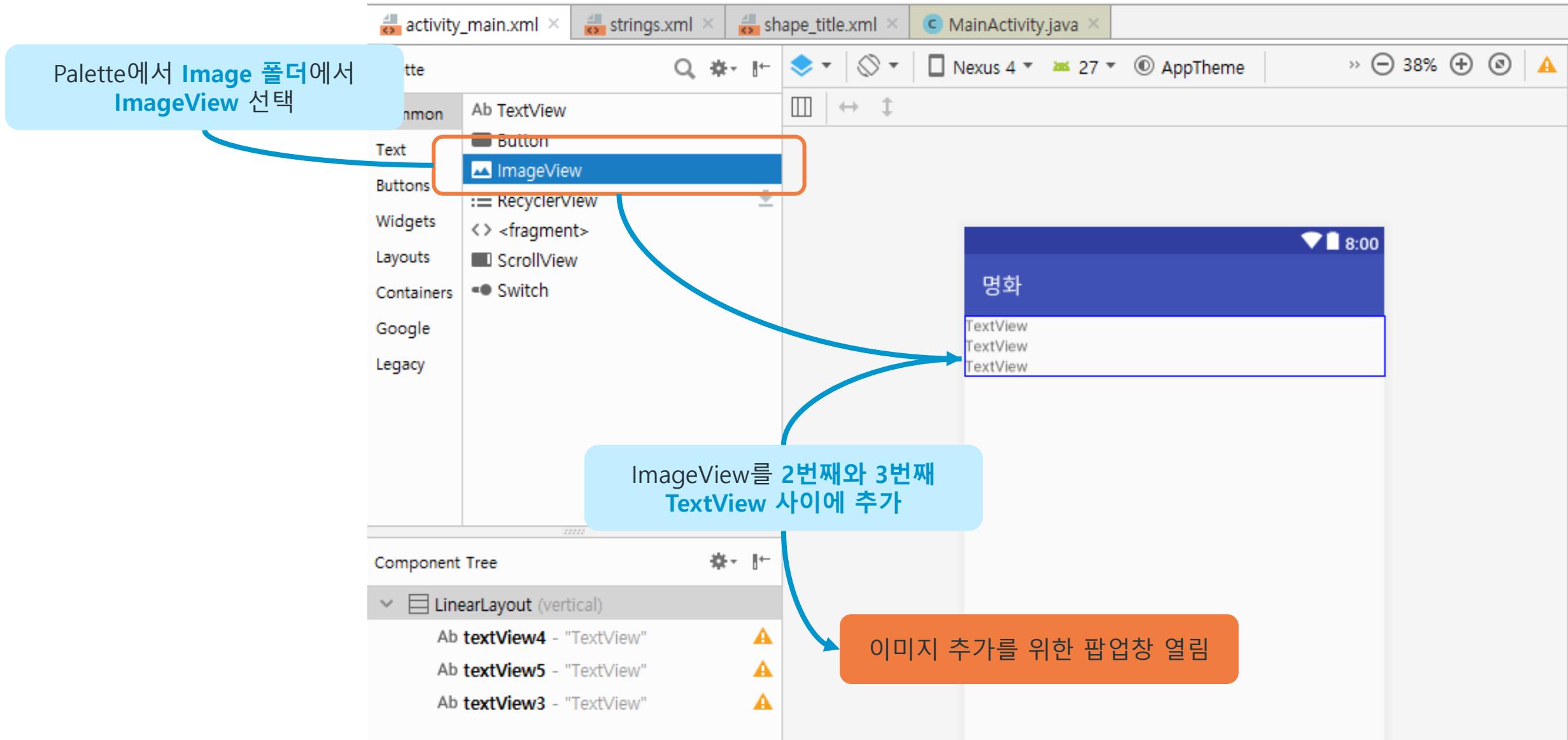
The screenshot shows the Android Studio interface with the `shape_title.xml` file open in the code editor. The code defines a rectangle shape with a solid blue fill and padding on all sides. It also specifies rounded corners with a radius of 5dp. Annotations explain each part of the XML:

- Line 3:** `android:shape="rectangle"` - 출력모양을 사각형으로 지정 (Specifies the output shape as a rectangle)
- Line 5:** `<solid android:color="#3061380b"/>` - 출력모양을 내부의 색 (Specifies the output shape's color as the internal color)
- Lines 7-11:** `<padding android:bottom="5dp" android:left="5dp" android:right="5dp" android:top="5dp"></padding>` - 내부 패딩 정보 (Internal padding information)
- Lines 13-14:** `<corners android:radius="5dp"></corners>` - 출력모양 모서리를 둥근 모양으로 지정(반지름은 5dp) (Specify the output shape's corners as rounded (radius is 5dp))

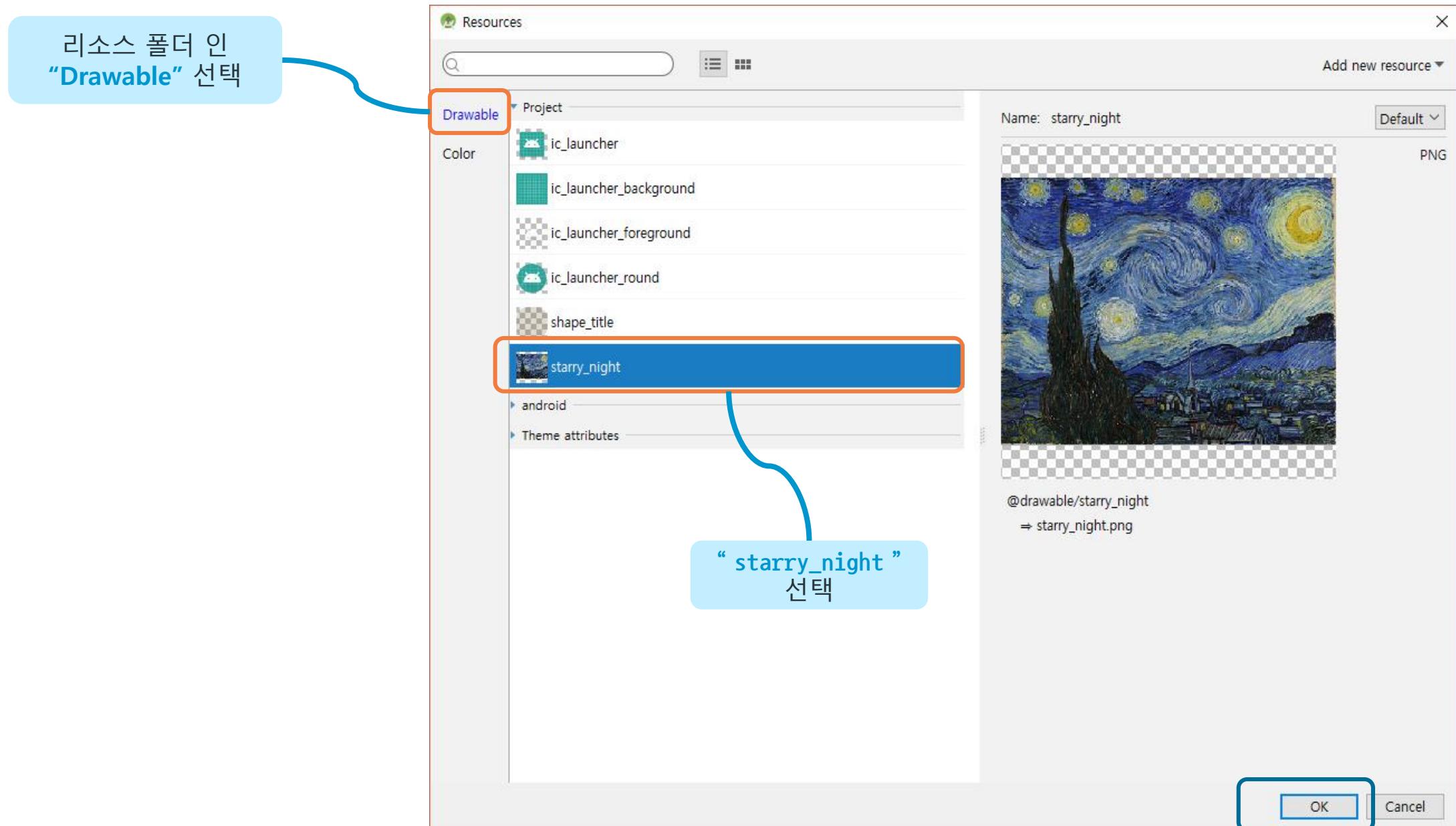
- LinearLayout에 TextView와 ImageView 추가하기



## • LinearLayout에 TextView와 ImageView 추가하기



- ImageView에 이미지 소스 선택 팝업 창



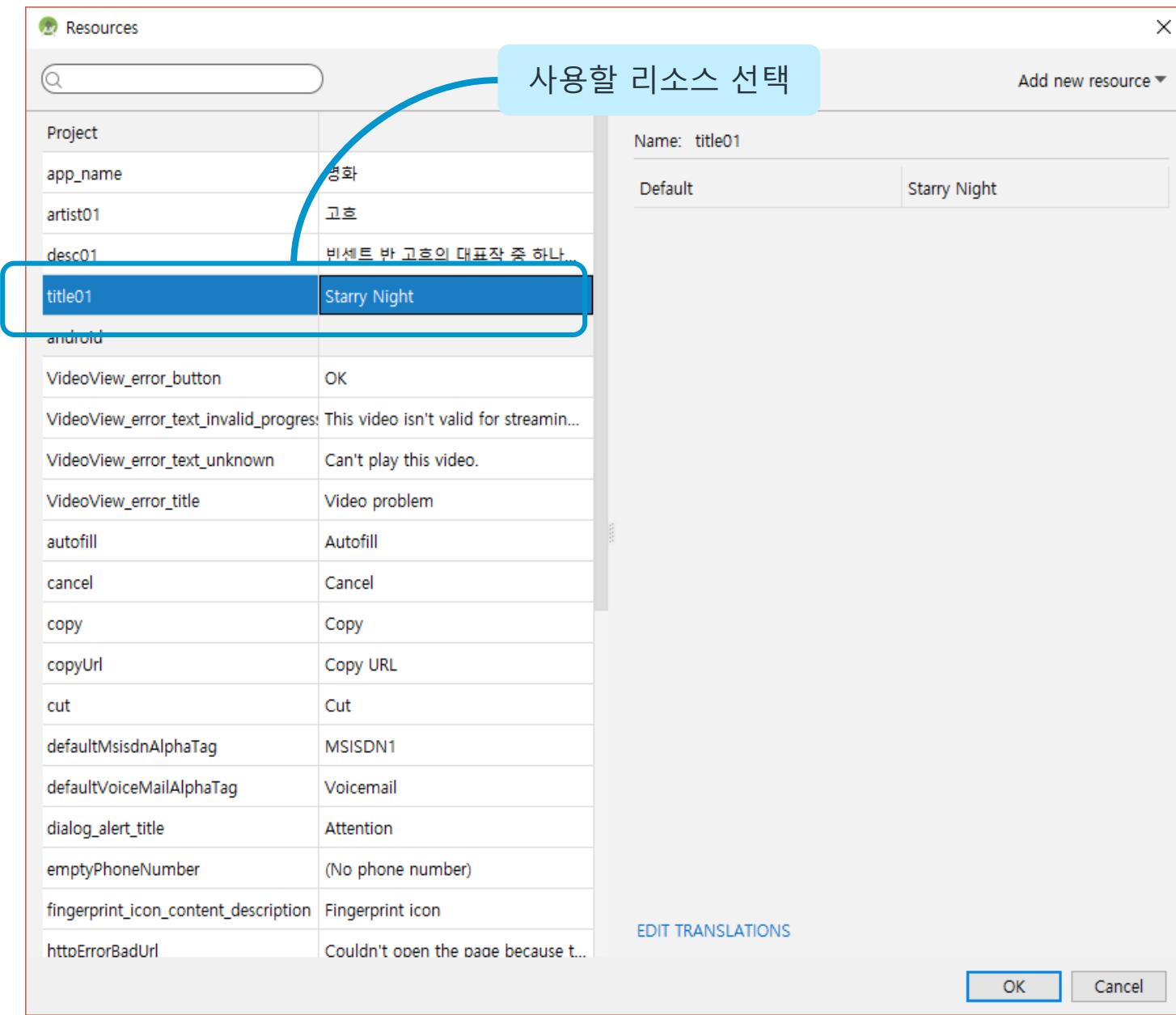
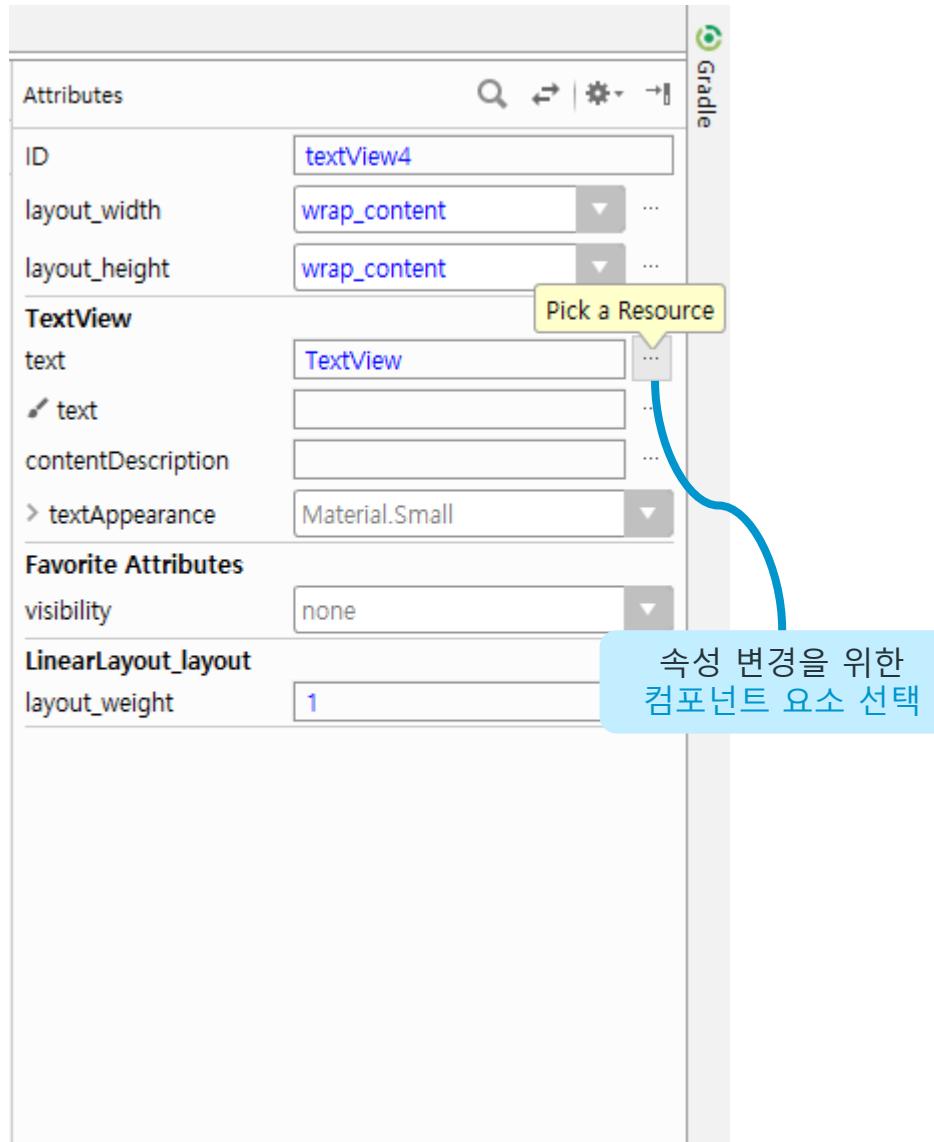
# 완성된 화면



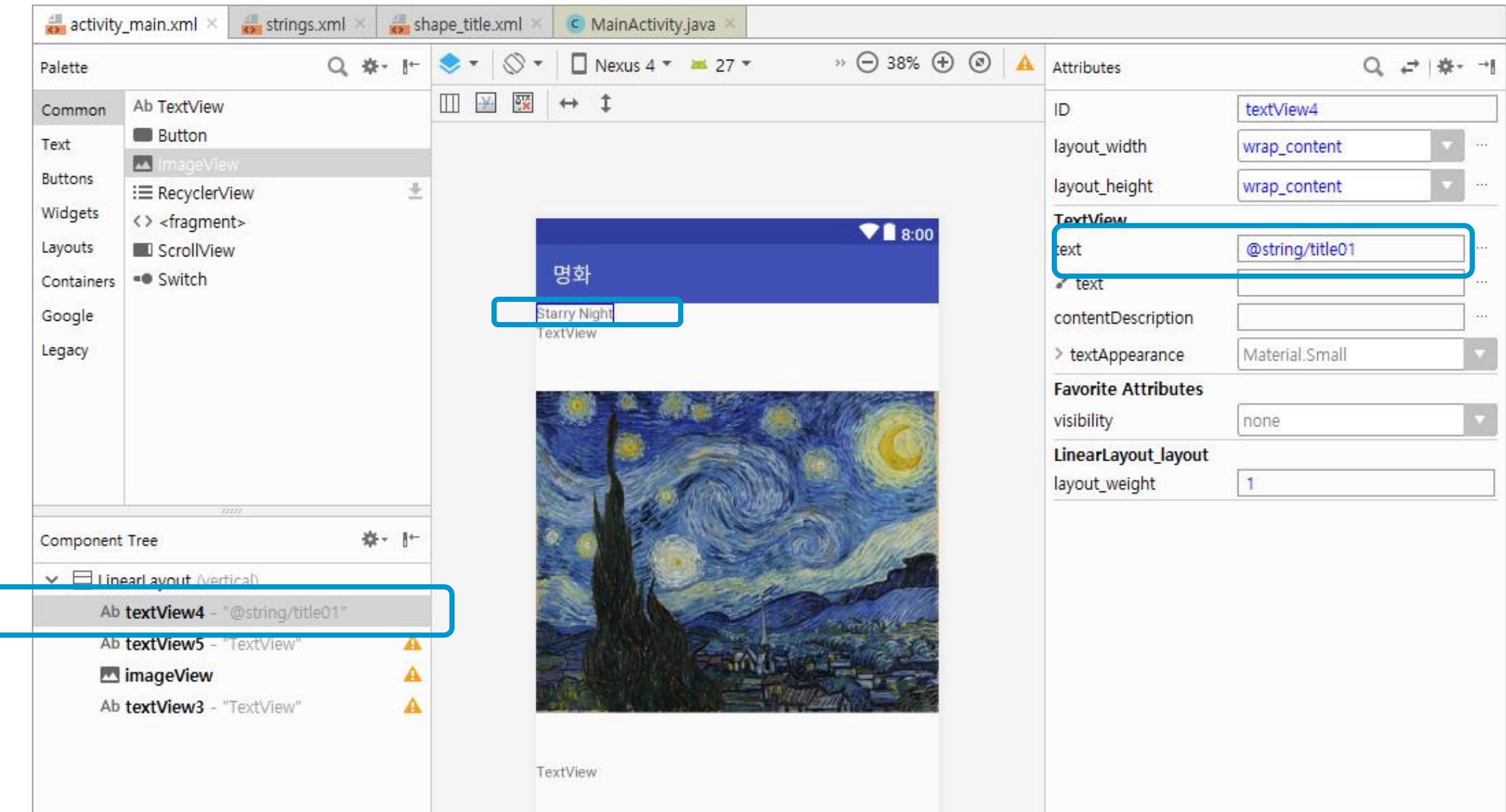
## • Title TextView의 “Text” 속성 변경하기

The screenshot shows the Android Studio interface with the following components visible:

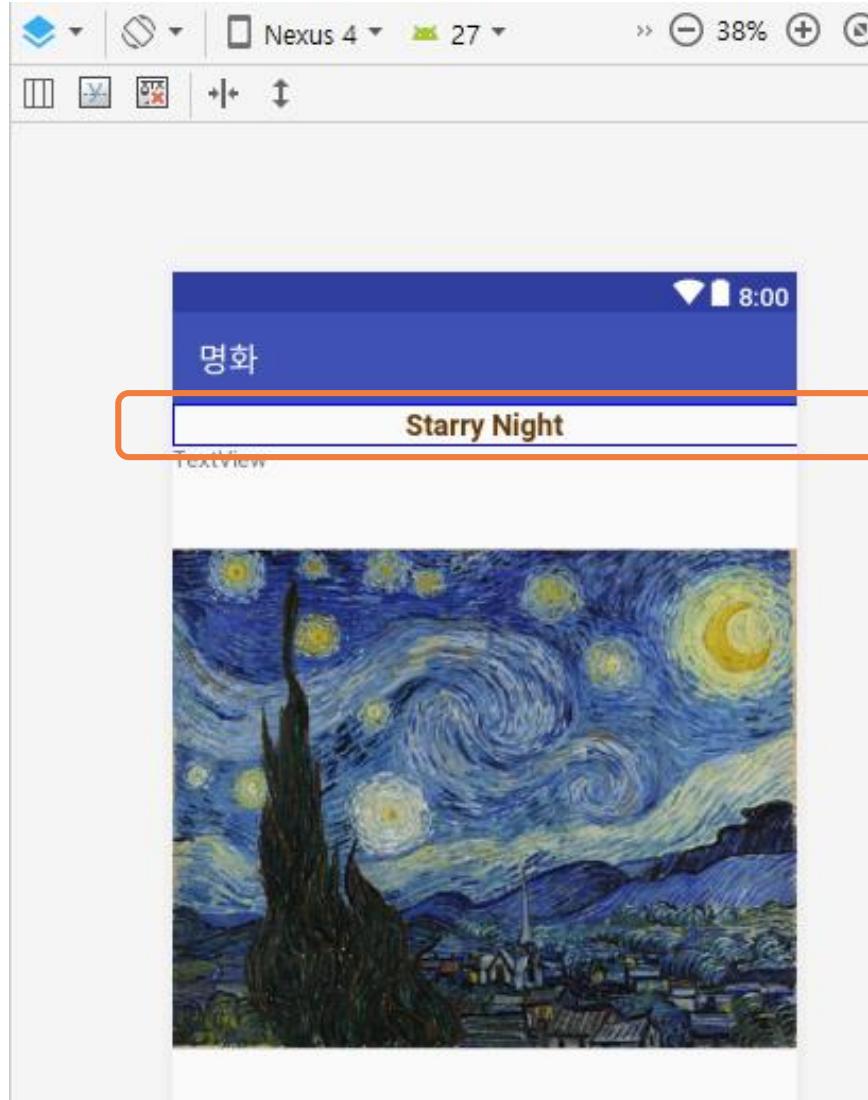
- Top Bar:** activity\_main.xml, strings.xml, shape\_title.xml, MainActivity.java.
- Toolbar:** Search, Settings, Undo, Nexus 4, 27, 38%, +, -.
- Palette:** Common (selected), Text, Buttons, Widgets, Layouts, Containers, Google, Legacy.
- Component Tree:** Linear layout (vertical) containing:
  - Ab textView4 - "TextView"
  - Ab textView5 - "TextView"
  - imageView
  - Ab textView3 - "TextView"
- Main View:** Preview of an Android application screen titled "명화". It displays a painting of "The Starry Night" by Vincent van Gogh. A blue box highlights the "textView" component in the preview.
- Attributes Panel:** Shows the properties for the selected component:
  - ID: textView4
  - layout\_width: wrap\_content
  - layout\_height: wrap\_content
  - TextView:
    - text: Pick a Resource (highlighted with a red box)
    - text (checkbox): checked
    - contentDescription
    - textAppearance: Material.Small
  - Favorite Attributes
  - visibility: none
  - LinearLayout\_layout: layout\_weight: 1
- Annotations:**
  - A blue box labeled "속성 변경을 위한 컴포넌트 요소 선택" (Select component element for attribute modification) points to the Component Tree.
  - A blue callout box labeled "해당 속성을 선택 후 값을 변경→ title01" (Select the attribute and change its value → title01) points to the "text" field in the Attributes panel.



## • title TextView의 “Text” 속성 변경 완료



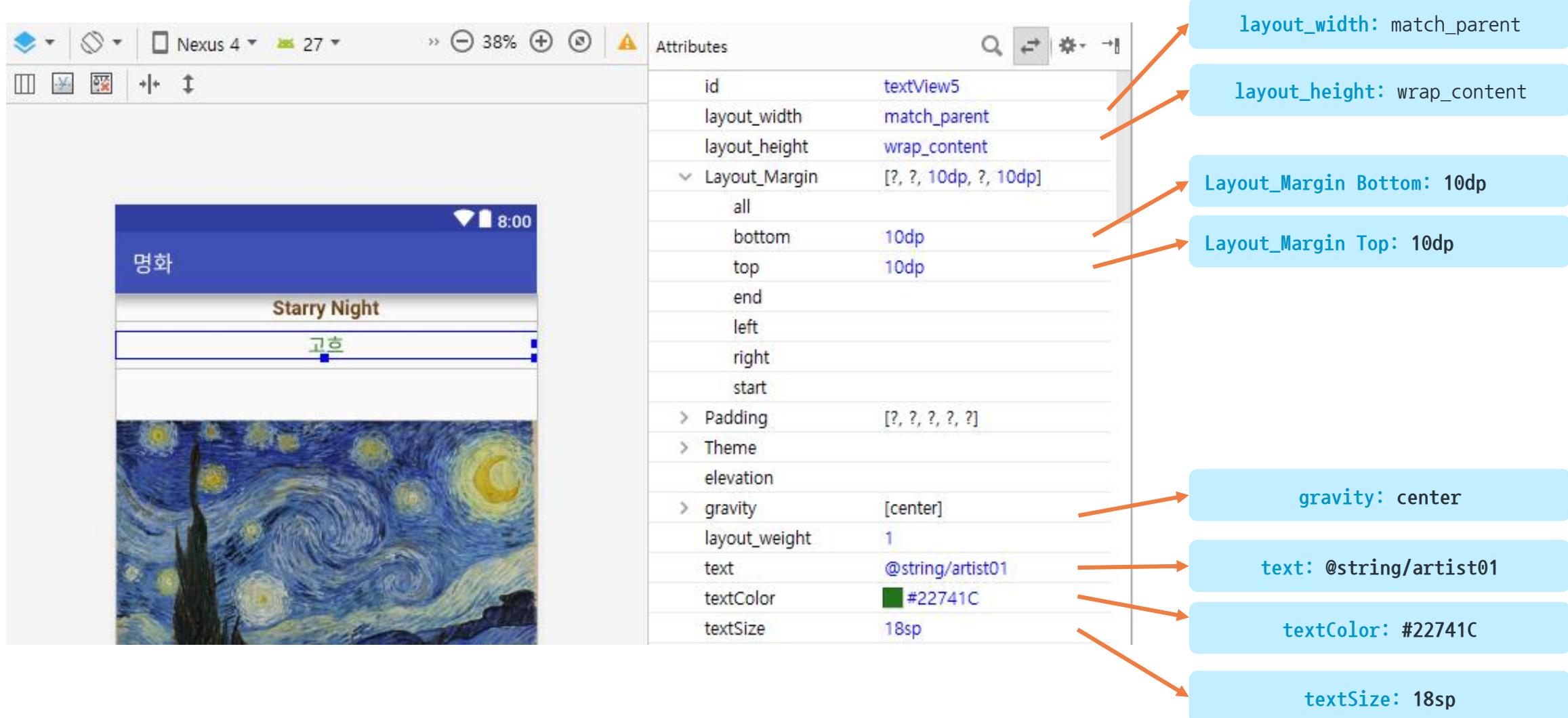
## • title 컴포넌트의 기타 속성 변경하기



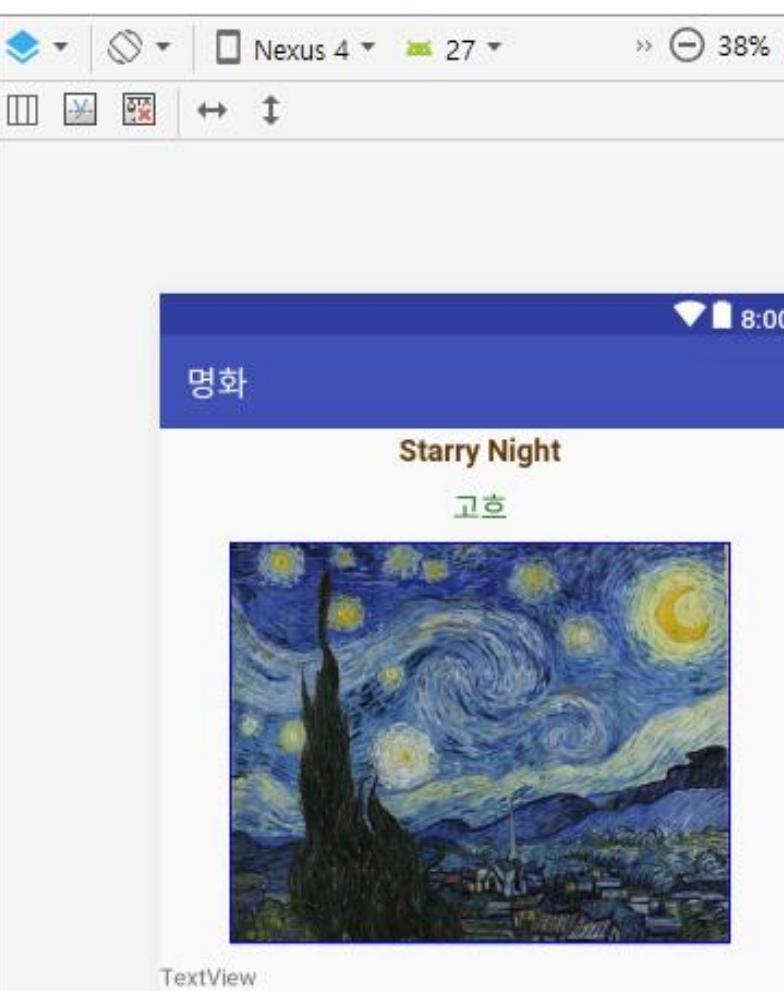
The screenshot shows the Android Studio interface with the XML layout editor open. The layout consists of a title bar with the text "명화" and a subtitle "Starry Night". Below the title bar is a large image of Vincent van Gogh's "Starry Night". The "Starry Night" text is highlighted with an orange border. The XML code on the right side defines the TextView for the subtitle.

Attributes	Value	Description
id	textView4	
layout_width	match_parent	layout_width: match_parent
layout_height	wrap_content	layout_height: wrap_content
> Layout_Margin	[?, ?, ?, ?, ?]	
> Padding	[?, ?, ?, ?, ?]	
> Theme		
elevation		
> gravity	[center]	gravity: center
layout_weight	1	
text	@string/title01	text: @string/title01
textColor	#613808	textColor: #613808
textSize	18sp	textSize: 18sp
> textStyle	[bold]	textStyle: bold
accessibilityLiveRegion		
accessibilityTraversalAf		
accessibilityTraversalBe		
allowUndo	<input checked="" type="checkbox"/>	
alpha		
> autoLink	[]	
autoSizeMaxTextSize		
autoSizeMinTextSize		
autoSizePresetSizes		
autoSizeStepGranularit		

## • artist TextView의 속성 변경



## • 이미지의 ImageView의 속성 변경



The screenshot shows the Android Studio Layout Editor with a Nexus 4 device configuration. On the left, a preview window displays a blue header bar with the text "명화" and "Starry Night" below it. Below that is a green footer bar with the text "고흐". The main content area contains a painting of "Starry Night" by Vincent van Gogh. At the bottom, there is a TextView with the text "Text View". On the right, the "Attributes" panel lists the properties of the ImageView:

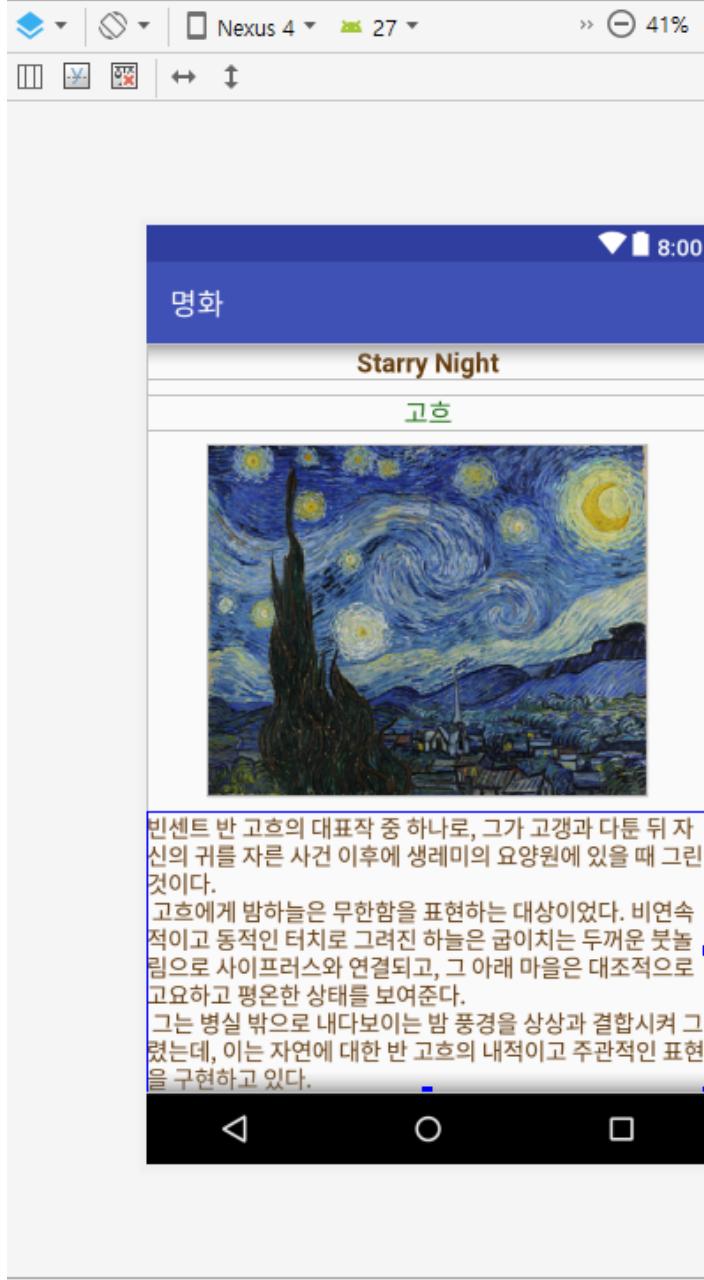
id	imageView
layout_width	wrap_content
layout_height	wrap_content
Layout_Margin	[?, ?, ?, ?, 10dp]
all	
bottom	10dp
end	
left	
right	
start	
top	
> Padding	[?, ?, ?, ?, ?]
> Theme	
elevation	
adjustViewBounds	<input checked="" type="checkbox"/>
> layout_gravity	[center]
layout_weight	1
maxHeight	300dp
maxWidth	300dp
srcCompat	@drawable/starry_night

Annotations with arrows point from specific attributes to their corresponding descriptions:

- layout\_width: wrap\_content
- layout\_height: wrap\_content
- Layout\_Margin Bottom: 10dp
- adjustViewBounds: true(check)
- layout\_gravity: center
- maxHeight: 300dp
- maxWidth: 300dp
- srcCompat: @drawable/starry\_night

※ `adjustViewBounds` 이미지의 종횡비를 맞추기 위해 이미지 뷰의 크기를 적당히 조정할 것인가를 지정

## • desc의 TextView의 속성 변경



The screenshot shows the Android Studio Layout Editor with the following details:

- Device:** Nexus 4
- Orientation:** 27°
- Zoom:** 41%
- Attributes Panel:** Shows the properties for the selected TextView (textView3). The properties are listed below:

Property	Value
id	textView3
layout_width	wrap_content
layout_height	wrap_content
Layout_Margin	[?, ?, ?, ?, ?]
Padding	[?, ?, ?, ?, ?]
Theme	
elevation	
layout_weight	1
text	@string/desc01
textColor	#61380b
textSize	16sp
accessibilityLiveRegion	
accessibilityTraversalAf	
accessibilityTraversalBe	
allowUndo	<input type="checkbox"/>
alpha	
autoLink	<input type="checkbox"/>
autoSizeMaxTextSize	
autoSizeMinTextSize	
autoSizePresetSizes	
autoSizeStepGranularity	
autoSizeTextType	
autoText	<input type="checkbox"/>
autofillHints	
background	
backgroundTint	
backgroundTintMode	
breakStrategy	
bufferType	
capitalize	
clickable	<input type="checkbox"/>
contentDescription	
contextClickable	<input type="checkbox"/>

Annotations on the right side highlight specific properties:

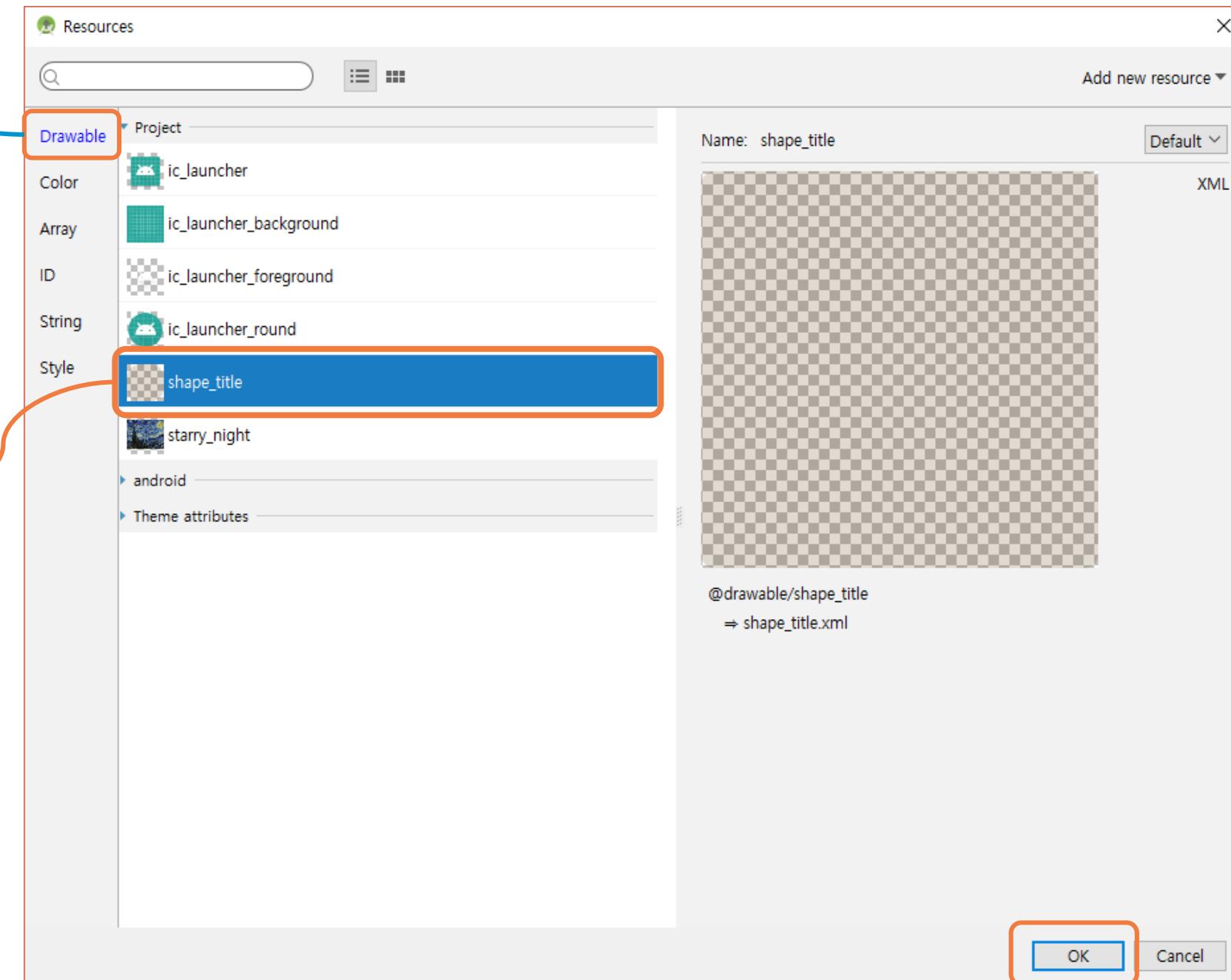
- layout\_width: wrap\_content**
- layout\_height: wrap\_content**
- text: @string/artist01**
- textColor: #61380B**
- textSize: 16sp**

## • `shape_title.xml`을 이용한 제목바의 배경 처리

The screenshot shows the Android Studio interface with the following details:

- Top Bar:** Shows tabs for `activity_main.xml`, `strings.xml`, `shape_title.xml`, and `MainActivity.java`.
- Palette:** On the left, under the "Common" category, `Ab TextView` is selected.
- Component Tree:** Shows a `LinearLayout (vertical)` containing:
  - `Ab textView4 - "@string/title01"` (highlighted with a blue box)
  - `Ab textView5 - "@string/artist01"`
  - `imageView` (highlighted with a yellow box)
  - `Ab textView3 - "@string/desc01"`
- Preview:** Displays the layout with a blue title bar labeled "명화" and "Starry Night". Below it is the painting of "Starry Night" by Vincent van Gogh. The main content area contains descriptive text about the painting.
- Attributes Panel:** On the right, the attributes for `textView4` are listed, including:
  - `id`: `textView4`
  - `layout_width`: `match_parent`
  - `layout_height`: `wrap_content`
  - `gravity`: `[center]`
  - `text`: `@string/title01`
  - `textColor`: `#613808`
  - `textSize`: `18sp`
  - `textStyle`: `[bold]`
  - `background` (highlighted with a blue box)

Resources 종류:  
Drawable



activity\_main.xml × strings.xml × shape\_title.xml × MainActivity.java ×

Palette 🔍 ⚙️ ⌛ Nexus 4 27% 41% + ⚡ Attributes 🔍 ⌛ ⚙️ ⌛

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

Component Tree ⚙️ ⌛

LinearLayout (vertical)  
Ab textView4 - "@string/title01"  
Ab textView5 - "@string/artist01"  
imageView  
Ab textView3 - "@string/desc01" ⚠️

명화  
Starry Night  
고흐  


빈센트 반 고흐의 대표작 중 하나로, 그가 고생과 다툴 뒤 자신의 귀를 자른 사건 이후에 생레미의 요양원에 있을 때 그린 것이다.  
고흐에게 밤하늘은 무한함을 표현하는 대상이었다. 비연속적이고 동적인 터치로 그려진 하늘은 굽이치는 두꺼운 붓놀림으로 사이프러스와 연결되고, 그 아래 마을은 대조적으로 고요하고 평온한 상태를 보여준다.  
그는 병실 밖으로 내다보이는 밤 풍경을 상상과 결합시켜 그렸는데, 이는 자연에 대한 반 고흐의 내적이고 주관적인 표현을 구현하고 있다.

## • 화면 상하좌우에 여백주기

The screenshot shows the Android Studio interface with the layout editor open for the file `activity_main.xml`. The `ImageView` component is selected in the palette, and its properties are displayed in the properties panel. A red box highlights the `Layout_Margin` property, which is set to `[5sp, ?, ?, ?, ?]`. The `layout_margin` value is explicitly set to `5sp`.

**Properties Panel:**

- `id`
- `layout_width`: `match_parent`
- `layout_height`: `match_parent`
- `Constraints`
- `Layout_Margin`: `[5sp, ?, ?, ?, ?]`
  - `layout_margin`: `5sp`
  - `layout_marginBottom`
  - `layout_marginEnd`
  - `layout_marginLeft`
  - `layout_marginRight`
  - `layout_marginStart`
  - `layout_marginTop`
- `Padding`: `[?, ?, ?, ?, ?]`
- `Theme`
- `elevation`
- `context`: `com.example.kyungtae.famouspaintings`
- `orientation`: `vertical`
- `accessibilityLiveRegion`
- `accessibilityTraversalAfter`
- `accessibilityTraversalBefore`
- `actionBarNavMode`
- `addStatesFromChildren`
- `alpha`
- `alwaysDrawnWithCache`
- `animateLayoutChanges`
- `animationCache`
- `background`
- `backgroundTint`
- `backgroundTintMode`
- `baselineAligned`

**Component Tree:**

```
LinearLayout (vertical)
  textView - "@string/title01"
  textView2 - "@string/artist01"
  imageView2
  textView3 - "@string/desc01"
```

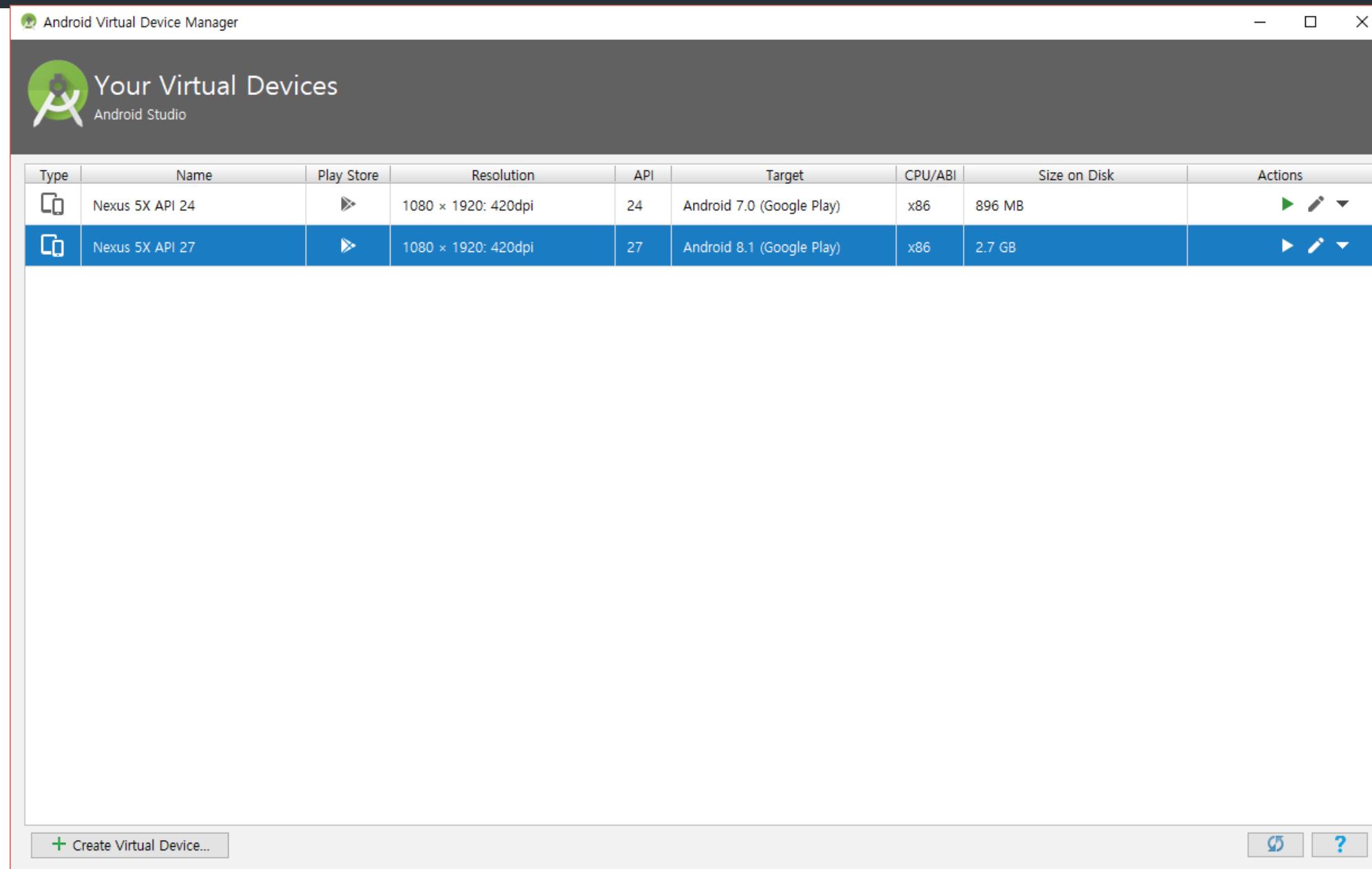
**Properties of Selected ImageView:**

- `id`: `img_starry_night`
- `layout_width`: `match_parent`
- `layout_height`: `wrap_content`
- `src`: `res/drawable/starry_night.jpg`
- `layout_margin`: `5sp`

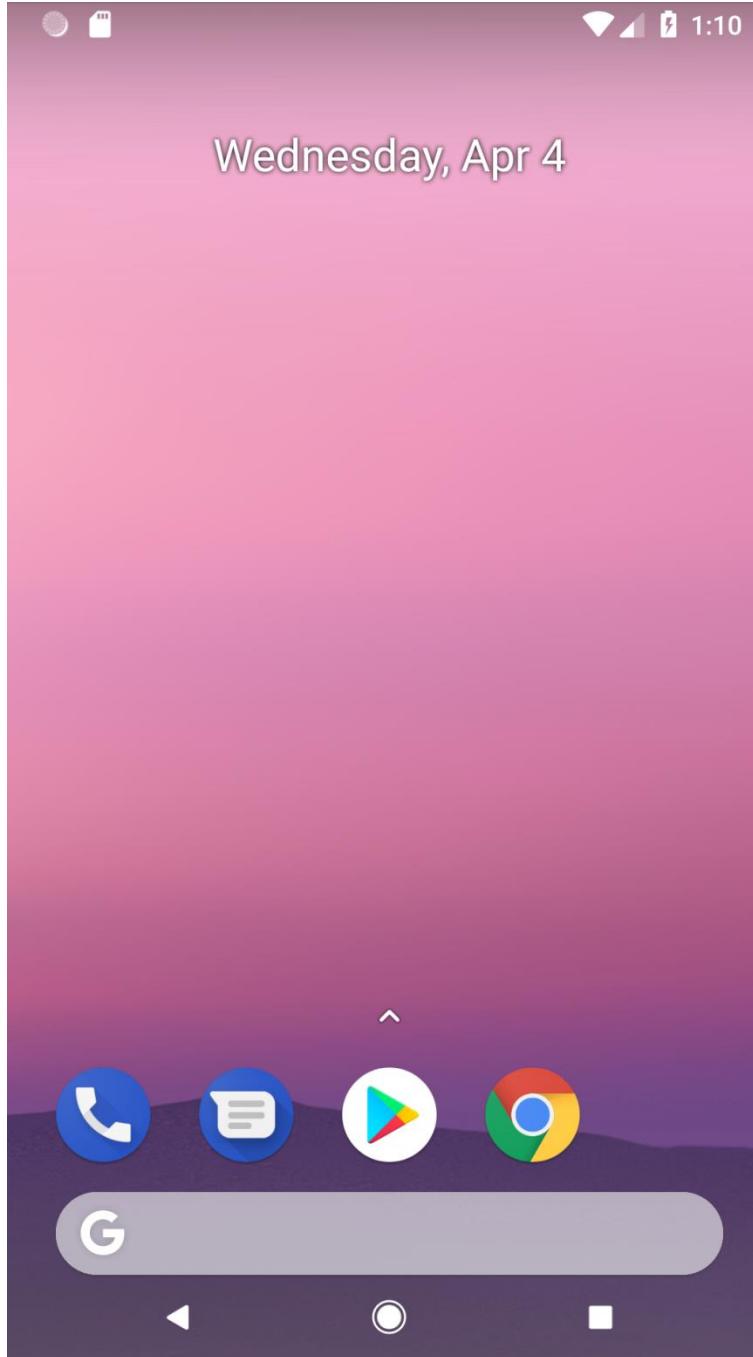
**Preview:** The preview shows a mobile application screen with a title bar, a toolbar with the text "Starry Night", a subtitle "고흐", and the painting "The Starry Night" by Vincent van Gogh.

# Step 3. 프로젝트 실행 – AVD(Android 8.1 Oreo)

48



# Android 8.1 Oreo 초기화면



빈센트 반 고흐의 대표작 중 하나로, 그가 고갱과 다툼 뒤 자신의 귀를 자른 사건 이후에 생레미의 요양원에 있을 때 그린 것이다.

고흐에게 밤하늘은 무한함을 표현하는 대상이었다. 비연속적이고 동적인 터치로 그려진 하늘은 굽이치는 두꺼운 붓놀림으로 사이프러스와 연결되고, 그 아래 마을은 대조적으로 고요하고 평온한 상태를 보여준다.

그는 병실 밖으로 내다보이는 밤 풍경을 상상과 결합시켜 그렸는데, 이는 자연에 대한 반 고흐의 내적이고 주관적인 표현을 구현하고 있다.

# O utputs



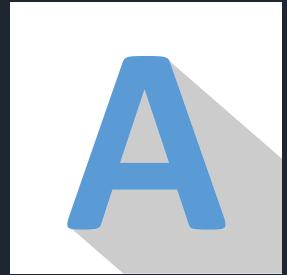
빈센트 반 고흐의 대표작 중 하나로, 그가 고갱과 다툼 뒤 자신의 귀를 자른 사건 이후에 생레미의 요양원에 있을 때 그린 것이다.

고흐에게 밤하늘은 무한함을 표현하는 대상이었다. 비연속적이고 동적인 터치로 그려진 하늘은 굽이치는 두꺼운 붓놀림으로 사이프러스와 연결되고, 그 아래 마을은 대조적으로 고요하고 평온한 상태를 보여준다.

그는 병실 밖으로 내다보이는 밤 풍경을 상상과 결합시켜 그렸는데, 이는 자연에 대한 반 고흐의 내적이고 주관적인 표현을 구현하고 있다.



&



question  
&  
answer

