

# 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)

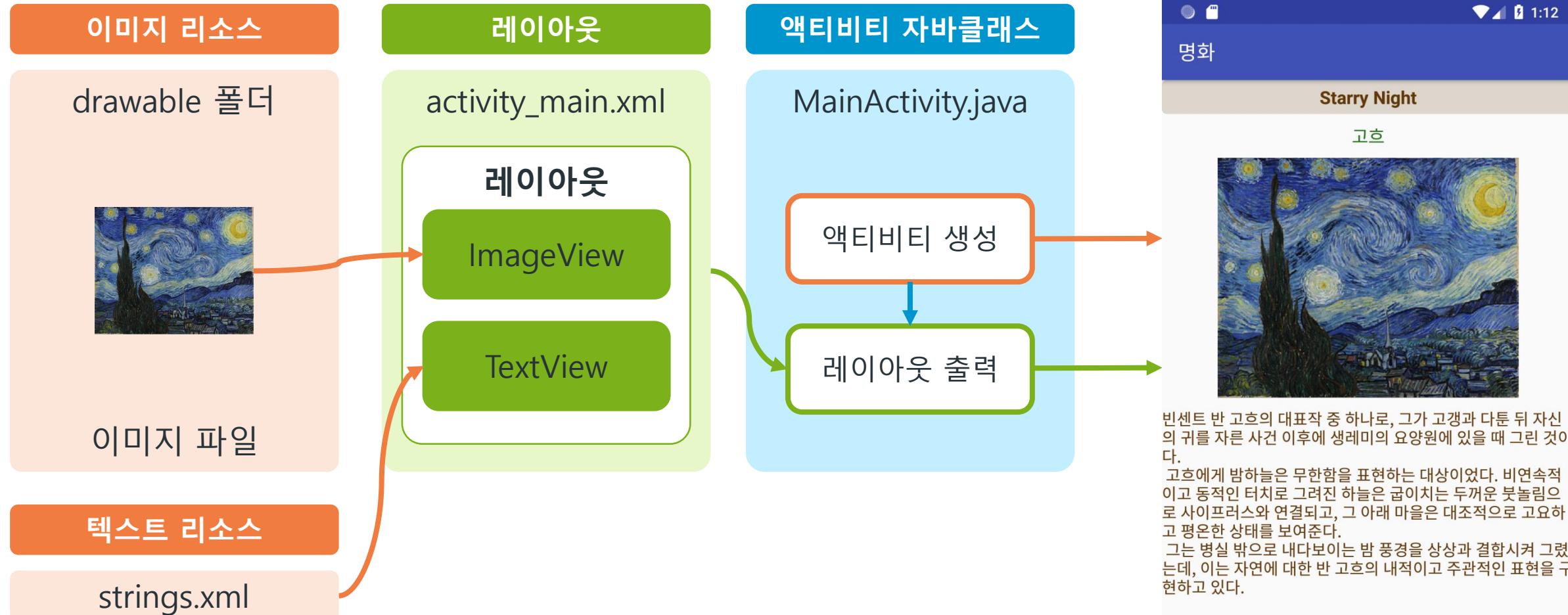


(b) 음식 배달 메뉴(배달의민족)

• 이미지 중심 앱의 예

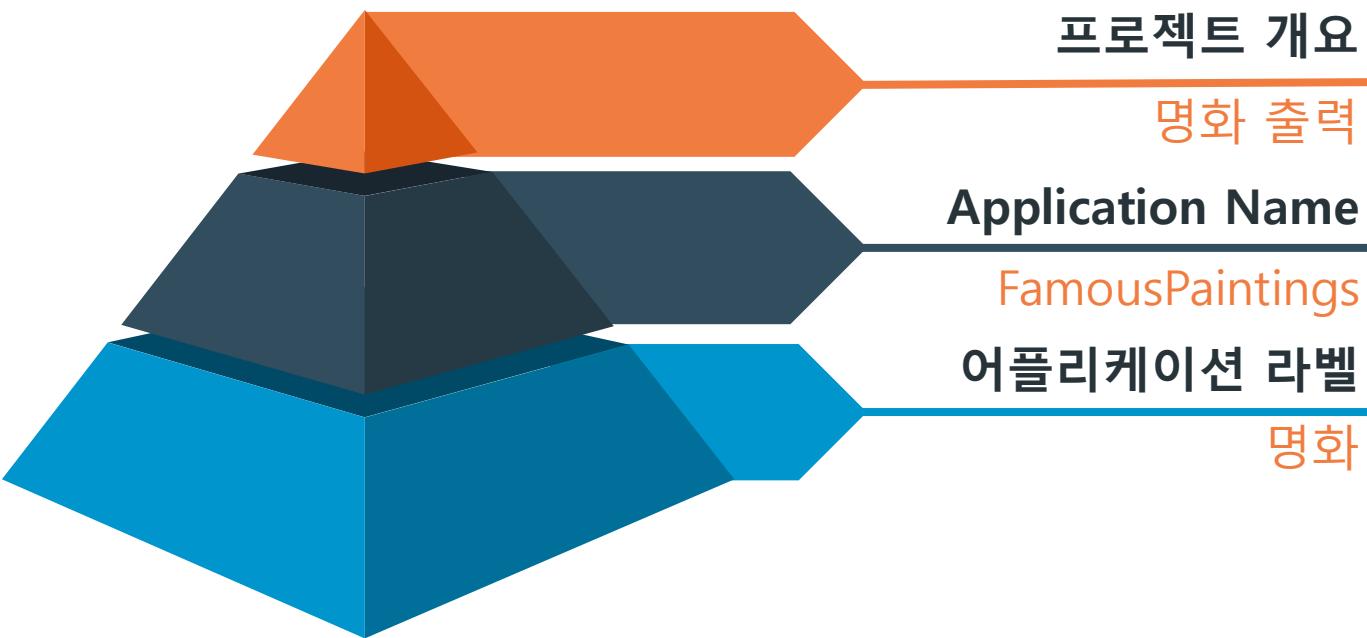
# 이지미 출력 원리

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 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.

# 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	
res	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>
res	layout	activity_main.xml	<ul style="list-style-type: none"><li>명화 화면 구성(제목, 작가, 이미지, 설명)</li></ul>
	mipmap	ic_launcher.png	
	values	dimens.xml	
		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  
autho01 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

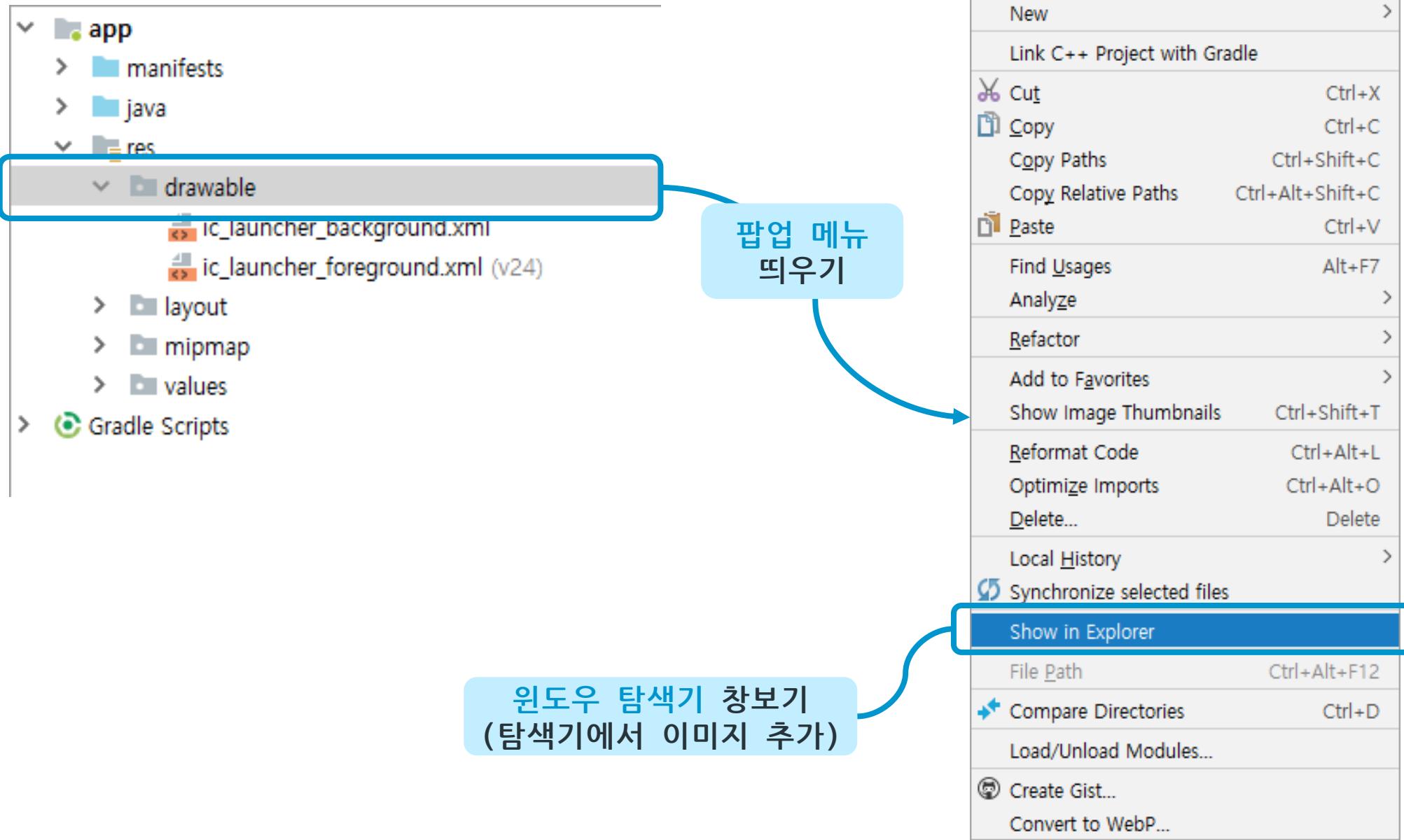
10

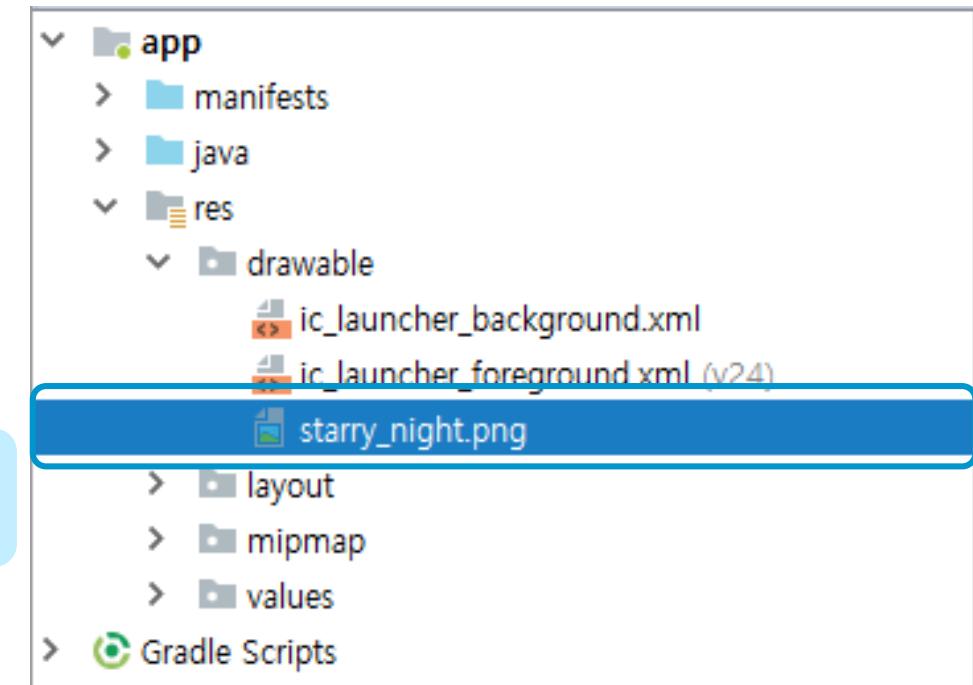
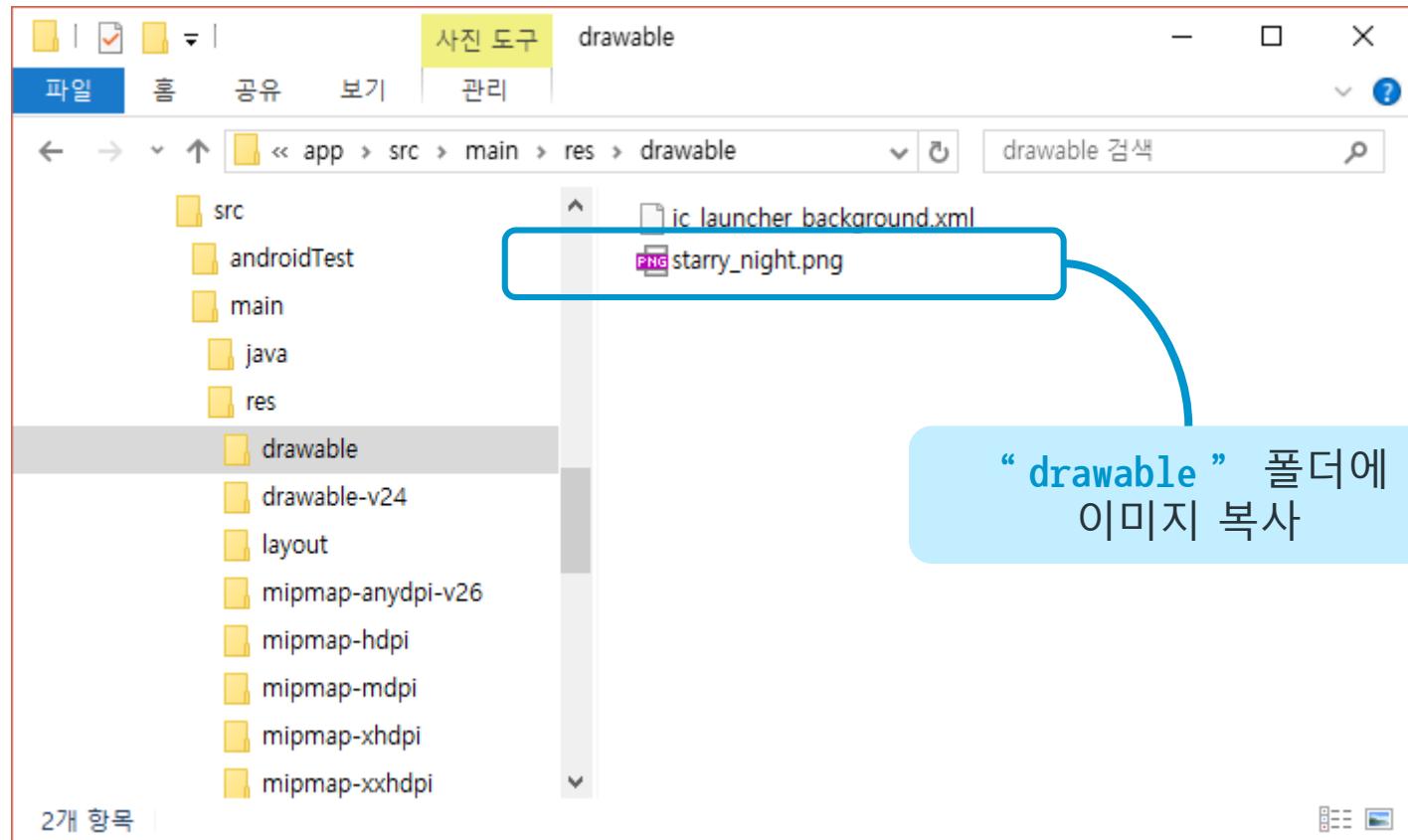
- res 폴더에 있는 **starry\_night.png** 파일을 **drawable** 폴더에 저장

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





"drawable" 폴더에  
이미지 복사

# ConstraintLayout을 LinearLayout로 바꾸기

14

The screenshot shows the Android Studio interface with two tabs: 'activity\_main.xml' and 'MainActivity.java'. On the left, there is a code editor and a preview window on the right.

**Top Editor (ConstraintLayout):**

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout 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="com.example.kyungtae.famouspainting.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>
```

**Bottom Editor (LinearLayout):**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.kyungtae.famouspainting.MainActivity">

</LinearLayout>
```

A large orange arrow points from the ConstraintLayout code up towards the preview window, indicating the transformation process.

**Preview Window:**

The preview window shows a blue header bar with the text "FamousPainting". Below it is a white content area containing the text "Hello World!" centered on the screen.

# Step 2.2 텍스트 자원의 편집

- strings.xml

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

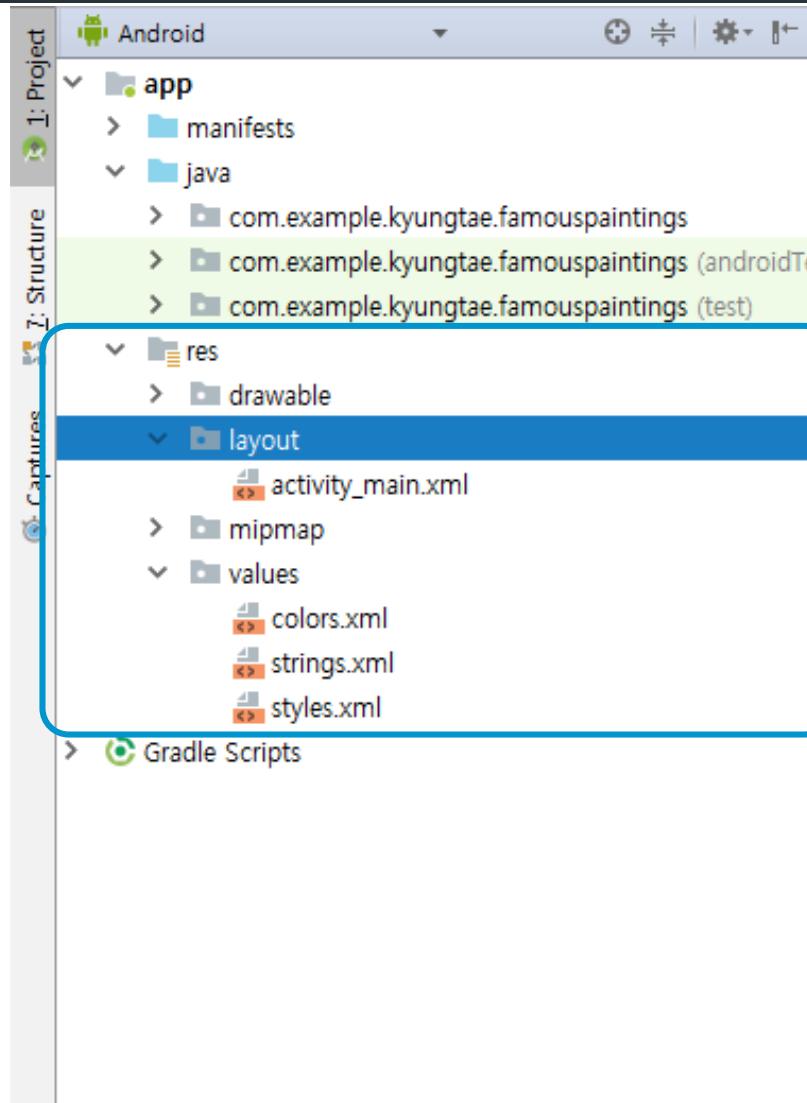
Edit translations for all locales in the translations editor.

```
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

16



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

# Drawable Resource

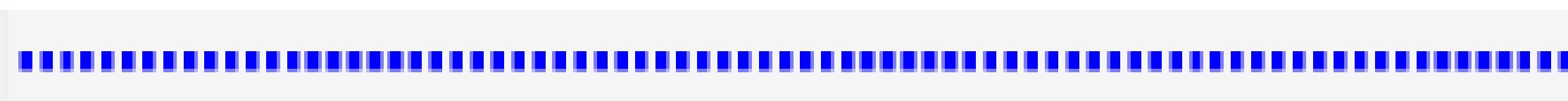
17

- 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

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

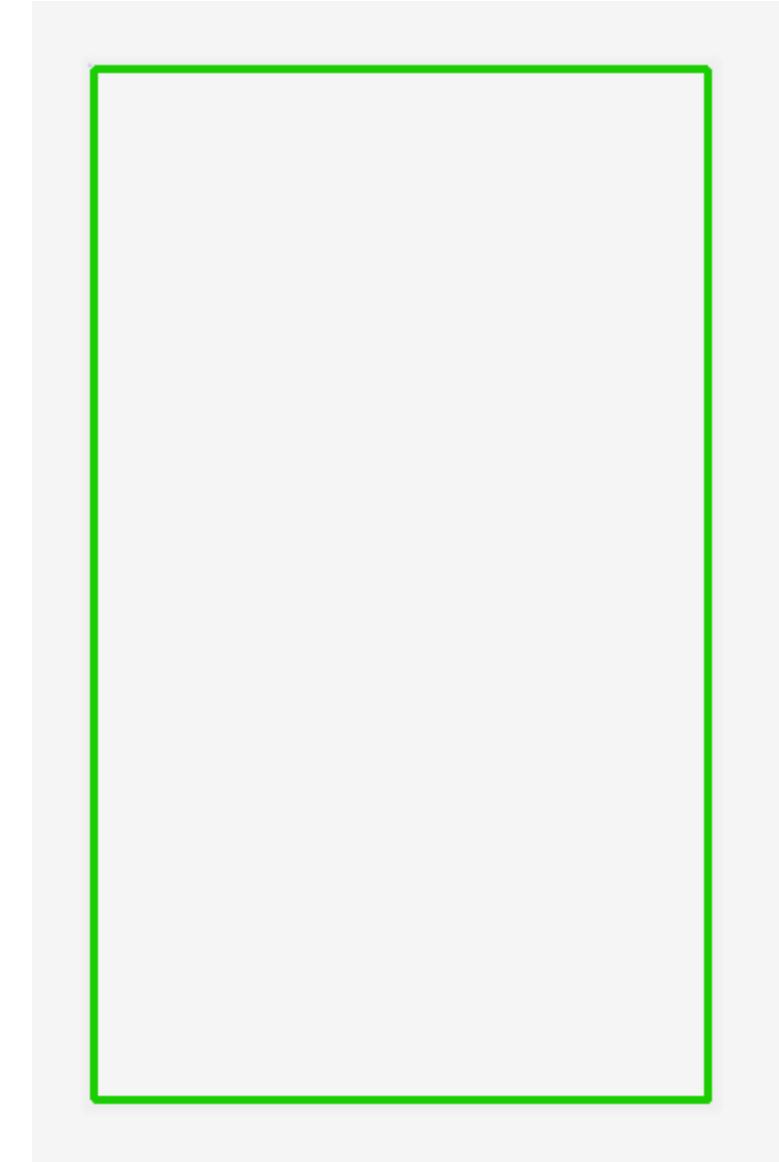
    <stroke
        android:width="5dp"
        android:color="#FF0000FF"
        android:dashWidth="3dp"
        android:dashGap="2dp">
    </stroke>
</shape>
```



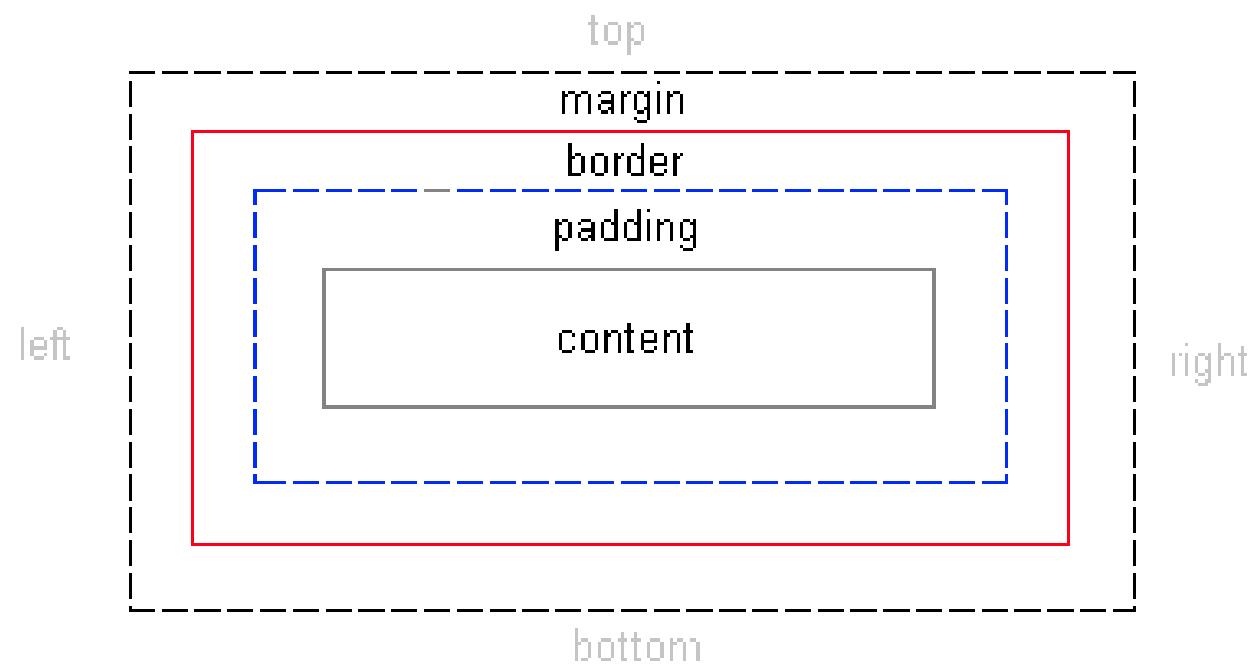
- 예제-사각형 그리기(출처: <http://aroundck.tistory.com/140>)

```
<?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



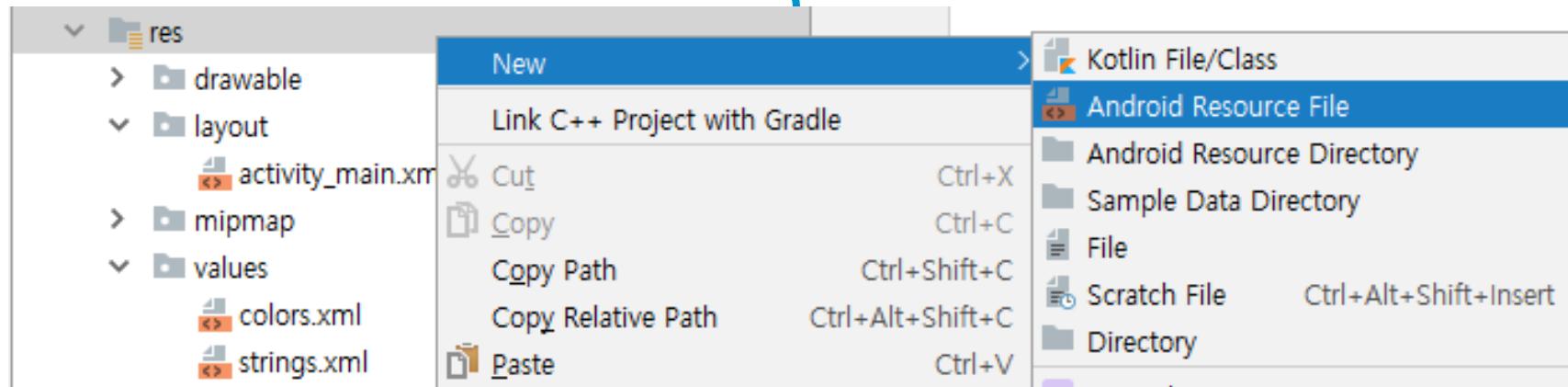
- margin edge
- border edge
- padding edge
- content edge

## • `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

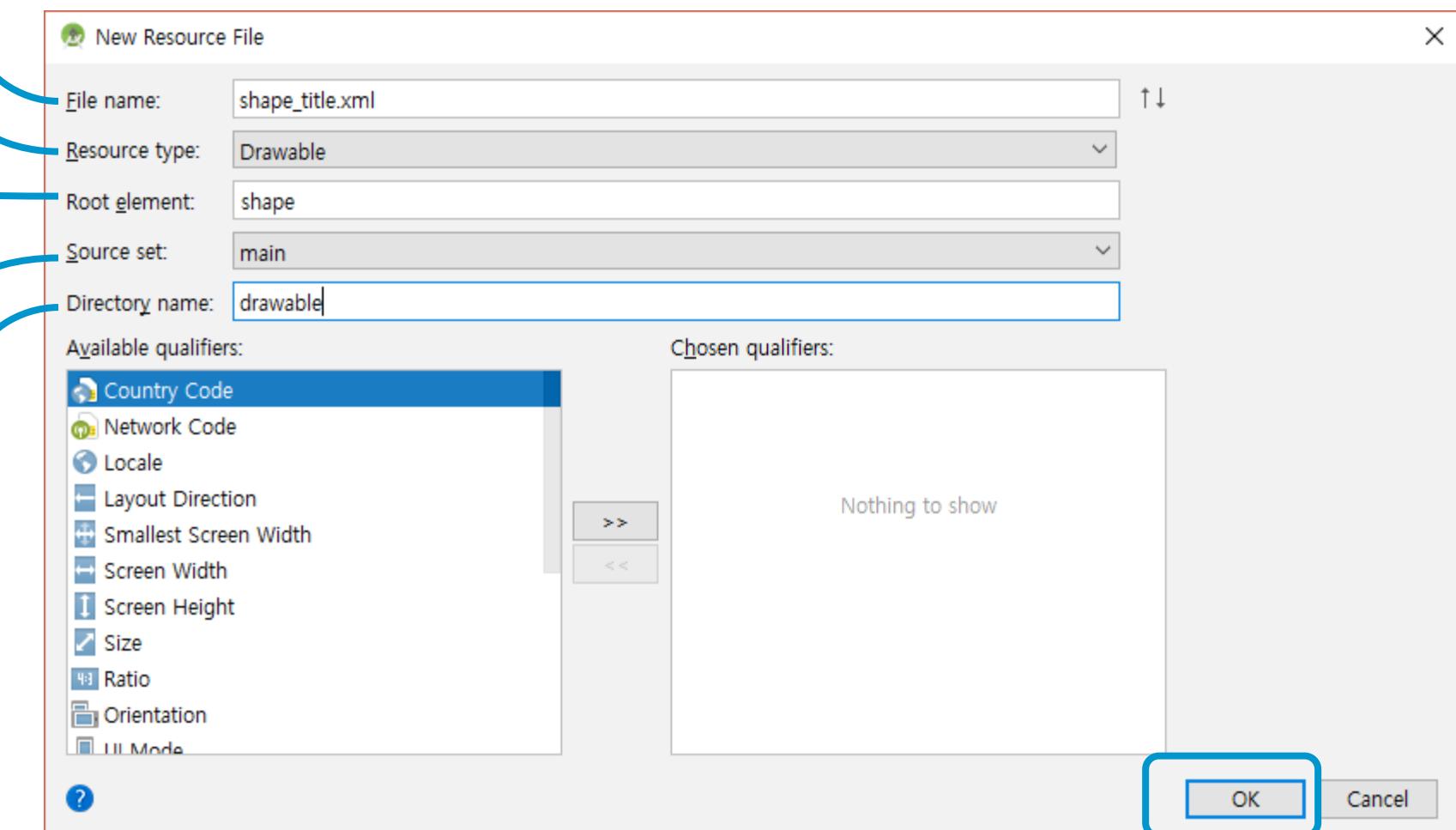
**File name:** shape\_title.xml

**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 several XML files: ic\_launcher\_background.xml, ic\_launcher\_foreground.xml (v24), shape\_title.xml (which is selected and highlighted in grey), and starry\_night.png. On the right, the code editor shows the content of shape\_title.xml:

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

- 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="#3061380b"/>
    <padding
        android:bottom="5dp"
        android:left="5dp"
        android:right="5dp"
        android:top="5dp"></padding>
    <corners android:radius="5dp">
    </corners>
</shape>
```

1 <?xml version="1.0" encoding="utf-8"?>  
2 <shape xmlns:android="http://schemas.android.com/apk/res/android"  
3 android:shape="rectangle">  
4 <solid android:color="#3061380b"/>  
5 출력모양을 사각형으로 지정  
6 출력모양을 내부의 색  
7 내부 패딩 정보  
8 출력모양 모서리를 둥근 모양으  
9 로 지정(반지름은 5dp)  
10 출력모양을 내부의 색  
11 내부 패딩 정보  
12 출력모양 모서리를 둥근 모양으  
13 로 지정(반지름은 5dp)  
14 출력모양을 내부의 색  
15 내부 패딩 정보  
16 출력모양 모서리를 둥근 모양으  
17 로 지정(반지름은 5dp)  
18 출력모양을 내부의 색

- activity\_main.xml  
android.support.constraint.ConstraintLayout → LinearLayout 변경

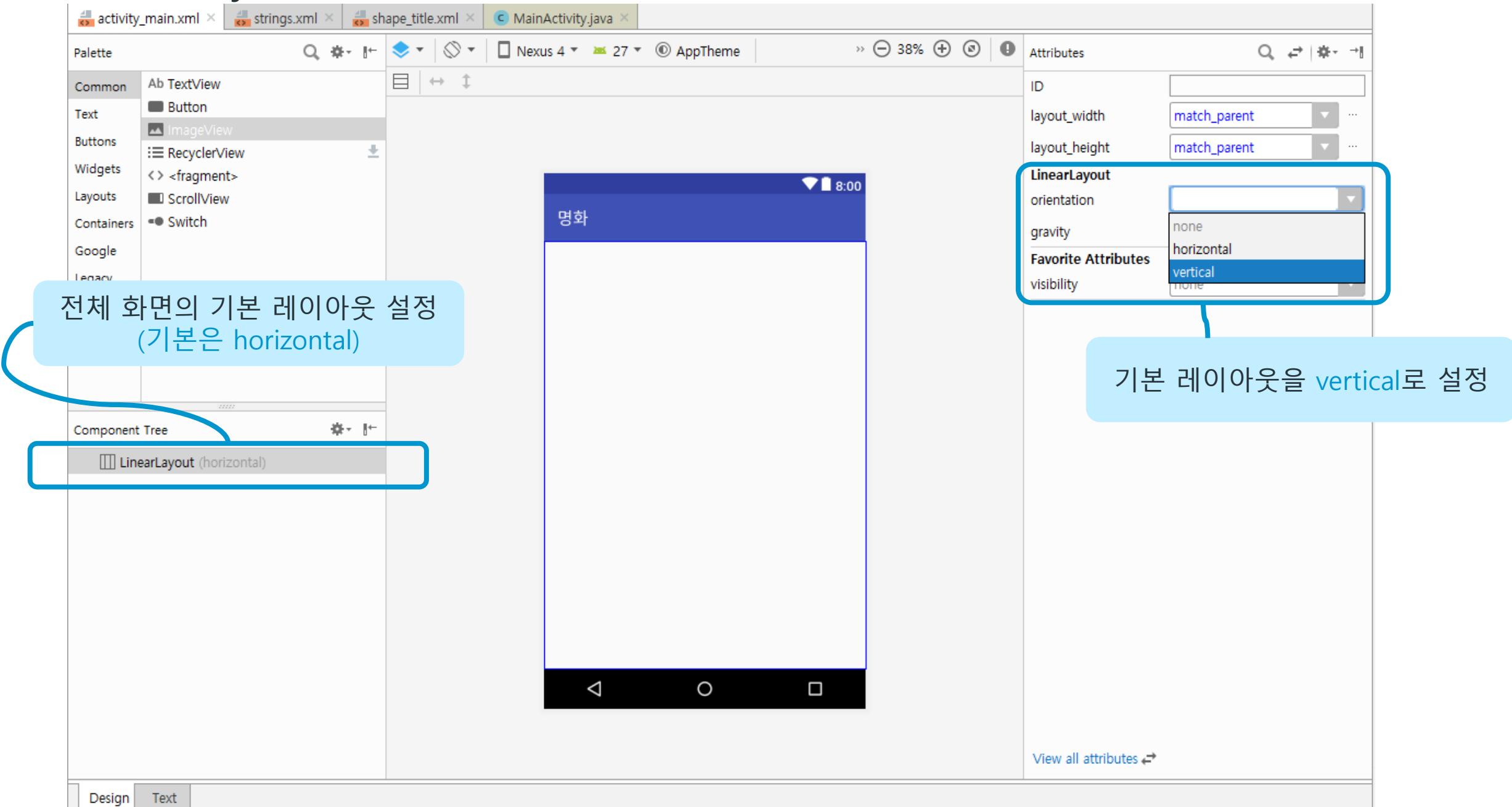
```
activity_main.xml MainActivity.java strings.xml shape_list.xml
1 <?xml version="1.0" encoding="utf-8"?>
2 3 <ConstraintLayout android.support.constraint.ConstraintLayout 4 xmlns:android="http://schemas.android.com/apk
5   6   xmlns:app="http://schemas.android.com/apk/res-auto"
6     7   xmlns:tools="http://schemas.android.com/tools"
7     8   android:layout_width="match_parent"
8     9   android:layout_height="match_parent"
9       10  tools:context=".MainActivity">
11
12 <TextView
13   android:layout_width="wrap_content"
14   android:layout_height="wrap_content"
15   android:text="Hello World!"
16   app:layout_constraintBottom_toBottomOf="parent"
17   app:layout_constraintLeft_toLeftOf="parent"
18   app:layout_constraintRight_toRightOf="parent"
19   app:layout_constraintTop_toTopOf="parent" />
20
21 <22  ConstraintLayout>
```

```
activity_main.xml × MainActivity.java × strings.xml × shape_list.xml ×  
1 <?xml version="1.0" encoding="utf-8"?>  
2 <L xmlns:android="http://schemas.android.com/apk/res/android"  
3     LinearLayout  
4         ListView  
5             android.support.v7.app.AppCompatDelegateImplV9.L...  
6             android.support.v7.view.menu.ListMenuItemView  
7             android.support.v7.widget.LinearLayoutCompat  
8         FrameLayout  
9         GridLayout  
10            TwoLineListIte...  
11            android.support.constraint.ConstraintLayout  
12            android.support.design.widget.CoordinatorLayout  
13            android.support.v4.widget.ContentLoadingProgressBar  
14            Press Ctrl+Space to view tags from other namespaces  
15            app:layout_constraintRight_toRightOf="parent"  
16            app:layout_constraintTop_toTopOf="parent" />  
17  
18        </L>
```

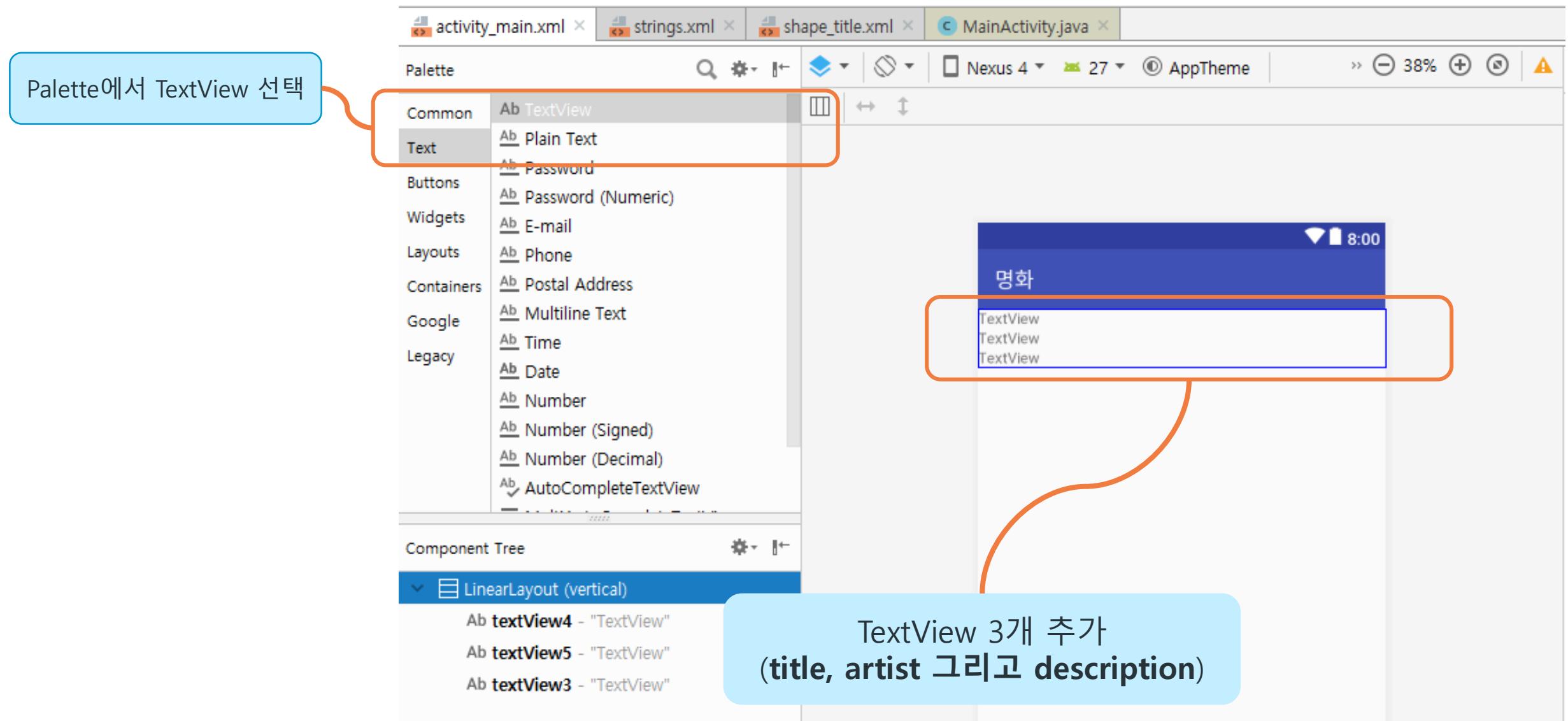
"L"을 입력하면 팝업창에  
뜬 LinearLayout 을 선택한  
다.

```
activity_main.xml MainActivity.java strings.xml shape_list.xml
1 <?xml version="1.0" encoding="utf-8"?
2 c <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     tools:context=".MainActivity">
8
9     <TextView
10        android:layout_width="wrap_content"
11        android:layout_height="wrap_content"
12        android:text="Hello World!"
13        app:layout_constraintBottom_toBottomOf="parent"
14        app:layout_constraintLeft_toLeftOf="parent"
15        app:layout_constraintRight_toRightOf="parent"
16        app:layout_constraintTop_toTopOf="parent" />
17
18     <LinearLayout>
```

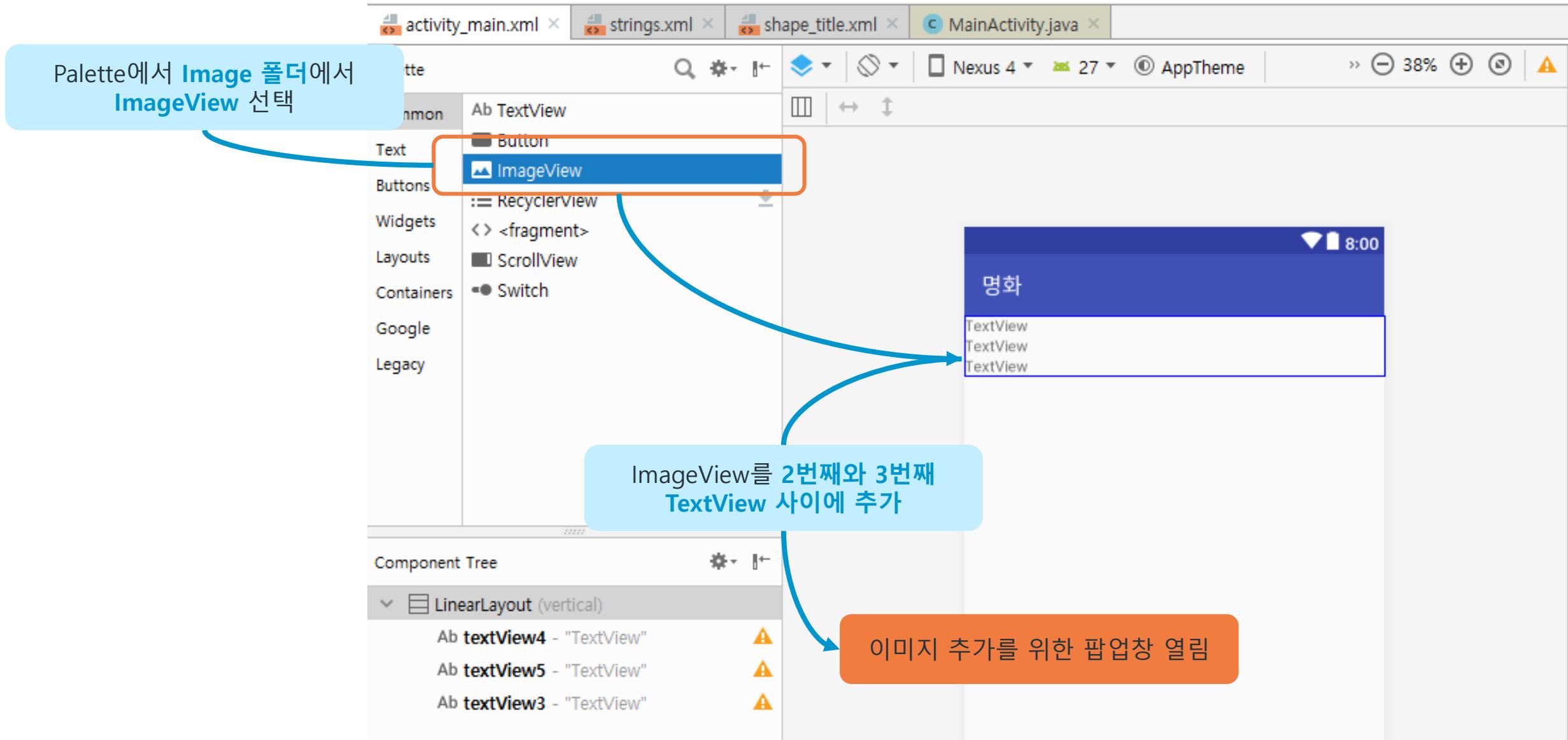
- LinearLayout의 orientation을 vertical로 설정



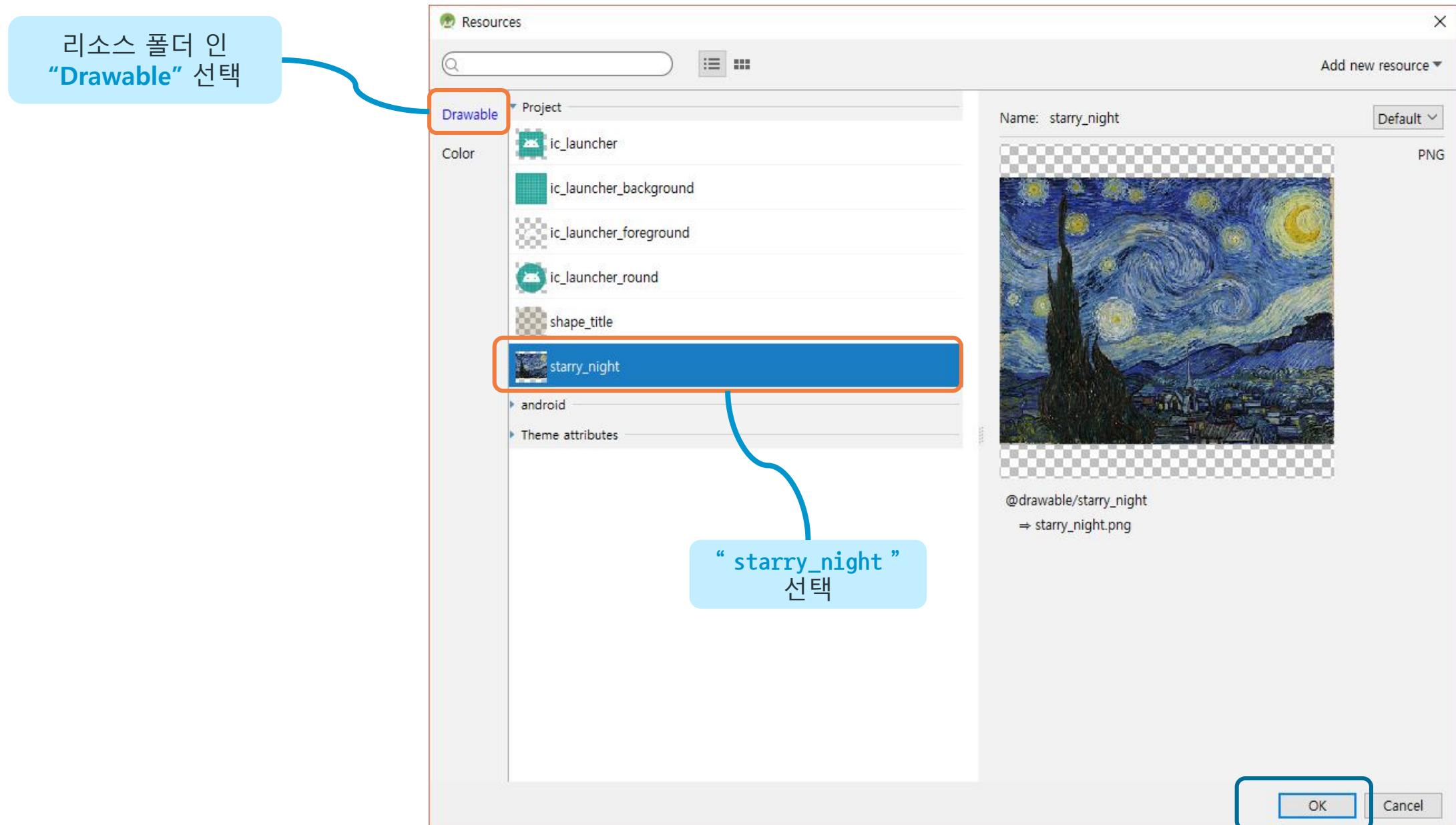
## • LinearLayout에 TextView와 ImageView 추가하기



## • LinearLayout에 TextView와 ImageView 추가하기



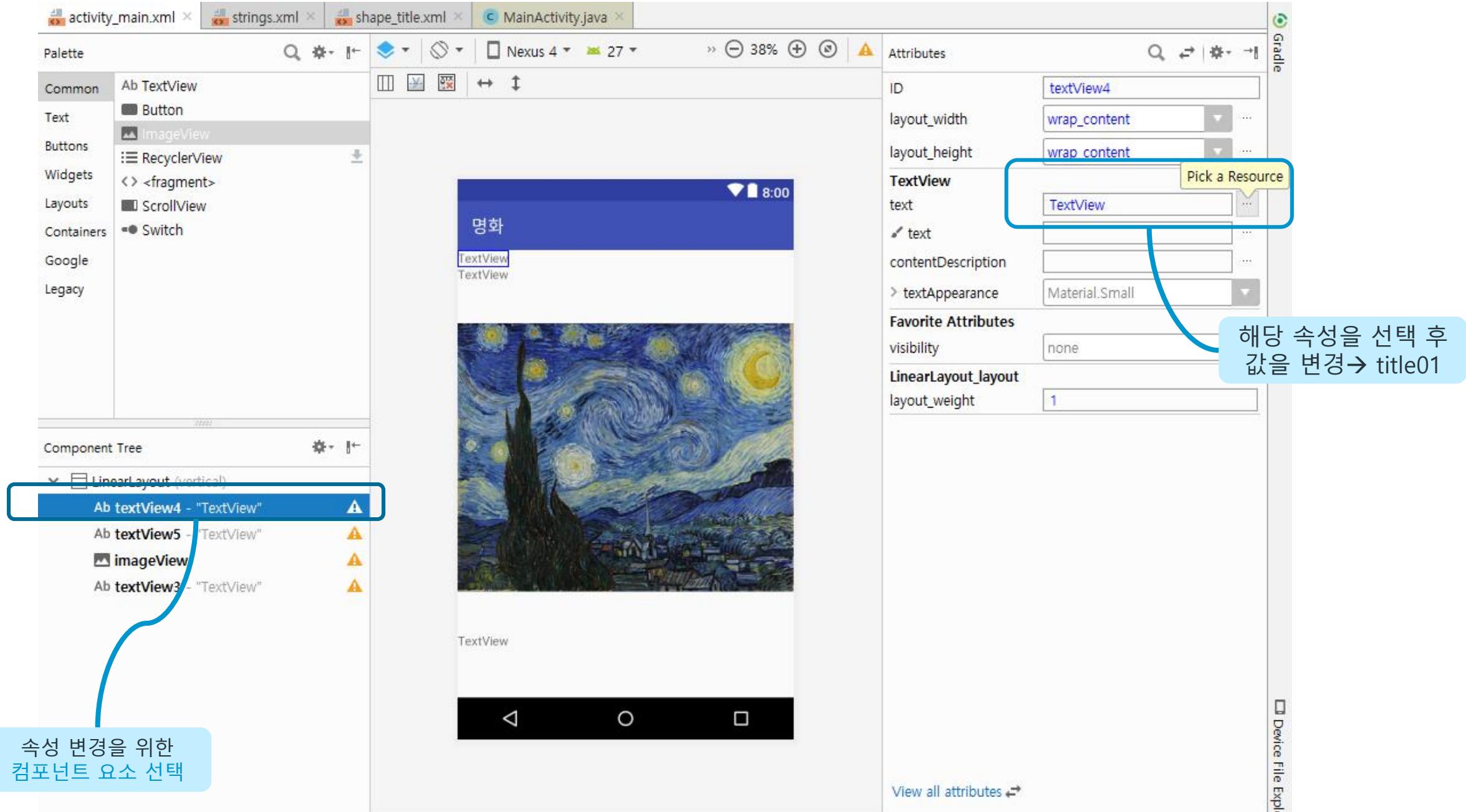
- ImageView에 이미지 소스 선택 팝업 창

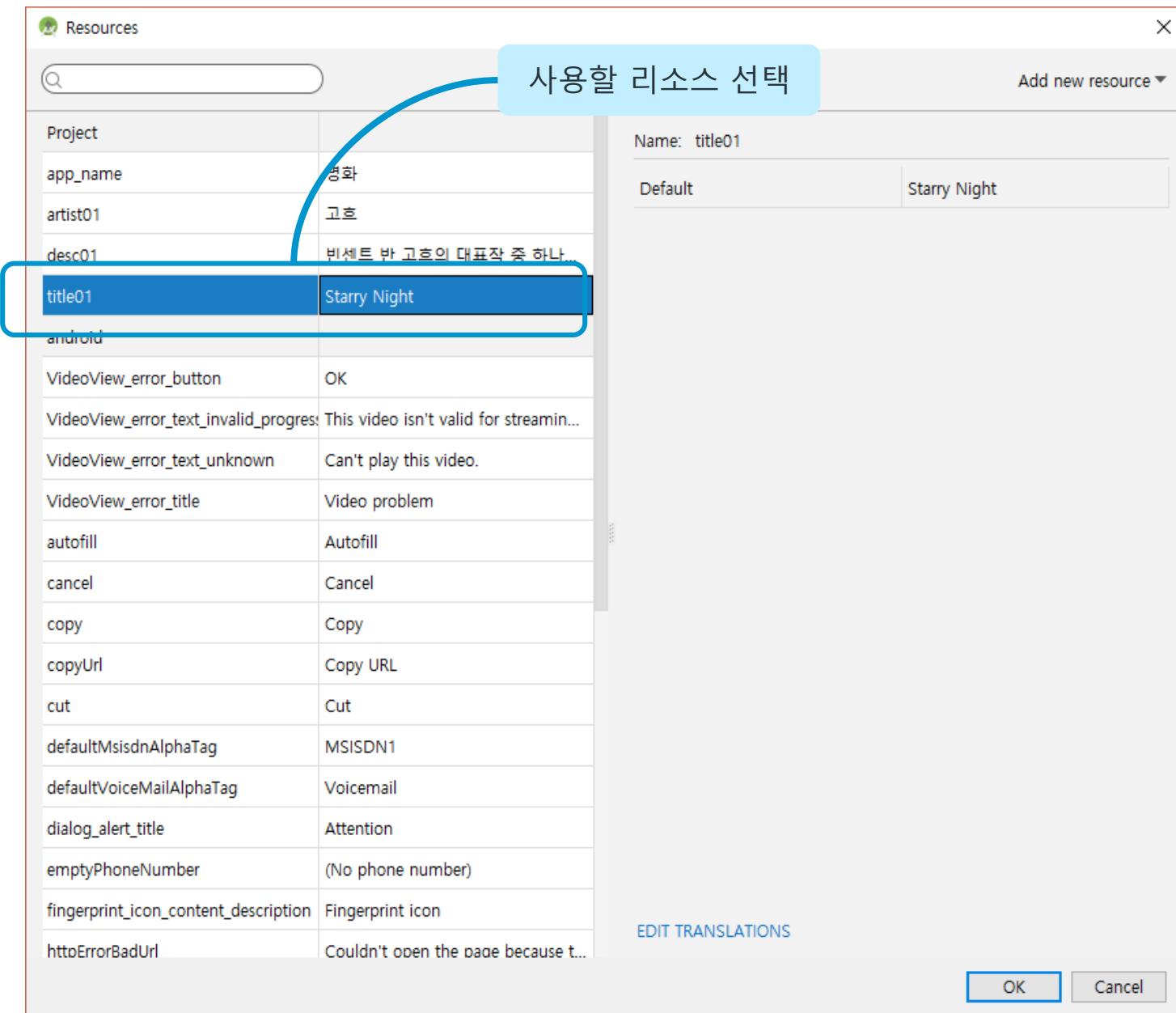
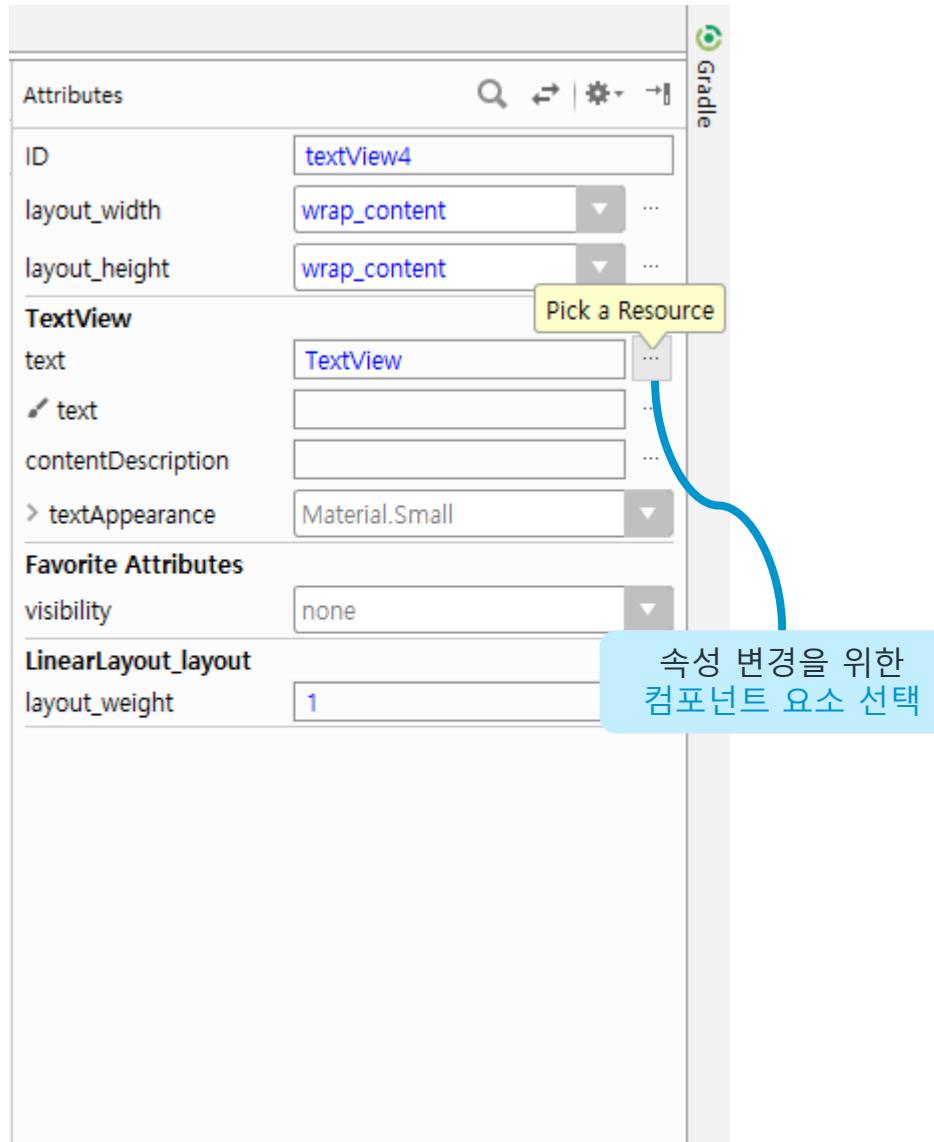


# 완성된 화면

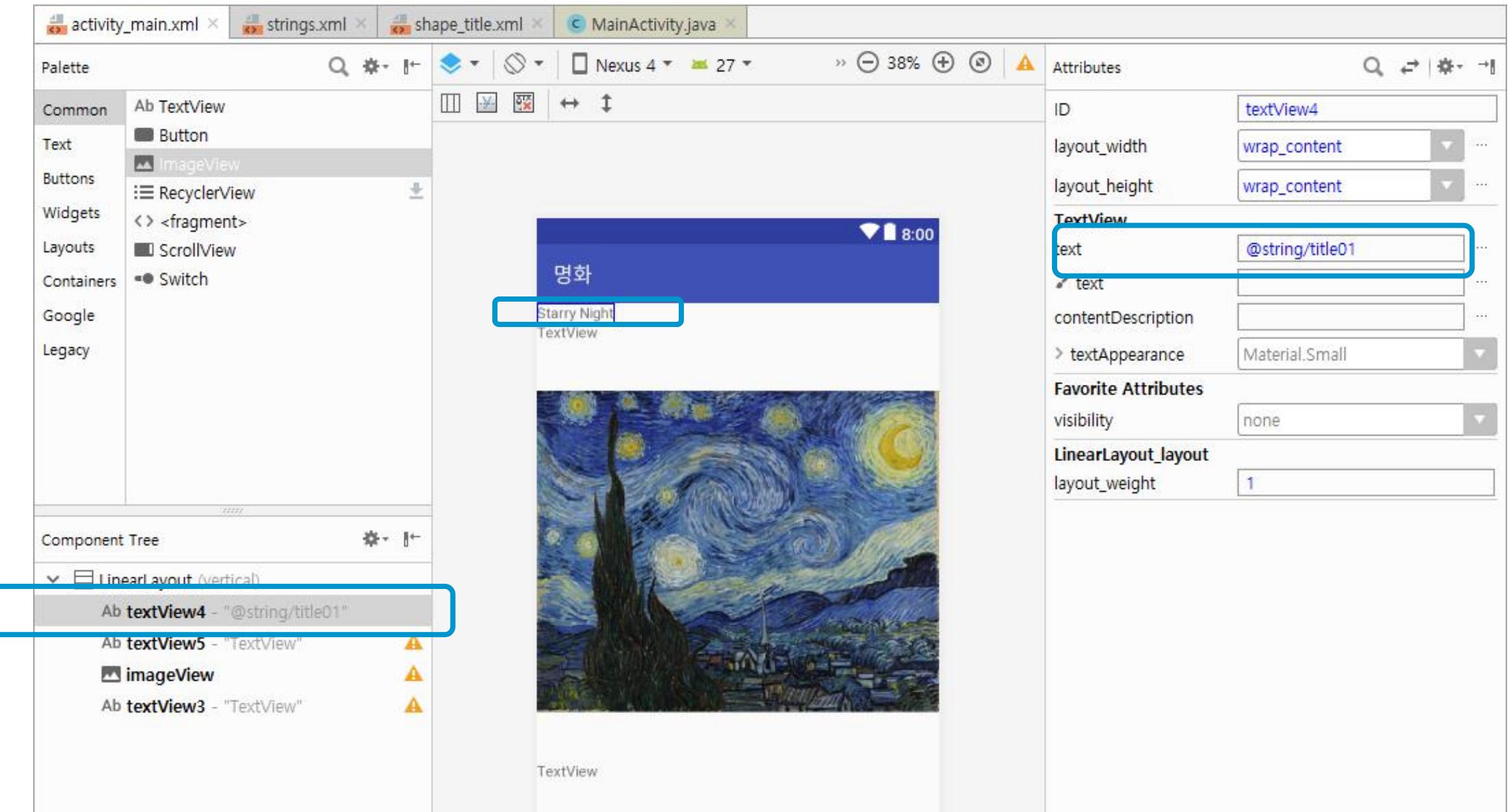


## • Title TextView의 “Text” 속성 변경하기

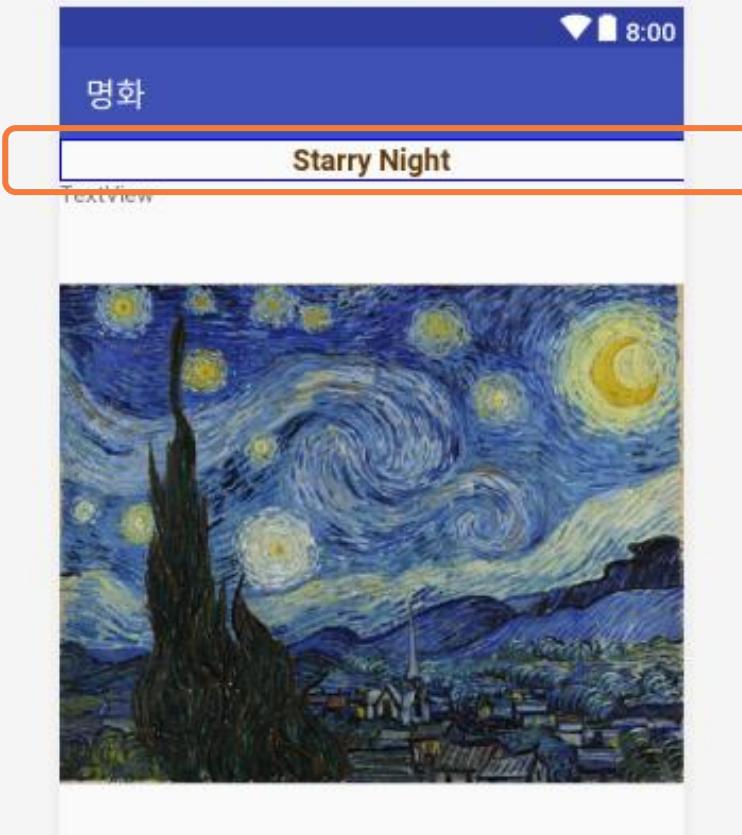




## • title TextView의 “Text” 속성 변경 완료



## • title 컴포넌트의 기타 속성 변경하기



The screenshot shows the Android Studio XML layout editor. On the left, there's a preview of an Android application interface featuring a title bar with the text "Starry Night". The title bar is highlighted with an orange border. Below it is a large image of Vincent van Gogh's "Starry Night". The top of the screen displays device information (Nexus 4, 27), zoom level (38%), and various tool icons.

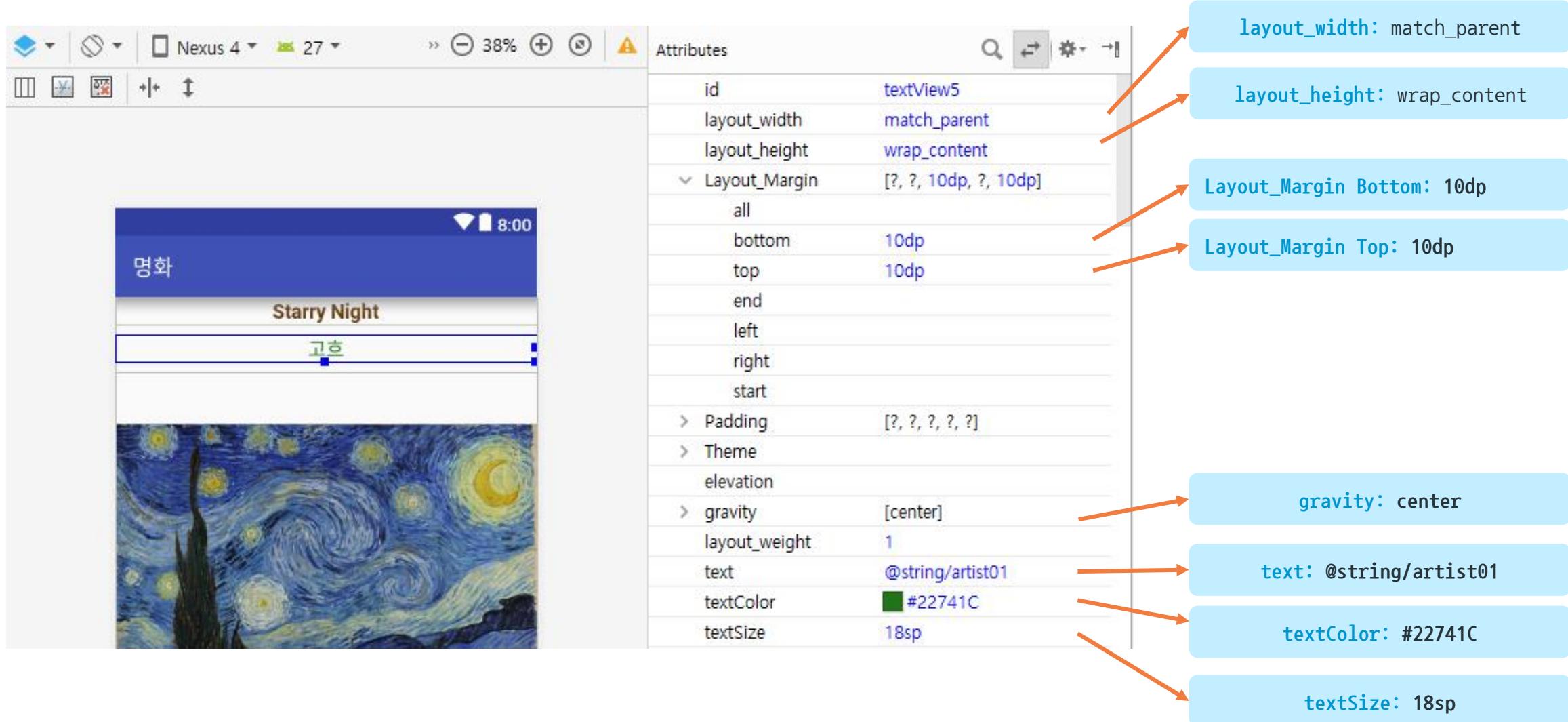
**Attributes**

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

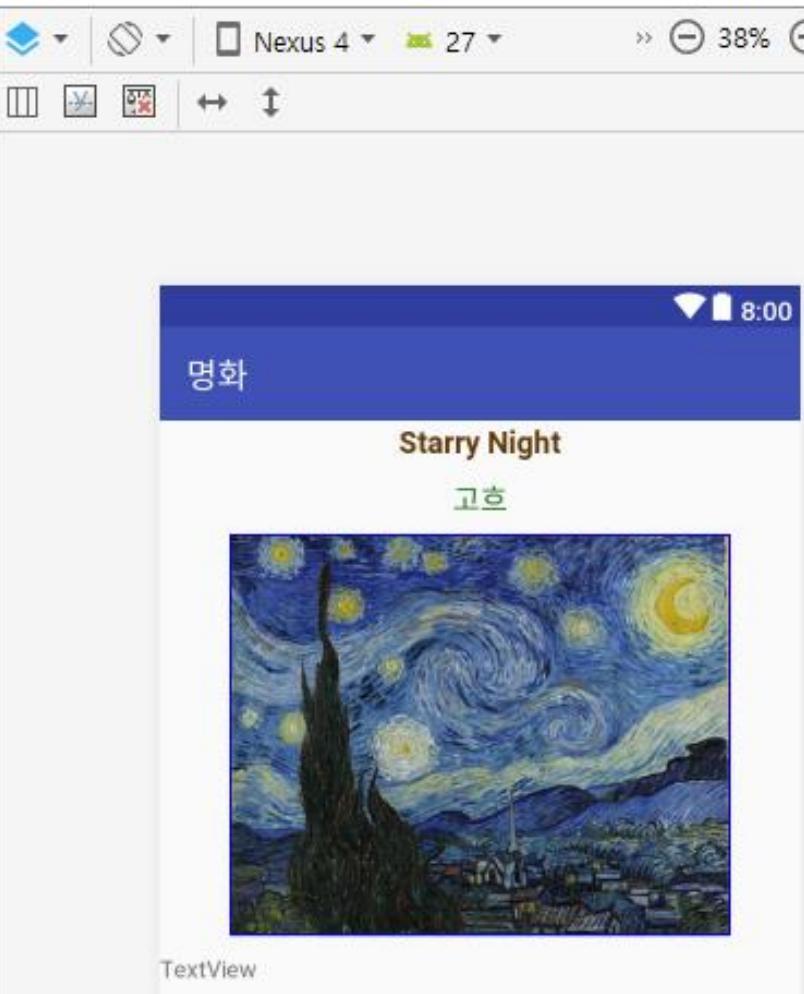
Arrows point from specific attributes in the table to their corresponding visual changes in the title bar:

- layout\_width: match\_parent → Title bar has a matching width relative to its parent.
- layout\_height: wrap\_content → Title bar has a height that wraps its content.
- gravity: center → Title bar text is centered.
- text: @string/title01 → Title bar text is set to the string resource @string/title01.
- textColor: #613808 → Title bar text color is set to #613808.
- textSize: 18sp → Title bar text size is set to 18sp.
- textStyle: bold → Title bar text style is set to bold.

## • 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, there's a preview of a mobile application interface featuring a blue header with Korean text, a title "Starry Night", and a subtitle "고흐". Below is a large image of Vincent van Gogh's "Starry Night". A TextView at the bottom contains the text "Text View". On the right, the "Attributes" panel lists the properties of the ImageView:

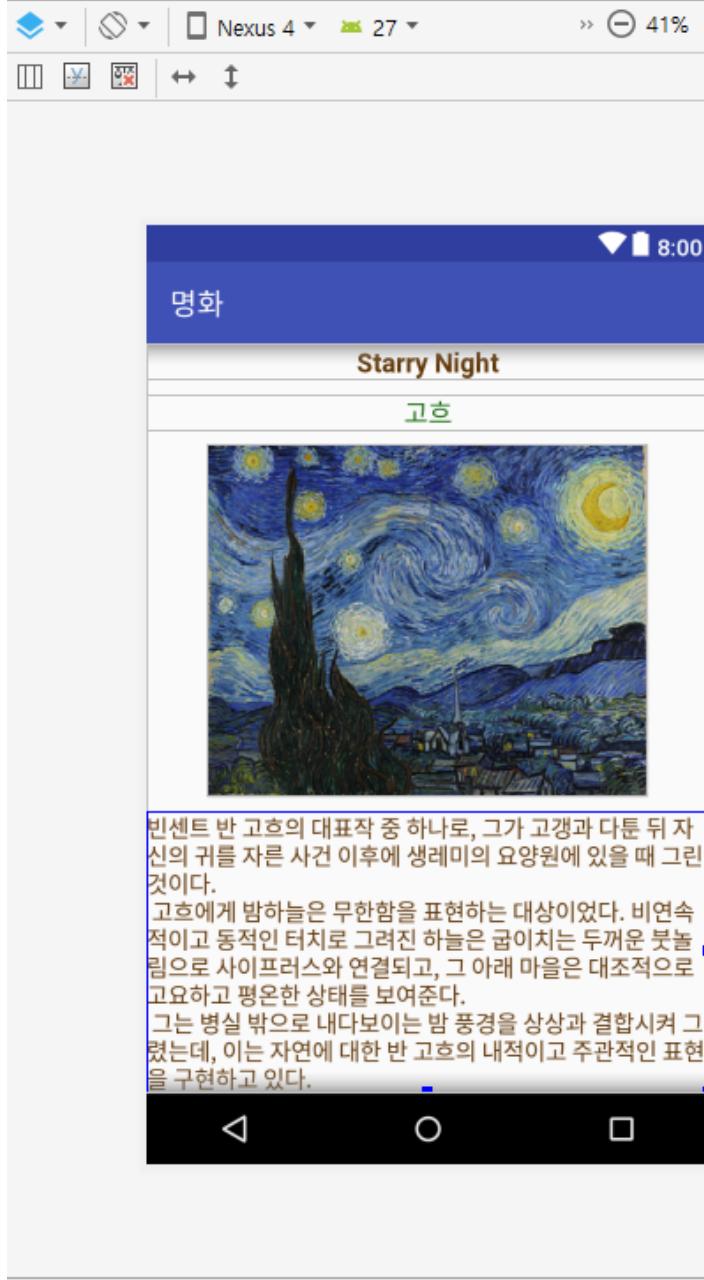
id	imageView
layout_width	wrap_content
layout_height	wrap_content
Layout_Margin	[?, ?, ?, ?, 10dp]
all	
bottom	10dp
end	
left	
right	
start	
top	
> Padding	[?, ?, ?, ?, ?]
> Theme	
elevation	
adjustViewBounds	<input checked="" type="checkbox"/>
> layout_gravity	[center]
layout_weight	1
maxHeight	300dp
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 listed are:
  - id: textView3
  - layout\_width: wrap\_content
  - layout\_height: wrap\_content
  - Layout\_Margin: [?, ?, ?, ?, ?]
  - Padding: [?, ?, ?, ?, ?]
  - Theme: (empty)
  - elevation: (empty)
  - layout\_weight: 1
  - text: @string/desc01
  - textColor: #61380b
  - textSize: 16sp
  - accessibilityLiveRegion: (empty)
  - accessibilityTraversalAf: (empty)
  - accessibilityTraversalBe: (empty)
  - allowUndo: (checkbox)
  - alpha: (empty)
  - autoLink: (checkbox)
  - autoSizeMaxTextSize: (empty)
  - autoSizeMinTextSize: (empty)
  - autoSizePresetSizes: (empty)
  - autoSizeStepGranularit: (empty)
  - autoSizeTextType: (empty)
  - autoText: (checkbox)
  - autofillHints: (empty)
  - background: (empty)
  - backgroundTint: (empty)
  - backgroundTintMode: (empty)
  - breakStrategy: (empty)
  - bufferType: (empty)
  - capitalize: (empty)
  - clickable: (checkbox)
  - contentDescription: (empty)
  - contextClickable: (checkbox)
- Text Preview:** The text preview shows the title 'Starry Night' and '고흐' above the painting, and a descriptive paragraph below it.

Annotations on the right side highlight specific attributes:

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

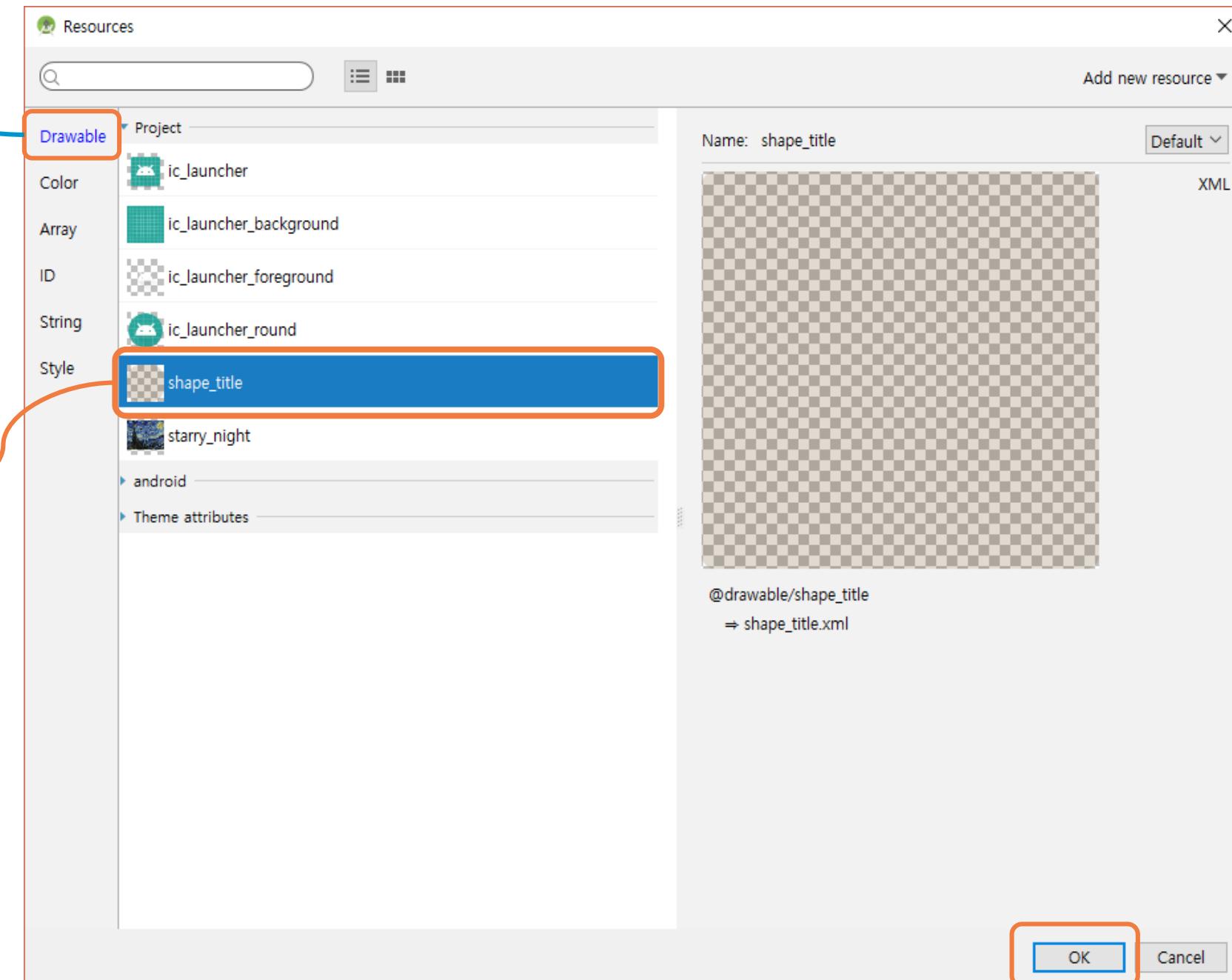
## • `shape_title.xml`을 이용한 제목바의 배경 처리

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:** Shows categories like Common, Text, Buttons, Widgets, Layouts, Containers, Google, and Legacy. `ImageView` is currently 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 triangle)
  - `Ab textView3 - "@string/desc01"`
- Preview:** Displays the UI with a blue title bar labeled "명화" and "Starry Night", and a main content area showing "고흐" and "The Starry Night" painting.
- Attributes Panel:** Shows the properties for the `textView4` component:

id	<code>textView4</code>
layout_width	<code>match_parent</code>
layout_height	<code>wrap_content</code>
> Layout_Margin	<code>[?, ?, ?, ?, ?]</code>
> Padding	<code>[?, ?, ?, ?, ?]</code>
> Theme	
elevation	
> gravity	<code>[center]</code>
layout_weight	<code>1</code>
text	<code>@string/title01</code>
textColor	<code>#613808</code>
textSize	<code>18sp</code>
> textStyle	<code>[bold]</code>
accessibilityLiveRegion	
accessibilityTraversalAf	
accessibilityTraversalBe	
allowUndo	<input checked="" type="checkbox"/>
alpha	
> autoLink	<input type="checkbox"/>
autoSizeMaxTextSize	
autoSizeMinTextSize	
autoSizePresetSizes	
autoSizeStepGranularity	
autoSizeTextType	
autoText	<input checked="" type="checkbox"/>
autofillHints	
- Bottom Navigation:** Shows standard Android navigation icons (back, home, recent apps).

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.

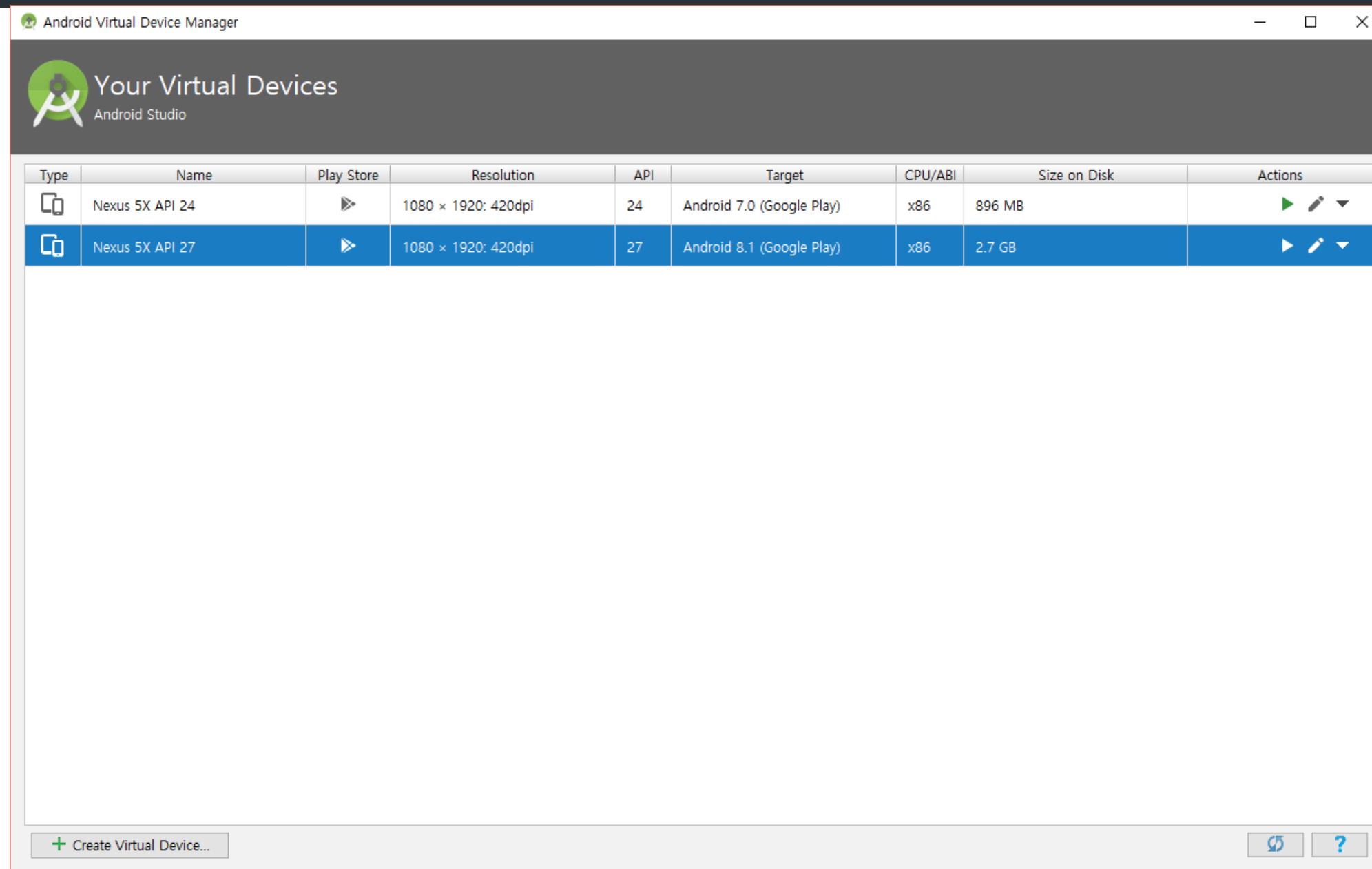
## • 화면 상하좌우에 여백주기

The screenshot shows the Android Studio layout editor with the following details:

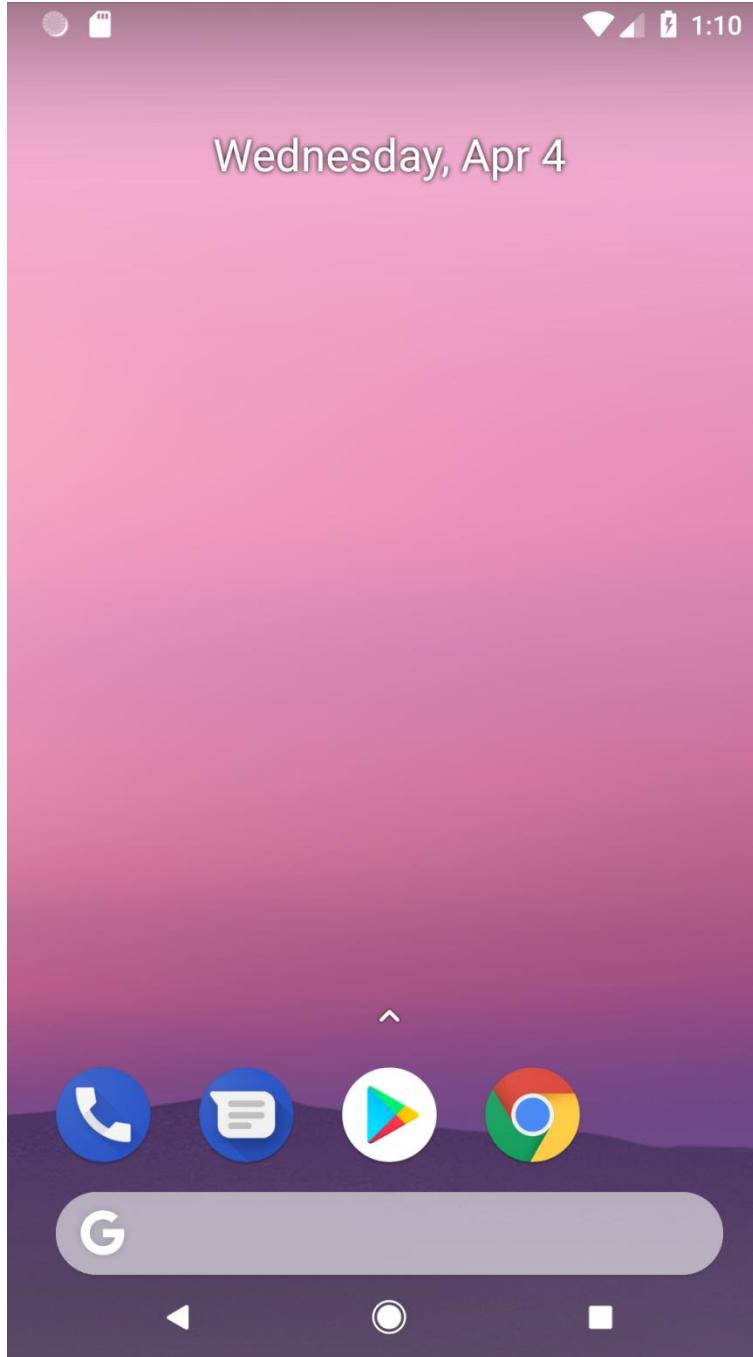
- Palette:** The "Images" category is selected, showing an "ImageView" component.
- Properties:** The "Layout\_Margin" section is highlighted with a red box, showing the value [5sp, ?, ?, ?, ?].
- Component Tree:** A `LinearLayout` (vertical) contains:
  - `textView` - "@string/title01"
  - `textView2` - "@string/artist01"
  - `imageView2` (selected)
  - `textView3` - "@string/desc01"
- Preview:** The preview shows a blue header with "명화" and "7:00", a title bar with "Starry Night", a subtitle "고흐", and the painting itself.
- Text Labels:** There are two text labels in Korean describing the painting.

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

46



# 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





question

&



answer

