

Week04.  
이미지의 출력



# 개발환경 구축 절차

2

주 차	수 업 내 용
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>



# https://goo.gl/vSdwLq

3

The screenshot shows a web browser window displaying the GitHub repository page for 'hopypark / LectureNotes'. The browser's address bar shows the URL 'https://github.com/hopypark/LectureNotes/tree/master/AndroidApp'. The repository page includes a header with the GitHub logo, repository name, and navigation links like 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the header, there are buttons for 'Unwatch', 'Star', and 'Fork'. The main content area shows a list of files and folders in the 'AndroidApp' directory, including 'README.md', 'Week02.Chap02.앱 개발환경 구축.pdf', 'Week02.Chap03.앱 프로젝트 구조와 실행원리.pdf', 'Week03.Chap06.텍스트 출력과 레이아웃.pdf', 'Week04.Chap07.이미지의 출력.pdf', 'ic\_launcher\_new-web.png', and 'starry\_night.png'. Each file entry shows its name, the commit message, and the time since the last commit.

Branch: master ▾ **LectureNotes** / AndroidApp /

Create new file Upload files Find file History

hopypark Week04.Chap07.이미지의 출력 Latest commit 20be68a a minute ago

..

README.md	Create readme.md	a month ago
Week02.Chap02.앱 개발환경 구축.pdf	Add files via upload	a month ago
Week02.Chap03.앱 프로젝트 구조와 실행원리.pdf	Add files via upload	26 days ago
Week03.Chap06.텍스트 출력과 레이아웃.pdf	Week03.Chap06.텍스트 출력과 레이아웃.pdf	12 days ago
Week04.Chap07.이미지의 출력.pdf	Week04.Chap07.이미지의 출력	a minute ago
ic_launcher_new-web.png	Add files via upload	29 days ago
starry_night.png	starry_night.png	4 hours ago

README.md

# 이미지 출력 앱의 예



(a) 사진 목록(인스타그램)

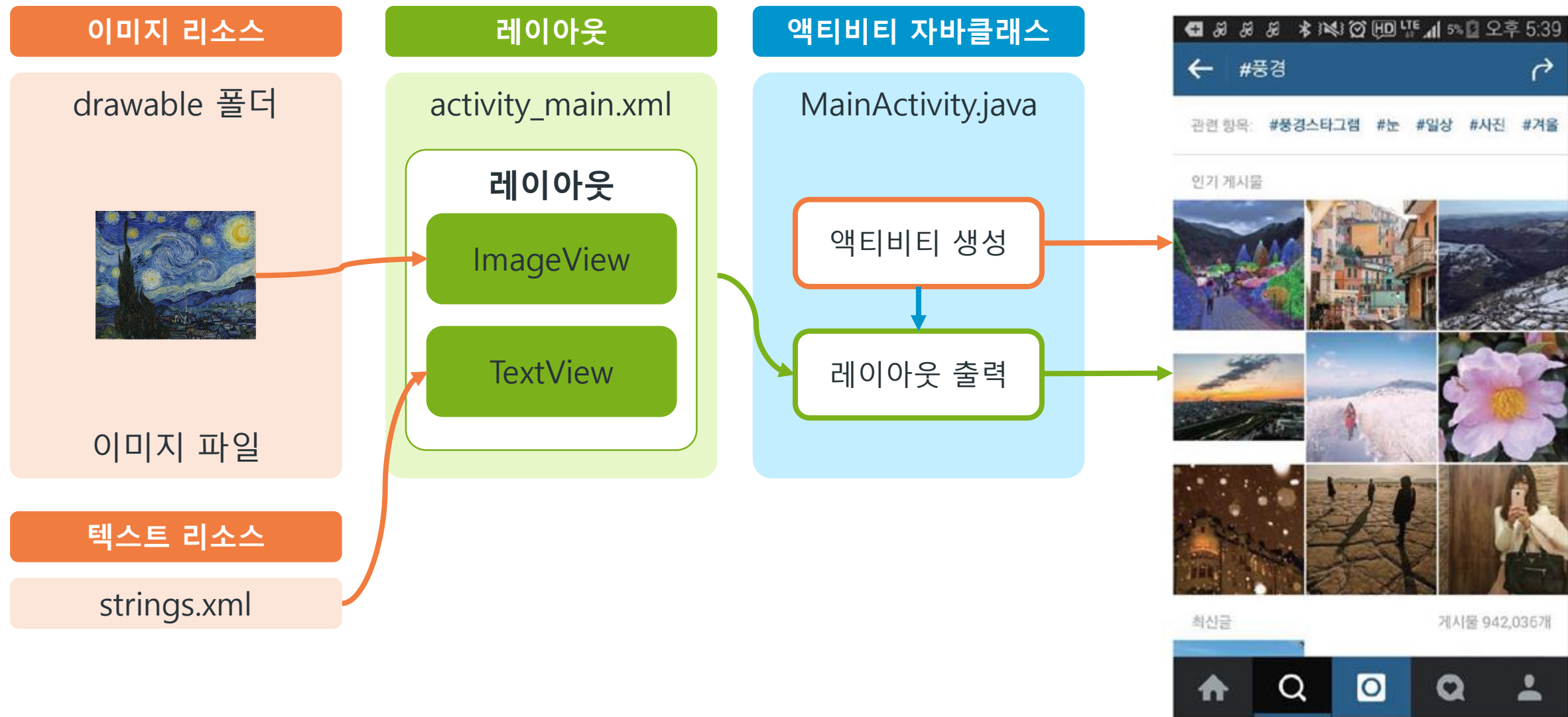


(b) 음식 배달 메뉴(배달의민족)

- 이미지 중심 앱의 예

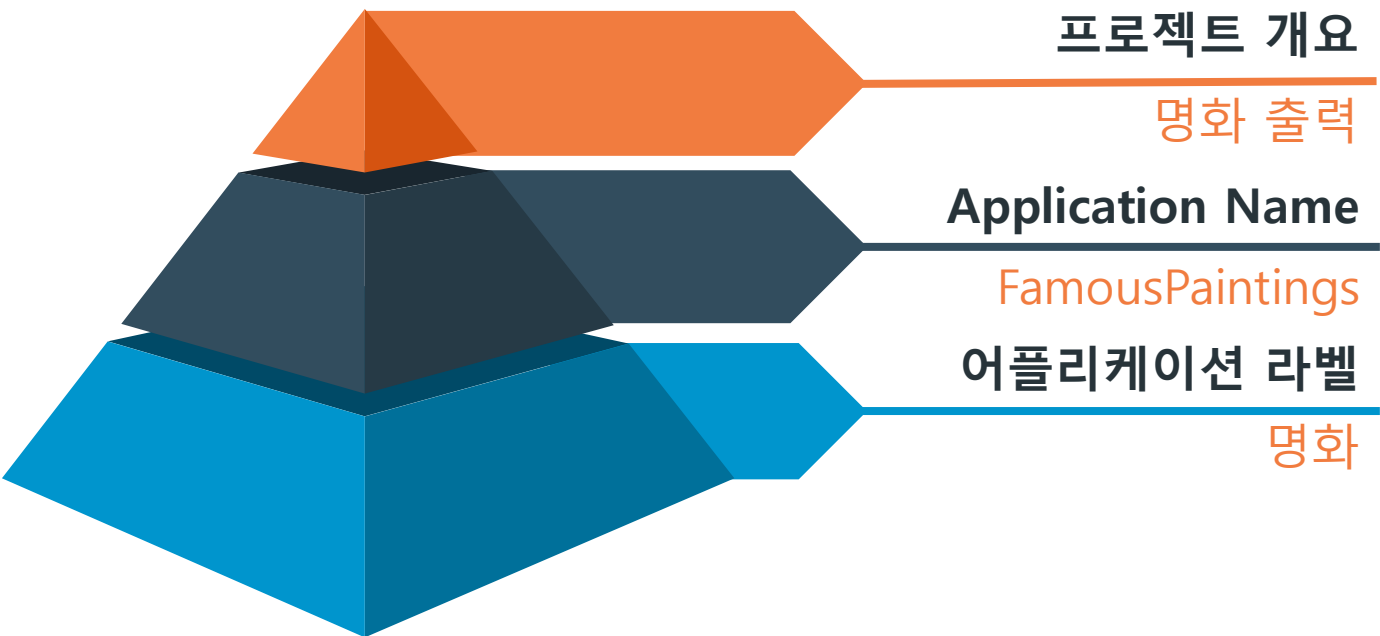
# 이미지 출력 원리

5



# Step 0.프로젝트 개요

6



# Step 1. 프로젝트 생성

7

절차	내 용
①프로젝트 시작	메뉴에서 ‘ <code>File → New Project</code> ’ 클릭
②프로젝트 구성	Application Name: <code>FamousPaintings</code>
	Company Domain: <code>사용자계정.example.com</code> (디폴트 사용)
③제품형태	<code>Phone and Tablet</code> (사용할 안드로이드 버전 지정: <code>Android 7.0 Nougat</code> )
④액티비티 유형	<code>Empty Activity</code>
⑤파일 옵션	Activity Name: <code>MainActivity</code>
	Layout Name: <code>activity_main</code>

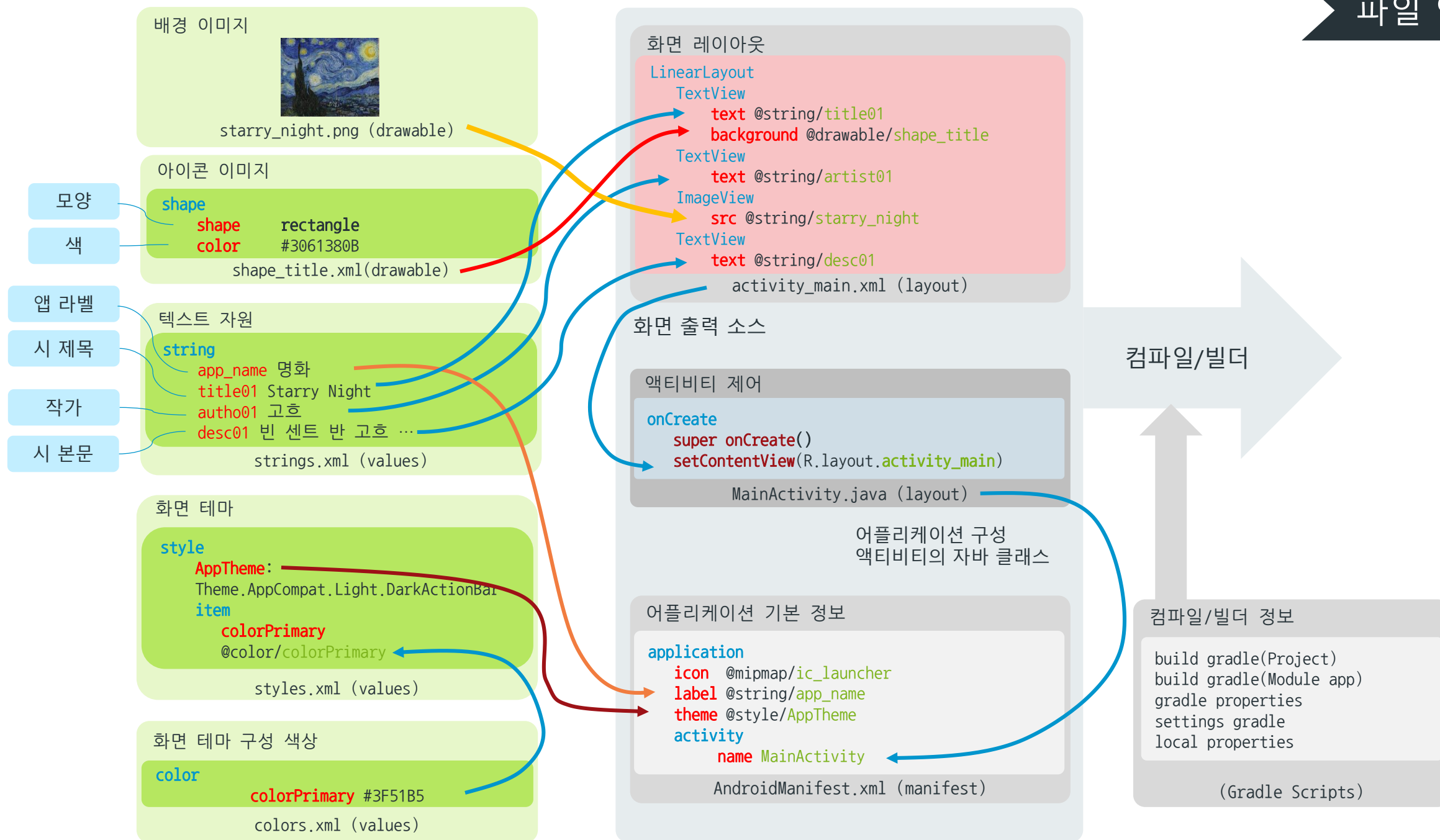


# Step 2. 파일 편집

8

모듈	폴더	소스 파일	편집 내용
manifests		AndroidManifest.xml	
java	com.example.kyungtae.famouspaintings	MainActivity.java	
res	drawable	starry_night.png	• 영화 이미지
		shape_title.xml	• 영화 제목의 출력모양 설계(배경색, 패딩, 모서리)
	layout	activity_main.xml	• 영화 화면 구성(제목, 작가, 이미지, 설명)
	mipmap	ic_launcher.png	
	values	dimens.xml	
		strings.xml	• 영화 목록 아이템(제목, 작가, 설명)
		styles.xml	






# 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	

# ConstraintLayout을 LinearLayout로 바꾸기

11

The image shows the Android Studio IDE with two tabs: `activity_main.xml` and `MainActivity.java`. The `activity_main.xml` file is open, displaying XML code for a layout. An orange arrow points from the `ConstraintLayout` code in the top pane to the `LinearLayout` code in the bottom pane, indicating a transformation.

**Top Pane (Original Code):**

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-
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="com.example.kyungtae.famouspainting.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 </android.support.constraint.ConstraintLayout>
```

**Bottom Pane (Modified Code):**

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   tools:context="com.example.kyungtae.famouspainting.MainActivity">
7
8
9 </LinearLayout>
```

The right preview pane shows the app's title "FamousPainting" and the time "7:00". The bottom navigation bar shows the standard Android navigation icons.

drawable 폴더에  
이미지 추가하기

# drawable 폴더에 이미지 추가하기

The image shows an IDE interface with a project tree on the left and a context menu open on the right. The project tree shows the following structure:

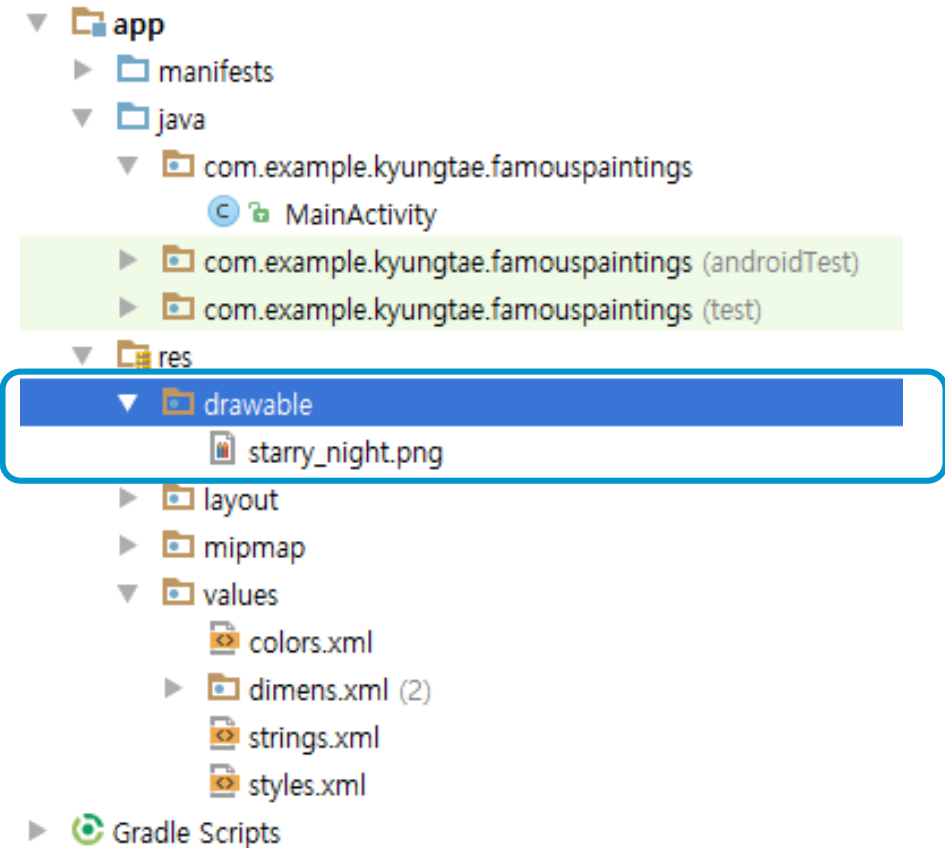
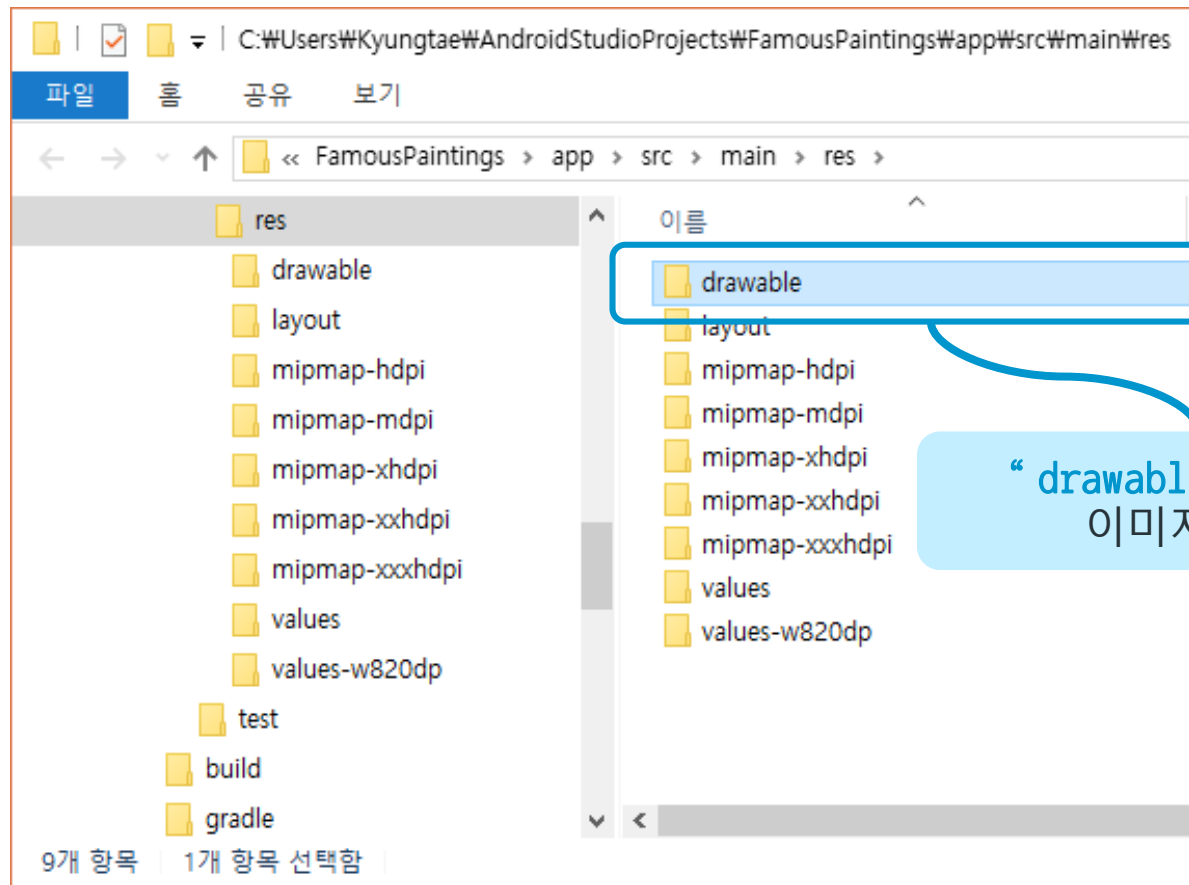
- app
  - manifests
  - java
    - com.example.kyungtae.famouspaintings
      - MainActivity
      - com.example.kyungtae.famouspaintings (androidTest)
      - com.example.kyungtae.famouspaintings (test)
    - res
      - drawable** (highlighted with a blue box)
      - layout
      - mipmap
      - values
        - colors.xml
        - dimens.xml (2)
        - strings.xml
        - styles.xml
    - Gradle Scripts

A blue callout box with the text "팝업 메뉴 띄우기" (Show context menu) has an arrow pointing to the 'drawable' folder in the project tree.

The context menu is open, showing various options. A blue callout box with the text "윈도우 탐색기 창보기 (탐색기에서 이미지 추가)" (Show in Explorer (add image in Explorer)) has an arrow pointing to the "Show in Explorer" option in the context menu.

The context menu options include:

- New
- Link C++ Project with Gradle
- Cut (Ctrl+X)
- Copy (Ctrl+C)
- Copy Path (Ctrl+Shift+C)
- Copy as Plain Text
- Copy Reference (Ctrl+Alt+Shift+C)
- Paste (Ctrl+V)
- Find Usages (Alt+F7)
- Find in Path... (Ctrl+Shift+F)
- Replace in Path... (Ctrl+Shift+R)
- Analyze
- Refactor
- Add to Favorites
- Show Image Thumbnails (Ctrl+Shift+T)
- Reformat Code (Ctrl+Alt+L)
- Optimize Imports (Ctrl+Alt+O)
- Delete... (Delete)
- Run 'Tests in 'drawable'' (Ctrl+Shift+F10)
- Debug 'Tests in 'drawable''
- Run 'Tests in 'drawable'' with Coverage
- Create 'Tests in 'drawable''...
- Local History
- Synchronize 'drawable'
- Show in Explorer** (highlighted with a blue box)
- File Path (Ctrl+Alt+F12)
- Compare With... (Ctrl+D)
- Create Gist...



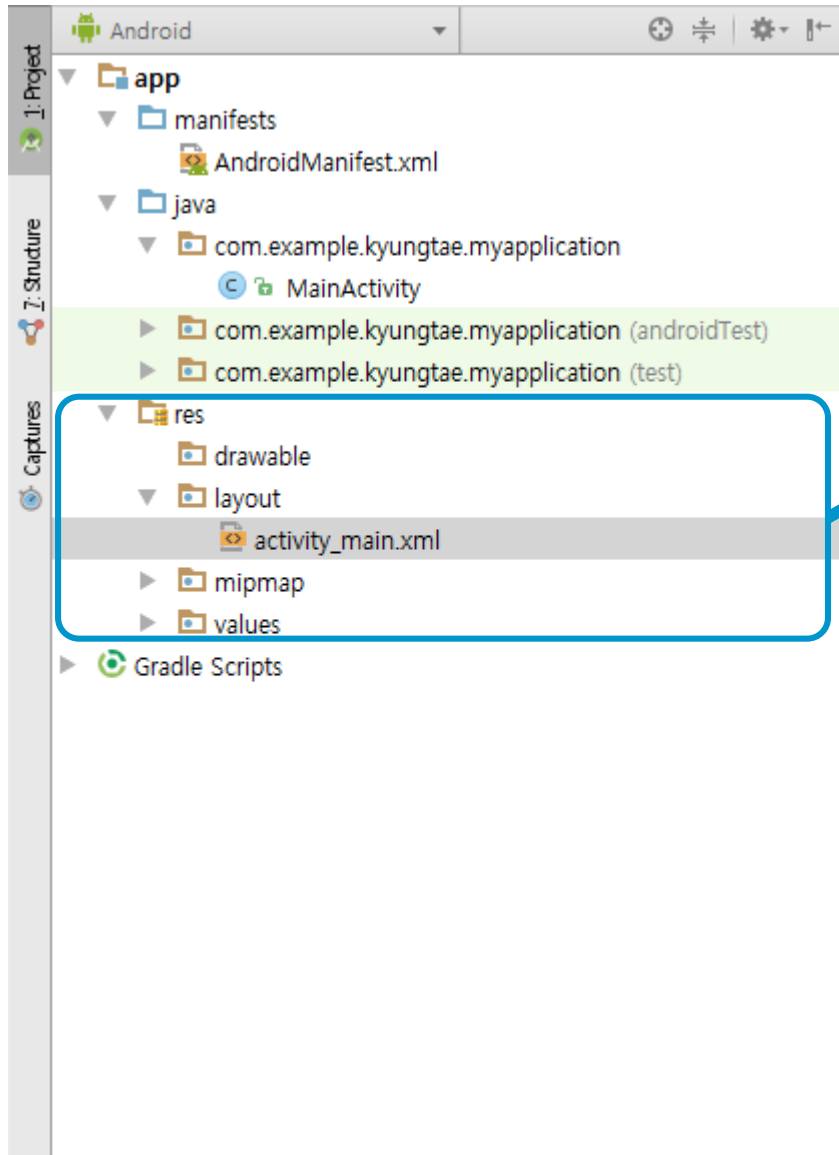
# Step 2.2 텍스트 자원의 편집

- strings.xml

```
1 <resources>
2   <string name="app_name">영화</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 다윈 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 굵이치는 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>
```



# Drawable Resource



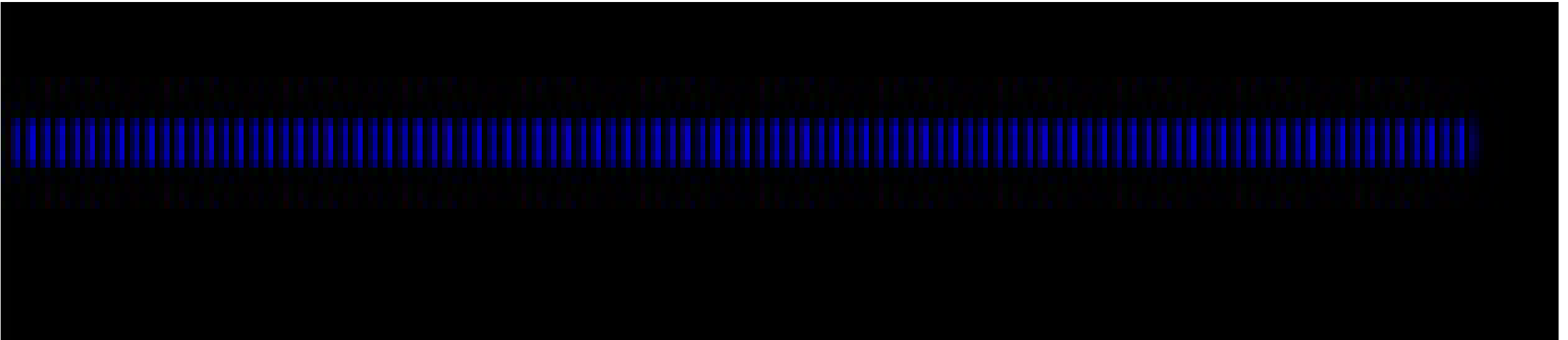
모듈에 사용되는 리소스 파일 관리  
(화면 레이아웃 설계, 텍스트와 이미지 리소스 등)

# Drawable Resource

- 화면에 출력하는 그래픽을 위한 일반적인 개념으로 여러 유형이 존재한다.
- **Shape Drawable**은 색상, 테두리, 그라데이션을 넣거나 가장자리를 둥글게 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(출처: <http://aroundck.tistory.com/140>)

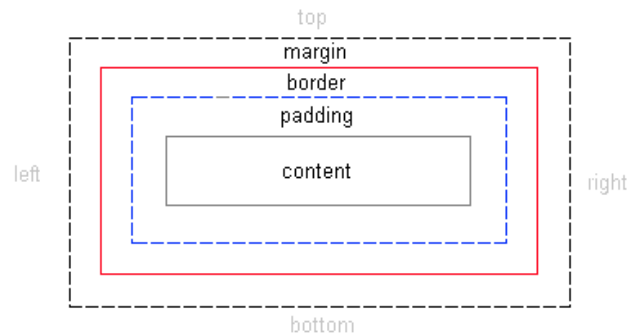
```
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="line">
    <stroke
        android:width="5dp"
        android:color="#FF0000FF"
        android:dashWidth="1dp"
        android:dashGap="2dp" />
</shape>
```



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

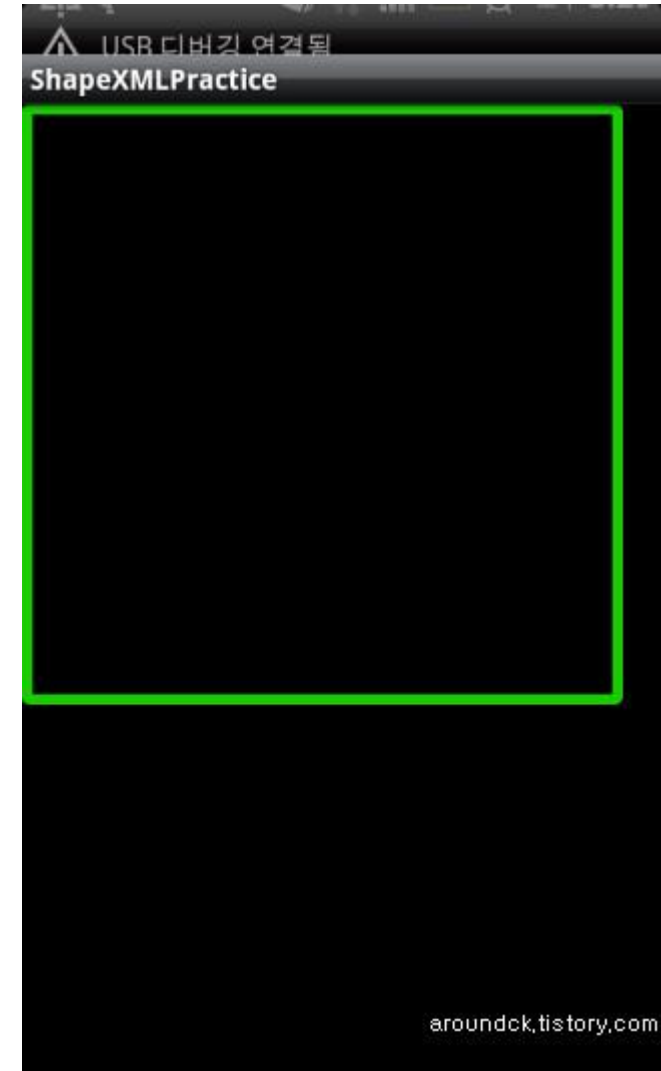
19

```
<?xml version="1.0" encoding="UTF-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <stroke
        android:width="5dip"
        android:color="#19CD00" />
    <corners
        android:bottomRightRadius="1dip"
        android:bottomLeftRadius="1dip"
        android:topLeftRadius="1dip"
        android:topRightRadius="1dip"/>
    <padding
        android:left="1dip"
        android:top="1dip"
        android:right="1dip"
        android:bottom="1dip" />
</shape>
```



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

Padding은 content와 border 사이의 공간

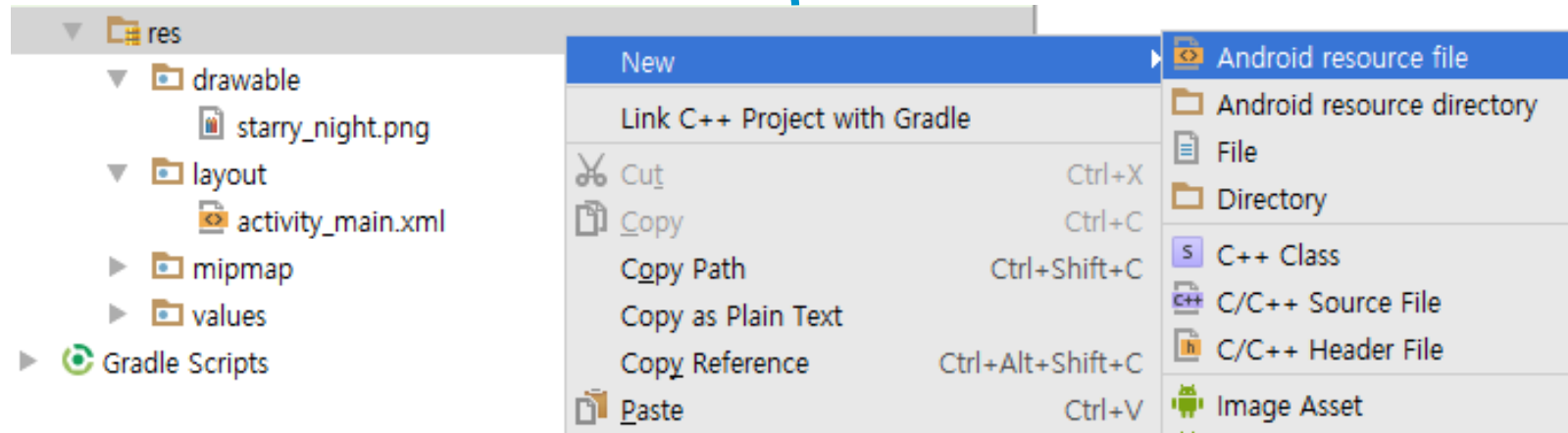


- **shape\_title.xml** 생성(res/drawable 폴더)

- 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

New Resource File

File name: shape\_title.xml ↑↓

Resource type: Drawable ▼

Root element: shape

Source set: main ▼

Directory name: drawable

Available qualifiers:

- Country Code
- Network Code
- Locale
- Layout Direction
- Smallest Screen Width
- Screen Width
- Screen Height
- Size
- Ratio
- Orientation

Chosen qualifiers:

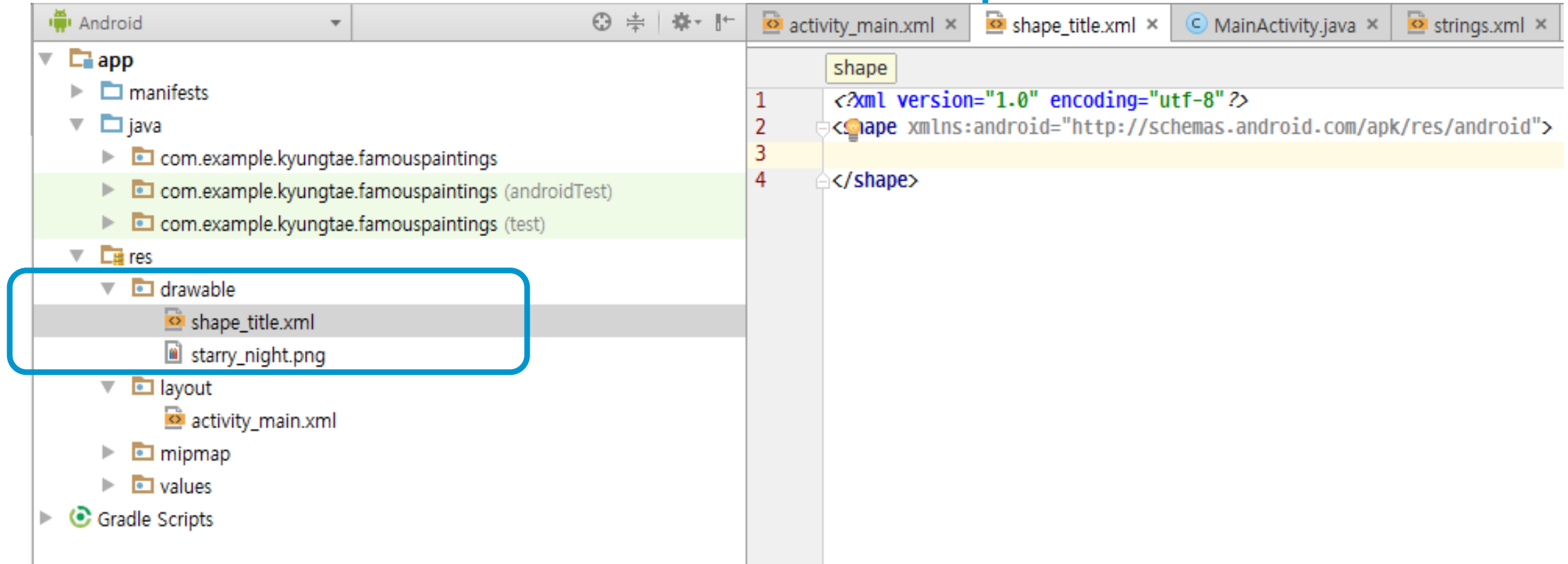
Nothing to show

>> <<

OK Cancel Help

- 생성된 shape\_title.xml

shape\_title.xml 파일





## • shape\_title.xml 소스

The image shows the Android Studio IDE with the `shape_title.xml` file open. The XML code is as follows:

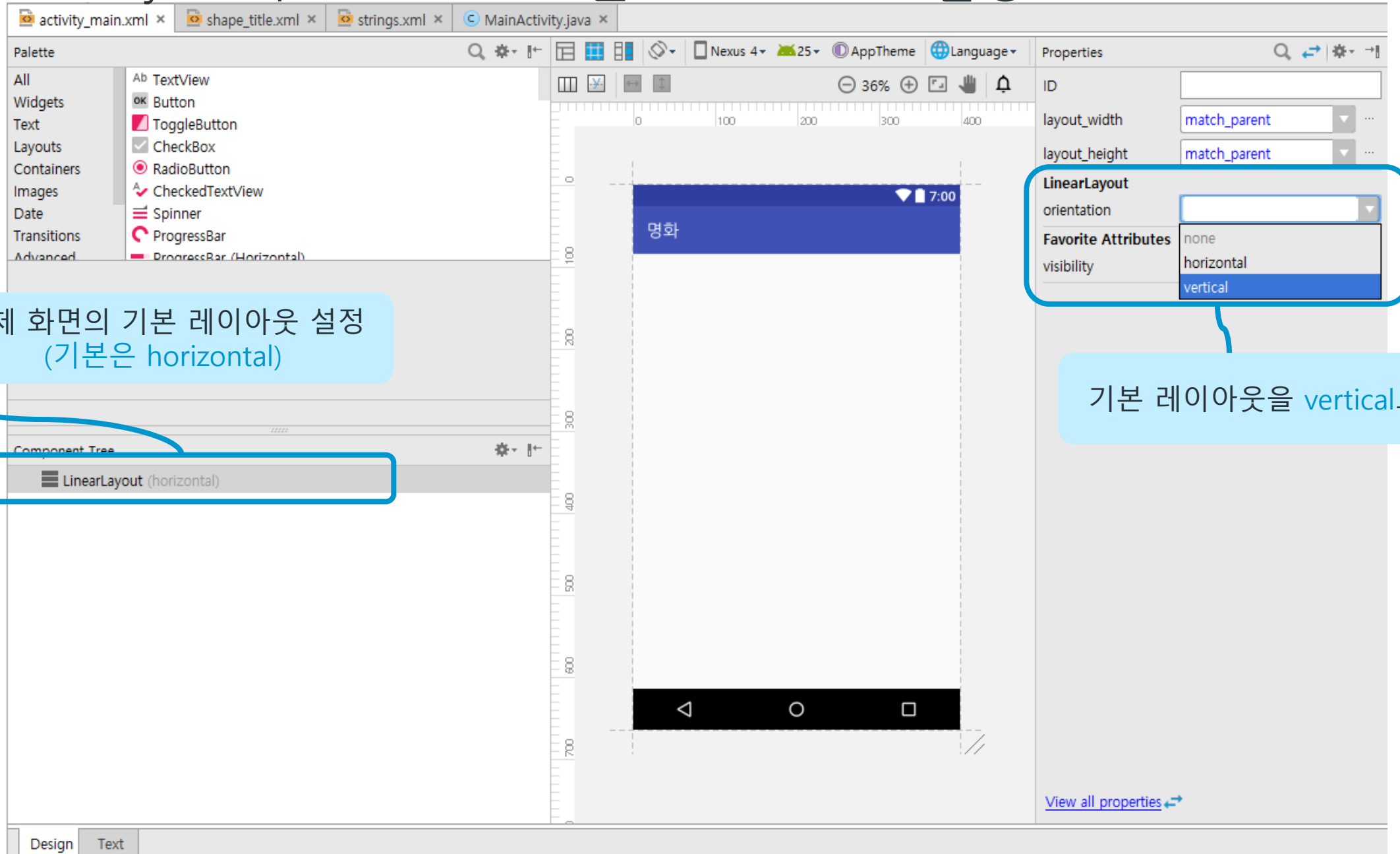
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <shape xmlns:android="http://schemas.android.com/apk/res/android"
3     android:shape="rectangle">
4
5     <solid android:color="#3061380B"/>
6
7     <padding
8         android:top="2dp"
9         android:bottom="2dp"
10        android:left="10dp"
11        android:right="10dp">
12     </padding>
13     <corners android:radius="5dp">
14     </corners>
15
16 </shape>
```

Callouts explaining the XML attributes:

- `android:shape="rectangle"`: 출력모양을 사각형으로 지정 (Specify the output shape as a rectangle)
- `android:color="#3061380B"`: 출력모양을 내부의 색 (Specify the output shape's internal color)
- `padding` (with `android:top`, `android:bottom`, `android:left`, `android:right`): 내부 패딩 정보 (Internal padding information)
- `android:radius="5dp"`: 출력모양 모서리를 둥근 모양으로 지정(반지름은 5dp) (Specify the output shape's corners as rounded (radius is 5dp))

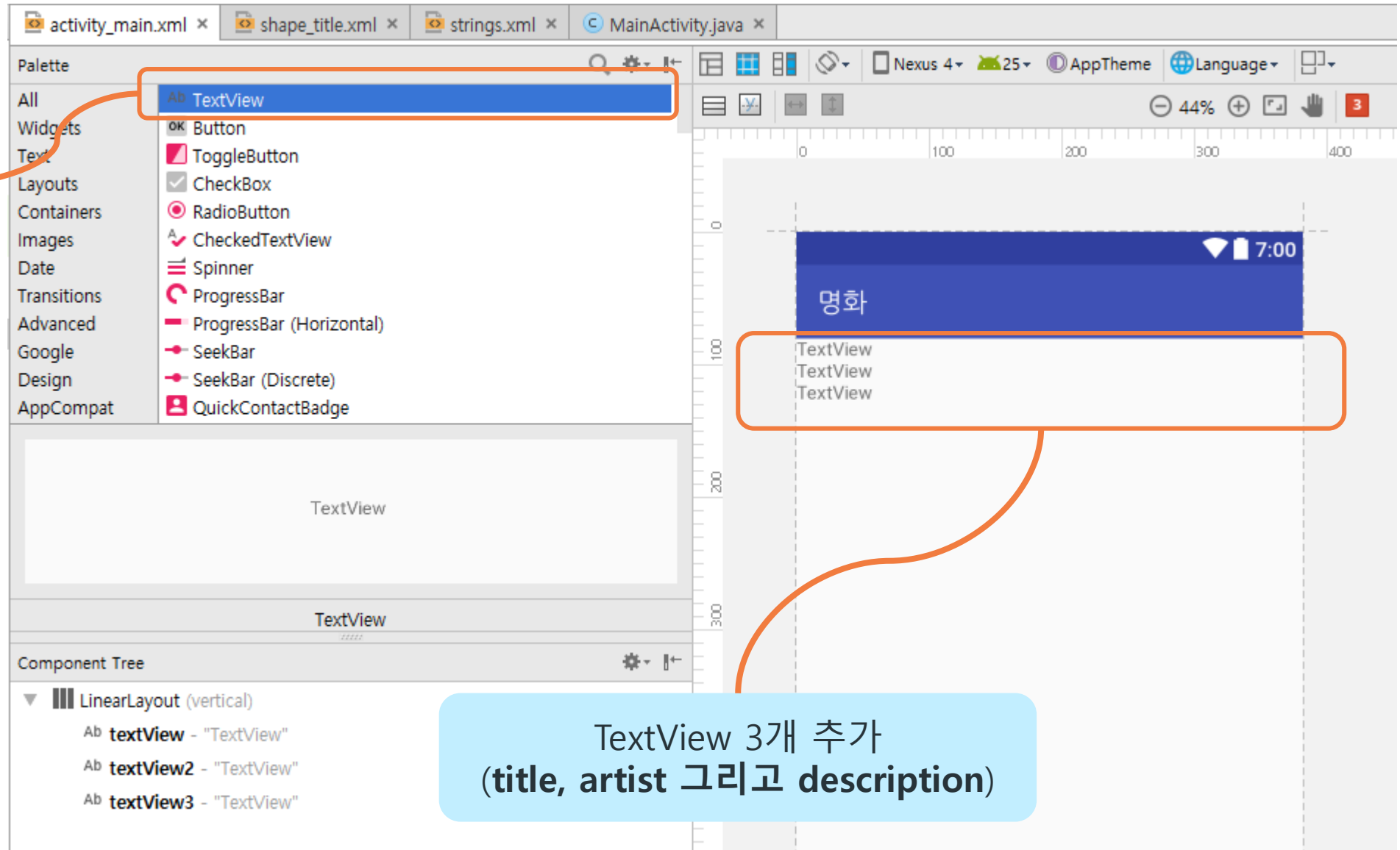
The right side of the image shows the Preview window, which displays a light gray rounded rectangle, matching the visual representation of the XML code.

- LinearLayout의 **orientation**을 **vertical**로 설정



- LinearLayout에 **TextView**와 ImageView 추가하기

Palette에서 TextView 선택



# • LinearLayout에 TextView와 **ImageView** 추가하기

Palette에서 **Image** 폴더에서 **ImageView** 선택

ImageButton  
ImageView  
videoview

Text  
Layouts  
Containers  
Images  
Date  
Transitions  
Advanced  
Google  
Design  
AppCompat

ImageView를 2번째와 3번째 **TextView** 사이에 추가

imageView

LinearLayout (vertical)  
Ab textView - "TextView"  
Ab textView2 - "TextView"  
Ab textView3 - "TextView"

이미지 추가를 위한 팝업창 열림

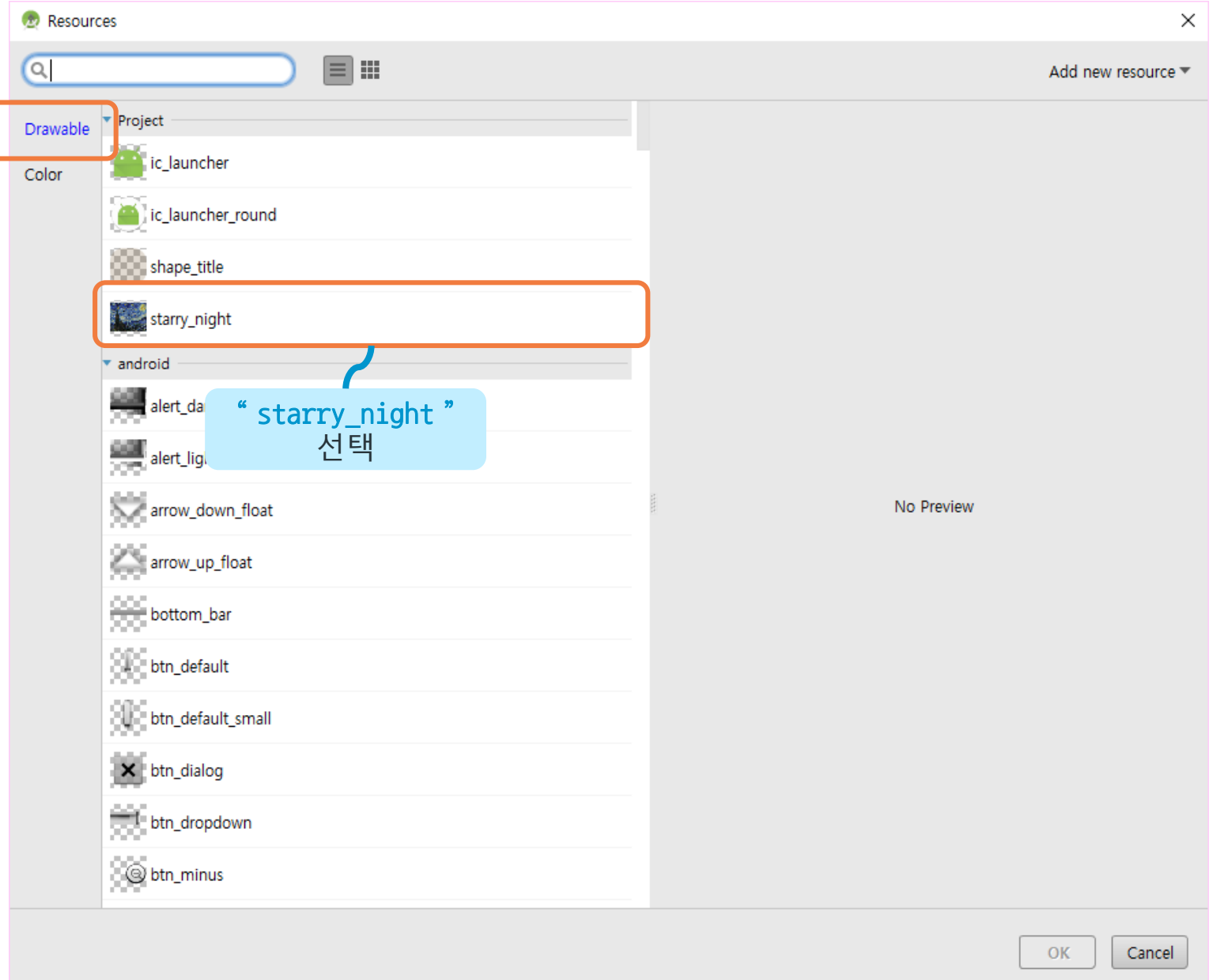
textView  
textView  
textView

영화

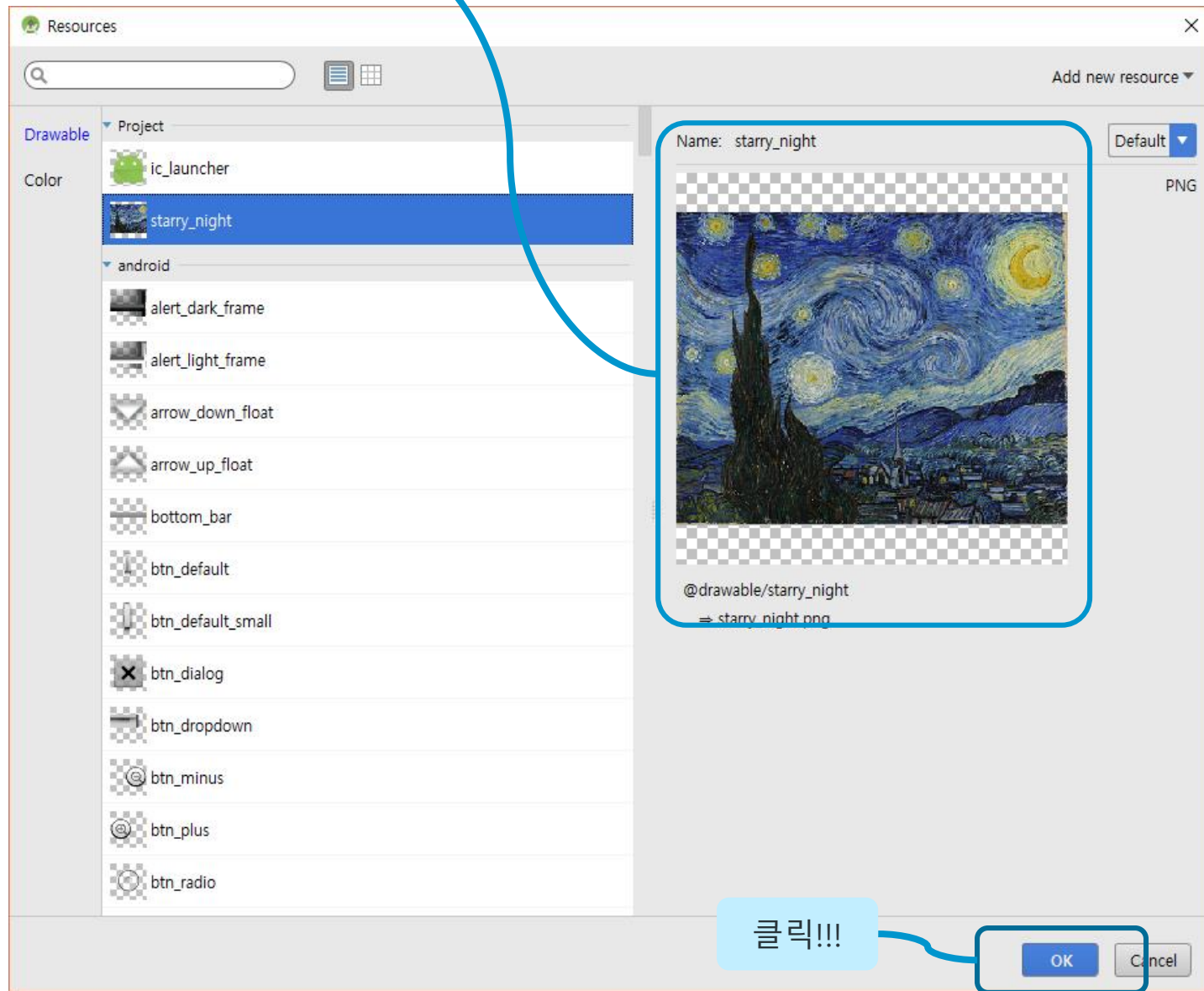
7:00

- ImageView에 이미지 소스 선택 팝업 창

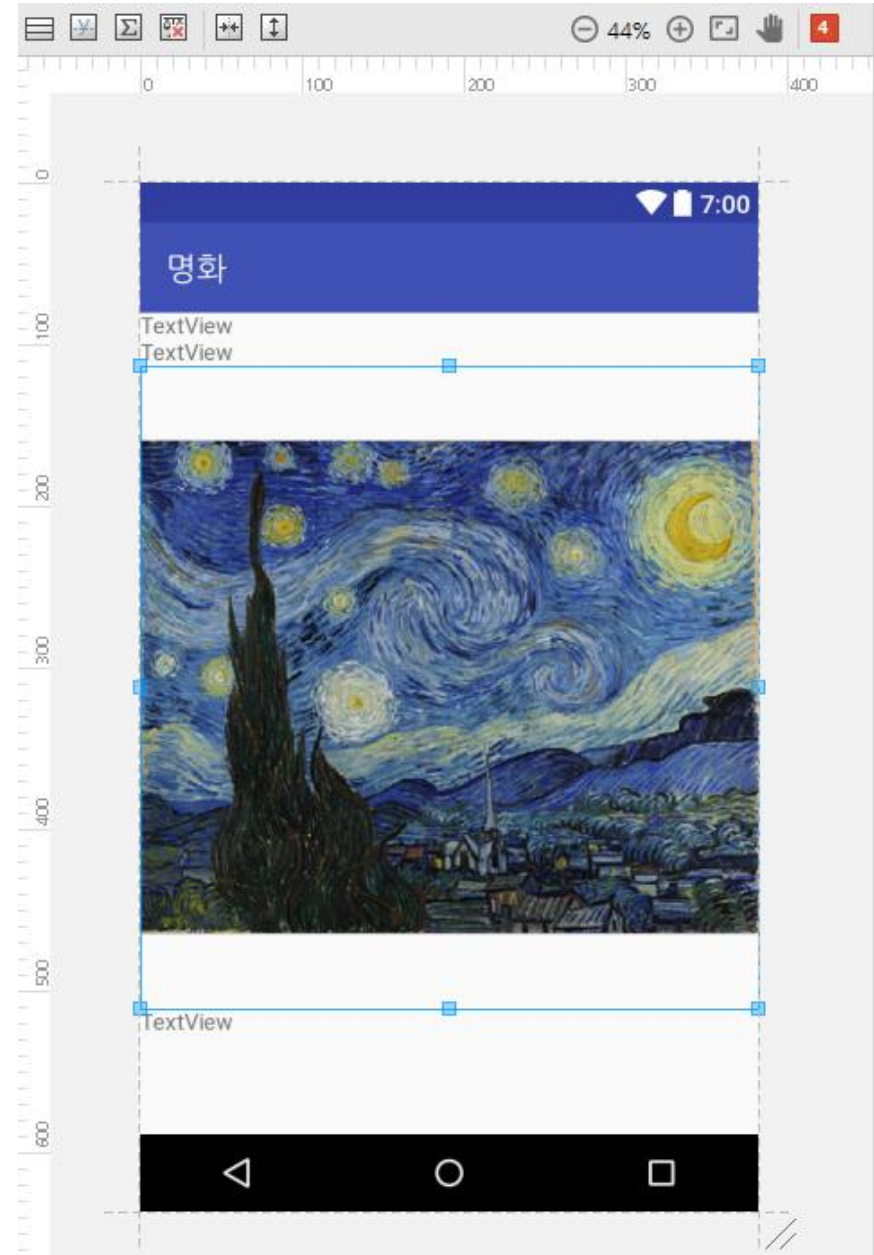
리소스 폴더 인  
"Drawable" 선택



선택한 이미지  
미리보기



완성된 화면





# • Title TextView의 “Text” 속성 변경하기

The screenshot shows the Android Studio IDE with the following components:

- Palette:** A list of UI widgets including ImageButton, ImageView, VideoView, and TextView. The 'TextView' widget is highlighted.
- Design View:** A visual representation of the app's layout. It shows a blue header bar with the text '영화' (Movie) and a large image of 'The Starry Night' painting below it. A blue selection box is around the '영화' text.
- Component Tree:** A hierarchical view of the UI components. It shows a 'LinearLayout (vertical)' containing four items: 'textView - "TextView"', 'textView2 - "TextView"', 'imageView2', and 'textView3 - "TextView"'. The 'textView - "TextView"' item is selected and highlighted with a blue box.
- Properties Panel:** A list of properties for the selected 'textView'. The 'text' property is highlighted with a blue box. A blue callout bubble points to this property with the text: '해당 속성을 선택 후 값을 변경→ title01' (Select this property and change the value to title01).

속성 변경을 위한  
컴포넌트 요소 선택



Properties

id	
layout_width	wrap_content
layout_height	wrap_content
Layout_Margin	[?, ?, ?, ?]
Padding	[?, ?, ?, ?]
Theme	
elevation	
text	Hello World!
accessibilityLiveRegion	
accessibilityTraversalAfter	
accessibilityTraversalBefore	
allowUndo	[-]
alpha	
autoLink	[ ]
autoText	[-]
background	
backgroundTint	
backgroundTintMode	
breakStrategy	high_quality
bufferType	
capitalize	
clickable	[-]
contentDescription	
contextClickable	[-]
cursorVisible	[-]
digits	
drawableBottom	

Click to pick a resource

속성 변경을 위한  
컴포넌트 요소 선택

Resources

사용할 리소스 선택

Add new resource ▾

Name: title01

Default Starry Night

Project	
app_name	명화
artist01	고흐
desc01	빈센트 반 고흐의 대표작 중 하나...
title01	Starry Night
android	
VideoView_error_button	OK
VideoView_error_text_invalid_progres	This video isn't valid for streamin...
VideoView_error_text_unknown	Can't play this video.
VideoView_error_title	Video problem
cancel	Cancel
copy	Copy
copyUrl	Copy URL
cut	Cut
defaultMsisdnAlphaTag	MSISDN1
defaultVoiceMailAlphaTag	Voicemail
dialog_alert_title	Attention
emptyPhoneNumber	(No phone number)
fingerprint_icon_content_description	Fingerprint icon
httpErrorBadUrl	Couldn't open the page because t...
httpErrorUnsupportedScheme	The protocol isn't supported.

[EDIT TRANSLATIONS](#)

OK Cancel

# • title TextView의 “Text” 속성 변경 완료

The screenshot displays the Android Studio IDE with the following components:

- Palette:** Shows various widget categories. Under the "Images" category, "ImageButton", "ImageView", and "VideoView" are listed.
- Component Tree:** Located at the bottom left, it shows the hierarchy of UI components. A `TextView` with the text `"@string/title01"` is highlighted with a blue box.
- Design View:** The central area shows a visual representation of the app's layout. It features a blue header bar with the text "영화" (Movie) and "Starry Night" below it. Below the header is a large image of the "Starry Night" painting. The text "Starry Night" is highlighted with a blue box.
- Properties Panel:** Located on the right, it shows the properties of the selected `TextView`. The `text` property is highlighted with a blue box and set to `@string/title01`.

The `Properties` panel lists various attributes for the `TextView` widget, including:

- `id`: `textView`
- `layout_width`: `match_parent`
- `layout_height`: `wrap_content`
- `Layout_Margin`: `[?, ?, ?, ?]`
- `Padding`: `[?, ?, ?, ?]`
- `Theme`: `elevation`
- `text`: `@string/title01` (highlighted)
- `accessibilityLiveRegion`
- `accessibilityTraversable`
- `allowUndo`: `[-]`
- `alpha`
- `autoLink`: `[]`
- `autoText`: `[-]`
- `background`
- `backgroundTint`
- `backgroundTintMode`
- `breakStrategy`
- `bufferType`
- `capitalize`
- `clickable`: `[-]`
- `contentDescription`
- `contextClickable`: `[-]`
- `cursorVisible`: `[-]`
- `digits`
- `drawableBottom`
- `drawableEnd`

# • title 컴포넌트의 기타 속성 변경하기

The screenshot displays the Android Studio interface. On the left, the design view shows a mobile app layout with a blue header bar containing the text '영화' (Movie). Below the header, a TextView component is highlighted with an orange border, displaying the text 'Starry Night'. Below the TextView is an image of the 'Starry Night' painting. On the right, the 'Properties' panel lists the attributes of the selected TextView. Arrows point from specific properties in the panel to callout boxes on the right side of the image.

id	textView
layout_width	match_parent
layout_height	wrap_content
Layout_Margin	[?, ?, ?, ?, ?]
Padding	[?, ?, ?, ?, ?]
Theme	
elevation	
gravity	[center]
text	@string/title01
textColor	#613808
textSize	18sp
textStyle	[bold]
accessibilityLiveReg	
accessibilityTraversa	
accessibilityTraversa	
allowUndo	<input type="checkbox"/>
alpha	

layout\_width: match\_parent

layout\_height: wrap\_content

gravity: center

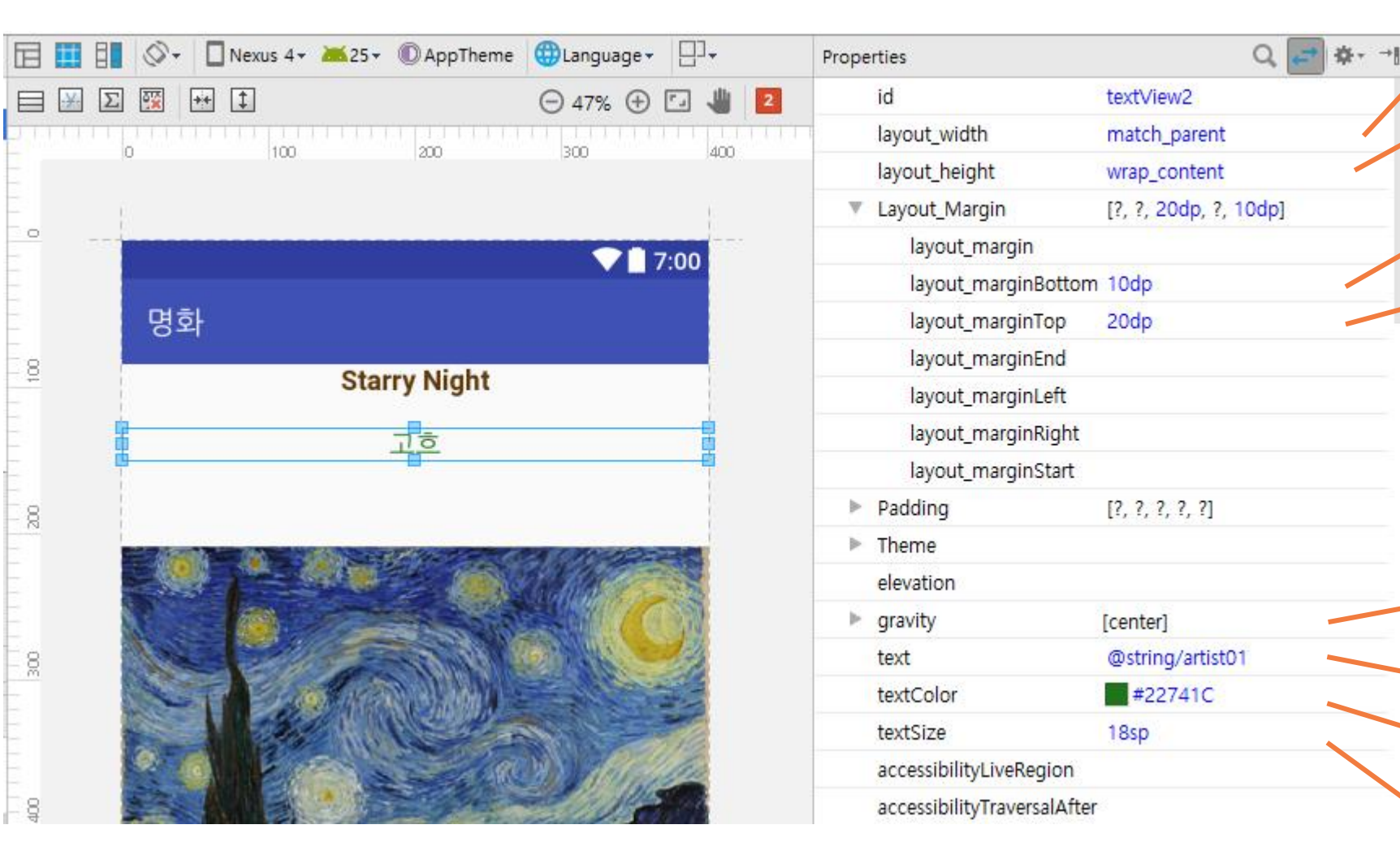
text: @string/title01

textColor: #613808

textSize: 18sp

textStyle: bold

# • artist TextView의 속성 변경



The screenshot displays the Android Studio interface. On the left, the design view shows a mobile app layout with a blue header bar containing the text '영화' (Movie) and a status bar at the top showing '7:00'. Below the header, the text 'Starry Night' is displayed in a large, bold, brown font. At the bottom, there is a reproduction of the 'Starry Night' painting. On the right, the 'Properties' panel for the selected TextView widget (textView2) is visible. The properties are listed in a table-like format. To the right of the Properties panel, several blue callout boxes with orange arrows point to specific properties, highlighting the changes being made.

Property	Value
id	textView2
layout_width	match_parent
layout_height	wrap_content
Layout_Margin	[?, ?, 20dp, ?, 10dp]
layout_margin	
layout_marginBottom	10dp
layout_marginTop	20dp
layout_marginEnd	
layout_marginLeft	
layout_marginRight	
layout_marginStart	
Padding	[?, ?, ?, ?, ?]
Theme	
elevation	
gravity	[center]
text	@string/artist01
textColor	#22741C
textSize	18sp
accessibilityLiveRegion	
accessibilityTraversalAfter	

- layout\_width: match\_parent
- layout\_height: wrap\_content
- Layout\_marginBottom: 10dp
- Layout\_marginTop: 20dp
- gravity: center
- text: @string/artist01
- textColor: #22741C
- textSize: 18sp



# • 이미지의 ImageView의 속성 변경

The screenshot shows the Android Studio interface with a movie poster design. The poster includes a blue header with the text '영화' (Movie) and '7:00', a title 'Starry Night' in brown, a subtitle '고흐' (Gogh) in green, and a central image of 'The Starry Night' by Van Gogh. Below the image is a 'TextView' label.

The Properties panel for 'imageView2' is shown on the right, with the following attributes:

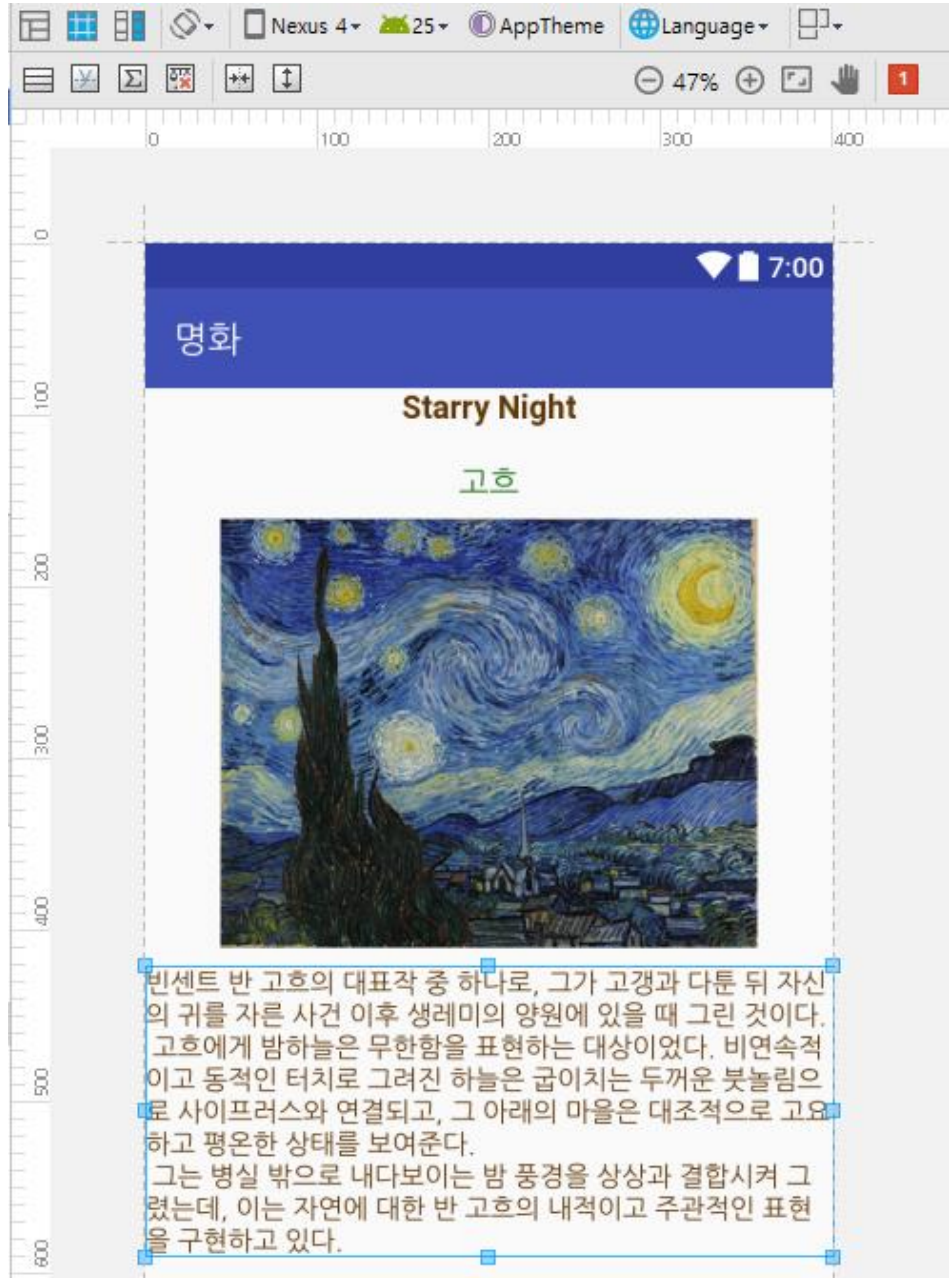
- id: imageView2
- layout\_width: wrap\_content
- layout\_height: wrap\_content
- Layout\_Margin: [?, ?, ?, ?, 10dp]
- layout\_marginBottom: 10dp
- Padding: [?, ?, ?, ?, ?]
- Theme: [?]
- elevation: [?]
- adjustViewBounds: ☒
- layout\_gravity: [center]
- maxHeight: 300dp
- maxWidth: 300dp
- srcCompat: @drawable/starry\_night
- accessibilityLiveRegion: [?]
- accessibilityTraversalAfter: [?]
- accessibilityTraversalBefore: [?]
- alpha: [?]

Callout boxes on the right highlight the following attributes:

- layout\_width: wrap\_content
- layout\_height: wrap\_content
- Layout\_marginBottom: 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



Properties	
id	textView3
layout_width	match_parent
layout_height	wrap_content
Layout_Margin	[?, ?, ?, ?]
Padding	[?, ?, ?, ?]
Theme	
elevation	
text	@string/desc01
textColor	#61380B
textSize	15sp
accessibilityLiveRegion	
accessibilityTraversalAfter	
accessibilityTraversalBefore	
allowUndo	<input type="checkbox"/>
alpha	
autoLink	<input type="checkbox"/>
autoText	<input type="checkbox"/>
background	
backgroundTint	
backgroundTintMode	
breakStrategy	
bufferType	
capitalize	
clickable	<input type="checkbox"/>
contentDescription	
contextClickable	<input type="checkbox"/>
cursorVisible	<input type="checkbox"/>
digits	

layout\_width: match\_parent

layout\_height: wrap\_content

text: @string/artist01

textColor: #61380B

textSize: 15sp

# • shape\_title.xml을 이용한 제목바의 배경 처리

The screenshot displays the Android Studio IDE with the following components:

- Palette:** Located on the top left, it shows various widget categories. The 'Images' category is selected, and 'TextView' is highlighted.
- Component Tree:** Located on the bottom left, it shows the hierarchy of UI components. The 'textView - "@string/title01"' is highlighted.
- Design Canvas:** The central area shows a visual representation of the layout. It features a blue header bar with the text '영화' (Movie) and 'Starry Night'. Below this is a reproduction of the painting 'Rain, Steam, and Great Smokes' by J.M.W. Turner, labeled '고흐' (Gogh). Below the painting is a paragraph of Korean text.
- Properties Panel:** Located on the right, it shows the properties of the selected 'textView'. The 'background' property is highlighted.

The design canvas shows a blue header bar with the text '영화' (Movie) and 'Starry Night'. Below this is a reproduction of the painting 'Rain, Steam, and Great Smokes' by J.M.W. Turner, labeled '고흐' (Gogh). Below the painting is a paragraph of Korean text.

textView - "@string/title01"

textView2 - "@string/artist01"

imageView2

textView3 - "@string/desc01"

background

backgroundTint

backgroundTintMode

breakStrategy

bufferType

capitalize

clickable

contentDescription

contextClickable

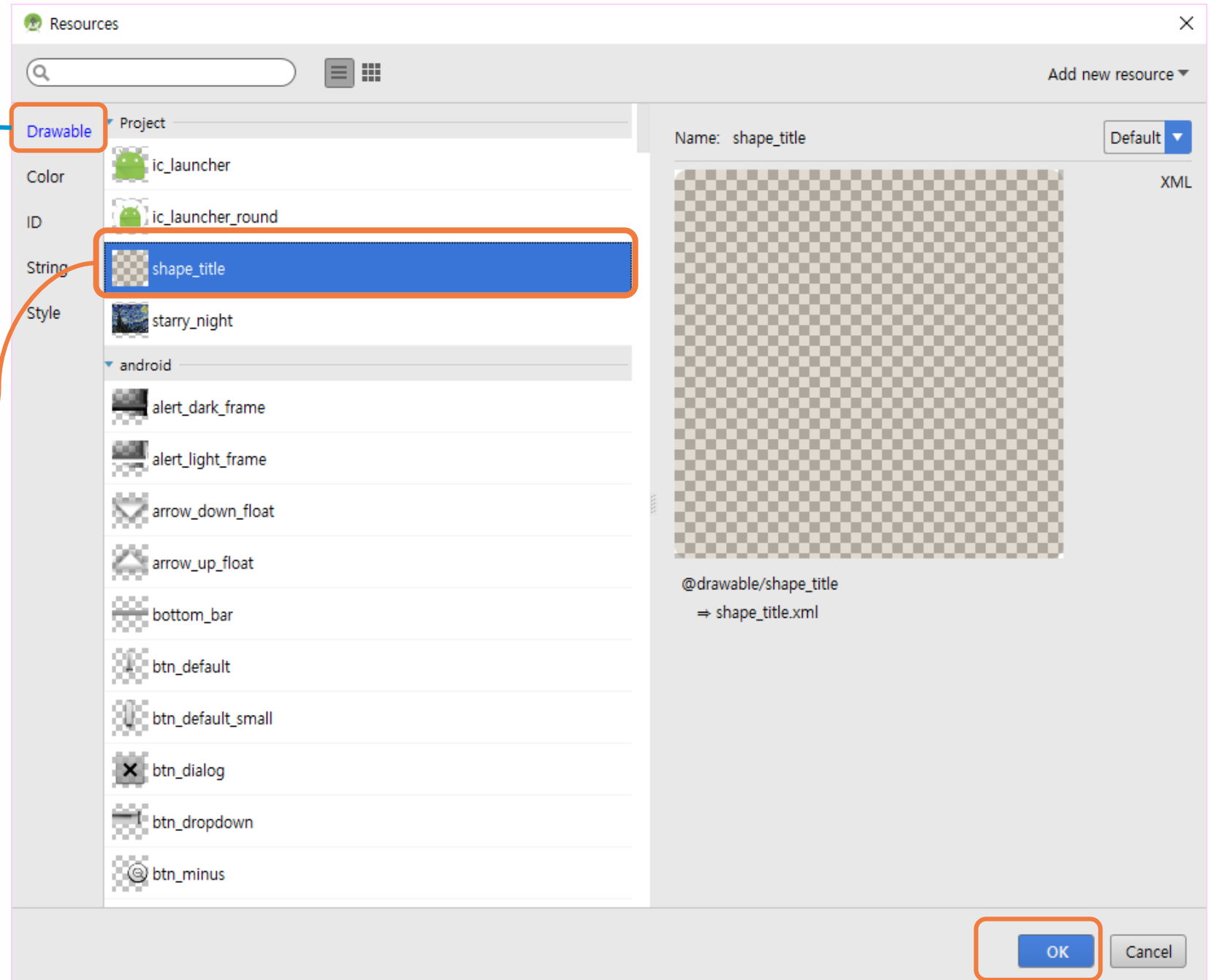
cursorVisible

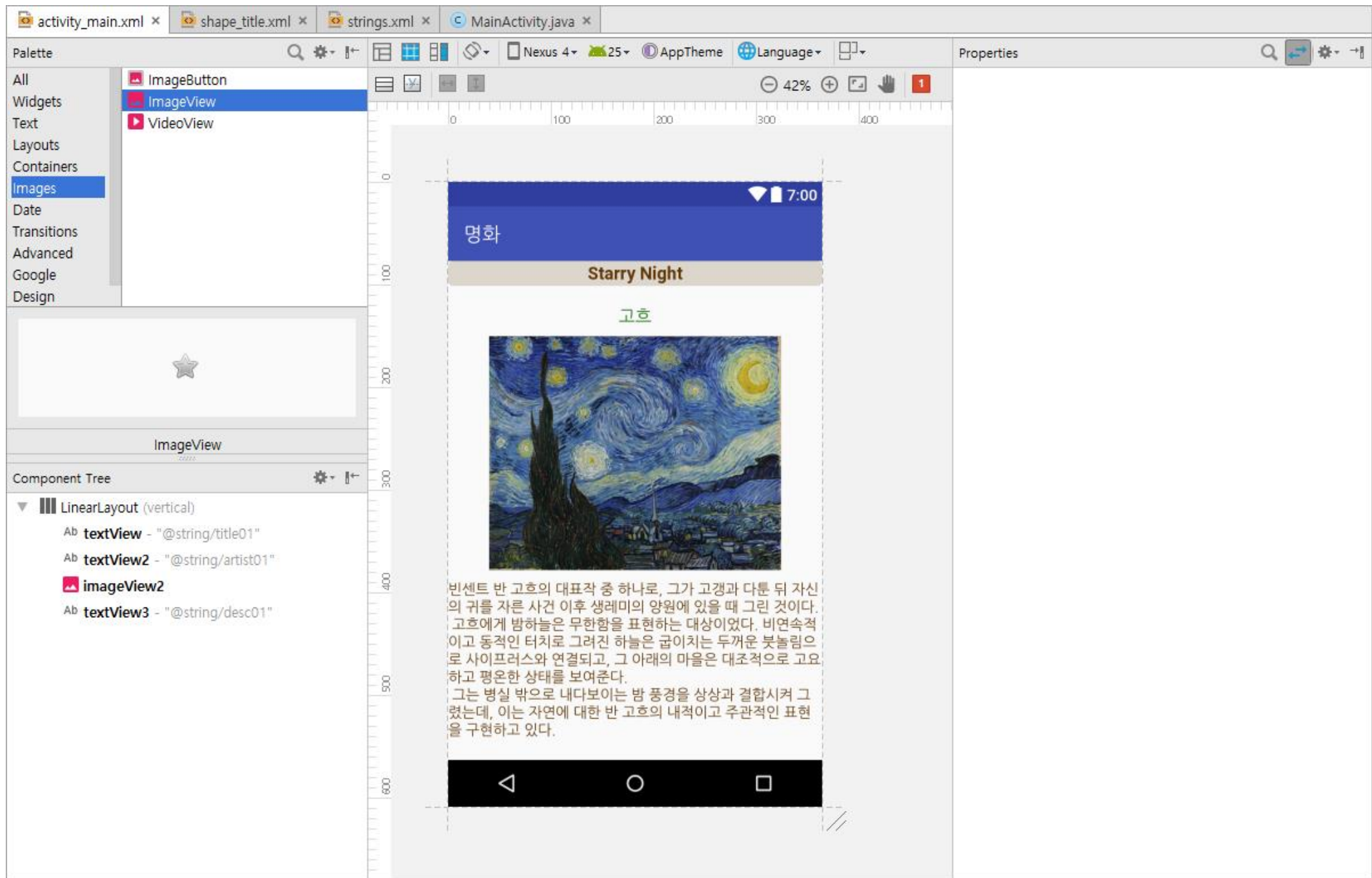
digits



Resources 종류:  
Drawable

선택





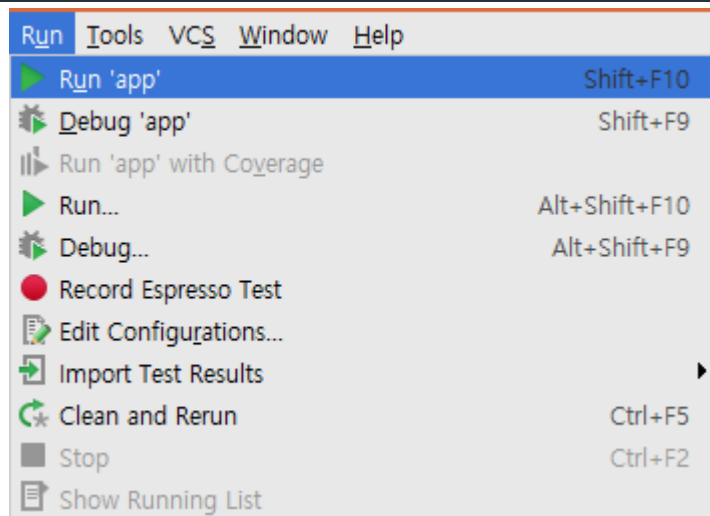
# • 화면 상하좌우에 여백주기

The screenshot displays the Android Studio IDE with the following components:

- Palette:** Shows widget categories like All, Widgets, Text, Layouts, Containers, Images, Date, Transitions, Advanced, Google, and Design. The 'Images' category is selected, showing 'ImageButton', 'ImageView', and 'VideoView'.
- Component Tree:** Shows a vertical `LinearLayout` containing:
  - `textView` - "@string/title01"
  - `textView2` - "@string/artist01"
  - `imageView2`
  - `textView3` - "@string/desc01"
- Design View:** A visual representation of the UI. It features a blue header with the text '영화', a tan title bar with 'Starry Night', a green text label '고흐', and a painting of 'The Starry Night'. Below the painting is a Korean description of the artwork.
- Properties Panel:** Lists various properties for the selected `ImageView`. The `Layout_Margin` property is highlighted with an orange box, showing a value of `[5sp, ?, ?, ?]`. The `layout_margin` property is also visible, set to `5sp`.

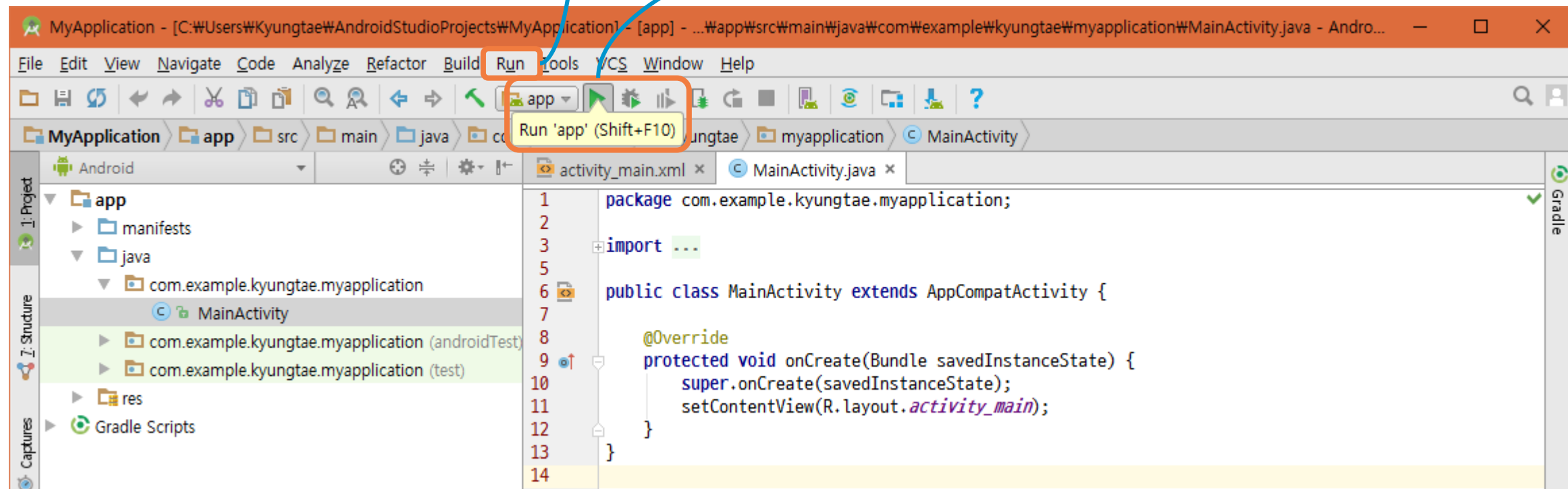
# Step 3. 프로젝트 실행

43



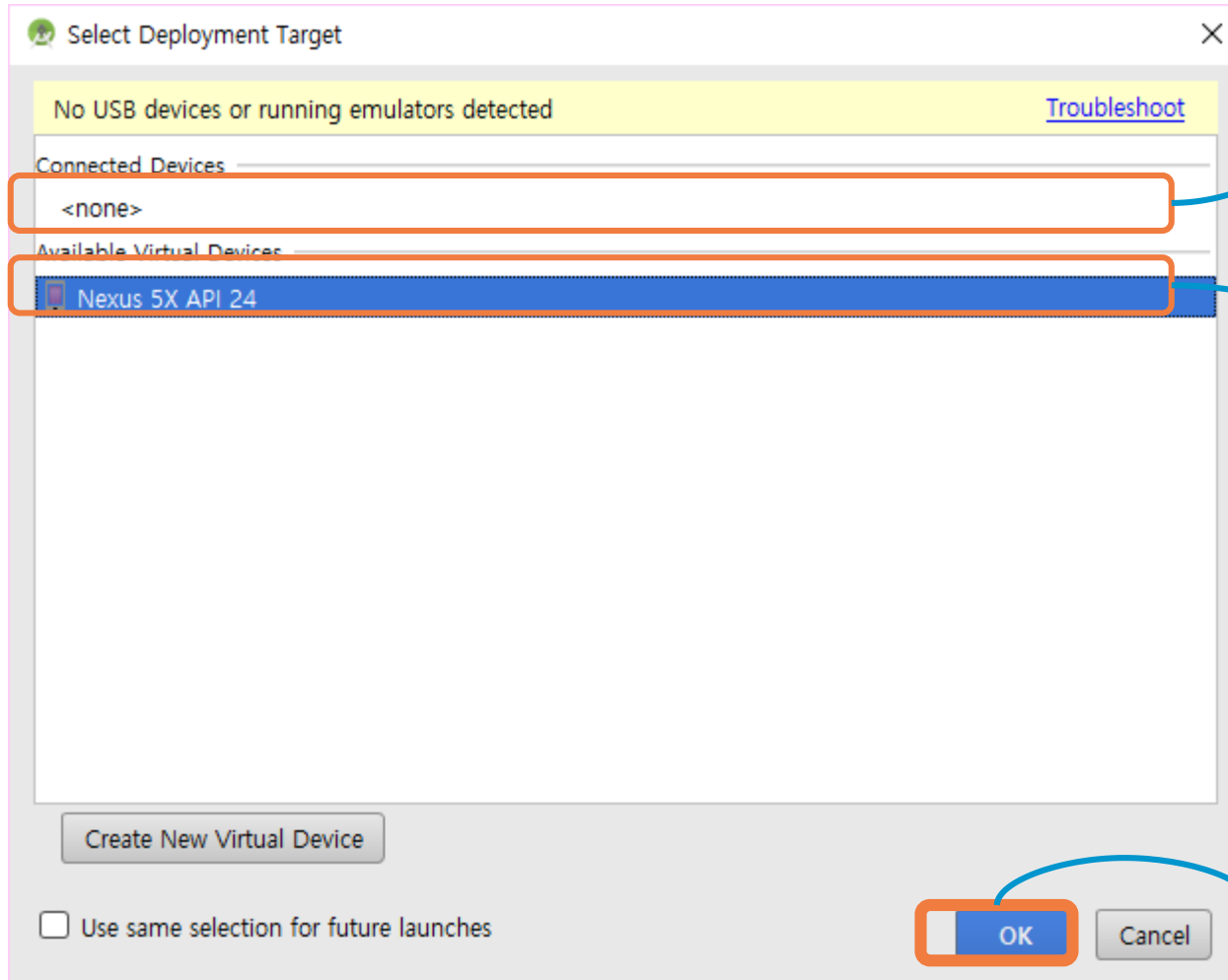
Run → Run 'app' 메뉴 클릭

앱 실행 아이콘 클릭



## • AVD 장비 선택하기

44



데이터 케이블로 연결된  
스마트폰

AVD

스마트폰 또는 AVD를 선택하고  
'OK' 버튼을 클릭



# O utputs



# Q & A uestion nswer

46

