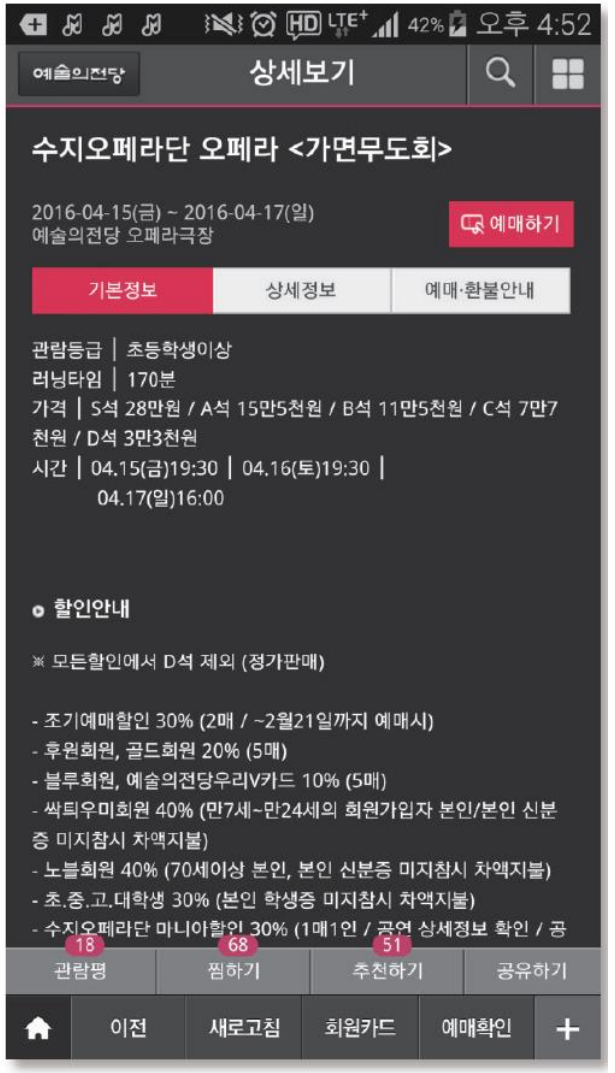
Week05.
이벤트 처리와 액티비티 간 이동

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	**이벤트 처리와 액티비티 간 이동**
6	오디오 재생
7	비디오 재생
8	**중간고사**
9	애니메이션
10	사물인터넷과 센서 – 터치 센서, 모션 센서
11	사물인터넷과 센서 – 위치 센서, 환경 센서
12	NFC 활용
13	공공 DB 오픈 API 활요
14	구글 맵과 위치 추적
15	**기말 고사**

사용자 클릭 처리 앱의 예

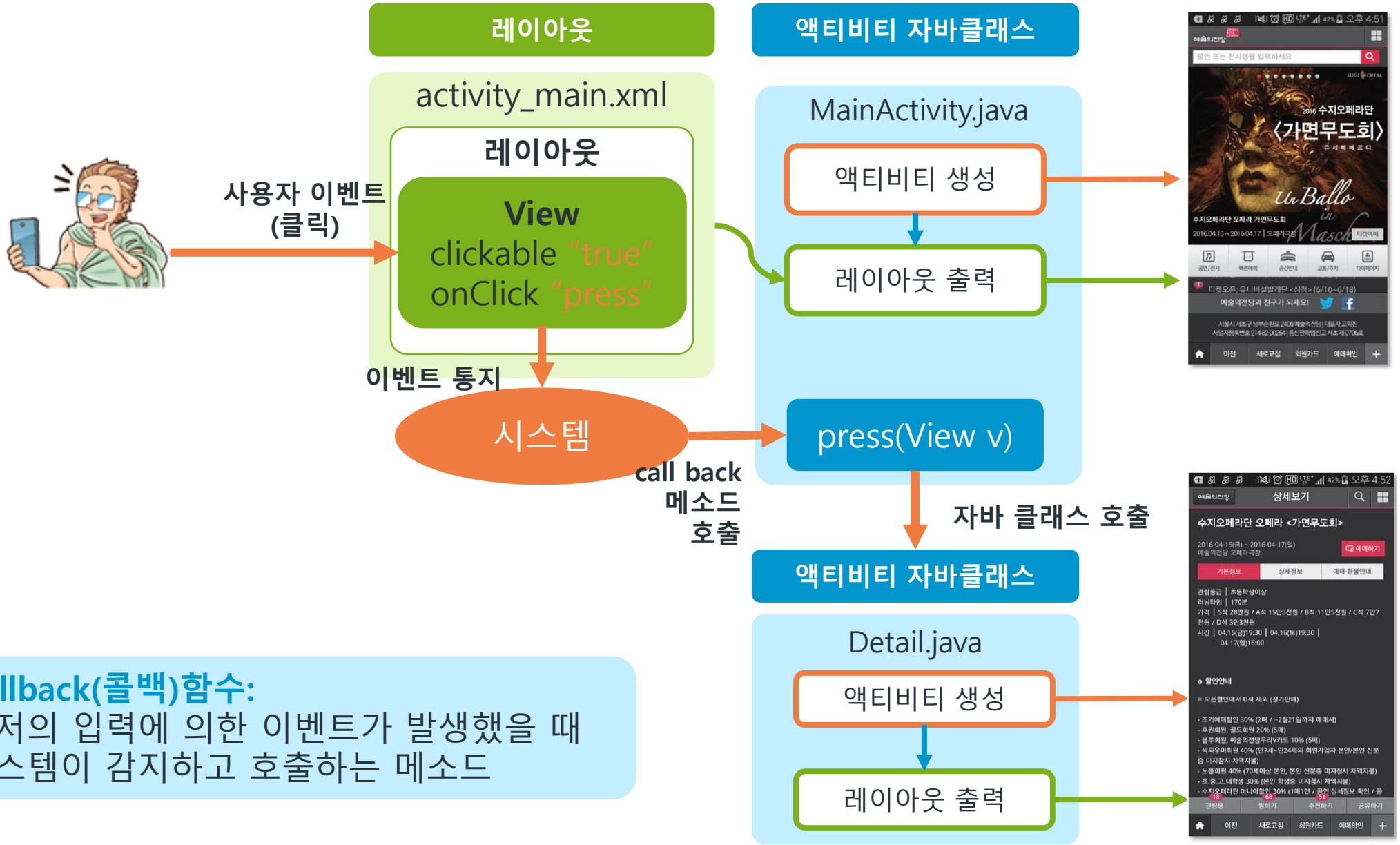
액티비티 1(초기 화면)



액티비티 2(상세 화면)

- 예술의 전당 앱

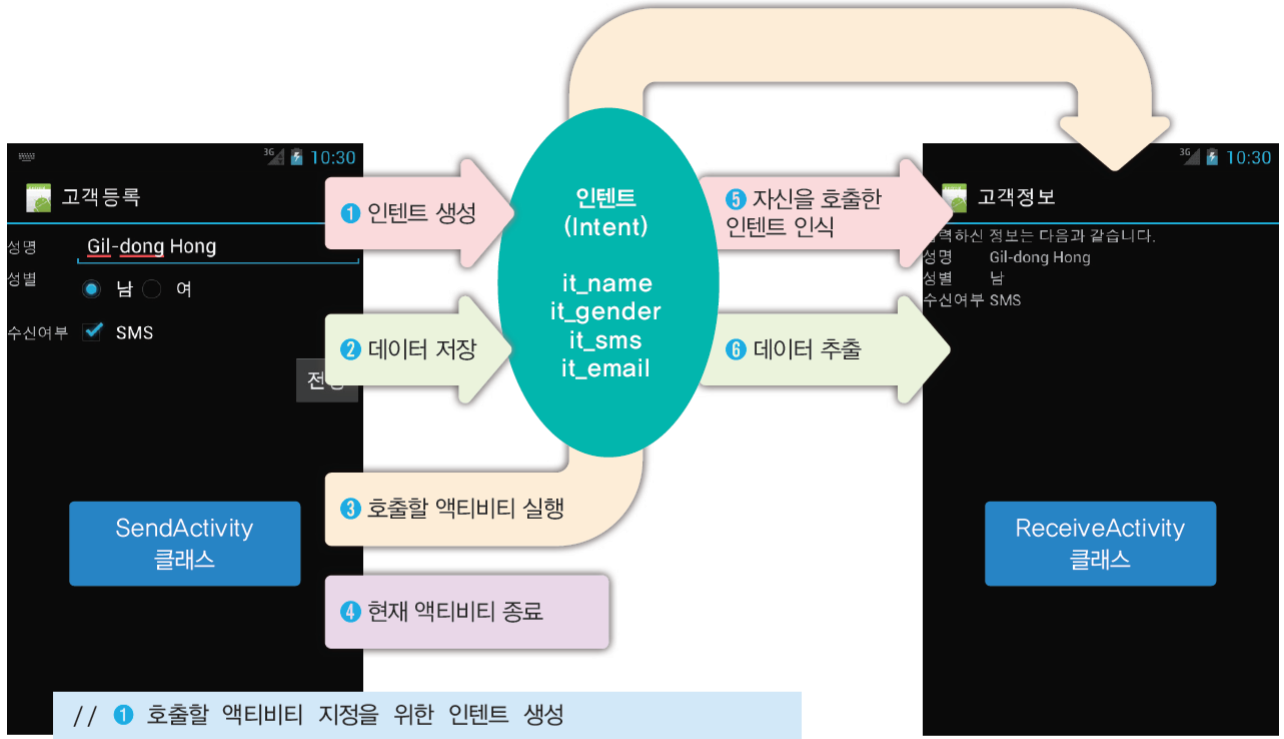
사용자 클릭 처리와 콜백 메소드



Callback(콜백)함수:
유저의 입력에 의한 이벤트가 발생했을 때
시스템이 감지하고 호출하는 메소드

액티비티 간 이동과 정보 전달 – 인텐트(Intent)


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

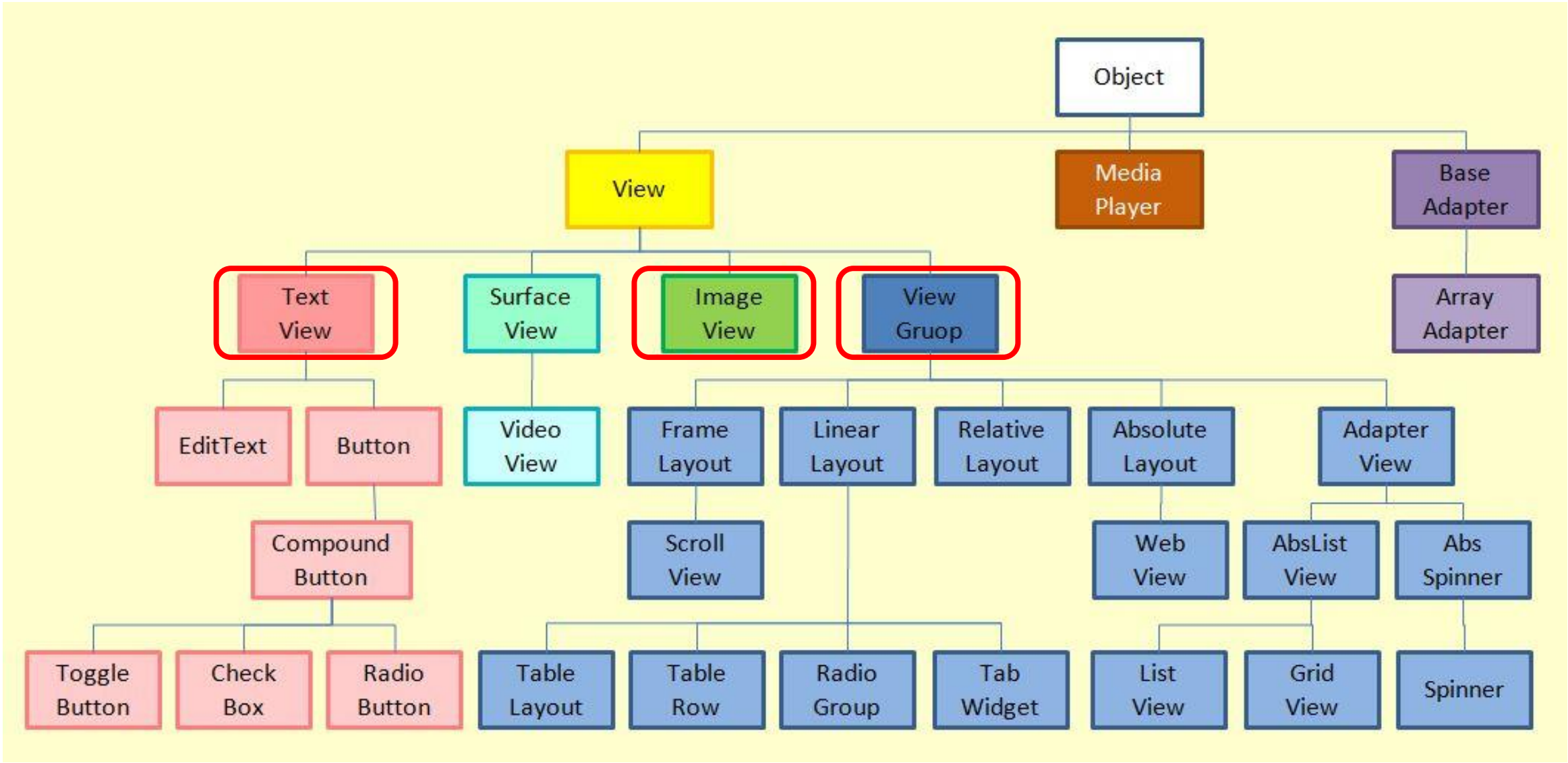



```
// ❶ 호출할 액티비티 지정을 위한 인텐트 생성
Intent it= new Intent(this, ReceiveActivity.class);
// ❷ 인텐트에 데이터 저장(성명)
// str_name의 값을 it_name 변수에 저장함
it.putExtra("it_name", str_name);
// ❸ 인텐트에서 지정한 액티비티 실행
startActivity(it);
// ❹ 현재 액티비티 종료
finish();
```

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

- **Bundle**은 상태/값 등을 저장하기 위한 객체
- **Intent**는 저장이 아닌 전달하는 수단으로의 객체

안드로이드 자바 클래스 계층도



출처: http://wiki.gurubee.net/pages/viewpage.action?pageId=26743178

상속의 예


```
class Phone{
    전화걸기
    전화받기
}

class MobilePhone extends Phone{
   전화걸기, 전화받기
   무선기지국 연결
   배터리 충전하기
}

class MusicPhone extends MobilePhone {
  전화걸기, 전화받기
  무선기지국 연결, 배터리 충전하기
  음악 다운로드
  음악 재생하기
}
```

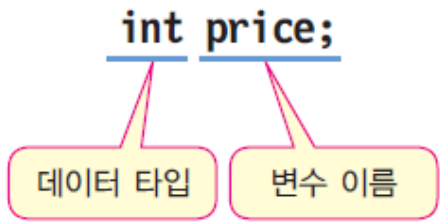
- **View 객체의 특성**
  - Object 객체를 상속받음
  - **그리기 및 이벤트 처리**가 가능
  - View는 위젯의 기반이 되는 클래스
    (위젯: 사용자와 상호작용하는 UI 컴포넌트)
  - TextView, ImageView, SurfaceView, ViewGroup 등이 상속

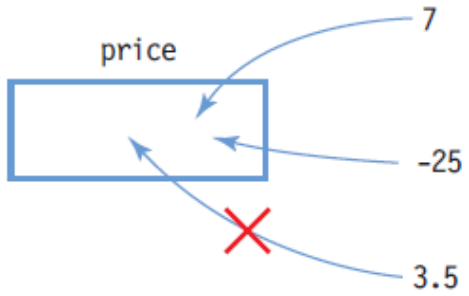
- **ViewGroup**
  - 여러 개의 View 또는 뷰클래스를 담고 있으면서 레이아웃의 특성을 정의하며 Layout의 기반이 되는 클래스

자바 변수란?

- 변수란?
  - 컴퓨터 메모리에 데이터를 기록하는 임시 저장 장소
  - 데이터의 종류에 따라 만들어서 사용

- 변수의 정의(선언)
  - 데이터타입 변수이름;





  - ‘count’라는 정수형 변수 선언→ `int count;`
  - 변수에 값을 할당하는 방법 → `count = 20;`
  - 변수 선언 및 할당을 동시에 하는 방법 → `int count = 20;`

- 데이터 형의 종류
  - 변수가 가질 수 있는 데이터 형은 문자열, 정수형, 부동소수형, 논리형

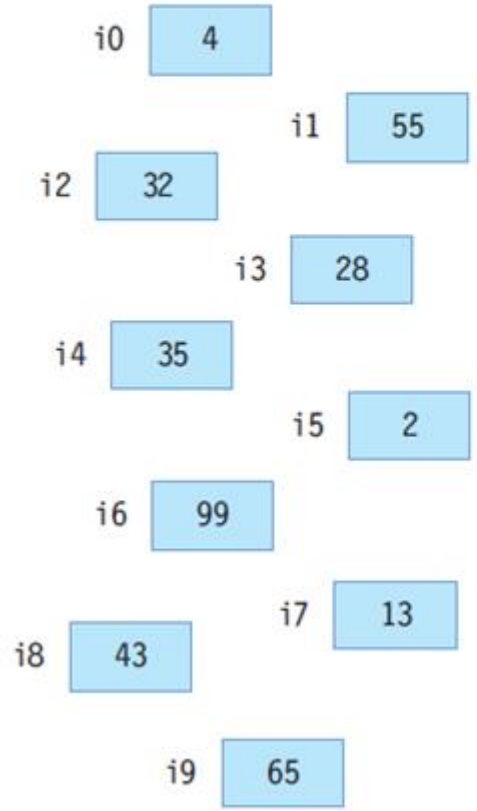
구분	데이터형(byte)	유효값	사용예
문자형	char(2)	작은 따옴표로 묶인 1개 문자	char ch ='가';
정수형	byte(1)	-128 ~ 127	byte count = 10;
	short(2)	-32768 ~ 32767	short count = 100;
	int(4)	-2147483648 ~ 2147483647	int count = 1000;
	long(8)	-9223372036854775808 ~ 9223372036854775807	long count = 10000**L**;
부동소수형	float(4)	1.4E-45 ~ 3.4028235E+38	float sum = 10.3**F**;
	double(8)	4.9E-324 ~ 1.7976931348623157E+308	double sum = 20.4;
논리형	boolean(1)	true, false	Boolean check = true;

자바 배열이란?

- 같은 이름과 같은 데이터 형을 가진 연속적인 저장 공간

(1) 10개의 정수형 변수를 선언하는 경우

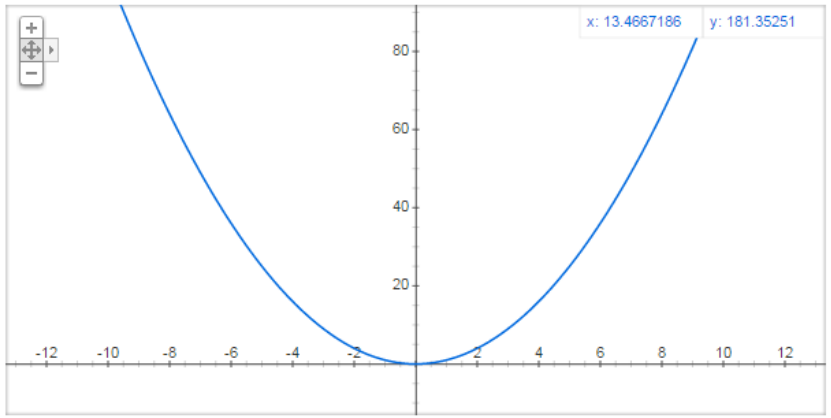
```
int i0, i1, i2, i3, i4, i5, i6, i7, i8, i9;
```

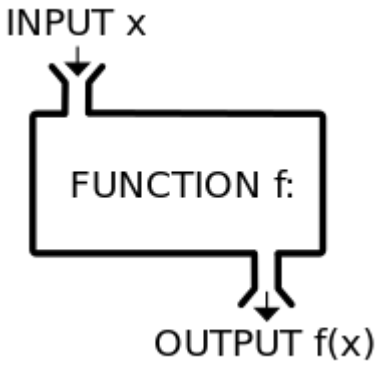


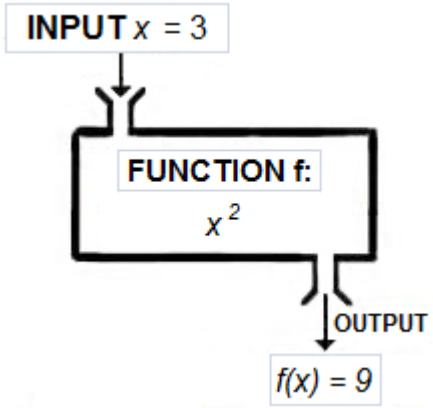
배열 인덱스: 인덱스는 배열의 시작 위치에서부터 데이터가 있는 상대적인 위치. 0부터 시작함

메소드란?

- 함수(Function) **f (x) → x^2 OR f(x) = x^2**








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

Java

```
int x = 3;

int f(int x)
{

}
```

f: function name

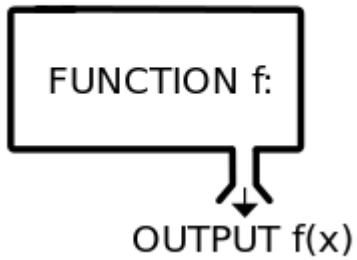
int: return type

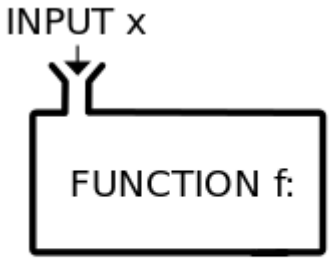
int x: parameters or arguments

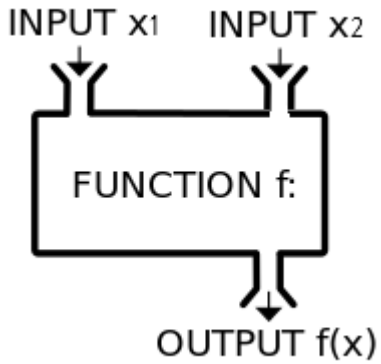
```
return x * x;
```

function body

• 함수의 다양한 형태







```
int f(void)
{
        return x * x;
}
```

```
void f(int x)
{
        int y = x * x;
}
```

```
int f(int x1, int x2)
{
        int y = x1 * x2;
        return y;
}
```

Step 0.프로젝트 개요

프로젝트 개요

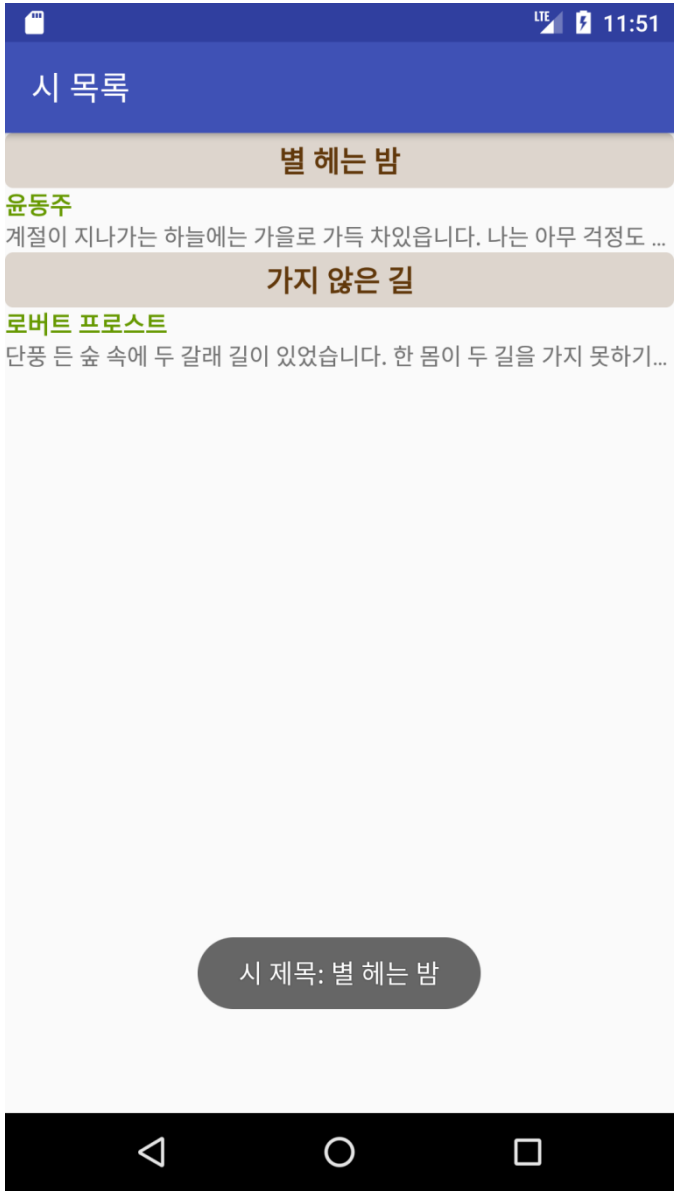
시 목록 클릭

Application Name

EventPoem

어플리케이션 라벨

시목록클릭



Step 1. 프로젝트 생성

절차	내 용
①프로젝트 시작	메뉴에서 ‘File → New Project’ 클릭
②프로젝트 구성	Application Name: EventPoem
	Company Domain: 사용자계정.example.com(디폴트 사용)
③제품형태	Phone and Tablet(사용할 안드로이드 버전 지정: Android 4.4 kitkat)
④액티비티 유형	Empty Activity
⑤파일 옵션	Activity Name: MainActivity
	Layout Name: activity_main

Step 2. 파일 편집

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

파일 연관 관계

클릭

목록 아이템의 제목 모양

```
shape
    shape     rectangle
    color     #3061380B
```

shape_title.xml(drawable)

모양

색

앱 라벨

시 제목

작가

시 본문

텍스트 자원

```
string
    app_name 시 목록
    title01 별 헤는 밤
    autho01 윤동주
    body01 계절이 …
```

strings.xml (values)

화면 레이아웃

```
LinearLayout
    TextView
        text @string/title01
        background @drawable/shape_title
        onClick displayToast
    TextView
        text @string/author01
    TextView
        text @string/body01
```

activity_main.xml (layout)

화면 출력 소스

액티비티 제어

```
onCreate()
    super onCreate()
    setContentView(R.layout.activity_main)
displayToast()
```

MainActivity.java (layout)

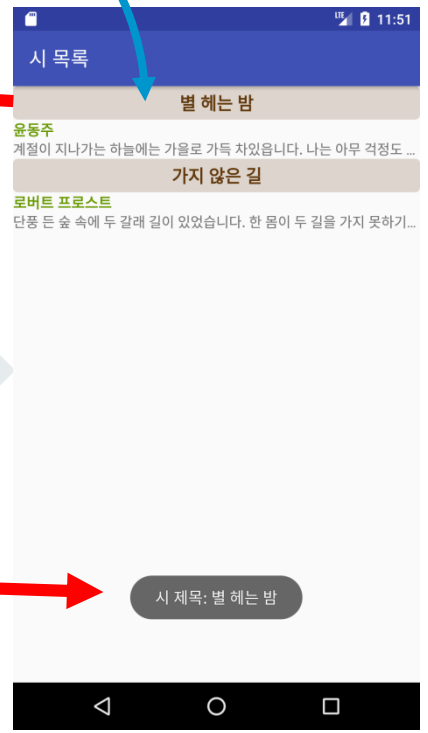
어플리케이션 구성
액티비티의 자바 클래스

어플리케이션 기본 정보

```
application
    icon  @mipmap/ic_launcher
    label @string/app_name
    theme @style/AppTheme
    activity
        name MainActivity
```

AndroidManifest.xml (manifest)

컴파일/빌더



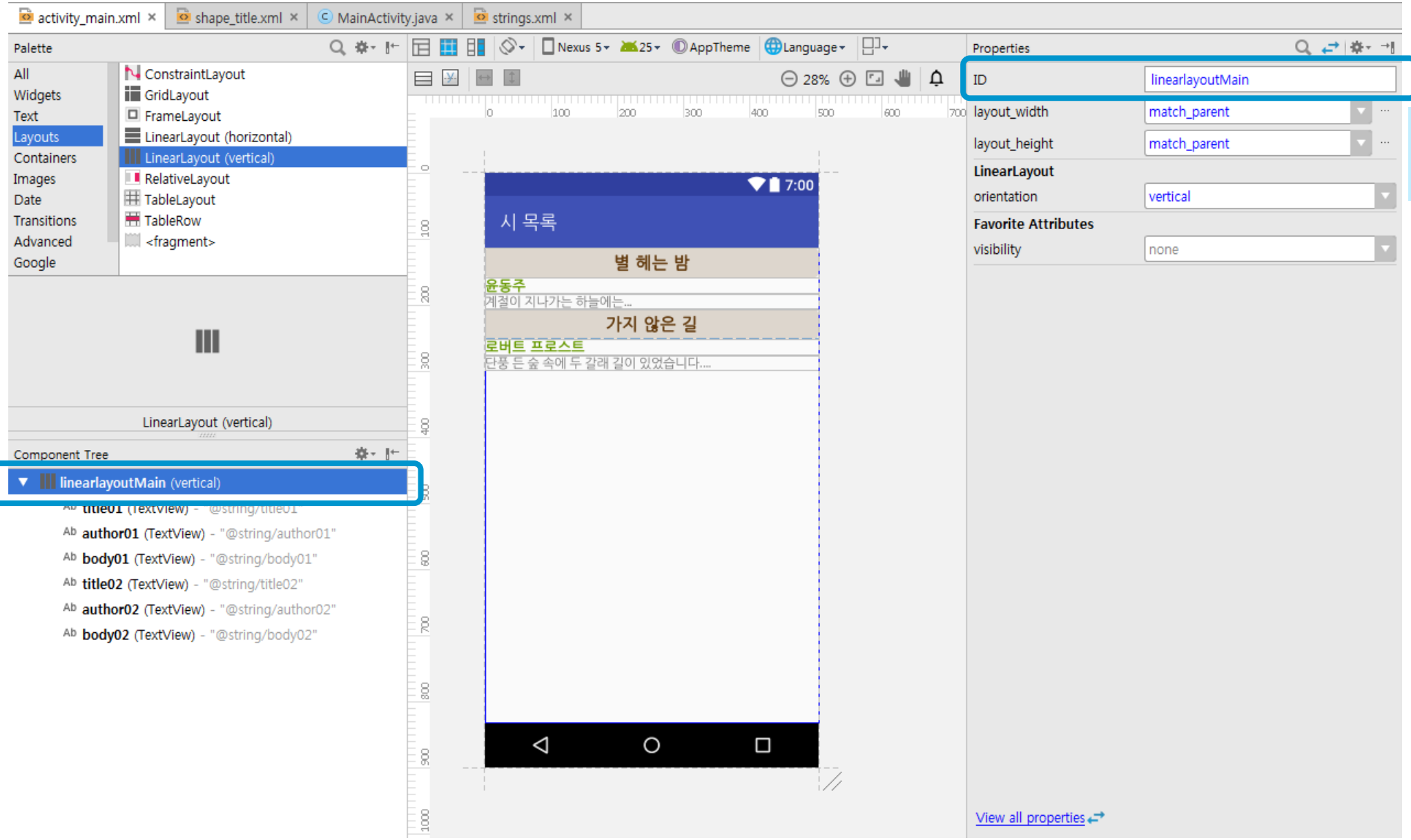
컴파일/빌더 정보

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

(Gradle Scripts)

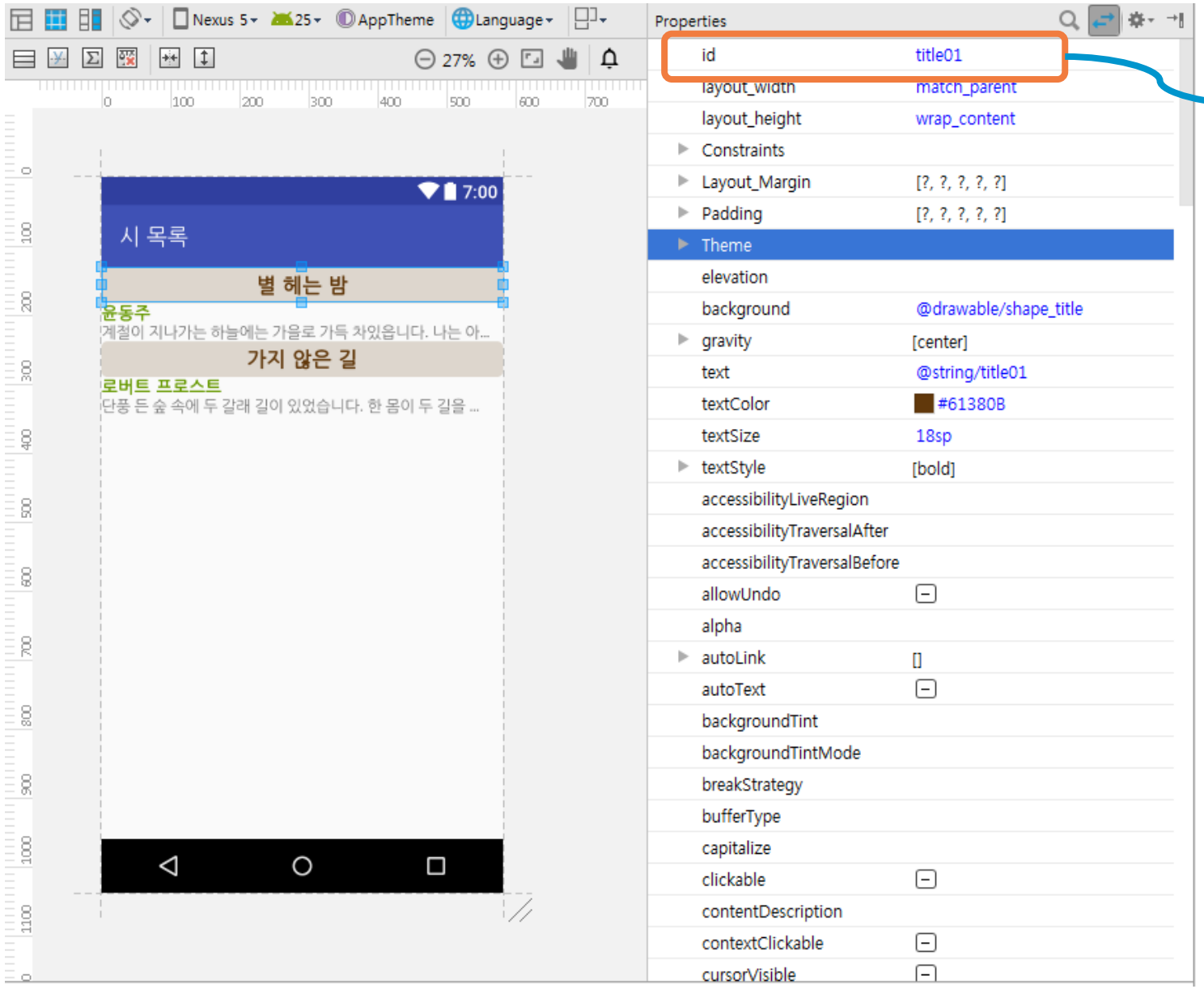
기존 파일 수정

2.1기존 예제(텍스트 출력과 레이아웃)에서 내용 수정



ID:
linearlayoutMain

• 기존 예제(텍스트 출력과 레이아웃)에서 내용 수정



첫번째 시의 제목 TextView 의 id 속성값: title01

항목	세부구분	id 속성값
첫번째 시	제목	title01
	저자	author01
	내용	body01
두번째 시	제목	title02
	저자	author02
	내용	body02

Step 2.2 텍스트 자원의 편집

- strings.xml - 변경된 부분은??? 변경하세요!!

activity_main.xml × | shape_title.xml × | MainActivity.java × | strings.xml ×

Edit translations for all locales in the translations editor.

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

- shape_title.xml 소스(동일)

activity_main.xml | shape_title.xml | MainActivity.java | strings.xml

```
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:left="10dp"
9          android:top="2dp"
10         android:right="10dp"
11         android:bottom="2dp">
12     </padding>
13
14     <corners
15         android:radius="5dp">
16     </corners>
17
18 </shape>
```

출력모양을 사각형으로 지정

출력모양을 내부의 색

내부 패딩 정보

출력모양 모서리를 둥근 모양으로 지정(반지름은 5dp)

클래스와 속성/메소드 – View 클래스의 속성

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

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

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

activity_main.xml

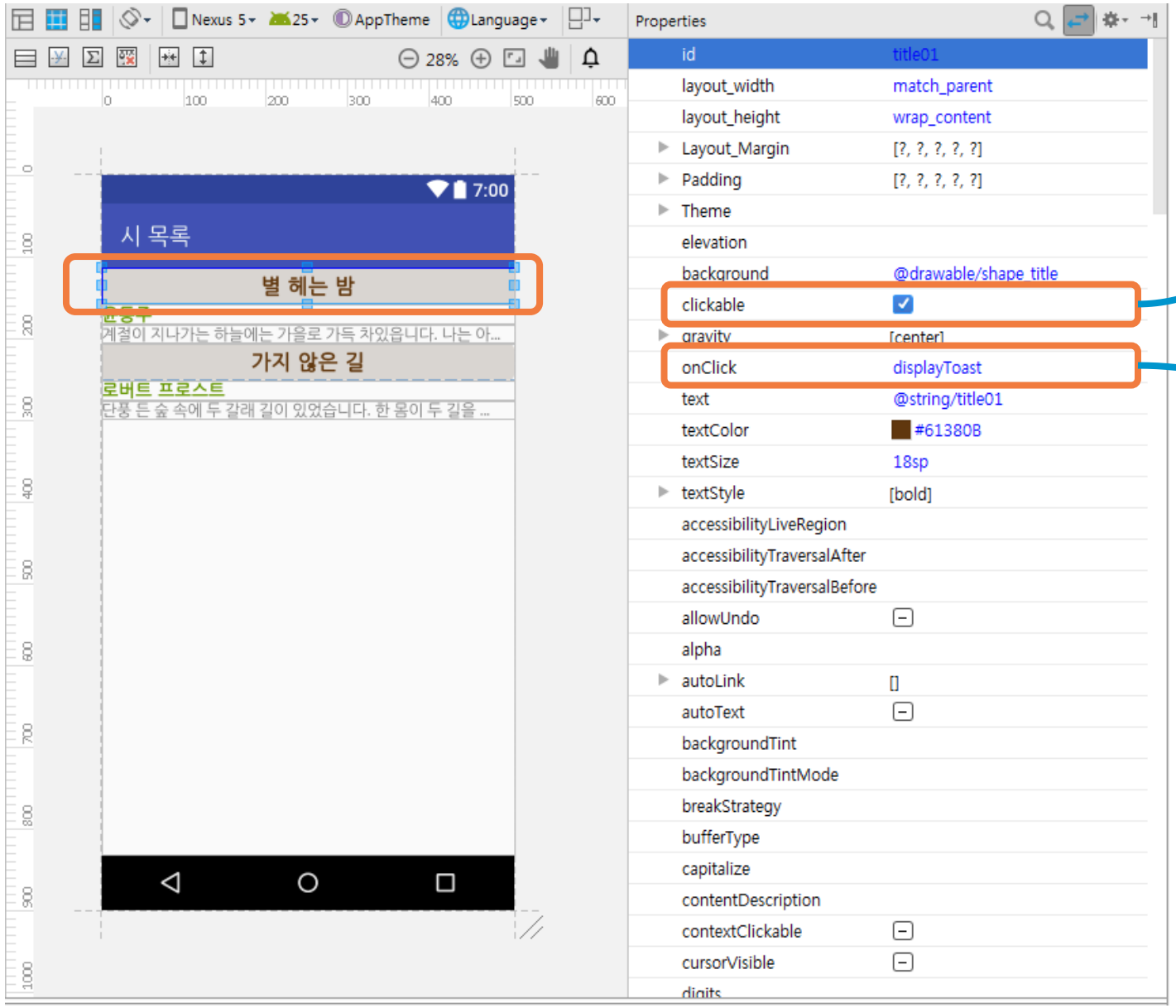
activity_main.xml × | shape_title.xml × | MainActivity.java × | strings.xml ×

LinearLayout | TextView

```
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:id="@+id/linearlayoutMain"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      android:orientation="vertical"
9      tools:context="com.example.research.famouspaintings.MainActivity">
10
11     <TextView
12         android:id="@+id/title01"
13         android:layout_width="match_parent"
14         android:layout_height="wrap_content"
15         android:background="@drawable/shape_title"
16         android:gravity="center"
17         android:text="별 헤는 밤"
18         android:textColor="#61380B"
19         android:textSize="18sp"
20         android:textStyle="bold" />
21
22     <TextView
23         android:id="@+id/author01"
24         android:layout_width="match_parent"
25         android:layout_height="wrap_content"
26         android:text="윤동주"
27         android:textColor="@android:color/holo_green_dark"
28         android:textSize="15sp"
29         android:textStyle="bold" />
```

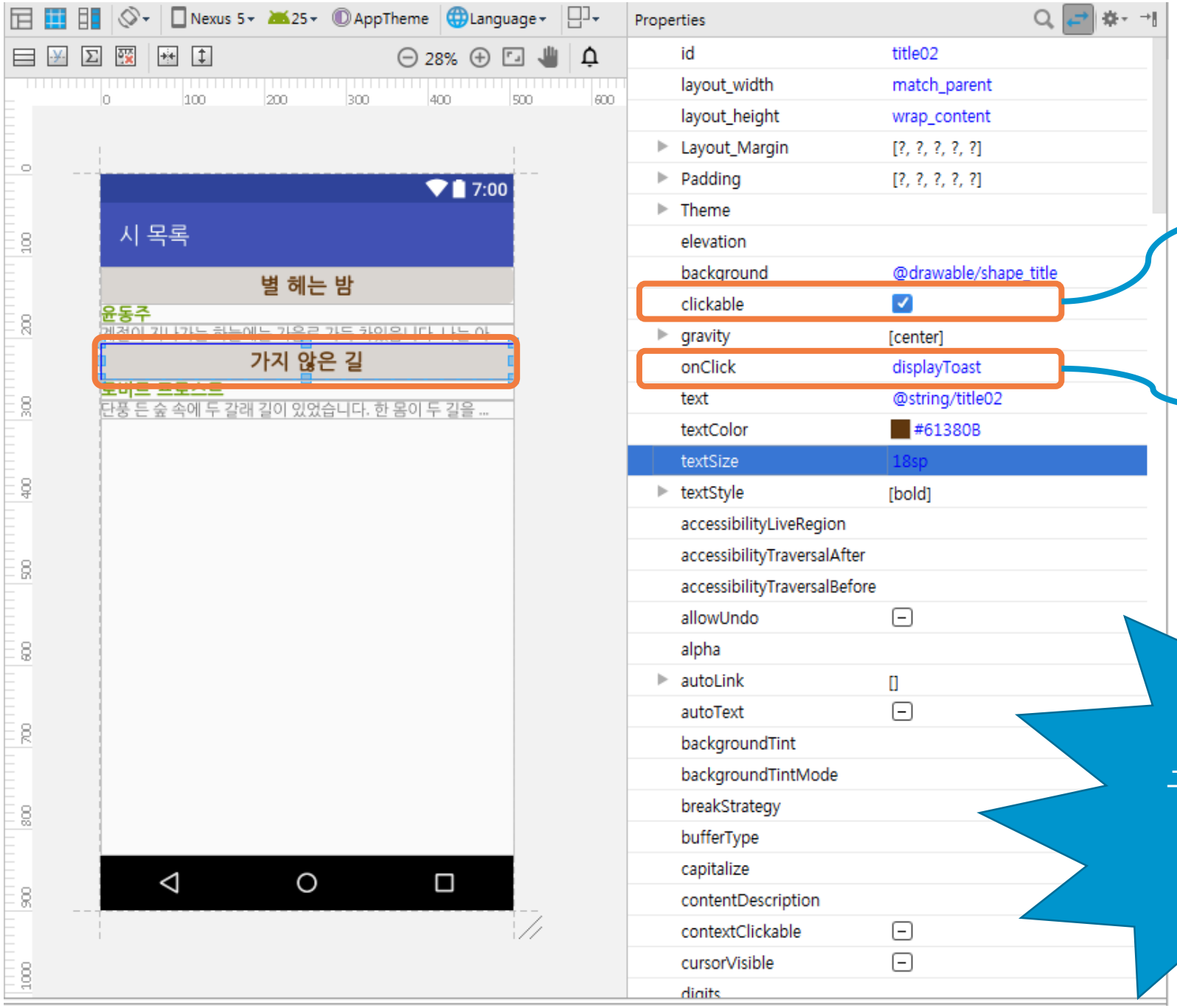
TextView의 다양한 속성이 설정되어 있음

제목(title)을 클릭으로 알림창(Toast)을 띄우기



clickable: check(true)

onClick: displayToast



clickable: check(true)

onClick: displayToast

같은 메소드를
호출했다.
구별은 어떻게???
ID 속성값

2.2 파일 편집 – 메소드 만들기

activity_main.xml × | shape_title.xml × | MainActivity.java × | strings.xml ×

```
4   import android.os.Bundle;
5   import android.view.View;
6   import android.widget.TextView;
7   import android.widget.Toast;
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 displayToast(View view){
18          int vID = view.getId(); // 클릭한 객체의 ID를 얻는다.
19
20          TextView textView = (TextView) findViewById(vID); // id가 vID인 위젯요소를 가져온다.
21          String title = (String) textView.getText(); // 위젯요소의 Text 내용을 가져온다.
22
23          // 팝업 알림창을 생성한다. - 내용과 보여줄 시간
24          // Toast.LENGTH_SHORT =
25          Toast toast = Toast.makeText(this, "시 제목: " + title, Toast.LENGTH_SHORT);
26          // 알림창을 보여준다.
27          toast.show();
28      }
29
30  }
31
```

displayToast 메소드 추가

클래스 속성과 메소드

- 클래스

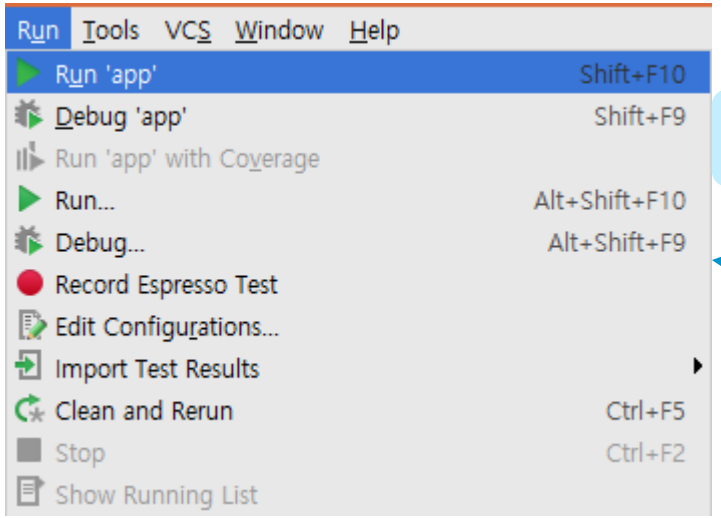
클래스	설명
Toast	화면에 작은 팝업 창을 통해 사용자에게 간단한 메시지를 전달하는 데 사용된다. 일정 시간이 지나면 자동으로 사라짐

- 메소드

클래스	메소드	설명
Toast	static Toast makeText(Context context, CharSequence text, int duration)	문자를 포함하는 표준 toast를 정의함. makeText(this,"문자열",출력시간)을 지정하면 현재 화면 "문자열"을 지정한 출력시간만큼 나타내고 사라짐 미리 정의된 LENGTH_SHORT, LENGTH_LONG을 사용할 수 있음.
	void show()	지정된 시간 동안 뷰를 보여줌

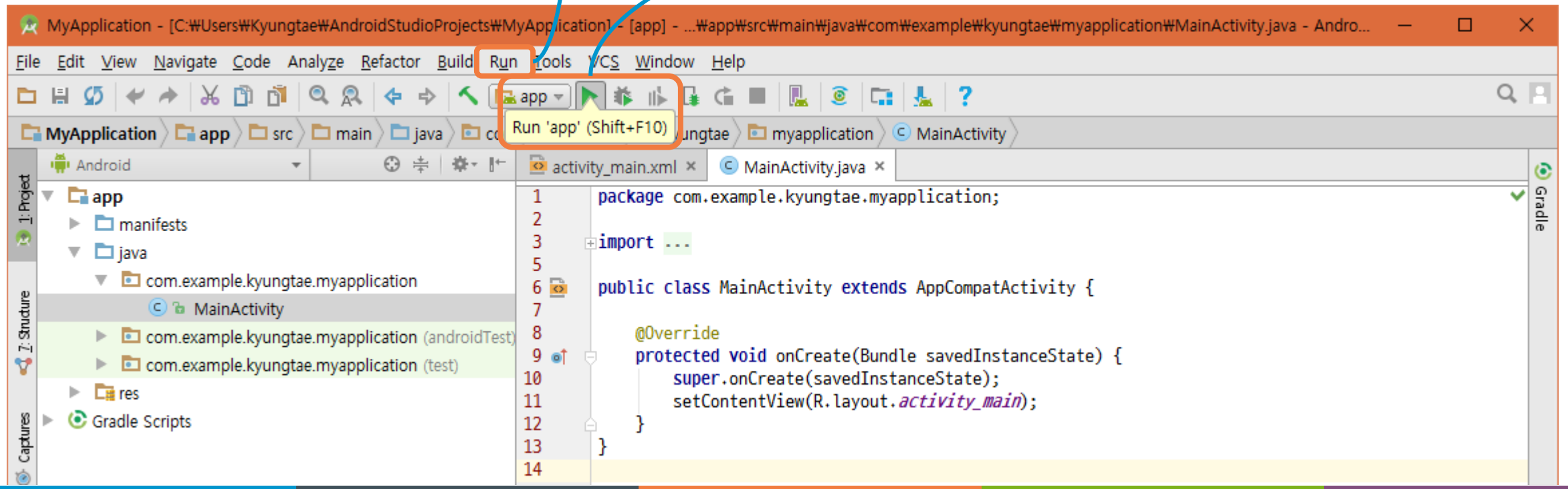
클래스	메소드	설명
View	final View findViewById(int id)	id 값에 해당하는 View를 인식함
	int getId()	View의 ID를 반환함
	Object getTag()	View의 태그값을 반환함

Step 3. 프로젝트 실행

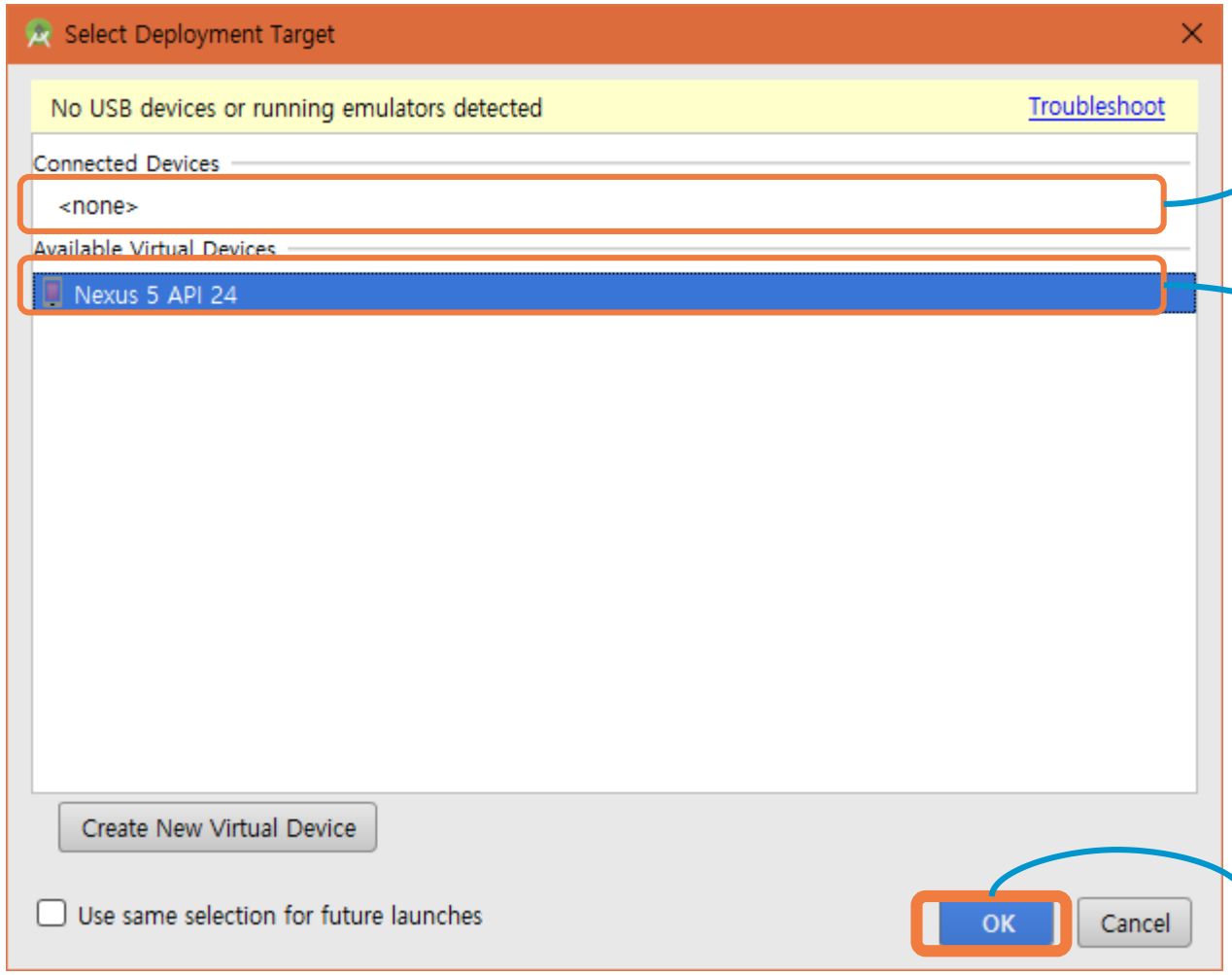


Run → Run 'app' 메뉴 클릭

앱 실행 아이콘 클릭



• AVD 장비 선택하기

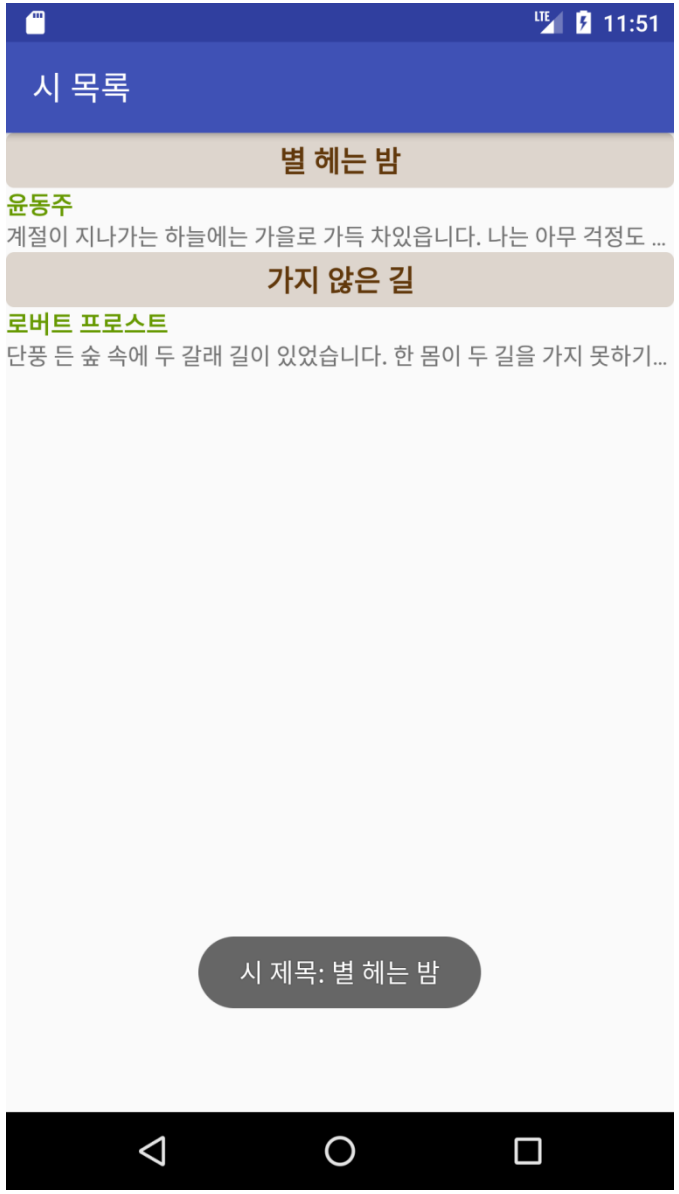


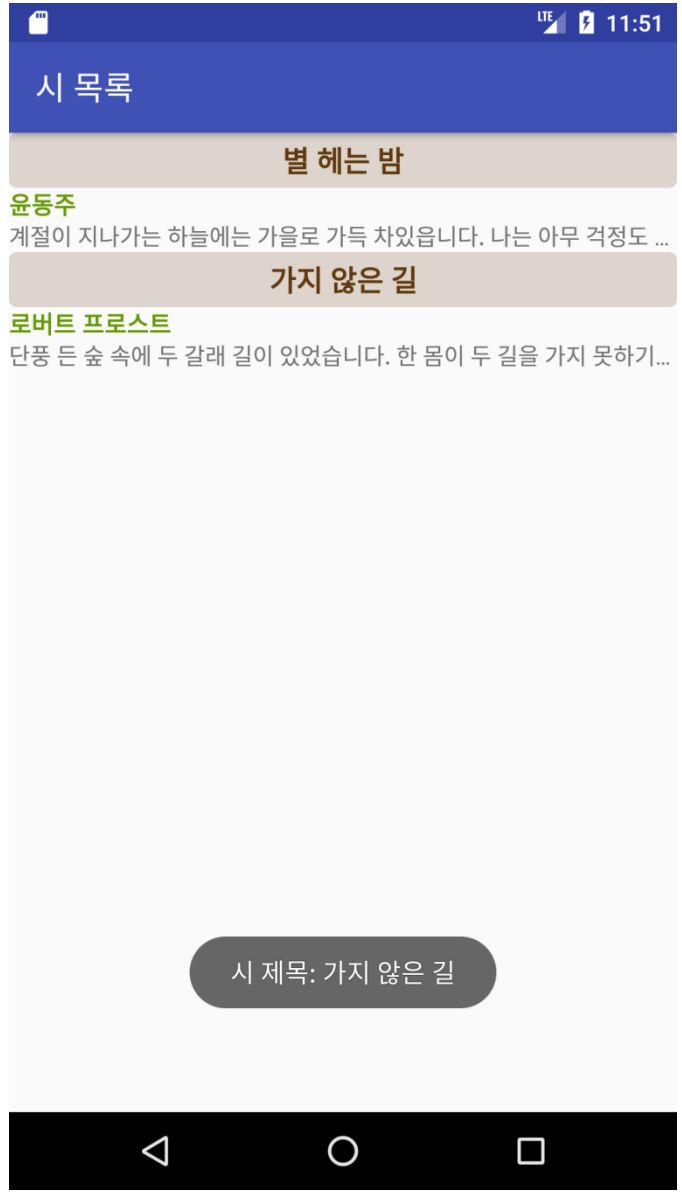
데이터 케이블로 연결된 스마트폰

AVD

스마트폰 또는 AVD를 선택하고 'OK' 버튼을 클릭

Outputs


11:51
시 목록
별 헤는 밤
윤동주
계절이 지나가는 하늘에는 가을로 가득 차있읍니다. 나는 아무 걱정도 ...
가지 않은 길
로버트 프로스트
단풍 든 숲 속에 두 갈래 길이 있었습니다. 한 몸이 두 길을 가지 못하기...
시 제목: 별 헤는 밤


11:51
시 목록
별 헤는 밤
윤동주
계절이 지나가는 하늘에는 가을로 가득 차있읍니다. 나는 아무 걱정도 ...
가지 않은 길
로버트 프로스트
단풍 든 숲 속에 두 갈래 길이 있었습니다. 한 몸이 두 길을 가지 못하기...
시 제목: 가지 않은 길

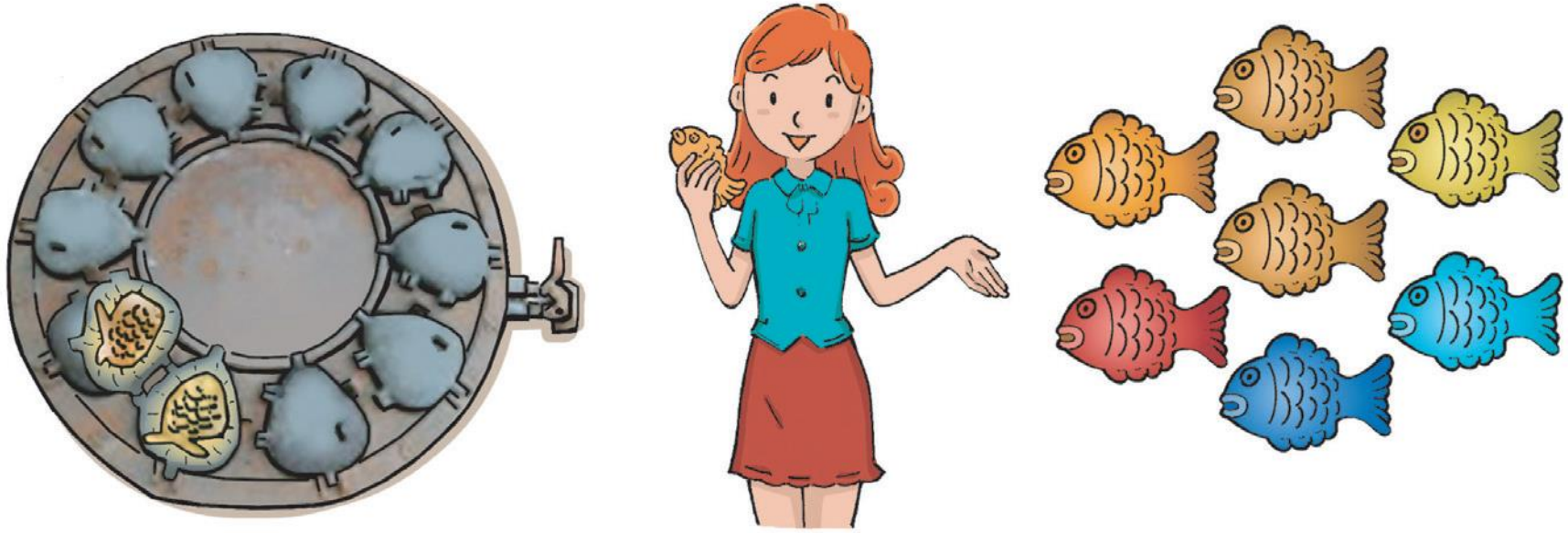
쉼터 – 클래스

클래스와 객체

- 클래스
  - 객체의 공통된 특징 기술
  - 객체의 특성과 행위 선언
- 객체
  - 물리적 공간을 갖는 구체적인 실체
  - 클래스의 인스턴스(실체)
    - 클래스를 구체화한 객체를 인스턴스(instance)라고 부름
    - 객체와 인스턴스는 같은 뜻으로 사용
- 사례
  - 신의 구상?: 클래스 객체: 실존 사람

클래스와 객체와의 관계

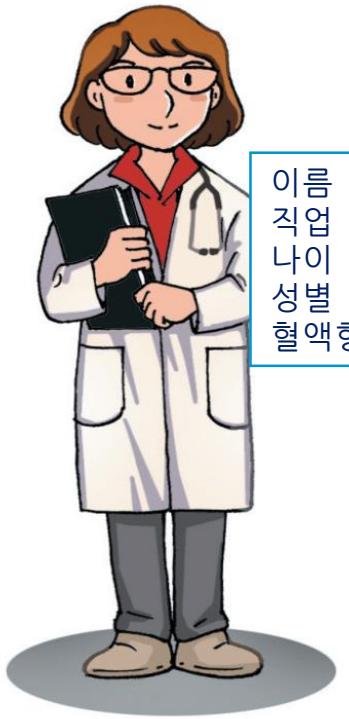
붕어빵 틀은 클래스이며, 이 틀의 형태로 구워진 붕어빵은 바로 객체입니다. 붕어빵은 틀의 모양대로 만들어지지만 서로 조금씩 다릅니다.
치즈붕어빵, 크림붕어빵, 앙코붕어빵 등이 있습니다.
그래도 이들은 모두 붕어빵입니다.



사람을 사례로 든 클래스와 객체 사례

클래스: 사람

특성-이름, 직업,나이, 성별, 혈액형
행위-밥 먹기, 잠자기, 말하기, 걷기

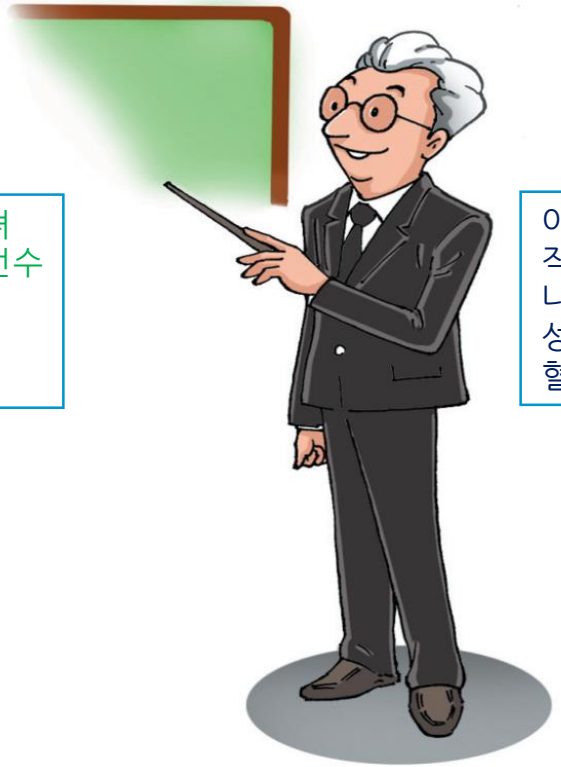


이름 최승희
직업 의사
나이 45
성별 여
혈액형 A

객체 : 최승희

이름 이미녀
직업 골프선수
나이 28
성별 여
혈액형 O

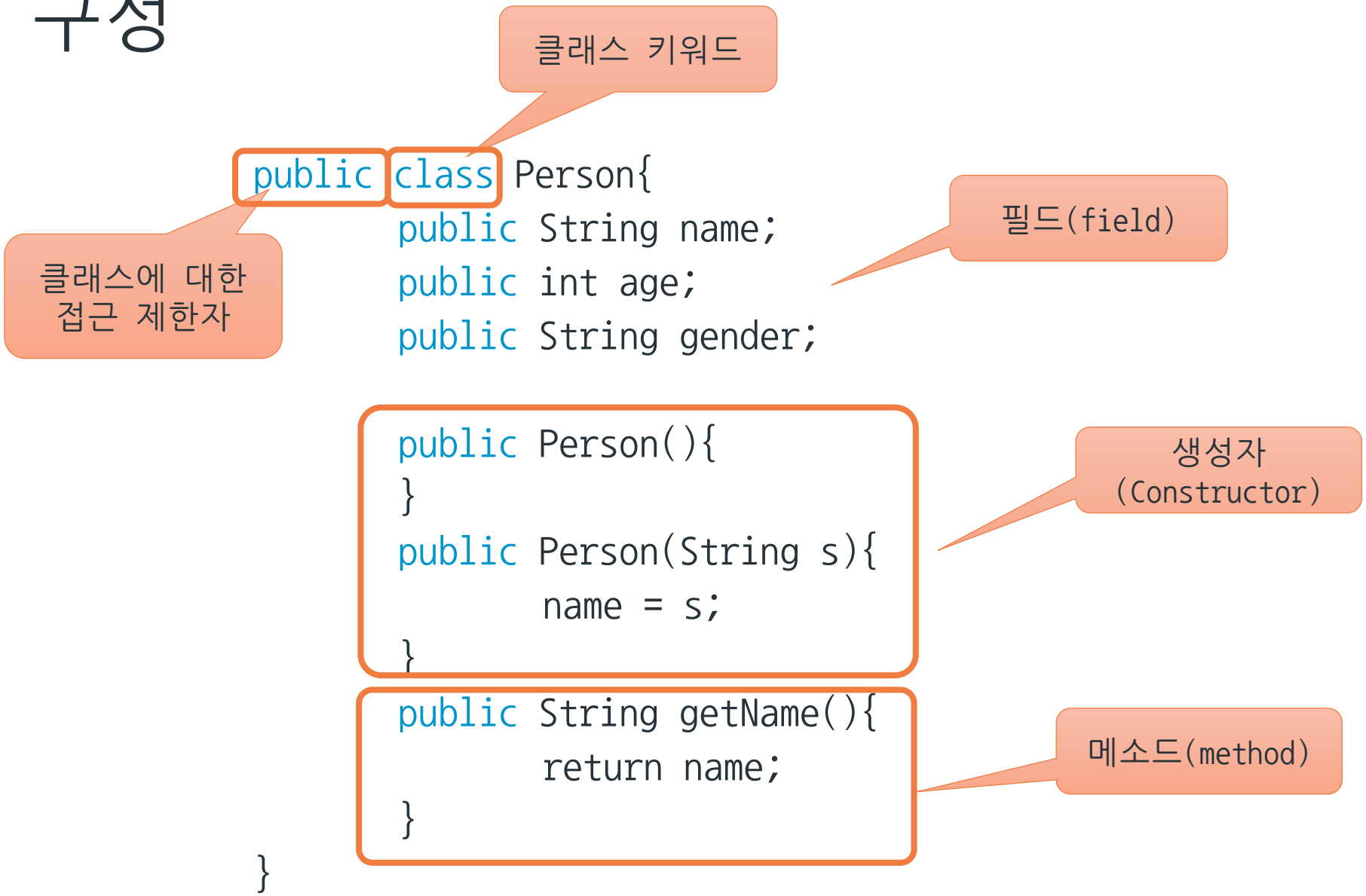
객체 : 이미녀



이름 김미남
직업 교수
나이 47
성별 남
혈액형 AB

객체:김미남

- 클래스 구성

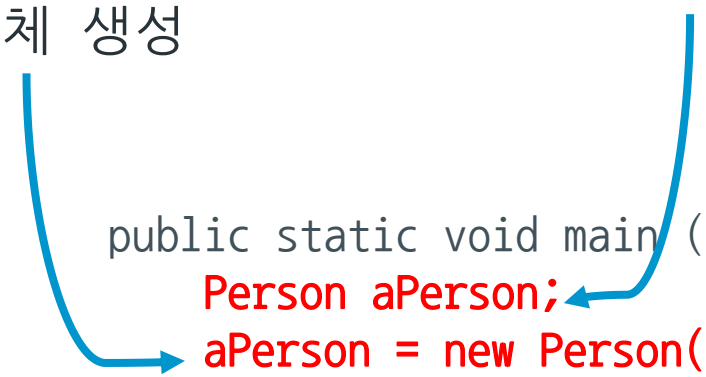

클래스 키워드
public class Person{
public String name;
public int age;
public String gender;
클래스에 대한
접근 제한자
필드(field)
public Person(){
}
public Person(String s){
name = s;
}
생성자
(Constructor)
public String getName(){
return name;
}
메소드(method)
}

클래스 선언

- 클래스 접근 권한, public
  - public 접근 권한은 다른 클래스들이 이 클래스에 대해 사용 혹은 접근이 가능함을 의미
- class Person
  - Person이라는 이름의 클래스 정의
  - class 다음에 클래스의 이름을 선언
  - 클래스는 {로 시작하여 }로 닫으며 이곳에 모든 멤버 필드와 메소드 구현
- 필드(field)
  - 값을 저장할 멤버 변수를 선언
  - 멤버 변수 혹은 필드라고 함
  - 필드 앞에 붙은 접근 지정자 public
    - 이 필드가 다른 클래스에서 접근될 수 있도록 공개한다는 의미
- 생성자(constructor)
  - 클래스의 이름과 동일한 메소드
  - 클래스의 객체가 생성될 때만 호출되는 메소드
- 메소드(method)
  - 메소드는 함수이며 객체의 행위를 구현
  - 메소드 앞에 붙은 접근 지정자 public
    - 메소드가 다른 클래스에서 접근될 수 있도록 공개한다는 의미

클래스 생성

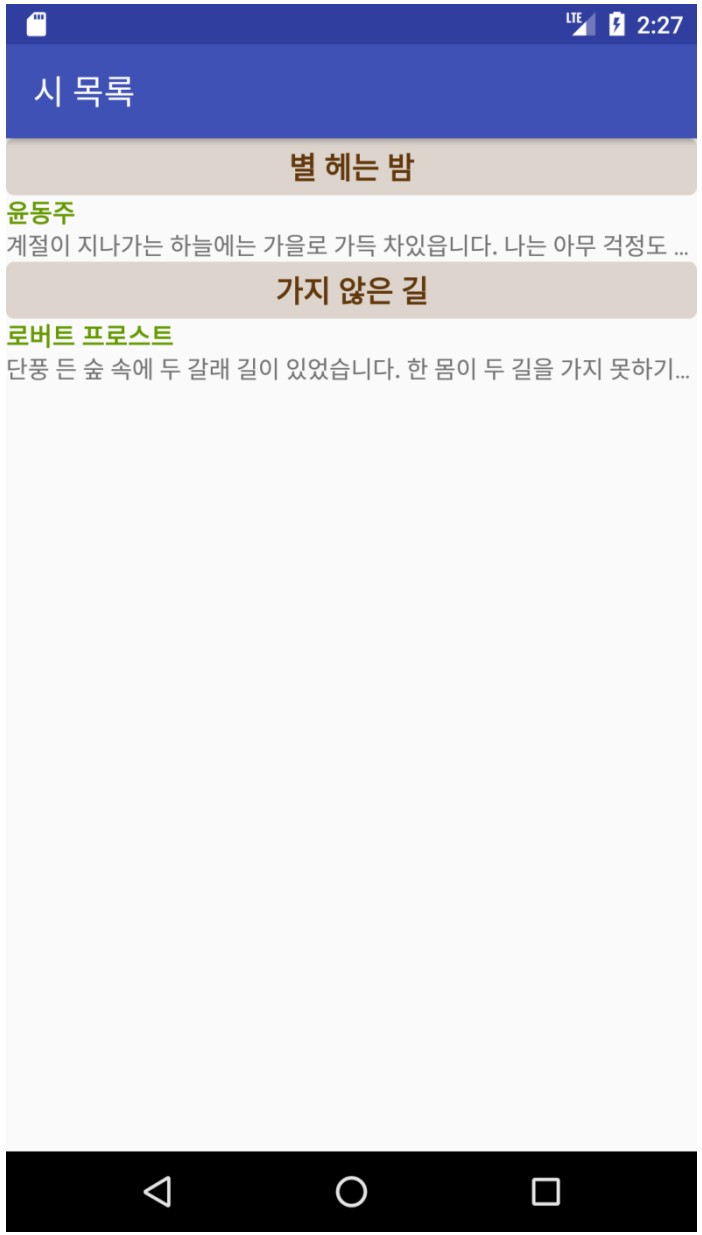
- 객체 생성
  - 객체는 new 키워드를 이용하여 생성
    - new는 객체의 생성자 호출
- 객체 생성 과정
  - 객체에 대한 레퍼런스 변수 선언
  - 객체 생성

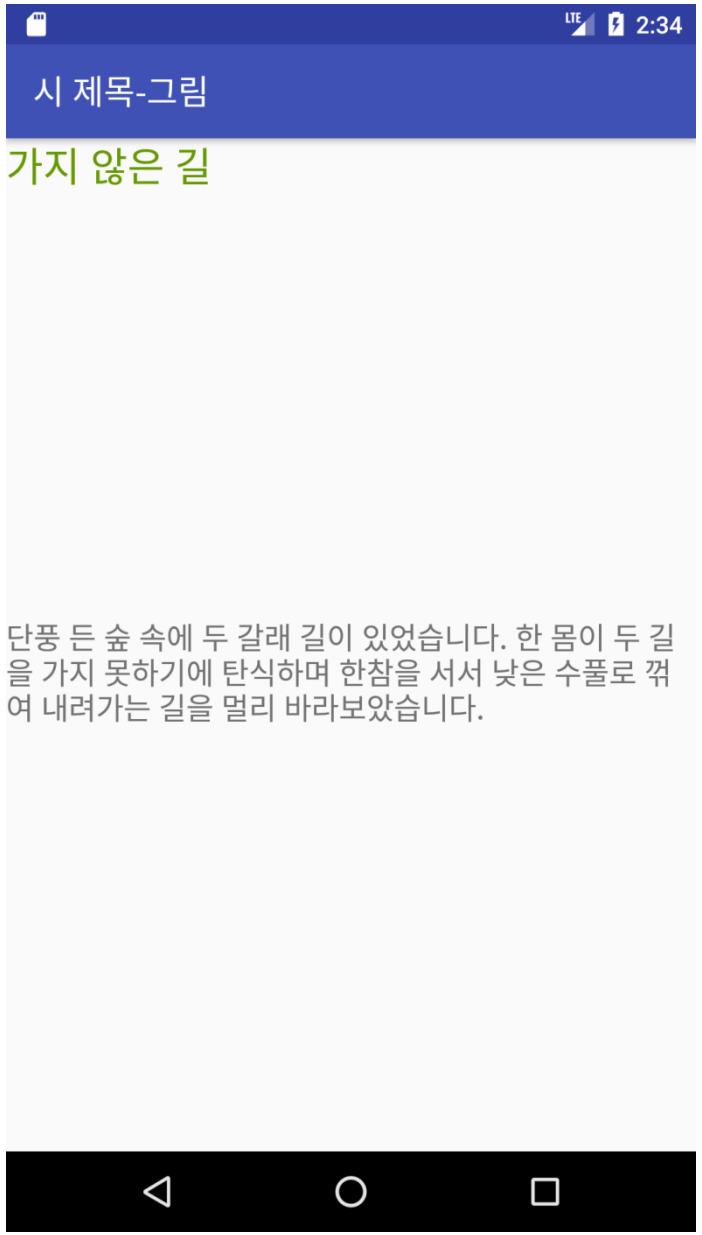


```
public static void main (String args[]) {
    Person aPerson;                     // 레퍼런스 변수 aPerson 선언
    aPerson = new Person("김미남");     // Person 객체 생성
    // Person aPerson = new Person("김미남");
    aPerson.age = 30;                   // 객체 멤버 접근
    int i = aPerson.age;                // 30
    String s = aPerson.getName();       // 객체 메소드 호출
}
```

액티비티 전환

Outputs


2:27
시 목록
별 헤는 밤
윤동주
계절이 지나가는 하늘에는 가을로 가득 차있읍니다. 나는 아무 걱정도 ...
가지 않은 길
로버트 프로스트
단풍 든 숲 속에 두 갈래 길이 있었습니다. 한 몸이 두 길을 가지 못하기...


2:34
시 제목-그림
가지 않은 길
단풍 든 숲 속에 두 갈래 길이 있었습니다. 한 몸이 두 길을 가지 못하기에 탄식하며 한참을 서서 낮은 수풀로 꺾여 내려가는 길을 멀리 바라보았습니다.

두번째 액티비티

- 두번째 액티비티(화면)을 띄워서 시 제목과 내용을 보여준다.
- intent를 사용해 시제목과 시 내용을 넘겨 준다.

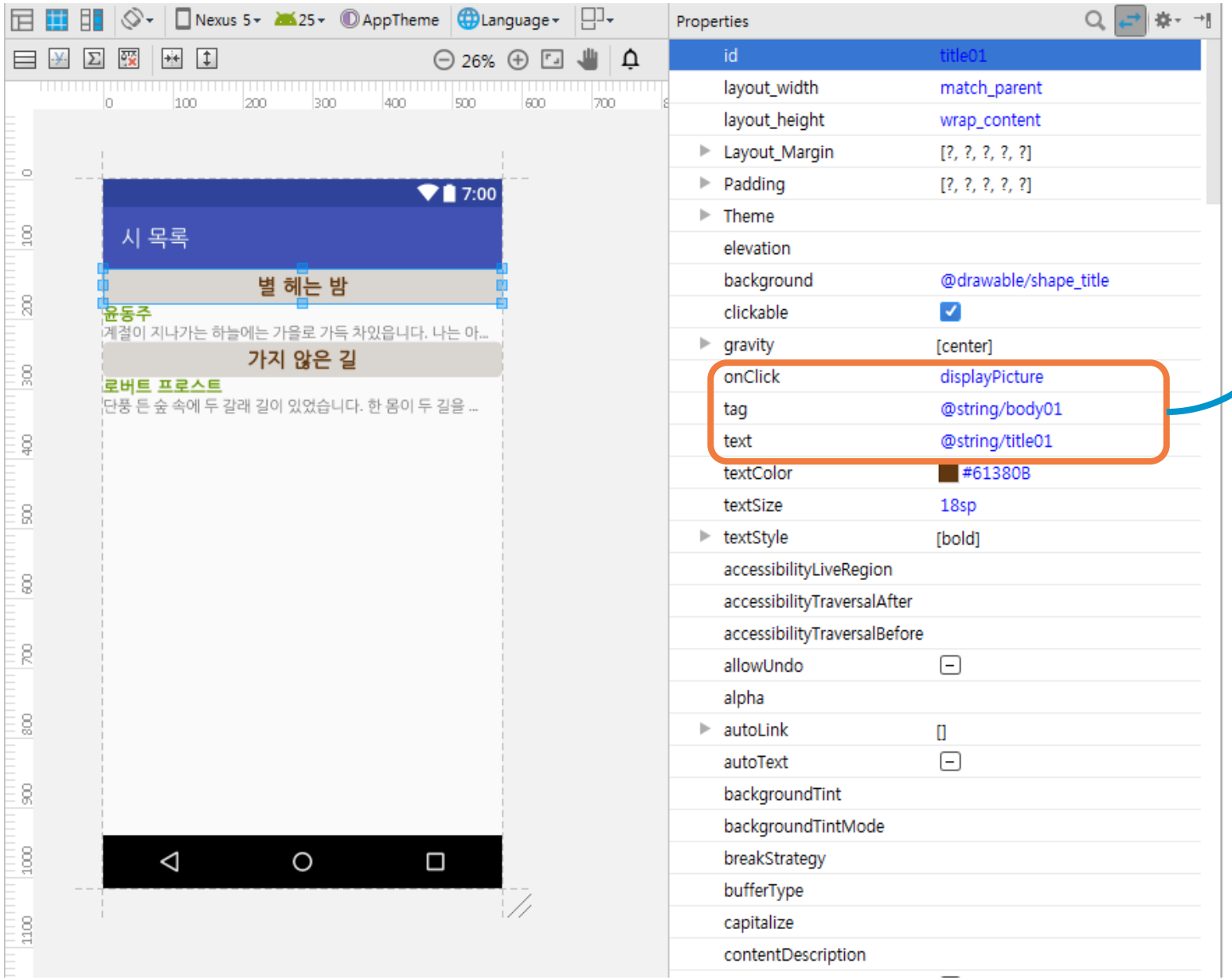
Strings.xml 파일

activity_main.xml × | activity_detail_view.xml × | strings.xml ×

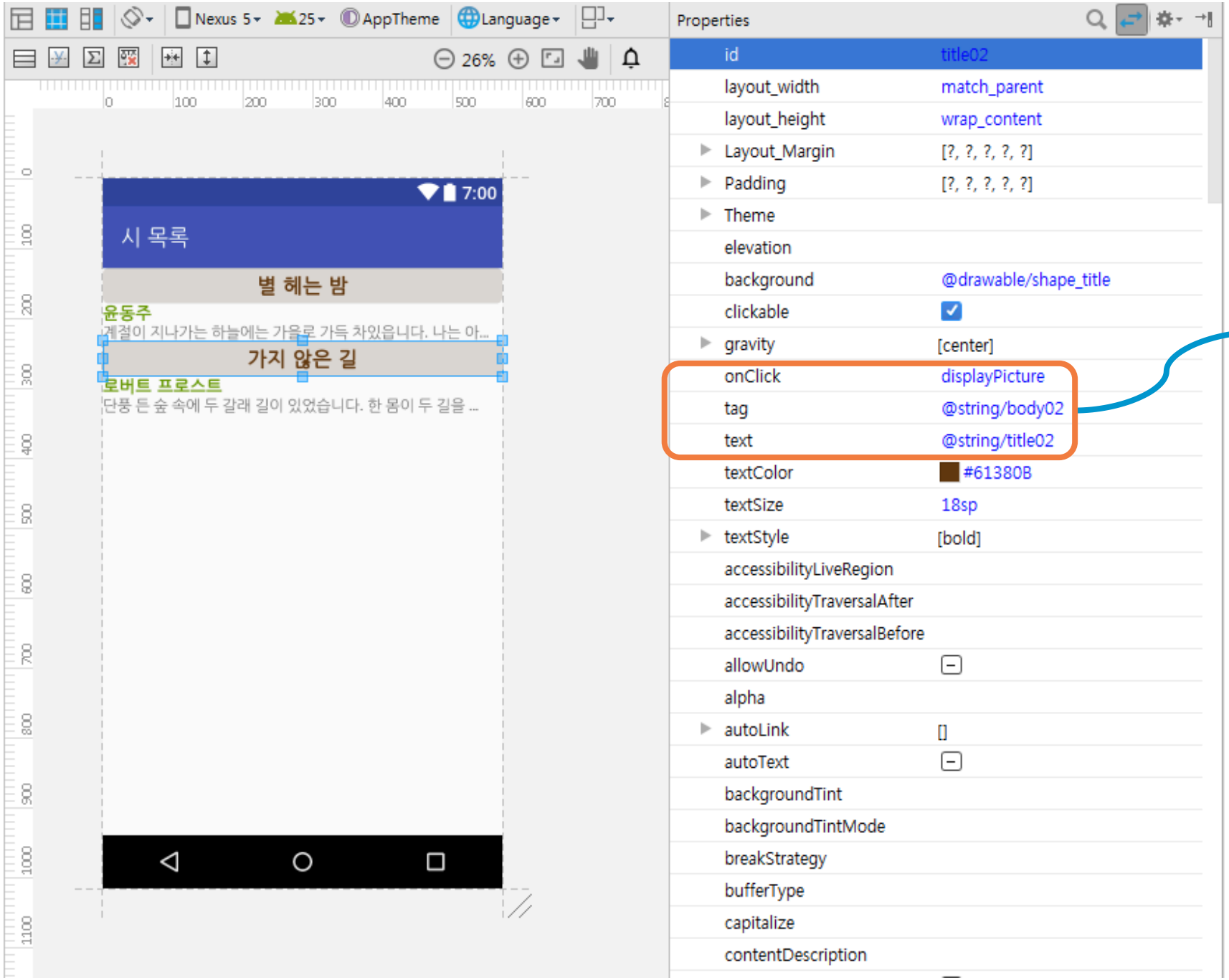
Edit translations for all locales in the translations editor.

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

Step 2.2 파일 편집-기존 예제에서 속성 수정



onClick: displayPicture
tag: @string/body01
text: @string/title01



onClick: displayPicture
tag: @string/body02
text: @string/title02

Step 2.2 파일 편집– 자바 메소드 만들기

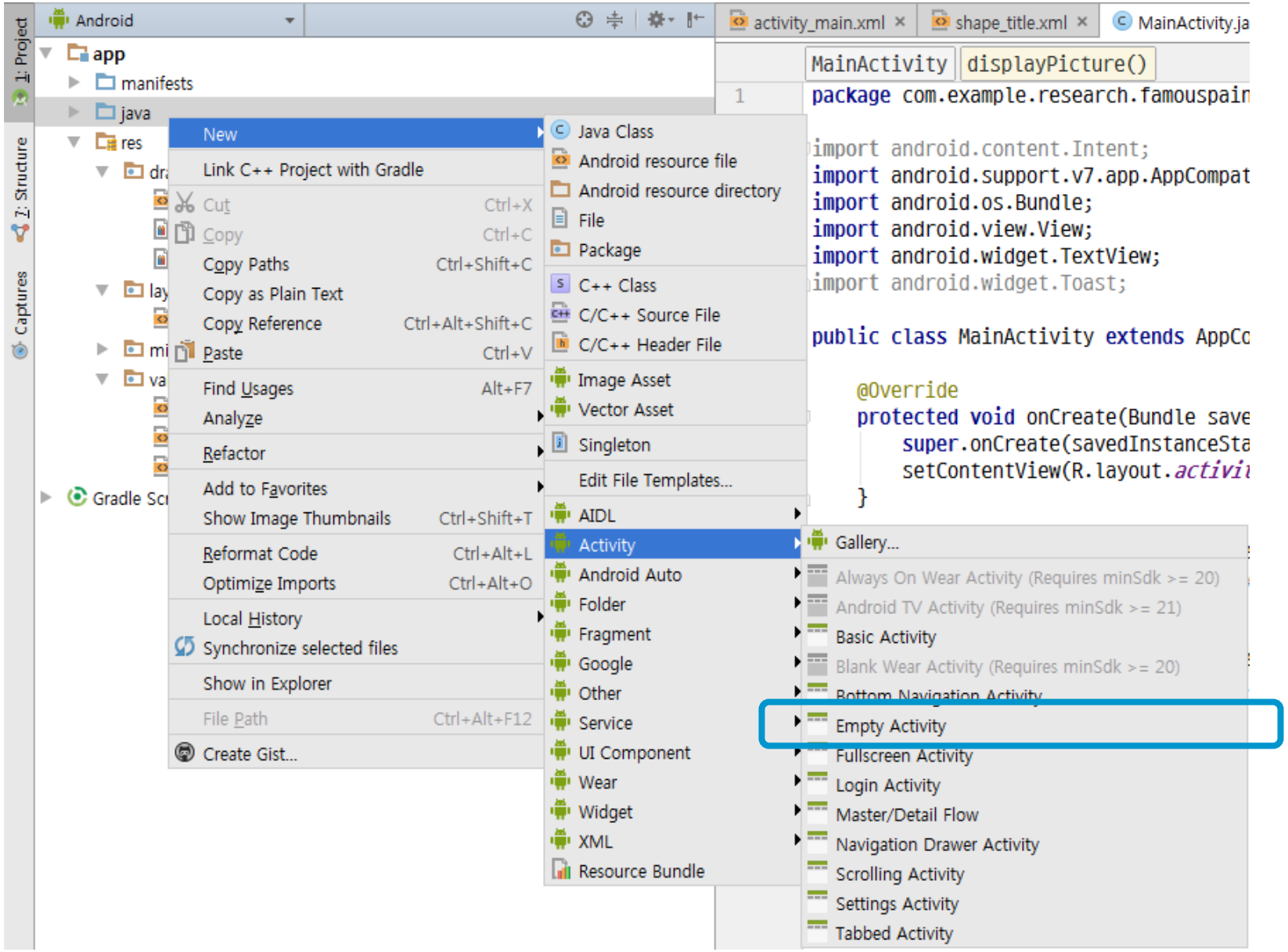
activity_main.xml × | shape_title.xml × | MainActivity.java × | strings.xml ×

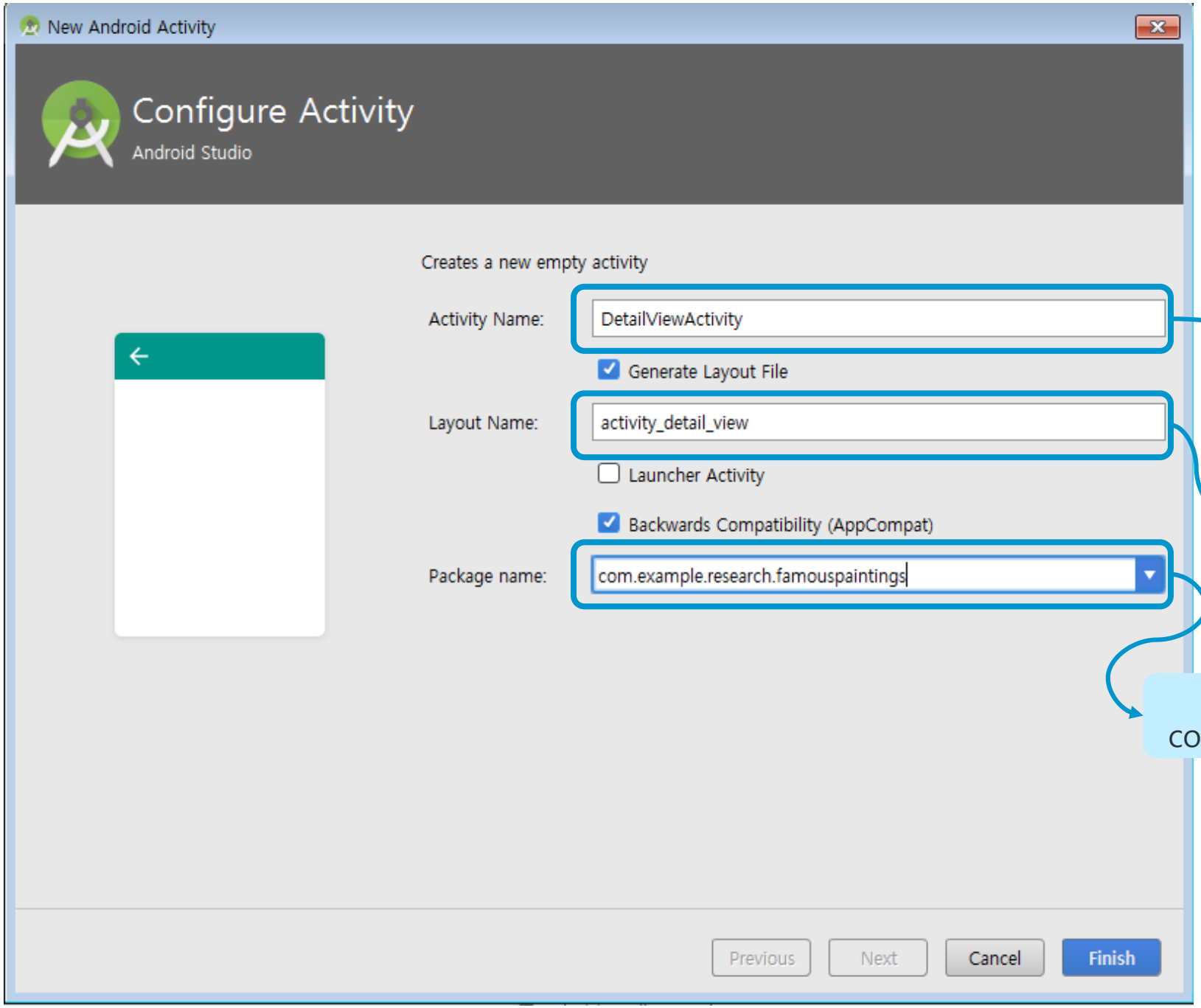
MainActivity.java

```
1  package com.example.research.famouspaintings;
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  import android.widget.Toast;
9
10 public class MainActivity extends AppCompatActivity {
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16     }
17
18     public void displayPicture(View view){
19         int vID = view.getId(); // 클릭한 객체의 ID를 얻는다.
20
21         TextView textView = (TextView) findViewById(vID); // id가 vID인 위젯요소를 가져온다.
22         String title = (String) textView.getText(); // 위젯요소의 Text 내용을 가져온다.
23         String tag = (String) textView.getTag(); // 위젯요소의 tag 내용을 가져온다.
24
25         // Intent에 저장하여 보내기
26
27     }
28
29 }
```

displayPicture 메소드 추가

Step 2.2 파일 편집 – Activity 추가



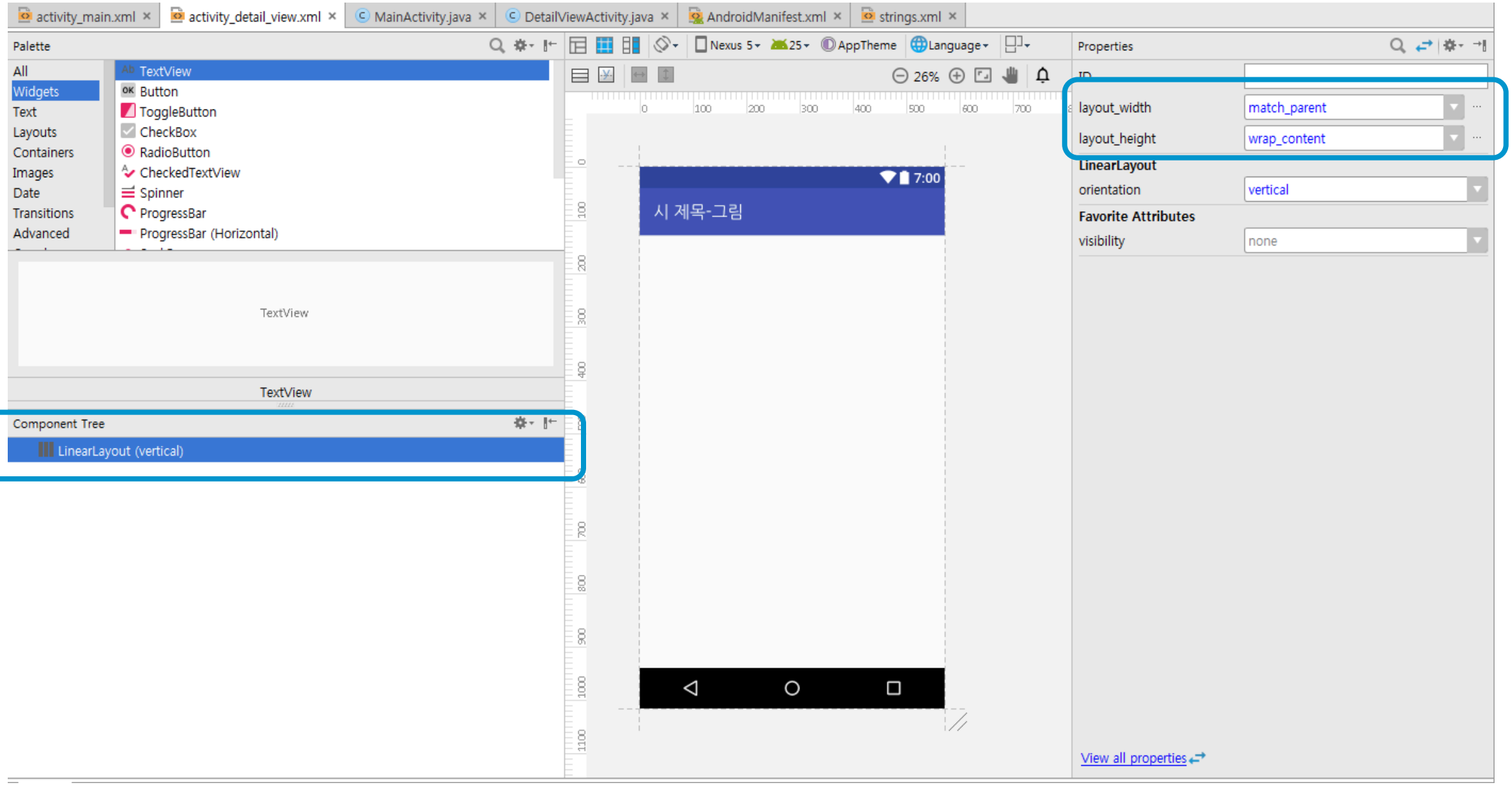


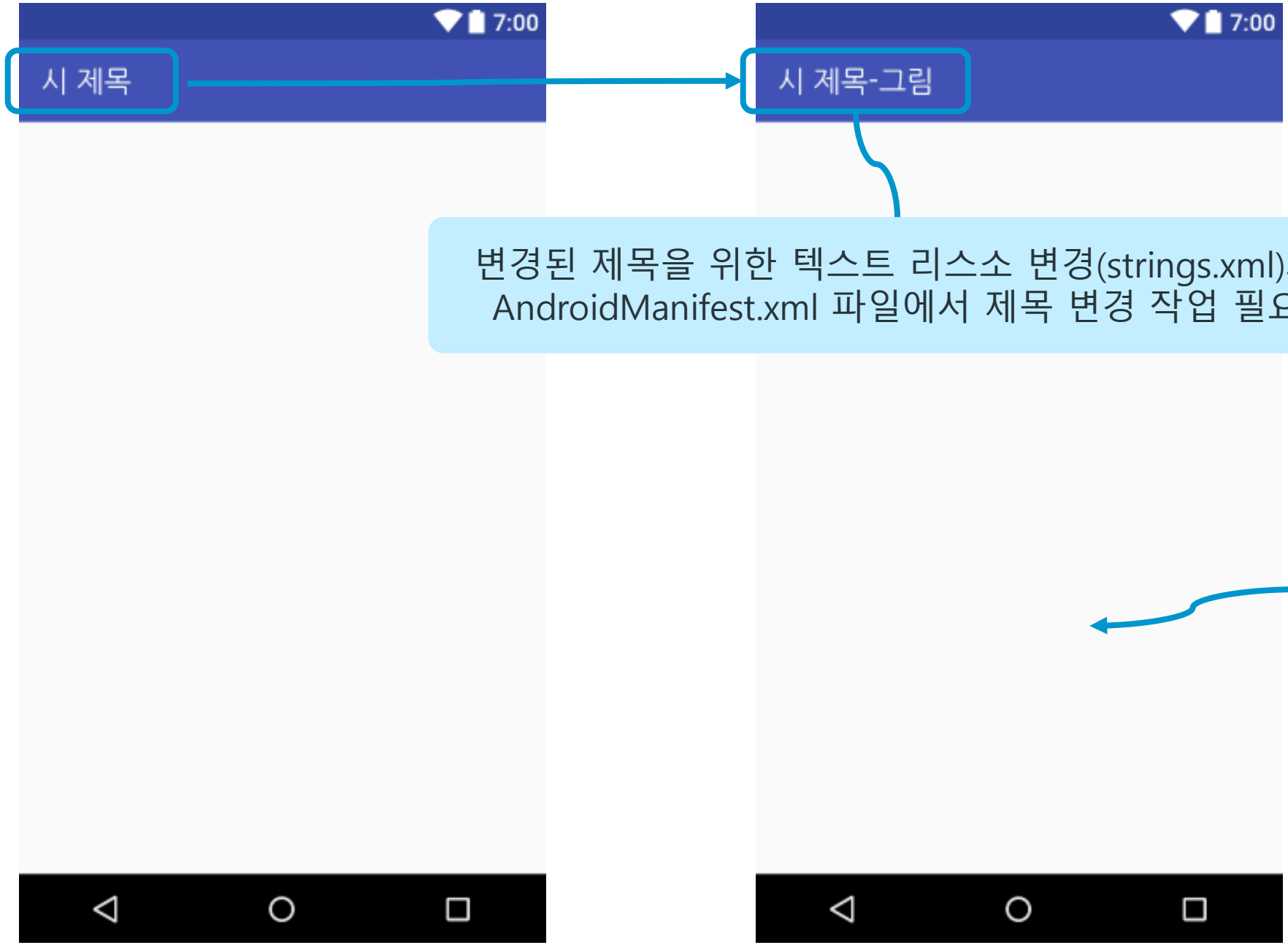
Activity Name:
DetailViewActivity

Layout Name:
activity_detail_view

Package name:
com.example.research.famouspintings

Step 2.2 파일 편집–activity_detail_view.xml




7:00
시 제목
7:00
시 제목-그림
변경된 제목을 위한 텍스트 리스소 변경(strings.xml)과
AndroidManifest.xml 파일에서 제목 변경 작업 필요
화면에 시제목과 내용 표시
➔ 위젯을 넣어야 함.

Step 2.2 파일 편집 – Strings.xml

activity_main.xml × | activity_detail_view.xml × | strings.xml ×

Edit translations for all locales in the translations editor.

```
<resources>
    <string name="app_name">시 목록</string>

    <string name="title01">별 헤는 밤</string>
    <string name="author01">윤동주</string>
    <string name="body01">
    계절이 지나가는 하늘에는
    가을로 가득 차있읍니다.
    나는 아무 걱정도 없이
    가을 속의 별들을 다 헤일듯합니다.
    </string>

    <string name="title02">가지 않은 길</string>
    <string name="author02">로버트 프로스트</string>
    <string name="body02">
    단풍 든 숲 속에 두 갈래 길이 있었습니다.
    한 몸이 두 길을 가지 못하기에
    탄식하며 한참을 서서
    낮은 수풀로 꺾여 내려가는 길을
    멀리 바라보았습니다.
    </string>

    <string name="detail_name">시 제목-그림</string>
</resources>
```

두번째 화면의 제목

Step 2.2 파일 편집–AndroidManifest.xml 파일 수정

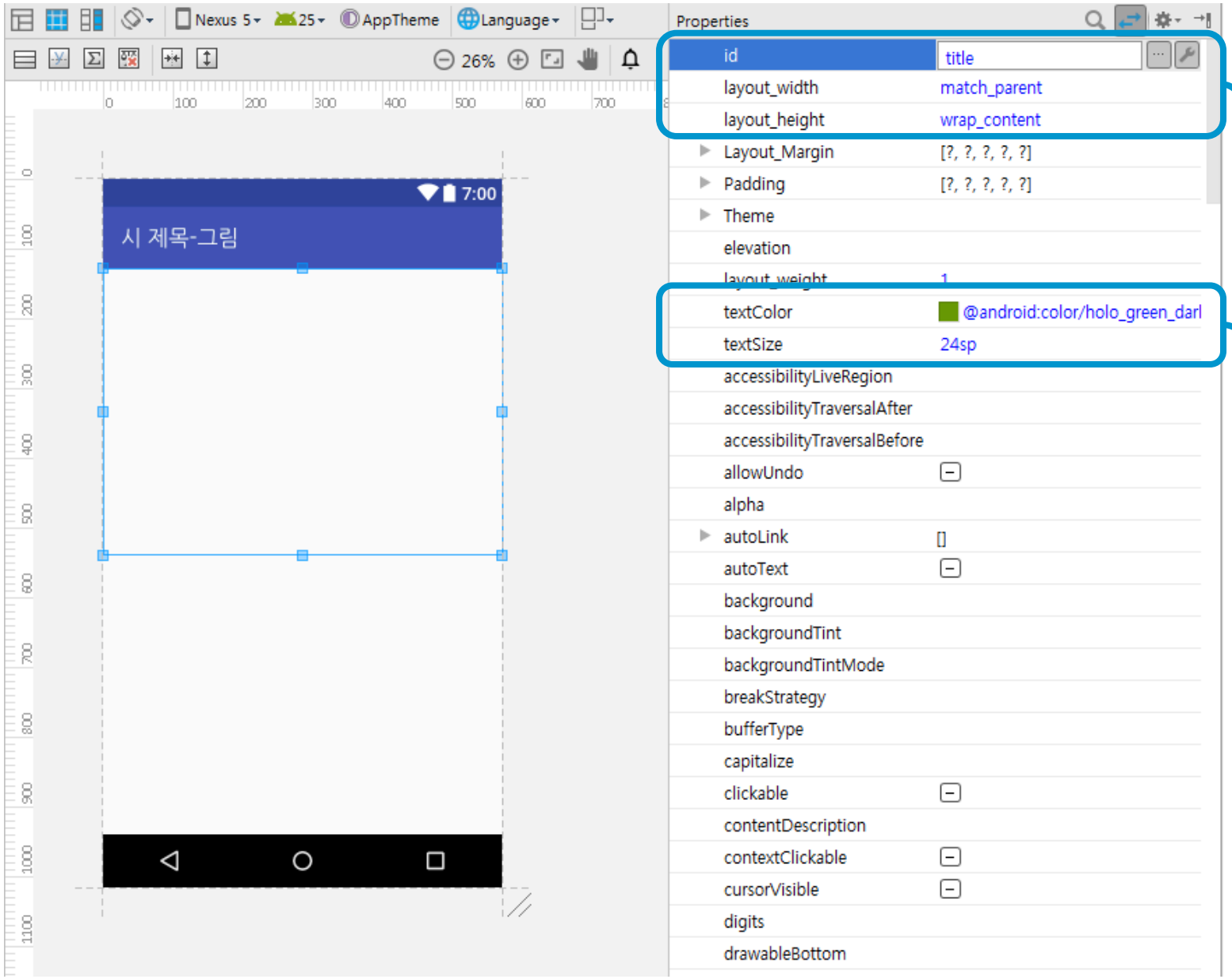
activity_main.xml × | activity_detail_view.xml × | AndroidManifest.xml × | strings.xml ×

```
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:supportsRtl="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
```

activity_main.xml × | activity_detail_view.xml × | AndroidManifest.xml × | strings.xml ×

```
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:supportsRtl="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/detail_name">
21         </activity>
22     </application>
23
24 </manifest>
25
26
```

Step 2.2 파일 편집–activity_detail_view.xml

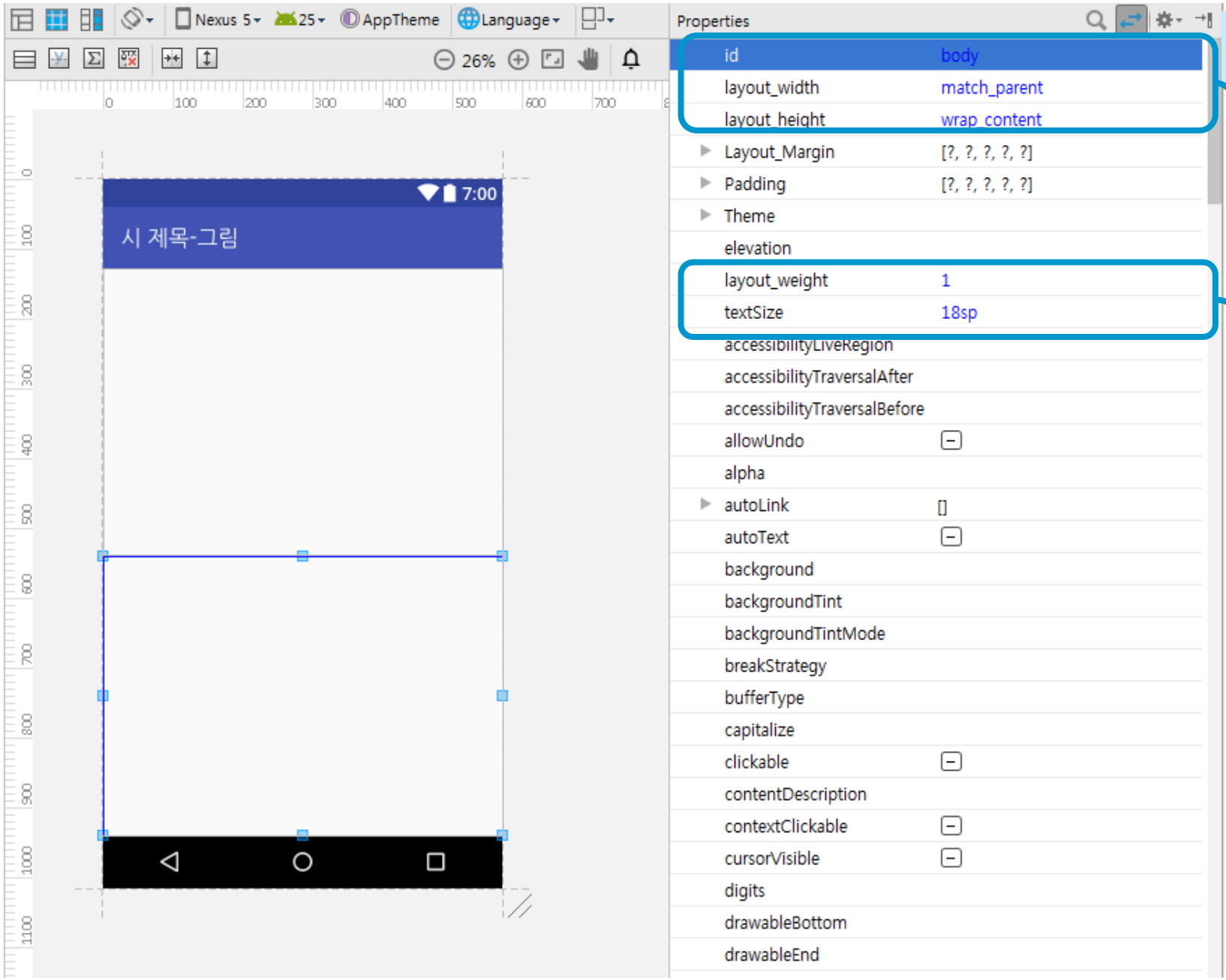


TextView 추가

id: title00
layout_width: match_parent
layout_height: wrap_content

textSize: 24sp
textColor:
@android:color/holo_green_dark
text: 공백

TextView 추가



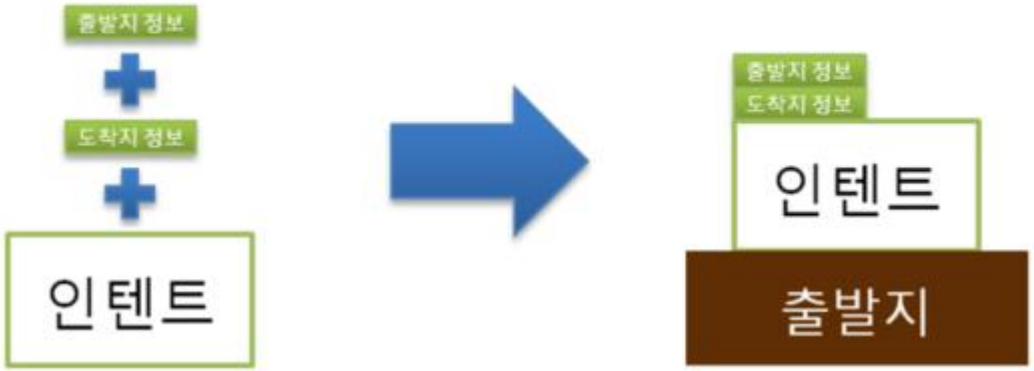
id: body00
layout_width: match_parent
layout_height: wrap_content

textSize: 18sp
text: 공백

Intent – http://m.blog.naver.com/eominsuk55/220228053631

① 출발지 액티비티에서 인텐트를 만든다.

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

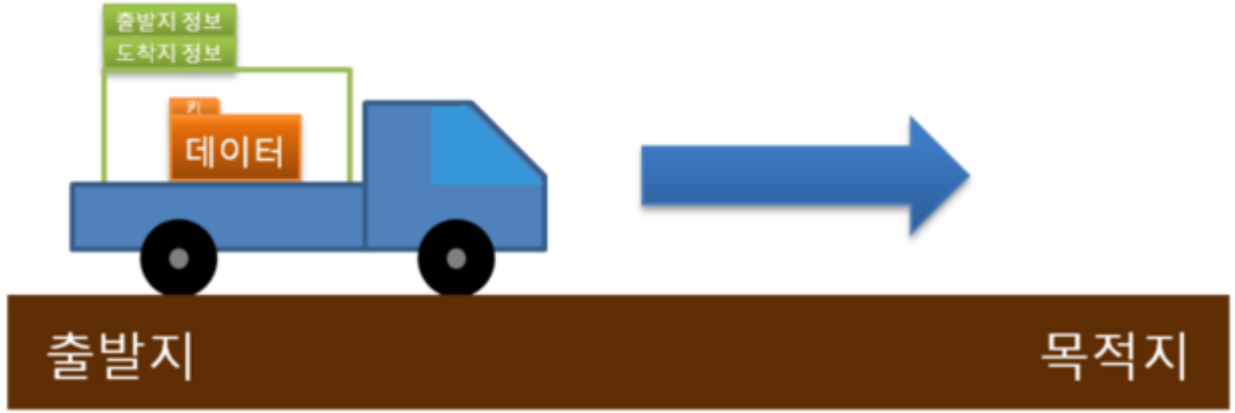


② 인텐트에 데이터를 적재한다.

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

③ 인텐트를 목적지 액티비티로 발송한다.

```
startActivity(intent);
```

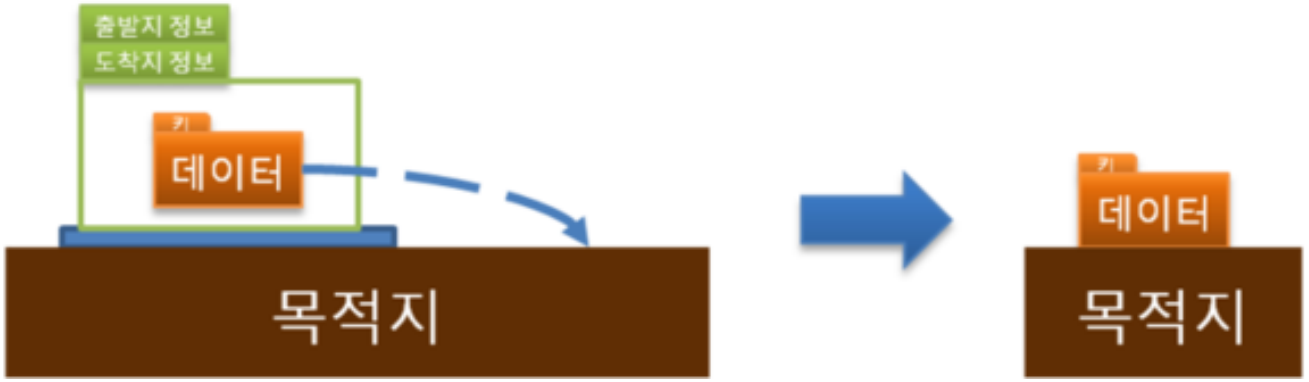


④ 인텐트를 트럭에서(?) 내린다.

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

⑤ 인텐트에서 데이터를 꺼낸다.

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



Step 2.2 파일 편집–MainActivity.java

activity_main.xml × | activity_detail_view.xml × | MainActivity.java × | DetailViewActivity.java × | AndroidManifest.xml × | strings.xml ×

```java
package com.example.research.famouspaintings;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void displayPicture(View view){
        int vID = view.getId(); // 클릭한 객체의 ID를 얻는다.

        TextView textView = (TextView) findViewById(vID); // id가 vID인 위젯요소를 가져온다.
        String title = (String) textView.getText(); // 위젯요소의 Text 내용을 가져온다.
        String tag = (String) textView.getTag(); // 위젯요소의 tag 내용을 가져온다.

        // Intent에 저장하여 보내기
        // DetailViewActivity로 보낼 Intent 생성
        Intent it = new Intent(this, DetailViewActivity.class);
        it.putExtra("it_title", title); // Intent에 it_title이라는 이름으로 title 저장
        it.putExtra("it_body", tag);    // Intent에 it_body이라는 이름으로 tag 저장
        startActivity(it);  // DetailViewActivity 화면 시작
    }

}
```

displayPicture 메소드 추가

클래스 속성과 메소드

- 클래스

클래스	설명
Intent	액티비티 호출과 정보 전달에 사용됨.
Uri	Uniform Resource Identifier로서 인터넷에 있는 자원을 나타내는 유일한 주소를 의미

- 메소드

클래스	메소드	설명
Context	void startActivity(intent)	인텐트에서 지정한 액티비티를 실행함
Intent	Intent(String action, Uri uri)	주어진 uri에 대해 주어진 액션을 하는 인텐트를 생성함
	Intent putExtra(String name, String value)	인텐트에 name의 값을 value로 할당함

Step 2.2 파일 편집–DetailViewActivity.java

activity_main.xml | activity_detail_view.xml | MainActivity.java | DetailViewActivity.java | AndroidManifest.xml | strings.xml

```java
package com.example.research.famouspaintings;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;

public class DetailViewActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail_view);

        Intent it = getIntent(); // 현재 Activity로 전송된 Intent를 얻는다.
        String title = it.getStringExtra("it_title"); // it_title 이름으로 저장된 내용 가져온다.
        String body = it.getStringExtra("it_body"); // it_body 이름으로 저장된 내용 가져온다.

        // TextView에 내용 입력하기
        TextView tvTitle = (TextView) findViewById(R.id.title00);
        tvTitle.setText(title);

        TextView tvBody = (TextView) findViewById(R.id.body00);
        tvBody.setText(title);

    }
}
```

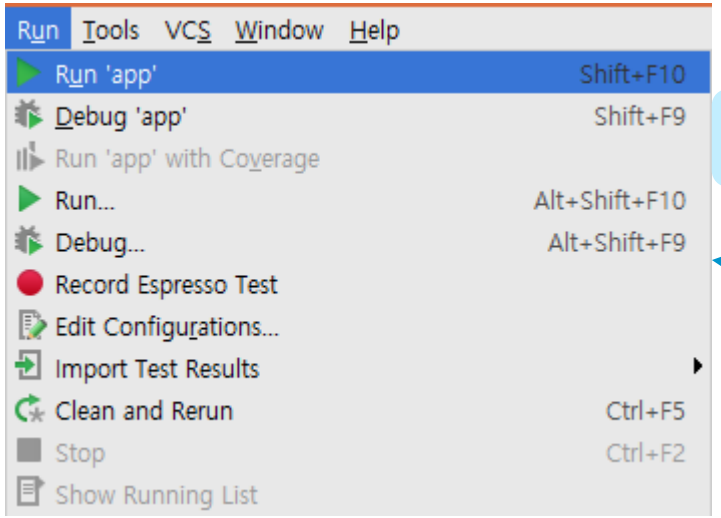
Intent로부터 받아온 내용을 TextView에 설정

클래스 속성과 메소드

- 클래스
  - 없음
- 메소드

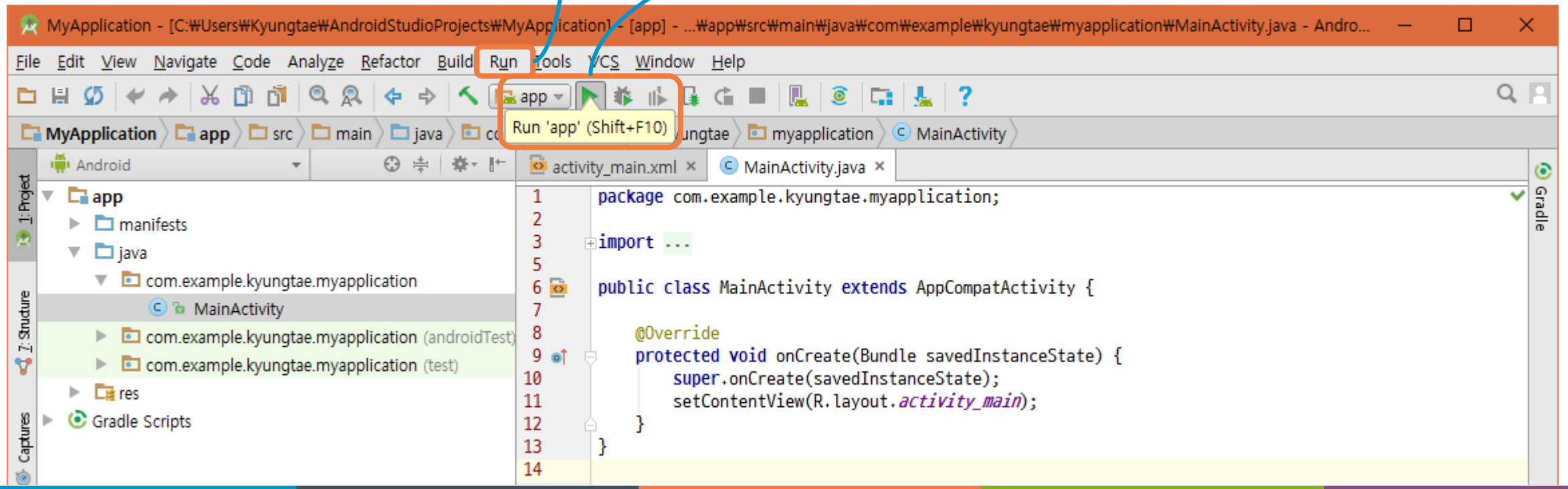
클래스	메소드	설명
Activity	Intent getIntent()	액티비티를 실행한 인텐트를 반환함
Intent	String getStringExtra(String name)	인텐트로부터 name에 해당하는 값을 추출
TextView	final void setText(CharSequence text)	TextView의 문자열을 text로 설정

Step 3. 프로젝트 실행

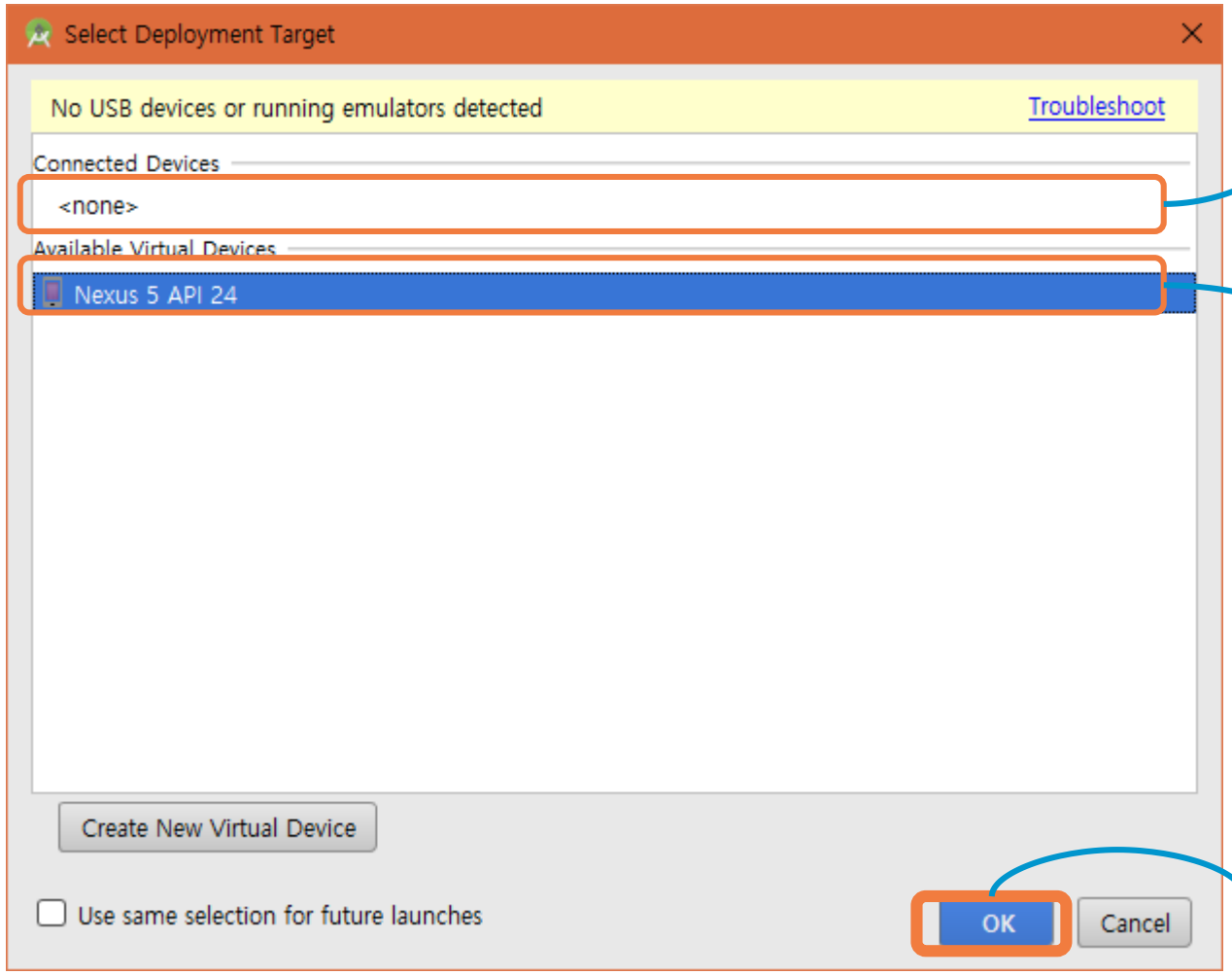


Run → Run 'app' 메뉴 클릭

앱 실행 아이콘 클릭



• AVD 장비 선택하기



데이터 케이블로 연결된 스마트폰

AVD

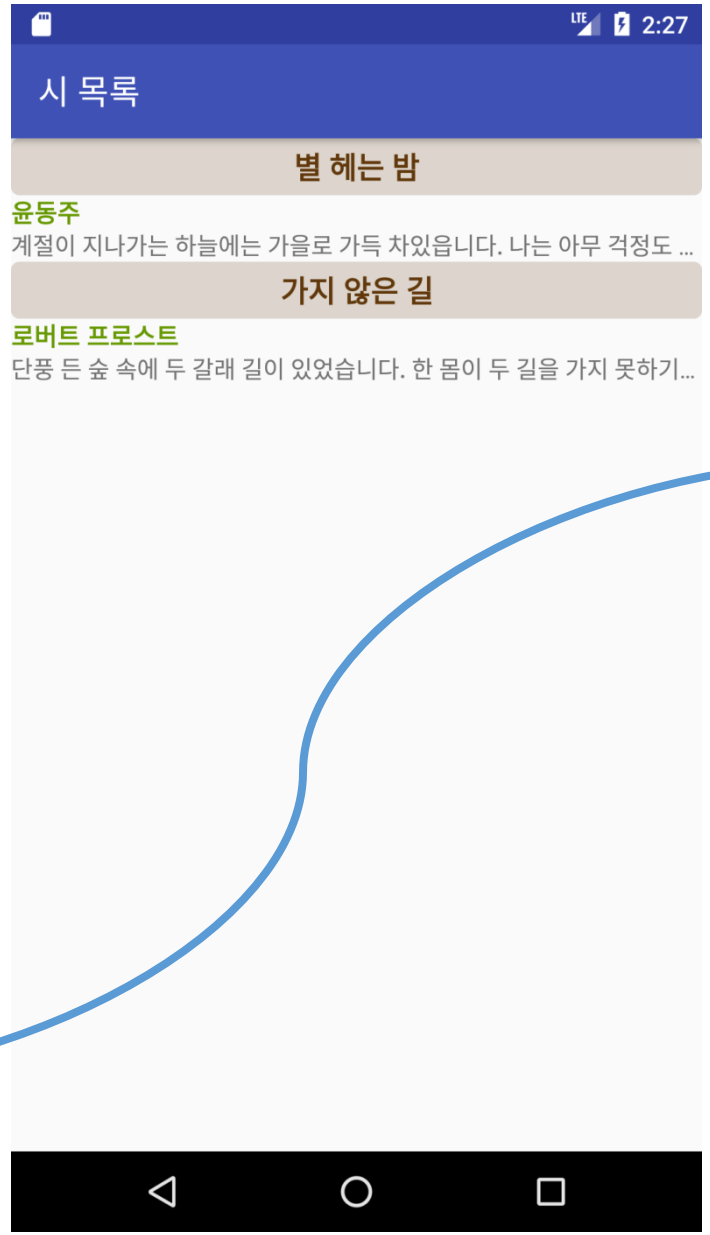
스마트폰 또는 AVD를 선택하고 'OK' 버튼을 클릭

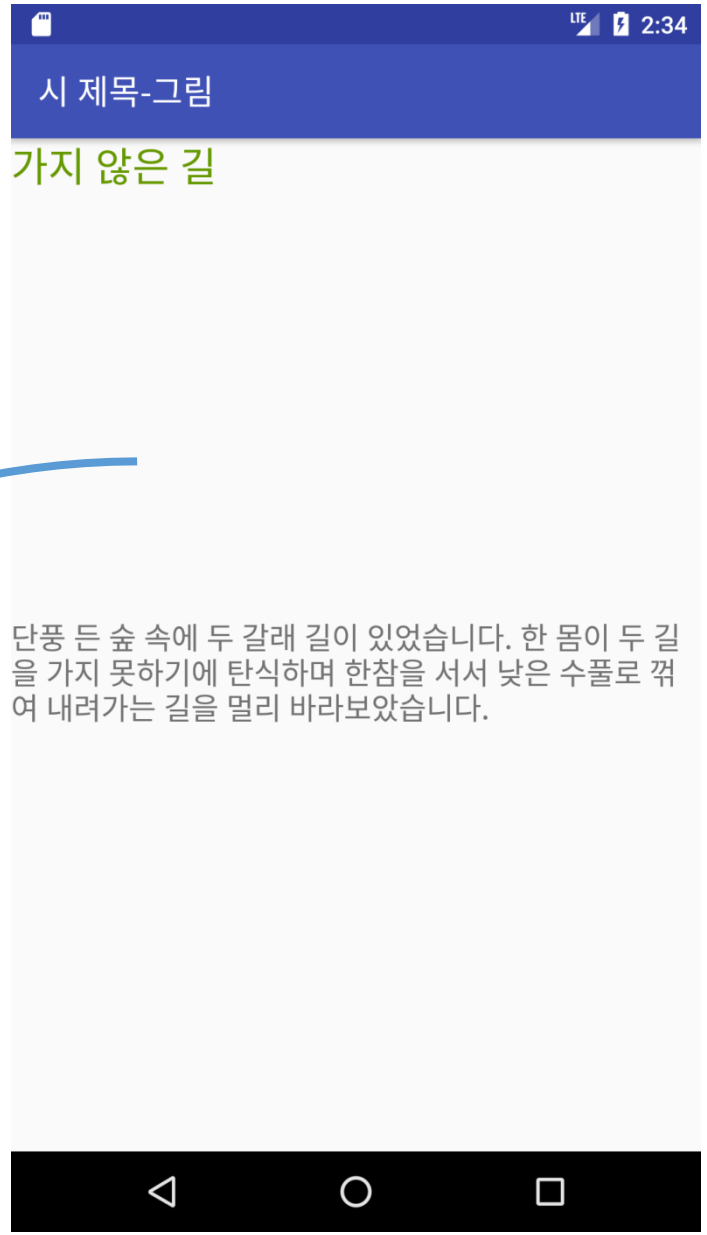
두번째 액티비티

텍스트 대신에 이미지 출력하기

Outputs

Layout의 Height속성을
wrap_content로 변경





Context?

Context

added in API level 1

Summary: Constants | Ctors | Methods | Inherited Methods | [Expand All]

```
public abstract class Context
extends Object
```

java.lang.Object

↳ android.content.Context

Known Direct Subclasses

ContextWrapper, MockContext

Known Indirect Subclasses

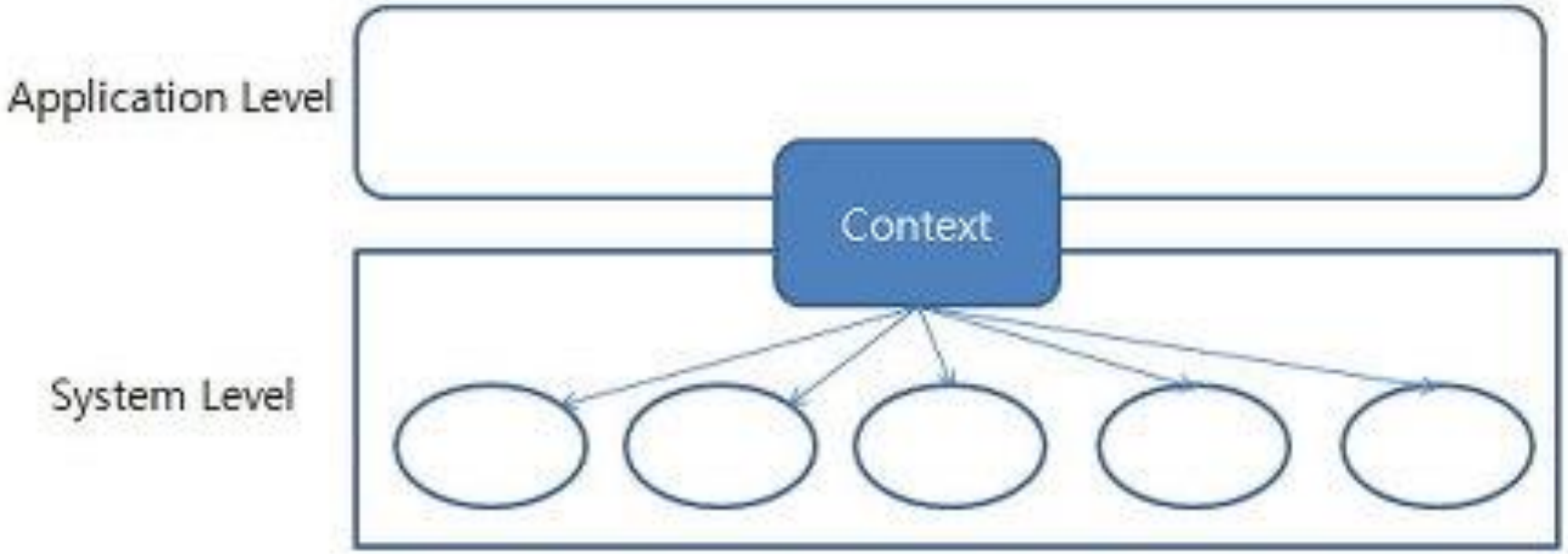
AbstractInputMethodService, AccessibilityService, AccountAuthenticatorActivity, ActionBarActivity, Activity, ActivityGroup, AliasActivity, AppCompatActivity, Application, AutoFillService, BackupAgent, BackupAgentHelper, CallScreeningService, and 48 others.

Interface to global information about an application environment. This is an abstract class whose implementation is provided by the Android system. It allows access to application-specific resources and classes, as well as up-calls for application-level operations such as launching activities, broadcasting and receiving intents, etc.

➔ **Application 환경에 관한 global 정보 접근**을 위한 Interface. Abstract 클래스이며, 구현은 Android 시스템에 의해 제공된다. Context를 통해서 **Application에 특화된 리소스 or 클래스에 접근** 가능하고, 추가적으로 **Application level의 작업(Activity 실행/Intent 브로드 캐스팅/Intent 수신 등)을 수행**하기 위한 API를 호출 할 수도 있다.

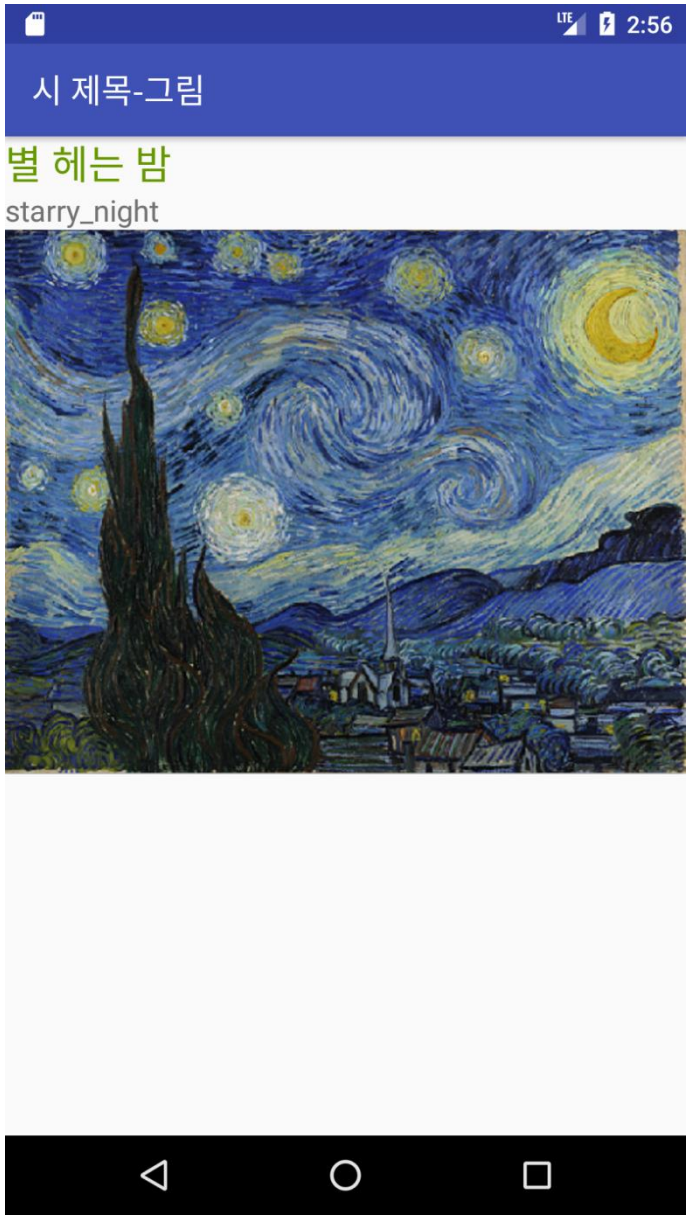
출처: http://wengdiiiy.tistory.com/1 [웽디의 개발블로그]

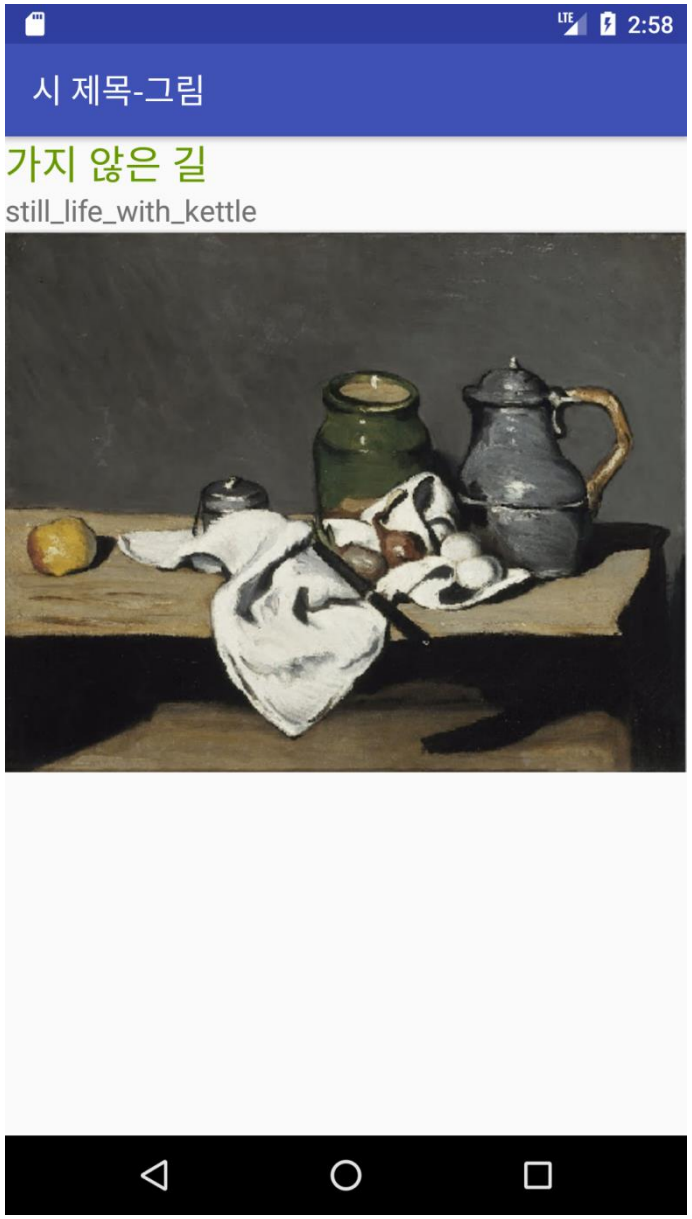
- **Android**에서의 **Context**의 의미?
  - Android의 실행 & 종료 등 관리하는 책임자는 개발자가 아니라,
  - System Level에서 적절히 Life-cycle을 유지해가며 수행시켜 주기 때문에, 실행하기 위한 객체들은 **System Level**에 등록을 해 놓아야 후에 실행이 가능합니다



출처: http://wengdiiiy.tistory.com/1 [웽디의 개발블로그]

확장하기


2:56
시 제목-그림
별 헤는 밤
starry_night

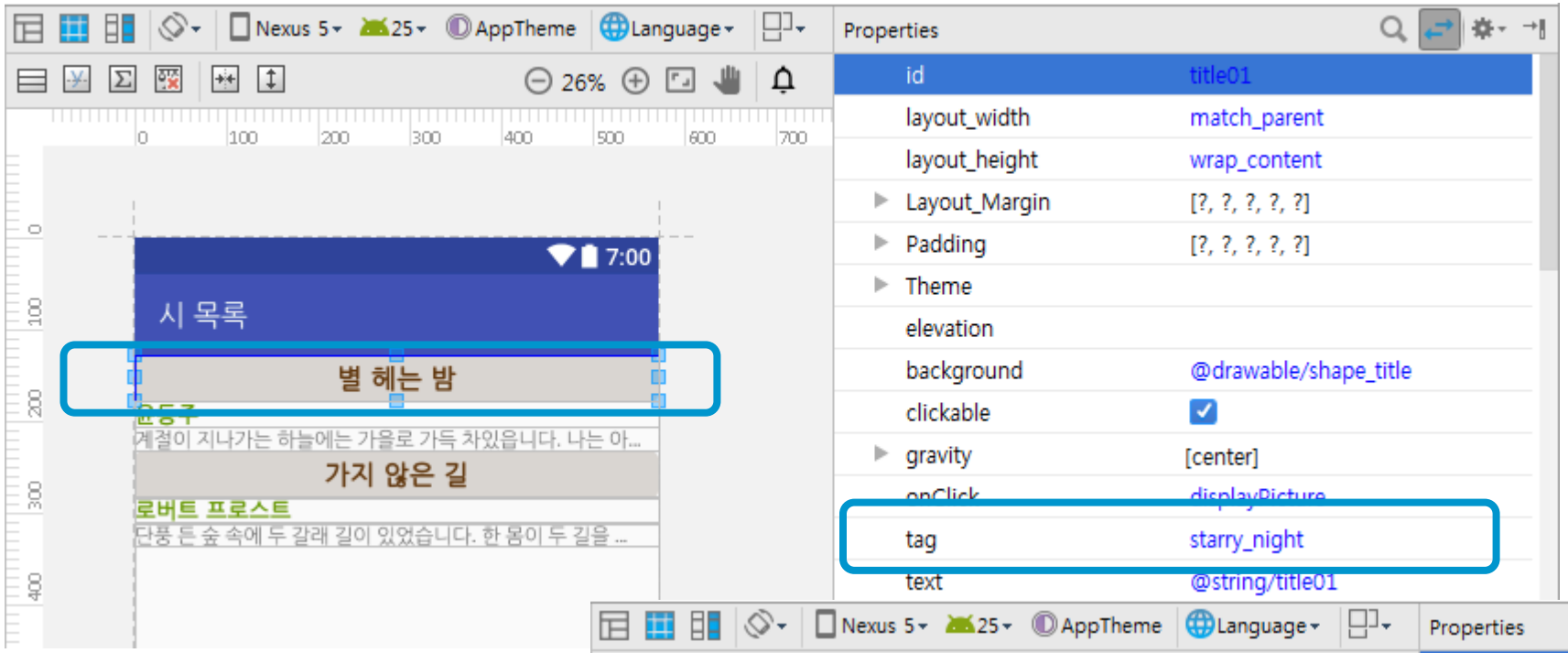

2:58
시 제목-그림
가지 않은 길
still_life_with_kettle

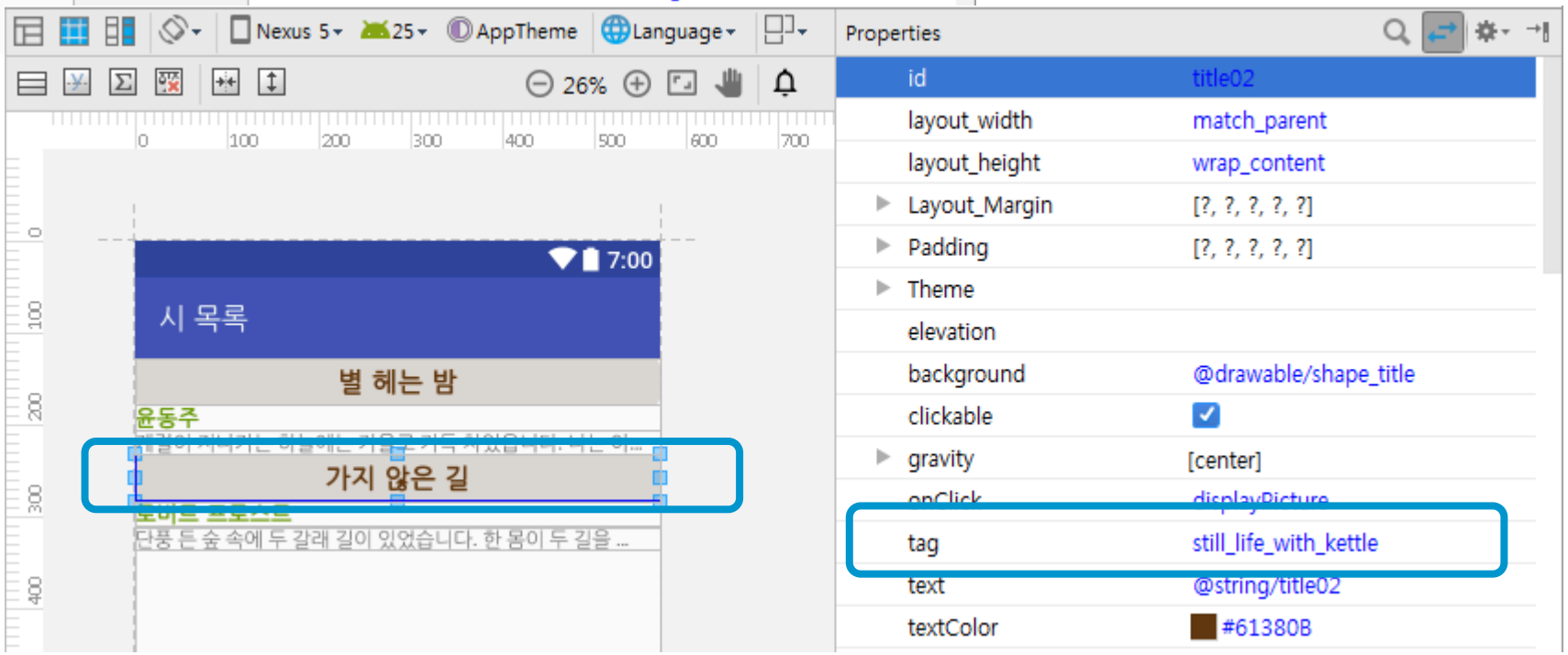
- activity_detail_view.xml 레이아웃 파일에 ImageView 위젯 추가

activity_main.xml | activity_detail_view.xml | MainActivity.java | DetailViewActivity.java | AndroidManifest.xml

```xml
<?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="wrap_content"
    android:orientation="vertical"
    tools:context="com.example.research.famouspaintings.DetailViewActivity">

    <TextView
        android:id="@+id/title00"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:textColor="@android:color/holo_green_dark"
        android:textSize="24sp" />

    <TextView
        android:id="@+id/body00"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:textSize="18sp" />

    <ImageView
        android:id="@+id/picture"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:adjustViewBounds="true" />
</LinearLayout>
```

Text로 입력


Nexus 5
25
AppTheme
Language
26%
0
100
200
300
400
500
600
700
7:00
시 목록
별 헤는 밤
윤동주
계절이 지나가는 하늘에는 가을로 가득 차있읍니다. 나는 아...
가지 않은 길
로버트 프로스트
단풍 든 숲 속에 두 갈래 길이 있었습니다. 한 몸이 두 길을 ...
Properties
id
title01
layout_width
match_parent
layout_height
wrap_content
Layout_Margin
[?, ?, ?, ?, ?]
Padding
[?, ?, ?, ?, ?]
Theme
elevation
background
@drawable/shape_title
clickable
gravity
[center]
onClick
displayPicture
tag
starry_night
text
@string/title01


Nexus 5
25
AppTheme
Language
26%
0
100
200
300
400
500
600
700
7:00
시 목록
별 헤는 밤
윤동주
가지 않은 길
로버트 프로스트
단풍 든 숲 속에 두 갈래 길이 있었습니다. 한 몸이 두 길을 ...
Properties
id
title02
layout_width
match_parent
layout_height
wrap_content
Layout_Margin
[?, ?, ?, ?, ?]
Padding
[?, ?, ?, ?, ?]
Theme
elevation
background
@drawable/shape_title
clickable
gravity
[center]
onClick
displayPicture
tag
still_life_with_kettle
text
@string/title02
textColor
#61380B

activity_main.xml × | activity_detail_view.xml × | MainActivity.java × | DetailViewActivity.java × | AndroidManifest.xml × | strings.xml ×

```java
package com.example.research.famouspaintings;

import android.content.Intent;
import android.content.res.Resources;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.ImageView;
import android.widget.TextView;

public class DetailViewActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail_view);

        Intent it = getIntent(); // 현재 Activity로 전송된 Intent를 얻는다.
        String title = it.getStringExtra("it_title"); // it_title 이름으로 저장된 내용 가져온다.
        String body = it.getStringExtra("it_body"); // it_body 이름으로 저장된 내용 가져온다.

        // TextView에 내용 입력하기
        TextView tvTitle = (TextView) findViewById(R.id.title00);
        tvTitle.setText(title);

        TextView tvBody = (TextView) findViewById(R.id.body00);
        tvBody.setText(body);

        Resources res = getResources(); // 어플리케이션의 리소스 객체 생성
        // 이미지를 표시할 ImageView의 객체를 가져온다.
        ImageView ivPicture = (ImageView) findViewById(R.id.picture);
        // 이름이 body인 drawable 객체의 id를 얻는다.
        int id_img = res.getIdentifier(body, "drawable", getPackageName());
        // id가 id_img인 이미지 리소스를 ImageView의 리소스로 사용한다.
        ivPicture.setImageResource(id_img);

    }
}
```

추가된 소스

Question & Answer