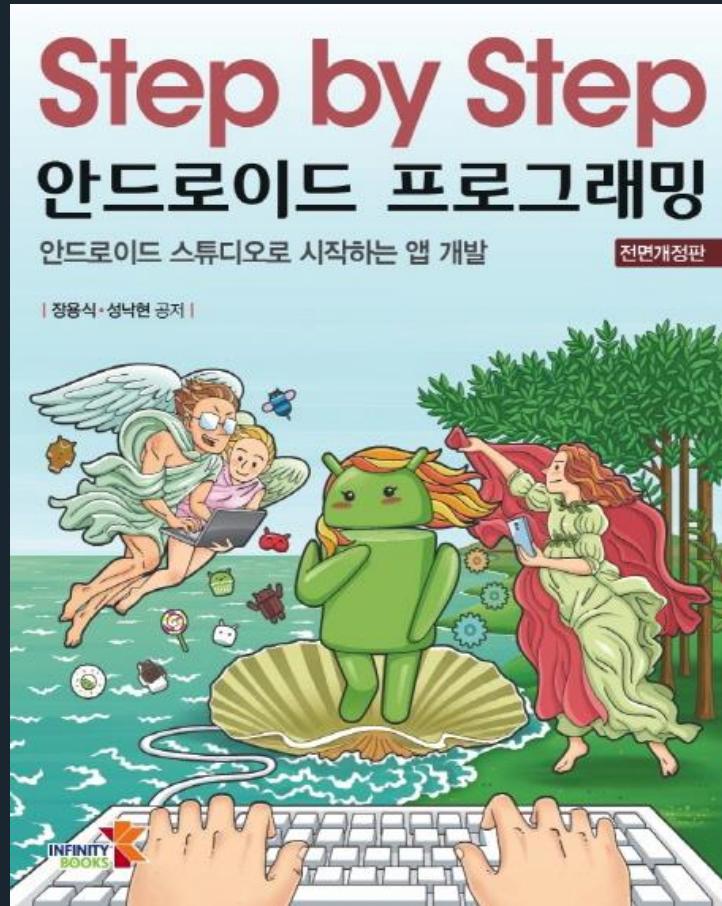


앱의 프로젝트 구조와 실행원리



강의자료 – <https://github.com/hopypark>

The screenshot shows a Microsoft Edge browser window with multiple tabs open. The active tab is for the GitHub profile of user 'hopypark' at <https://github.com/hopypark>. The GitHub interface displays the user's profile picture (a person in a cowboy hat outdoors), the name 'hopypark', and two buttons: 'Add a bio' and 'Edit profile'. Below the profile picture, it says '134 contributions in the last year' and shows a heatmap of activity by month and day. A pinned repository named 'Lecture2018' is highlighted with a red box. On the right, there's a beta feature announcement about activity overview across repositories and organizations, with a 'Got it!' button.

Overview Repositories 6 Stars 0 Followers 0 Following 0

Pinned repositories

Lecture2018

Lecture notes of the first half of 2018

hopypark

Add a bio

Edit profile

134 contributions in the last year

Contribution settings ▾

Sep Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug

Mon
Wed
Fri

Learn how we count contributions.

Less More

The screenshot shows a GitHub repository page for 'hopypark/Lecture2018'. The repository has 24 commits, 1 branch, 0 releases, 1 contributor, and is licensed under MIT. A red box highlights the commit for 'AndroidApp' which was added 6 days ago. The repository contains files like 'LICENSE', 'README.md', and 'README.md' (the last one is being edited).

No description, website, or topics provided.

Add topics

24 commits 1 branch 0 releases 1 contributor MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

hopypark Add files via upload Latest commit 813b1b3 2 days ago

AndroidApp	Add files via upload	6 days ago
Appinventor2	Add files via upload	2 days ago
JavaSpring	Add files via upload	7 days ago
LICENSE	Initial commit	6 months ago
README.md	Update README.md	7 days ago

README.md

Lecture2018

Lecture2018/AndroidApp

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

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

hopypark / Lecture2018 Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

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

hopypark Add files via upload Latest commit 4b14af2 20 hours ago

..

Week01.강의 소개.pdf Add files via upload 7 days ago

Week02.Chap02.앱 개발환경 구축.pdf Add files via upload 20 hours ago

Week02.Chap03.앱 프로젝트 구조와 실행원리.pdf Add files via upload 20 hours ago

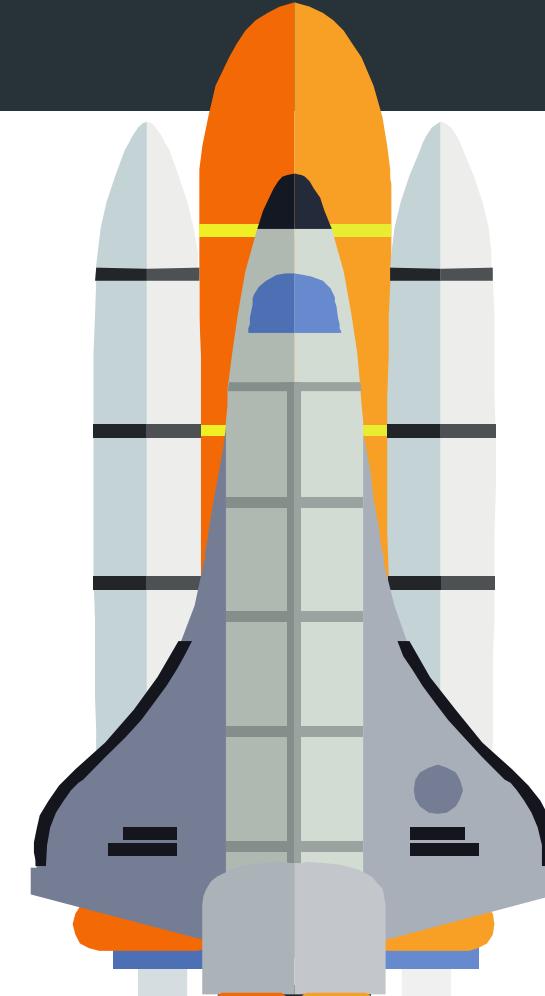
readme.md Create readme.md 7 days ago

readme.md

© 2018 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

Table of Contents

- 안드로이드 프로그램의 구조
- 프로젝트 예제: Hello Android 프로젝트
- 프로젝트 개발 과정
- 프로젝트 파일 구조
- 앱의 실행 원리
- 프로젝트 소스 간의 연관성
- 어플리케이션 아이콘의 변경



안드로이드 프로그램의 구조

6

액티비티(Activity)	서비스(Service)
<ul style="list-style-type: none">사용자 인터페이스 화면을 가지며, 특정한 작업을 담당하는 컴포넌트사용자로부터 입력을 받고 처리 결과를 화면에 보여주는 기능을 한다.	<ul style="list-style-type: none">사용자와의 상호작용과는 상관없이 항상 백그라운드에서 동작한다(사용자 인터페이스 화면이 없다).음악재생<ul style="list-style-type: none">사용자가 곡을 선택화면과는 별도로 실제로 음악을 재생하는 작업을 진행

방송 수신자(Broadcast receiver)	컨텐트 프로바이더(Content Provider)
<ul style="list-style-type: none">방송을 받고 반응하는 컴포넌트로 대부분 시스템에서 방송이 발생한다.화면이 꺼졌다거나 배터리 사용 가능량의 상태 등을 시스템이 알리기 위해 사용	<ul style="list-style-type: none">데이터를 관리하고 다른 어플리케이션에서도 데이터를 이용할 수 있도록 제공하는 기능

액티비티(ACTIVITY)

- 간단히 말하면 액티비티는 우리가 보는 화면 자체를 말함
- 안드로이드는 태생적 특성인지 모바일에 특화 되어 있음
 - 작은 화면에서 정보를 보여주고 입력을 받아야 한다.
- 동시에 여러 화면을 띄워 줄 수는 없다
 - 최근 동시에 두개의 앱을 실행 시킬 수 있는 기술이 등장했지만, 이것과는 별개
- 한 개의 애플리케이션 안에 여러 개의 액티비티가 있을 수 있지만 동시에 여러 개가 동작하지 않고 반드시 한 가지 액티비티만 동작한다.





서비스(SERVICE)

- 액티비티는 시작이 되면 언젠가 화면에서 사라지며 종료가 됨
- 그러나, **서비스는 앱이 종료가 된 이후에도 계속 동작하고 있는 것을 말한다.** 따라서 서비스는 액티비티를 반드시 필요로 하지 않는다.
 - 예를 들자면, MP3 플레이어 앱

방송 수신자(BROADCAST RECEIVER)

- 방송을 받고 반응하는 컴포넌트로 대부분 시스템에서 방송이 발생한다.
 - 화면이 꺼졌다거나 배터리 사용 가능량의 상태 등을 시스템이 알리기 위해 사용
 - 어떤 데이터가 다운로드 되어서 사용가능하게 되었다는 것을 다른 어플리케이션에 알릴 목적으로 방송을 사용할 수도 있다.
- 방송수신자
 - 방송 수신자가 수신된 정보에 반응하여 새로운 액티비티를 시작할 수 있음
 - 사용자에게 경고를 보내기 위하여 통지 매니저(Notification Manager)를 사용할 수 있음
 - 예) 백라이트를 점멸, 장치를 진동시키거나 사운드를 재생





컨텐트 프로바이더(CONTENT PROVIDER)

- 컨텐츠 프로바이더는 **여러 앱들의 데이터 공유를 위한 인터페이스를 제공**하는 것
- 앱과 앱 저장소 사이에서 데이터 접근을 쉽게 하도록 관리.
- 데이터는 파일 시스템이나 SQLite 데이터베이스에 저장될 수 있다.
 - 컨텐트 프로바이더를 사용하여 다른 어플리케이션에서 데이터를 쿼리하거나 변경할 수 있다.
 - 예를 들어 주소록에 저장되어 있는 데이터를 기반으로 카카오톡의 친구가 자동으로 등록되어 있는 것과 같은 원리임.

인텐트(Intent)

11

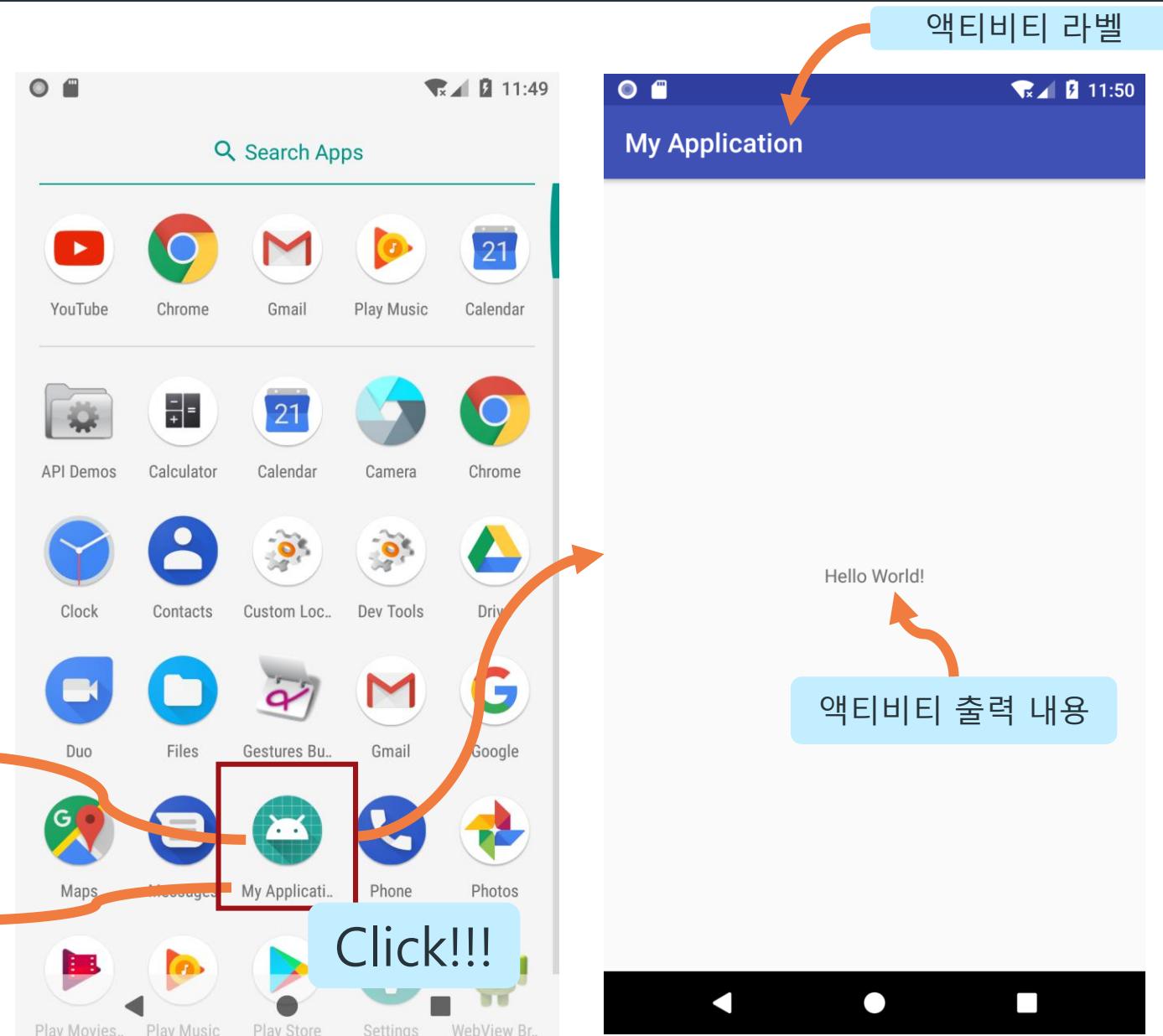
- 인텐트는 "의도 " 라는 의미
- 어플리케이션이 의도를 적어서 안드로이드에 전달하면 안드로이드가 가장 적절한 컴포넌트를 찾아서 활성화하고 실행
 - 즉, A라는 어플리케이션이 의도한 내용을 처리할 B어플리케이션을 호출하여 실행시키고 필요한 메시지를 전달
 - 예) 이미지를 화면에 표시하거나 웹페이지를 오픈하는 요청을 기술 가능

3.1 프로젝트 예제

Hello Android 프로젝트(프로젝트의 시작)

Step 1. Hello Android 프로젝트

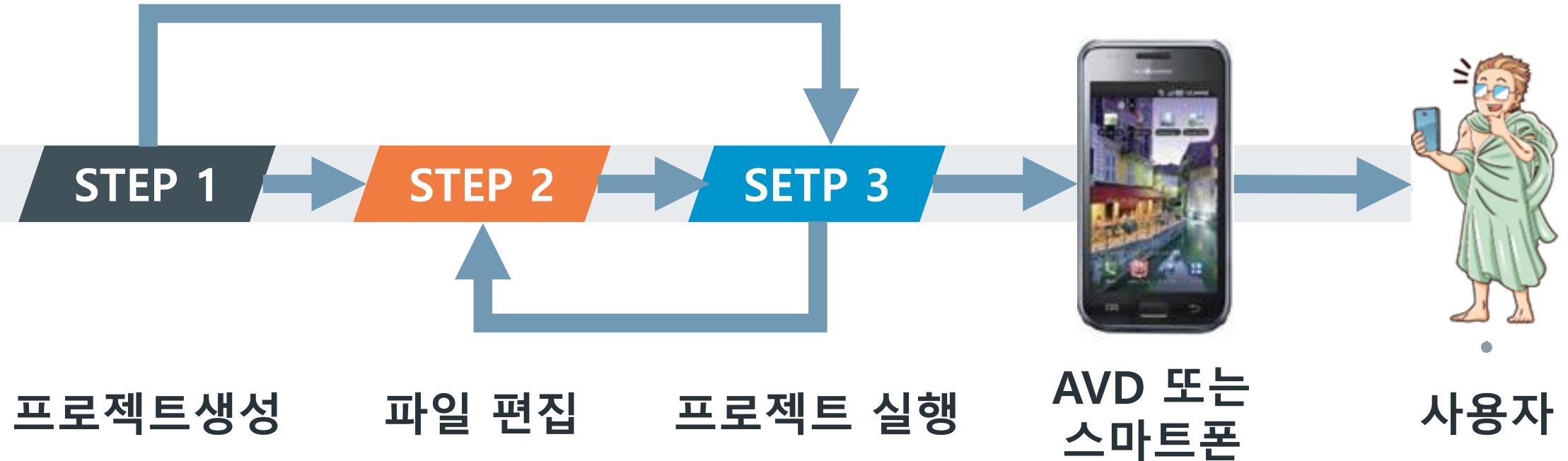
- **프로젝트 개요:** "Hello World!"
- **Application Name:** My Application
- **어플리케이션 라벨:** 안녕
(기본적으로 액티비티 라벨과 같지만 다르게 지정가능)



3.2 프로젝트 개발

Step 0. 프로젝트 개발 과정

16



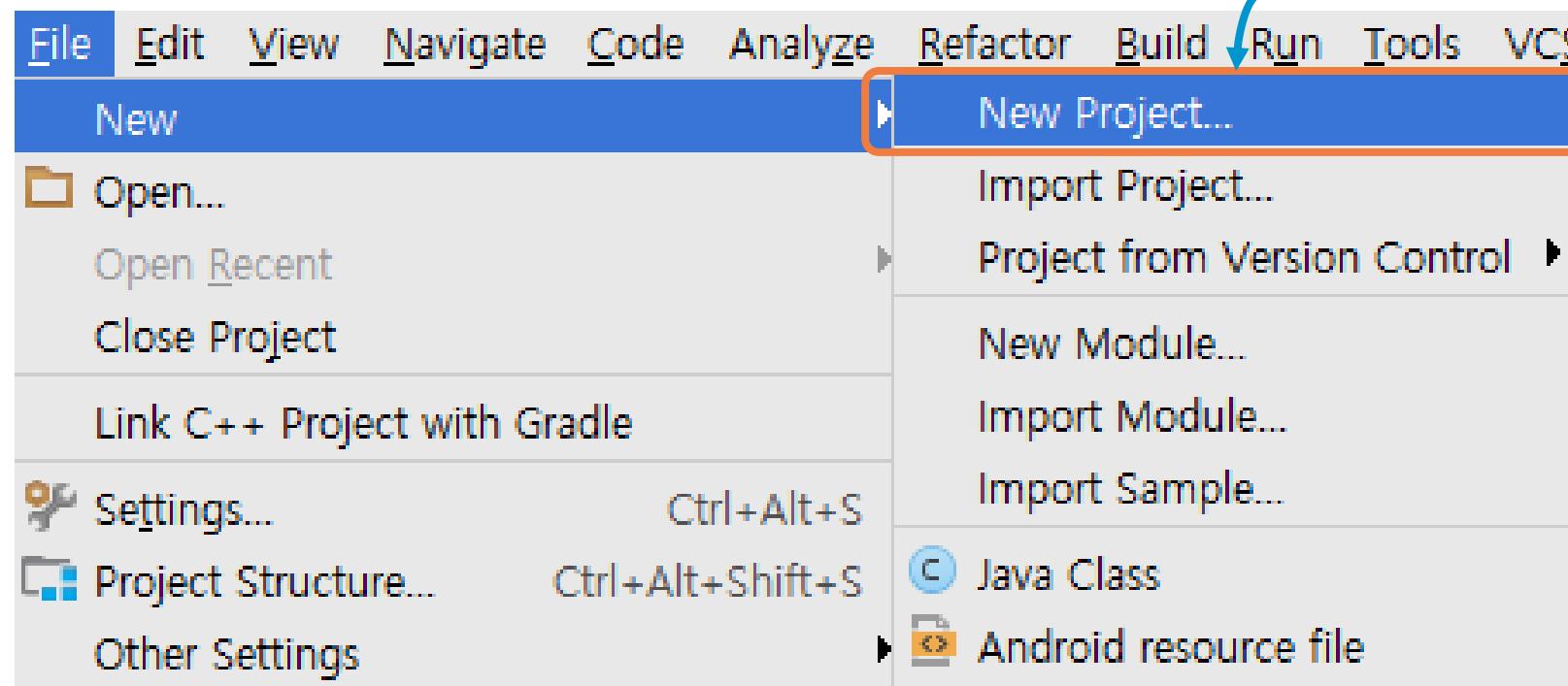
Step 1. 프로젝트 생성

17

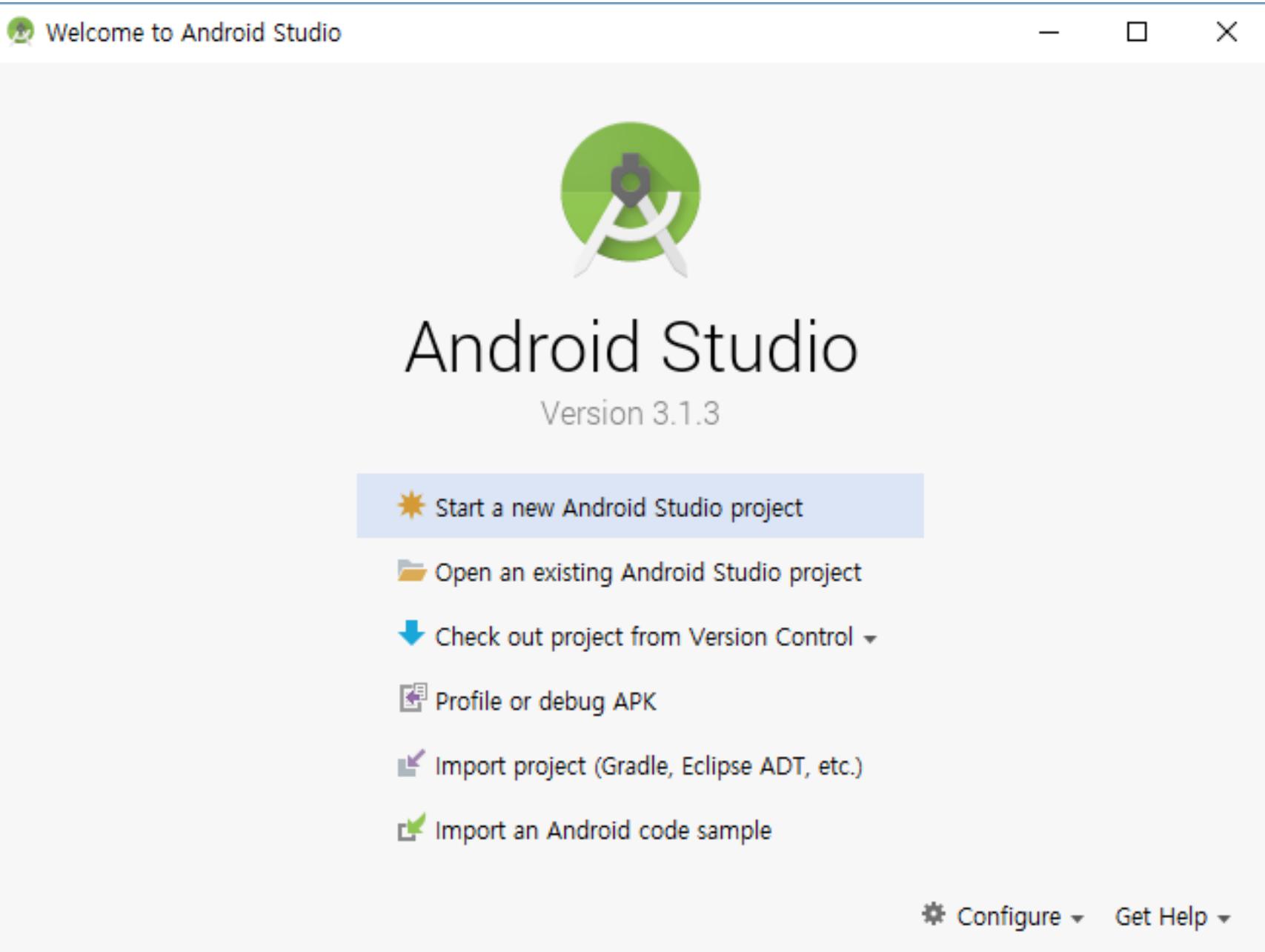
- 프로젝트는 다음 5단계 과정을 거쳐 만들게 된다.

① 프로젝트 시작

Create new project

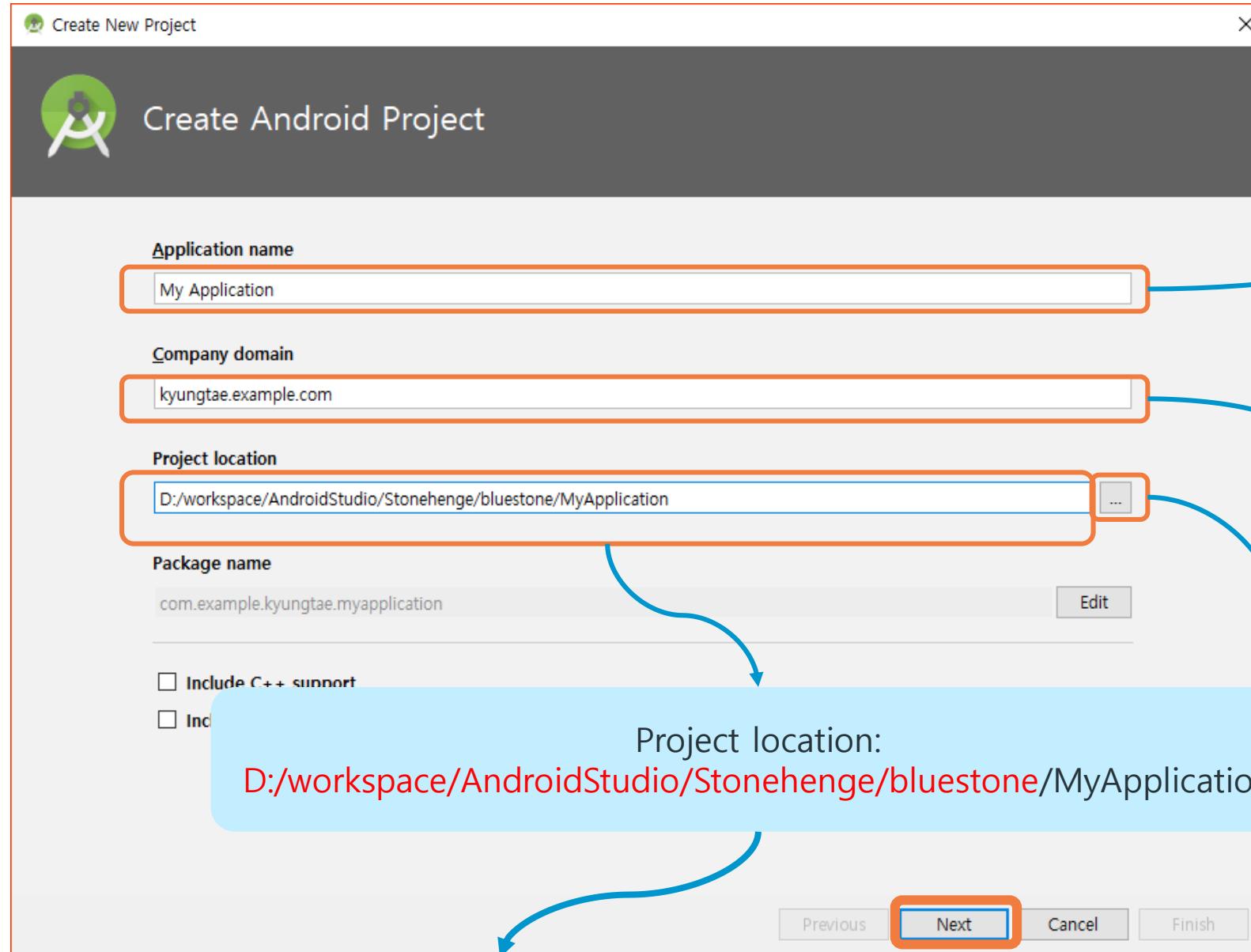


또는



② 프로젝트 구성

19



Application name:
"My Application"

Company Domain:
"username.example.com"
으로 지정됨

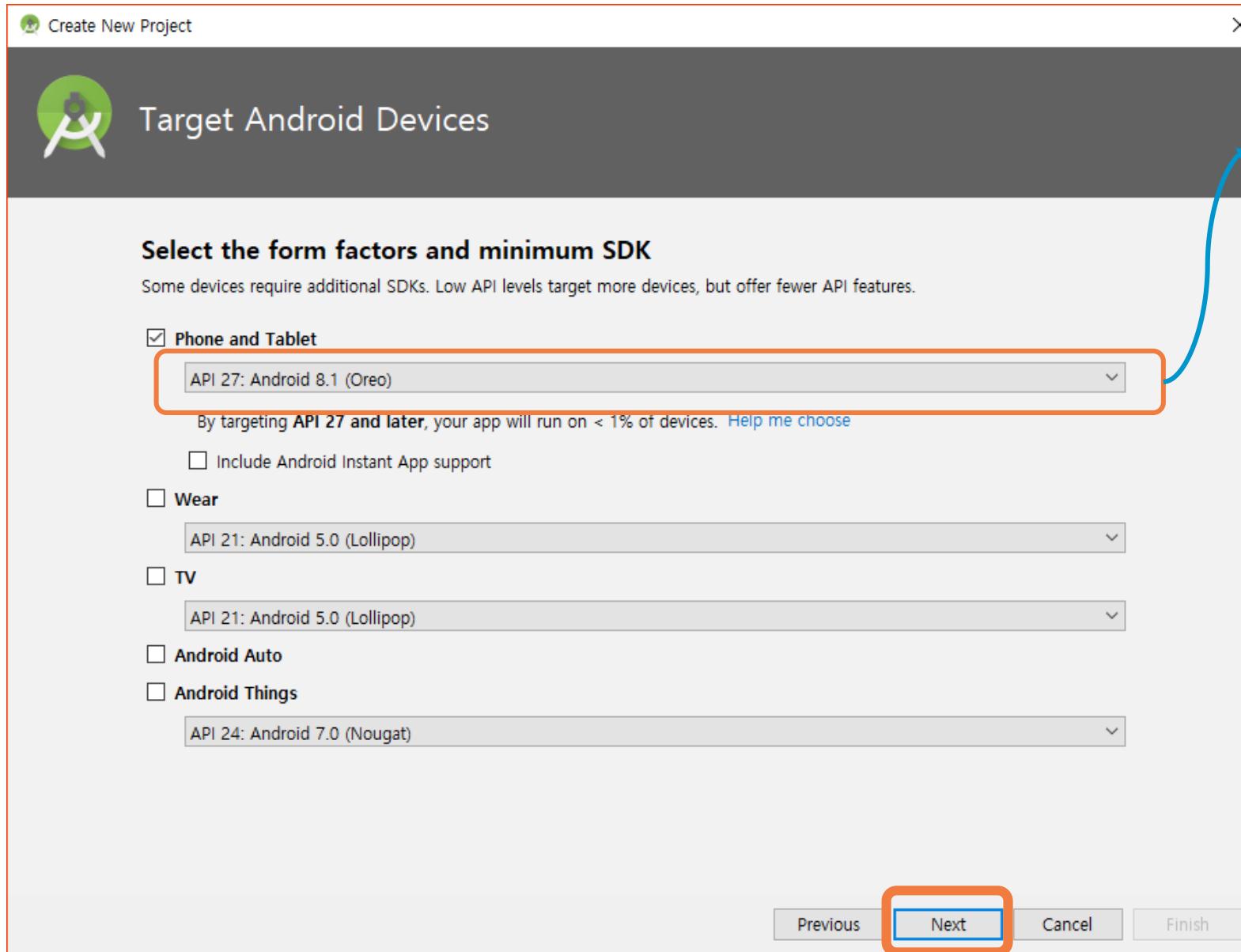
폴더 위치 변경 버튼

Project location:
D:/workspace/AndroidStudio/Stonehenge/bluestone/MyApplication

빨간색 부분은 개인별로 상이함.

③ 실행 디바이스

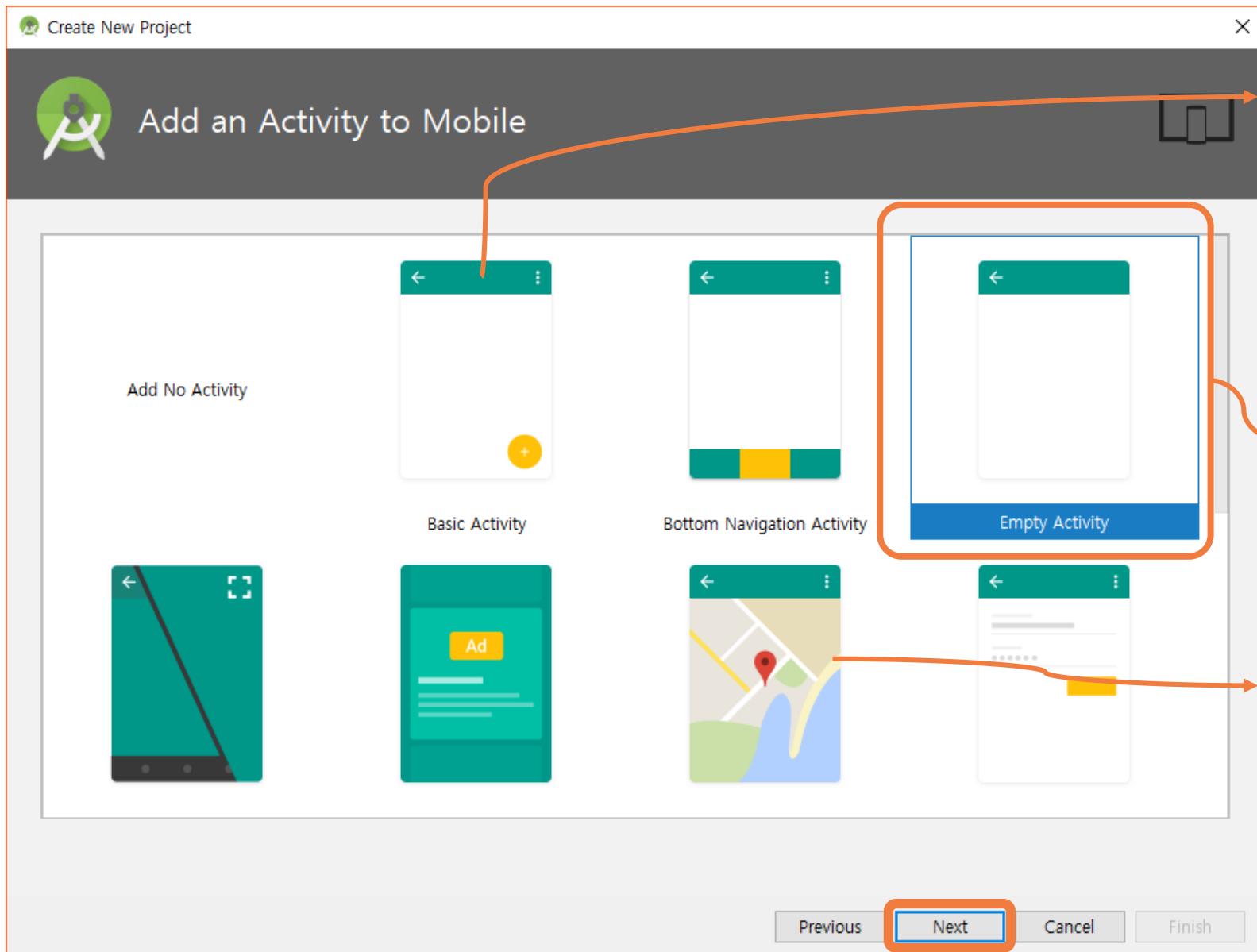
20



실행할 스마트폰의 안드로이드 버전을 고려한 SDK 버전 설정:
Android 8.1 (Oreo)

④ 액티비티 유형(Empty Activity)

21



Basic Activity

액티비티 라벨을 표시하는 **타이틀 바**(좌측 상단)와 네비게이션이 가능하도록 메뉴가 나타나는 **액션 바**(우측 상단) 외에 화면 위에 떠다니는 버튼인 **Floating Action Button**(우측하단)으로 구성

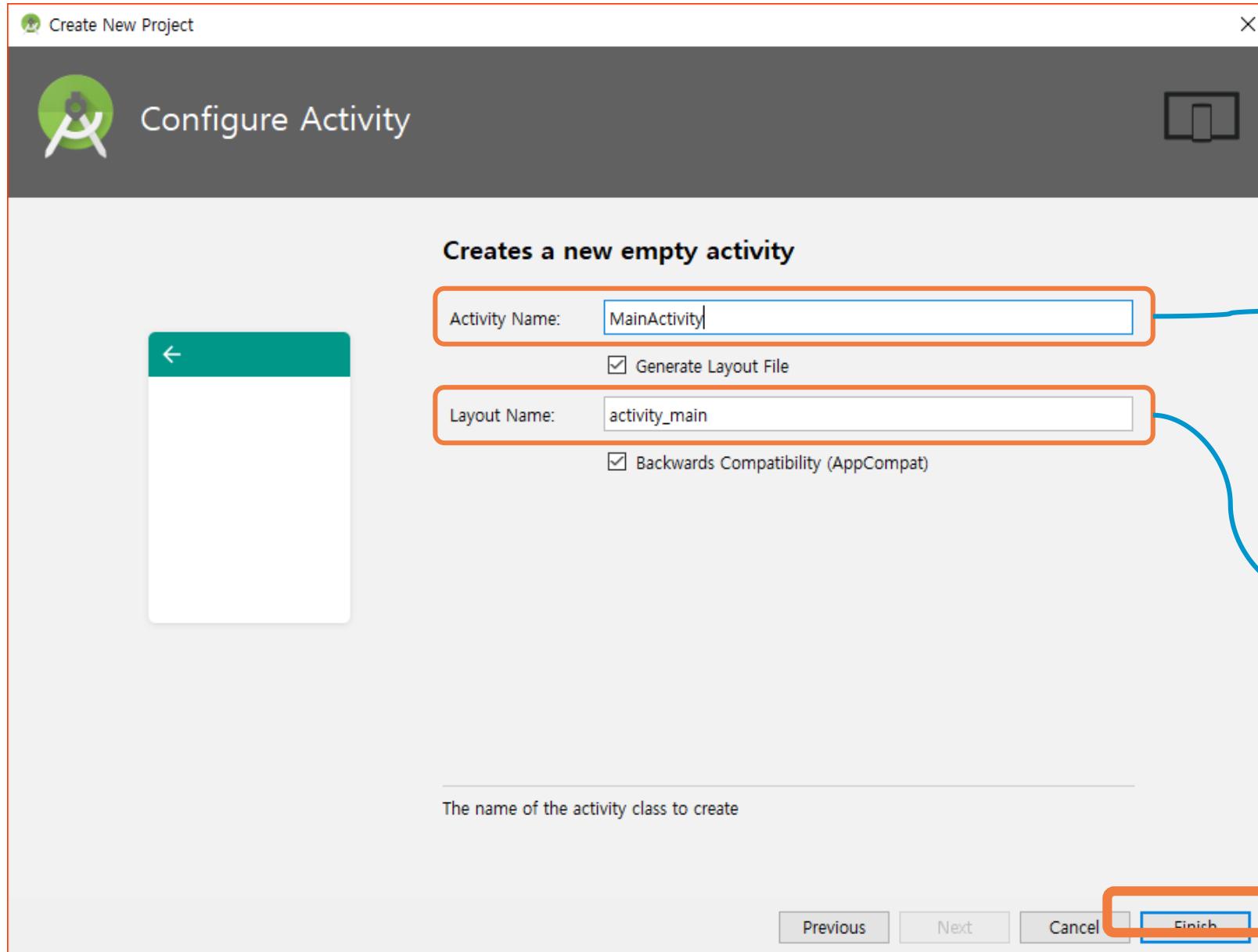
Empty Activity

액티비티 라벨을 표시하는 **타이틀 바**만 나타나는 가장 단순한 화면 구성
(대부분의 예제에서 사용)

Google Map Activity

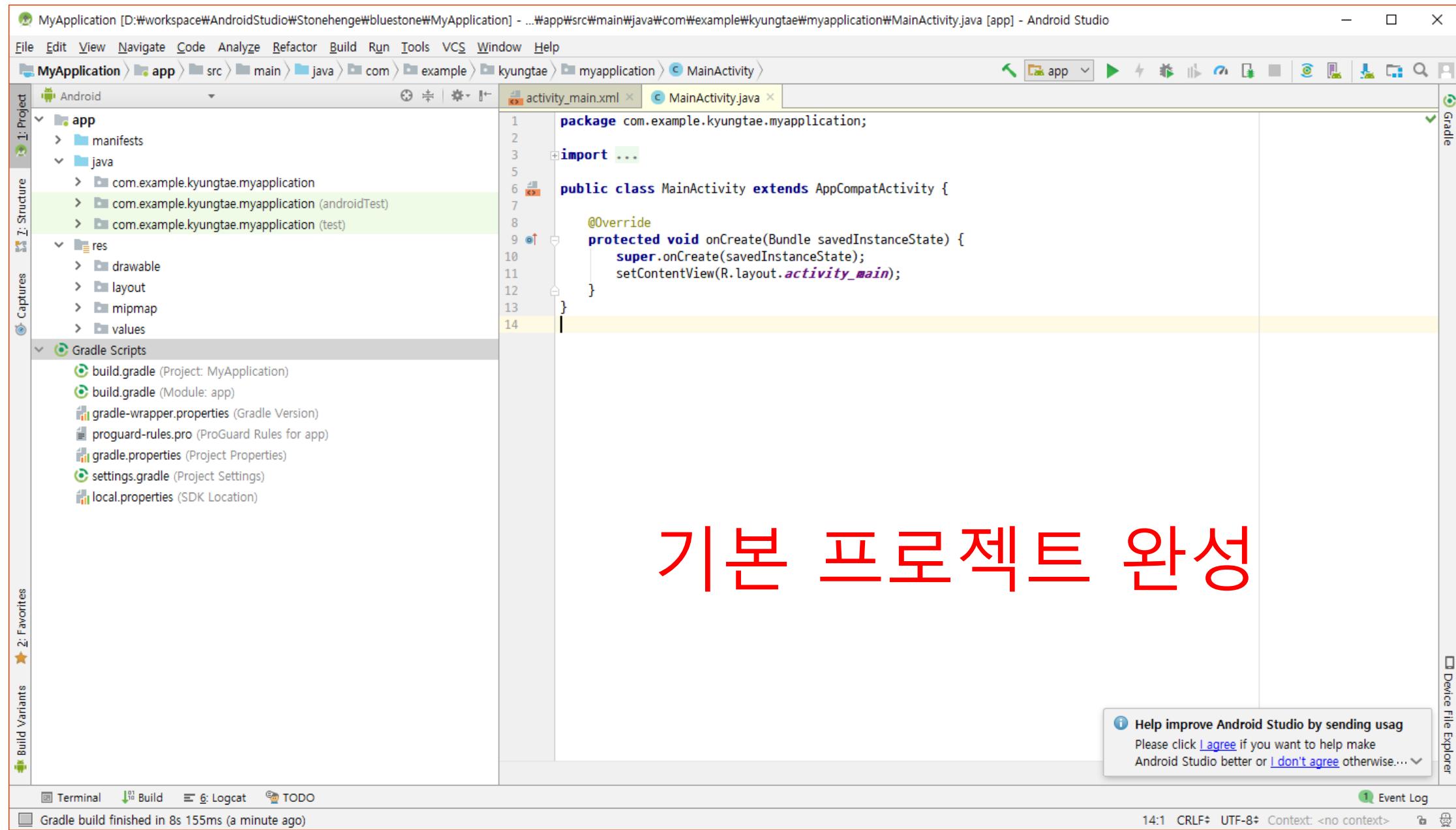
액티비티 라벨을 표시하는 **타이틀 바**와 **액션바**가 나타나며, 구글맵을 표시할 수 있는 화면 구성

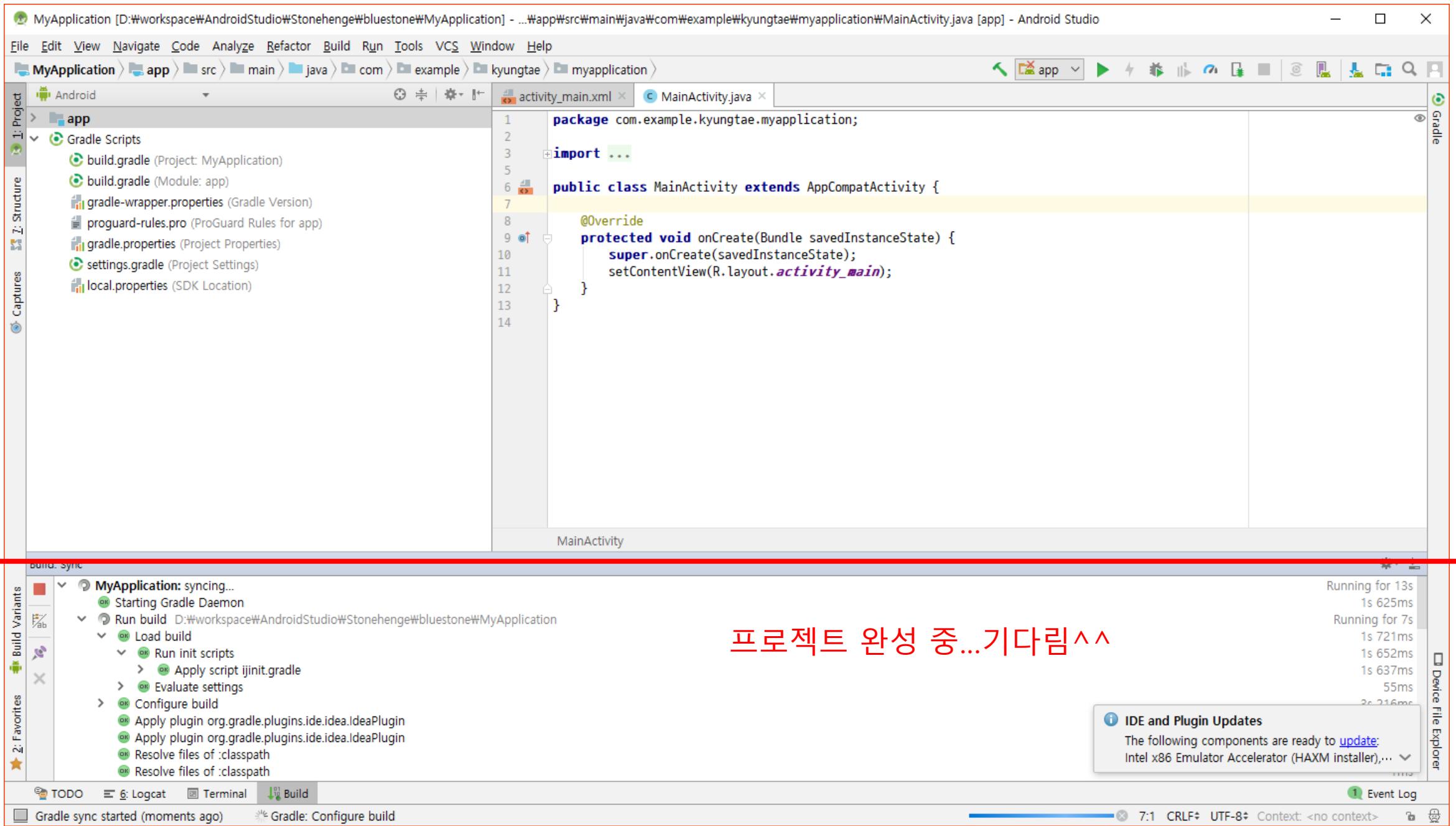
⑤ 액티비티 구성 파일 이름 설정



앱이 실행 시 처음 실행되는 액티비티 이름(**그대로 둠**): MainActivity

액티비티에 출력될 레이아웃 이름
(**그대로 둠**): activity_main





프로젝트 생성 완료 – 프로젝트 보기

MainActivity.java 클래스

The screenshot shows a development environment with two tabs open: 'activity_main.xml' and 'MainActivity.java'. The 'MainActivity.java' tab is active, displaying the following Java code:

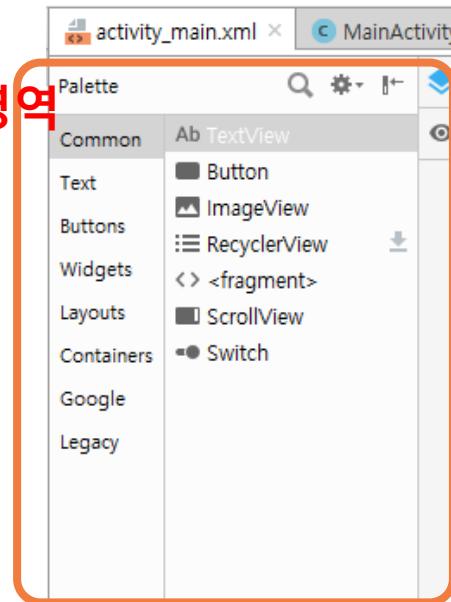
```
1 package com.example.research.myapplication;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.os.Bundle;
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13 }
14
```

The code defines a class named 'MainActivity' that extends 'AppCompatActivity'. It overrides the 'onCreate' method to set the content view to 'activity_main'. The code is numbered from 1 to 14. Line 14 is highlighted with a yellow background.

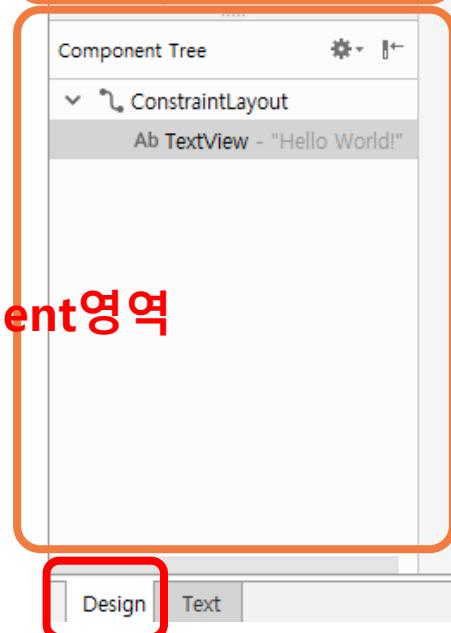
activity_main.xml – Design view

27

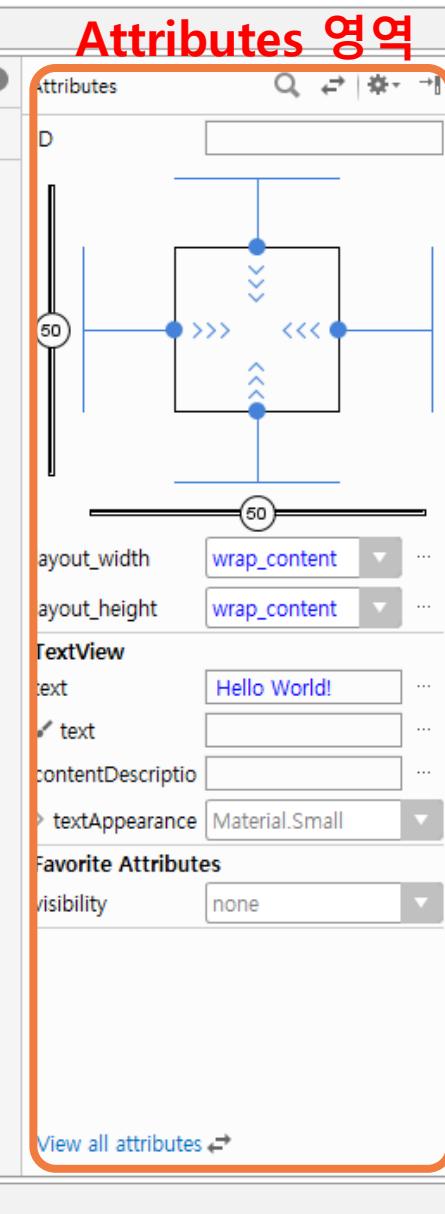
Palette 영역



Component 영역

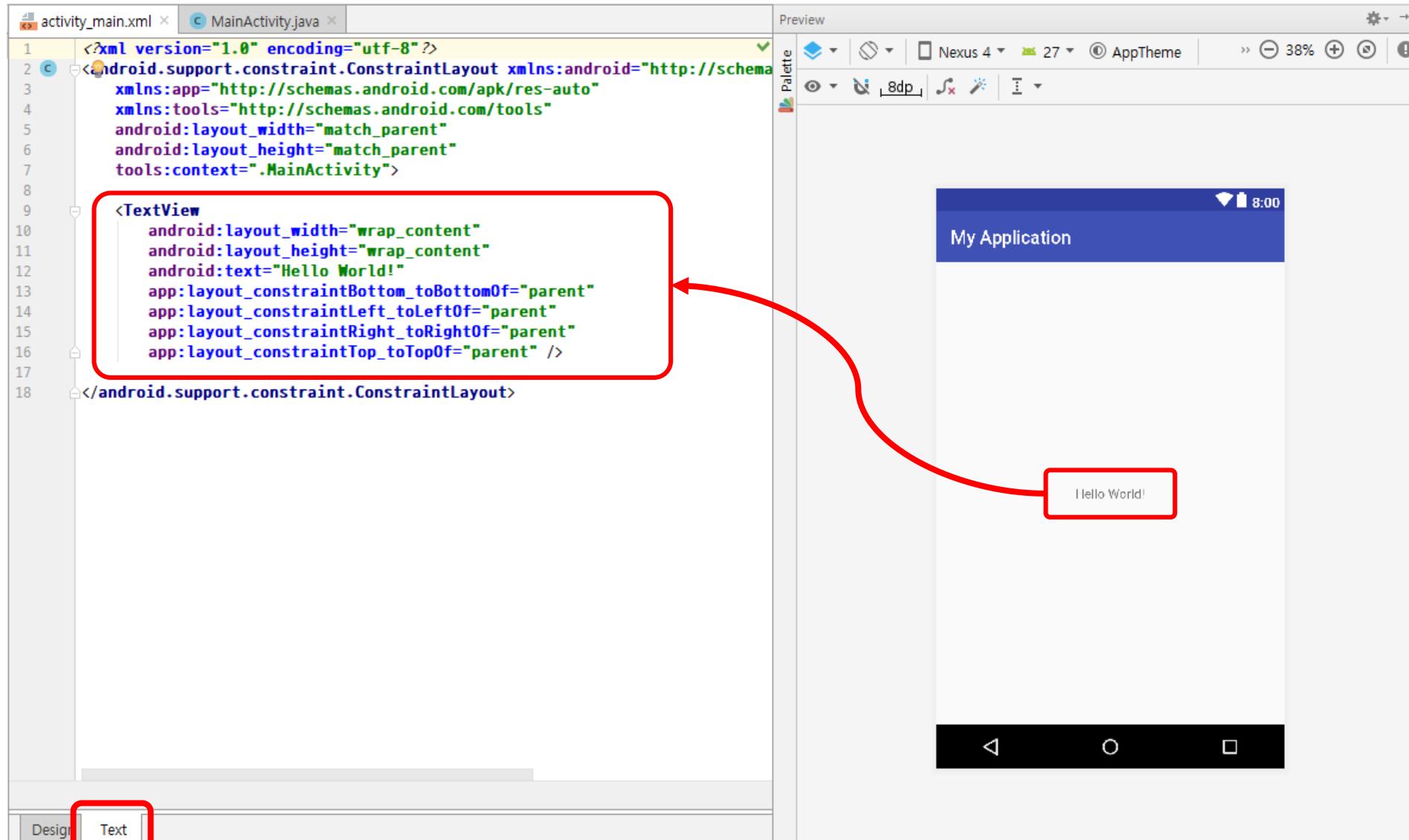


Attributes 영역



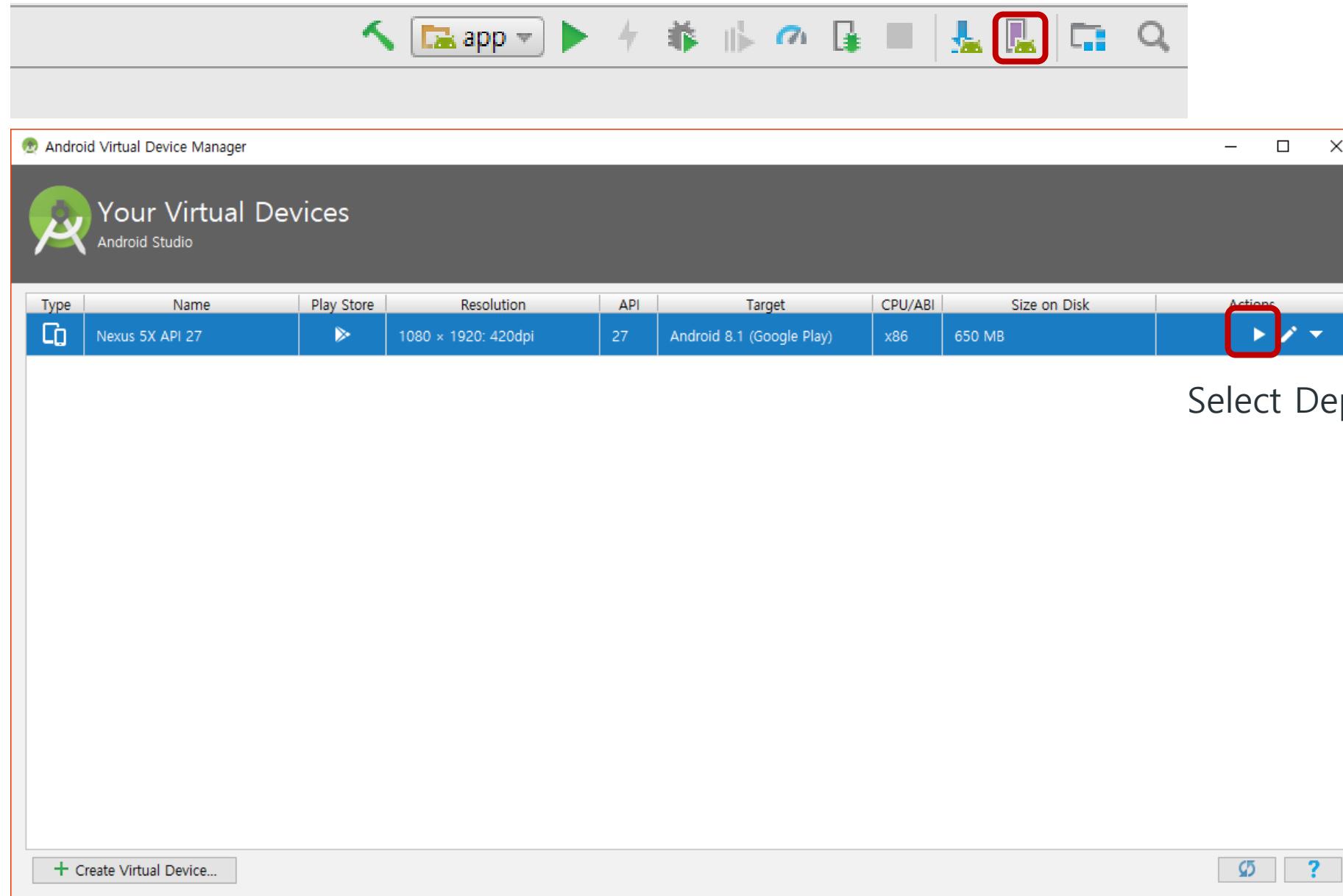
activity_main.xml – Text view

- 안드로이드는 기본적으로 java language로 만들어 지지만 디자인과 관련된 layout, property등 리소스의 상당부분이 **xml 문서**로 작성



우선, AVD Manager 실행

29



AVD 실행 화면

30

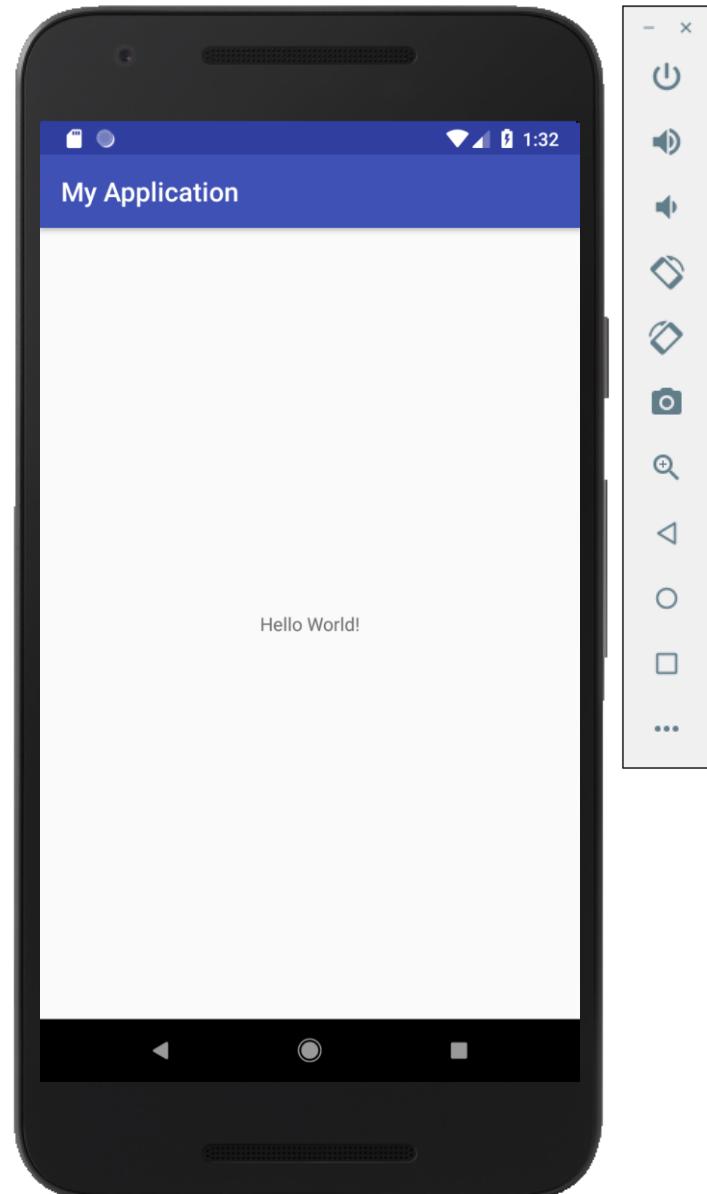
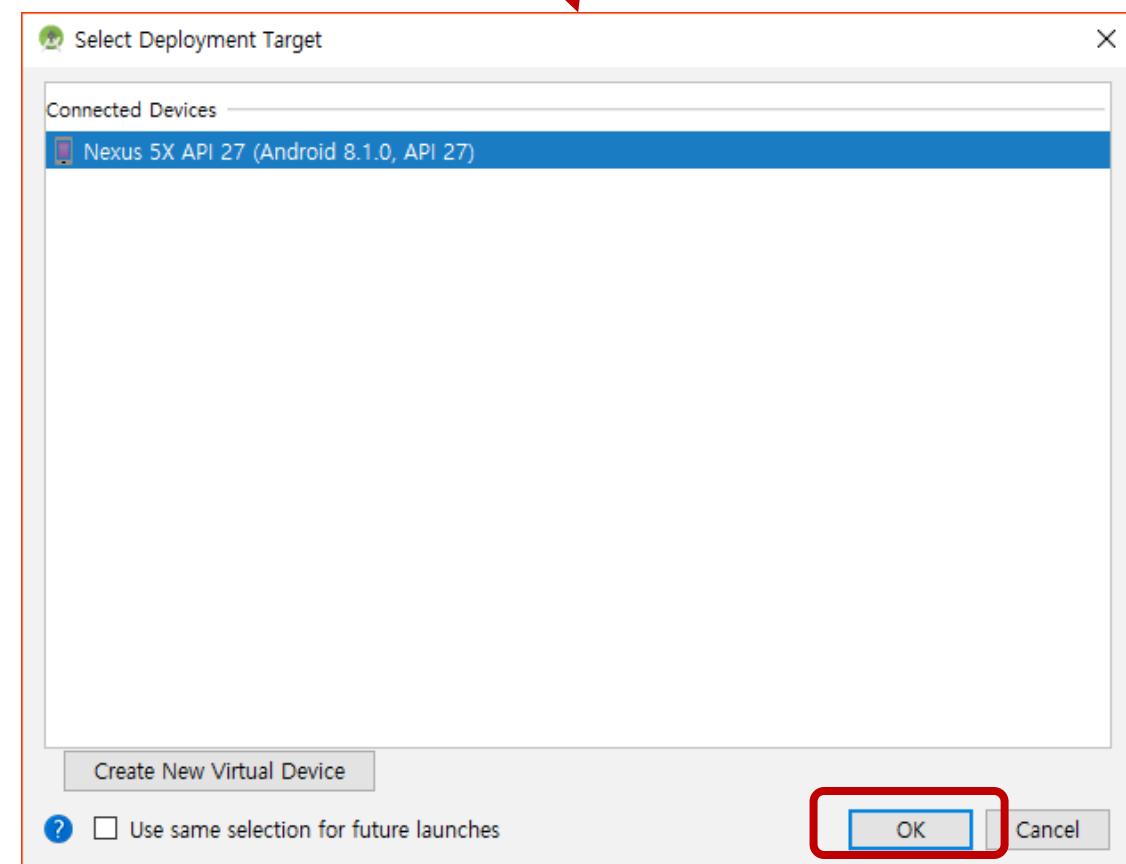


2. 프로젝트 실행

31

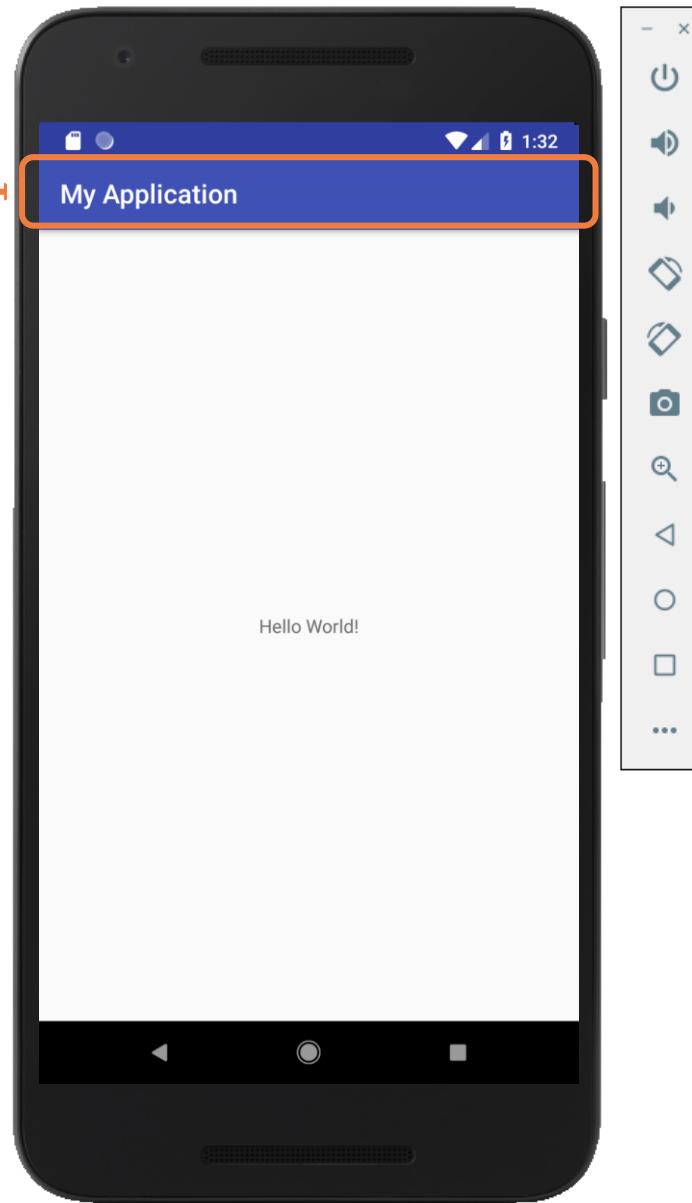


스마트폰 앱을 AVD를 통해 실행



AVD 실행 결과

32



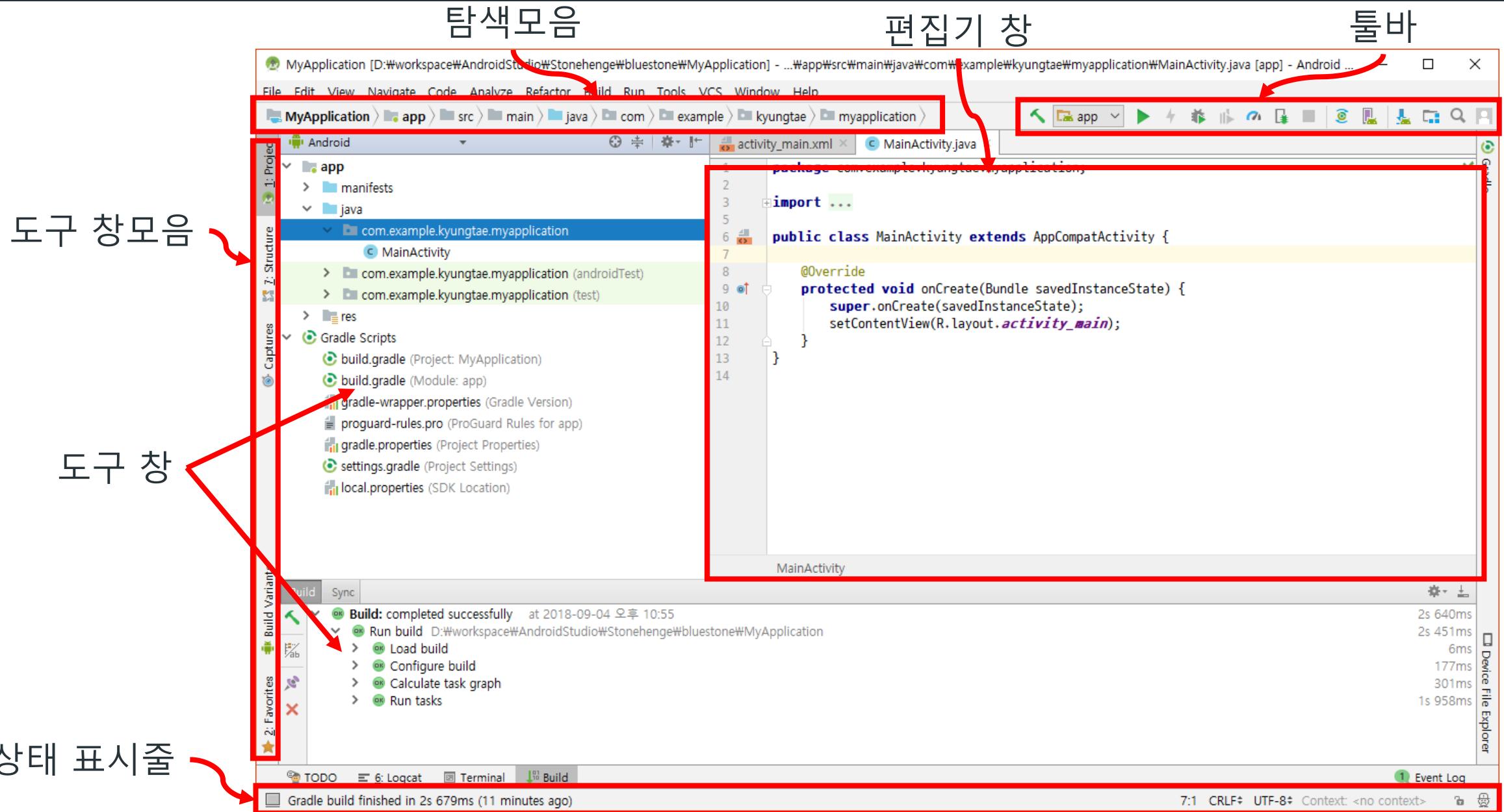
액티비티 라벨:
"My Application"

사용자 인터페이스 (User Interface)



사용자 인터페이스

38

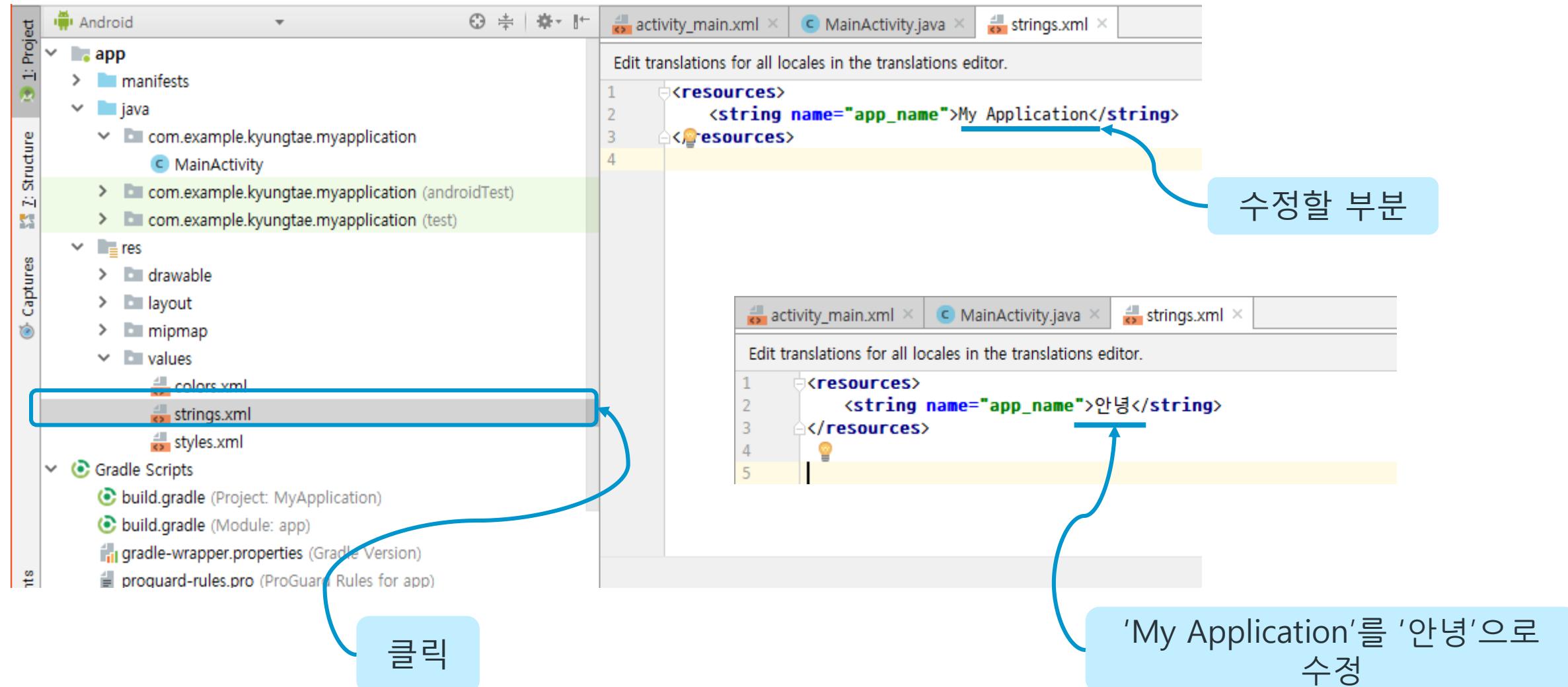


- ① **툴바**를 사용하면 앱 실행, Android 도구 시작과 같은 다양한 작업을 수행 할 수 있습니다.
- ② **탐색 모음**을 사용하면 **프로젝트를 탐색하고 편집할 파일을 열 수 있습니다.** 탐색 모음은 **프로젝트** 창에 나타나는 구조를 보다 간략하게 표시합니다.
- ③ **편집기 창**에서는 **코드를 작성하고 수정할 수 있습니다.** **현재의 파일 유형** **에 따라 편집기가 바뀔 수 있습니다.** 예를 들어, 레이아웃 파일을 볼 때 편집기는 Layout Editor를 표시합니다.
- ④ **도구 창 모음**은 **IDE 창 바깥 주변에 있으며 개별 도구 창을 펼치거나 접 을 수 있게 해주는 버튼**이 있습니다.
- ⑤ **도구 창**에서는 **프로젝트 관리, 검색, 버전 제어 등**의 특정 작업에 액세스 할 수 있습니다. 창을 확대/축소할 수 있습니다.
- ⑥ **상태 표시줄**은 **프로젝트와 IDE의 상태**를 표시하며, 또한 **경고나 메시지도 표시**합니다.

STEP 2. 파일 편집 – strings.xml

40

- 어플리케이션 라벨과 액티비티 라벨로 사용되는 **텍스트 리소스를 관리하는 strings.xml**에 있는 **app_name** 속성의 데이터를 '안녕'으로 수정한다.



Continued–activity_main.xml

41

- 액티비티 화면에 나타나는 텍스트는 `activity_main.xml`에 있는 `TextView`의 `text` 속성의 데이터를 ‘안녕, 안드로이드’로 수정.

The screenshot shows the Android Studio interface with the project structure on the left and the XML code editor on the right. The XML code defines a ConstraintLayout containing a TextView with the text "안녕, 안드로이드". The 'Text' tab in the bottom navigation bar is highlighted with a blue border, and a red arrow points from it to the 'android:text' attribute in the XML code. A blue callout bubble contains the Korean text "Text' 탭 클릭". Another blue callout bubble contains the English text "Hello World" followed by a Korean translation "→ 안녕, 안드로이드".

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
    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=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="안녕, 안드로이드"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

Step 3. 프로젝트 실행(편집 후)

43

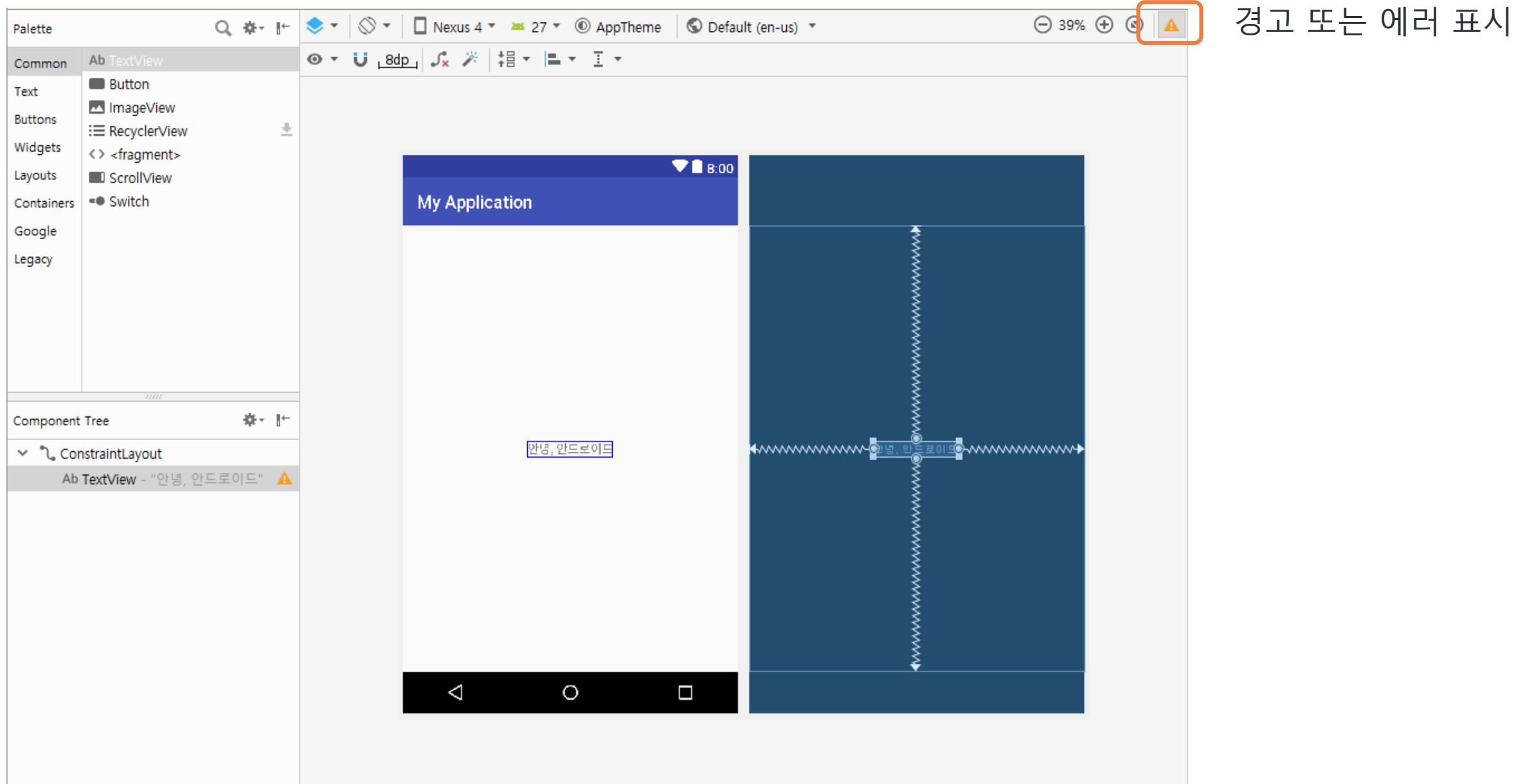


액티비티 라벨과
텍스트 내용 변경

안녕, 안드로이드

Warnings and Errors

44



Warnings and Errors…이어서

45

1 Warning

Message

Hardcoded text

Hardcoded string "안녕, 안드로이드", should use @string resource

Hardcoding text attributes directly in layout files is bad for several reasons:

- * When creating configuration variations (for example for landscape or portrait) you have to repeat the actual text (and keep it up to date when making changes)
- * The application cannot be translated to other languages by just adding new translations for existing string resources.

string “안녕, 안드로이드” 가 하드코딩되어 있다. → 좋지 않다. 소프트코딩으로 바꿔라...

The screenshot shows the Android Studio interface with three tabs at the top: activity_main.xml, strings.xml, and MainActivity.java. The activity_main.xml tab is active, displaying XML code for a ConstraintLayout. A red box highlights the android:text="@string/app_name" attribute in line 12. A red arrow points from this highlighted text to the right, where the string value "안녕, 안드로이드" is displayed. In the bottom right corner of the editor, there is a toolbar with icons for zooming and a warning sign, which is also enclosed in a red box.

```
<?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=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/app_name" <-- This line is highlighted with a red box
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

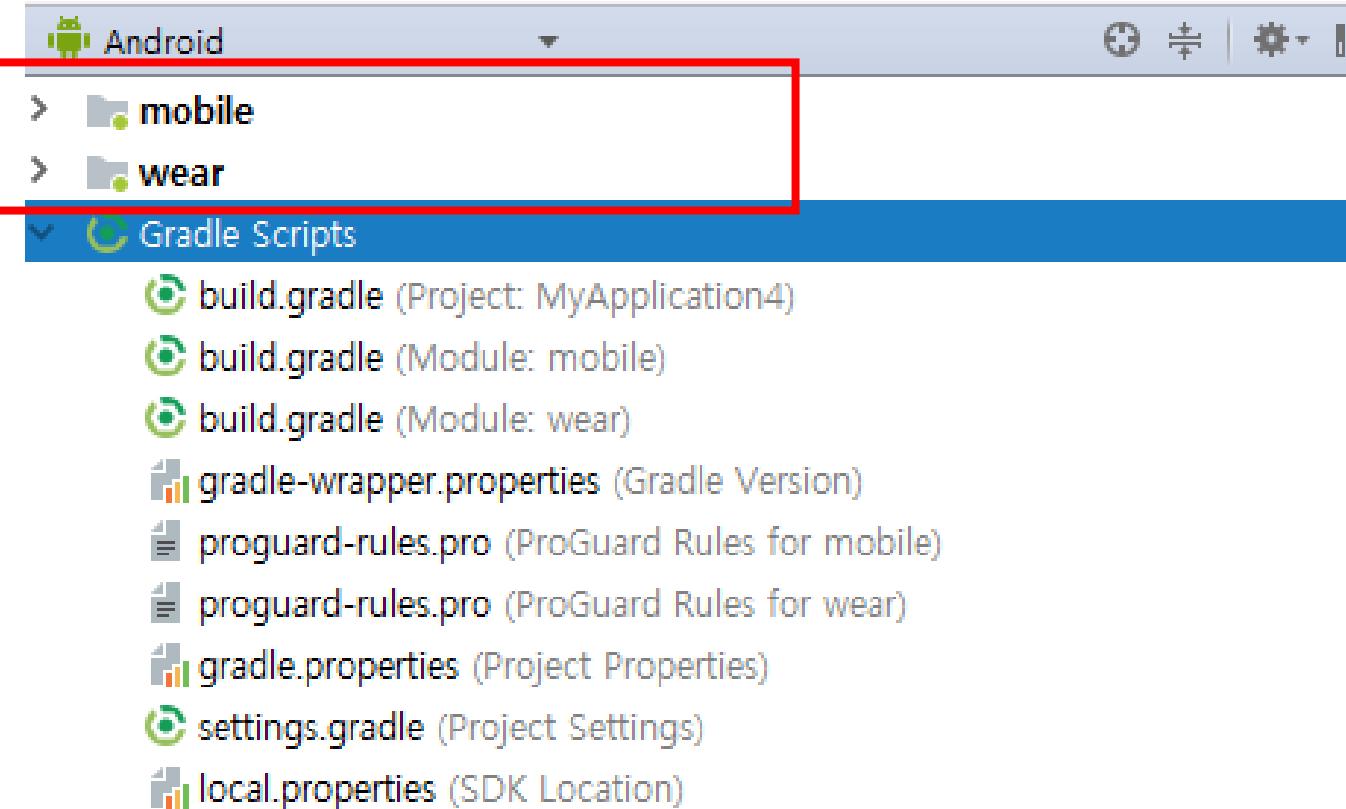
3.3 프로젝트 파일 구조

소스 코드 및 리소스(자산)에서 테스트 코드 및 빌드 구성에 이르기까지 앱에 대한 작업 영역을 정의하는 모든 항목이 포함되고 **최소 1개 이상의 모듈**을 가진다.

모듈? – 2종류의 스마트기기일 경우

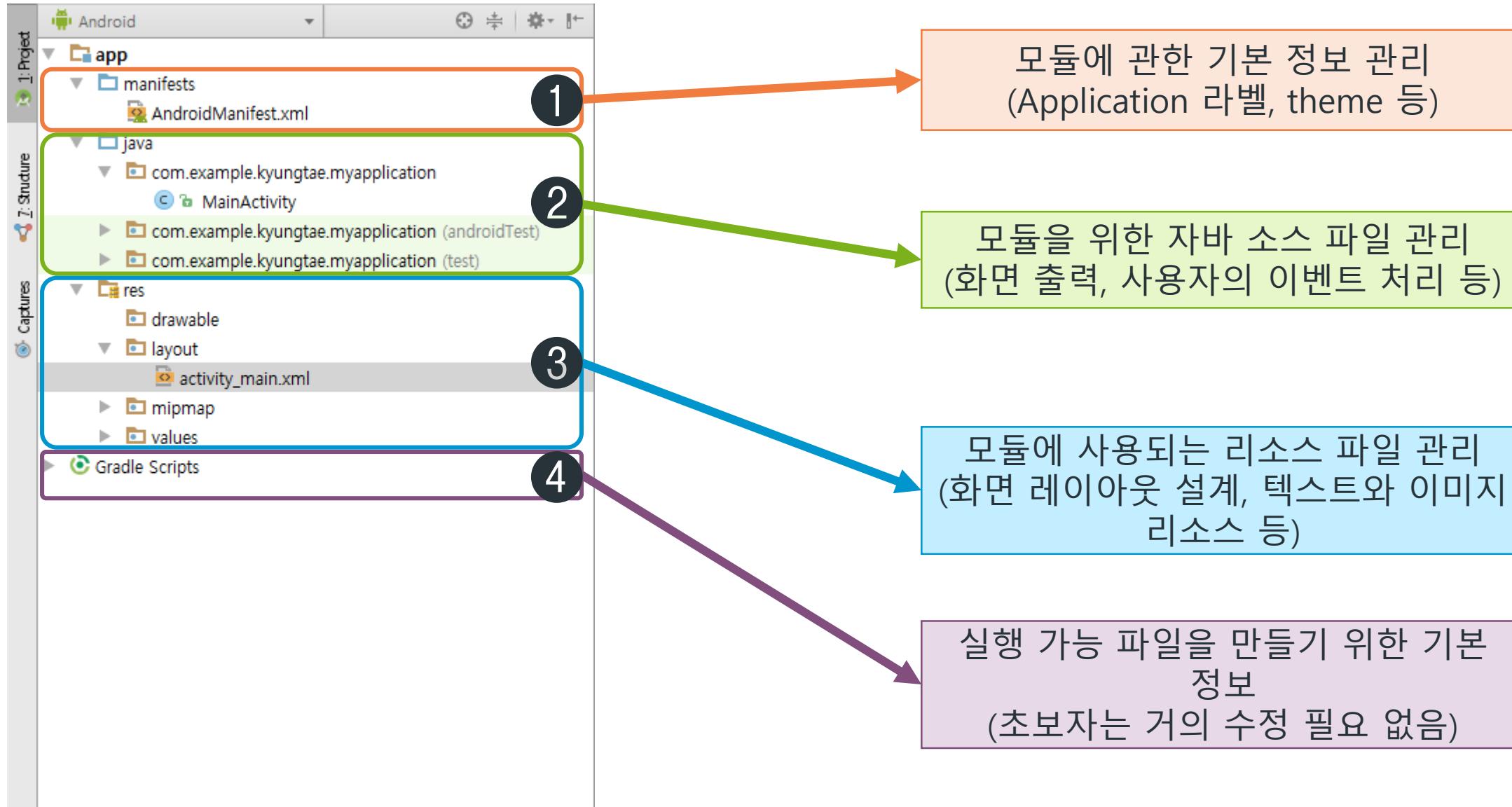
49

- 독립되어 있는 하나의 소프트웨어 또는 하드웨어 단위를 지칭
- 스마트폰과 스마트시계가 통신을 하는 앱을 만든다고 예를 들었을 때의 화면
 - mobile 모듈은 스마트폰 디바이스에 올라갈 수 있는 앱
 - wear 모듈은 스마트워치에 올라갈 앱



프로젝트 모듈의 구조와 기능

50



모듈내 각 소스의 기능

51

모듈분류	폴더(위치)	소스 파일	기능
① manifests		AndroidManifest.xml	<p>어플리케이션에 관한 정보들이 설정되어 있다. 어플리케이션 라벨, 어플리케이션 아이콘, 액티비티 라벨, 처음 실행 될 액티비티명(자바 클래스)을 포함한다.</p> <p>또한, 어플리케이션이 실행될 때 필요한 권한 등이 기술된다. (인터넷 접속 허용 설정 등)</p>
② java	com.example. kyungtae. myapplication	MainActivity.java	어플리케이션을 구성하는 액티비티를 구현하는 자바 클래스 (어플리케이션 실행 시에 처음 실행되는 자바 클래스로 사용됨)
	com.example. kyungtae. myapplication (androidTest)	ApplicationTest.java	실행 테스트를 위한 자바 클래스(삭제해도 무관함)

소스	폴더(위치)	소스 파일	기능
③ res	drawable		화면에 그려지는 그래픽을 위한 Drawable resource를 저장(png, jpg, gif 이미지 파일과 XML 파일)
	layout	activity_main.xml	액티비티 실행 시에 화면에 나타나는 레이아웃 을 설계함. strings.xml에 정의된 텍스트 리소스나 drawable 폴더에 있는 이미 리소스들을 출력할 위치에 배치함
	mipmap	ic_launcher.png(hdpi)	홈 화면의 아이콘(72 x 72 픽셀) , 240dpi 화면밀도에서 사용
		ic_launcher.png(mdpi)	홈 화면의 아이콘(48 x 48 픽셀) , 160dpi 화면밀도에서 사용
		ic_launcher.png(xhdpi)	홈 화면의 아이콘(96 x 96 픽셀) , 320dpi 화면밀도에서 사용
		ic_launcher.png(xxhdpi)	홈 화면의 아이콘(144 x 144 픽셀) , 480dpi 화면밀도에서 사용
		ic_launcher.png(xxxhdpi)	홈 화면의 아이콘(192 x 192 픽셀) , 640dpi 화면밀도에서 사용
	values	colors.xml	화면 테마에 사용되는 색상 정의
		dimens.xml	화면 구성 자원의 크기(여백, 글자 크기 등)를 정의
		strings.xml	어플리케이션 라벨과 액티비티 라벨을 포함하며, 액티비티 화면에 출력될 여러 텍스트 리소스 를 정의
		styles.xml	화면의 스타일(화면 테마, 텍스트 폰트 등)을 정의

소스	폴더(위치)	소스 파일	기능
④ Gradle Scripts		build.gradle(project:프로젝트명)	모듈 전체에 적용되는 빌드 정보
		build.gradle(Module app)	app 모듈에 적용되는 빌드 정보
		proguard-rule.pro	사용되지 않는 코드를 지워 사이즈를 줄여 최적화하거나 코드를 난독화하여 안전하게 보호하기 위해 사용
		gradle.properties	빌드 관련 속성 관리
		settings.gradle	빌드 관련 환경설정, 빌드 될 하위 모듈 정보 포함
		local.properties	빌드 진행 시, 필요한 환경변수 정보를 저장하는 파일임. 안드로이드 SDK의 경로가 이곳에 저장됨.

3.4 앱의 실행 원리

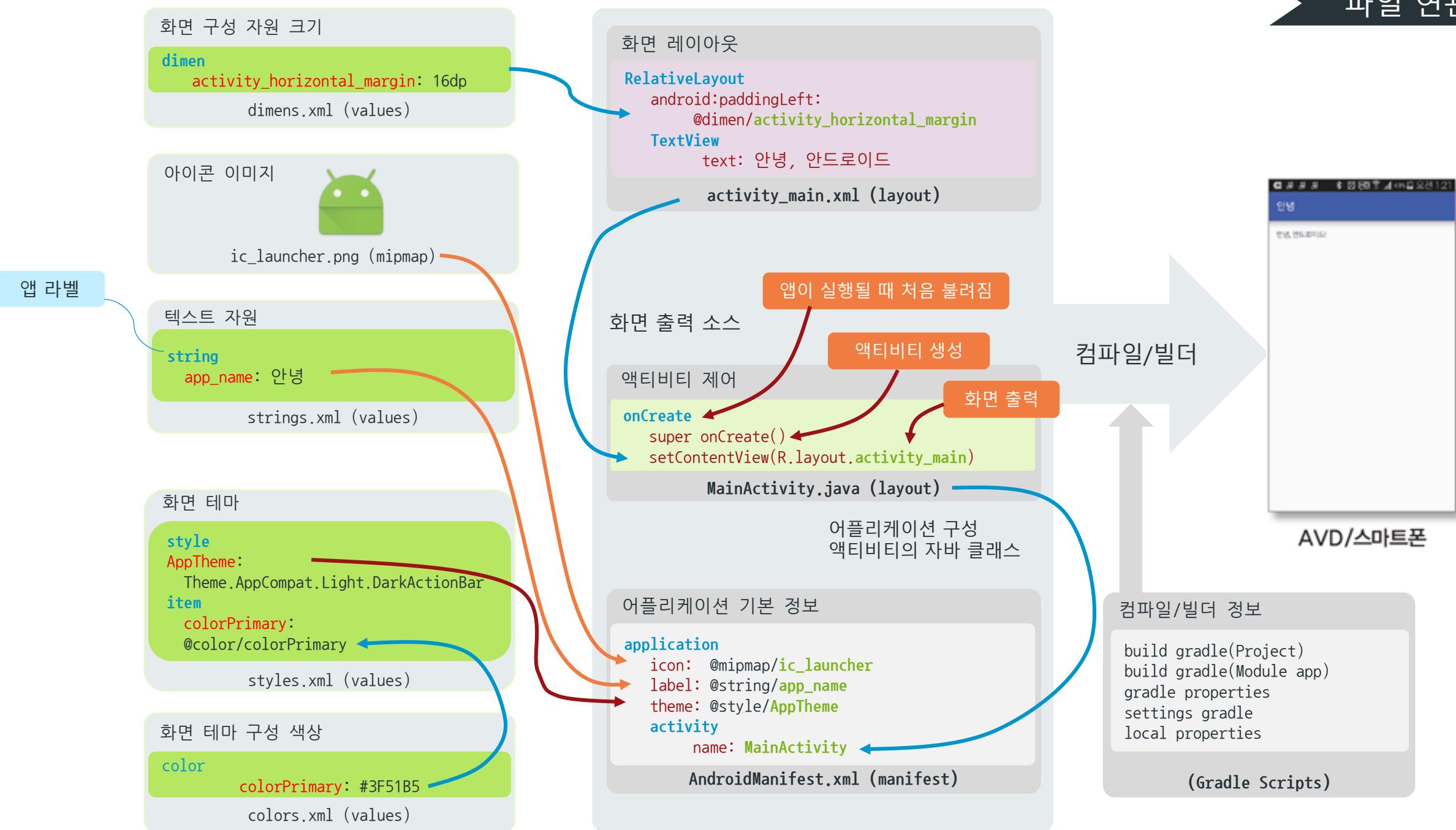
앱의 실행 원리

55

- 자바와 xml 파일로 구성되는 기본 파일들은 서로 연관성을 가지고 있음



파일	내용
activity_main.xml	화면을 설계하며, dimens.xml 파일에 정의된 여백의 크기를 설정
MainActivity.java	액티비티를 생성하고 activity_main.xml에 정의된 레이아웃을 출력
AndroidManifest.xml	어플리케이션 아이콘과 라벨, 화면 스타일 등을 지정하고, 액티비티를 구성하는 자바 클래스를 지정한다.
Gradle Scripts	컴파일/빌더 정보를 담고 있으며 실행파일을 만드는 역할



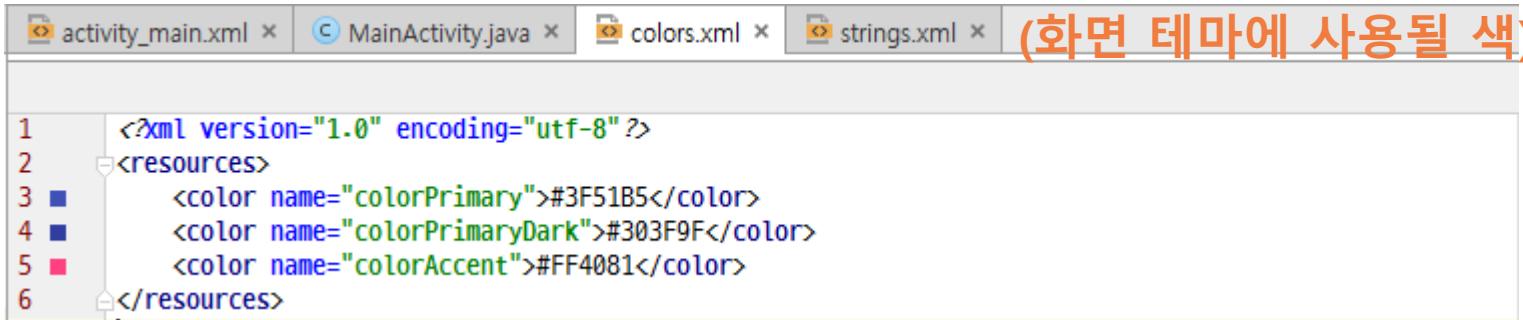
3.5 프로젝트 소스 간의 연관성

- 프로젝트 모듈을 구성하는 주요 파일의 소소내의 연관 부분 간 흐름을 살펴 보자. 구체적인 문법은 다음 4장부터 설명한다.

- **res/mipmap 폴더**



- res/values 폴더의 colors.xml



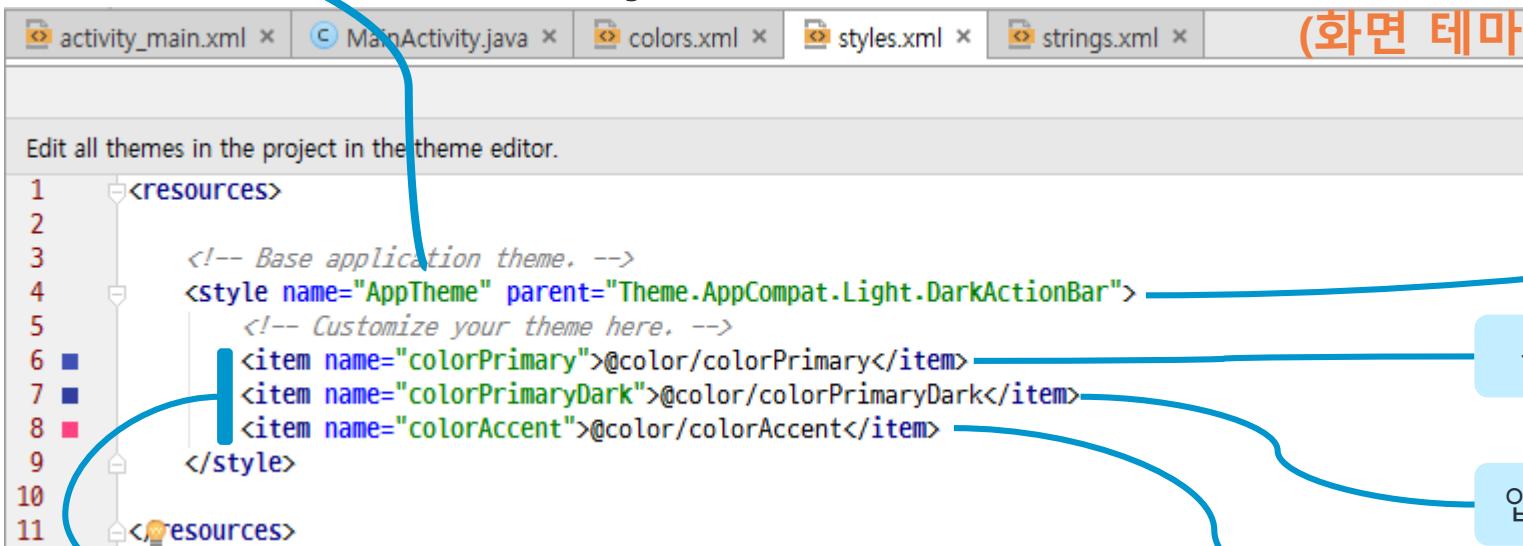
```

1 <?xml version="1.0" encoding="utf-8"?
2 <resources>
3   <color name="colorPrimary">#3F51B5</color>
4   <color name="colorPrimaryDark">#303F9F</color>
5   <color name="colorAccent">#FF4081</color>
6 </resources>

```

②

- res/values 폴더의 styles.xml



```

1 <resources>
2
3   <!-- Base application theme. -->
4   <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
5     <!-- Customize your theme here. -->
6     <item name="colorPrimary">@color/colorPrimary</item>
7     <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
8     <item name="colorAccent">@color/colorAccent</item>
9   </style>
10
11 </resources>

```

AppTheme 속성값을 '어두운 색의
액션 바를 가진 밝은 테마'로 할당

상태 바(status bar)의 색

앱 바(application bar)의 색

체크박스, 텍스트 입력 필
드와 같은 UI 컨트롤의 색

화면 테마의 추가적인 정보

- res/values 폴더의 dimens.xml과 strings.xml

```
1 <resources>
2     <!-- Default screen margins, per the Android Design guidelines. --&gt;
3     &lt;dimen name="activity_horizontal_margin"&gt;16dp&lt;/dimen&gt;
4     &lt;dimen name="activity_vertical_margin"&gt;16dp&lt;/dimen&gt;
5 &lt;/resources&gt;</pre>
```

```
1 <resources>
2     <string name="app_name">안녕</string>
3 </resources>
```

app_name을 '안녕'으로 수정

- res/layout 폴더의 activity_main.xml

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <android.support.constraint.ConstraintLayout 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="com.example.kyungtae.myapplication.MainActivity">
8
9     <TextView
10        android:layout_width="wrap_content"
11        android:layout_height="wrap_content"
12        android:text="안녕, 안드로이드!"
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>
```

화면의 하단,좌측,우측,상단 패딩의
크기

• java/com.example.Kyungtae.myapplication 패키지

5

```

activity_main.xml x MainActivity.java x dimens.xml x strings.xml x (액티비티 제어)
1 package com.example.kyungtae.myapplication;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.os.Bundle;
5
6 public class MainActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13 }

```

MainActivity 자바 클래스가 호출될 때 처음 실행되는 메소드

액티비티 생성

activity_main.xml에서 정의된 화면 레이아웃을 액티비티에 출력

4

• Manifests 폴더

(어플리케이션 기본 정보)

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3   package="com.example.kyungtae.myapplication">
4
5   <application
6     android:allowBackup="true"
7     android:icon="@mipmap/ic_launcher"
8     android:label="안녕"
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    </application>
20
21 </manifest>
22

```

① 어플리케이션 아이콘
 ② 화면 테마
 ③ 어플리케이션 라벨
 ⑤ 어플리케이션을 구성하는 액티비티 자바 클래스로 앱 실행시 처음 실행됨



question

&



answer

