

# 안드로이드 튜토리얼

## 제 1장 안드로이드 스튜디오



동아리 ALOM

01/ 프로젝트  
생성

02/ 프로젝트  
파일 구성

03/ IDE 기능  
알아보기

04/ AVD  
실행하기

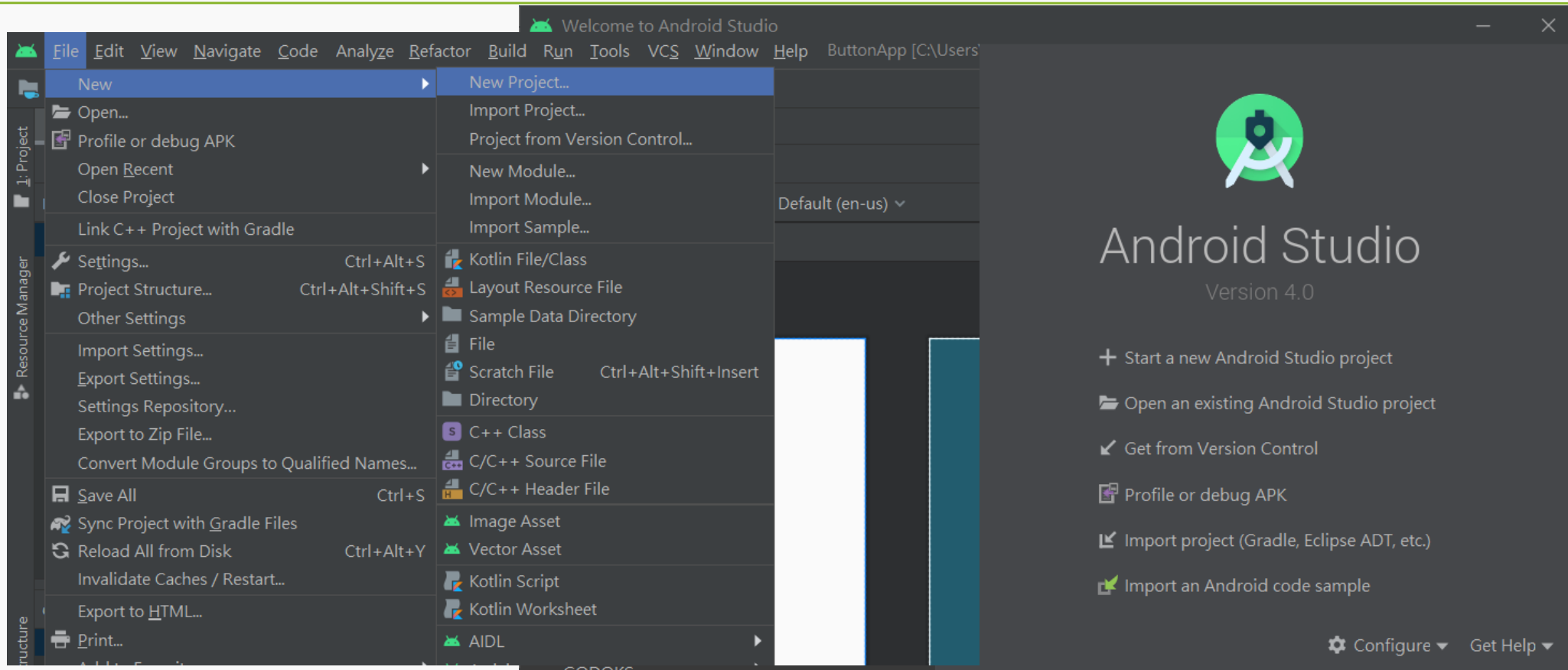
05/ 혼자  
해보기





# 1. 프로젝트 생성

TITLE | TITLE | TITLE | TITLE | TITLE



기존 프로젝트 실행 중  
새 프로젝트 생성시

File->New->New Project 선택

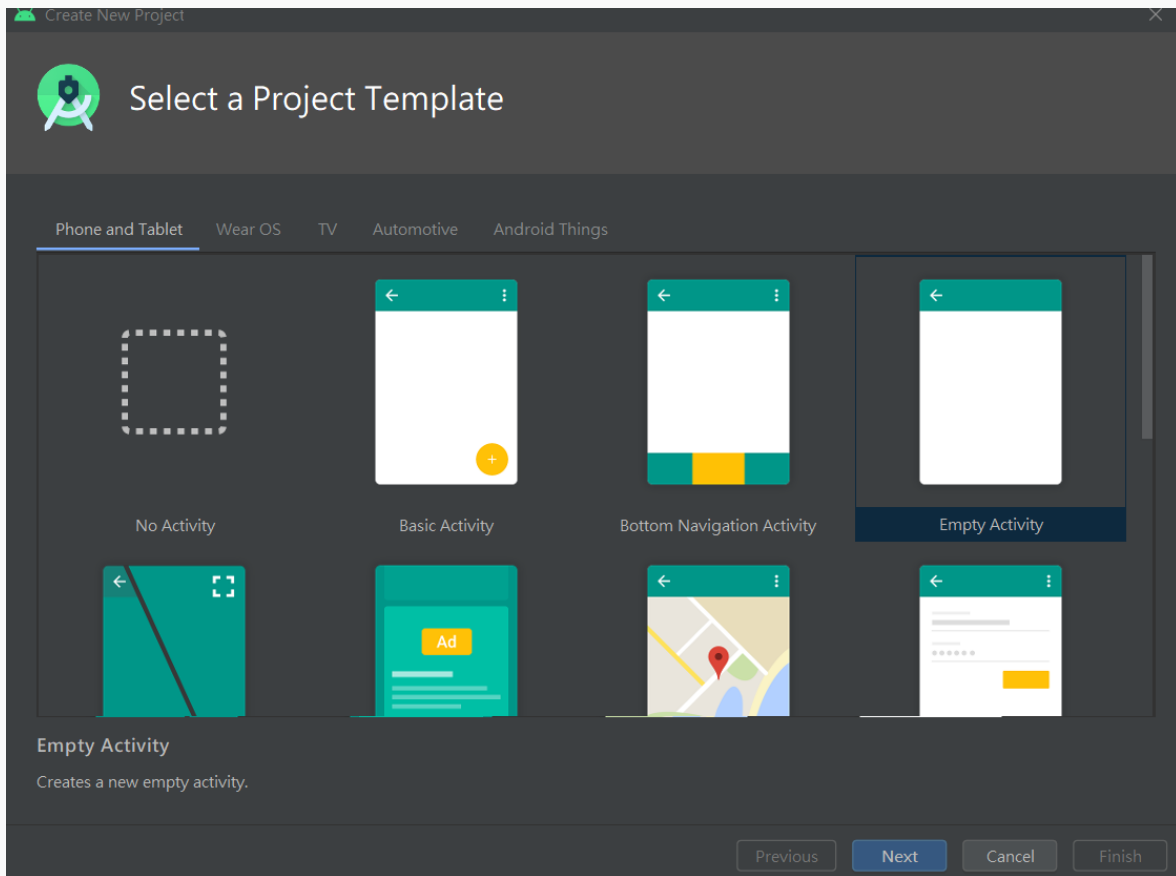
맨 위줄 프로젝트 시작 선택





# 1. 프로젝트 생성

TITLE | TITLE | TITLE | TITLE | TITLE



Empty Activity 선택





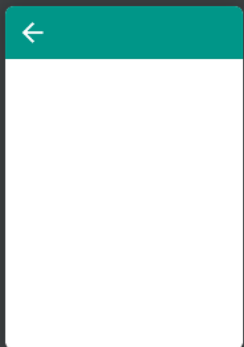
# 1. 프로젝트 생성

TITLE | TITLE | TITLE | TITLE | TITLE



Create New Project

## Configure Your Project

 Empty Activity  
Creates a new empty activity.


Name  
My Application


Package name  
com.example.myapplication

Save location  
C:\Users\lys74\AndroidStudioProjects\MyApplication

Language  
Java

Minimum SDK  
API 21: Android 5.0 (Lollipop)

 Your app will run on approximately 94.1% of devices.  
[Help me choose](#)

☐ Use legacy android.support libraries 

Previous Next Cancel Finish

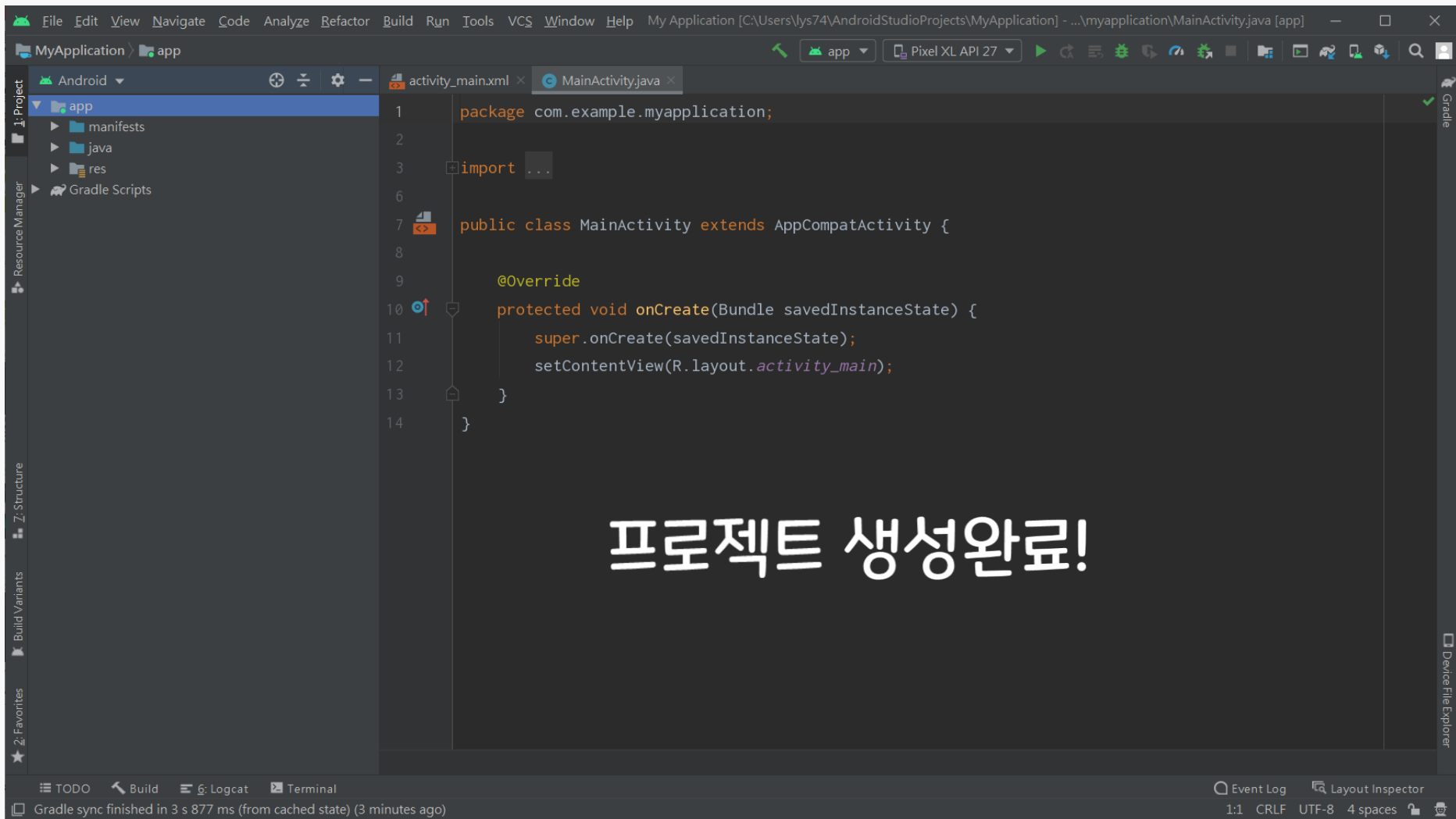
Name : 마음대로  
Package name: Name에  
따라 자동으로 생성됨  
Save location: 저장위치  
Language: JAVA  
Minimum SDK: 21  
(일단은 몰라도 된다, 넘어가자)





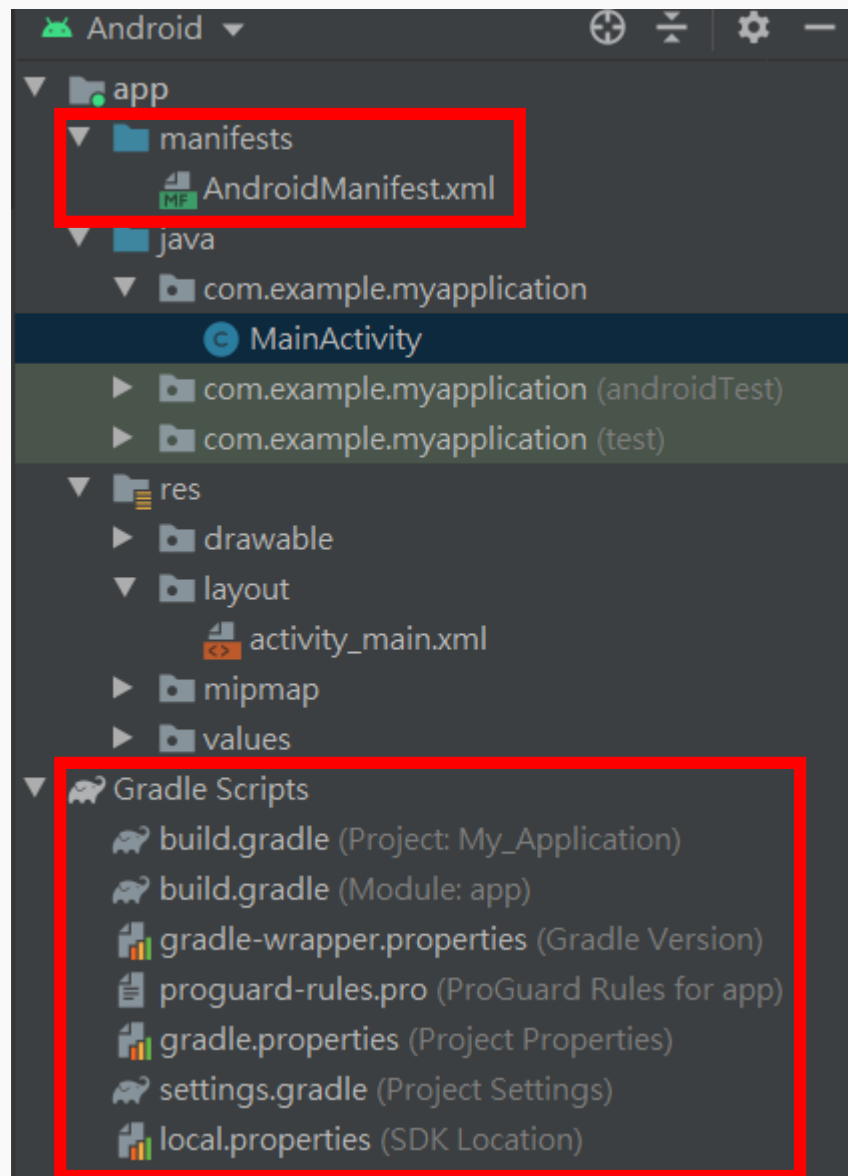
# 1. 프로젝트 생성

TITLE | TITLE | TITLE | TITLE | TITLE





## 2. 프로젝트 파일구성



AndroidManifest는 안드로이드의 컨트롤 타워이다. 안드로이드 어플리케이션을 구동 하는데 필요한 설정 값을 관리해주는 곳이다.

Gradle Scripts는 어플리케이션을 빌드 하기위해 필요한 옵션, 라이브러리 정보들이 들어있다.

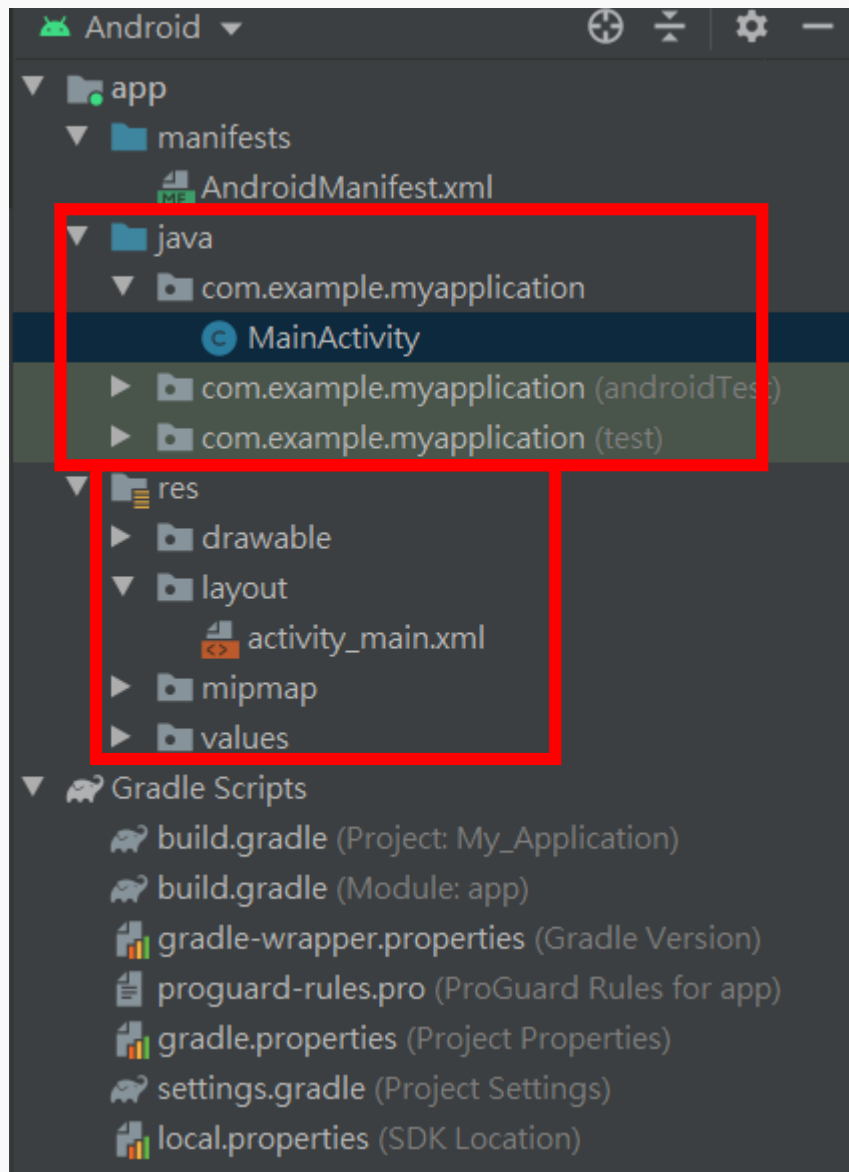
둘 다 지금 당장은 자세히 알 필요 없다. 이런 것들이 있구나 하고 넘어가면 된다.





## 2. 프로젝트 파일구성

TITLE | TITLE | TITLE | TITLE | TITLE



res : Resource 폴더로 UI와 관련된 파일과 디자인 리소스, 문자열 리소스를 담고있다.

앱에서 겉으로 보이는 모든 것들을 담당

Ex) UI, 아이콘, 폰트, 사진파일

java : 클래스를 관리하는 폴더이다.

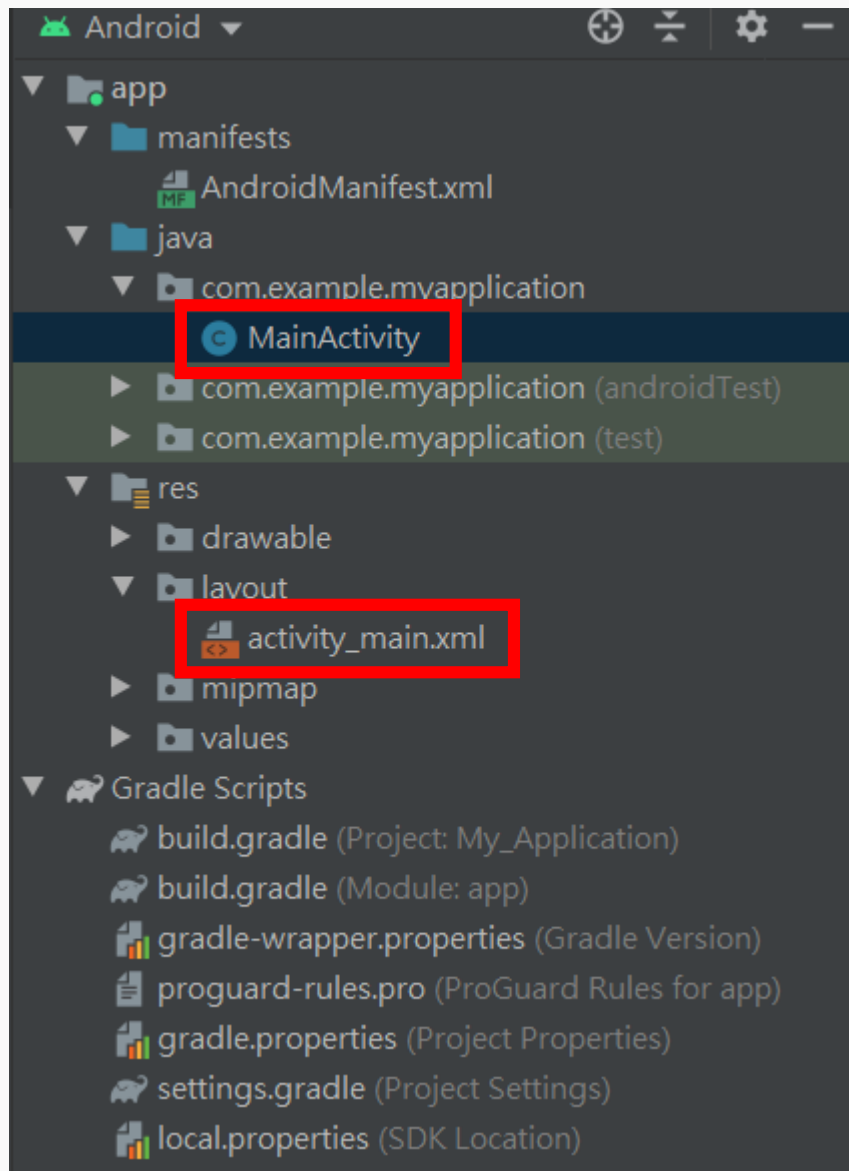
앱이 작동하는 방식에 대해 정의하는 .java 파일들이 모두 이곳에 들어있다.

앱에서 보이지 않는 모든 기능들을 담당

Ex) 버튼 동작, 파일 전송 기능, 사진 촬영 기능







처음 프로젝트를 생성했을 때  
기본적으로 MainActivity.java와  
Activity\_main.xml 두 파일이 있다.

Activity\_main.xml은  
어플리케이션 위에 실행될 화면을 보여준다.

MainActivity.java는  
Activity를 어플리케이션에 띄워준다.

```
package com.example.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

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

MainActivity.java 코드

코드에서 표시된 줄을 보면  
setContentView라는 함수가 보인다.  
이 함수는 표시될 화면을 지정한다.

R.layout.activity\_main은  
activity\_main.xml을 가리킨다.  
즉, xml 파일에서 화면을 구성하고,  
java코드에서 화면을 띄우고,  
여러 기능을 수행한다.

```
package com.example.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

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

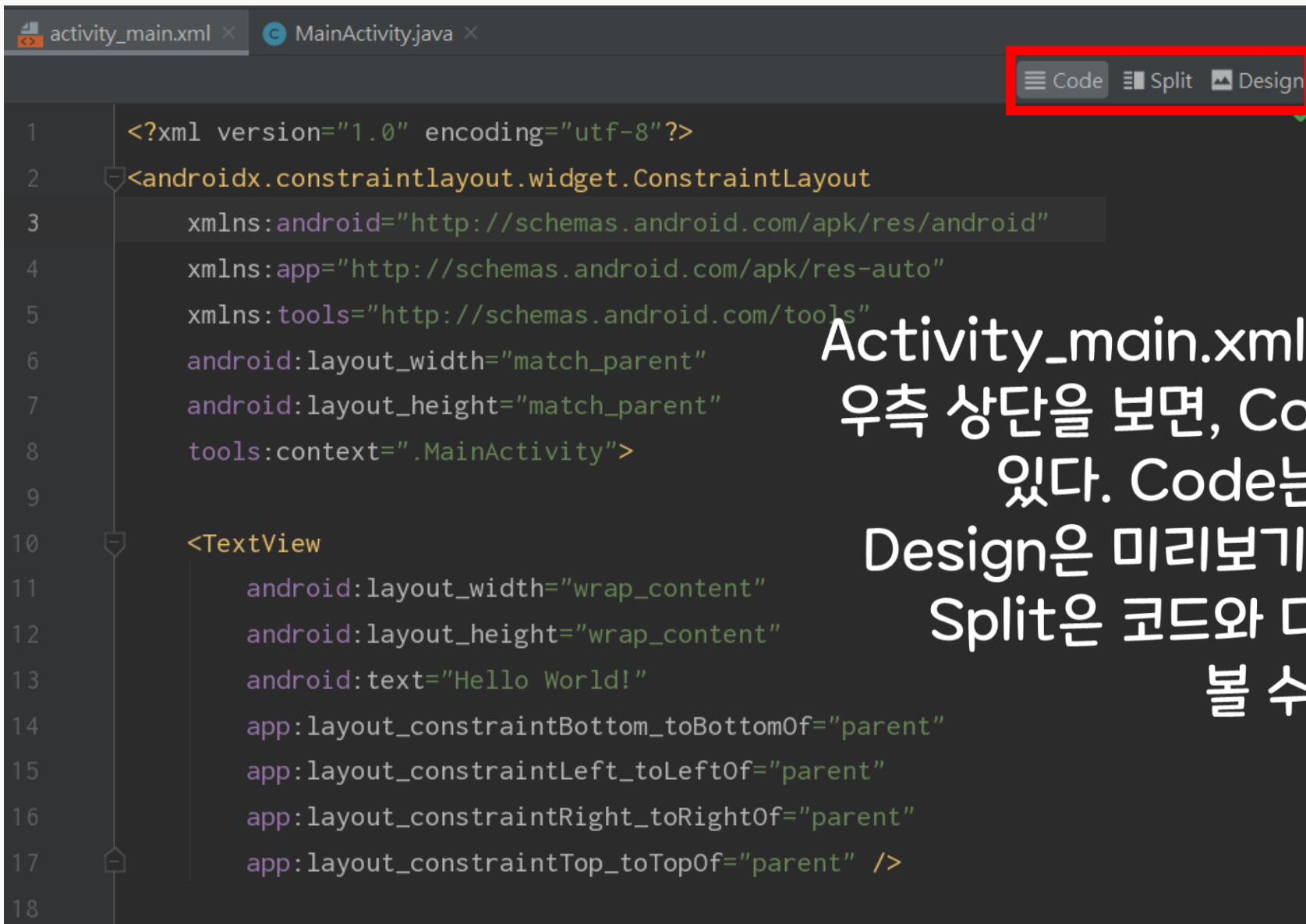
MainActivity.java 코드

만약 앱에서 메인화면과 설정화면을 만들었다고 생각해보자.  
설정화면은 activity\_setting.xml 이라고 할 때,  
메인화면에서 설정화면으로 이동하려면  
setContentView  
(R.layout.activity\_setting);  
으로 이동할 수 있다.



## 3. IDE 기능 알아보기

TITLE | TITLE | TITLE | TITLE | TITLE



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

Activity\_main.xml 에서 Code 화면이다.  
우측 상단을 보면, Code, Split, Design이  
있다. Code는 텍스트 코드를,  
Design은 미리보기 화면과 디자인 툴을,  
Split은 코드와 디자인 탭을 한번에  
볼 수 있다.



# 3. IDE 기능 알아보기

TITLE | TITLE | TITLE | TITLE | TITLE



Android Studio IDE interface showing the XML editor and Design view. The 'Split' button in the top right is highlighted with a red box.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Split 탭

Design 탭  
Palette, Attributes,  
Component Tree  
등으로 이루어져 있으며  
처음에는 거의 디자인 툴로  
화면과 구성요소들을 만든다.

디자인 탭에서는  
직접 마우스로 드래그 하여  
버튼 등의 위젯들을 집어넣고  
화면을 구성한다.  
(마치 파워포인트와 같다)

androidx.constraintlayout.widget.ConstraintLayout

위 내용을 담기에는 너무 방대하고  
다른 정리된 좋은 자료가 많으므로

구글에 “안드로이드 스튜디오 avd 실행” 검색  
구글 공식 문서 또는 블로그 포스팅 참조

AVD 는 안드로이드 가상 디바이스를 뜻하며,  
안드로이드 기종의 휴대폰을 사용한다면 USB로 직접 연결하여  
가상기기 대신 자신의 휴대폰에 앱을 빌드해볼 수 있다.





## DO IT YOURSELF!!!

1. 안드로이드 프로젝트 생성
2. 프로젝트 폴더 확인
3. AVD 실행 후 앱 빌드



android