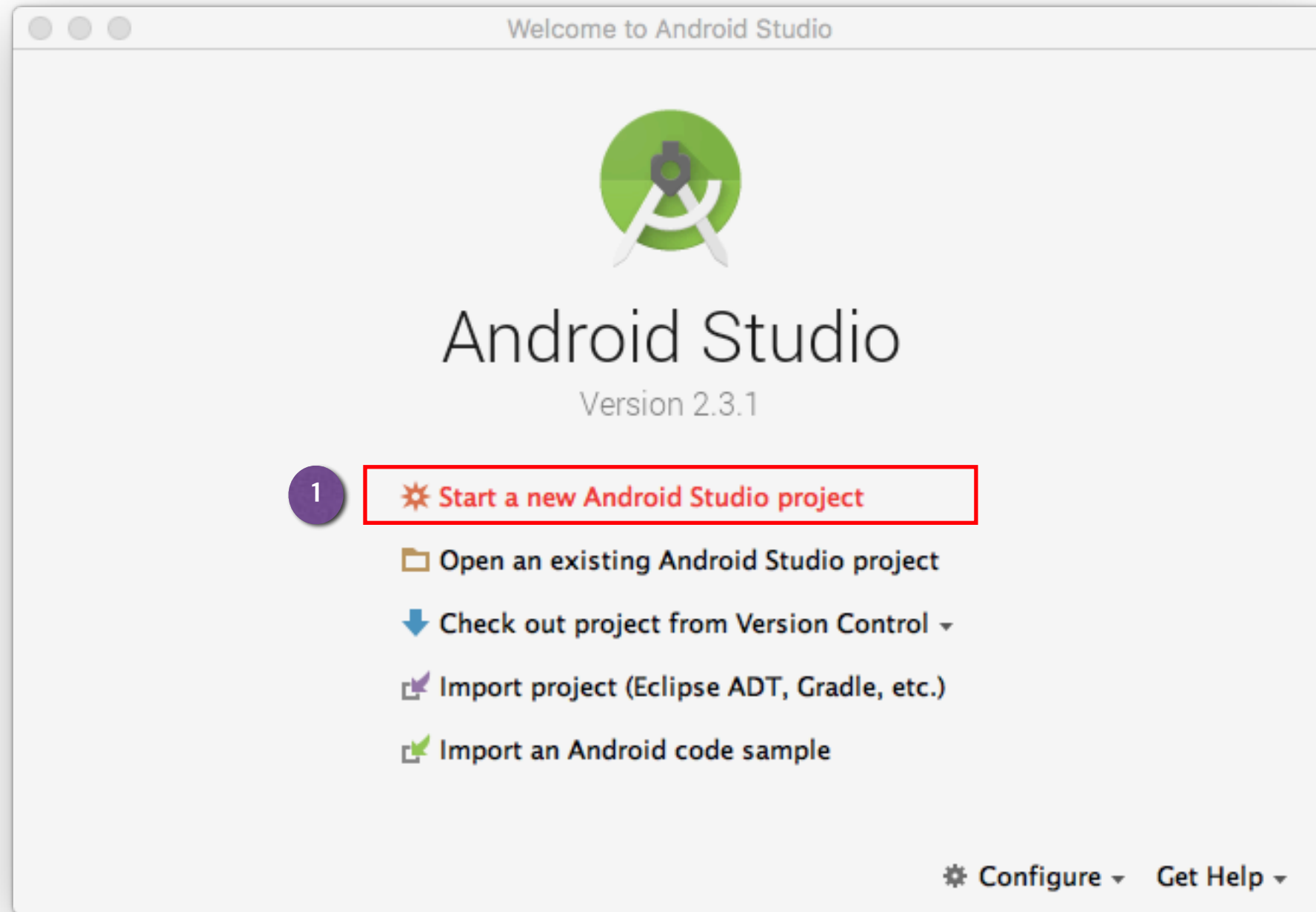


# Android 1주차


# 프로젝트 생성



- 새 프로젝트 만들기

# 프로젝트 생성

Create New Project

 **New Project**  
Android Studio

**Configure your new project**

Application name:

Company domain:

Package name:  [Edit](#)

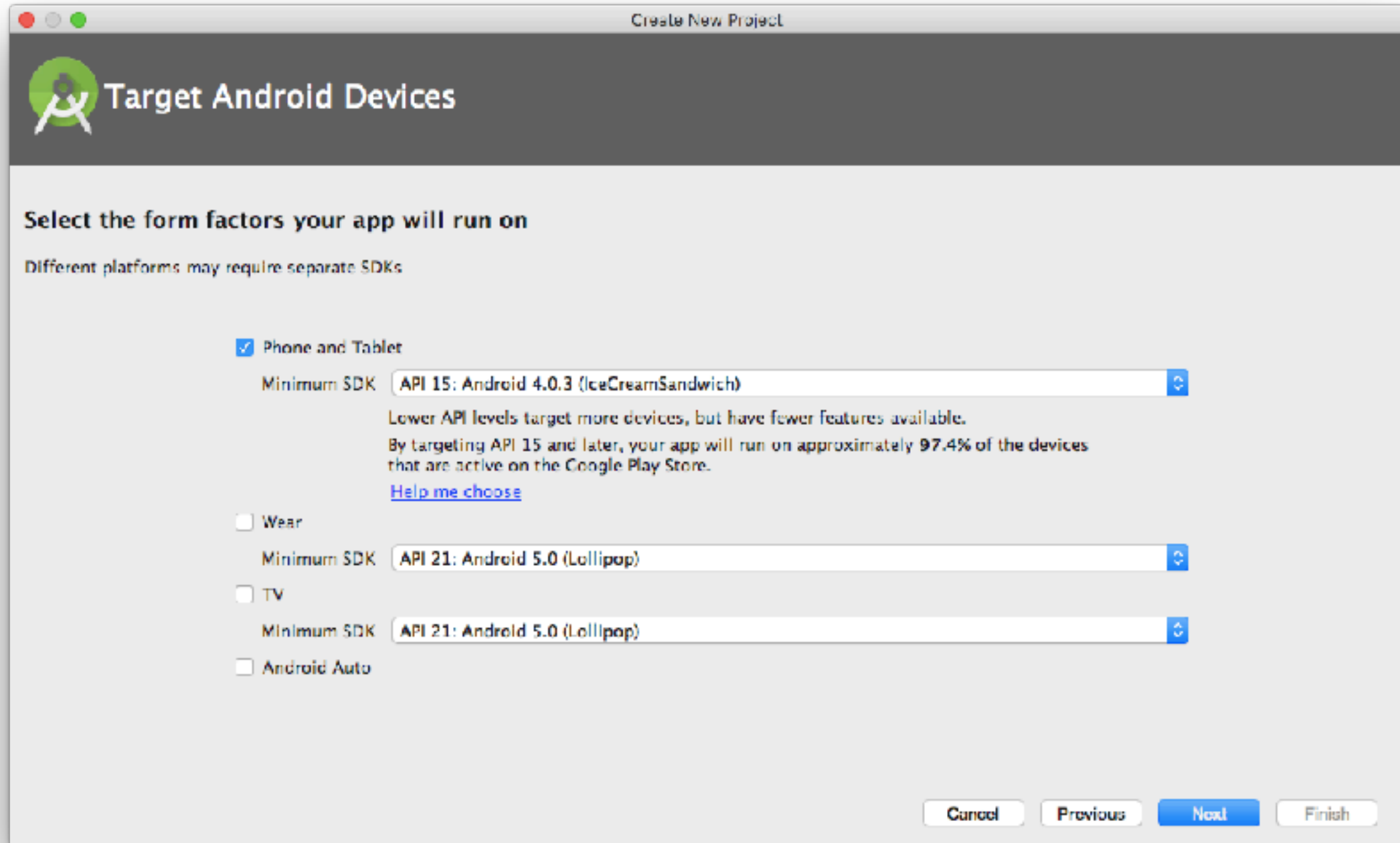
☐ Include C++ support

Project location:  [...](#)

[Cancel](#) [Previous](#) [Next](#) [Finish](#)

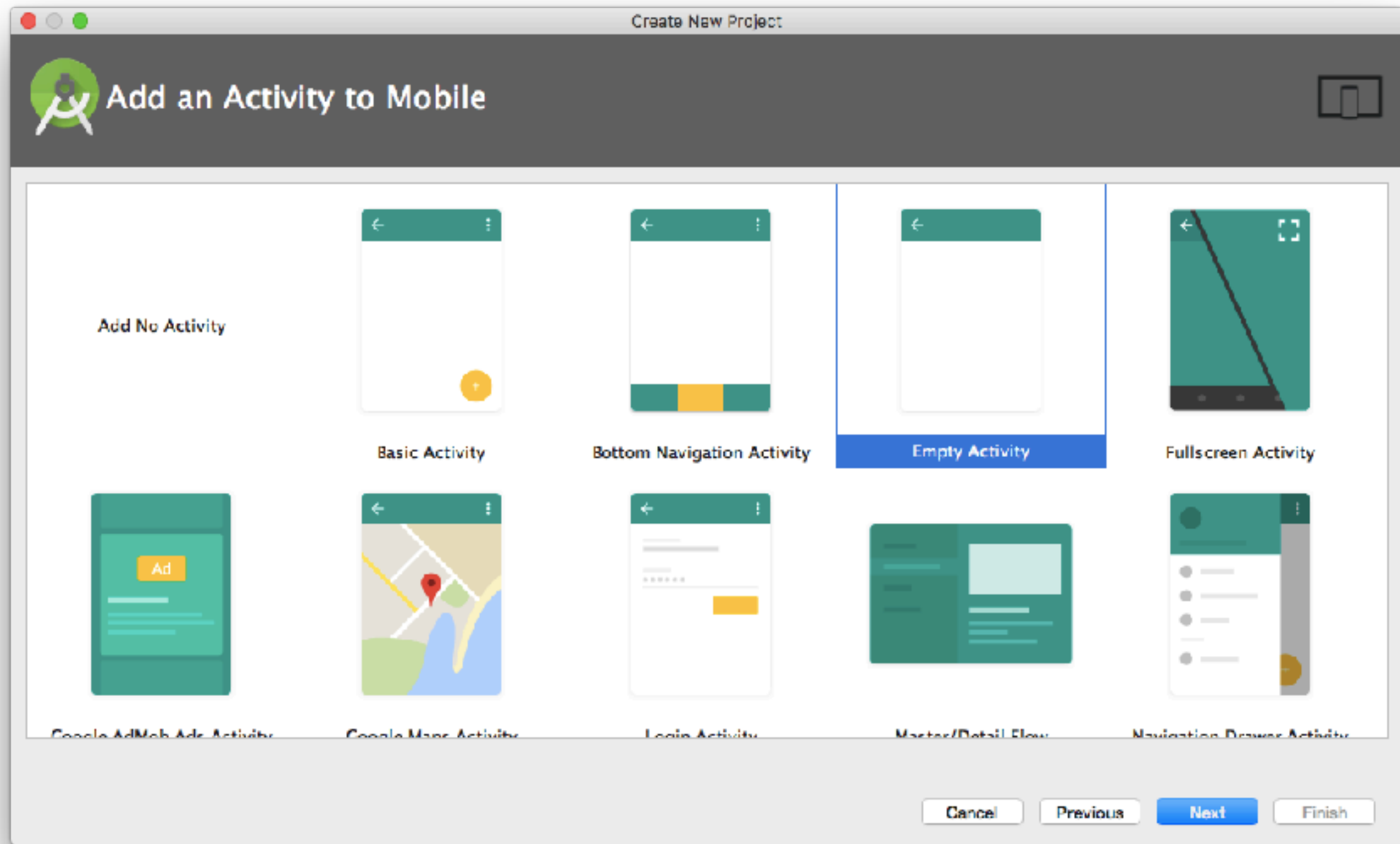
- 어플리케이션 이름
- 도메인(다른 프로젝트와 겹치지 않게)

# 프로젝트 생성



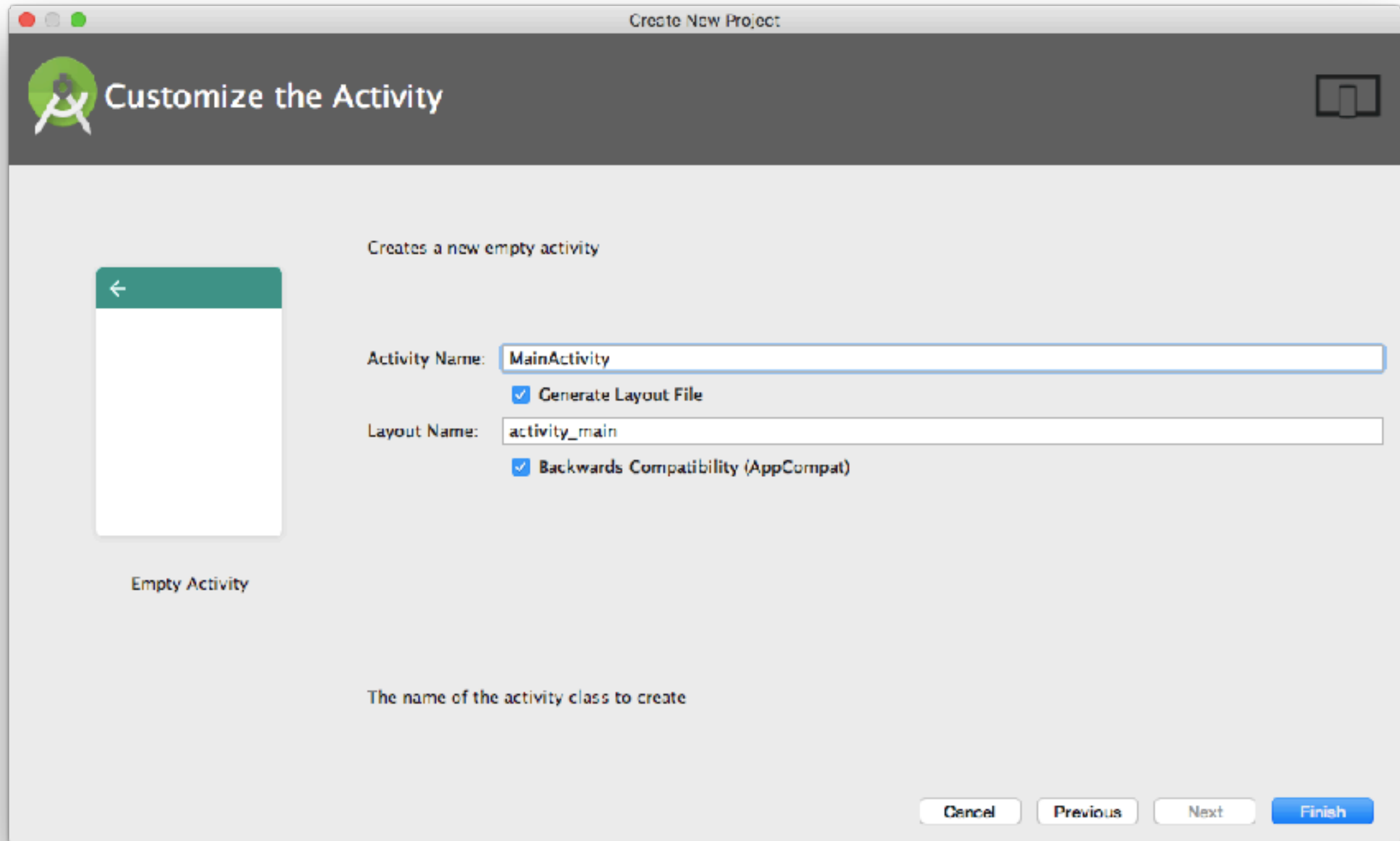
- 지원할 매체, 최소 지원 SDK(Software development kit) 설정

# 프로젝트 생성



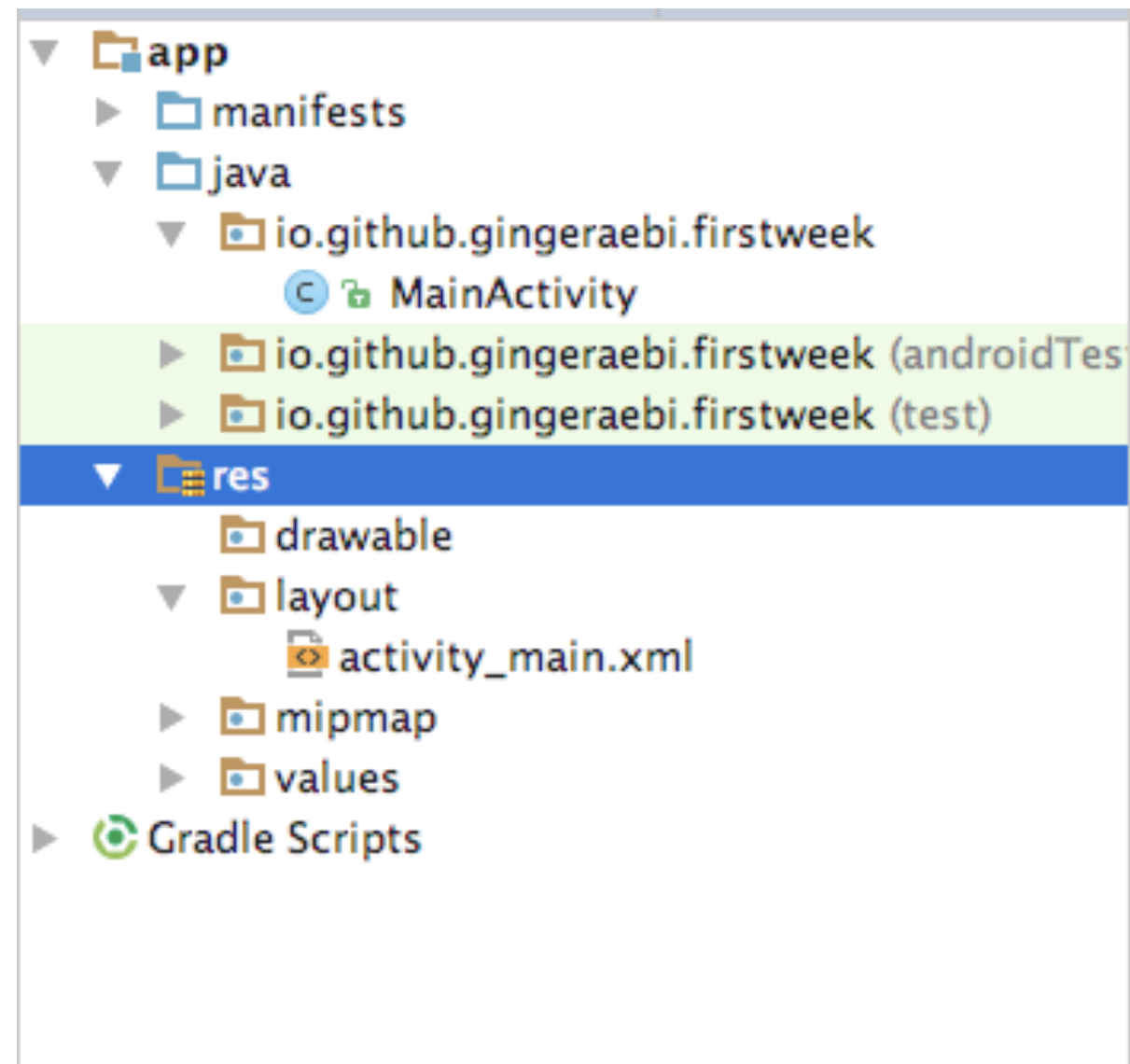
- 처음 만들 화면에 대한 템플릿 제공 (고르기)

# 프로젝트 생성



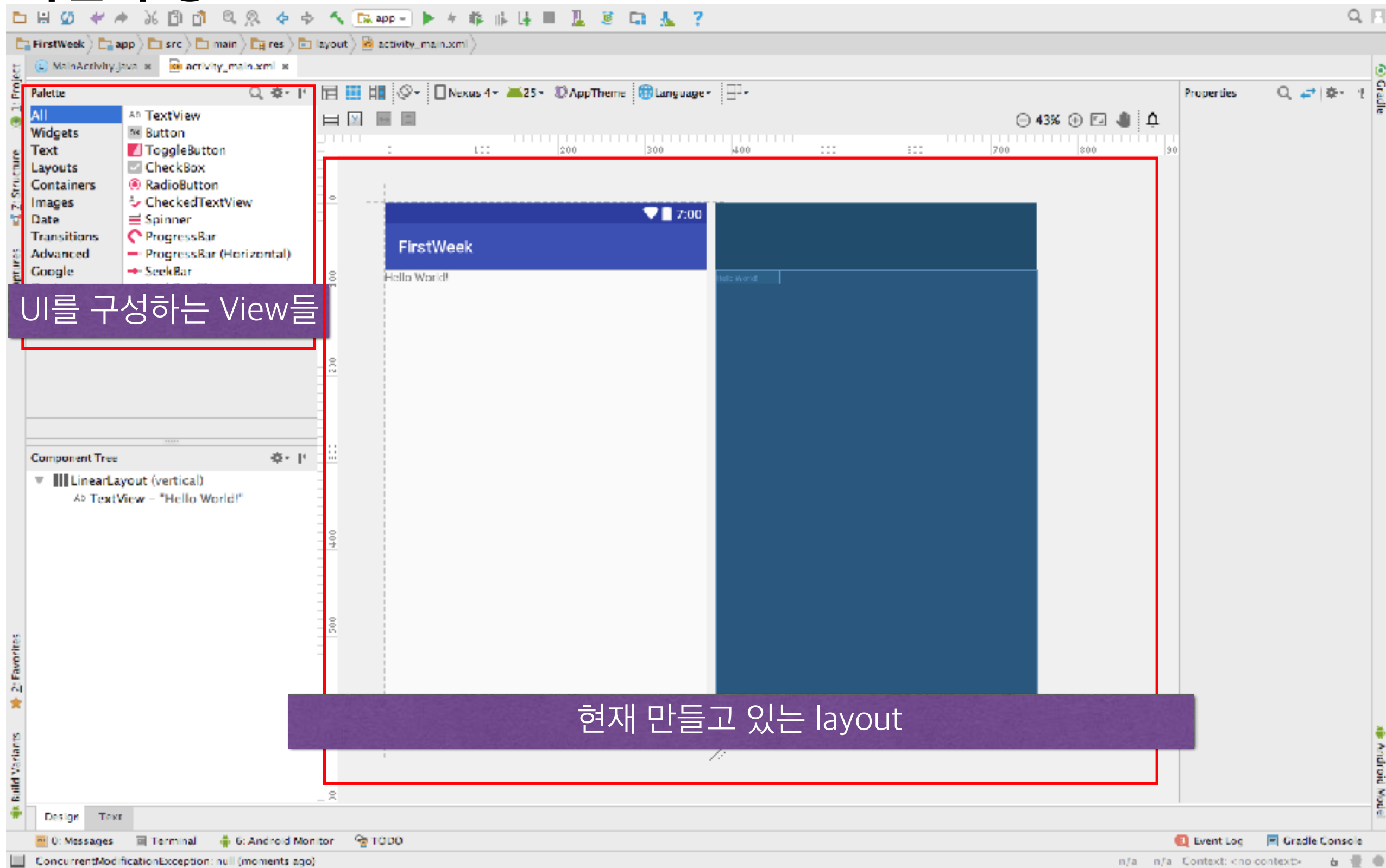
- 화면이 처음 시작될 지점에 대한 액티비티명 설정

# 프로젝트 구조



- 프로젝트 구조
- app > java (자바 파일들)
- app > res ( 리소스 파일들 )

# 화면 구성





# LinearLayout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context="io.github.gingeraebi.firstweek.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!">

</TextView>
</LinearLayout>
```

- XML 형식으로 전 슬라이드의 화면을 나타냄
- tools:context="~~~~" -> xml과 연결되어 있는 액티비티를 나타냄

# XML 형식?

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context="io.github.gingeraebi.firstweek.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!">

    </TextView>
</LinearLayout>
```

- <> 와 </>가 한쌍을 이룸
- 처음 <와 붙어있는 텍스트가 그 필드의 이름
- <와 > 사이에 키=값 형식으로 값을 지정 가능

# LinearLayout

```
package io.github.gingeraebi.firstweek;

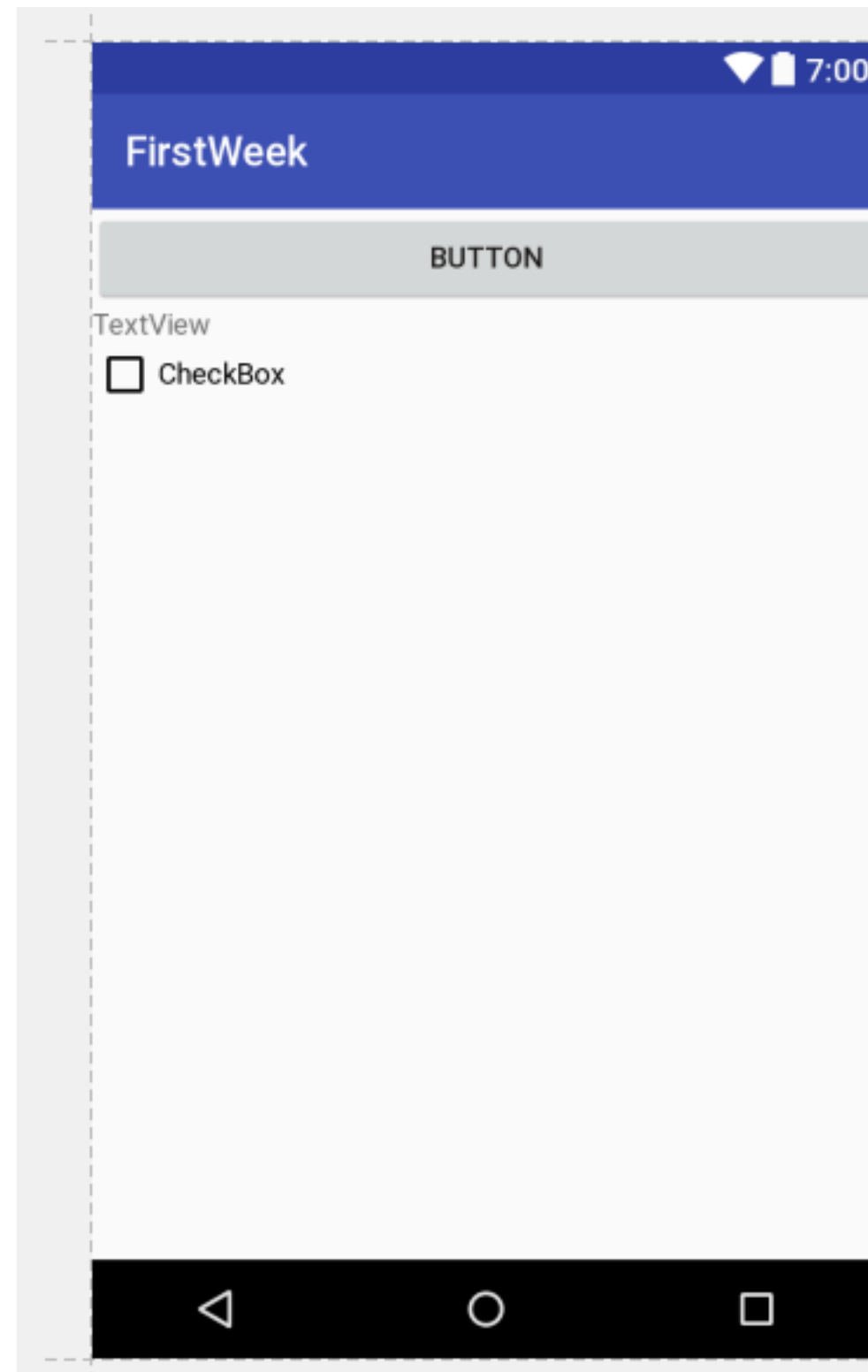
import ...

public class MainActivity extends AppCompatActivity {

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

- AppCompatActivity를 상속
- onCreate() -> 화면이 처음 만들어질 때 하는 동작들을 정의해놓은 메서드

# LinearLayout



- 아무거나 Palette에서 drag & drop해서 가져다놔 보자.
- 화면 요소들이 블록처럼 아래로 쌓이는 것을 볼 수 있다.

# LinearLayout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context="io.github.gingeraebi.firstweek.MainActivity">

    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Button" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="TextView" />

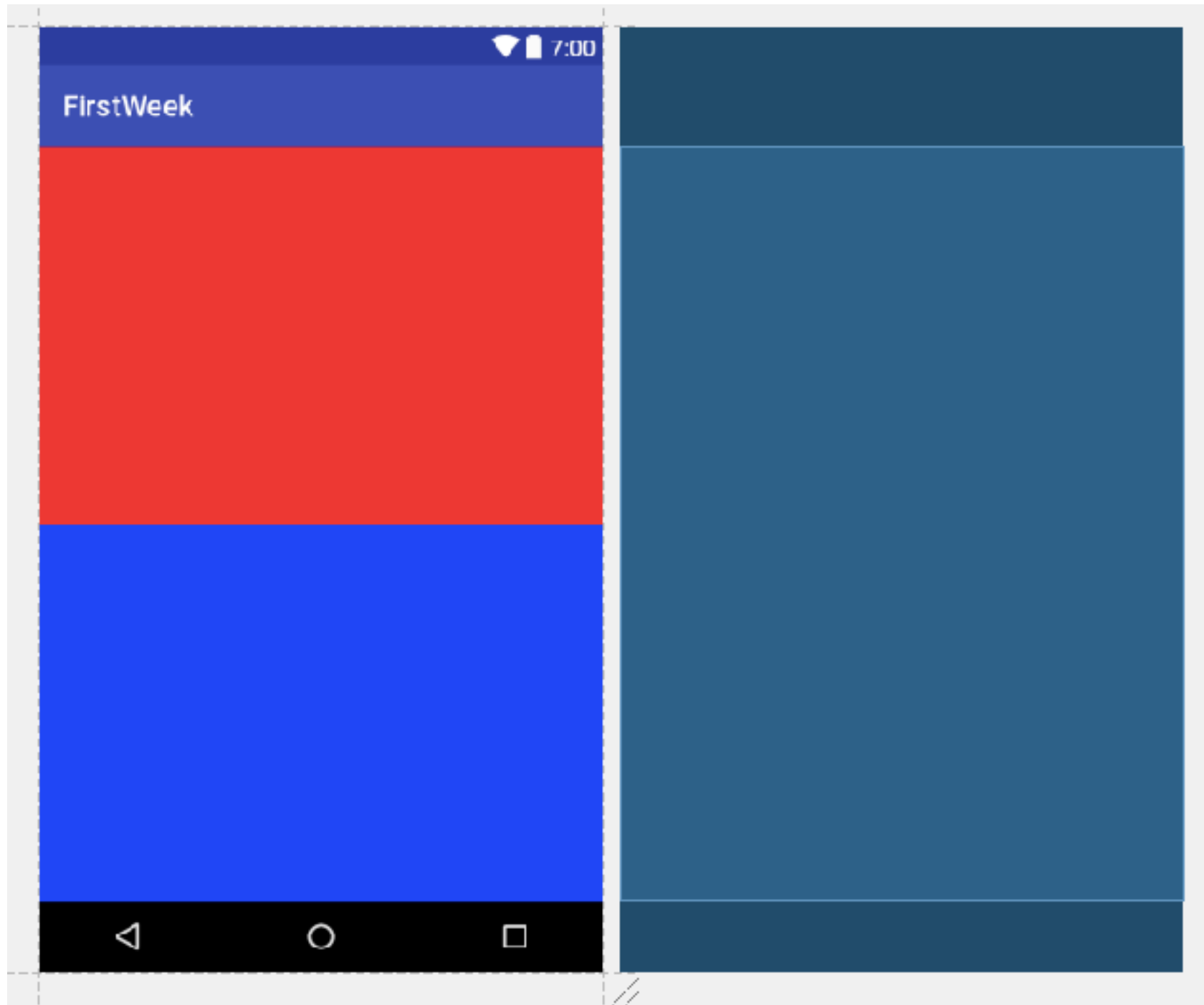
    <CheckBox
        android:id="@+id/checkbox"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="CheckBox" />
</LinearLayout>
```

- XML 창으로 가보면 LinearLayout 안에 여러 View 요소들이 생긴걸 볼수 있다.
- LinearLayout의 속성중 orientation에 주목해보자
- Android:orientation="horizontal"로 바꾸면 어떤 현상이 벌어질까?

# LinearLayout

- 안드로이드 가장 기본적인 화면을 이루는 단위인 Layout의 한 종류
- layout의 위에는 다른 view들이 올라갈 수 있다.
- layout위에 layout이 올라갈 수 있다. (layout도 뷰)
- orientation이라는 속성값이 있다.
- 뷰들이 박스처럼 쌓인다. (vertical : 위아래, horizontal : 좌우)
- 보통 큰 틀을 잡을 때 많이 사용된다.
- layout\_weight값을 통해 포함된 view들간의 비율을 정할 수 있다.

# LinearLayout



- height를 0으로 주고 layout\_weight를 1로 주면 화면을 같은 비율로 나눠서 차지하게 된다.

# LinearLayout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context="io.github.gingeraebi.firstweek.MainActivity">

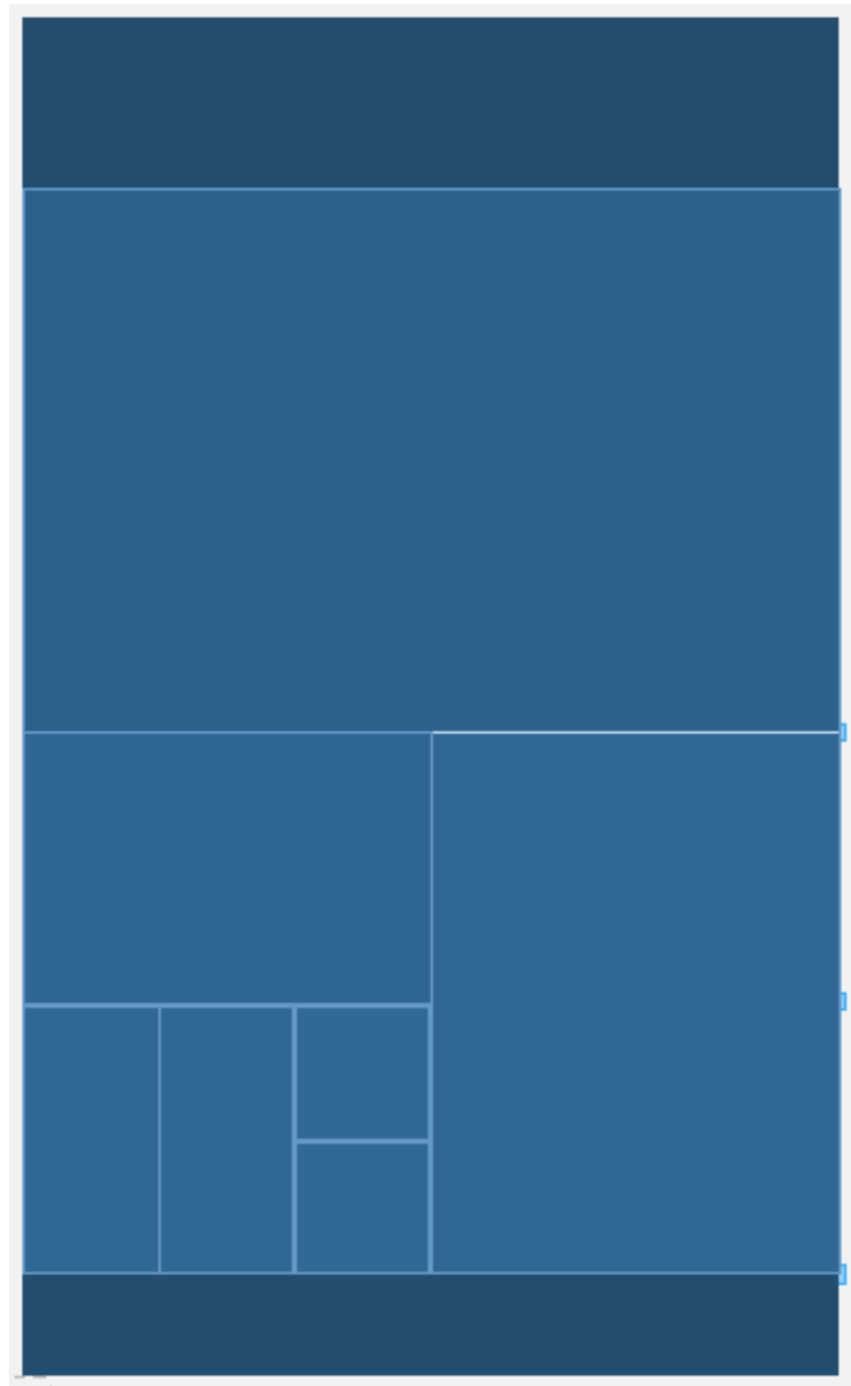
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:background="#FF0000"
        android:orientation="horizontal"></LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:background="#0000FF"
        android:orientation="horizontal"></LinearLayout>
</LinearLayout>
|
```

- 3개의 레이아웃이 화면을 같은 비율로 나누려면 어떻게 해야할까?

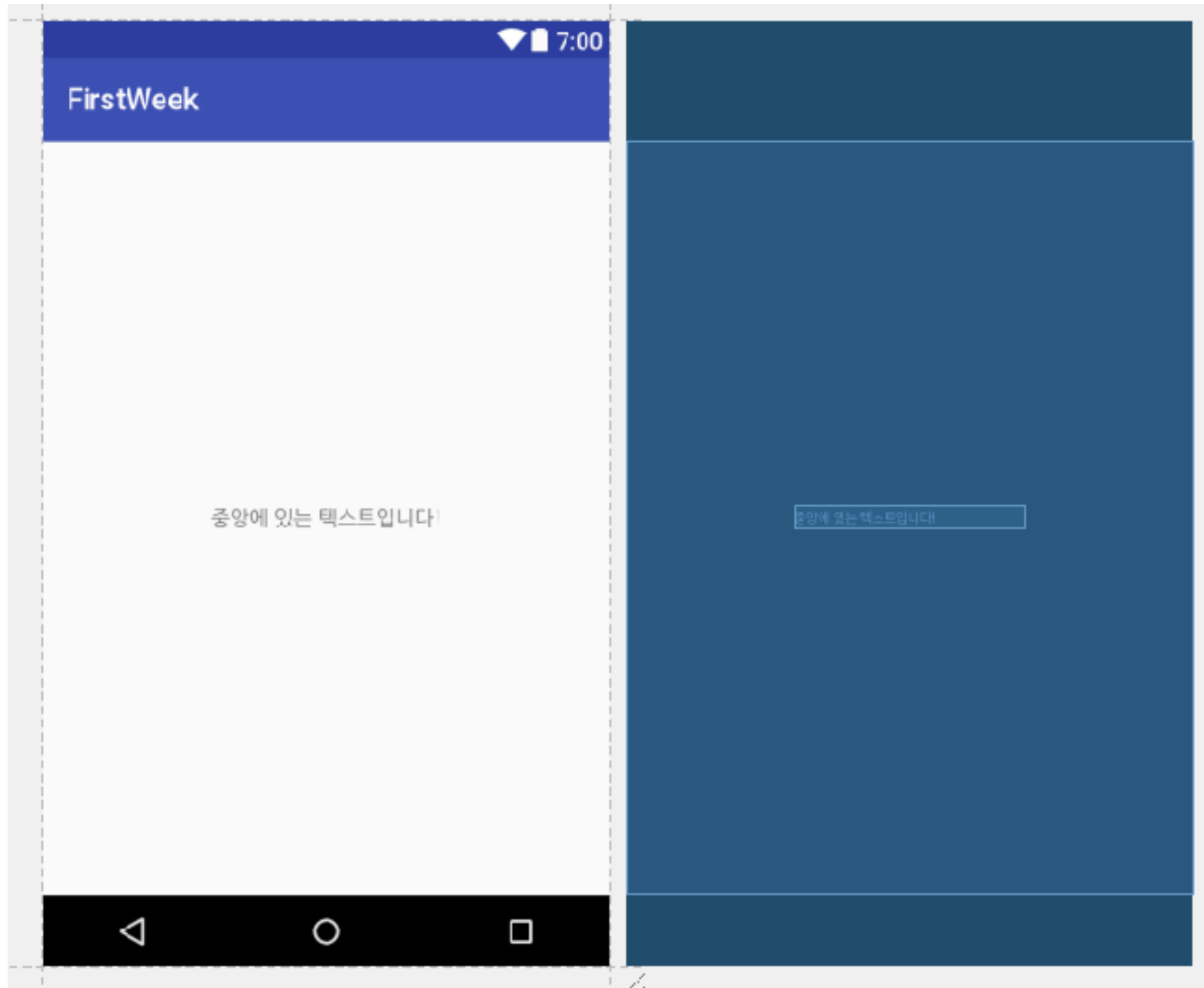


# LinearLayout 연습



- 위와 같은 구조를 가진 레이아웃을 만들어보자.

# RelativeLayout



- 레이아웃 내부 뷰끼리 관계를 통해 뷰들의 위치를 정함

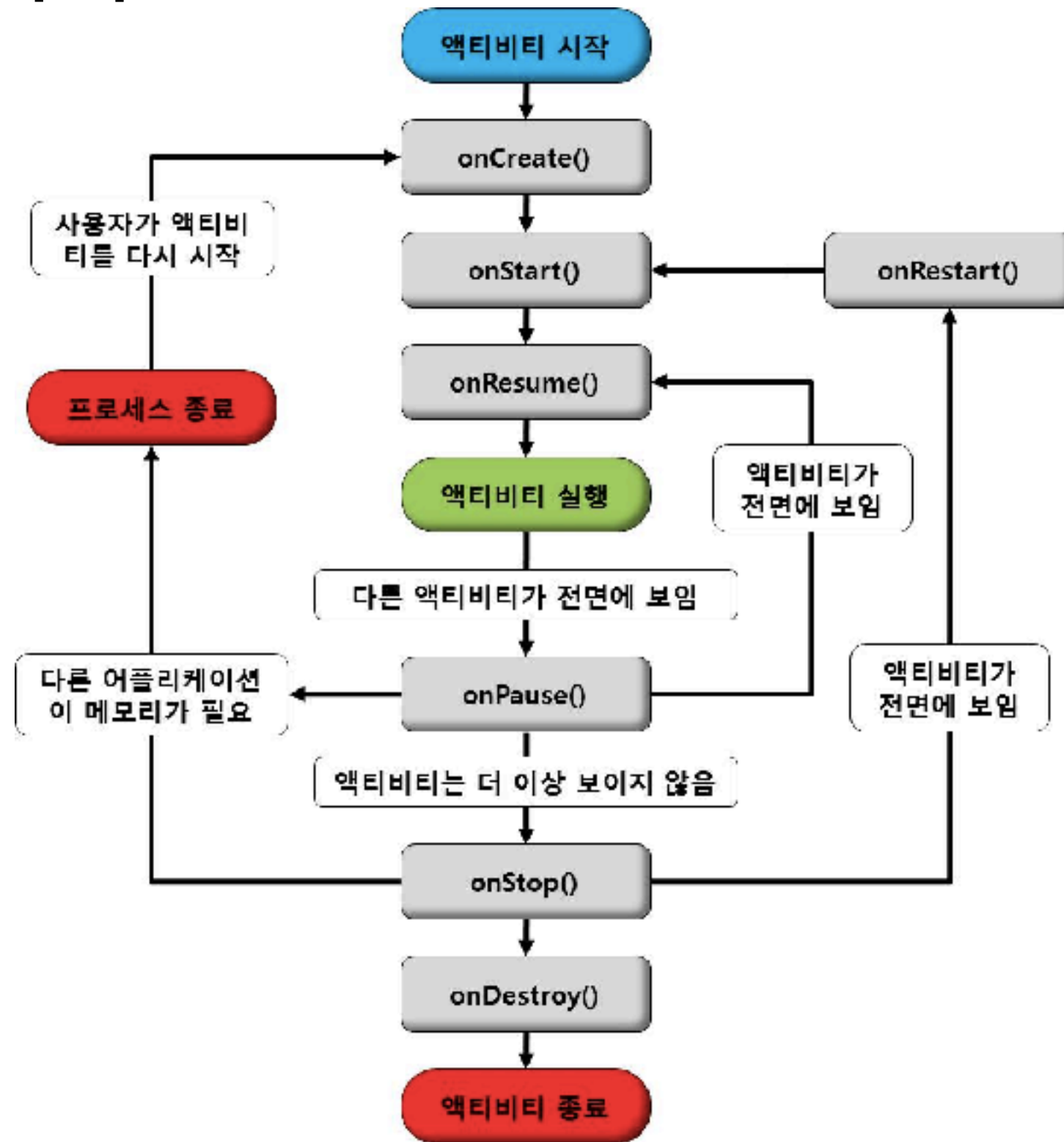
# RelativeLayout

- 레이아웃 내부 뷰끼리 관계를 통해 뷰들의 위치를 정함
- 부모와의 관계를 통해 뷰의 위치를 정하거나
- 같은 깊이(depth)에 있는 뷰끼리의 관계를 통해 위치를 정한다.
- 자세한 설명과 예제는 라이브로 보여드리는것으로...

# Activity

- 뷰들을 올려서 쓸 수 있는 도화지의 개념
- 기본적으로 액티비티 1 : ~~Activity.java 1 : activity\_~~.xml 1 의 관계를 가진다
- 실행, 일시정지, 정지의 상태를 가진다.
- 실행 : 사용자가 화면을 보고 있는 상태
- 일시정지 : 액티비티가 보여지고 있지만 조작이 불가능한 상태 (팝업)
- 정지 : 화면이 다른 액티비티에 완전히 가려진 상태
- 액티비티 위에는 액티비티가 올라갈 수 없다.

# Activity 생명주기



- 자세한건 예제를 통해 알아보자