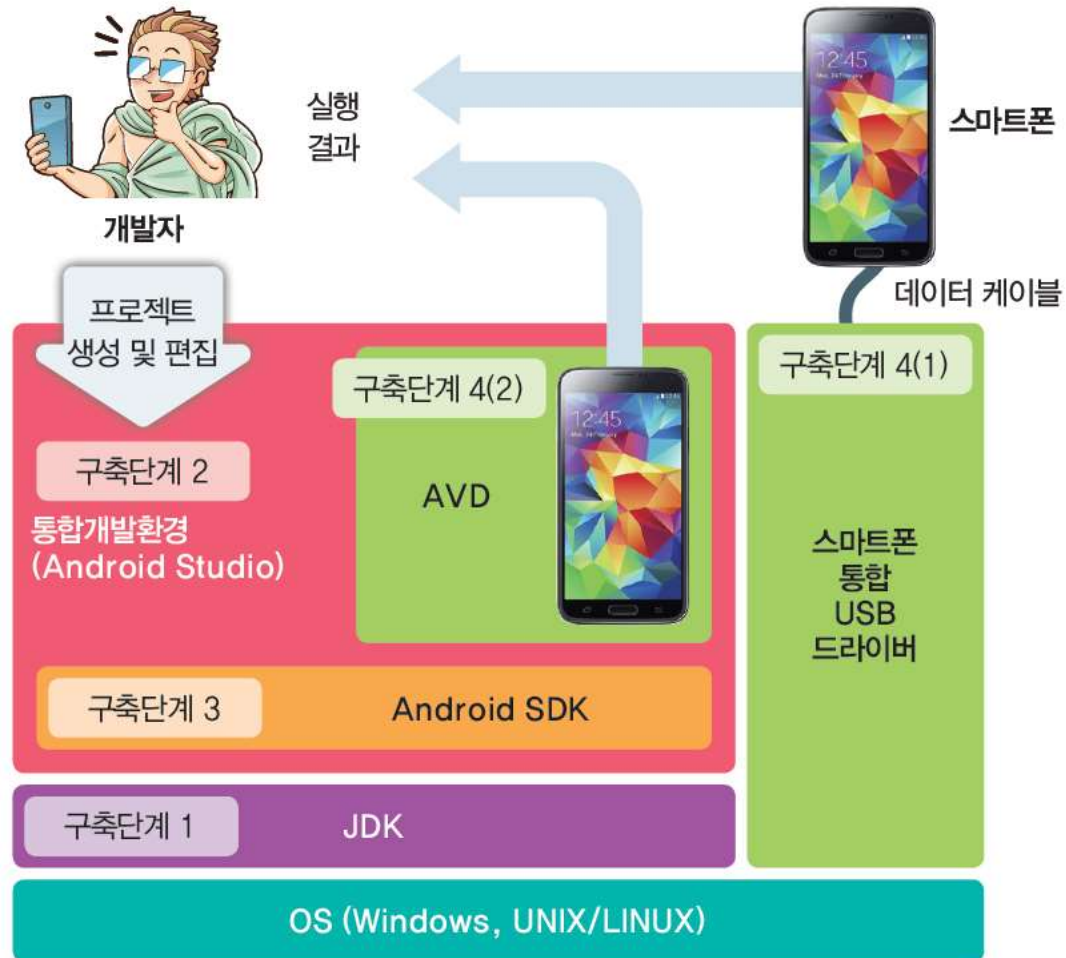


# 첫 번째 애플리케이션

Mobile Software  
2022 Fall

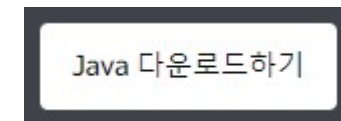
All rights reserved, 2022, Copyright by Youn-Sik Hong (편집, 배포 불허)

# App. 개발 환경

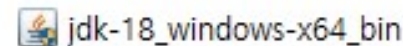


# 단계 1: JDK 설치

- <http://www.oracle.com/kr/java>
  - 화면 오른쪽 위에 놓인 다운로드 아이콘 클릭
  - JDK SE **18** (2022.8. 기준) 선택
    - **JDK**: *Java Development Kit*, **SE**: *Standard Edition*



- **JDK 18 다운로드**
  - Windows 운영체제 64비트 기준

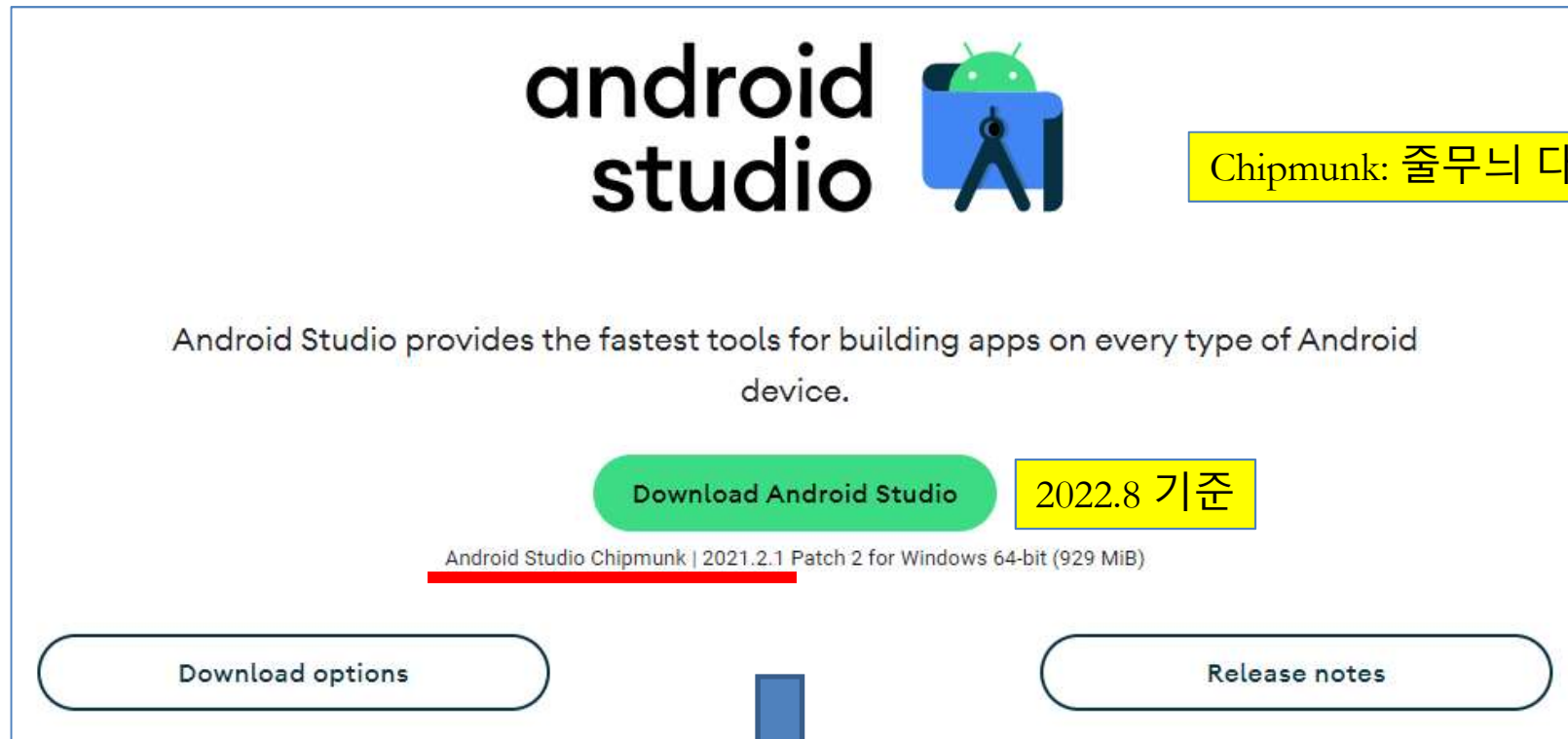


- DOS Command 창에서 Java 설치 확인  
c:\w> **java -version**

EE : Enterprise Edition, ME : Micro Edition, JRE : java Runtime Environment

# 단계 2: Android Studio 설치

- <http://developer.android.com/studio>



Chipmunk: 줄무늬 다람쥐

2022.8 기준

Android Studio Chipmunk | 2021.2.1 Patch 2 for Windows 64-bit (929 MiB)

Download options

Release notes

Platform 별로  
다운로드 할 Studio 선택 가능

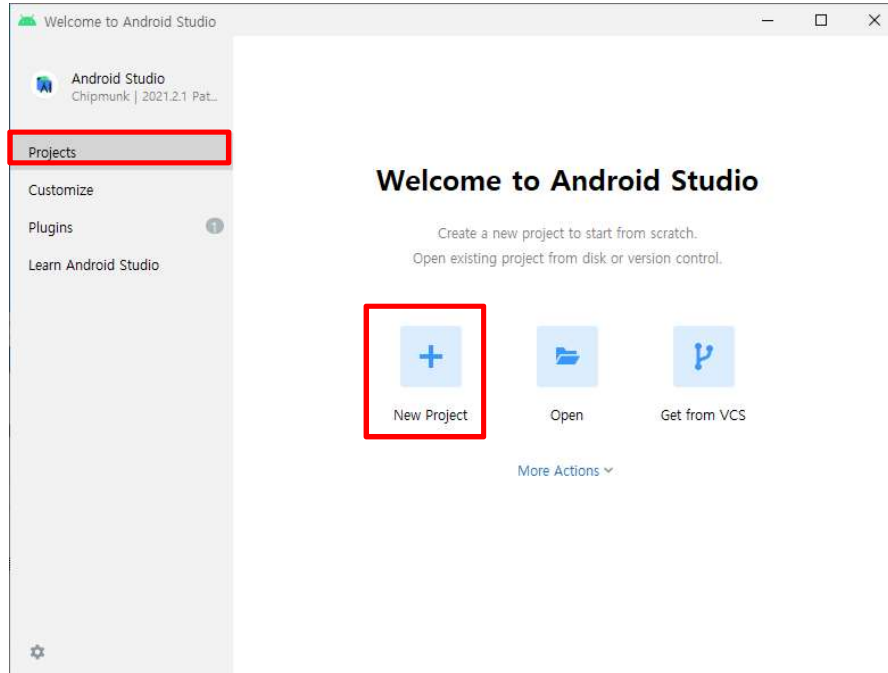
이용약관 동의 필요

Chipmunk에서 추가된  
새 기능과  
개선된 사항

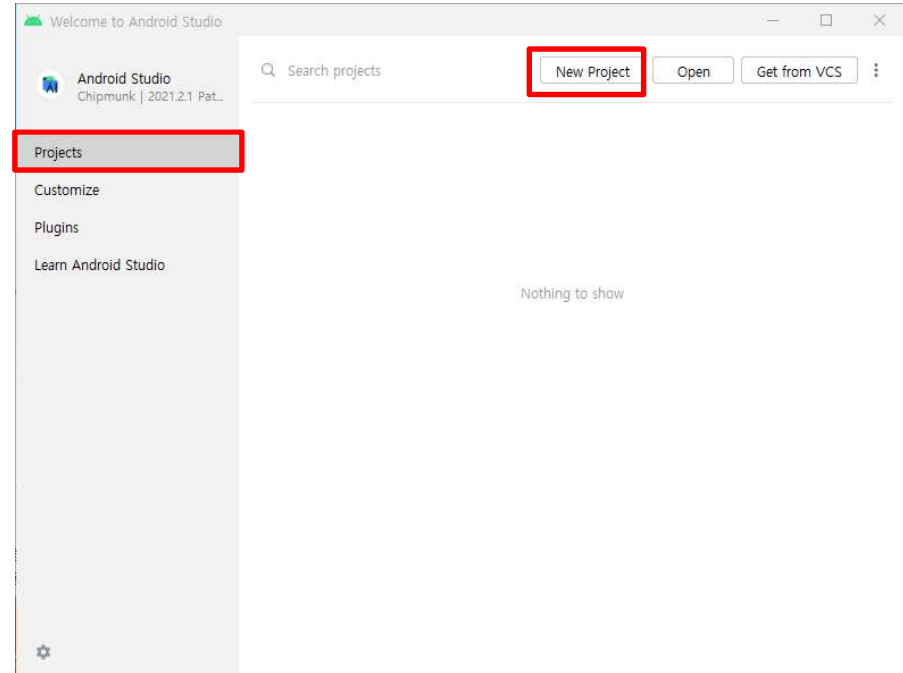
# What to do next?

- **MyApplication 프로젝트 만들고 실행**
  - **Android studio 화면 이해**
  - AVD를 사용한 App. 실행
- **용어**
  - **Project, Package, Empty Activity**

# Creating New Project

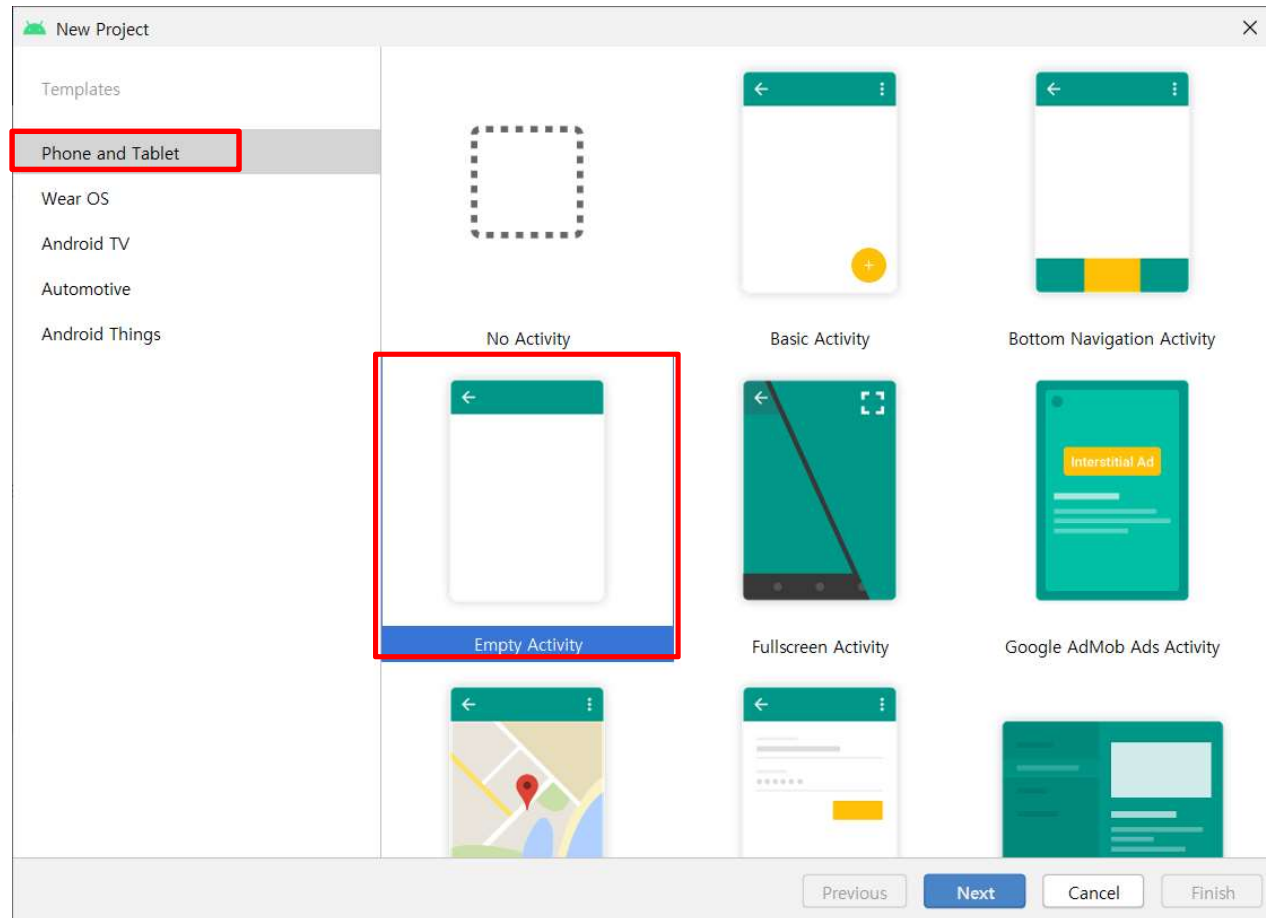


Welcome screen



초기 화면  
(한 번 이상 project를 만들었을 때)

# Creating an Activity: **Empty Activity**



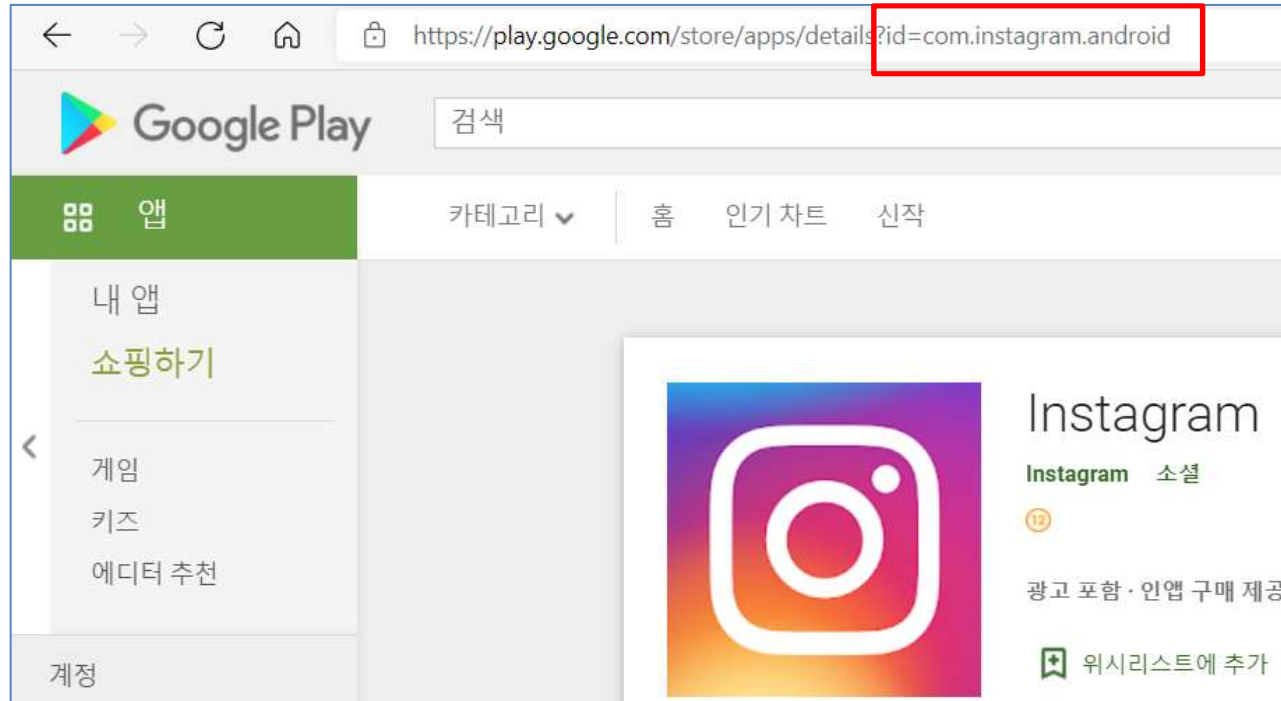
# Defining the Project and SDK Settings

The screenshot shows the 'New Project' dialog in Android Studio. The dialog is titled 'New Project' and has a close button (X) in the top right corner. It contains the following fields and options:

- Empty Activity**: A section header.
- Creates a new empty activity**: A descriptive text.
- Name**: A text field containing 'My Application'. A red box highlights the text, and a yellow box with the text '대소문자 구분(camel case)' points to it.
- Package name**: A text field containing 'com.example.myapplication'. A red box highlights the text, and a yellow box with the text 'small case only' points to it.
- Save location**: A text field containing 'C:\Users\YSHong\AndroidStudioProjects\MyApplication'. A red box highlights the folder icon on the right, and a yellow box with the text 'Project 저장 경로 및 폴더' points to it.
- Language**: A dropdown menu showing 'Kotlin'. A yellow box with the text '개발 언어: kotlin' points to it.
- Minimum SDK**: A dropdown menu showing 'API 21: Android 5.0 (Lollipop)'. A red box highlights the text.
- Information**: A section with a blue 'i' icon and the text 'Your app will run on approximately 98.6% of devices.' and a link 'Help me choose'.
- Use legacy android.support libraries**: A checkbox that is unchecked. A dashed blue box surrounds this section, and a yellow box with the text 'Legacy support libraries: 이전 버전' points to it.
- Buttons**: 'Previous', 'Next', 'Cancel', and 'Finish' buttons at the bottom.

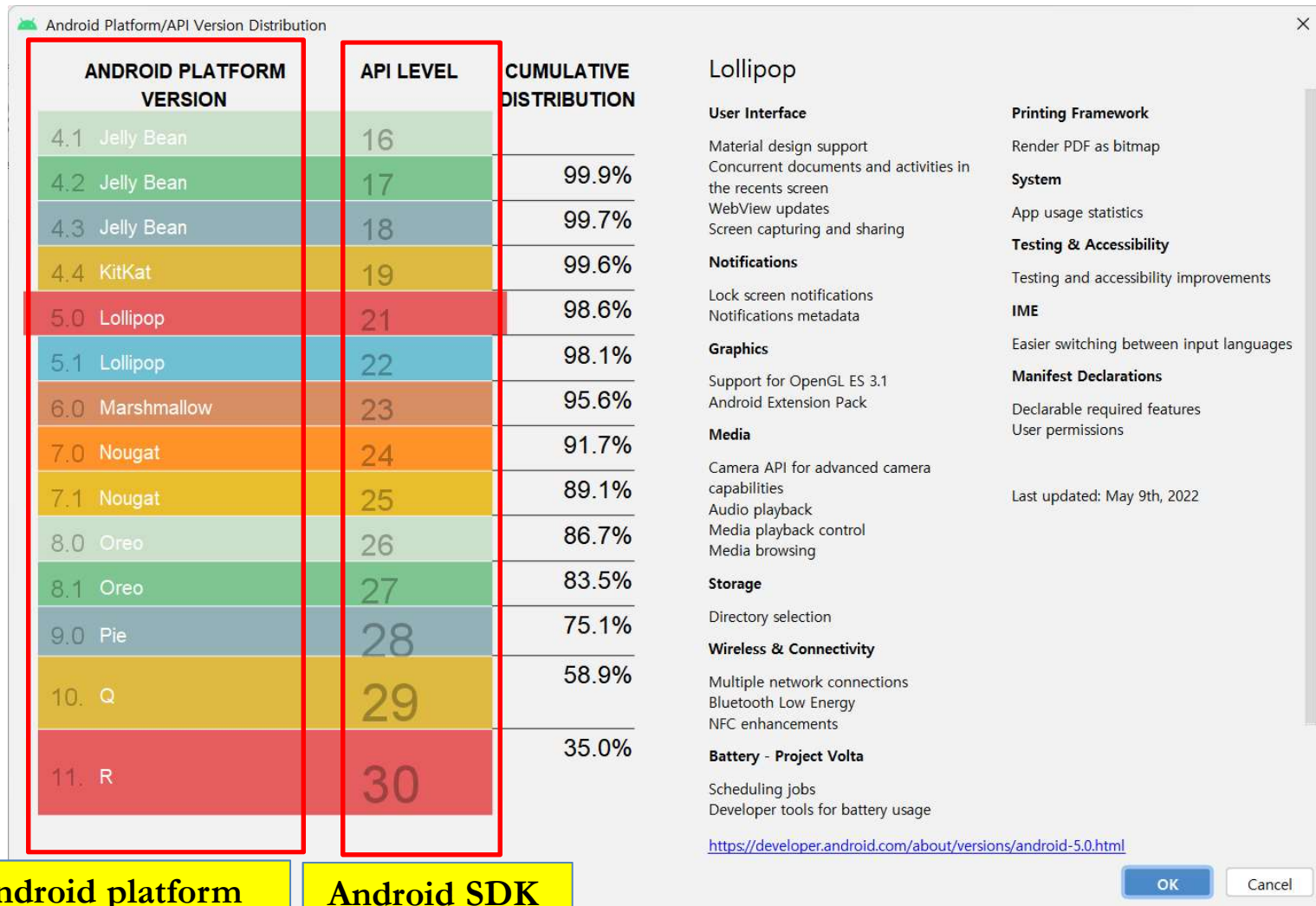


# Package name should be unique



It should *not clash* with others app's package name in **Google Play Store**.

# Android Platform, SDK, and API



Android platform  
= Android System

Android SDK  
= Android API

# SDK manager

Search projects

New Project Open Get from VCS

Virtual Device Manager  
Profile or Debug APK  
SDK Manager  
Import an Android Code Sample

My Application  
~\AndroidStudioProjects\MyApplication

Appearance & Behavior

- Appearance
- Menus and Toolbars
- System Settings
  - HTTP Proxy
  - Data Sharing
  - Date Formats
  - Updates
  - Process Elevation
  - Passwords
  - Android SDK
  - Memory Settings
  - Notifications
  - Quick Lists
  - Path Variables
- Keymap
- Editor
- Build, Execution, Deployment
- Languages & Frameworks
- Tools
- Advanced Settings

Manager for the Android SDK and Tools used by the IDE

Android SDK Location: C:\Users\Wyshon\AppData\Local\Android\Sdk Edit Optimize disk space

SDK Platforms SDK Tools SDK Update Sites

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, the IDE will automatically check for updates. Check "show package details" to display individual SDK components.

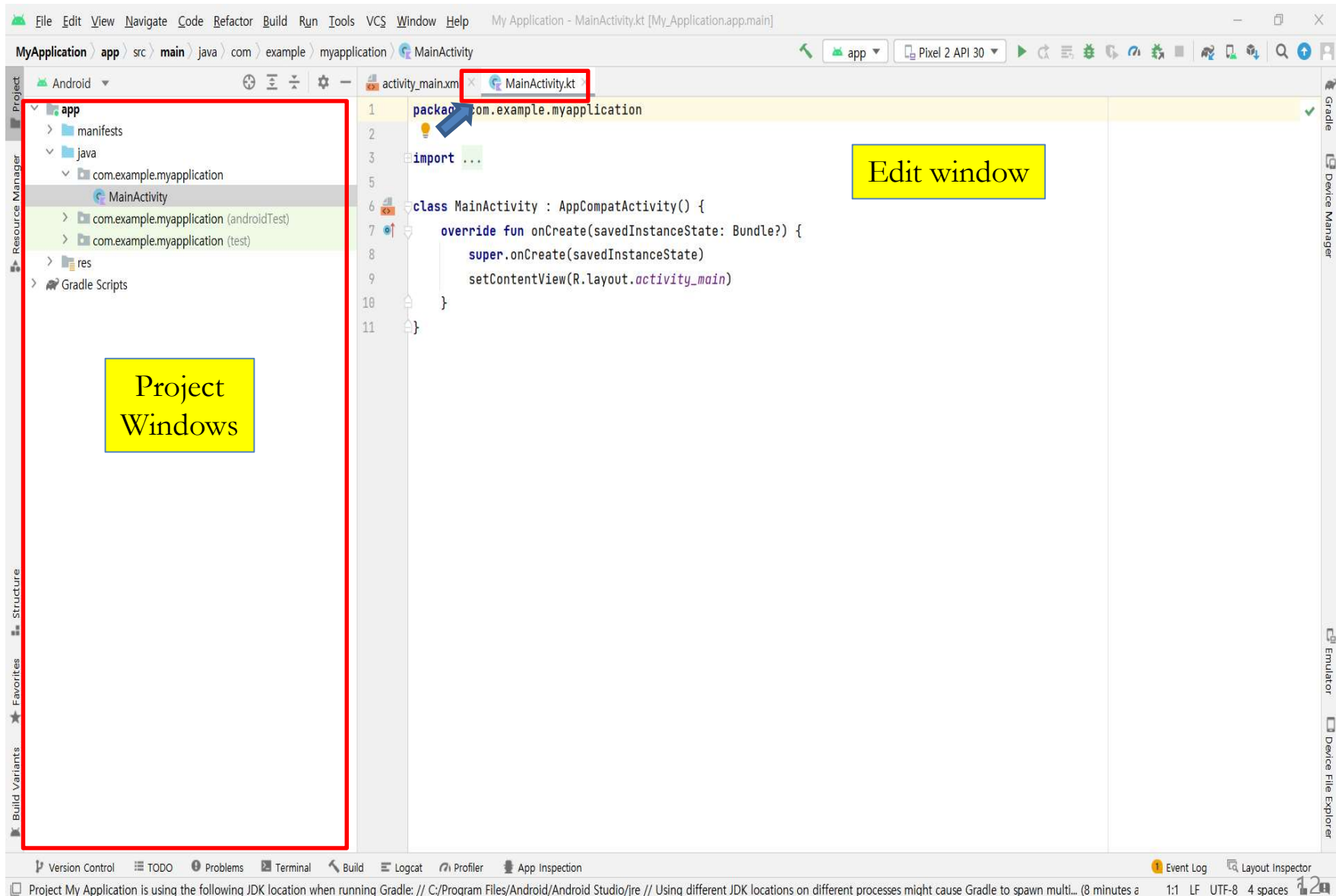
Name	API Level	Revision	Status
<input type="checkbox"/> Android API 33	33	1	Not installed
<input type="checkbox"/> Android Tiramisu Privacy Sandbox Preview	Tiramisu Privacy Sandbox	2	Not installed
<input checked="" type="checkbox"/> Android API 32 (Sv2)	32	1	Partially installed
<input type="checkbox"/> Android 12.0 (S)	31	1	Not installed
<input type="checkbox"/> Android 11.0 (R)	30	3	Not installed
<input type="checkbox"/> Android 10.0 (Q)	29	5	Not installed
<input type="checkbox"/> Android 9.0 (Pie)	28	6	Not installed
<input type="checkbox"/> Android 8.1 (Oreo)	27	3	Not installed
<input type="checkbox"/> Android 8.0 (Oreo)	26	2	Not installed
<input type="checkbox"/> Android 7.1.1 (Nougat)	25	3	Not installed
<input type="checkbox"/> Android 7.0 (Nougat)	24	2	Not installed
<input type="checkbox"/> Android 6.0 (Marshmallow)	23	3	Not installed
<input type="checkbox"/> Android 5.1 (Lollipop)	22	2	Not installed
<input type="checkbox"/> Android 5.0 (Lollipop)	21	2	Not installed
<input type="checkbox"/> Android 4.4W (KitKat Wear)	20	2	Not installed
<input type="checkbox"/> Android 4.4 (KitKat)	19	4	Not installed
<input type="checkbox"/> Android 4.3 (Jelly Bean)	18	3	Not installed
<input type="checkbox"/> Android 4.2 (Jelly Bean)	17	3	Not installed

☒ Hide Obsolete Packages ☐ Show Package Details

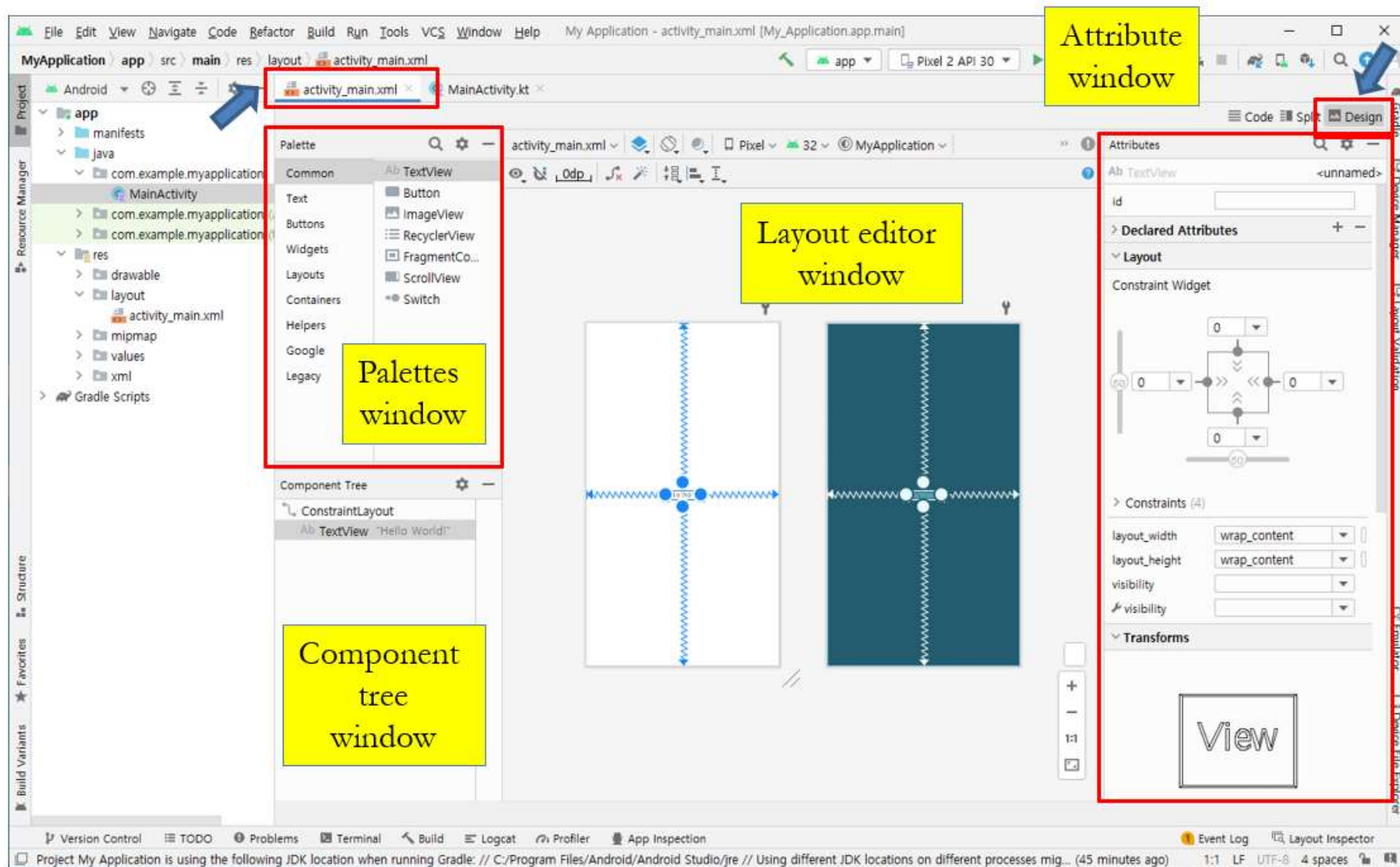
Project-level settings will be applied to new projects

OK Cancel Apply

# Android Studio - code



# Android Studio - layout



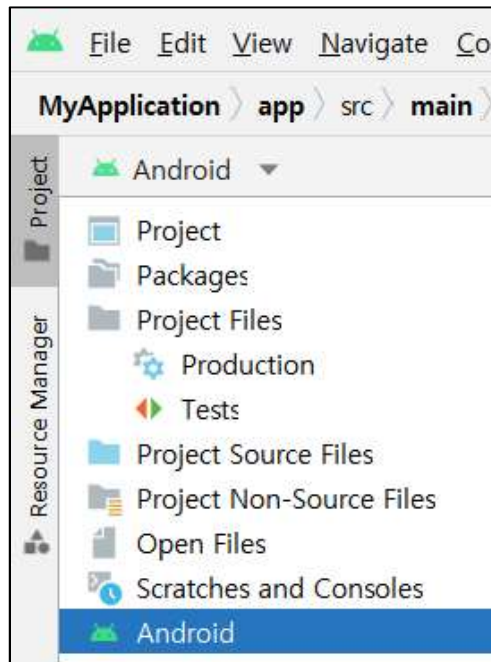
# What to do next?

- 관련 파일을 하나씩 자세히 들여다 보자
  - MainActivity.kt
  - activity\_main.xml

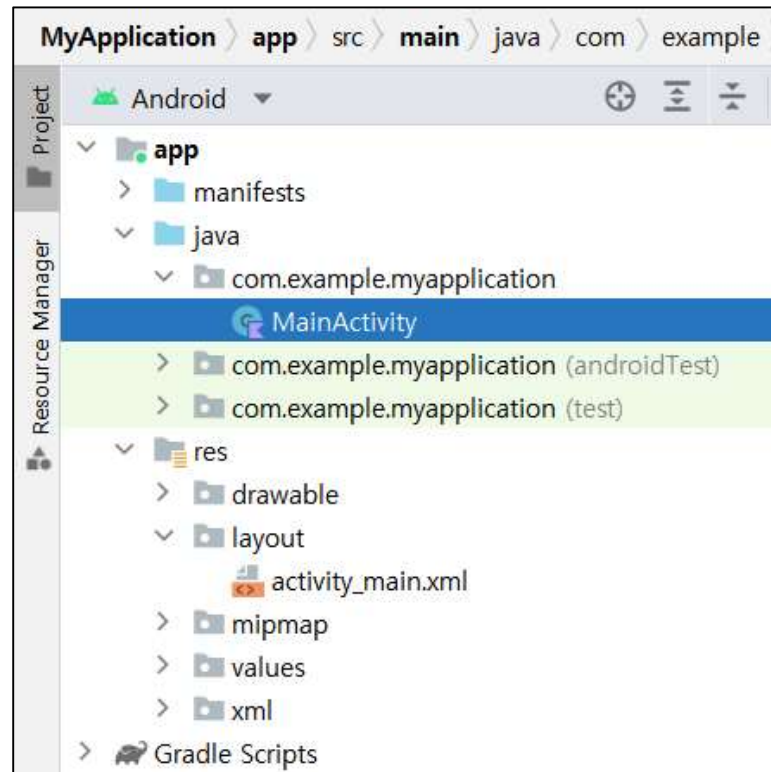


# Where is MainActivity ?

Project  
Windows



kt는 kotlin 확장자



# Code – MainActivity (1/6)

```
activity_main.xml x MainActivity.kt x
1 package com.example.myapplication
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5
6 class MainActivity : AppCompatActivity() {
7     override fun onCreate(savedInstanceState: Bundle?) {
8         super.onCreate(savedInstanceState)
9         setContentView(R.layout.activity_main)
10    }
11 }
```

**Compat** →  
*compatible*  
(호환 가능한)

- **package**
  - class를 보관하는 container
    - 다른 사람이 만든 것과 중복되지 않도록 고유한 이름을 사용
    - 주로 domain name을 패키지 이름에 사용
  - **com.example.프로젝트이름**

- **import**
  - 패키지\_이름.클래스\_이름
    - 외부에서 정의한
    - 클래스를 사용할 때 필요
  - **androidx.xxx**

**androidx**: android extension



# Code – MainActivity (2/6)

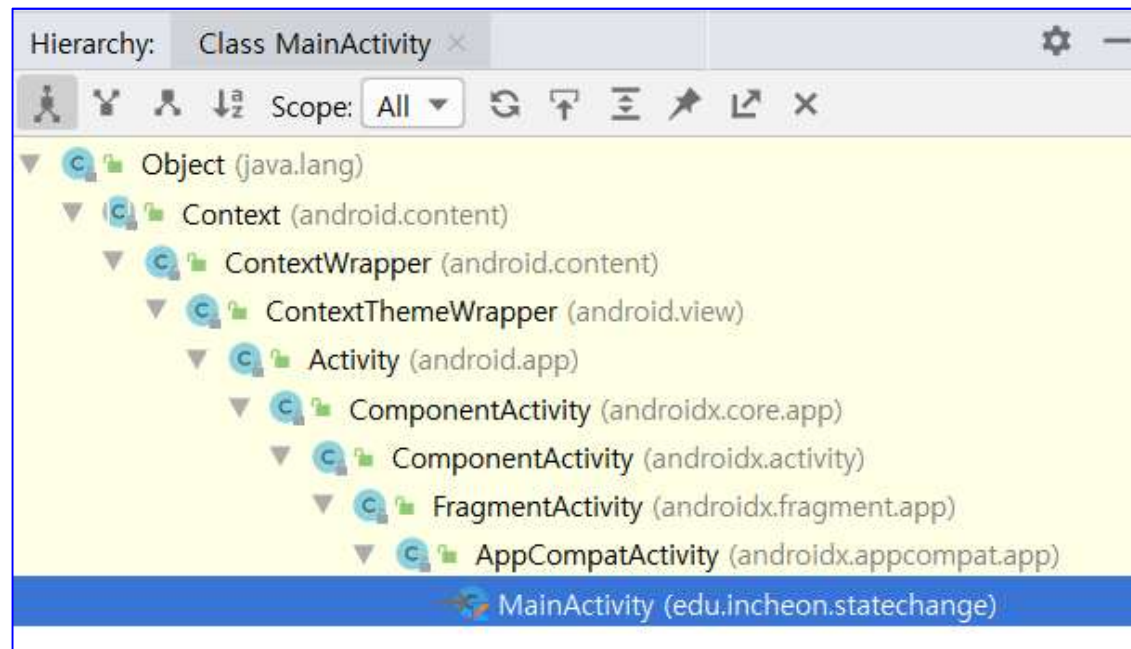
```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
}
```

- **MainActivity** 클래스 : 클래스 이름
  - 클래스 이름이 파일 이름
  - **':' (colon)** → **클래스 상속 기호** (자식 클래스 : 부모 클래스)
- **AppCompatActivity**
  - 부모 클래스(**super class**) : AppCompatActivity
    - 사용자 인터페이스(UI)를 갖는 **Activity** 컴포넌트
  - **AppCompatActivity ( )** 에서
    - 괄호는 클래스의 **생성자(constructor)** 호출
    - 클래스 상속

# 잠깐! Class hierarchy

```
class MainActivity : AppCompatActivity() {
```

ctrl-h



# Code – MainActivity (3/6)

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
}
```

이 method는  
return 값이 없음.

- **override**
  - 부모 클래스로부터 상속받은 method를 자식 클래스에서 재정의
    - MainActivity : AppCompatActivity( )
- **fun** : 함수 (function)
- 함수 onCreate의 매개변수(parameter)는 1개
  - 파라미터 이름 : 데이터 타입

객체지향언어(OOP)에서  
클래스에서 정의한 멤버 함수를  
method라 부름.

savedInstanceState: Bundle?

**OOP:** object oriented programming

# Code – MainActivity (4/6)

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
}
```

**super**  
→ 슈퍼 클래스(super class)  
→ AppCompatActivity

- *dot-notation*
  - 객체(object)의 **method**나 **property** (member field)를 호출하는 문장  
**Object-이름 . 메소드\_이름 ( ... )**  
**Object-이름 . 프로퍼티\_이름**
  - 상속받은 슈퍼 클래스의 method를 호출할 경우  
**super . 메소드\_이름 ( ... )**
- 슈퍼 클래스인 **AppCompatActivity** 에서 정의한 **onCreate** 메소드를 호출  
**super.onCreate(savedInstanceState)**

함수 → method  
변수 → property

# Code – MainActivity (5/6)

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
}
```

- 메소드 이름이 **onXXX, OnXXX** 이면 **event handler**
  - 이벤트가 발생했을 때 이벤트를 처리하는 **callback 메소드**
- **onCreate( )** 메소드는 언제, 누가 호출할까?
  - Activity가 생성될 때 한 번 호출됨.
    - Android system에서 호출.
    - 사용자가 직접 이 메소드를 호출할 수 없음.
  - **Activity 의 lifecycle** (생명 주기)과 관련 있음.

이벤트(event) → 휴대폰 화면을 터치, 가상 키보드로 키를 입력, ...  
Callback 이란? → Don't call me directly. I'll call you.  
내(시스템)가 전화(호출)할 때까지 기다려...  
언제 전화할 지 모르지만 늘 준비하고 기다려...

# Code – MainActivity (6/6)

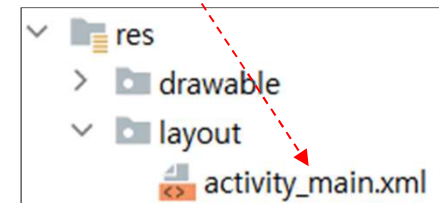
```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
}
```

- **setContentView**

- 사용자 인터페이스(User Interface, UI) 화면을 출력.
- (UI를 정의한) XML 레이아웃을 출력.
  - 이 메소드를 호출하지 않으면 화면에 아무 것도 나타나지 않음.

- **R.layout.activity\_main**

- **activity\_main.xml**을 가리킴
  - UI 화면을 정의한 XML 파일의 이름
- 정수 상수 : XML 파일에 해당하는 resource-id



```
setContentView(R.layout.activity_main)
```

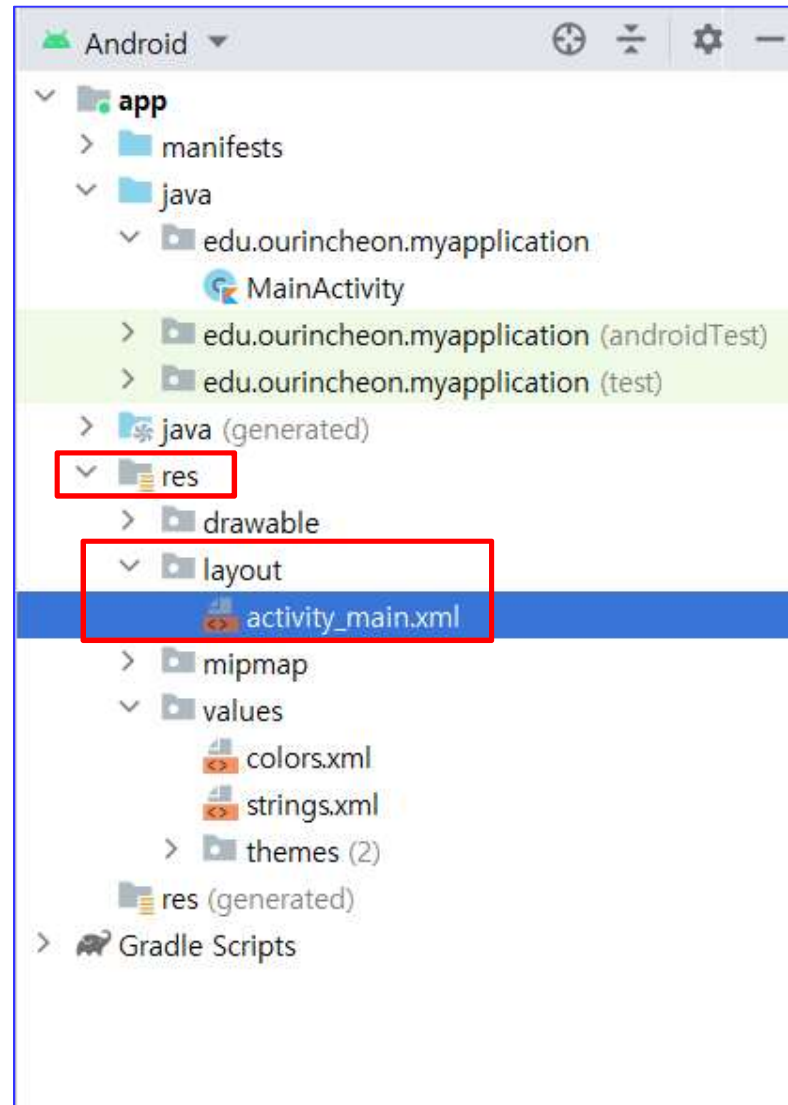
```
edu.ourincheon.myapplication.R.layout  
public static final int activity_main = 1300085 ;
```

# What to do next?

- 관련 파일을 하나씩 자세히 들여다 보자
  - MainActivity.kt
  - **activity\_main.xml**

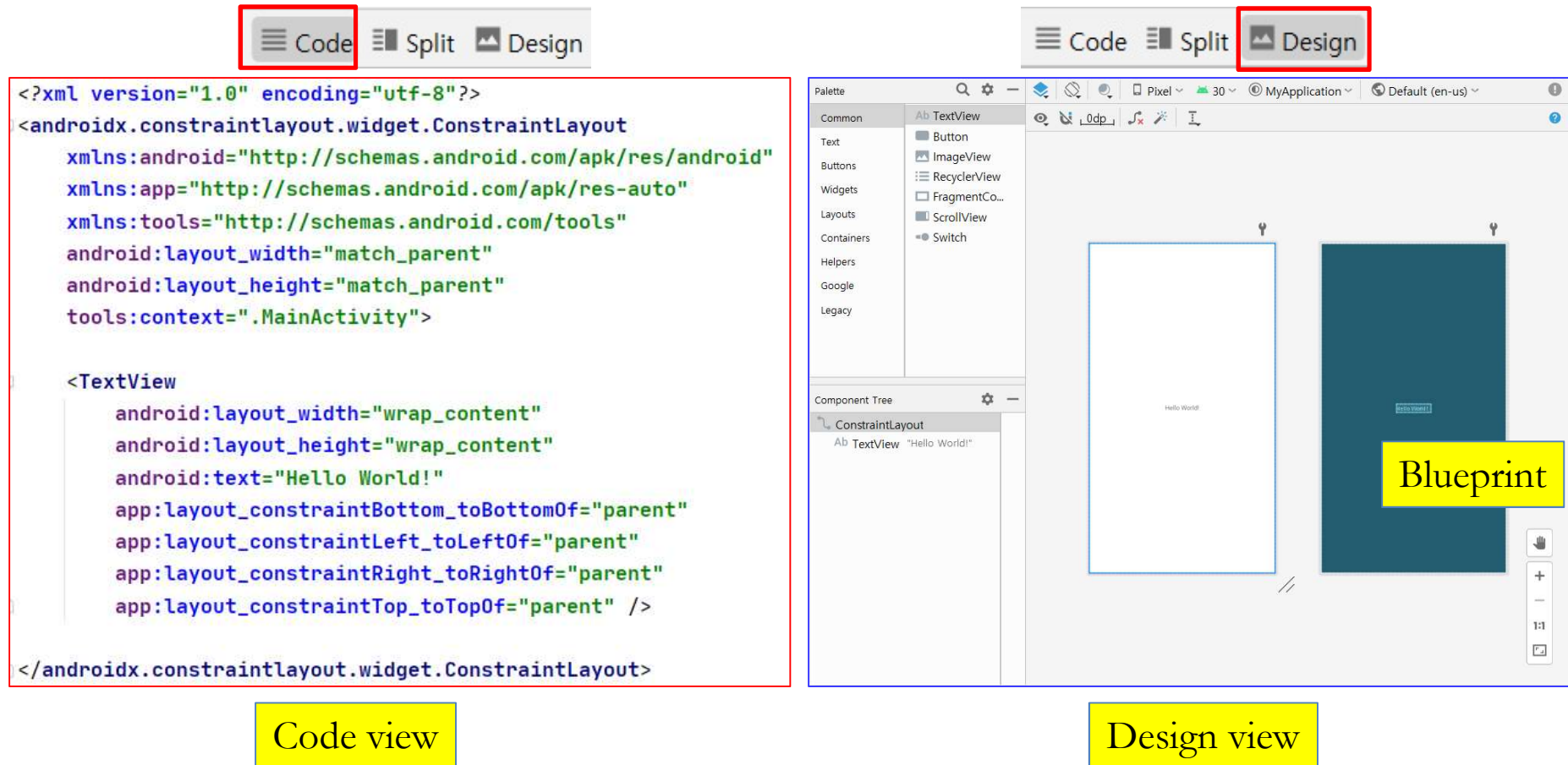
# Where is activity\_main.xml?

Project  
Windows





# activity\_main.xml – code, design



Code view

Design view

- XML 파일은 텍스트 파일이지만
- 사용자 인터페이스(UI) 화면을 설계 → 화면 구성(layout)

# Layout - activity\_main.xml (1/3)

```
<?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>
```

## xmlns:android

- Android SDK에 정의된 tag 이름 및 속성을 사용
  - ns : namespace
- xmlns:xxx : xxx 는 prefix

## xmlns:tools

- Android studio에서만 사용
- Preview 화면에서 XML로 설계한 UI 속성을 테스트할 때 사용

## xmlns:android

- android에서 정의한 tag 이름 및 속성을 사용

## xmlns:app

- 외부 라이브러리에서 정의한 tag 이름 및 속성을 사용

# Layout - activity\_main.xml (2/3)

```
<?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">

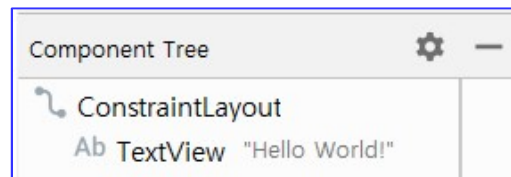
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

- 계층 구조(트리 구조)
- **Root element**
  - **ConstraintLayout**
  - Root element는 하나
- Root element에서만
  - name space 속성 지정

- 2개의 view를 정의
- **ConstraintLayout**
  - 외부 라이브러리에서 가져오기 때문에 클래스 경로를 지정
- **TextView**
  - Android SDK에 포함

Component Tree



# Layout - activity\_main.xml (3/3)

```
<?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>
```



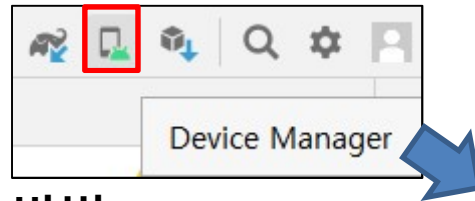


# What to do next?

- **MyApplication 프로젝트 만들고 실행**
  - Android studio 화면 이해
  - **AVD를 사용한 App. 실행**
- **용어**
  - **AVD**
    - emulator, ABI, 가상화 기술

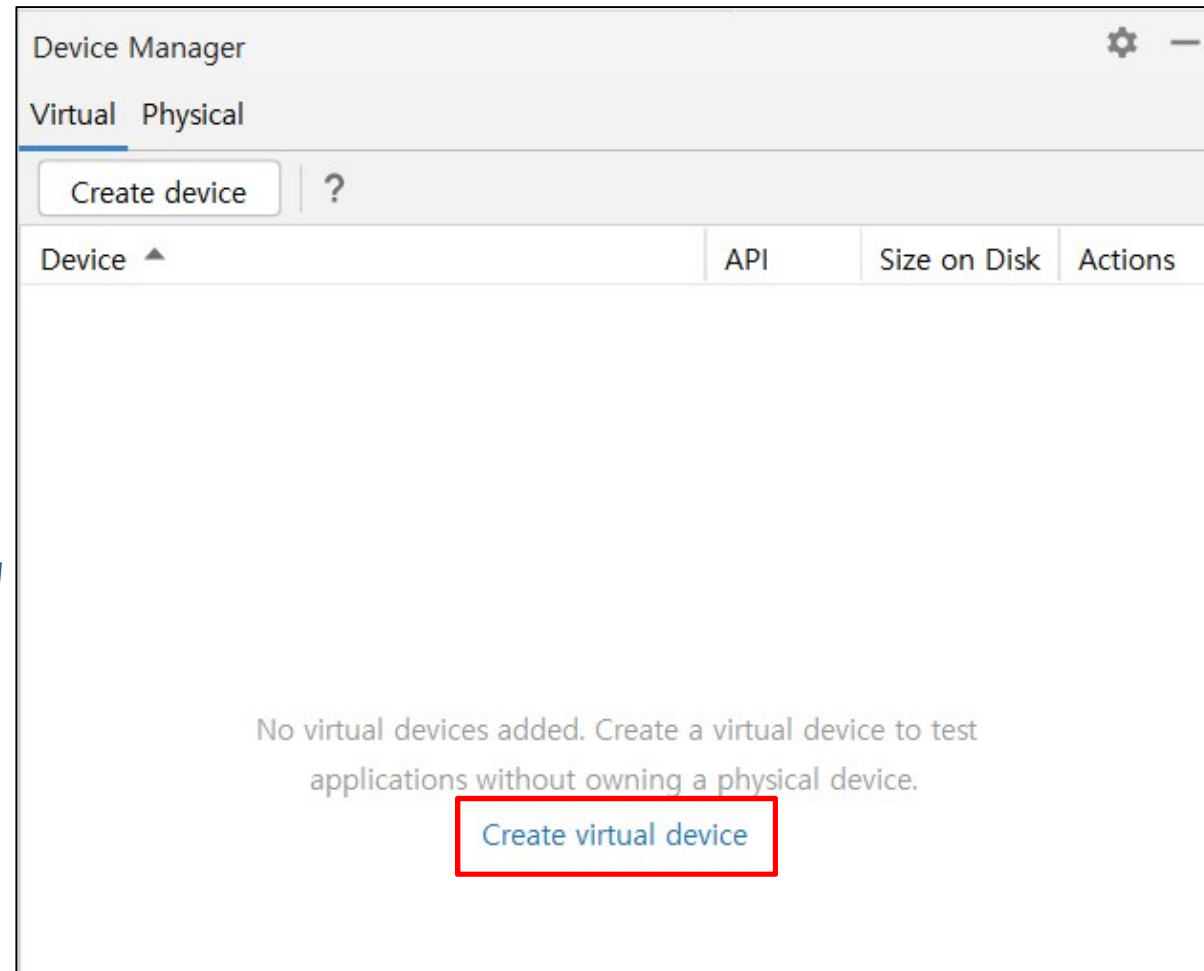
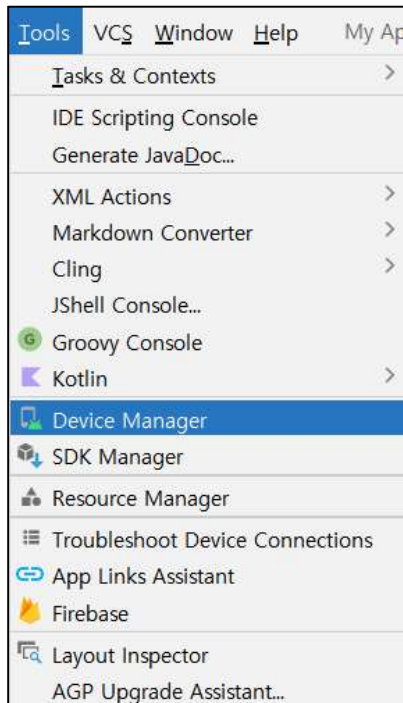
# Create Virtual Device(1/4) : run **AVD manger**

**방법 1** 화면 오른쪽 위에 놓인 휴대폰 모양 아이콘

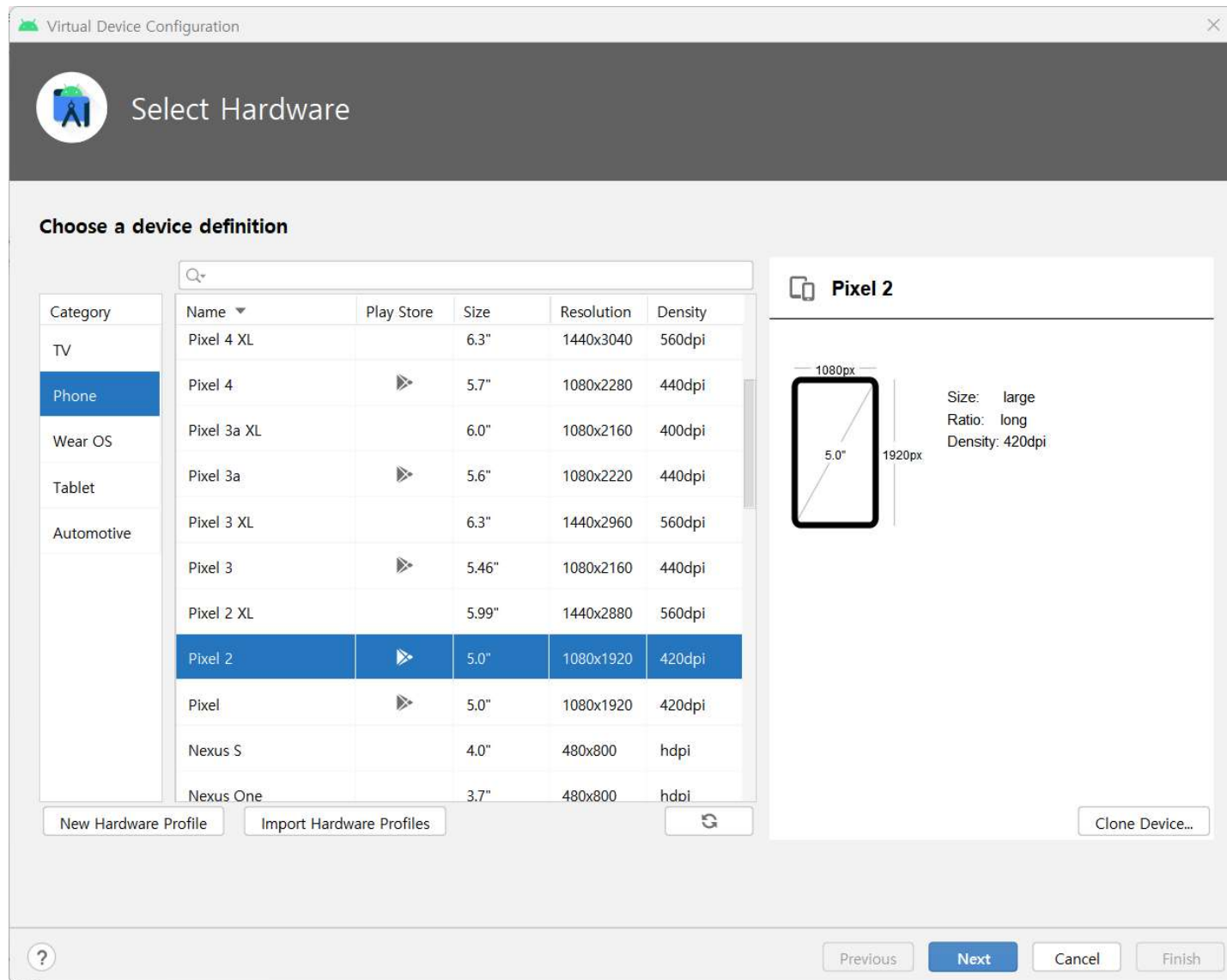


**방법 2**

Tools > Device Manager



# Create Virtual Device(2/4) : Hardware - Pixel 2



# Create Virtual Device(3/4) : System Image - Sv2

- 가상 단말 실행에 필요한 이미지 - 안드로이드 version 선택
- emulator가 모든 안드로이드 version을 시뮬레이션 할 수 없기 때문에, 개발자가 테스트가 필요한 version을 선택.

Release Name	API Level	ABI	Target
<a href="#">TiramisuPrivacySandbox</a>	<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>
<a href="#">API 33</a>	35	x86_64	Android API 33 (Google Play)
<a href="#">Sv2</a>	32	x86_64	Android API 32 (Google Play)
<a href="#">S</a>	31	x86_64	Android 12.0 (Google Play)
<a href="#">R</a>	30	x86	Android 11.0 (Google Play)
<a href="#">Q</a>	29	x86	Android 10.0 (Google Play)
<a href="#">Pie</a>	28	x86	Android 9.0 (Google Play)
<a href="#">Oreo</a>	27	x86	Android 8.1 (Google Play)
<a href="#">Oreo</a>	26	x86	Android 8.0 (Google Play)
<a href="#">Nougat</a>	25	x86	Android 7.1.1 (Google Play)
<a href="#">Nougat</a>	24	x86	Android 7.0 (Google Play)

### Sv2

API Level  
**32**

Android  
**Google Inc.**

System Image  
**x86\_64**

We recommend these Google Play images because this device is compatible with Google Play.

Questions on API level?  
[See the API level distribution chart](#)

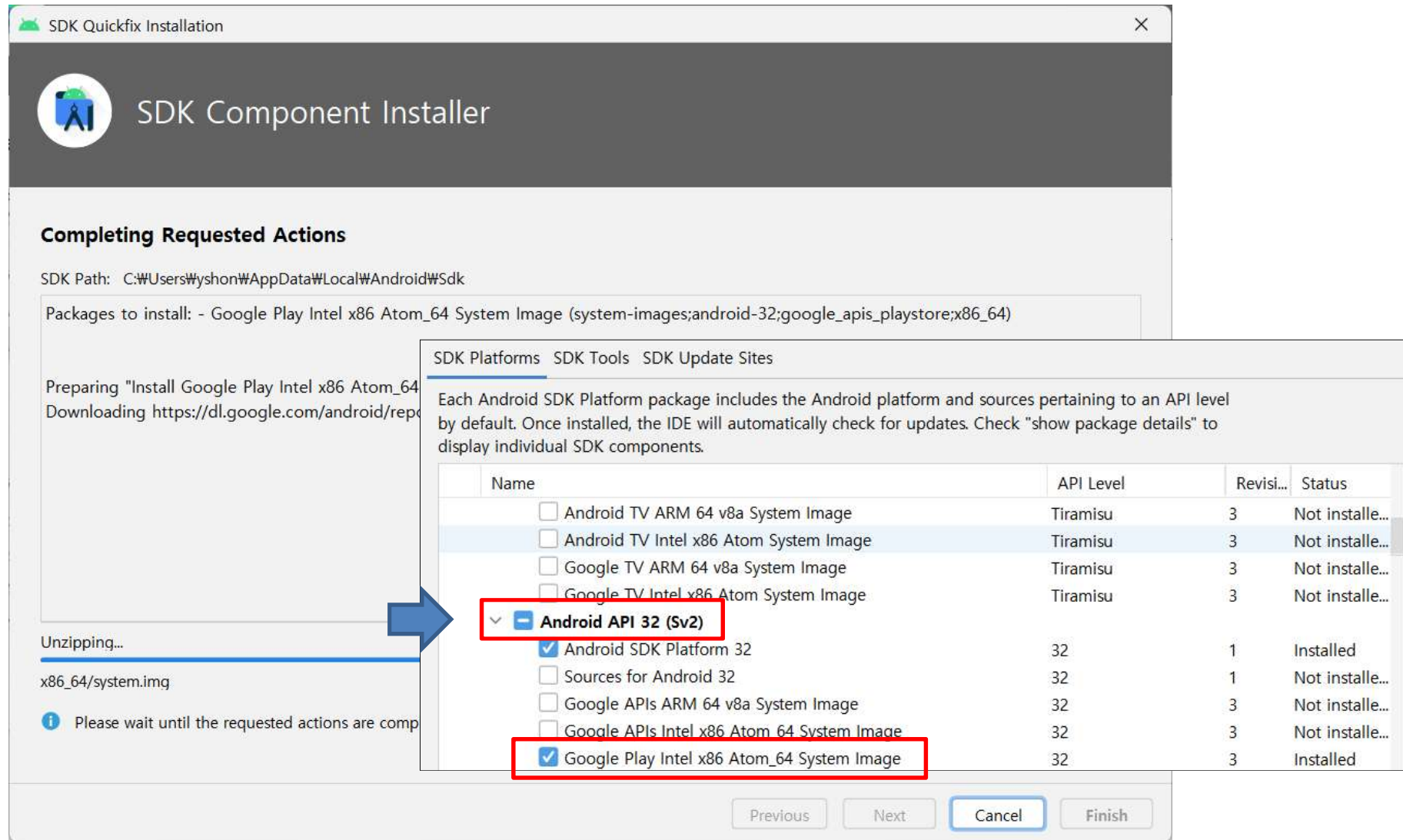
**ABI** : Application Binary Interface  
**Emulator**: 가상 장치에서 동작을 테스트(시뮬레이션)

A system image must be selected.

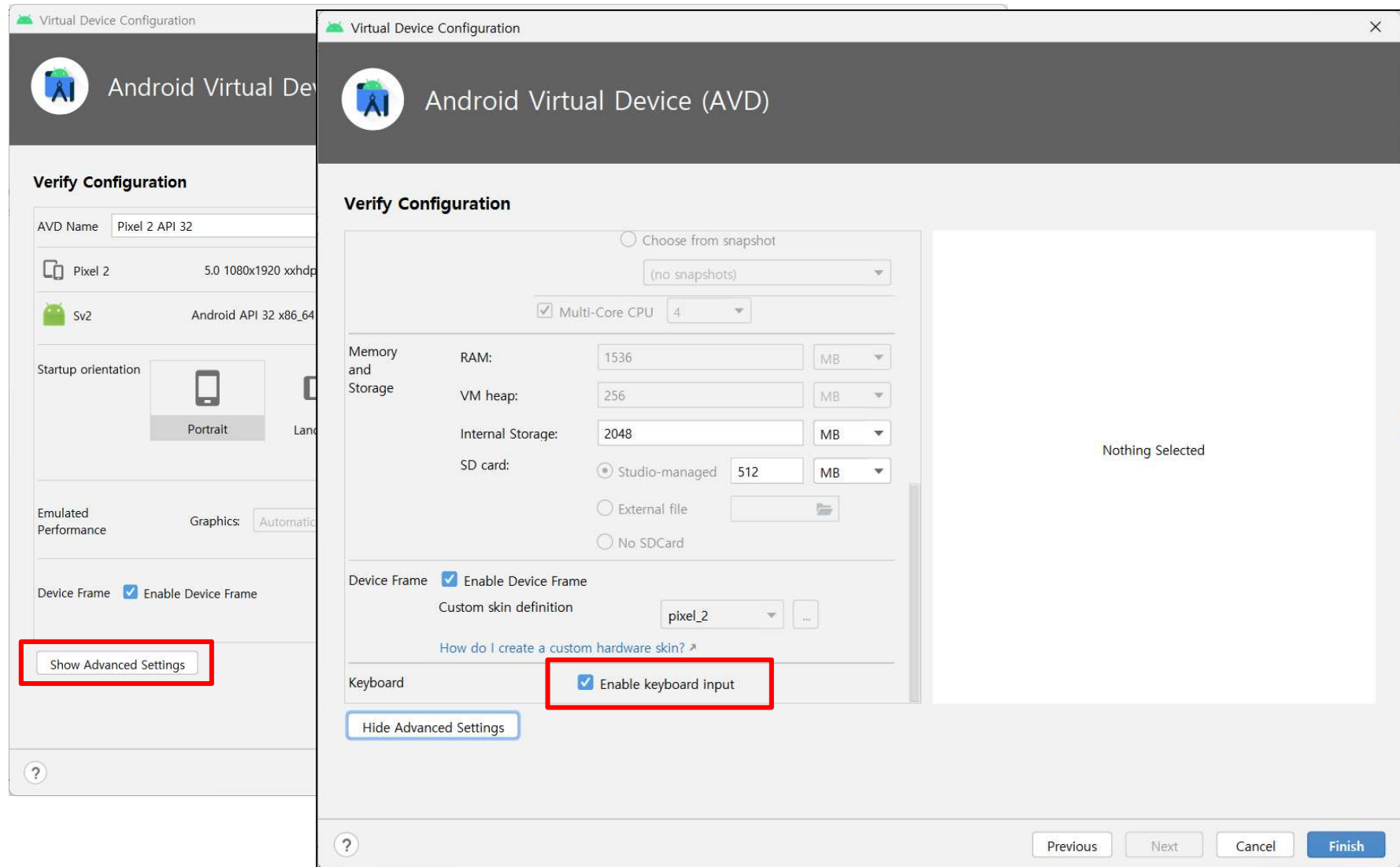
Previous Next Cancel Finish



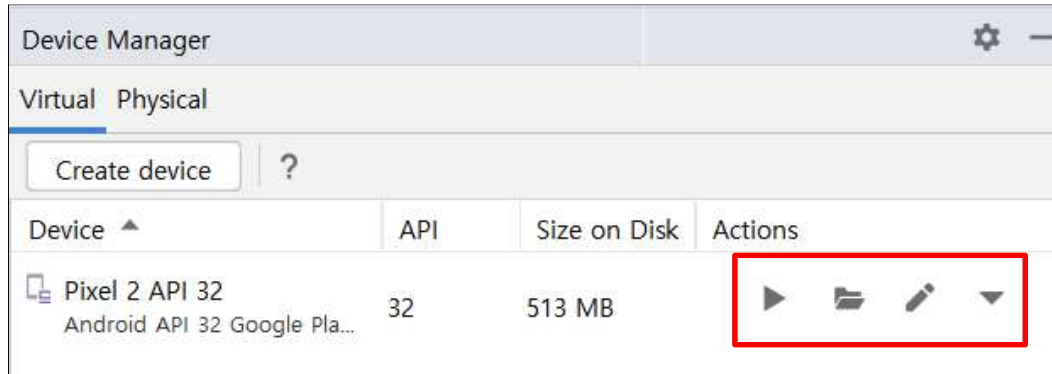
# Create Virtual Device(3/4) : System Image - Sv2



# Create Virtual Device(4/4) : **verify configuration**



# Launch the Virtual Device



Open Device File Explorer for this device



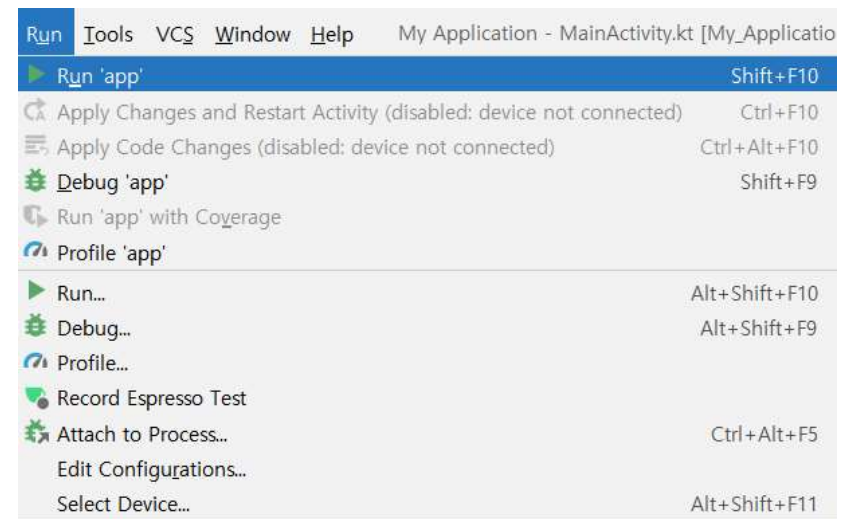
Launch this AVD in the emulator

Edit this AVD

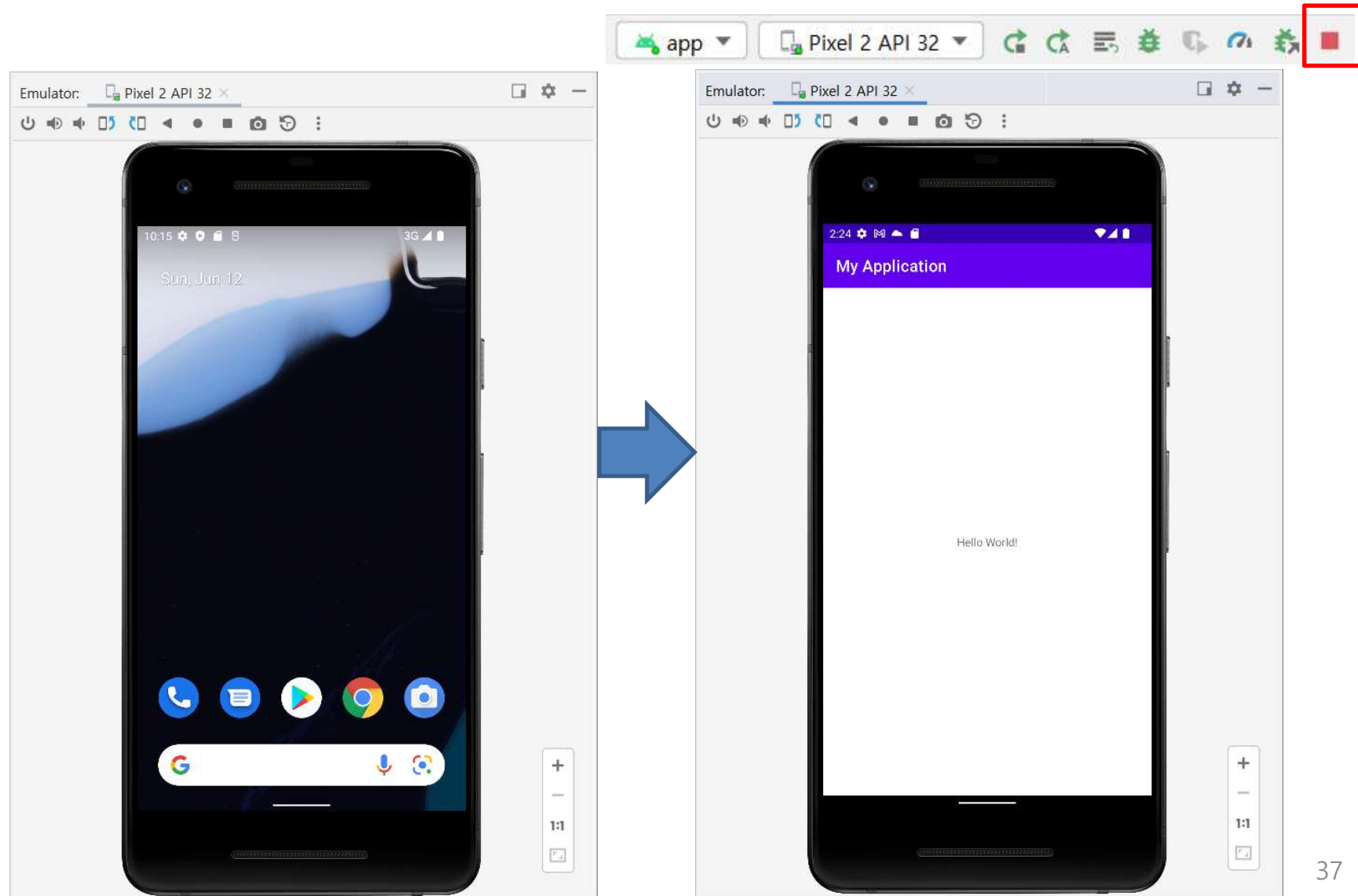
Duplicate  
Wipe Data  
Cold Boot Now  
Show on Disk  
View details  
Delete



# Run the App. (1/2)



# Run the App. (2/2)

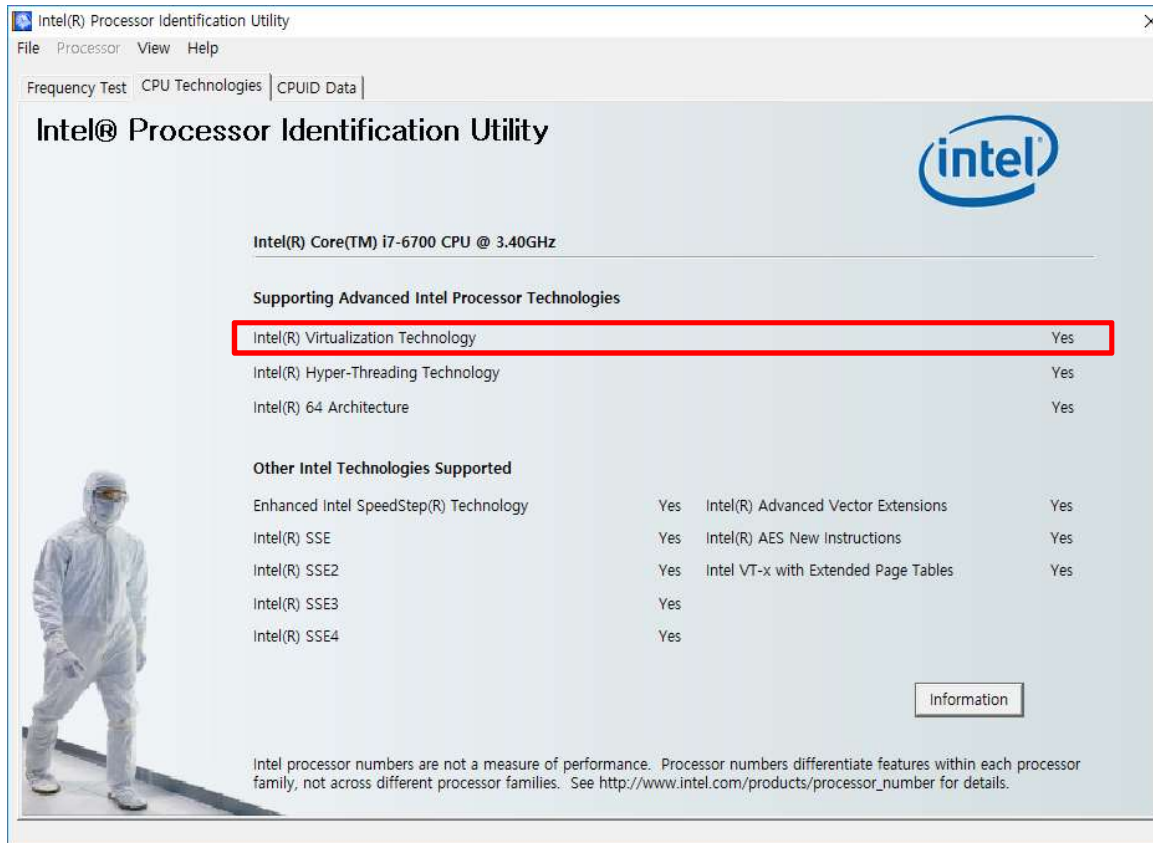


# 잠깐! ABI

- 초기 개인용 PC는 Apple Mac을 제외하면 OS나 컴파일러 없이 출시
  - IBM사는 IBM-PC compatible과 같이 하드웨어 표준만 제시
    - 하드웨어 확장 및 소프트웨어 표준은 시장 경쟁에 맡김
  - Microsoft가 Windows OS와 Visual C++ 등으로 시장 지배
  - Borland, Novell 외에 gcc와 같은 open source project 도 등장
- Android 에서도 제조사에 따라 단말에 장착되는 CPU (**ARM, AMD, x86**)가 다름
  - **기계어 코드 간 호환성을 위해 낮은 수준의** 인터페이스가 필요

- **ABI : Application Binary Interface**
  - 다양한 종류의 CPU와 android 시스템 간 인터페이스 정의
- **EABI : Embedded ABI**
  - ARM 계열 CPU가 대표적
- **AMD : Advanced Micro Devices**
  - PC용 CPU 시장에서 Intel사의 강력한 라이벌
- **x86과 x86\_64**
  - **x86** : Intel 계열 CPU이름에서 유래(8086, 80186, ... )
  - **x64** : AMD에서 x86을 64 bits로 확장하고 붙인 이름
    - 32bits CPU와 64bits CPU를 구분할 목적으로 사용
  - **x86**은 Intel 계열 CPU를, **x86\_64**는 AMD 계열 CPU를 가리킴

# 잠깐! 가상화 기술



- AVD로 **x86**을 선택했을 때는 **virtualization technology**를 **enable**시켜야 함.
  - BIOS에서 설정
- **가상 머신**(Virtual Machine, **VM**)
  - CPU 처리 과정을
  - 소프트웨어로 구현
- **x86 가상화** (x86 virtualization)
  - **CPU 가상화**
  - VM에서 실행되는 소프트웨어는 마치 전용 CPU에서 동작하는 것처럼
  - 어떠한 성능 및 호환성 문제 없이 동작

# What to do next?

- MyApplication 프로젝트 만들고 실행
  - Android studio 화면 이해
  - AVD를 사용한 App. 실행
- 용어
  - AVD
    - emulator, ABI, 가상화 기술
- **Wrap-up**



# Separation of layout and code

Layout

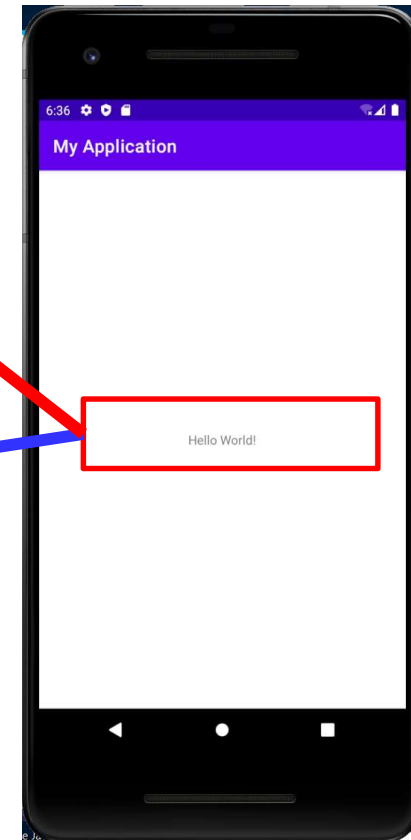
activity\_main.xml

```
<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" />
```

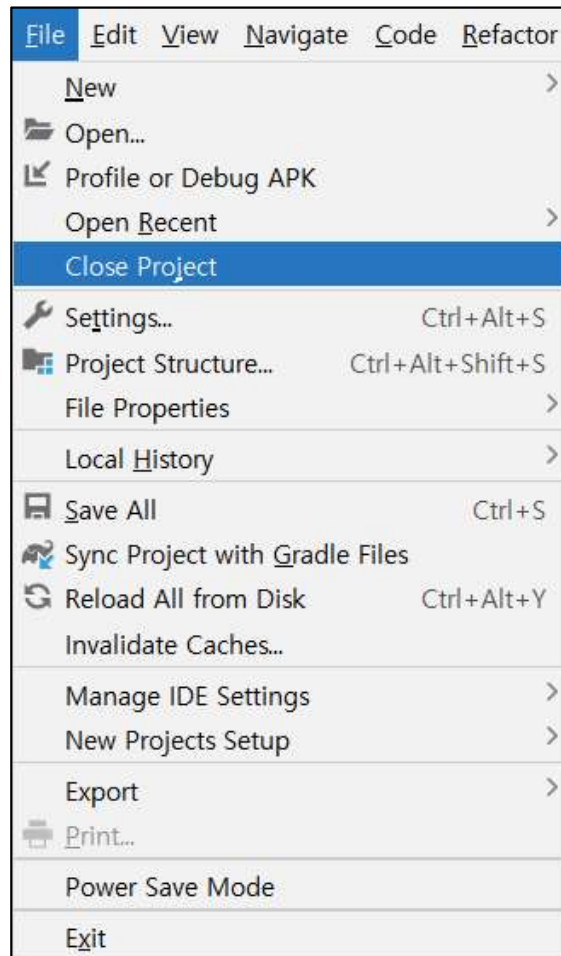
Code

MainActivity

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```



# Close the project and check the files



내 PC > 로컬 디스크 (C:) > 사용자 > yshon > AndroidStudioProjects > MyApplication

이름	수정한 날짜	유형
.gradle	2022-06-09 오후 10:27	파일 폴더
.idea	2022-06-10 오후 10:34	파일 폴더
app	2022-06-09 오후 10:27	파일 폴더
gradle	2022-06-09 오후 10:27	파일 폴더
.gitignore	2022-06-09 오후 10:27	GITIGNORE 파일
build.gradle	2022-06-09 오후 10:27	GRADLE 파일
gradle.properties	2022-06-09 오후 10:27	PROPERTIES 파일
gradlew	2022-06-09 오후 10:27	파일
gradlew	2022-06-09 오후 10:27	Windows 배치 파일
local.properties	2022-06-09 오후 10:27	PROPERTIES 파일
settings.gradle	2022-06-09 오후 10:27	GRADLE 파일