

---

# **Application Development for Mobile Computer**

**<Week 2>**

---

**Youn Kyu Lee**

# Introduction to Android

## Android

- Android is based on Linux, an open-source operating system.
- Android apps are developed using Java or Kotlin.
- Most of the core parts of the Android OS, libraries, and some Google apps are open-source.
- Android apps can be distributed not only through Google Play Store but also in various other ways.

Android 

# Introduction to Android

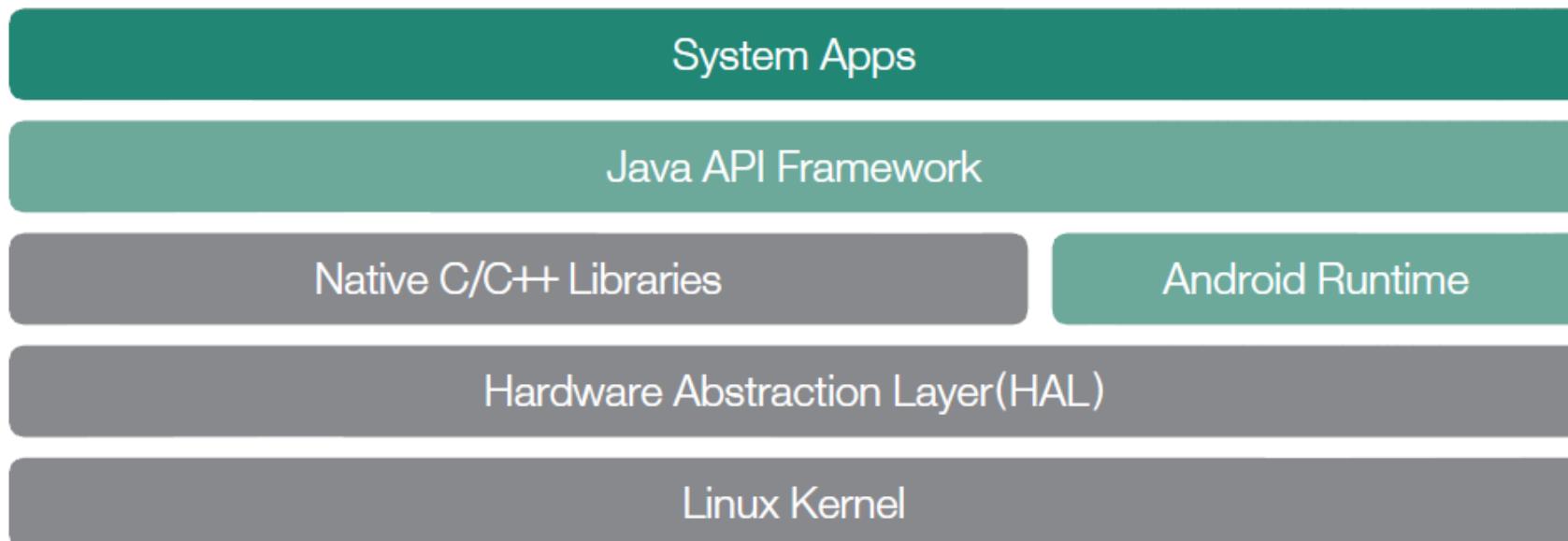
## Android OS

- The structure of the Android platform combines the Linux kernel and the Java API framework.
- In the past, explanations of the Android platform always mentioned the JVM (actually Dalvik VM, but usually written as JVM) because apps ran on the Java Virtual Machine.
- Now, **Android Runtime (ART)** is used instead of the JVM.
- Therefore, although the JVM is no longer present, apps still run on a virtual machine, adopting the execution model of Java.

# Introduction to Android

## Android OS

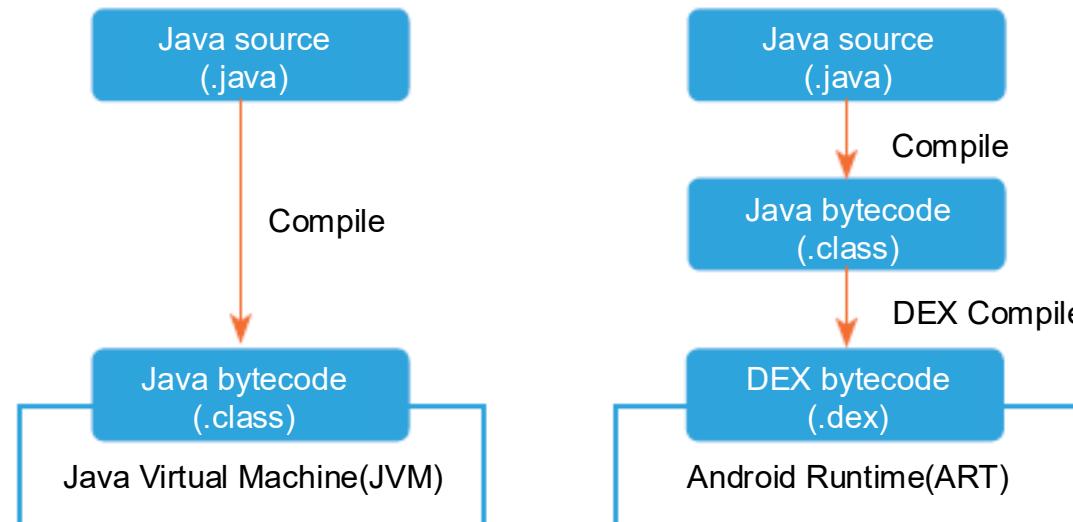
- Linux kernel: Android is an open-source software stack based on Linux.
- **Android Runtime (ART)**: Responsible for running applications.
- Java API Framework: Provides Java APIs used for app development.



# Introduction to Android

## Android

- Android does not run Java classes directly at runtime but compiles them into DEX files.
- These DEX files are executed in the Android Runtime (ART), which interprets them

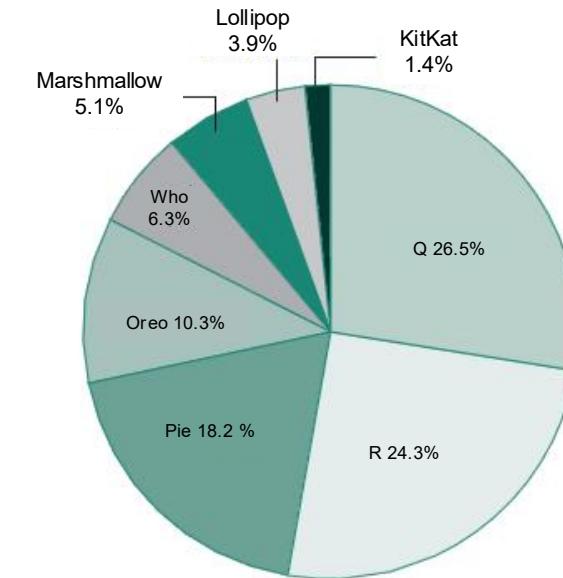


# Android Versions

## Android Versions

- Android has an API level that exists separately from the platform version.
- Depending on the API level, new features may be added, or major code changes may be required.

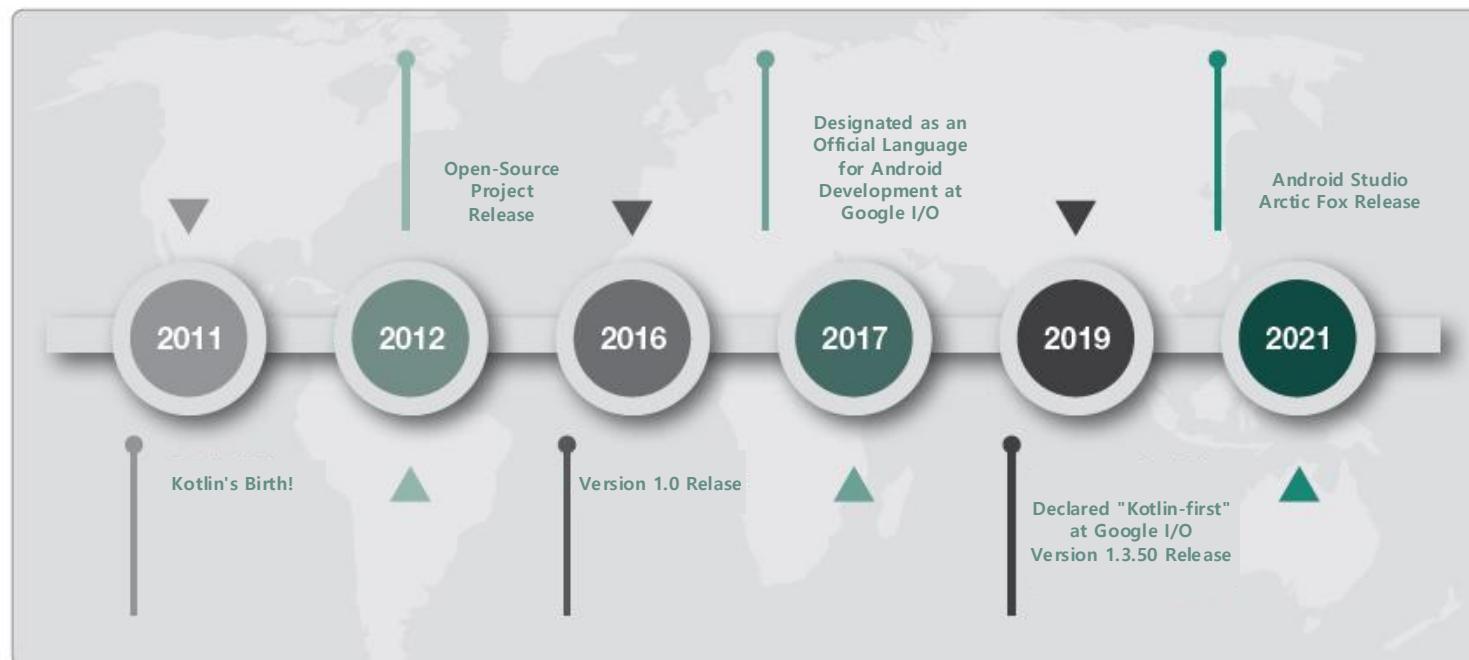
Platform Version	Code Name	API Level	Version Code
12.0	Android 12	31	S
11.0	Android 11	30	R
10.0	Android 10	29	Q
9.0	Pie	28	P
8.1	Oreo	27	O_MR1
8.0	Oreo	26	O
7.1.1	Nougat	25	N_MR1
7.0	Nougat	24	N
6.0	Marshmallow	23	M



# Kotlin

## Kotlin

- In the past, Java was mainly used as the primary app development language.
- At Google I/O in May 2017, Kotlin was announced as an official language, and since then more companies have gradually adopted it for app development.



# Kotlin

## Kotlin

- Kotlin supports functional programming.
- In object-oriented programming, logic is written only in functions inside a class.
- In functional programming, logic can be written anywhere without such restrictions.

### Java code - Object-Oriented Programming

```
class Hello {  
    public static void main(String args[]) {  
        System.out.print("Hello World");  
    }  
}
```

### Kotlin code - Functional Programming

```
System.out.print("Hello World");
```

# Kotlin

## Kotlin

- When developing Android apps, Kotlin requires writing logic inside classes.
- Kotlin is 100% compatible with Java.
- Code written in Kotlin is usually much shorter compared to Java.

### Java code - ButtonActivity.java

---

```
CheckBox check = findViewById(R.id.check);
ImageView image = findViewById(R.id.image);
check.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if (isChecked) {
            image.setVisibility(View.VISIBLE);
        } else {
            image.setVisibility(View.GONE);
        }
    }
});
```

### Kotlin code - ButtonActivity.kt

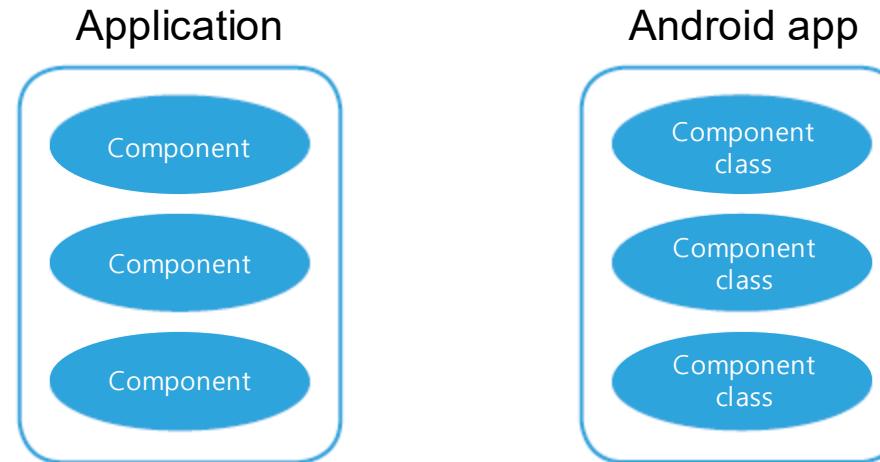
---

```
binding.check.setOnCheckedChangeListener { buttonView, isChecked ->
    binding.image.visibility = if(isChecked).View.VISIBLE else View.GONE
}
```

# Android App Development

## Android Components

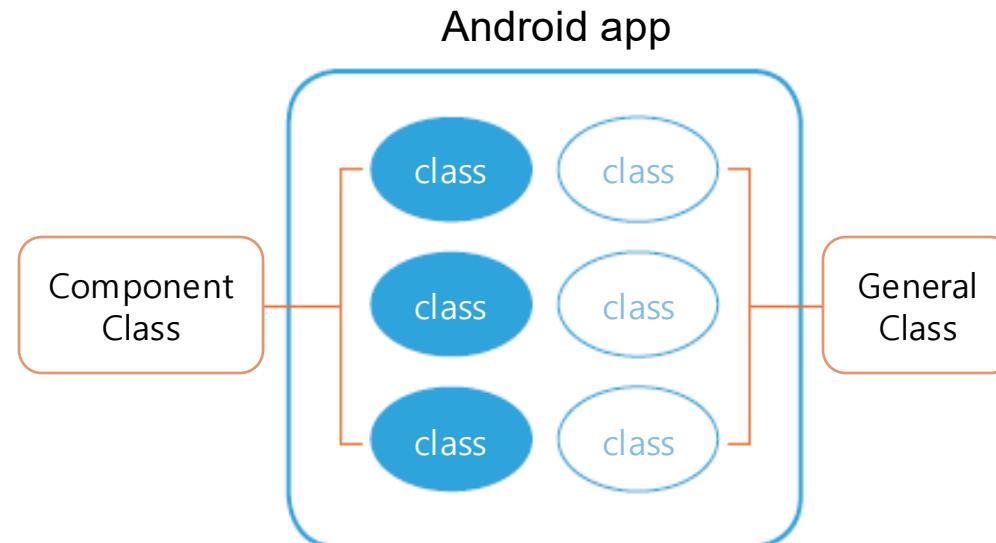
- A component is a building block of an application.
- A component is an independent unit of execution within an app.



# Android App Development

## Android Components

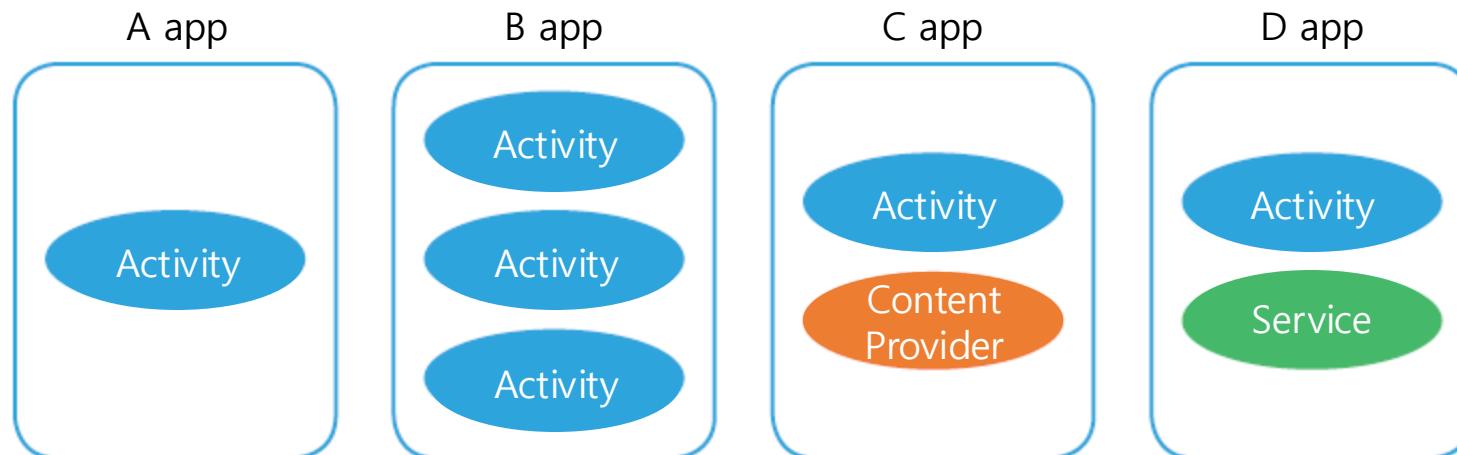
- An app is made up of many classes, mainly divided into component classes and regular classes.
- If the lifecycle (from object creation to destruction) is managed by developer code, it is a regular class.
- If the lifecycle is managed by the Android system, it is a component class.



# Android App Development

## Android Components

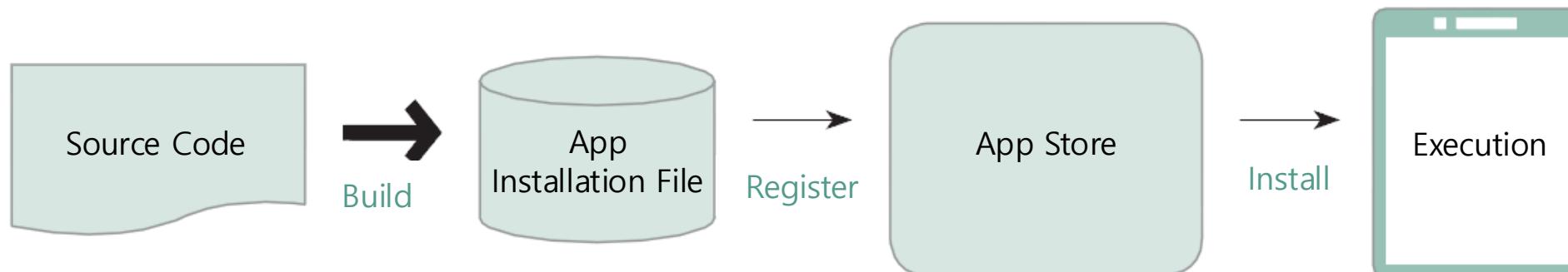
- Activity: A component that provides the user interface (screen).
- Service: A component that performs background tasks.
- Content Provider: A component that shares app data.
- Broadcast Receiver: A component that responds to system events.



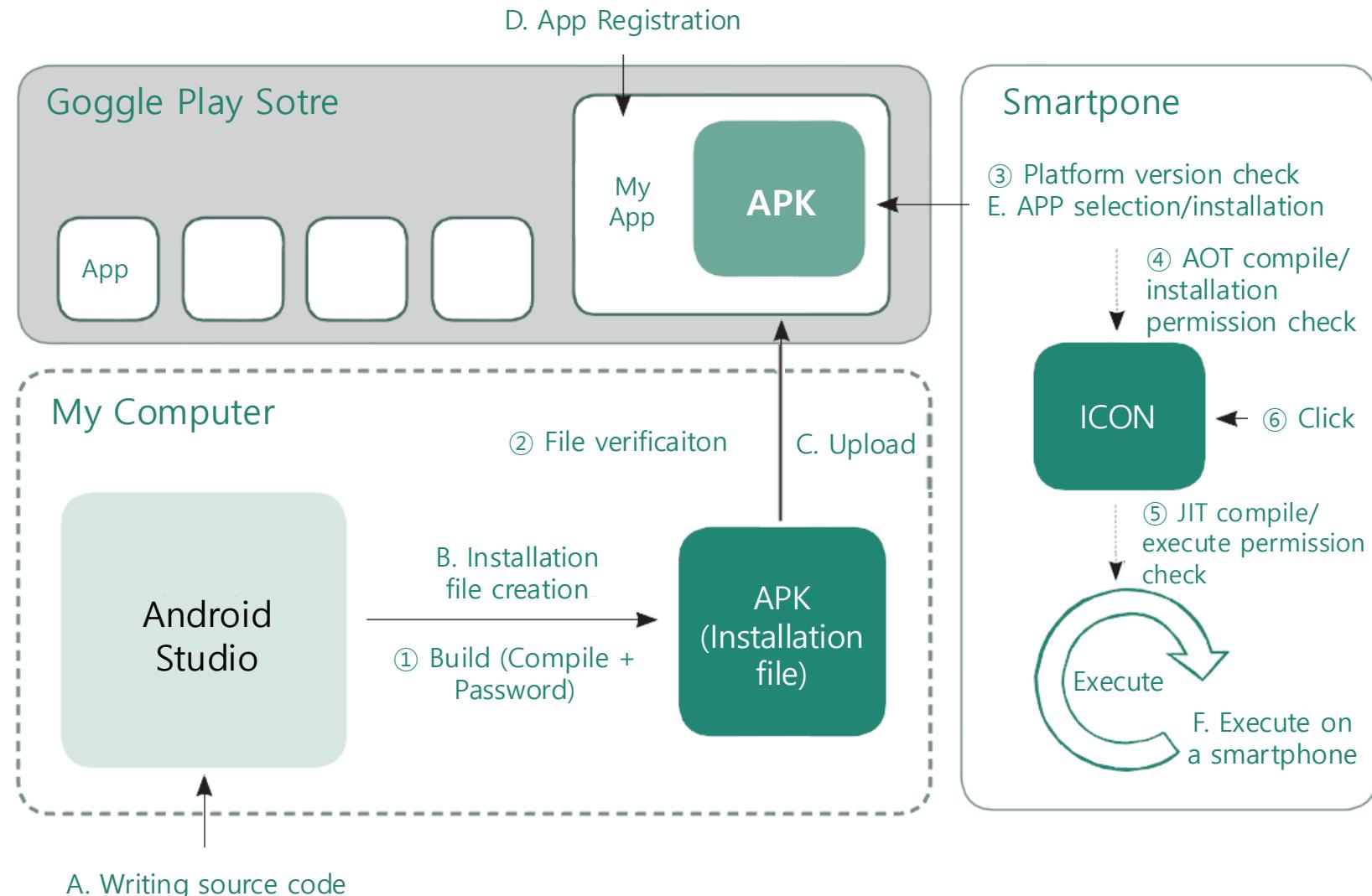
# Android App Development

## Development Process

- Write the source code.
- Use build commands to convert it into an installable Android package.
- Upload the app to the Google Play Store.
- Register the app in the Google Play Console.
- On a smartphone, open the Play Store, select the app, and install it.
- Tap the app icon to launch it.



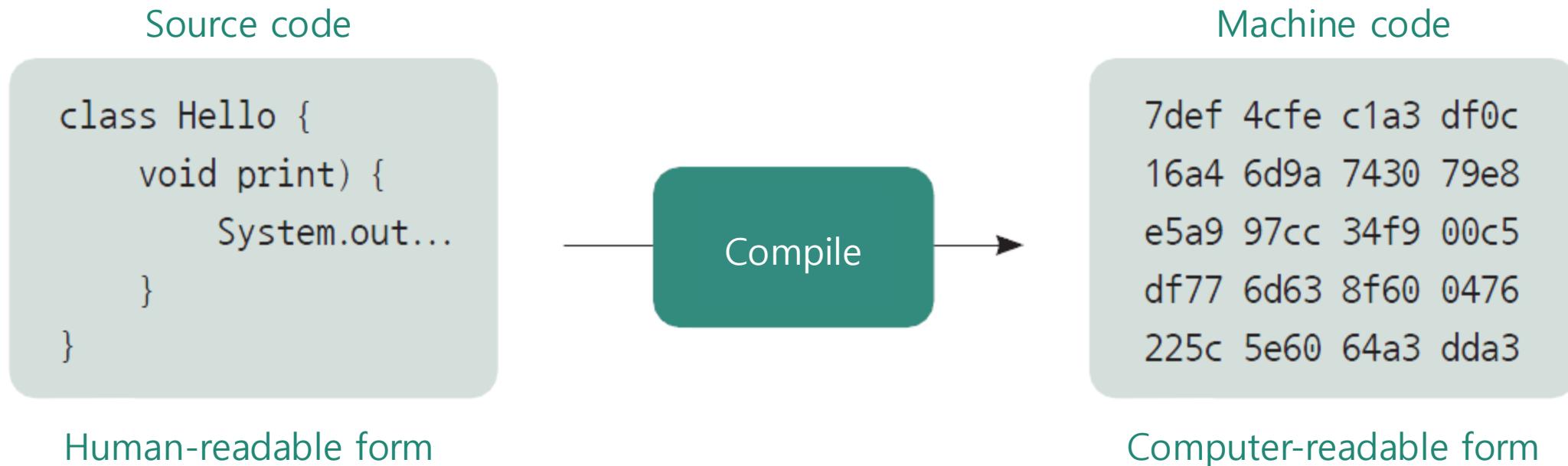
# Android App Development



# Android App Development

## Compile

- Converting human-readable source code into machine code that a computer can understand.



# Android App Development

## Build in Linux

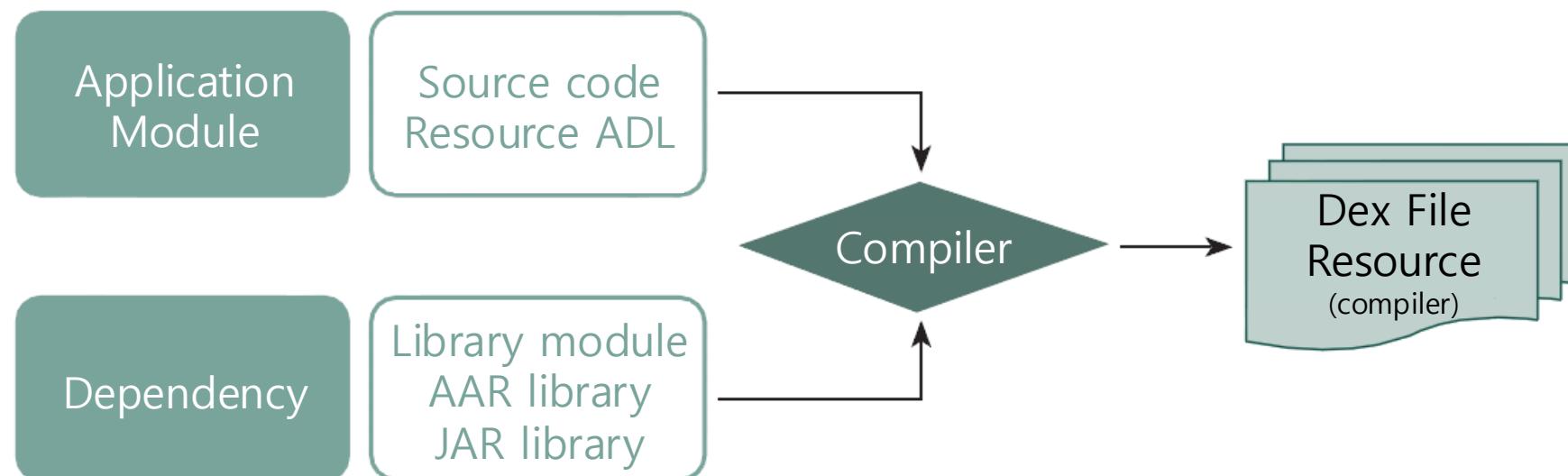
- Translating (compiling) the source code into machine code and linking it with the libraries used, to produce the final executable file.



# Android App Development

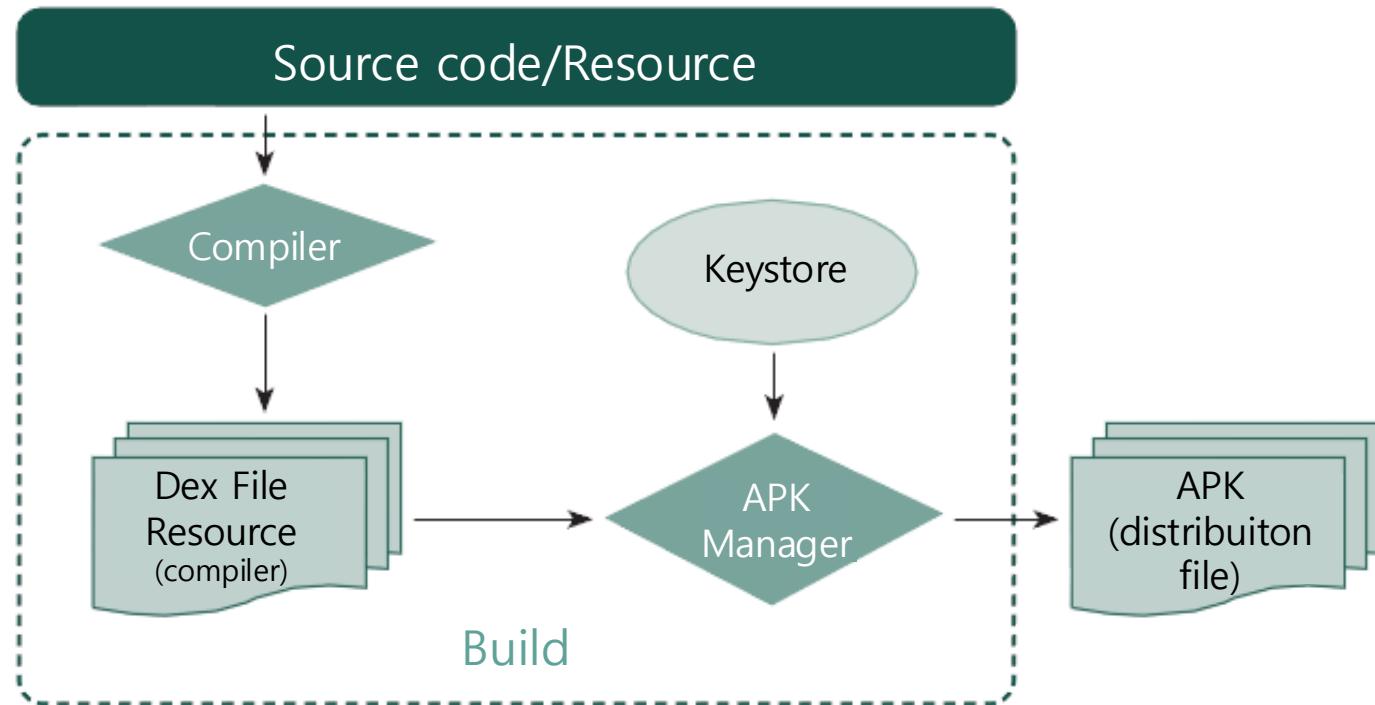
## Build in Android

- The difference from Linux compilation is that Android has the concept of resources.
- Android compilation happens in two steps:
  - Step 1: Bytecode generation
  - Step 2: APK file creation



# Android App Development

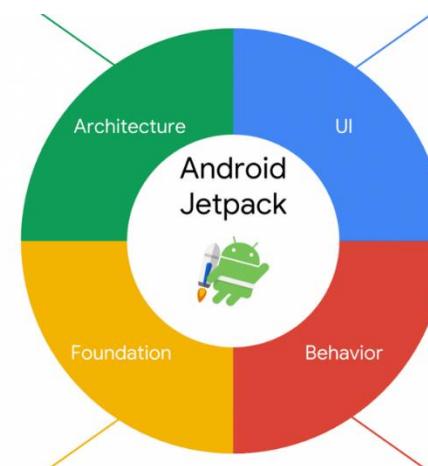
## Build in Android



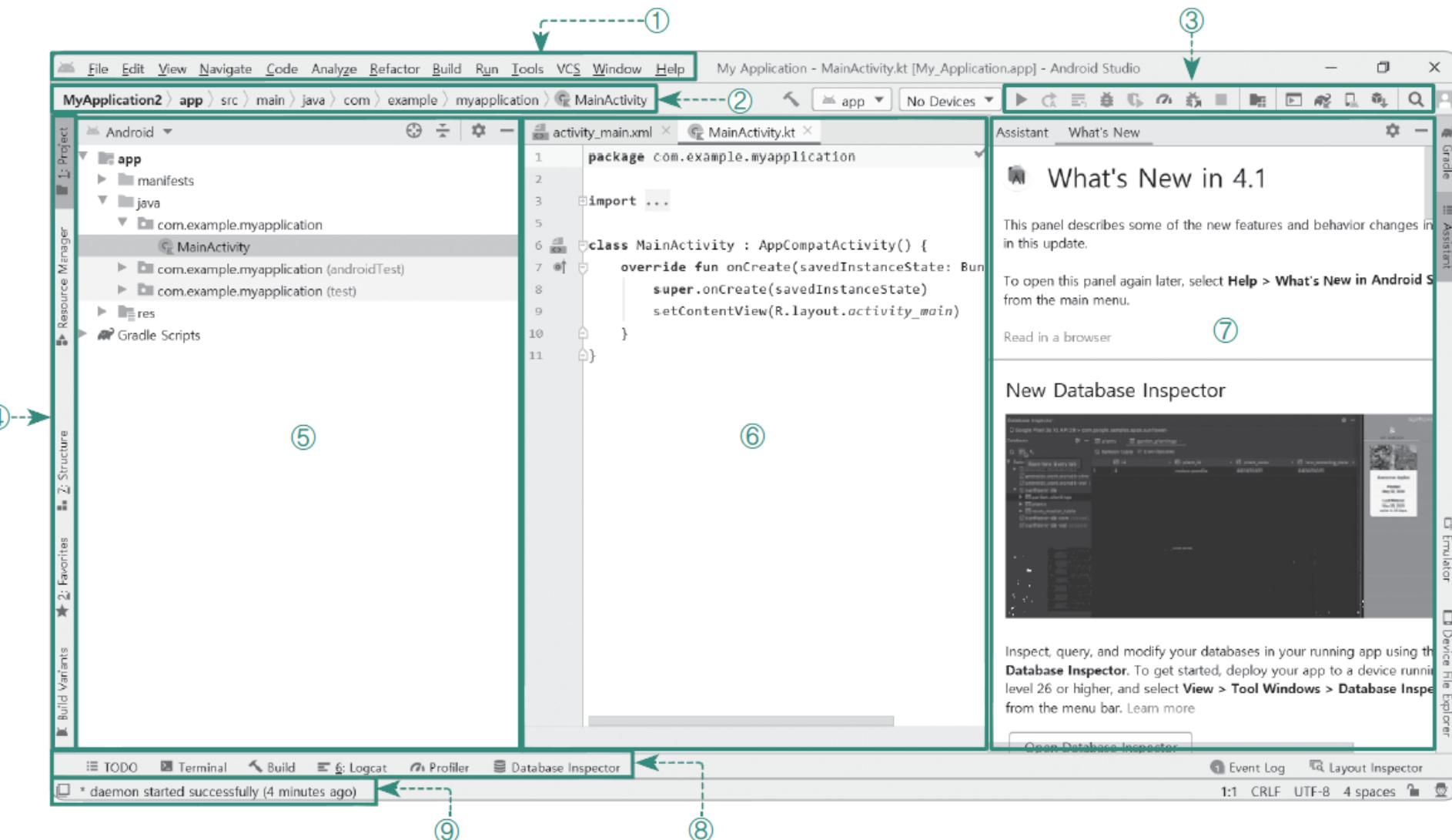
# Android App Development

## Jetpack

- Consistent libraries: Work uniformly across all versions and devices.
- Backward compatibility: Jetpack libraries, built on modern design principles, include compatibility features for older versions, reducing crashes and memory leaks.
- Reduced boilerplate code: Jetpack removes repetitive tasks like background work and lifecycle management, allowing developers to focus on business logic.
- Lower complexity: Provides consistent behavior across versions and devices, simplifying code.

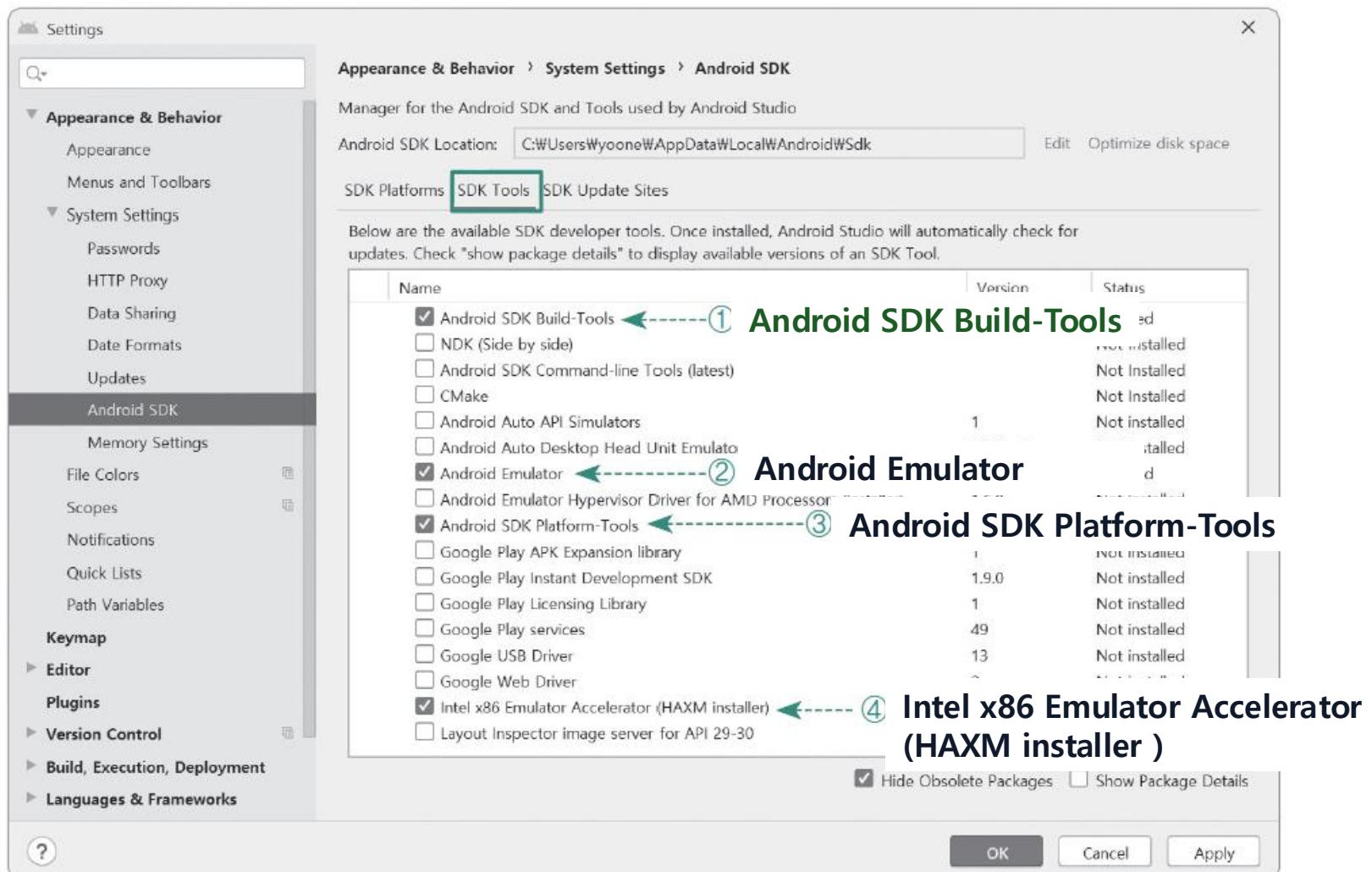


# Android Studio



# Android Studio

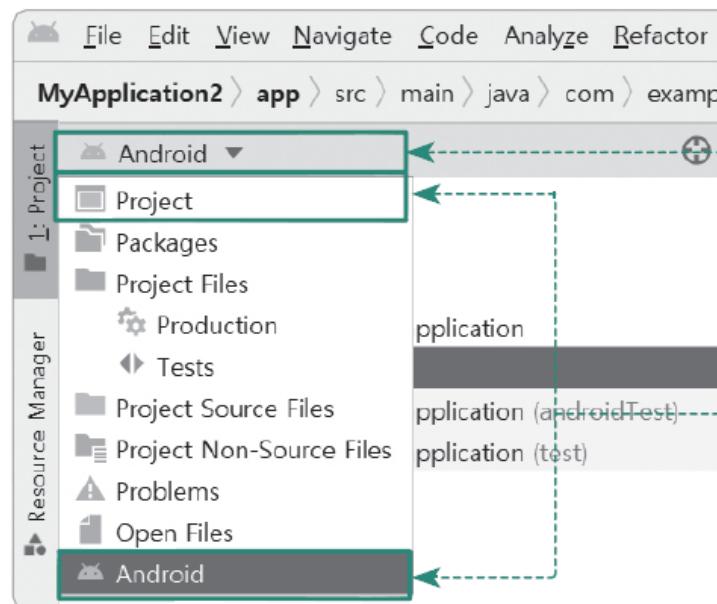
## Android SDK



# Android Studio

## Android Project

- The top-level directory that systematically manages source code, images, music, text files, and other resources.

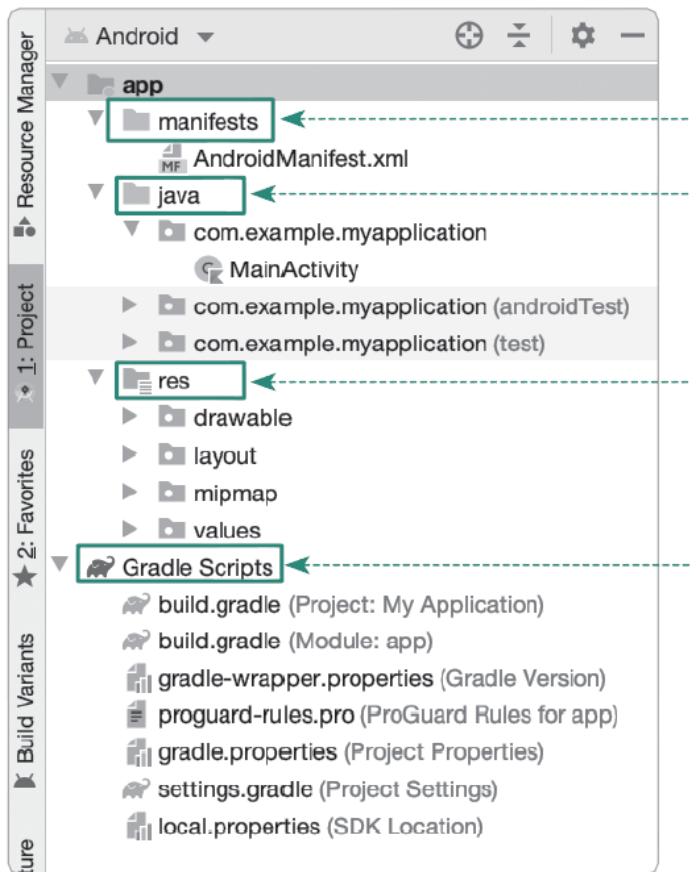


When clicked, the list of views expands.

Mainly uses Project and Android views.

# Android Studio

## Android View



Directory for installation-related information

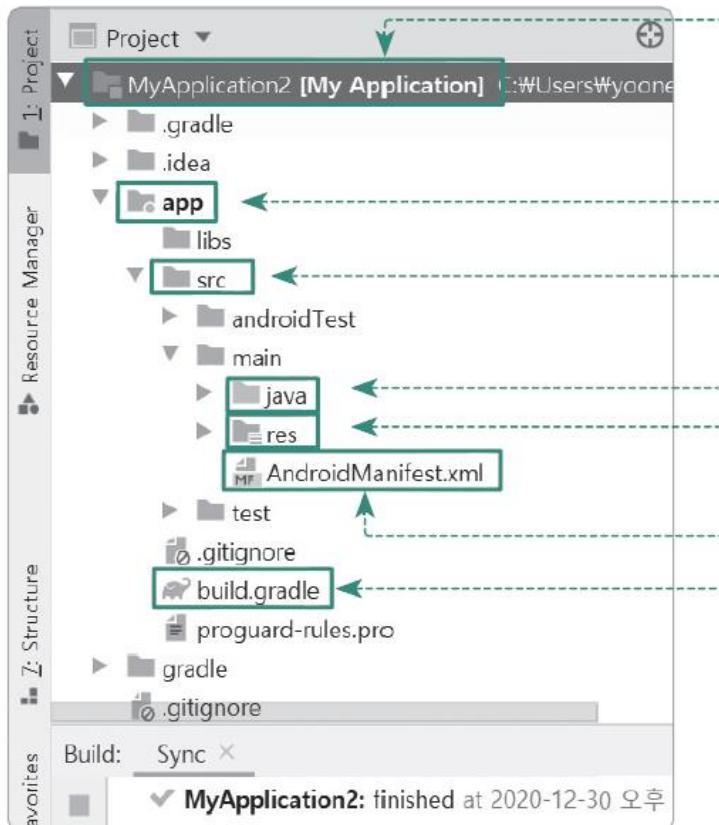
Directory for source code

Directory for resources such as images, layouts, and MP3 files

Directory for build-related configuration information

# Android Studio

## Project View



Directory for the actual project

Directory containing all app-related sources (code, libraries, images, etc.)

Directory for user-created sources (code, layouts, images, etc.)

Directory containing source code (still shown as java)

Directory for resources such as images, layouts, and MP3 files

Installation information file

Build information file

# Android Studio

## Emulator



# Q & A

---

aiclasscau@gmail.com