

소프트웨어설계및실험

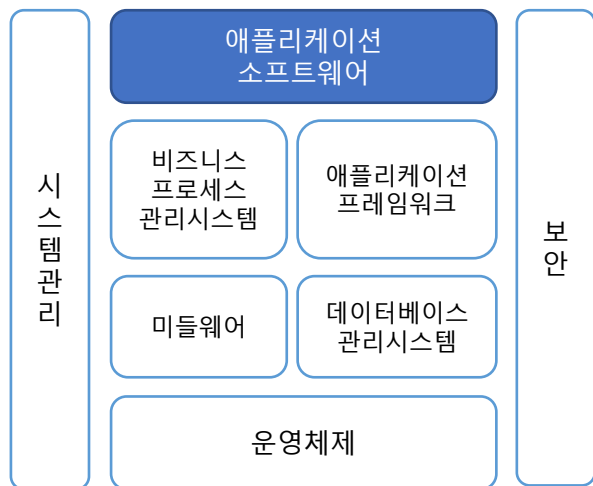
2024 Spring

안드로이드 프로그래밍 개요:
안드로이드 프로그래밍, 안드로이드 스튜디오 설치

이론

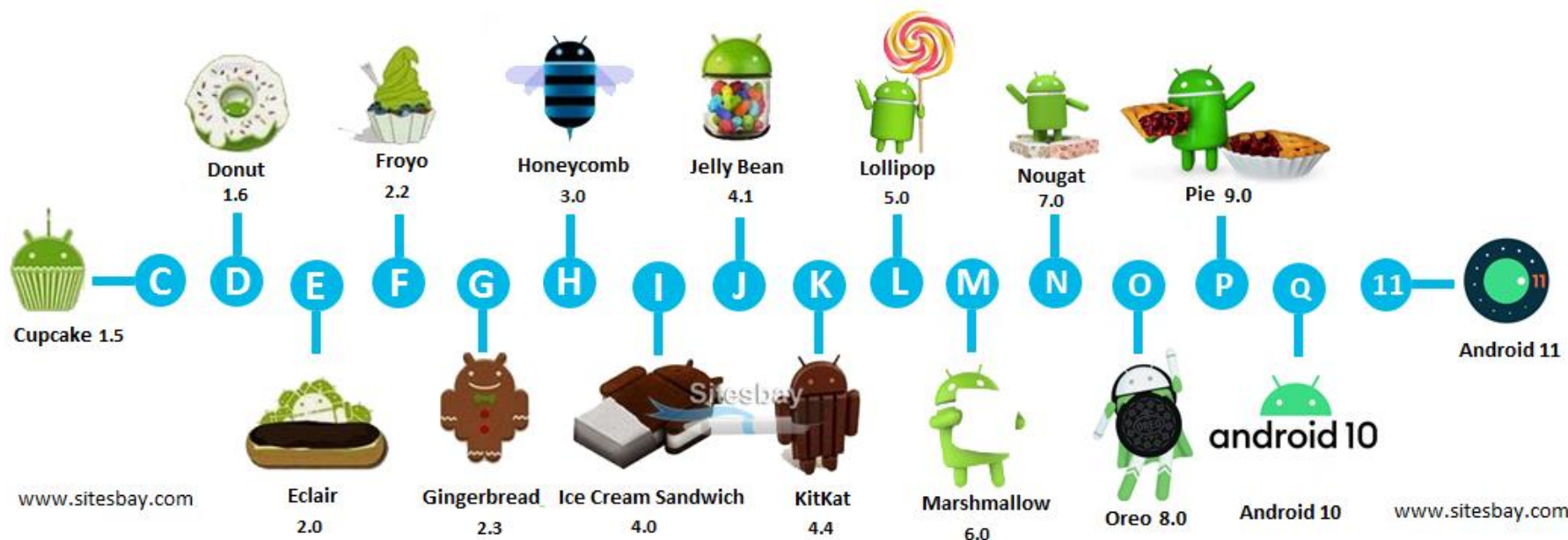
• Android 란?

- Linux Kernel을 기반으로 제작한 모바일 운영체제
- 2008년 구글에서 Android 1.0 버전 출시
- Software Stack 방식
- 전 세계 모바일 플랫폼 시장의 70~80%를 차지



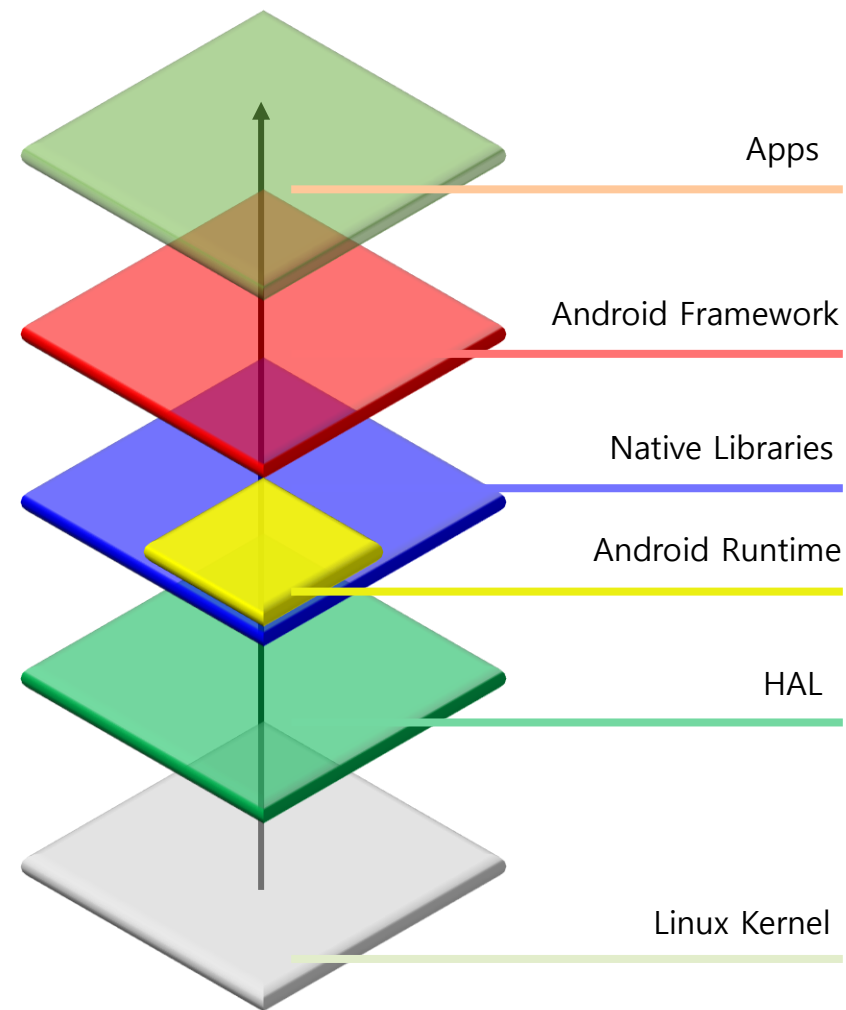
❖ Android의 특징

- Java나 Kotlin 언어를 이용해 개발
- 개발된 Android 앱은 다양한 방법으로 사용자에게 배포 가능
- Open Source Software Stack (Platform)
- 소스 코드를 아파치 v2 license로 배포하여 기업이나 사용자는 Android 프로그램을 독자적으로 개발하여 탑재 가능



• Android의 운영체제의 구조

- Linux Kernel
 - Android 플랫폼의 기반으로 **운영체제의 핵심 기능**을 수행
 - 하드웨어와 소프트웨어 사이에서 **하드웨어 제어 관리**
 - 다양한 하드웨어 드라이버 제공
- HAL (Hardware Abstraction Layer)
 - 같은 종류의 하드웨어에 대한 공통 명령어 집합
 - Android 프레임워크에서 **하드웨어 기능을 이용할 수 있게 표준 인터페이스 제공**
- Native Libraries
 - Java Framework 외의 C/C++ 라이브러리 제공
 - Webkit, OpenMAX AL, Media Framework, ...
- Android Runtime (ART)
 - **앱에 대한 전체 실행을 주관**
 - Android 버전 5.0 이전에는 Dalvik이 ART 역할 수행
- Android Framework
 - Java로 구현된 API Framework
 - 모든 Android 앱이 사용하는 Toolkit
- Apps
 - Android 운영체제에 기본적으로 설치되어 있는 앱
 - 사용자가 설치한 앱



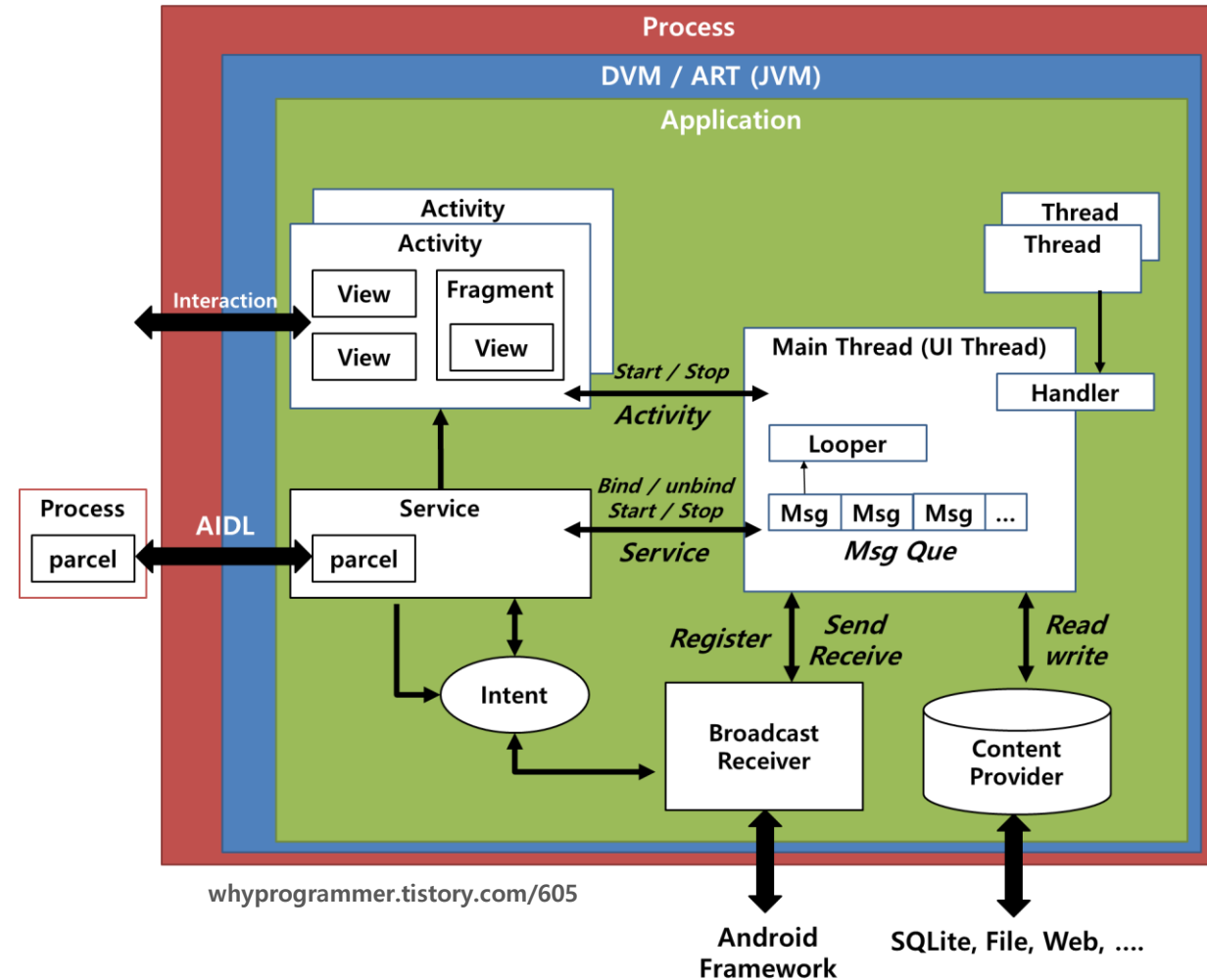
Android 앱의 구성 요소

• Android 앱의 구성 요소

- 애플리케이션은 **일반 클래스**와 **컴포넌트 클래스**로 구성
 - **일반 클래스**는 개발자가 정의한 클래스
 - **컴포넌트 클래스**는 Android 시스템의 컴포넌트 클래스를 상속받아 생성하는 클래스

• Android의 4대 컴포넌트

- **Activity**
 - **화면을 구성**하는 컴포넌트, 주로 UI(User Interface) 화면을 담당
- **Service**
 - 기기의 화면에 표시되지 않고 **백그라운드에서 실행되는 작업을 처리**
 - Ex.) 네트워크 데이터 송수신, 음원 재생 등
- **Content provider**
 - 앱 내에서 생성·관리하는 **데이터를 다른 앱에 제공**하는 기능 표준화
 - Ex.) 카카오톡 앱 프로필 변경 시 갤러리 앱의 사진 사용
- **Broadcast receiver**
 - **시스템 이벤트**가 발생할 때 **실행**되게 하는 컴포넌트
 - Ex.) 부팅 완료, 배터리 방전, 앱의 팝업



- 컴포넌트의 실행 특징

- 앱 내부 독립된 실행 단위

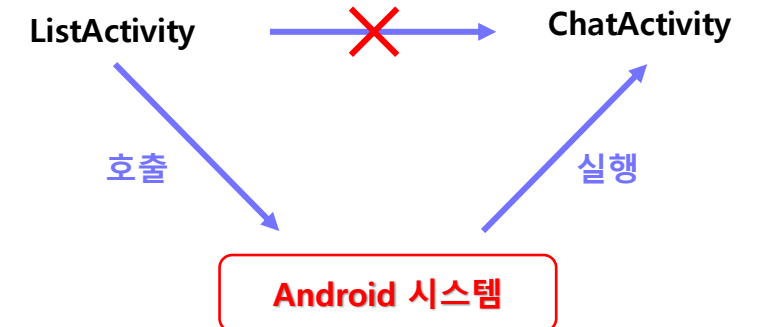
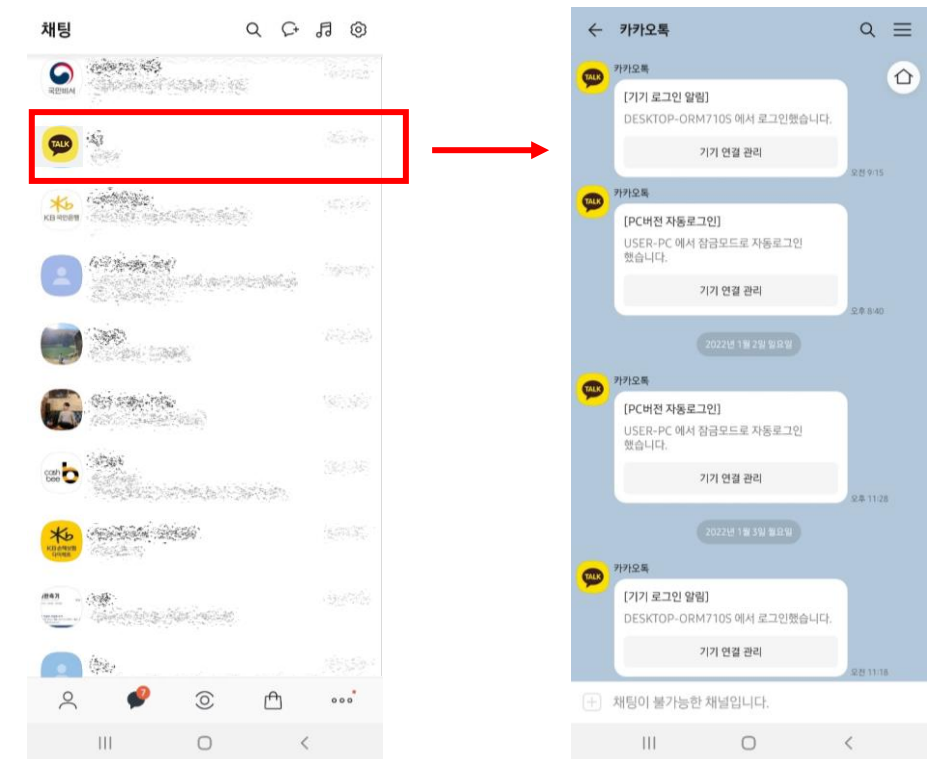
- 컴포넌트는 **독립적인 실행 단위**로 코드 결합이 발생하지 않는다.
- 컴포넌트 내에서 새로운 컴포넌트 실행 시 **Android 시스템을 호출하여 실행**한다.
 - Ex.) 카카오톡의 목록 화면(ListActivity)과 채팅 화면(ChatActivity)

- 애플리케이션 실행 시점의 다양성

- 메시지 수신 알림을 통해 목록화면 (ListActivity) 을 실행하지 않고 채팅화면 (ChatActivity) 을 바로 실행 가능

- 애플리케이션 라이브러리 사용 가능

- 애플리케이션 라이브러리**란? 현재 실행 중인 애플리케이션에서 다른 애플리케이션을 라이브러리처럼 이용하는 것
 - Ex.) 카카오톡 앱에서 카메라 앱을 이용



• Android 리소스 란?

- 코드에서 사용되는 **추가 파일**과 **정적인 콘텐츠**
- 독립적인 유지관리를 위해서 **코드에서 외부화로 분리한 데이터**

문자열을 리소스로 등록하기

```
<string name="mytxt">
```

앱에서 발생하는 데이터나 사용자 이벤트에 따른 값이 항상 똑같은 값이라면
코드에 담지 않고 리소스로 분리해서 개발
문자열 이외에 색상, 크기, 레이아웃 등 많은 요소를 리소스로 활용 가능
대부분의 리소스는 **XML** 파일로 작성

```
</string>
```

문자열을 코드로 작성하기

```
textView.text = ""
```

앱에서 발생하는 데이터나 사용자 이벤트에 따른 값이 항상 똑같은 값이라면
코드에 담지 않고 리소스로 분리해서 개발
문자열 이외에 색상, 크기, 레이아웃 등 많은 요소를 리소스로 활용 가능
대부분의 리소스는 **XML** 파일로 작성

```
""
```



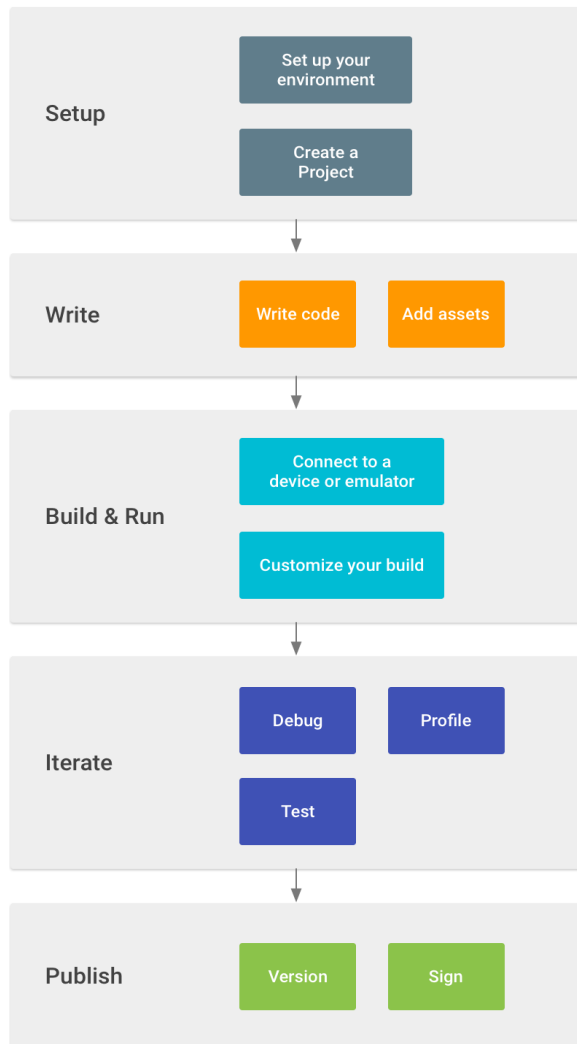
문자열 리소스 사용 예시

```
textView.text = resources.getString(R.string.mytxt)
```

• Android 리소스를 활용한 개발

- 앱에서 발생하는 데이터나 사용자 이벤트에 따른 결과 값이 **동일한 값일 경우 코드에 담지 않고 리소스로 등록하여 사용**한다.
- 문자열 이외에 **색상, 크기, 레이아웃, 이미지, 메뉴 등** 많은 요소를 리소스로 등록하여 활용할 수 있다.

❖ Android 개발자 워크플로우



❖ 작업공간 설정

- Android 스튜디오를 설치하고 프로젝트를 생성

❖ 앱 작성

- Android 스튜디오에서 제공하는 사용 가능한 도구와 기능을 활용하여 앱을 작성

❖ 빌드와 실행

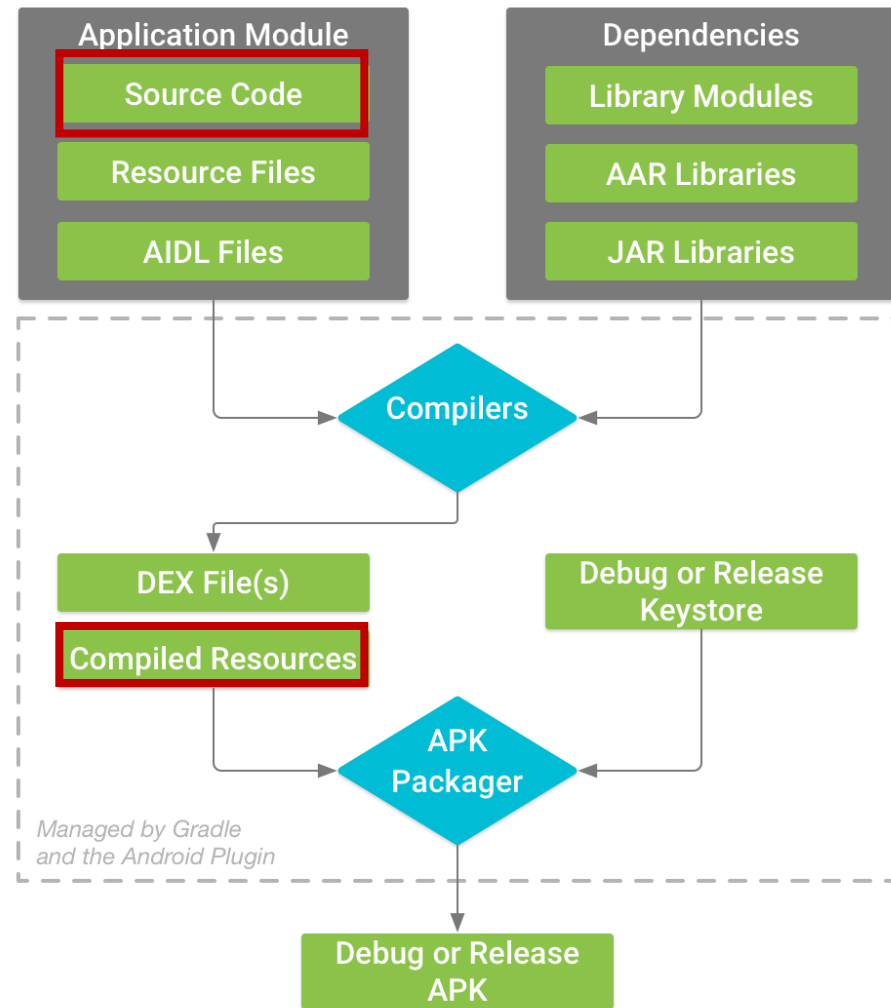
- 디버그 가능한 APK 패키지로 프로젝트를 빌드

❖ 디버깅, 프로파일링 및 테스트

- 버그를 제거하고 앱 성능을 최적화하는데 중점을 두고 앱을 계속 작성

❖ 게시

- 앱 버전 관리, Android App Bundle 빌드, 키 서명과 같은 몇 가지 사항을 고려하여 앱을 출시



• Android Studio란?

- Android 전용 애플리케이션 제작을 위한 공식 통합 개발환경(IDE)
- IntelliJ IDEA를 기반으로 제작
- Kotlin, Java, C++ 프로그래밍 언어 지원

android
studio



Kotlin



Java



❖ Android Studio의 강점 및 특징

Strengths	Characteristics
범용성	모든 Android 기기용으로 개발할 수 있는 통합 환경 제공
유연성	Gradle 빌드 시스템을 사용하여 유연한 프로그래밍 환경 제공
효율성	다중 APK 지원으로 효율적인 APK 생성
호환성	Windows, Linux, Mac 운영체제에서 사용 가능
편리성	인라인 디버깅 및 성능 분석 도구 제공
확장성	다양한 오픈 소스 라이브러리 사용 가능

• Android 프로젝트 파일

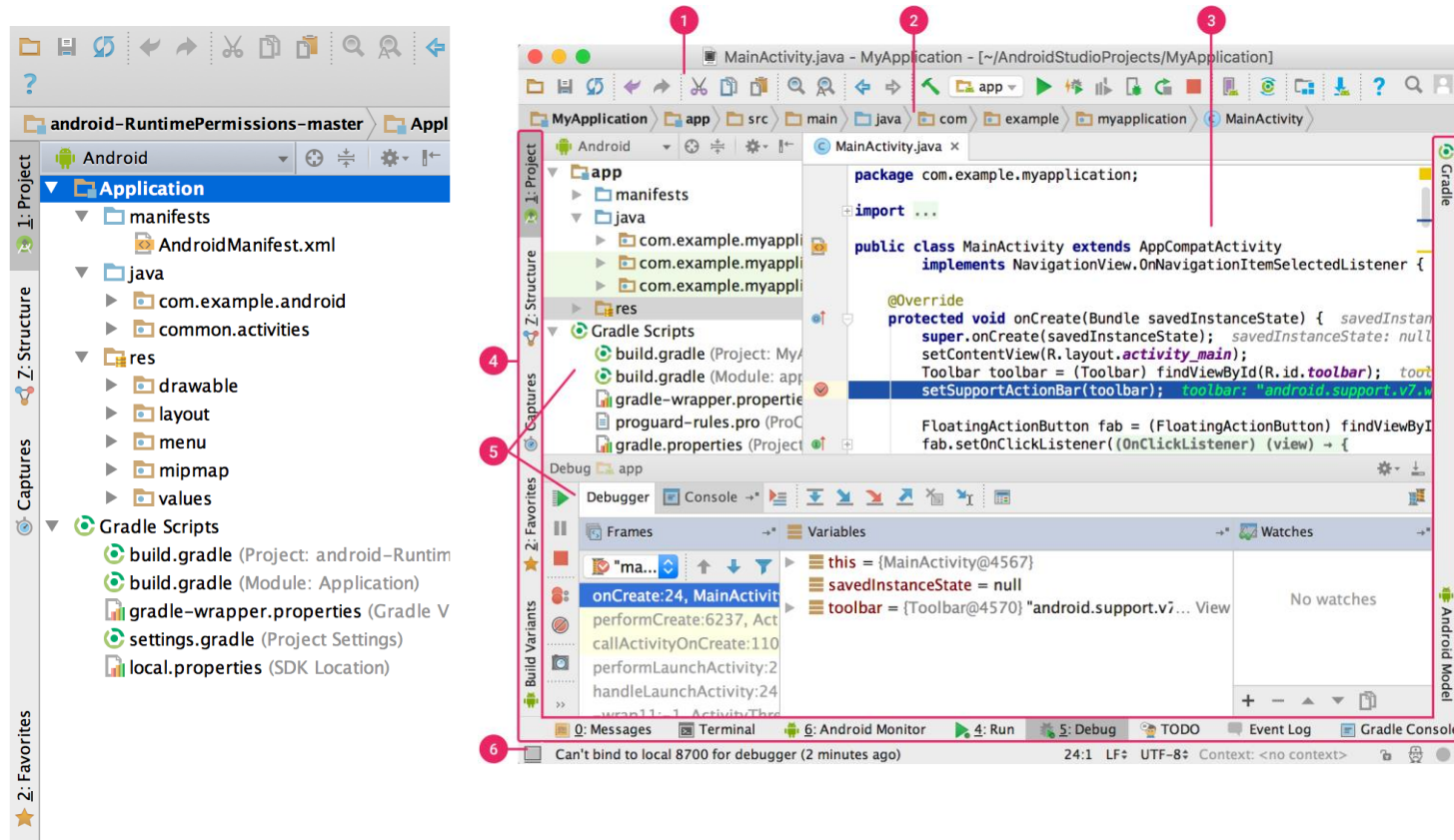
• Application

- **manifests**: AndroidManifest.xml 파일을 포함
 - Android 빌드 도구, Android 운영체제 및 Google Play에 **앱 구성에 관한 필수 정보**를 설명
- **java**: JUnit 테스트 코드를 비롯한 **자바 소스 코드 파일**을 포함
- **res**: 코드가 아닌 모든 리소스를 포함 (XML 레이아웃, UI 문자열, 비트맵 이미지 등).

• Gradle Scripts

- 프로젝트의 **빌드 정보, 라이브러리** 등을 관리하기 위한 **설정 파일**을 포함

❖ 사용자 인터페이스



실습

- ◆ Android Studio 설치

- ◆ 개발 환경 시험하기
 - ◆ 앱 생성
 - ◆ 앱 실행
 - ◆ 기초 조작 및 실행 실습
 - ◆ 스마트폰에서 앱 실행
 - ◆ Android App Bundles(AAB) 파일 생성

- ◆ 실습
 - ◆ 예제 1 - 학번 이름 출력하기
 - ◆ 예제 2 - GUI 화면 조작해보기

◆ 개발환경 구축하기

- ◆ <https://developer.android.com/studio/install?hl=ko>

Android 개발자 > 문서 > Android 스튜디오 > Android 스튜디오 편집기

Android 스튜디오 설치

클릭 몇 번으로 Android 스튜디오를 설정할 수 있습니다. 먼저 시스템 요구사항을 확인하세요. 그런 다음 [최신 버전의 Android 스튜디오를 다운로드](#)합니다.

이 페이지의 내용

[Windows](#)

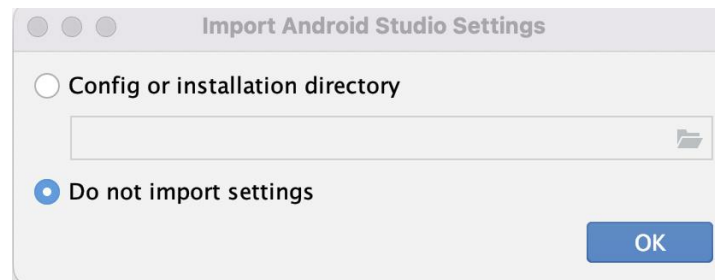
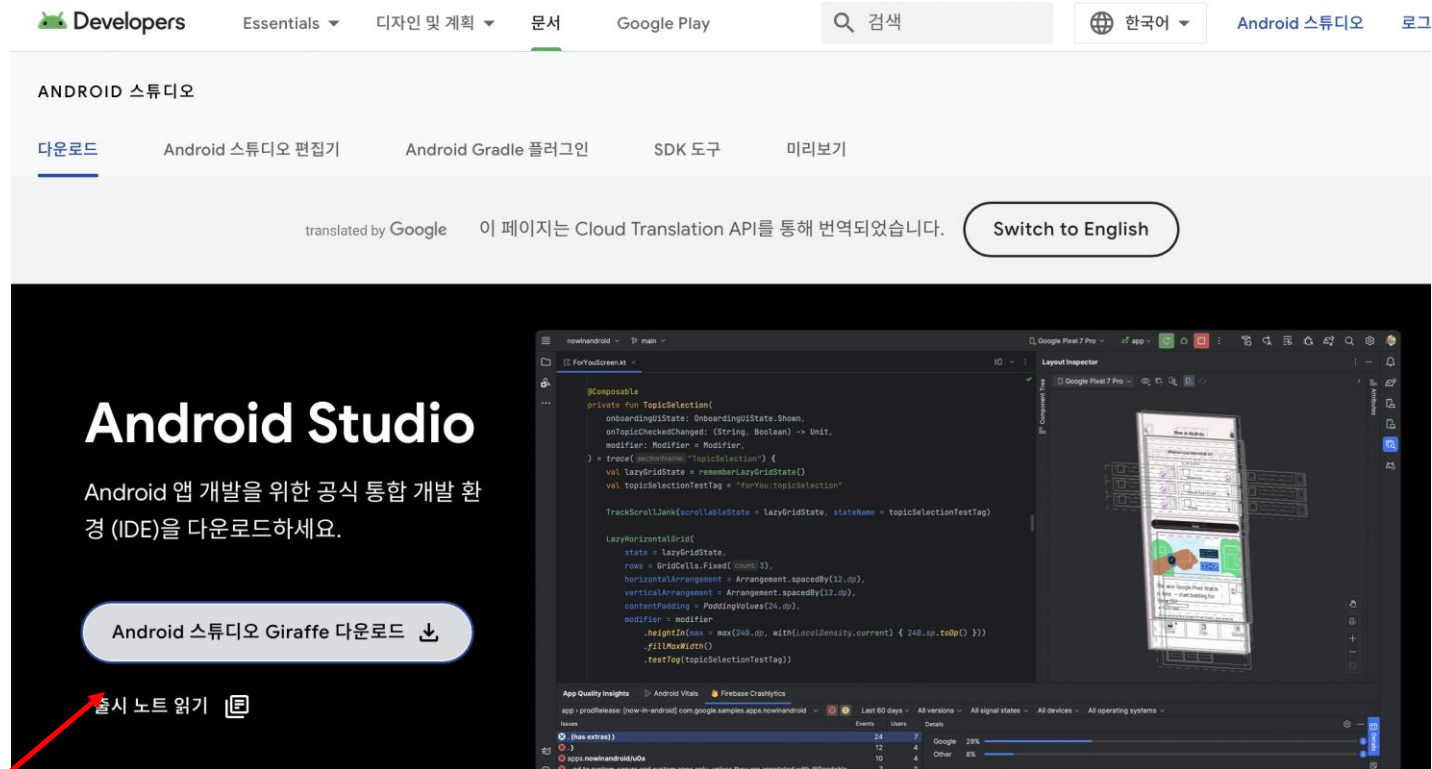
[Mac](#)

[Linux](#)

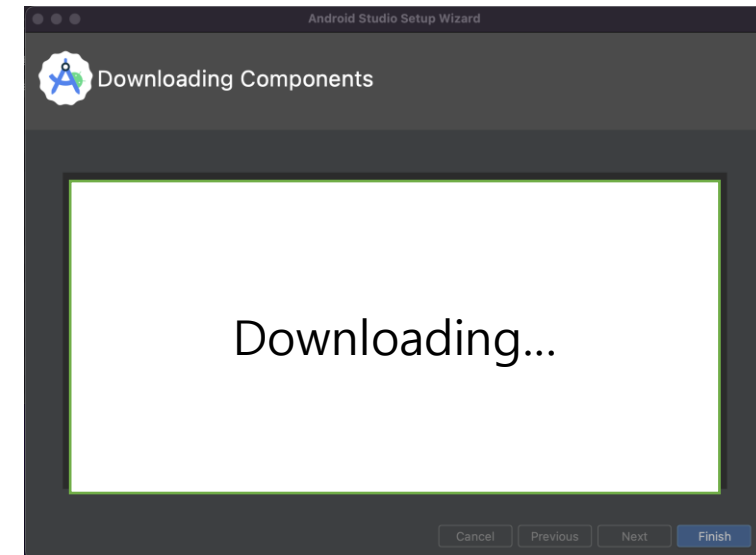
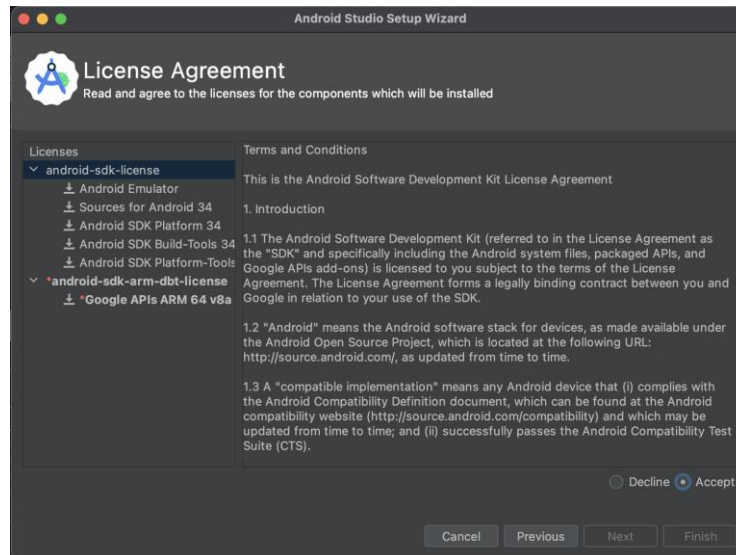
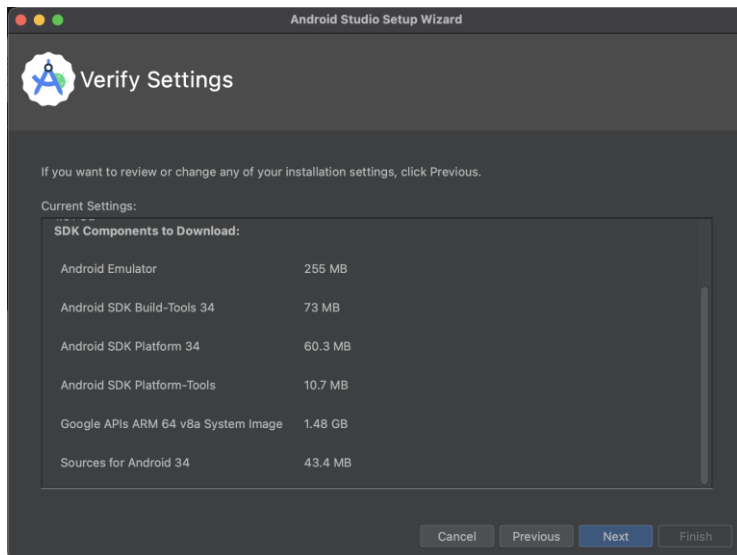
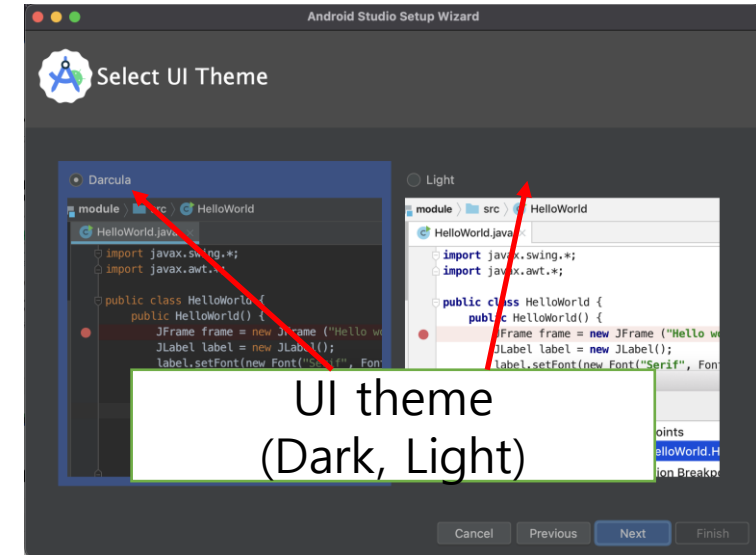
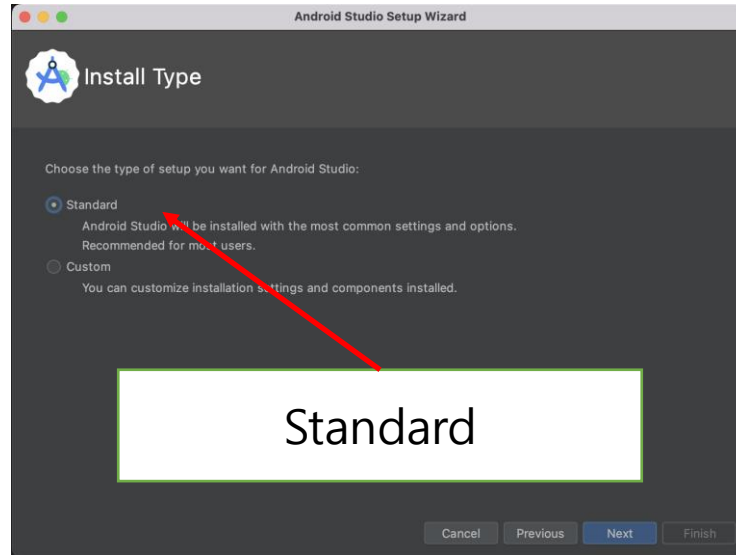
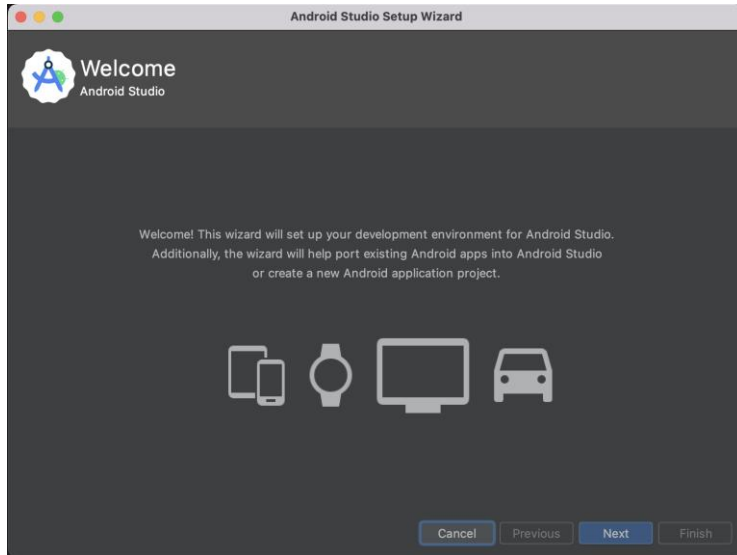
64비트 시스템에
필요한 라이브러리

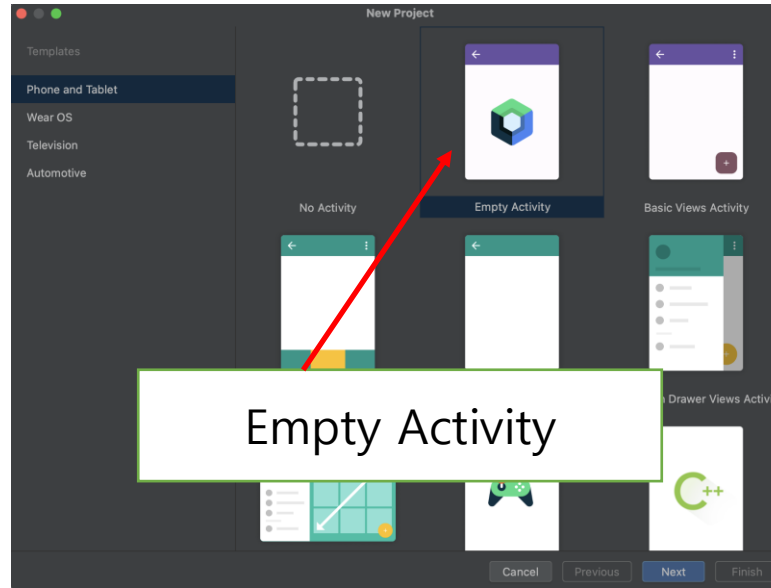
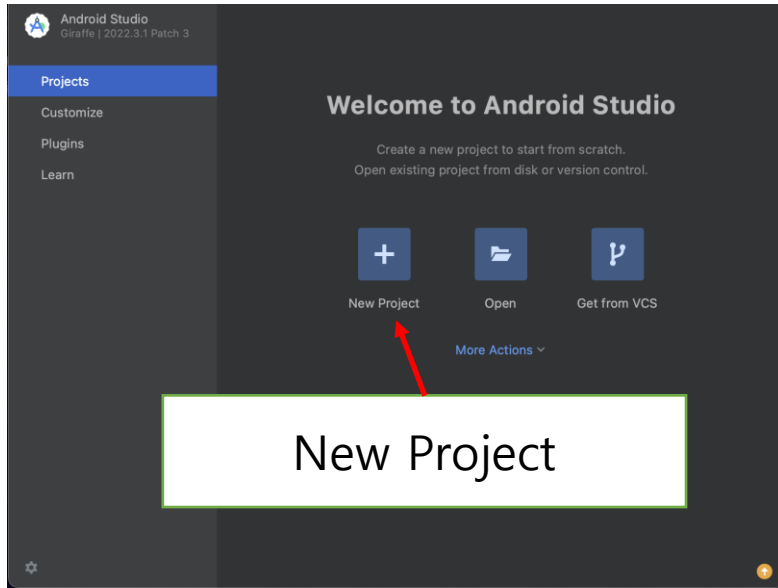
[ChromeOS](#)

Android Studio 설치



Android Studio 설치





New Project

Empty Activity

Create a new empty activity with Jetpack Compose

Name: My Application

Package name: com.example.myapplication

Save location: StudioProjects/MyApplication

Minimum SDK: API 24 ("Nougat"; Android 7.0)

Build configuration language: Kotlin DSL (build.gradle.kts) [Recommended]

Buttons: Cancel, Previous, Next, Finish

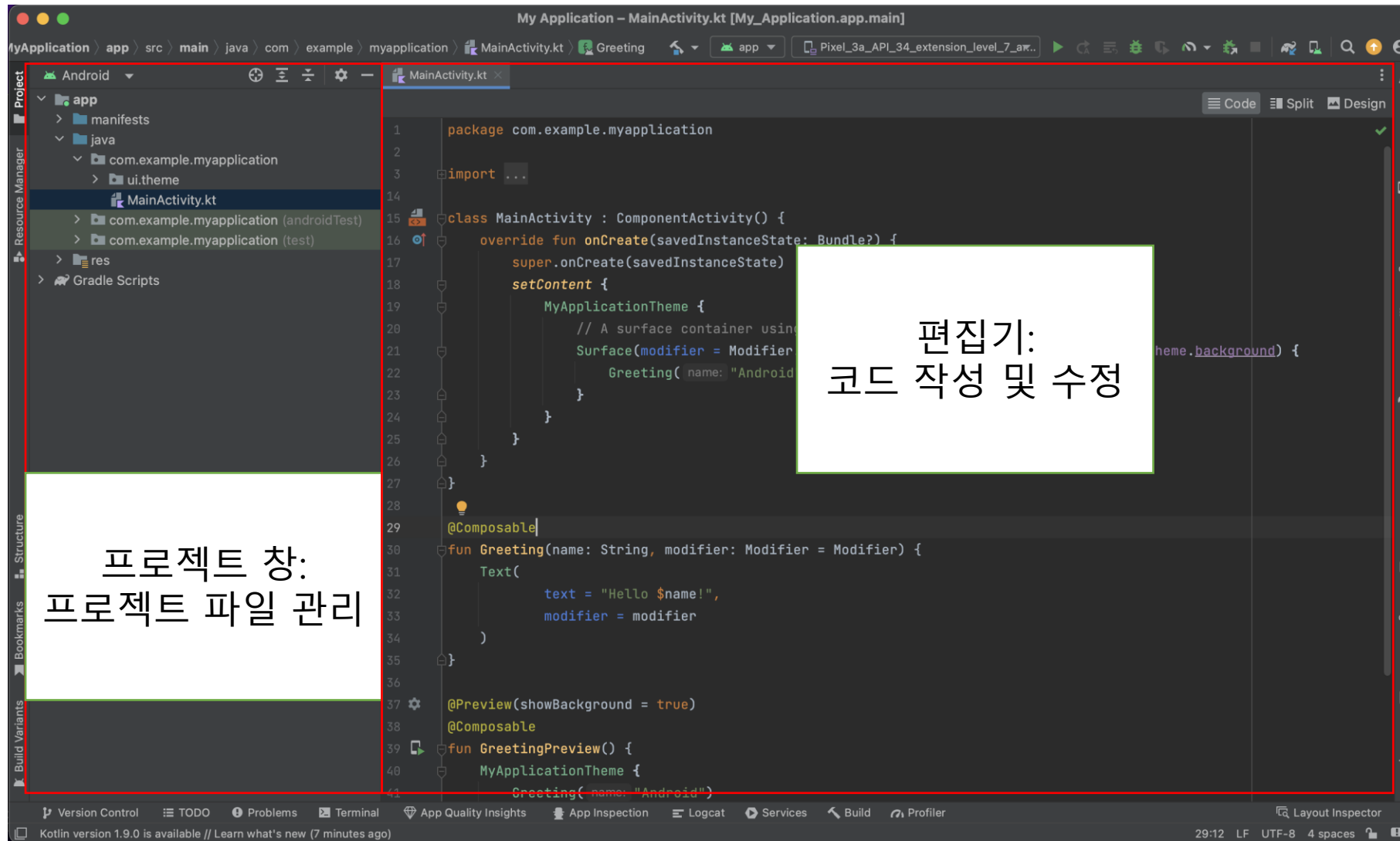
프로젝트 이름

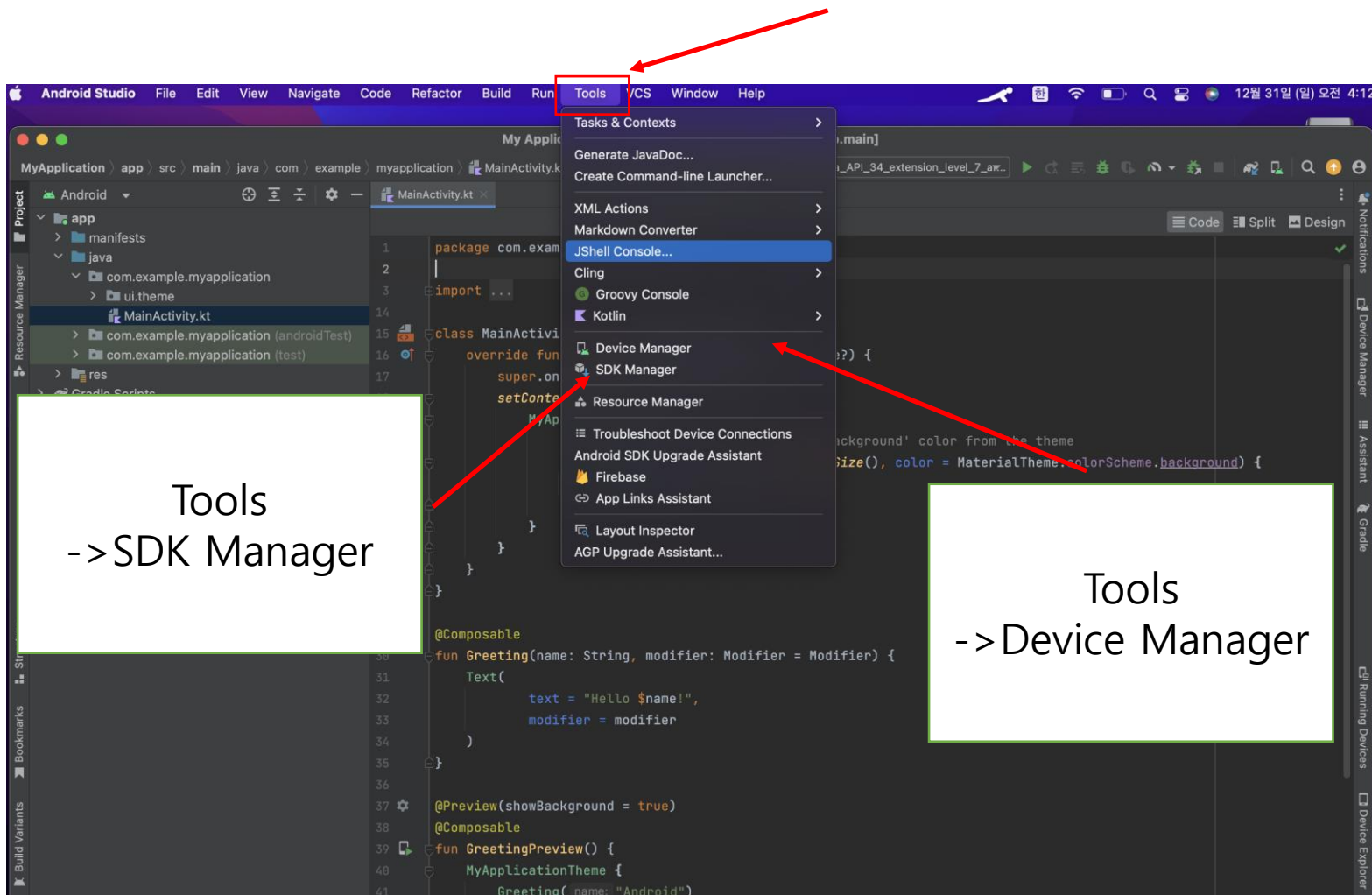
패키지 이름, 프로젝트 식별 값

프로젝트 저장 위치

SDK 버전

프로젝트 개발 언어 (Kotlin)

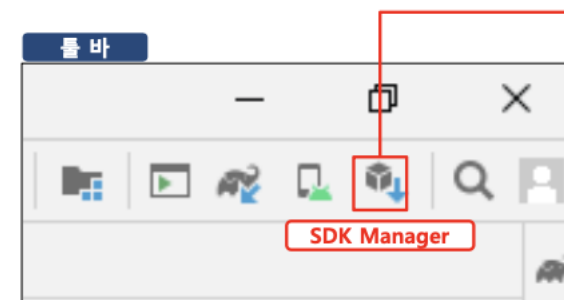




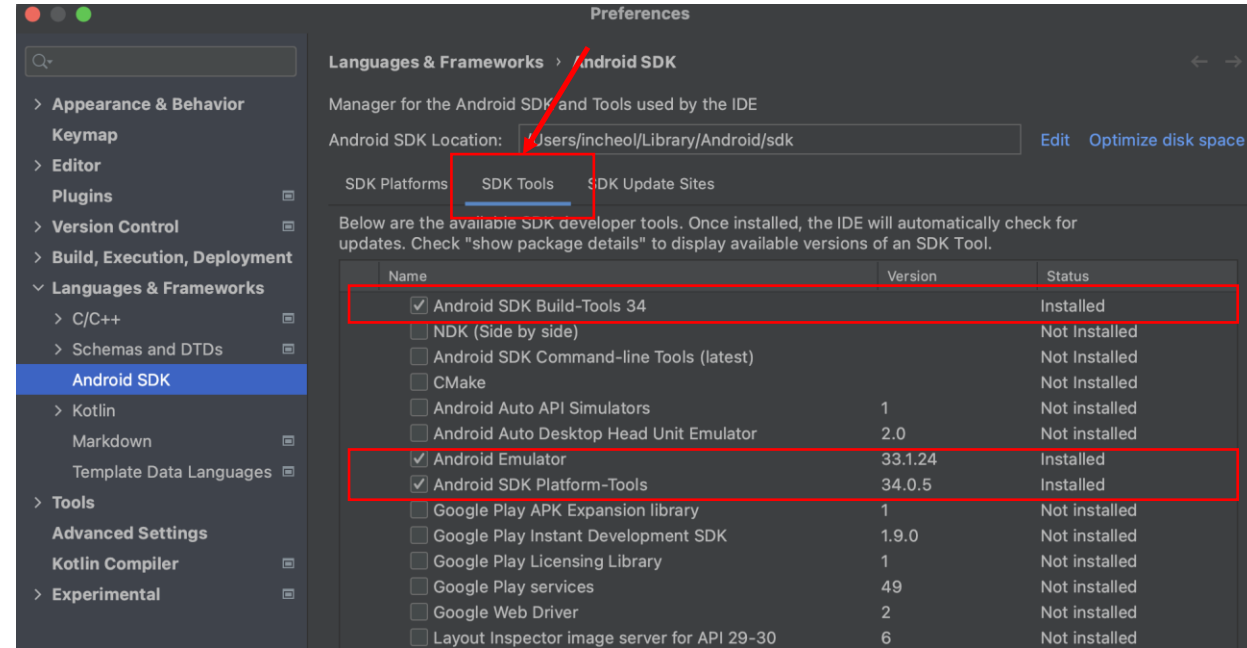
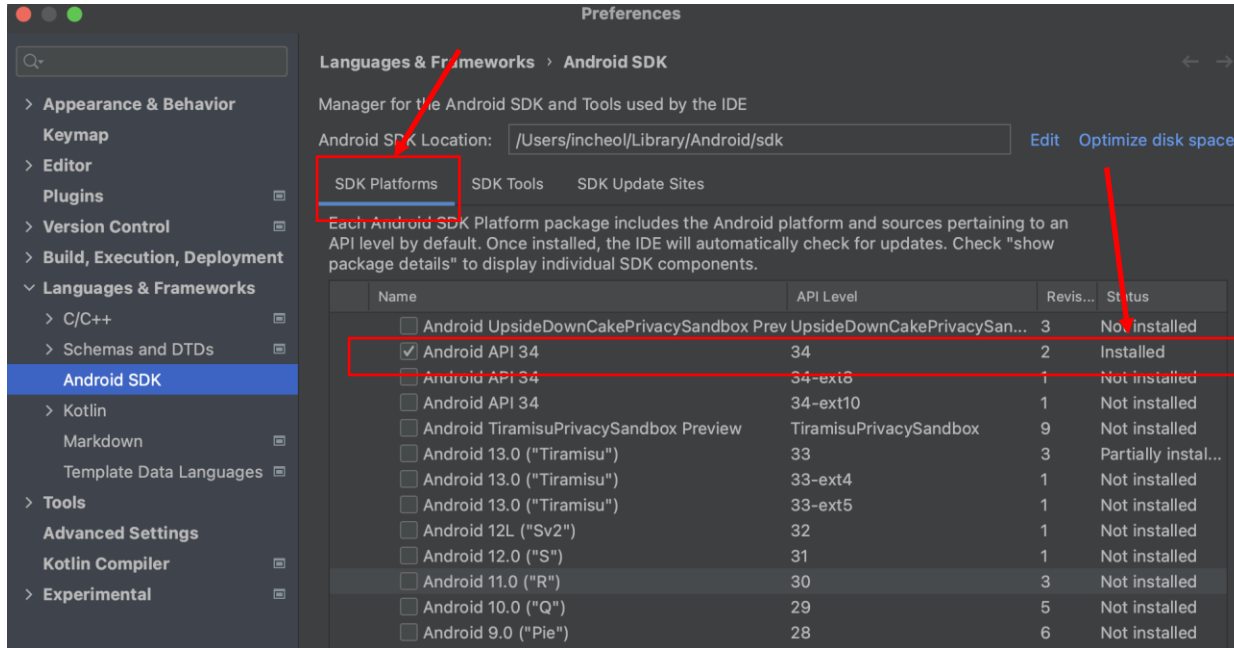
Tools
-> SDK Manager

Tools
-> Device Manager

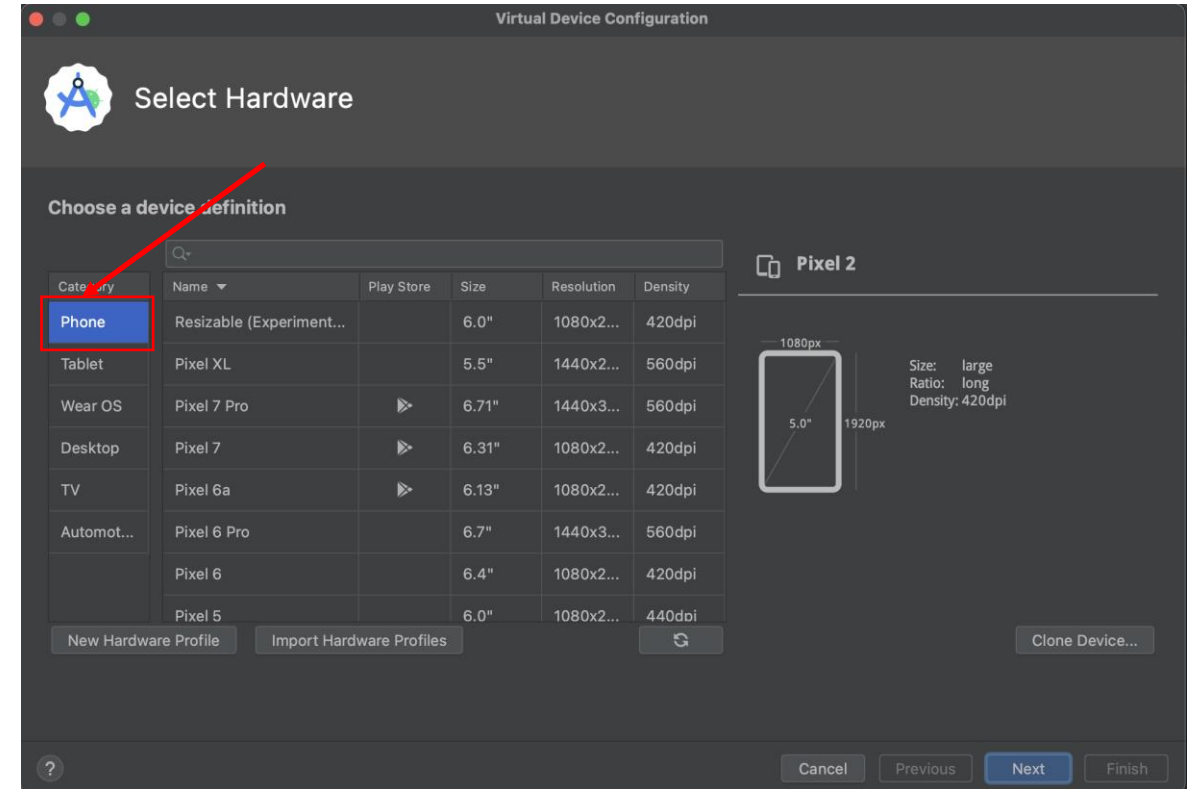
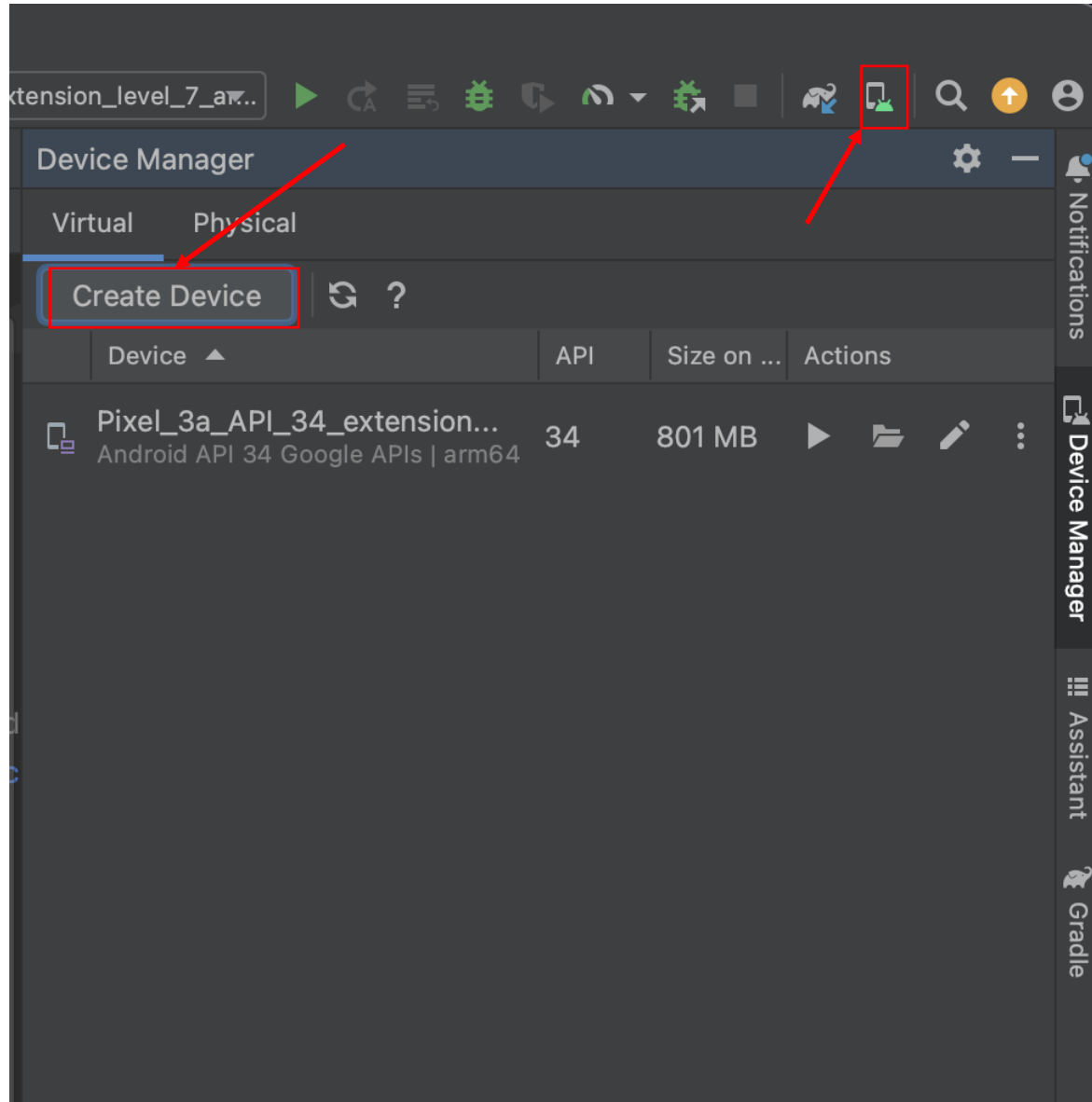
MAC

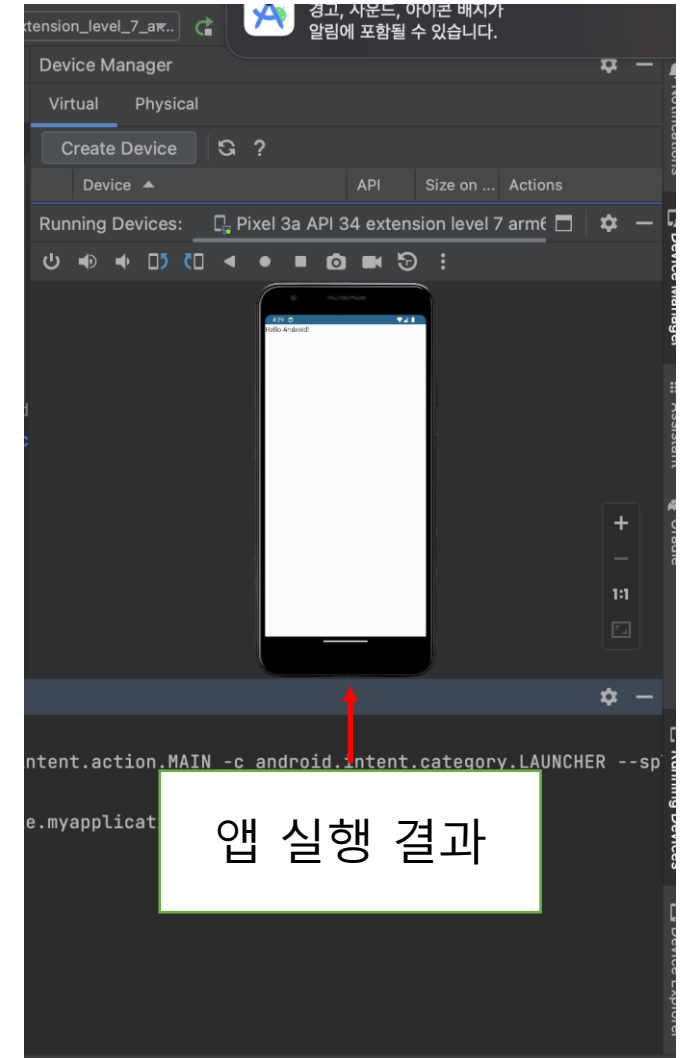
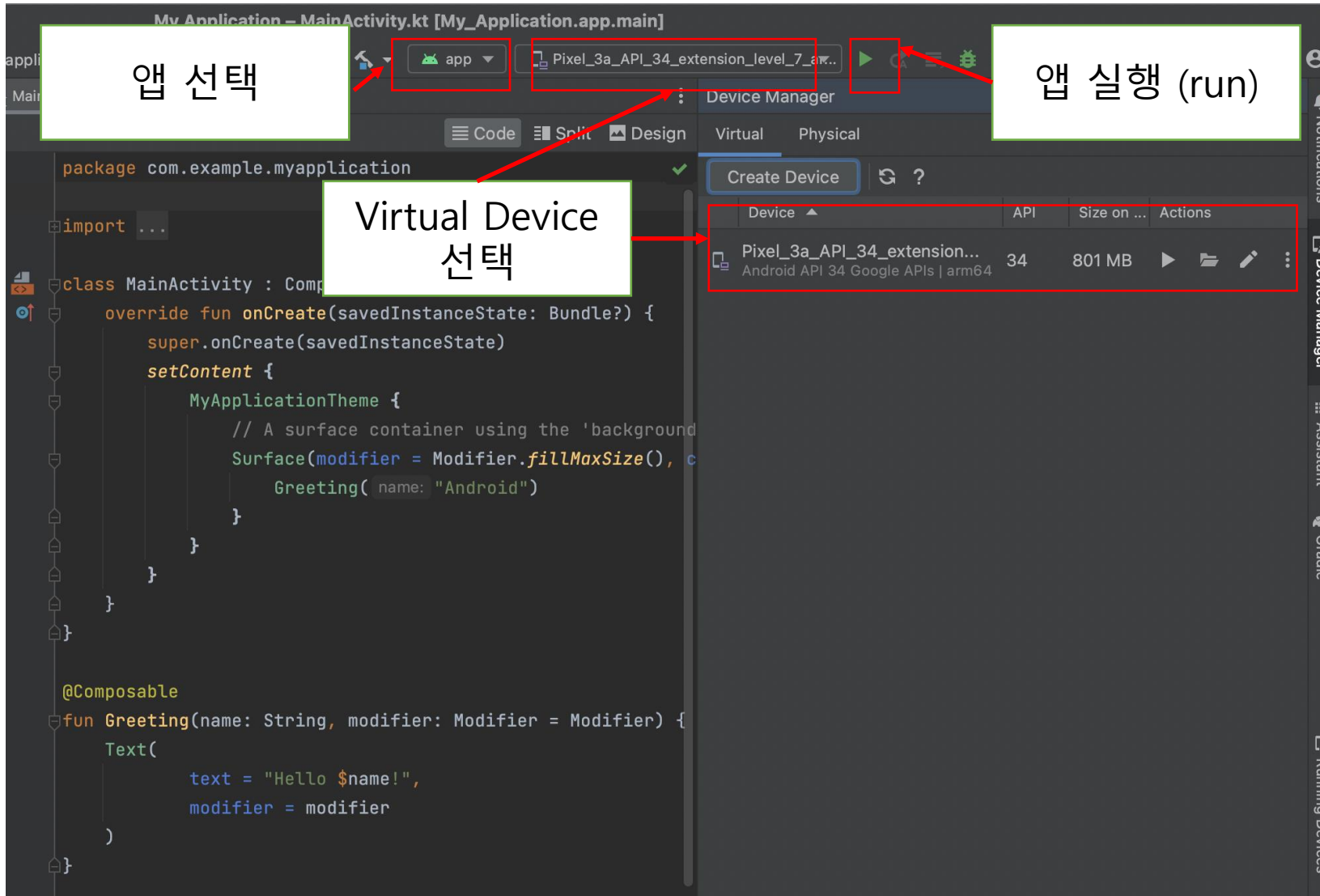


Window



Virtual Device





My Application – MainActivity.kt [My_Application.app.main]

Project Manager: MyApplication > app > src > main > res

Code Editor: MainActivity.kt

```

1 package com.example.myapplication
2
3 import ...
4
5 class MainActivity : AppCompatActivity() {
6     override fun onCreate(savedInstanceState: Bundle?) {
7         super.onCreate(savedInstanceState)
8         setContentView(R.layout.activity_main)
9         // A surface container using the 'background' attribute of a
10         // SurfaceView to draw the background.
11         Greeting(name = "Android")
12     }
13 }
14
15 @Composable
16 fun Greeting(name: String, modifier: Modifier = Modifier) {
17     Text(
18         text = "Hello $name!",
19         modifier = modifier
20     )
21 }
22
23 
```

Device Manager: Virtual, Physical, Create Device, Pixel 3a API 34

Callout 1: 앱 재실행(restart)
코드 수정된 부분
있으면 적용하고
재실행

Callout 2: Code and app design

- 윈도우용 USB 드라이버 설치

- <https://developer.android.com/studio/run/oem-usb?hl=ko>

OEM 드라이버 가져오기

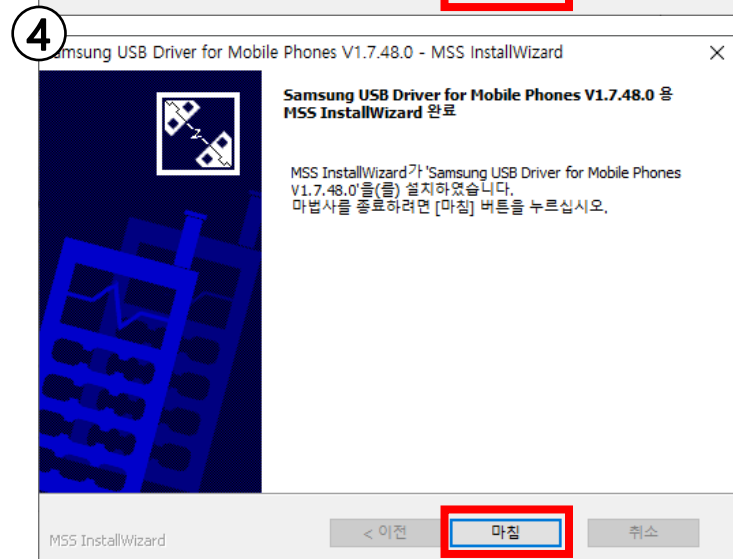
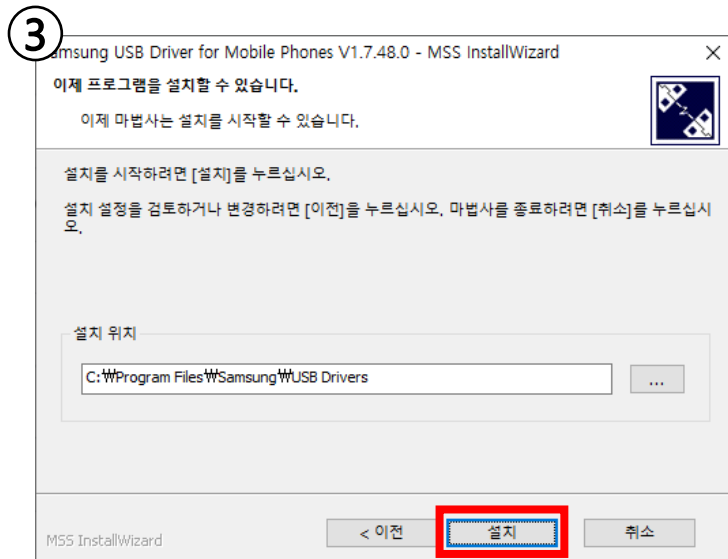
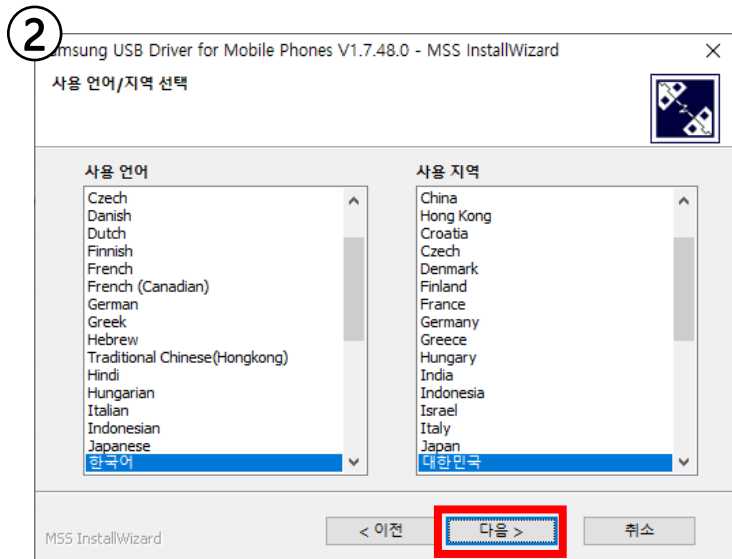
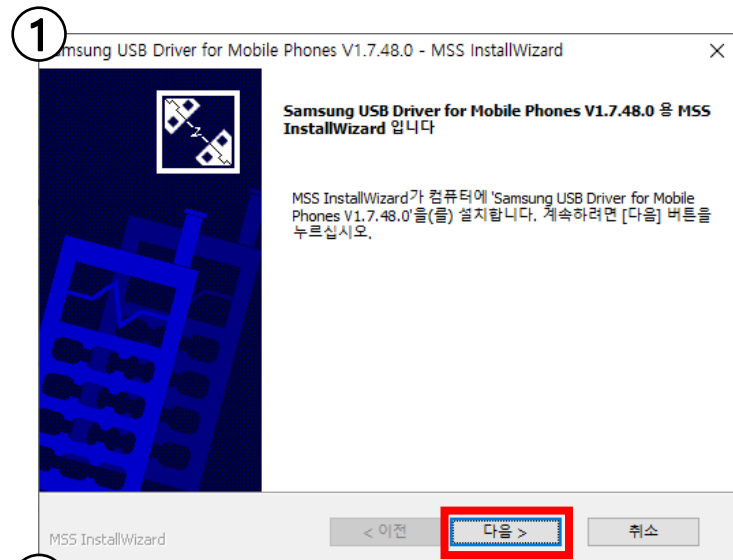
OEM	드라이버 URL
Acer	http://www.acer.com/worldwide/support/
ALCATEL ONE TOUCH	http://www.alcatelonetouch.com/global-en/support/
Asus	https://www.asus.com/support/Download-Center/
LGE	http://www.lg.com/us/support/software-firmware
Motorola	https://motorola-global-portal.custhelp.com/app/answers/detail/a_id/88481/
MTK	http://online.mediatek.com/Public%20Documents/MTK_Android_USB_Driver.zip (ZIP 다운로드)
삼성	http://developer.samsung.com/galaxy/others/android-usb-driver-for-windows

• 윈도우용 USB 드라이버 설치

Samsung Android USB Driver for Windows

You need the driver only if you are developing on Windows and want to connect a Samsung Android device to your development environment over USB.

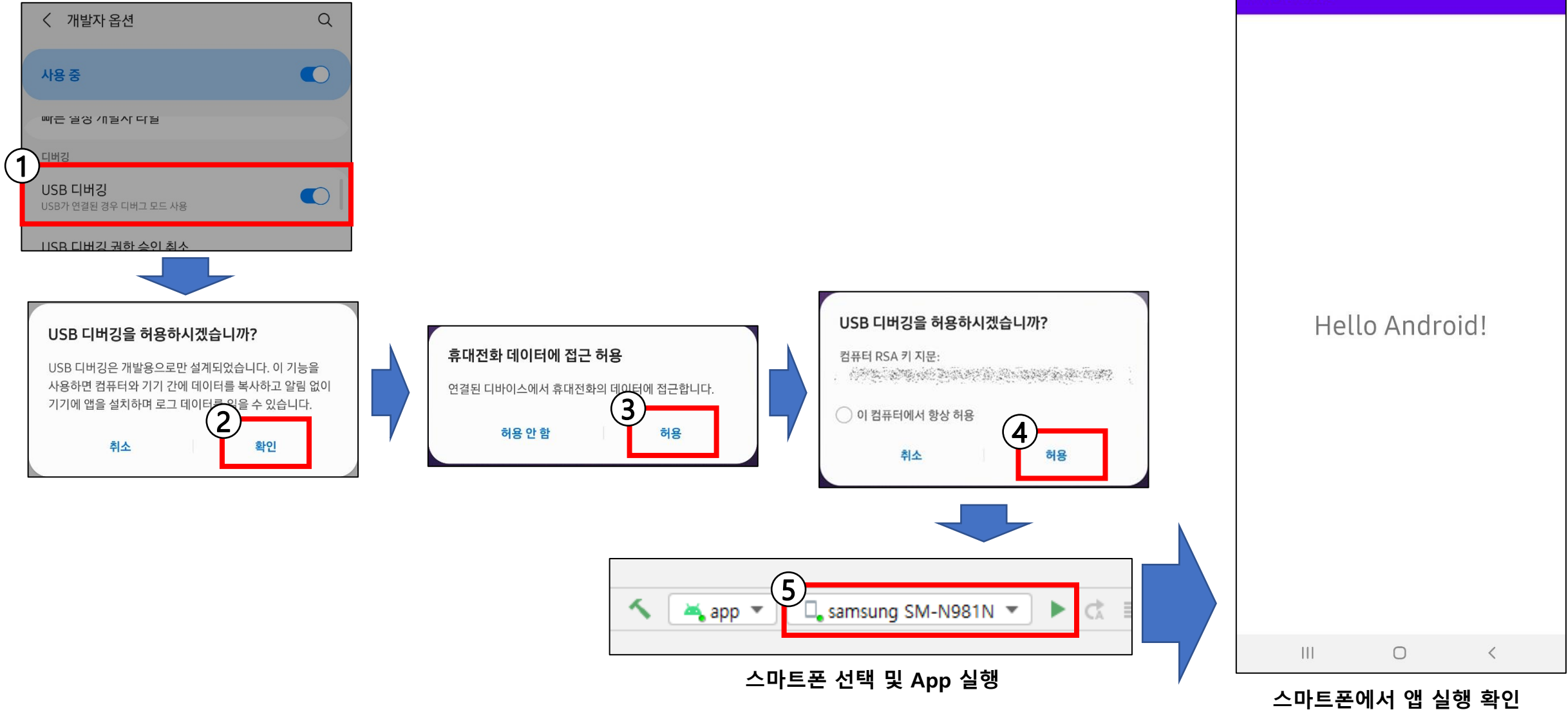
↓ Samsung Android USB Driver for Windows v1.7.48 (36.89MB) | Oct 25, 2021



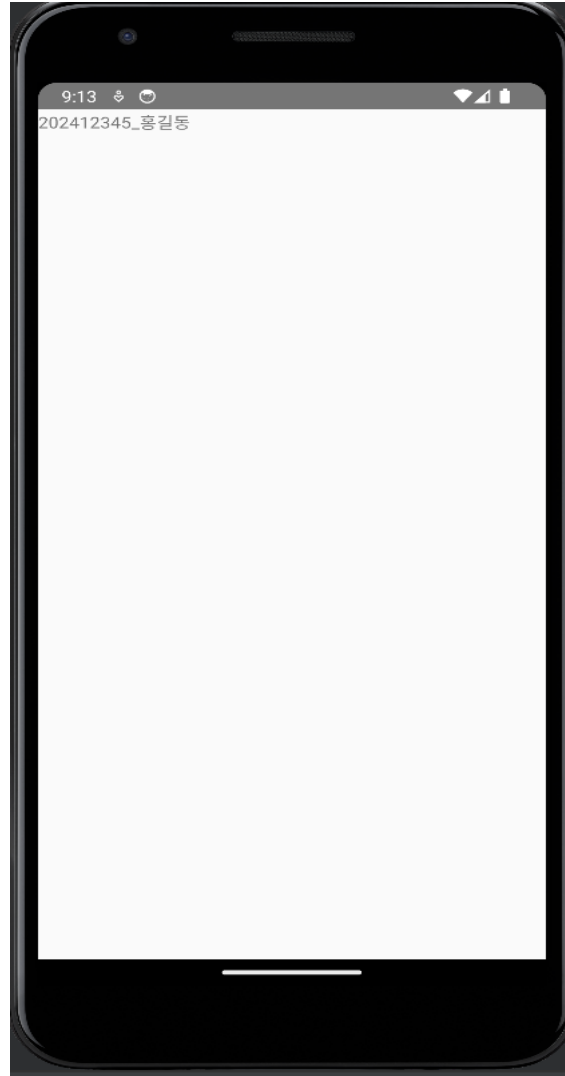
• 스마트폰 설정



• 스마트폰 설정



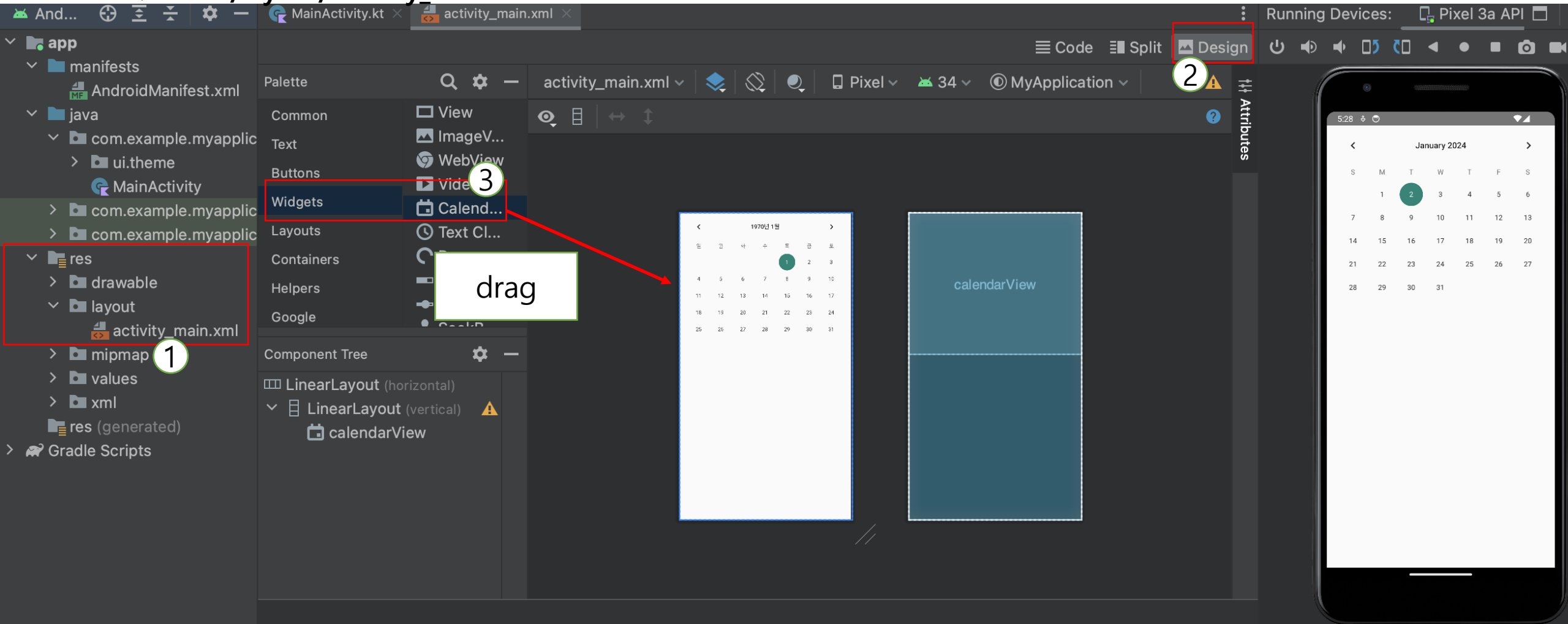
예제 1 – 학번, 이름 출력하기



예제 2 – GUI 화면 조작하기

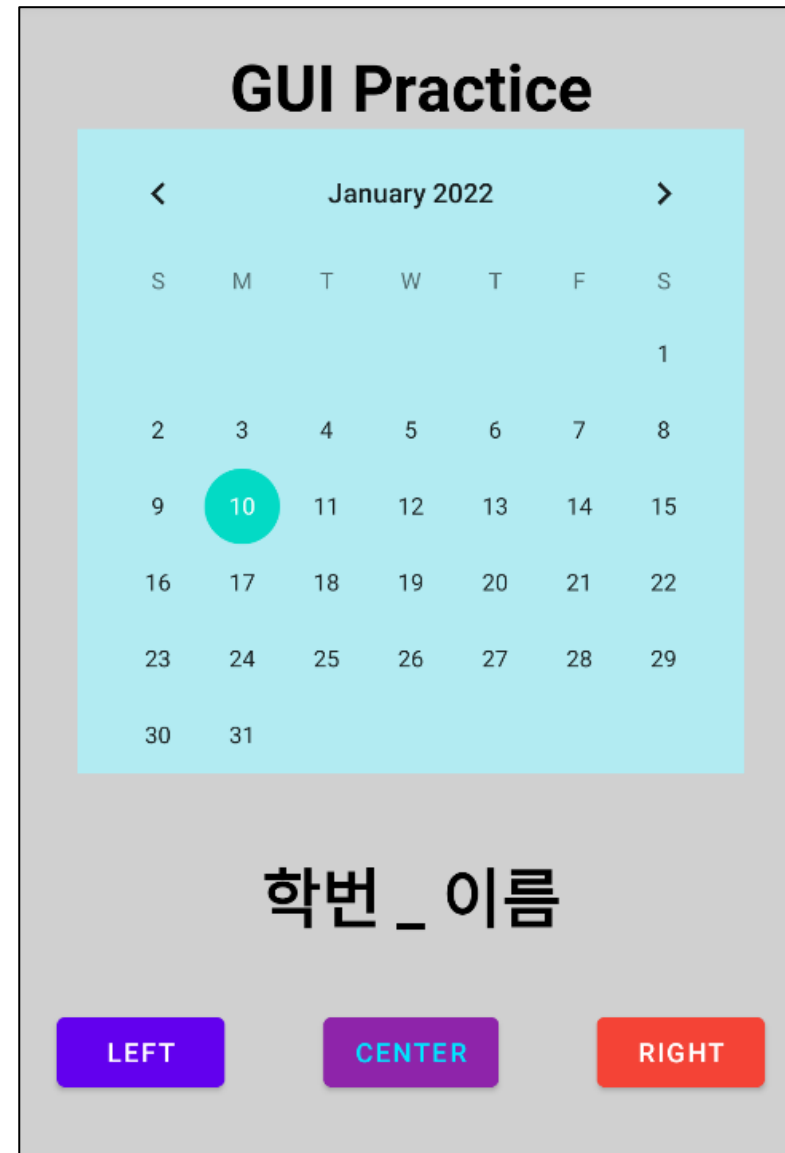
◆ Widgets 사용

◆ Res/layout/activity_main.xml



예제 2 – GUI 화면 조작하기

- ◆ CalendarView 삽입
- ◆ Button 삽입
- ◆ TextView 삽입



감사합니다!