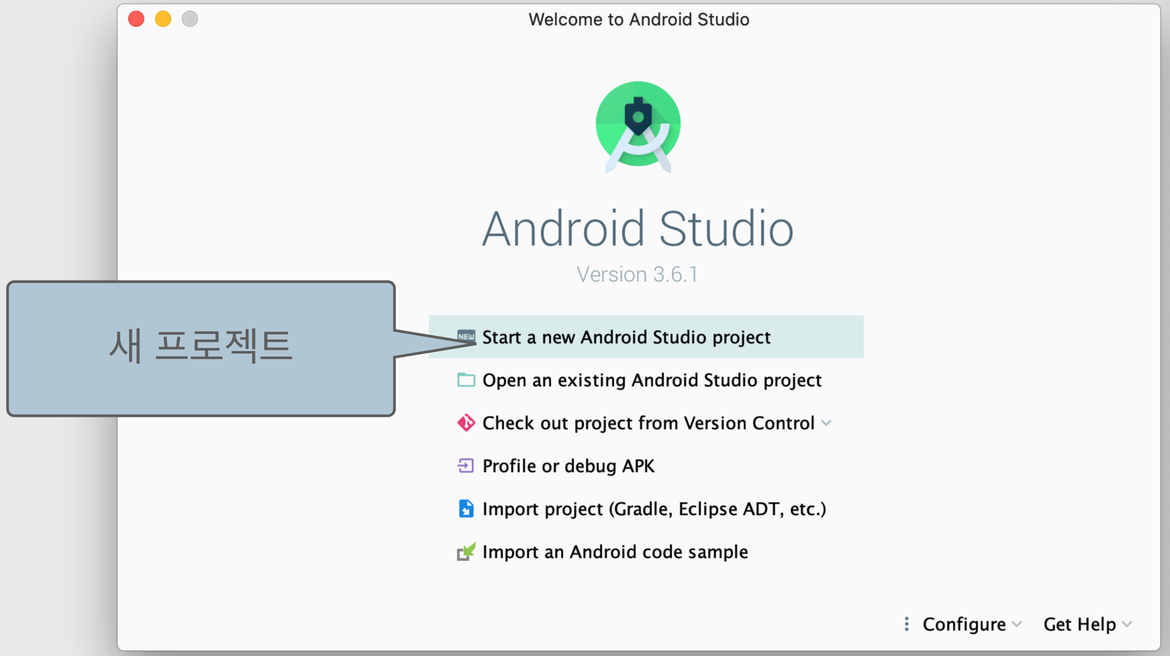
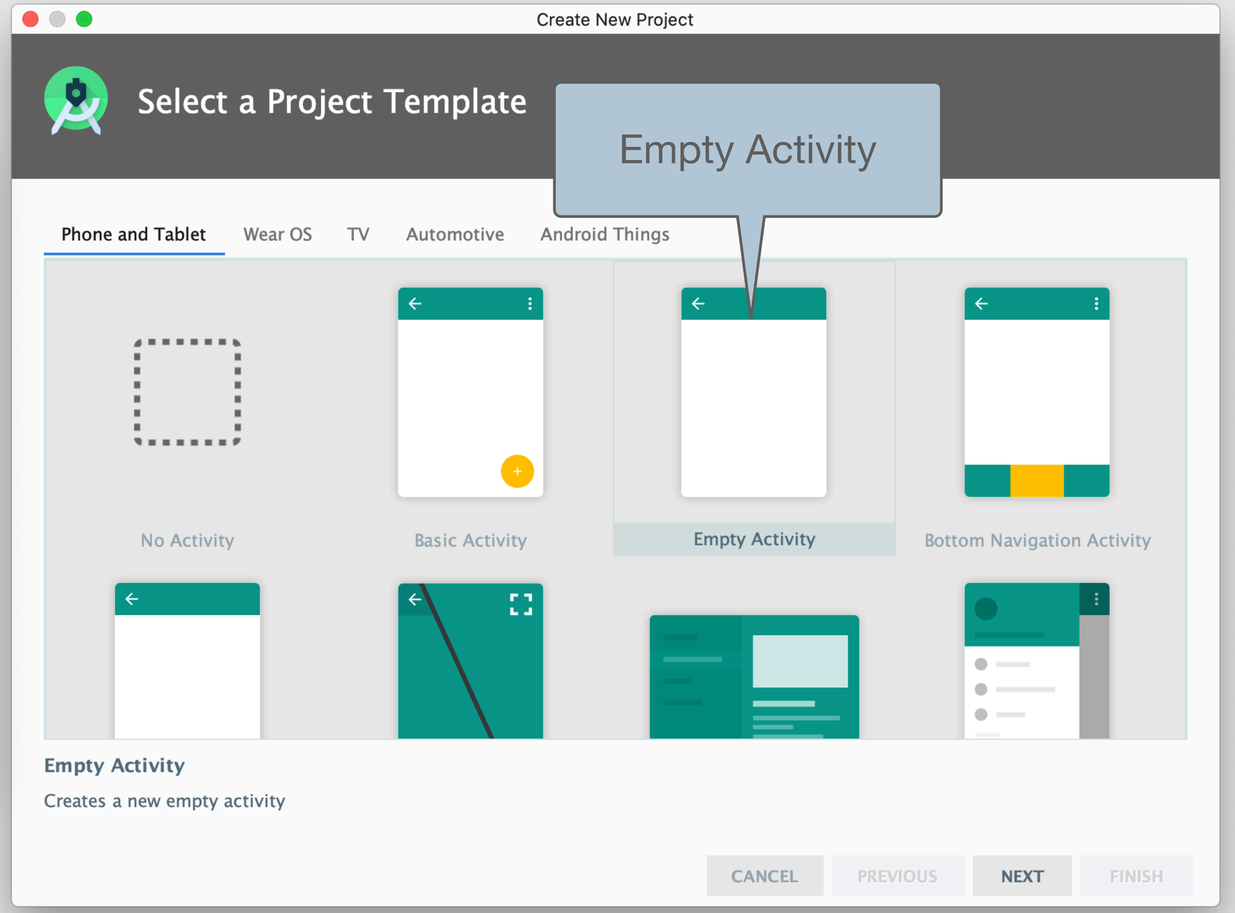
Android application 만들기

2. Hello World

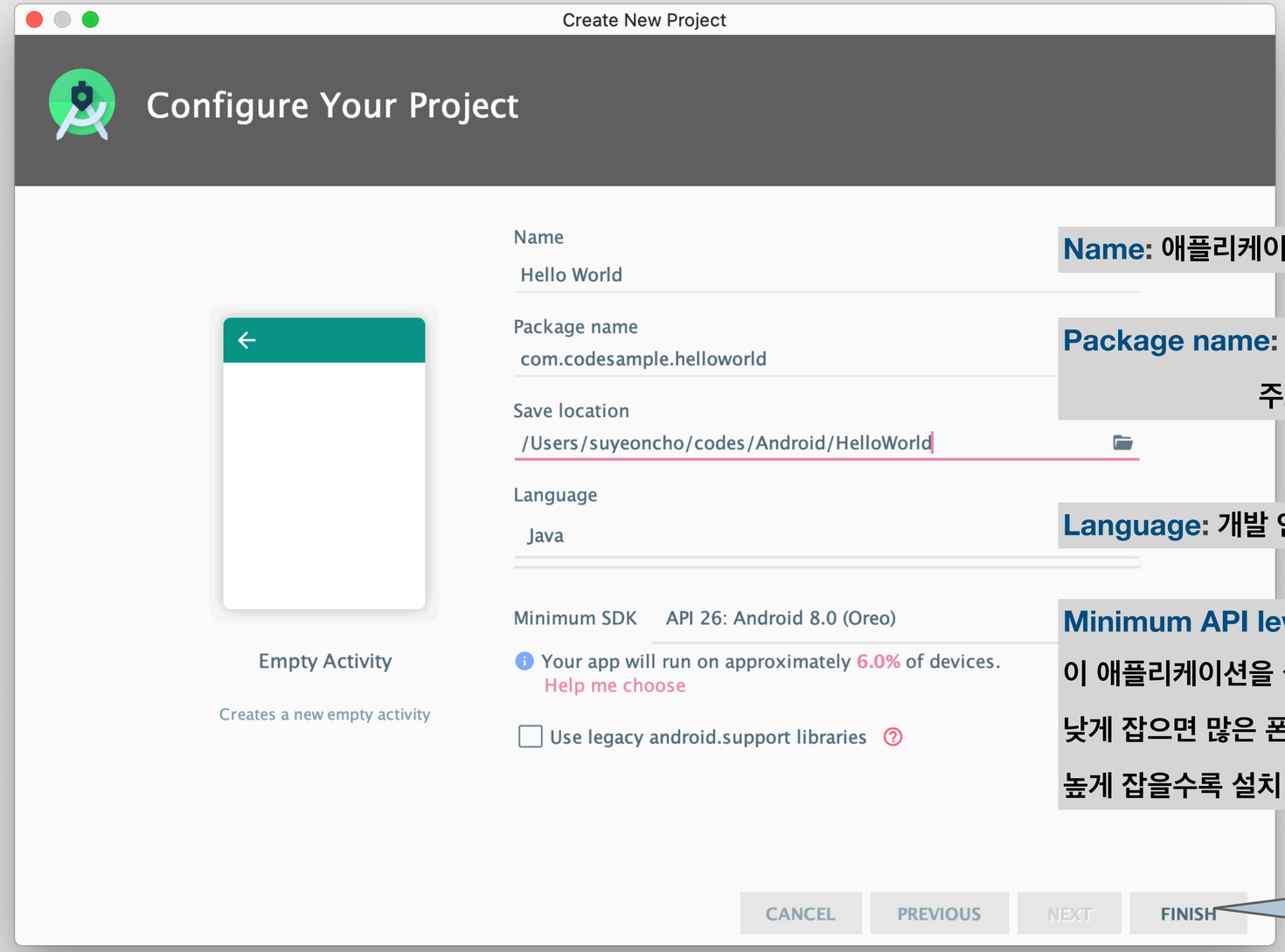
Hello World, AVD, USB Debugging

새 프로젝트 만들기


Welcome to Android Studio
Android Studio
Version 3.6.1
Start a new Android Studio project
Open an existing Android Studio project
Check out project from Version Control
Profile or debug APK
Import project (Gradle, Eclipse ADT, etc.)
Import an Android code sample
Configure
Get Help
새 프로젝트


Create New Project
Select a Project Template
Empty Activity
Phone and Tablet
Wear OS
TV
Automotive
Android Things
No Activity
Basic Activity
Empty Activity
Bottom Navigation Activity
Empty Activity
Creates a new empty activity
CANCEL
PREVIOUS
NEXT
FINISH

새 프로젝트 만들기



Name: 애플리케이션 이름. 너무 길지 않게 정한다.

Package name: 애플리케이션을 식별할 패키지 이름.

주로 회사 도메인을 역순 + 애플리케이션 이름으로 적는다.

Language: 개발 언어. Java와 Kotlin 중 하나를 선택할 수 있다.

Minimum API level:

이 애플리케이션을 설치하기 위한 폰의 최소 운영체제 버전.

낮게 잡으면 많은 폰에 설치 가능하지만 최신 API 들을 사용할 수 없고

높게 잡을수록 설치 가능한 폰의 수가 줄어든다.

모두 확인 후 Finish

새 프로젝트 살펴보기

제일 위에 Android로 표기 된 경우의 구성으로 실제 폴더 구성과 다르다.

이 모양으로 출력된다면 컴파일이 성공한 것.

Java: 소스코드 작업 폴더. MainActivity가 있는 패키지에서 작업한다.

test 폴더는 테스트를 위한 곳으로 실제 동작하는 코드가 아니다.

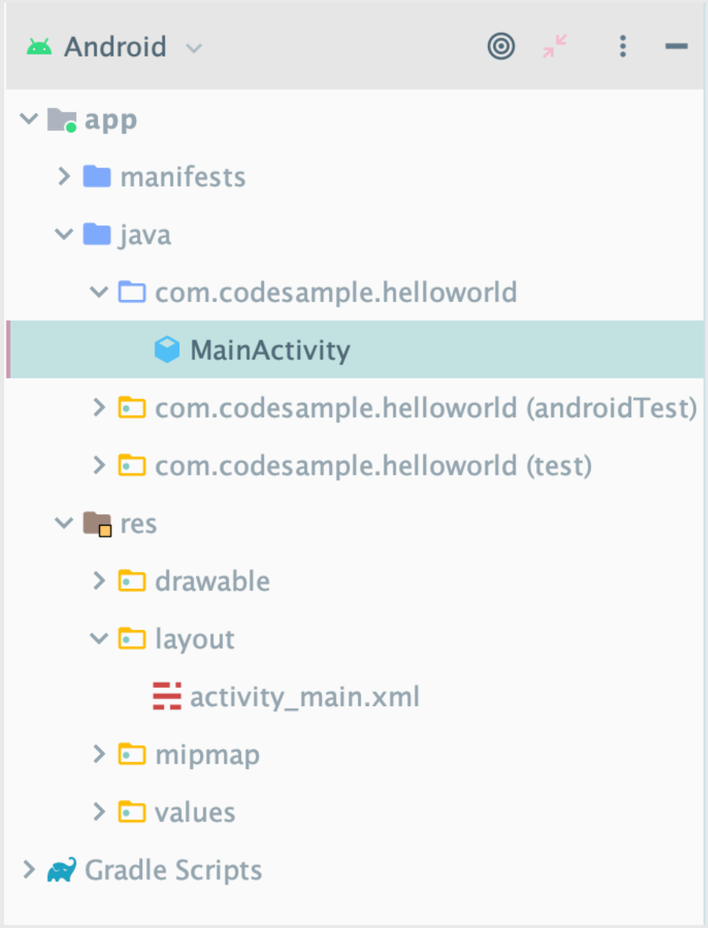
res: 소스코드를 제외한 모든 정적인 파일들과 설정을 하는 곳.

drawable: 모든 그림 파일들을 관리

layout: 화면의 디자인 관리

mipmap: 애플이케이션이 설치되었을 때 보이는 아이콘 관리

values: 테마, 컬러, 문자열 등 각종 설정 파일이나 정적인 콘텐츠를 관리.



com.codesample.helloworld.MainActivity.java

```java
package com.codesample.helloworld;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

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

res/layout/activity_main.xml

에디터 우측 상단의 다음 버튼으로 코드 보기 가능 

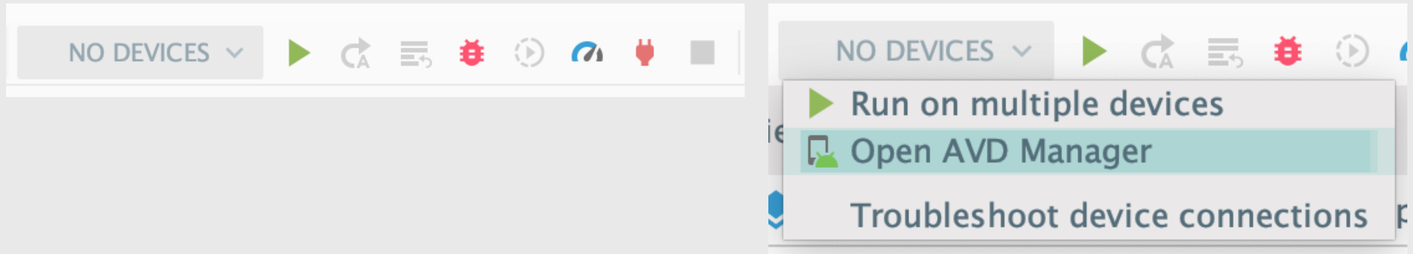
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    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>
```

AVD로 실행

AVD(Android Virtual Device)

PC 환경에서 가상의 안드로이드 디바이스를 실행한다.

디바이스가 없는 상태. Open AVD Manager를 실행한다.

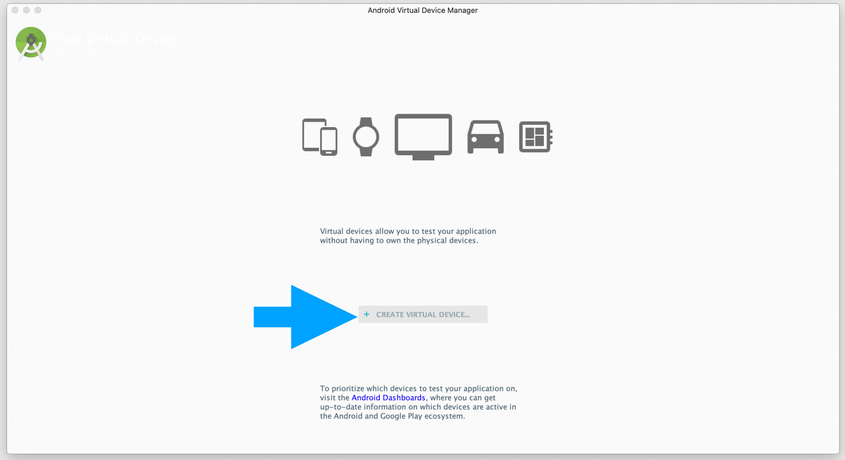


연결 된 디바이스가 있을 경우 NO DEVICES자리에 해당 디바이스의 모델명이 출력되고

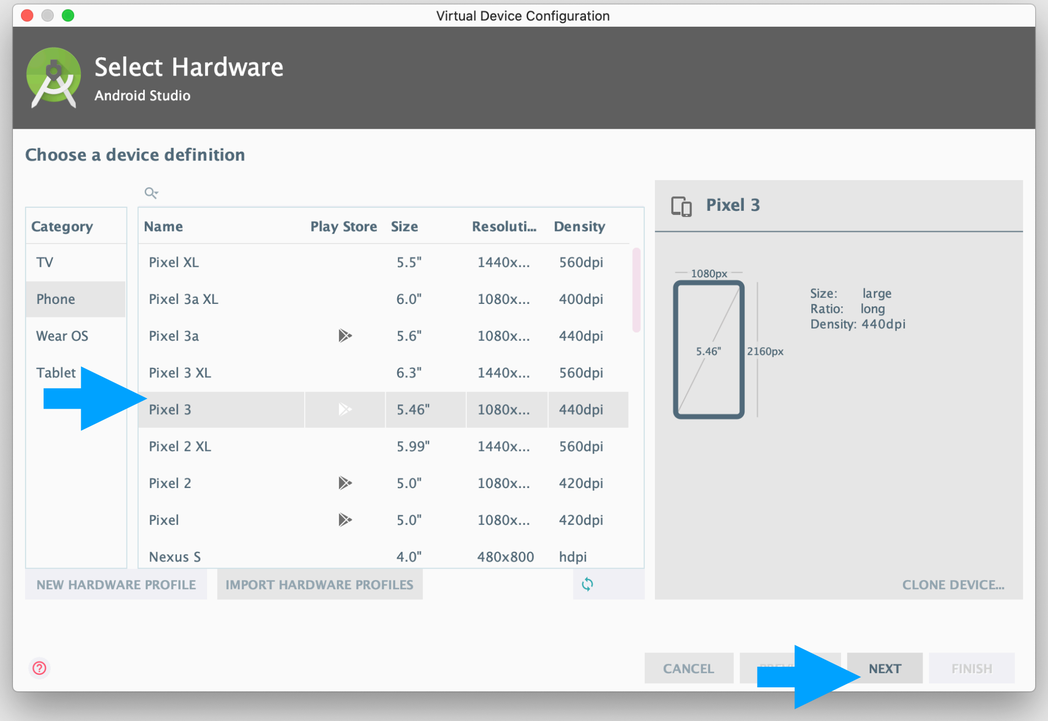
이 경우 녹색 플레이 버튼으로 바로 실행 가능하다.

AVD로 실행

AVD(Android Virtual Device) Manager



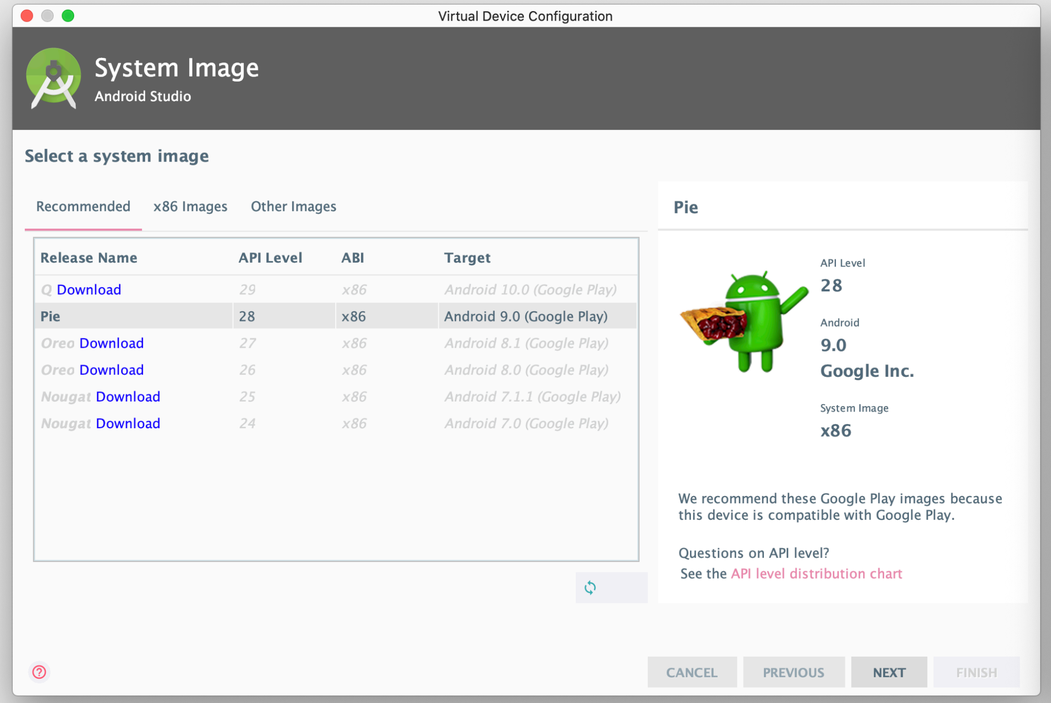
CREATE VIRTUAL DEVICE
버튼 클릭

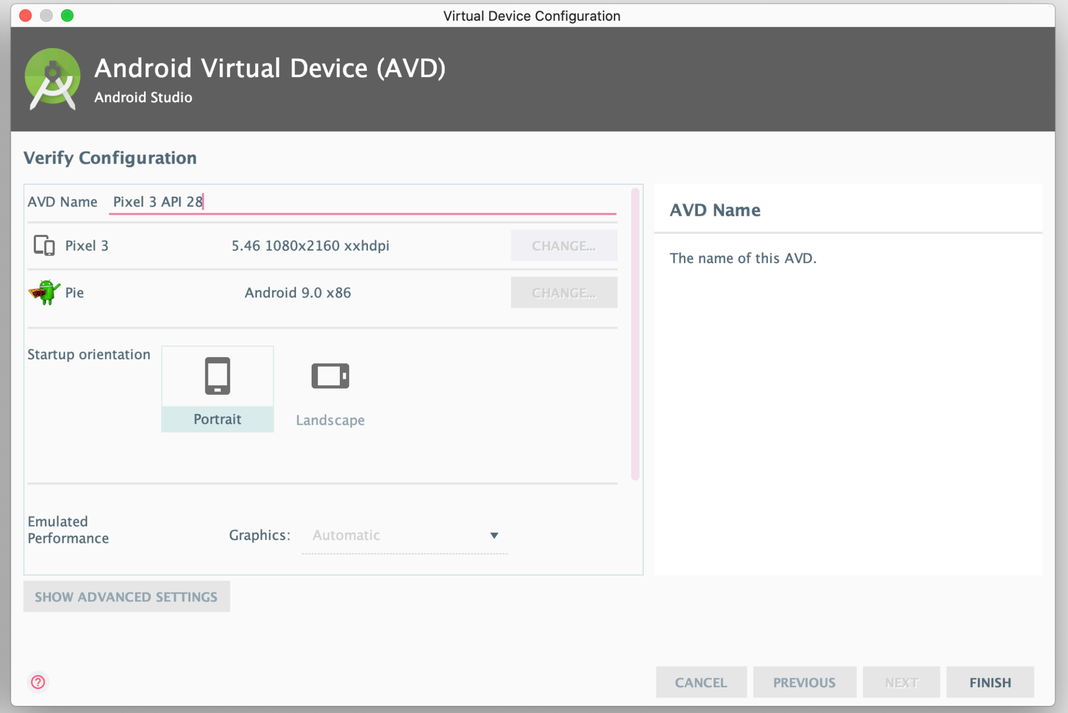


Pixel2 시리즈는 9:16 화면
Pixel3 시리즈는 좀 더 세로 큰 화면.
구글 API를 사용하려면 Play Store 아이콘이 있는 것을 선택한다.

AVD로 실행

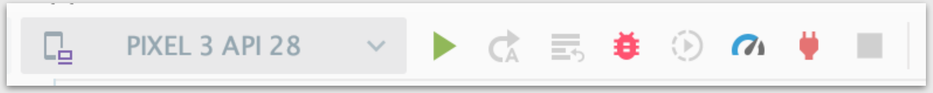
AVD(Android Virtual Device) Manager





System version을 선택하여 다운받는다.

기기의 이름 등을 확인하고 FINISH. 완료 후 기기 이름이 뜨는지 확인



AVD로 실행

폰 부팅 후 앱을 설치하고 실행 화면을 보여준다.

장점

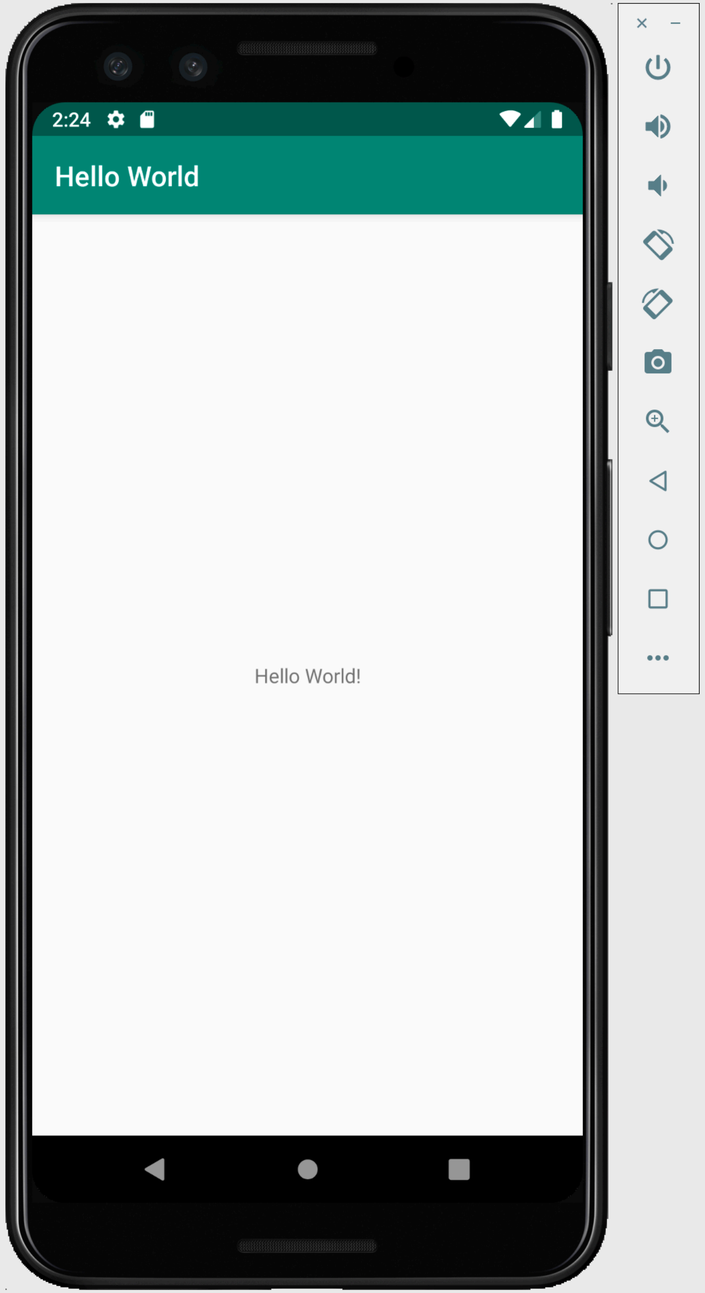
구하기 힘든 버전의 디바이스에서 앱을 실행해 볼 수 있다.

다양한 해상도에서의 실행을 확인 할 수 있다.

단점

시스템 메모리를 나누어 쓴다.

블루투스 등 일부 기능은 지원하지 않는다.



실제 안드로이드 디바이스로 실행

PC 준비

사용하고자 하는 디바이스의 통합 USB 드라이버를 설치한다.

설치 전에 디바이스와 연결 된 USB 케이블은 뺀다.

구글 피쳐폰은 안드로이드 스튜디오에서 다운 받을 수 있지만 다른 모델은 제조사 홈페이지에서 구한다.

디바이스 준비

개발자 모드를 활성화한다.

설정 > 디바이스정보 > 소프트웨어정보 > 빌드번호 7번 연속 터치

개발자 모드 메뉴에서 USB 디버깅을 허용한다.

최초로 USB 케이블을 연결하면 디바이스쪽에 보안 팝업이 뜨는데 디버깅을 허용하면 된다.