



[한국ICT인재개발원] 안드로이드

1. 안드로이드 개요 및 환경설정

前) 광고데이터 분석 1년

前) IT강의 경력 2년 6개월

前) 머신러닝을 활용한 데이터 분석 프로젝트반 운영 1년

前) 리그오브 레전드 데이터 분석 등...

現) 국비반 강의 진행중

구글이 2007년에 안드로이드사를 인수하며 시작했습니다.

이클립스와 안드로이드 스튜디오를 지원하지만

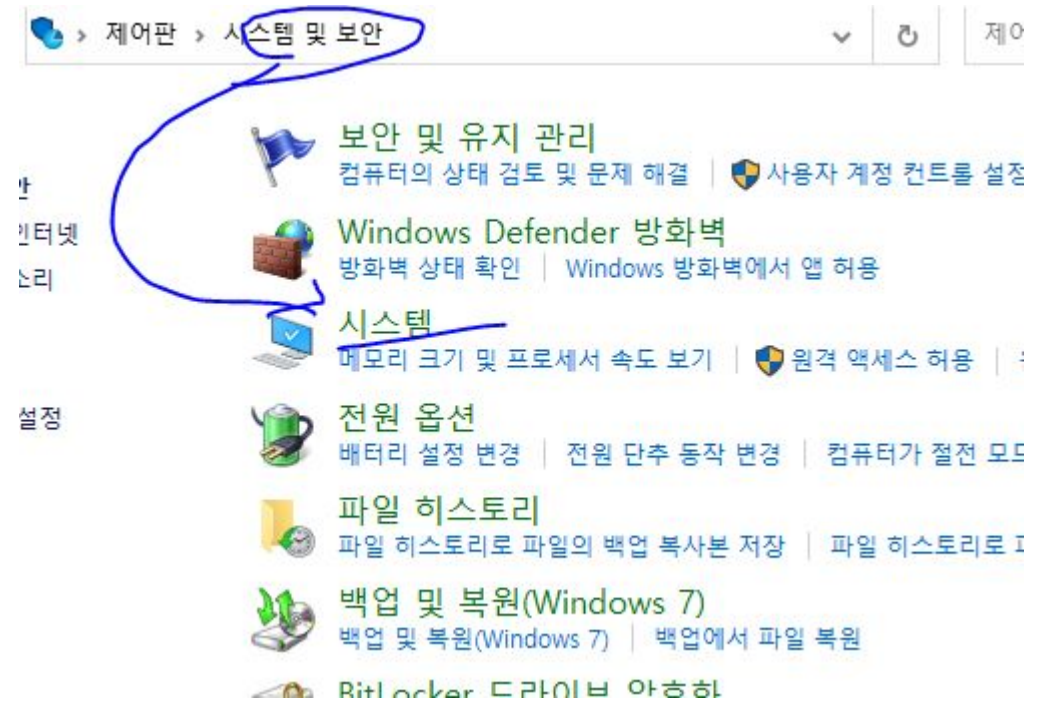
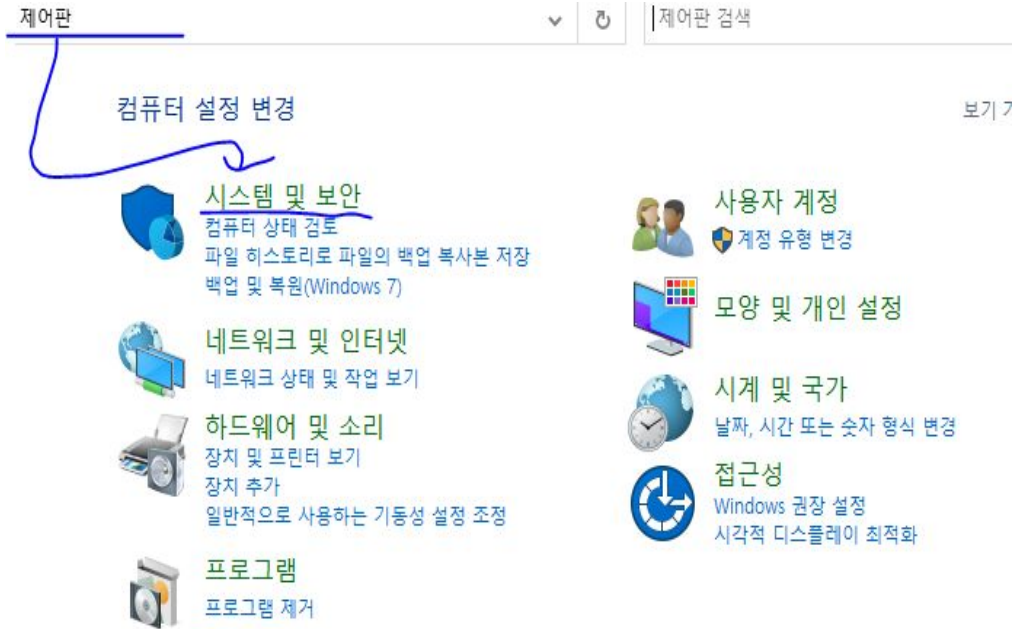
현재 이클립스는 거의 사장된 상태입니다.

안드로이드 스튜디오는 4.1버전을 사용할 예정입니다.

개발 언어는 자바와 코틀린이 있으며, 최근 코틀린쪽이 대세이지만

코틀린 모드에서도 자바코드를 입력해도 돌아가기도 하고

일정상 수업진행은 자바로 진행합니다.



환경설정을 시작합니다. 먼저 인텔 HAXM을 설치합니다.
HAXM은 가상의 휴대폰을 컴퓨터로 구현해 어플리케이션 동작여부를
체크하도록 도와줍니다.

장치 사양

디바이스 이름	DESKTOP-0GBL09R
프로세서	Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz
설치된 RAM	8.00GB(7.82GB 사용 가능)
장치 ID	428D55C7-4B38-4B58-AC4E-E2290AF749A5
제품 ID	00330-80000-00000-AA765
시스템 종류	64비트 운영 체제, x64 기반 프로세서
펜 및 터치	이 디스플레이에 사용할 수 있는 펜 또는 터치식 입력이 없습니다.



HAXM v7.6.6

wcwang released this 4 days ago · 18 commits to master since this release

Change Log

- Optimized the CPUID module and added support for setting two new CPUID leaves (#335).
- Fixed some vulnerability issues of loading DRs and MSRs (#347).
- Fixed some minor issues from static code scan (#351).
- Fixed the download URL in the homepage (#348).

Filename	SHA-256
haxm-windows_v7_6_6.zip	b970149a0b9d4f65fdd7879a71a497929e758d9a173eb2363b9250e104c07f49
haxm-macosx_v7_6_6.zip	ba2ecae32acc4d635bf2b8edc2dffc0679076e0a74a707597b53291aa0fa4263

Assets 4

haxm-macosx_v7_6_6.zip	606 KB
haxm-windows_v7_6_6.zip	501 KB
Source code (zip)	
Source code (tar.gz)	

다음, 시스템이 인텔임이 확인되었다면
<https://github.com/intel/haxm/releases/>
에 접속해 최신 버전 윈도우용 HAXM 압축파일을 찾아 다운로드합니다.

장치 사양

디바이스 이름	DESKTOP-0GBL09R
프로세서	Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz
설치된 RAM	8.00GB(7.82GB 사용 가능)
장치 ID	428D55C7-4B38-4B58-AC4E-E2290AF749A5
제품 ID	00330-80000-00000-AA765
시스템 종류	64비트 운영 체제, x64 기반 프로세서
펜 및 터치	이 디스플레이에 사용할 수 있는 펜 또는 터치식 입력이 없습니다.



HAXM v7.6.6

wcwang released this 4 days ago · 18 commits to master since this release

Change Log

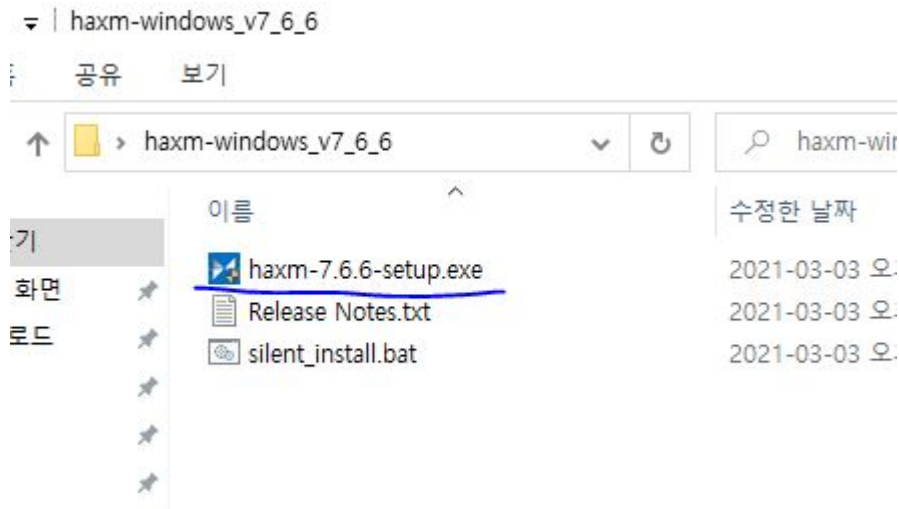
- Optimized the CPUID module and added support for setting two new CPUID leaves (#335).
- Fixed some vulnerability issues of loading DRs and MSRs (#347).
- Fixed some minor issues from static code scan (#351).
- Fixed the download URL in the homepage (#348).

Filename	SHA-256
haxm-windows_v7_6_6.zip	b970149a0b9d4f65fdd7879a71a497929e758d9a173eb2363b9250e104c07f49
haxm-macosx_v7_6_6.zip	ba2ecae32acc4d635bf2b8edc2dffc0679076e0a74a707597b53291aa0fa4263

Assets 4

haxm-macosx_v7_6_6.zip	606 KB
haxm-windows_v7_6_6.zip	501 KB
Source code (zip)	
Source code (tar.gz)	

다음, 시스템이 인텔임이 확인되었다면
<https://github.com/intel/haxm/releases/>
에 접속해 최신 버전 윈도우용 HAXM 압축파일을 찾아 다운로드합니다.



다운 받은 압축파일을 풀고
haxm-7.6.6-setup.exe 파일을 실행합니다.
이어지는 화면에서는 계속 next나 install을 눌러서 진행해주시면 됩니다.
진행 완료 후 finish를 눌러 마무리해주세요.

Android 스튜디오 다운로드 자료실

이 페이지는 Android 스튜디오 출시 자료실입니다.

하지만 [최신 정식 버전](#) 또는 [최신 미리보기 버전](#)을 다운로드하는 것이 좋습니다.

Android 스튜디오 Arctic Fox(2020.3.1) Canary 11 2021년 3월 22일

Android 스튜디오 4.1.3 2021년 3월 18일

설치 프로그램

Windows IDE 전용 (64비트): [android-studio-ide-201.7199119-windows.exe](#) (939674592바이트)

Chrome OS: [android-studio-ide-201.7199119-cros.deb](#) (778710260바이트)

Mac: [android-studio-ide-201.7199119-mac.dmg](#) (919746098바이트)

SHA-256 체크섬

다음으로 안드로이드 스튜디오를 다운받겠습니다.

<https://developer.android.com/studio/archive.html>

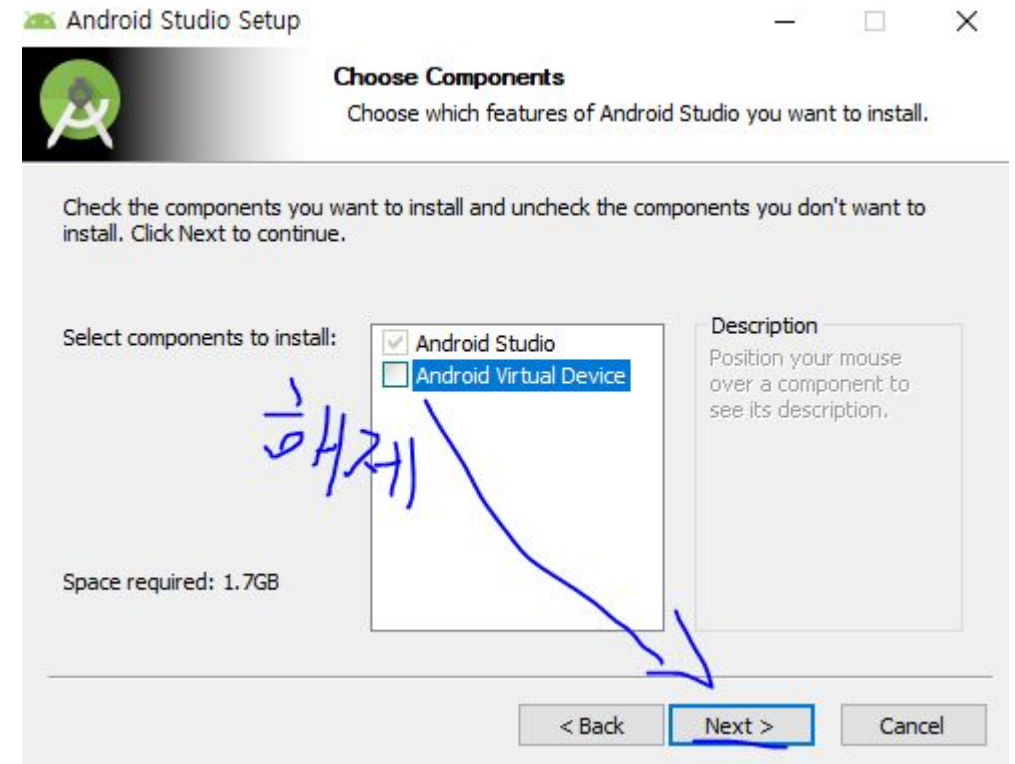
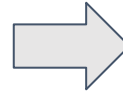
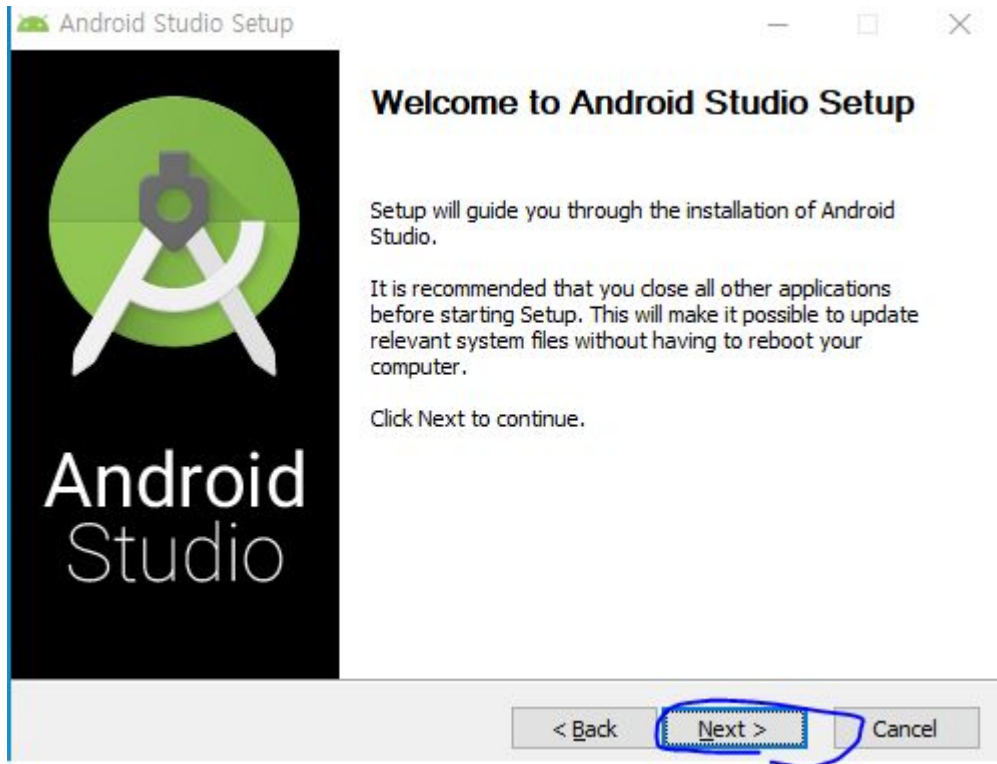
에 접속해서 약관 동의 후, 4.1버전, 윈도우 64bit 버전을 받아주세요.



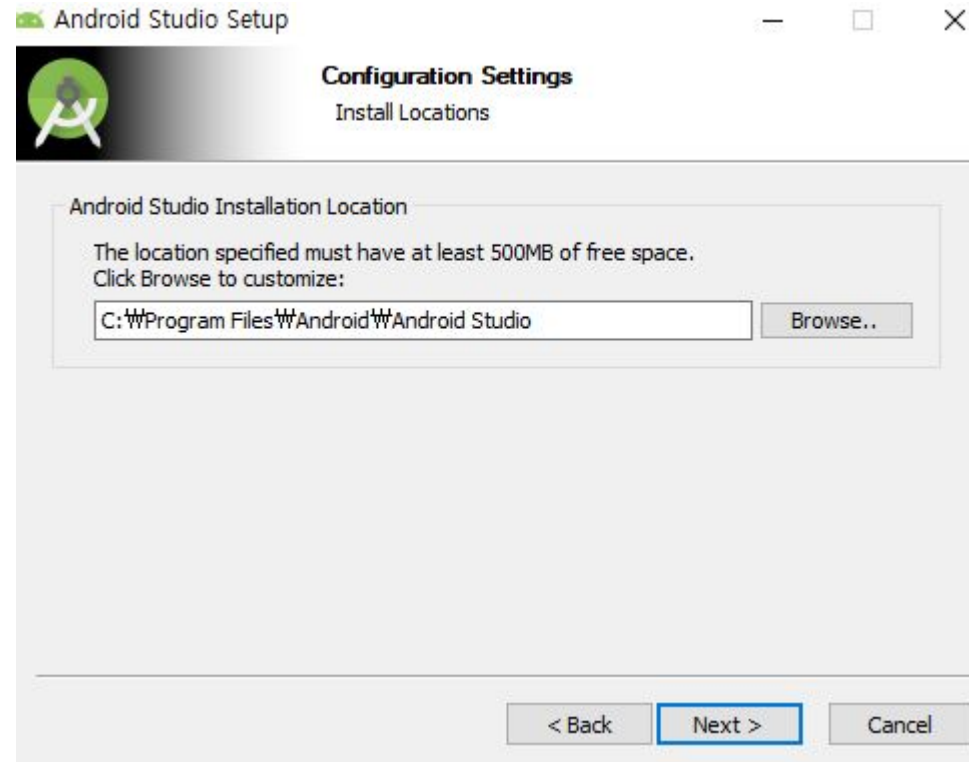
다운받은 파일은 약 896메가바이트입니다.

이제 위 파일을 실행합니다.

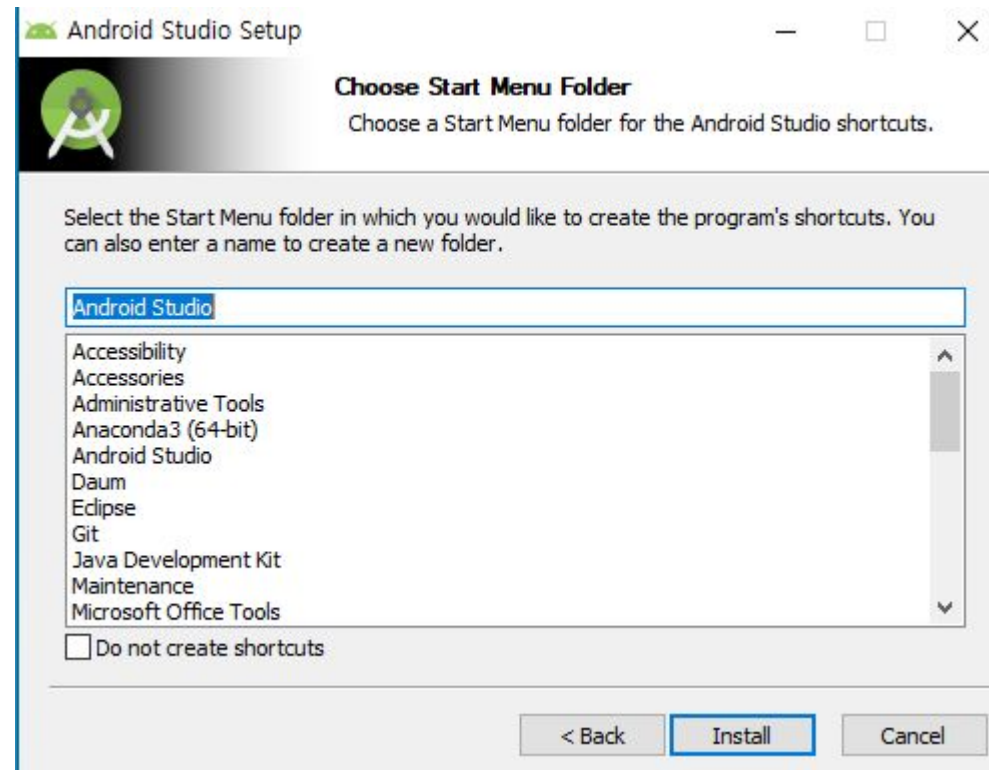
JavaJDK파일이 원래는 필요했지만 최신버전은 포함되어 나오니 신경쓰지 않으셔도 됩니다.



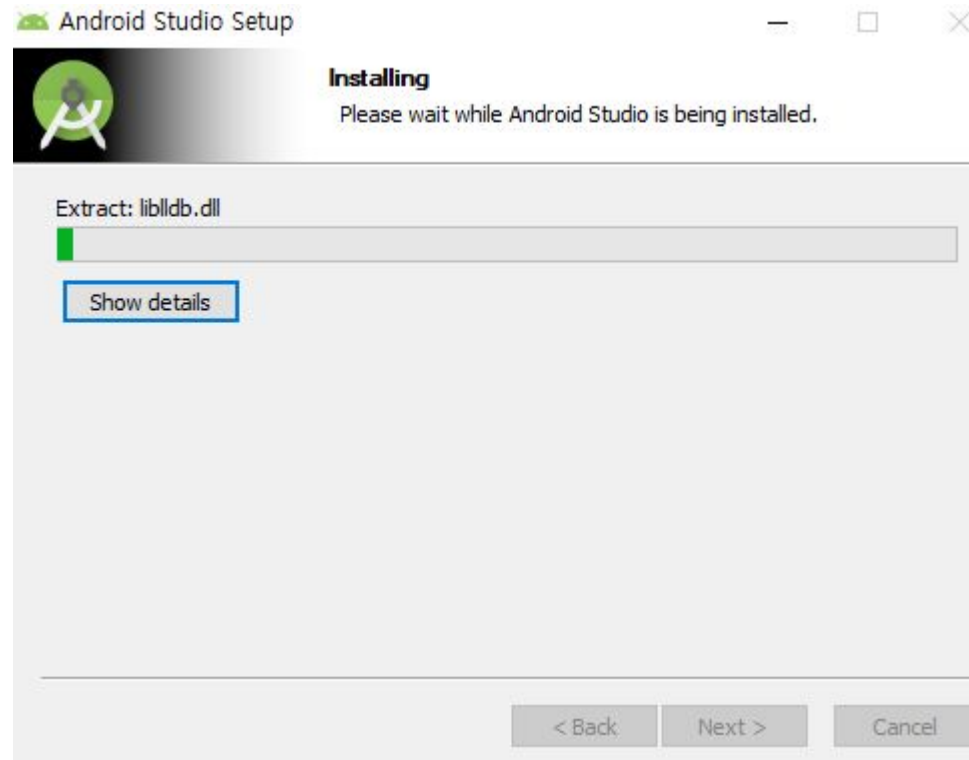
next를 한 번 누른 다음 AVD 설정을 해제하고 다음으로 넘어갑니다.
아까 설치한 HAXM이 AVD설정을 대신할것이기 때문입니다.



폴더는 특별한 사유가 없다면 기본 폴더 그대로 설치합니다.

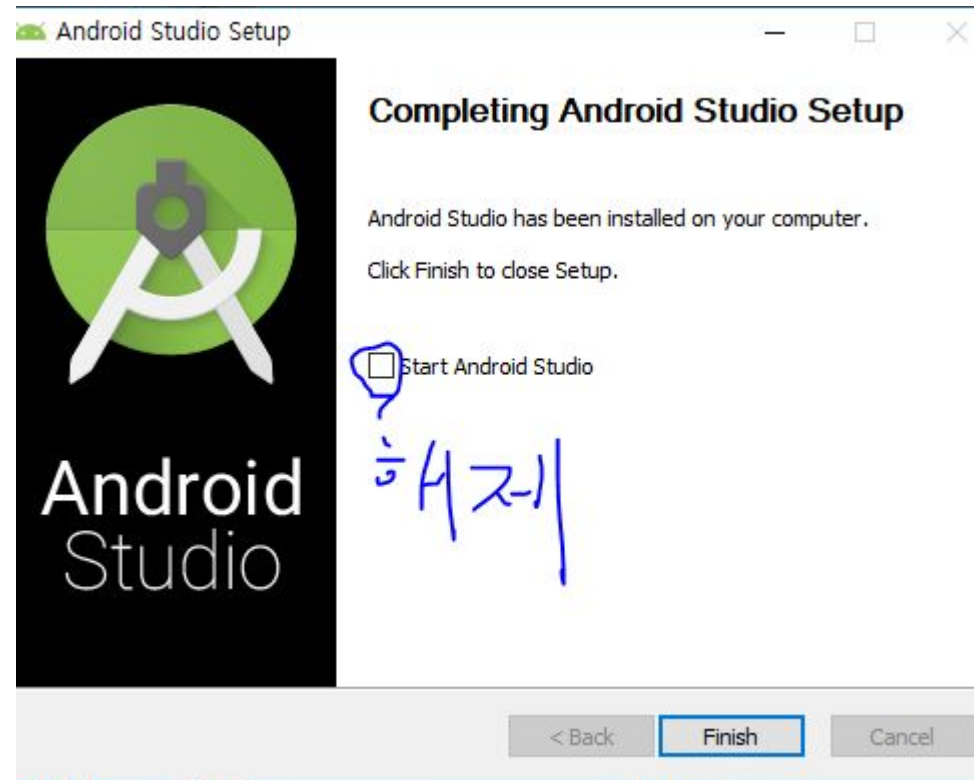


메뉴 폴더 역시 기본값으로 두고 다음으로 넘어갑니다.

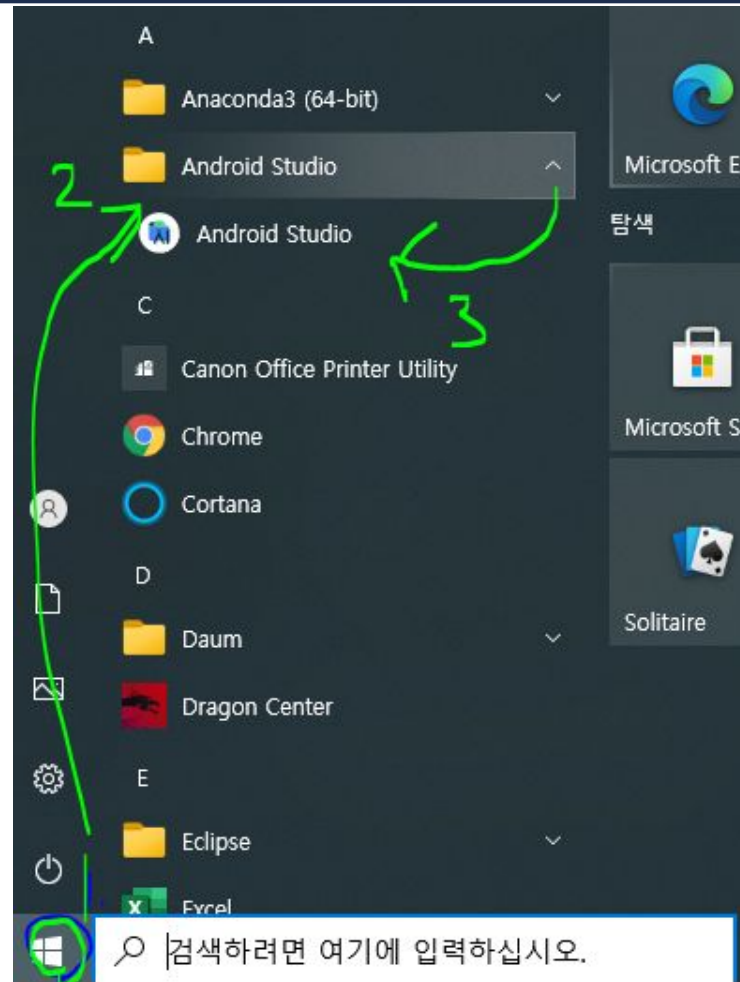


install을 눌러 설치를 진행합니다.

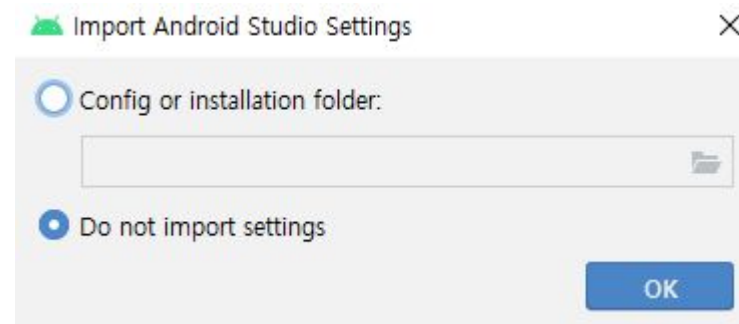
완료되면 Next를 누릅니다.



설치 완료 화면에서 Start Android Studio를 체크풀고 마무리합니다.

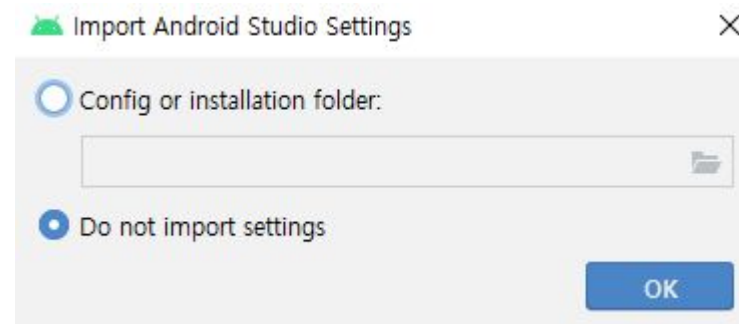


설치가 끝났으면 안드로이드 스튜디오의 기본 설정을 시작합니다.
위와 같이 하단 윈도우버튼 -> 안드로이드 스튜디오 폴더 -> 실행합니다.



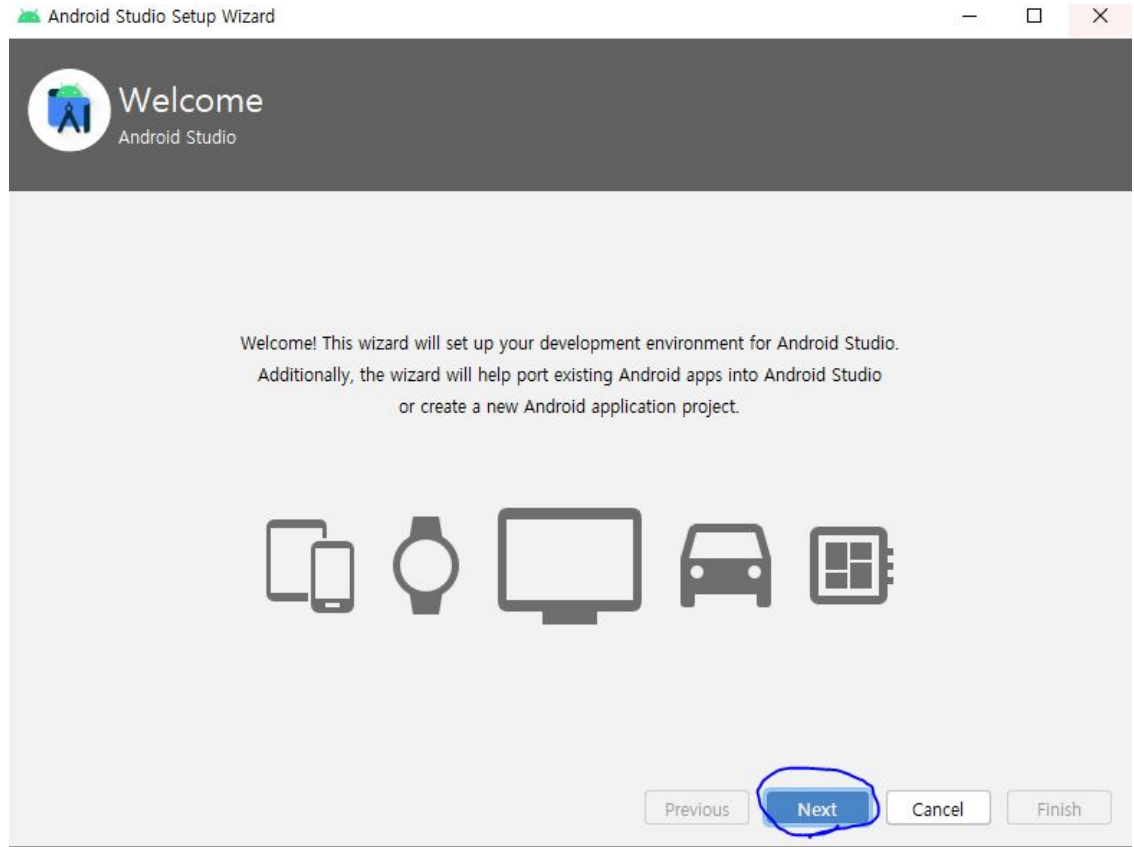
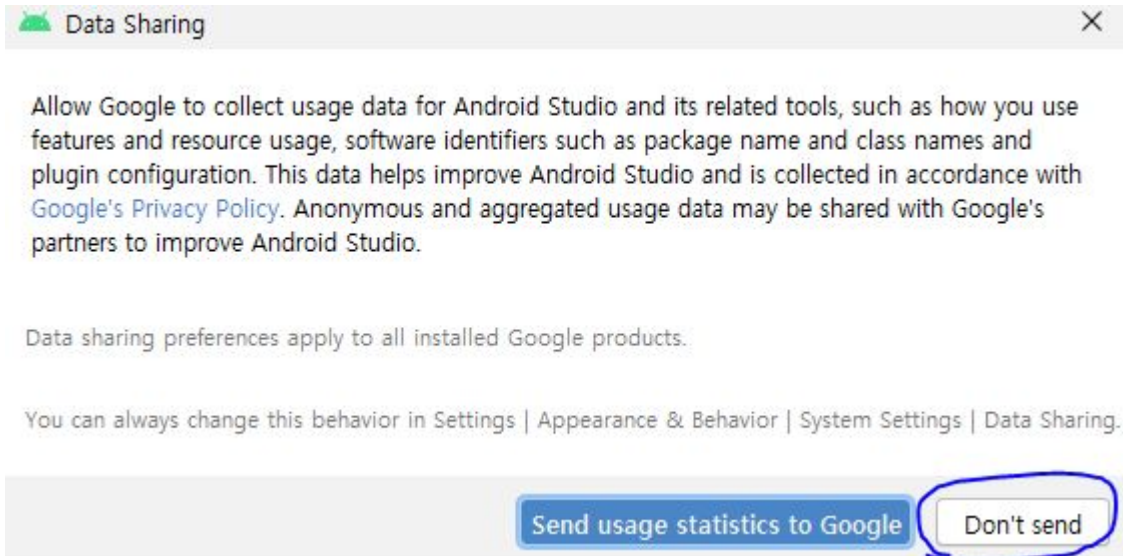
실행시 먼저 이 PC에 처음으로 안드로이드 스튜디오를 설치하는 경우는 위와 같이 **import settings**가 나옵니다.

Do not impoert settings를 선택해주세요.

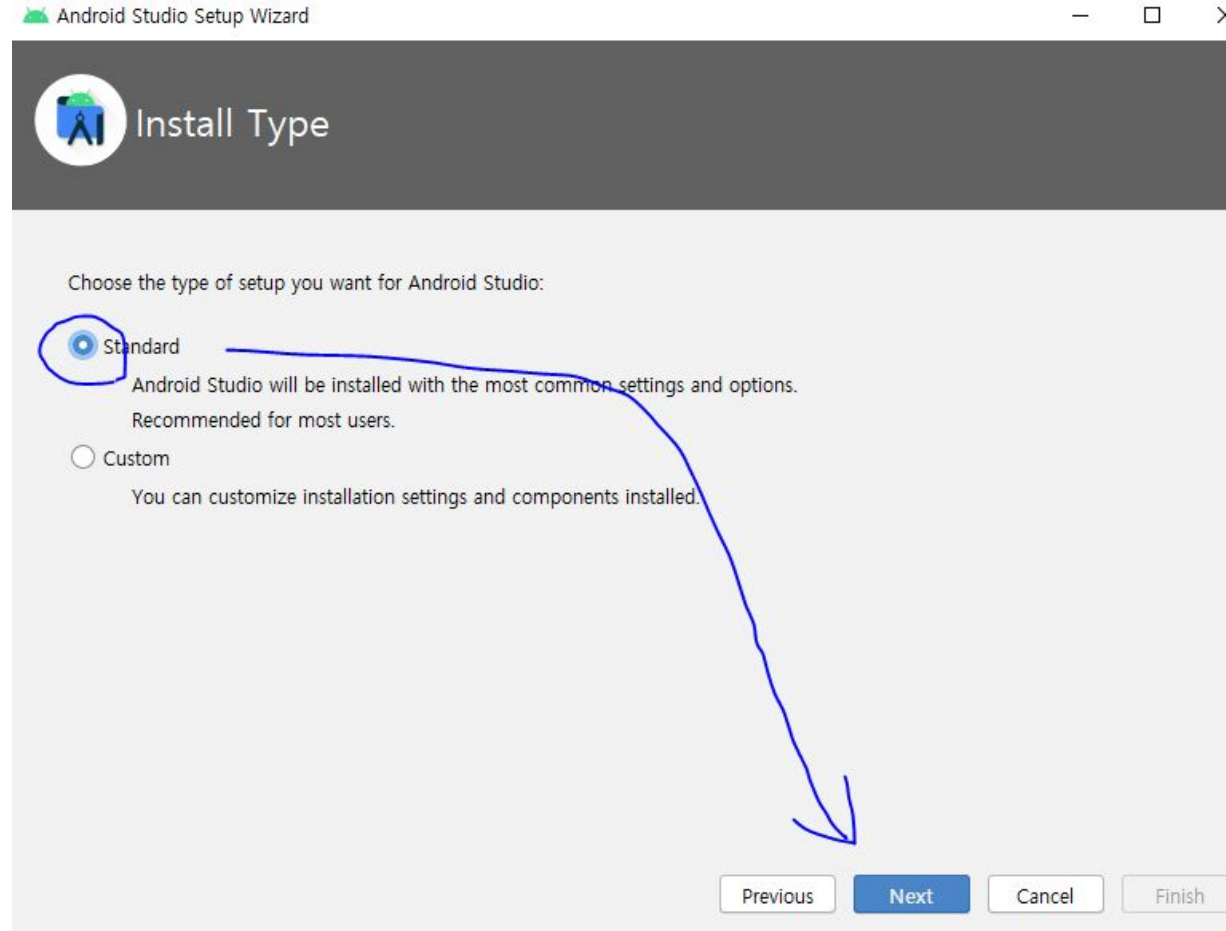


실행시 먼저 이 PC에 처음으로 안드로이드 스튜디오를 설치하는 경우는 위와 같이 **import settings**가 나옵니다.

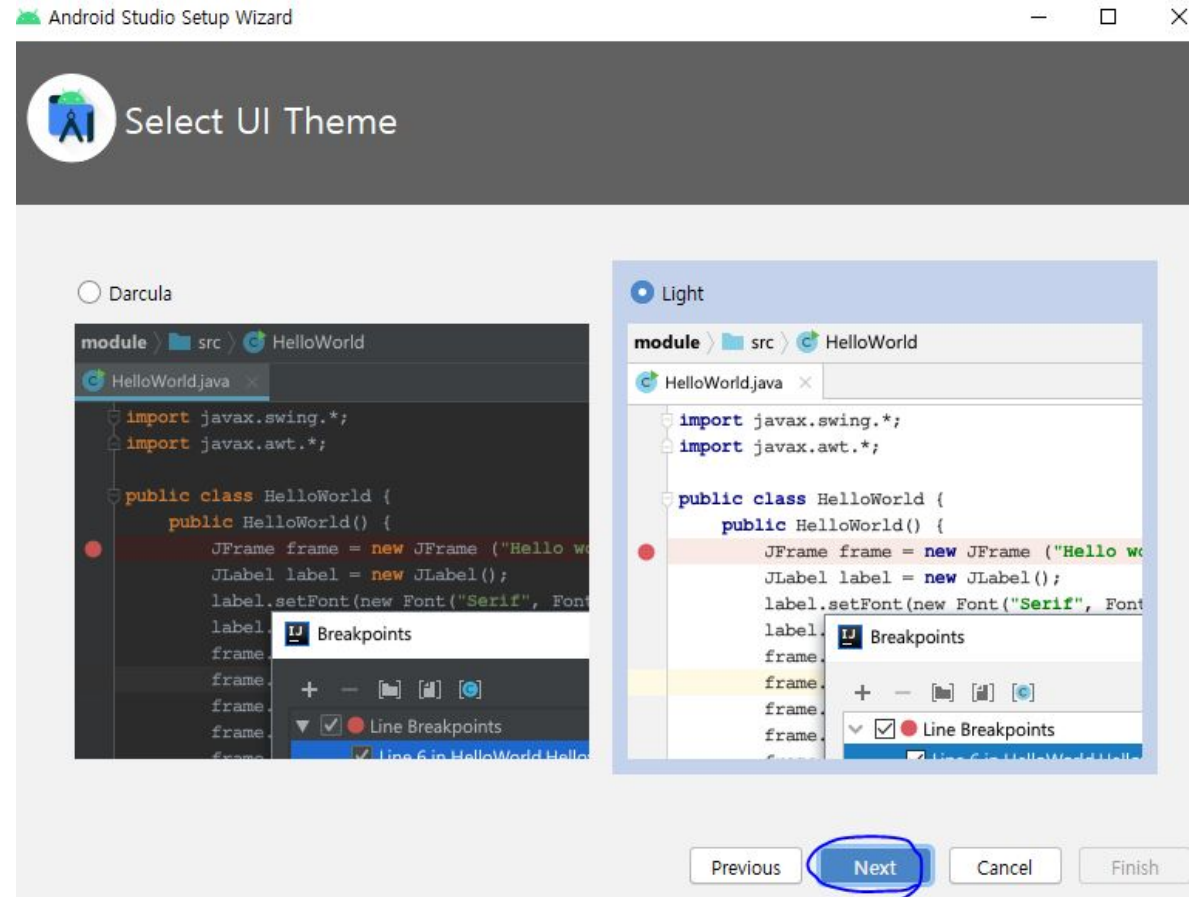
Do not impoert settings를 선택해주세요.



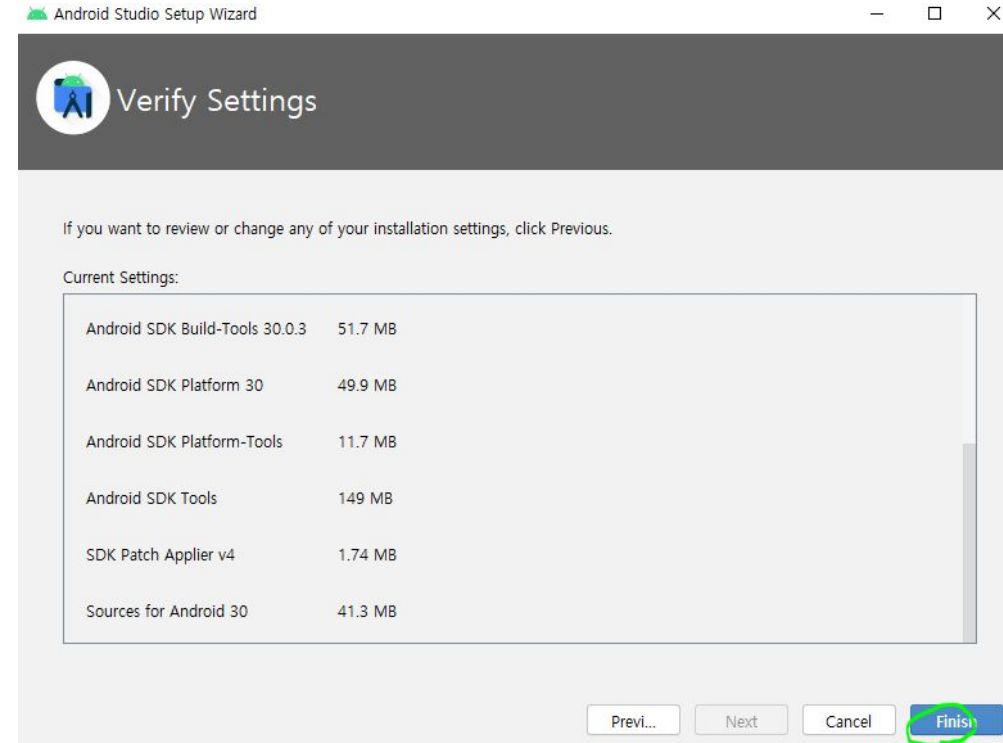
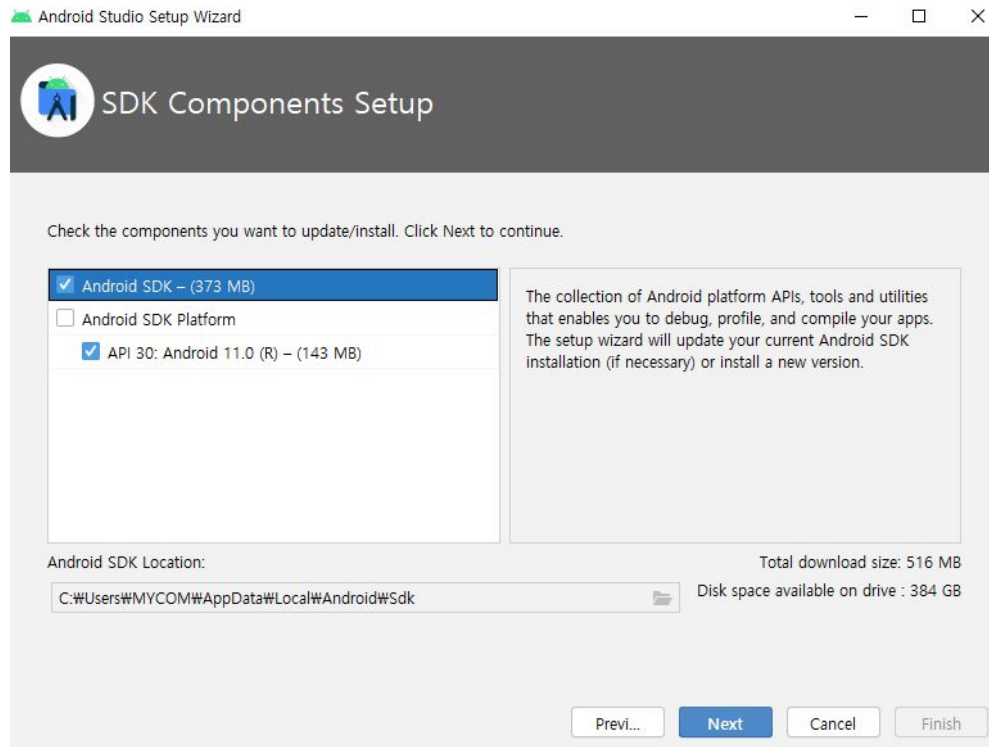
데이터 공유는 전송하지 않으면 됩니다.
다음 설치 마법사는 **Next**를 먼저 누릅니다.



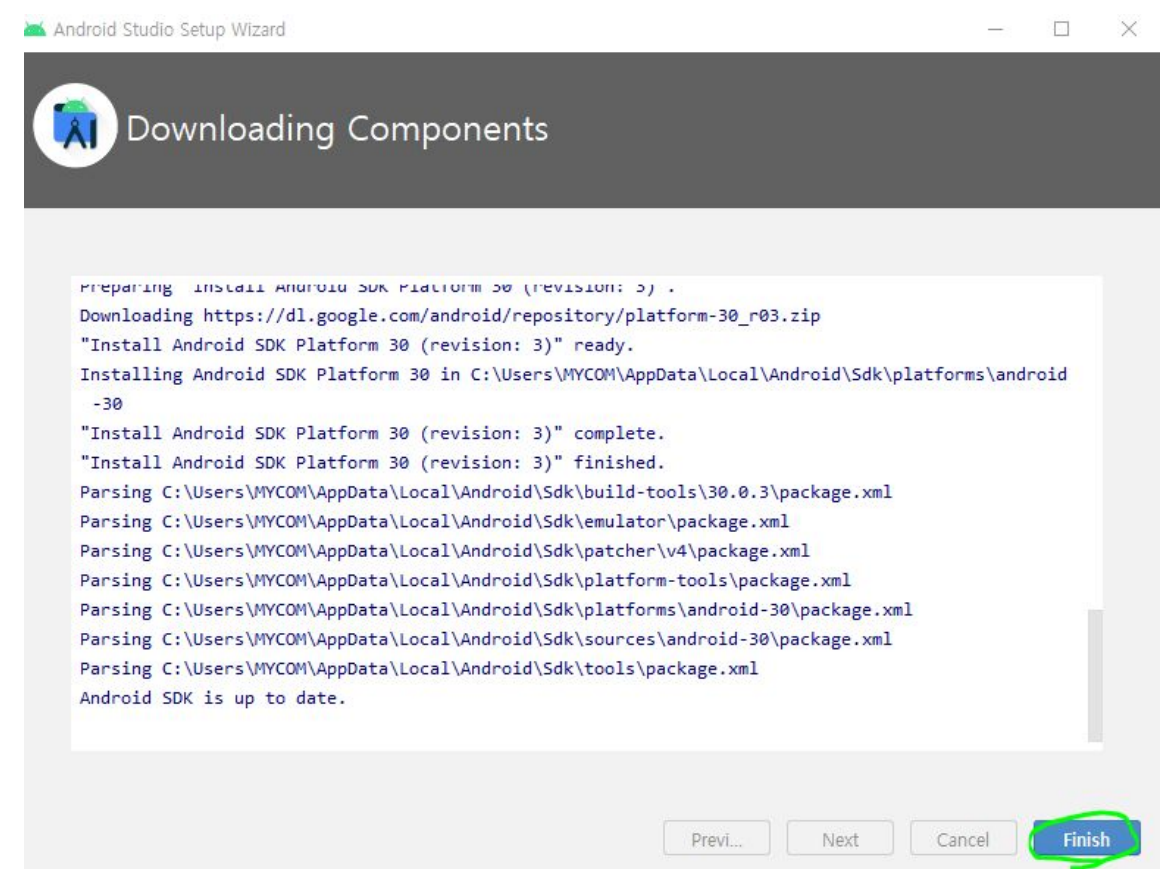
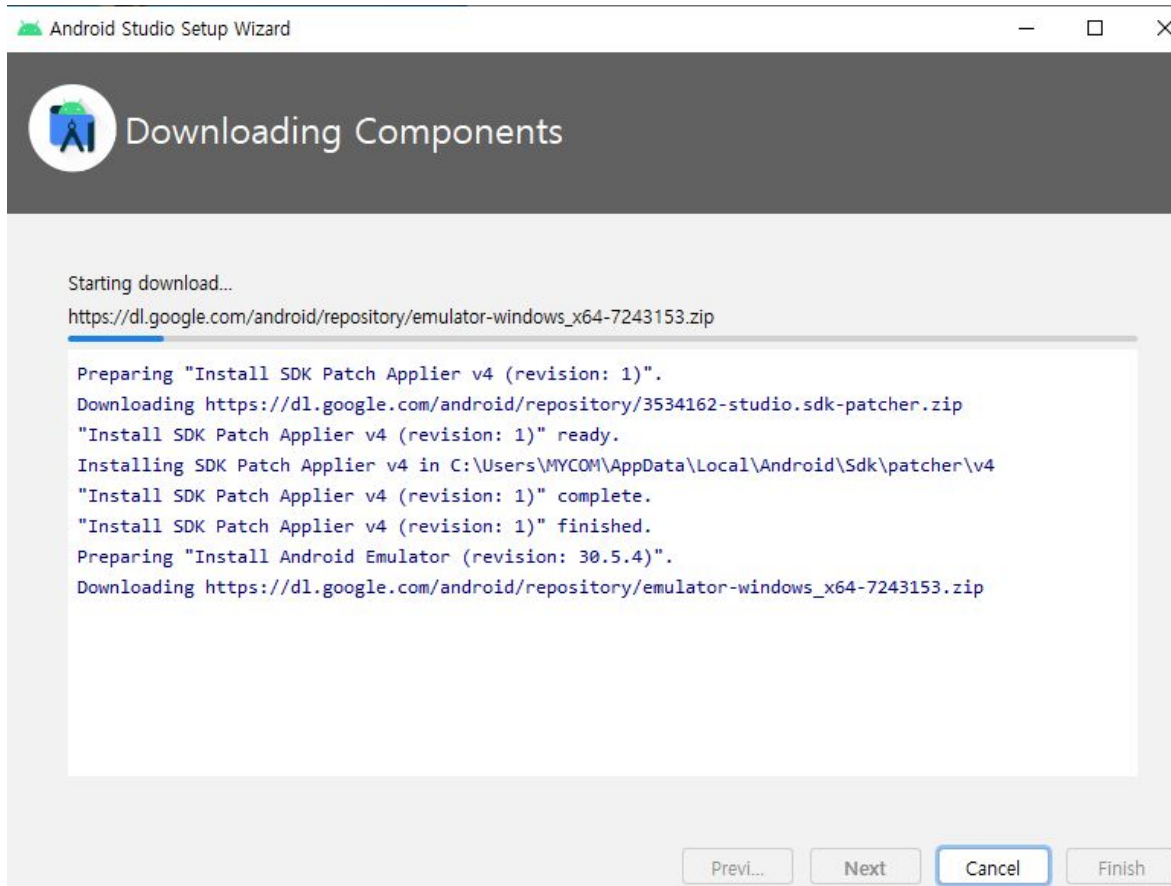
설치 유형은 Standard를 그대로 둡니다.



UI 테마도 그냥 편한대로 두시면 됩니다.

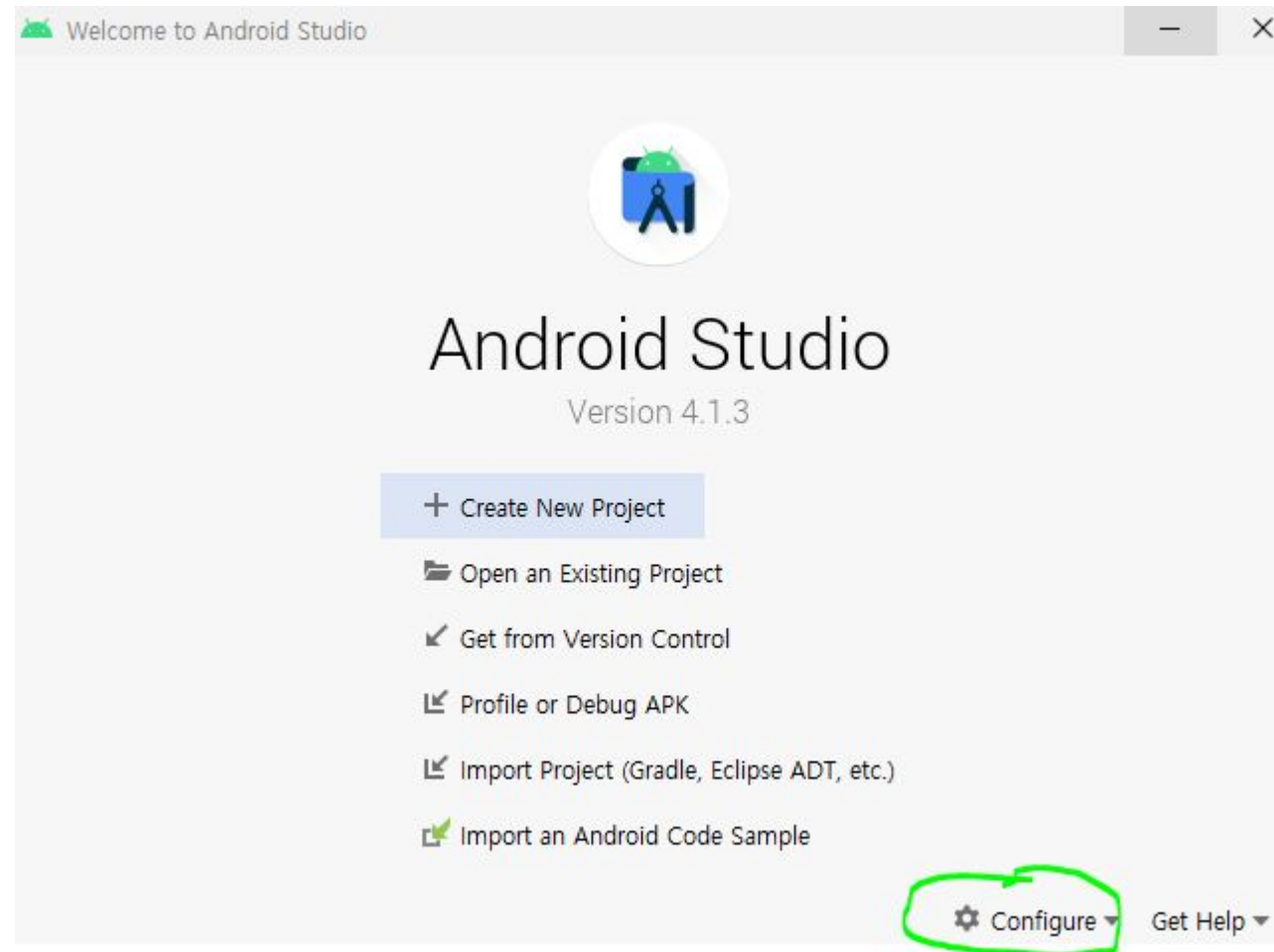


이후 SDK가 설치된적이 없다면 설치하라고 나옵니다.

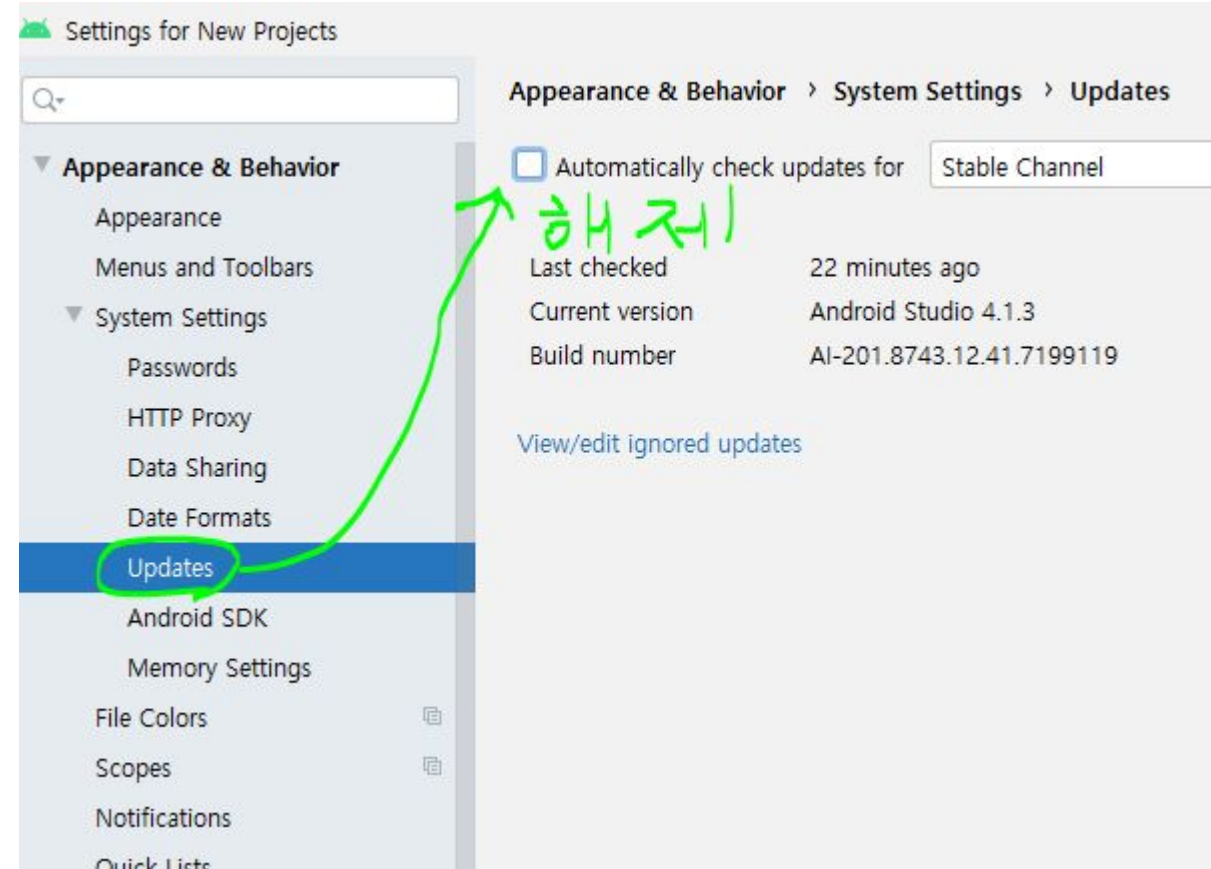
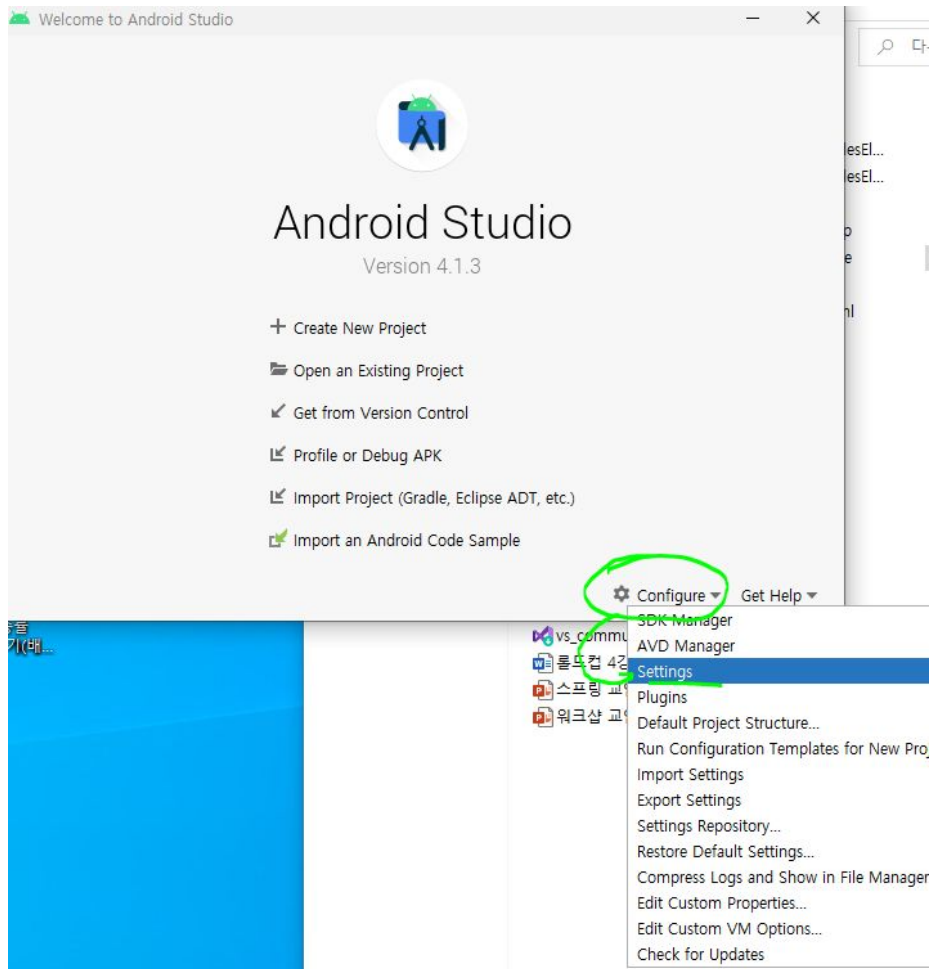


위와 같이 설치가 진행됩니다.

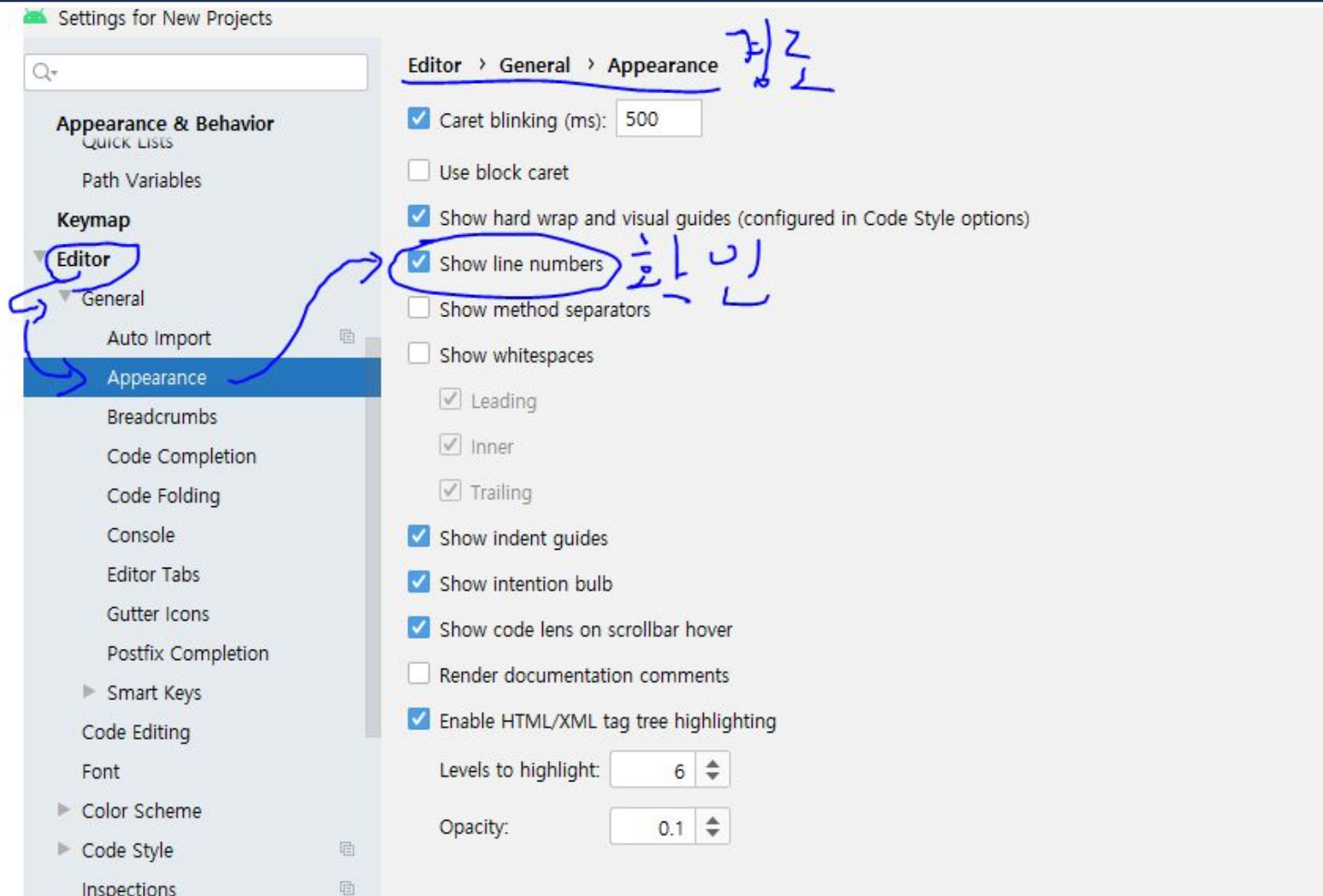
설치가 완료되면 Finish를 눌러 마무리합니다.



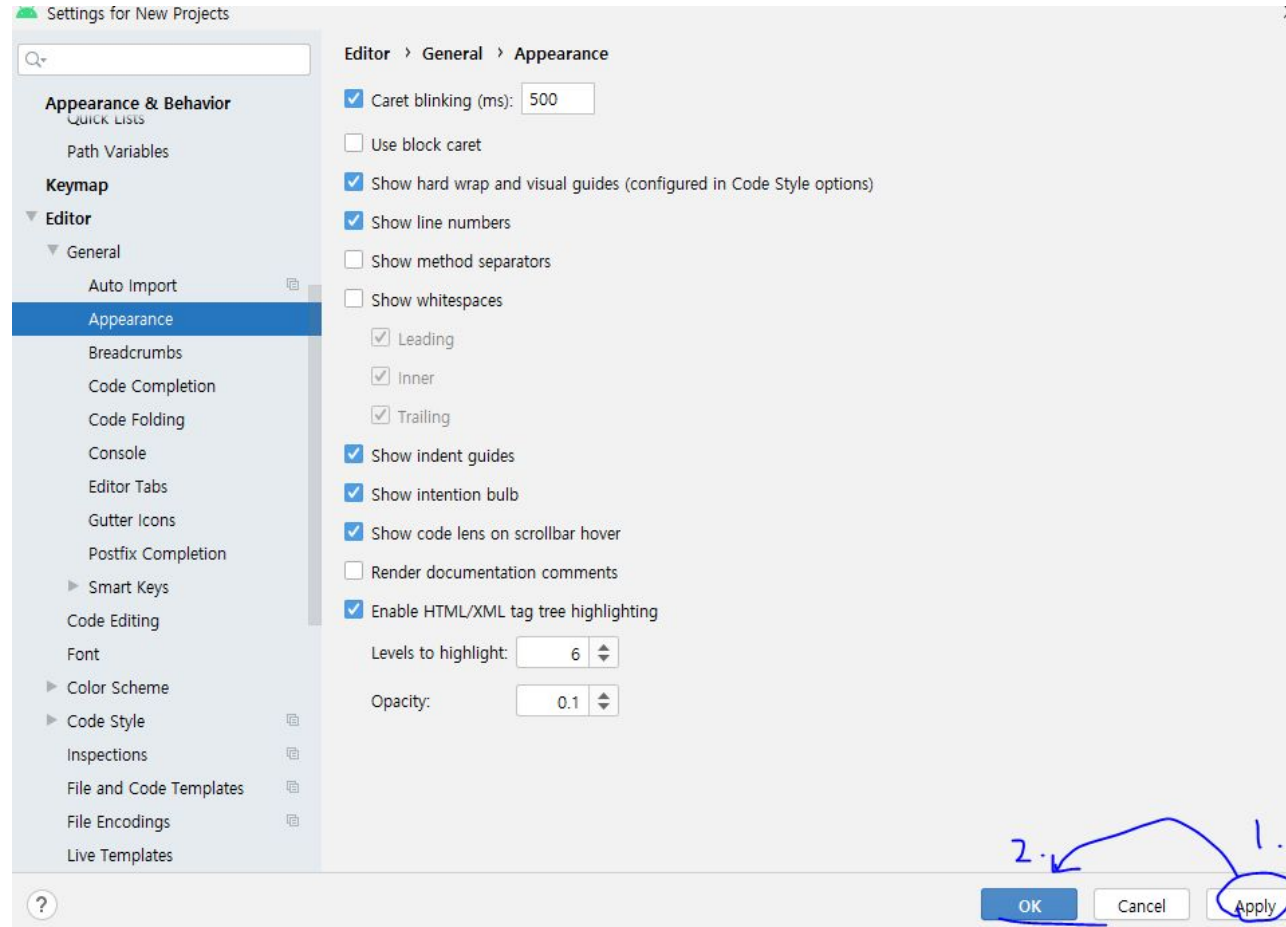
SDK설치 후 위와 같은 화면이 나오는데, 하단 **Configure**를 눌러서 몇 가지 옵션을 바꿔야 합니다.



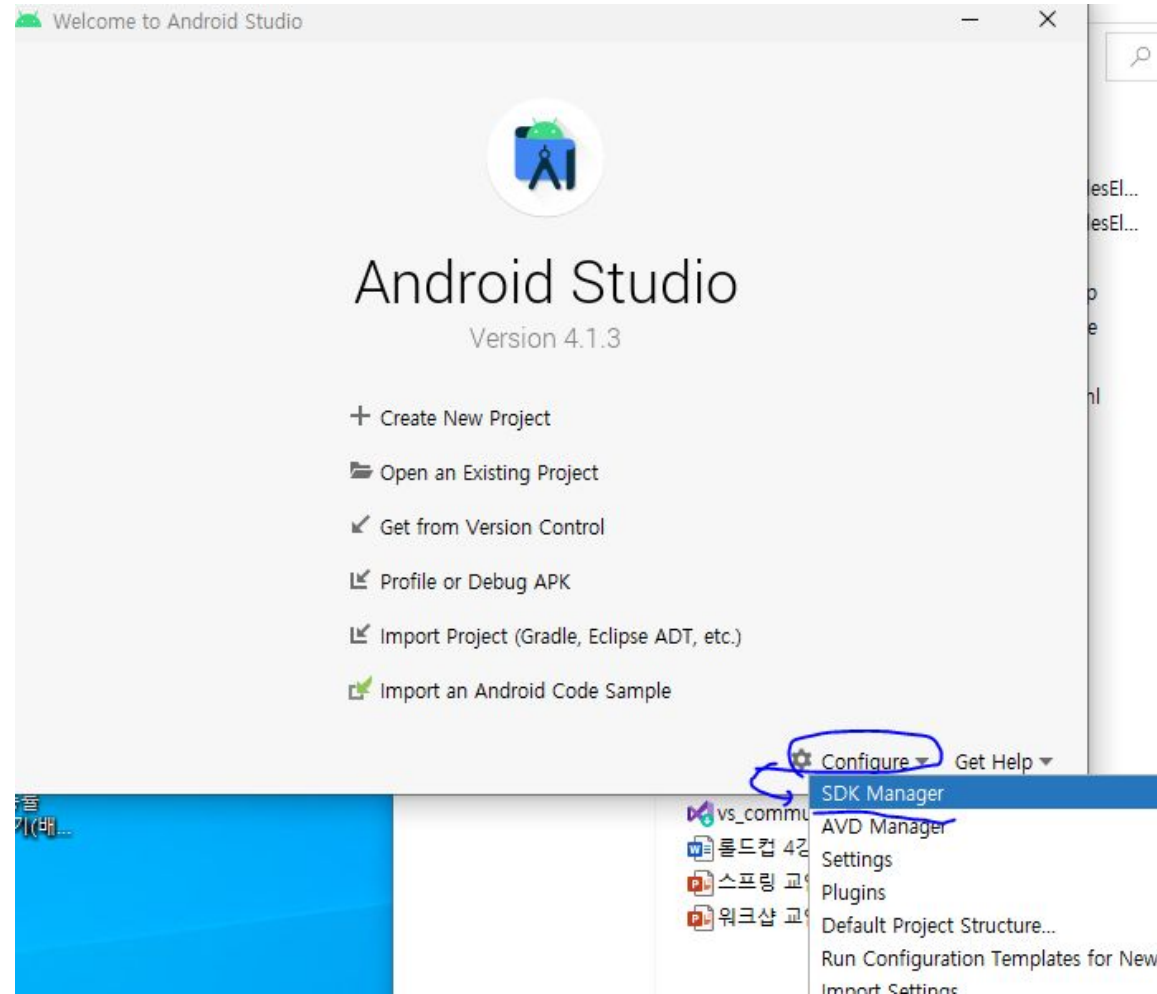
Setting를 들어가서 자동 업데이트기능을 끄겠습니다.



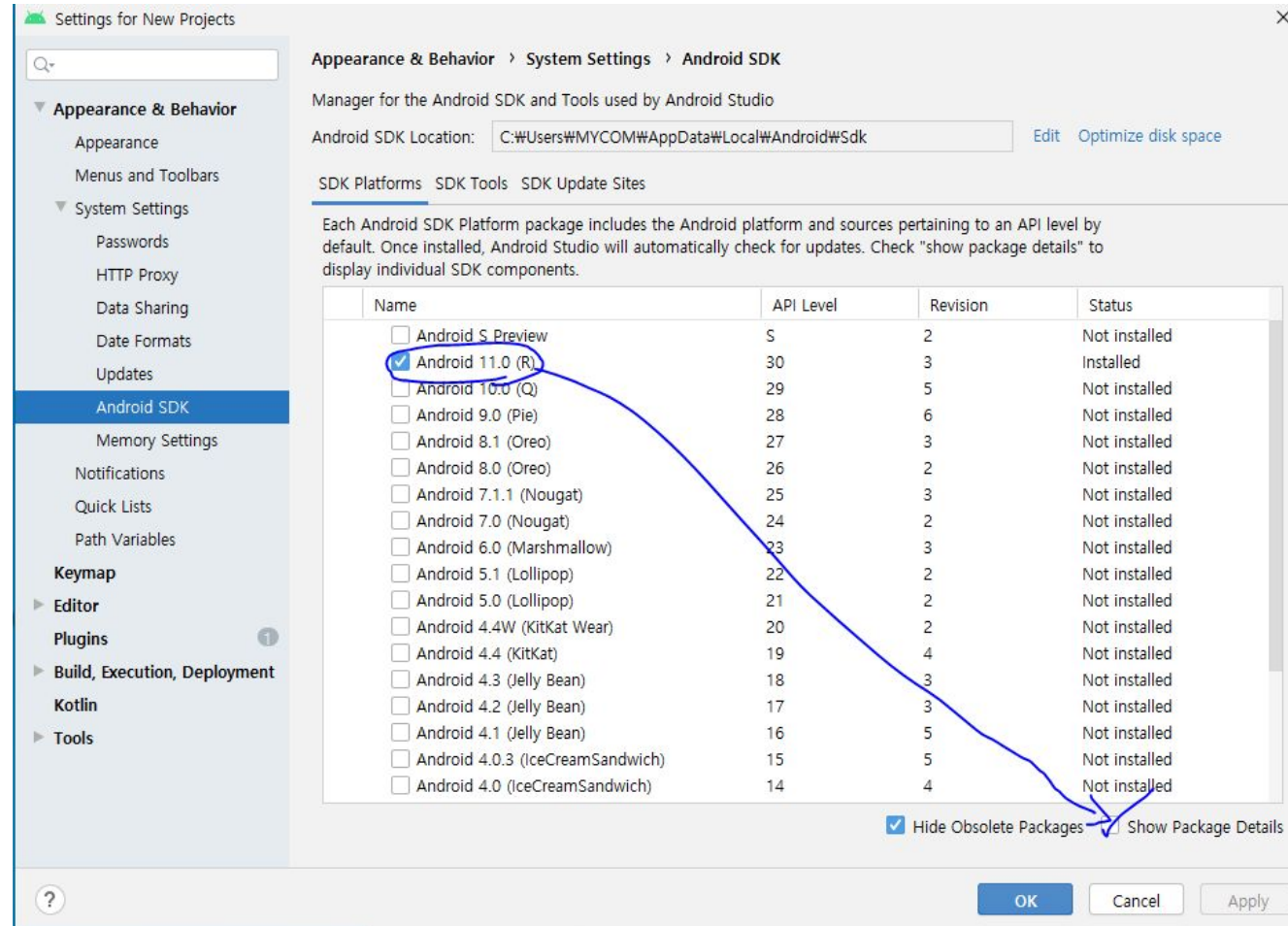
다음 좌측 Editor -> Appearance -> Show line numbers가 체크되어 있는지 확인부터 해 주세요.



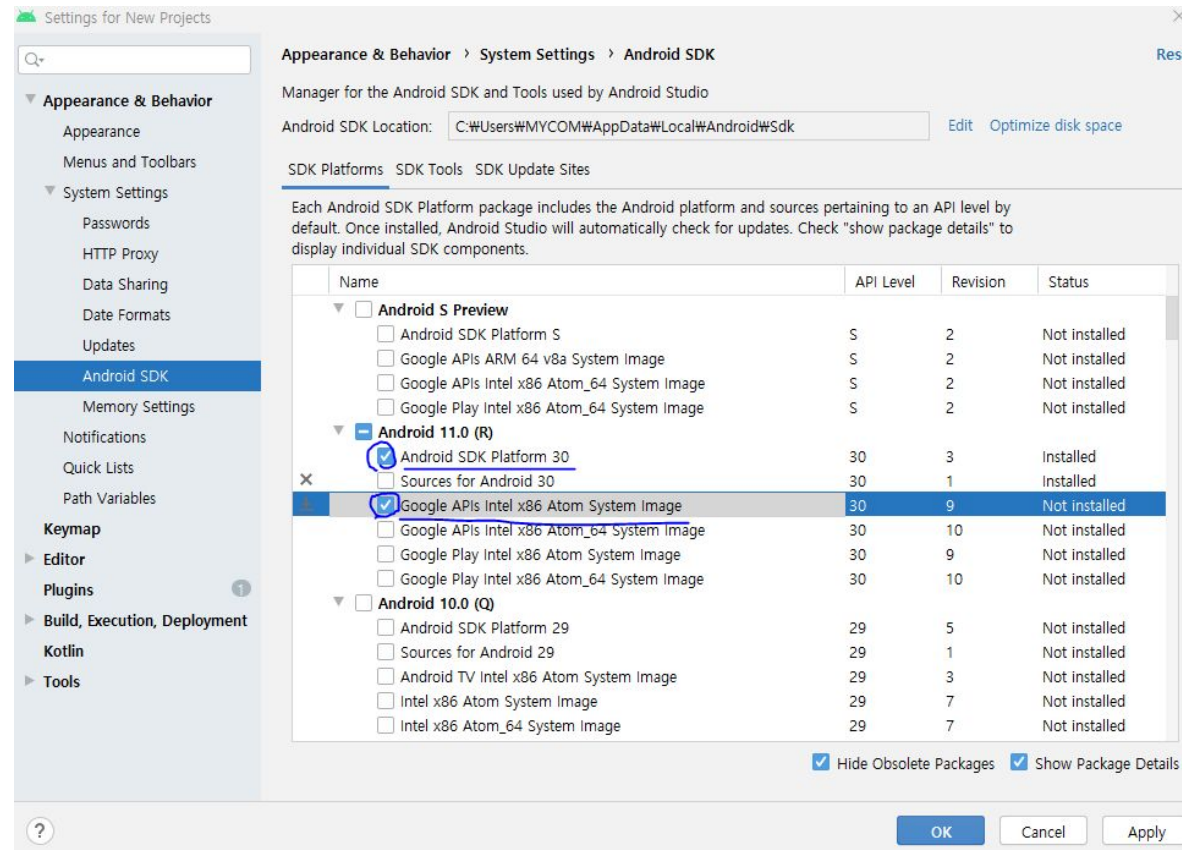
다음 Apply를 먼저 선택한 후 OK를 눌러서 설정을 마칩니다.



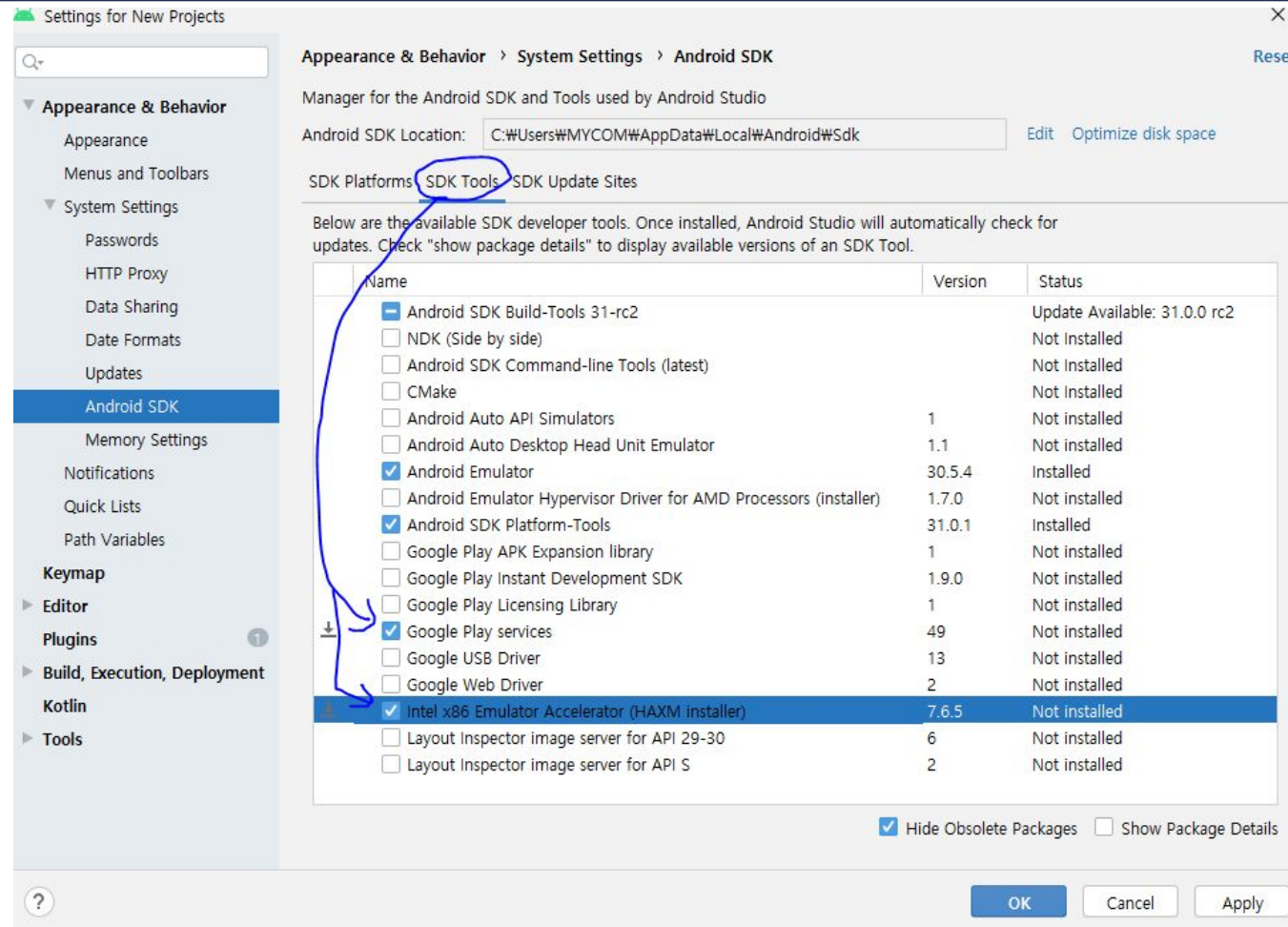
다음, 다시 **Configure**를 선택해 **SDK Manager**로 갑니다.
초기 설치에 더해 추가설치가 필요한 부분을 세팅합니다.



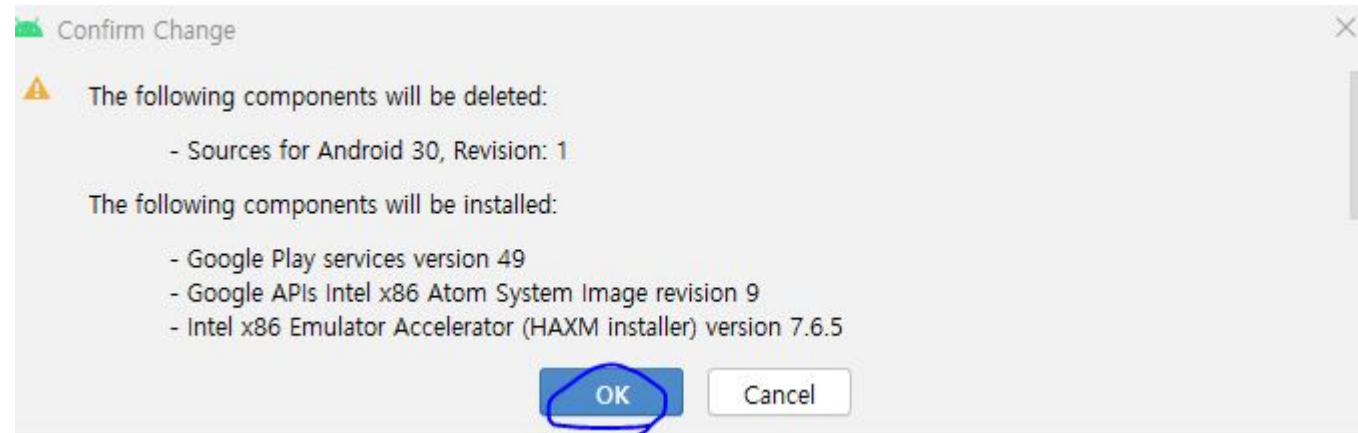
먼저 11.0버전이 설치되었는지 확인 후 Show Package Detail를 체크합니다.



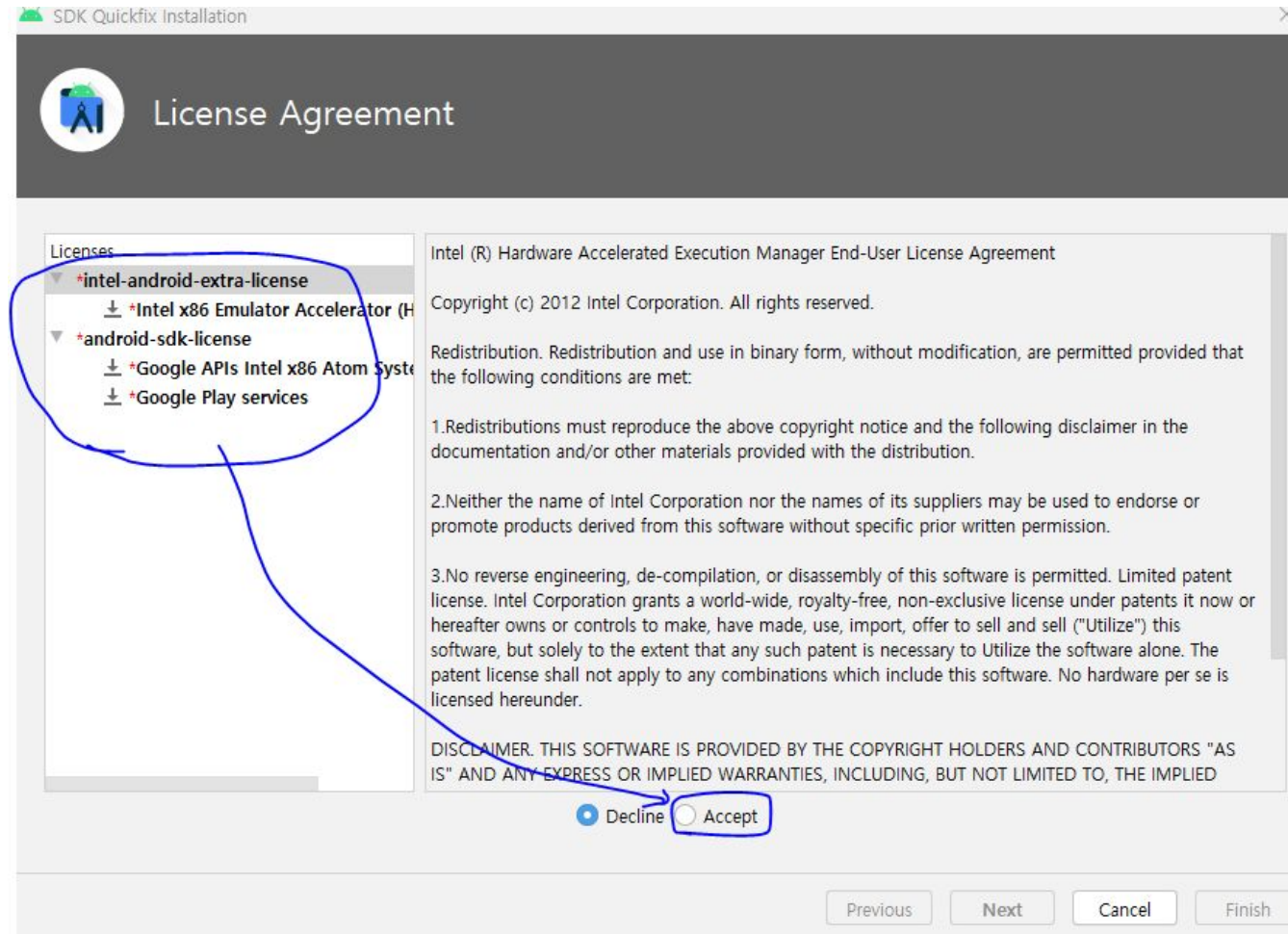
다음, Sources for Android 30은 체크를 풀고
한 칸 아래의 Google APIs intel x86 atom System image를 체크합니다.
OK는 누르지 말고 대기합니다.



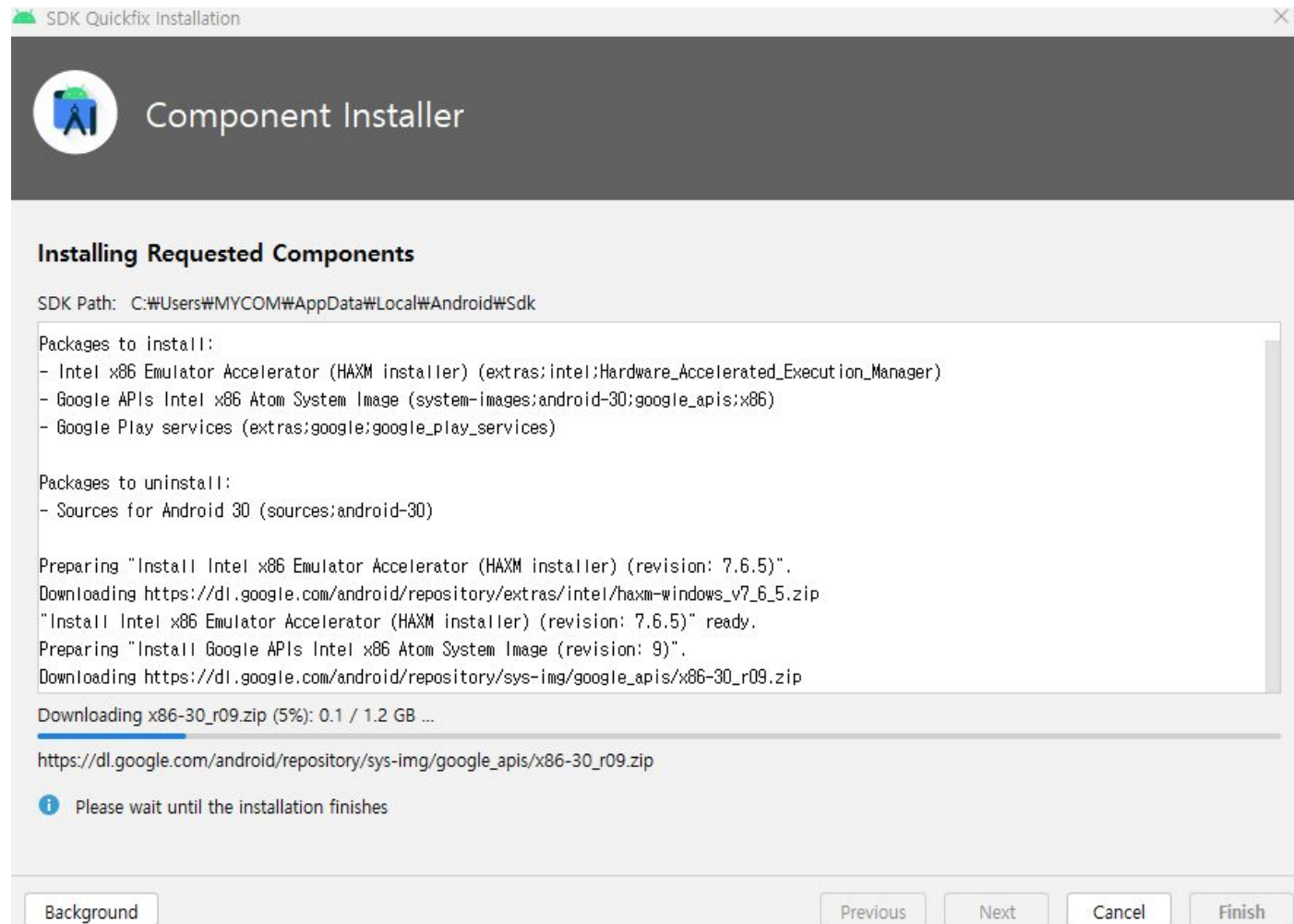
다음, SDK Tools로 이동한 다음
Google Play services, Intel x86 Emulator Accelerator(HAXM installer)
 두 개를 체크하고 OK를 누릅니다.



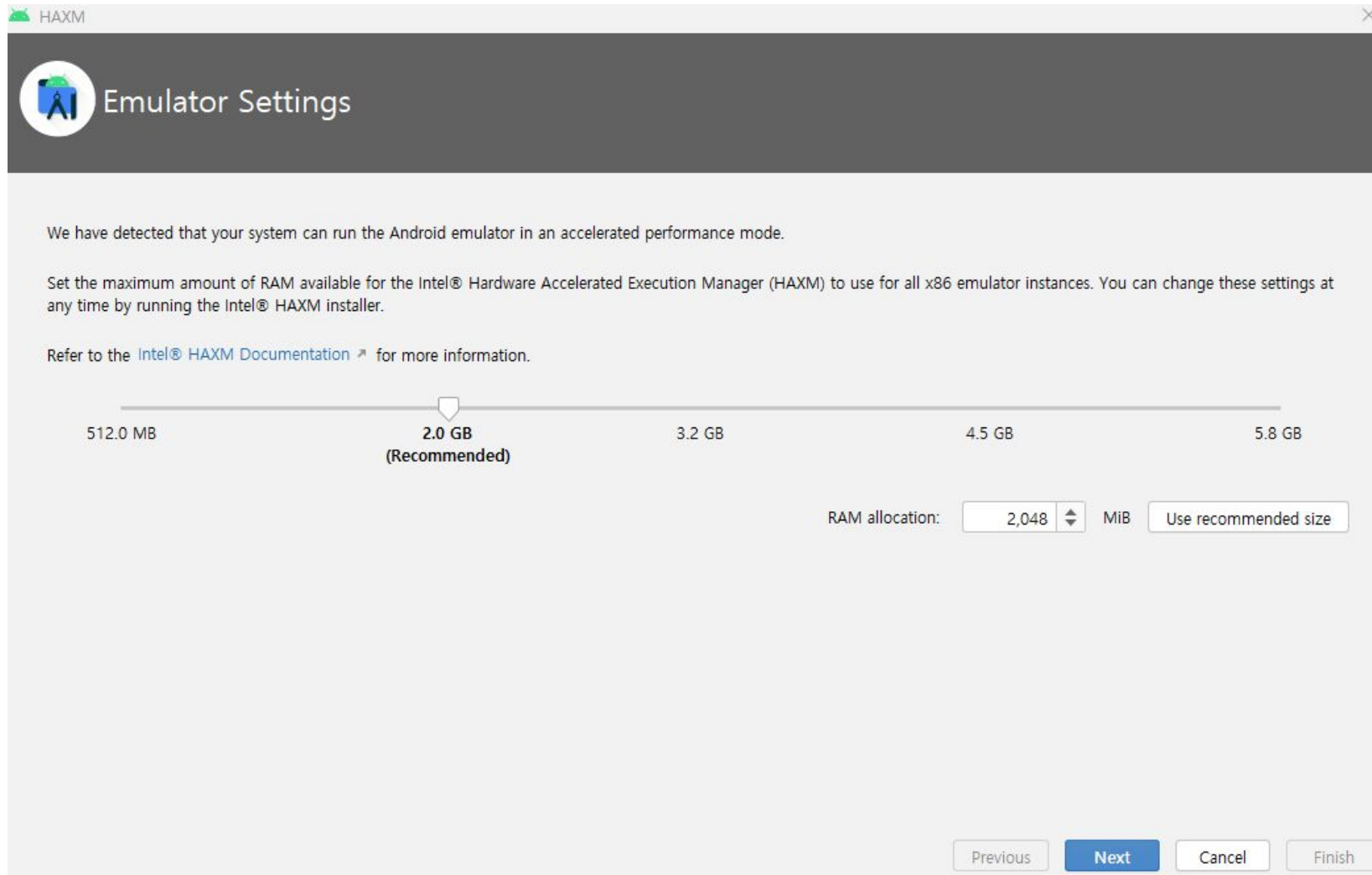
Confirm Change역시 **OK**를 체크합니다.



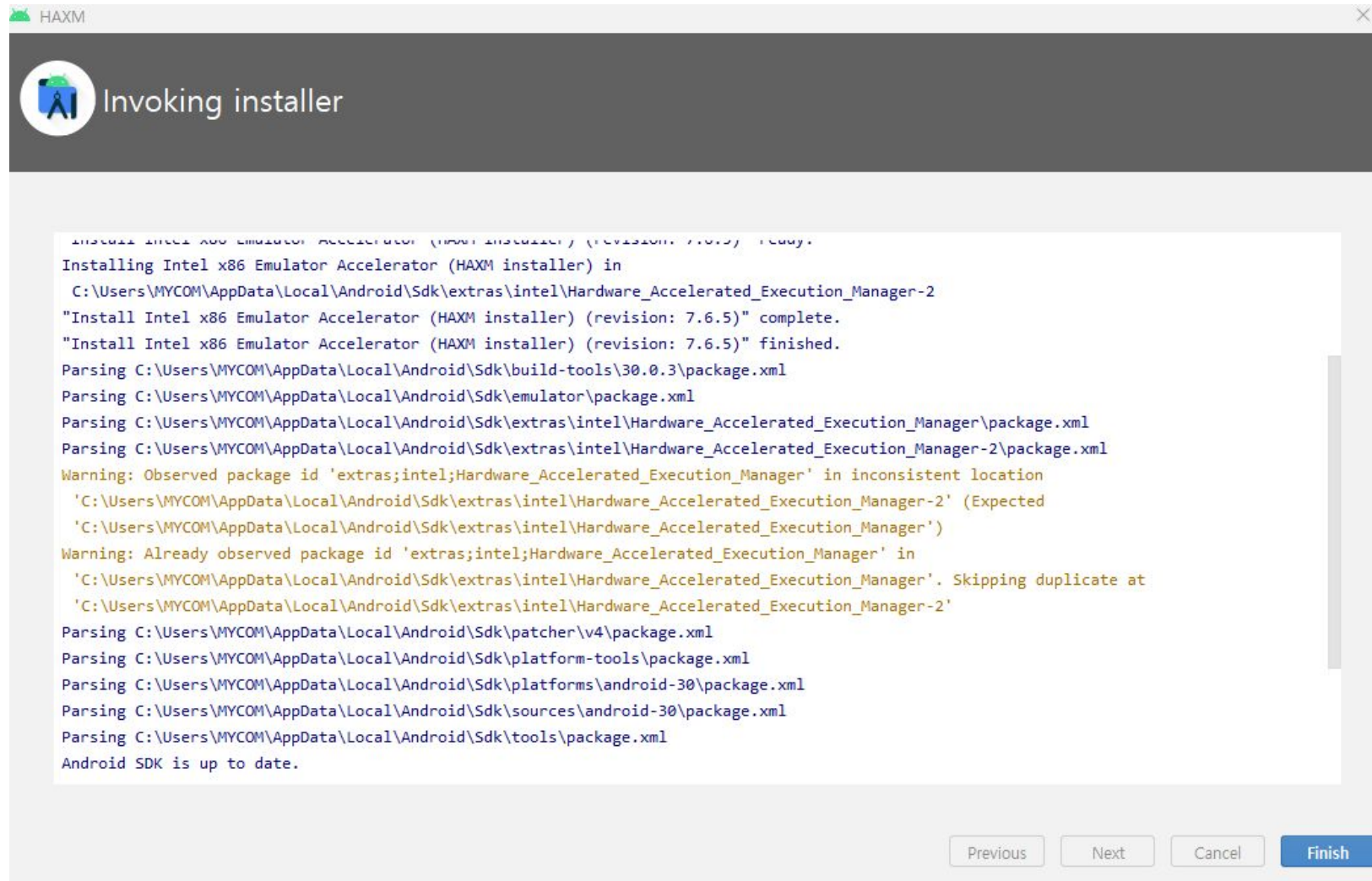
동그라미 친 모든 항목을 Accept로 바꿔주고 설치를 시작합니다.



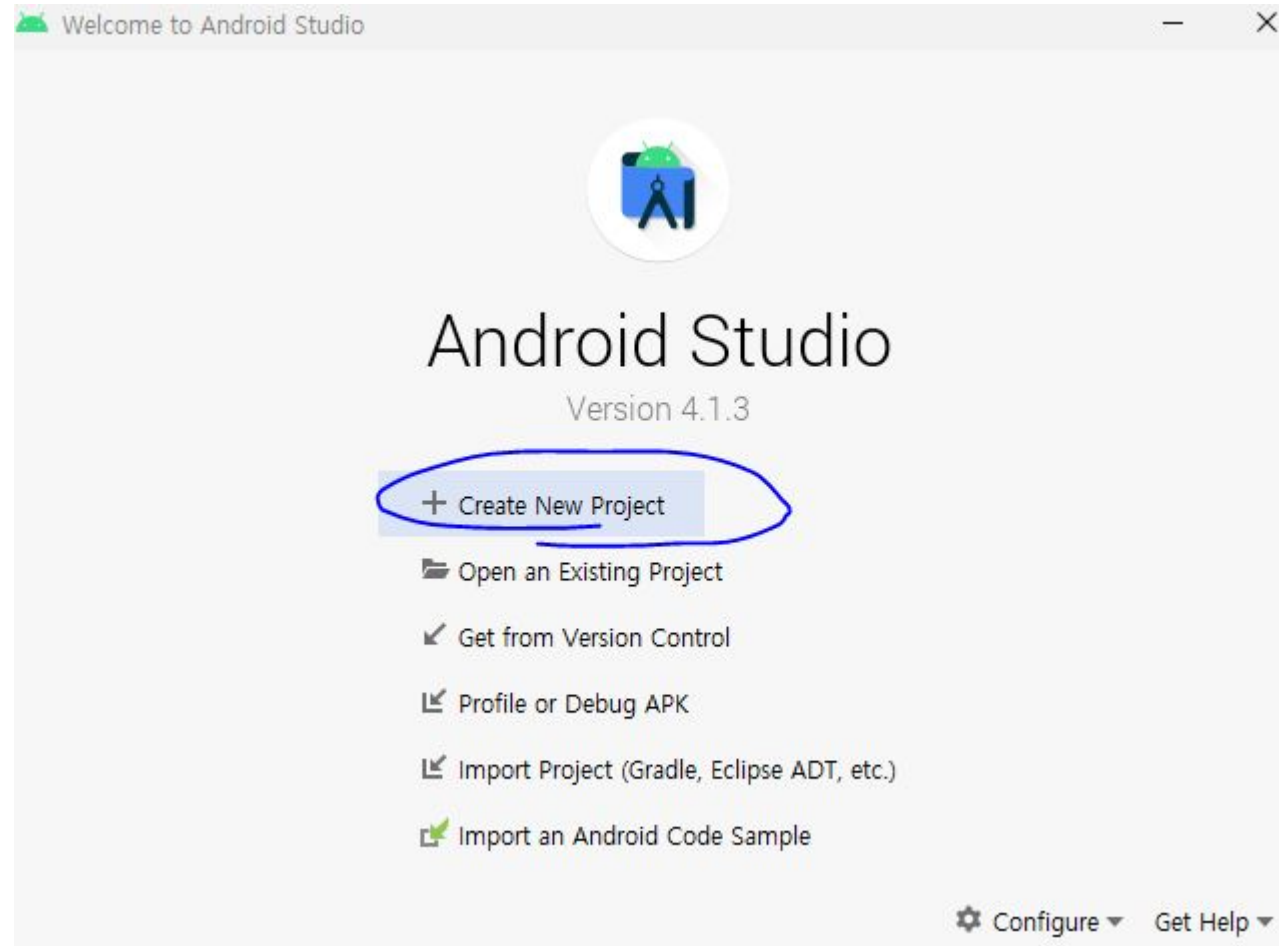
설치되는동안 기다리겠습니다.



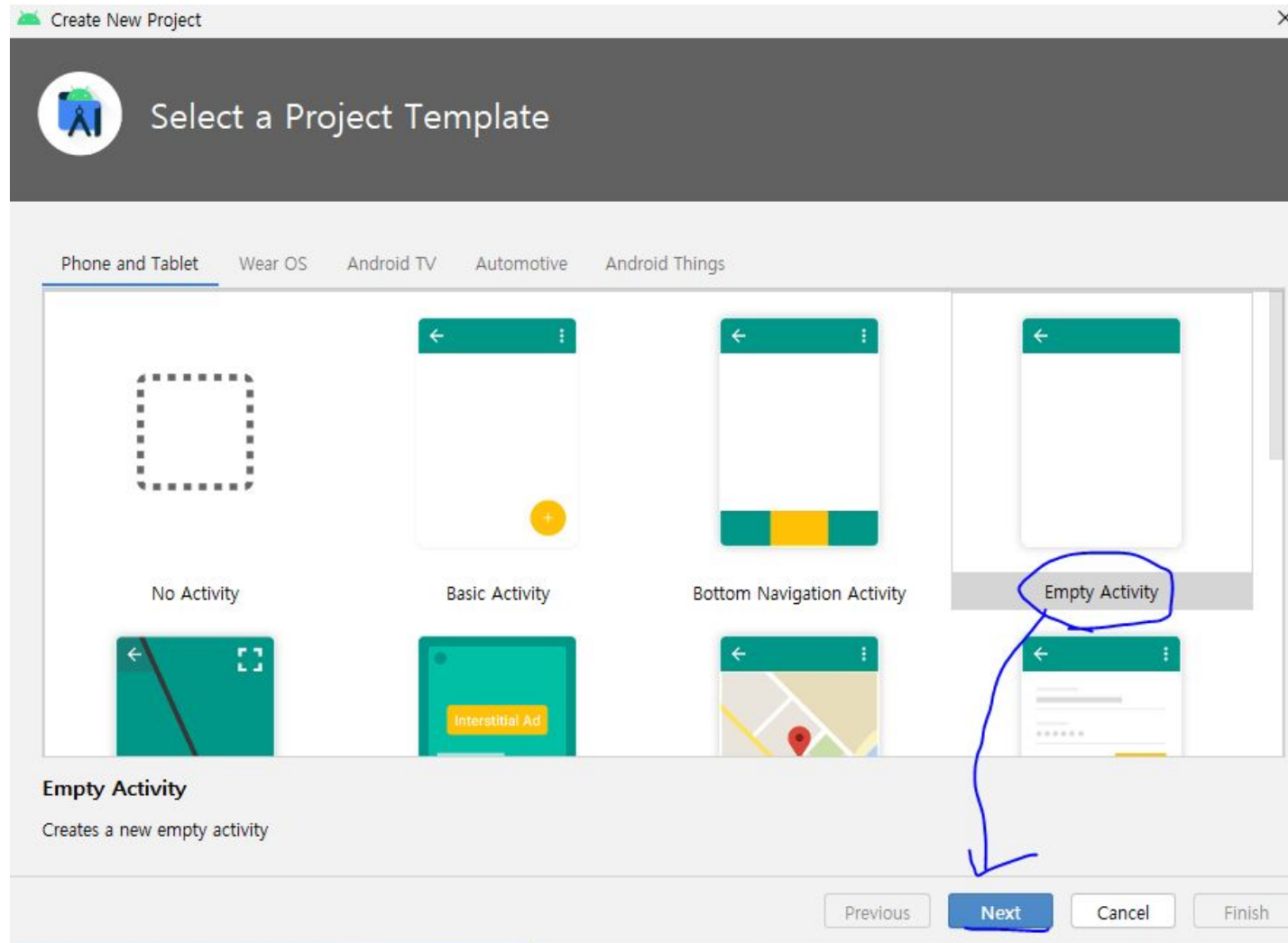
가상휴대폰에 할당할 메모리를 정해야 하는데 어차피 3GB이상 활용하려면 창을 여러개 띄워야 하므로 그냥 그대로 둡니다.



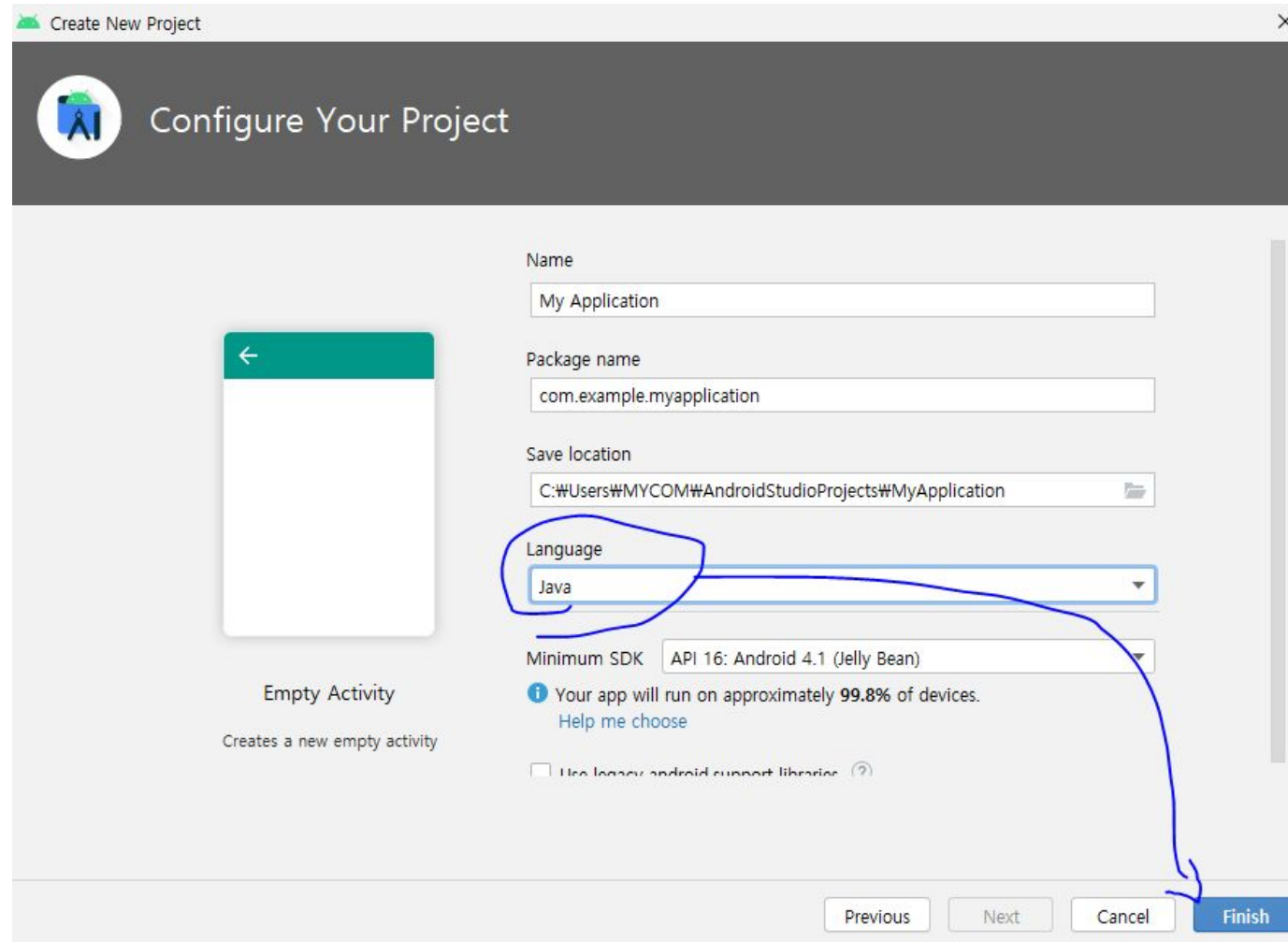
완료되면 finish를 눌러서 마무리 지어주세요.



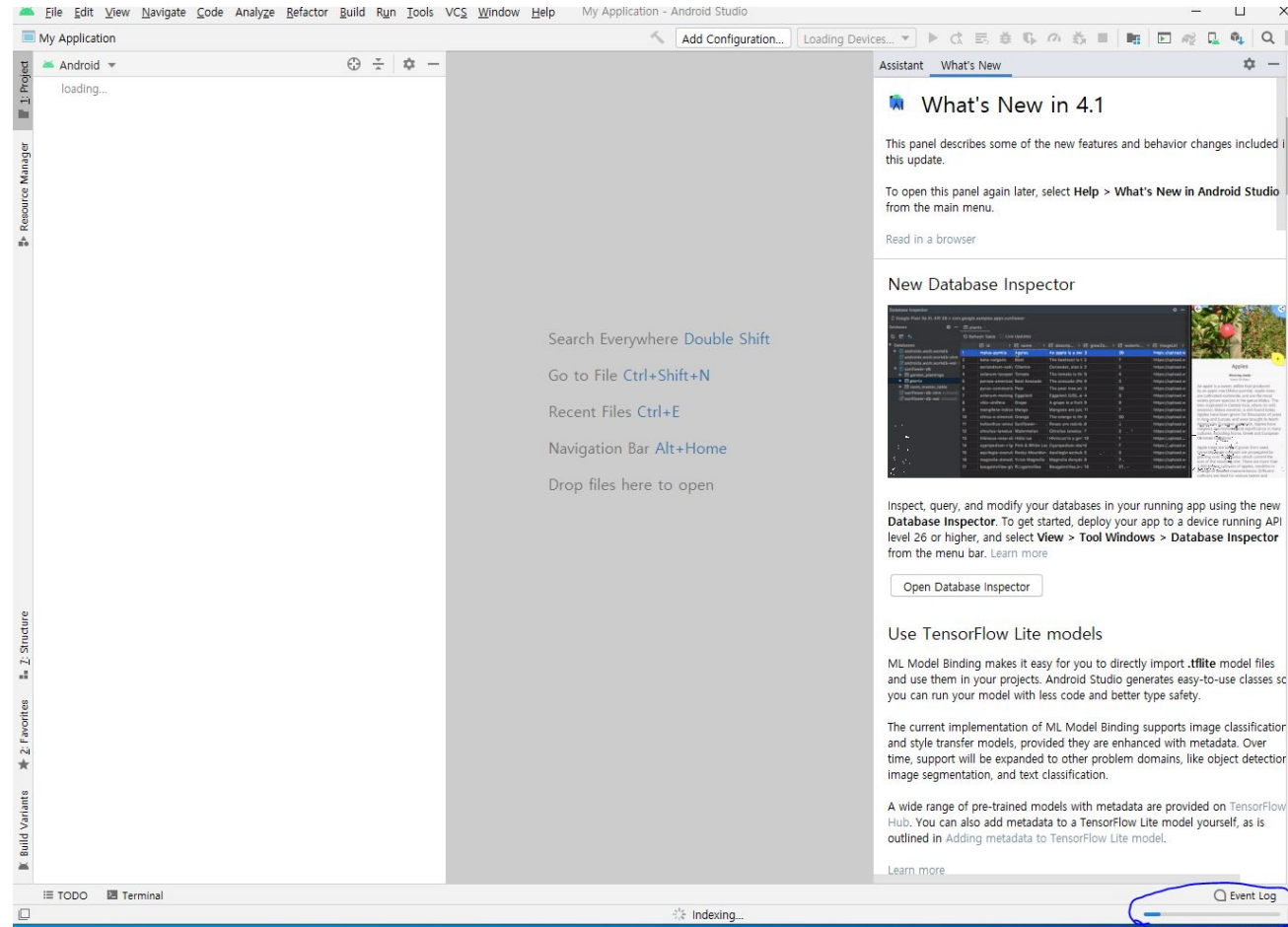
이제 프로젝트를 생성해 **AVD**를 간단하게 써보겠습니다.
Create New Project를 선택해주세요.



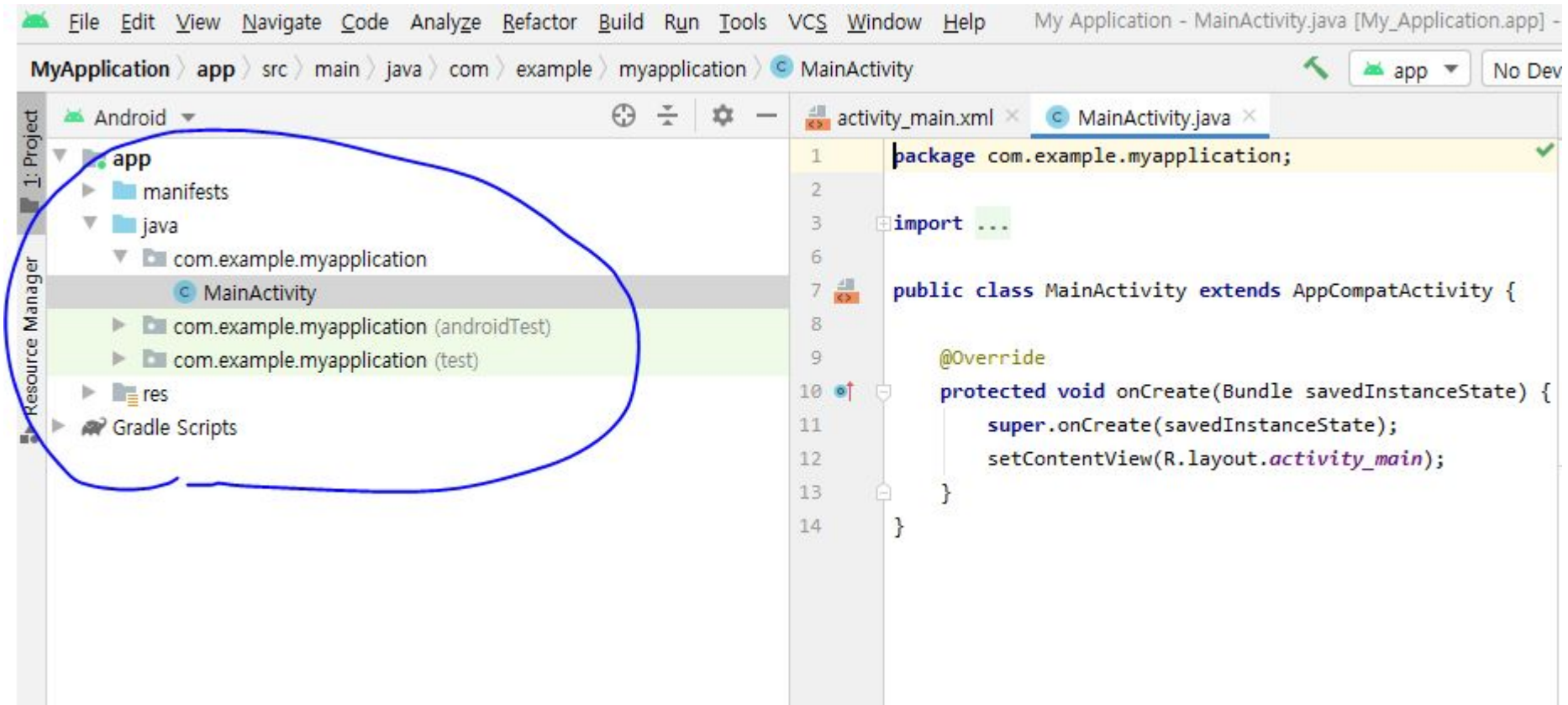
다음 나오는 창에서는 우선 **Empty Activity**를 선택해주시고 **Next**를 눌러주세요.



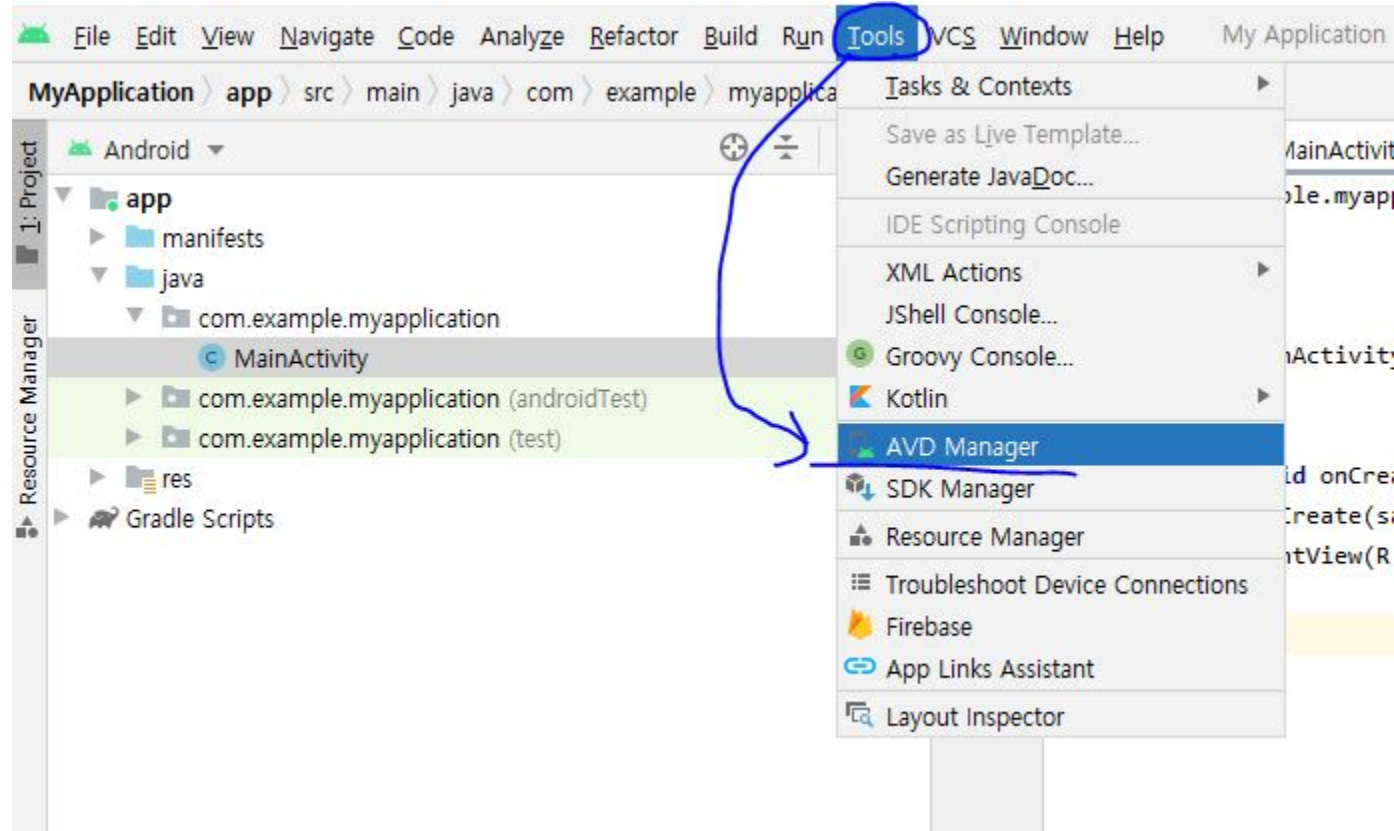
Language를 Java로 바꿔줍니다. Kotlin으로 해놓고 자바 문법을 써도 작동은 하지만 속도차이가 있으므로 Java로 우선 돌려놓습니다.



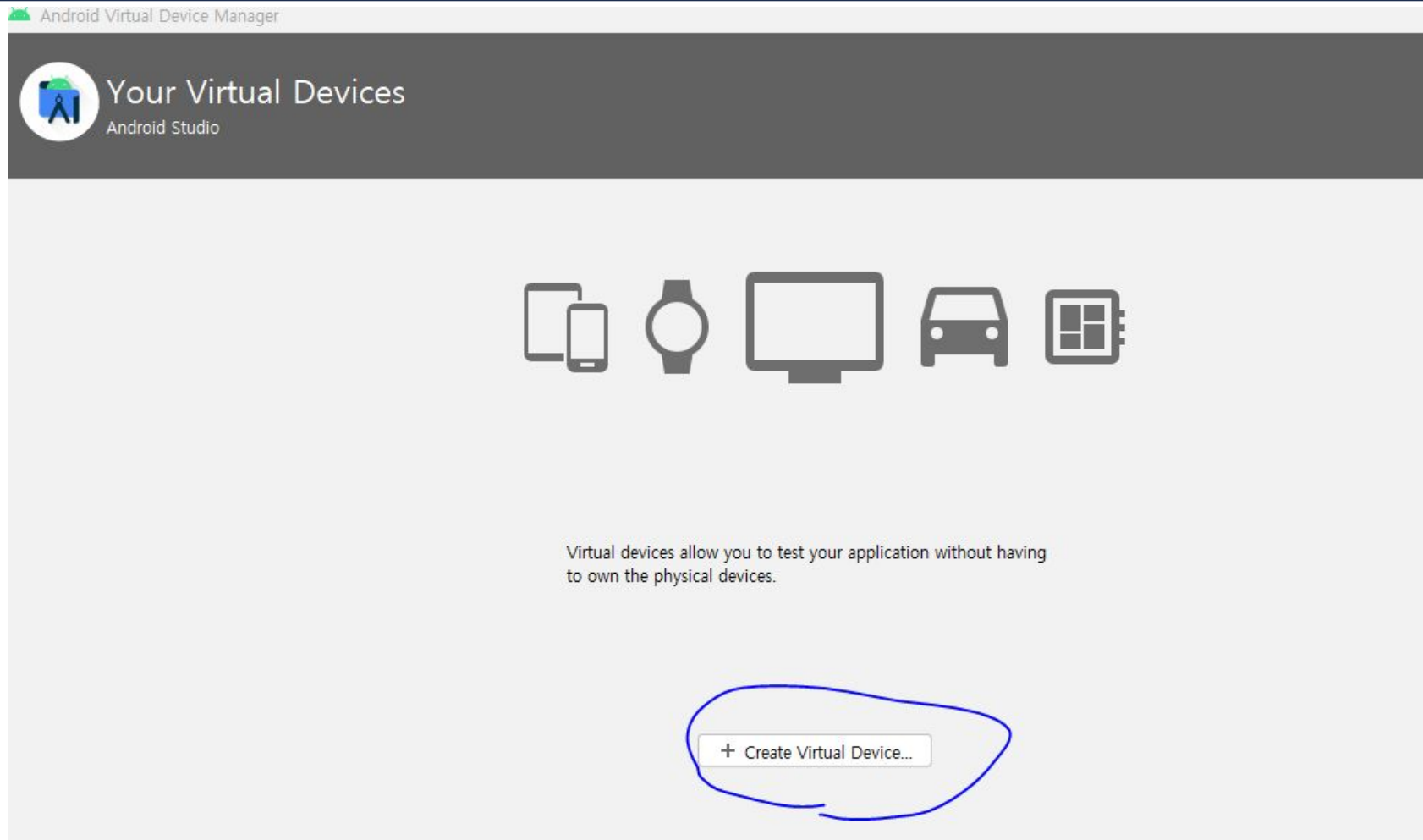
위와 같이 창이 뜨고, 우측하단에 Spring이 그랬듯 의존성 설치를 진행합니다.
완료될때까지 기다려주세요.



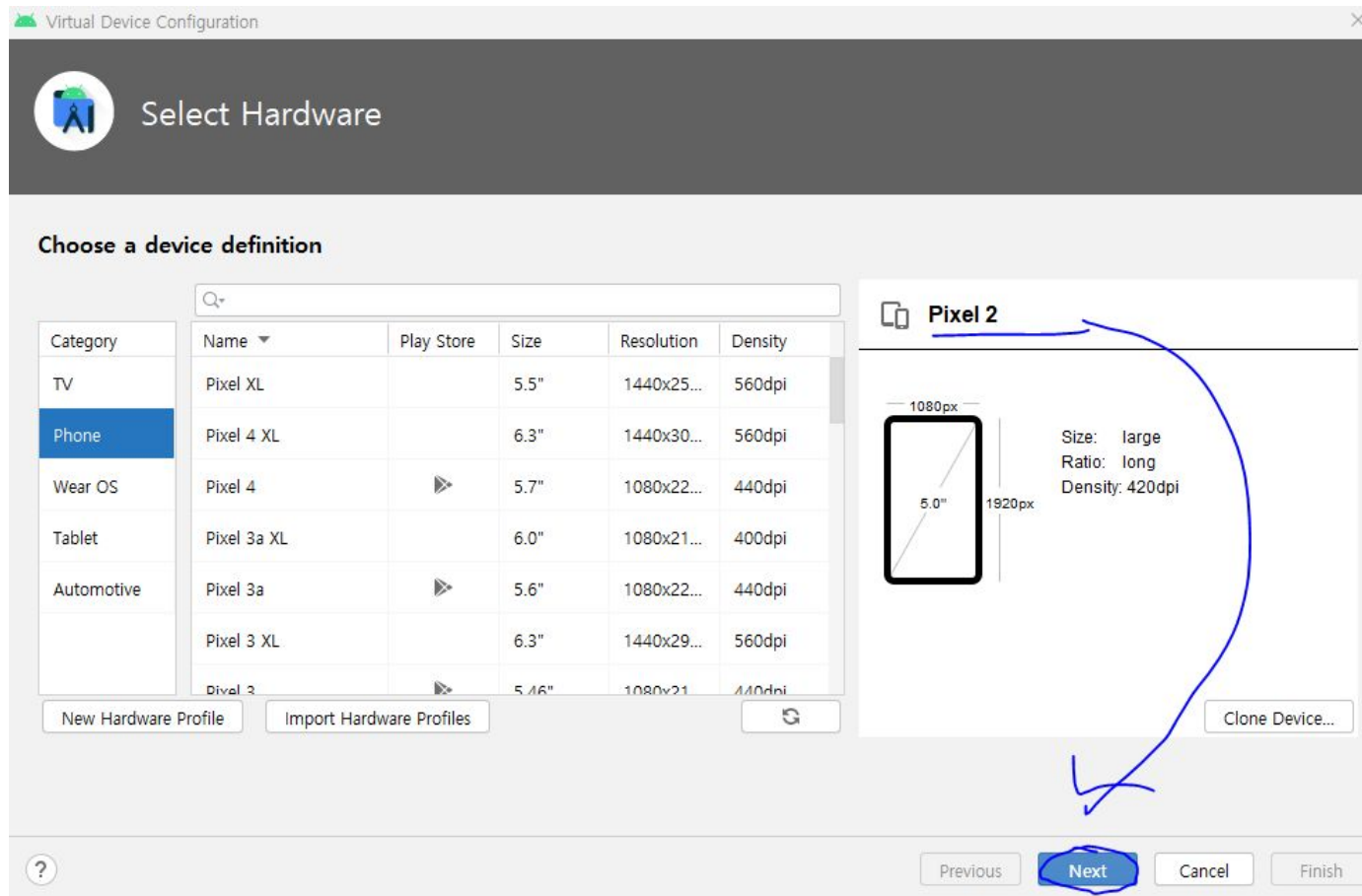
완료시 위와 같이 제일 왼쪽 탭이 활성화됩니다.
안드로이드 프로그래밍이 가능해진 상태이며 **AVD** 활용 후 본격적 코딩에
들어가면 다시 설명하겠습니다.



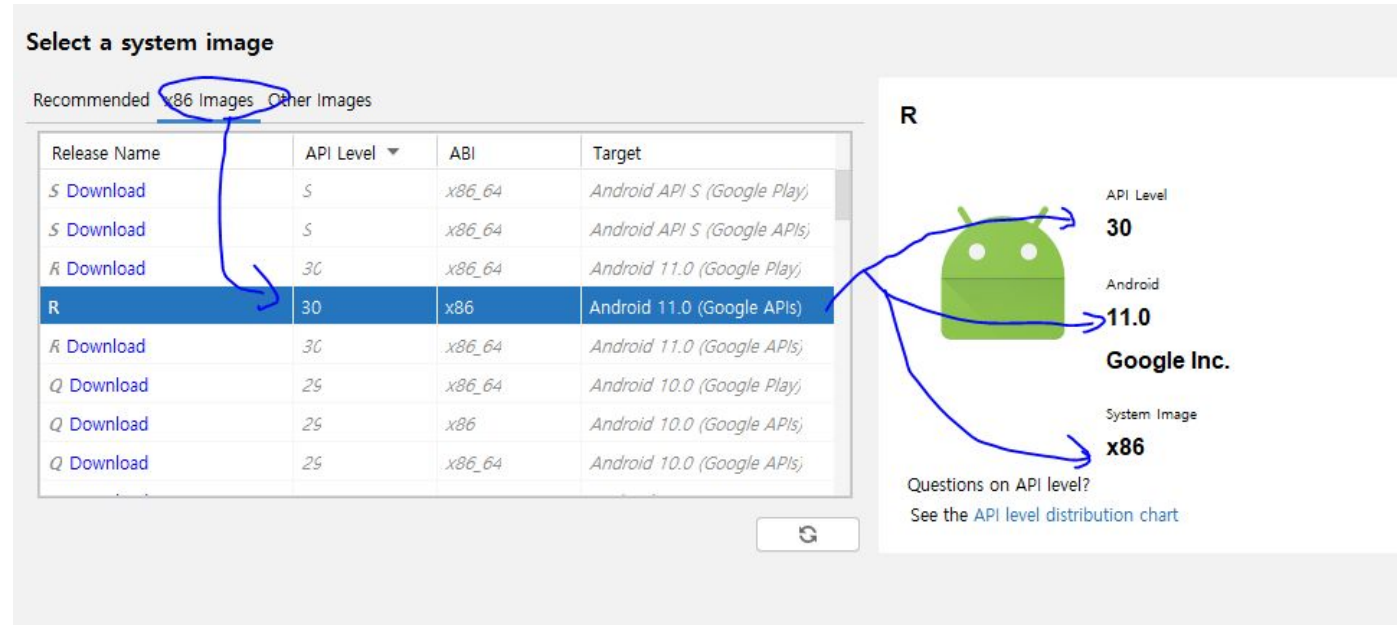
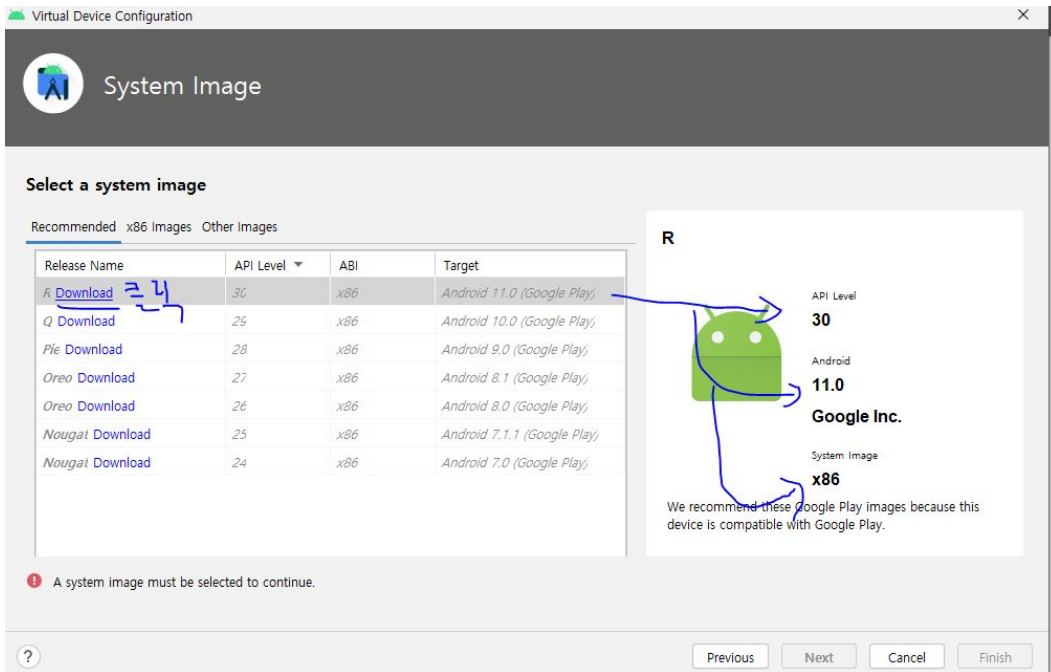
이제 상단의 Tools를 고르고, AVD Manager를 선택합니다.



현재 가상 휴대폰이 없으므로 화면 가운데 **Create Virtual Device**를 선택합니다.

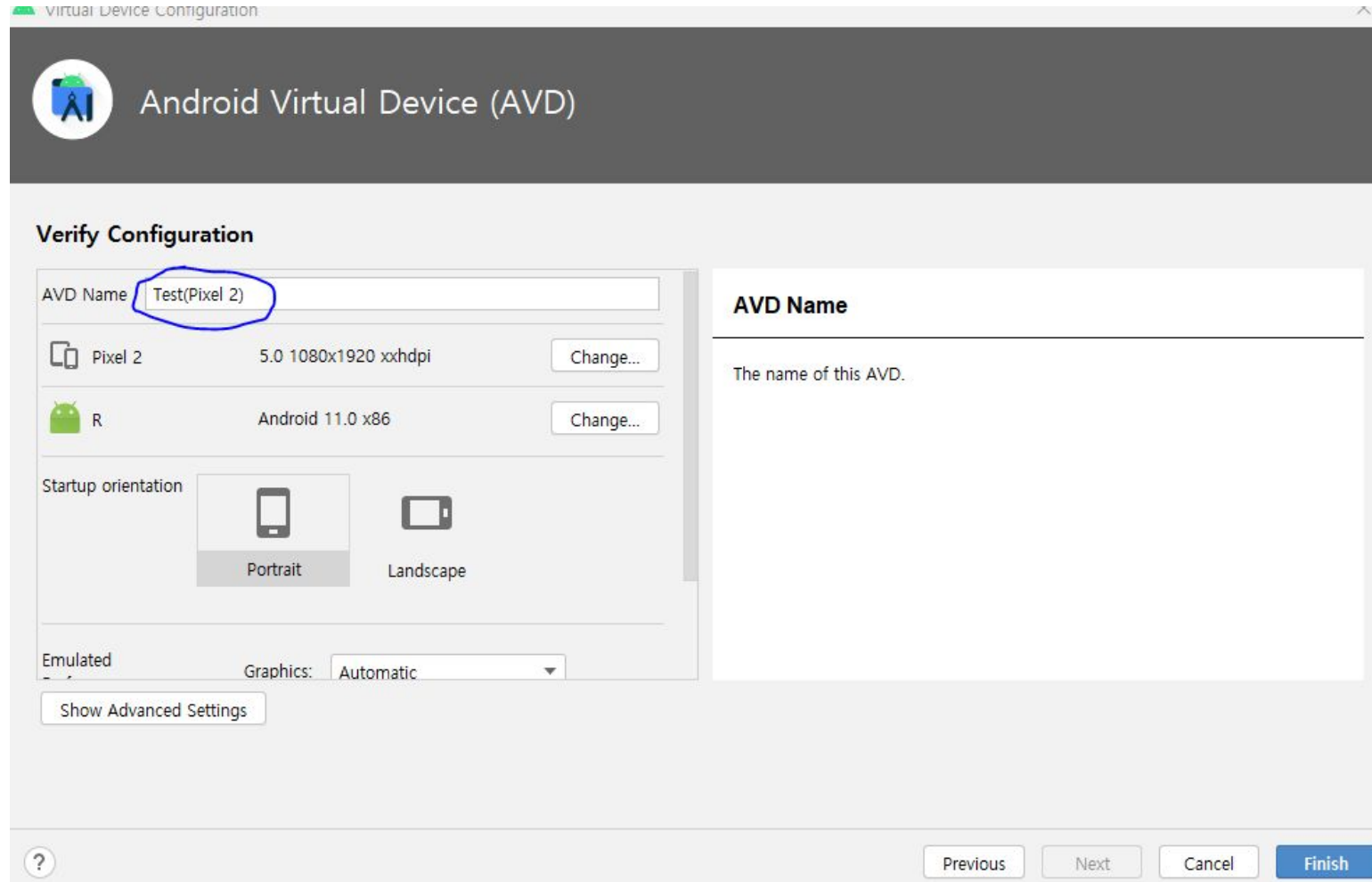


우선 Pixel2를 선택하면 안드로이드 11버전과 호환이 잘 되기 때문에 Pixel 2를 선택합니다.

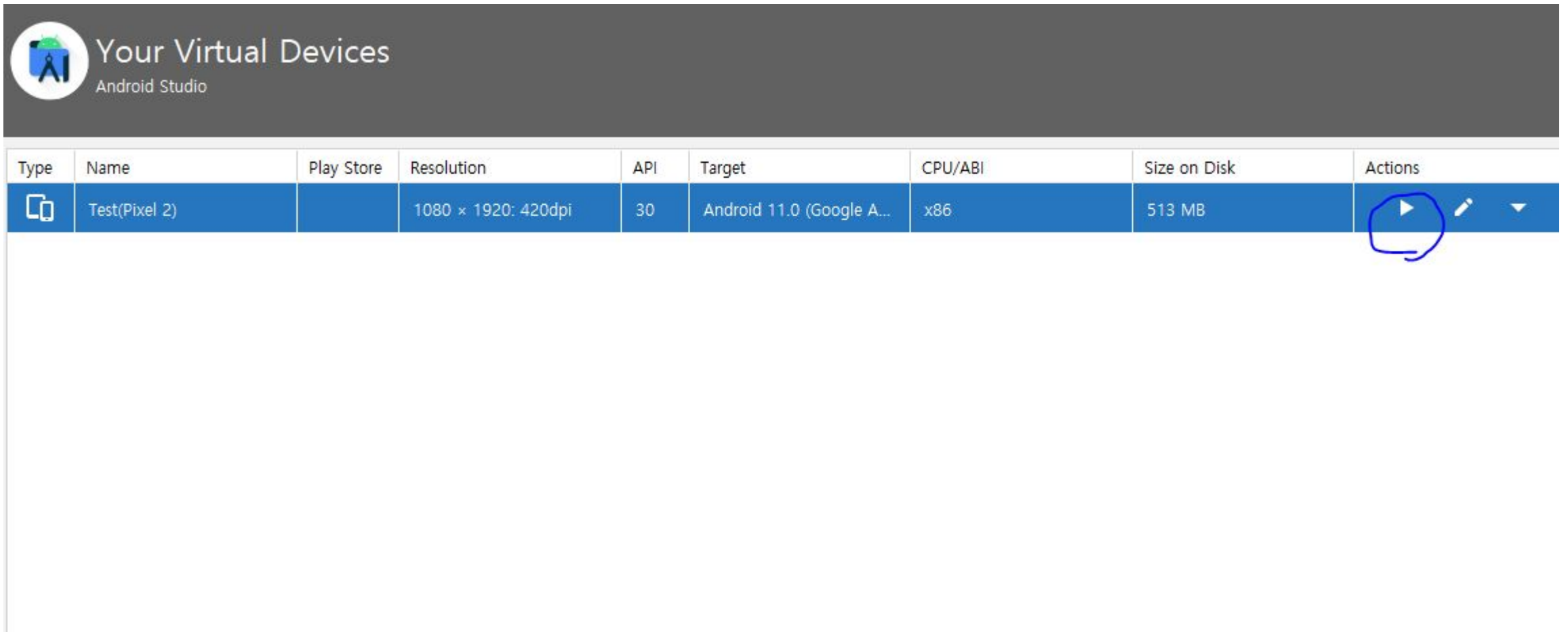


다음, 화면과 같이 API Level 30, Android 11.0, System Image X86을 선택합니다.

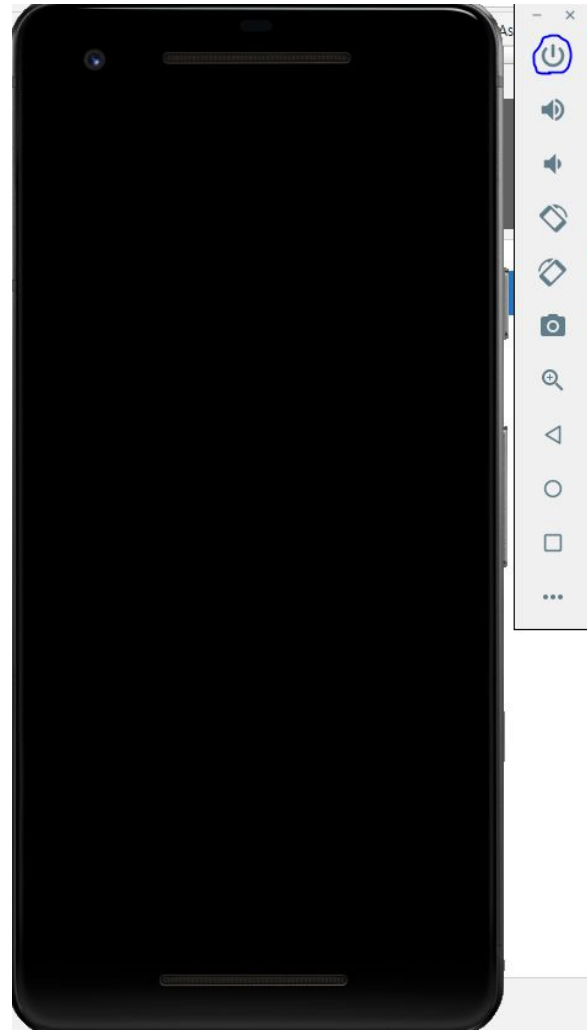
Download를 눌러 SDK를 선택하시되, 만약 다운이 되어있다면 우측처럼 다운로드 탭이 아예 없습니다.



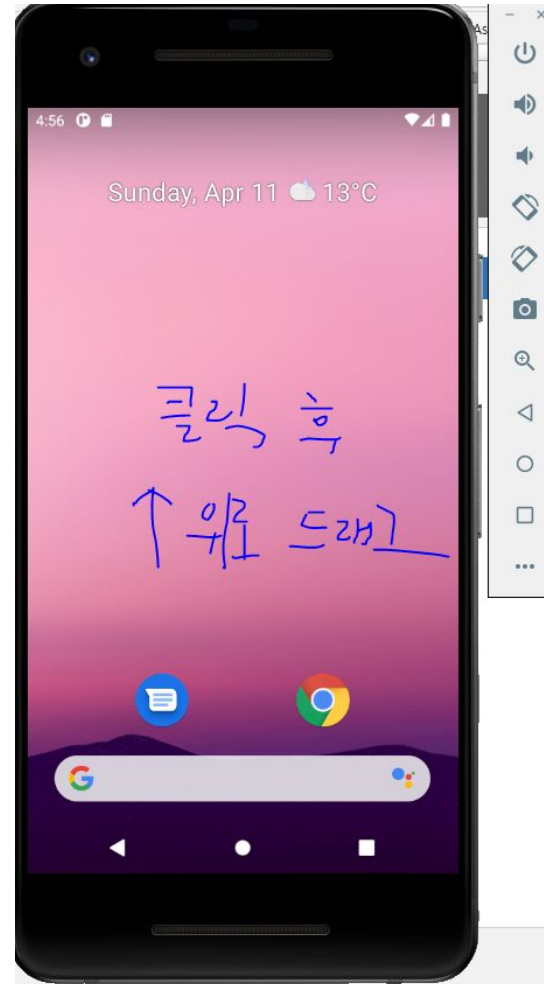
AVD Name은 위와 같이 Test(Pixel 2)로 고쳐주세요.
다 되면 Finish를 눌러줍니다.



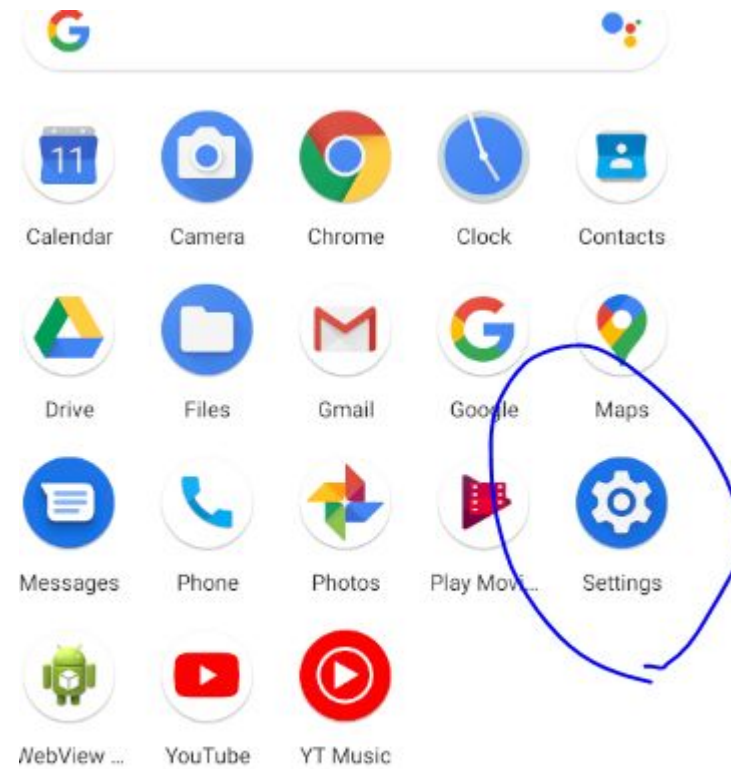
Your Virtual Devices가 나오면 우측 **Actions**의 재생버튼을 눌러 **AVD**를 화면에 띄워봅니다.



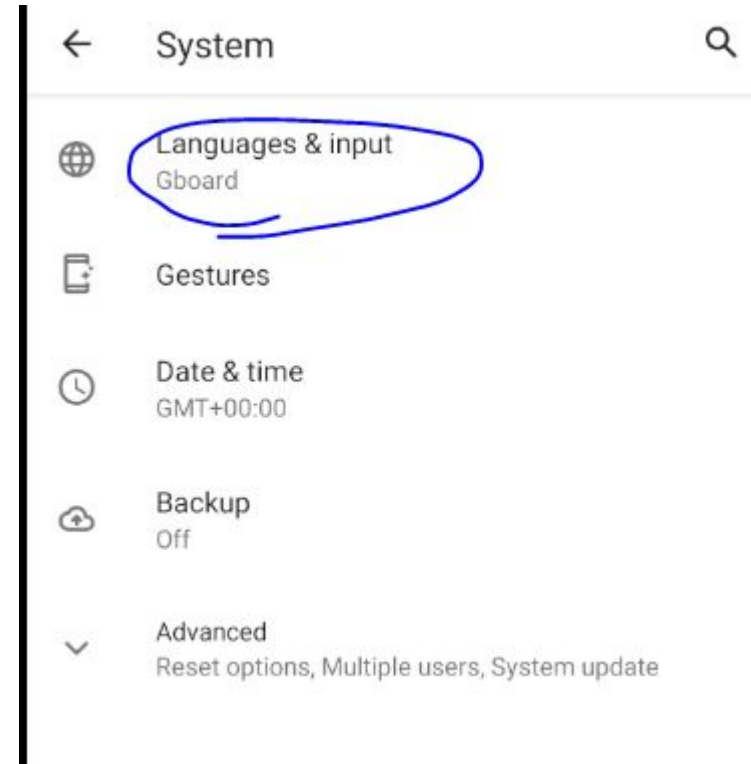
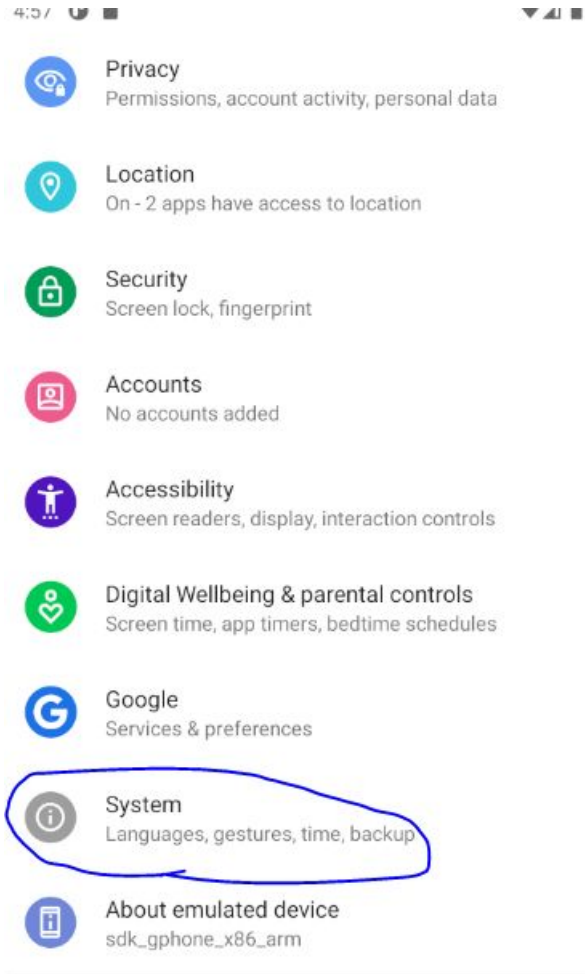
휴대폰이 정확하게 나오면 우측의 전원버튼을 눌러 폰을 켭니다.



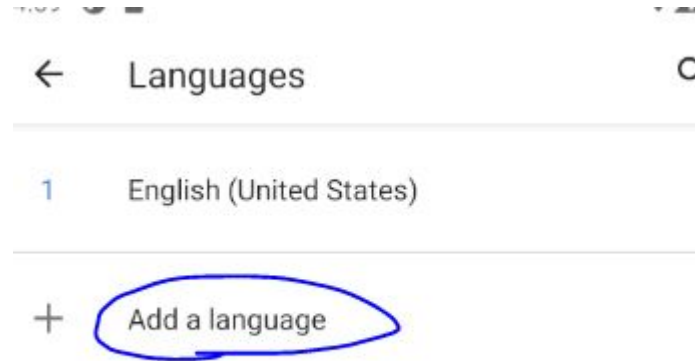
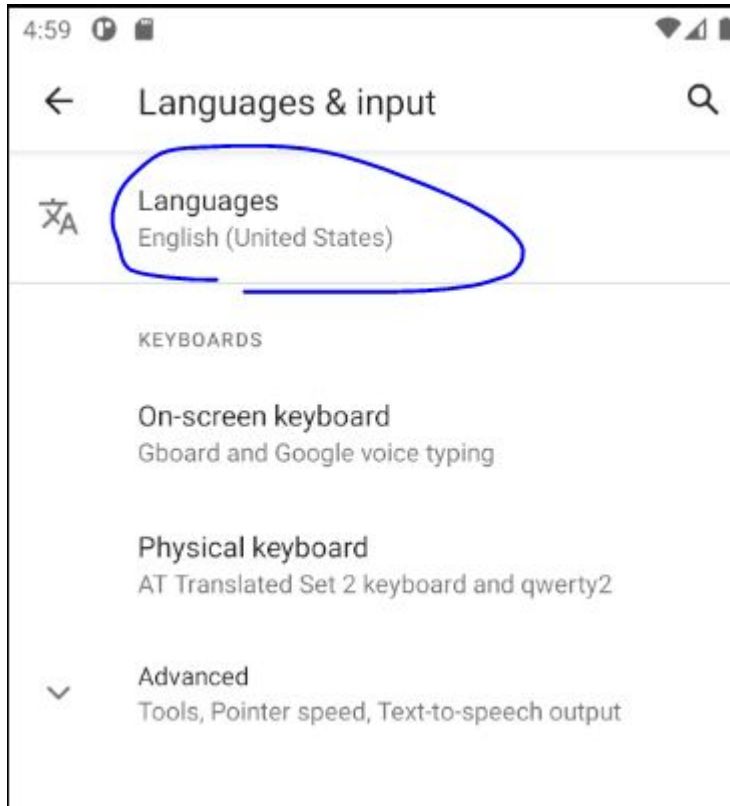
먼저, 한글설정을 위해 화면을 클릭해 위로 드래그하여 설정창을 열어줍니다.



다음 나오는 설정창에서 **Settings**를 클릭해 언어설정을 바꿉니다.



최 하단의 **System** 탭으로 넘어간 다음, **Languages & Input**을 클릭합니다.



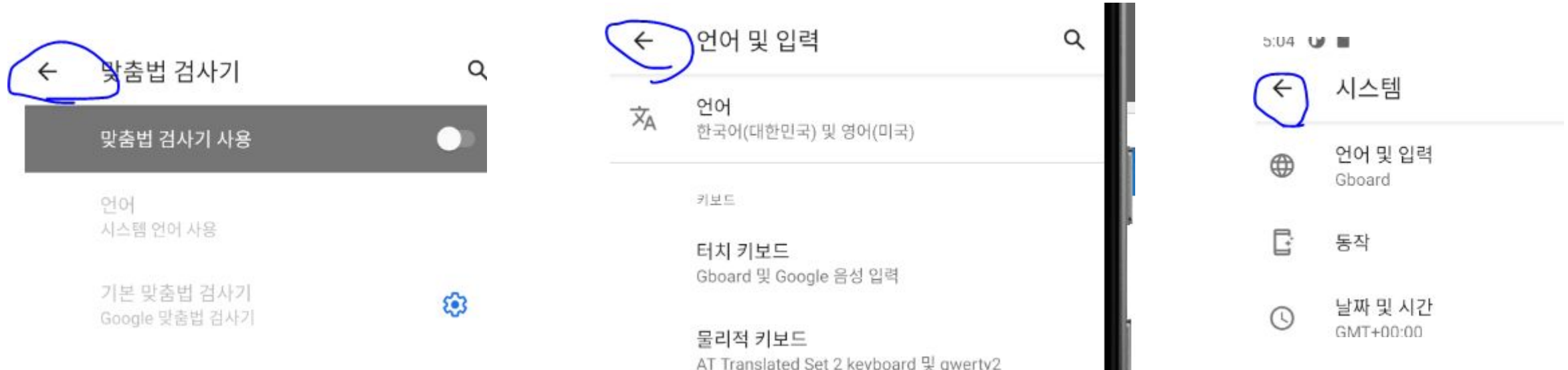
다음 Languages 탭을 클릭하고, Add a language를 클릭한 후 하단으로 쪽 내려와 한국어를 찾아 클릭합니다.



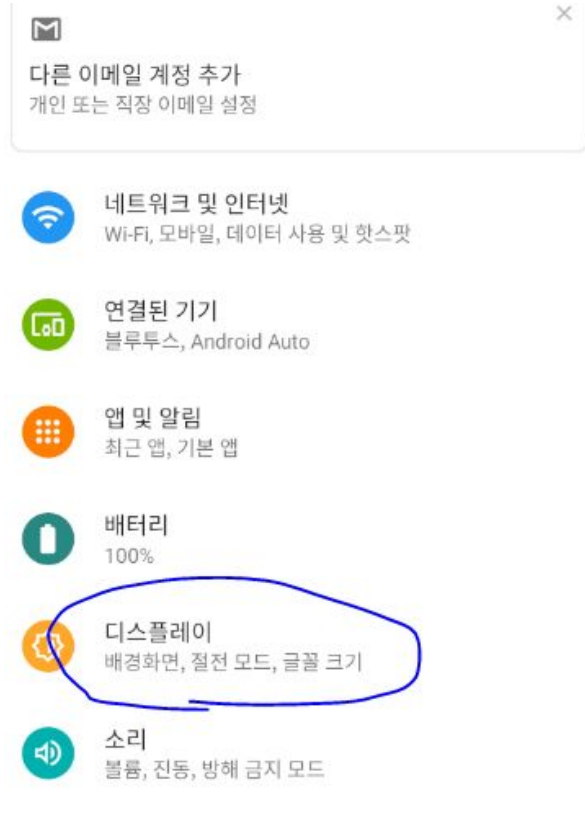
설마 북한...을 고를 분은 없으실테고, 대한민국을 고른 다음
한국어를 위로 끌어올려줍니다.
최종적으로 가장 우측처럼 한국어가 1언어로 설정되면 됩니다.
설정 확인 후 <- 버튼을 클릭해 뒤로 옵니다.



다음 고급을 골라주신 다음, 맞춤법 검사기를 끄겠습니다.
켜 두면 개발상의 문제는 아니고 그냥 빨간 밑줄이 쳐져서 보기 안 좋기
때문입니다.



또한 화면이 1분마다 절전모드가 되게 설정되어있어 불편하니 시간을 30분으로 늘려줍니다.
먼저 <- 버튼을 눌러 뒤로 와주세요.

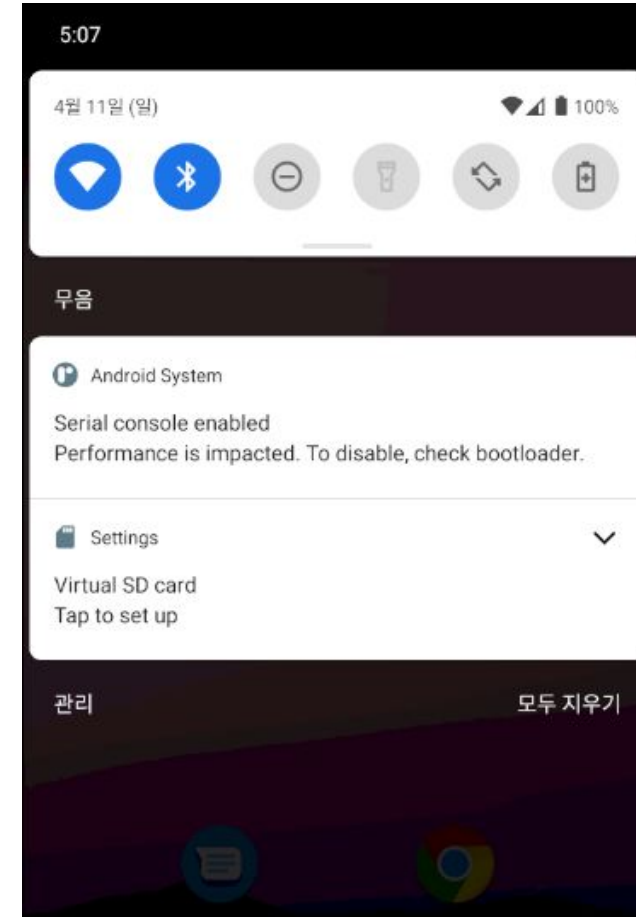
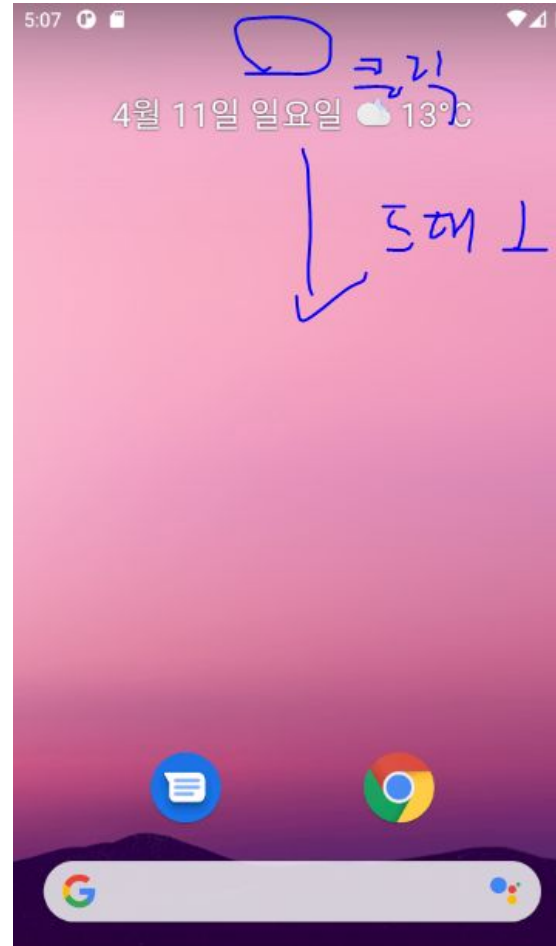
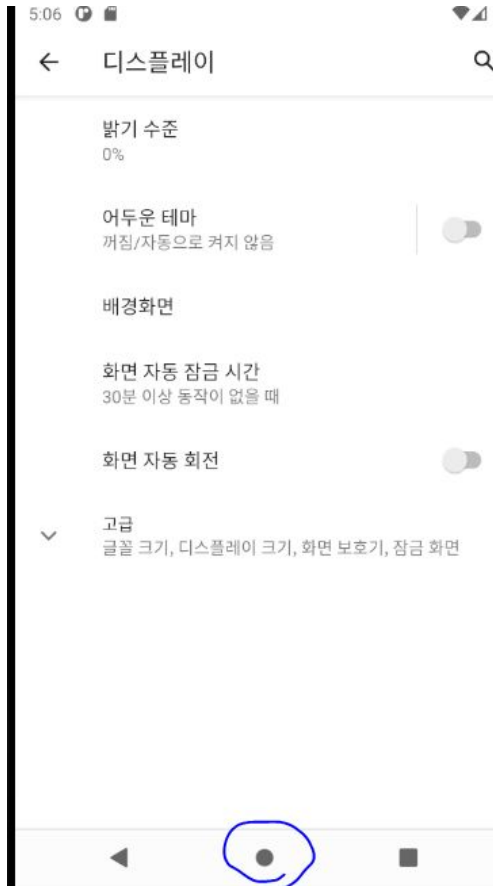


화면 자동 잠금 시간

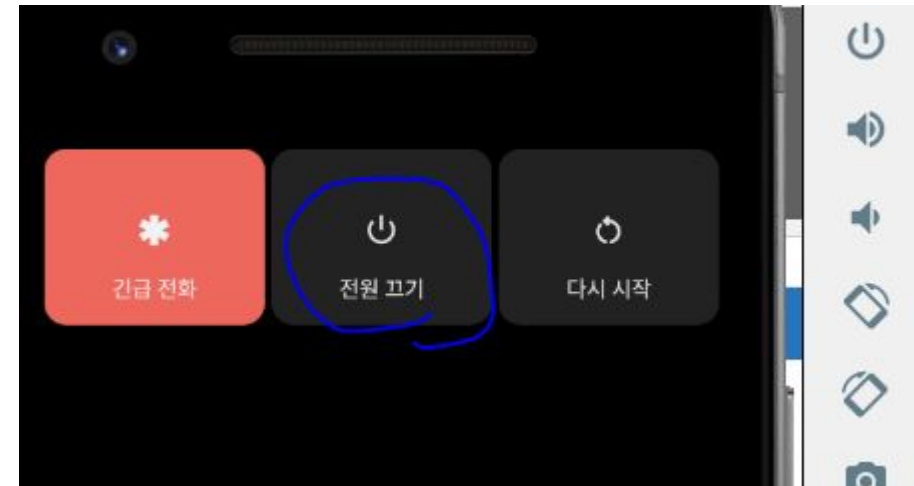
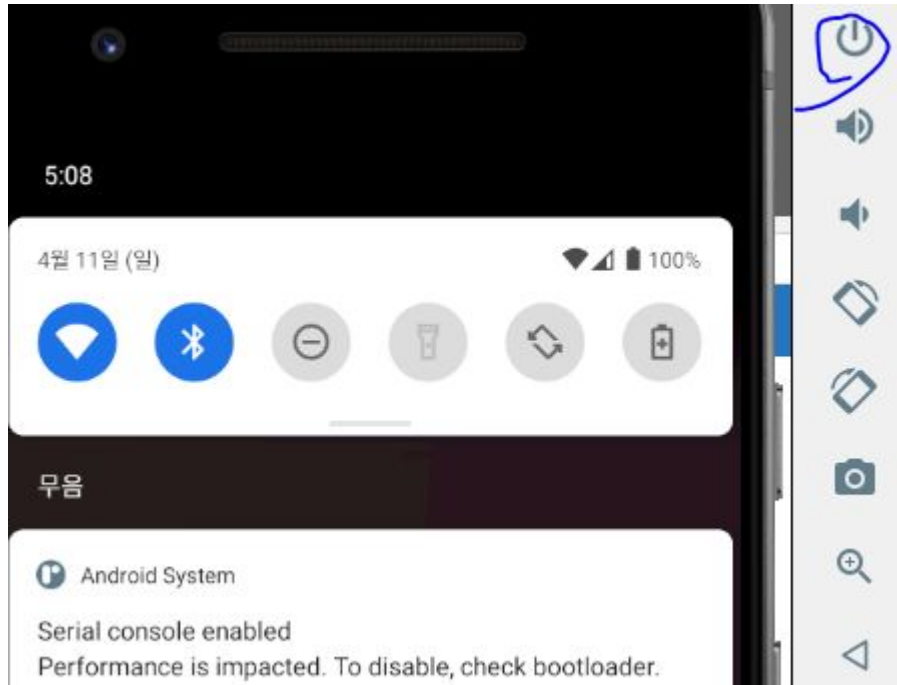
- ☐ 15초
- ☐ 30초
- ☐ 1분
- ☐ 2분
- ☐ 5분
- ☐ 10분
- ☒ 30분

취소

다음으로 켜진 화면에서 디스플레이를 선택하시고,
화면 자동 잠금 시간을 선택한 다음
30분으로 선택합니다.



설정 후 하단의 ● 버튼을 클릭해 초기화면으로 옵니다.
12시 방향을 클릭 후 드래그로 내리면 역시 우리가 쓰는 폰처럼 옵션창이
켜집니다.



그리고 마지막으로 우측 전원버튼을 오래 누르면 종료창이 켜지고,
그 상태로 전원 끄기를 선택하면 AVD가 꺼집니다.