

Android Programming

An Introduction

Android 소개



Android 소개

- ▶ Android는 Google이 Mobile Device에 탑재할 수 있도록 개발한 Open Platform
- ▶ 주목 받는 이유
 - ▶ 완전 개방형 확장 플랫폼
 - ▶ Google의 막강한 영향력
- ▶ 사용 언어
 - ▶ 응용 프로그램 : JAVA 프로그래밍 언어
 - ▶ Middleware : C/C++ 프로그래밍 언어
 - ▶ Linux Kernel : C 프로그래밍 언어
- ▶ License
 - ▶ Open Source 및 상용화 할 수 있는 License - 이중화 전략
 - ▶ Kernel : Embed Linux Kernel - Open Source
 - ▶ 나머지 부분 : BSD혹은 Apache License로 처리하여 상용화에 따른 문제를 적절하게 처리

Android 소개

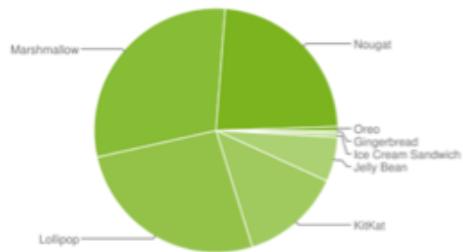
▶ Android 특징

- 1) **Application Framework** : Application의 재사용과 대체가 가능함
- 2) **Dalvik Virtual Machine** : Mobile Device에 최적화됨 (4.4.4 이하)
 - 1) ART(Android Runtime)으로 변경되어 현재에 이름
- 3) **통합 Browser** : Open Source WebKit Engine 기반
- 4) **graphic 최적화** : 2D/3D (OpenGL ES 1.0사용 – Android 2.0부터는 OpenGL 2.0지원) 그래픽, 옵션으로 하드웨어 가속 기능 지원
- 5) **SQLite** : 데이터 저장을 위한 데이터베이스 엔진 제공
- 6) **미디어 지원** : 다양한 오디오, 비디오 및 이미지 포맷 지원(MPEG4, H.264, MP3, JPG, PNG, GIF 등) 기존의 Phone에 적용된 포맷만 지원, PC에서 사용하는 포맷은 지원하지 못함(예:AVI, WMV등)
- 7) **하드웨어 의존적인 기술들** : GSM telephony 기술, bluetooth, 3G, WiFi, Camera, GPS, compass, accelerometer
- 8) **유용한 개발환경 지원** : 디버깅을 위한 IntelliJ기반의 device emulator를 지원, memory 및 성능 profiling 기능 지원, IntelliJ에 기반한 x86기반의 크로스 통합 개발 환경(IDE) 지원

Android 소개

: Android Version

| Version | Codename | API | Distribution |
|---------------|--------------------|-----|--------------|
| 2.3.3 - 2.3.7 | Gingerbread | 10 | 0.4% |
| 4.0.3 - 4.0.4 | Ice Cream Sandwich | 15 | 0.5% |
| 4.1.x | Jelly Bean | 16 | 2.0% |
| 4.2.x | | 17 | 3.0% |
| 4.3 | | 18 | 0.9% |
| 4.4 | KitKat | 19 | 13.4% |
| 5.0 | Lollipop | 21 | 6.1% |
| 5.1 | | 22 | 20.2% |
| 6.0 | Marshmallow | 23 | 29.7% |
| 7.0 | Nougat | 24 | 19.3% |
| 7.1 | | 25 | 4.0% |
| 8.0 | Oreo | 26 | 0.5% |



- ▶ 2005년 구글이 안드로이드를 인수
- ▶ 2017년 현재 안드로이드 최신 버전은 Oreo 버전(8.0)

[참조] <https://developer.android.com/about/dashboards/index.html?hl=ko>

Android 소개

: Android Platform

▶ Android Platform

- ▶ 1) 안드로이드는 운영체제와 프레임워크 그리고 핵심 애플리케이션을 포함하고 있는 모바일 플랫폼
- ▶ 2) 안드로이드는 하나의 모바일 기기의 운영체제이자 다른 애플리케이션들을 개발하여 실행하는 소프트웨어 프레임워크이다.
- ▶ 3) 핵심적인 커널뿐만 아니라 애플리케이션 개발을 위한 다양한 컴포넌트 제공하여 안드로이드 개발자들은 이러한 컴포넌트를 이용하여 손쉽게 안드로이드 애플리케이션을 빠르게 개발할 수 있다.

Android 소개

: Android Platform

- ▶ Linux 커널
 - ▶ Android 플랫폼의 기반
 - ▶ 보안, 메모리 관리, 프로세스 및 쓰레드 관리, 네트워크, 전원 관리, 그리고 각종 하드웨어 드라이버 등과 같은 핵심 시스템 서비스들을 처리
- ▶ HAL(Hardware Abstraction Layer: 하드웨어 추상화 계층)
 - ▶ 물리적인 하드웨어를 추상화하여 안드로이드 플랫폼에서 접근 가능하도록 하는 역할을 담당
 - ▶ 상위 수준의 Java API 프레임워크에 기기 하드웨어 기능을 노출하는 표준 인터페이스 제공



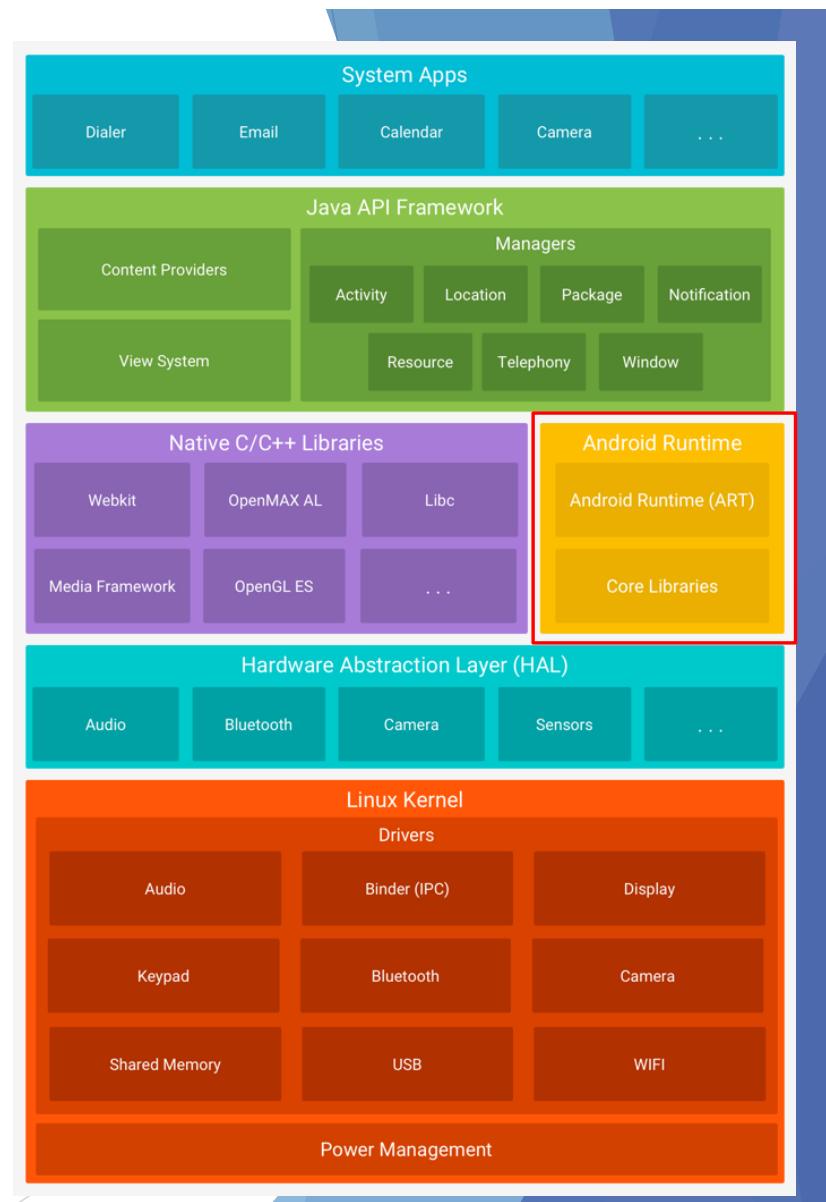
[출처] <https://developer.android.com/guide/platform/index.html>

Android 소개

: Android Platform

▶ Android Runtime

- ▶ 런타임 시에 필요한 핵심 라이브러리들과 ART로 구성
- ▶ Android 버전 5.0 이상을 실행하는 기기의 경우, 각 앱이 자체 프로세스 내에서 자체 ART 런타임 인스턴스로 실행
 - ▶ 이전 버전은 Dalvik VM으로 실행
 - ▶ 하나의 모바일 기기에서 여러 개의 인스턴스가 동시에 실행되는 멀티태스킹 환경
- ▶ 안드로이드 런타임의 핵심으로, 모바일 환경에 최적화된 자바 VM이다.
- ▶ SUN(Oracle)의 오리지널 Java와 호환되지 않음

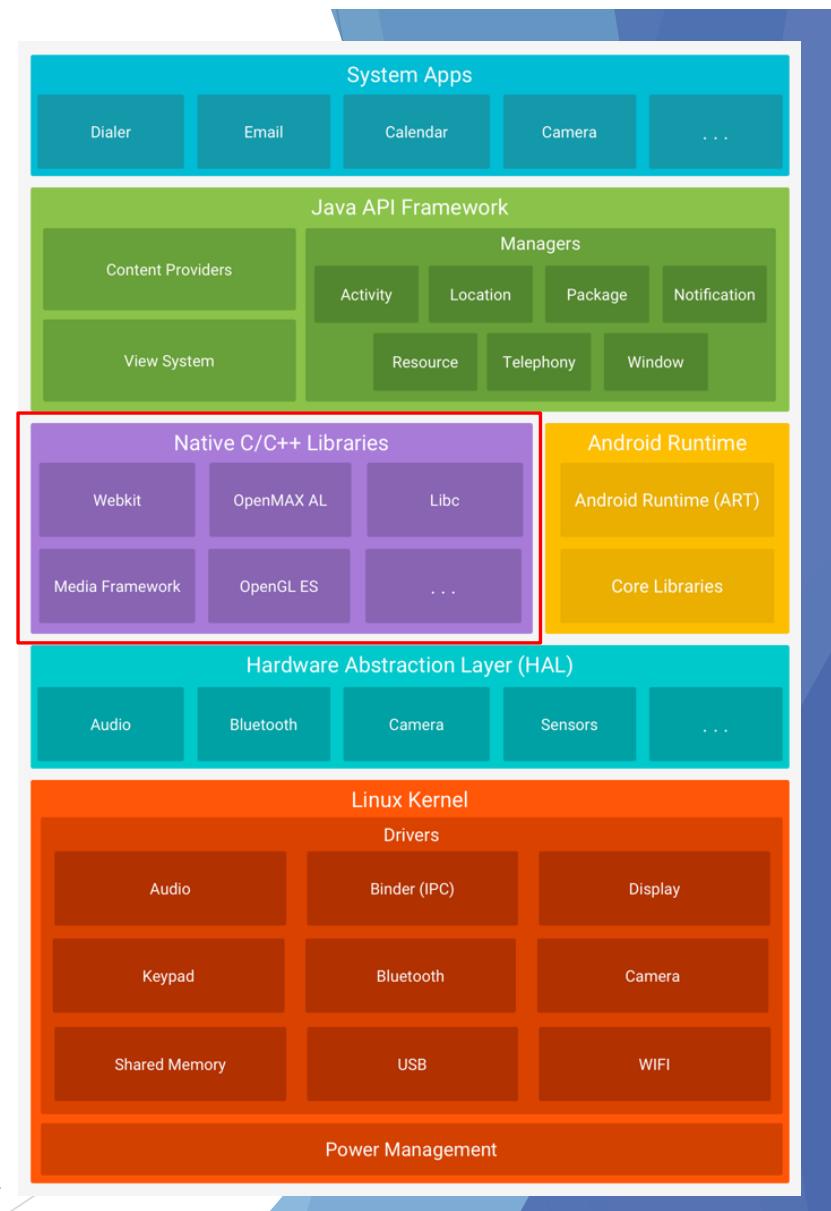


[출처] <https://developer.android.com/guide/platform/index.html>

Android 소개

: Android Platform

- ▶ Native C/C++ 라이브러리
 - ▶ ART 및 HAL 등 많은 핵심 Android 시스템 구성요소가 C/C++ 네이티브 코드를 기반으로 빌드
 - ▶ Android는 이러한 네이티브 라이브러리의 일부 기능을 Java 프레임워크 API를 제공하여 기능을 앱에 노출
 - ▶ 예) OpenGL ES -> Java OpenGL API (2D, 3D 그래픽을 그리고 조작)
 - ▶ C/C++ 코드가 필요한 앱을 개발하는 경우, Android NDK를 이용하여 네이티브 라이브러리에 접근, 사용 가능

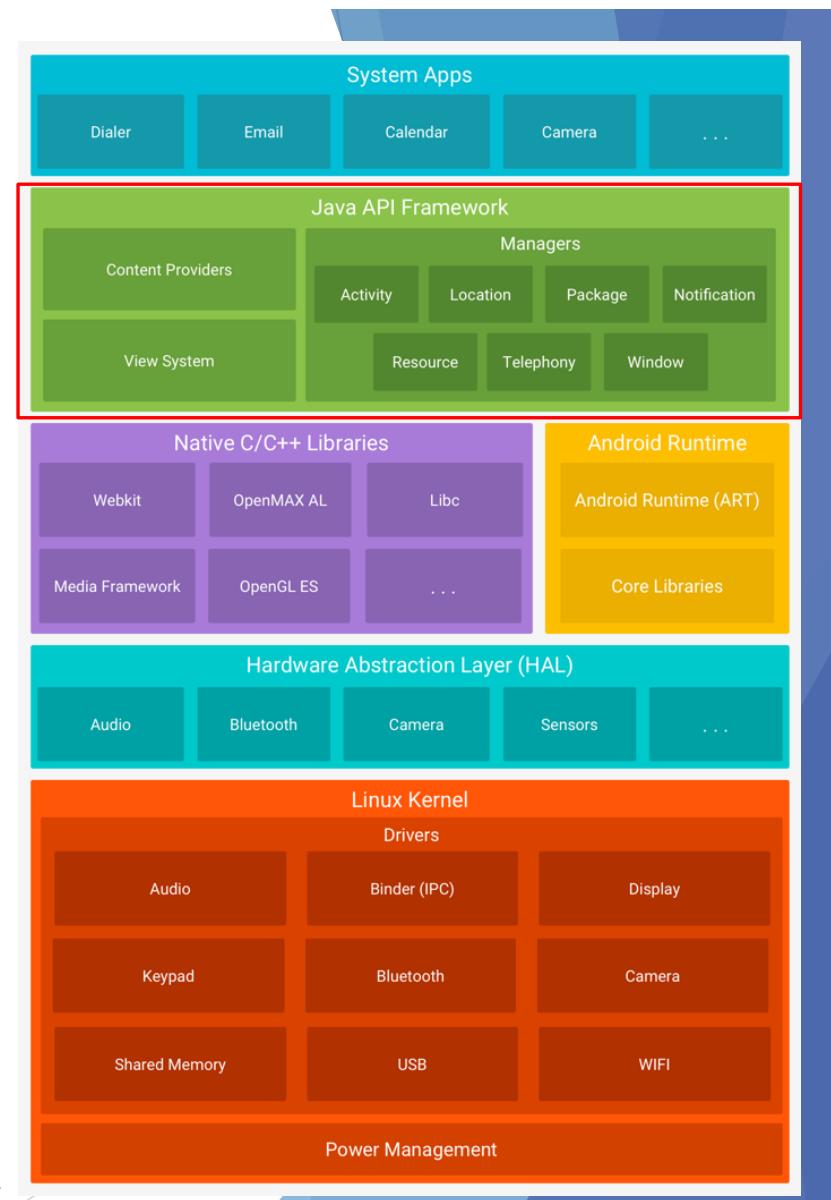


[출처] <https://developer.android.com/guide/platform/index.html>

Android 소개

: Android Platform

- ▶ Java API 프레임워크
 - ▶ Android OS의 전체 기능 세트는 Java 언어로 작성된 API를 통해 액세스
 - ▶ 핵심 모듈식 시스템 구성 요소 및 서비스 재활용을 단순화
 - ▶ Android 앱을 제작하는 데 필요한 빌딩 블록을 구성
 - ▶ 기능이 풍부하며 확장 가능한 뷰 시스템 - 목록, 그리드, 텍스트 상자, 버튼 및 삽입 가능한 웹 브라우저를 포함하여 앱의 UI를 빌드하는데 사용 가능
 - ▶ Resource Manager - 혼자화된 문자열, 그래픽 및 레이아웃 파일과 같은 코드가 아닌 리소스에 대한 액세스 제공
 - ▶ Notification Manager - 모든 앱이 상태 표시줄에 사용자 지정 알림을 표시할 수 있도록 지원
 - ▶ Activity Manager - 앱의 수명 주기를 관리, 하고 공통 탐색 백 스택 제공
 - ▶ Contents Manager - 앱이 다른 앱의 데이터에 액세스하거나 자신의 데이터를 공유할 수 있도록 지원
 - ▶ 개발자에게 Android 시스템 앱이 사용하는 것과 동일한 프레임워크 API에 대한 전체 액세스 권한을 부여



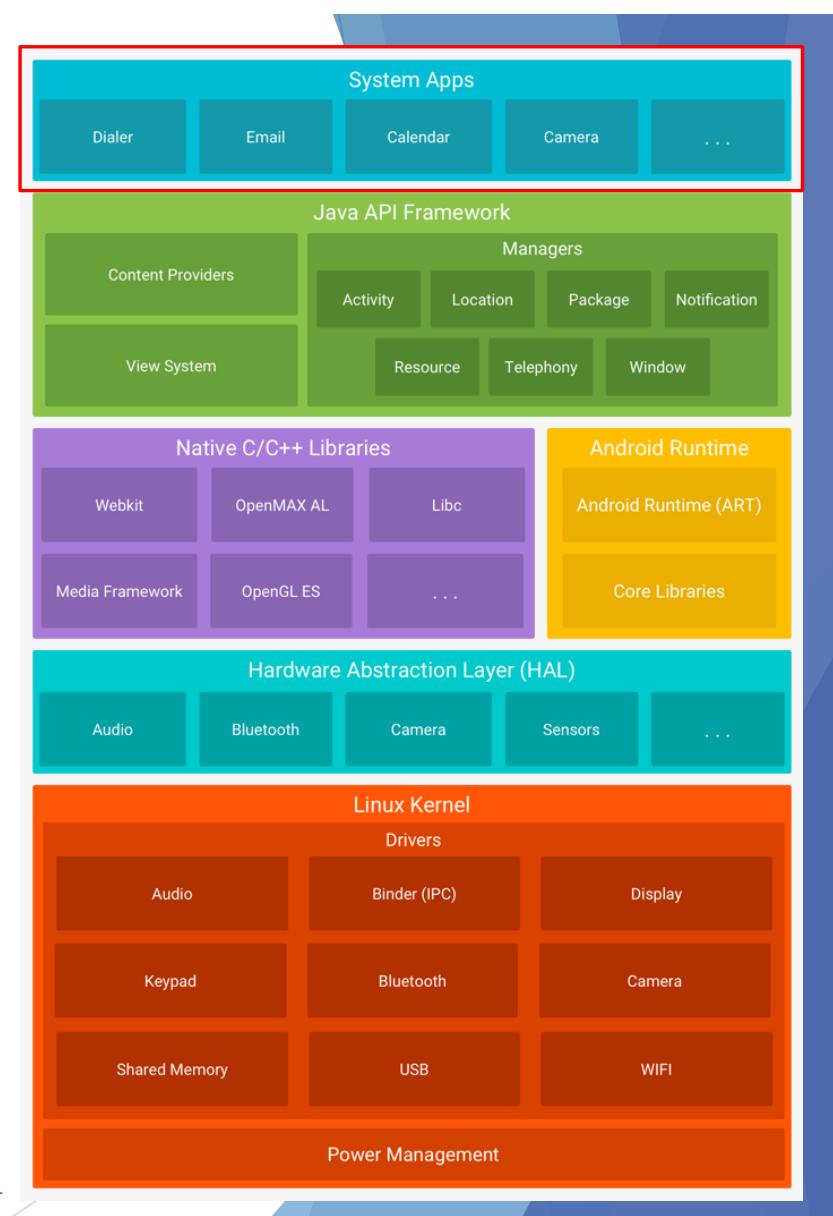
[출처] <https://developer.android.com/guide/platform/index.html>

Android 소개

: Android Platform

▶ 시스템 앱

- ▶ 이메일 클라이언트, SMS 문자메시지 프로그램, 달력, 지도, 브라우저, 주소록 등과 같은 핵심 애플리케이션을 탑재
- ▶ 모든 애플리케이션은 자바 프로그래밍 언어를 사용하여 작성
- ▶ 사용자를 위한 앱으로도 작동하고 개발자가 자신의 앱에서 액세스할 수 있는 주요 기능을 제공하기 위한 용도로도 사용
 - ▶ 예) SMS 전송 기능 탑재 -> 직접 개발 대신, SMS 앱을 호출하여 기능 제공

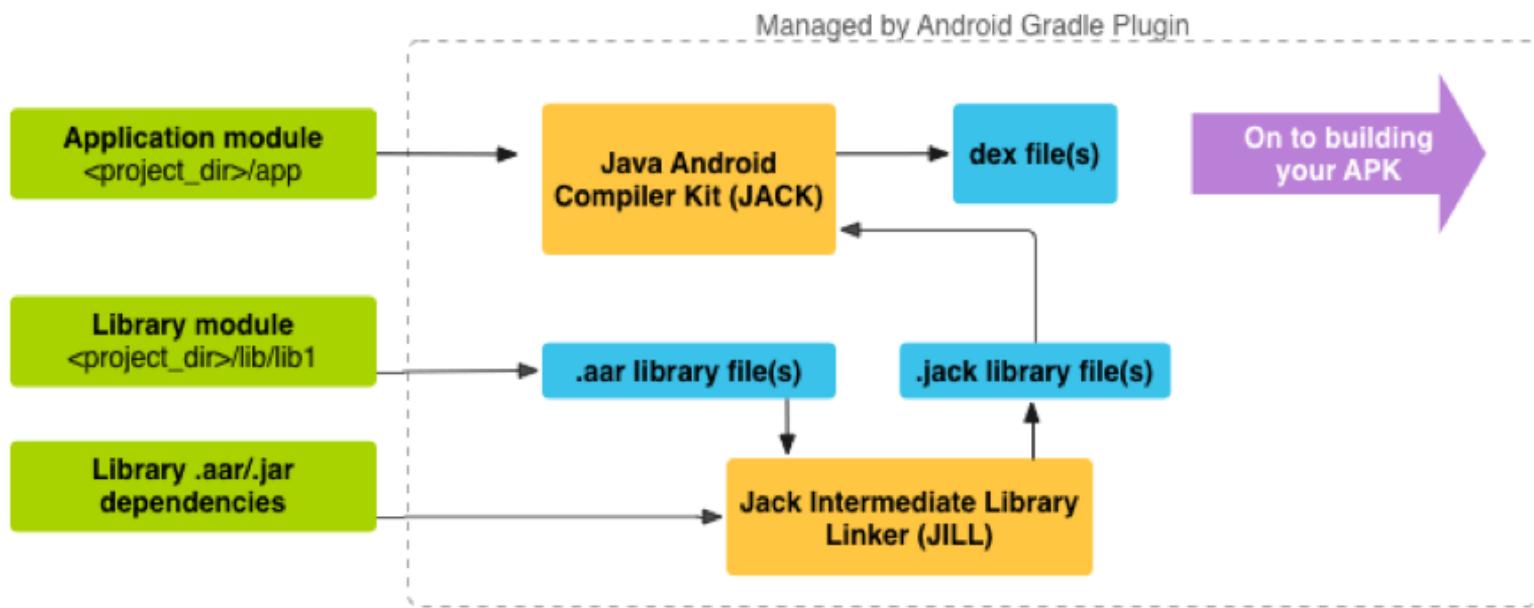


[출처] <https://developer.android.com/guide/platform/index.html>

Android 소개

: Android 개발 과정

Typical Jack and Jill Application Build



Android Studio

- Java로 컴파일

DX Converter

- class + resources
- .apk로 패키징

ART

- 앱을 실행

Android 생태계

android

- ▶ Smart Phones
- ▶ Tablets
- ▶ Smart TVs
- ▶ Automotive
- ▶ Wearables
- ▶ IoT Devices



Visit [Android.com](#)

: for more information



왜 ANDROID인가요? ▾

스마트폰

WEAR

태블릿

TV

엔터프라이즈



마음껏 사진 찍기

Google 포토의 저장용량은 무제한, 무료입니다.
마음껏 사진을 찍어 보세요.

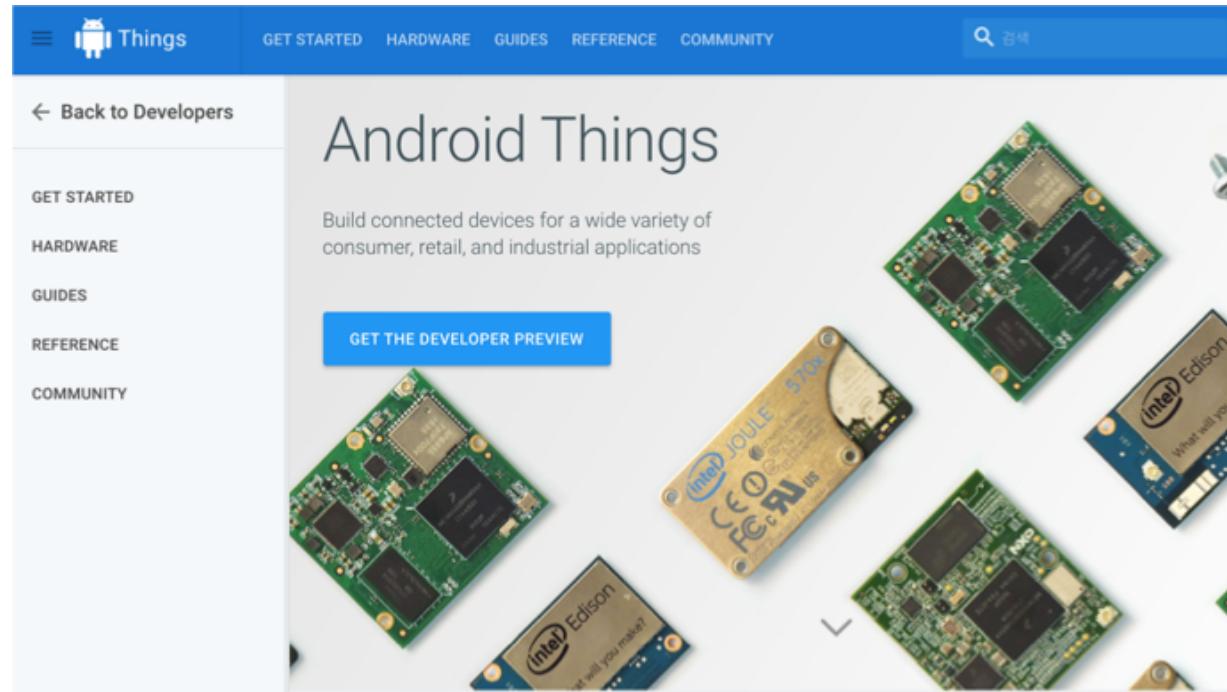
[자세히 알아보기](#)



And more...

: Android Things, Google I/O

<https://events.google.com/io/>

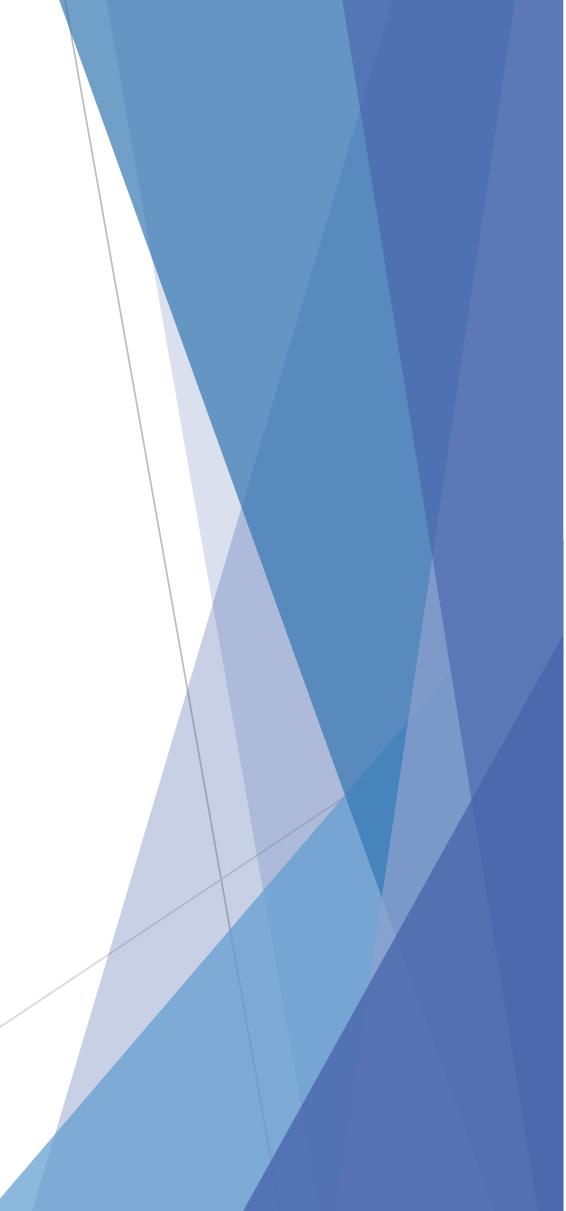


Android Things: <https://developer.android.com/things/index.html>

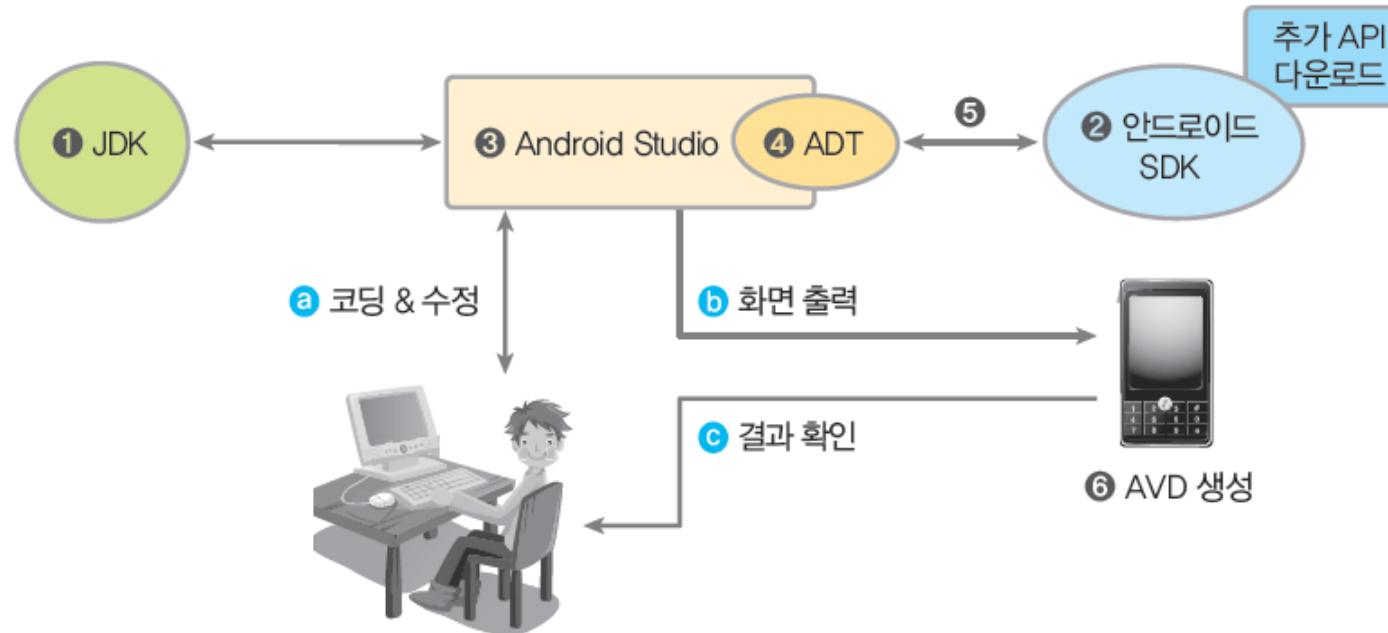


Android 개발 환경

Java와 Android Studio



Android 개발 환경 구성

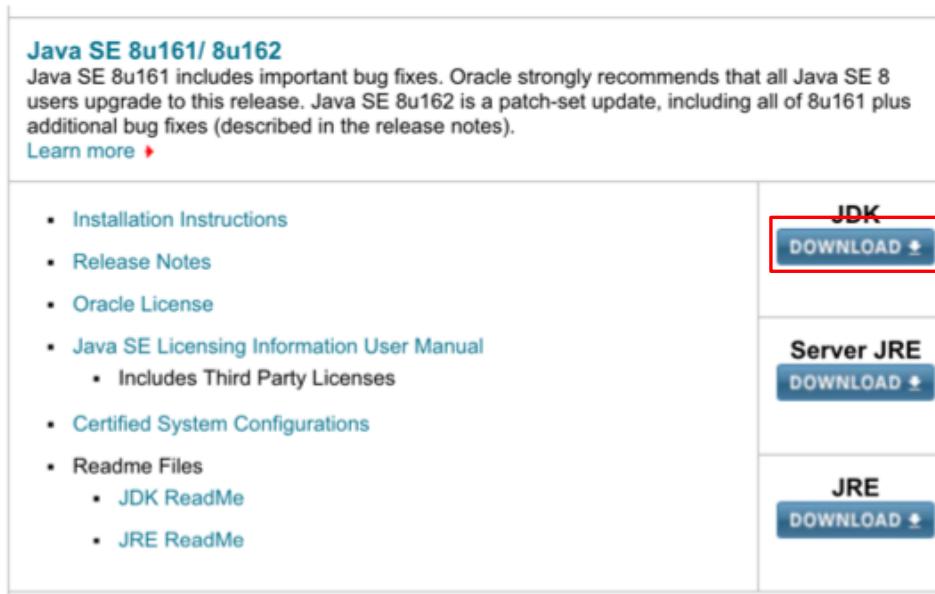


- 안드로이드 스튜디오를 이용하여 코드를 개발, 수정하고
- AVD를 이용하여 앱을 실행, 테스트한다

안드로이드 개발을 위한 JDK 설치

: JDK 설치

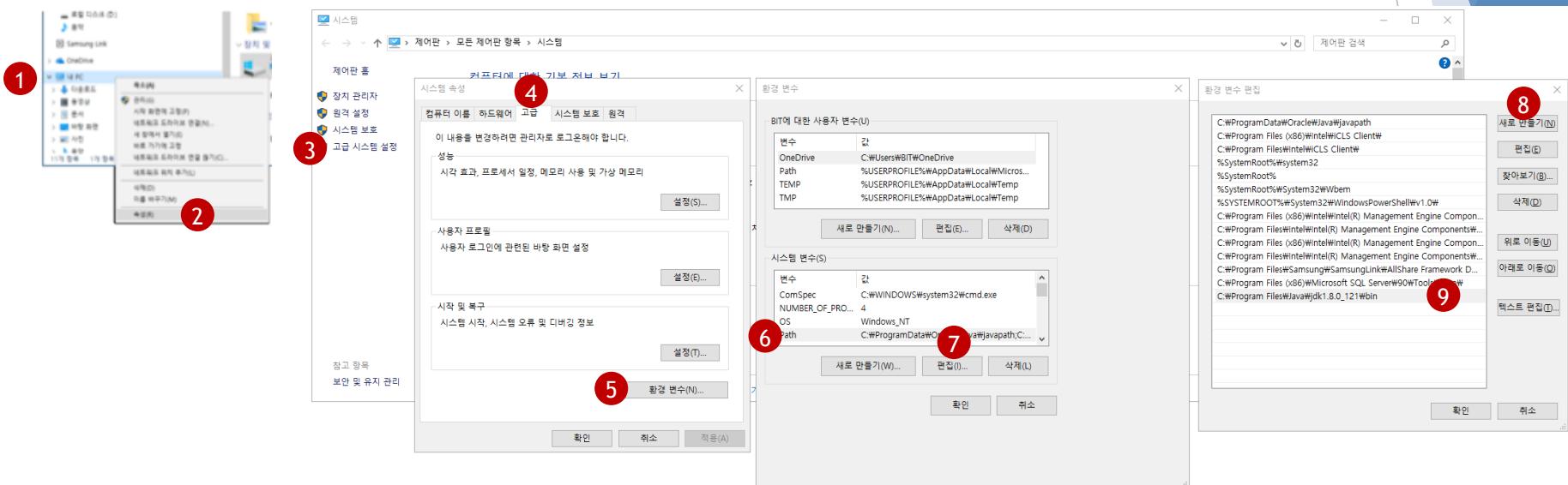
- ▶ <http://java.oracle.com> 에 접속하여 [Software Downloads]의 [Java SE]를 클릭
- ▶ 자신의 운영체제와 플랫폼에 맞는 버전을 다운로드, 설치한다



안드로이드 개발을 위한 JDK 설치

: 환경변수의 설정

- ▶ 윈도 환경 변수에 JAVA_HOME과 Path를 추가해준다



C:\Program Files\Java\jdk1.8.0_131\bin

안드로이드 개발을 위한 JDK 설치

: 환경변수의 설정 확인

- ▶ Win + R > cmd 입력 후 다음 내용을 확인
 - ▶ java
 - ▶ javac

javac 입력

```
C:\Users\BIT>javac
Usage: javac <options> <source files>
where possible options include:
  -g                         Generate all debugging info
  -g:none                     Generate no debugging info
  -g:{lines,vars,source}        Generate only some debugging info
  -nowarn                     Generate no warnings
  -verbose                    Output messages about what the compiler is doing
  -deprecation                Output source locations where deprecated APIs are used
  -classpath <path>           Specify where to find user class files and annotation processors
  -cp <path>                  Specify where to find input source files
  -sourcepath <path>          Override location of bootstrap class files
  -bootclasspath <path>       Override location of installed extensions
  -extdirs <dirs>             Override location of endorsed standards path
  -endorseddirs <dirs>        Control whether annotation processing and/or compilation is done.

  -processor <class1>[,<class2>,<class3>...] Names of the annotation processors to run; bypass
es default discovery process
  -processorpath <path>        Specify where to find annotation processors
  -parameters                 Generate metadata for reflection on method parameters
  -d <directory>              Specify where to place generated class files
  -s <directory>              Specify where to place generated source files
  -h <directory>              Specify where to place generated native header files
  -implicit:{none,class}      Specify whether or not to generate class files for implicitly ref-
erenced files
  -encoding <encoding>        Specify character encoding used by source files
  -source <release>           Provide source compatibility with specified release
  -target <release>           Generate class files for specific VM version
```

java 입력

```
C:\Users\BIT>java
사용법: java [-options] class [args...]
          (클래스 실행)
          또는 java [-options] -jar jarfile [args...]
          (jar 파일 실행)
여기서 options는 다음과 같습니다.
  -d32    사용 가능한 경우 32비트 데이터 모델을 사용합니다.
  -d64    사용 가능한 경우 64비트 데이터 모델을 사용합니다.
  -server  "server" VM을 선택합니다..
  기본 VM은 server입니다..

  -cp <디렉토리 및 zip/jar 파일의 클래스 경로>
  -classpath <디렉토리 및 zip/jar 파일의 클래스 경로>
            클래스 파일을 검색할 :(으)로 구분된 디렉토리,
            JAR 아카이브 및 ZIP 아카이브 목록입니다.

  -D<name>=<value>
            시스템 속성을 설정합니다.

  -verbose:[class][c][l][n]
            상세 정보 출력을 사용으로 설정합니다.

  -version   제품 버전을 인쇄한 후 종료합니다.

  -version:<value>
            경고: 이 기능은 사용되지 않으며
            이후 릴리스에서 제거됩니다.

  -showversion  제품 버전을 인쇄한 후 계속합니다.

  -jre-restrict-search | -no-jre-restrict-search
            경고: 이 기능은 사용되지 않으며
            이후 릴리스에서 제거됩니다.
```

Android 개발자 포털과 Android Studio

- ▶ 안드로이드 개발자 포털 : <http://developers.android.com>
 - ▶ 안드로이드 개발을 지원하는 각종 도구들과 도움말을 제공
- ▶ Android Studio
 - ▶ 안드로이드 앱 개발을 위한 공식 통합 개발 환경(IDE)
 - ▶ IntelliJ IDEA를 기반으로 함
 - ▶ 이전에는 Eclipse에 ADT(Android Development Tools)를 설치하여 사용

Android Studio

- ▶ 강력하고 편리한 코드 편집기와 개발자 도구
- ▶ 안드로이드 앱을 빌드할 때 생산성을 높여주는 다양한 기능을 제공
 - ▶ 유연한 Gradle 기반 빌드 시스템
 - ▶ 빠르고 풍부한 기능을 갖춘 에뮬레이터
 - ▶ 모든 Android 기기용 앱을 개발할 수 있는 통합 환경
 - ▶ Instant Run : 새로운 APK를 빌드하지 않고도 변경사항만 실행중인 앱에 푸시
 - ▶ 다양한 코드 템플릿 및 GitHub 통합
 - ▶ 테스트 도구 및 프레임워크
 - ▶ C++ 및 NDK 지원
 - ▶ Google Cloud Platform 기본 지원
 - ▶ Google Cloud Messaging 및 App Engine의 통합을 용이하게

Android Studio 설치

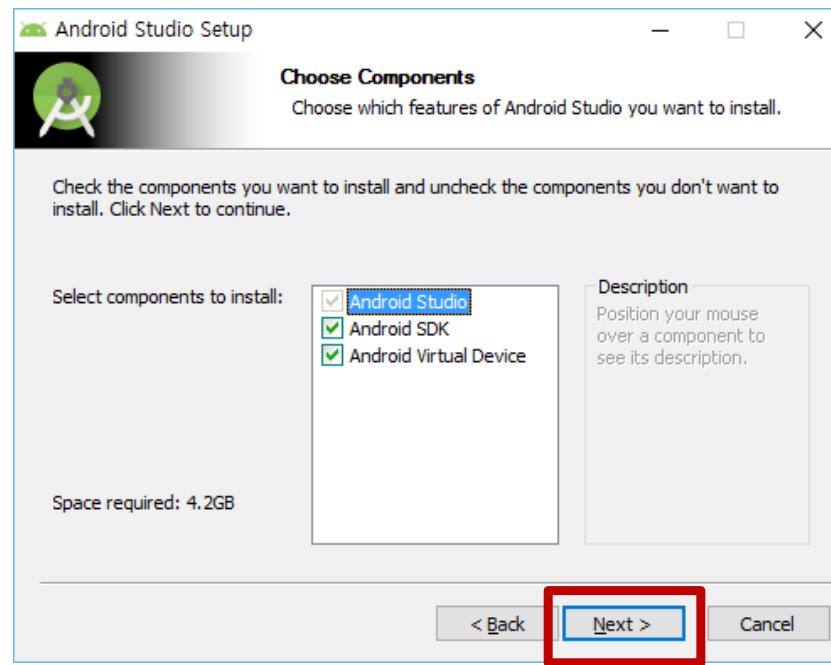
: 다운로드 및 설치

- ▶ 안드로이드 개발자 포털(<https://developer.android.com/studio/index.html>)에서 다운로드 후 설치
- ▶ 설치 과정은 버전/플랫폼 별로 다를 수 있음



Android Studio 설치

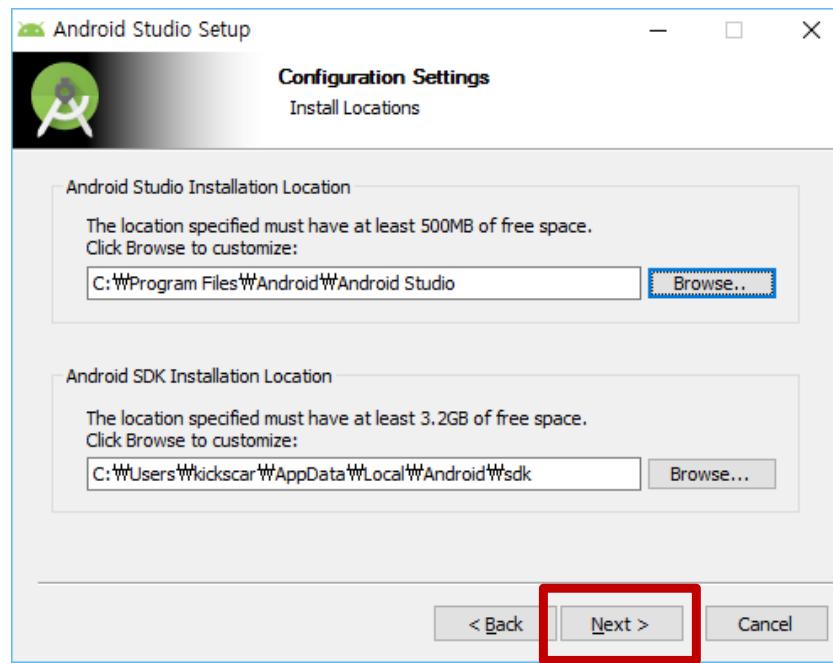
: 설치 구성 요소 선택



Android Studio 설치

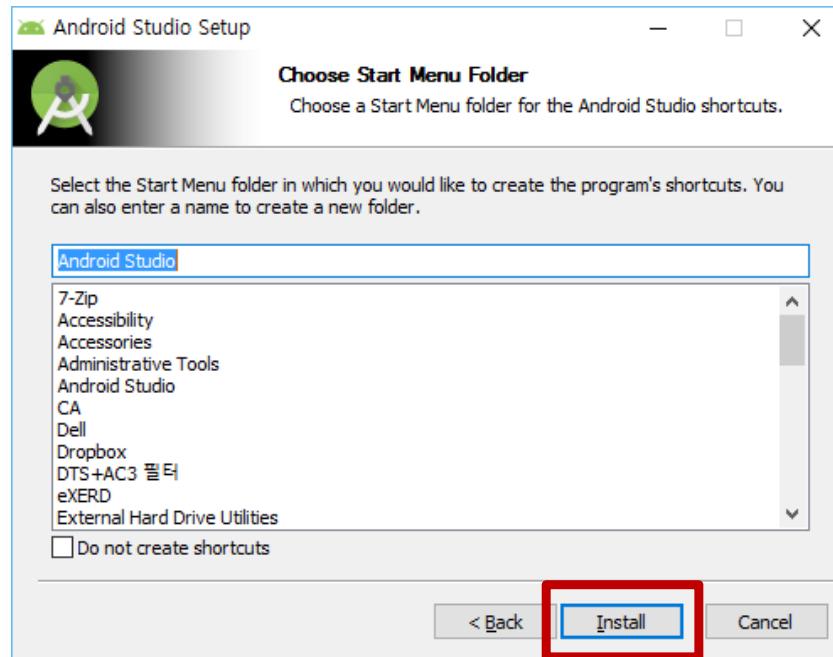
: 설치 위치 지정

- ▶ IDE와 SDK를 다른 위치에 설치하고 기억



Android Studio 설치

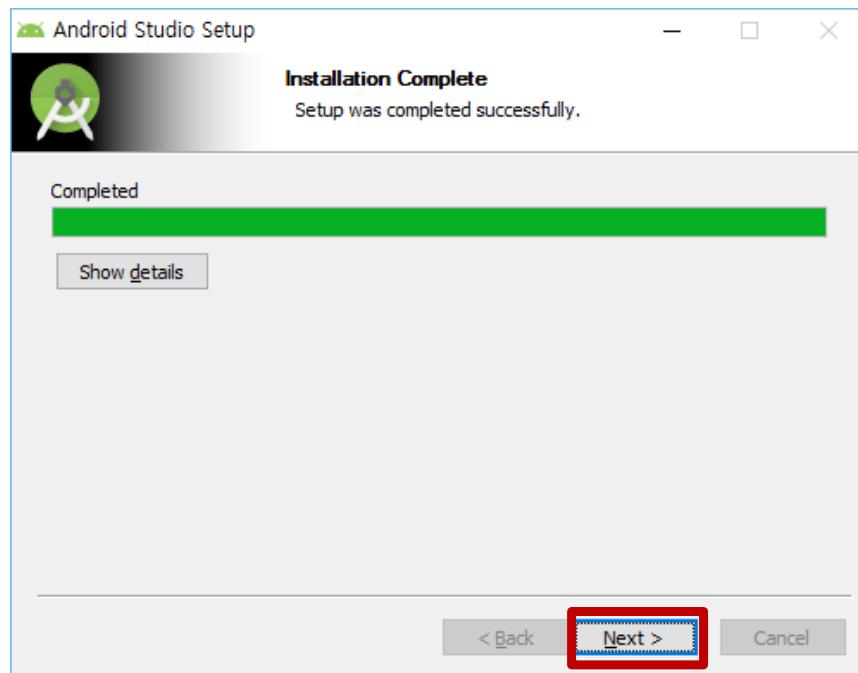
: 시작 메뉴 폴더 지정



Android Studio 설치

: 설치 완료

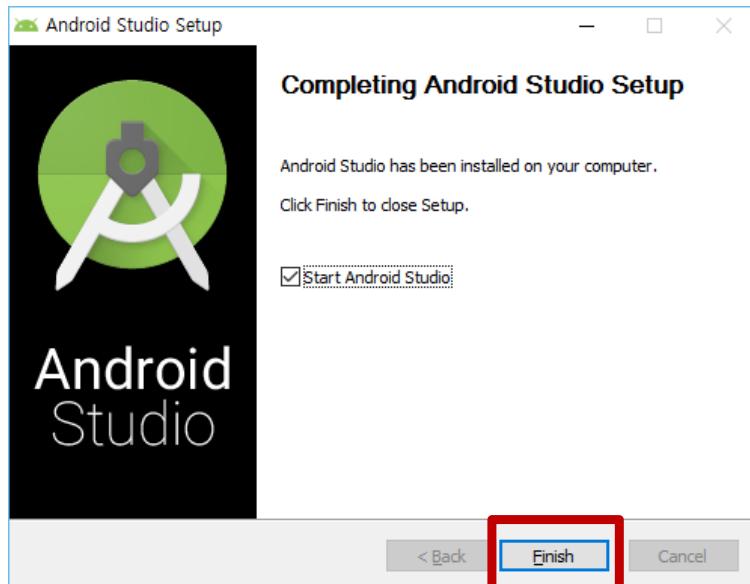
다 끝난 줄 알았으나...



Android Studio 설치

: 일단 실행

- ▶ 설정과 추가 설치가 남아 있습니다



이전 설정은 무시하도록 합니다

A dialog box titled 'Complete Installation' is shown. It contains the message 'You can import your settings from a previous version of Studio.' followed by three radio button options:

- I want to import my settings from a previous version (C:\Users\kickscar\AndroidStudio1.4\config)
- I want to import my settings from a custom location
Specify config folder or installation home of the previous version of Studio:
C:\Users\kickscar\AndroidStudio1.4
- I do not have a previous version of Studio or I do not want to import my settings

At the bottom right is an 'OK' button.

추가 설정의 시작

The 'Welcome to Android Studio' window is displayed. It shows the Android Studio logo and the text 'Android Studio Version 2.1.1'. Below this are five options with icons:

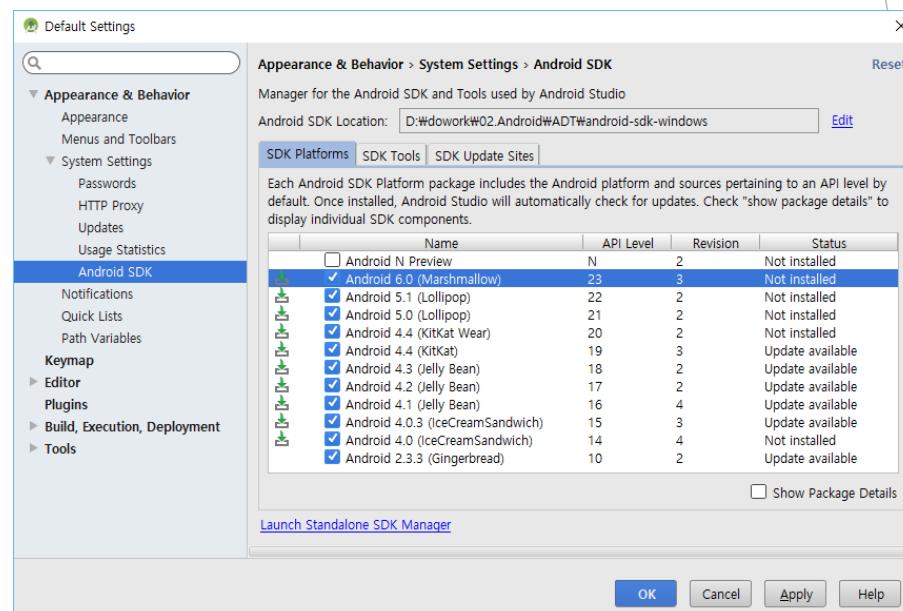
- Start a new Android Studio project
- Open an existing Android Studio project
- Check out project from Version Control ▾
- Import project (Eclipse ADT, Gradle, etc.)
- Import an Android code sample

At the bottom right, there is a 'Configure ▾' button highlighted with a red box, and 'Get Help ▾' next to it.

Android Studio 초기 셋업

: SDK 추가 인스톨

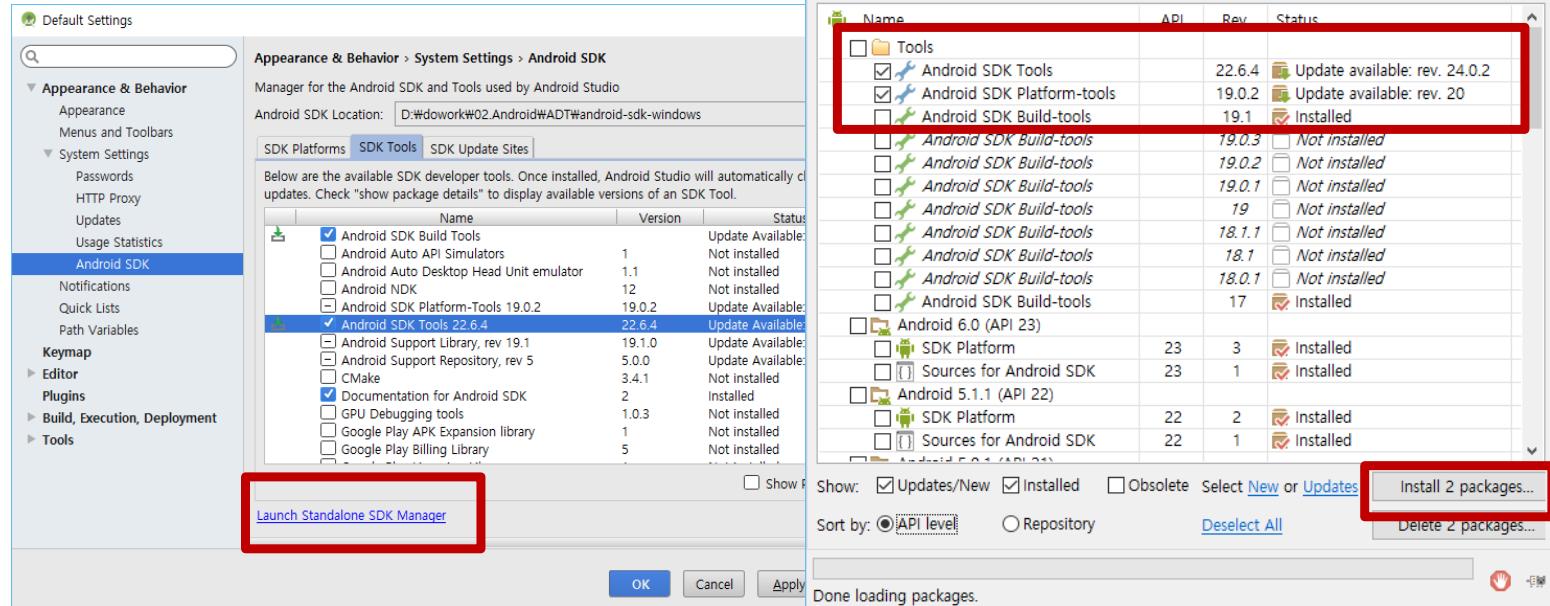
- ▶ Android SDK 항목에서 SDK를 추가 인스톨
- ▶ 권장 인스톨 SDK
 - ▶ 최신 SDK
 - ▶ 6.0 (Marshmallow)
 - ▶ 4.4 (KitKat)



Android Studio 초기 셋업

: Tool Update

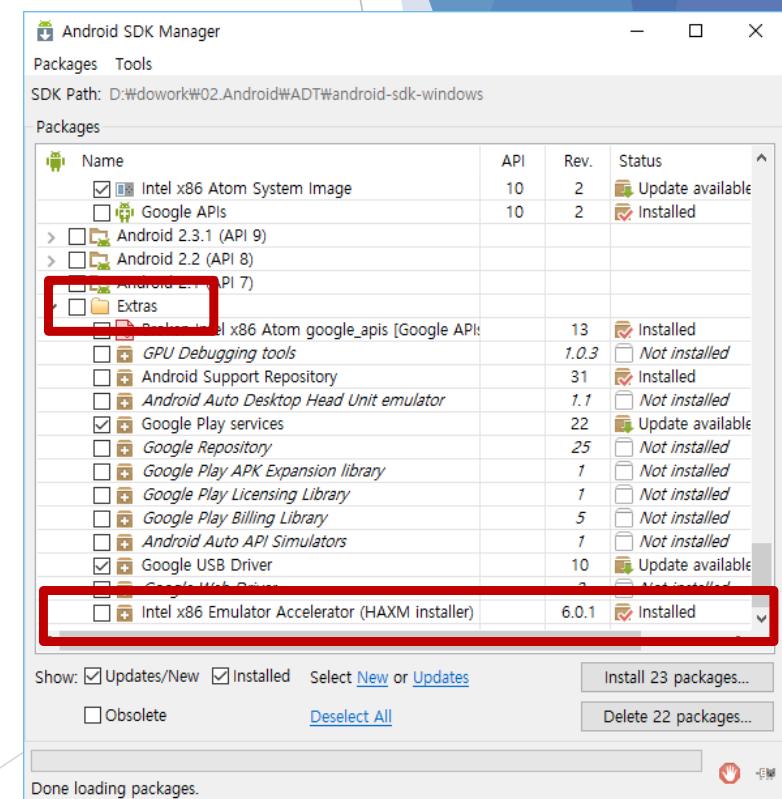
- ▶ Android SDK Build-Tools, Platform-Tools, SDK-Tools 업데이트 및 설치
- ▶ Android Support Repository, Support Library 업데이트



Android Studio 초기 셋업

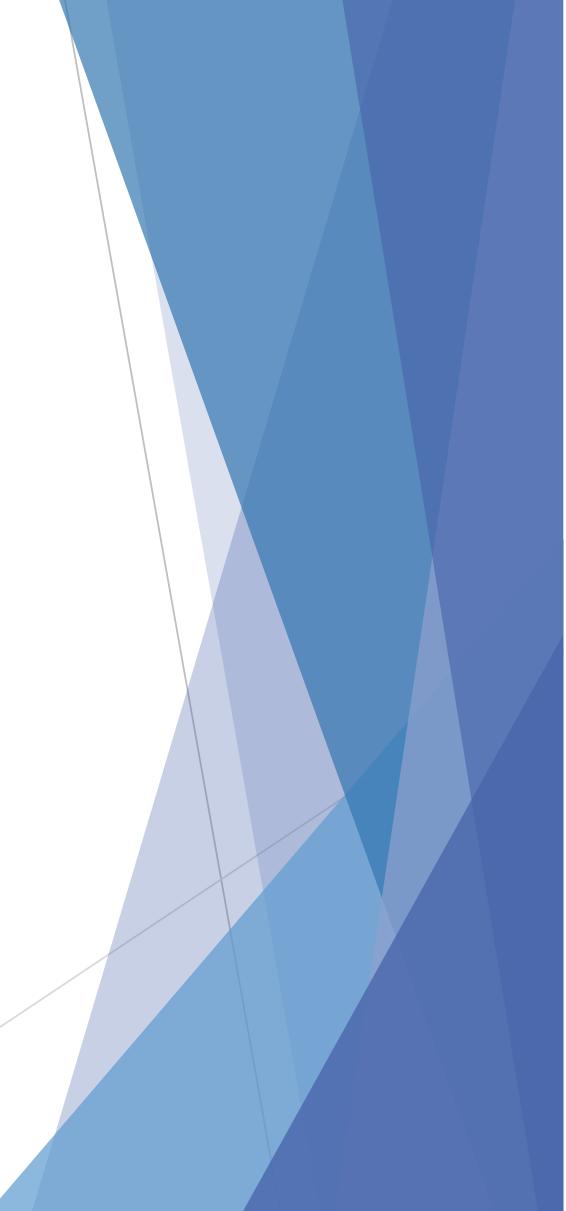
: HAXM(Hardware Accelerated Execution Manager) 설치

- ▶ HAXM : Intel CPU 기반의 Emulator 속도를 개선하기 위해 설치



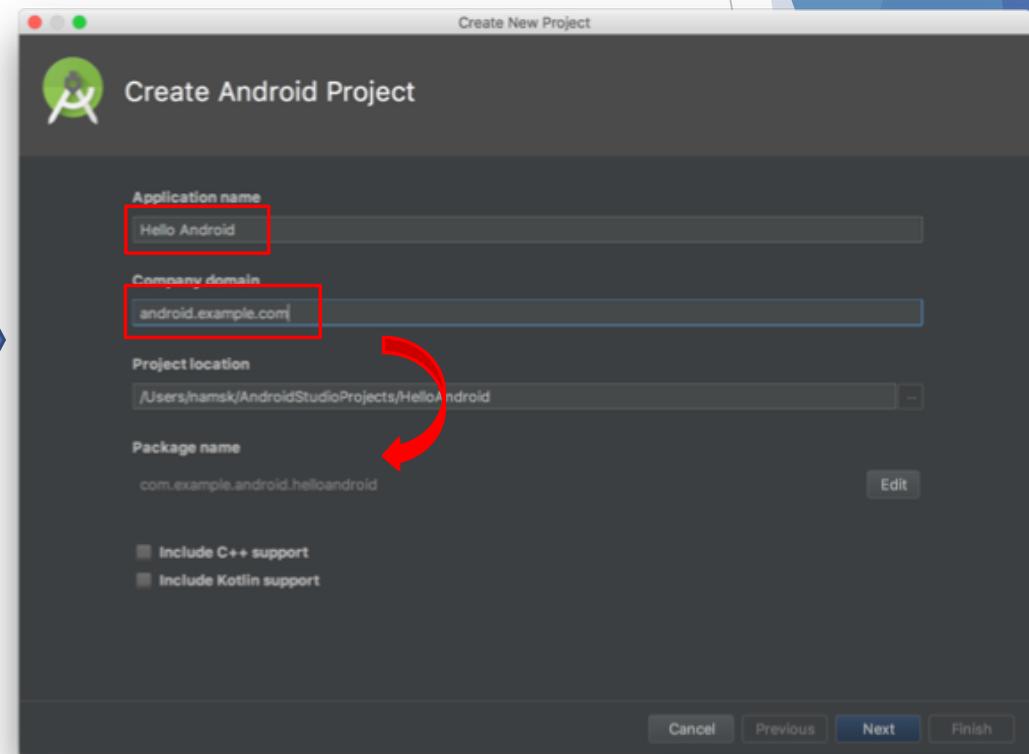
Hello, Android

첫 번째 안드로이드 Application



Hello, Android

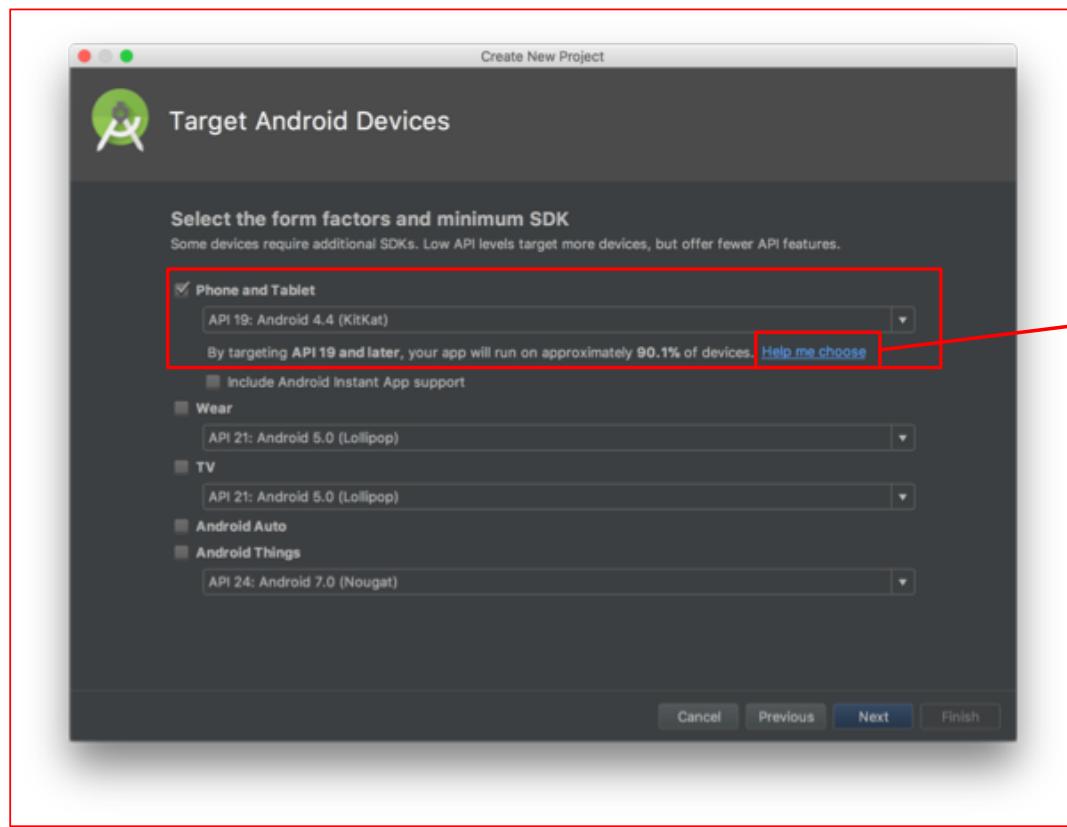
: 프로젝트의 생성



Company Domain과 Application name은
Package name으로 조합되어 사용된다

Hello, Android

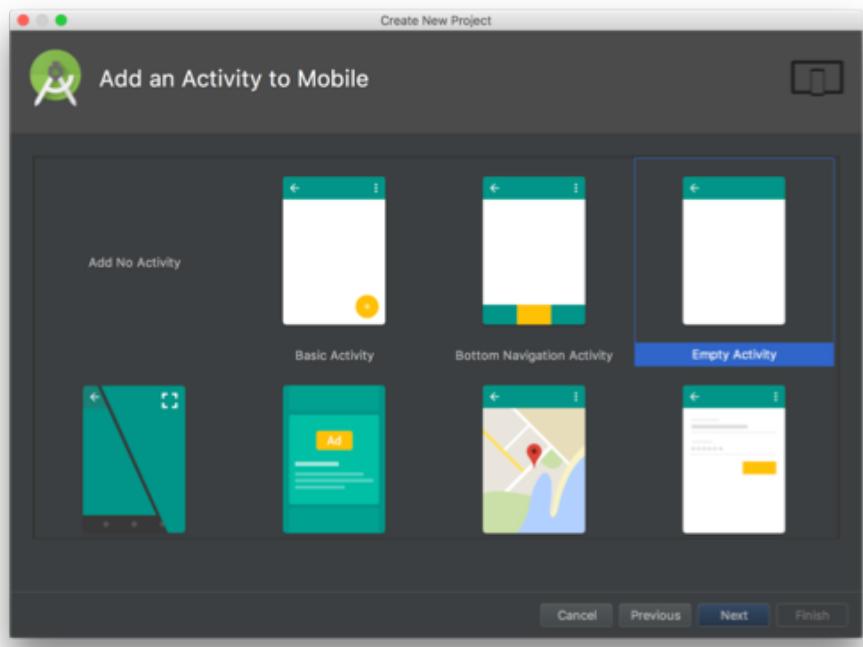
: Target Device의 선택



Target Device 버전 선택을 도와준다

Hello, Android

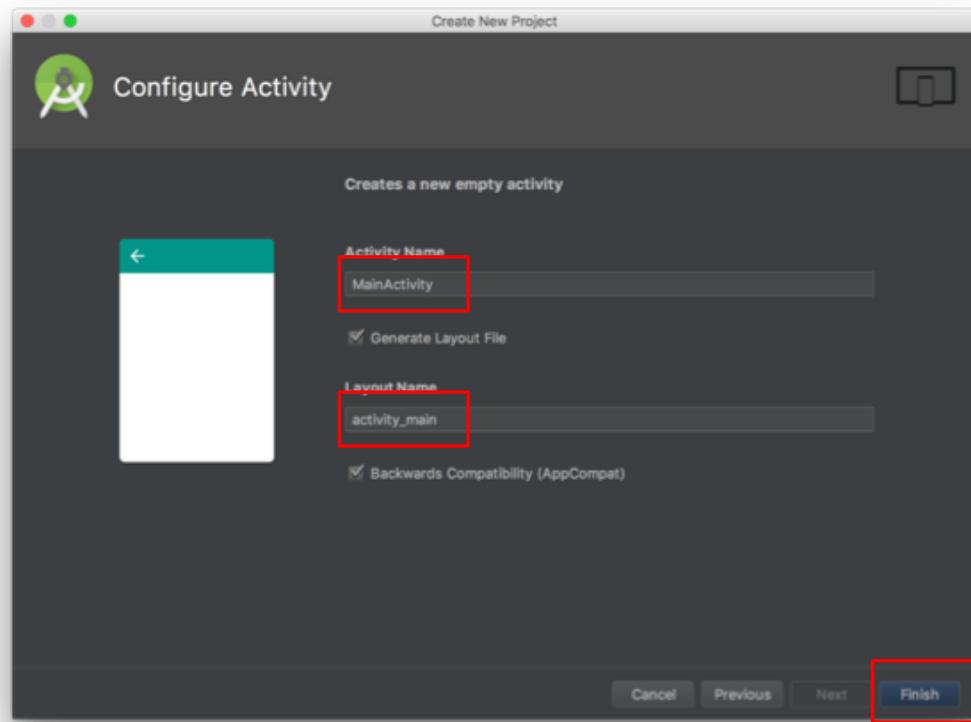
: Template의 선택



- ▶ 학습을 위해 Blank Activity를 선택

Hello, Android

: Activity 설정



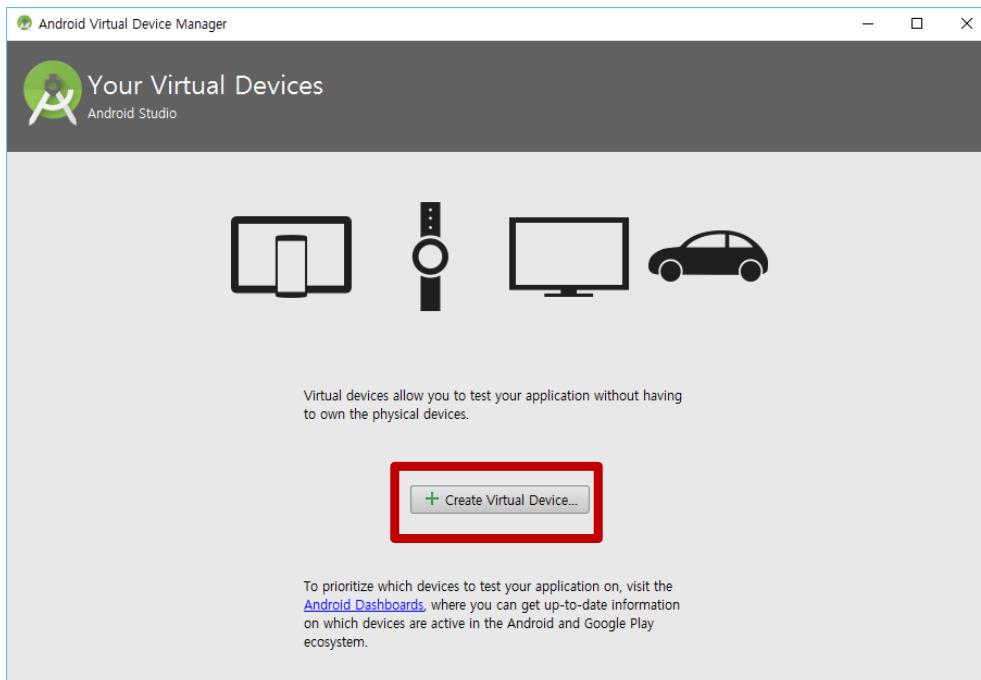
- ▶ Activity 이름과 Layout 이름을 설정
- ▶ Generate Layout File, AppCompat 도 체크해 둔다

앱 기본 틀의 완성

Hello, Android

: AVD(Android Virtual Device) 생성

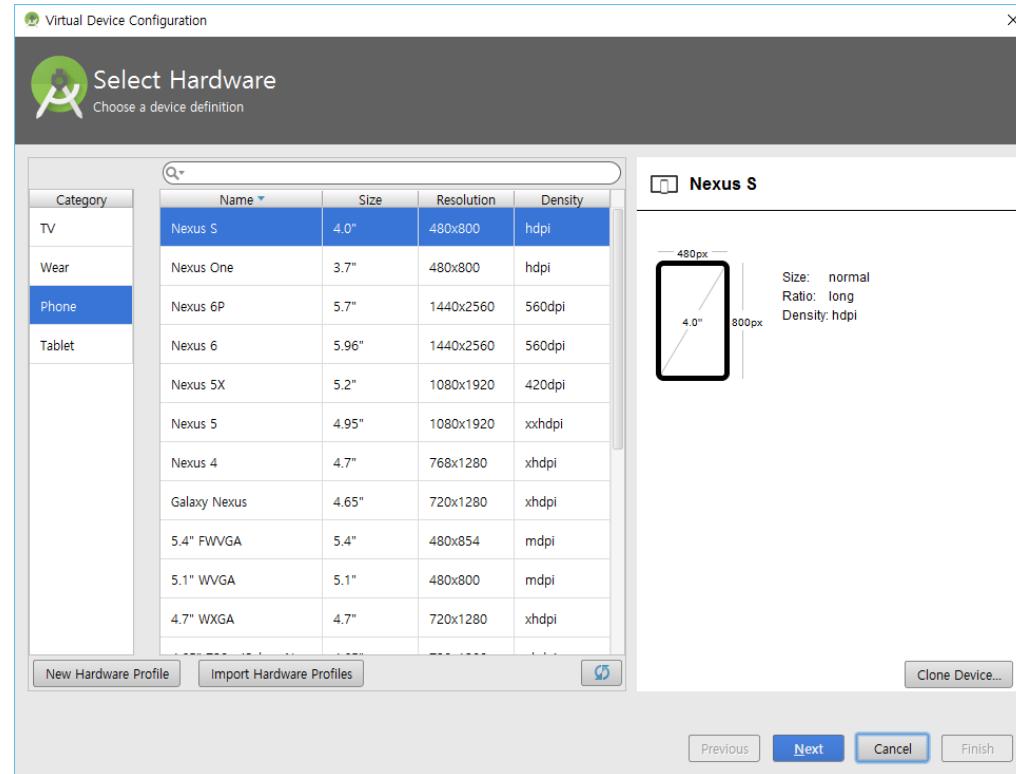
- ▶ AVD라 부르는 가상 장비를 통해 테스트 환경을 구축한다
- ▶ 테스트용 에뮬레이터를 생성하기 위해 AVD 관리자를 실행



Hello, Android

: AVD(Android Virtual Device) 생성

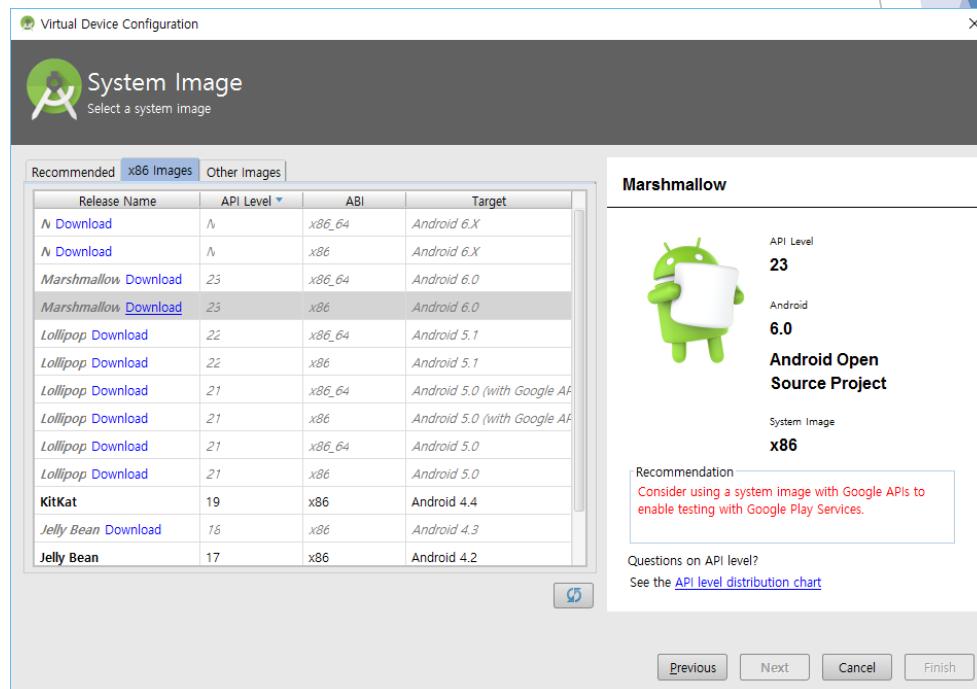
- ▶ 원하는 Device 선택



Hello, Android

: AVD(Android Virtual Device) 생성

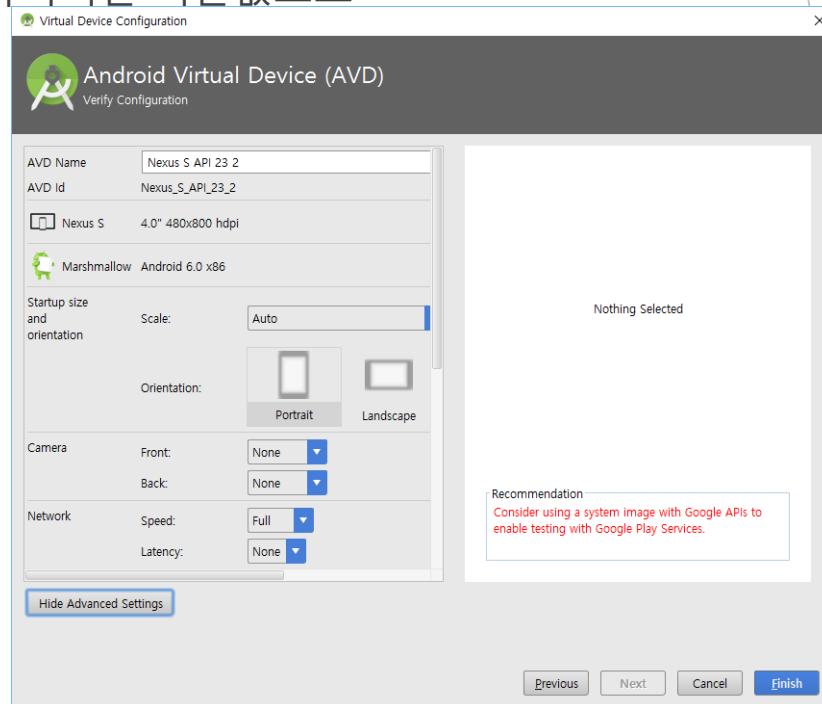
- ▶ 원하는 Device 선택
- ▶ 필요하면 시스템 이미지 다운로드



Hello, Android

: AVD(Android Virtual Device) 생성

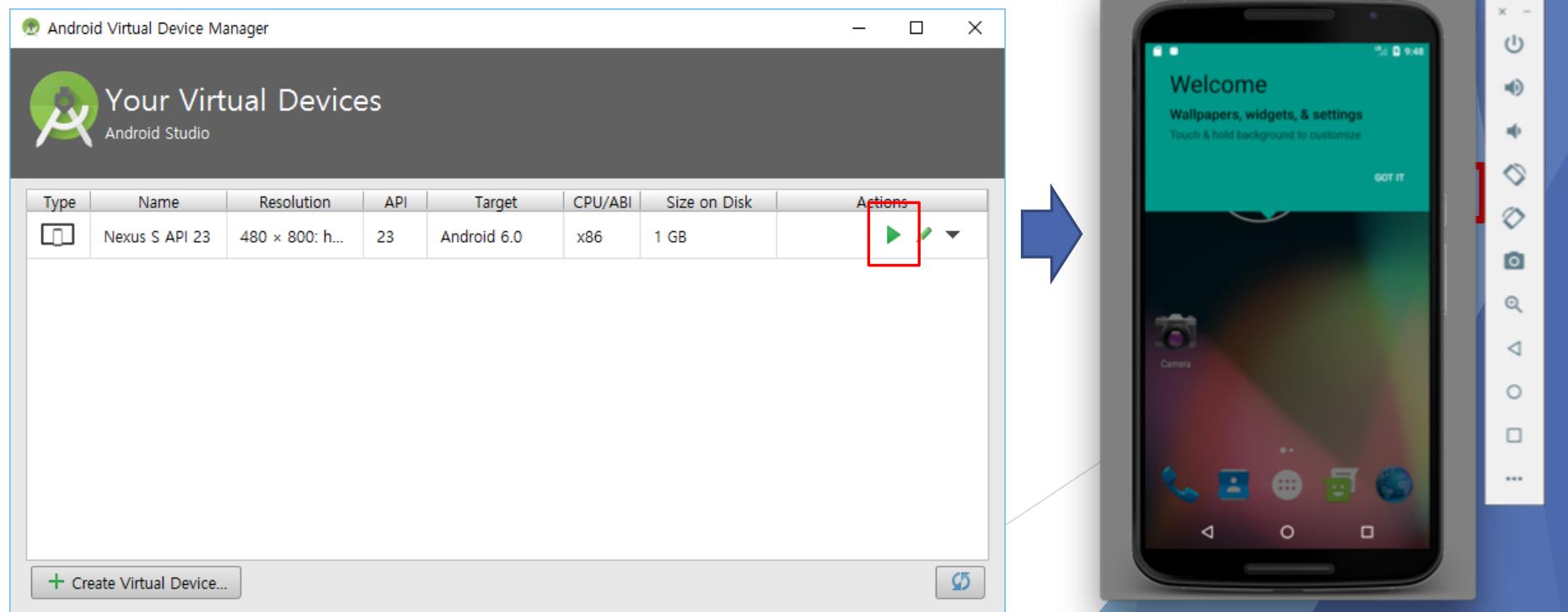
- ▶ AVD 설정 정보 입력
- ▶ AVD 이름과 키보드 설정 해제만 해 주고 나머지는 기본값으로



Hello, Android

: AVD(Android Virtual Device) 생성

- ▶ Emulator 실행

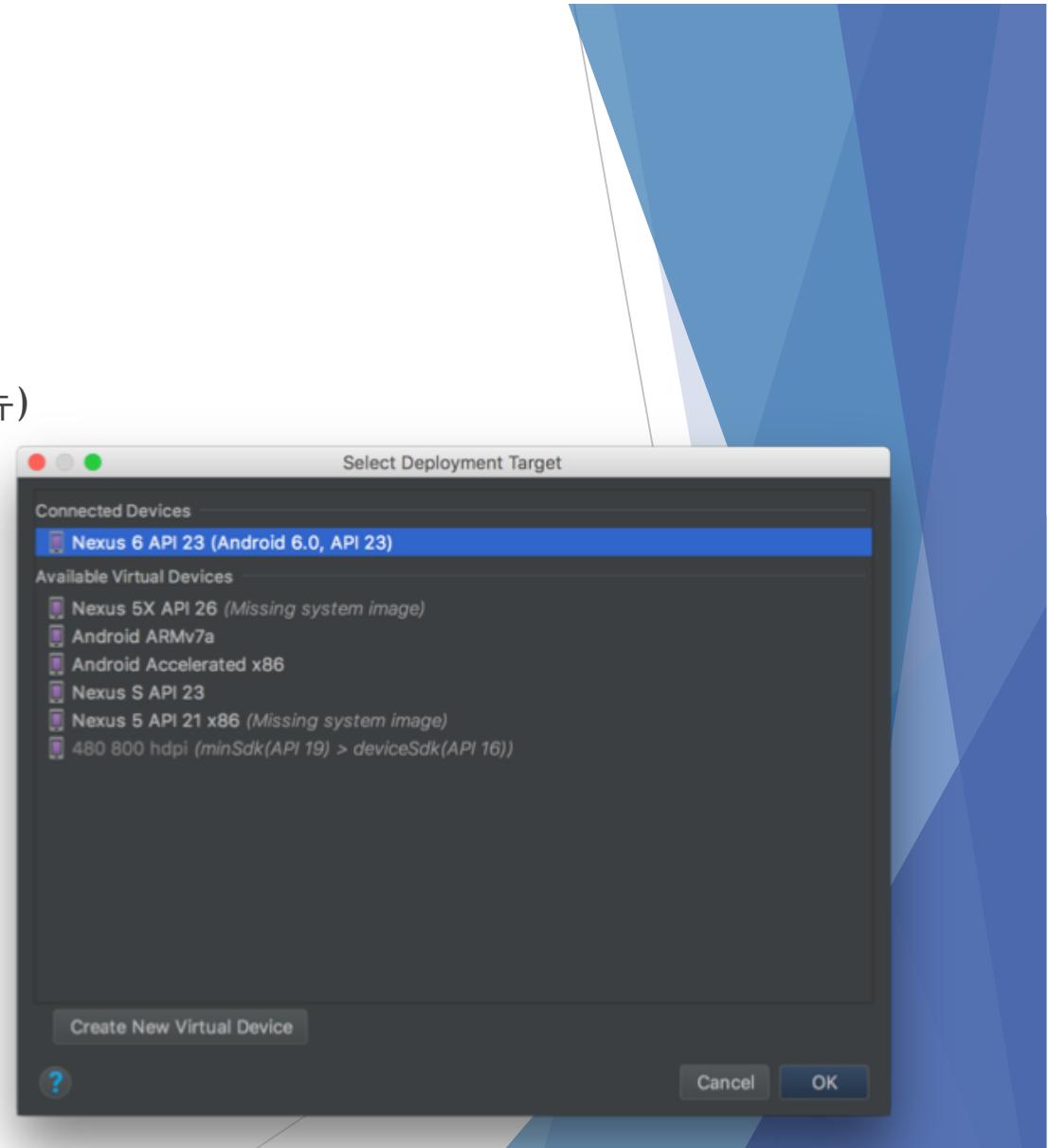


Hello, Android

: 앱을 AVD에서 실행하기

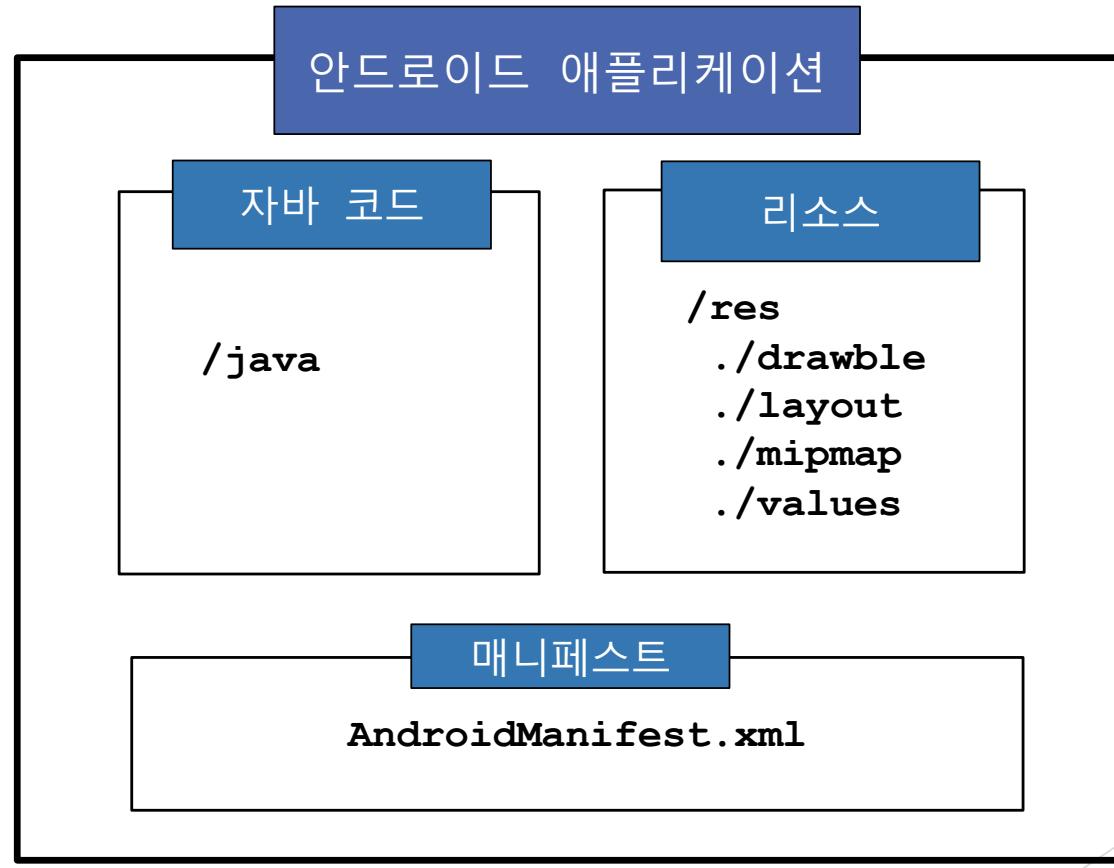
- ▶ 상단 툴바에서 Run App 버튼(Run > Run App 메뉴)
- ▶ 실행할 AVD를 선택
- ▶ 에뮬레이터에서 실행 확인

AVD 실행 후, Setting 앱에서
Language & Input을 한국어로 설정해 줍시다



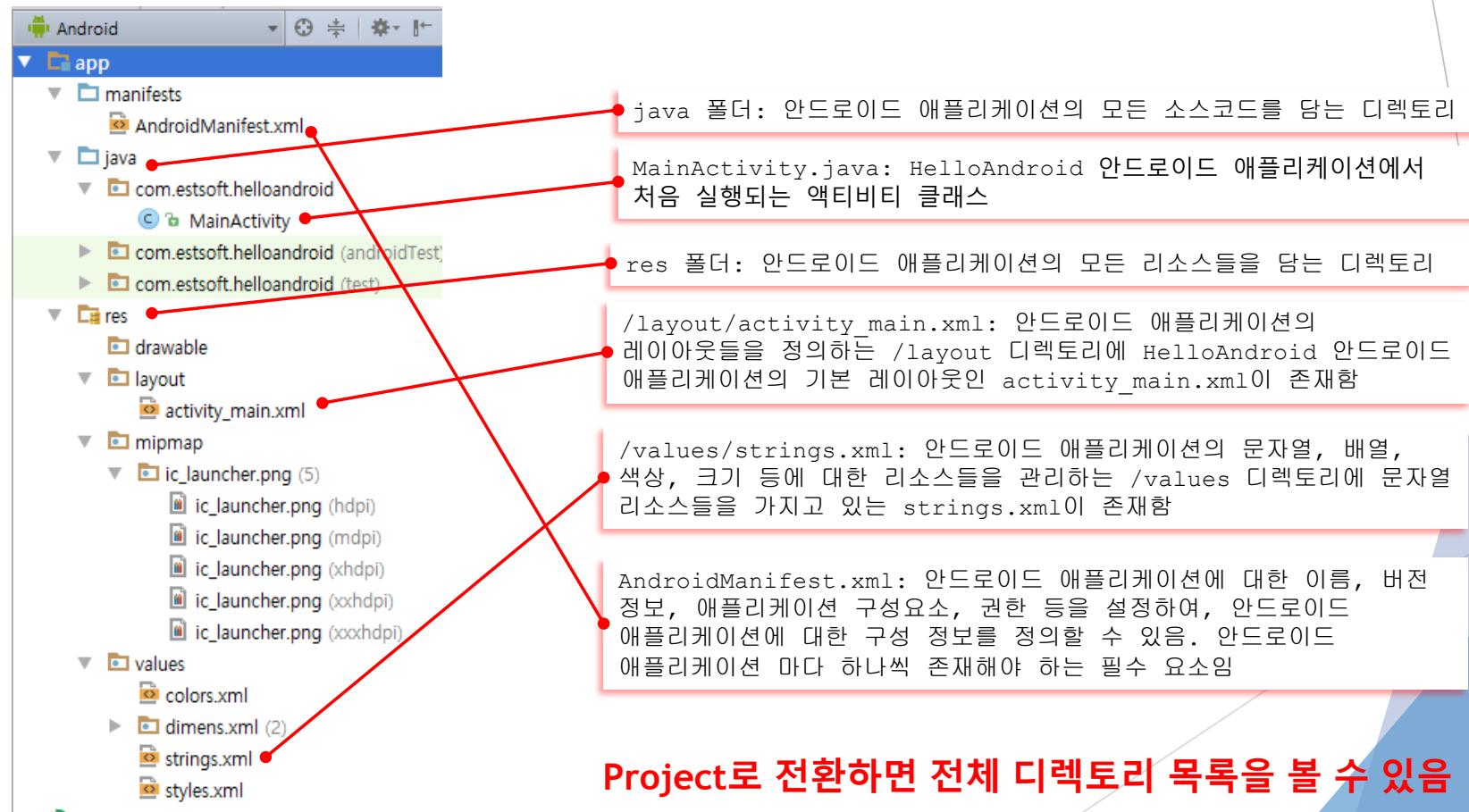
Hello, Android 살펴보기

: 개발 환경 관점에서 본 구성요소



Hello, Android 살펴보기

: 프로젝트의 구조



Hello, Android 살펴보기

: Java 클래스 (in java 폴더)

- ▶ java 폴더: 애플리케이션 구성 요소와 애플리케이션에서 사용되는 소스코드가 저장
- ▶ HelloAndroid의 Main Activity가 설정한 java 패키지 아래 있다
- ▶ MainActivity.java
 - ▶ main() 메서드 없음
 - ▶ 시작점이 여러 곳이 될 수 있다
 - ▶ Activity는 한 화면의 구성 요소

```
public class MainActivity extends AppCompatActivity {

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

Hello, Android 살펴보기

: 리소스 (in res 폴더)

- ▶ 자바 클래스와 함께 애플리케이션을 구성하는 요소. 주로 XML 형태로 저장
- ▶ 리소스가 추가되면 AS가 자동으로 인식하여 리소스 접근 식별자를 정의
 - ▶ 정의된 리소스 ID는 R.java에 클래스로 생성 (임의로 수정하면 안됨)
- ▶ 리소스 이름은 영문 **소문자**, 숫자, 밑줄(_), 마침표(.)만 사용 가능

ANDROID RESOURCE NAMING CHEAT SHEET

by @MOLSJEROEN

<WHAT>_<WHERE>_<DESCRIPTION>_<SIZE>

fixed set of options
choose the right one below

custom part Android view subclass
"all" if reused in ≠ screens

differentiate multiple elements in one screen

always optional
[xdp] or bucket [small]

LAYOUTS

<WHAT>_<WHERE>.XML

<WHAT> is activity, fragment, view, item or layout

e.g. activity_main.xml

STRINGS

<WHERE>_<DESCRIPTION>

e.g. main_intro

all_done

DRAWABLES

<WHERE>_<DESCRIPTION>_<SIZE>

e.g. all_infoicon_small

main_background

IDS

<WHAT>_<WHERE>_<DESCRIPTION>

<WHAT> is name of Android/Custom view class

e.g. linearlayout_main_fragmentcontainer

DIMENSIONS

<WHAT>_<WHERE>_<DESCRIPTION>_<SIZE>

<WHAT> is width, height, size, margin, padding, elevation, keyline or textsize

e.g. keyline_all_text

Hello, Android 살펴보기

: 리소스 종류

- ▶ drawables
 - ▶ in /res/drawable)
 - ▶ 앱에서 사용하는 그림 파일이나 XML 형태로 정의된 그리기 객체를 포함
- ▶ 화면 레이아웃
 - ▶ in /res/layout
 - ▶ 액티비티, 다이얼로그 등에 표시되는 화면의 구성을 담고 있음
 - ▶ 레이아웃 에디터를 이용한 WYSIWYG 방식과 직접 파일을 편집할 수도 있음
 - ▶ XML로 구성되어 있고 위젯을 어떻게 배치할지 지정
 - ▶ 화면 표시 방법에 대한 속성을 지정

Hello, Android 살펴보기

: 리소스 종류

- ▶ values
 - ▶ in /res/values)
 - ▶ 문자열(strings), 색상(colors), 스타일(styles) 등 특정 목적의 값을 가지는 파일을 포함
- ▶ mipmap
 - ▶ in /res/mipmap
 - ▶ 앱을 식별하는 아이콘을 저장하고 있는 리소스

Hello, Android 살펴보기

: 매니페스트

- ▶ 매니페스트 (`AndroidManifest.xml`)
 - ▶ 애플리케이션과 관련된 핵심 정보들을 담고 있다
 - ▶ 패키지 이름, 버전 정보, 앱 이름, 앱 구성요소와 속성이 정의
 - ▶ 앱에서 사용하는 권한, 최소 요구 SDK 버전 등이 정의

Hello, Android

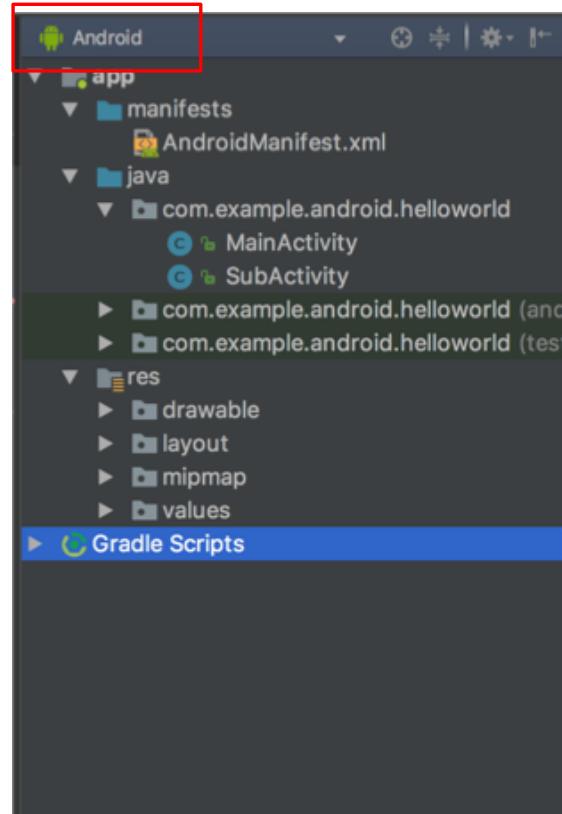
: 남은 내용

- ▶ Hello, Android의 소스에서 Hello World라는 텍스트를 Hello, Android로 변경해 보고 다시 실행해 봅시다
- ▶ 에뮬레이터 툴 버튼들을 한번씩 사용해보고, 에뮬레이터가 어떤 기능을 하는지 익숙해져봅시다 (실제 디바이스 사용 경험과 비교해서)
- ▶ 에뮬레이터 설정에서 Language를 한국어로 설정해 봅시다
- ▶ 에뮬레이터의 Contacts에 연락처 몇 개를 입력해 봅시다
- ▶ AndroidManifest.xml을 열어보고, 어떤 구성 요소들을 포함하고 있는가 확인해 봅시다

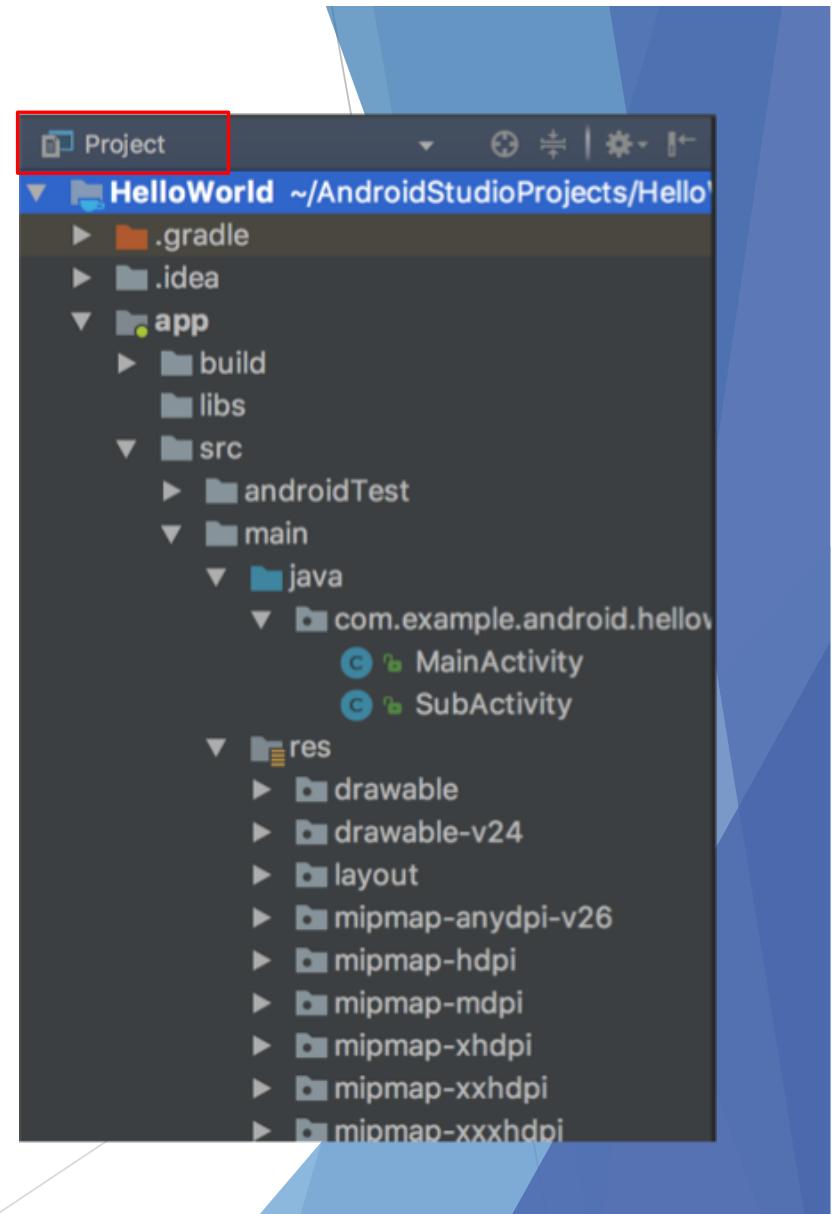
Android Studio



Android Studio



Project 상태일 때는 전체
파일 디렉토리 구조의 내용을
볼 수 있다



지원 화면의 범위

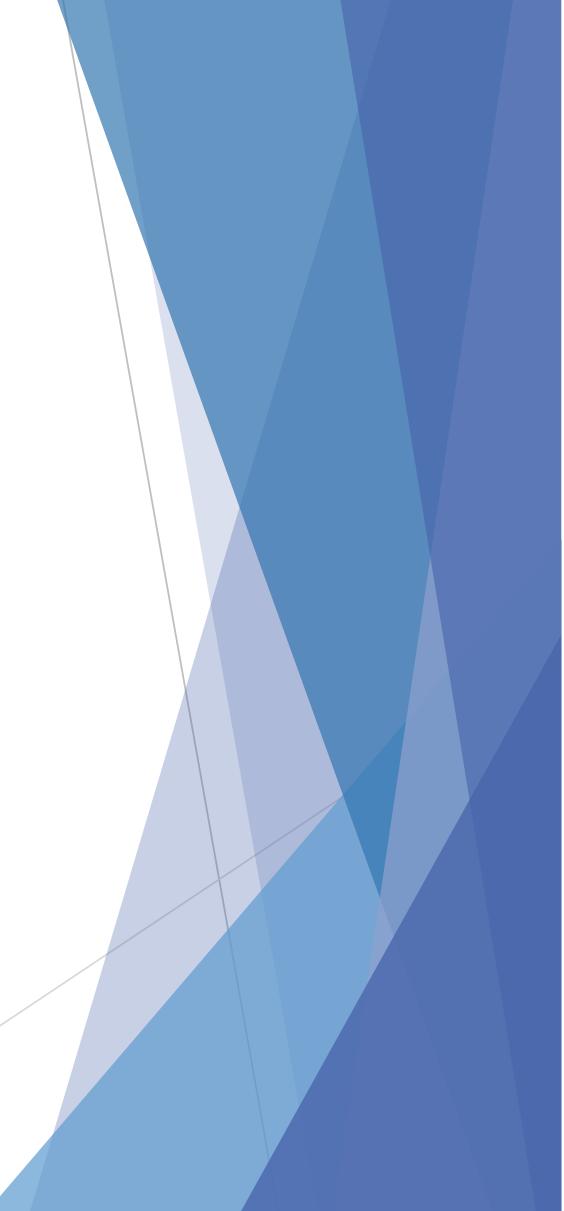
: 크기와 밀도

- ▶ 네 가지 일반화된 크기

- ▶ 소형
- ▶ 보통
- ▶ 대형
- ▶ 초대형

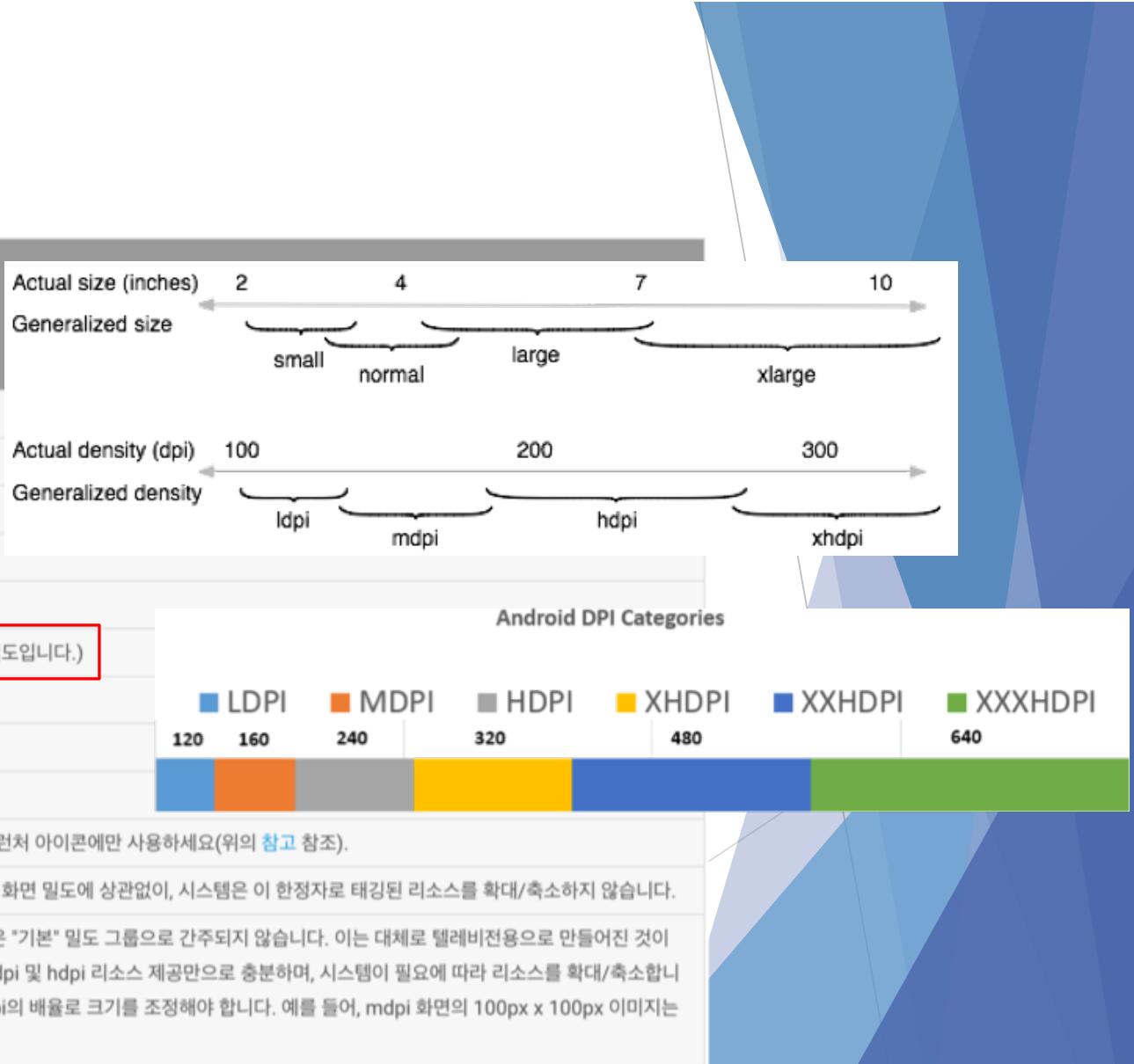
- ▶ 여섯 가지 일반화된 밀도

- ▶ *ldpi* (저밀도) ~120dpi
- ▶ *mdpi* (중간 밀도) ~160dpi
- ▶ *hdpi* (고밀도) ~240dpi
- ▶ *xhdpi* (초고밀도) ~320dpi
- ▶ *xxhdpi* (초초고밀도) ~480dpi
- ▶ *xxxhdpi* (초초초고밀도) ~640dpi



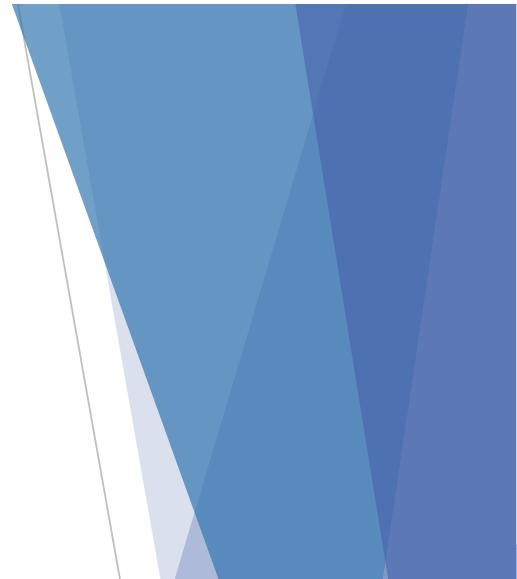
다중 화면 지원

| 화면 특성 | 한정자 | 설명 |
|-------|----------------|---|
| 크기 | small | 소형 크기 화면의 리소스. |
| | normal | 보통 크기 화면의 리소스. (이것이 기준 크기입니다.) |
| | large | 대형 크기 화면의 리소스. |
| | xlarge | 초대형 크기 화면의 리소스. |
| 밀도 | ldpi | 저밀도(<i>ldpi</i>) 화면(~120dpi)의 리소스. |
| | mdpi | 중간 밀도(<i>mdpi</i>) 화면(~160dpi)의 리소스. (이것이 기준 밀도입니다.) |
| | hdpi | 고밀도(<i>hdpi</i>) 화면(~240dpi)의 리소스. |
| | xhdpi | 초고밀도(<i>xhdpi</i>) 화면(~320dpi)의 리소스. |
| | xxhdpi | 초초고밀도(<i>xxhdpi</i>) 화면(~480dpi)의 리소스. |
| | xxxhdpi | 초초초고밀도(<i>xxxhdpi</i>) 화면(~640dpi)의 리소스. 이것은 런처 아이콘에만 사용하세요(위의 참고 참조). |
| | nodpi | 모든 밀도의 리소스. 이들은 밀도 독립적 리소스입니다. 현재 화면 밀도에 상관없이, 시스템은 이 한정자로 태깅된 리소스를 확대/축소하지 않습니다. |
| | tvdpi | mdpi와 hdpi 사이에 있는 화면의 리소스, 약 213dpi. 이것은 "기본" 밀도 그룹으로 간주되지 않습니다. 이는 대체로 텔레비전용으로 만들어진 것이며 대부분의 앱에는 필요하지 않습니다. 대부분의 앱에는 mdpi 및 hdpi 리소스 제공만으로 충분하며, 시스템이 필요에 따라 리소스를 확대/축소합니다. tvdpi 리소스를 제공할 필요가 있는 경우에는 1.33*mdpi의 배율로 크기를 조정해야 합니다. 예를 들어, mdpi 화면의 100px x 100px 이미지는 tvdpi에서 133px x 133px가 되어야 합니다. |



다중 화면 지원

| | | |
|-----|----------------------|---|
| 방향 | <code>land</code> | 가로 모드 방향(가로 방향의 화면비) 화면의 리소스. |
| | <code>port</code> | 세로 모드 방향(세로 방향의 화면비) 화면의 리소스. |
| 화면비 | <code>long</code> | 세로 모드 방향(또는 가로 모드 방향)에서 기준 화면 구성보다 세로 방향(또는 가로 방향)이 훨씬 더 긴 화면의 리소스. |
| | <code>notlong</code> | 기준 화면 구성과 화면비가 유사한 화면의 리소스. |



```
res/layout/my_layout.xml           // layout for normal screen size ("default")
res/layout-large/my_layout.xml     // layout for large screen size
res/layout-xlarge/my_layout.xml   // layout for extra-large screen size
res/layout-xlarge-land/my_layout.xml // layout for extra-large in landscape orientation

res/drawable-mdpi/graphic.png      // bitmap for medium-density
res/drawable-hdpi/graphic.png     // bitmap for high-density
res/drawable-xhdpi/graphic.png    // bitmap for extra-high-density
res/drawable-xxhdpi/graphic.png   // bitmap for extra-extra-high-density

res/mipmap-mdpi/my_icon.png        // launcher icon for medium-density
res/mipmap-hdpi/my_icon.png       // launcher icon for high-density
res/mipmap-xhdpi/my_icon.png      // launcher icon for extra-high-density
res/mipmap-xxhdpi/my_icon.png     // launcher icon for extra-extra-high-density
res/mipmap-xxxhdpi/my_icon.png    // launcher icon for extra-extra-extra-high-density
```

안드로이드는 가장 적합한
리소스를 선택하며
없을 경우, 그보자 작은
사이즈의 리소스를 선택한다

Android Studio

: 디자인 도구

컴포넌트 레이아웃

컴포넌트 트리

디자인 프리뷰

디자인 블루프린트

어트리뷰트

