# 첫 번째 애플리케이션 (1/2)

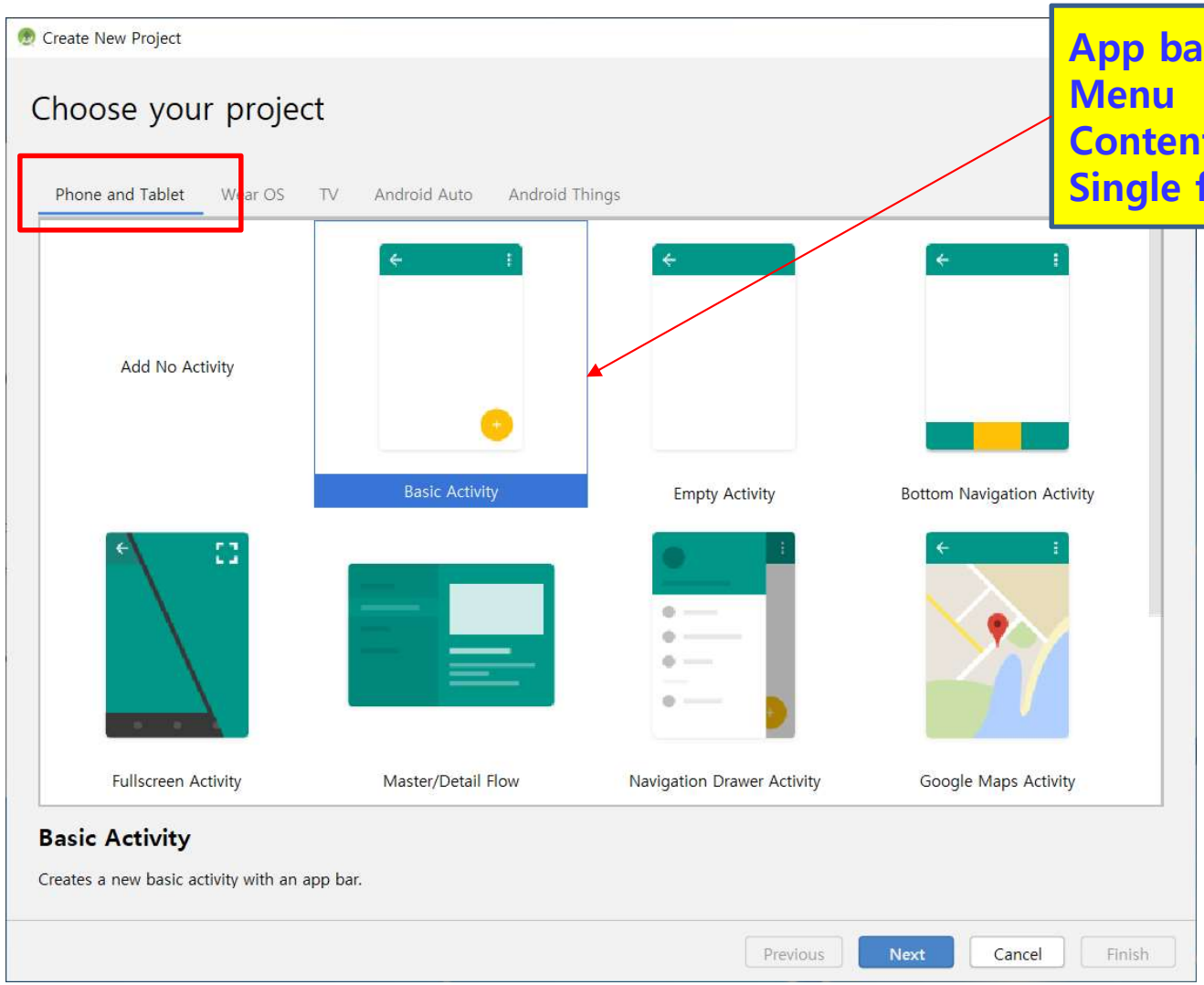
Mobile Software

2019 Fall

# What to do?

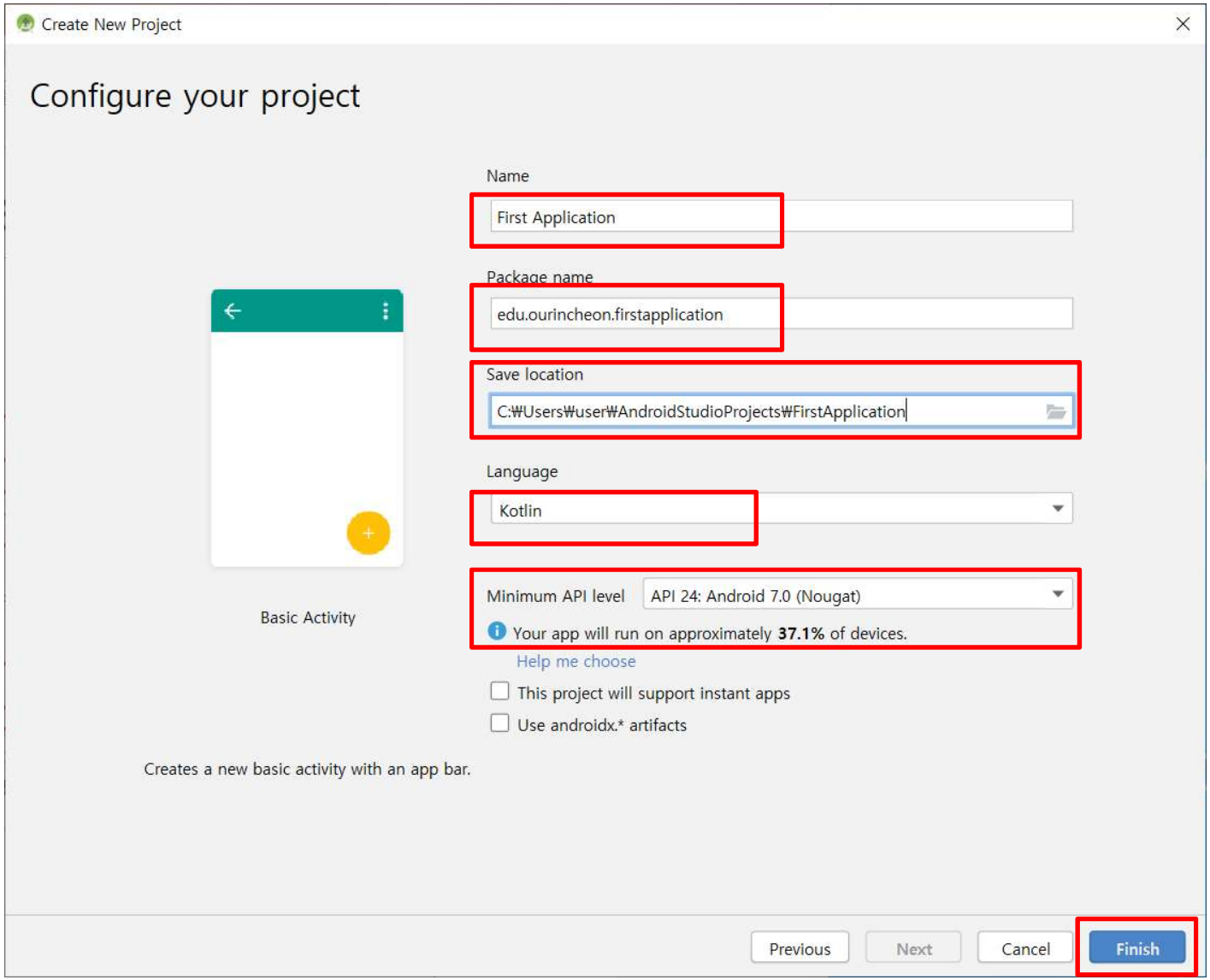
- **First Application 프로젝트 생성과 실행**
  - **Android studio 화면 이해**
  - **AVD를 사용한 App. 실행**
- 실습 1: 출력 문자열을 바꿔보자.
- 소스 코드간 연관성에 대해 알아보자.
- 실습 2: XML 리소스를 소스 코드에서 참조해 보자.

# Creating an Activity: Basic Activity

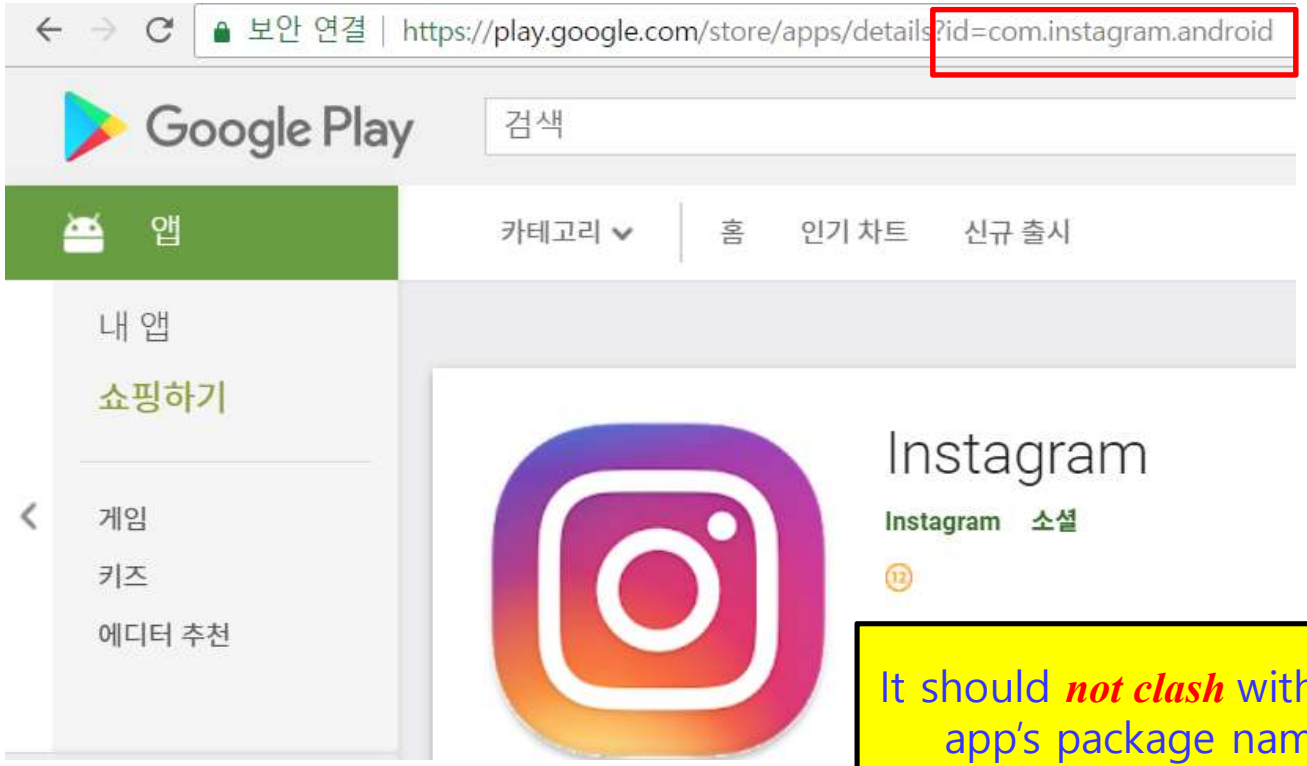



App bar
Menu
Content area
Single floating action button

# Defining the Project and SDK Settings

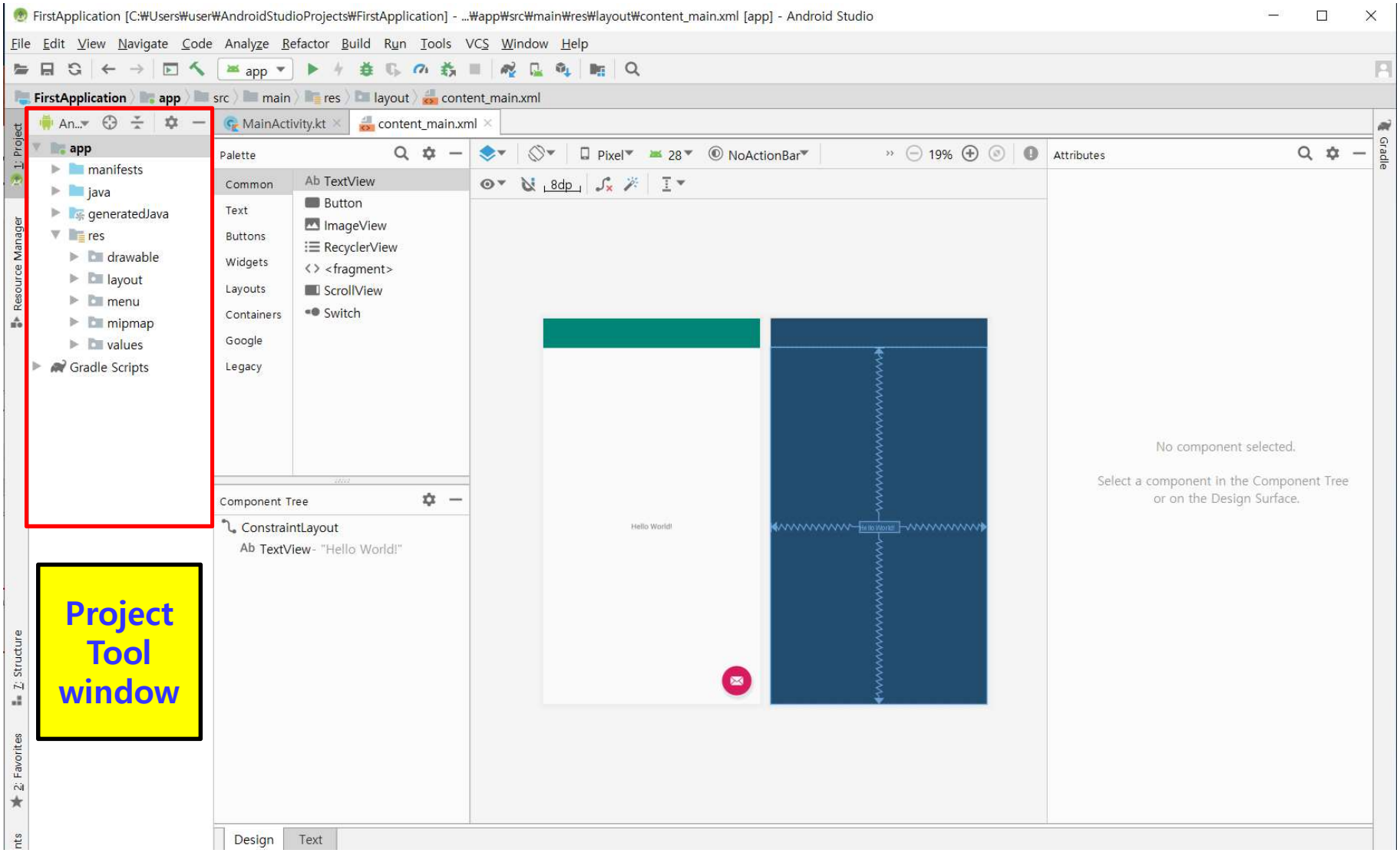


# Package name should be unique

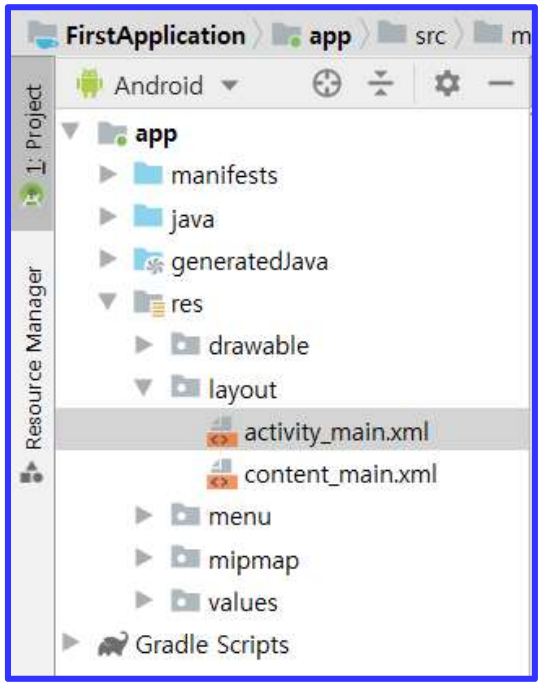



It should ***not clash*** with others app's package name in **Google Play Store.**

# Android Studio 화면 구성



# UI design: activity_main.xml



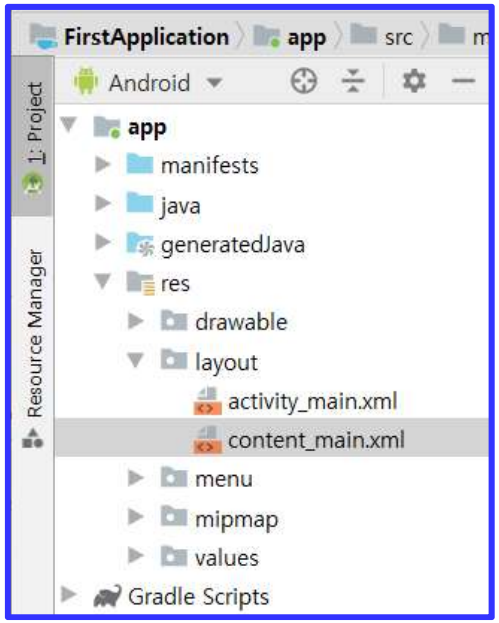


App bar = action bar

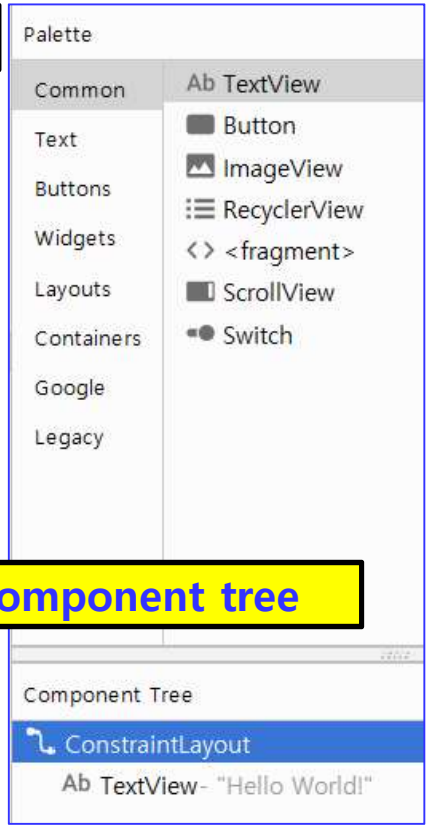
Content layout
➔ content_main.xml

Floating
Action
button

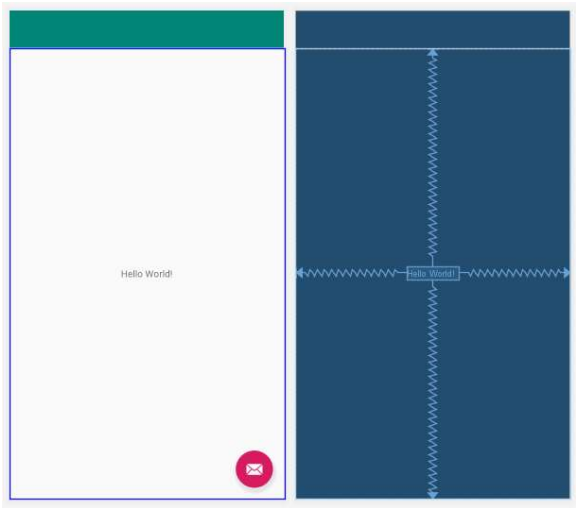
# UI design: content_main.xml



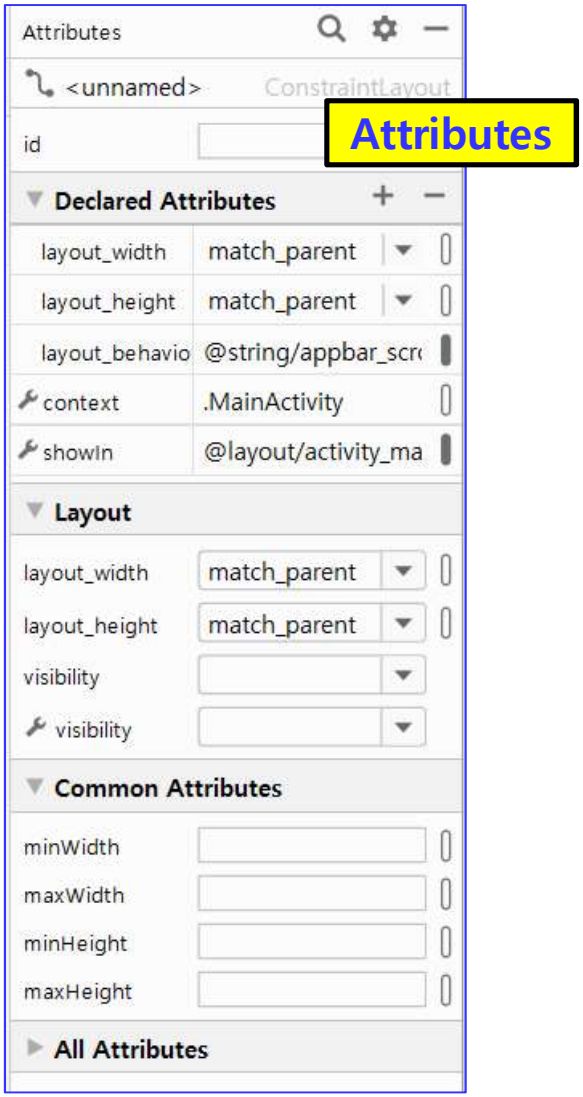

palette




Component tree

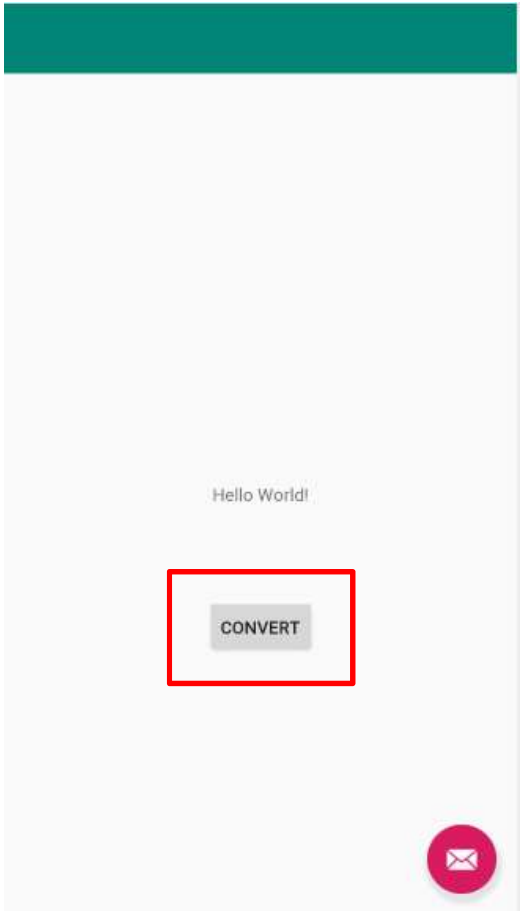


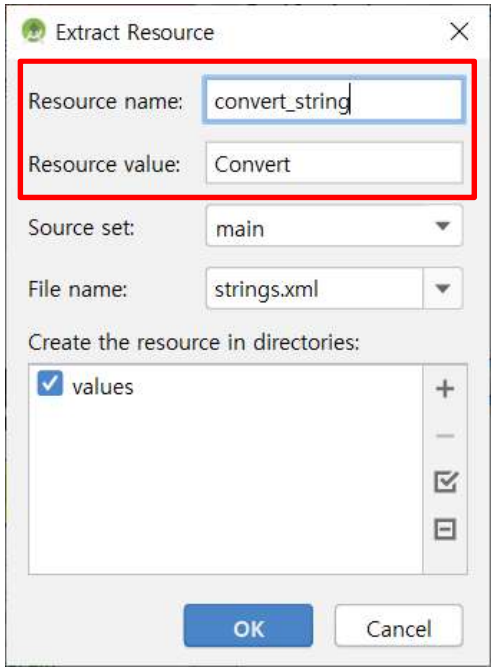

UI layout editor




Attributes

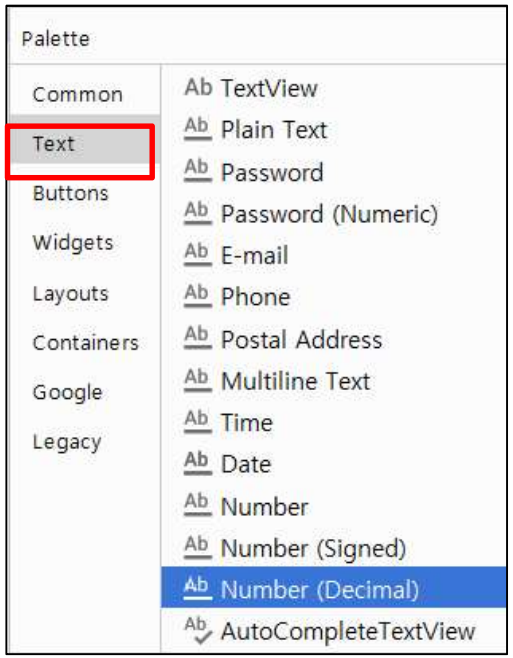
# Add Button

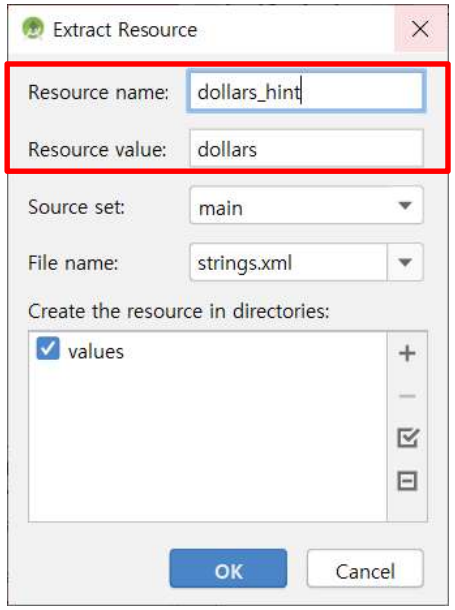

Hello World!
CONVERT

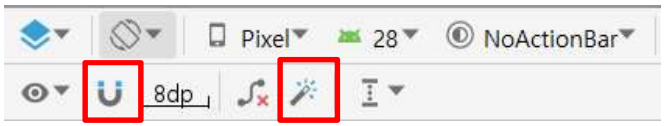

Extract Resource
Resource name:
convert_string
Resource value:
Convert
Source set:
main
File name:
strings.xml
Create the resource in directories:
values
OK
Cancel


Common Attributes
style
@android:styl
stateListAnimat...
@android:anim/butt
onClick
elevation
background
@android:drawa
backgroundTint
backgroundTint...
text
string/convert_string
text

# Add EditText






**dollarText**

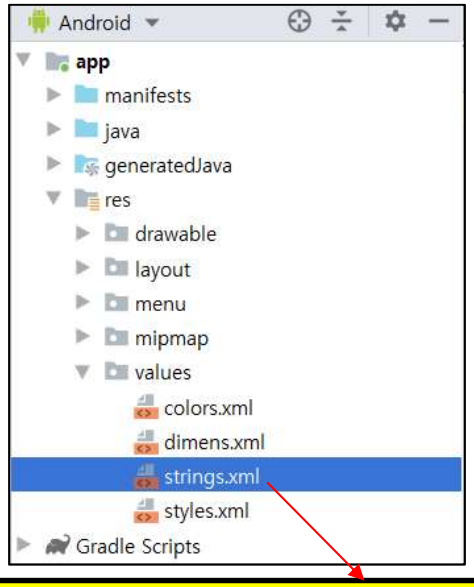
# UI design: content_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="android.support.design.widget.AppBarLayout$ScrollingVi
        tools:showIn="@layout/activity_main"
        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"
            android:id="@+id/textView"/>
    <Button
            android:text="Convert"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            android:id="@+id/button" android:layout_marginTop="73dp"
            app:layout_constraintTop_toBottomOf="@+id/textView"/>
    <EditText
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:inputType="numberDecimal"
            android:ems="10"
            android:id="@+id/editText"
            android:hint="dollars"
            app:layout_constraintStart_toStartOf="parent"
            android:layout_marginBottom="58dp"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintBottom_toTopOf="@+id/textView"/>

</android.support.constraint.ConstraintLayout>
```

Design Text

# Practice 1: Do and see it

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
        xmlns:android="http://schemas.androi
        xmlns:tools="http://schemas.android.
        xmlns:app="http://schemas.android.co
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="android.support
        tools:showIn="@layout/activity_main"
        tools:context=".MainActivity"
        android:background="#ff2438">
```




오른쪽 버튼 >Open Translation Editor

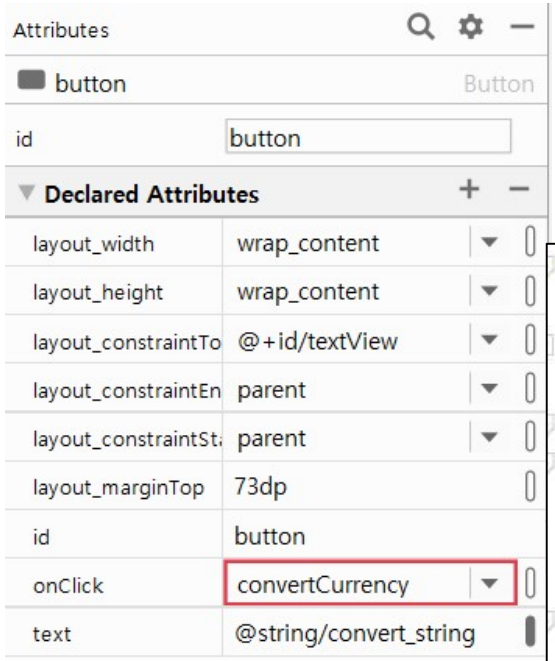
```
<resources>
    <string name="app_name">First Application</string>
    <string name="action_settings">Settings</string>
    <string name="convert_string">Convert</string>
    <string name="dollars_hint">dollars</string>
</resources>
```

Convert to Korean Won

```
<Button
        android:text="@string/convert_string"
        android:layout_width="wrap_content"
```

Ctrl-B

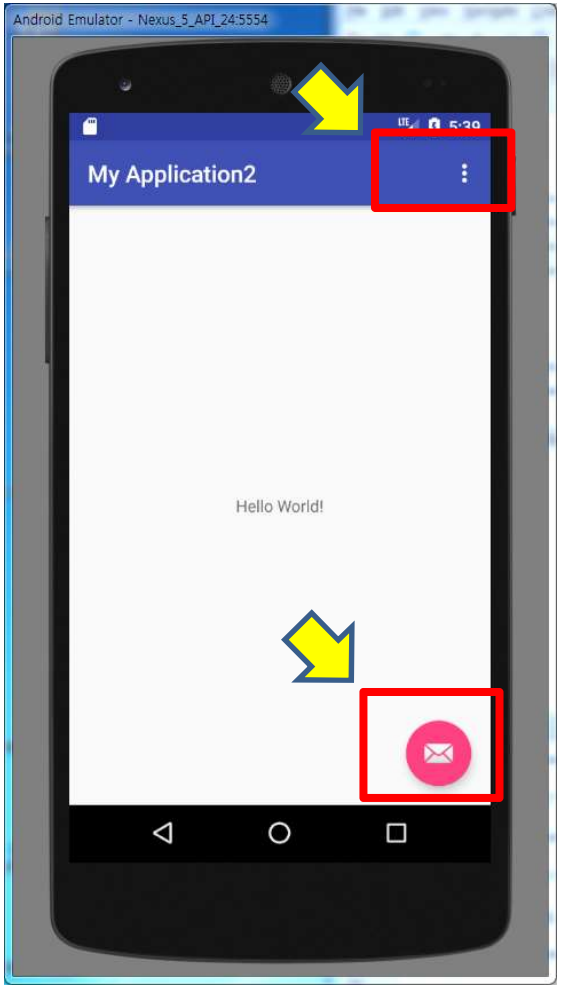
# Android Studio : MainActivity.kt



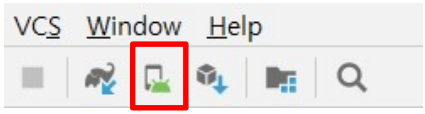

```
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {...}

    fun convertCurrency(view: View) {
        if (dollarText.text.isNotEmpty()) {
            val dollarValue = dollarText.text.toString().toFloat()
            val wonValue = dollarValue * 1127f
            textView.text = wonValue.toString()
        } else {
            textView.text = "No Value"
        }
    }
```

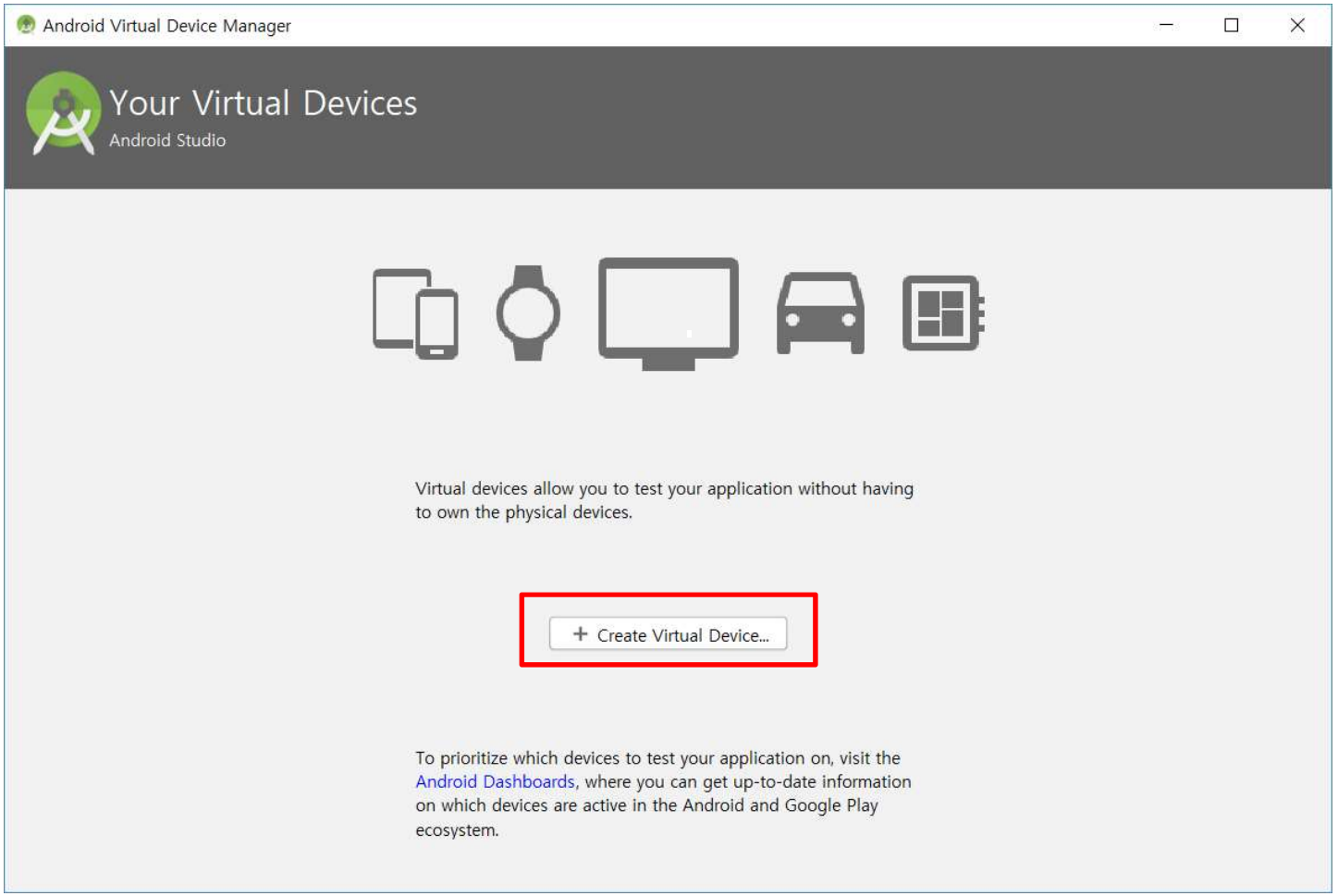
# Basic Activity와 Empty Activity

- **Basic Activity**와 **Empty Activity**는 뭐가 다른가요?
  - **Basic Activity**
    - *Action bar* 와 *floating action bar*가 화면에 포함
    - 2개의 layout 파일과 1개의 메뉴 파일이 자동 생성됨
      - **activity_main.xml**
      - **content_main.xml**
      - **menu_main.xml**
  - **Empty Activity**
    - 빈 화면

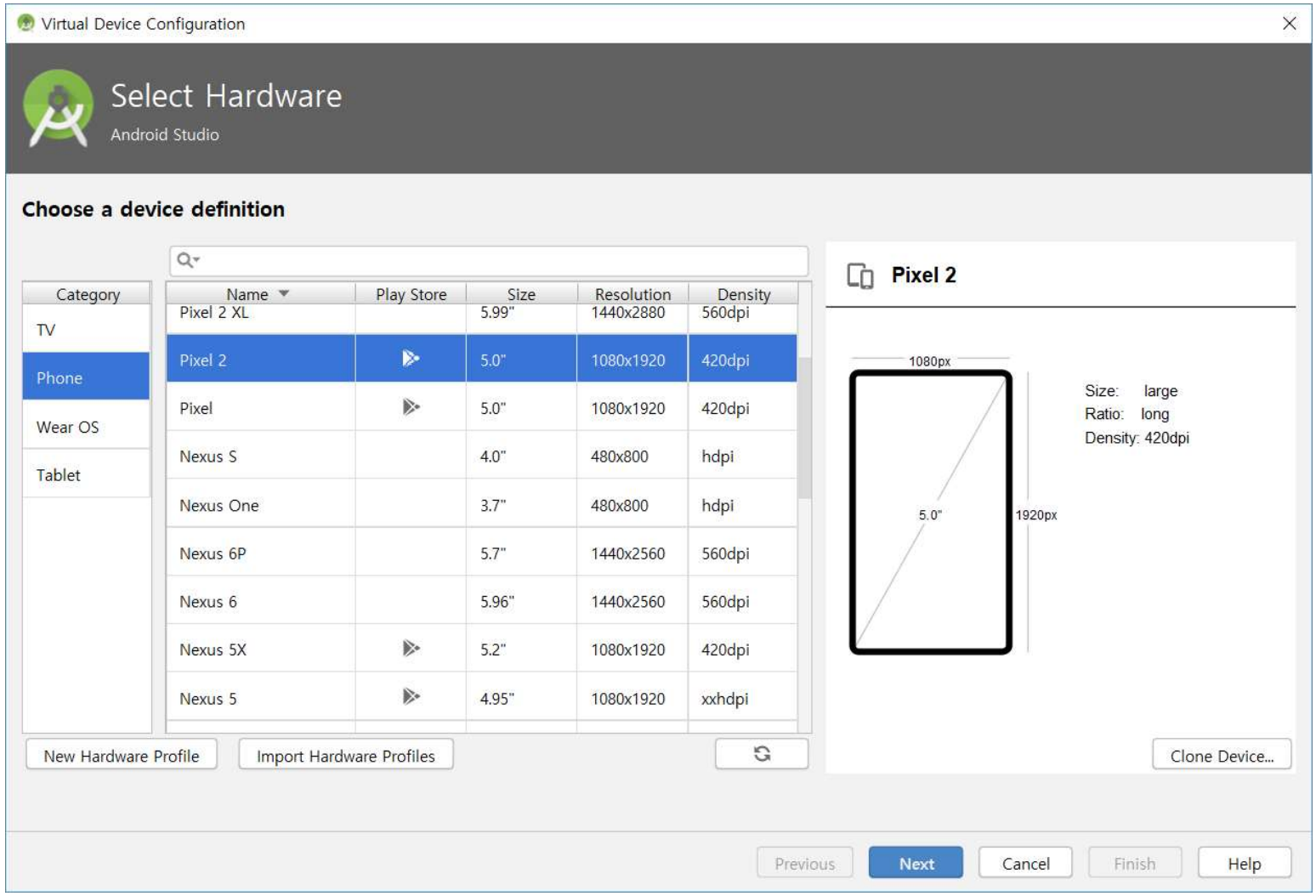


# AVD 만들기(1/4): AVD manger 실행

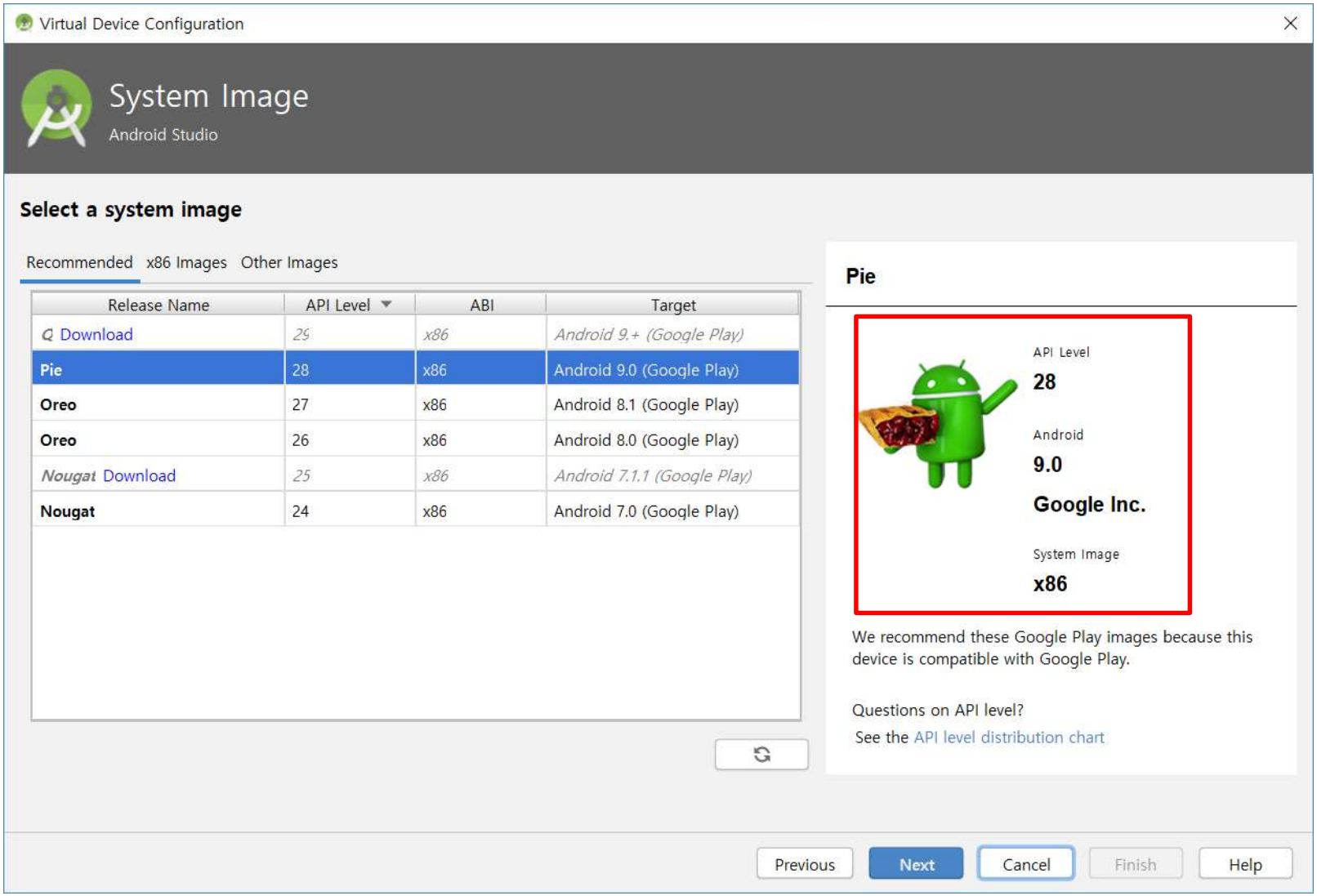


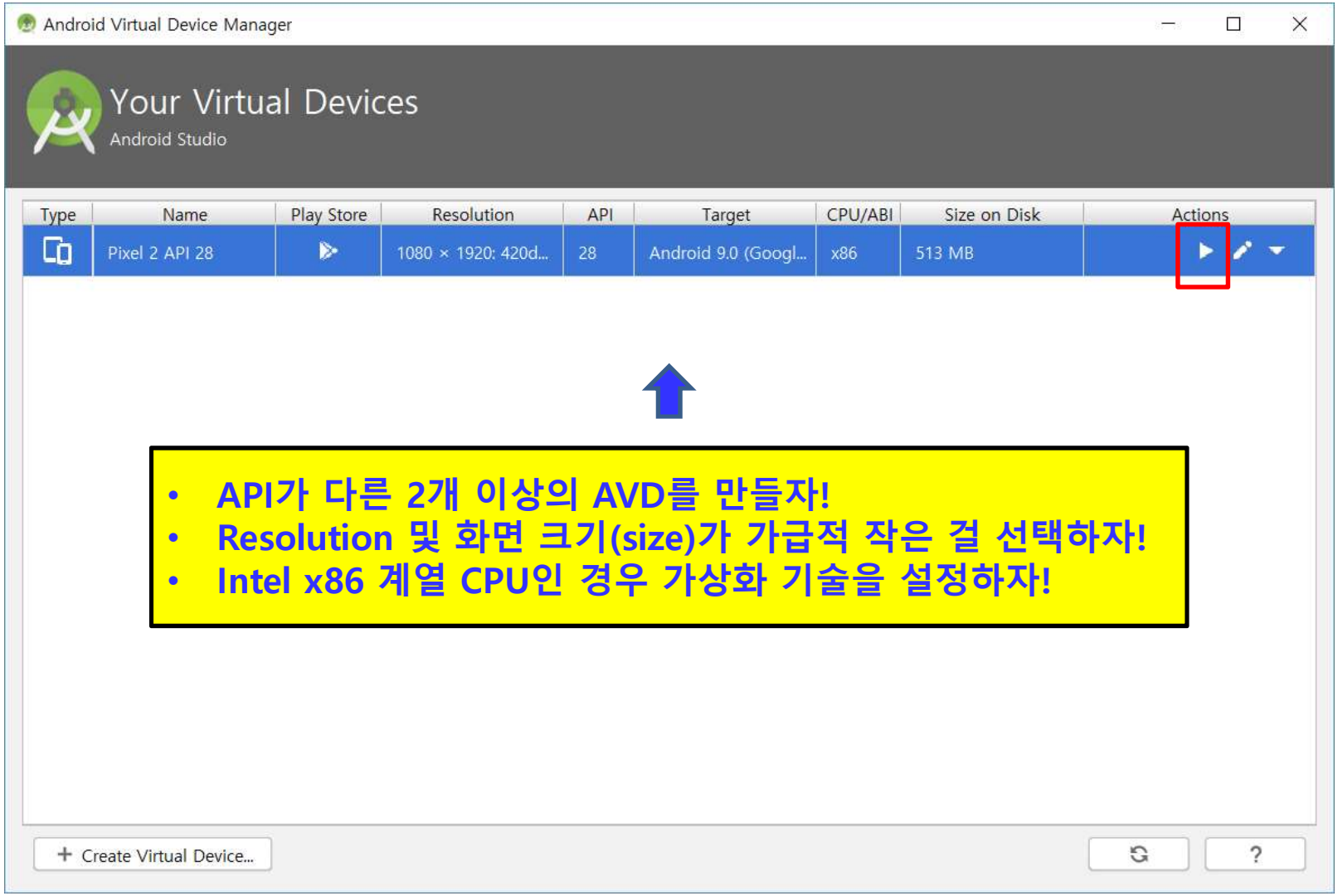
# AVD 만들기(2/4): Pixel 2



# AVD 만들기(3/4): Pie



# AVD 만들기(4/4) : 실행


Android Virtual Device Manager
Your Virtual Devices
Android Studio
Type
Name
Play Store
Resolution
API
Target
CPU/ABI
Size on Disk
Actions
Pixel 2 API 28
1080 × 1920: 420d...
28
Android 9.0 (Googl...
x86
513 MB
• API가 다른 2개 이상의 AVD를 만들자!
• Resolution 및 화면 크기(size)가 가급적 작은 걸 선택하자!
• Intel x86 계열 CPU인 경우 가상화 기술을 설정하자!
+ Create Virtual Device...
?

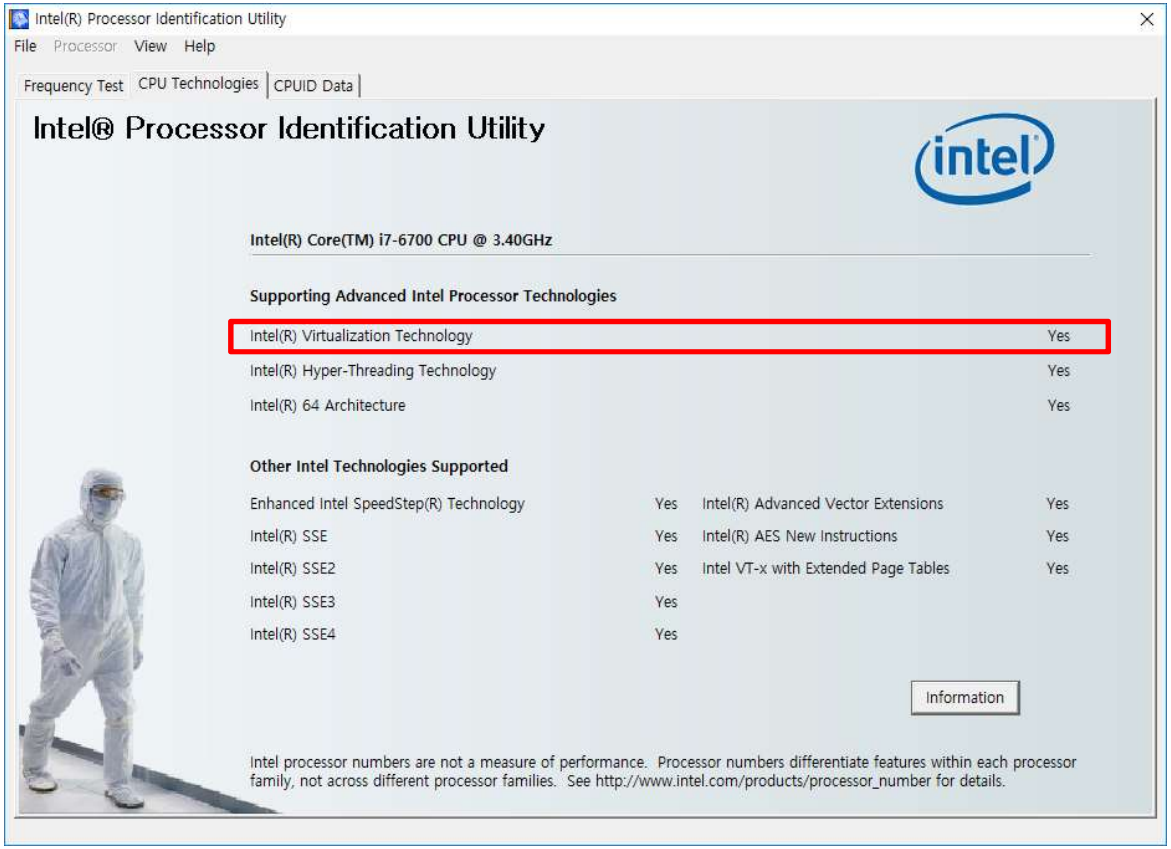
# ABI

- 초기 개인용 PC는 Apple Mac을 제외하면 OS나 컴파일러 없이 출시
  - IBM사는 IBM-PC compatible과 같이 하드웨어 표준만 제시
    - 하드웨어 확장 및 소프트웨어 표준은 시장 경쟁에 맡김
  - Microsoft가 Windows OS와 Visual C++ 등으로 시장 지배
  - Borland, Novell 외에 gcc와 같은 open source project 도 등장
- Android 에서도 제조사에 따라 단말에 장착되는 CPU (**ARM, AMD, x86**)가 다름
  - 기계어 코드 간 호환성을 위해 **낮은 수준** 인터페이스가 필요

- **ABI : Application Binary Interface**
  - 다양한 종류의 CPU와 android 시스템 간 인터페이스 정의
- **EABI** : **Embedded ABI**
  - ARM 계열 CPU가 대표적

- **AMD** : Advanced Micro Devices
  - PC용 CPU 시장에서 Intel사의 강력한 라이벌
- **x86과 x86_64**
  - **x86 :** Intel 계열 CPU이름에서 유래(8086, 80186, … )
  - **x64 :** AMD에서 x86을 64 bits로 확장하고 붙인 이름
    - 32bits CPU와 64bits CPU를 구분할 목적으로 사용
  - **x86**은 Intel 계열 CPU를, **x86_64**는 AMD 계열 CPU를 가리킴

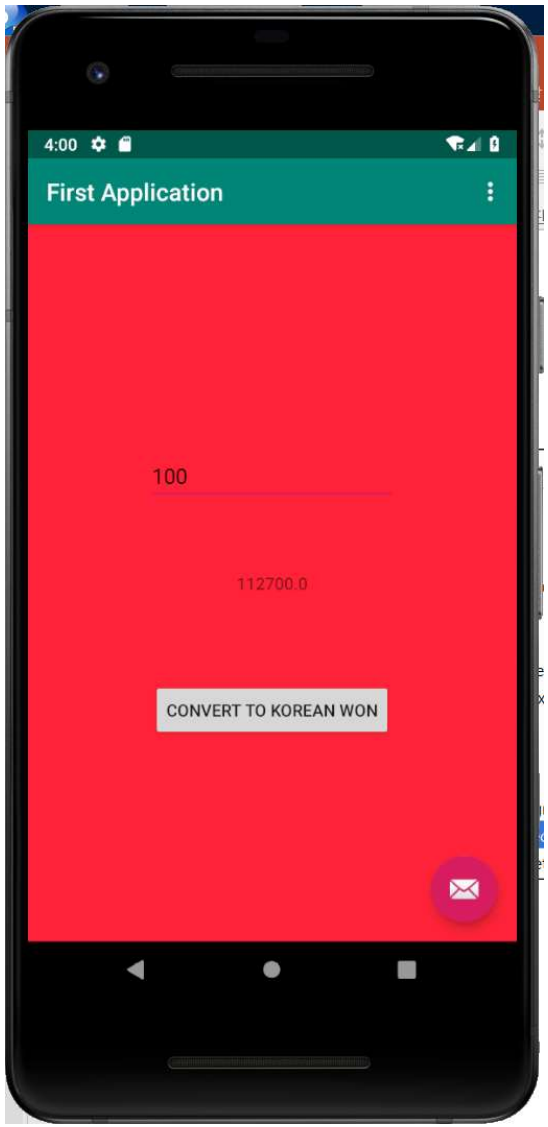
# 가상화 기술




- AVD로 **x86**을 선택했을 때는 **virtualization technology**를 **enable** 시켜야 함.
  - **BIOS**에서 설정
- **가상 머신**(Virtual Machine, **VM**)
  - CPU 처리 과정을 소프트웨어로 구현
- **x86 가상화** (x86 virtualization)
  - **CPU 가상화**
  - VM에서 실행되는 소프트웨어는 마치 전용 CPU에서 동작하는 것처럼 어떠한 성능 및 호환성 문제 없이 동작

참고 사이트 https://downloadcenter.intel.com/ko/download/7838

# App. 실행

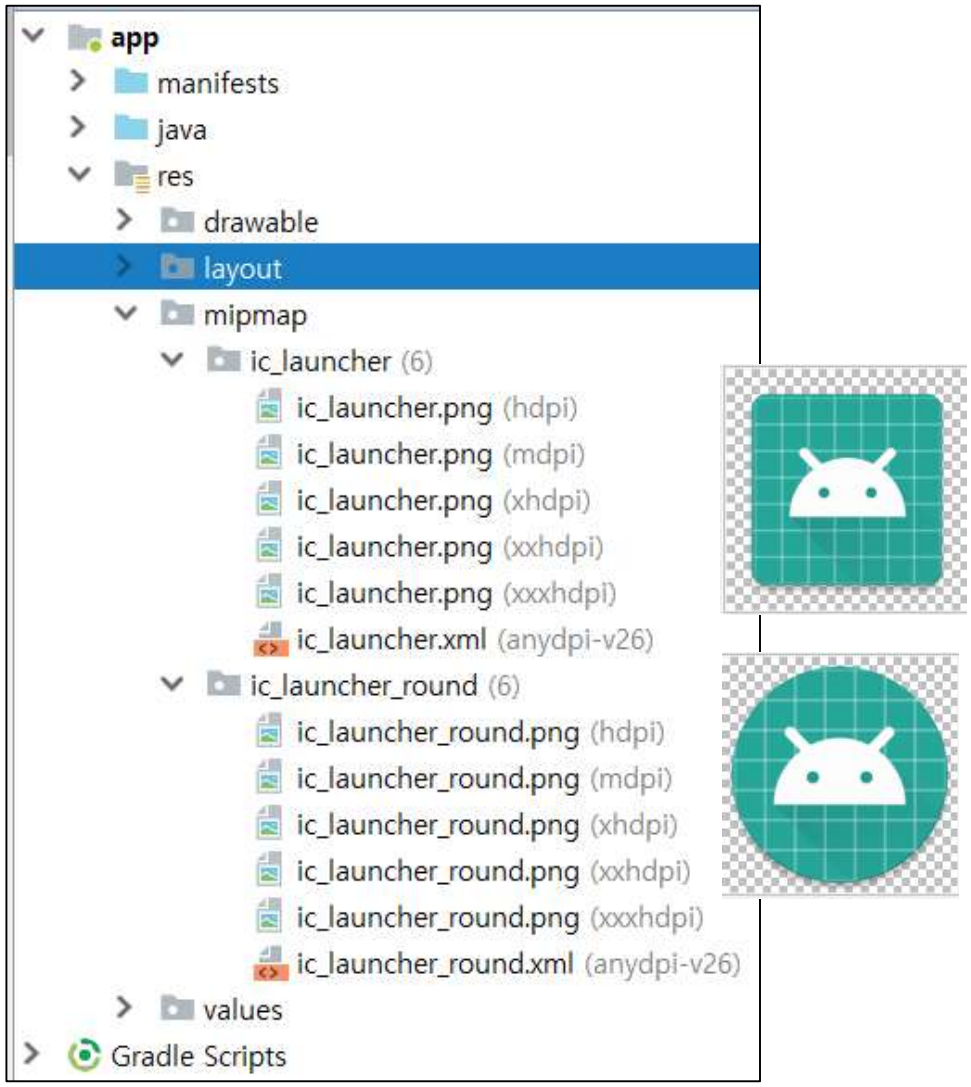



Contains Kotlin code

MainActivity.kt
+
activity_main.xml

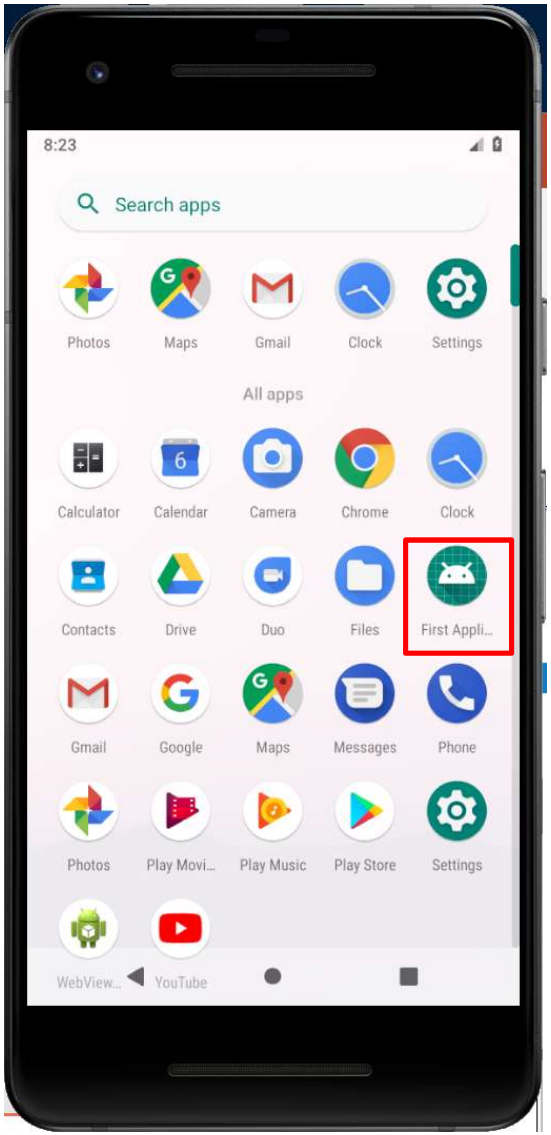
Decides how your Activity looks like

# App. icon


app
manifests
java
res
drawable
layout
mipmap
ic_launcher (6)
ic_launcher.png (hdpi)
ic_launcher.png (mdpi)
ic_launcher.png (xhdpi)
ic_launcher.png (xxhdpi)
ic_launcher.png (xxxhdpi)
ic_launcher.xml (anydpi-v26)
ic_launcher_round (6)
ic_launcher_round.png (hdpi)
ic_launcher_round.png (mdpi)
ic_launcher_round.png (xhdpi)
ic_launcher_round.png (xxhdpi)
ic_launcher_round.png (xxxhdpi)
ic_launcher_round.xml (anydpi-v26)
values
Gradle Scripts

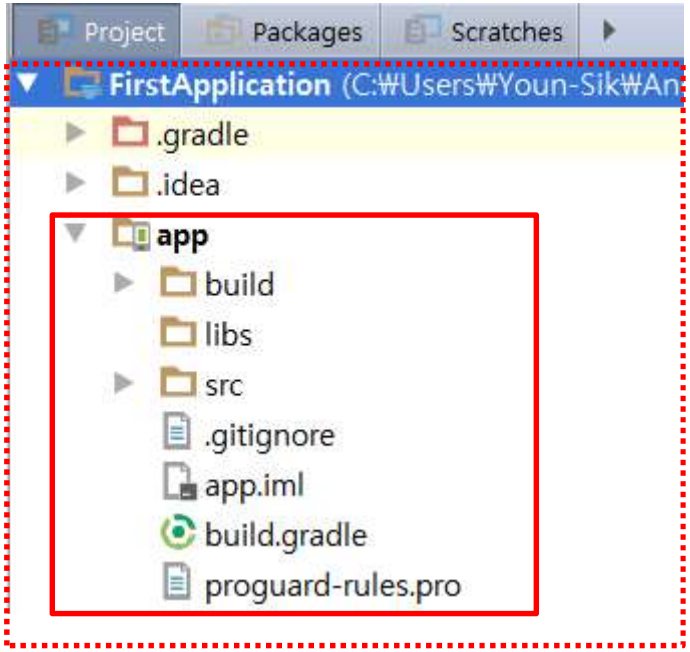




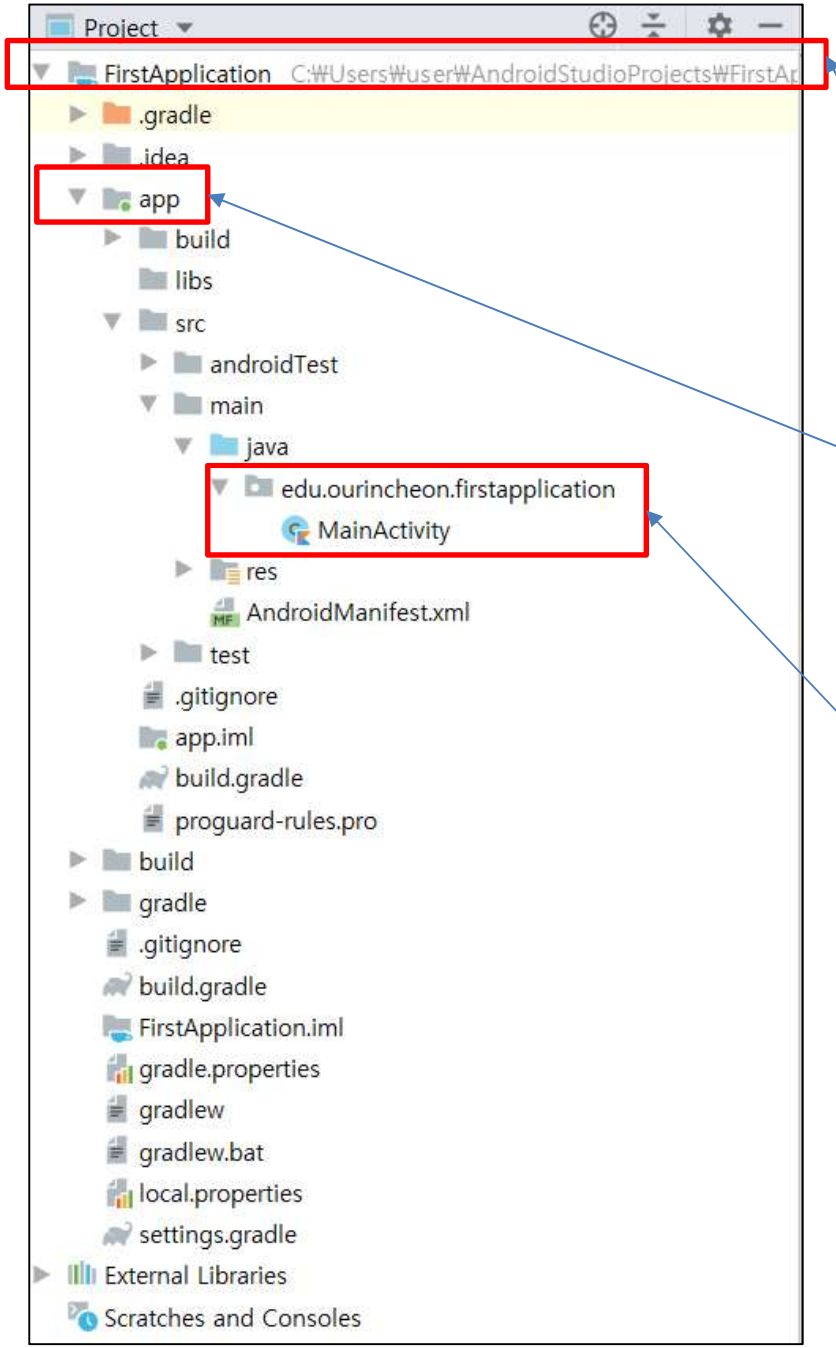

8:23
Search apps
Photos
Maps
Gmail
Clock
Settings
All apps
Calculator
Calendar
Camera
Chrome
Clock
Contacts
Drive
Duo
Files
First Appli...
Gmail
Google
Maps
Messages
Phone
Photos
Play Movi...
Play Music
Play Store
Settings
WebView...
YouTube

# project와 module

- **project(프로젝트) 와 module(모듈)은 어떻게 다른가요?**
  - project와 module은 **Android Studio** 에서 사용하는 용어
- **프로젝트 (project)**
  - project는 여러 개의 module로 이루어짐.
- **모듈 (module)**
  - 모듈 = 앱(App.)
  - module은 프로젝트에 포함됨.




File > New > **New Module** ...
을 선택해서 새 모듈을
만들어 보자!



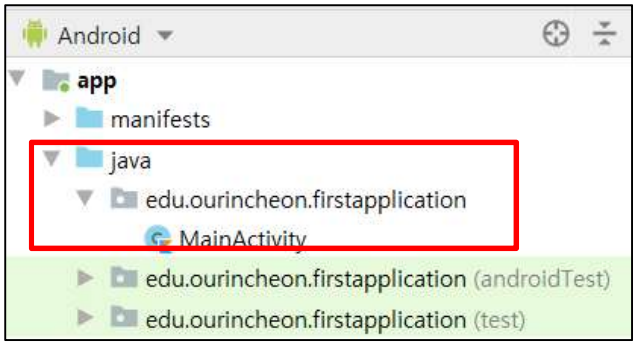

**FirstApplication**
- Parent Project Name
- Contains sub-projects and its files

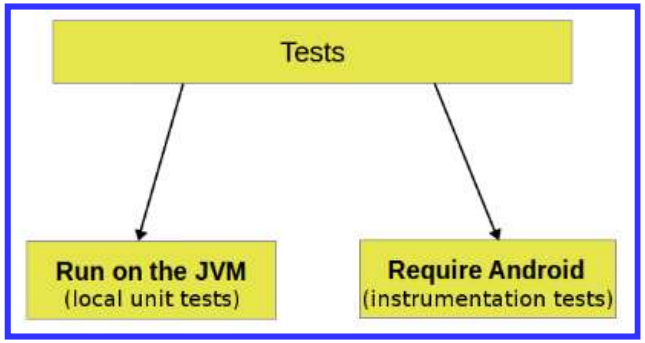
**app**
- Sub-project
- Also known as a module

**edu.ourincheon.firstapplication**
- Package name
- Contains Java or Kotlin files

# 잠깐! 이상한 폴더가 너무 많이 생겨요

- **app > java 폴더 밑에 왜 3개의 하위 폴더가 만들어지나요?**
  - (package) **androidTest**
    - for unit tests that involves android instrumentation.
    - **To test code that *use Android framework***
  - (package) **test**
    - for pure unit test that do not involve android framework.
    - **To test code that are *pure java classes***

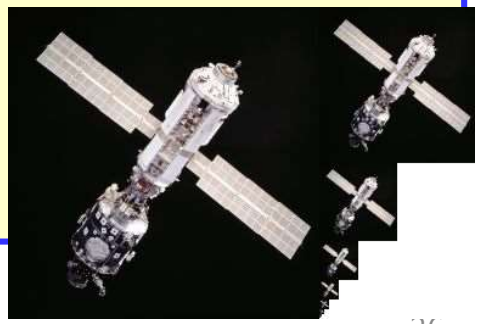


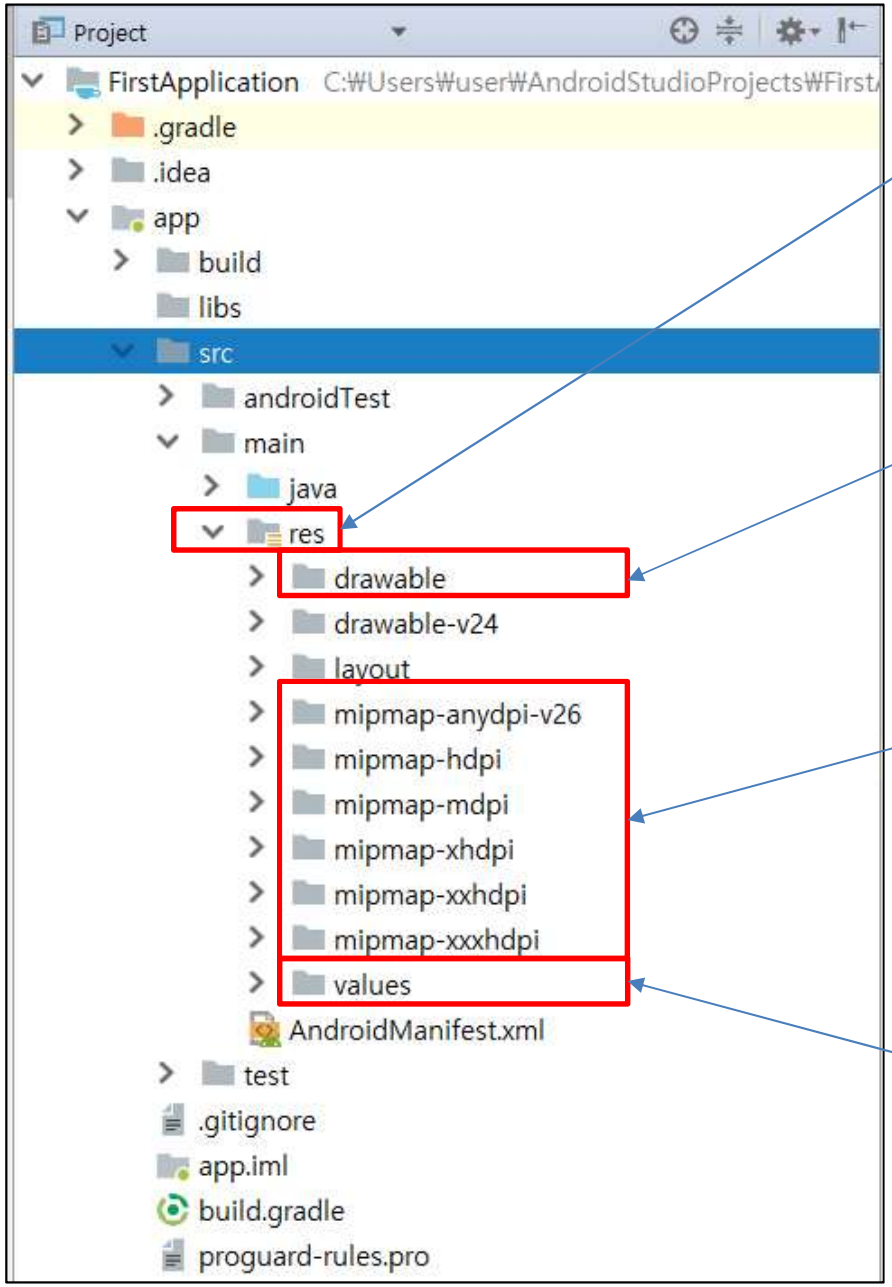




참고 사이트 http://www.vogella.com/tutorials/AndroidTesting/article.html
https://developer.android.com/studio/test/index.html

# Drawable 과 mipmap 차이

- **drawable**
  - For bitmap files (PNG, JPEG, or GIF), 9-Patch image files, and XML files that describe Drawable shapes or Drawable objects that contain multiple states (*normal*, *pressed*, or *focused*).
- **mipmap**
  - For **app launcher icons**
  - Android 4.2(API 17)부터 도입
  - The Android system retains the resources in this folder (and *density-specific folders* such as mipmap-xxxhdpi) regardless of the screen resolution of the device where your app is installed.
  - This behavior allows launcher apps *to pick the best resolution icon* for your app to display on the home screen.
  - **mip-map**이란 무슨 뜻인가요?
    - 원본 이미지 축소판의 집합



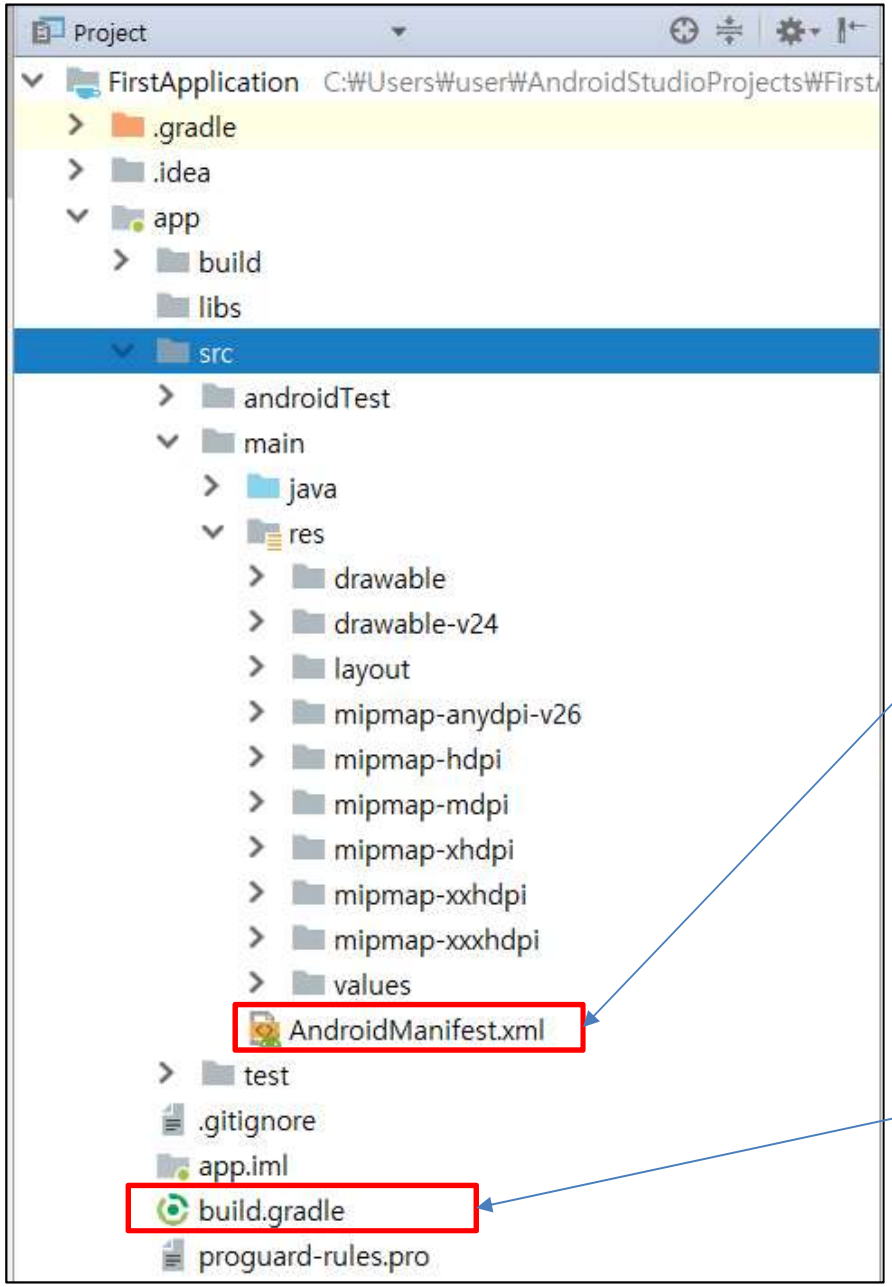



**res**
- Contains all UI resources
- Layouts, images, audio files, etc.

**res/drawable**
- Image assets
- Vector assets

**res/mipmap**
- App launcher icons

**res/values**
- App styles and themes
- Color details
- Localized strings (texts used in app UI)




**AndroidManifest.xml**

- contains application components details
- Declaration of Activity, Service, BroadcastReceiver and ContentProvider
- Define necessary permission
  - USES INTERNET, USES CAMERA
  - READ SD CARD, etc.
- It is like summary of the application

**build.gradle**

- Build configuration
- Plugins to be used
- External libraries or dependencies to be included

# AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
          package="edu.ourincheon.firstapplication">

    <application
            android:allowBackup="true"
            android:icon="@mipmap/ic_launcher"
            android:label="First Application"
            android:roundIcon="@mipmap/ic_launcher_round"
            android:supportsRtl="true"
            android:theme="@style/AppTheme">
        <activity
                android:name=".MainActivity"
                android:label="First Application"
                android:theme="@style/AppTheme.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>

                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>

</manifest>
```

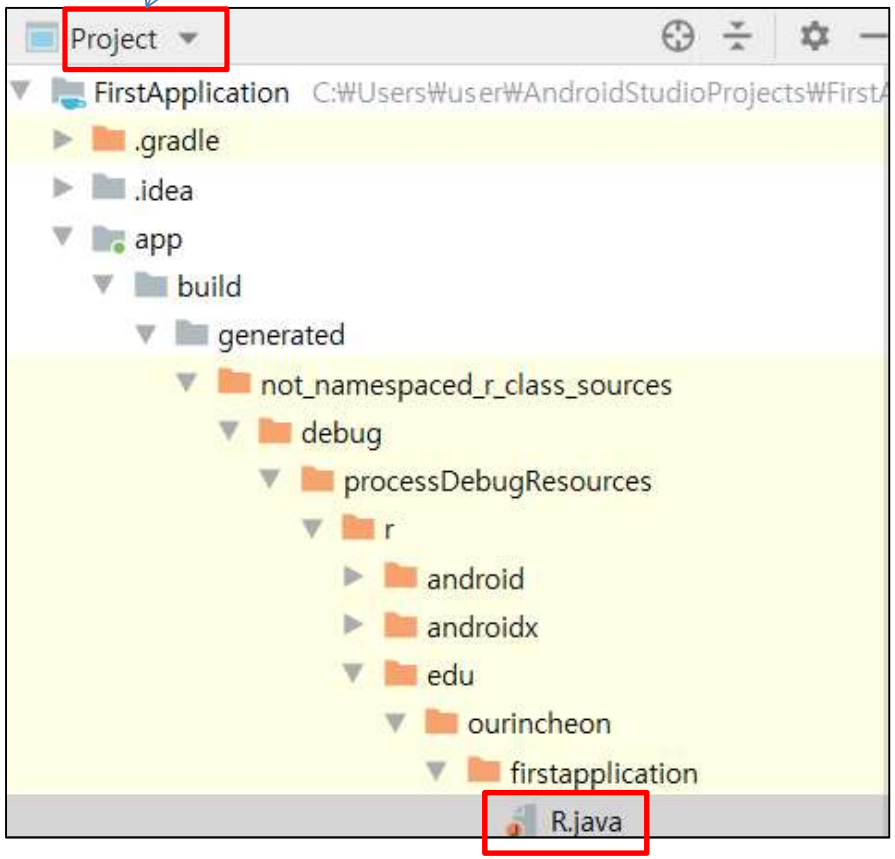
# Resource 참조 방법

- **@리소스타입/리소스이름**
  - XML 문서에서 리소스를 참조할 때
    - **strings**.xml ➔ @**string**/xxx
    - **dimens**.xml ➔ @**dimen**/xxx

- **R.리소스타입.리소스이름**
  - Java 코드에서 리소스를 참조할 때
    - **리소스_id** 는 **R.java**에 정수 상수로 선언되어 있음.
      - R.layout.**activity_main** ➔ 리소스_id
      - R.string.**new_string** ➔ 리소스_id

Android view 를
Project view로 전환

# Where is R.java?




```
public final class R {
  public static final class anim {...}
  public static final class animator {...}
  public static final class attr {...}
  public static final class bool {...}
  public static final class color {...}
  public static final class dimen {...}
  public static final class drawable {...}
  public static final class id {...}
  public static final class integer {...}
  public static final class interpolator {...}
  public static final class layout {...}
  public static final class menu {...}
  public static final class mipmap {...}
  public static final class string {...}
  public static final class style {...}
  public static final class styleable {...}
}
```

```
public static final class id {
    public static final int dimensions=0x7f08003c;
    public static final int direct=0x7f08003d;
    public static final int disableHome=0x7f08003e;
    public static final int dollarText=0x7f08003f;
    public static final int edit_query=0x7f080040;
    public static final int end=0x7f080041;
```

```
public static final class string {
    public static final int convert_string=0x7f0e002d;
    public static final int dollars_hint=0x7f0e002e;
```

# 새 resource를 XML 문서에 직접 추가

res/values/strings.xml

새 문자열(string) 리소스를 추가하려면
리소스 이름(name)과 리소스 값(value)을 정의
<string **name**="**리소스 이름**">**리소스 값**</string>

예:
<string **name**="**new_string**">**새 문자열**</string>

```
<resources>
    <string name="app_name">First Application</string>
    <string name="action_settings">Settings</string>
    <string name="convert_string">Convert to Korean Won</string>
    <string name="dollars_hint">dollar</string>
    <string name="hello_msg">안녕, 안드로이드</string>
    <string name="new_string">새 문자열</string>
</resources>
```

**지금 추가한 문자열 리소스 id를 R.java에서 찾을 수 없는데요?**
**Build > Rebuild Project 를 하거나**
**Run > Run app 을 하면 R.java에 리소스 id가 추가됩니다.**

# TextView를 소스 코드에서 참조


**MainActivity.kt**

```
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        setSupportActionBar(toolbar)

        textView.text = "내용이 바뀌었지!"
```

# String 리소스에 정의한 문자열 출력

**MainActivity.kt**

```
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        setSupportActionBar(toolbar)

        val str = getString(R.string.hello_msg)
        textView.text = str
```

# 잠깐! 코드 순서를 바꿔도 되나요?

```kotlin
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)

    val str = getString(R.string.hello_msg)
    textView.text = str


}
```

```kotlin
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)


    val str = getString(R.string.hello_msg)
    textView.text = str

    setContentView(R.layout.activity_main)
}
```

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

    TextView tv = (TextView)findViewById(R.id.textView);
    tv.setText("내용이 바뀌었지?");
}
```

```java
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    TextView tv = (TextView)findViewById(R.id.textView);
    tv.setText("내용이 바뀌었지?");

    setContentView(R.layout.activity_main);
```

# Android project Build 과정