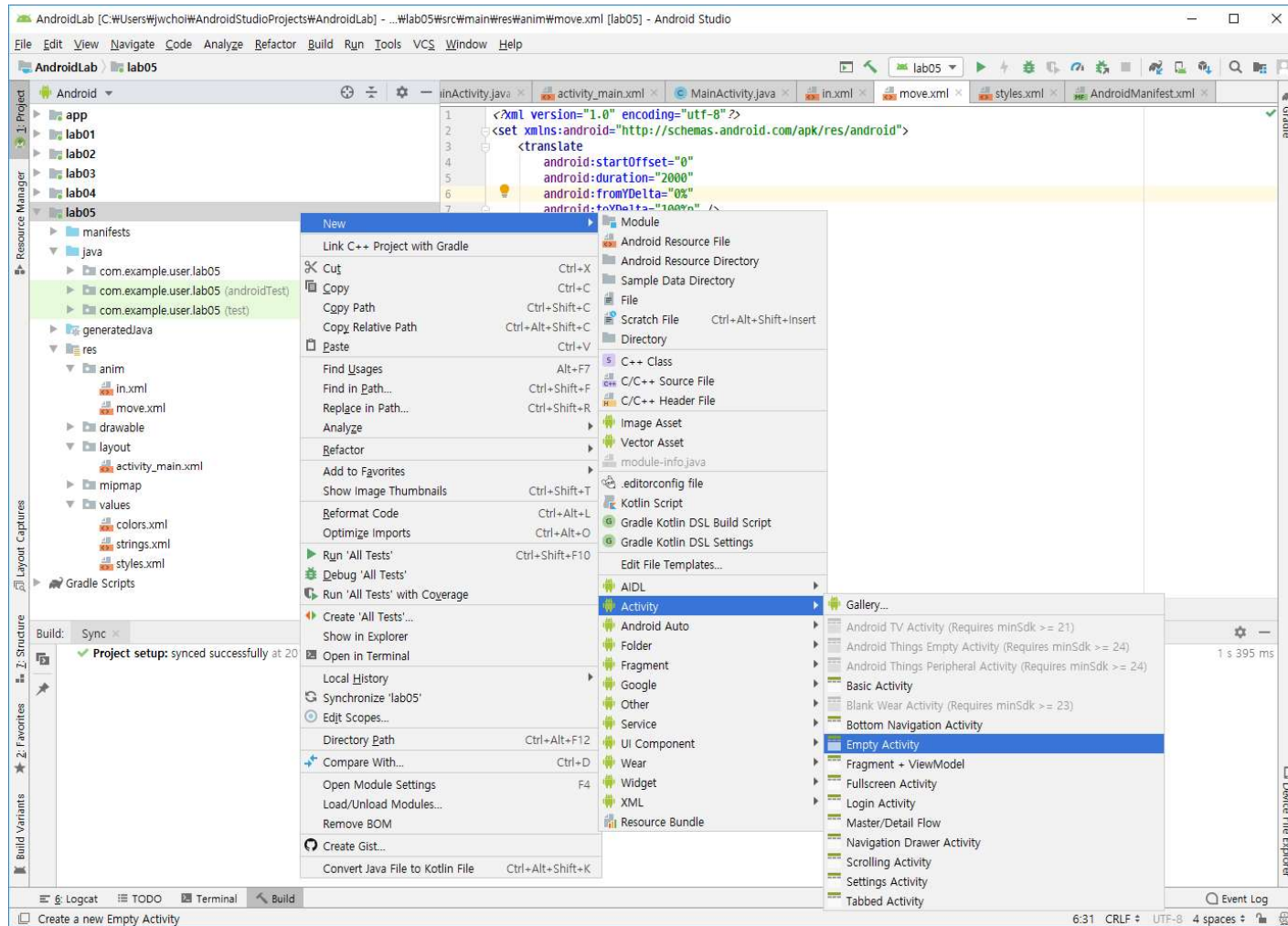




리소스 폴더명 조건 활용하기

Step 1 _ 액티비티 생성




New Android Activity

 **Configure Activity**
Android Studio



Creates a new empty activity



Activity Name:

Lab05_2Activity

☒ Generate Layout File

Layout Name:

activity_lab05_2

☒ Launcher Activity

Package name:

com.example.user.lab05

Source Language:

Java

Target Source Set:

main

If true, this activity will have a CATEGORY_LAUNCHER intent filter, making it visible in the launcher

Previous

Next

Cancel

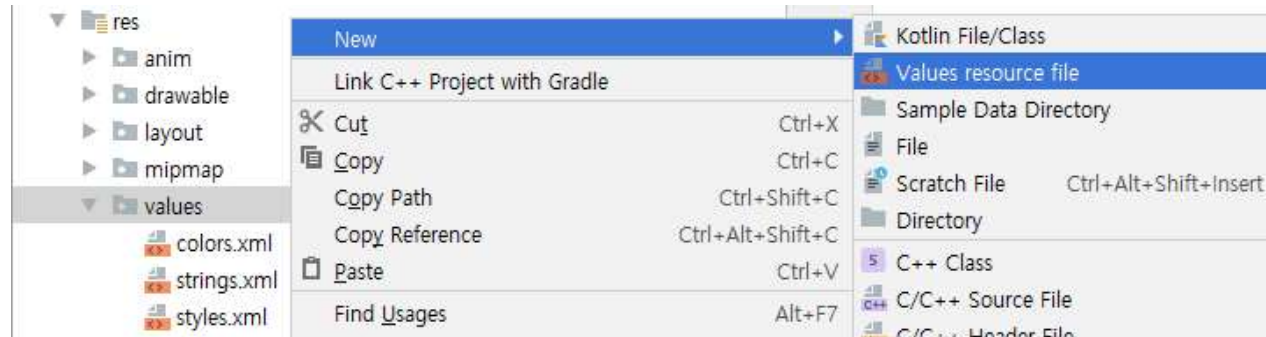
Finish

Step 2 _ 문자열 리소스 등록

- res/values/strings.xml

```
<resources>  
    <string name="app_name">Lab05</string>  
    <string name="note">To be, or not to be: that is the question</string>  
</resources>
```

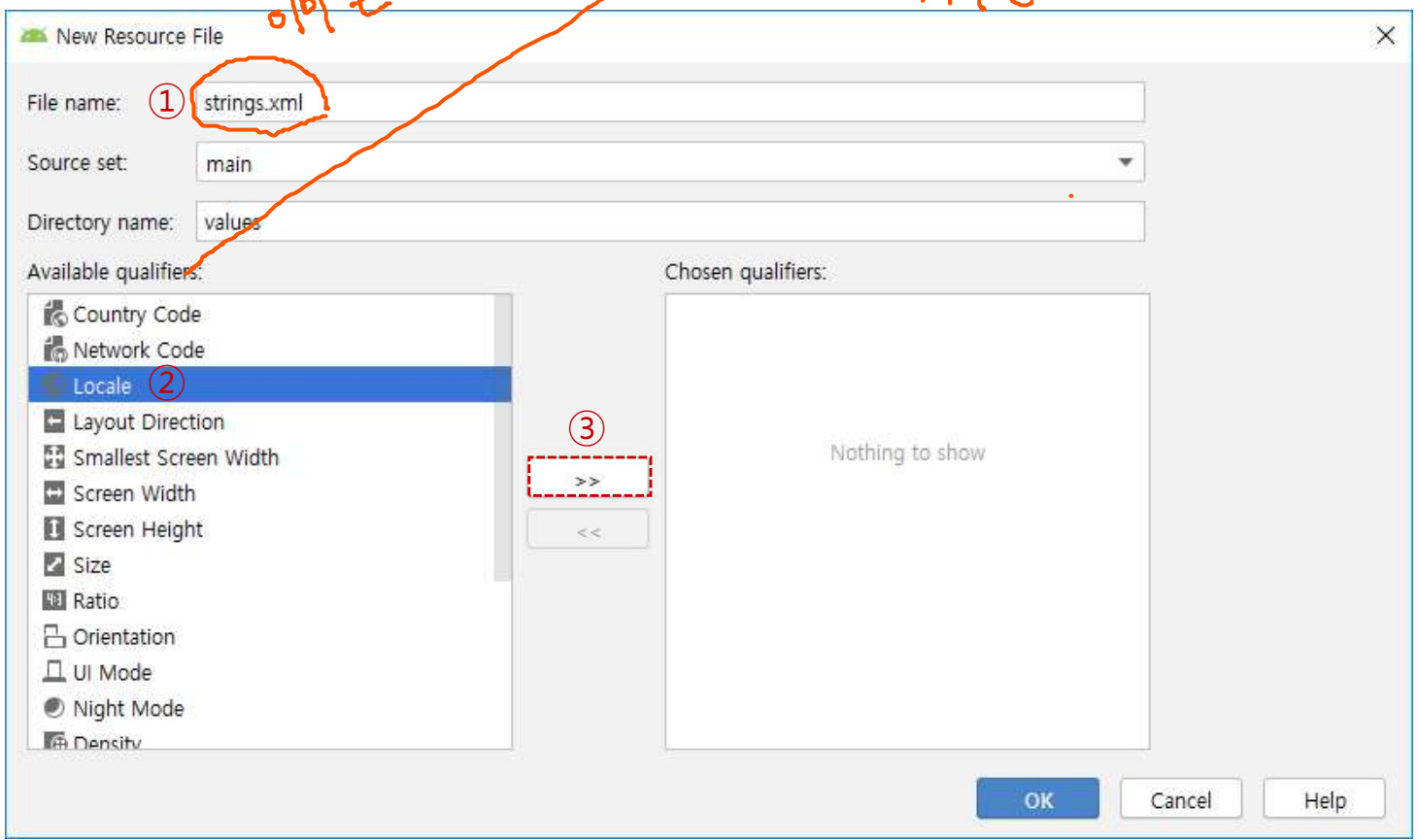
Step 3 _ 한국어 문자열 리소스 파일 생성



drawable - V24

이미 존재

기기설정상태에서 콜라풍.



New Resource File

File name: strings.xml

Source set: main

Directory name: values-ko-rKR

Available qualifiers:

Country Code

Network Code

Layout Direction

Smallest Screen Wi

Screen Width

Screen Height

Size

Ratio

Orientation

UI Mode

Night Mode

Density

Touch Screen

>>

<<

Chosen qualifiers:

ko, KR

Language:

kk: Kazakh

kl: Kalaallisut

km: Khmer

kn: Kannada

ko: Korean

kr: Kanuri

ks: Kashmiri

ku: Kurdish

kv: Komi

kw: Cornish

ky: Kyrgyz

la: Latin

Tip: Type in list to filter

Specific Region Only:

Any Region

KR: South Korea

KP: North Korea

Show All Regions

OK

Cancel

Help

Step 4 _ 한글 문자열 리소스 등록

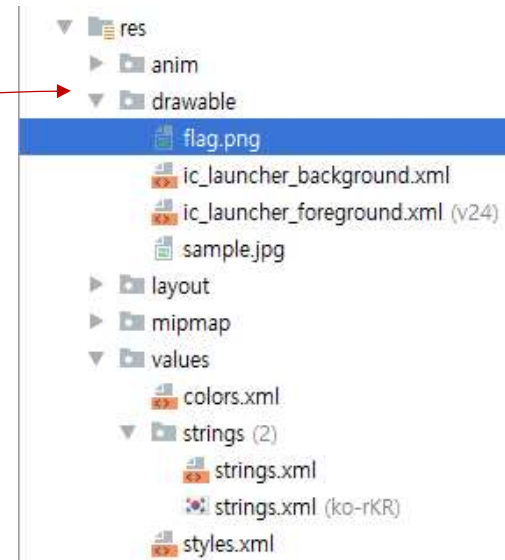
- res/values-ko-^{region}rKR/strings.xml

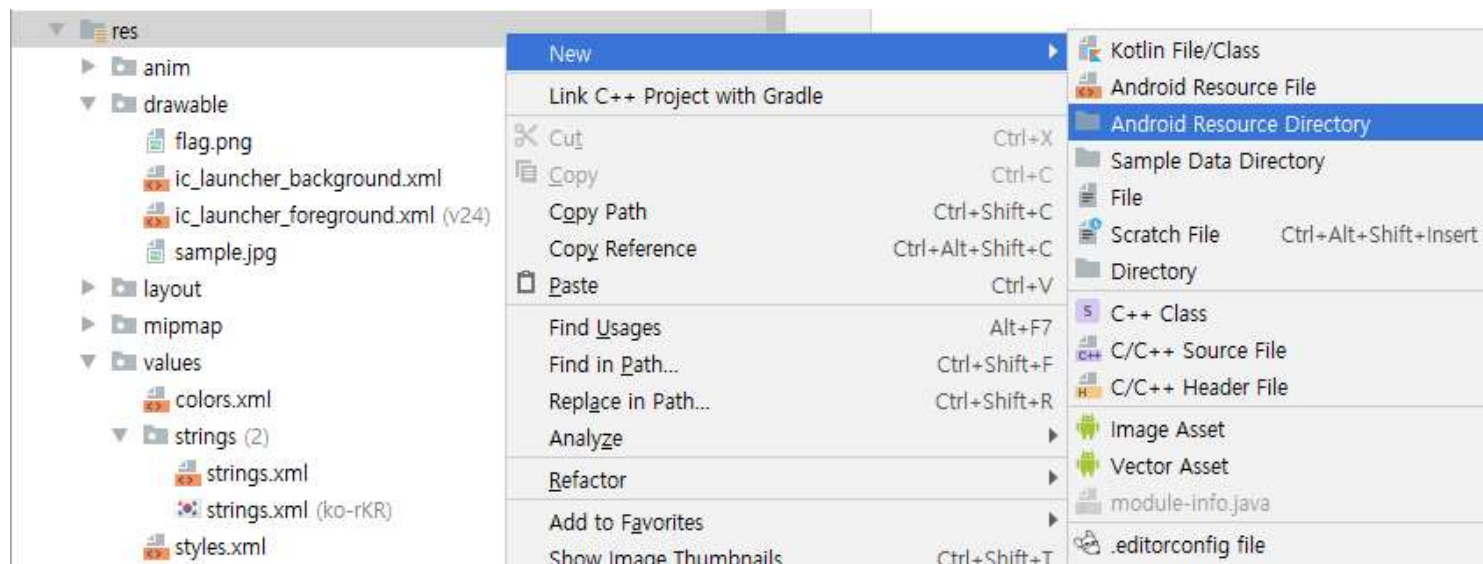
```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Lab05</string>
    <string name="note">사느냐 죽느냐 이것이 문제로다</string>
</resources>
```

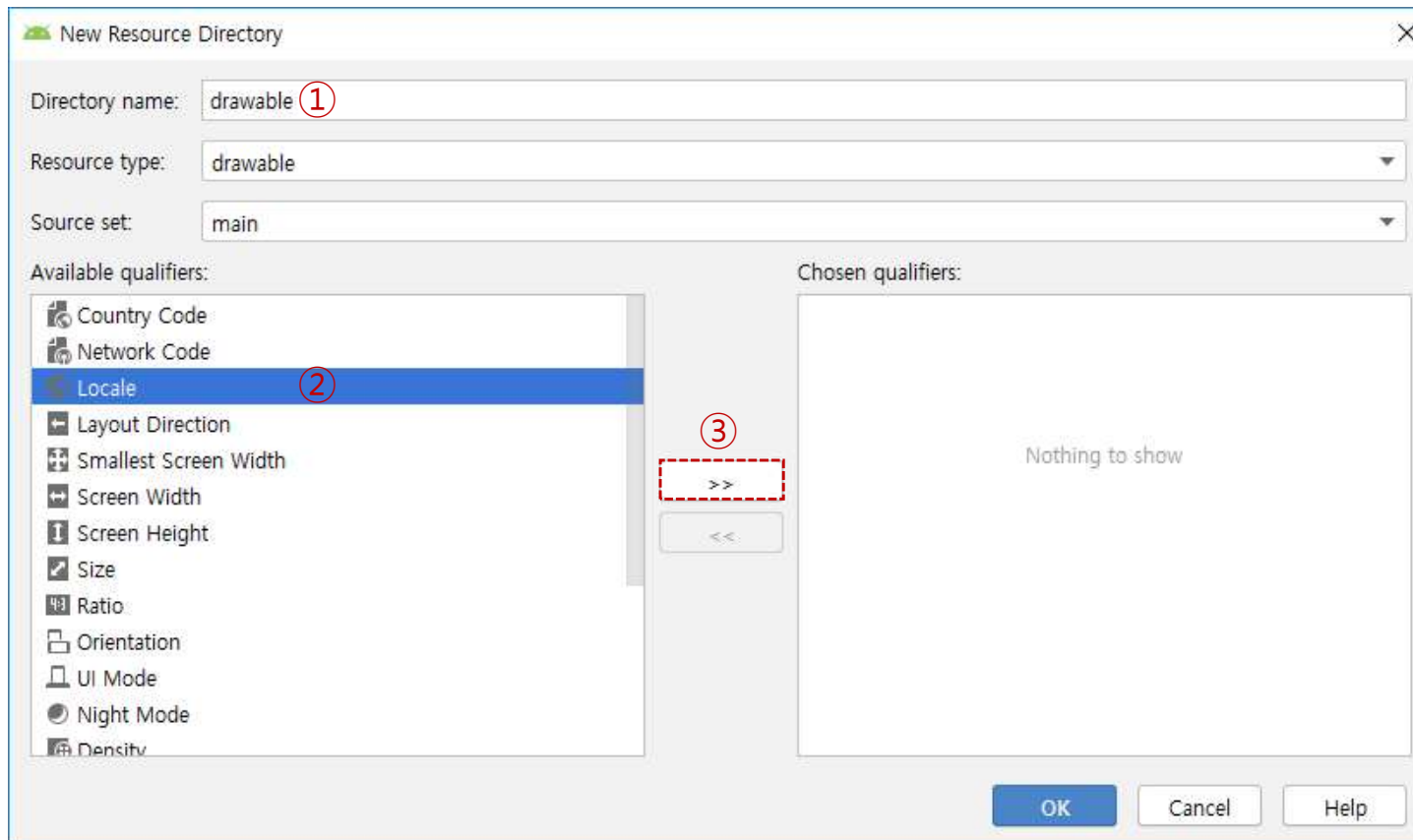

Step 5 _ 이미지 리소스 복사

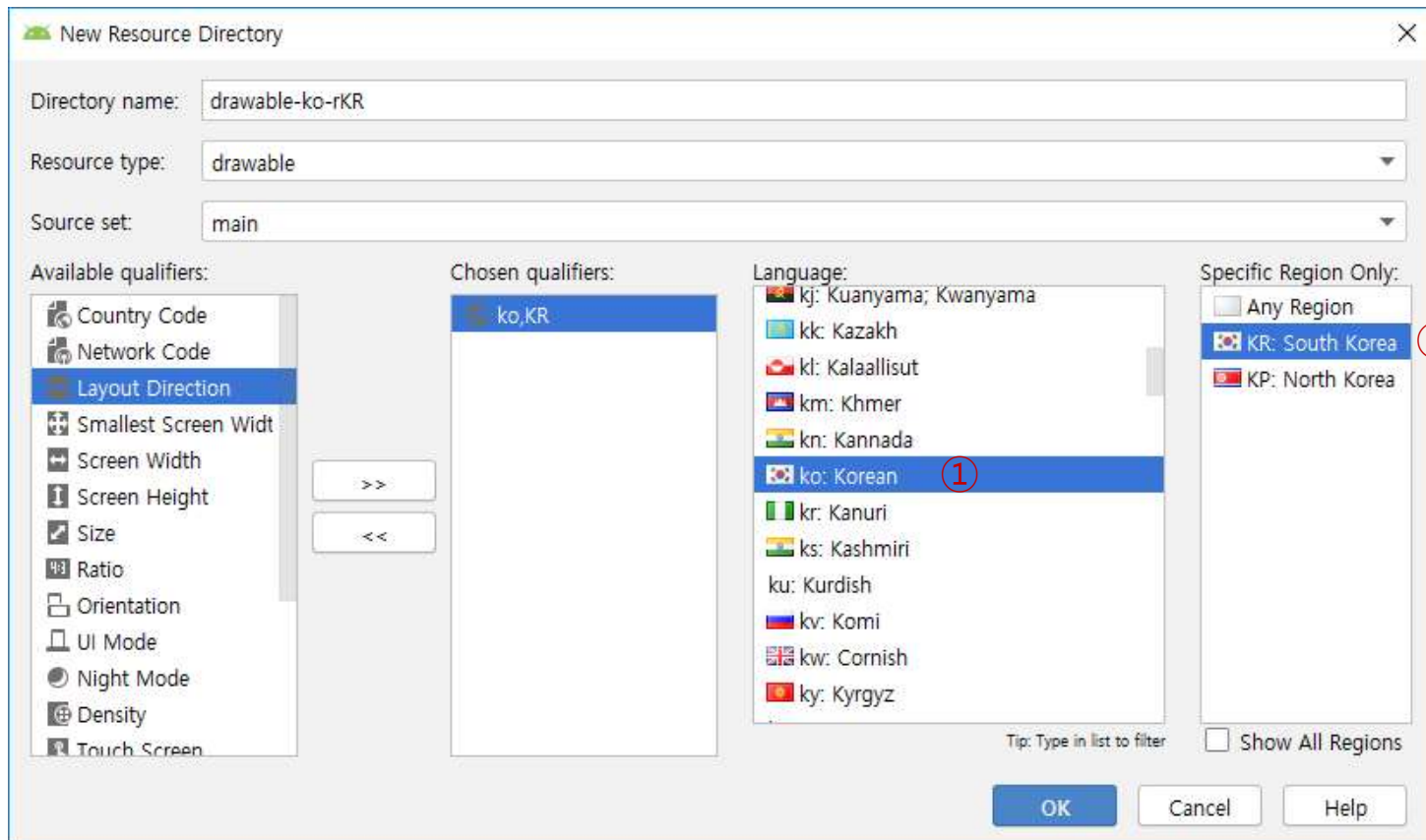


flag.png



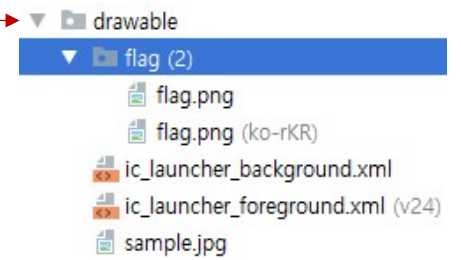
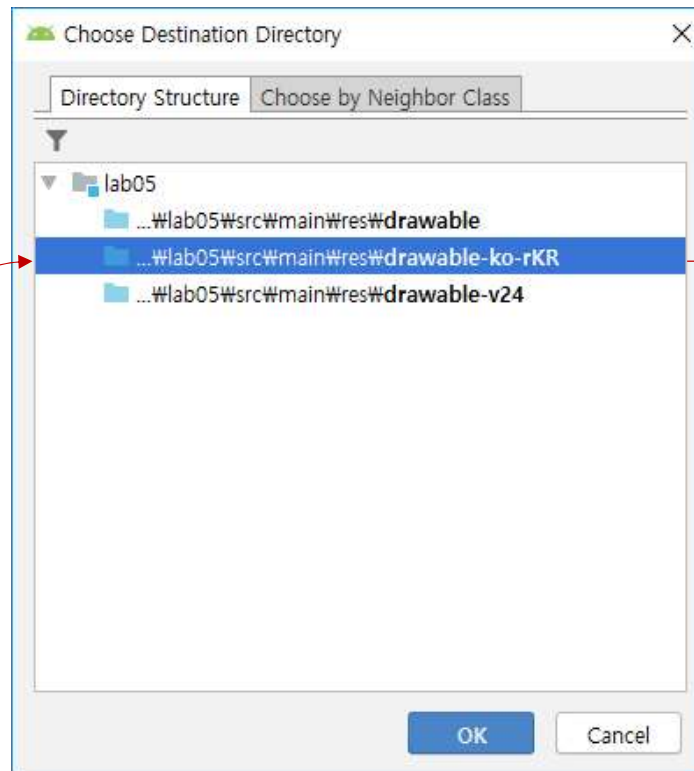








flag.png



Step 6 _ activity_lab05_2.xml

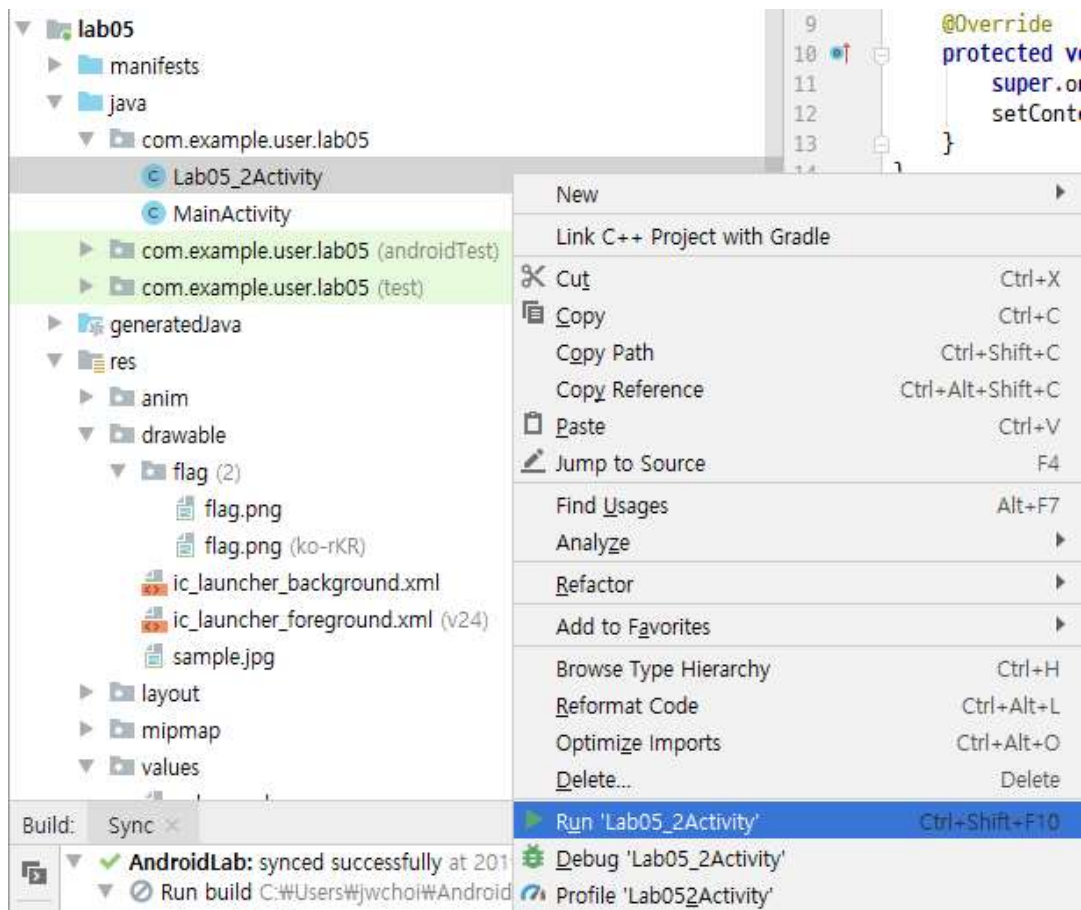
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center">

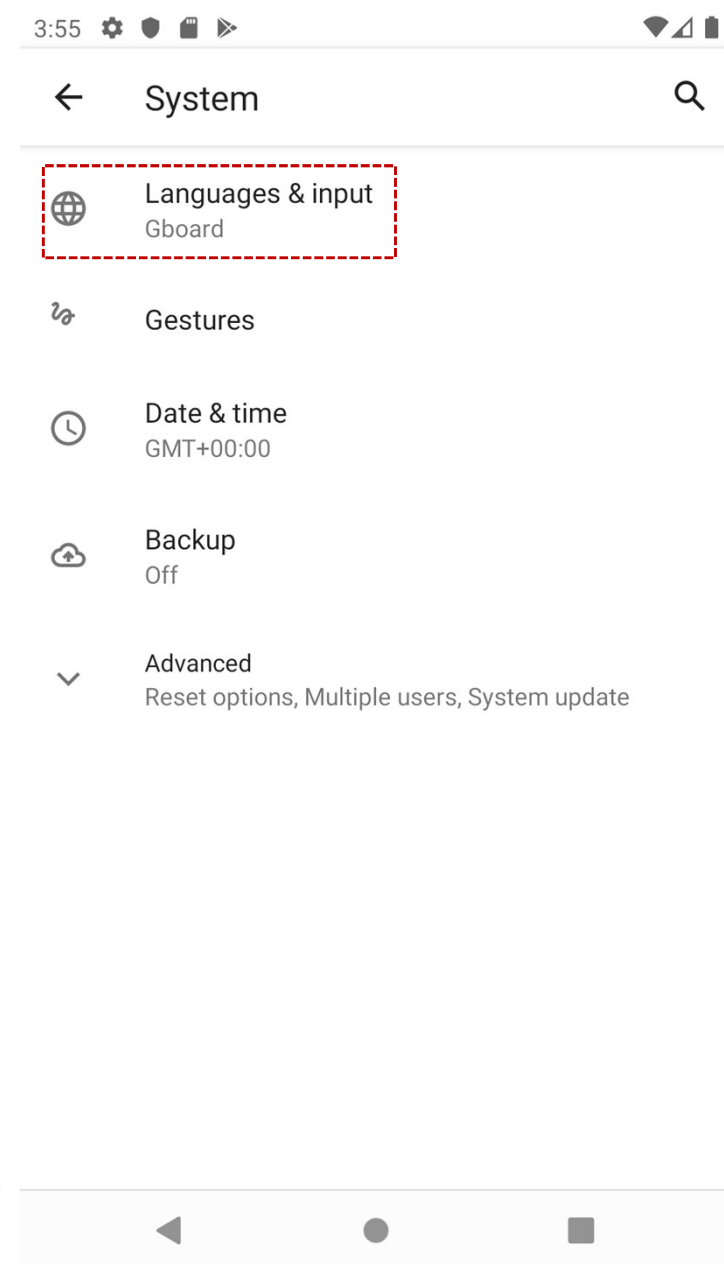
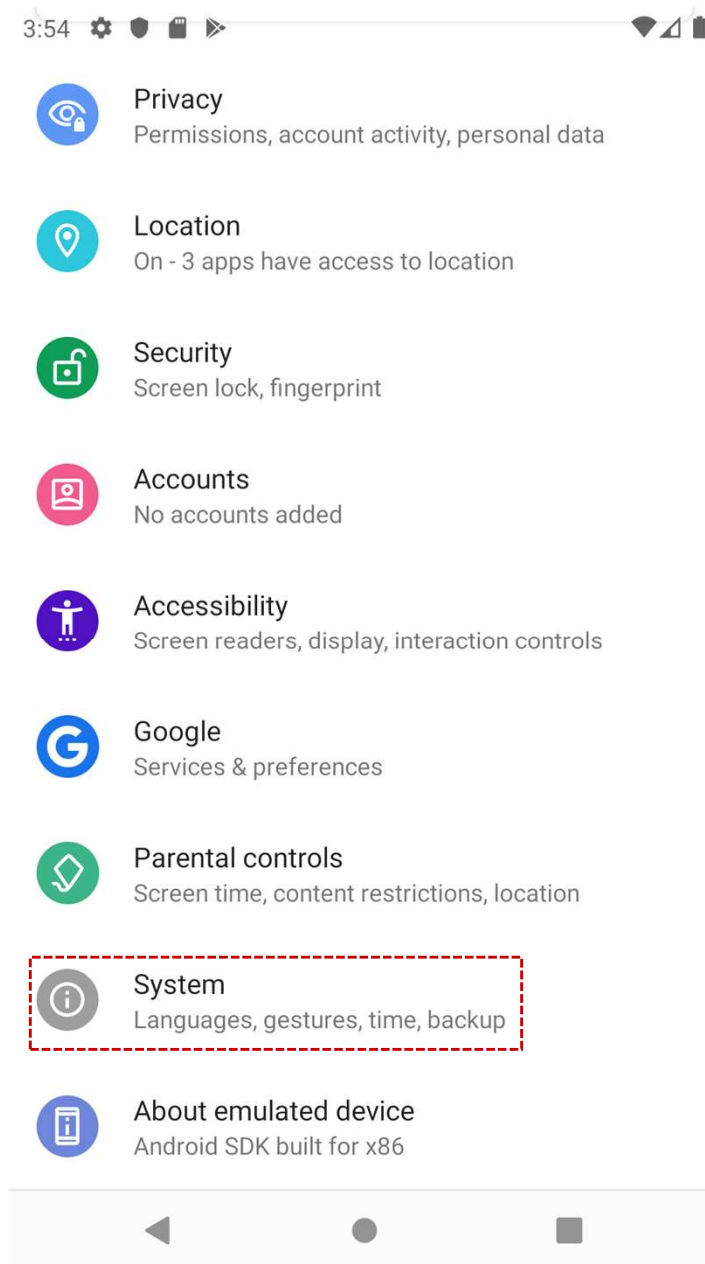
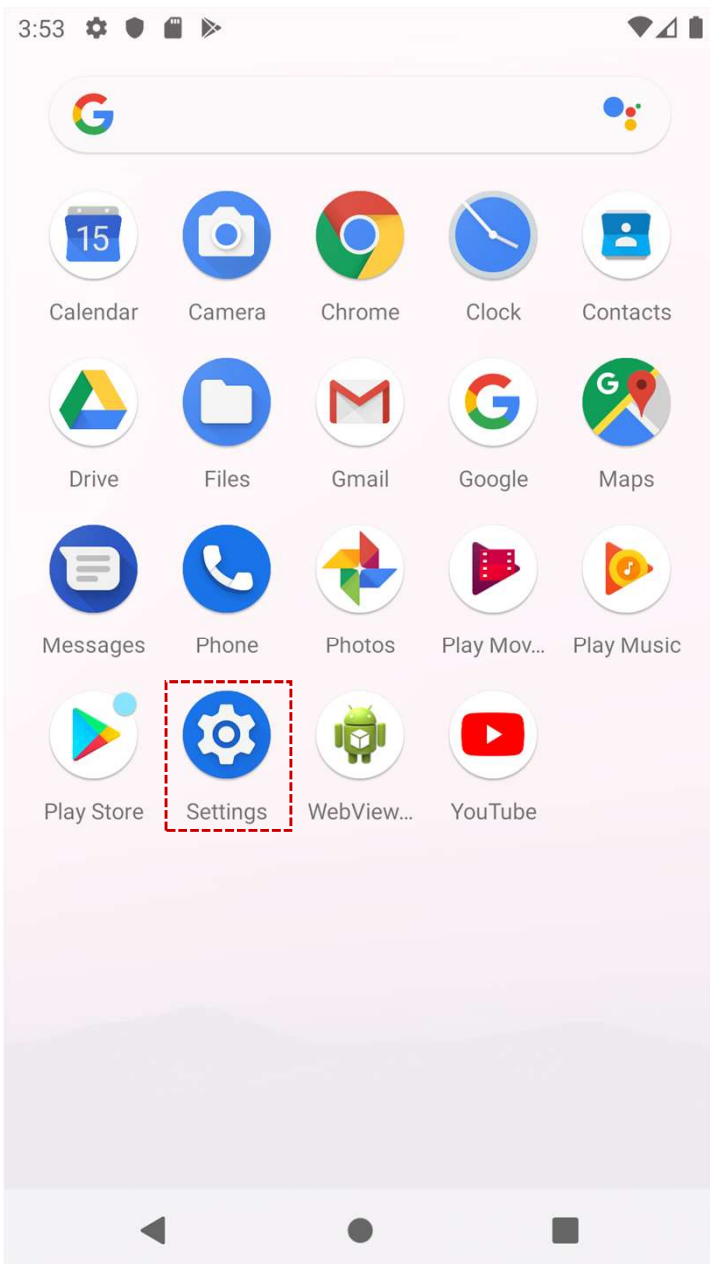
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/flag" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/note" />

</LinearLayout>
```

Step 7 _ Lab05_2Activity.java 실행





Languages & input



Languages
English (United States)

KEYBOARDS

Virtual keyboard

Gboard, Google voice typing, and Google Korean Input

Physical keyboard

Not connected



Advanced

Tools, Pointer speed, Text-to-speech output

Languages

1 English (United States)



Add a language

Add a language

GWY

한국어

日本語

日本語

简体中文

粤语 (简体)

粵語 (繁體)

繁體中文

4:00



4:01



4:02



← 한국어

대한민국

조선민주주의인민공화국

← Languages

1 English (United States)

2 한국어(대한민국)

+ Add a language

🔍 ⋮

← 언어

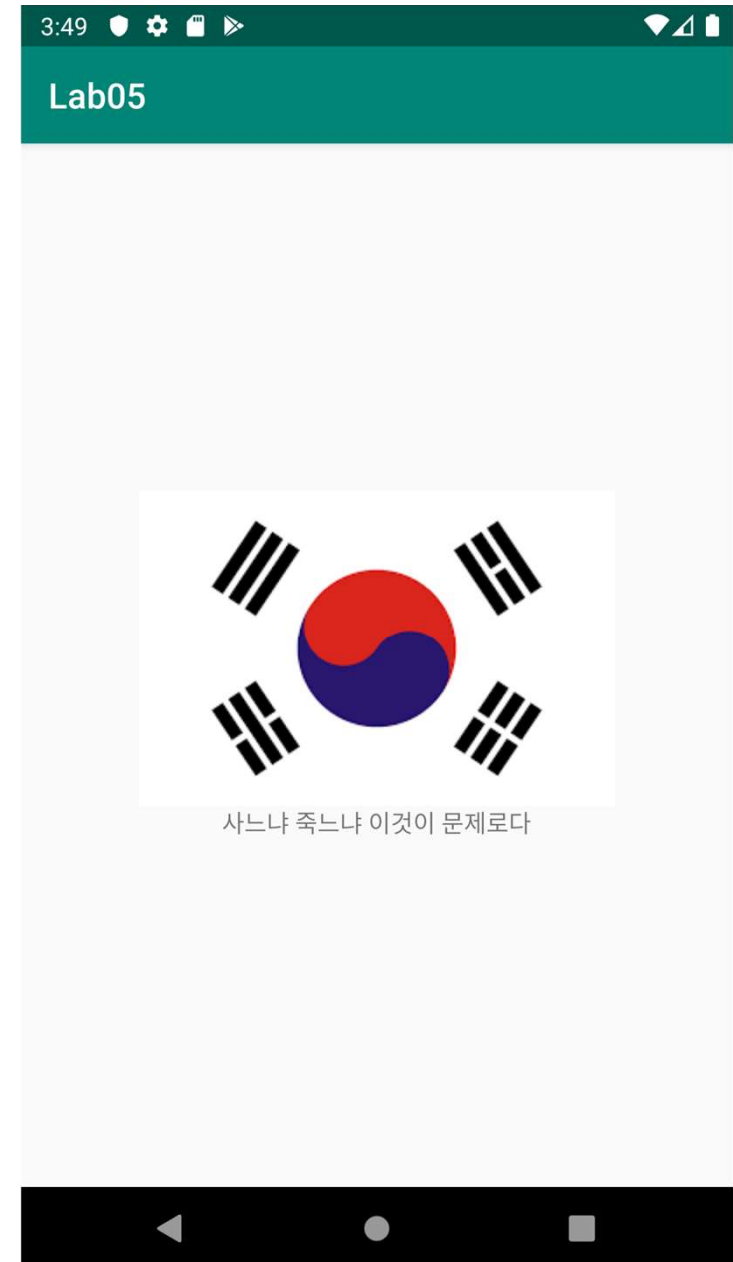
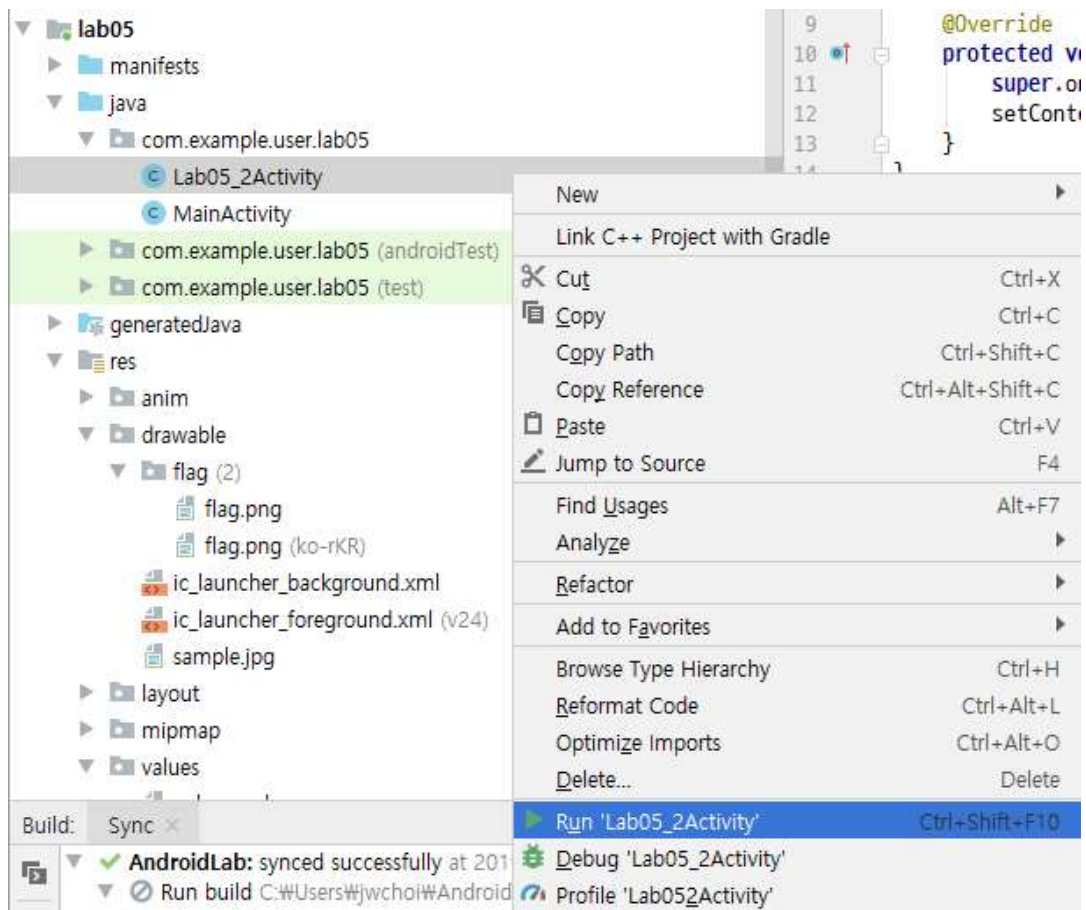
🔍 ⋮

1 한국어(대한민국)

2 English (United States)

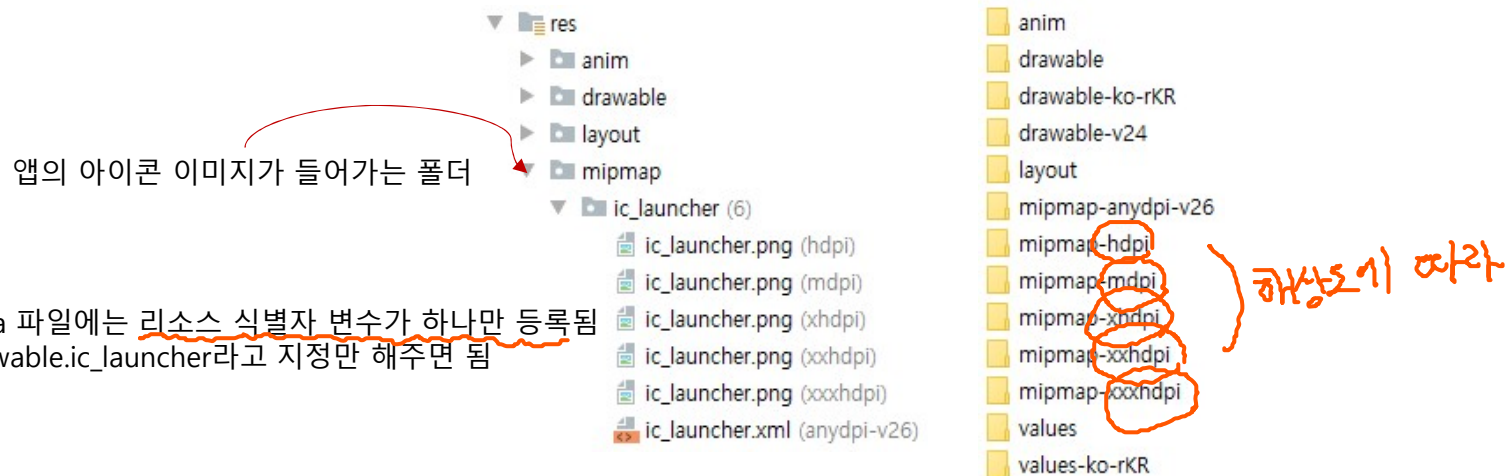
+ 언어 추가





리소스 폴더명 조건 명시법

- 리소스 폴더명에 대시(-)를 이용한 기기의 조건을 주고 이에 따른 리소스 적용을 시스템에서 알아서 적용해주는 구조



이름이 같아서 R.java 파일에는 리소스 식별자 변수가 하나만 등록됨
∴ 코드에서는 R.drawable.ic_launcher라고 지정만 해주면 됨

폴더명에 대시를 이용해 명시할 때 지켜야 할 규칙

- 대시로 구분하여 하나의 폴더에 여러 조건을 명시할 수 있음
예) drawable-en-rUS-land
 - 여러 조건을 모두 만족할 때 해당 폴더의 리소스가 적용 (즉, And 연산)
- 조건이 나열되는 순서가 있음
예) drawable-hdpi-port (O), drawable-port-hdpi (X)
- 폴더에 서브 폴더로 조건을 세분화하는 건 불가능
예) res/drawable/drawable-en (X)
- 조건에 대한 대소문자는 구분하지 않음
 - 대문자는 가독성을 위한 것일 뿐
- 하나의 폴더에는 각 조건마다 하나의 값만 명시할 수 있음
예) drawable-rES-rFR (X) → drawable-rES, drawable-rFR
 - 즉, 2개의 폴더를 만들어야 함

논리적 단위로 스마트폰 크기 호환성 확보

가기에 해상도가 다르더라도 크기가 비슷하게 나오게 한다.
view의

개념은 같다

- dp(dip)
 - Density-Independent Pixels.
 - 스크린의 물리적 밀도에 기초한 단위
 - 160dpi(dots per inch)에 상대적 수치
- sp(sip)
 - Scale-independent Pixels.
 - dp와 유사하며 폰트 크기에 적용
- pt
 - Points.
 - 화면 크기의 1/72를 1pt
- px
 - 픽셀
- mm
 - 밀리미터
- in
 - 인치

권장

기기 설정에서 글자 크기를 변경 가능 (dp는 불가)

"Baseline" density
1dp=1px

Device의 특성

LOGICAL DENSITY (DPI) - 1인치 안에 몇 픽셀	FRIENDLY NAME	SCALE <small>px = dp * (dpi / 160)</small>
~120	LDPI	0.75x
~160	MDPI	1x
~240	HDPI	1.5x
~320	XHDPI	2x
~480	XXHDPI	3x
~640	XXXHDPI	4x

자바 코드에서 크기가 명시되는 경우

- px 단위로만 지정할 수 있음

16sp가 변환된 값이 들어감. (px)

$$px = dp * (dpi / 160)$$

Device마다 고유한 특성

MDPI
Baseline

```
Resources res = getResources();  
float fontSize = res.getDimension(R.dimen.font_size);
```

```
// Get the screen's density scale  
final float scale = getResources().getDisplayMetrics().density;
```

res/values/dimens.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
  <dimen name="textview_height">25dp</dimen>  
  <dimen name="textview_width">150dp</dimen>  
  <dimen name="ball_radius">30dp</dimen>  
  <dimen name="font_size">16sp</dimen>  
</resources>
```

예제는 dp 가능

"java 코드에서는 px로만 가능"

$dpi / 160$ 값 사용.

<https://developer.android.com/guide/topics/resources/more-resources#Dimension>