

CS361

Mobile Application Development การพัฒนาโปรแกรมประยุกต์สำหรับอุปกรณ์พกพา



Pakorn Leesutthipornchai, Ph.D.

Assistant Professor

ผศ.ดร.ปกรณ์ ลิ้มสุทธิพรชัย

pakornl@tu.ac.th



MA05: Mobile and Application Resources
ทรัพยากรของโทรศัพท์และโปรแกรมประยุกต์

ทรัพยากรของโทรศัพท์

- หน้าจอ Touch Screen
 - การแตะหน้าจอทำให้เกิด event → Click, KeyPress, MotionEvent
- Sensors: SensorManager
 - Location: LocationManager
 - Location: Latitude, Longitude เช่น 14.0771, 100.5935
 - Motion sensors
 - Accelerometer ความเร่งเชิงเส้นตามแกน x, y, z มีหน่วยเป็น m/s^2
 - Gyroscope ความเร็วเชิงมุมตามแกน x, y, z มีหน่วยเป็น rad/s
 - Gravity ความเร่ง
 - Position sensors
 - Magnetic Field: x, y, z มีหน่วยเป็น Micro Tesla
 - Orientation ทิศทาง x, y, z มีหน่วยเป็น $^{\circ}$
 - Proximity วัดความใกล้: มีหน่วยเป็น cm เช่น 8.0 cm
 - Environmental sensors
 - Light: มีหน่วยเป็น lux เช่น 40.0 lux
 - Atmospheric Pressure: มีหน่วยเป็น mb, mmHg, atm, Pascal
 - Relative Humidity ความชื้นสัมพัทธ์
 - Ambient Temperature อุณหภูมิภายนอก
- Battery Status: Temperature, Level, Voltage
- Sound Level: มีหน่วยเป็น dB เช่น 40.233 dB

ทรัพยากรของแอนดรอยด์

■ Application Framework and Libraries

- ☐ Activity Manager
- ☐ Resource Manager
- ☐ Location Manager
- ☐ Sensor Manager
- ☐ Write/Read File
- ☐ Media Framework
- ☐ SQLite
- ☐ OpenGL

▶ อ่านไฟล์ภาพ

- JPEG (.jpg)
- GIF (.gif)
- PNG (.png)
- BMP (.bmp)
- WebP (.webp)

▶ อ่านไฟล์เสียง

- 3GPP (.3gp)
- MPEG-4 (.mp4, .m4a)
- AAC (.aac)
- MP3 (.mp3)
- RTTTL/RTX (.rtttl, .rtx)
- OTA (.ota)
- iMelody (.imy)
- WAVE (.wav)
- Ogg (.ogg)
- Matroska (.mkv, Android 4.0+)
- etc.

▶ อ่านไฟล์ Video

- 3GPP (.3gp)
- MPEG-4 (.mp4)
- WebM (.webm)
- Matroska (.mkv, Android 4.0+)

ช่องทางการบันทึกข้อมูลของแอปพลิเคชันบนแอนดรอยด์

▶ ตัวแปรทั่วไป

- มีอายุชั่วคราว หลังจากปิด application ตัวแปรนี้จะถูกลบทิ้ง
- ประกาศและเรียกใช้งาน เหมือนการประกาศตัวแปรทั่วไป
- ข้อมูลจะถูกบันทึกใน `"/data/data/com.example.app_name/cache"`

▶ ตัวแปรถาวร

- มีอายุการใช้งานถาวร จนกว่า application จะถูก uninstall
- ข้อมูลจะถูกบันทึกใน `"/data/data/com.example.app_name/shared_prefs"`

▶ ไฟล์

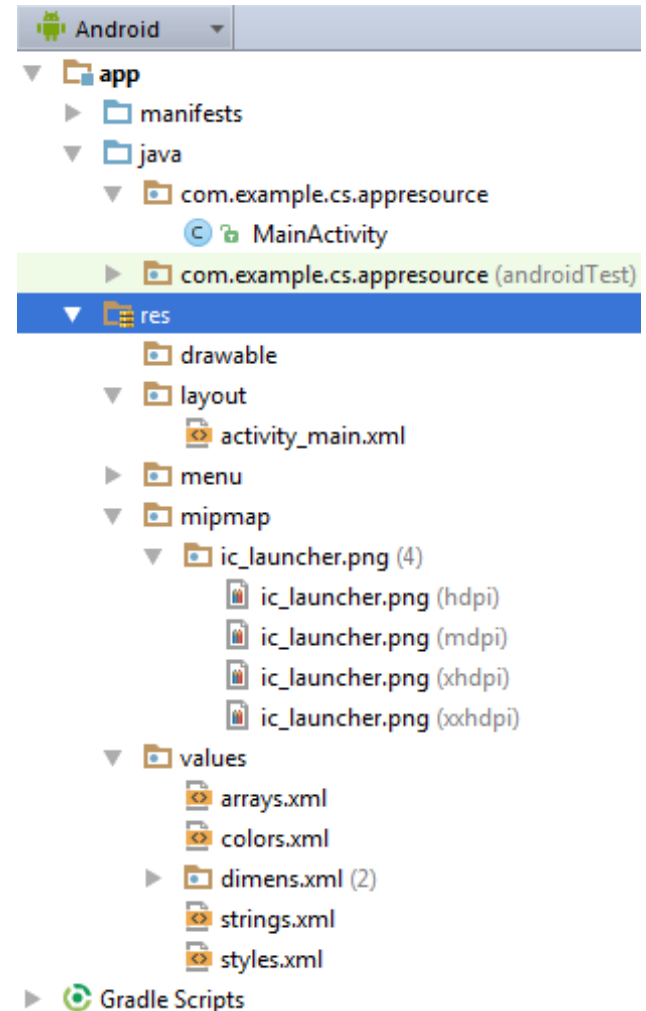
- บันทึกลงใน sub directory ของ application
 - ข้อมูลจะถูกบันทึกใน `"/data/data/com.example.app_name/files"`
- บันทึกลงใน public directory ของเครื่อง
 - เช่น `"/storage/sdcard0/Download"`
- บันทึกลงใน สื่อบันทึกภายนอก เช่น SD card
 - ข้อมูลจะถูกบันทึกลงใน `"storage/extSdCard"`
- บันทึกลงใน สื่อบันทึกออนไลน์ เช่น File Server

▶ ฐานข้อมูล

- ฐานข้อมูลภายในเครื่อง : SQLite
- ฐานข้อมูลออนไลน์ : SQL Server
 - จำเป็นต้องมี Internet Connection

Application Resources

- ▶ เราสามารถ อ้างถึงทรัพยากรใน application ของเราได้โดยใช้ resource IDs ที่อยู่ใน R class
- ▶ ทรัพยากรของ application จะอยู่ใน res/ directory
- ▶ MyProject/
 - src/
 - MainActivity.java
 - res/
 - drawable/
 - graphic.png
 - layout/
 - main.xml
 - info.xml
 - mipmap/
 - icon.png
 - values/
 - strings.xml



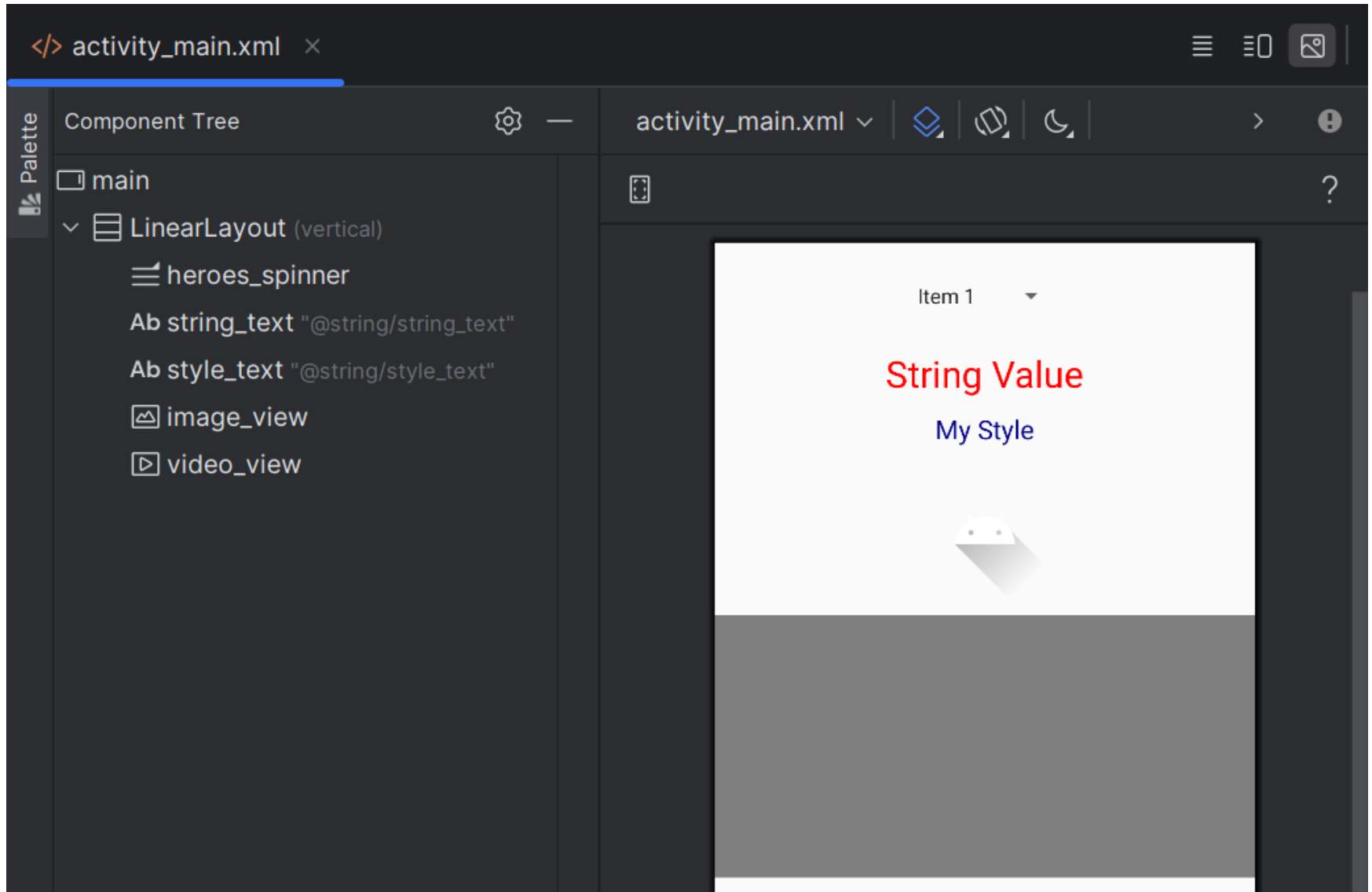
Resource Types

Directory	Resource Type
drawable/	Bitmap files (.png, .jpg, .gif) or XML files
mipmap/	Drawable files for different launcher icon densities.
layout/	XML files that define a user interface layout.
menu/	XML files that define application menus.
raw/	Arbitrary files to save in their raw form. To open these resources with a raw InputStream , call Resources.openRawResource() with the resource ID, which is R.raw.filename .
values/	<p>Each resource is defined with its own XML element. XML files that contain simple values, such as strings, colors and styles. For example, a <string> element creates an R.string resource and a <color> element creates an R.color resource.</p> <ul style="list-style-type: none"> ❖ arrays.xml for resource arrays ❖ colors.xml for color values ❖ dims.xml for dimension values ❖ strings.xml for string values ❖ styles.xml --> themes.xml for styles

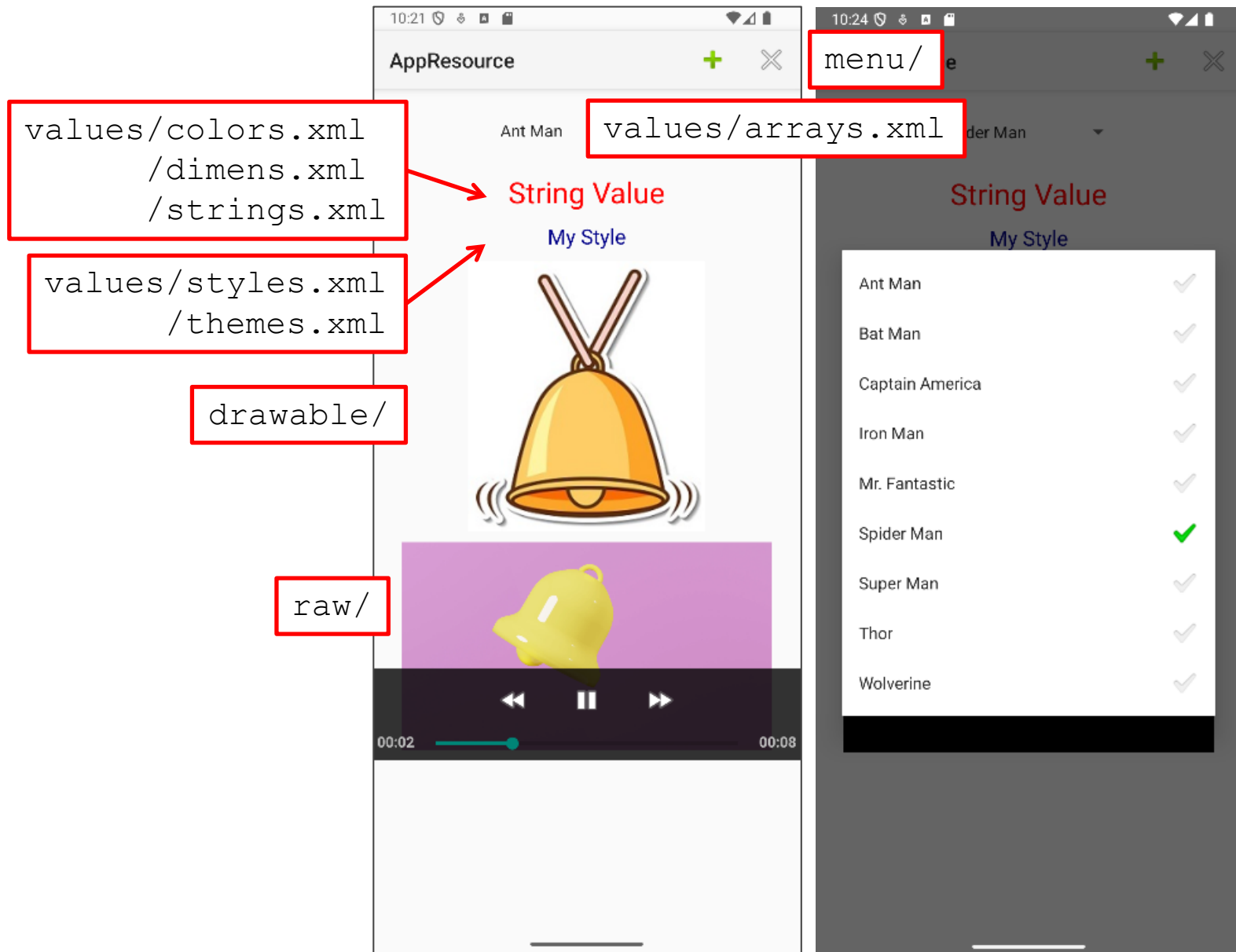
Caution: Never save resource files directly inside the res/ directory—it will cause a **compiler error**.

<https://developer.android.com/guide/topics/resources/providing-resources>

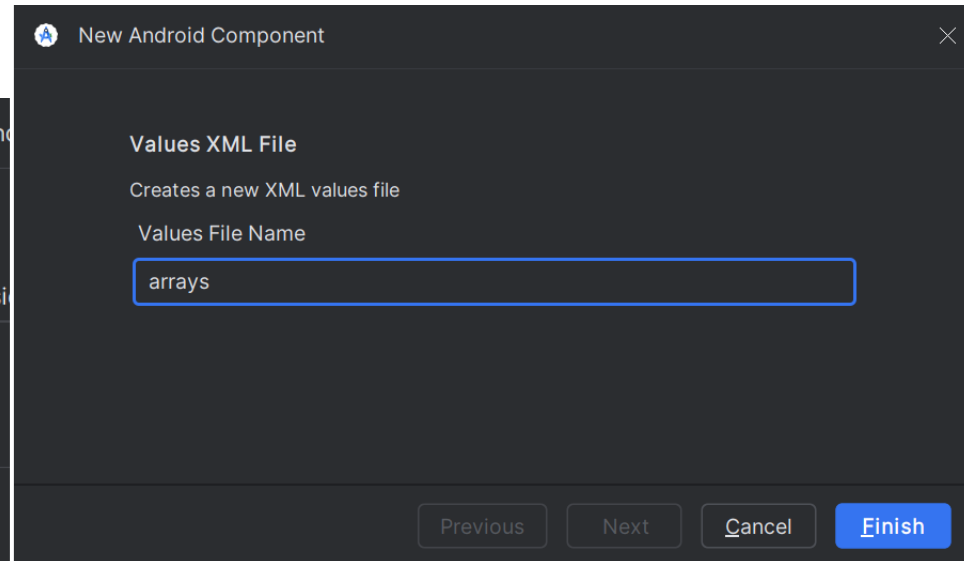
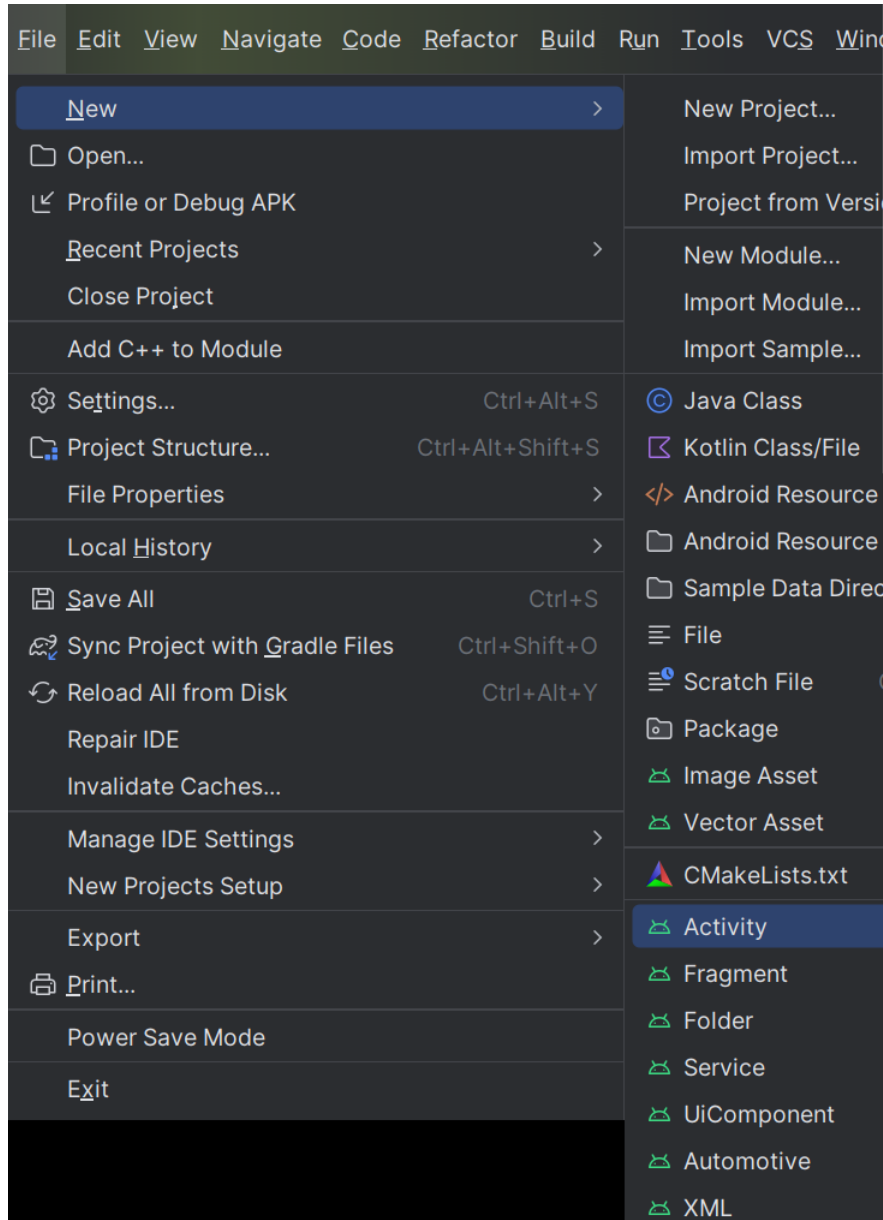
Create Layout



Run on Mobile Device



Create "arrays.xml"



arrays.xml

- ▶ ไฟล์ arrays.xml
- ▶ XML file save ไว้ที่ res/values/arrays.xml

Android ▾



▼ app

> manifests

> java

java (generated)

▼ res

> drawable

> layout

> menu

> mipmap

> raw

▼ values

</> arrays.xml

</> colors.xml

</> dims.xml

</> strings.xml

> themes (2)

> xml

res (generated)

> Gradle Scripts

</> arrays.xml ×

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3      <array name="heroes">
4          <item>Ant Man</item>
5          <item>Bat Man</item>
6          <item>Captain America</item>
7          <item>Iron Man</item>
8          <item>Mr. Fantastic</item>
9          <item>Spider Man</item>
10         <item>Super Man</item>
11         <item>Thor</item>
12         <item>Wolverine</item>
13     </array>
14 </resources>
```

ทรัพยากรอาเรย์ : res/values/arrays.xml

▶ ที่ไฟล์ Code เพิ่มใน onCreate()

```
1. Spinner spinner = (Spinner) findViewById(R.id.heroes_spinner);
2. ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(
    this, R.array.heroes, android.R.layout.simple_spinner_item
);
3. adapter.setDropDownViewResource(
    android.R.layout.simple_list_item_checked);
4. spinner.setAdapter(adapter);
```

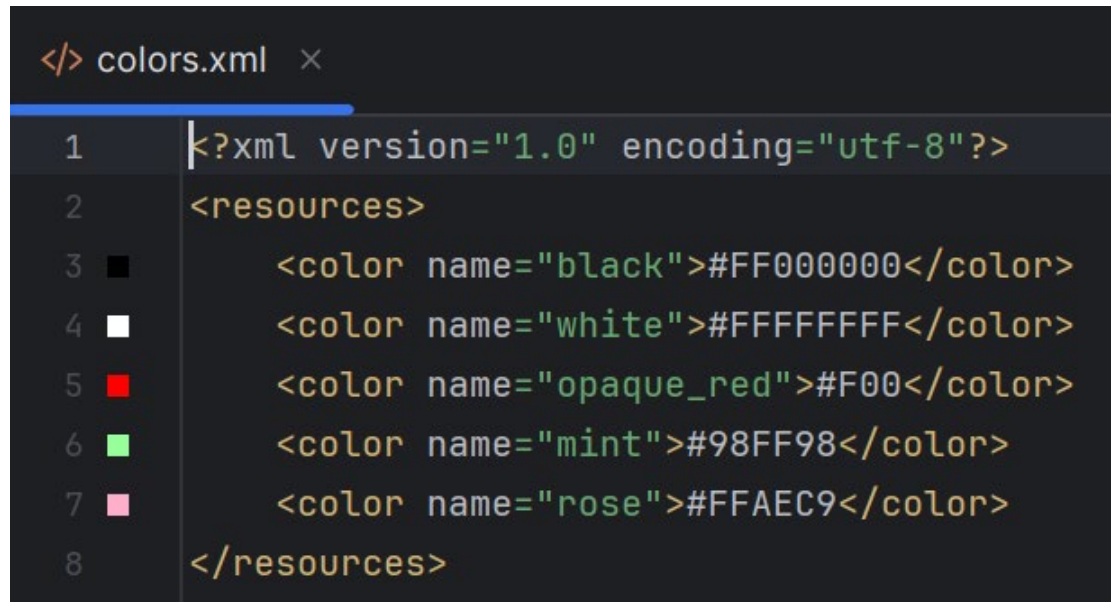
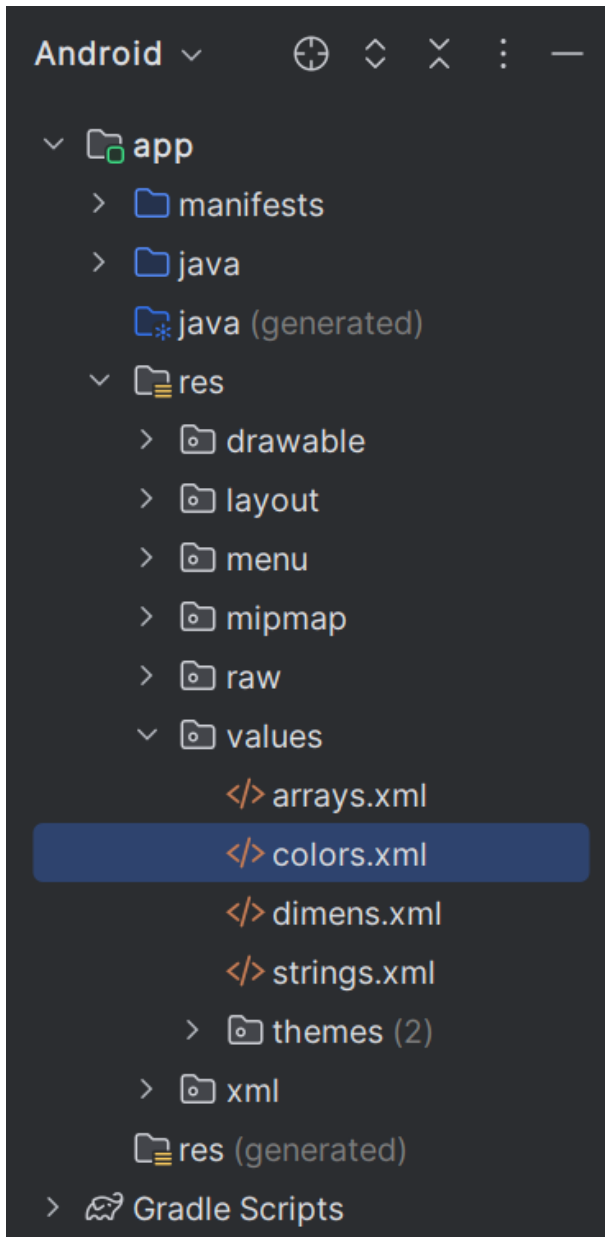
▶ ที่ไฟล์ Layout เพิ่ม spinner tag

<Spinner

```
    android:id="@+id/heroes_spinner"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:spinnerMode="dialog"
    android:minHeight="60dp" />
```

ทรัพยากรสี res/values/colors.xml

อยู่ในรูปแบบ
#RGB หรือ #RRGGBB



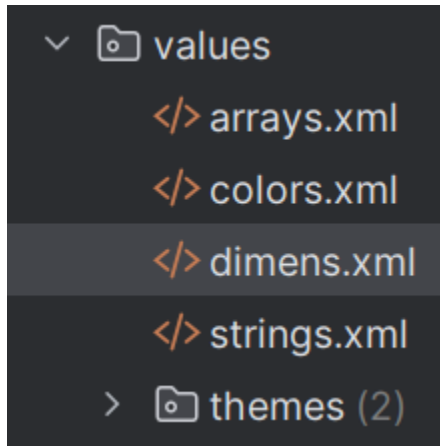
▶ This application code retrieves the color resource:

- 1) Resources res = getResources();
- 2) int color = res.getColor(**R.color.opaque_red**);

▶ This layout XML applies the color to an attribute:

- 1) <TextView
- 2) android:id="@+id/string_text"
- 3) android:layout_width="wrap_content"
- 4) android:layout_height="wrap_content"
- 5) android:layout_marginTop="10dp"
- 6) android:text="@string/string_text"
- 7) **android:textColor="@color/opaque_red"**
- 8) android:textSize="@dimen/large_size" />

ทรัพยากรขนาดอักษรและระยะห่าง : `dimens.xml`



```
</> dimens.xml ×
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3      <dimen name="large_size">28sp</dimen>
4      <dimen name="normal_size">16sp</dimen>
5      <dimen name="small_size">8sp</dimen>
6  </resources>
```

dp: Density-independent
Pixels - 160 dpi (dots per
inch)

sp: Scale-independent
Pixels - user's font size
preference

pt: Points - 1/72 of an inch
based on the physical size
of the screen.

px: Pixels - Corresponds to
actual pixels on the screen.
not recommended because
the actual representation
can vary across devices

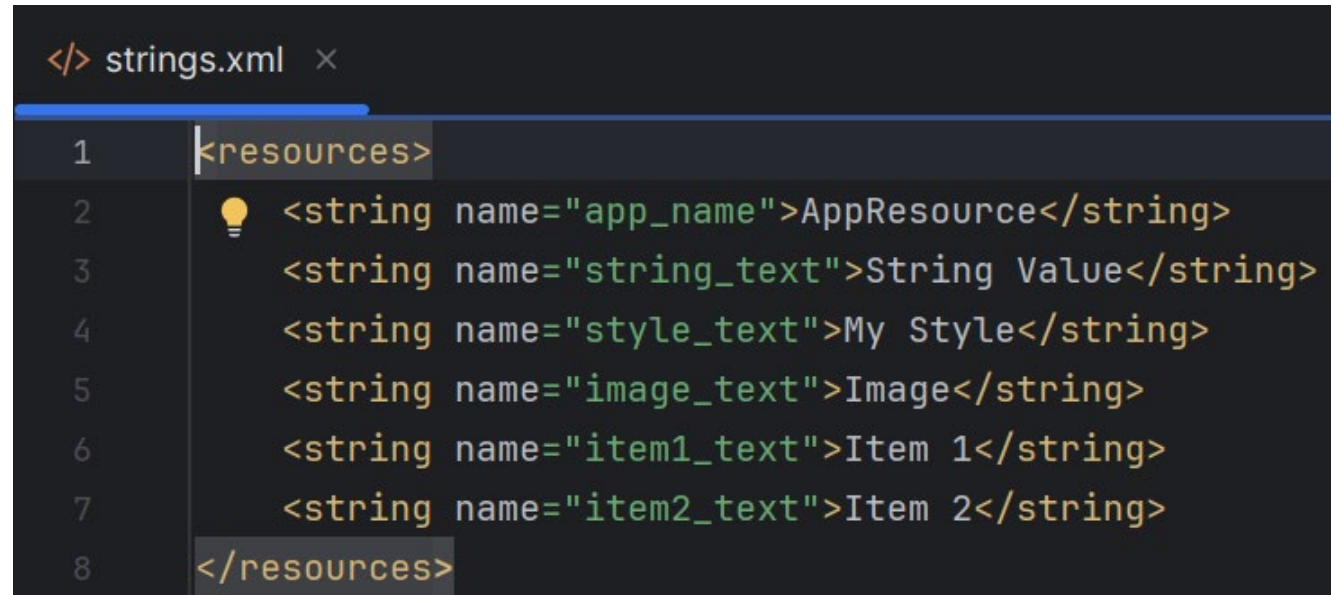
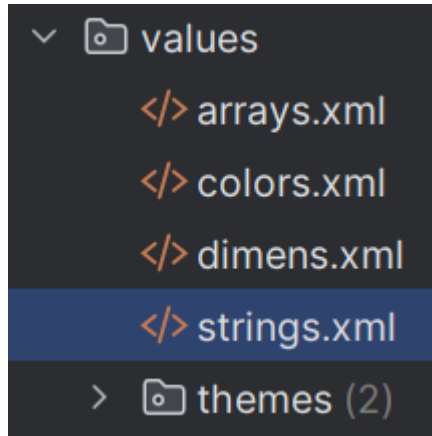
▶ This application code retrieves a dimension:

- 1) Resources res = `getResources()`;
- 2) float fontSize = res.`getDimension(R.dimen.normal_size)`;

▶ This layout XML applies dimensions to attributes:

- 1) `<TextView`
- 2) `android:id="@+id/string_text"`
- 3) `android:layout_width="wrap_content"`
- 4) `android:layout_height="wrap_content"`
- 5) `android:layout_marginTop="10dp"`
- 6) `android:text="@string/string_text"`
- 7) `android:textColor="@color/opaque_red"`
- 8) `android:textSize="@dimen/large_size" />`

ทรัพยากรตัวแปรข้อความ : res/values/strings.xml



- ▶ This layout XML applies a string to a View:

- 1) <TextView
- 2) android:id="@+id/string_text"
- 3) android:layout_width="wrap_content"
- 4) android:layout_height="wrap_content"
- 5) android:layout_marginTop="10dp"
- 6) **android:text="@string/string_text"**
- 7) android:textColor="@color/opaque_red"
- 8) android:textSize="@dimen/large_size" />


- ▶ This application code retrieves a string:

- 1) String string = getString(**R.string.hello**);
- ▶ You can use either getString(int) or getText(int) to retrieve a string.

ทรัพยากรรูปแบบการแสดงผล : themes.xml (styles.xml)

- ▶ A style resource defines the format and look for a UI.
 - A style can be applied to an individual View (from within a layout file)

</> themes.xml ×

```
1 <resources xmlns:tools="http://schemas.android.com/tools">
2    <style name="Base.Theme.AppResource" parent="Theme.Material3.DayNight.NoActionBar">
3     </style>
4
5   <style name="Theme.AppResource" parent="Theme.AppCompat.DayNight" />
6   <style name="my_style" parent="Theme.AppResource">
7     <item name="android:textSize">20sp</item>
8     <item name="android:textColor">#008</item>
9   </style>
10 </resources>
```

values

</> arrays.xml

</> colors.xml

</> dimens.xml

</> strings.xml

themes (2)

</> themes.xml

</> themes.xml (night)

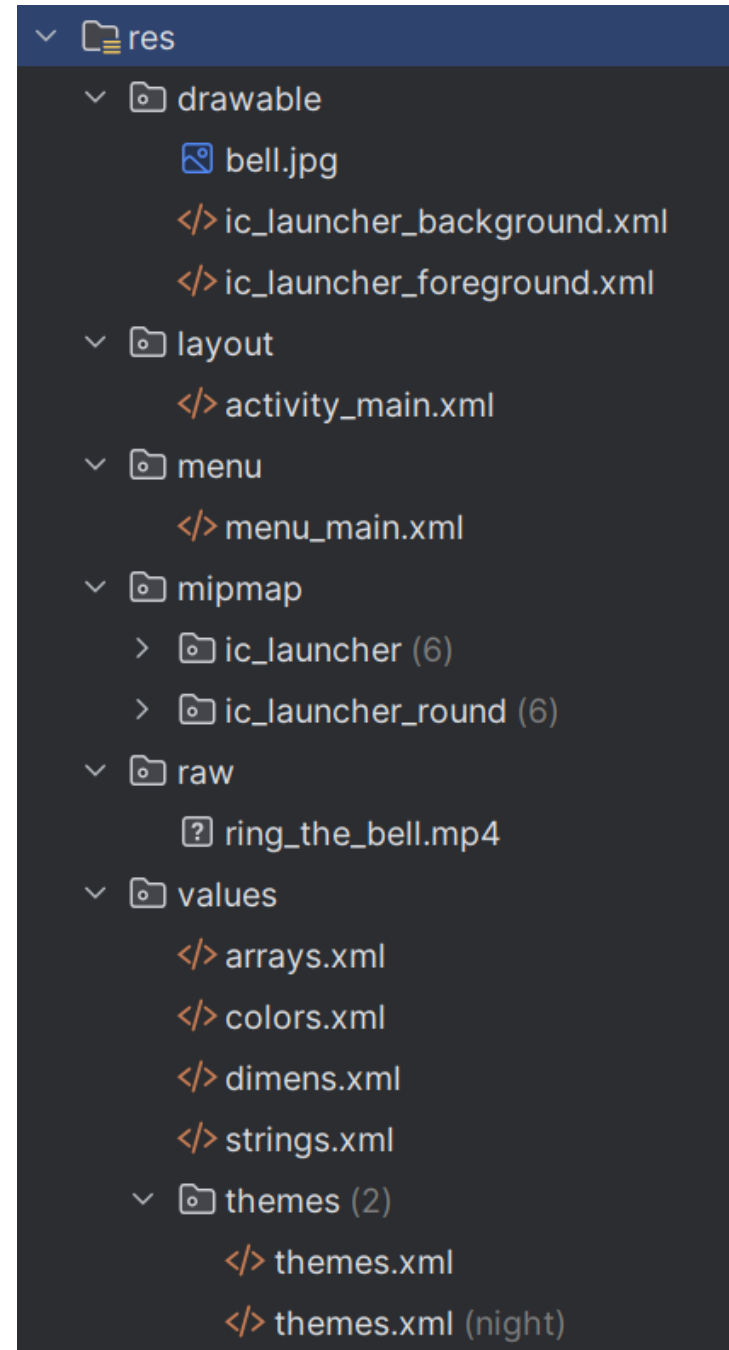
- ▶ XML file that applies the style to a TextView :

1. <TextView
2. android:id="@+id/style_text"
3. **style="@style/my_style"**
4. android:layout_width="wrap_content"
5. android:layout_height="wrap_content"
6. android:layout_marginTop="10dp"
7. android:text="@string/style_text" />

Accessing Resources

- ▶ **R** : Project Resources in “**res/**” directory
- ▶ **Resource ID** : Integer
 - Resource type
 - drawable, layout, menu, mipmap, string
 - Resource name
 - Filename without extension, or,
 - Value in XML android:name
 - Syntax:
 - “Project” + “Resource type” + “Resource name”
- ▶ Two ways to access a resource:
 - Code (ไฟล์ java) : R.string.hello
 - string : “resource type”
 - hello : “resource name”
 - Layout (ไฟล์ XML): @string/hello
 - string : “resource type”
 - hello : “resource name”

Folder “values” พิเศษกว่า Folder อื่น อย่างไร?



Accessing Resources in Code (1/3)

- ▶ Syntax : [`<package_name>`].`R.<resource_type>.<resource_name>`
 - `<package_name>` is the name of the package in which the resource is located
 - Not required when referencing resources from your own package
 - `<resource_type>` is the R subclass for the resource type
 - `<resource_name>` is the resource filename without extension or
 - `android::name` attribute value in the XML element

Example

1) `ImageView img = findViewById(R.id.image_view);`

2) `img.setImageResource(R.drawable.bell);` อ้างถึง resource
 ในหน้า layout ใช้ R.id.???

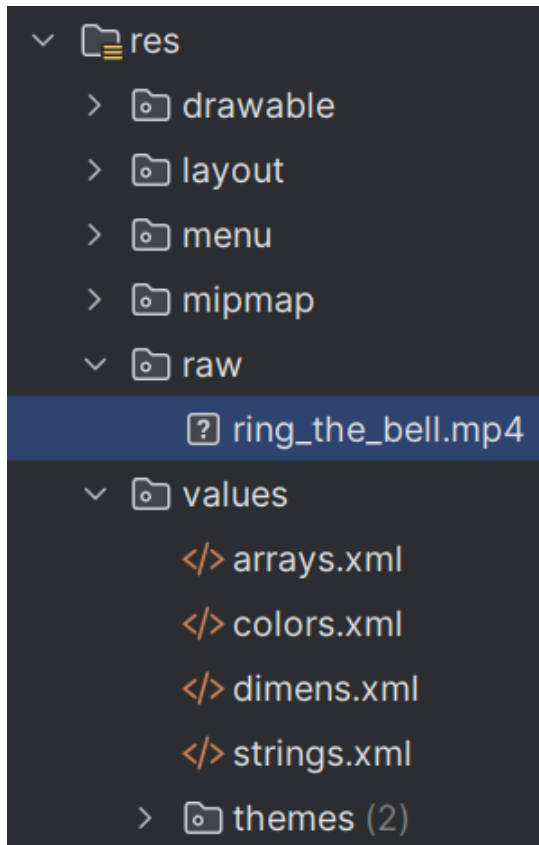
- ▶ `"res/"` : get resource ID
 - reference them through the R class or @ from XML resources
- ▶ `"assets/"` : get original file
 - read raw data using **AssetManager**
- ▶ Video / Audio files
 - save file in the `"res/raw/"` directory and then read a stream of bytes

Accessing Resources in Code (2/3)

Video

- 1) `VideoView mVideoView = findViewById(R.id.video_view);`
- 2) `mVideoView.setVideoURI(Uri.parse("android.resource://" + getPackageName() + "/" + R.raw.ring_the_bell));`
- 3) `mVideoView.setMediaController(new MediaController(this));`
- 4) `mVideoView.requestFocus();`

มีข้อจำกัดในการตั้งชื่อ
ไฟล์วีดีโอ อย่างไร?



`<ImageView`

```
    android:id="@+id/image_view"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:contentDescription="@string/image_text"
    android:maxHeight="200dp"
    android:src="@drawable/ic_launcher_foreground"
    tools:ignore="ImageContrastCheck" />
```

`<VideoView`

```
    android:id="@+id/video_view"
    android:layout_width="wrap_content"
    android:layout_height="200dp"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="10dp" />
```

Accessing Resources in Code (3/3)

ตัวอย่างการอ้างถึงทรัพยากรของแอปพลิเคชัน

1. Load a background for the current screen from a drawable resource
 - `getWindow().setBackgroundDrawableResource(R.drawable.view) ;`
2. ตั้งค่า Activity Title จากข้อความใน Resource Object
 - `getWindow().setTitle(getResources().getText(R.string.app_name));`
 - ฟังก์ชัน `setTitle` ต้องการ parameter เป็น character sequence ไม่ใช่ resource ID
3. Load a custom layout for the current screen
 - `setContentView(R.layout.activity_main);`
4. ตั้งค่าข้อความใน TextView ด้วย resource ID
 - `TextView msg = findViewById(R.id.msg);`
 - `msg.setText(R.string.hello_message);`

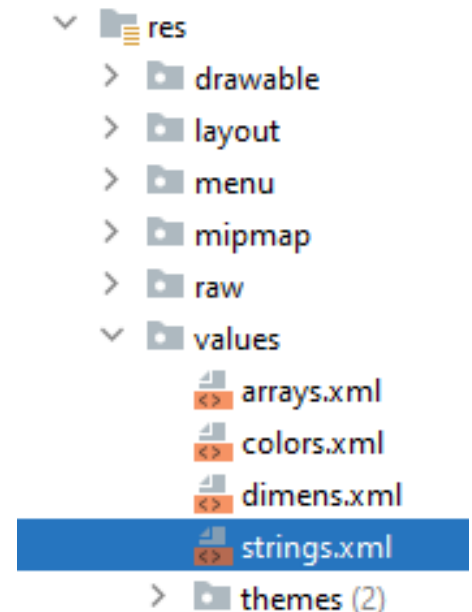
Accessing Resources from XML (1/2)

▶ Syntax

- `@[<package_name>:]<resource_type>/<resource_name>`
- `<package_name>` is the name of the package in which the resource is located
 - Not required when referencing resources from the same package
- `<resource_type>` is the R subclass for the resource type
- `<resource_name>` is the resource filename without extension or
 - `android::name` attribute value in the XML element

▶ ตัวอย่างในไฟล์ Layout

- ▶ `<Button` `android: layout_width = "match_parent"`
 `android: layout_height = "wrap_content"`
 `android: text="@string/submit" />`



Accessing Resources from XML (2/2)

ใน res/values/strings.xml

```
1) <?xml version="1.0" encoding="utf-8"?>
2) <resources>
3)   <string name="hello">Hello!</string>
4) </resources>
```

ใน res/values/colors.xml

```
1) <?xml version="1.0" encoding="utf-8"?>
2) <resources>
3)   <color name="opaque_red">#f00</color>
4) </resources>
```

การอ้างถึง resource ในไฟล์ XML

```
1) <?xml version="1.0" encoding="utf-8"?>
2) <EditText      android:layout_width="match_parent"
3)               android:layout_height="match_parent"
4)               android:textColor="@color/opaque_red"
5)               android:text="@string/hello" />
```

To reference a system resource

```
1) <?xml version="1.0" encoding="utf-8"?>
2) <EditText      android:layout_width="match_parent"
3)               android:layout_height="match_parent"
4)               android:textColor="@android:color/secondary_text_dark"
5)               android:text="@string/hello" />
```

References

1. App resources overview, URL:
<http://developer.android.com/guide/topics/resources/providing-resources.html> [Available on August 2020]
2. Providing alternative resources, URL:
<http://developer.android.com/guide/topics/resources/providing-resources.html#AlternativeResources> [Available on August 2020]