

Android

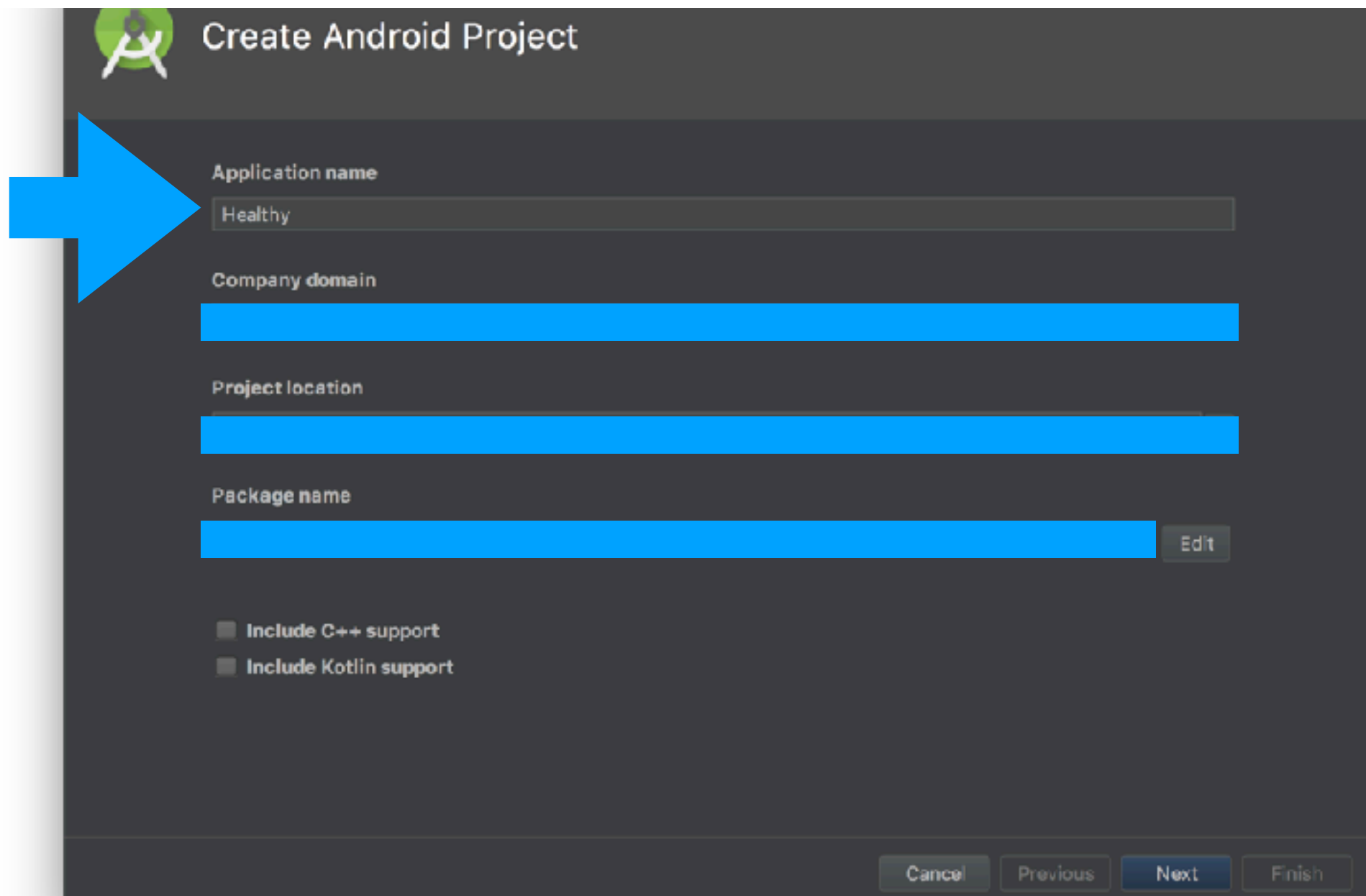
Working with Fragment

“ในโรงเรียน ไม่มีคนโกง หรือคนฉลาด
มีแต่คนที่ตั้งใจเรียน กับคนที่ไม่ตั้งใจเรียน
เพราะไม่มีมิดด้ามไหนด ที่ลับแล้วไม่คม”

-ใครสักคนนึง

สิ่งที่ต้องมีก่อนเริ่ม

- Account github.com
- สำหรับคนที่มิแล้วให้ส่ง email มาที่ [sommai.k@gmail.com](mailto:sommaik@gmail.com)
 - หัวข้อของ email
 - 999999 นาย สมหมาย กรังพานิช
 - เนื้อหาใน email
 - รหัสนักศึกษา : 9999999
 - ชื่อ : นาย สมหมาย กรังพานิช
 - github_url : <https://github.com/sommaik>
- Clone slide ได้ที่ <https://github.com/sommaik/android-2018-01>



Create Android Project

Application name
Healthy

Company domain

Project location

Package name

☐ Include C++ support
☐ Include Kotlin support

Cancel Previous Next Finish

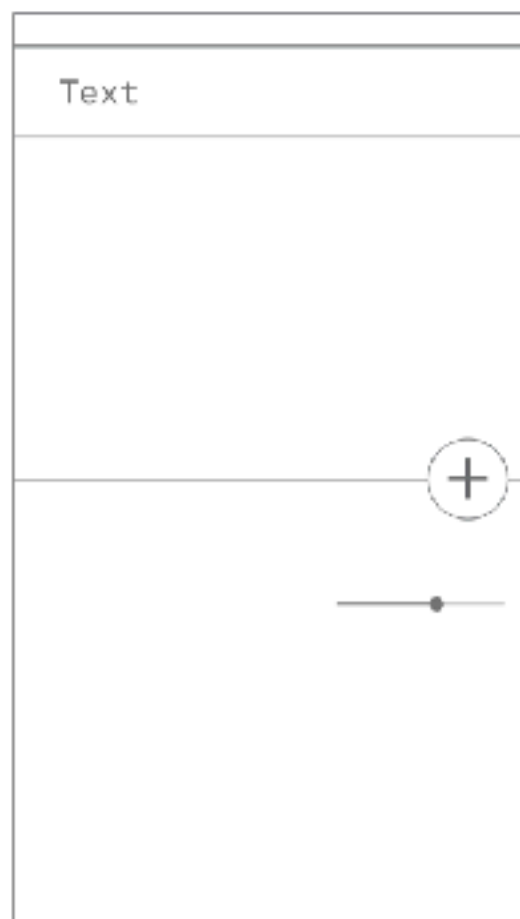
สร้าง project ใหม่

ชื่อ Healthy

เปลี่ยน theme

เครื่องมือช่วย

<https://material.io/tools/color>

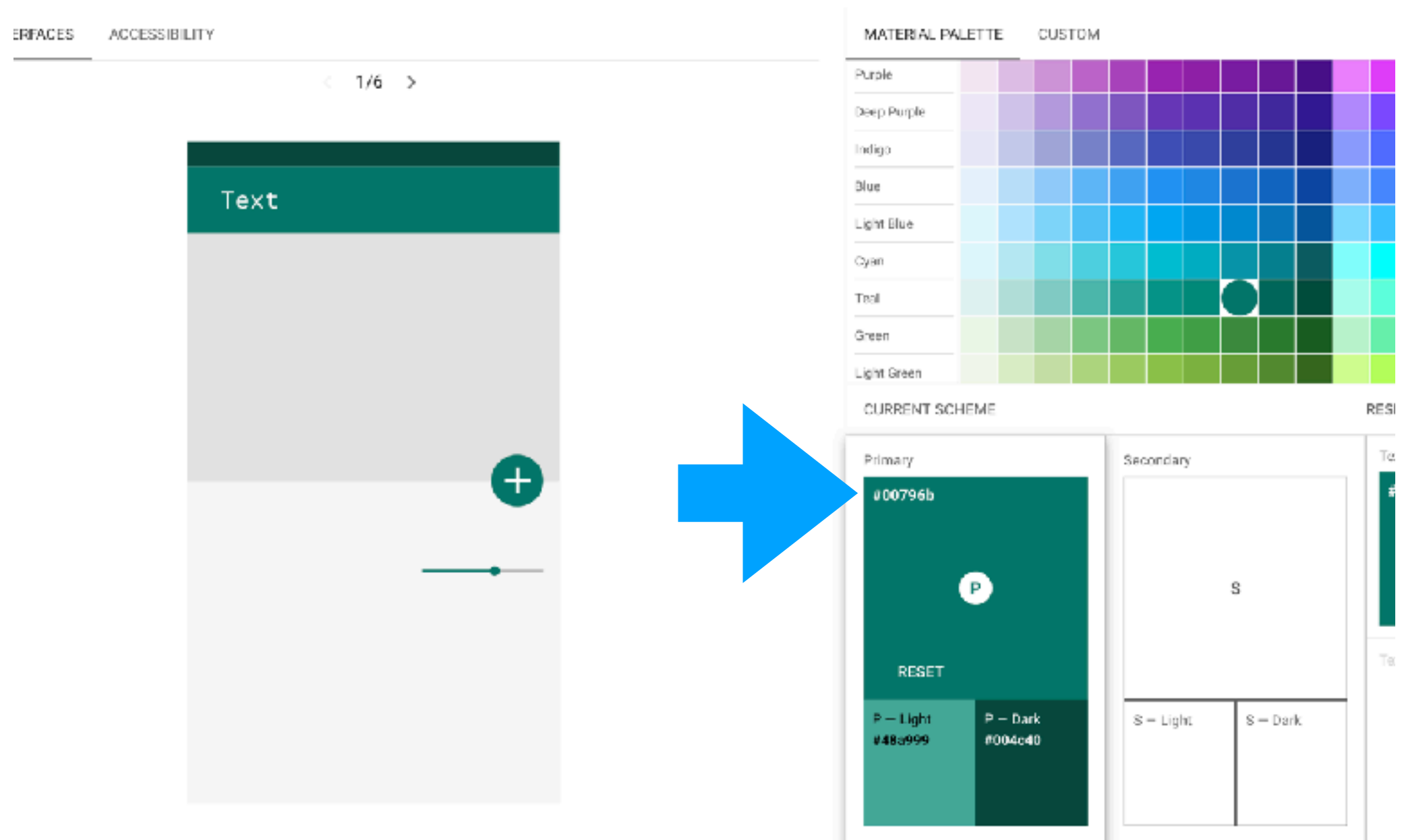


	50	100	200	300	400	500	600	700	800	900	A	A	A
Red													
Pink													
Purple													
Deep Purple													
Indigo													
Blue													
Light Blue													

Primary	Secondary	Text on P
P	S	T
P - Light	S - Light	Text on S
		T
P - Dark	S - Dark	

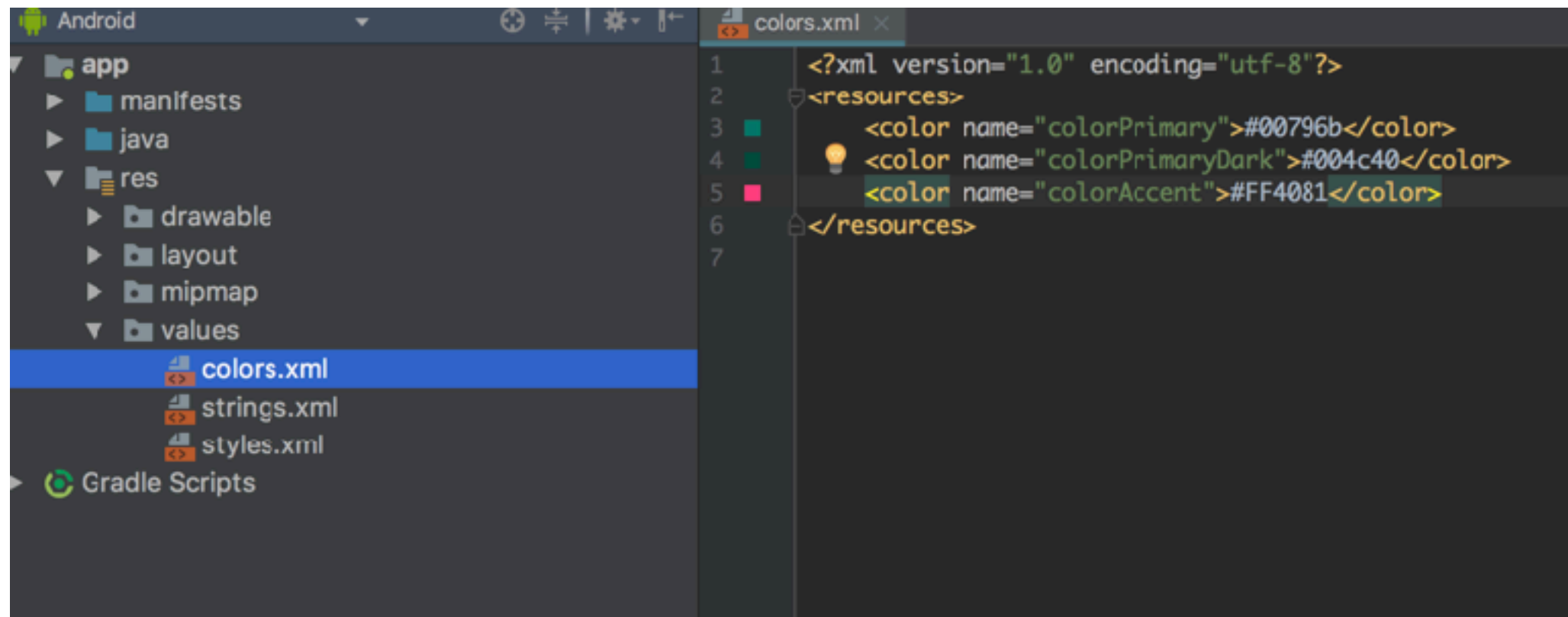
เลือกสีที่ต้องการ

กดตามลูกศร



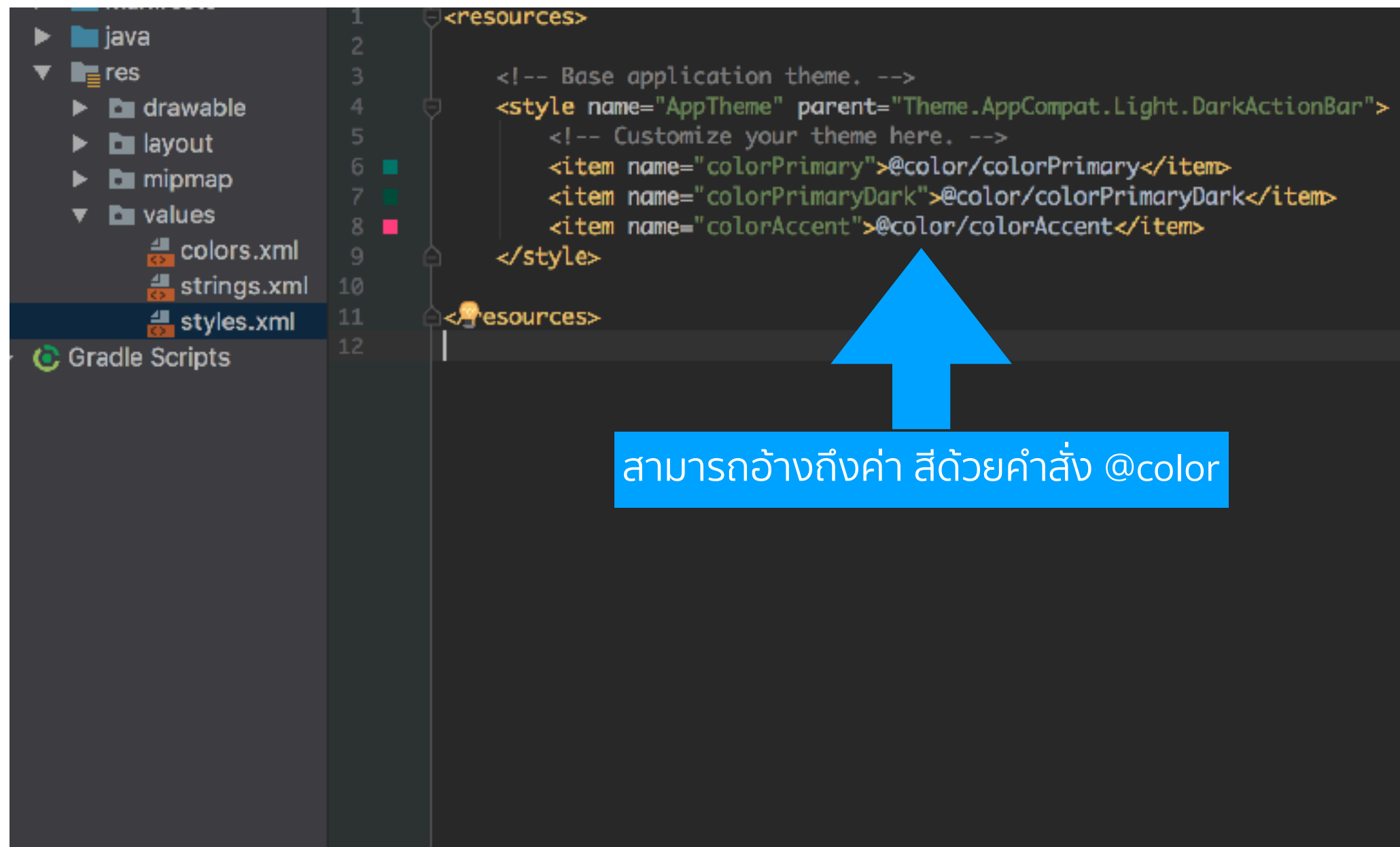
เมื่อได้สีที่ต้องการแล้ว

ให้ copy รหัสสีตามลูกศร



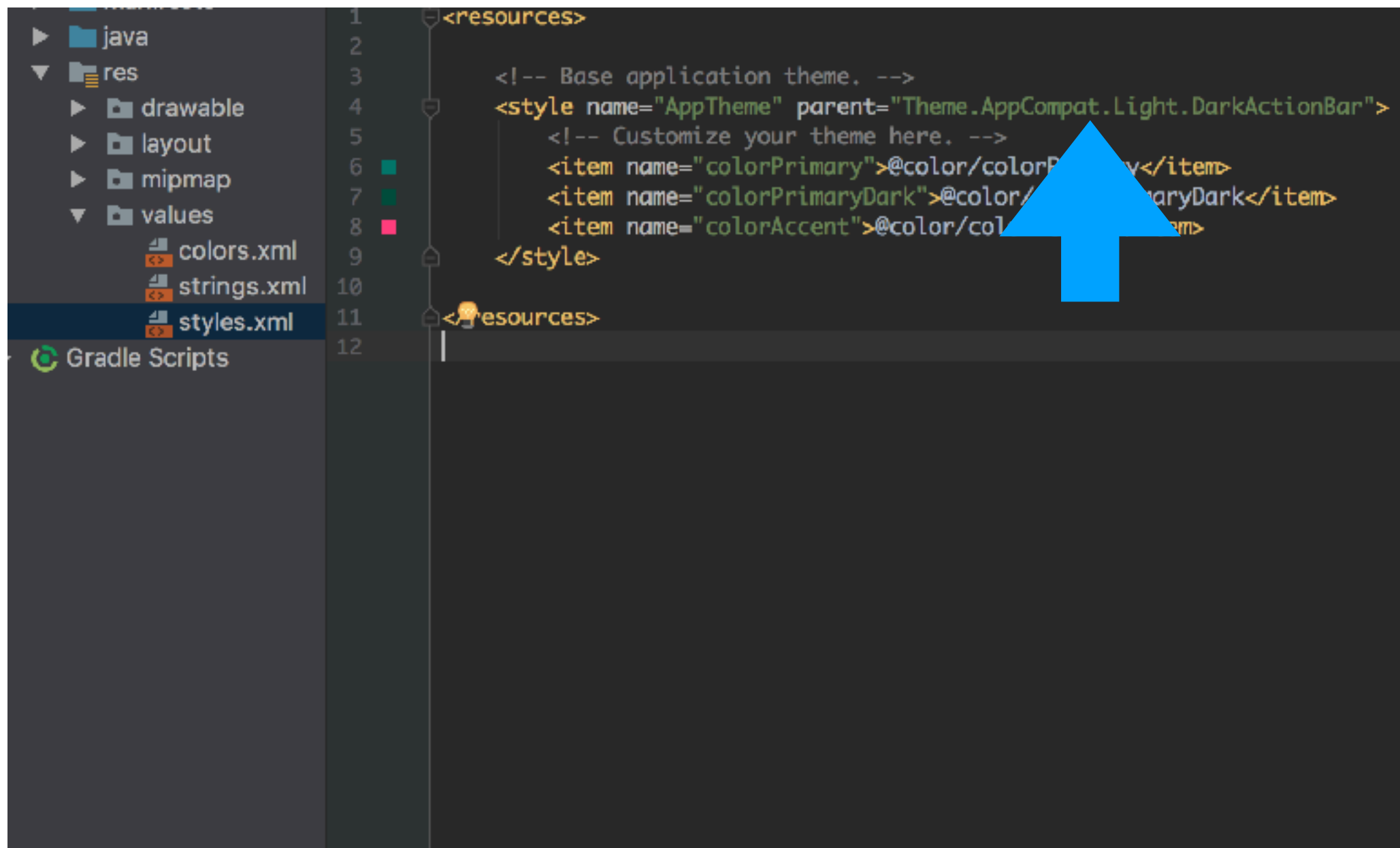
นำค่าที่ได้มาใส่

ที่ file colors.xml



ตั้งค่า Theme

แก้ไข file styles.xml

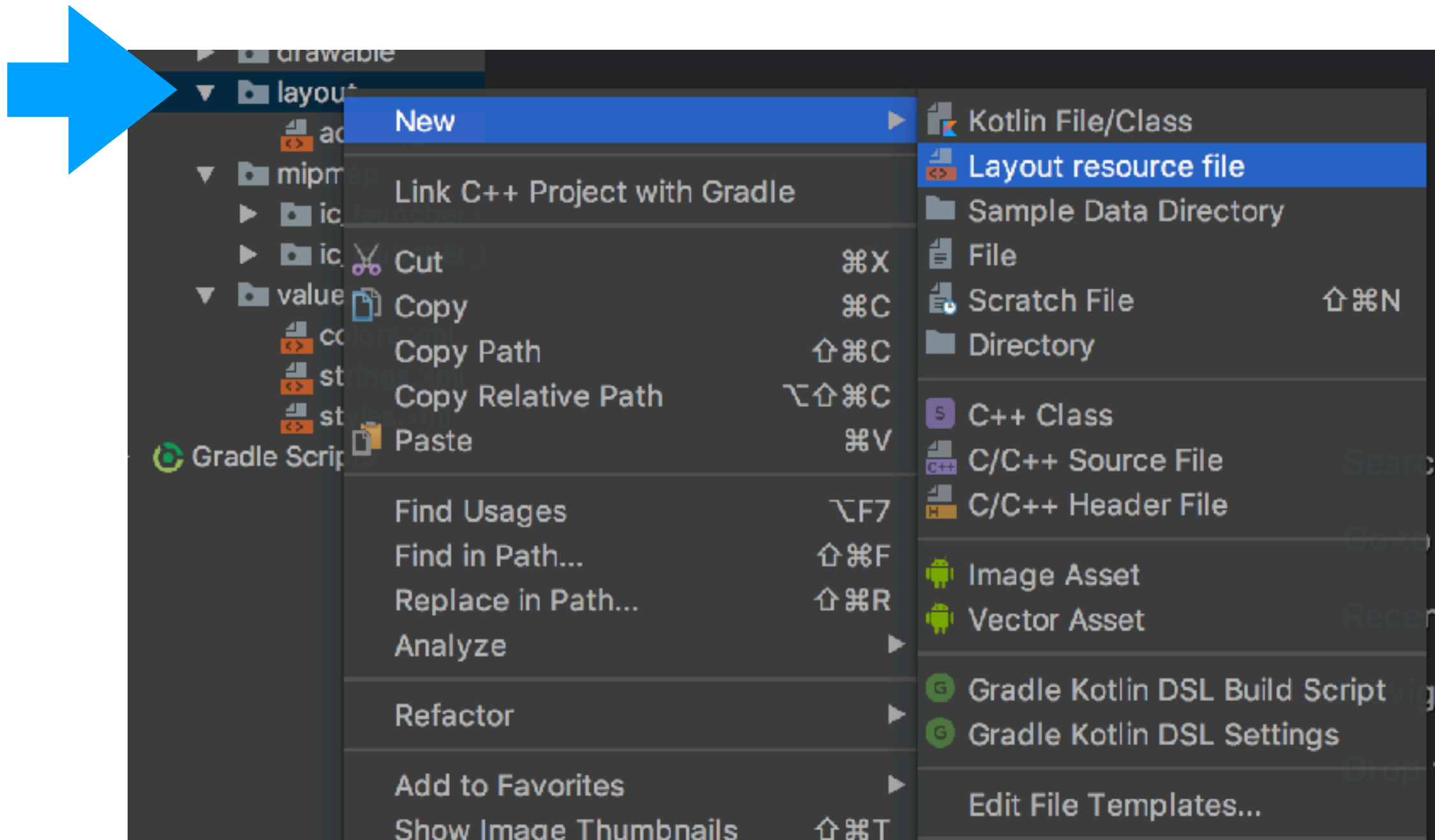


เอา default title bar ออก

เปลี่ยนจาก Theme.AppCompat.Light.DarkActionBar
เป็น Theme.AppCompat.Light.NoActionBar

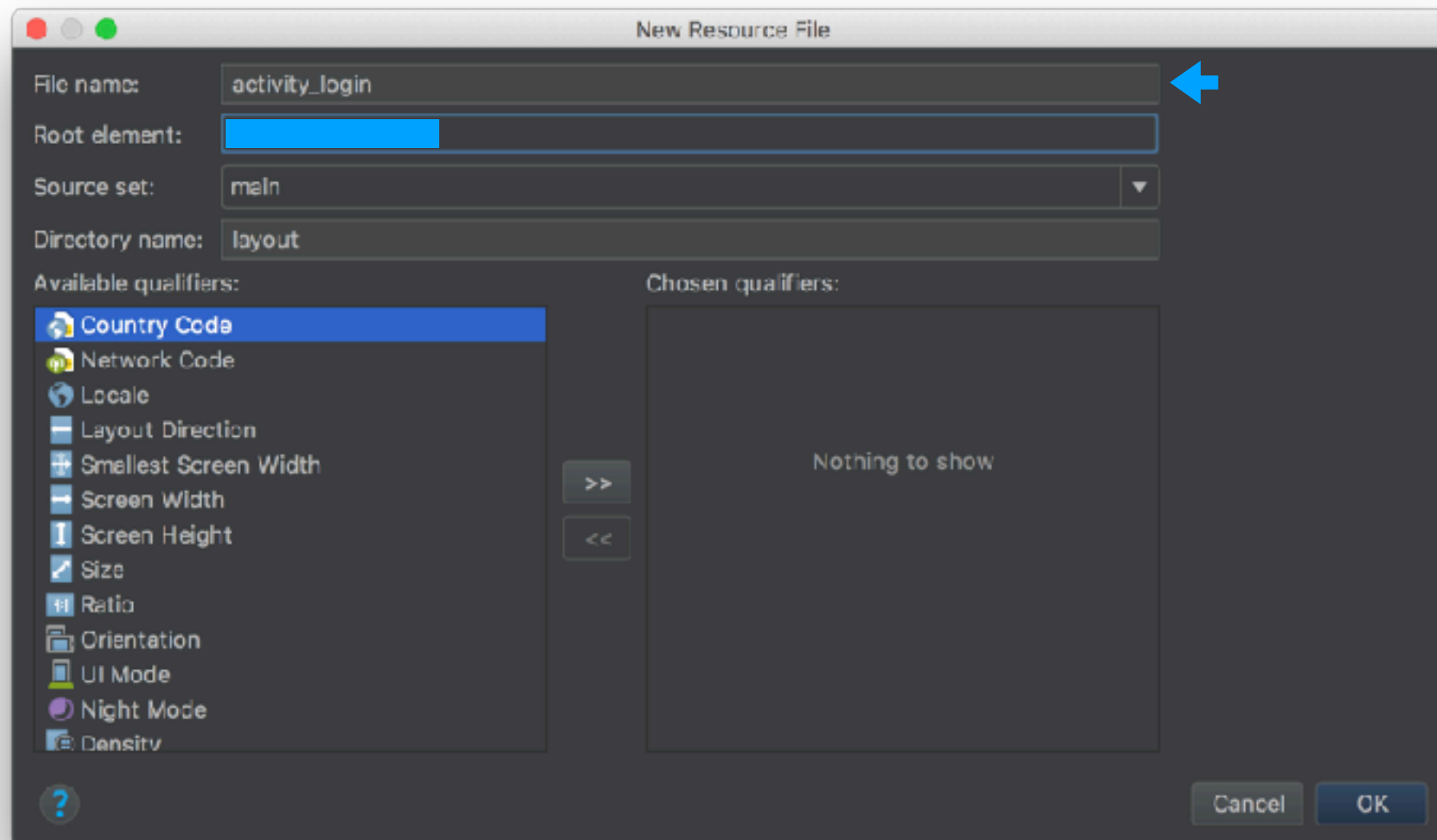
วิธีการสร้าง

Fragment



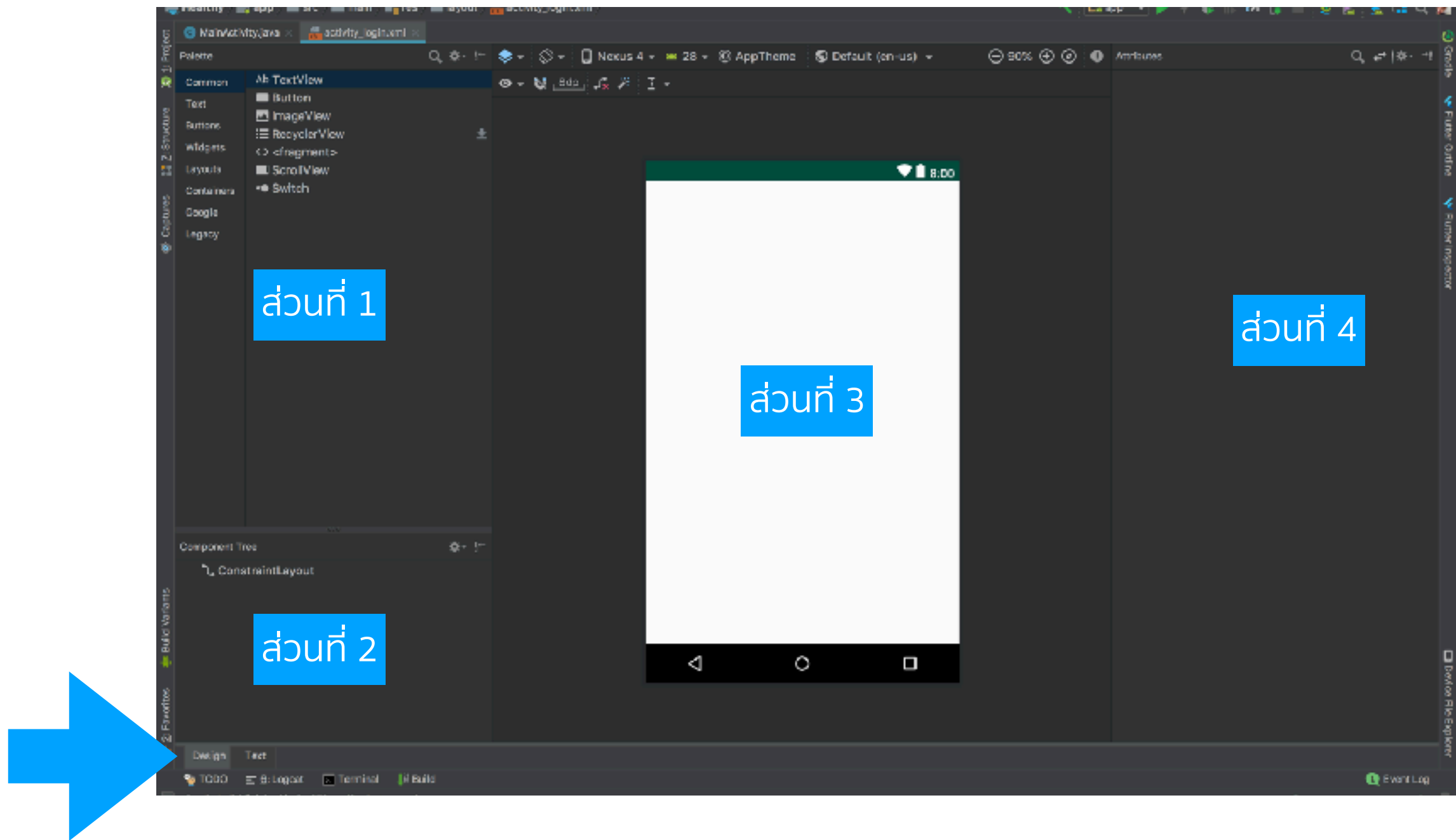
สร้าง file layout

Click ขวาที่ layout เลือก New เลือก Layout resource file



Sະຸ File name

Name ໃກ້ໄກຢູ່ fragment_login



ส่วนประกอบของหน้าจอ

Mode Design

ส่วนที่ 1

- เป็นส่วนที่เก็บรวบรวม component ทั้งหมดของ android
- สามารถลาก วาง component ที่ต้องการได้
- โดยการลากมาวางในส่วนที่ 3

ส่วนที่ 2

- ส่วนนี้จะแสดงรายการของ component ทั้งหมดที่มีอยู่ของหน้าจอ
- เราสามารถย้ายลำดับของ component ได้
- จะแสดงผลอยู่ในรูปแบบของ tree แสดงชั้นของ component


ส่วนที่ 3

- ส่วนแสดงผล
- ในส่วนนี้เราจะเห็นผลของหน้าโปรแกรมคร่าวๆ
- เมื่อเราเลือกไปที่ component ใด ก็จะมี properties ของ component นั้นๆ แสดงอยู่ในส่วนที่ 4

ส่วนที่ 4

- ส่วนนี้เป็นส่วนของการตั้งค่า properties ของ component
- เราสามารถแก้ไขค่าของ component ได้
- หลังจากที่เราแก้ไขค่าแล้วโปรแกรมก็จะแสดงผลในส่วนที่ 3

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        // setContentView(R.layout.activity_main);  
        setContentView(R.layout.fragment_login);  
    }  
}
```



วิธีการอ้างอิง Fragment

จาก code java จะสามารถอ้างอิงถึง fragment ผ่าน Class R.layout.ชื่อ_fragment

Layouts

Android layouts

Layouts

- A layout defines the structure for a user interface in your app, such as in an activity
- All elements in the layout are built using a hierarchy of View and ViewGroup objects.
- A View usually draws something the user can see and interact with.
- Whereas a ViewGroup is an invisible container that defines the layout structure for View and other ViewGroup objects

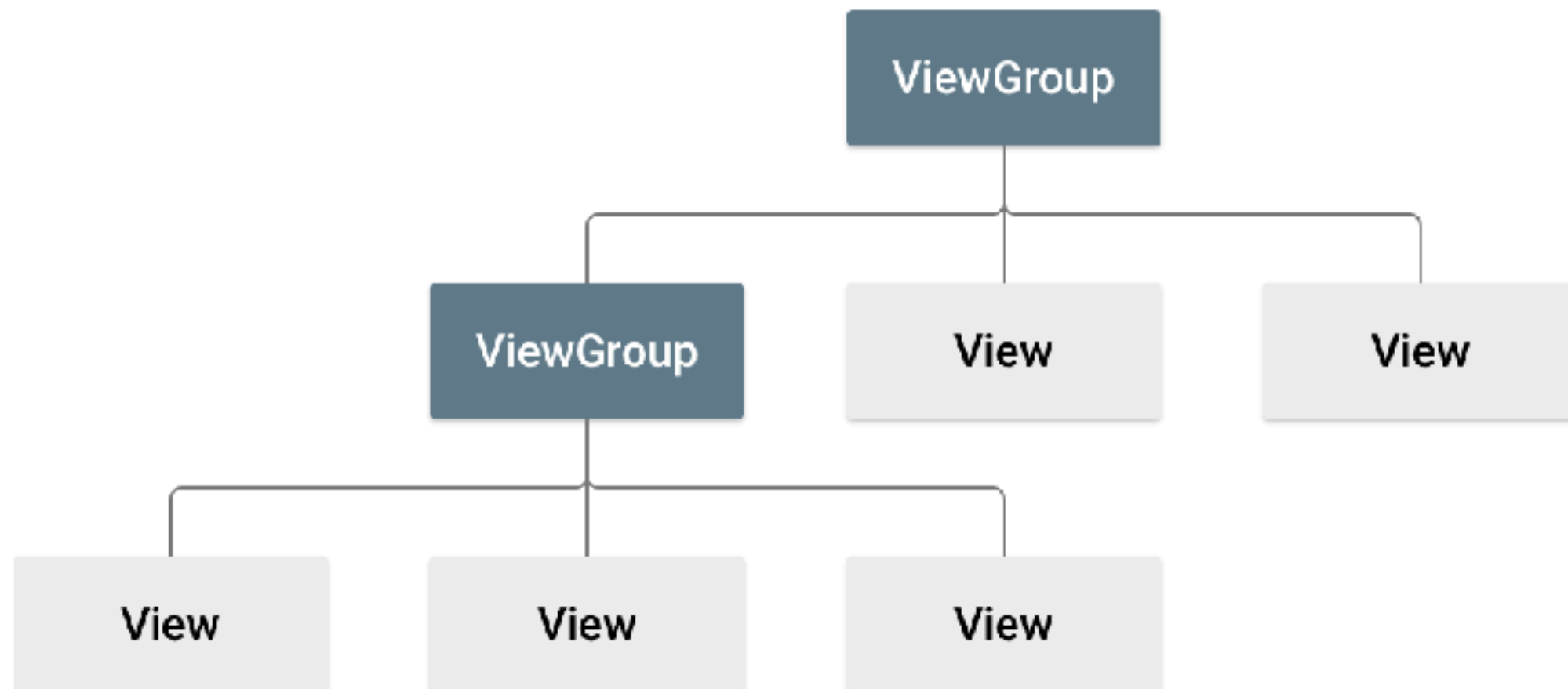


Illustration of a view hierarchy
which defines a UI layout

You can declare a layout in two ways

- **Declare UI elements in XML.** Android provides a straightforward XML vocabulary that corresponds to the View classes and subclasses, such as those for widgets and layouts. You can also use Android Studio to build your XML layout using a drag-and-drop interface.
- **Instantiate layout elements at runtime.** Your app can create View and ViewGroup objects (and manipulate their properties) programmatically.

Common Layout

LinearLayout

A layout that organizes its children into a single horizontal or vertical row. It creates a scrollbar if the length of the window exceeds the length of the screen.

All children of a LinearLayout are stacked one after the other, so a vertical list will only have one child per row, no matter how wide they are, and a horizontal list will only be one row high (the height of the tallest child, plus padding).

A LinearLayout respects margins between children and the gravity(right, center, or left alignment) of each child.



LinearLayout

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    android:paddingLeft="16dp"
```

```
    android:paddingRight="16dp"
```

```
    android:orientation="vertical" >
```

```
.....
```

```
</LinearLayout>
```

Relative Layout

RelativeLayout is a view group that displays child views in relative positions. The position of each view can be specified as relative to sibling elements (such as to the left-of or below another view) or in positions relative to the parent RelativeLayout area (such as aligned to the bottom, left or center).



Layout properties

android:layout_alignParentTop

- If True makes the top edge of this view match the top edge of the parent.

android:layout_centerVertical

- If True centers this child vertically within its parent.

android:layout_below

- Positions the top edge of this view below the view specified with a resource ID.

android:layout_toRightOf

- Positions the left edge of this view to the right of the view specified with a resource ID.

RelativeLayout

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp" >
    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/reminder" />
    <Spinner
        android:id="@+id/dates"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/name"
        android:layout_alignParentLeft="true"
        android:layout_toLeftOf="@+id/times" />
    <Spinner
        android:id="@id/times"
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/name"
        android:layout_alignParentRight="true" />
    <Button
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/times"
        android:layout_alignParentRight="true"
        android:text="@string/done" />
</RelativeLayout>
```

Webview

If you want to deliver a web application (or just a web page) as a part of a client application, you can do it using WebView. The WebView class is an extension of Android's View class that allows you to display web pages as a part of your activity layout. It does not include any features of a fully developed web browser, such as navigation controls or an address bar. All that WebView does, by default, is show a web page.

A common scenario in which using WebView is helpful is when you want to provide information in your app that you might need to update, such as an end-user agreement or a user guide. Within your Android app, you can create an Activity that contains a WebView then use that to display your document that's hosted online.

Another scenario in which WebView can help is if your app provides data to the user that always requires an Internet connection to retrieve data, such as email. In this case, you might find that it's easier to build a WebView in your Android app that shows a web page with all the user data, rather than performing a network request, then parsing the data and rendering it in an Android layout. Instead, you can design a web page that's tailored for Android devices and then implement a WebView in your Android app that loads the web page.

WebView

```
// Xml file
```

```
<WebView  
    android:id="@+id/webview"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
>
```

```
// Java File
```

```
WebView myWebView = (WebView) findViewById(R.id.webview);  
myWebView.loadUrl("https://www.example.com");
```


```
// Manifest
```

```
<manifest ... >  
    <uses-permission android:name="android.permission.INTERNET" />  
    ...  
</manifest>
```


ประกาศค่า String

ใน file strings.xml

```
resources>  
  <string name="app_name">Healthy</string>  
  <string name="user_id">User Id</string>  
  <string name="password">Password</string>  
resources>
```



ใส่ค่า name และ value

Name จะเป็นชื่อที่เอาไว้อ้างอิงในระบบ


```
android:layout_height="match_parent">

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <EditText
        android:hint="@string/user_id"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</LinearLayout>

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



ใส่ค่าเริ่มต้นด้วย @string

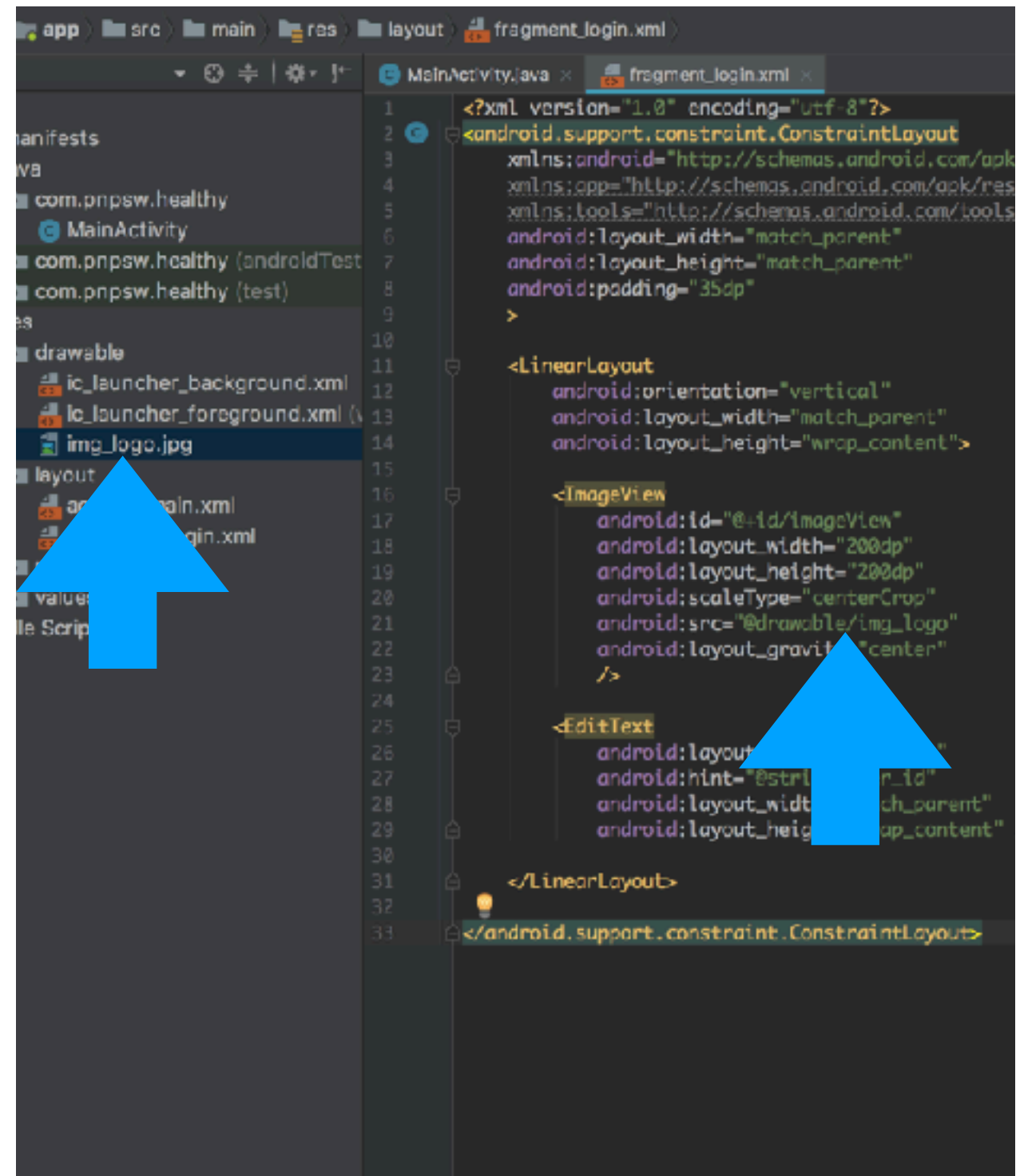
แล้วตามด้วย name ที่เราตั้งไว้

วิธีการนำรูปเข้ามาใช้

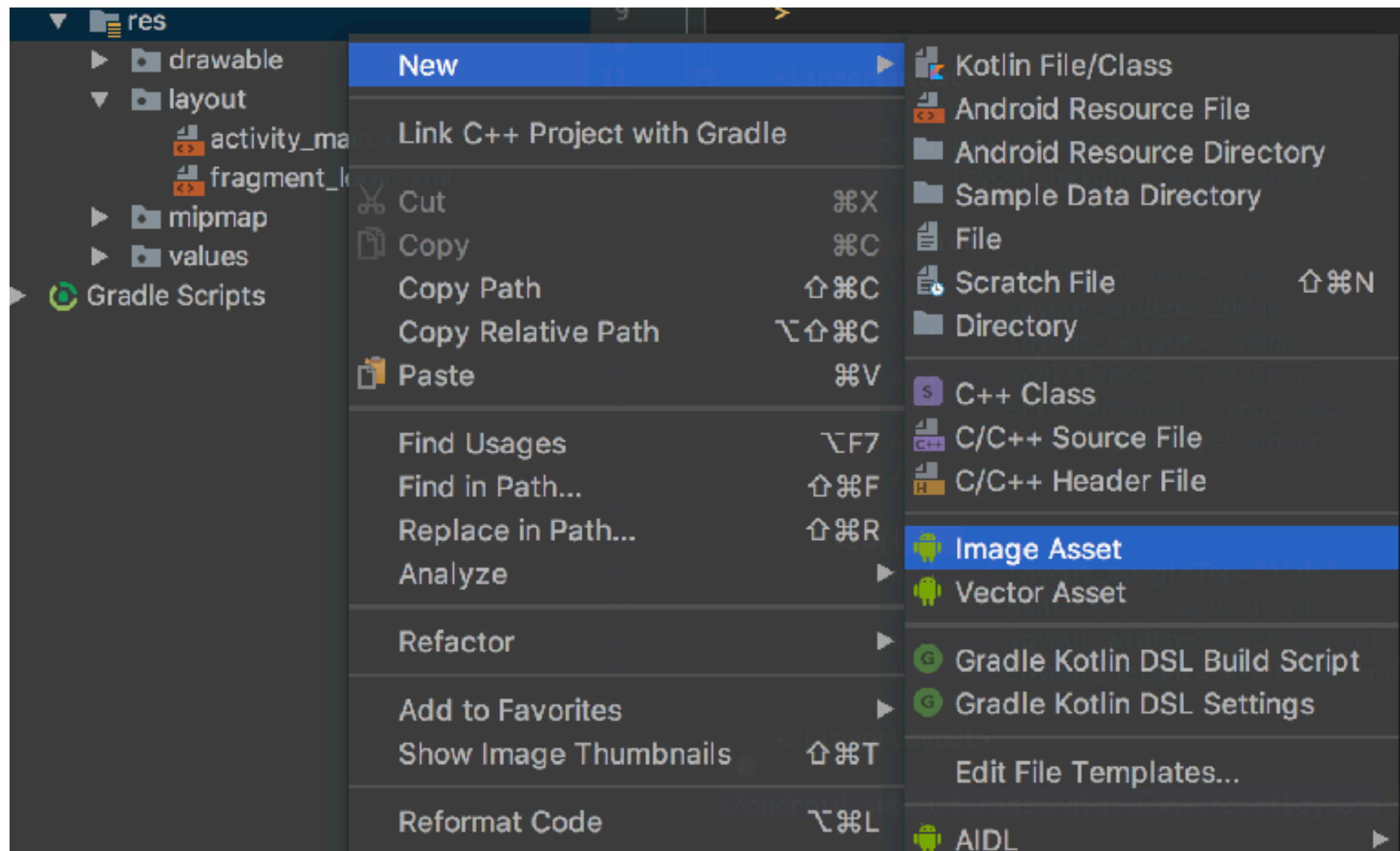
ใน android studio

Copy file

- Copy file ให้มาอยู่ใน folder res/drawable
- ตั้งชื่อยาวมากและห้ามมีตัวอักษรพิเศษ
- อย่าตั้งชื่อเป็นภาษาไทย
- ขนาดรูปไม่ควรใหญ่เกินไป
- สามารถอ้างถึง file ด้วยชื่อ

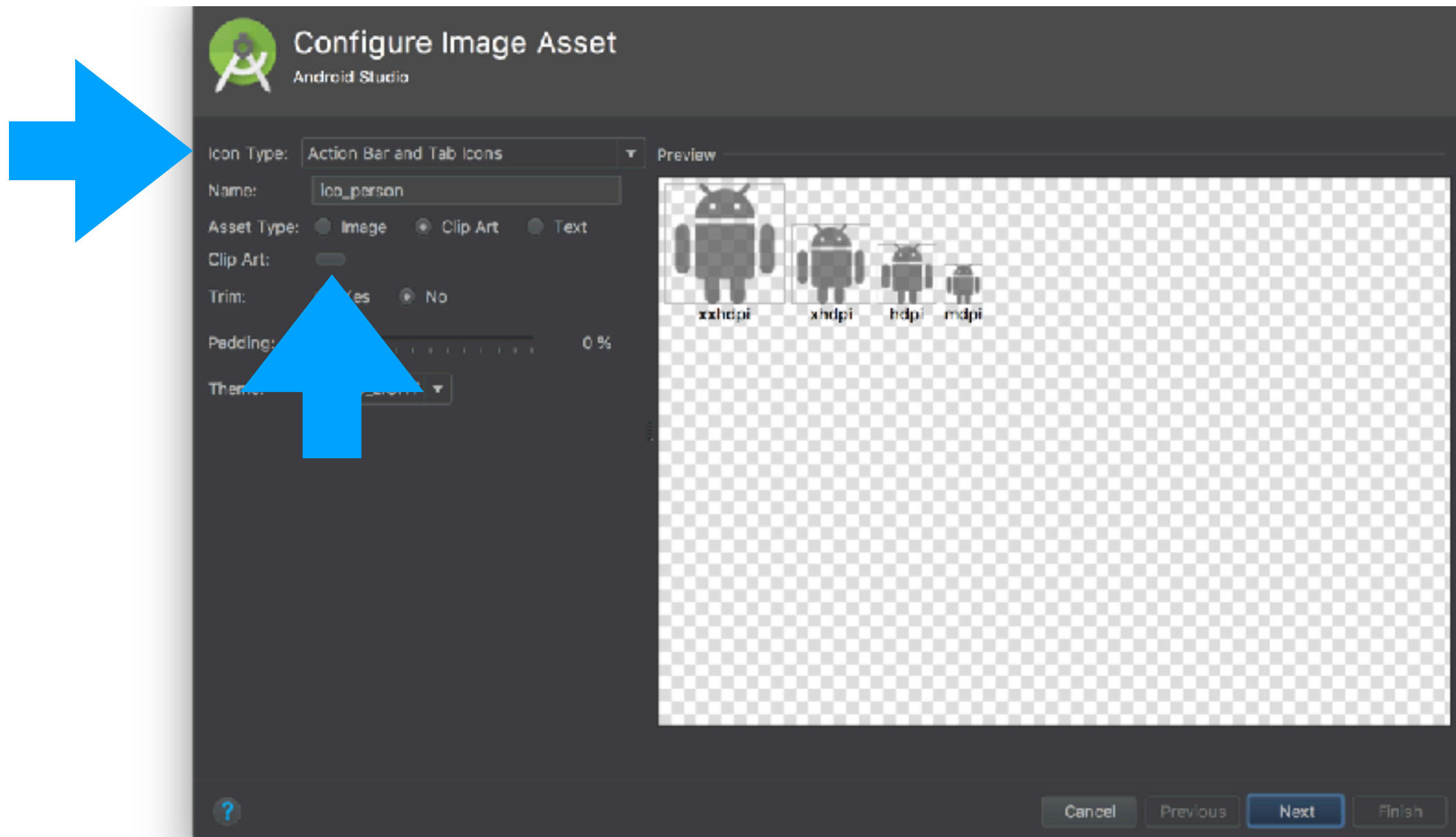


Create Image Asset



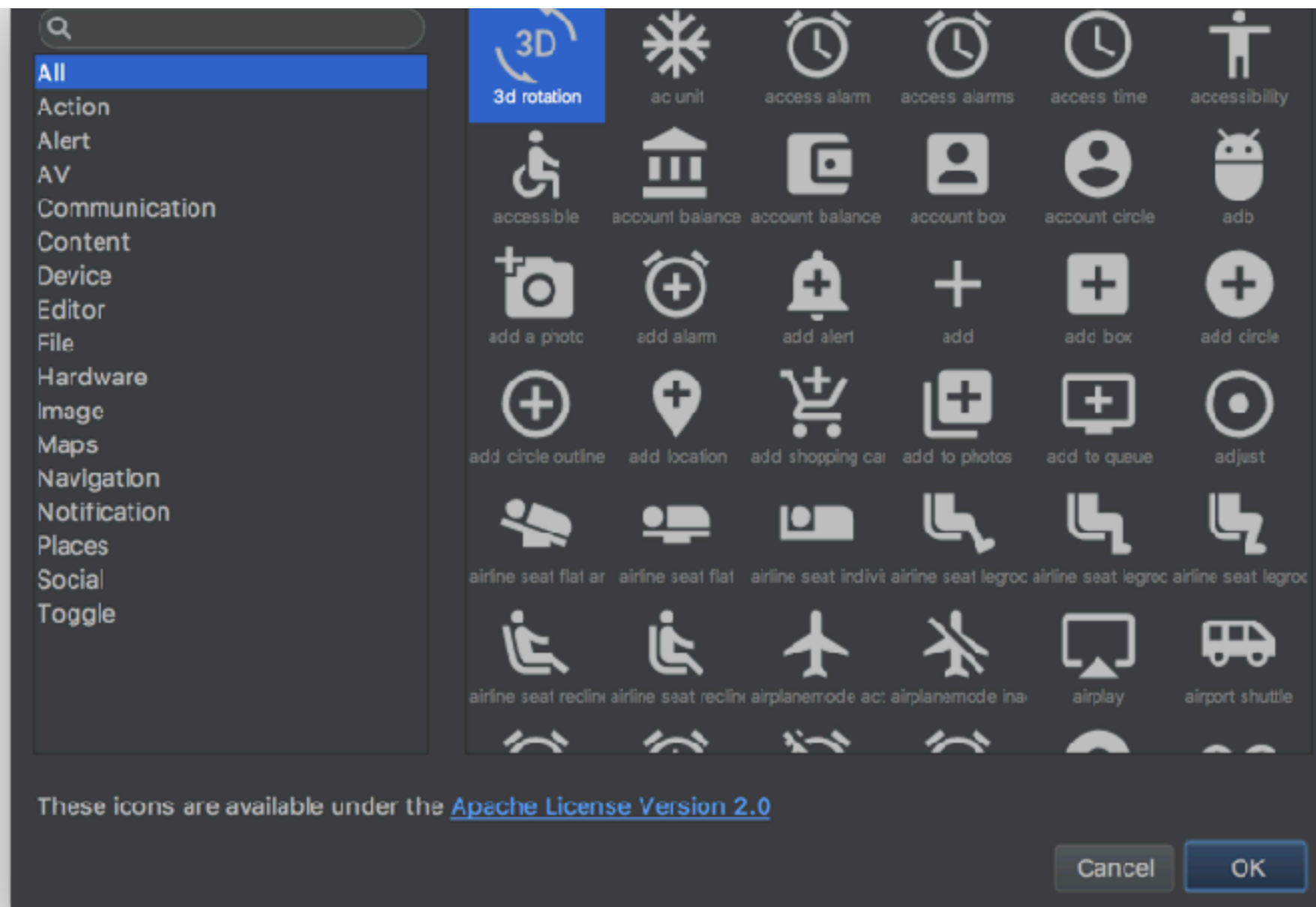
คลิกขวาที่ layout เลือก

New / Image Asset



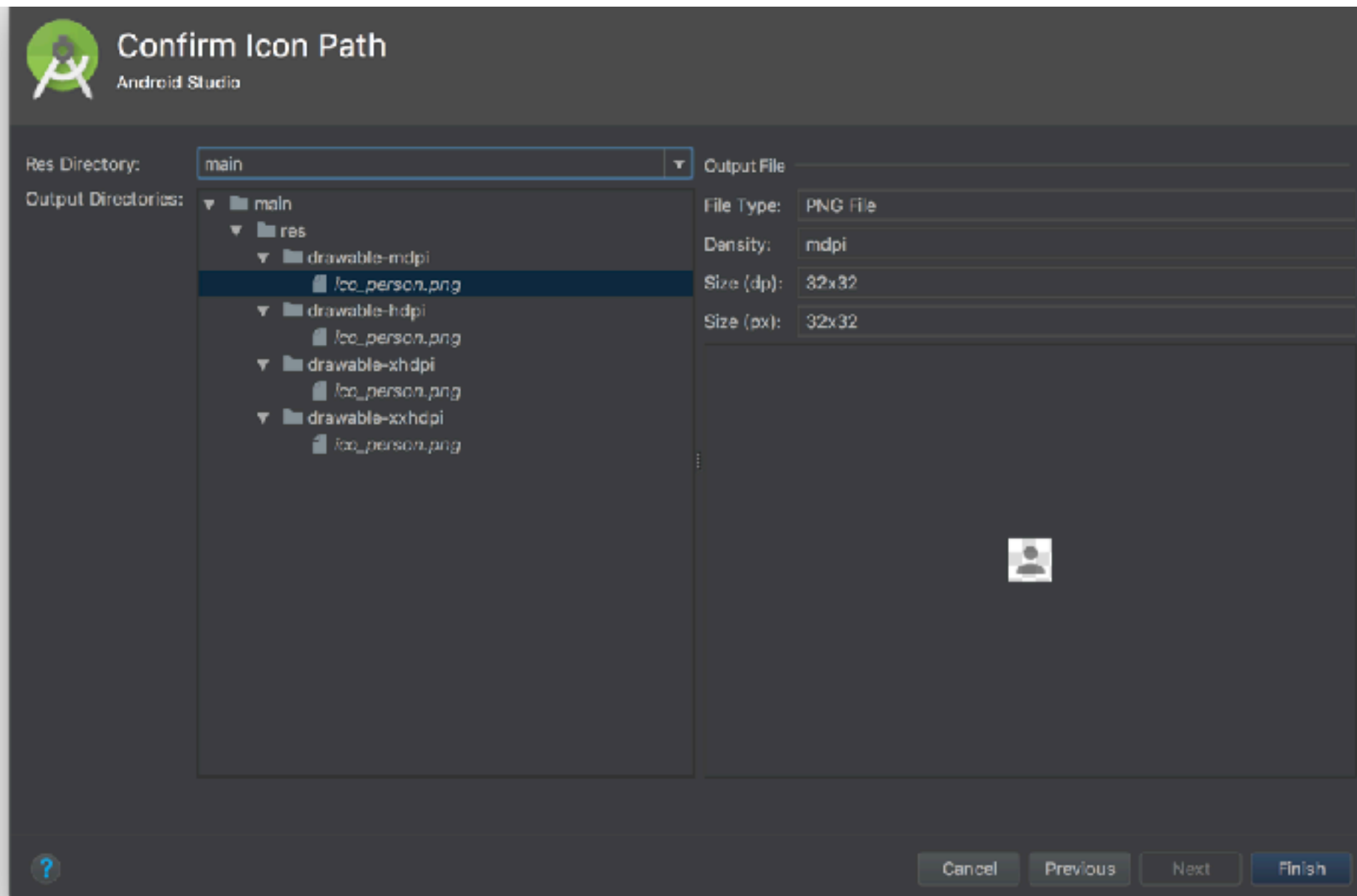
เลือก Icon Type เป็น action bar and tab icons

Asset Type เป็น Clip Arts เสร็จแล้วกดปุ่ม Clip Art



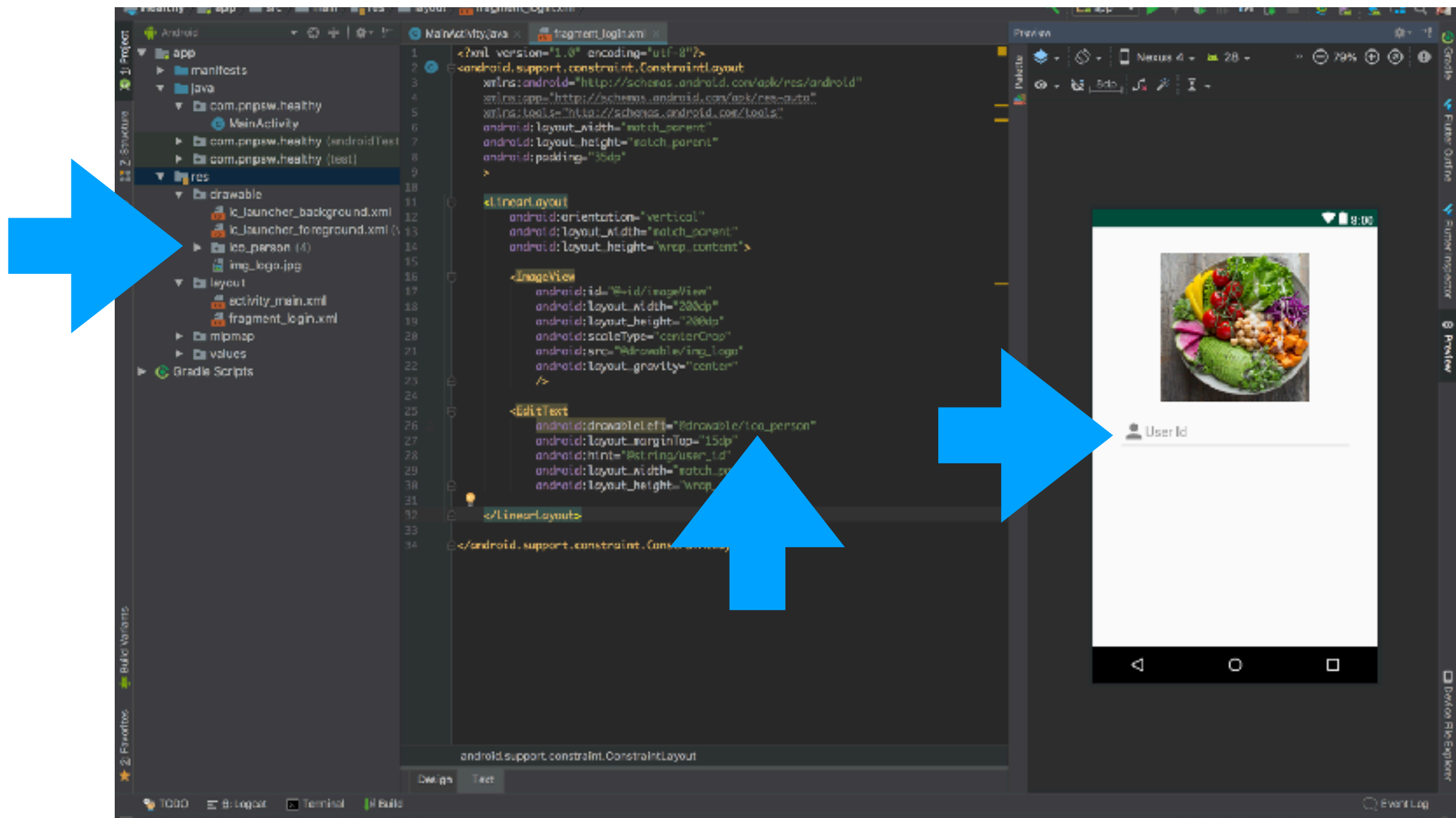
ค้นหา icon ที่ต้องการ

เสร็จแล้วกดปุ่ม OK



ให้ confirm path ที่เก็บรูป

แนะนำว่าให้ใช้ default



เราสามารถอ้างอิงถึงรูปด้วยคำสั่ง

@drawable/ico_person

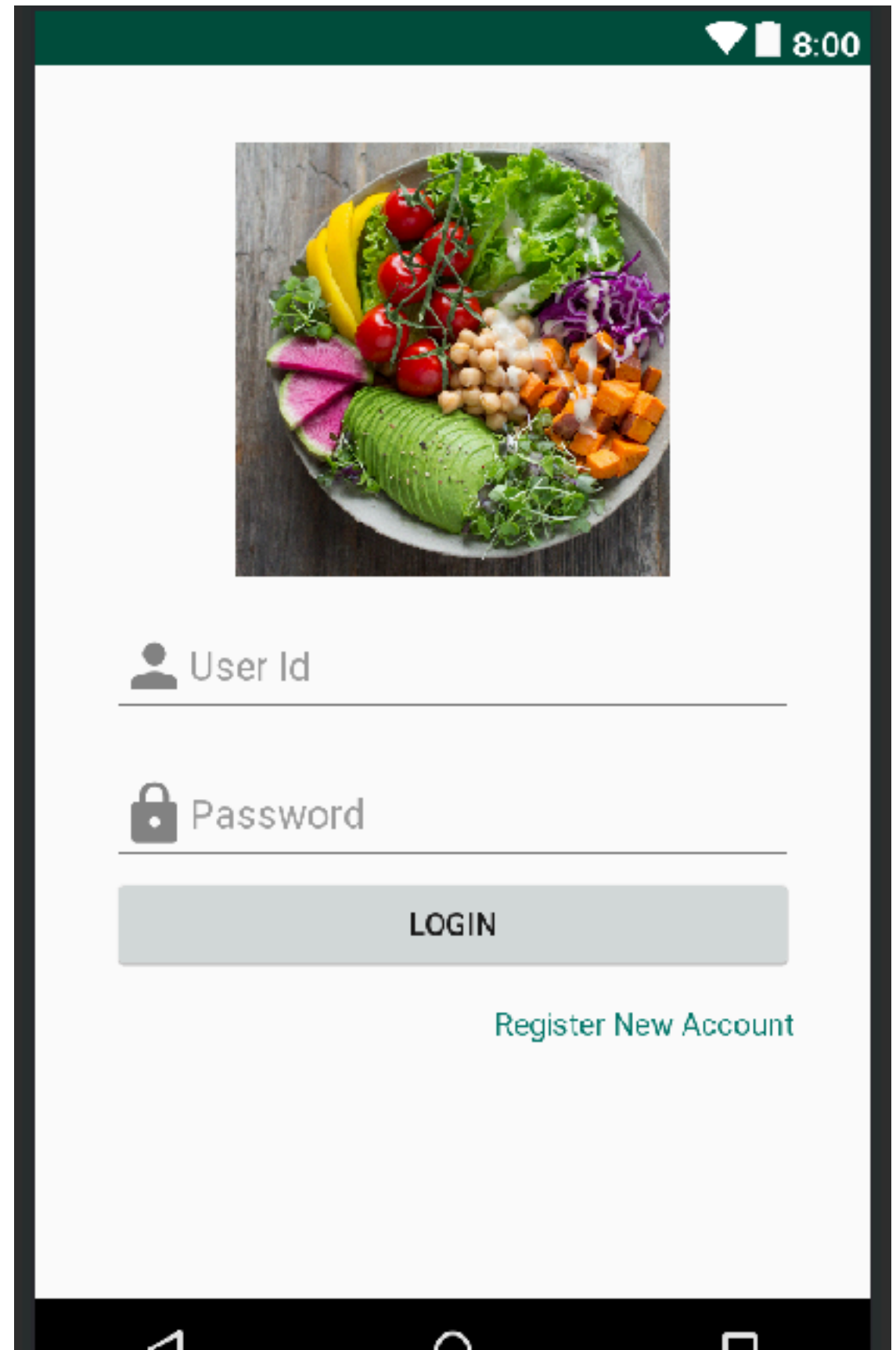
Workshop #1

Health App

Login

fragment_login

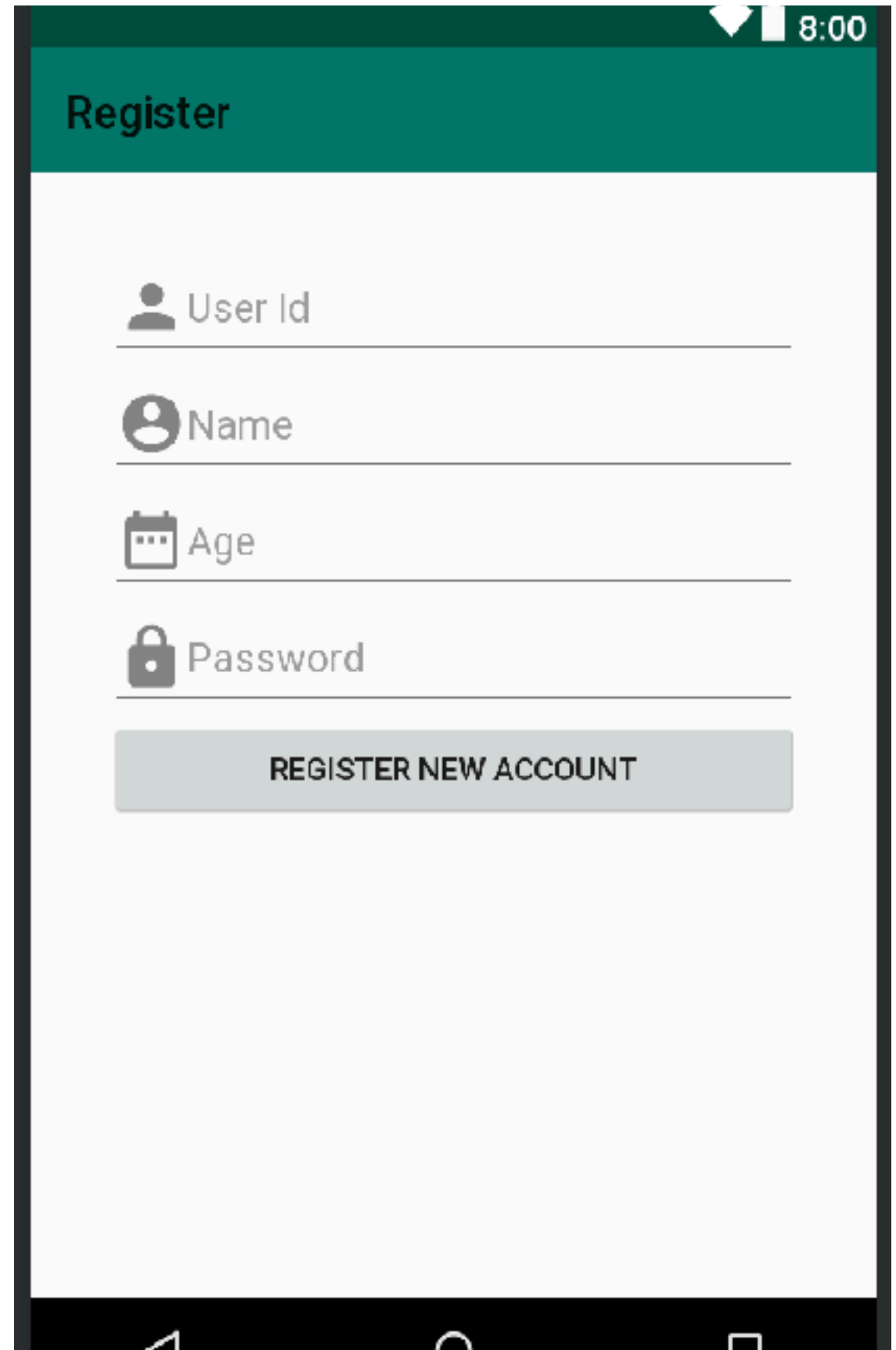
เอาไว้เพื่อให้ user ใส่ user id,
password ของตัวเอง
ในกรณีไม่มี ให้กดปุ่ม Register
New Account



Register

fragment_register

เอาไว้เพื่อให้ผู้ใช้งานกรอกข้อมูล
เพื่อทำการลงทะเบียนสมาชิกใหม่

A mobile application interface for a registration form. The title bar is dark green with the word "Register" in white. The form has four input fields, each with a grey icon on the left: a person icon for "User Id", a person icon for "Name", a calendar icon for "Age", and a lock icon for "Password". Each field has a horizontal line below the text. At the bottom of the form is a grey button with the text "REGISTER NEW ACCOUNT" in black. The top status bar shows a signal strength indicator, a battery icon, and the time "8:00". The bottom navigation bar is black with three white icons: a triangle, a circle, and a square.

Register

User Id

Name

Age

Password

REGISTER NEW ACCOUNT

BMI

fragment_bmi

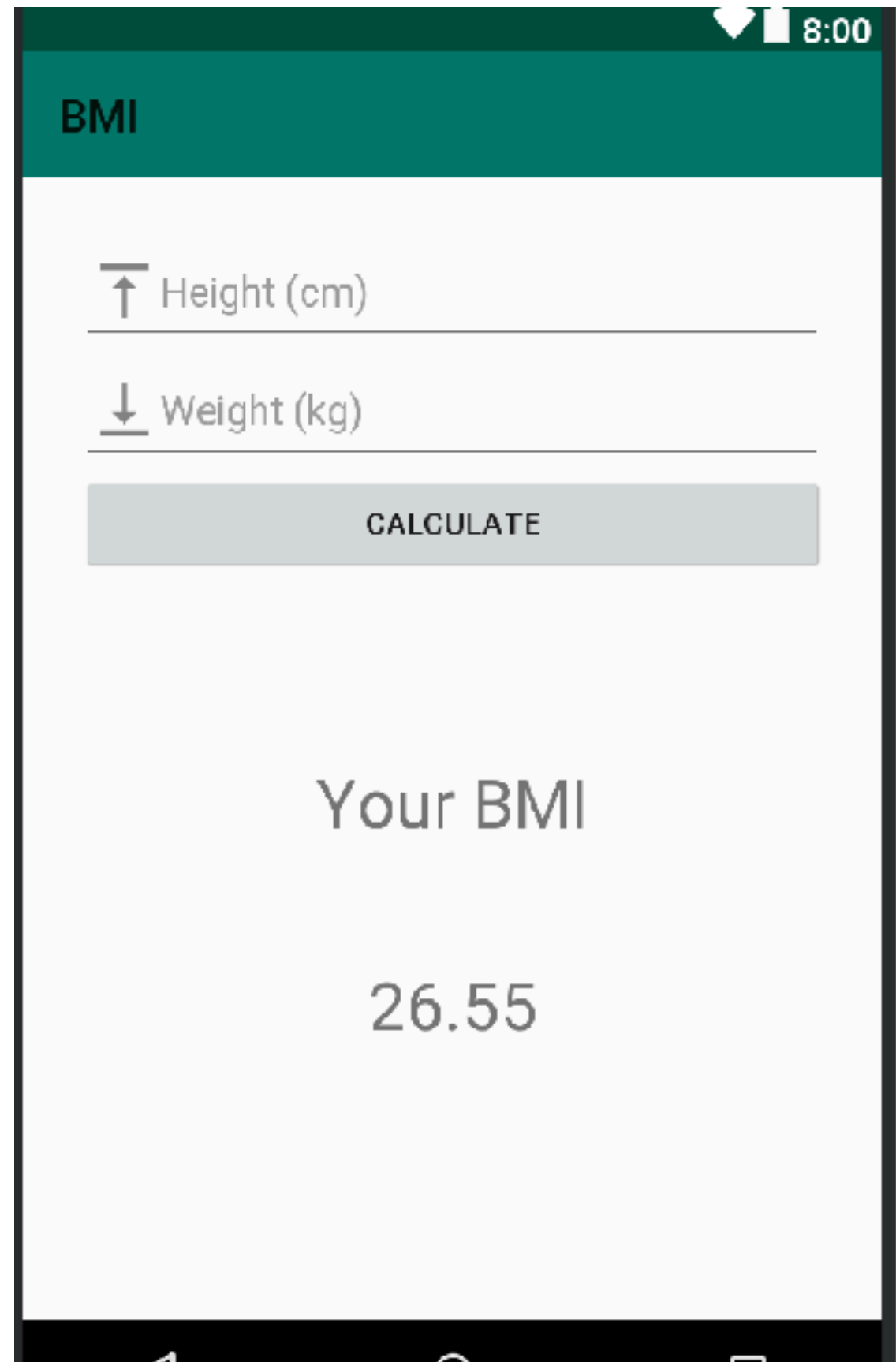
เมื่อใส่ส่วนสูงและน้ำหนักให้

คำนวณผลดังนี้

$\text{weight(kg)} / \text{height(m} \times \text{m)}$

เช่น น้ำหนัก 68 สูง 168

$\text{Bmi} = 68 / (1.68 \times 1.68)$



The screenshot shows a mobile application interface for calculating BMI. At the top, there is a dark green header with the text "BMI". Below the header, there are two input fields: "Height (cm)" with an upward arrow icon and "Weight (kg)" with a downward arrow icon. A grey "CALCULATE" button is positioned below these fields. The result is displayed in the center of the screen, showing "Your BMI" followed by the value "26.55". The status bar at the top right indicates the time is 8:00.