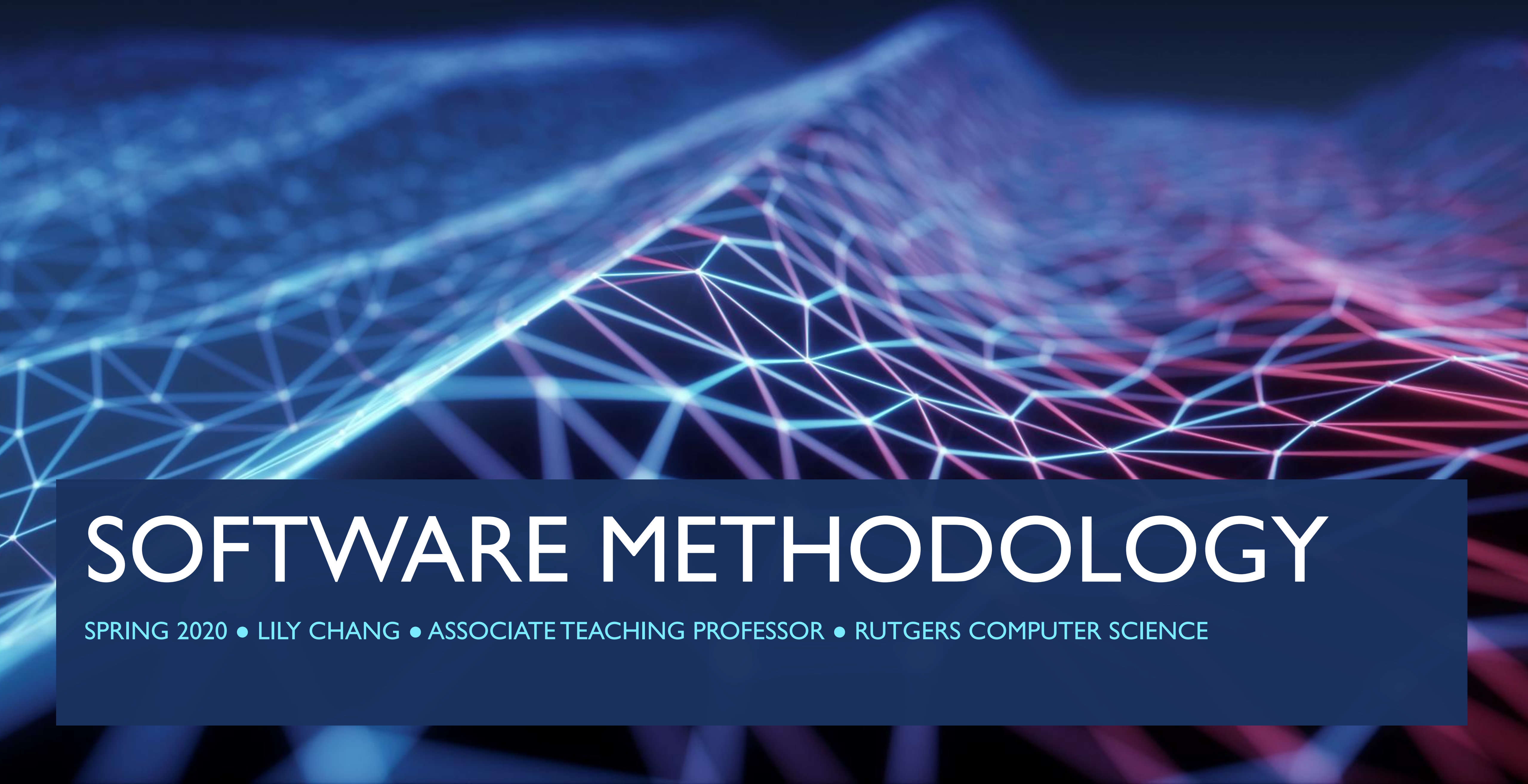


SOFTWARE METHODOLOGY

SPRING 2020 • LILY CHANG • ASSOCIATE TEACHING PROFESSOR • RUTGERS COMPUTER SCIENCE

MORE ON ANDROID APP GUI COMPONENTS

ANDROID CLASSES RELEVANT TO PROGRAM 5

- Activity
- Intent
- ConstraintLayout
- TextView
- Text/Plain Text (EditText)
- ImageView
- Buttons
  - Button
  - RadioGroup
  - RadioButton

ACTIVITY

- AndroidManifest.xml
  - Register the activities in your app
  - Define the Intent Filter
    - Explict intent
    - Implict intent
- Each activity has a .java file and a .xml file; for example,
  - MainActivity.java → activity_main.xml
  - SecondActivity.java → activity_second.xml
- The .xml file is where you create the UI components
  - Use the Android Layout Editor
  - Write the xml code directly

ADDING THE ACTIVITY IN THE ANDROIDMANIFEST.XML

SecondActivity.java

```
<activity android:name=".SecondActivity"
          android:label = "@string/activity2_name"
          android:parentActivityName=".MainActivity">
</activity>
```

Title of the activity
A constant String value

INTENT OBJECT

```
Intent intent = new Intent(this, SignInActivity.class);
startActivity(intent);
```

- The Intent object specifies either the exact activity you want to start or describes the type of action you want to perform (and the system selects the appropriate activity for you, which can even be from a different application).
- An Intent object can also carry small amounts of data to be used by the activity that is started.

INTENT CLASS/OBJECT

- The primary pieces of information in an intent are:
  - action – The general action to be performed, such as ACTION_VIEW, ACTION_EDIT, ACTION_MAIN, etc.
  - data – the data to operate on, such as a person record in the contacts database, expressed as a Uri.
- For example,

```
ACTION_VIEW content://contacts/people/1
```

  - Display information about the person whose identifier is "1".

INTENT EXTRA

- This is a **Bundle** of any additional information. This can be used to provide extended information to the component.
- For example, if we have a action to send an e-mail message, we could also include extra pieces of data here to supply a subject, body, etc.

EXTRA DATA – AN EXAMPLE

public Intent putExtra (String name, Bundle value)

- Name – String: the name of the extra data, with package prefix.
- Value – Bundle: the Bundle data value; this value may be null.

Could be String, int, double, or other data types

```
//in MainActivity
public static final String EXTRA_MESSAGE =
            "com.example.android.twoactivities.extra.MESSAGE";
…
String message = ….;
 …
intent.putExtra(EXTRA_MESSAGE, message);
```

Package prefix

```
//in SecondActivity
String message = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);
```

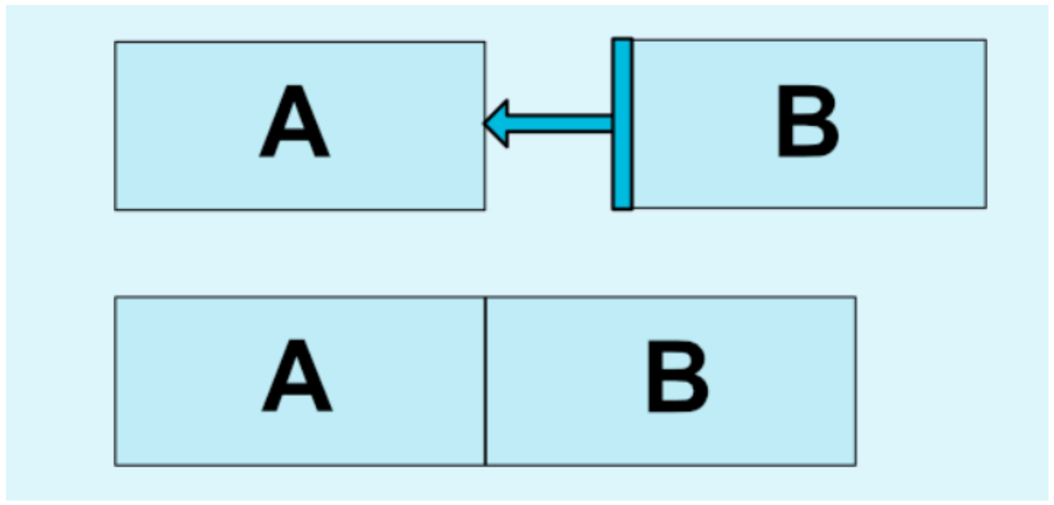
EXTRA DATA – ANOTHER EXAMPLE

```
//in MainAtivity
String message = "..." ;
intent.putExtra(Intent.EXTRA_TEXT, message);

//in SecondActivity
String message = intent.getStringExtra(intent.EXTRA_TEXT);
```

CONSTRAINT LAYOUT

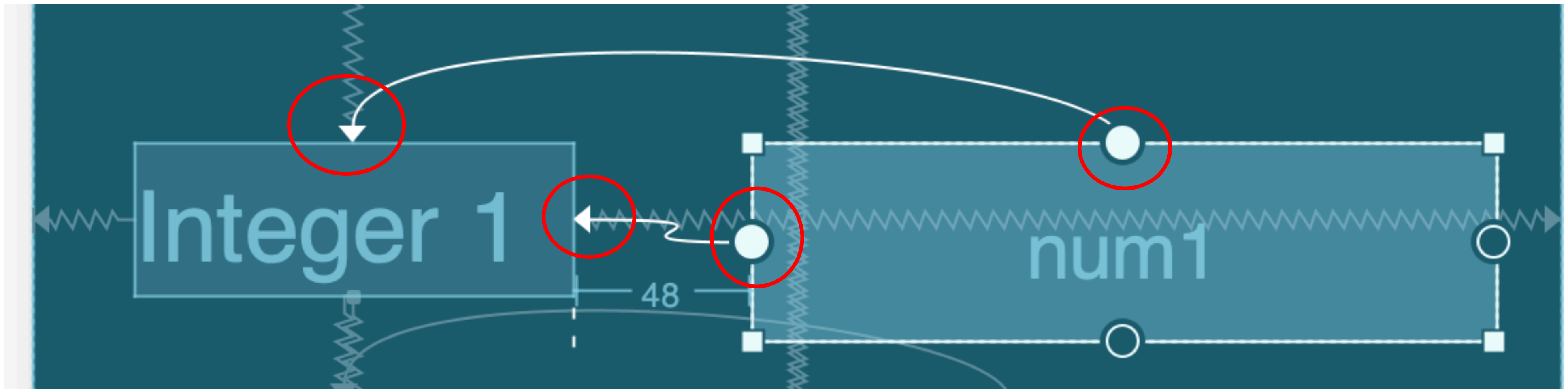
position button B to the right of button A



- Relative positioning is one of the basic building blocks of creating layouts in ConstraintLayout. Those constraints allow you to position a given widget relative to another one. You can constrain a widget on the horizontal and vertical axis:
  - Horizontal Axis: left, right, start and end sides
  - Vertical Axis: top, bottom sides and text baseline
  - The general concept is to constrain a given side of a widget to another side of any other widget.

CONSTRAINT LAYOUT

- Position num1 relative to the left of Integer 1 by setting the left and top constraints to the left and top constraints of Integer 1 with vertical constraint



TEXTVIEW

- A user interface element that displays text to the user

```
<TextView  android:id="@+id/text_view_id"
           android:layout_height="wrap_content"
           android:layout_width="wrap_content"
           android:text="@string/hello"
/>
```

1. match_constraint - as big as its parent (minus padding)
2. wrap_content - just big enough to enclose its content (plus padding).
3. exact values

EDITTEXT

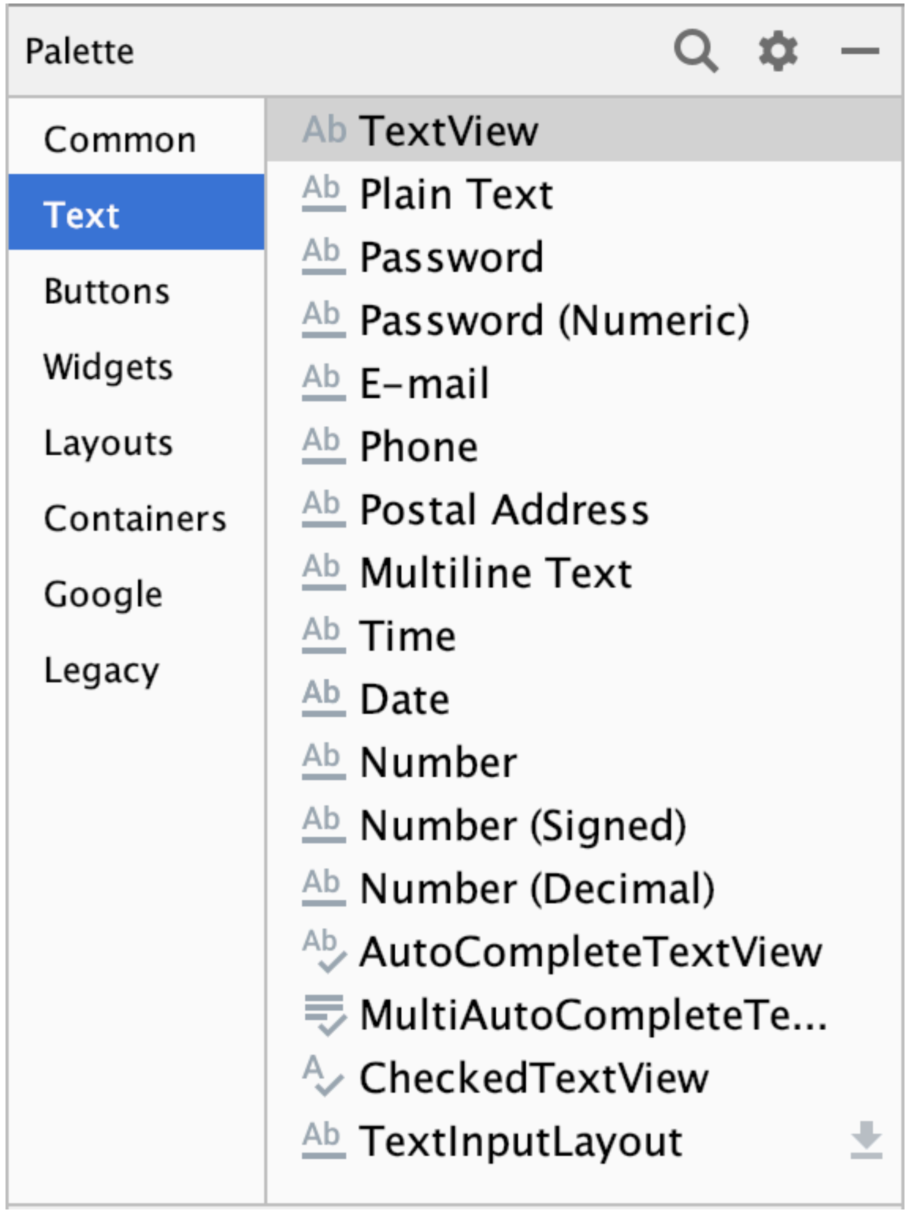
- A user interface element for entering and modifying text.
- When you define an edit text widget, you must specify the R.styleable.TextView_inputType attribute.
- For example, for plain text input set inputType to "text":

```
<EditText   android:id="@+id/plain_text_input"
             android:layout_height="wrap_content"
             android:layout_width="match_parent"
             android:inputType="text”
/>
```

- Choosing the input type configures the keyboard type that is shown, acceptable characters, and appearance of the edit text.
- For example, if you want to accept a secret number, like a unique pin or serial number, you can set inputType to "numericPassword”

EDITTEXT

- Layout editor groups different EditText views into "Text" in the Palette
- Could choose different views based on the input data type



IMAGEVIEW

- Displays image resources, for example Bitmap or Drawable resources. ImageView is also commonly used to apply tints to an image and handle image scaling.

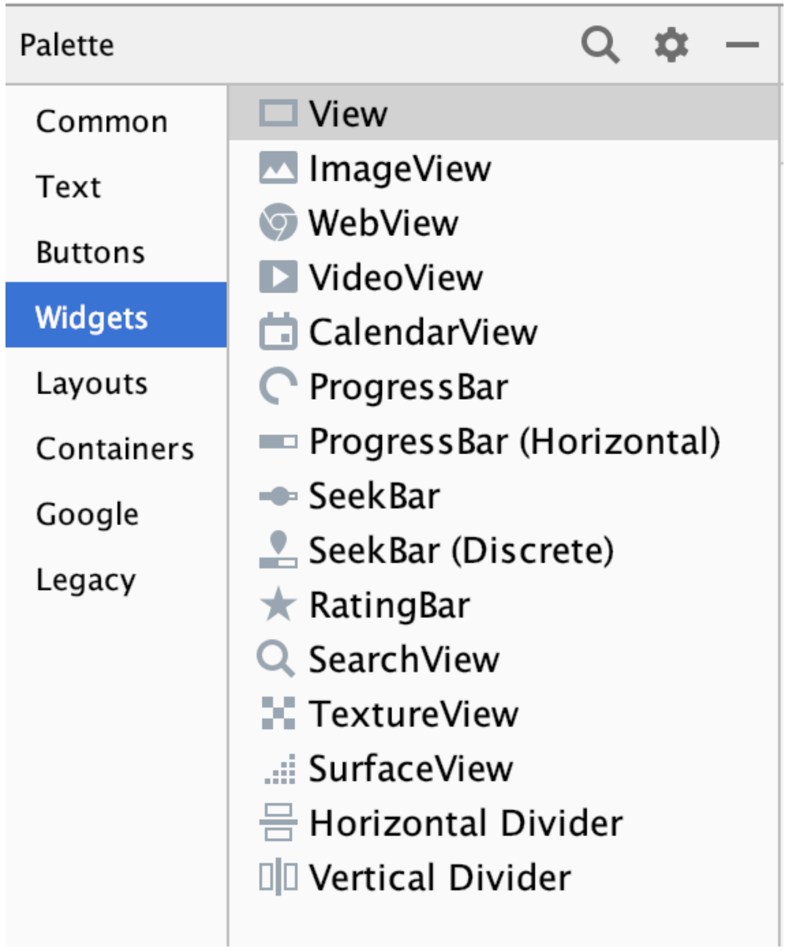
```
<ImageView android:layout_width="wrap_content"
           android:layout_height="wrap_content"
           android:src="@drawable/my_image"
           android:contentDescription="@string/my_image_description"
/>
```

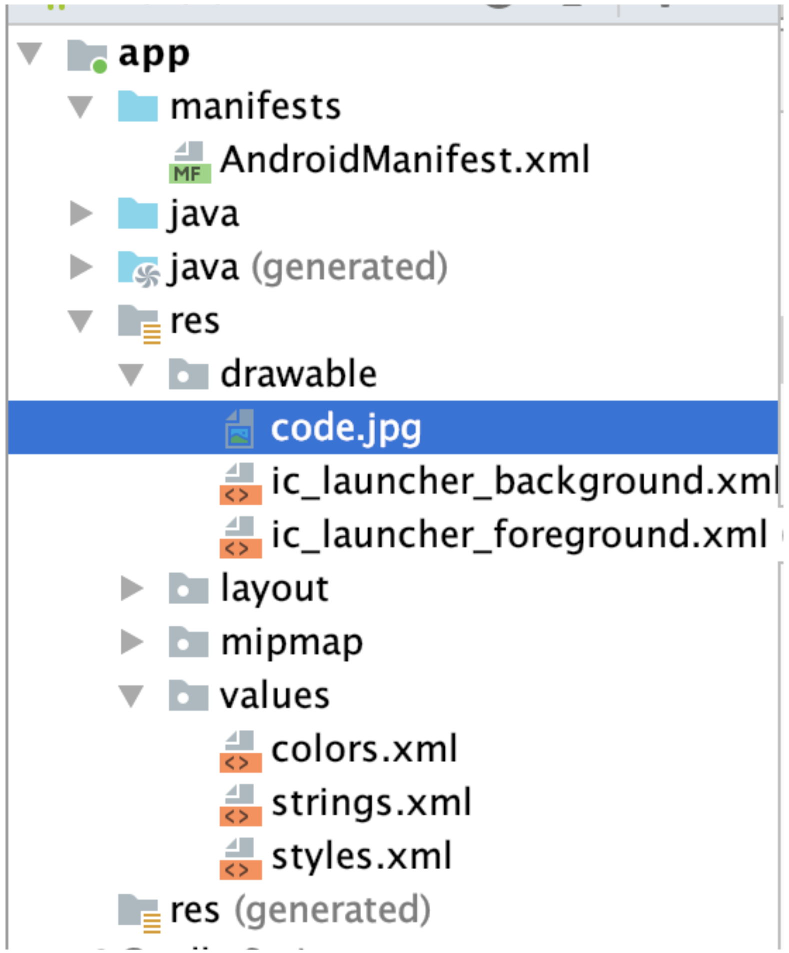
in res/drawable

- setImageResource method sets a drawable as the content of this ImageView

```
public void setImageResource (int resId)
```

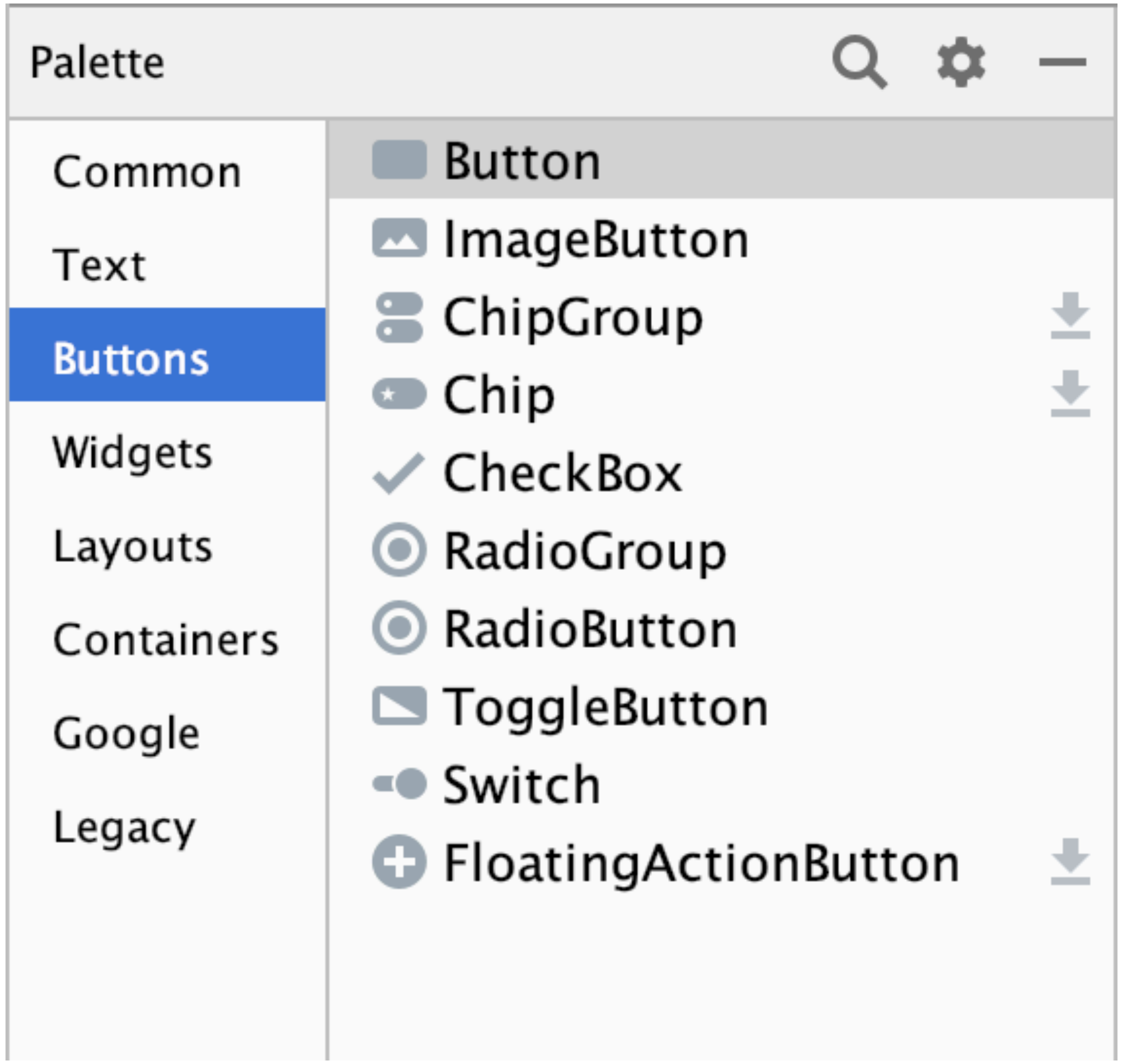
IMAGEVIEW


Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Google
Legacy
View
ImageView
WebView
VideoView
CalendarView
ProgressBar
ProgressBar (Horizontal)
SeekBar
SeekBar (Discrete)
RatingBar
SearchView
TextureView
SurfaceView
Horizontal Divider
Vertical Divider


app
manifests
AndroidManifest.xml
java
java (generated)
res
drawable
code.jpg
ic_launcher_background.xml
ic_launcher_foreground.xml
layout
mipmap
values
colors.xml
strings.xml
styles.xml
res (generated)

BUTTONS

- RadioGroup
  - This class is used to create a multiple-exclusion scope for a set of radio buttons. Checking one radio button that belongs to a radio group unchecks any previously checked radio button within the same group. Intially, all of the radio buttons are unchecked.
- RadioButton
  - A radio button is a two-states button that can be either checked or unchecked.
  - When the radio button is unchecked, the user can press or click it to check it. However, contrary to a CheckBox, a radio button cannot be unchecked by the user once checked.
  - Radio buttons are normally used together in a RadioGroup. When several radio buttons live inside a radio group, checking one radio button unchecks all the others.


Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Google
Legacy
Button
ImageButton
ChipGroup
Chip
CheckBox
RadioGroup
RadioButton
ToggleButton
Switch
FloatingActionButton

BUTTONS

THANK YOU