Royal University of Bhutan

Unit II: Android Views, Layouts and Resources

CTE308- AS2025

Tutor: Pema Galey

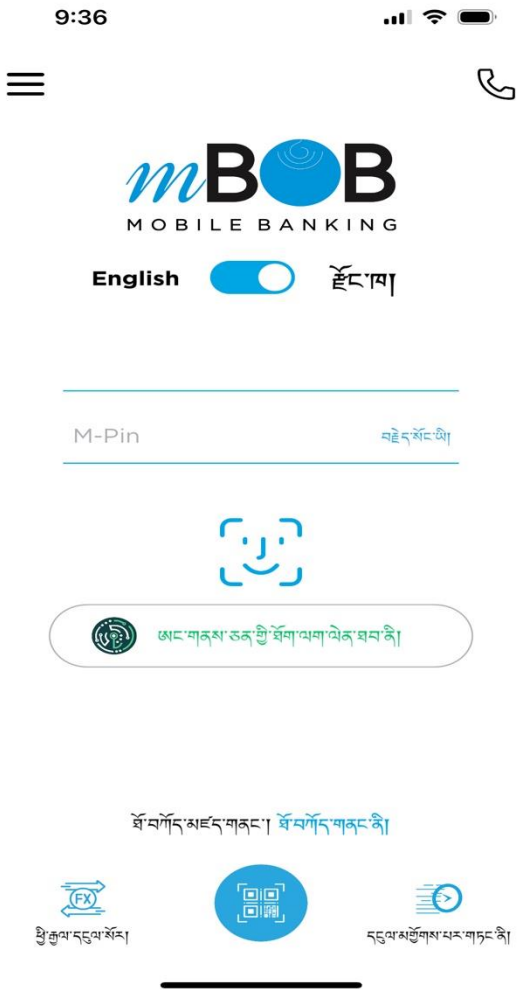
#17682761

Outline

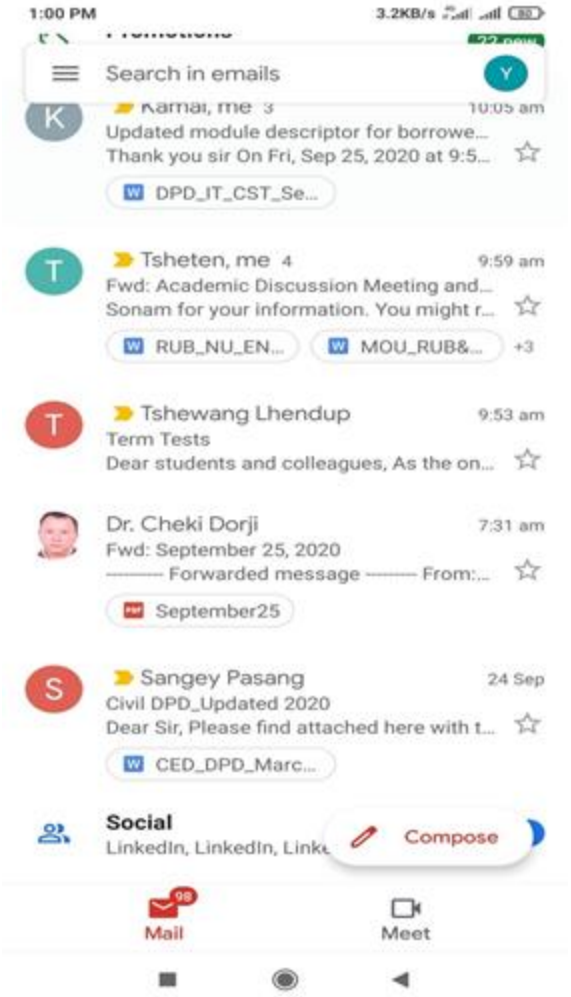
- Views, view groups, and view hierarchy
- Layouts in XML and Java code
- Event Handling
- Resources
- Screen Measurements

Views

Everything You see is a view…



- TextView
- EditText
- Buttons
- Images
- Icons
- Menu
- Etc..



What Is View?

- Android's basic user interface building blocks

UI Building Block	Control Class
Display Text	TextView, EditText
Buttons	Button class, menus, other control
Scrollable	ScrollView, RecyclerView
Show images	ImageView

Views have properties

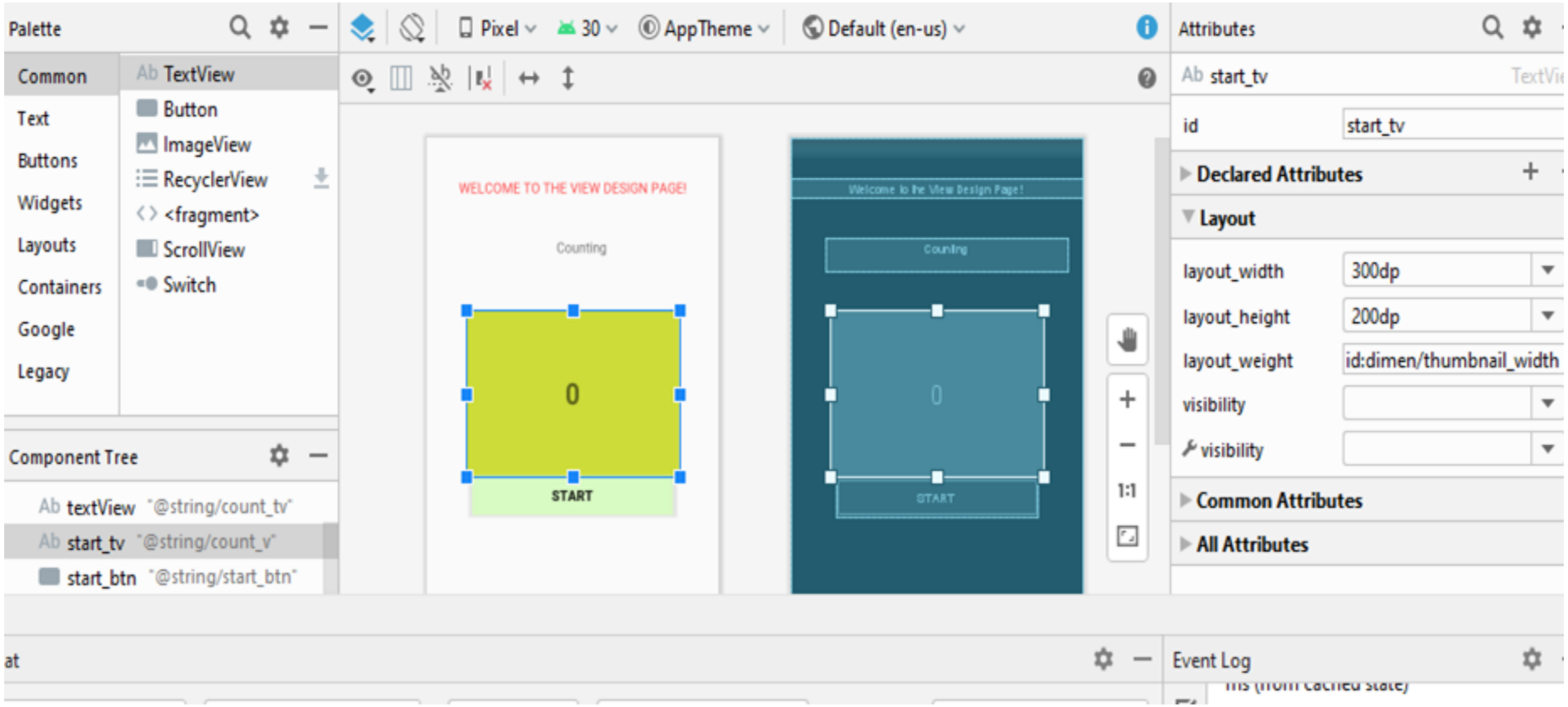
- Have properties (e.g., color, dimensions, positioning)
- May have focus (e.g., selected to receive user input)
- May be interactive (respond to user clicks)
- May be visible or not
- Have relationships to other views

VIEWS: Create and Layout

Creating and Laying out Views

- Graphically within Android Studio (Design)
- XML Files (XML Code/Text)
- Programmatically (Java\Kotlin Code)

Views defined in Visual Editor



Visual Representation in XML files.

Views defined in XML Code

activity_main.xml | MainActivity.java

```
38      <TextView
39          android:id="@+id/start_tv"
40          android:layout_width="300dp"
41          android:layout_height="200dp"
42          android:layout_gravity="center"
43          android:layout_weight="@android:dimen/thumbnail_width"
44          android:background="#CDDC39"
45          android:gravity="center"
46          android:shadowColor="#FFC107"
47          android:text="count_view"
48          android:textAlignment="center"
49          android:textSize="36sp"
50          android:textStyle="bold"
51          tools:text="0" />
52
```

Views Properties in XML

android:<property_name>="<property_value>"
Example: android:layout_width="match_parent"

android:<property_name>="@<resource_type>/resource_id"
Example: android:text="@string/button_label_next"

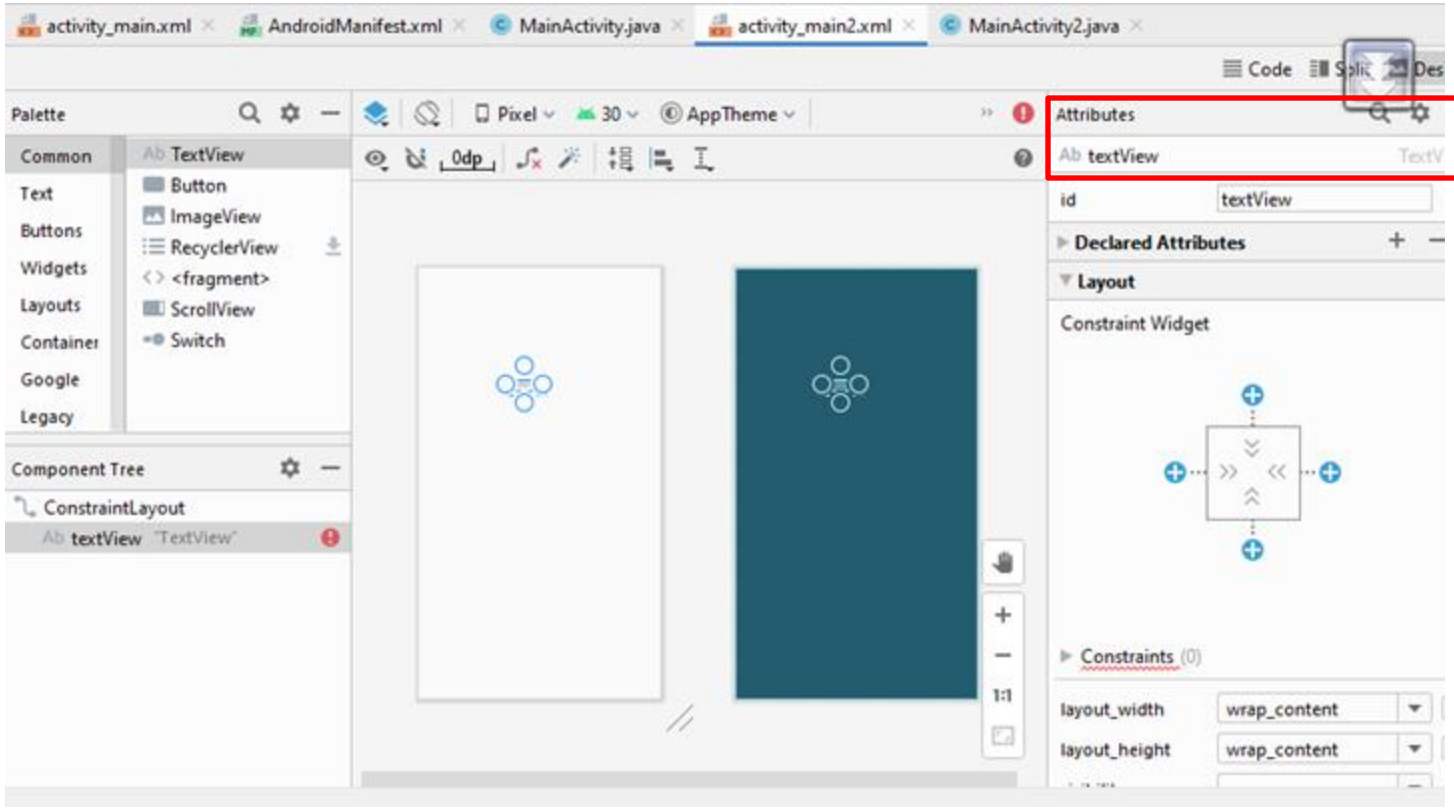
android:<property_name>="@+id/view_id"
Example: android:id="@+id/show_count"

Create View in Java Code (in Activity)

```
import android.widget.*;

myText = new TextView(context: this);
myText.setText("Text Message");
```

Displaying View (XML + Java)




```
TextView textView = (TextView) findViewById(R.id.textView);
textView.setText("Message display from TextView");
```

What is Context?

- Context is an interface to global information about an application environment
- Get the context:
```
Context context=getApplicationContext();
```
- An activity is its own context:
```
TextView myText = new TextView(context: this);
```

Custom Views

- Different types of views (over 100!) available from the Android system, all children of the **View** class
- If necessary, create custom views by subclassing existing views or the View class

ViewGroup & View Hierarchy

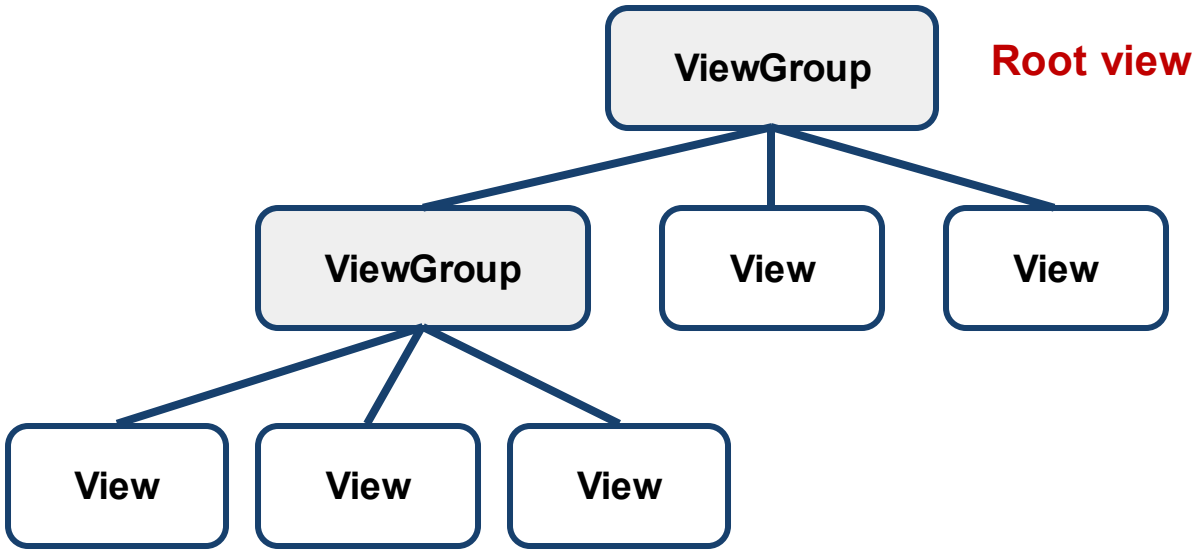
ViewGroup Views

A ViewGroup (parent) is a type of view that can contain other views (children)

ViewGroup is the base class for layouts and view containers

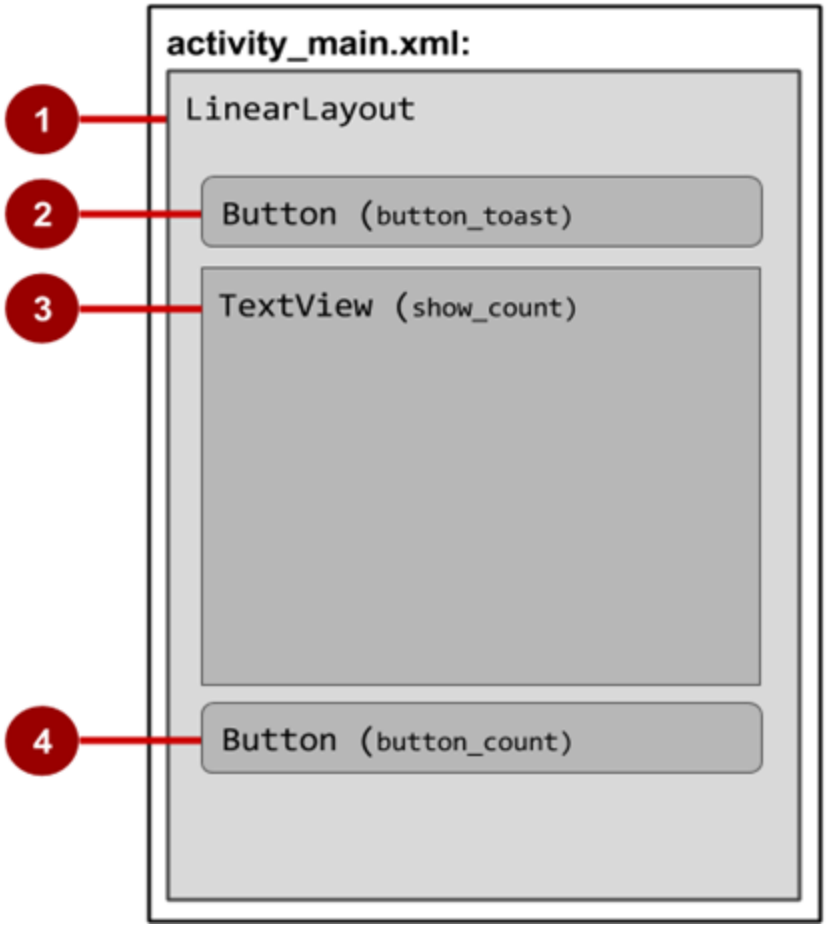
- ScrollView—scrollable view that contains one child view
- LinearLayout—arrange views in horizontal/vertical row
- RecyclerView—scrollable "list" of views or view groups

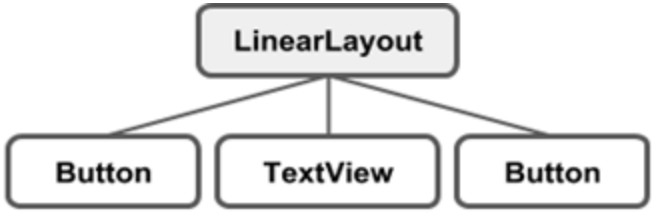
Hierarchy View Groups and Views

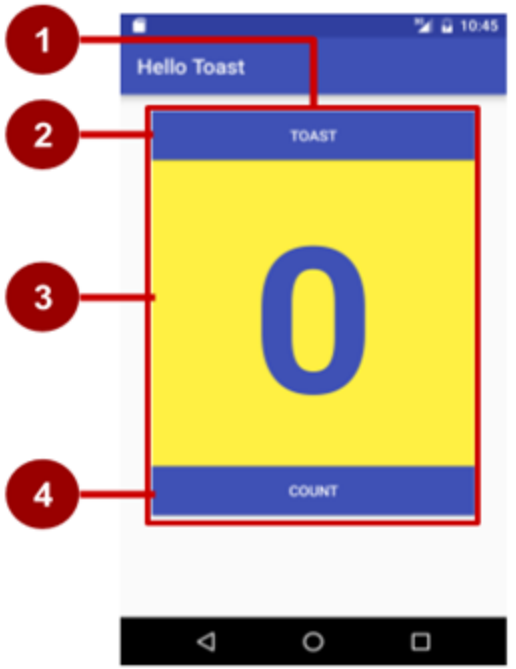


Root view is always a ViewGroup

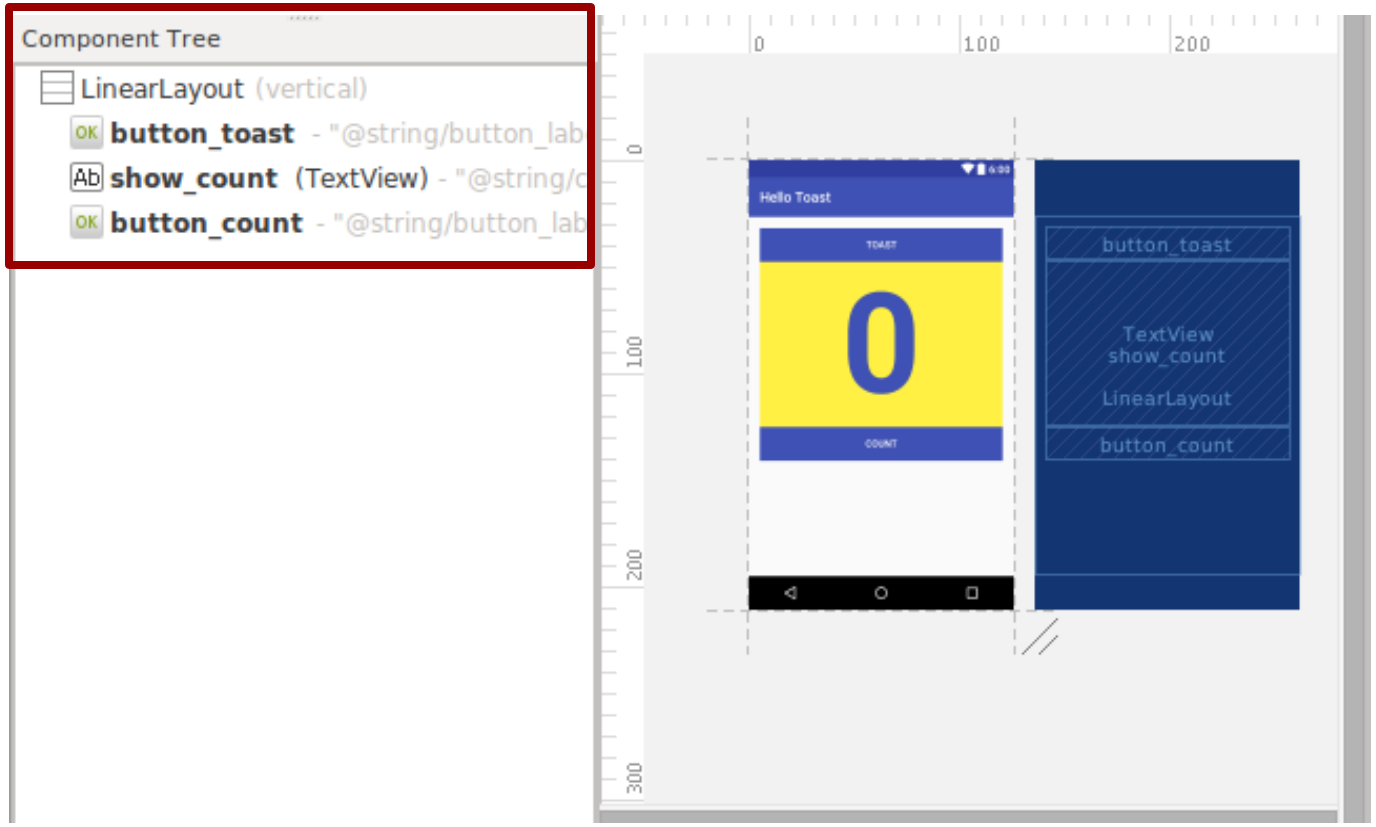
View Hierarchy and Screen Layout







View Hierarchy in Component Tree



Best Practices for View Hierarchy

- Arrangement of view hierarchy affects app performance
- Use smallest number of simplest views possible
- Keep the hierarchy flat—limit nesting of views and view groups

Layout

Layout View

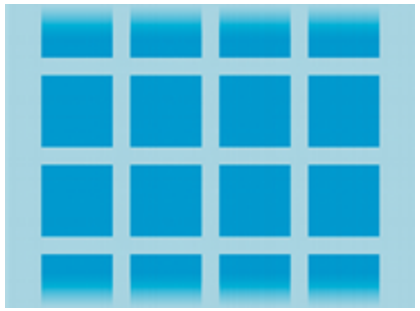
Layouts ?

- are specific types of view groups
- are subclasses of ViewGroup
- contain child views
- can be in a row, column, grid, table, absolute

Common Layouts

LinearLayout

RelativeLayout



GridLayout

TableLayout

Common Layout Classes

- **ConstraintLayout -** connect views with constraints
- **LinearLayout** - horizontal or vertical row
- **RelativeLayout** - child views relative to each other
- **TableLayout** - rows and columns
- **FrameLayout** - shows one child of a stack of children
- **GridView** - 2D scrollable grid

Class Hierarchy Vs Layout Hierarchy

- View class-hierarchy is standard object-oriented class inheritance
  - For example, Button is-a TextView is-a View is-a Object
  - Superclass-subclass relationship

- Layout hierarchy is how Views are visually arranged
  - For example, LinearLayout can contain Buttons arranged in a row
  - Parent-child relationship

Layout Created in XML

```
<LinearLayout
  android:orientation="vertical"
  android:layout_width="match_parent"
  android:layout_height="match_parent">
    <EditText
        ... />
    <Button
        ... />
</LinearLayout
```

Linear Layout Created in Java Activity Code

```
LinearLayout linearL = new LinearLayout(this);
linearL.setOrientation(LinearLayout.VERTICAL);
TextView myText = new TextView(this);
myText.setText("Display Message in TextView!");
linearL.addView(myText);
setContentView(linearL);
```

Event Handling

Events?

Something that happens

- In UI: Click, tap, drag
- Device: DetectedActivity such as walking, driving, tilting
- Events are "noticed" by the Android system

Events Handlers?

Methods that do something in response to a click

- A method, called an **event handler**, is triggered by a specific event and does something in response to the event

Handling Click event in XML and Java?

Attach handler to view in layout:

```
android:onClick=
"showToast"
```

Implement handler in activity:

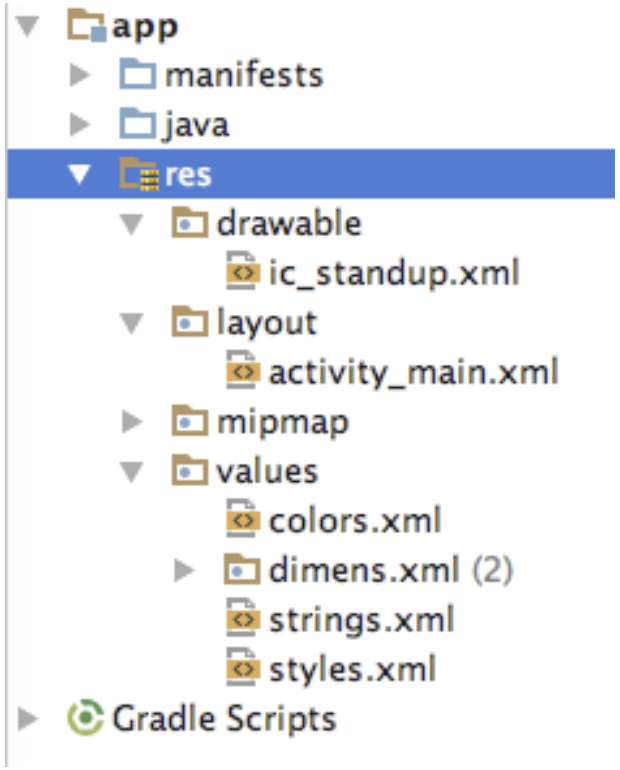
```
public void showToast(View view){
    String str="You have clicked button";
    Toast toast=Toast.makeText(this,str,
                    Toast.LENGTH_LONG);
    toast.show();
}
```

Resources

Resources

- Separate static data from code in your layouts.
- Strings, dimensions, images, menu text, colors, styles
- Useful for localization

Resources



resources and resource files stored in **res** folder

Refer to Resources in the Code

- Layout:
  `R.layout.activity_main`
  ```
  setContentView(R.layout.activity_main);
  ```
- View:
  `R.id.recyclerview`
  ```
  rv = (RecyclerView)
     findViewById(R.id.recyclerview);
  ```
- String:
  In Java: `R.string.title`
  In XML: `android:text="@string/title"`

Measurements

- Device Independent Pixels (dp) - for Views
- Scale Independent Pixels (sp) - for text

Don't use device-dependent units:

- Actual Pixels (px)
- Actual Measurement (in, mm)
- Points - typography 1/72 inch (pt)

Practical – Using Views, Layout, and Resources

- Views
- Layout
- Resources

Method:

1) Using Visual Editor (Design)
2) Using XML Code
3) Programmatically

Any Questions?

Class Work

- Design the UI/UX for a User Login form for a social media application (similar to Facebook, Instagram, X, etc.
- Requirements:
  - **Prototype Design**: Develop a detailed prototype showcasing the layout, interactions, and user flow for the login page. Ensure the design is intuitive and user-friendly.
  - **Layout of the Application**: Structure the layout to accommodate essential elements like username, password fields, login button, and "Forgot Password" and "Sign Up" links.
  - **Color Combination**: Choose a color scheme that aligns with the brand's identity, ensuring readability and a visually appealing interface.
  - **Responsiveness**: Design the login page to be responsive, ensuring optimal performance and appearance across various devices (smartphones, tablets, and desktops).
  - **Generate APK and AAB**: After finalizing the design and development, generate the APK and AAB files. Test using your physical devices.

Thank you!