FALL 2018 - NARGES NOROUZI

MOBILE APPLICATIONS

LECTURE 3

Resource: https://developers.google.com/training/android/

VIEWS, LAYOUTS, & RESOURCES

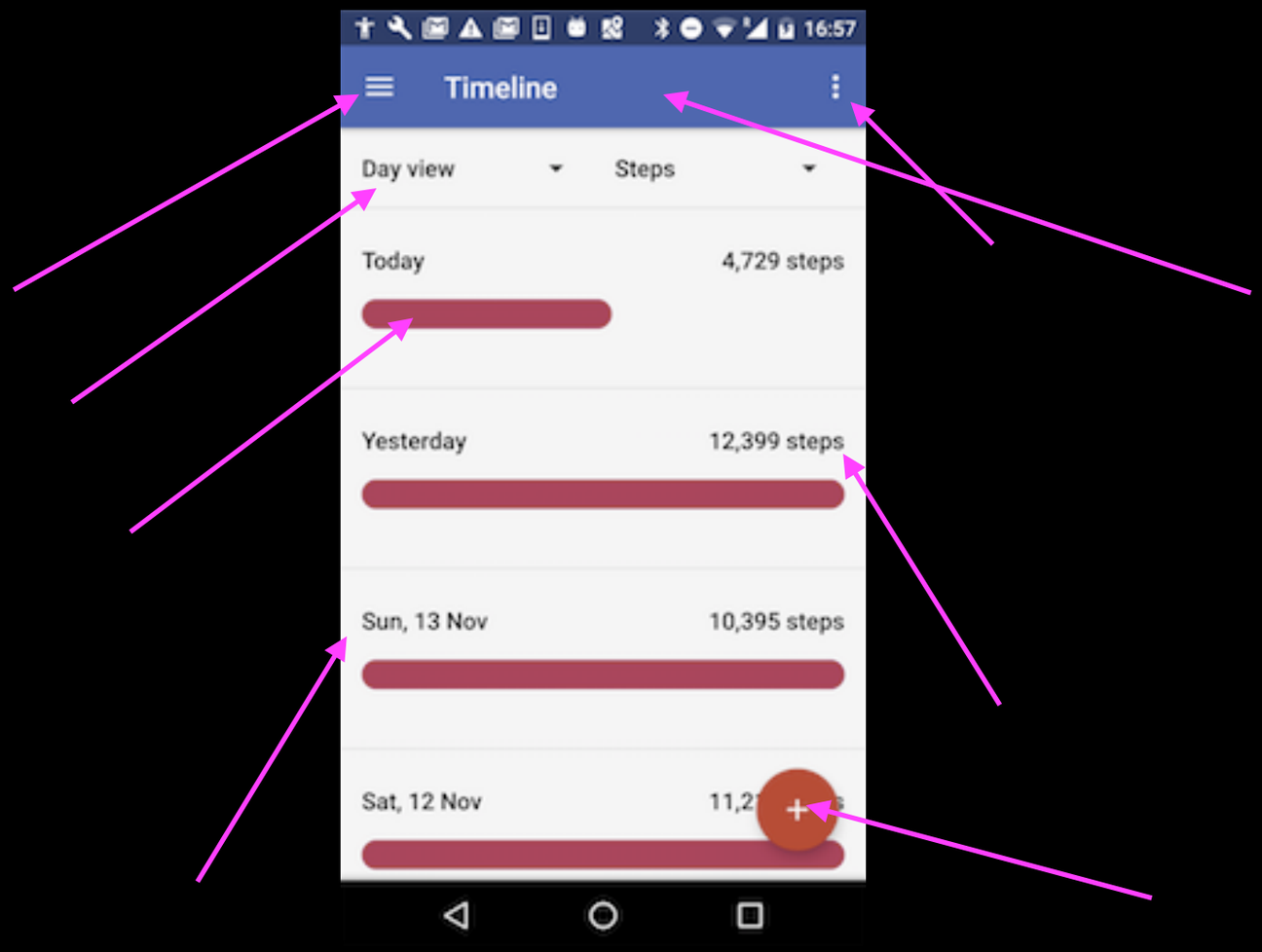
CONTENT

- Views, view groups, and view hierarchy
- Layouts in XML and Java code

VIEWS

EVERYTHING YOU SEE IS A VIEW

- If you look at your mobile device, every user interface element that you see is a View.



WHAT IS A VIEW

Views are Android's basic user interface building blocks.

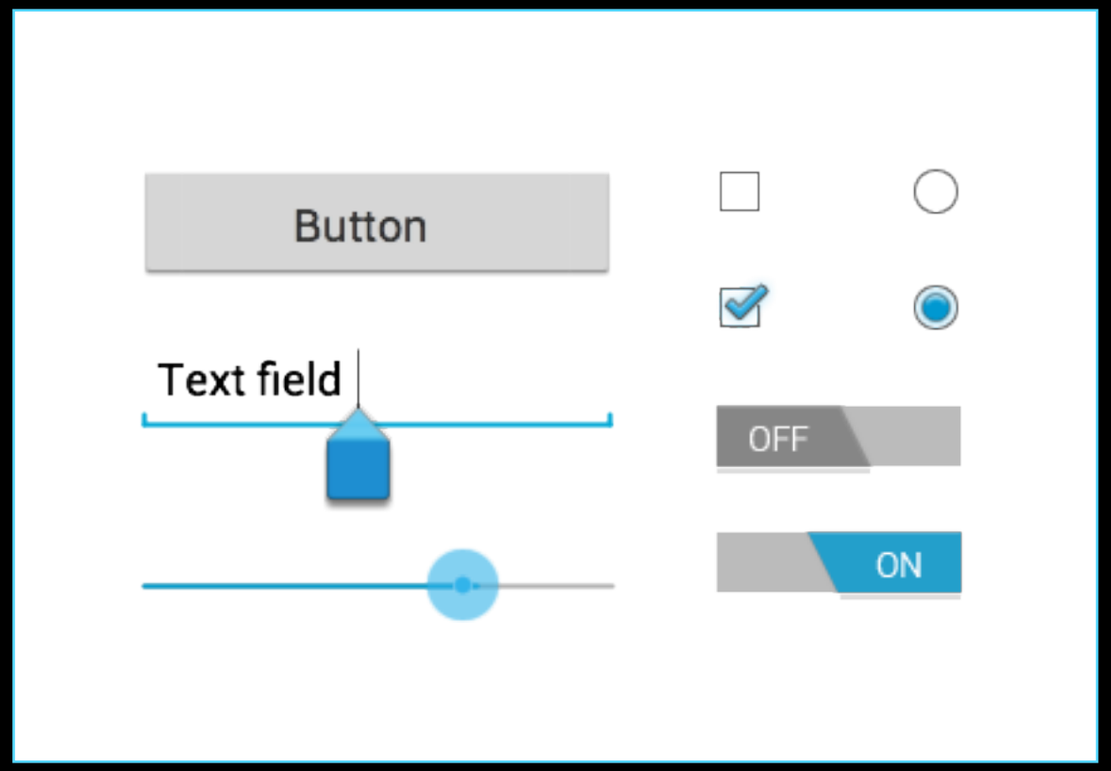
- display text (TextView class), edit text (EditText class)
- buttons (Button class), menus, other controls
- scrollable (ScrollView, RecyclerView)
- show images (ImageView)
- subclass of View class

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

EXAMPLES OF VIEWS

Button
EditText
SeekBar



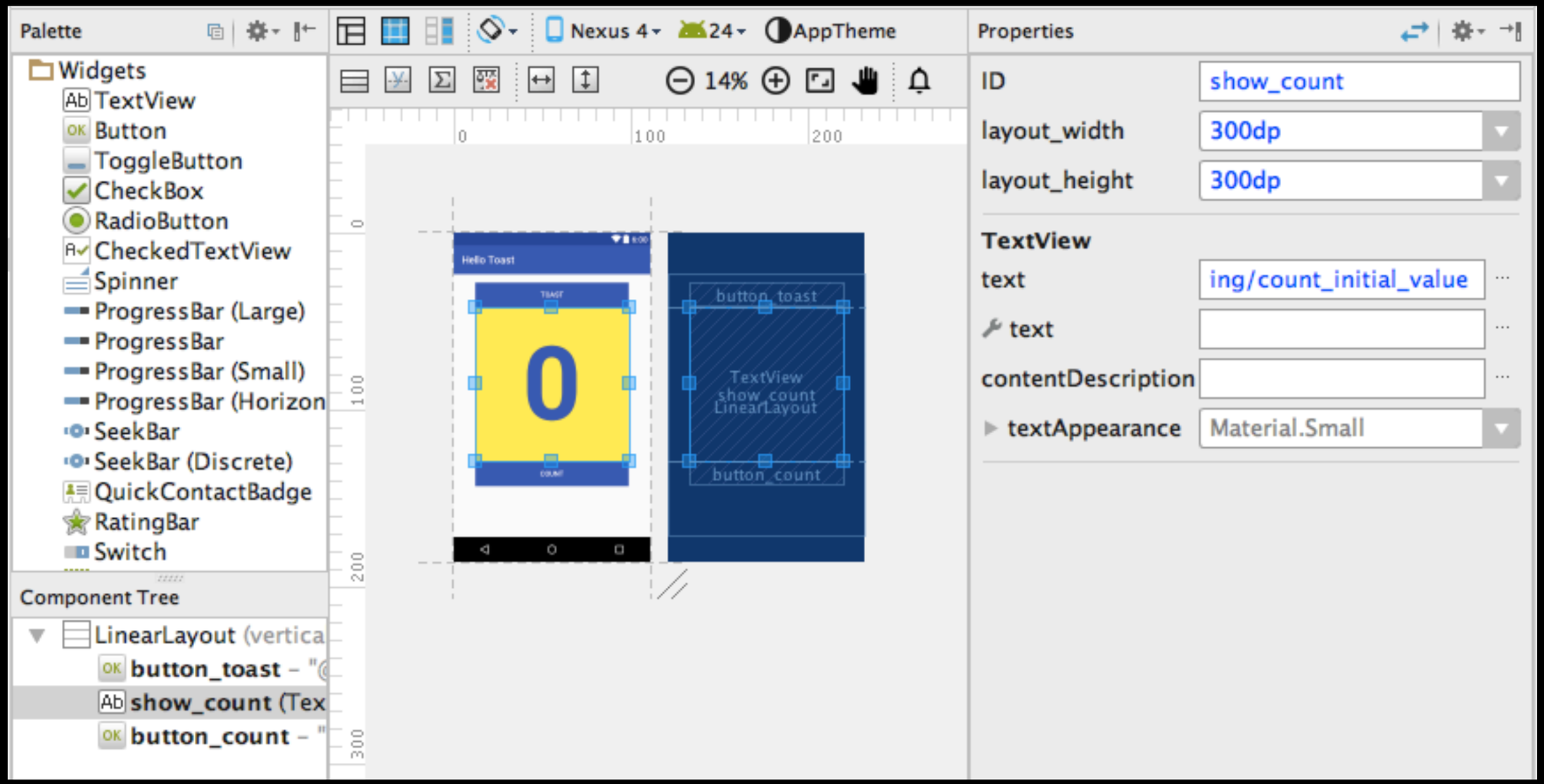
CheckBox
RadioButton
Switch

CREATING AND LAYING OUT VIEWS

- Graphically within Android Studio
- XML Files
- Programmatically

VIEWS DEFINED IN LAYOUT EDITOR

Visual representation of what's in XML



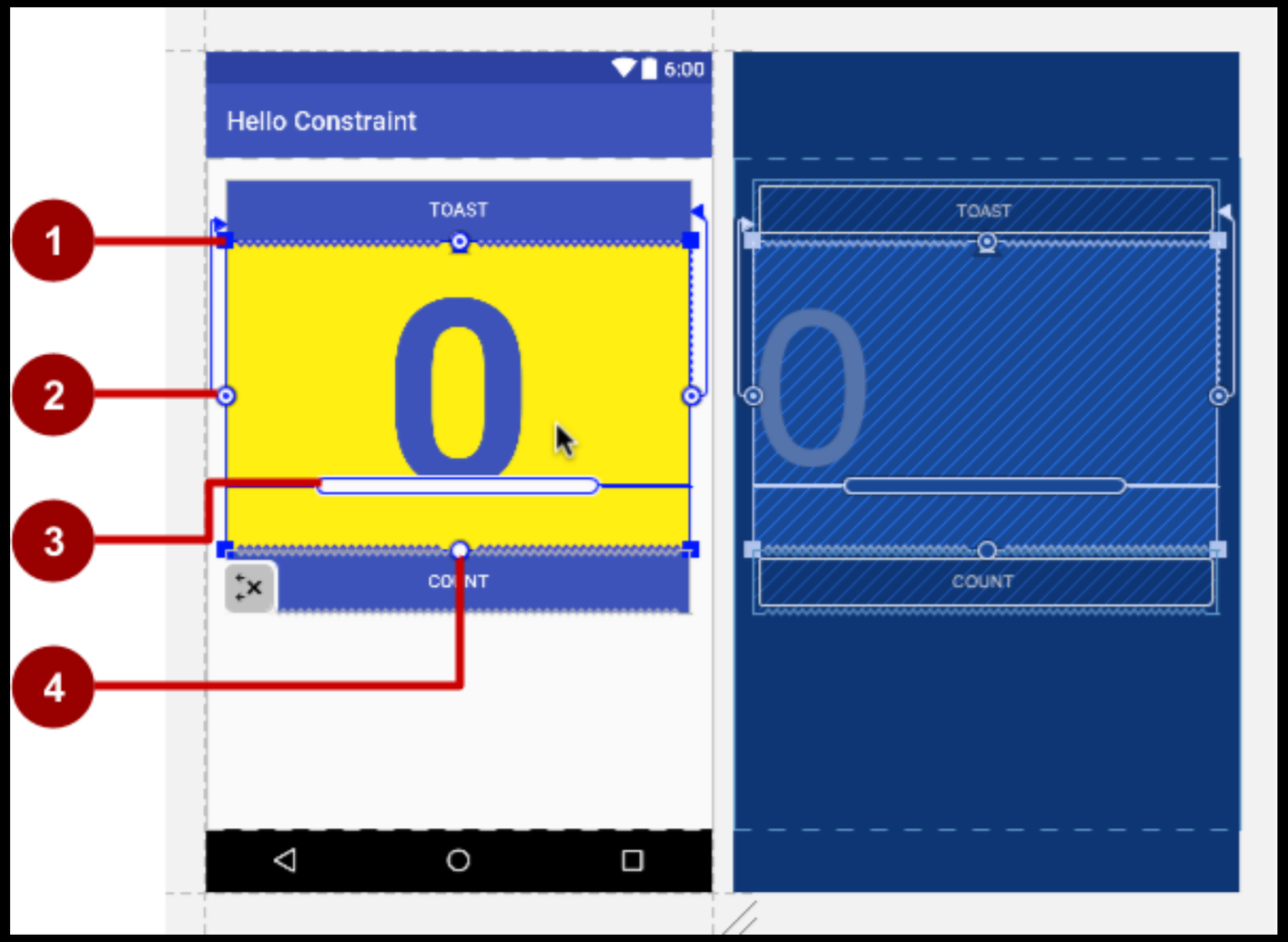
USING THE LAYOUT EDITOR

1. Resizing handle

2. Constraint line and handle

3. Baseline handle

4. Constraint handle



VIEWS DEFINED IN XML

```
<TextView
        android:id="@+id/show_count"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/myBackgroundColor"
        android:text="@string/count_initial_value"
        android:textColor="@color/colorPrimary"
        android:textSize="@dimen/count_text_size"
        android:textStyle="bold"
/>
```

VIEW 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 VIEWS IN JAVA

In an Activity:

Context

```
TextView myText = new TextView(this);

myText.setText("Display this text!");
```

WHAT IS THE 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(this);
```

CUSTOM VIEWS

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

TEXTVIEW

TEXTVIEW FOR TEXT

- TextView is a view for displaying single and multi-line text
- EditText is a subclass of TextView with editable text
- Controlled with layout attributes
- Set text statically from a string resource in XML, or dynamically from Java code and any source

FORMATTING TEXT IN STRING RESOURCE

- Use <b> and <i> HTML tags for bold and italics
- All other HTML tags are ignored
- String resources: one unbroken line = one paragraph
- \n starts a new a line or paragraph

CREATING TEXTVIEW IN XML

```
<TextView android:id="@+id/textview"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="@string/my_story"/>
```

COMMON TEXTVIEW ATTRIBUTES

- android:text—text to display
- android:textColor—color of text
- android:textAppearance—predefined style or theme
- android:textSize—text size in sp
- android:textStyle—normal, bold, italic, or bold|italic
- android:typeface—normal, sans, serif, or monospace
- android:lineSpacingExtra—extra space between lines in sp

FORMATTING ACTIVE WEB LINKS

```
<string name="article_text">... www.rockument.com ...</string>

<TextView

    android:id="@+id/article"

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:autoLink="web"

    android:text="@string/article_text"/>
```

autoLink values: “web", "email", "phone", "map", "all"

CREATING TEXTVIEW IN JAVA CODE

```
TextView myTextview = new TextView(this);

myTextView.setWidth(LayoutParams.MATCH_PARENT);

myTextView.setHeight(LayoutParams.WRAP_CONTENT);

myTextView.setMinLines(3);

myTextView.setText(R.string.my_story);

myTextView.append(userComment);
```

DEMO

5 MINUTES BREAK

LINK

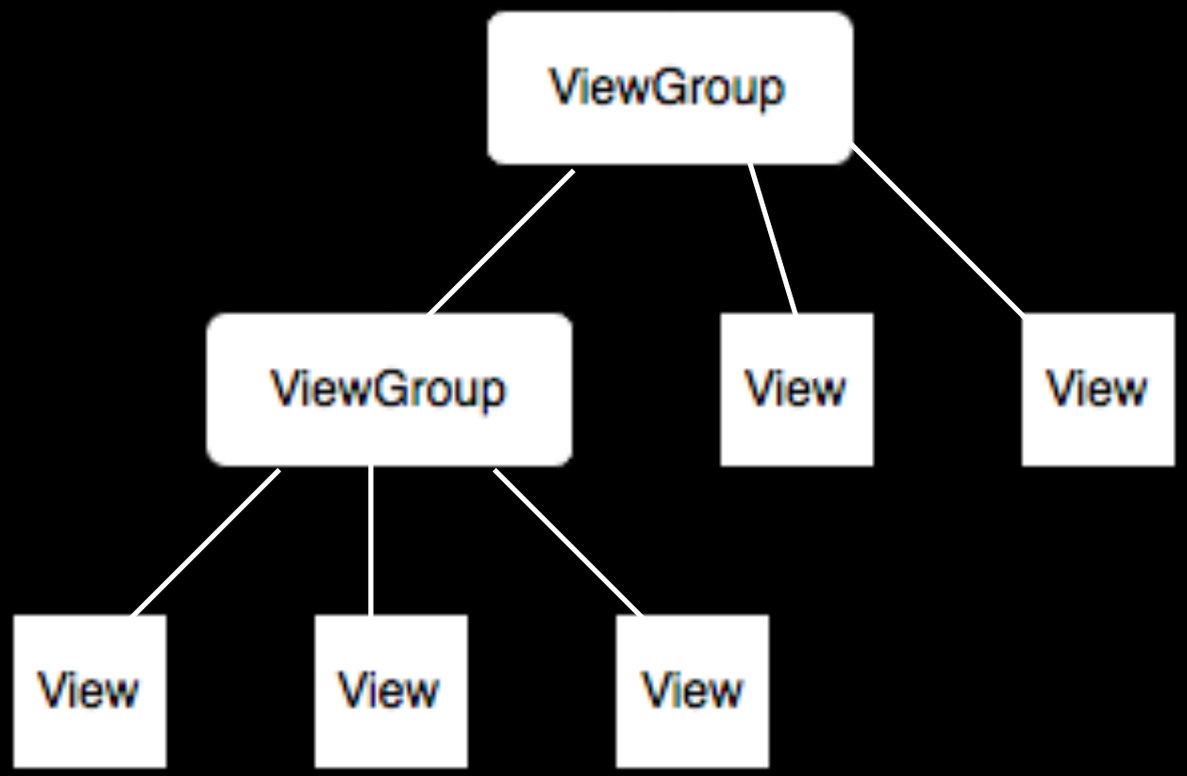
VIEWGROUPS & VIEW HIERARCHY

VIEWGROUP VIEWS

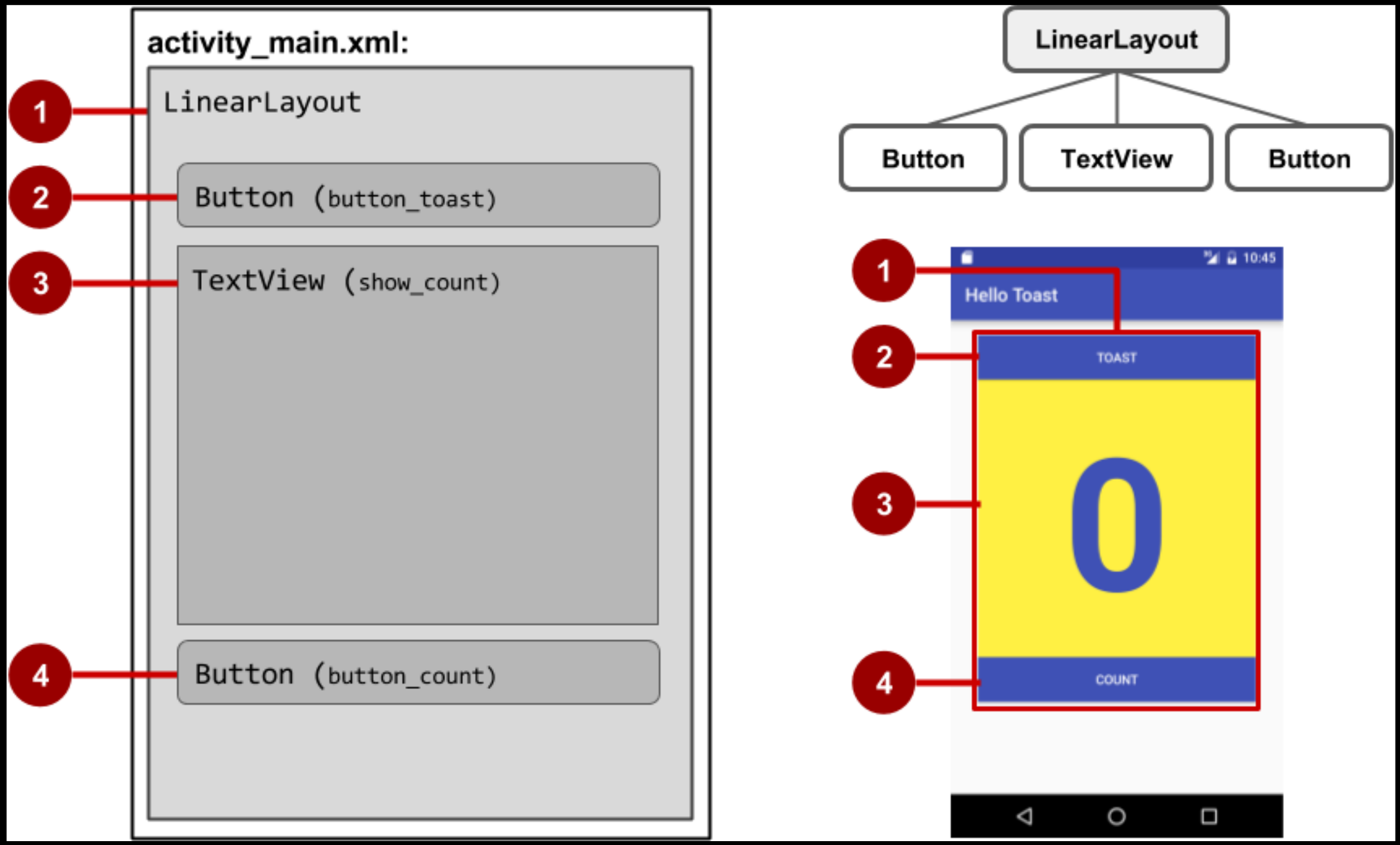
- A ViewGroup (parent) is a type of view that can contain other views (children)
- ViewGroup is the base class for layouts and view containers
- ScrollView—scrollable view that contains one child view
- LinearLayout—arrange views in horizontal/vertical row
- RecyclerView—scrollable "list" of views or view groups

HIERARCHY OF VIEWGROUPS & VIEWS

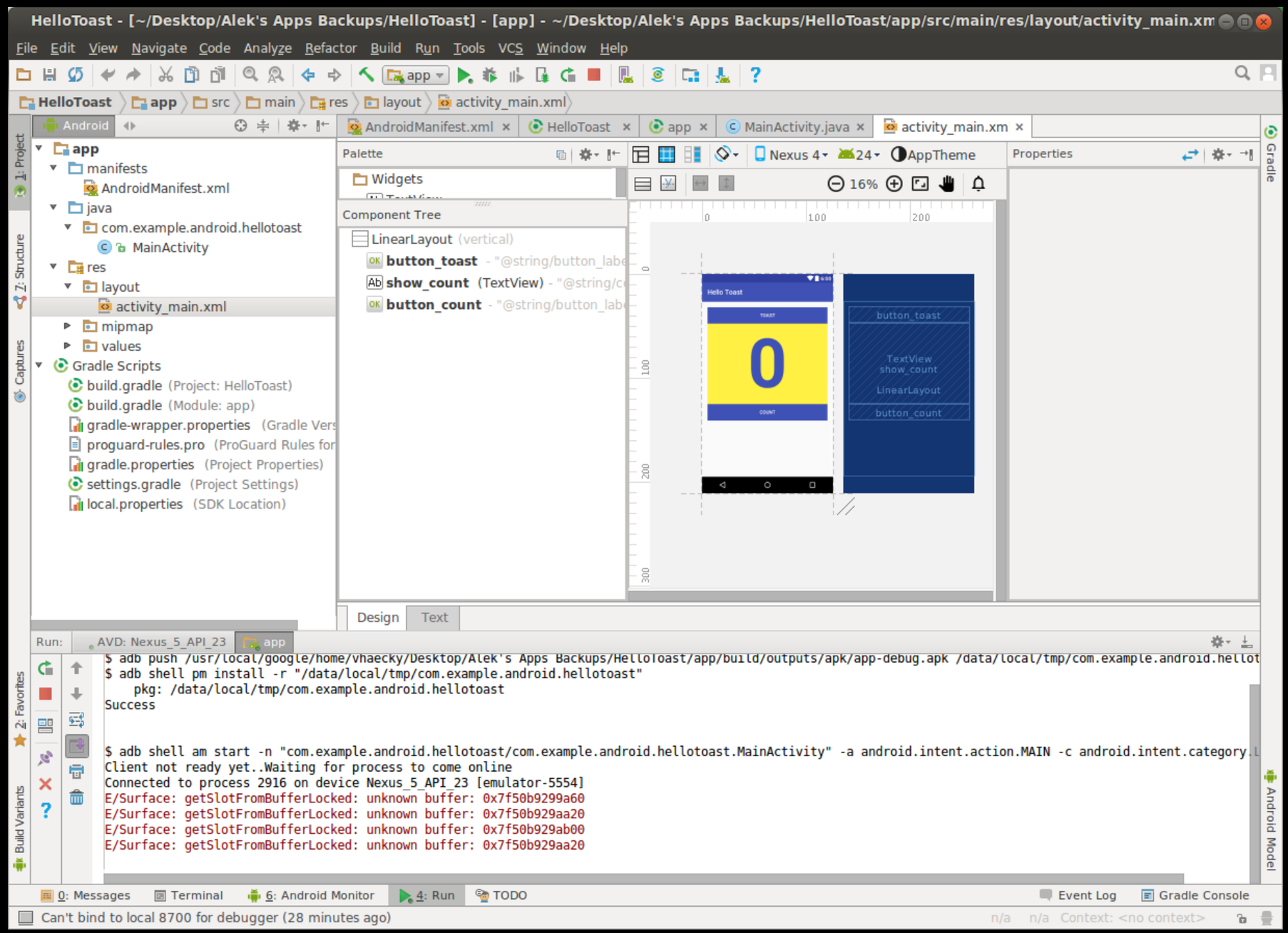
Root view is always a view group



VIEW HIERARCHY AND SCREEN LAYOUT



VIEW HIERARCHY IN THE COMPONENT TREE



BEST PRACTICES FOR VIEW HIERARCHIES

- 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

LAYOUTS

LAYOUT VIEWS

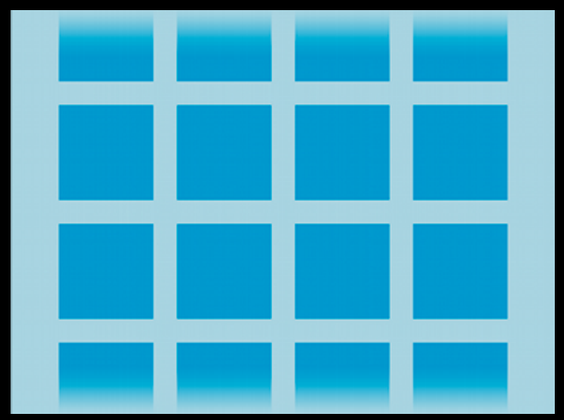
Layouts

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

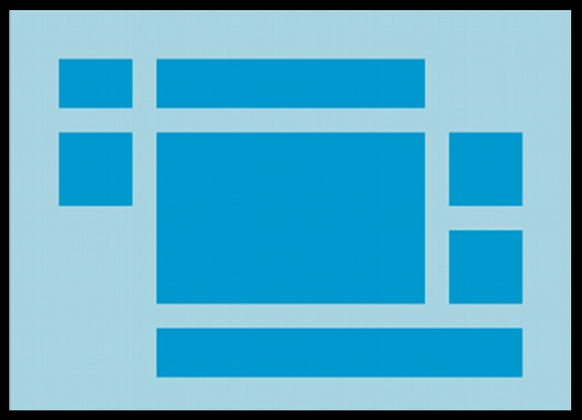
COMMON LAYOUT CLASSES

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
```

LAYOUT CREATED IN JAVA ACTIVITY CODE

```
LinearLayout linearL = new LinearLayout(this);

linearL.setOrientation(LinearLayout.VERTICAL);

TextView myText = new TextView(this);

myText.setText("Display this text!");

linearL.addView(myText);

setContentView(linearL);
```

SETTING WIDTH AND HEIGHT IN JAVA CODE

Set the width and height of a view:

```
LinearLayout.LayoutParams layoutParams =
   new Linear.LayoutParams(

      LayoutParams.MATCH_PARENT,
      LayoutParams.WRAP_CONTENT);

myView.setLayoutParams(layoutParams);
```

QUESTIONS?