4. Lesson 07/03/23

Views, view groups, and view hierarchy

Everything you see is a view

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

UI consists of a hierarchy of objects called views.

What is a view?

View subclasses are basic user interface building blocks:

- **Display text** (TextView), **edit text** (EditText)
- **Buttons**, **menus**, other controls
- **Scrollable** (ScrollView, RecyclerView)
- **Show images** (ImageView)
- **Group views** (ConstraintLayout and LinearLayout)

View Attributes

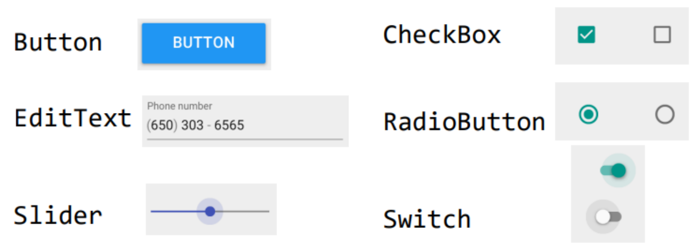
- Location (positioning): expressed as coordinates
- Dimensions: width and height

(The unit for location and dimensions is the density-independent pixel (dp)).

- Color
- May have focus
- May be interactive
- May be visible or not
- Relationship to other views

Examples of view subclasses

The Android system provides hundreds of predefined View subclasses.



Create views and layouts

View elements can be created in XML layout resource files.

To do it use:

- Android Studio layout editor using the visual representation of XML.
- XML editor.

Layout resources are listed within the layout folder in the res folder in the Project > Android pane.

View can be also created programmately, using Java code.

Android Studio Layout editor

Elements inside the layout editor:

1. XML layout file
2. Design and Text tabs
3. Palette pane
4. Component Tree
5. Design and blueprint panes
6. Attributes tab

You can see both the code pane and the design pane at the same time by clicking “split”.

View defined in XML

For example a TextView with some attributes:

```
<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 attributes in XML

```
android:text="@string/button_label_next“
```

This means that the string associated to the “text” attribute is defined inside the “*string.xml*” file.

```
android:background="@color/myBackgroundColor"
```

This means that the color associated to the “background” attribute is defined inside the “*color.xml*” file.

Custom views

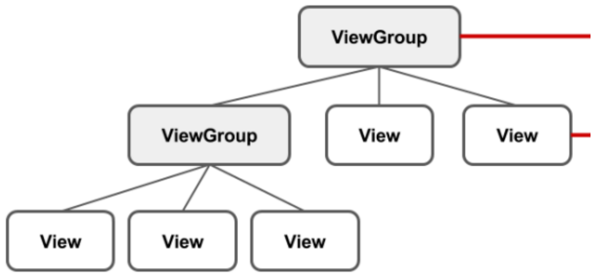
Over 100 different types of views available from the Android system, all children of the View class. It is also possible to create custom views by subclassing existing views.

View Group and View Hierarchy

ViewGroup contains "child" views

View elements for a screen are organized in a hierarchy.

At the root of this hierarchy there is a **ViewGroup (parent).** It contains the layout of the entire screen.



ViewGroup can contain child View elements or other ViewGroups.

Commonly used ViewGroups:

- **ContraintLayout: p**ositions UI elements using constraint connections to other elements and to the layout edges.
- **ScrollView:** contains one element and enables scrolling.
- **RecycleView:** contains a list of elements and enables scrolling.

ViewGroups for layouts

Some **ViewGroup groups** are designed as **layouts**:

- Organize child View elements in a specific way.
- Typically used as the root ViewGroup.
- Can be ina row, column, grid, table, absolute.

Common Layout Classes

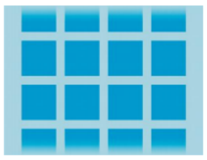
ConstraintLayout

A group of child View elements using constraints, edges, and guidelines to control how the elements are positioned relative to other elements in the layout.

LinearLayout

A group of child View elements positioned and aligned horizontally or vertically.

GridLayout



A group that places its child View elements in a rectangular grid that can be scrolled.

Class hierarchy vs. layout hierarchy

- **View class-hierarchy** is standard object-oriented class inheritance
- **Layout hierarchy** is how views are visually arranged.

Best practice for view hierarchies

- Use smallest number of simplest views possible.
- Keep the hierarchy flat-limit nesting of views and view groups.

Exercize

01.2A Views, view groups, and view hierarchy - Exercise.pdf

- **RadioButtons** implemented with a **RadioGroup** (e RadioGroup is a subclass of LinearLayout)
- **ImageView**
- Method **clickFlag**: recognizes which option button has been selected and changes the image displayed in the image view.
- Four **.png images**