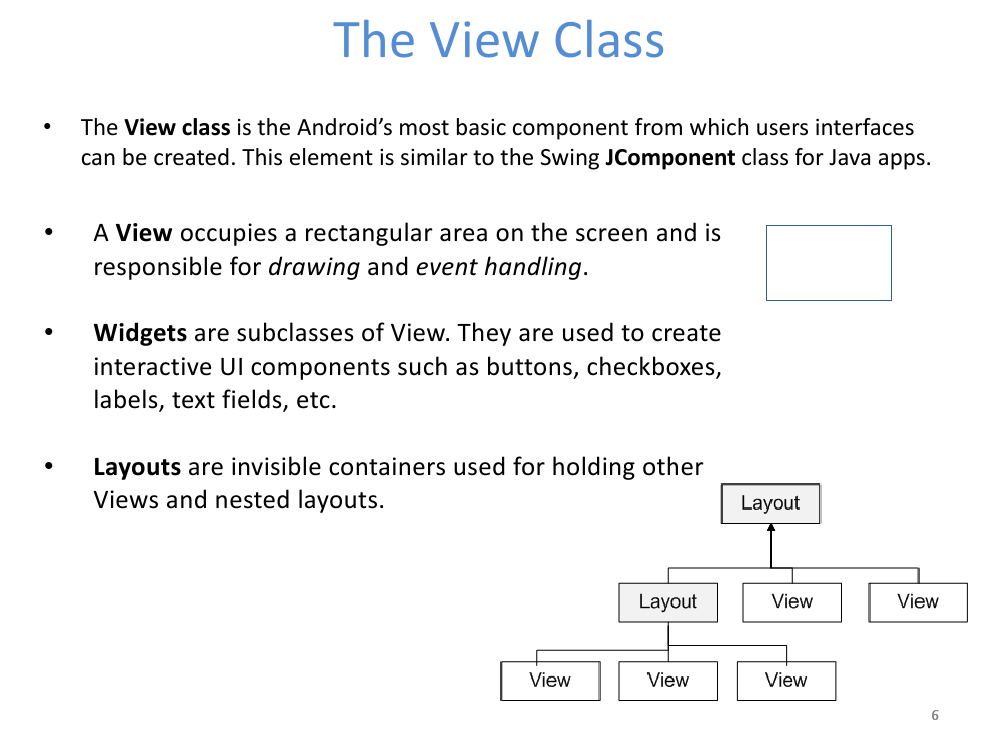
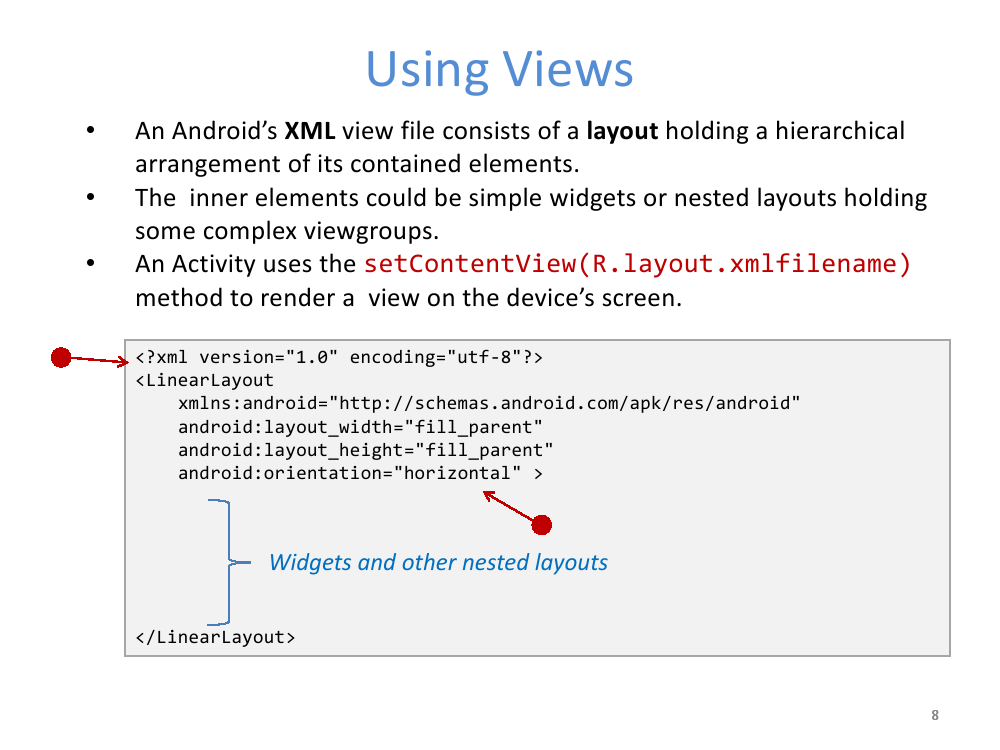
**Agenda**

* UI Overview
* View Class
* Using View
* Graphical User Interfaces
  + Android Layouts
  + Input Controls
  + Toast, Dialog and Snackbar
  + Navigation, ActionBar and Menus
* Assignments

**UI Overview**

* All user interface elements in an Android app are built using View and View Group objects.
* A View is an object that draws something on the screen that the user can interact with.
* A ViewGroup is an object that holds other View (and ViewGroup ) objects in order to define the layout of the interface.
* Android provides a collection of both View and ViewGroup subclasses that offer you common input controls (such as buttons and text fields) and various layout models (such as a linear or relative layout).



 **Common Layouts**

* Linear Layout
* Relative Layout
* Table Layout
* Frame Layout
* Absolute Layout
* Scroll View (Vertical)
* Horizontal Scroll View
* Simple List View
* List View in Android

**Graphical User Interface: Input Controls**

* Android Button
* Image Button
* Edit Text
* Check Box
* Radio Button
* Switch Button
* Toggle Button
* Rating Bar
* Spinner Example
* Date Picker
* Time Picker

**Toast, Dialog and Snackbar**

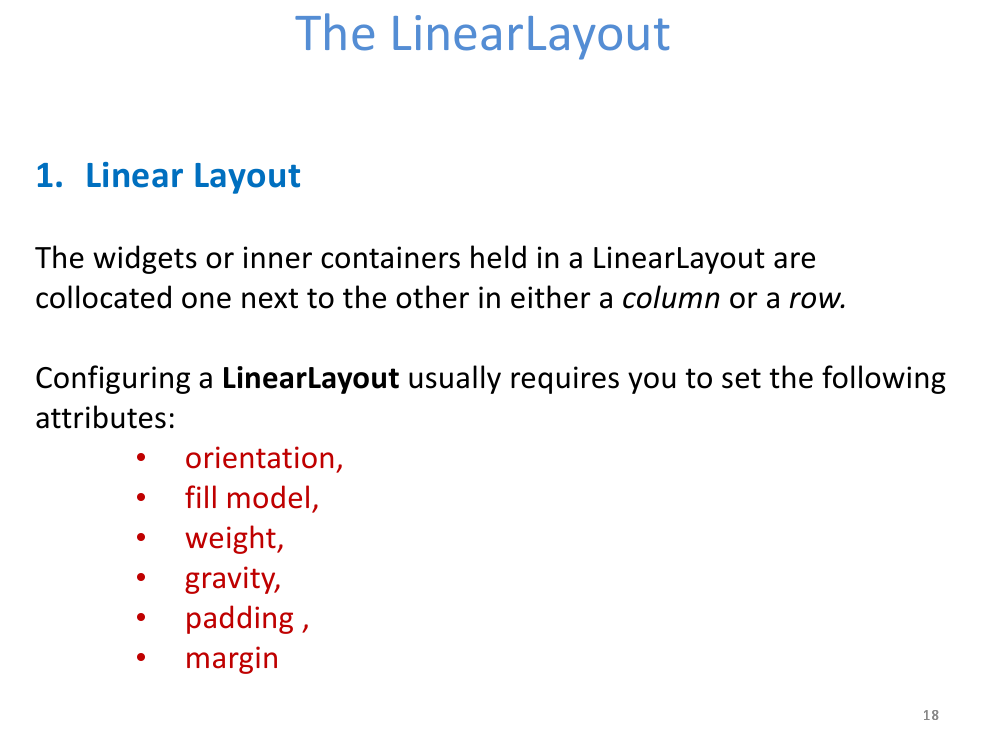
* Toast Message
* Alert Dialog
* Snackbar

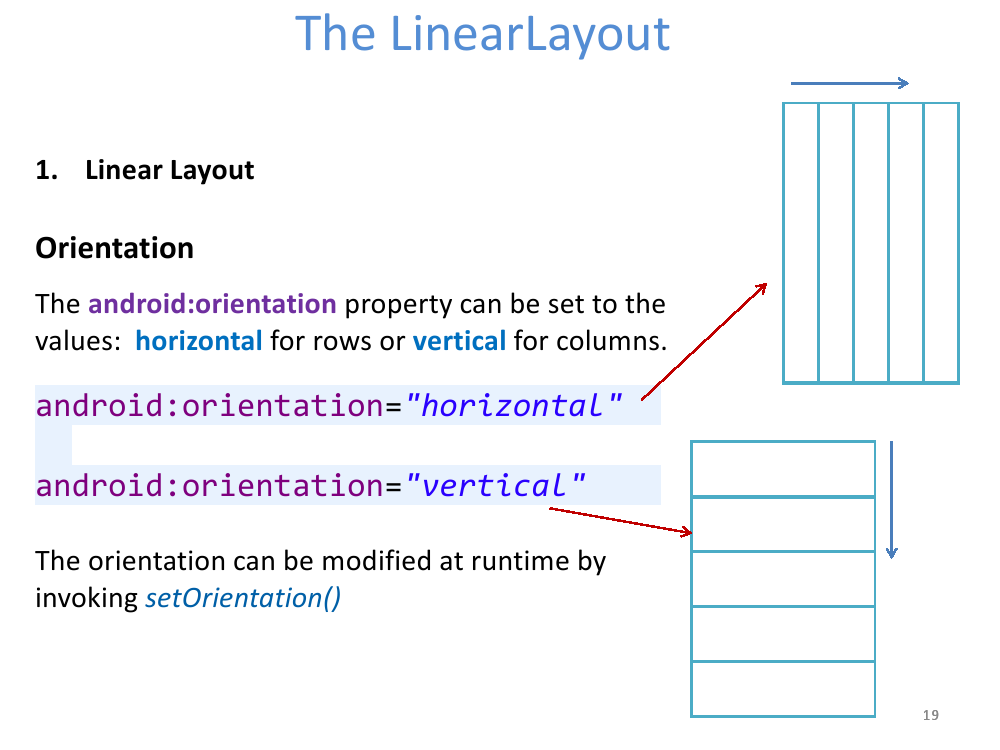
**Android User Interface Design: Navigation, ActionBar and Menus**

* Navigation Drawer
* Sliding Menu with WebView
* Dropdown Menu
* Action Bar/App Bar
* Toolbar
* Action Bar Tabs

**Linear Layout**

* Linear Layout is a layout which aligns the widgets or elements in a linear (Straight) fashion. Linear Layout consists of two types of orientation:
* Vertical Orientation,
* Horizontal Orientation.
* Vertical Orientation is shown above where the widgets such as Text View, Edit Text, and Button are aligned in a Vertical manner.





**1.2 Linear Layout: Fill Model**

All widgets inside a Linear Layout must include ‘width’ and ‘height’ attributes to establish the issue of empty space around them.

android:layout\_width;

android:layout\_height;

Values used in defining height and width can be:

1. 125dp
2. wrap\_content
3. match\_parent = fill\_parent

**1.2Linear Layout : Weight**

Indicates how much of the extra space in the LinearLayout will be allocated to the view. Use 0 of if the view should not be stretched. The bigger the weight the larger the extra space given to that widget.

android:layout\_weight=”1”;

**Gravity**

Indicates how to place an object within a contaioner. In the example the text is centered.

android:gravity=”center”;

**The Linear Layout – Gravity**

android:layout\_gravity=”left,center,right,top,bottom,etc”;

**1.4 Linear Layout: Padding**

The padding specifies how much extra space there is between the boundaries of the widget’s “cell” and the actual widget contents.

android:padding=”30dp”;

**1.4 Linear Layout: (External) Margin**

android:layout\_margin=”6dp”