Linear Layout

Linear layouts are one of the simplest and most common types of layouts used by Android developers to organize controls within their user interfaces. The linear layout works much as its name implies: it organizes controls linearly in either a vertical or horizontal fashion. When the layout's orientation is set to vertical, all child controls within it are organized in a single column; when the layout's orientation is set to horizontal, all child controls within it are organized in a single row.

The following figure shows a linear layout with seven TextView controls. This linear layout's orientation is set to vertical, causing each of the TextView controls to display in a single column. Each TextView control has its text attribute set to a color, with that same color set as the background for the control; each control is stretched the width of the screen by setting the controls' layout_width attribute to fill_parent.

Defining an XML Layout Resource with a Linear Layout

The most convenient and maintainable way to design application user interfaces is by creating XML layout resources. This method greatly simplifies the UI design process, moving much of the static creation and layout of user interface controls and definition of control attributes, to the XML, instead of littering the code.

XML layout resources must be stored in the /res/layout project directory hierarchy. This screen is basically a vertical linear layout set to fill the entire screen, which is achieved by setting its layout_width and layout_height attributes to fill_parent. This layout resource file is defined in XML as follows:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent" android:layout_height="fill_parent"
   android:orientation="vertical">
   <TextView android:text="RED" android:id="@+id/TextView01"
      android:layout_height="wrap_content" android:background="#f00"
      android:layout_width="fill_parent" android:layout_weight=".14"
```

```
      android:gravity="center" android:textColor="#000"></TextView>
   <TextView android:text="ORANGE" android:id="@+id/TextView02"
      android:layout_height="wrap_content" android:layout_width="fill_parent"
      android:layout_weight=".15" android:background="#ffa500"
      android:gravity="center" android:textColor="#000"></TextView>
   <TextView android:text="YELLOW" android:id="@+id/TextView03"
      android:layout_height="wrap_content" android:layout_width="fill_parent"
      android:layout_weight=".14" android:background="#ffff00"
      android:gravity="center" android:textColor="#000"></TextView>
   <TextView android:text="GREEN" android:id="@+id/TextView04"
      android:layout_height="wrap_content" android:layout_width="fill_parent"
      android:layout_weight=".15" android:background="#0f0" android:gravity="center"
      android:textColor="#000"></TextView>
   <TextView android:text="BLUE" android:id="@+id/TextView05"
      android:layout_height="wrap_content" android:layout_width="fill_parent"
      android:layout_weight=".14" android:background="#00f" android:gravity="center"
      android:textColor="#fff"></TextView>
   <TextView android:text="INDIGO" android:id="@+id/TextView06"
      android:layout_height="wrap_content" android:layout_width="fill_parent"
      android:layout_weight=".14" android:background="#4b0082"
      android:gravity="center" android:textColor="#fff"></TextView>
   <TextView android:text="VIOLET" android:id="@+id/TextView07"
      android:layout_height="wrap_content" android:layout_width="fill_parent"
      android:layout_weight=".14" android:background="#ee82ee"
      android:gravity="center" android:textColor="#000"></TextView>
</LinearLayout>
```

Exploring the Important Linear Layout Properties and Attributes

Now let's talk a bit about the attributes that help configure a linear layout and its child controls.

Some specific attributes apply to linear layouts. Some of the most important attributes you'll use with linear layouts include:

- The orientation attribute (required), which can be set to vertical or horizontal (class: LinearLayout)
- The gravity attribute (optional), which controls how all child controls are aligned and displayed within the linear layout (class: LinearLayout)
- The layout_weight attribute (optional, applied to each child control) specifies each child control's relative importance within the parent linear layout (class: LinearLayout.LayoutParams)

Also, general ViewGroup-style attributes apply to linear layouts. These include:

- Generic Layout Parameters such as layout_height (required) and layout_width (required) (class: ViewGroup.LayoutParams)
- Margin Layout Parameters such as margin_top, margin_left, margin_right and margin_bottom (class: ViewGroup. MarginLayoutParams)
- Layout Parameters such as layout_height and layout_width (class: ViewGroup.LayoutParams)