Slide 9.1: Android user interface
Slide 9.3: Hypertext
Home

PRINT VERSION

Android User Interface (UI) (Cont.)

The name of an XML element for a view is respective to the Android class it represents. So a `TextView` element creates a `TextView` widget in your UI, and a `LinearLayout` element creates a `LinearLayout` view group. For example, a simple vertical layout with a text view and a button looks like this:

```
01 <?xml version="1.0" encoding="utf-8"?>
02 <LinearLayout
03   xmlns:android            = "http://schemas.android.com/apk/res/android"
04   android:layout_width     = "fill_parent"
05   android:layout_height    = "fill_parent"
06   android:orientation      = "vertical" >
07   <TextView
08     android:id             = "@+id/text"
09     android:layout_width   = "wrap_content"
10     android:layout_height  = "wrap_content"
11     android:text           = "I am a TextView" />
12   <Button
13     android:id             = "@+id/button"
14     android:layout_width   = "wrap_content"
15     android:layout_height  = "wrap_content"
16     android:text           = "I am a Button" />
17 </LinearLayout>
```

When you load a layout resource in your app, Android initializes each node of the layout into a runtime object you can use to define additional behaviors, query the object state, or modify the layout. For a complete guide to creating a UI layout, see XML Layouts.

User Interface Components

You don't have to build all of your UI using `View` and `ViewGroup` objects. Android provides several app components that offer a standard UI layout for which you simply need to define the content. These UI components each have a unique set of APIs that are described in their respective documents, such as Adding the App Bar, Dialogs, and Status Notifications.

Slide 9.1: Android user interface
Slide 9.3: Hypertext
Home