

Android Styles & Themes



Agenda

- ❑ Define Android App Styles
- ❑ Apply Android App styles to the UI
- ❑ Platform Styles and Themes

Defining Styles

- Style is used to define the look of screen
- It can be applied on individual view or entire activity
- It is defined in XML format
- It will be inside the **res/values/** directory of the specified project
- The name of the XML resource file can be arbitrary, but it must use the .xml extension

```
<?xml version="1.0" encoding="utf-8"?> <resources>
<style name="CustomFontStyle">
<item name="android:layout_width">fill_parent</item>
<item name="android:layout_height">wrap_content</item>
<item name="android:capitalize">characters</item>
<item name="android:typeface">monospace</item>
<item name="android:textSize">12pt</item>
<item name="android:textColor">#00FF00</item>/> </style> </resources>
```

Using Styles

```
<?xml version="1.0" encoding="utf-8"?>

<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:id="@+id/text_id" style="@style/CustomFontStyle"

android:text="@string/hello_world" />

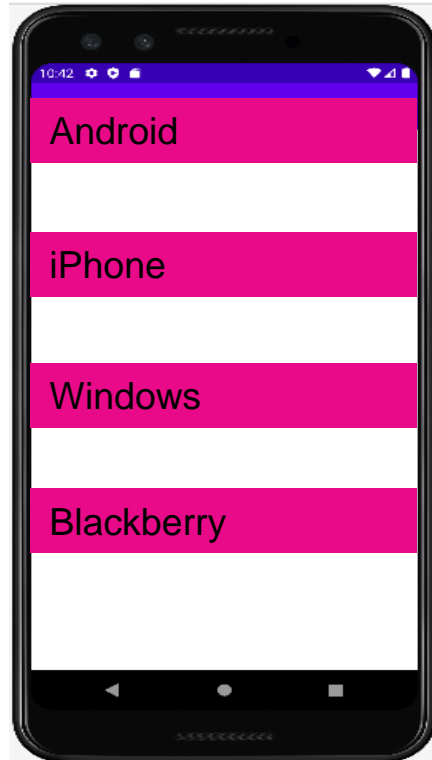
</LinearLayout>
```

Android app Style Inheritance

AndroidManifest.xml is used to apply style

```
<activity  
    android:name="com.myapp.MyActivity"  
    ...  
    android:theme="@style/MyCustomTheme"  
/>
```

Style Inheritance (Real Interface)



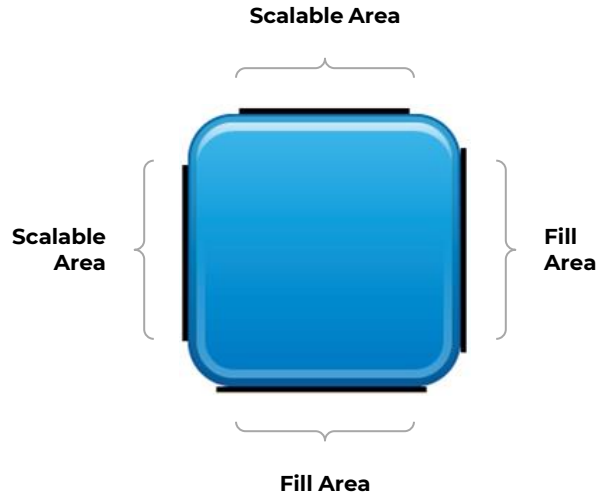
Android app Colors to Theme Attribute

- <item> tag is used to specify color and other in style.xml.
- All configuration is done in style.xml file.

```
<resources>
    ...
    <style name="MyCustomTheme" ...>
        <item name="android:windowBackground">@color/my_custom_color</item>
        <item name="android:colorBackgroundCacheHint">@color/my_custom_color</item>
    </style>
    ...
</resources>
```

Android app Nine-Patch with Button

- This is a drawable entity used to auto scale or scale down as per requirement
- It is most useful when the size and position can be re-configured at run time.



Custom Nine-Patch with Button

- Define a new Style

```
<resources>
...
<style name="MyCustomButton" parent="android:Widget.Button">
  <item name="android:background">@drawable/my_nine_patch</item>
</style>
...
</resources>
```

- Apply to the Theme

```
<resources>
...
<style name="MyCustomTheme" parent=...>
  ...
  <item name="android:buttonStyle">@style/MyCustomButton</item>
</style>
...
</resources>
```

Android App Theme

Theme is an Android App Style which is applied to entire screen components not individual view.

- On applying a theme each view which belongs to that category will consume that style and converts itself.
- To apply a theme in an Android app, an attribute named **android:theme** need to configured

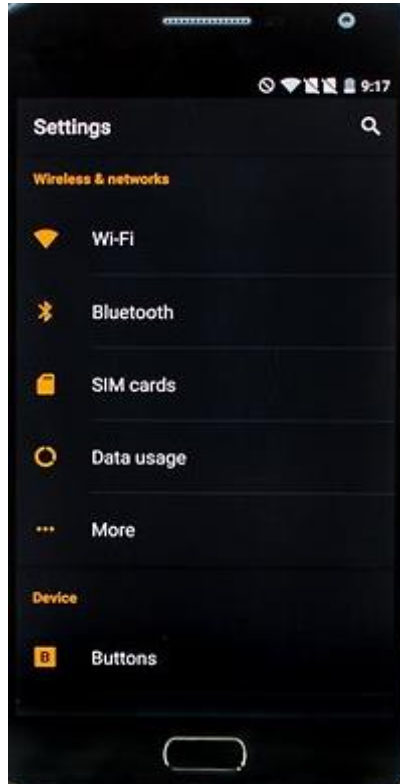
To set a theme for all the activities of your application, open the **AndroidManifest.xml** file and edit the **<application>** tag to include the **android:theme** attribute with the style name

```
<application android:theme="@style/CustomFontStyle">
```

But if you want a theme applied to just one Activity in your application, then add the android:theme attribute to the <activity> tag only.

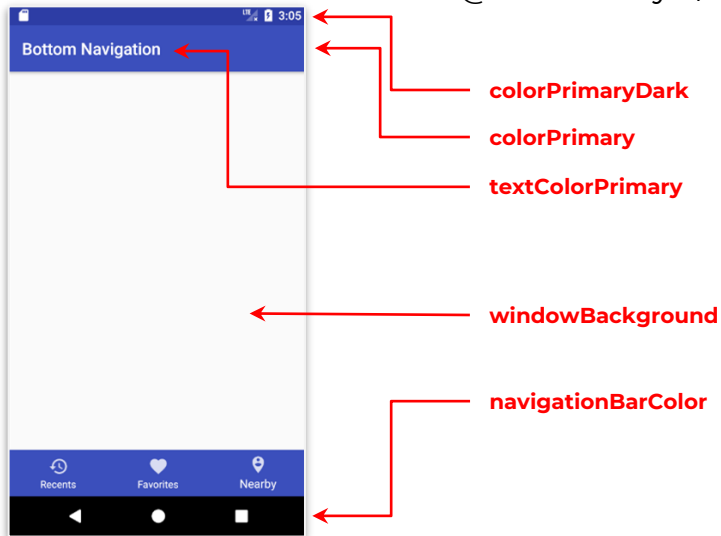
```
<activity android:theme="@style/CustomFontStyle">
```

Android App Theme (Real Interface)



Default Styles & Themes

- The Android platform provides a large collection of styles and themes that you can use in your applications. You can find a reference of all available styles in the **R.style** class
- To use the styles listed here, replace all underscores in the style name with a period
- For example, you can apply the Theme_NoTitleBar theme with "@android:style/Theme.NoTitleBar"



Questions