App Styling & Theming in Android (Colors, Styles, Themes, Material Design)

1. What is App Styling & Theming?

App styling & theming means defining how an Android app **looks and feels**, including:

- Colors
- Fonts
- Text sizes
- Buttons appearance
- Shapes
- Dark mode / Light mode
- Overall visual design consistency

Android provides a structured way to define these look-and-feel elements using:

- colors.xml
- styles.xml
- themes.xml
- Material Design guidelines

2. Colors in Android

Colors are stored inside:

app/src/main/res/values/colors.xml

Example:

```
<resources>
   <color name="purple_500">#6200EE</color>
   <color name="purple_700">#3700B3</color>
   <color name="teal_200">#03DAC5</color>
   <color name="white">#FFFFFF</color>
   <color name="black">#000000</color>
</resources>
```

These color names can be used in layouts, buttons, text, backgrounds, etc.

3. Styles in Android

A **style** is like a reusable formatting rule.

Example in styles.xml:

```
<style name="MyButtonStyle">
   <item name="android:background">@color/purple_500</item>
   <item name="android:textColor">@color/white</item>
   <item name="android:padding">12dp</item>
   <item name="android:textStyle">bold</item>
</style>
```

Use it on a button:

```
<Button
   style="@style/MyButtonStyle"
   android:text="Click Me" />
```

4. Themes

A **theme** applies styles to the entire app or full Activity.

File:

res/values/themes.xml

Example:

```
<style name="Theme.MyStyledApp" parent="Theme.Material3.DayNight.NoActionBar">
   <item name="colorPrimary">@color/purple_500</item>
   <item name="colorSecondary">@color/teal_200</item>
</style>
```

This theme defines:

- Primary color
- Secondary color
- Default button color
- Toolbar color
- etc.

Set it in AndroidManifest:

```
<application
   android:theme="@style/Theme.MyStyledApp" />
```

5. Material Design (Intro)

Material Design is Google’s design system for:

- Cards
- Buttons
- AppBar
- Navigation
- Shapes
- Elevation (shadow)
- Responsive layout

Material Design components come from:

implementation("com.google.android.material:material:1.9.0")

Examples:

- MaterialButton
- CardView
- TextInputLayout
- FloatingActionButton

6. EXAMPLE APP

“Beautiful UI App Using Styling & Theming”

This app will show:

- Custom colors
- Custom styles
- Material button
- CardView
- Themed activity

FILE 1: colors.xml

(Place in res/values/colors.xml)

```
<resources>
   <color name="primaryColor">#673AB7</color>
   <color name="secondaryColor">#9575CD</color>
   <color name="buttonColor">#7E57C2</color>
   <color name="white">#FFFFFF</color>
   <color name="lightGray">#F5F5F5</color>
</resources>
```

FILE 2: themes.xml

(Place in res/values/themes.xml)

```
<resources xmlns:tools="http://schemas.android.com/tools">
   <style name="Theme.StyledDemo" parent="Theme.Material3.DayNight.NoActionBar">
      <item name="colorPrimary">@color/primaryColor</item>
      <item name="colorSecondary">@color/secondaryColor</item>
      <item name="android:navigationBarColor">@color/primaryColor</item>
   </style>
</resources>
```

FILE 3: styles.xml

```
<resources>
   <!-- Custom Button Style -->
   <style name="MyRoundedButton">
      <item name="android:backgroundTint">@color/buttonColor</item>
      <item name="android:textColor">@color/white</item>
      <item name="android:padding">14dp</item>
      <item name="android:fontFamily">sans-serif-medium</item>
   </style>

   <!-- Text style -->
   <style name="HeadingStyle">
```

```
    <item name="android:textSize">26sp</item>
    <item name="android:textStyle">bold</item>
    <item name="android:textColor">@color/primaryColor</item>
  </style>
</resources>
```

FILE 4: activity_main.xml
A beautiful UI layout using Material Components.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  android:padding="20dp"
  android:background="@color/lightGray">

  <TextView
    android:id="@+id/tvTitle"
    style="@style/HeadingStyle"
    android:text="Beautiful Styled App"
    android:layout_marginBottom="20dp"
    android:gravity="center"/>

  <com.google.android.material.card.MaterialCardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="20dp"
    app:cardBackgroundColor="@color/white"
    app:cardCornerRadius="12dp"
    app:cardElevation="6dp">

    <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="This is a Material Card with themed styling."
      android:textSize="18sp"/>
  </com.google.android.material.card.MaterialCardView>

  <Button
    android:id="@+id/btnClick"
    style="@style/MyRoundedButton"
    android:text="Click Me"
    android:layout_marginTop="30dp" />

</LinearLayout>
```

FILE 5: MainActivity.kt

package com.example.styledemo

```
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.Toast

class MainActivity : AppCompatActivity() {
   override fun onCreate(savedInstanceState: Bundle?) {
      super.onCreate(savedInstanceState)
      setContentView(R.layout.activity_main)

      val btn = findViewById<Button>(R.id.btnClick)

      btn.setOnClickListener {
         Toast.makeText(this, "Button Clicked!", Toast.LENGTH_SHORT).show()
      }
   }
}
```

FILE 6: AndroidManifest.xml (Important)

```
<application
   android:theme="@style/Theme.StyledDemo"
   ... >
```

Final Output of This App

You will see:

- A **clean, modern UI**
- A header text styled using styles
- A beautiful **Material Card**
- A **themed button** with rounded corners
- Consistent colors across the entire app
- Proper Material Design look

Note:

This app helps you to learn

✓ colors.xml usage

✓ styles.xml (reusable styles)

✓ themes.xml (full app theme)

✓ Material Components

✓ visually appealing UI design

✓ How to apply styles to components