

App Design

Drawables

- A **drawable** is a graphic that can be drawn to the screen. The **Drawable** class is used for adding rich images in an app with a minimal impact on its performance.
- Android supports the following image formats:
 - WebP – a modern image format developed by Google that provides superior lossless and lossy compression for images on the web.
 - PNG (Portable Network Graphics)
 - JPG/JPEG (Joint Photographic Experts Group)
 - GIF (Graphics Interchange Format)
 - BMP (Bitmap)
- WebP, PNG, and JPG are the recommended image formats.
- The image file should be saved in the **app/src/main/res/drawable** folder of the project.
- To display a drawable, create an **ImageView** element in XML:

```
<ImageView
    android:id="@+id/ic"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/ice_cream" />
```

- To use an image in the Java file:

```
ImageView img = new ImageView(this);
img.setImageResource(R.drawable.ice_cream);
```

- To match the bounds of the image with the dimension of the drawable:

```
img.setAdjustViewBounds(true);
img.setLayoutParams(new ViewGroup.LayoutParams(
    ViewGroup.LayoutParams.WRAP_CONTENT,
    ViewGroup.LayoutParams.WRAP_CONTENT));
```

- To add the image to the layout and set the layout as the content view:

```
ConstraintLayout cl = new ConstraintLayout(this);
cl.addView(img);
setContentView(cl);
```

Styles and Themes

- A **style** is a collection of attributes that define the look and format of a view.
- The styles that Android provides are called **platform styles**.
- To create a style, add a **<style>** element inside a **<resources>** element in any XML file located in the **values** folder inside the project's **res** folder.
- A **<style>** element includes the following:
 - name
 - parent (optional)
 - Any number of **<item>** elements as child elements of **<style>**. Each **<item>** element includes one (1) style attribute.

```
<resources>
    <style name="MyFont">
        <item name="android:typeface">monospace</item>
        <item name="android:textColor">#D7D6D7</item>
    </style>
</resources>
```

- To apply the style to a component:

```
<TextView
    style="@style/MyFont"
    android:text="@string/code_string" />
```

- Inheritance** allows a style to inherit the properties of an existing style.
- To inherit a platform style, use the parent attribute to specify the resource ID of the style you want to inherit.

```
<style name="Text1" parent="@android:style/TextAppearance">
    <item name="android:textColor">#00FF00</item>
</style>
```

- To apply this style, use **@style/Text1**.
- To inherit a custom style, use the name of the style you want to inherit as the first part of the new style's name and separate the parts with a period. Syntax: **name="StyletoInherit.NewStyle"**

```
<style name="MyFont.BlueLarge">
  <item name="android:textColor">#0000FF</item>
  <item name="android:textSize">36sp</item>
</style>
```

The original text color is overridden.

To apply this style, use `@style/MyFont.BlueLarge`.

- A **theme** is a collection of attributes that define the look and format of an activity or an entire app. Any style can be used as a theme.
- A default theme looks like this:

```
<!-- Base application theme. -->
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
  <!-- Customize your theme here. -->
  <item name="colorPrimary">@color/colorPrimary</item>
  <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
  <item name="colorAccent">@color/colorAccent</item>
</style>
```

- To apply a theme to an app, declare it inside an `<application>` element in the manifest file.
Ex. `android:theme="@style/AppTheme"`
- To apply a theme to an activity, declare it inside an `<activity>` element in the manifest file.
Ex. `<activity android:theme="@android:style/Theme.Dialog">`

References:

- DiMarzio, J. (2017). *Beginning Android programming with Android Studio*. Indiana: John Wiley & Sons, Inc.
- Google Developers Training Team. (2018). *Android developer fundamentals (version 2)*. Retrieved from <https://google-developer-training.github.io>