**✅ Section 13: Android App Resources – Ultimate Master Note**

**📚 What You Will Learn**

* Complete structure and purpose of the res/ folder
* How to use and customize each resource type (drawable, layout, mipmap, menu, values, font)
* Implementing shapes, themes, adaptive icons, custom fonts, and app menus
* Best practices for resource usage, performance, and scalability
* Advanced techniques (qualifiers, vector assets, shrinking, Compose equivalents)

**🗂️ 1. res/ Folder Overview**

css

CopyEdit

res/

├── drawable/ → Images, vector assets, and XML shape files

├── layout/ → UI XML layout files (screens)

├── mipmap/ → App launcher icons (multi-density)

├── values/ → Strings, colors, themes, dimensions, styles

├── font/ → Custom font files (.ttf/.otf)

├── menu/ → App bar and overflow menus

├── raw/, assets/ → Media, HTML, JSON, etc.

✅ Central location for all **non-code assets** that define the look, text, and feel of the app.

**🖼️ 2. drawable/ – Image & Shape Resources**

**✅ Add Images:**

* Formats: .png, .jpg, .webp, .xml
* Naming: lowercase only, no spaces, no leading digits

xml

CopyEdit

<ImageView android:src="@drawable/image1" />

**✅ Create Custom Shape Drawable:**

xml

CopyEdit

<!-- res/drawable/rounded\_button.xml -->

<shape xmlns:android="http://schemas.android.com/apk/res/android"

android:shape="rectangle">

<solid android:color="#2196F3"/>

<corners android:radius="20dp"/>

</shape>

xml

CopyEdit

<Button

android:background="@drawable/rounded\_button"

app:backgroundTint="@null" />

**✅ Use Vector Drawables (Recommended):**

xml

CopyEdit

<vector xmlns:android="..." android:width="24dp" android:viewportWidth="24">

<path android:fillColor="#000000" android:pathData="M5,10L15,10L10,20Z"/>

</vector>

**🧩 3. layout/ – Define UI Screens**

* Right-click res/layout → New → Layout Resource File

xml

CopyEdit

<LinearLayout>

<TextView android:text="@string/welcome\_text"/>

</LinearLayout>

🔁 Use **Split**, **Design**, or **Code** view to edit.

**📱 4. mipmap/ – Launcher Icons**

* Supports adaptive and legacy icons for all screen densities
* Use: New → Image Asset → Launcher Icons

xml

CopyEdit

<adaptive-icon xmlns:android="...">

<background android:drawable="@color/icon\_bg"/>

<foreground android:drawable="@mipmap/ic\_foreground"/>

</adaptive-icon>

**🌈 5. values/colors.xml – Centralized Colors**

xml

CopyEdit

<color name="blue\_sky">#2196F3</color>

<TextView android:textColor="@color/blue\_sky"/>

🎯 Avoid hardcoded color values in layouts.

**📝 6. values/strings.xml – Centralized Text**

xml

CopyEdit

<string name="welcome\_text">Hello, World!</string>

<TextView android:text="@string/welcome\_text"/>

🗺️ Enables easy **localization** and reusability.

**🎨 7. values/themes.xml – Styling App-Wide**

xml

CopyEdit

<item name="colorPrimary">@color/blue\_sky</item>

* themes.xml → Light mode
* themes.xml (night) → Dark mode
* Use Theme.Material3.DayNight (remove NoActionBar if menu not visible)

**🍔 8. menu/ – App Menus**

**✅ XML Menu Resource:**

xml

CopyEdit

<menu xmlns:android="http://schemas.android.com/apk/res/android">

<item

android:id="@+id/action\_home"

android:title="Home"

android:icon="@drawable/ic\_home"

app:showAsAction="ifRoom"/>

</menu>

**✅ Inflate and Handle Menu:**

java

CopyEdit

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.my\_menu, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

if (item.getItemId() == R.id.action\_home) {

Toast.makeText(this, "Home clicked", Toast.LENGTH\_SHORT).show();

return true;

}

return super.onOptionsItemSelected(item);

}

**🔤 9. font/ – Custom Fonts**

**✅ Add and Use:**

* Add .ttf/.otf in res/font/

xml

CopyEdit

<TextView android:fontFamily="@font/poppins\_bold"/>

**✅ Load Programmatically:**

kotlin

CopyEdit

val tf = ResourcesCompat.getFont(this, R.font.poppins\_bold)

**⚙️ 10. Resource Qualifiers – Adaptive UI**

| **Qualifier** | **Folder** | **Use Case** |
| --- | --- | --- |
| -land | layout-land/ | Landscape mode |
| -night | values-night/ | Dark mode |
| -fr | values-fr/ | French language |
| -sw600dp | layout-sw600dp/ | Tablets |
| -v26 | values-v26/ | Android 8+ |

**🛡️ 11. Performance & Optimization Tips**

**✅ Shrink Resources:**

gradle

CopyEdit

buildTypes {

release {

shrinkResources true

minifyEnabled true

}

}

**✅ Use:**

* .webp for images (30% smaller)
* Vector assets for resolution independence

**♿ 12. Accessibility**

xml

CopyEdit

<Button

android:contentDescription="@string/btn\_submit\_desc"/>

📣 Add meaningful descriptions for all interactive UI elements.

**🧭 13. Jetpack Compose Equivalents**

| **XML Resource** | **Compose Equivalent** |
| --- | --- |
| @drawable/ | painterResource() |
| @string/ | stringResource() |
| @font/ | FontFamily(Font(R.font.font\_name)) |
| Menu | DropdownMenu(), TopAppBar() |
| Colors | MaterialTheme.colorScheme.primary |

**💎 Best Practices Cheat Sheet**

| **Practice** | **Benefit** |
| --- | --- |
| Use @string/, @color/ | Supports localization |
| Prefer vector drawables | Smaller APK |
| Use shape drawables | Flexible UI |
| Consistent file naming | Avoids build errors |
| Use dimens.xml, styles.xml | Reuse styles, ensure UI consistency |
| Use qualifiers | Responsive and adaptive UI |
| Remove NoActionBar if needed | Show app bar menus |

**📘 BONUS: Extra Tools You Must Know**

**📏 dimens.xml**

xml

CopyEdit

<dimen name="default\_margin">16dp</dimen>

**🧬 styles.xml**

xml

CopyEdit

<style name="TitleStyle">

<item name="android:textSize">20sp</item>

<item name="android:textColor">@color/primary\_blue</item>

</style>

**🎞 Animated Vector Drawable**

xml

CopyEdit

<animated-vector android:drawable="@drawable/ic\_base">

<target android:name="path1" android:animation="@animator/anim1" />

</animated-vector>

**🌐 Localization**

xml

CopyEdit

values-fr/strings.xml → French

values-hi/strings.xml → Hindi

**🎵 raw/ and assets/**

* raw/ → .mp3, .mp4, etc.
* assets/ → .html, custom JSON, configs