



# ANDROID DEVELOPMENT

## LAYOUTS & RESOURCES

Tan Cher Wah ([isstcw@nus.edu.sg](mailto:isstcw@nus.edu.sg))



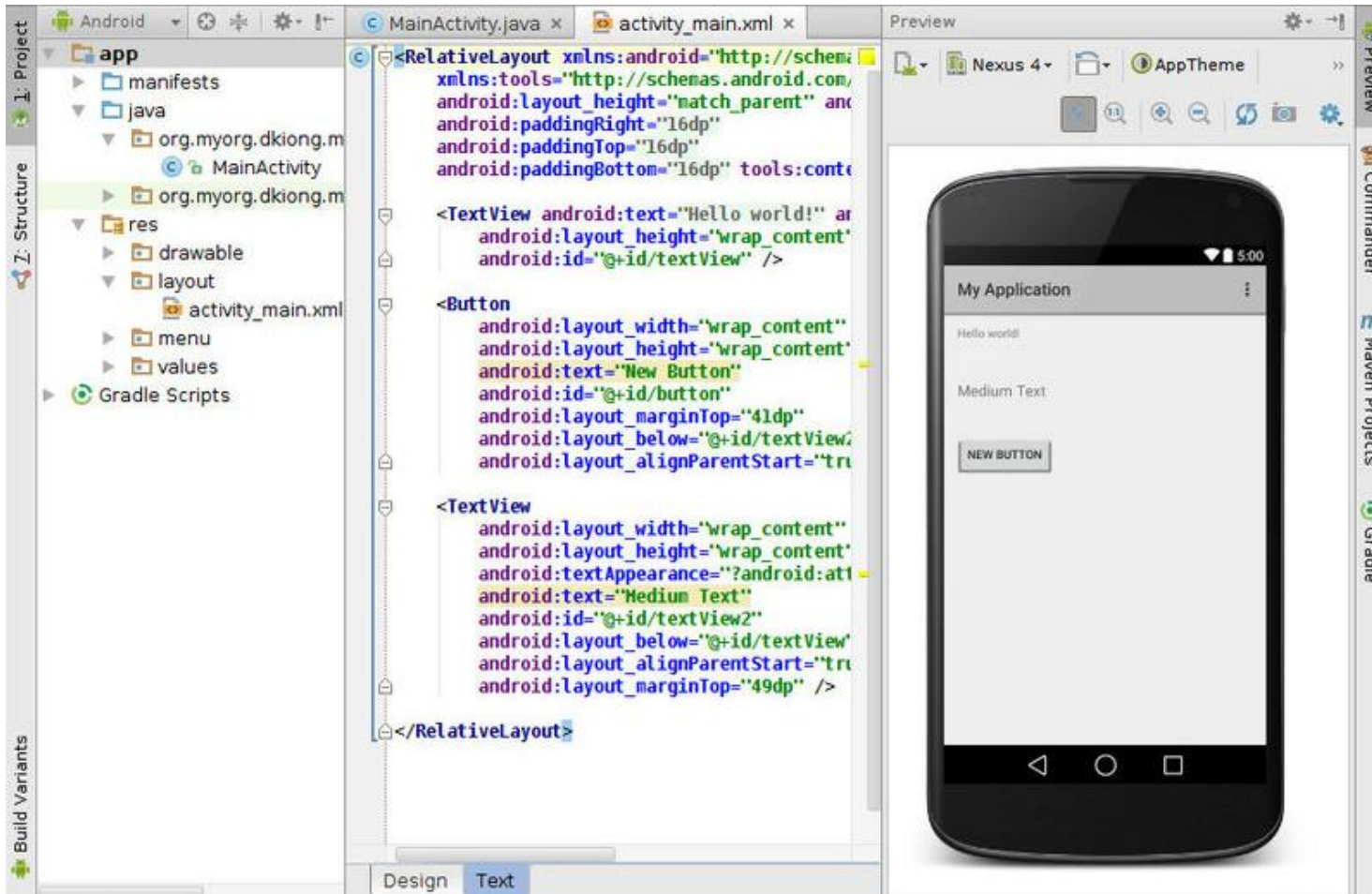
# Agenda

- Layouts
- Resource Files
- Permissions



# UI Component Layout in Activity

- Layout specified via XML files

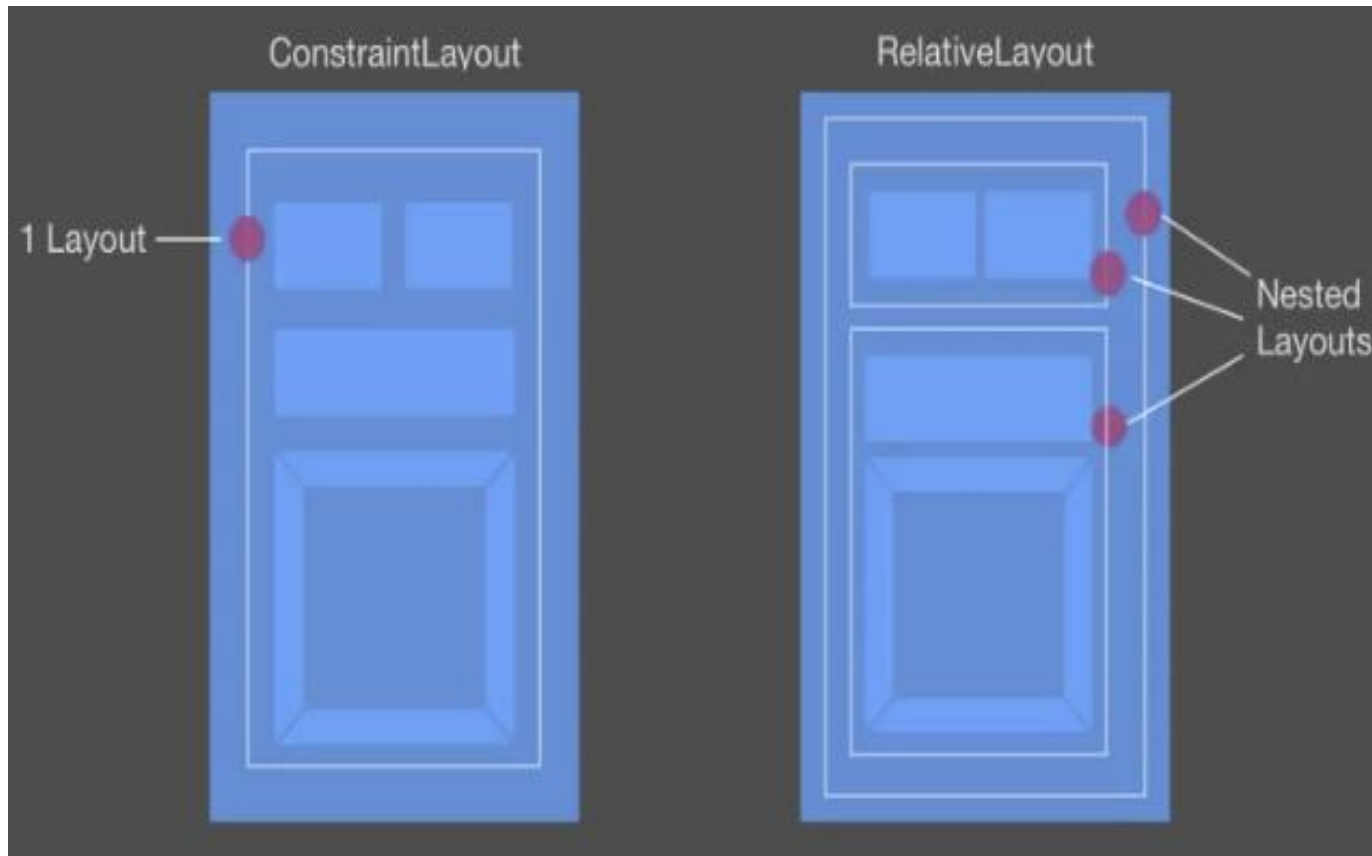




# Layout Managers

- Layouts organize child elements differently
  - **RelativeLayout** – child elements aligned relative to each other; has a nested view hierarchy
  - **ConstraintLayout** – similar to RelativeLayout but has a flat view hierarchy (recommended)
  - **LinearLayout** – with horizontal/vertical orientation so that child elements stacked in row/column
  - **FrameLayout** – stack child elements
  - **AbsoluteLayout** – child elements have exact [x,y] positions; non-responsive design (obsolete)

# Constraint vs Relative layout



# Android XML Namespaces

- Namespaces to resolve name-conflicts
- 3 key namespaces
  - **android** (android's platform attributes)
  - **app** (attributes declared in libraries)
  - **tools** (attributes that are visible in Design preview)

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  package="com.cherwah.datapersistence"
  tools:ignore="GoogleAppIndexingWarning">

  <application
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme">
    <activity
      android:name=".MainActivity"
      android:label="@string/app_name"
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
    </activity>
  </application>
</manifest>
```



# Loading a Layout

- Layout specification is loaded in **setContentView()** when an **Activity** is being created

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```



# Avoid hardwiring in code

- Resource files allow properties to be switched even at runtime
  - Separation of presentation from code
  - Localization
  - Screen rotation
- Landscape layout specified via
  - `res/layout-land/activity_main.xml`
  - Android will use this layout when device is in landscape mode





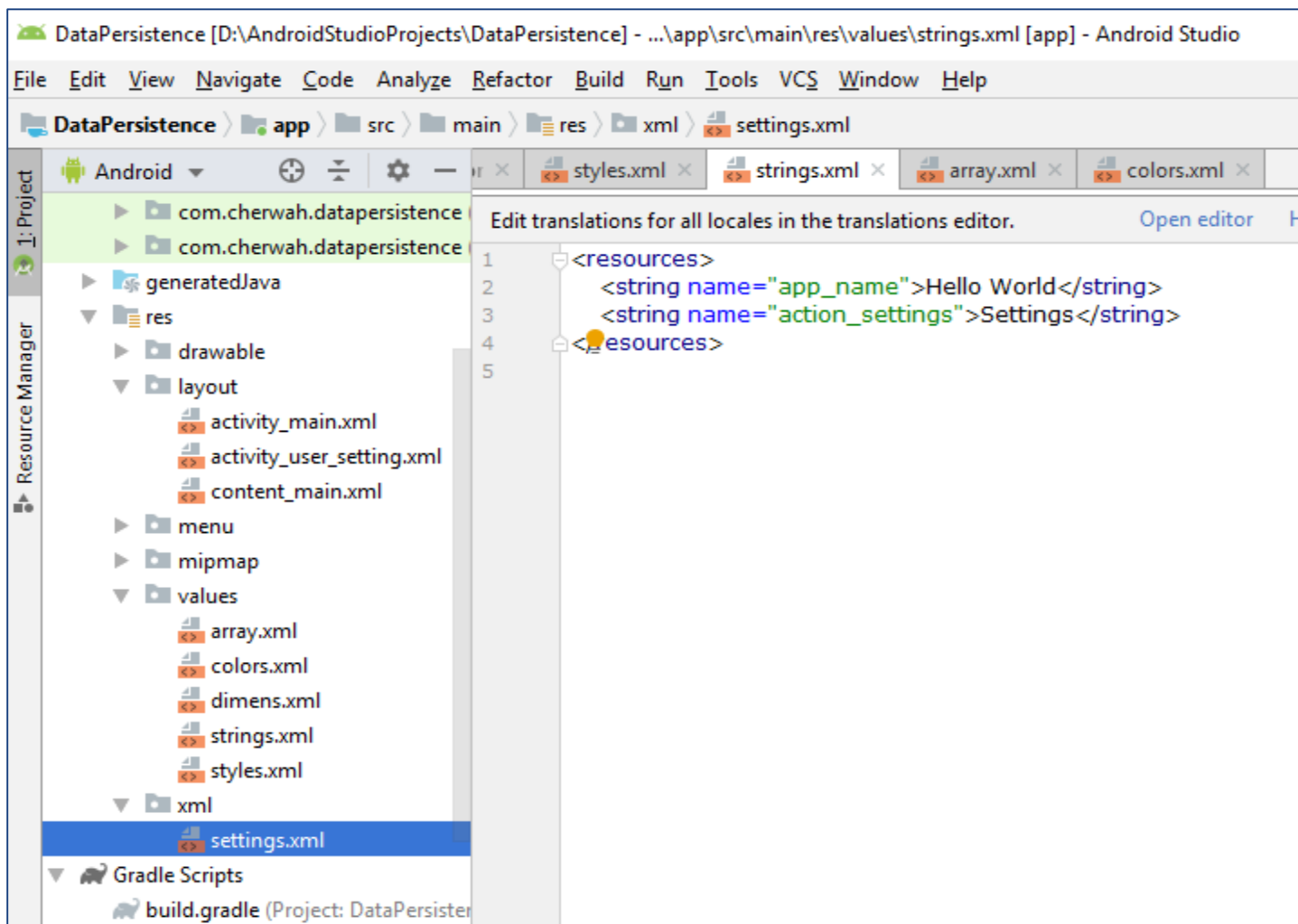
# Different Screen Sizes

- Layout maybe tailored for different screen sizes
  - res/layout/activity\_main.xml
  - res/layout-large/activity\_main.xml
  - res/layout-sw600dp/activity\_main.xml



# Constant Values in res

- Key-value associations in **res/values/strings.xml**



- Styles allow for consistency

```
<TextView  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:textColor="#00FF00" android:typeface="monospace"  
    android:text="@string/hello" />
```

- Neater style

```
<TextView  
    style="@style/CodeFont" android:text="@string/hello" />
```

```
<resources>  
    <style name="CodeFont" parent="@android:style/TextAppearance.Medium">  
        <item name="android:layout_width">fill_parent</item>  
        <item name="android:layout_height">wrap_content</item>  
        <item name="android:textColor">#00FF00</item>  
        <item name="android:typeface">monospace</item>  
    </style>  
</resources>
```

# View IDs => Java objects

- View in layout file with unique ID

```
<Button android:id="@+id/button1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/button1_text" />
```

- View-to-code mapping

```
Button button1 = (Button)findViewById(R.id.button1);
```



# Permissions

- To protect privacy of the Android user
- Android apps must request permission from user to access **data** (e.g. contacts) and **resources** (e.g. network)
- Different levels of permissions
  - Normal (accessed via **manifest** file)
  - Dangerous (accessed via **runtime** permissions)



# Permissions

- Refer to References slide for full list of permissions

Normal Permissions examples	
INTERNET	Allow applications to open network sockets
RECEIVE_BOOT_COMPLETED	Receive ACTION_BOOT_COMPLETED system broadcast
VIBRATE	Allow access to the vibrator

Dangerous Permissions examples	
WRITE_EXTERNAL_STORAGE	Allow writing to external storage
RECEIVE_SMS	Allow receiving of SMS messages
RECORD_AUDIO	Allow to record audio



# Normal Permissions

- Ask permissions via AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.iss.runtimepermissionexample">

    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>

    <application

        ...

    </application>

</manifest>
```



# Dangerous Permissions

- Execute code to explicitly ask for permissions

```
String[] permissions = new String[] {  
    Manifest.permission.WRITE_EXTERNAL_STORAGE  
};  
  
if (ContextCompat.checkSelfPermission(this, permissions[0]) == PackageManager.PERMISSION_GRANTED) {  
    writeToExtStg(data);  
}  
else {  
    ActivityCompat.requestPermissions(MainActivity.this,  
        permissions, WRITE_EXTERNAL_STORAGE_REQ);  
}
```

```
@Override  
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,  
                                       @NonNull int[] grantResults) {  
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
  
    switch (requestCode) {  
        case WRITE_EXTERNAL_STORAGE_REQ:  
            if (grantResults.length > 0) {  
                if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {  
                    writeToExtStg(data);  
                }  
            }  
            break;  
    }  
}
```





# References

- Layout Editor –  
<https://developer.android.com/studio/write/layout-editor>
- ConstraintLayout Tutorial for Android: Getting Started -  
<https://www.raywenderlich.com/9193-constraintlayout-tutorial-for-android-getting-started>
- Build a Responsive UI with ConstraintLayout -  
<https://developer.android.com/training/constraint-layout/index.html>
- Permissions -  
<https://developer.android.com/guide/topics/permissions/overview#normal-dangerous>



# Active Recall

