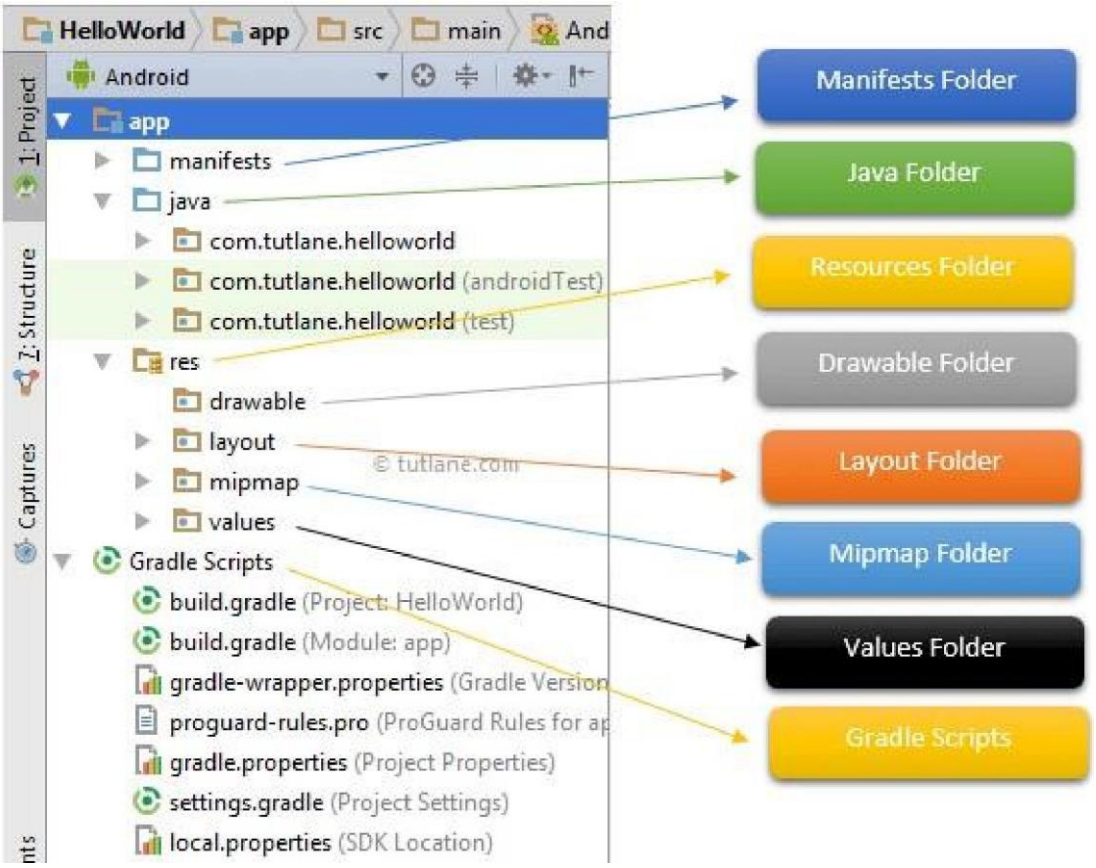
Lab 1
Installation of Android Studio, running first application and Understanding project folder structure

1. Understanding Android App Folder Structure

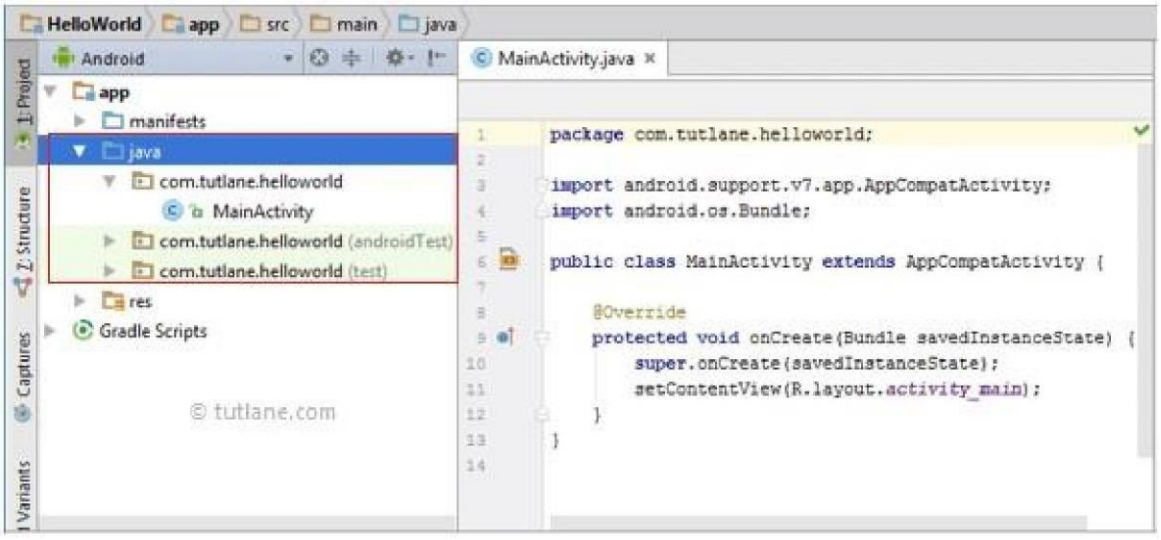


The Android project structure on the disk might be differs from the above representation. To see the actual file structure of the project, select **Project** from the **Project** dropdown instead of **Android**.

The android app project will contain different types of app modules, source code files, and resource files. We will explore all the folders and files in the android app.

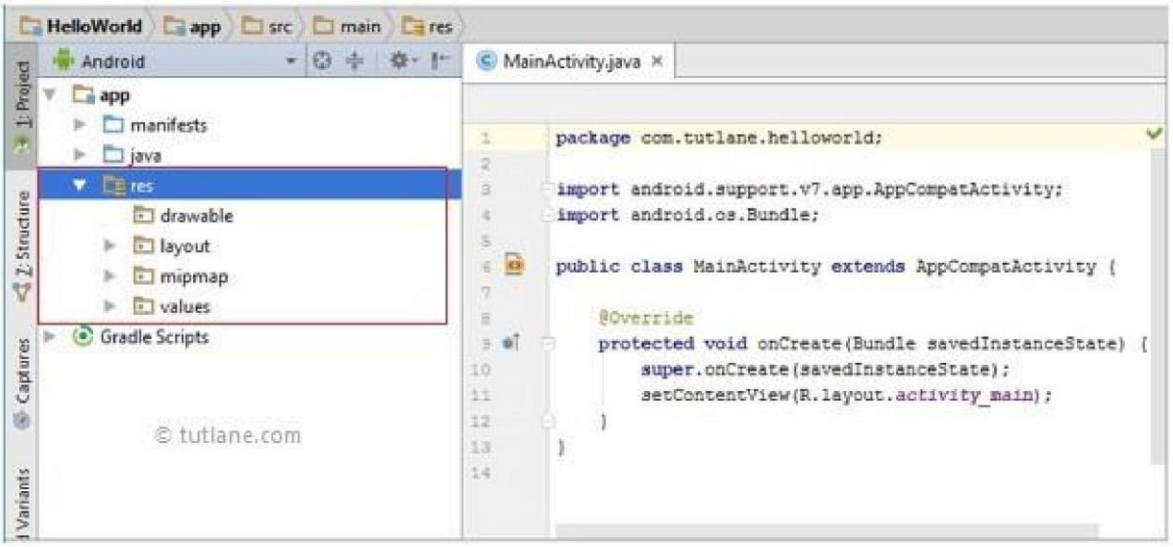
Java Folder

This folder will contain all the java source code (**.java**) files which we'll create during the application development, including JUnit test code. Whenever we create any new project/application, by default the class file **MainActivity.java** will create automatically under the package name "**com.tutlane.helloworld**" like as shown below.



res (Resources) Folder

It's an important folder that will contain all non-code resources, such as bitmap images, UI strings, XML layouts like as shown below.



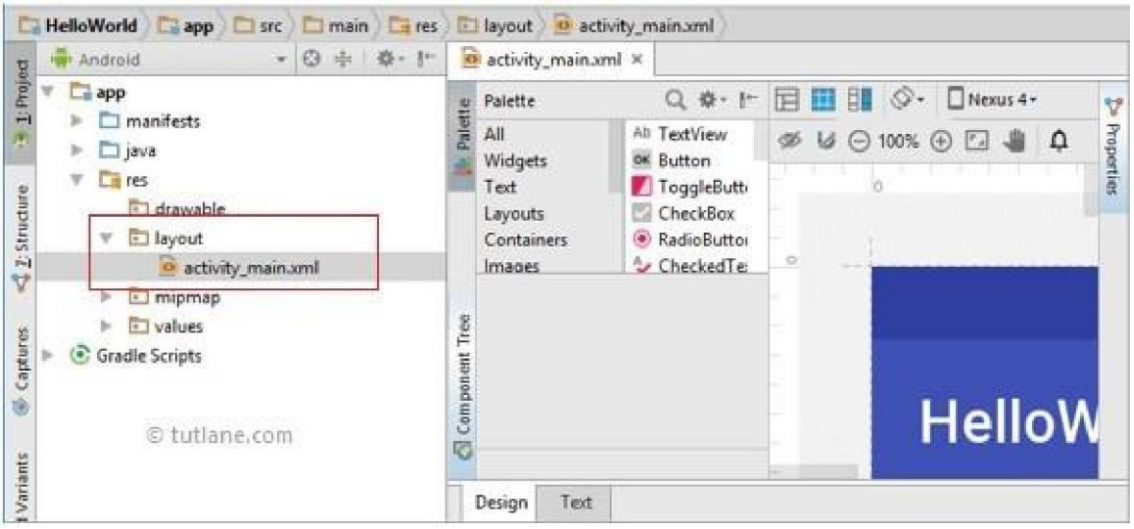
The res (**Resources**) will contain a different type of folders that are

Drawable Folder (res/drawable)

It will contain the different types of images as per the requirement of application. It's a best practice to add all the images in a **drawable** folder other than app/launcher icons for the application development.

Layout Folder (res/layout)

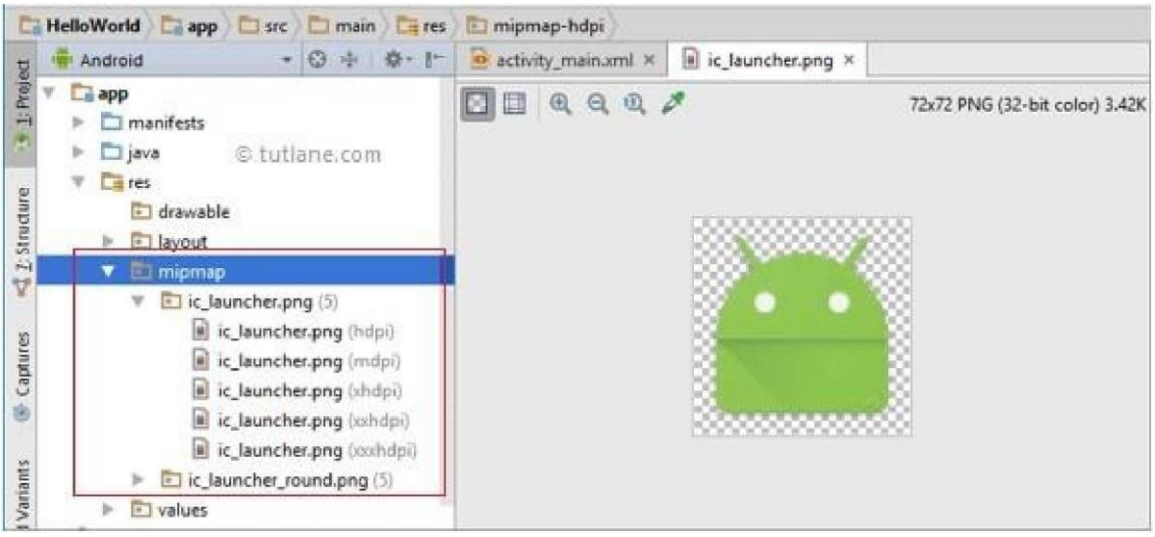
This folder will contain all XML layout files which we used to define the user interface of our application. Following is the structure of the **layout** folder in the android application.



Mipmap Folder (res/mipmap)

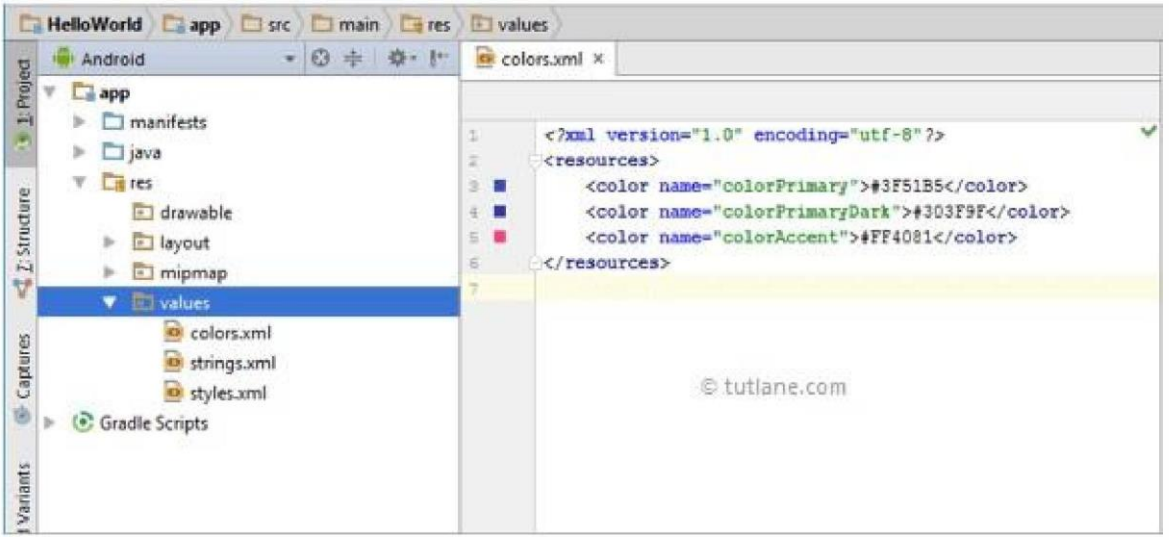
This folder will contain app / launcher icons that are used to show on the home screen. It will contain different density type of icons such as hdpi, mdpi, xhdpi, xxhdpi, xxxhdpi, to use different icons based on the size of the device.

Following is the structure of the **mipmap** folder in the android application.



Values Folder (res/values)

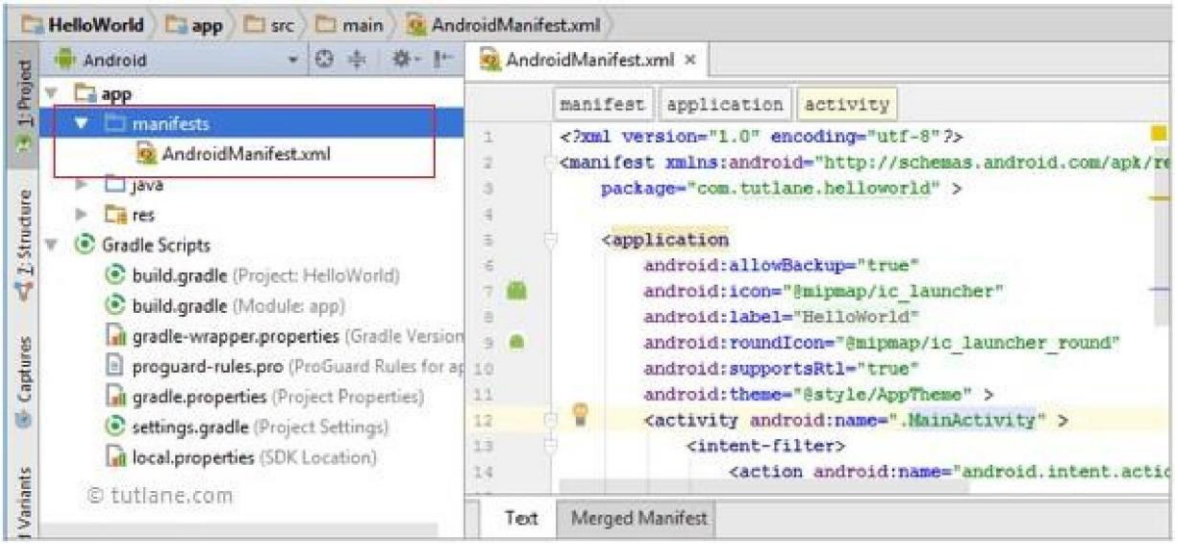
This folder will contain various XML files, such as strings, colors, style definitions and a static array of strings or integers. Following is the structure of the **values** folder in android application.



Manifests Folder

This folder will contain a manifest file (**AndroidManifest.xml**) for our android application. This manifest file will contain information about our application such as android version, access permissions, metadata, etc. of our application and its components. The manifest file will act as an intermediate between android OS and our application.

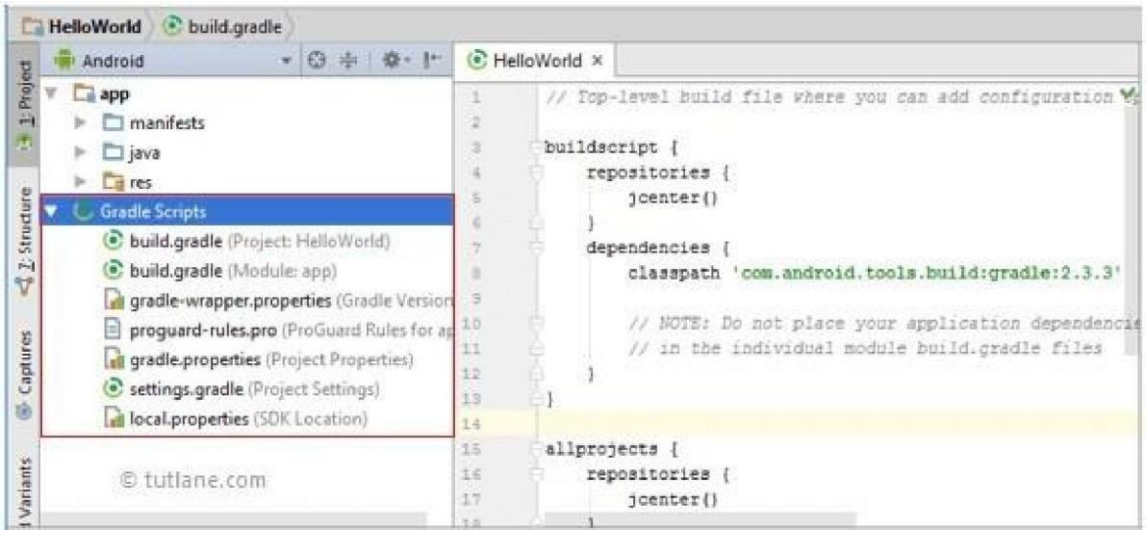
Following is the structure of the **manifests** folder in the android application.



Gradle Scripts

In android, Gradle means automated build system and by using this we can define a build configuration that applies to all modules in our application. In Gradle **build.gradle (Project)**, and **build.gradle (Module)** files are useful to build configurations that apply to all our app modules or specific to one app module.

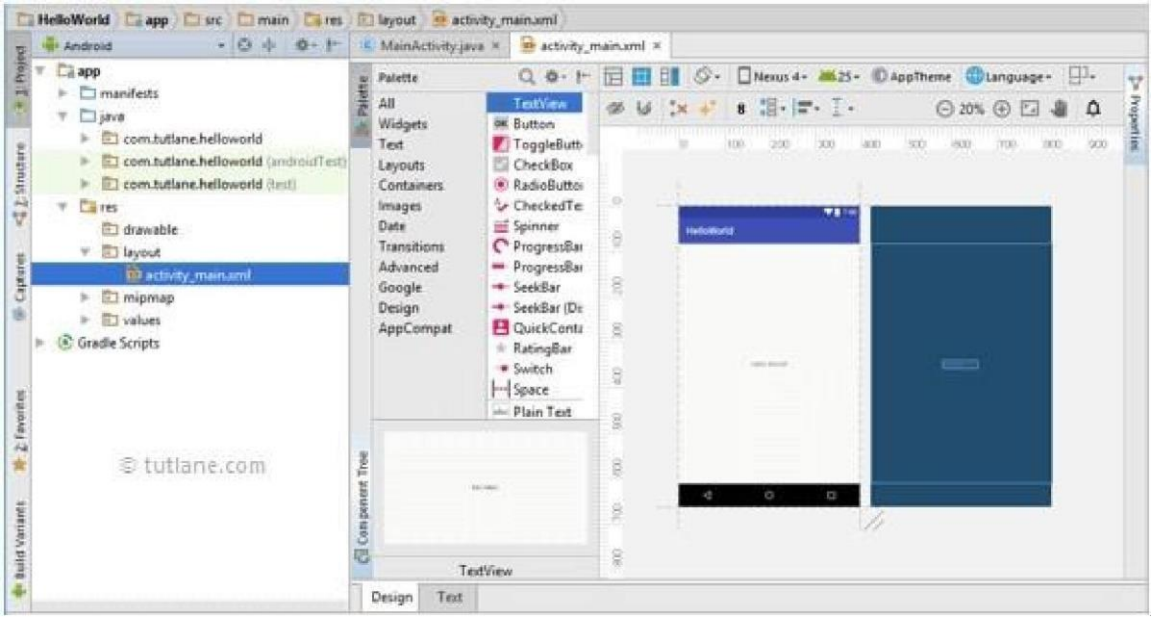
Following is the structure of **Gradle Scripts** in the android application.



Following are the important files which we need to implement an app in android studio.

- **Android Layout File (activity_main.xml)**

The UI of our application will be designed in this file and it will contain **Design** and **Text** modes. It will exists in the **layouts** folder and the structure of **activity_main.xml** file in **Design** mode like as shown below.



We can make required design modifications in **activity_main.xml** file either using **Design** or **Text** modes. If we switch to **Text** mode **activity_main.xml** file will contain a code like as shown below.

```
<?xml version="1.0" encoding="utf-8"?> <android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context="com.tutlane.helloworld.MainActivity">
   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Hello World!"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintLeft_toLeftOf="parent"
      app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent" />

    </android.support.constraint.ConstraintLayout>
```

Android Main Activity File (MainActivity.java)

The main activity file in the android application is **MainActivity.java** and it will exist in the **java** folder. The **MainActivity.java** file will contain the java code to handle all the activities related to our app.

Following is the default code of **MainActivity.java** file which is generated by our .HelloWorld application

```
package com.tutlane.helloworld;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
   }
     }
```

Android Manifest File (AndroidManifest.xml)

Generally, our application will contain multiple activities and we need to define all those activities in the **AndroidManifest.xml** file. In our manifest file, we need to mention the main activity for our app using the **MAIN** action and **LAUNCHER** category attributes in **intent filters** (<intent-filter>).

In case if we didn't mention MAIN action or LAUNCHER category for the main activity, our app icon will not appear in the home screen's list of apps.

Following is the default code of **AndroidManifest.xml** file which is generated by our **HelloWorld** application.

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

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

These are the main folders and files required to implement an application in android studio. If you want to see the actual file structure of the project, select **Project** from the **Project** dropdown instead of **Android**.

OUTPUT OF HELLO WORLD