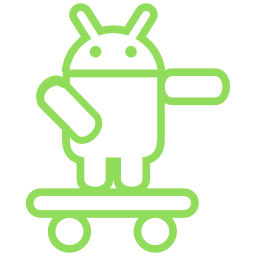
**1.1**

**Android Introduction**



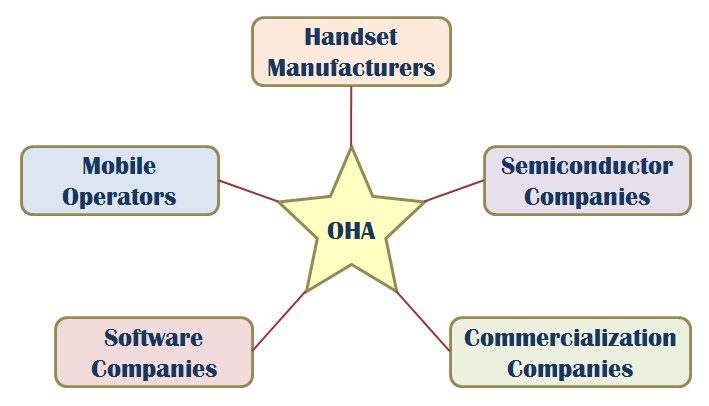
1. Define Android and Platforms.

**Android** is an open source operating system based on Linux with a Java programming interface for mobile devices such as Smartphone (Touch Screen Devices who supports Android OS) as well for Tablets too.

Any hardware or software environment in which a program runs, is known as a platform. Since Android has its own runtime environment (JRE) and API, it is called platform.

1. What is Open Handset Alliance (OHA) and what is its purpose?

The Open Handset Alliance (OHA) is a business alliance that was created for the purpose of developing open mobile device standards. The OHA has approximately 86 (2017)/84 (2020) member companies, including HTC, Dell, Intel, Motorola, Qualcomm and Google. The OHA's main product is the Android platform - the world's most popular smartphone platform.

****

The **Open Handset Alliance** (OHA) is consortium of multiple companies like Samsung, Sony, Intel and many more to provide a services and deploy handsets using android **platform.**

1. Mention some good platforms with example.

* Android – Open source mobile OS developed by the Open Handset Alliance led by Google. Based on Linux 2.6 kernel
* iOS – Apple’s proprietary mobile OS, iPhone, iPod Touch, iPad. Derived from OS X, very UNIX like
* Symbian – acquired by Nokia 2008
* Windows Phone 7 – Microsoft – Kin, discontinued 6 weeks after initial launch
* Blackberry OS – RIM (Research in Motion), proprietary OS

1. Mention some smartphone platforms.

Based on these features of

### 3G/4G/5G… connectivity

### WiFi connectivity

### Bluetooth connectivity

### Accelerometer w/compass

### Ambient light sensor

### Proximity sensor

### GPS

### Gyroscope

iOS, Symbian, Windows Phone-7, Blackberry OS are some smartphone platforms.

1. Make a list of Android features with their functions.

|  |  |
| --- | --- |
| **Sr. No.** | **Feature & Description** |
| 1 | **Beautiful UI**  Android OS basic screen provides a beautiful and intuitive user interface. |
| 2 | **Connectivity**  GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX. |
| 3 | **Storage**  SQLite, a lightweight relational database, is used for data storage purposes. |
| 4 | **Media support**  H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI,  Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP. |
| 5 | **Messaging**  SMS and MMS |
| 6 | **Web browser**  Based on the open-source WebKit layout engine, coupled with Chrome's V8 JavaScript engine supporting HTML5 and CSS3. |
| 7 | **Multi-touch**  Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero. |
| 8 | **Multi-tasking**  User can jump from one task to another and same time various application can run simultaneously. |
| 9 | **Resizable widgets**  Widgets are resizable, so users can expand them to show more content or shrink them to save space. |
| 10 | **Multi-Language**  Supports single direction and bi-directional text. |
| 11 | **GCM**  Google Cloud Messaging (GCM) is a service that lets developers send short message data to their users on Android devices, without needing a proprietary sync solution. |
| 12 | **Wi-Fi Direct**  A technology that lets apps discover and pair directly, over a high-bandwidth peer-to-peer connection. |
| 13 | **Android Beam**  A popular NFC-based technology that lets users instantly share, just by touching two NFC-enabled phones together. |

1. Graphically explain the Android features.



1. Make a list of Android version names, code name with their API levels.

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Release Date** | **Code Name** | **API Level** |
| 1.0 | 23 Sep, 2008 | Alpha/Apple Pie/NA | 1 |
| 1.1 | 9 Feb, 2009 | Banana Bread/NA | 2 |
| 1.5 | 30 Apr, 2009 | Cupcake | 3 |
| 1.6 | 15 Sep, 2009 | Donut | 4 |
| 2.0/2.1 | 26 Oct, 2009 | Éclair | 5-7 |
| 2.2 | 20 May, 2010 | Froyo | 8 |
| 2.3 | 6 Dec, 2010 | Gingerbread | 9-10 |
| 3.0/3.1/3.2 | 22 Feb, 2011 | Honeycomb | 11-13 |
| 4.0 | 18 Oct, 2011 | Ice Cream Sandwich | 14-15 |
| 4.1/4.2/4.3 | 9 Jul, 2012 | Jelly Bean | 16-18 |
| 4.4 | 31 Oct, 2013 | KitKat | 19-20 |
| 5.0/5.1 | 12 Nov, 2014 | Lollipop | 21-22 |
| 6.0 | 5 Oct, 2015 | Marshmallow | 23 |
| 7.0 | 2016 End | Nougat | 24 |
| 8.0 | 21 Aug, 2017 | Oreo | 26 |
| 9.0 | 6 Aug, 2018 | Pie | 28 |
| 10 | 23 Sept, 2019 | Android 10/ Quince tart | 29 |
| 11 |  | Android 11/Red Velvet Cake | 30 |
| 12 |  | Android 12/ Snow cone | 31 |
| 13 |  | Android 13/Tiramisu | 32 |
| 14 |  | Android 14/Upside Down Cake | 33 |

1. Write some applications of Android.

* Android applications are usually developed in the Java language using the Android Software Development Kit.
* Once developed, Android applications can be packaged easily and sold out either through a store such as
* Google Play,
* SlideME,
* Opera Mobile Store,
* Mobango,
* F-droid and
* Amazon Appstore

**Google Play:** Formerly known as the Android Market, is the official app store for Android smartphones and tablets. Google makes software applications, music, movies and books available for purchase and download through the store.

**SlideME:** A Community & Content Marketplace, uniting developers and users, offers products, services and experience that help promote small Android developers and their creative efforts, without locking them into any closed standards.

**Opera Mobile Store:** Offers a large number of applications for Android besides other *mobile* platforms.

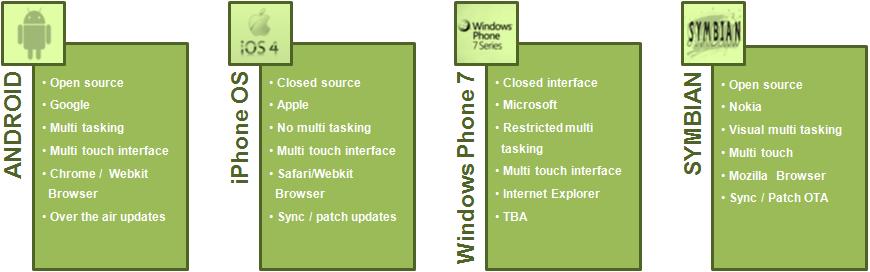
**MOBANGO:** A mobile community enabling mobile users to publish, convert and share user generated content with others.

***F*-*Droid*:** A software repository (or "app store") for Android applications, similar to the Google Play store.

**Amazon Appstore:** An app store for the Android operating system operated by Amazon.com.

1. What are the others Other Mobile OSS?

There are different other Mobile operating systems also present in market in competition with Android. Apple's iOS and Windows Phone give strong competition to Android. A simple comparison between features and specifications of latest version of Android and other Operating Systems can be seen in the table given below.



**Fig.** Various Mobile Operating System Available in Markets

1. Make a list of Mobile Devices and Mobile Companies for Android.

## Android Devices in Market

A device that runs Android OS comes in all shapes and sizes. Various devices that run Android OS and Apps are as follows:

* Smartphones
* Smart-watches
* Tablets
* E-reader Devices
* Netbooks
* MP4 Players
* Internet TVs and more.

**Android Mobile Company**

A large no of mobile companies are using Android. A list of companies supporting Android in their hardware are:

* + - Acer Inc
    - ALCATEL ( TCL corporation )
    - Bluelans Communications
    - NCE casio Mobile Communications
    - Cherry Mobile
    - CSL
    - Dell
    - Garmin
    - Geeks Phone
    - General Mobile
    - High screen
    - HKC
    - HTC coroporation
    - Huawei
    - I-mobile
    - Lenovo
    - LG
    - Motorola
    - Samsung
    - Sony Ericsson
    - Videocon
    - ZTE

1. Describe about Brick Model.

* It made calls, and there was a simple contacts application included in the operating system
* First marketed in 1983, it was 13 x 1.75 x 3.5 inches in dimension, weighed about 2.5 pounds, and allowed us to talk for a little more than half an hour.
* It retailed for $3,995, plus hefty monthly service fees and per-minute charges.
* The Motorola DynaTAC 8000X was the first commercially available cell phone.

EXTRA

Android was developed by the **Open Handset Alliance** (OHA), which is led by Google.

**OHA (Open Handset Alliance)**

### **The Smartphone Platform**

### With the iPhone being the first to the marketplace it sets the configuration of the Smartphone Platform

**Android Features**

Android is a powerful open source operating system which provides a lot of great features, those are

* It’s an open source and we can customize the OS based on our requirements.
* It support a connectivity for GSM, CDMA, WIFI, NFC, Bluetooth, etc. for telephony or data transfer. It will allow us to make or receive a calls / SMS messages and we can send or retrieve a data across mobile networks
* By using WIFI technology we can pair with other devices using apps
* Android have a multiple APIs to support a location-based services such as GPS
* We can perform all data storage related activities by using light weight database [SQLite](https://www.tutlane.com/tutorial/sqlite).
* It have a wide range of media supports like AVI, MKV, FLV, MPEG4 etc. to play or record variety of audio / video and having a different image formats like JPEG, PNG, GIF, BMP, MP3, etc.
* It has an extensive support for multimedia hardware control to perform playback or recording using camera and microphone
* It has an integrated open source Webkit layout based web browser to support HTML5, CSS3
* It supports a multi-tasking, we can move from one task window to another and multiple applications can run simultaneously
* It will give a chance to reuse the application components and the replacement of native applications.
* We can access the hardware components like Camera, GPS, and Accelerometer
* It has a support for 2D/3D/4G Graphics

### **Android Versions and API Levels**

* API (Application Program Interface) Level is an integer value that uniquely identifies the framework API revision offered by a version of the Android platform.
* Each Android version is assigned a unique integer identifier, called the *API Level*. Therefore, each Android version corresponds to a single Android API Level. Because users install apps on older as well as the most recent versions of Android, real-world Android apps must be designed to work with multiple Android API levels.

Since the release, Android has gone through number of updates. Here is the list of its different versions and their code names:

**Android History**

Initially Google launched a first version of Android platform on Nov 5, 2007 from that onwards Google released a lot of android versions under a codename based on desserts, such as Apple Pie, Banana Bread, Cupcake, Donut, Éclair, Froyo, Gingerbread, Jellybeans, Kitkat, Lollipop, marshmallow, etc. and made a lot of changes and additions to the android platform.

In **2007**, Google released a first beta version of the Android Software Development Kit (SDK) and the first commercial version of Android **1.0** (with name **Alpha**), was released in September **2008**.

In **2012**, Google released another version of android, 4.1 **Jelly Bean**. It’s an incremental update and it improved a lot in terms of user interface, functionality and performance.

In **2014**, Google announced another Latest Version, 5.0 **Lollipop**. In Lollipop version Google completely revamped the UI by using Material Designs, which is good for the User Interface as well for the themes related.

All the source code for Android is available free on Git-Hub, Stack overflow and many more websites. Google publishes most of the code under the Apache License version 2.0.

Following table shows the version details of android which is released by Google from 2007 to till date.

For each version Google has made a lot of changes and introduced lot of new features due to that the usage of android in mobile market increased drastically.

## Android 9.0 Popcorn (by Google)

Android 9.0 Pastry Android 9.0 Pasta

Android Pastilla Android 9.0 Puff

Android 9.0 Pandoro Android 9.0 Panna Cotta

Android 9.0 Parfait Android 9.0 Popover

Android 9.0 PanCake Android 9.0 Peanut Brittle

Android 9.0 Pumpkin Pie Android 9.0 Popsicle

Android 9.0 Pecan Pie Android 9.0 Poached Pears

Android 9.0 Praline Android 9.0 Pastille

Android 9.0 Petit Four Android 9.0 Pinka

Android Piano Android Pillow

Android Pilot Android Pudding

Android Pie

**Other suggested Android 9.0 version**

* Penuche
* Pignolo
* Pizzelle
* Pecan Pie
* Pavlova
* Profiterole
* Pop Tart

**Android P Indian Names**

* Android 9.0 Peda
* Android Pista
* Android Pilu

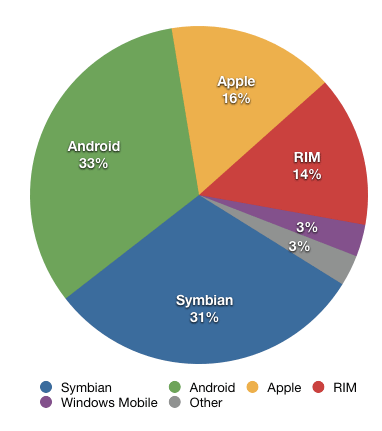
**Key additions of Different Versions**

|  |  |  |
| --- | --- | --- |
| **No** | **Code Name/Version** | **Key Additions/Features** |
| 1. | **No Codename** (v1.0) | Google Android 1.0 officially arrived September 23rd, 2008, as the only release of Android (so far) to not include a codename |
| 2. | **Petit Four** (v1.1) | "Petit Four" began rolling out in early 2009 as the first update for the new Google Android mobile operating system. |
| 3. | [Cupcake](https://www.webopedia.com/TERM/C/Cupcake.html) (v1.5) | Speech recognition tools, a virtual keyboard, video upload support for YouTube and support for live data feeds and live folders. |
| 4. | [Donut](https://www.webopedia.com/TERM/D/donut.html) (v1.6) | Support for CDMA smartphones, additional screen sizes and a text-to-speech engine. |
| 5. | [Eclair](https://www.webopedia.com/TERM/E/eclair.html) (v2.0) | Support for multi-touch devices, new browser interface, Microsoft Exchange support, single interface for managing multiple online accounts, soft keys support, and an enhanced camera app (with digital zoom and flash support). |
| 6. | [FroYo](https://www.webopedia.com/TERM/F/froyo.html) (v2.2) | USB [tethering](https://www.webopedia.com/TERM/T/tethering.html) support (for turning a smartphone into a [Wi-Fi](https://www.webopedia.com/TERM/W/Wi_Fi.html)hotspot), significant speed improvements, Flash 10.1 support, voice dialing over Bluetooth, the ability to store apps on external memory cards, updated browser with Google Chrome's V8 JavaScript. |
| 7. | [Gingerbread](https://www.webopedia.com/TERM/G/gingerbread.html) (v2.3) | Google Voice over Wi-Fi, enhanced gaming functionality, improved Google Apps. |
| 8. | [Honeycomb](https://www.webopedia.com/TERM/A/android_honeycomb.html) (v3.0) | A tablet-centric update that delivered a new interface optimized for devices with larger screen sizes (particularly [tablets](https://www.webopedia.com/TERM/T/tablet_PC.html)), video chat support based on Google Talk protocols, new System Bar for global status and notifications and Action Bar for application control, [tabbed Web browsing](https://www.webopedia.com/TERM/T/tabbed_browsing.html), optimized soft keyboard and a new email interface. |
| 9. | [Ice Cream Sandwich](https://www.webopedia.com/TERM/I/ice_cream_sandwich.html) (v4.0) | A smartphone-centric update based on the [Linux](https://www.webopedia.com/TERM/L/Linux.html) kernel v3.0.1 that brings many of Honeycomb's features to smartphones, including Face Unlock [facial recognition](https://www.webopedia.com/TERM/F/face_recognition.html) software, tabbed Web browsing capabilities, unified social networking contacts, 1080p video recording capabilities and video chat support based on Google Talk protocols. |
| 10. | [Jelly Bean](https://www.webopedia.com/TERM/J/jelly_bean.html) (v4.1, v4.2 and v4.3) | Advanced natural language voice command capabilities akin to Apple's [Siri](https://www.webopedia.com/TERM/S/siri.html), enhanced interface and overall responsiveness via "[Project Butter](https://www.webopedia.com/TERM/G/google_project_butter.html)," Google Now support, an improved Web browser, enhanced file management capabilities and more. |
| 11. | [KitKat](https://www.webopedia.com/TERM/K/kitkat.html) (v4.4) | Full-screen immersive mode, new transitions framework, and "Project Svelte," a project initiated to reduce the memory needs of the Android OS. Originally internally referred to as [Key Lime Pie](https://www.webopedia.com/TERM/K/key_lime_pie.html), Google announced in early September 2013 that it would be using the iconic candy bar as the codename for the 4.4 Android release. |
| 12. | [Lollipop](https://www.webopedia.com/TERM/L/lollipop.html) (v5.0) | Enhanced Material Design user interface, improved continuity across Android devices, multiple user support, a guest user account option, a new notification system, support for 64-bit CPUs, and more. |
| 13. | [Marshmallow](https://www.webopedia.com/TERM/M/marshmallow.html) (v6.0) | Now on Tap functionality and other Google Now enhancements, native fingerprint authentication support, Android Pay integration, USB Type-C support, improved battery life, better app management and more. |
| 14. | [Oreo](https://www.webopedia.com/TERM/O/oreo.html)  (v8.0) | Android Go minimized version of Android mobile OS, picture-in-picture video support, Notification Grouping, Bluetooth 5 support, Wi-Fi Aware feature, and more. |
| 15. | [Pie](https://www.webopedia.com/TERM/P/pie.html) (v9.0) | Adaptive Battery, Digital Well-being Dashboard (monitors screen time usage), App Slices, Intuitive Gestures, Adaptive Brightness, and more. |
| 16 | Android 10 | Dark mode, QR Code for WiFi, Audio features, Thermal API, Settings Panels, Bubbles, Location permissions, Privacy, Foldable devices, and more. |
|  | Android 11 |  |

# Android Applications

## Other Mobile OSs

 According to Canalys, In Q2 2009 Android had 2.8% market share which had grown to 33% market share by Q4 2010 which made Android leader of smart phone OSs worldwide. The market share for commonly used mobile OSs is shown in the following pie chart.

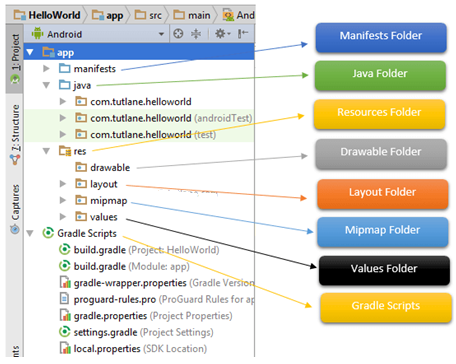


# Fig. Diagram Showing Market Share for Commonly Used Mobile Oss

**1.12**

**Android App/Project Folder Structure**

1. Sketch the Android App/Project Folder Structure.



1. What are the main folders and files required to implement an application in android studio?

FOLDERS

Java Folder

res/ Resources Folder

Drawable Folder (res/drawable)

Layout Folder (res/layout)

Mipmap Folder (res/mipmap)

Values Folder (res/values)

Mainfest Folder

Gradle Scripts

FILES

Android Layout File (activity\_main.xml)

Android Main Activity File (MainActivity.java)

Android Manifest File (AndroidManifest.xml)

1. What is for res (Resources) folder?

It’s an important folder which contains all non-code resources, such as bitmap images, UI strings, XML layouts. It contains total 4 folders which are: Drawable Folder (res/drawable), Layout Folder (res/layout), Mipmap Folder (res/mipmap) & Values Folder (res/values)

1. What is for Drawable folder (res/drawable)?

It contains the different type of images as per the requirement of application. It’s a best practice to add all the images in drawable folder other than app / launcher icons for the application development.

1. What is for Mipmap folder (res/mipmap)?

This folder contains app / launcher icons which 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 device. (ic\_launcher.png in different formats figure)

1. What is for Values folder (res/values)?

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

It contains colors, strings, styles file in xml format.

1. What is for Gradle Scripts in Android?

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

1. What is for Manifests folder?

This folder contains 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.

1. What is for Android Main Activity File (MainActivity.java)?

The main activity file in android application is MainActivity.java and it exists in java folder. The MainActivity.java file contains the java code to handle all the activities related to our app.

1. What is for 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 exists in layouts folder and the structure of activity\_main.xml file in Design mode. We can make a required design modification in activity\_main.xml file either using Design or Text modes.

Extra

To implement android apps, **Android Studio** is the official IDE (Integrated Development Environment) which is freely provided by Google for android app development.

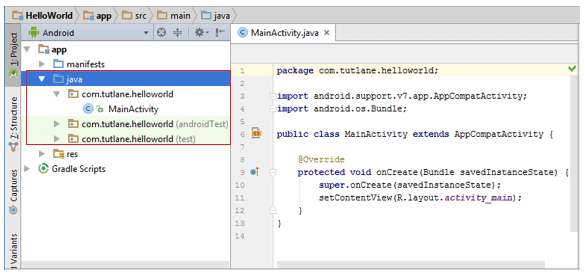
Once we [setup android development environment](https://www.tutlane.com/tutorial/android/android-development-environment-setup) using android studio and if we create a sample application using android studio, our project folder structure will be like as shown below.

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 a different type of app modules, source code files and resource files. We will explore all the folders and files in android app.

**Java Folder**

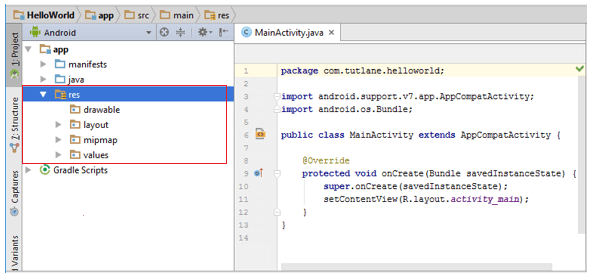
This folder contains all the java source code (**.java**) files which we’ll create during the application development, including JUnit test code. Whenever we create any new project / application, by default the class file **MainActivity.java** will create automatically under the package name “**com.mms.helloworld**”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



**Fig. 1.12.2:** Android Java Folder Structure.

**res (Resources) Folder**

It’s an important folder which contains all non-code resources, such as bitmap images, UI strings, XML layouts like as show below.

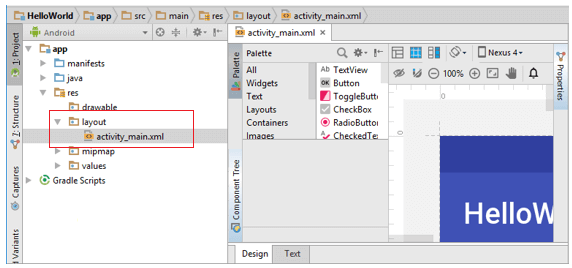


**Fig. 1.12.3:** Android res (Resources) Folder.

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

**Layout Folder (res/layout)**

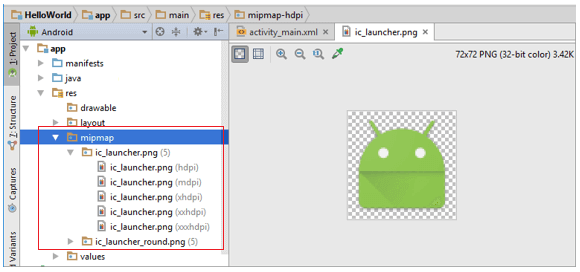
This folder contains all XML layout files which we used to define the user Interface of our application. Following is the structure of **layout** folder in android application.



**Fig. 1.12.4:** Android Layout Folder (res/layout)

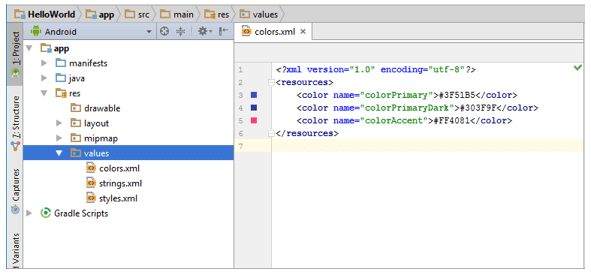
**Mipmap Folder (res/mipmap)**

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



**Fig. 1.12.5:** Android Mipmap Folder (res/mipmap)

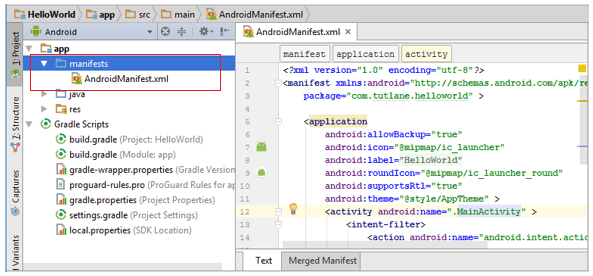
**Values Folder (res/values)**



**Fig. 1.12.6:** Android Values Folder (res/values)

**Manifests Folder**

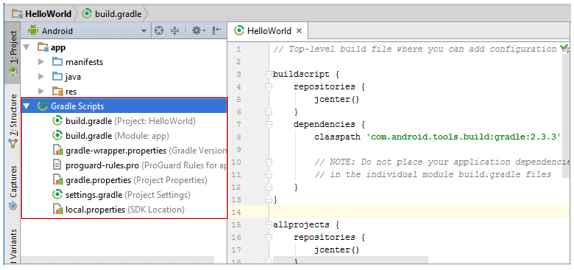
Following is the structure of **mainfests** folder in android application.



**Fig. 1.12.7:** Android Manifests Folder

**Gradle Scripts**

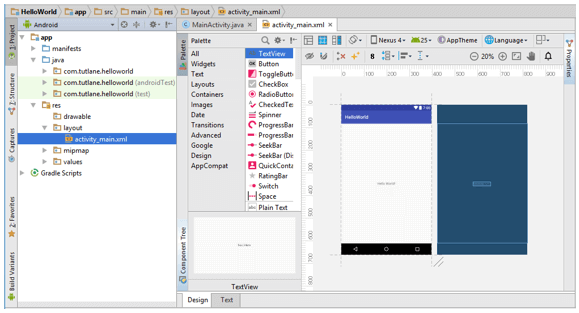
Following is the structure of **Gradle Scripts** in android application.



**Fig. 1.12.8:** Android Gradle Scripts

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

**Android Layout File (activity\_main.xml)**



**Fig. 1.12.9:** Android Layout File (activity\_main.xml)

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="sarker.com.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)**

Following is the default code of **MainActivity.java** file which is generated by our [HelloWorld application](https://www.tutlane.com/tutorial/android/android-hello-world-app-example).

package com.sarker.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](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) and we need define all those [activities](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) in **AndroidManifest.xml** file. In our manifest file we need to mention the main activity for our app using **MAIN** action and **LAUNCHER** category attributes in **intent filters** (<intent-filter>).

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.sarker.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.

**1.2**

**Android Architecture**

# Draw the Android Architecture (Software Stack of Android) and explain the roles and features of each layers.

# Android architecture is a software stack of components to support a mobile device needs. Android software stack contains a Linux Kernel, collection of c/c++ libraries which is exposed through an application framework services, runtime and application

# Android Architecture Diagram with Multiple Components

# 

1. How many layers are in Android Architecture? Explain the roles and features of each layers.

Following are main components/layers of android architecture those are

1. Applications
2. Android Framework
3. Android Runtime
4. Platform Libraries
5. Linux Kernel

In these components **Linux Kernel** is the main component in android to provide its operating system functions to mobile and **Dalvik Virutal Machine** (**DVM**) which is responsible for running a mobile application.

[roles & features included in q 3-5]

1. What are the roles and features of Linux Kernel, Hardware Abstraction Layer (HAL) and Applications?

Linux Kernel is a bottom layer and heart of the android architecture. It manage all the drivers such as display drivers, camera drivers, Bluetooth drivers, audio drivers, memory drivers, etc. which are mainly required for the android device during the runtime.

The Linux Kernel will provides an abstraction layer between the device hardware and the remainder of the stack. It is responsible for memory management, power management, device management, resource access, etc.

The top layer of android architecture is **Applications**. The native and third party applications like contacts, email, music, gallery, clock, games, etc. whatever we will built those will be installed on this layer only.

The application layer runs within the Android run time using the classes and services made available from the application framework.

1. What are the roles and features of Android Runtime (ART)?

**Android Runtime** environment is an important part of Android rather than an internal part and it contains a component like **core libraries** and the **Dalvik virtual machine**. The Android run time is the engine that powers our applications along with the libraries and it forms the basis for the application framework. The **core libraries** in android runtime will enable us to implement an android applications using standard JAVA programming language.

[[**Dalvik Virtual Machine** (**DVM**) is a register-based virtual machine like Java Virtual Machine (JVM). It is specially designed and optimized for android to ensure that a device can run multiple instances efficiently. It relies on the Linux kernel for threading and low-level memory management.]]

1. What are the roles and features of Android Libraries?

The Android **Platform Libraries** includes various C/C++ core libraries and Java based libraries such as SSL, libc, Graphics, SQLite, Webkit, Media, Surface Manger, OpenGL etc. to provide a support for android development. (….. support)

Following are the summary details of some core android libraries available for android development. (… features)

* Media library for playing and recording an audio and video formats
* The Surface manager library to provide a display management
* SGL and OpenGL Graphics libraries for 2D and 3D graphics
* SQLite is for database support and FreeType for font support
* Web-Kit for web browser support and SSL for Internet security.

1. What are the roles and features of Android Application Framework?

The **Application Framework** provides the classes used to create an Android applications. It also provides a generic abstraction for hardware access and manages the user interface and application resources. It basically provides the services through which we can create the particular class and make that class helpful for the Applications creation.

The application framework includes services like telephony service, location services, notification manager, NFC service, view system, etc. which we can use for application development as per our requirements.

**1.3**

**Android Setup and Installation**

1. How many ways can be setup android development environment and what are the different components of Android Studio?

We can setup android development environment using following two ways

1. Setup Eclipse IDE Manually (**Depreciated**)
2. Android Studio

Android Studio is the combination of following components to allow users to implement android applications.

1. Eclipse IDE
2. Android SDK
3. Android Virtual Device
4. Eclipse Plugin

EXTRA

Generally to build an applications for Android we should have Java Development Kit (JDK), Android SDK, and a development environment.

The Android SDK is compatible with Windows, Mac and Linux operating systems to build android applications based on our requirements.

Initially Google supported a Manual **Eclipse IDE Setup** for android development environment by downloading required components like Eclipse IDE, Android SDK, Java Development Kit (JDK) etc. from official site. Afterwards Google introduced a component called **Android Studio** to make environment setup process simple.

By using **Android Studio** bundle we can easily setup android development environment in any operating system to implement android applications.

**Installation**

Initially to setup an android development environment in our system we need to install following components manually by downloading from different sites.

1. Eclipse IDE
2. Eclipse Plugin
3. Android SDK

To make android development environment setup process simple Google introduced a new android IDE called **Android Studio**. The **Android Studio** will contain all the required components like Eclipse IDE, Eclipse Plugin and Android SDK so we don’t need to download the components separately.

**Android Studio** is the official IDE for android development and it’s based on **IntelliJ IDEA** software. It’s available for Windows, MAC and LINUX operating systems.

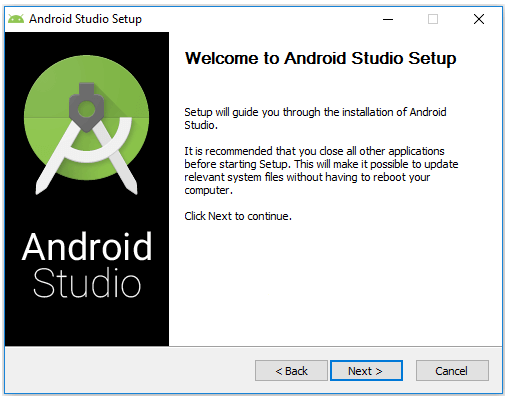
We can download latest version of Android Studio from following URL.

[**Download Android Studio**](https://developer.android.com/studio/index.html)

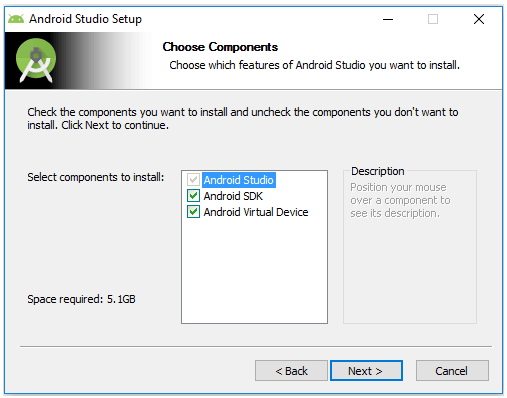
In this tutorial we are going to explain how to install android studio on windows machine which is having windows 10 operating system.

Download the latest version of Android Studio from above URL and launch **Android Studio.exe** file by double clicking on it.

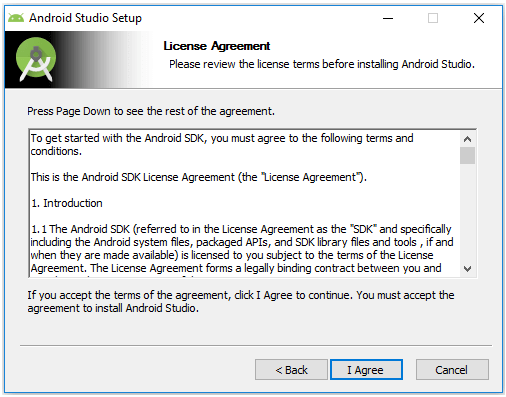
The initial android studio setup screen will open like as shown below in that click **Next** to continue for further steps of environment setup.



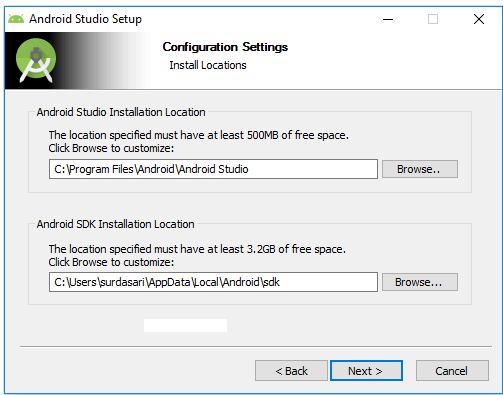
Now we need to select a required components to setup an android environment. Here we selected all three components (**Android Studio**, **Android SDK** and **Android Virtual Device**) and click **Next** like as shown below.



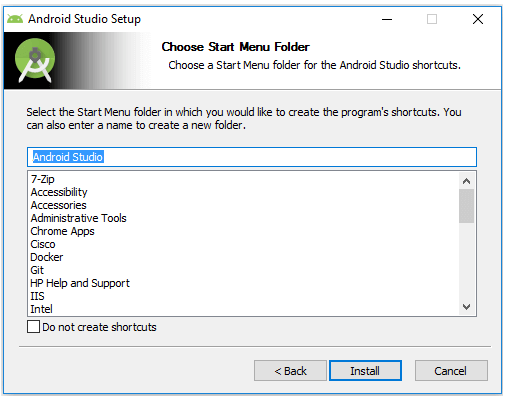
Now we need to agree the License agreements to proceed further, click on **I Agree** button like a shown below.



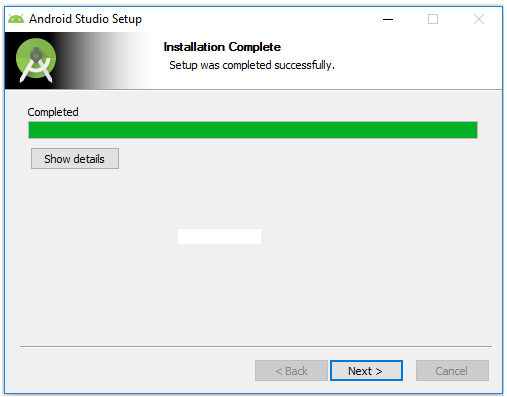
Now we need to specify the local machine drive location to install Android Studio and Android SDK. After selecting the location path to install required components, click **Next** like as shown below.



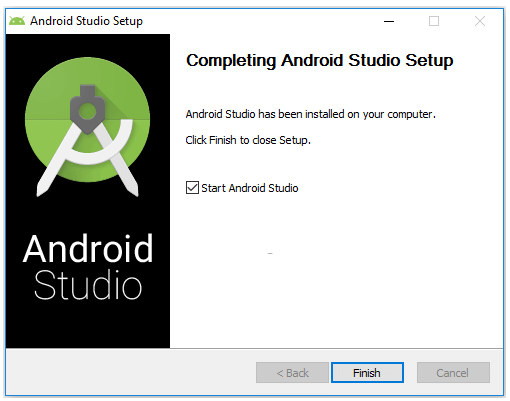
 Now select the start menu folder to create a shortcuts for android studio and click **Install** like as shown below.



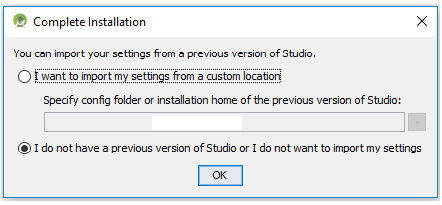
Once we click Install button the installation process will start and click **Next** after completion of installation like as shown below.



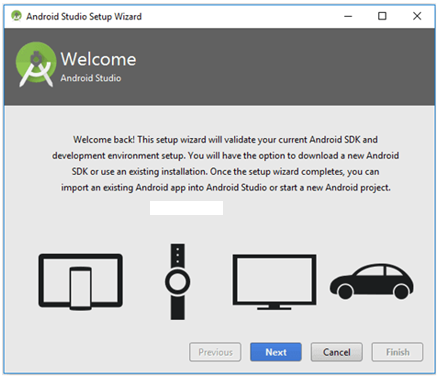
After that it will show installation completion wizard in that click **Finish** to launch android studio like as shown below.



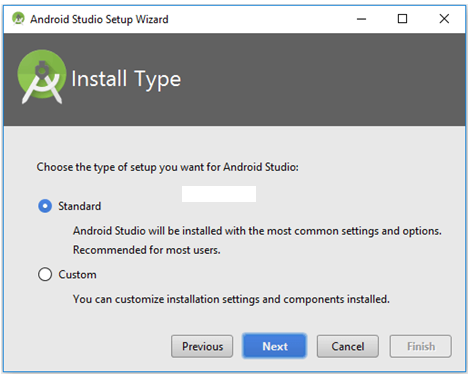
While launching **Android Studio** it will give us an option to import settings from previous version of studio. In case if we don’t have any previous version, select second option and click **OK** like as shown below.



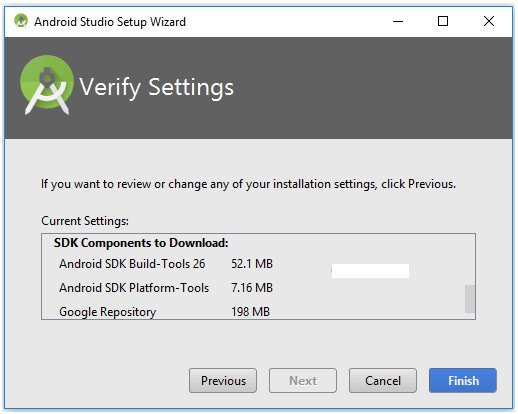
 Now android studio will open a welcome wizard window in that click **Next** to validate our current Android SDK and development environment setup like as shown below.



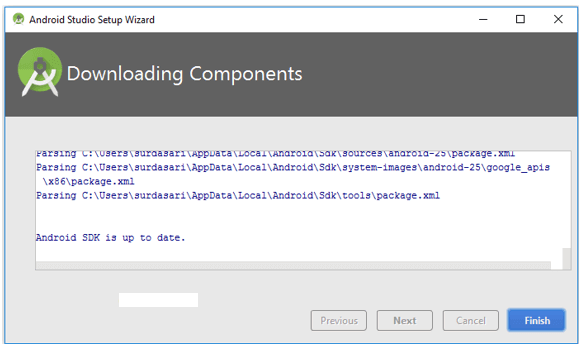
Now select a **Standard** installation type and click **Next** to install a common settings and options like as shown below.



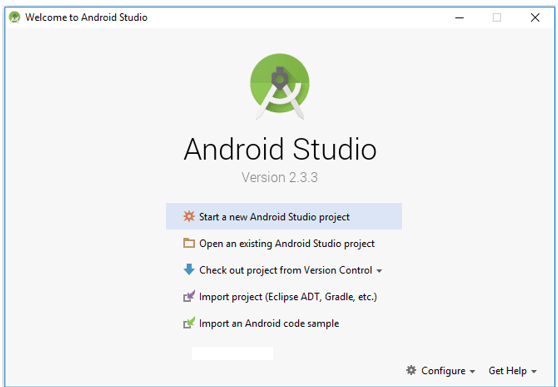
 Now verify settings and click **Finish** to complete android studio setup process like as shown below.



After completion of required components installation click on **Finish** like as shown below.



After completion of all required components installation we will be able to see Android Studio welcome window like as shown below.



This is how we can setup android development environment on windows machine which is having windows 10 operating system using android studio IDE.

**1.4**

**Android Hello World App Example**

1. What is for AndroidManifest.xml file? Write necessary code for AndroidManifest.xml.

Generally our application will contain multiple **activities** and we need define all those **activities** in **AndroidManifest.xml** file. In our manifest file we need to mention the main activity for our app using **MAIN** action and **LAUNCHER** category attributes in **intent filters** (<intent-filter>). In case if we didn’t mention MAIN action or LAUNCHER category for main activity, our app icon will not appear in home screen’s list of apps.

1. What is for MainActivity.java file? Write necessary code for MainActivity.java file.

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

1. What is for activity\_main.xml file? Write necessary code for activity\_main.xml file.

The UI of our application will be designed in this file and it will contain **Design** and **Text** modes. It will exists in **layouts** folder and the structure of **activity\_main.xml** file in **Design** mode . We can make a required design modifications in **activity\_main.xml** file either using **Design**

or **Text** modes.

1. What are the two modes of UI and why they are for?

**Design** and **Text** modes. We can make a required design modifications in **activity\_main.xml** file either using **Design** or **Text** modes.

1. Write the meaning of the following:
2. android:layout\_width="match\_parent"
3. android:layout\_height="match\_parent"

[The answer is taken from: Slide 5.2 Android UI Layouts]

|  |  |
| --- | --- |
| android:layout\_width | It is used to define the width for View and ViewGroup elements in layout |
| android:layout\_height | It is used to define the height for View and ViewGroup elements in layout |

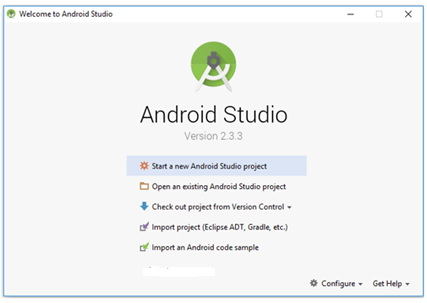
If we set value match\_parent, then the View or ViewGroup will try to match with parent width or height.

EXTRA

By using android studio IDE (Integrated Development Environment) we can implement required android apps based on our requirements.

To implement android hello world app first we need to setup development environment using android studio IDE which is freely provided by Google for android developers. In case if you are not aware of how to setup android development environment, follow step by step [Android Environment Setup](https://www.tutlane.com/tutorial/android/android-studio-installation-for-development-environment) tutorial.

Once we are done with [Android Studio Installation](https://www.tutlane.com/tutorial/android/android-studio-installation-for-development-environment), open android studio and that will be like as shown below.



Here we're going to choose the **New Project** option because we haven’t created any other project and we need to create a new one. So, we will select the New Project from the given options.

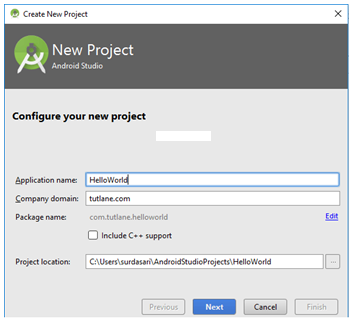
However, we can choose **Import Project** if we’d like to import a project from any other way, for example, Eclipse project into Android Studio. Android Studio will convert the Eclipse project to an Android Studio project, adding the necessary configuration files for us.

If we select **Open Project** from the list of options, we can open projects created with either Android Studio or IntelliJ IDEA.

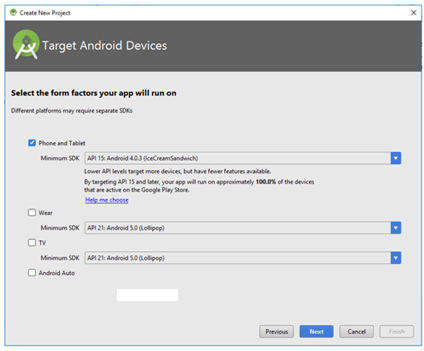
Check out from Version Control, we can check out a copy of a project that's under version control. This is a great way to quickly get up to speed with an existing project.

To get us started, choose **New Project** from the list of options. This will show us a list of options to configure our new project.

As we click on “**New Project**” from the above option, then the next screen will be open like this, where we have to mention our **Project’s name**, **Company domain** and **Project location** (we called it the main path where this application will be save) because the **Package name** will be created automatically as we create the project in Android Studio.

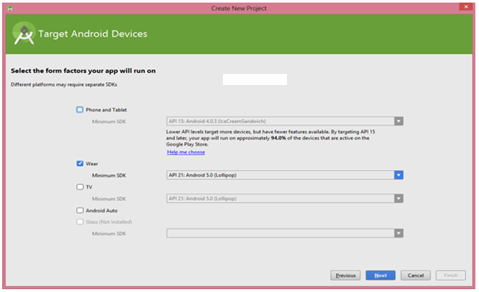


After entering all the details if we click on “**Next**” button another screen will be appear where we have select the different platforms and SDK targets like as shown below based on our requirements.

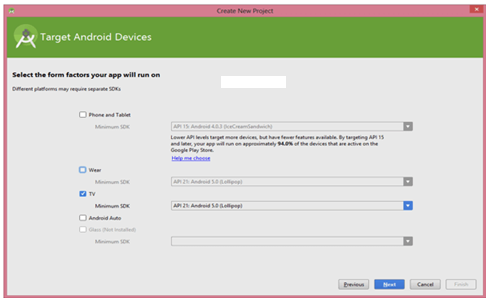


Here we need to select the type of Platform which we are going to use for the Application development like if we select “**Phone and Tablet**”, then it will show it’s different **API** and **SDK** version and similarly to others.

If we choose “**Wear**”, then it will show it’s API and SDK versions like as shown below.



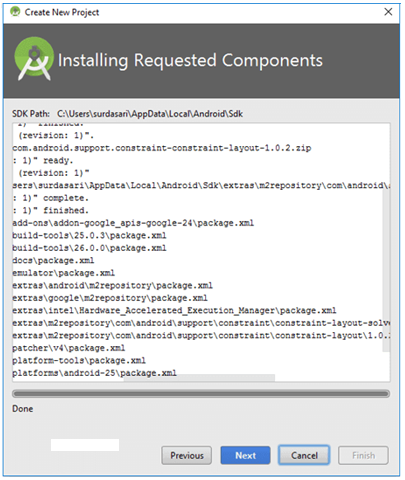
In case if we choose “**TV**”, then it will show it’s API and SDK versions like as shown below.



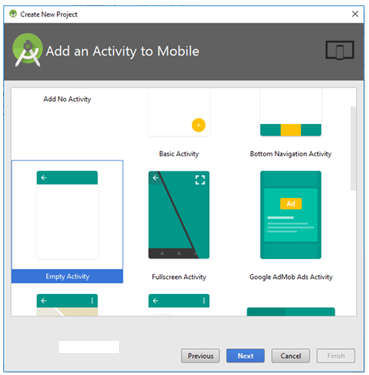
**Wear**: We use this option for **Android Watches** which we can wear to our hand and use the same functionality as we do with the Android devices. We can call, set the alarm, capture images and many more things easily.

**TV**: We use this option for **SmartIPTV** which is very much common these days. We can see our favorite channels like we see in our Home Televisions and make the changes in the channel easily.

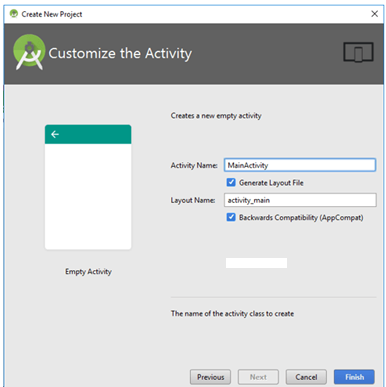
Here we are going to implement app for phone and tablets, so we selected a **Phone and Tablet** option and click “**Next**” and it will install required components like as shown below.



Now click **Next** to select the particular Activity for our requirement. If we will select the “**Empty Activity**”, then it will show the empty activity in our layout. In case if we choose other options, then it will show the activity which we have chosen. Here we are selecting **Empty Activity** like as shown below.

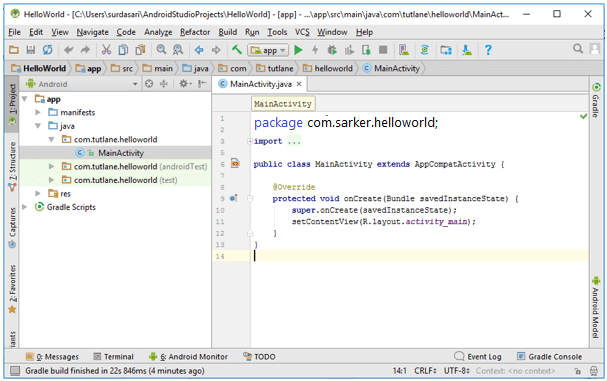


After choosing the “**Activity**” for our application, then click on the “**Next**” button and it will take us to the next screen like as shown below.



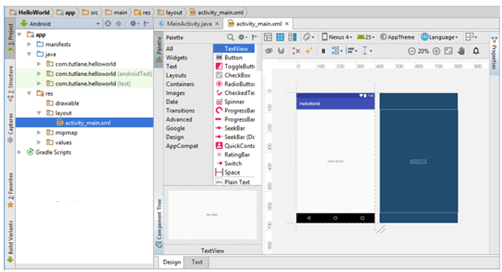
Here we can see that the Activity i.e. **EmptyActivity** which we selected in previous section and the java file name i.e. “**MainActivity**”. Now we are ready for final step, just click on “**Finish**” button and it will took us to the Main page where we have to do the coding and create new layouts over there.

After clicking **Finish**, we will be presented with Android Studio's user interface with the project explorer on the left and the workspace on the right like as shown below.



Following are the important files which we need to build our app in android studio.

**Android Layout File (activity\_main.xml)**



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.sarker.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)**

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

application.

package com.sarker.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);  
    }  
}

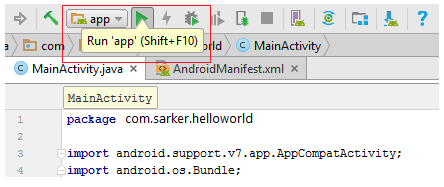
**Android Manifest File (AndroidManifest.xml)**

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

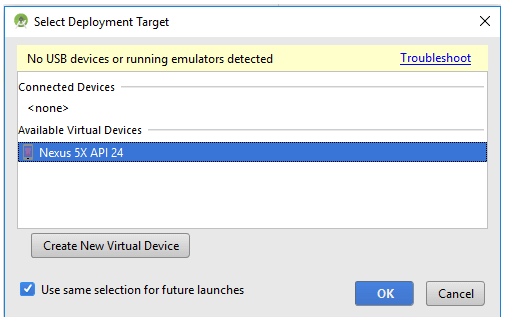
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.sarker.helloworld" >  
    <application  
        android:allowBackup="true"  
        android:icon="@mipmap/ic\_launcher"  
        android:label="@string/app\_name"  
        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>

**Run Android Hello World App**

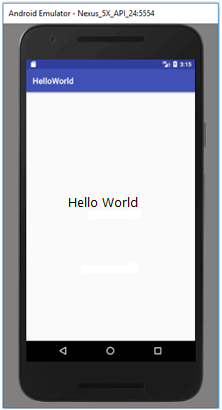
To run android applications we need to click on **Run** button or press Shift + F10 like as shown below:



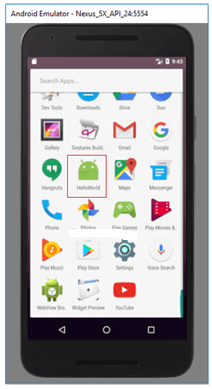
After click on play button new window will open in that select **Android Virtual Device** (AVD) and click **OK** like as shown below.



Now our android hello world application will show the result like as shown below



In our **AndroidManifest.xml** file we mentioned MAIN action and LAUNCHER category attributes for our main activity file due to that our app icon will create in Home screen list of apps like as shown below.



This is how we can create an apps in android and execute applications based on our requirements.

**1.5**

**Setup Emulator or Create AVD (Android Virtual Device)**

* 1. What is Android virtual device (AVD) and how it can be setup and created in Android Studio?

Android virtual device (AVD) is an emulator which is used to replicate the functionality of an android phone, tablet, android wear or TV to test our android applications locally. By using AVD manager interface in android studio we can setup android virtual device emulator to test our applications.

**Create Android Virtual Device**

To test our android application we should have an Android Virtual Device (AVD). We can create virtual device by click on **AVD Manager**.

When we click on **AVD Manager**, new window will open in that click on **Create Virtual Device.**

Now select the required device type and Click **Next** to create a virtual device.

Now we need to download and select the system image and click **Next.**

Now verify the configuration of **android virtual device** (AVD) and click **Finish.**

This is how we need to add android virtual device (AVD) in android studio to test our android applications.

* 1. What are the functionality of Android Virtual Device (AVD)?
* Once we are done with setup of android virtual device in android studio, create sample application in android studio and run the app using AVD manager. In case if we are not aware of creating an app,
* To run android applications we need to click on **Run** button or press Shift + F10
* After click on play button new window will open in that select **Android Virtual Device** (AVD) and click **OK**
* This is how we can setup android virtual device (AVD) emulator in android studio to replicate the functionality of real android devices.