**1)Explain in details: Emulator**

**🡪**An Android Emulator is a device that simulates an Android device on your system. Suppose we want to run our android application that we code. One option is that we will run this on our Android Mobile by Enabling USB Debugging on our mobile. Another option is using Android Emulator.

In Android Emulator the virtual android device is shown on our system on which we run the Android application that we code.

Thus, it simply means that without needing any physical device Android SDK component “Android Emulator” provides a virtual device on the System where we run our Application.

The emulator’s come with the configuration for Various android phones, tablets, Wear OS, and Android TV devices.

**Advantages of Android Emulator**

* Transfer of files is faster in virtual devices, as here a file can be transferred through drag and drop.
* Emulator enables the programmer to work with physical sensors such as accelerometers, gyroscopes, proximity, etc.
* One can play games, browse the internet, change settings, etc.
* Android emulator enables you to choose any version of your choice and accordingly, a developer can build an app.
* Developer can check all possible test cases through android emulators.

**Disadvantages of Android Emulator**

* Android emulator works with slow speed as compared to actual physical devices.
* Emulators cannot recognize the speed and performance of the battery, location, and hardware-related activities.
* Testing on an emulator is not accurate as you are using a virtual device not an actual device.
* One cannot identify issues related to networks or notifications on an emulator.
* The emulator crashes at launch if it does not receive enough disk space.

2)Explain giving example : contax.

🡪Context: The context is the central command center for an Android application. All application-specific functionality can be accessed through the context.

Example of Context:

o Creating New objects: Creating new views, adapters, listeners

♣ TextView tv = new TextView(getContext());

♣ ListAdapter adapter = new SimpleCursorAdapter(getApplicationContext(), ...);

o Accessing Standard Common Resources: Services like LAYOUT\_INFLATER\_SERVICE, SharedPreferences:

♣ context.getSystemService(LAYOUT\_INFLATER\_SERVICE)

♣ getApplicationContext().getSharedPreferences(\*name\*, \*mode\*);

o Accessing Components Implicitly: Regarding content providers, broadcasts, intent

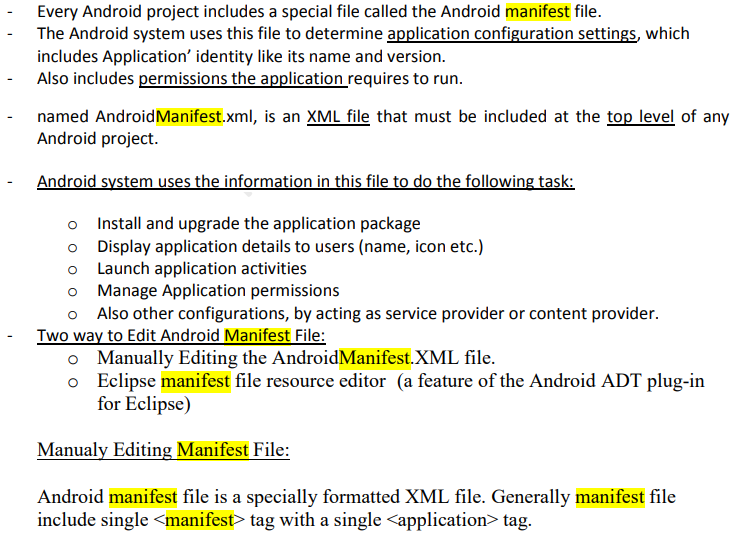
♣ getApplicationContext().getContentResolver().query(uri, ...)

3)various resources available in android.

* 🡪[**Animation Resources**](https://developer.android.com/guide/topics/resources/animation-resource.html)
  + Define pre-determined animations.
  + Tween animations are saved in res/anim/ and accessed from the R.anim class.
  + Frame animations are saved in res/drawable/ and accessed from the R.drawable class.
* [**Color State List Resource**](https://developer.android.com/guide/topics/resources/color-list-resource.html)
  + Define a color resources that changes based on the View state.
  + Saved in res/color/ and accessed from the R.color class.
* [**Drawable Resources**](https://developer.android.com/guide/topics/resources/drawable-resource.html)
  + Define various graphics with bitmaps or XML.  
    Saved in res/drawable/ and accessed from the R.drawable class.
* [**Layout Resource**](https://developer.android.com/guide/topics/resources/layout-resource.html)
  + Define the layout for your application UI.
  + Saved in res/layout/ and accessed from the R.layout class.
* [**Menu Resource**](https://developer.android.com/guide/topics/resources/menu-resource.html)
  + Define the contents of your application menus.
  + Saved in res/menu/ and accessed from the R.menu class.
* [**String Resources**](https://developer.android.com/guide/topics/resources/string-resource.html)
  + Define strings, string arrays, and plurals (and include string formatting and styling).
  + Saved in res/values/ and accessed from the R.string, R.array, and R.plurals classes.
* [**Style Resource**](https://developer.android.com/guide/topics/resources/style-resource.html)
  + Define the look and format for UI elements.
  + Saved in res/values/ and accessed from the R.style class.
* [**Font Resources**](https://developer.android.com/guide/topics/resources/font-resource.html)
  + Define font families and include custom fonts in XML.
  + Saved in res/font/ and accessed from the R.font class.
* [**More Resource Types**](https://developer.android.com/guide/topics/resources/more-resources.html)
  + Define other primitive values as static resources, including the following:
* [**Bool**](https://developer.android.com/guide/topics/resources/more-resources.html#Bool)
  + XML resource that carries a boolean value.
* [**Color**](https://developer.android.com/guide/topics/resources/more-resources.html#Color)
  + XML resource that carries a color value (a hexadecimal color).
* [**Dimension**](https://developer.android.com/guide/topics/resources/more-resources.html#Dimension)
  + XML resource that carries a dimension value (with a unit of measure).
* [**ID**](https://developer.android.com/guide/topics/resources/more-resources.html#Id)
  + XML resource that provides a unique identifier for application resources and components.
* [**Integer**](https://developer.android.com/guide/topics/resources/more-resources.html#Integer)
  + XML resource that carries an integer value.
* [**Integer Array**](https://developer.android.com/guide/topics/resources/more-resources.html#IntegerArray)
  + XML resource that provides an array of integers.
* [**Typed Array**](https://developer.android.com/guide/topics/resources/more-resources.html#TypedArray)
  + XML resource that provides a [TypedArray](https://developer.android.com/reference/android/content/res/TypedArray.html) (which you can use for an array of drawables).

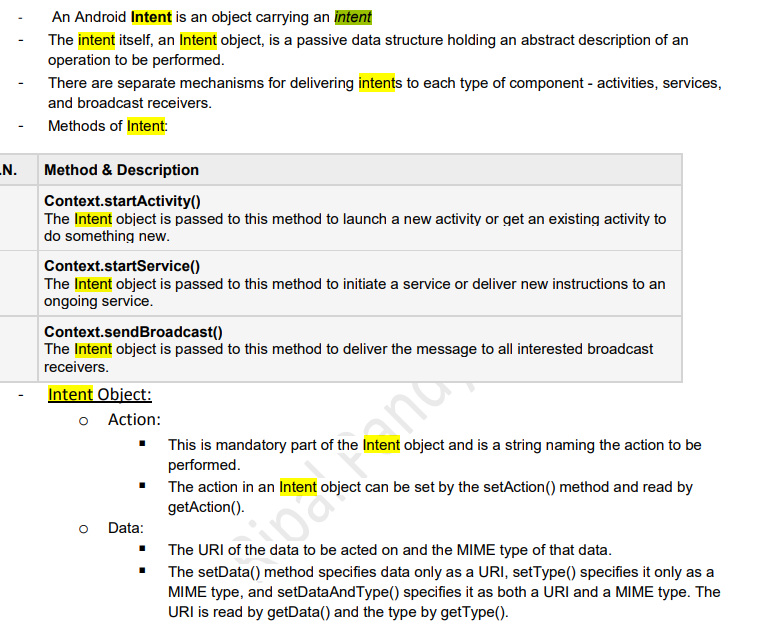
4)What is Android manifestfile ? Explain its usefulness

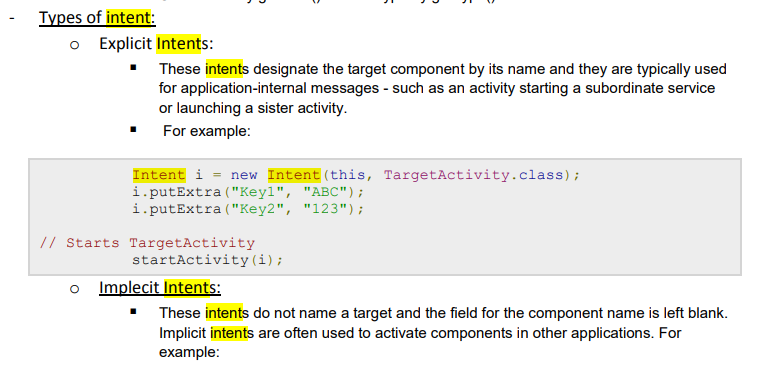
* 🡪Manifest file for an android application is a resource file which contains all the details needed by the android system about the application.
* It is a key file that works as a bridge between the android developer and the android platform.
* It helps the developer to pass on functionality and requirements of our application to Android.
* This is an xml file which must be named as AndroidManifest.xml and placed at application root.
* Every Android app must have AndroidManifest.xml file. AndroidManifest.xml allows us to define The packages, API, libraries needed for the application.
* Basic building blocks of application like activities, services and etc.
* Details about permissions.

🡪

The Android manifest file helps to declare the permissions that an app must have to access data from other apps. The Android manifest file also specifies the app's package name that helps the Android SDK while building the app.

5)What is Intent ? Explain various types of intent.

🡪





6)Explain Anatomy of android application

🡪

7)what is android SDK ? Explain it.

🡺Android SDK is a collection of libraries and Software Development tools that are essential for Developing Android Applications.

* Whenever Google releases a new version or update of Android Software, a corresponding SDK also releases with it.
* In the updated or new version of SDK, some more features are included which are not present in the previous version.
* Android SDK consists of some tools which are very essential for the development of Android Application.
* These tools provide a smooth flow of the development process from developing and debugging. Android SDK is compatible with all operating systems such as Windows, Linux, macOS, etc.

8)Explain R.java file.

🡪**Android R.java** is *an auto-generated file by aapt* (Android Asset Packaging Tool) that contains resource IDs for all the resources of res/ directory.

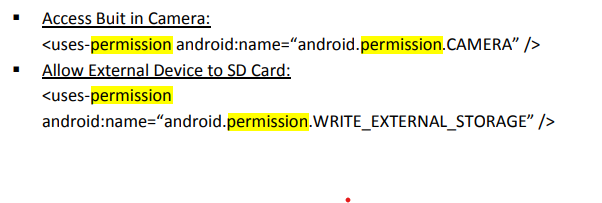
If you create any component in the activity\_main.xml file, id for the corresponding component is automatically created in this file

If you delete R.jar file, android creates it automatically.

9)Explain Detail: permission

🡪Android applications have no permissions by default

Permission can be set using tag inside Android Manifest file

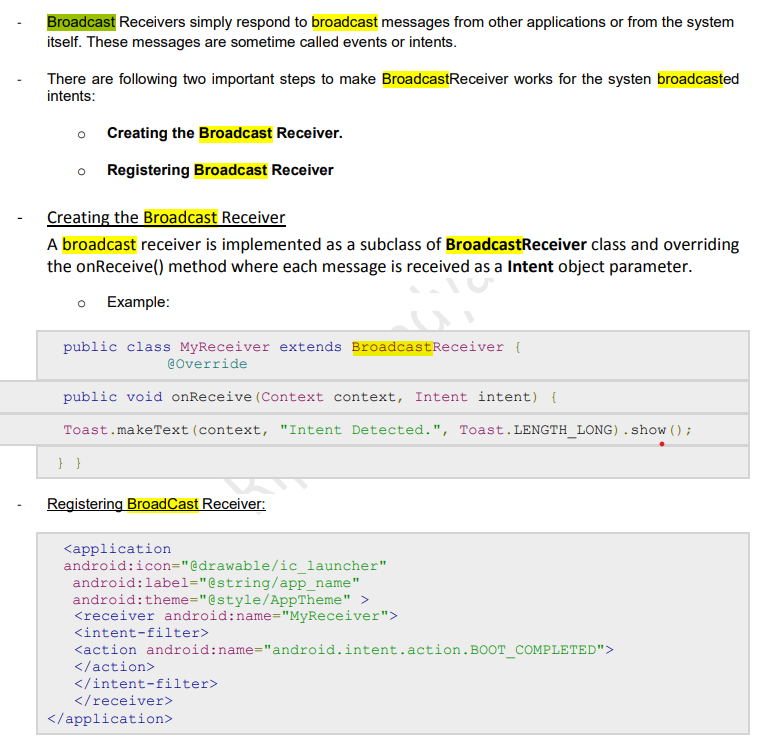


## Types of permissions

Android categorizes permissions into different types, including

* install-time permissions.
* runtime permissions.
* special permissions.

10)Explain Broadcast And receving intent in details.

🡪

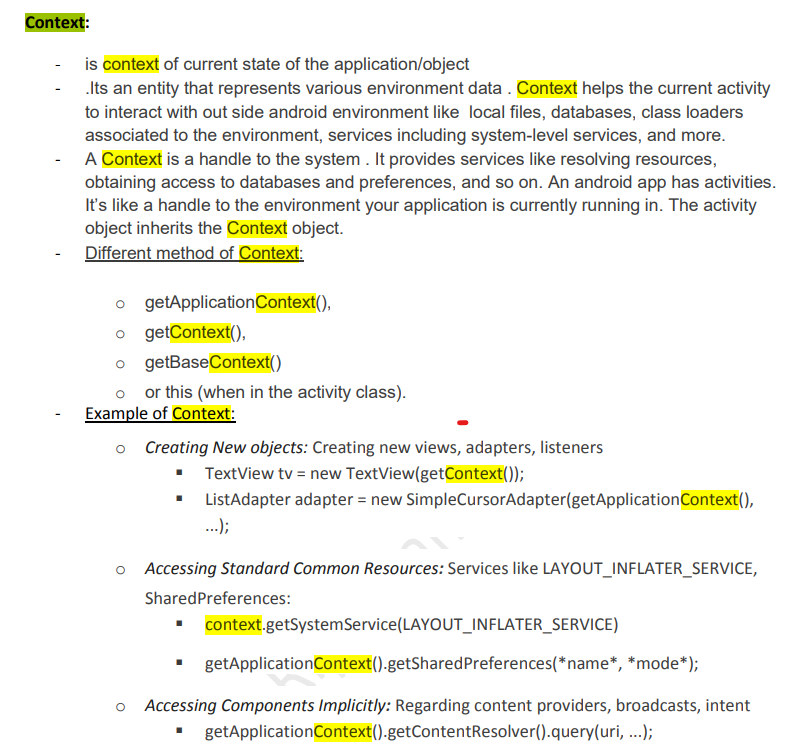
11)what id use of drawable folder in android?

🡪

12)What is use of XML file in android?

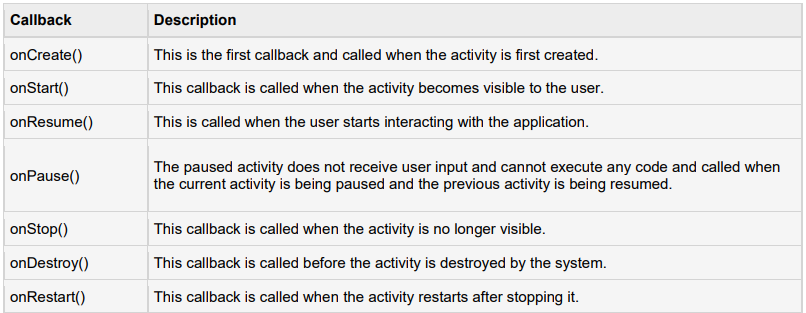
🡪XML tags define the data and used to store and organize data. It's easily scalable and simple to develop. In Android, the XML is used to implement UI-related data, and it's a lightweight markup language that doesn't make layout heavy. XML only contains tags, while implementing they need to be just invoked

13)what is contax and activities .

🡪

Activity: - An activity represents a single screen with a user interface.

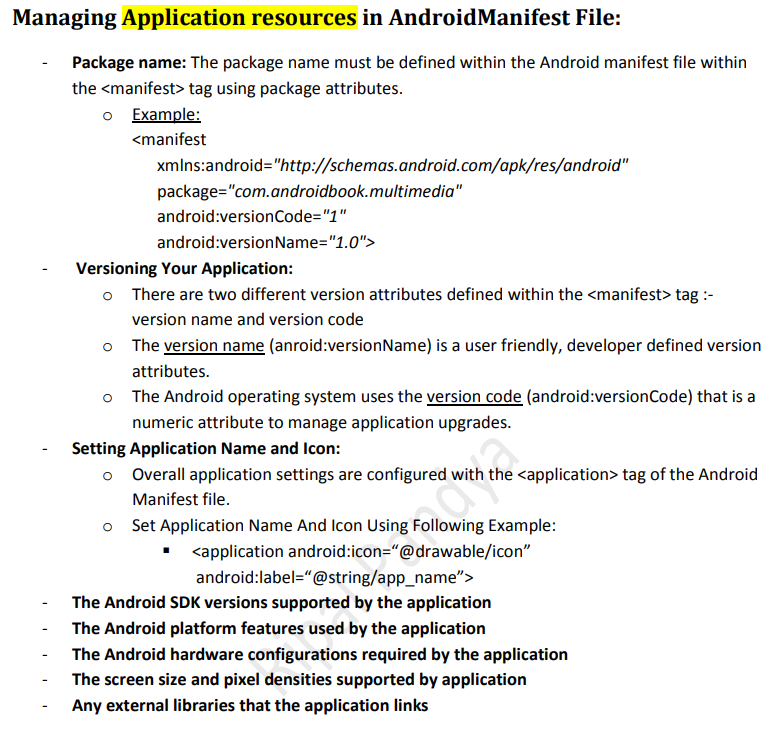
- Activity Lifecycle is below



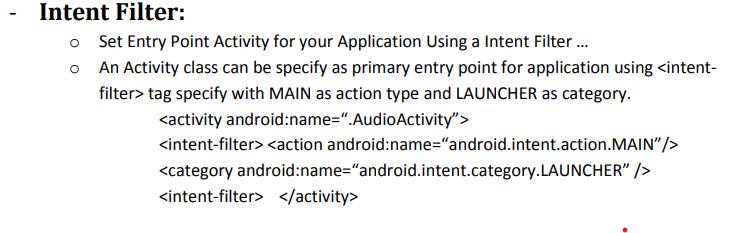
**13) Explain How to create AVD.**

**🡪**

**14)Explain Application Resouces in a hierarchy.**

**🡪**

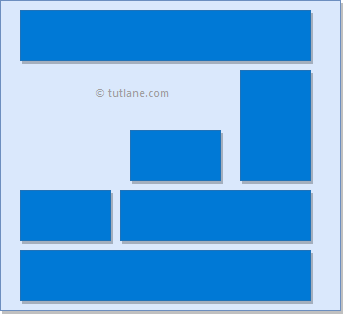
**18)Explain Intenet filter with example.**

****

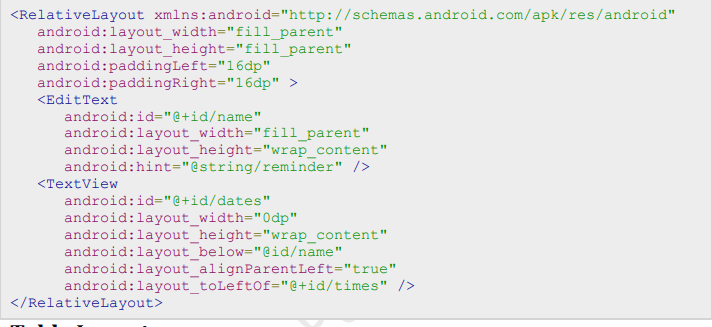
**Unit-2**

**1)Explain with example: Relative layout**

* **🡪**In android, **RelativeLayout** is a **ViewGroup** which is used to specify the position of child **View** instances relative to each other (Child **A** to the left of Child **B**) or relative to the parent (Aligned to the top of parent).
* Following is the pictorial representation of relative layout in android applications.

****

* In android, **RelativeLayout** is very useful to design user interface because by using relative layout we can eliminate the nested view groups and keep our layout hierarchy flat, which improves performance of application.
* A RelativeLayout is a very powerful utility for designing a user interface

****

**2)What is frame by frame animation?**

* **🡪**Frame-by-frame animation is where a series of images are shown in succession at quick intervals so that the final effect is that of an object moving or changing.
* In Android, frame-by-frame animation is implemented through the class AnimationDrawable. This class is a Drawable. These objects are commonly used as backgrounds for views. AnimationDrawable, in addition to being a Drawable, can take a list of other Drawable resources (like images) and render them at specified intervals.

To use this AnimationDrawable class, start with a set of Drawable resources (for example, a set of images) placed in the /res/drawable subdirectory. You will then construct an XML file that defines the AnimationDrawable using a list of these images. This XML file needs to be placed in the /res/drawable. /frame\_animation.xml subdirectory as well.

1. **Create the animation in XML**

When your Android application project is created, you need to put the bitmaps in a res/drawable directory.

**3)what is cross fading? Explain giving example.**

**4)List various Dialoges .Explain any one with example.**

* **🡪**A dialog is a small window that prompts the user to make a decision or enter additional information.
* A dialog does not fill the screen and is normally used for events that require users to take an action before they can proceed.
* In android, you can create following types of Dialogs:
  + - Alert Dialog
    - DatePicker Dialog
    - TimePicker Dialog
    - Custom Dialog

# Android - Alert Dialog

# **A Dialog is small window that prompts the user to a decision or enter additional information.**

# **Some times in your application, if you wanted to ask the user about taking a decision between yes or no in response of any particular action taken by the user, by remaining in the same activity and without changing the screen, you can use Alert Dialog.**

# **In order to make an alert dialog, you need to make an object of AlertDialogBuilder which an inner class of AlertDialog. Its syntax is given below**

AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);

* Now you have to set the positive (yes) or negative (no) button using the object of the AlertDialogBuilder class. Its syntax is

alertDialogBuilder.setPositiveButton(CharSequence text,

DialogInterface.OnClickListener listener)

alertDialogBuilder.setNegativeButton(CharSequence text,

DialogInterface.OnClickListener listener)

**5)Explain twined animation.**

* **🡪** Tween Animation Creates an animation by performing a series of transformations on a single image with an [Animation](https://developer.android.com/reference/android/view/animation/Animation).
* Tween Animation is defined as an animation which is used to Translate, Rotate, Scale and Alpha any type of view in Android.
* All the Tween Animations are coded in Android xml file which are placed together in folder name “anim” under “res” folder in Project directory.
* Tween Animation takes some parameters such as start value , end value, size , time duration , rotation angle e.t.c and perform the required animation on that object.
* It can be applied to any type of object. So in order to use this , android has provided us a class called Animation.
* In order to perform animation in android , we are going to call a static function loadAnimation() of the class AnimationUtils.
* We are going to receive the result in an instance of Animation Object. Its syntax is as follows −

Animation animation = AnimationUtils.loadAnimation(getApplicationContext(),

R.anim.myanimation);

* We have a different type of animations available in android, here we will discuss about most commonly used android animations such as [zoom in / zoom out](https://www.tutlane.com/tutorial/android/android-zoom-in-out-animations-with-examples), [fade in / fade out](https://www.tutlane.com/tutorial/android/android-fade-in-out-animations-with-examples), [slide up / slide down](https://www.tutlane.com/tutorial/android/android-slide-up-down-animations-with-examples) and [rotate clockwise or anti clockwise](https://www.tutlane.com/tutorial/android/android-rotate-animations-clockwise-anti-clockwise-with-examples), etc. with examples.
* This animation class has many useful functions which are listed below −

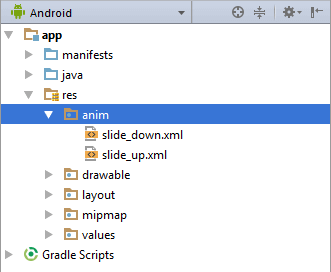
|  |  |
| --- | --- |
| **Sr.No** | **Method & Description** |
| 1 | **start():**This method starts the animation. |
| 2 | **setDuration(long duration):**This method sets the duration of an animation. |
| 3 | **getDuration():**This method gets the duration which is set by above method |
| 4 | **end():**This method ends the animation. |
| 5 | **cancel():**This method cancels the animation. |

To create an animation effect to the objects in our android application, we need to follow below steps.

## Create XML File to Define Animation

* We need to create an xml file that defines the type of animation to perform in a new folder **anim** under **res** directory (**res** **anim**  **animation.xml**) with required properties.
* In case **anim** folder not exists in **res** directory, create a new one.

 Following is the example of creating an XML files under **anim** folder to define [slide up / down](https://www.tutlane.com/tutorial/android/android-slide-up-down-animations-with-examples) animation properties.

****

* In case if we want to use different type of animations such as [fade in / out](https://www.tutlane.com/tutorial/android/android-fade-in-out-animations-with-examples), [zoom in / out](https://www.tutlane.com/tutorial/android/android-zoom-in-out-animations-with-examples), etc. we need to create a new xml files in **anim** folder with required properties.

6)List various UI screen Element . Explain any two with example

🡪

|  |
| --- |
| **UI Control & Description** |
| **1.TextView**  This control is used to display text to the user. |
| **2.EditView**  EditView is a predefined subclass of TextView that includes rich editing  capabilities. |
| **3.AutoCompleteTextView**  The AutoCompleteTextView is a view that is similar to EditText, except that it  shows a list of completion suggestions automatically while the user is typing. |
| **4.Button**  A push-button that can be pressed, or clicked, by the user to perform an action. |
| **5.ImageButton**  AbsoluteLayout enables you to specify the exact location of its children. |
| **6.CheckBox**  An on/off switch that can be toggled by the user. You should use checkboxes  when presenting users with a group of selectable options that are not mutually  exclusive. |
| **7.ToggleButton**  An on/off button with a light indicator. |
| **8.RadioButton**  The RadioButton has two states: either checked or unchecked. |
| **9.RadioGroup**  The RadioGroup is used to group together one or more RadioButtons. |
| **10.ProgressBar**  The ProgressBar view provides visual feedback about some ongoing tasks,  such as when you are performing a task in the background. |
| **11.Spinner**  A drop-down list that allows users to select one value from a set. |
| **12.TimePicker**  The TimePicker view enables users to select a time of the day, in either 24-hour  mode or AM/PM mode |
| **13.DatePicker**  The DatePicker view enables users to select a date of the day |

7)What is table layout ? lists its attribute.

* 🡪In android, **TableLayout** is a **ViewGroup** subclass which is used to display the child View elements in rows and columns.
* Following is the pictorial representation of table layout in android applications.

In android, TableLayout will position its children elements into rows and columns and it won’t display any border lines for rows, columns or cells.

* The TableLayout in android will work same as HTML table and table will have as many columns as the row with the most cells.
* The TableLayout can be explained as **<table>** and TableRow is like **<tr>** element.
* <?xml version="1.0" encoding="utf-8"?>  
  <TableLayout xmlns:android="http://schemas.android.com/apk/res/android"  
      android:layout\_width="match\_parent"  
      android:layout\_height="match\_parent"  
      android:layout\_marginTop="100dp"  
      android:paddingLeft="10dp"  
      android:paddingRight="10dp" >  
      <TableRow android:background="#0079D6" android:padding="5dp">  
          <TextView  
              android:layout\_width="wrap\_content"  
              android:layout\_height="wrap\_content"  
              android:layout\_weight="1"  
              android:text="UserId" />
* </TableRow>  
      <TableRow android:background="#DAE8FC" android:padding="5dp">  
          <TextView  
              android:layout\_width="wrap\_content"  
              android:layout\_height="wrap\_content"  
              android:layout\_weight="1"  
              android:text="1" />  
         </TableRow>  
  </TableLayout>

9)Explain How fragment enhace reusability in an android app?

* 🡪A **Fragment** is a piece of an activity which enable more modular activity design. A fragment encapsulates functionality so that it is easier to reuse within activities and layouts.
* Android devices exists in a variety of screen sizes and densities. Fragments simplify the reuse of components in different layouts and their logic.
* You can build single-pane layouts for handsets (phones) and multi-pane layouts for tablets.
* You can also use fragments also to support different layout for landscape and portrait orientation on a smartphone.
* The below image shows how two UI modules defined by fragments can be combined into one activity for a tablet design but separated for a handset design.
* Following are important points about fragment −
* A fragment has its own layout and its own behavior with its own life cycle callbacks.
* You can add or remove fragments in an activity while the activity is running.
* You can combine multiple fragments in a single activity to build a multi-pane UI.
* A fragment can be used in multiple activities.
* Fragment life cycle is closely related to the life cycle of its host activity which means when the activity is paused, all the fragments available in the activity will also be stopped.
* A fragment can implement a behavior that has no user interface component.
* Fragments were added to the Android API in Honeycomb version of Android which
* choose **New > Layout Resource File**. Name the new xml file fragment\_orange.
* import android.os.Bundle;
* import android.view.LayoutInflater;
* import android.view.View;
* import android.view.ViewGroup;
* import androidx.annotation.NonNull;
* import androidx.annotation.Nullable;
* import androidx.fragment.app.Fragment;
* public class OrangeFragment extends Fragment {
* @Nullable
* @Override
* public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
* View view = inflater.inflate(R.layout.*fragment\_orange*,
* container, false);
* return view;
* }
* }

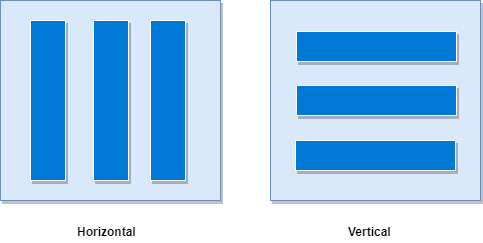
10)Explain Grid layout take suitable reusability in an android app?

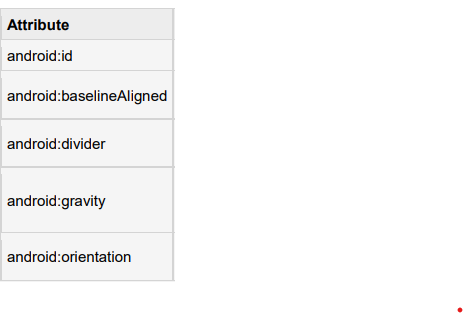
🡪

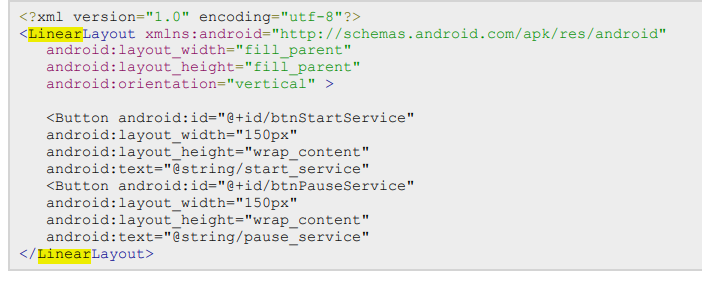
11)xplain Linear layout.

🡪**Linear Layout**

* In android, [LinearLayout](https://www.tutlane.com/tutorial/android/android-linearlayout-with-examples" \o "Android Linear Layout with Examples" \t "_blank) is a ViewGroup subclass which is used to render all child View instances one by one either in horizontal direction or vertical direction based on the orientation property.
* In android, we can specify the linear layout orientation using **android:orientation** attribute.
* Following is the pictorial representation of linear layout in android applications.



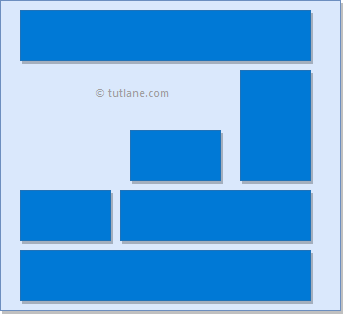
* In **LinearLayout**, the child **View** instances arranged one by one, so the horizontal list will have only one row of multiple columns and vertical list will have one column of multiple rows.
* 



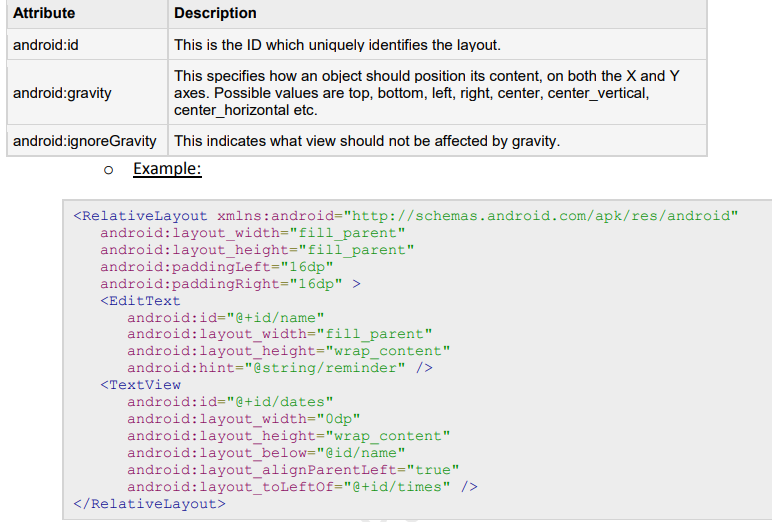
13)Explain relative layout attribute with example.

🡪**Relative Layout**

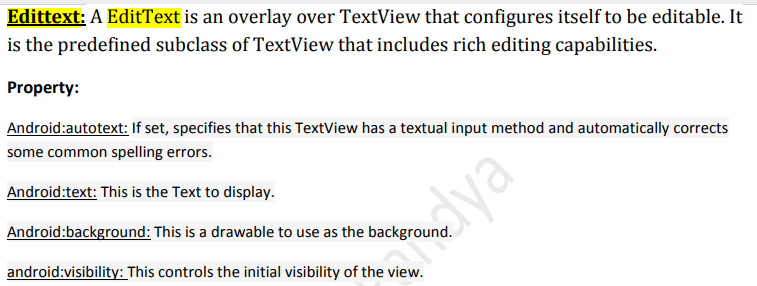
* In android, **RelativeLayout** is a **ViewGroup** which is used to specify the position of child **View** instances relative to each other (Child **A** to the left of Child **B**) or relative to the parent (Aligned to the top of parent).
* Following is the pictorial representation of relative layout in android applications.

****

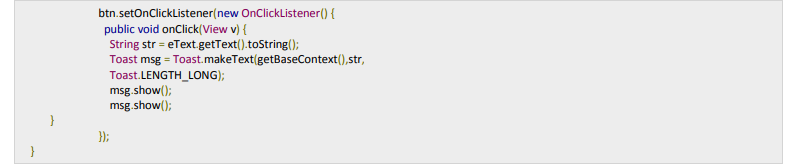
* In android, **RelativeLayout** is very useful to design user interface because by using relative layout we can eliminate the nested view groups and keep our layout hierarchy flat, which improves performance of application.
* A RelativeLayout is a very powerful utility for designing a user interface



**14)Explain Editext view.**

**🡪**

****

****

**15)Explain progressbar Element in android.**

* **🡪Android Progress Bar using ProgressDialog**
* Progress bars are used to show progress of a task. For example, when you are uploading or downloading something from the internet, it is better to show the progress of download/upload to the user.
* In android there is a class called ProgressDialog that allows you to create progress bar. In order to do this, you need to instantiate an object of this class. Its syntax is.

ProgressDialog progress = new ProgressDialog(this);

* there are methods that are provided by the ProgressDialog class

|  |  |
| --- | --- |
| **Sr. No** | **Title & description** |
| 1 | **getMax():**This method returns the maximum value of the progress. |
| 2 | **incrementProgressBy(int diff):**This method increments the progress bar by the difference of value passed as a parameter. |
| 3 | **setIndeterminate(boolean indeterminate):**This method sets the progress indicator as determinate or indeterminate. |
| 4 | **setMax(int max):**This method sets the maximum value of the progress dialog. |
| 5 | **setProgress(int value):**This method is used to update the progress dialog with some specific value. |
| 6 | **show(Context context, CharSequence title, CharSequence message):**This is a static method, used to display progress dialog. |

**16)Explain radio group element in android**

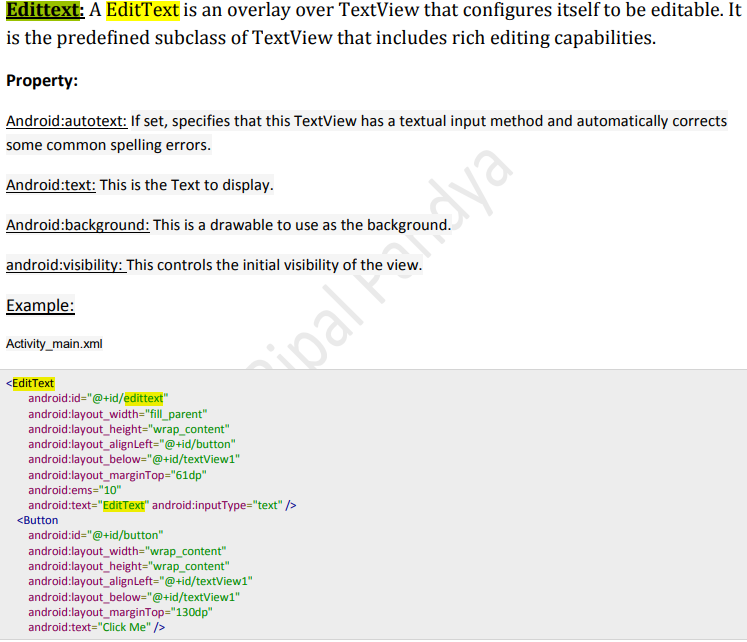
* **🡪Android Progress Bar using ProgressDialog**
* Progress bars are used to show progress of a task. For example, when you are uploading or downloading something from the internet, it is better to show the progress of download/upload to the user.
* In android there is a class called ProgressDialog that allows you to create progress bar. In order to do this, you need to instantiate an object of this class. Its syntax is.

ProgressDialog progress = new ProgressDialog(this);

* there are methods that are provided by the ProgressDialog class

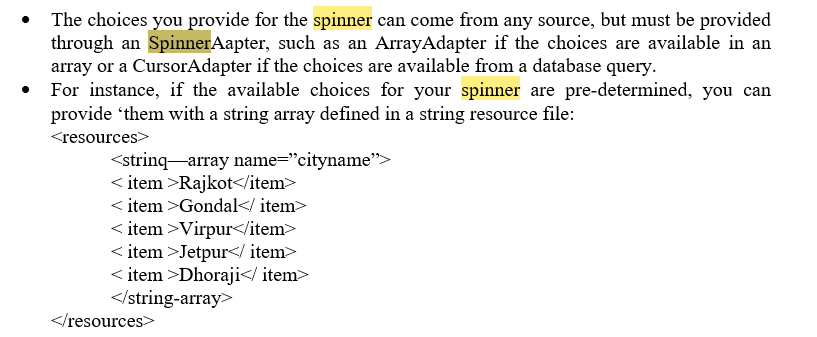
|  |  |
| --- | --- |
| **Sr. No** | **Title & description** |
| 1 | **getMax():**This method returns the maximum value of the progress. |
| 2 | **incrementProgressBy(int diff):**This method increments the progress bar by the difference of value passed as a parameter. |
| 3 | **setIndeterminate(boolean indeterminate):**This method sets the progress indicator as determinate or indeterminate. |
| 4 | **setMax(int max):**This method sets the maximum value of the progress dialog. |
| 5 | **setProgress(int value):**This method is used to update the progress dialog with some specific value. |
| 6 | **show(Context context, CharSequence title, CharSequence message):**This is a static method, used to display progress dialog. |

**17)Explain Listview Widget.**

**🡪**

**19)Explain Spinner widget.**

**🡪**

****

**UNIT-3**

**1)Explain With Syntaxt : Delete Query**

**🡪**

**2)What are Shared Preference?**

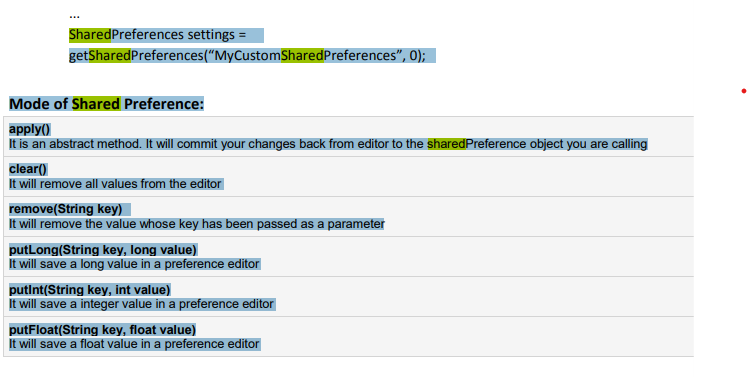
**🡺**SharedPreferences is an API from Android SDK to store and retrieve application preferences.

SharedPreferences are simply sets of data values that stored persistently.

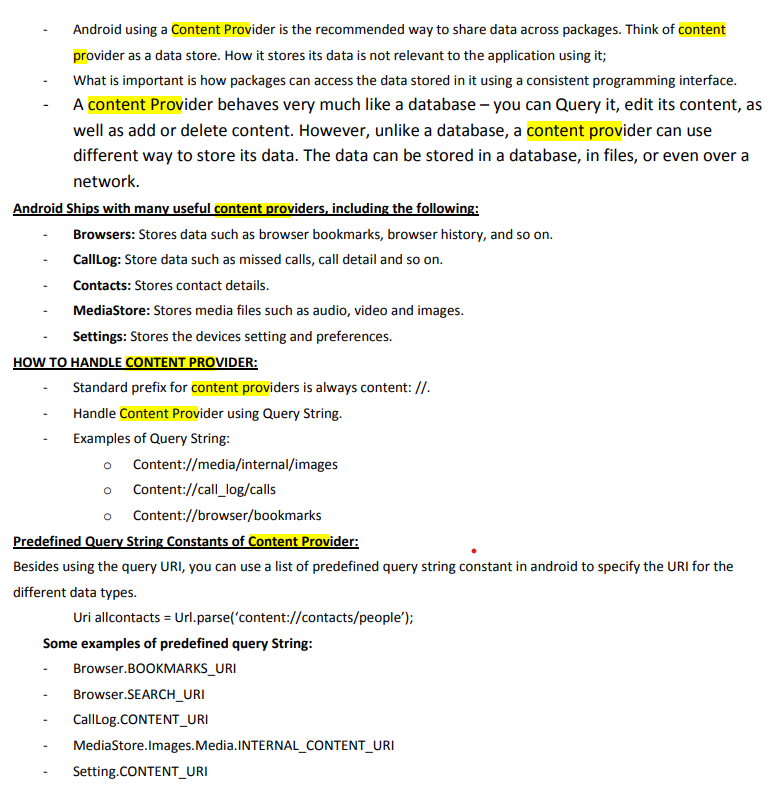
Persistently which mean data you stored in the SharedPreferences are still exist even if you stop the application or turn off the device.

Creating shared preferences is similar.The only two differences are that we must name our preference set and use a different call to get the preference instance:

import android.content.SharedPreferences**;**

****

**3)What is content provider ? How it is useful**

**🡪**

**4)Short Note on: Retrofit**

**🡪**

**5)Explain in details: SQLite**

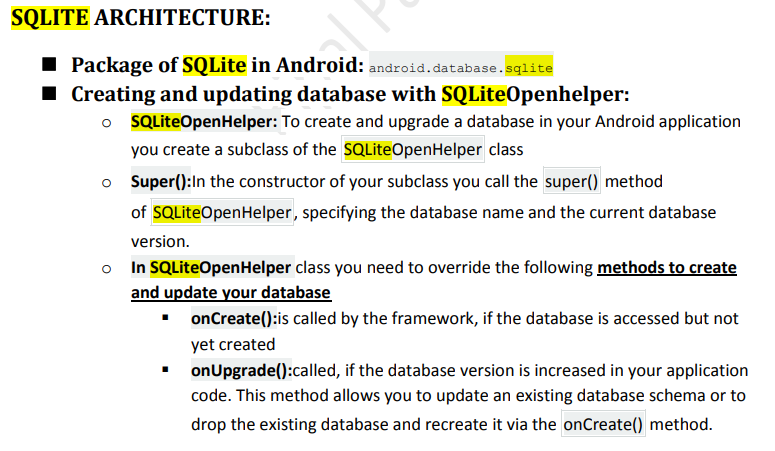
**🡪**SQLite is an Open Source database. SQLite supports standard relational database features like SQL syntax, transactions and prepared statements

SQLite supports the data types TEXT (similar to String in Java), INTEGER (similar to long in Java) and REAL (similar to double in Java). All other types must be converted into one of these fields before getting saved in the database

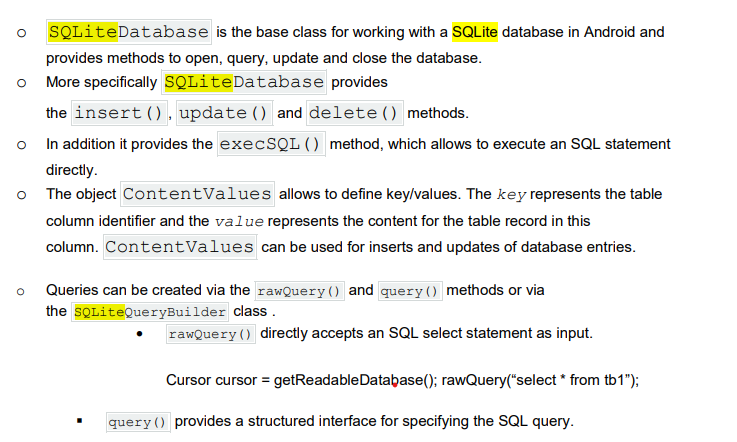
**SQLite in Android:**

SQLite is embedded into every Android device. Using an SQLite database in Android does not require a setup procedure or administration of the database

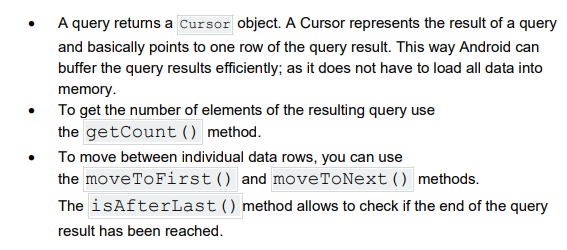
. If your application creates a database, this database is by default saved in the directory: DATA/data/APP\_NAME/databases/FILENAME.

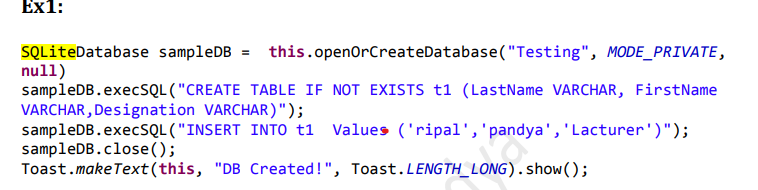
****

**DATABASE:**

****

**Cursor:**

****

****

**6)Explain Basic use of rest api.**

**🡪**The basic function of a RESTful API is the same as browsing the internet. The client contacts the server by using the API when it requires a resource. API developers explain how the client should use the REST API in the server application API documentation.

Access to Remote Services: REST APIs allow Android applications to interact with remote services, such as web servers or cloud platforms. This enables developers to access a wide range of functionalities and data from various sources, including social media platforms, weather services, payment gateways, and more.

**7)List various storage classes . Explain any one.**

**🡪**Mainly Two types of File Data Storage Available in Android:

- Internal Storage

- External Storage

shared preferences,

internal and external storage,

SQLite storage,

storage via network connection.

Internal storage(physical storage)

* It’s always available.
* Files saved here are accessible by only your app by default.
* When the user uninstalls your app, the system removes all your app’s files from internal storage.
* Internal storage is best when you want to be sure that neither the user nor other apps can access your files.
* We are able to save or read data from the device internal memory. FileInputStream and FileOutputStream classes are used to read and write data into the file.

**How to save file on Internal storage?**

Following two methods are use to save file.

* **getFilesDir** () : Returns a File representing an internal directory for your app.
* **getCacheDir** () : Returns a File representing an internal directory for your app’s temporary cache files.
* To create a new file in one - of these directories, you can use the File() constructor.
* Passing the File provided by one of the above methods that specifies your internal storage directory.

**8)List Basic function of Cursor classs.**

**🡪**

|  |  |
| --- | --- |
| **Sr.No** | **Method & Description** |
| 1 | **getColumnCount()**  This method return the total number of columns of the table. |
| 2 | **getColumnIndex(String columnName)**  This method returns the index number of a column by specifying the name of the column |
| 3 | **getColumnName(int columnIndex)**  This method returns the name of the column by specifying the index of the column |
| 4 | **getColumnNames()**  This method returns the array of all the column names of the table. |
| 5 | **getCount()**  This method returns the total number of rows in the cursor |
| 6 | **getPosition()**  This method returns the current position of the cursor in the table |
| 7 | **isClosed()**  This method returns true if the cursor is closed and return false otherwise |

**9)Explain Shared preference.**

**🡪Shared Preference**

* Android provides many way to storing data in application.
* One of from this is by shared preference.
* Android shared preference is used to store and retrieve small amount of primitive information. In android, string, integer, long, number etc. are considered as primitive data type.
* Android Shared preferences are used to store data in key and value pair so that we can retrieve the value on the basis of key.
* It is way in which data as key/value pairs to a file on the device storage.

**Get a Handle to a SharedPreferences:**

* You can create a new shared preference file or access an existing one by calling one of two methods:

 **getPreferences()** : used from within your Activity, to access activity-specific preferences

* **getSharedPreferences()** : used from within your Activity (or other application Context), to access application-level
* you have to call a method getSharedPreferences() that returns a SharedPreference instance pointing to the file that contains the values of preferences.

SharedPreferences sharedpreferences = getSharedPreferences(MyPREFERENCES, Context.MODE\_PRIVATE);

**UNIT-4**

**1)what is GeoCoding ?Explain**

**🡪** Geocoding (sometimes called forward geocoding) is the process of enriching a description of a location, most typically a postal address or place name, with geographic coordinates from spatial reference data such as building polygons, land parcels, street addresses, ZIP codes (postal codes) and so on. Geocoding facilitates spatial analysis using Geographic Information Systems and Enterprise Location Intelligence systems.

Reverse geocoding is the process of enriching geographic coordinates with a description of the location, most typically a postal address or place name. A geocoder is a piece of software or a (web) service that implements a geocoding process.

* Latitude and longitude are imaginary (unreal) lines drawn on maps to easily locate places on the Earth.
* Latitude is distance north or south of the equator (an imaginary circle around the Earth halfway between the North Pole and the South Pole) and
* longitude is distance east or west of the prime meridian (an imaginary line running from north to south through Greenwich, England).

**2)list and various types of api commonly use din web services.**

**🡪**

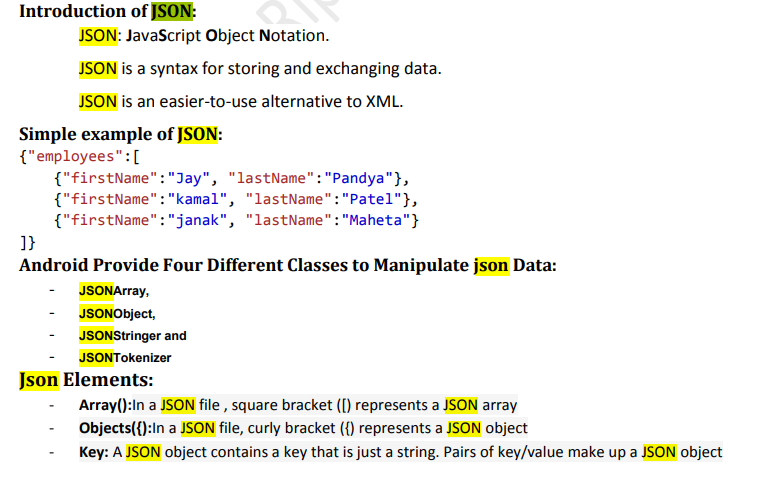
**3)writes steps to status bar notification in android.**

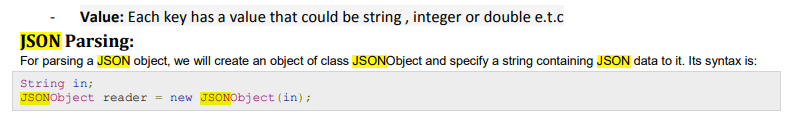
1. **🡪** Get a reference to the NotificationManager: String ns = Context. ...
2. Instantiate the Notification: int icon = R. ...
3. Define the Notification's expanded message and Intent: Context context = getApplicationContext(); ...
4. Pass the Notification to the NotificationManager:

**4)Explain how to connect to MYSQL Via post method.**

**🡪**

**5)Explain :JSON**

**🡪**

****

**Advantages:**

**-JSON is faster**

**-SCHEMA support**

**-server parsing**

**-tool for sharing data**

## ****JSON Disadvantages****

First and foremost, in JSON has no error handling for JSON calls. If the dynamic script insertion works, you get called and will get the response perfectly.

If not inserted, nothing happens. It just fails silently. For example, you are not able to catch a 404 error from the server, Nor can you cancel or restart the request. You can, however, timeout after waiting a reasonable amount of time.

**Uses:**

Android supports all the JSON classes such as JSONStringer, JSONObject, JSONArray, and all other forms to parse the JSON data and fetch the required information by the program. JSON's main advantage is that it is a language-independent, and the JSON object will contain data like a key/value pair.

**6)Explain Bifly:**

**🡪1)Telephony Api**

- The Android SDK provides a number of useful utilities for applications to integrate phone features available on the device.

- Applications might need to place a call or send a text message.

- The TelephonyManager object within the android.telephony package is a great place to start.

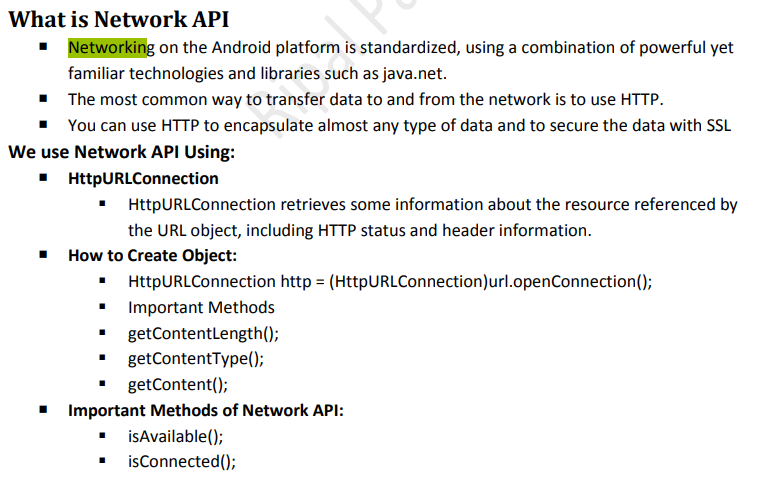
- The READ\_PHONE\_STATE permission is required to retrieve information such as the call state, handset phone number, and device identifiers or serial numbers.

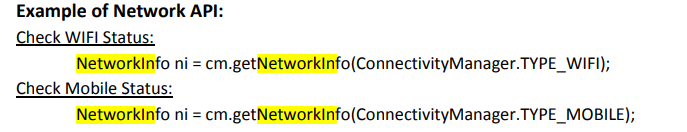
- You can use the TelephonyManager object to retrieve the state of the phone and some information about the phone service itself, such as the phone number of the handset.

- You can request an instance of TelephonyManager using the getSystemService() method, like this

: TelephonyManager telManager = (TelephonyManager)getSystemService(Context.TELEPHONY\_SERVICE);

**🡪2)Networking API**

****

****

**7)Explain Get current location on android.**

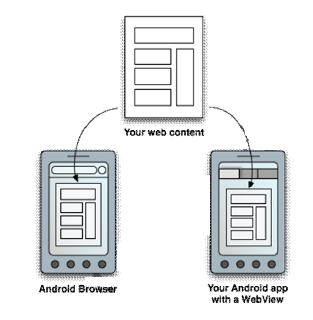
**🡪**

**8)Write steps to generate marker on map.**

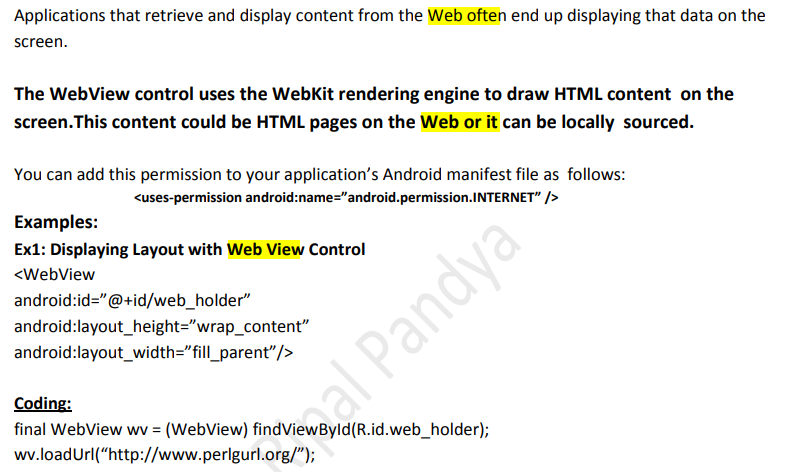
**🡪**

**9)Write note on Android Web API.**

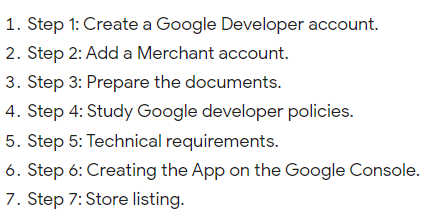
* **🡪**There are essentially **two ways to deliver an application on Android**: as a **client-side application** (developed using the Android SDK and installed on user devices in an APK) or as a **web application** (developed using web standards and accessed through a web browser—there's nothing to install on user devices).
* If you chose to provide a web-based app for Android-powered devices, you can rest assured that major web browsers for Android (and the WebView framework) allow you to specify viewport and style properties that make your web pages appear at the proper size and scale on all screen configurations.



* Above figure illustrates how you can provide access to your web pages from either a web browser or your own Android app.
* However, you shouldn't develop an Android app simply as a means to view your web site.
* Rather, the web pages you embed in your Android app should be designed especially for that environment.
* You can even define an interface between your Android application and your web pages that allows JavaScript in the web pages to call upon APIs in your Android application—providing Android APIs to your web-based application.
* To start developing web pages for Android-powered devices, see the following documents:

****

**10)writes a steps to published the android application.**

**🡪**

**11)Writes details : GPS**

**🡪**The Global Positioning System (GPS) is a space-based satellite navigation system that provides location and time information in all weather conditions, anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites.

The GPS project was developed in 1973 to overcome the limitations of previous navigation systems,[2] integrating ideas from several predecessors, including a number of classified engineering design studies from the 1960s.

GPS was created and realized by theU.S. Department of Defense (DoD) and was originally run with 24 satellites.

It became fully operational in 1995. Bradford Parkinson,Roger L. Easton, and Ivan A. Getting are credited with inventing it.

**How GPS Works?**

* A GPS system consists of a network of 24 orbiting satellites, called **NAVSTAR** (Navigation System with Time and Ranging), and placed in space in six different orbital paths with four satellites in each orbital plane and covering the entire earth under their signal beams.

|  |  |
| --- | --- |
| gps | gps2 |
| **NAVSTAR** | **triangulation technique**. |

* As above figure The GPS is based on well known concept called the **triangulation technique**.
* The receiver uses the messages it receives to determine the transit time of each message and computes the distance to each satellite using the speed of light.
* Each of these distances and satellites locations defines a area.
* The receiver is on surface of each of these areas when the distances and the satellites’ locations are correct.
* These distances and satellites locations are used to compute the location the receiver using the navigation equations.
* This location is then displayed, perhaps with a moving map display or latitude and longitude; -
* Basic GPS measurements surrender only a position, and neither speed direction.
* However, most GPS units can automatically derive speed and direction movement from two or more position measurements.
* The disadvantage of this principle is that changes in speed or direction can only be computed with and that derived direction becomes inaccurate when the distance travelled between two position measurements drops below or near the random error of posit measurement.
* More advanced navigation systems use additional sensors like compass or an inertial navigation system to complement GPS.

**How to Use Global Positioning Service(GPS)**

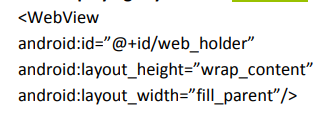
* The Android SDK provides resources for accessing location via a built-in GPS hardware if available in hardware. Generally speaking, just about every **Android phone has some LBS(Location based Services) capabilities.**
* For example, in the United States, mobile phone location information is used emergency services.
* That said, not all android devices are phones, nor do all enable consumer-usage of LBS services.
* If **GPS features are disabled, or an An device does not have LBS hardware,** the Android SDK provides additional APIs determining alternate location providers.
* These other providers might have advantage and disadvantages in terms of power use, speed, and accuracy of reporting.

The key **advantages** of using Global Positioning System (GPS) include precise location tracking, improved navigation and timing, and enhanced security and safety

GPS is extremely easy to navigate because it tells you to direction for every turns you're taking otherwise you need to fancy reach to your destination. GPS works altogether weather so you would like to not worry of climate as in other navigating devices. GPS costs you very low as compared other navigation systems

**12)Explain Web view with example.**

* **🡪**If you want to deliver a web application (or just a web page) as a part of a client application, you can do it using WebView.
* **The WebView class is an extension of Android's View class that allows you to display web pages as a part of your activity layout.**
* It does not include any features of a fully developed web browser, such as navigation controls or an address bar. All that WebView does, by default, is show a web page.
* A common scenario in which using WebView is helpful is when you want to provide information in your application that you might need to update, such as an end-user agreement or a user guide.
* Within your Android application, you can create an Activity that contains a WebView, and then use that to display your document that's hosted online.
* Another scenario in which WebView can help is if your application provides data to the user that always requires an Internet connection to retrieve data, such as email.
* In this case, you might find that it's easier to build a WebView in your Android application that shows a web page with all the user data, rather than performing a network request, then parsing the data and rendering it in an Android layout.
* Instead, you can design a web page that's tailored for Android devices and then implement a WebView in your Android application that loads the web page.

****

**UNIT-5**

**1)What is story board?**

* **🡪**The storyboard is first introduced in iOS 5 to save time building user interfaces for the iOS applications.
* A storyboard is a visual representation of the user interface of an iOS application, showing screens of content and the connections between those screens.
* A storyboard is composed of a sequence of scenes, each of which represents a view controller and its views; scenes are connected by segue objects, which represent a transition between two view controllers.
* Xcode provides a visual editor for storyboards, where you can lay out and design the user interface of your application by adding views such as buttons, table views, and text views onto scenes.
* In addition, a storyboard enables you to connect a view to its controller object, and to manage the transfer of data between view controllers. Using storyboards is the recommended way to design the user interface of your application because they enable you to visualize the appearance and flow of your user interface on one canvas.

**2)Explain Simulator?(3 or 5)**

**🡪**

**3)What is MVC? Explain in detail.**

**🡪**While developing an iOS application, we always face a challenge to manage the project so that it can be extended easily.

MVC is a software development pattern made up of three main objects:

**Model**

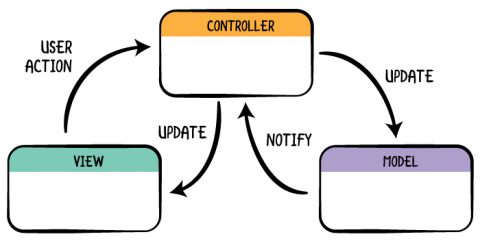
* The **Model** contains the things that are used to handle the data of the application. The models are used to parse the request and response with the server's API. Things like persistence, model objects, parsers, managers, and networking code reside here.

**View**

* The **View** is treated as the face of the application. The object like UILabel and UITextField are the view objects that present the data on the screen. The view doesn't contain any domain-specific logic.

**Controller**

* The **controller** is used as the mediator between the Model and the view via the delegation pattern. The controller doesn't need to know the concrete view it is working for. However, the controller contains the business logic to present the data that is parsed by the model and displayed by the view objects.



## The Model (M)

The model layer encompasses your app’s data. No surprise there, but there are usually other classes and objects in your projects that you can include in this layer:

* Network code:
* Persistence code:
* Parsing code:
* Managers and abstraction layers/classes:
* Data sources and delegates:
* Constants:

## The View (V)

When users interacts with your app, they are interacting with the view layer. It should not contain any business logic. In code terms, you’ll normally see:

* UIView subclasses: These range from a basic UIView to complex custom UI controls.
* Classes that are part of UIKit/AppKit.
* Core Animation.
* Core Graphics.

Typical [code smells](https://en.wikipedia.org/wiki/Code_smell) found in this layer manifest in different ways but boil down to include anything unrelated to UI in your view layer. This can include network calls, business logic, manipulating models and so on.

Use the following as a checklist when inspecting your view layer:

* Does it interact with the model layer?
* Does it contain any business logic?

From a testing perspective, UI testing in iOS allows you to test things like transitions, UI elements having specific attributes or UI interactions working as intended.

## The Controller (C)

The controller layer is the least reusable part of your app as it often involves your domain-specific rules. It should be no surprise that what makes sense in your app won’t always make sense in someone else’s. The controller will then use all the elements in your model layer to define the flow of information in your app.

Usually, you’ll see classes from this layer deciding things like:

* What should I access first: the persistence or the network?
* How often should I refresh the app?
* What should the next screen be and in which circumstances?
* If the app goes to the background, what should I tidy up?
* The user tapped on a cell; what should I do next?

Think of the controller layer as the brain, or engine, of the app; it decides what happens next.

You likely won’t do much testing in the controller layer as, for the most part, it’s meant to kick things off, trigger data loads, handle UI interactions, mediate between UI and model, etc.

**3)Explain COCOA Touch.**

* **🡪**Cocoa Touch is a user interface framework provided by Apple for building software applications for products like iPhone, iPad and iPod Touch.
* It is primarily written in Objective C language and is based on Mac OS X. Cocoa Touch was developed based on model view controller software architecture.
* The high-level application programming interfaces available in Cocoa Touch help to make animation, networking, and adding the appearance and behavior of the native platform to the developed applications possible with less code development.
* **The main features of Cocoa Touch include:**
* Core Animation: Helps to create rich user experiences by allowing for the smooth movement of visual elements. It also fills in the interim frames of animation with automatic timing and adjustment.
* Core Audio
* Core Data: Provides an object-oriented data management solution and aids in defining an application's data model in a logical and graphical way.
* **Cocoa Touch is made up of several frameworks, but the key ones are:**

|  |  |  |
| --- | --- | --- |
| **Audio and Video**   * Core Audio * OpenAL * Media Library * AV Foundation | **Data Management**   * Core Data * SQLite | **Graphics and Animation**   * Core Animation * OpenGL ES * Quartz 2D |
| **Networking and Internet**   * Bonjour * WebKit * BSD Sockets | **User Applications**   * Address Book * Core Location * Map Kit * Store Kit |  |

**5)Explain how X-code is useful in developing application.**

🡪**X Code is an integrated development environment (IDE).**

Xcode is an integrated development environment, or IDE, for…making iOS, macOS, tvOS and watchOS apps.…In short, it's an app for making apps.…

Xcode is also a code editor that…supports multiple languages.…

It's a user interface development tool…

for creating apps and games as well.…And finally,

it is the only officially-supported tool…created by Apple for the creation of publishing apps to Apple's app stores.…

* Xcode includes all of the tools needed to create an app within one software package; namely, a text editor, graphic user interface (UI) editor, debugging tools and much more.
* Most importantly, Xcode comes with an iPhone (and iPad) simulator so you can test your app without the real devices.
* With Xcode, you can write, compile, and debug your app, and when you’re finished you can submit it to the Apple app store.
* As a code editor, Xcode supports a huge variety of programming languages – C, C++, Objective-C, Objective-C++, Java, AppleScript, Python, Ruby, ResEdit, and Swift. It uses Cocoa, Carbon, and Java programming models.

**6)what is what is the textview control? Explain Giving example.**

**🡪**

**7)List Visual control in ios. Explain any one with example.**

**🡪**

**8)Explain Textbox in iphone.**

**🡪**

**9)Explain Button in Iphone.**

**🡪: A Button is a Push-button which can be pressed, or clicked, by the user to perform an action**