**Why do we need android studio?**

Android Studio provides a unified environment where you can build apps for Android phones, tablets, Android Wear, Android TV, and Android Auto. Structured code modules allow you to divide your project into units of functionality that you can independently build, test, and debug.

Features of Android

1. Beautiful UI
2. Connectivity
3. Storage
4. Media support
5. Messaging
6. Web browser
7. Multi-touch
8. Multi-tasking
9. Multi-Language
10. Wi-Fi Direct
11. GCM

**What is the purpose of using Android SDK?**

The Android SDK (Software Development Kit) is a set of development tools that are used to develop applications for the Android platform

The JDK provides tools, such as the Java compiler, used by IDEs and SDKs for developing Java programs

JVM(Java Virtual Machine)

* Java source code is compiled into Java bytecode format(.class file) that further translates into machine code.
* The executable file for the device is .jar file.
* Supports multiple operating systems like Linux, Windows, and Mac OS.

DVM(Dalvik Virtual Machine)

* Source code files are first of all compiled into Java bytecode format like JVM. Further, the DEX compiler(dx tool) converts the Java bytecode into Dalvik bytecode(classes.dex) file that will be used to create the .apk file.
* The executable file for the device is .apk file.
* Support only the Android operation system.

Android Activity Lifecycle

|  |  |
| --- | --- |
| Method | Description |
| onCreate | called when activity is first created. |
| onStart | called when activity is becoming visible to the user. |
| onResume | called when activity will start interacting with the user. |
| onPause | called when activity is not visible to the user. |
| onStop | called when activity is no longer visible to the user. |
| onRestart | called after your activity is stopped, prior to start. |
| onDestroy | called before the activity is destroyed. |

**text view**

A TextView displays text to the user and optionally allows them to edit it. A TextView is a complete text editor, however the basic class is configured to not allow editing.

**Edit Text**

A EditText is an overlay over TextView that configures itself to be editable. It is the predefined subclass of TextView that includes rich editing capabilities.

**Auto Complete Text View**

* Android AutoCompleteTextView is an editable text view which shows a list of suggestions when user starts typing text. When a user starts typing, a dropdown menu will be there based on the entered characters, defined in the threshold limit and the user can choose an item from the list to replace the text.
* The AutoCompleteTextView is a subclass of EditText class so we can easily inherit all the properties of EditText as per our requirements.

Attribute & Description

1 **android:completionHint** This defines the hint displayed in the drop-down menu

2 **android:completionHintView** This defines the hint view displayed in the drop-down menu

3 **android:completionThreshold** This defines the number of characters that the user must type before completion suggestions are displayed in a drop-down menu

4 **android:dropDownAnchor** This is the View to anchor the auto-complete dropdown to

**Button**

In [Android](https://www.geeksforgeeks.org/kotlin-android-tutorial/) applications, a Button is a user interface that is used to perform some action when clicked or tapped. It is a very common widget in Android and developers often use it.

**ImageButton**

An ImageButton is an AbsoluteLayout which enables you to specify the exact location of its children. This shows a button with an image (instead of text) that can be pressed or clicked by the user.

**Toggle Button**

Android Toggle Button can be used to display checked/unchecked (On/Off) state on the button.

It is beneficial if user have to change the setting between two states. It can be used to On/Off Sound, Wifi, Bluetooth etc.

Graphical user interface, application

Description automatically generated

**CheckBox**

* Android CheckBox is a type of two state button either checked or unchecked.
* There can be a lot of usage of checkboxes. For example, it can be used to know the hobby of the user, activate/deactivate the specific action etc.
* Android CheckBox class is the subclass of CompoundButton class.

**Radio button**

* Radio buttons allow the user to select one option from a set.
* RadioButton is a two states button which is either checked or unchecked
* RadioButton is generally used with RadioGroup. RadioGroup contains several radio buttons, marking one radio button as checked makes all other radio buttons as unchecked.

**RadioGroup**

* A RadioGroup class is used for set of radio buttons.
* If we check one radio button that belongs to a radio group, it automatically unchecks any previously checked radio button within the same group.

RadioGroup Attributes

|  |  |
| --- | --- |
| android:checkedButton | This is the id of child radio button that should be checked by default within this radio group. |
|  |  |

Following are the few methods of radio button:

1. check(id):

This sets the selection to the radio button whose identifier is passed in parameter. -1 is used as the selection identifier to clear the selection.

1. clearCheck() :

It clears the selection. When the selection is cleared, no radio button in this group is selected and getCheckedRadioButtonId() returns null.

1. getCheckedRadioButtonId() :

It returns the identifier of the selected radio button in this group. If its empty selection, the returned value is-1.

1. setOnCheckedChangeListener():

This registers a callback to be invoked when the checked radio button changes in this group. We must supply instance of Radio Group OnCheckedChangeListener() to setOnCheckedChangeListener() method.

**Progress Bars**

* 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);

**Indeterminate Progress –**

Use indeterminate mode for the progress bar when you do not know how long an operation will take. Indeterminate mode is the default for progress bar and shows a cyclic animation without a specific amount of progress indicated.

**Determinate Progress –**

Use determinate mode for the progress bar when you want to show that a specific quantity of progress has occurred. For example, the percent remaining of a file being retrieved, the amount records in a batch written to database, or the percent remaining of an audio file that is playing.

**Progress Dialog** is a class 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 dialog=new ProgressDialog(this);

• **wrap\_content** tells the view to size itself to the dimensions required by its content.

• **match\_parent** tells the view to become as big as its parent view group will allow.

**ListView**

Android ListView is a view which groups several items and display them in vertical scrollable list. The list items are automatically inserted to the list using an Adapter that pulls content from a source such as an array or database.

Graphical user interface, application

Description automatically generated

**GridView**

Android GridView shows items in two-dimensional scrolling grid (rows & columns) and the grid items are not necessarily predetermined but they automatically inserted to the layout using a ListAdapter

Graphical user interface

Description automatically generated

**ImageView**

* An ImageView control is used to display images in Android applications
* ImageView class is used to display any kind of image resource in the android application either it can be android.graphics.Bitmap or android.graphics.drawable.

**ScrollView**

In [Android](https://www.geeksforgeeks.org/introduction-to-android-development/), a ScrollView is a view group that is used to make vertically scrollable views. A scroll view contains a single direct child only. In order to place multiple views in the scroll view, one needs to make a view group(like [LinearLayout](https://www.geeksforgeeks.org/android-linearlayout-in-kotlin/)) as a direct child and then we can define many views inside it.

Table

Description automatically generated

**Date and Time Picker.**

Android provides controls for the user to pick a time or pick a date as ready-to-use dialogs. Each picker provides controls for selecting each part of the time (hour, minute, AM/PM) or date (month, day, year). Using these pickers helps ensure that your users can pick a time or date that is valid, formatted correctly, and adjusted to the user’s locale. Android Date Picker allows you to select the date consisting of day, month and year in your custom user interface. For this functionality android provides DatePicker and DatePickerDialog components.

1.getDayOfMonth()

This method gets the selected day of month

2 getMonth()

This method gets the selected month

3 getYear()

This method gets the selected year

4 setMaxDate(long maxDate)

This method sets the maximal date supported by this DatePicker in milliseconds since January 1, 1970 00:00:00 in getDefault() time zone

5 setMinDate(long minDate)

This method sets the minimal date supported by this NumberPicker in milliseconds since January 1, 1970 00:00:00 in getDefault() time zone

6 setSpinnersmode(boolean shown)

This method sets whether the spinners are shown

7 updateDate(int year, int month, int dayOfMonth)

This method updates the current date

8 getCalendarView()

This method returns calendar view

9 getFirstDayOfWeek()

This Method returns first day of the week