

Other UI controls

ViewFlipper

- ViewFlipper switches smoothly between two or more views (like TextView, ImageView or any layout) and thus provides a way of transitioning from one view to another through appropriate animations.
- It is an extension of the **ViewAnimator** class which helps to animate between views added to it.
- ViewFlipper makes it easy to switch view. To control over flipping between views ViewFlipper provides two methods **startFlipping()** and **stopFlipping()**.
- To automatically switch between views, add the **autoStart** tag and set its value to true.
- To give more control to user, add views dynamically in the ViewFlipper.
- A ViewFlipper can be used in the gallery app to navigate between the images or videos.

Slider

Create an application to make auto image slider using viewflipper

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
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=".MainActivity">
    <ViewFlipper
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/imgFlipper"></ViewFlipper>
</LinearLayout>
```

MainActivity.java

```
package com.example.slider;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.ImageView;
import android.widget.ViewFlipper;

public class MainActivity extends AppCompatActivity {
    ViewFlipper imgFlipper;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        int sliders[]={
            R.drawable.nature,
            R.drawable.nature1,
            R.drawable.nature2,
            R.drawable.nature3
        };
        imgFlipper=findViewById(R.id.imgFlipper);
        for(int slide:sliders)
        {
            sliderFlipper(slide);
        }
    }
    public void sliderFlipper(int image)
    {
        ImageView imageView=new ImageView(this);
        imageView.setBackgroundResource(image);
        imgFlipper.addView(imageView);
        imgFlipper.setFlipInterval(4000);
        imgFlipper.setAutoStart(true);
        imgFlipper.setInAnimation(this, android.R.anim.fade_in);
        imgFlipper.setOutAnimation(this, android.R.anim.fade_out);
    }
}
```

Progressbar

- In android, **ProgressBar** is a user interface control that is used to indicate the progress of an operation. For example, downloading a file, uploading a file.



Android ProgressBar Control Attributes

| Attribute | Description |
|-----------------------|---|
| android:id | It is used to uniquely identify the control |
| android:max | It is used to specify the maximum value of the progress can take |
| android:progress | It is used to specify default progress value. |
| android:background | It is used to set the background color for a progress bar. |
| android:indeterminate | It is used to enable the indeterminate progress mode. |
| android:padding | It is used to set the padding for left, right, top or bottom of a progress bar. |

Android ProgressBar Example

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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=".MainActivity">
    <ProgressBar
        android:id="@+id/pBar"
        style="?android:attr/progressBarStyleHorizontal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:layout_marginTop="200dp"
        android:minHeight="50dp"
        android:minWidth="200dp"
        android:max="100"
        android:indeterminate="false"
        android:progress="0" />
    <TextView
        android:id="@+id/tView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/pBar"
        android:layout_below="@+id/pBar" />
    <Button
        android:id="@+id/btnShow"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/tView"
        android:layout_marginLeft="130dp"
        android:layout_marginTop="20dp"
        android:text="Start Progress" />
</RelativeLayout>
```

MainActivity.Java

```
package com.example.progressbar;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.Button;
import android.widget.ProgressBar;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    private ProgressBar pgsBar;
    private int i = 0;
    private TextView txtView;
    private Handler hdlr = new Handler();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        pgsBar = findViewById(R.id.pBar);
        txtView = findViewById(R.id.tView);
        Button btn = findViewById(R.id.btnShow);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                i = pgsBar.getProgress();
                new Thread(new Runnable() {
                    public void run() {
                        while (i < 100) {
                            i += 1;
                            // Update the progress bar and display the current value in text view
                            hdlr.post(new Runnable() {
                                public void run() {
                                    pgsBar.setProgress(i);
                                    txtView.setText(i+"/"+pgsBar.getMax());
                                }
                            });
                        }
                    }
                }).start();
            }
        });
    }
}
```


Spinner

- In Android, Spinner provides a quick way to select one value from a set of values.
- Android spinners are nothing but the drop down-list seen in other programming languages.
- In a default state, a spinner shows its currently selected value.
- It provides a easy way to select a value from a list of values.
- In Simple Words we can say that a spinner is like a combo box of AWT or swing where we can select a particular item from a list of items.
- Spinner is a sub class of AsbSpinner class.

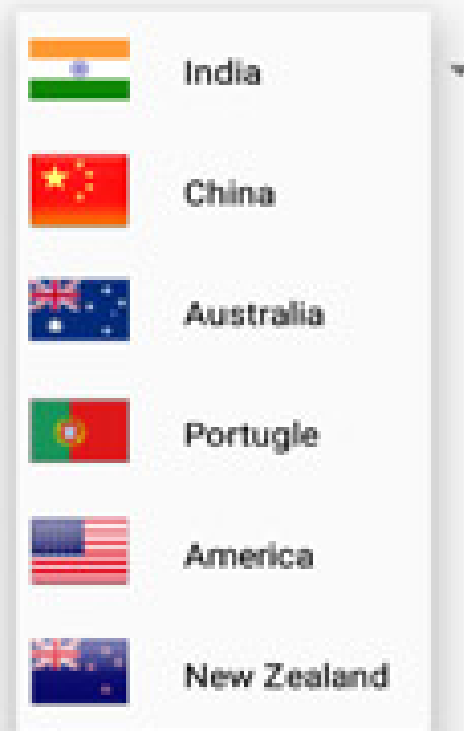
- To fill the data in a spinner we need to implement an adapter class.
- A spinner is mainly used to display only text field so we can implement Array Adapter for that.
- We can also use Base Adapter and other custom adapters to display a spinner with more customize list.
- Suppose if we need to display a textview and a imageview in spinner item list then array adapter is not enough for that.
- Here we have to implement custom adapter in our class.

ArrayAdapter

- An adapter is a bridge between UI component and data source that helps us to fill data in UI component.
- It holds the data and send the data to adapter view then view can takes the data from the adapter view and shows the data on different views like as list view, grid view, spinner.

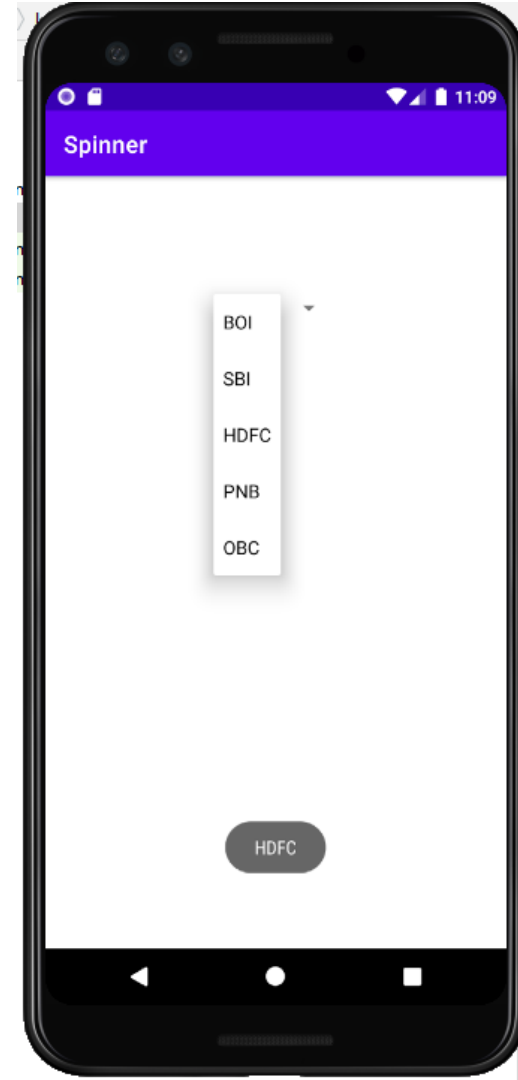


Spinner



Custom Spinner

- Create an application to implement spinner.



activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    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=".MainActivity">

    <Spinner
        android:id="@+id/simpleSpinner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="100dp" />

</RelativeLayout>
```

MainActivity.java

```
package com.example.spinner;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity
{
    String[] bankNames={"BOI","SBI","HDFC","PNB","OBC"};
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //Getting the instance of Spinner and applying OnItemSelectedListener
on it
        Spinner spin = findViewById(R.id.simpleSpinner);
        spin.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener()
        {
            @Override
            public void onItemSelected(AdapterView<?> adapterView, View view,
int i, long l) {
                Toast.makeText(getApplicationContext(), bankNames[i],
Toast.LENGTH_LONG).show();
            }

            @Override
            public void onNothingSelected(AdapterView<?> adapterView) {

            }
        });
    }
}
```

MainActivity.java contd...

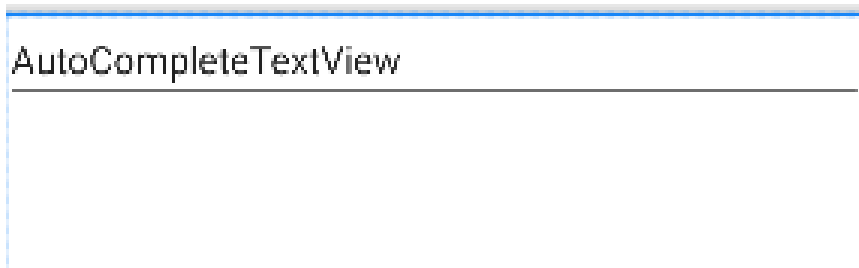
```
//Creating the ArrayAdapter instance having the bank name list
        ArrayAdapter aa = new
ArrayAdapter(this,android.R.layout.simple_spinner_item,ban
kNames);
        // set simple layout resource file for each item of spinner

aa.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
//Setting the ArrayAdapter data on the Spinner which binds data to spinner
        spin.setAdapter(aa);
    }
}
```


- Reference link for custom spinner
- <https://abhiandroid.com/ui/custom-spinner-examples.html>

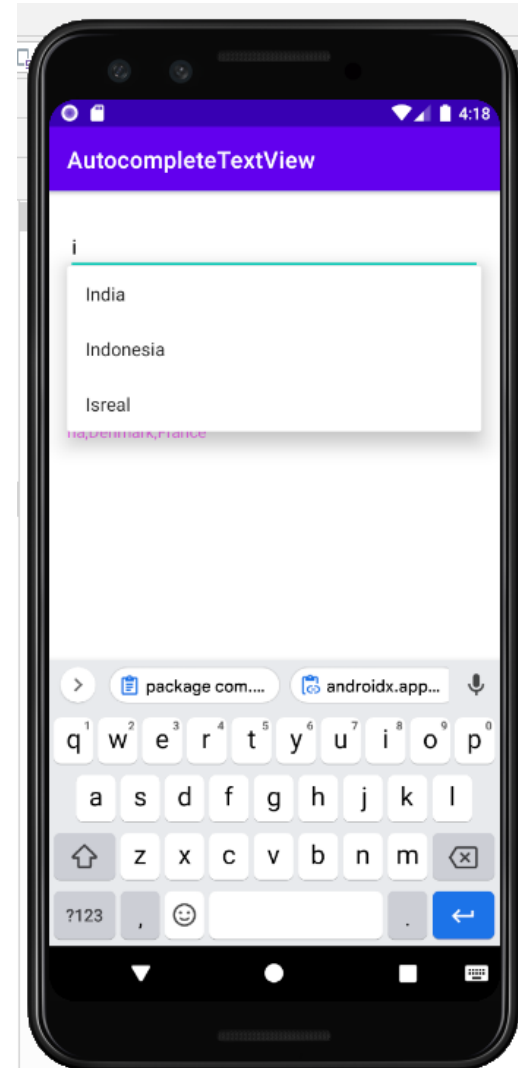
AutoCompleteTextView

- An editable text view that shows completion suggestions automatically while the user is typing.
- The list of suggestions is displayed in a drop down menu from which the user can choose an item to replace the content of the edit box with.
- The drop down can be dismissed at any time by pressing the back key or, if no item is selected in the drop down, by pressing the enter/dpad center key.



AutoCompleteTextView

- Perform AutoCorrect Demo using AutoCompleteTextView



activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:padding="16dp"
    tools:context=".MainActivity">

    <AutoCompleteTextView
        android:id="@+id/acTextView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10dp"
        android:completionThreshold="1"
        android:textColor="@color/black" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="150dp"
        android:textColor="#f6f" />

</RelativeLayout>
```

MainActivity.java

```
package com.example.autocompletetextview;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.AdapterView;
import android.widget.AutoCompleteTextView;
import android.widget.TextView;

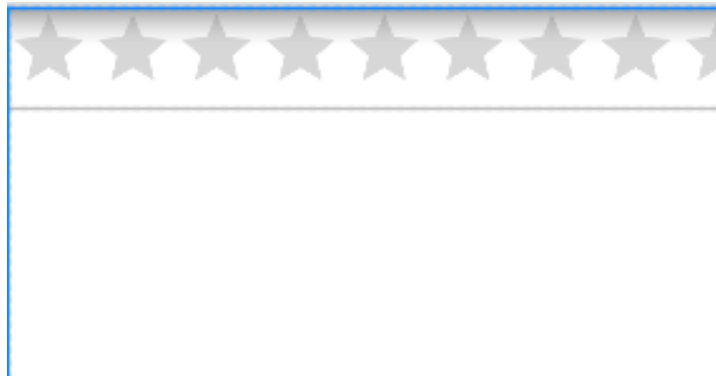
public class MainActivity extends AppCompatActivity {
    AutoCompleteTextView acTextView;
    TextView textView; String [] inputs =

{"India","Indonesia","Isreal","America","Austrilia","Canada","China","Denmark","France"};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        acTextView = (AutoCompleteTextView)findViewById(R.id.acTextView);
        textView = (TextView)findViewById(R.id.textView);
        textView.setText("Text Inputs:\n
India,Indonesia,Isreal,America,Austrilia,Canada,China,Denmark,France");
        ArrayAdapter<String> adapter = new
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,inputs);
        acTextView.setAdapter(adapter);
    }
}
```

Rating Bar

- A RatingBar is an extension of SeekBar and ProgressBar that shows a rating in stars.
- The user can touch/drag or use arrow keys to set the rating when using the default size RatingBar.
- The **getRating()** method of android RatingBar class returns the rating number.



- **Application to demonstrate rating bar**

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    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=".MainActivity">

    <RatingBar
        android:id="@+id/simpleRatingBar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="60dp"
        android:paddingLeft="5dp"
        android:paddingRight="5dp"
        android:rating="3.5" />

    <Button
        android:id="@+id/submitButton"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="109dp"
        android:layout_marginEnd="102dp"
        android:layout_marginBottom="219dp"
        android:background="#f00"
        android:padding="10dp"
        android:text="Submit"
        android:textColor="#fff"
        android:textSize="20sp"
        android:textStyle="bold"
        android:layout_marginRight="102dp"
        android:layout_marginLeft="109dp"
        android:layout_alignParentRight="true"
        android:layout_alignParentLeft="true" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="305dp"
        android:layout_height="70dp"
        android:layout_above="@+id/submitButton"
        android:layout_alignParentEnd="true"
        android:layout_marginEnd="47dp"
        android:layout_marginBottom="83dp"
        android:textSize="25dp"
        android:textStyle="bold"
        android:layout_alignParentRight="true"
        android:layout_marginRight="47dp" />

</RelativeLayout>
```


MainActivity.java

```
package com.example.ratingbar;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RatingBar;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    private EditText text;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        final RatingBar simpleRatingBar = (RatingBar)
findViewById(R.id.simpleRatingBar);
        Button submitButton = (Button) findViewById(R.id.submitButton);
        TextView textView1=findViewById(R.id.textView);
        submitButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String totalStars = "Total Stars:: " +
simpleRatingBar.getNumStars();
                String rating = "Rating :: " + simpleRatingBar.getRating();
                textView1.setText(totalStars+"\n"+rating);
            }
        });
    }
}
```