

AdapterView

AdapterView

- AdapterView is a [ViewGroup](#) that displays items loaded into an adapter. The most common type of adapter comes from an array-based data source

Types

- ArrayAdapter
- ListView-simple listview, multichoice listview, single choice listview
- Spinner
- AutoCompleteTextView
- GridView
- Gallery
- RecyclerView

ArrayAdapter

- The easiest adapter to use is **ArrayAdapter**. All we need to do is wrap around a Java array or `java.util.List` instance. Then, we have a fully functional adapter.

```
String[] arr = {"My", "first", "Android", "list"};
```

```
new ArrayAdapter<String> (this, android.R.layout.simple_list_item_1, arr);
```

The **ArrayAdapter** constructor takes following three parameters:

- The Context to use, typically, this will be our activity instance.
- The resource ID of a view to use, for example, built-in resource ID.
- The actual array or list.

ListView

- Displaying lists is a very common.
- The user gets a list of items that can be scrolled.
- **ListView** is a **ViewGroup** that creates a list of scrollable items.
- The list items are automatically inserted to the list using a **ListAdapter**.
- We make the class extend **ListActivity** instead of **Activity**.
- We call **setListAdapter()** which automatically adds a **ListView** to the **ListActivity**
- The **setTextFilterEnabled(boolean)** method turns on text filtering for the **ListView**, so that when the user begins typing, the list will be filtered.

```
package com.example.arrayadapter;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import android.app.ListActivity;
```

```
import android.os.Bundle;
```

```
import android.widget.ArrayAdapter;
```

```
import android.widget.ListView;
```

```
public class MainActivity extends ListActivity
```

```
{
```

```
    public void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setListAdapter(new ArrayAdapter<String>
```

```
            (this,android.R.layout.simple_list_item_1,countries));
```

```
        ListView l=getListView();
```

```
    }
```

```
    static final String countries[] = { "Algeria", "Argentina", "Australia", "Brazil", "Cote  
d'Ivoire",
```

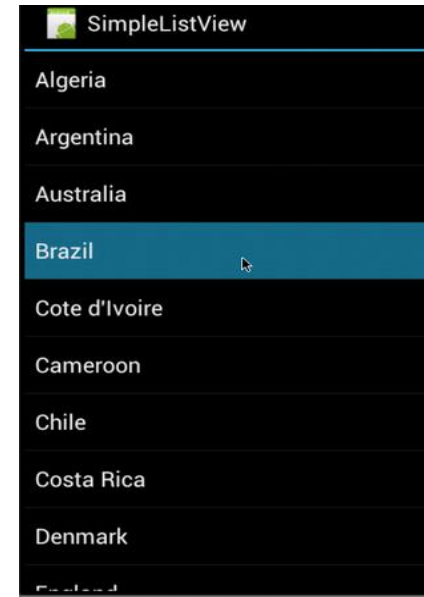
```
    "Cameroon", "Chile", "Costa Rica", "Denmark", "England", "France", "Germany",
```

```
    "Ghana", "Greece", "Honduras",
```

```
    "Italy", "Japan", "Netherlands", "New Zealand", "Nigeria", "North Korea", "Paraguay",
```

```
    "Portugal", "Serbia",
```

```
    "Slovakia", "Slovenia", "South Africa", "South Korea", "Spain", "Switzerland", "United  
States" "Uruguay" };
```



Selecting the Item

- we use attached listener via **setOnItemSelectedListener()** to find out when the selection has changed.
- When an item in the **ListView** is clicked, the **onItemClick()** method is called.

```

public class MainActivity extends ListActivity
{
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setListAdapter(new
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,countries));
        ListView l=getListView();
        l.setTextFilterEnabled(true);
        l.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

                Toast.makeText(getApplicationContext(),parent.getItemAtPosition(position).toString(),
                    Toast.LENGTH_SHORT).show();

            }
        });
    }

    static final String countries[] = { "Algeria", "Argentina", "Australia", "Brazil", "Cote d'Ivoire",
"Cameroon", "Chile", "Costa Rica", "Denmark", "England", "France", "Germany", "Ghana", "Greece",
"Honduras", "Italy", "Japan", "Netherlands", "New Zealand", "Nigeria", "North Korea", "Paraguay",
"Portugal", "Serbia", "Slovakia", "Slovenia", "South Africa", "South Korea", "Spain", "Switzerland",
"United States", "Uruguay" };
}

```

If we modify **onCreate()** method of the previous example, we can get multiple choice list.
Modify the line:

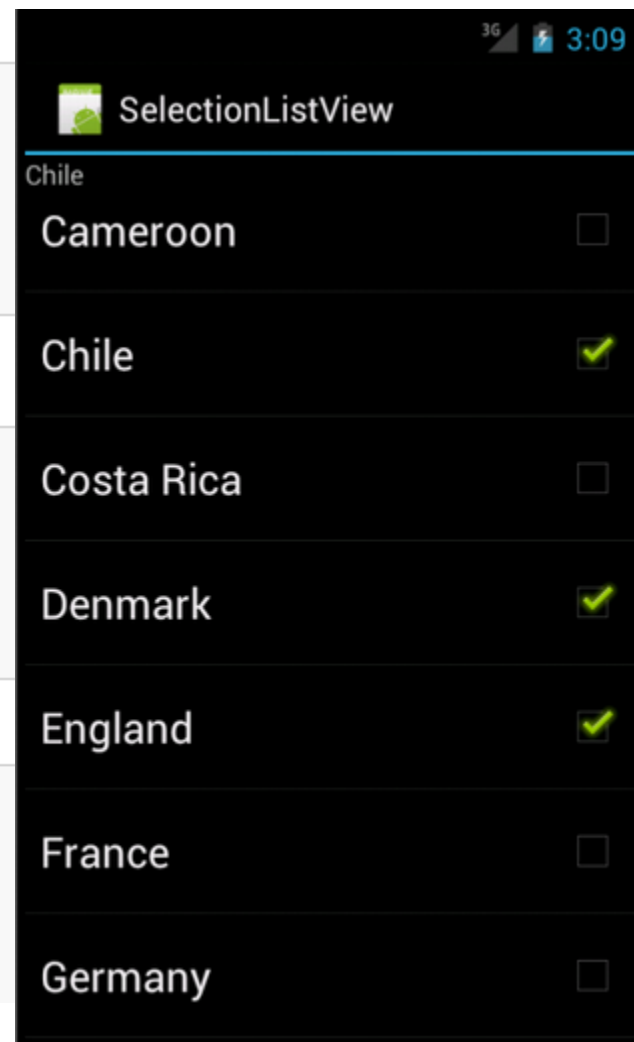
```
setListAdapter(new ArrayAdapter<String>(this,  
    android.R.layout.simple_list_item_1,  
    WORLDCUP2010));
```

To:

```
setListAdapter(new ArrayAdapter<String>(this,  
    android.R.layout.simple_list_item_multiple_choice,  
    WORLDCUP2010));
```

And add the following three line at the end of the method

```
final ListView listView = getListView();  
listView.setItemsCanFocus(false);  
listView.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);
```



Of course, we can do single choice, too.

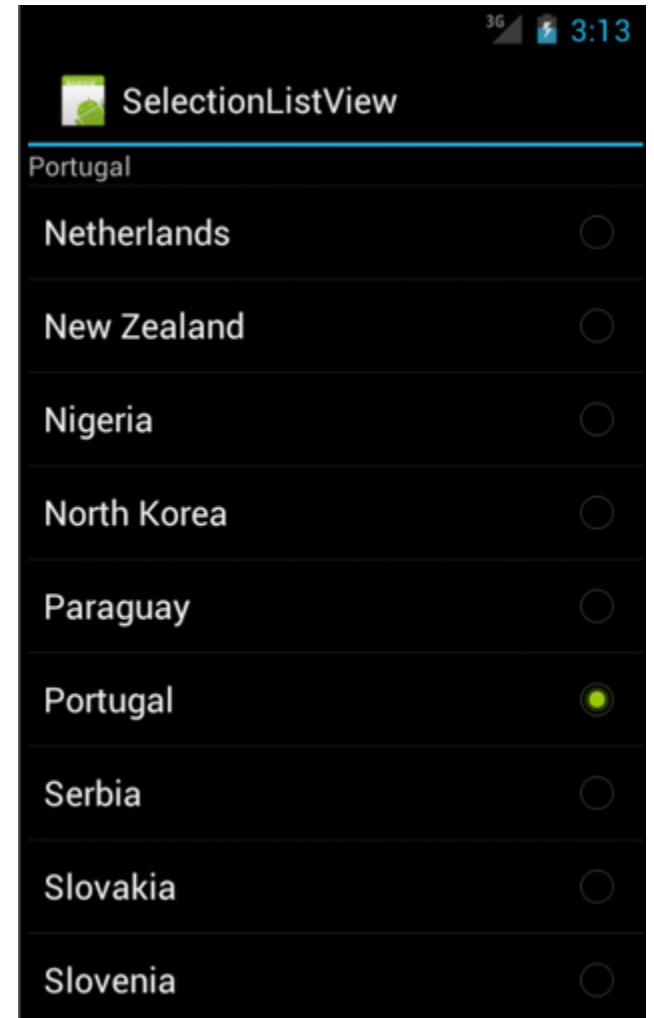
Change

```
android.R.layout.simple_list_item_multiple_choice,  
->  
android.R.layout.simple_list_item_single_choice,
```

and

```
listView.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);  
->  
listView.setChoiceMode(ListView.CHOICE_MODE_SINGLE);
```

Then, we get:



Custom Listview

```
public class MainActivity extends AppCompatActivity {
    static final String names[]={"Apple","Banana","Oranges","grapes","kiwi"};
    static final String desc[]={"Apple varieties","Banana varieties","Oranges
varieties","grapes varieties","kiwi varieties"};
    static final int
fruits[]={R.drawable.apple,R.drawable.banana,R.drawable.orange,R.drawable.grapes,R.drawable.kiwi};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ListView lv=findViewById(R.id.listview);
        lv.setAdapter(new MyAdapter(this));
    }
}
```

- ```

private class MyAdapter extends BaseAdapter {
 Context c;
 public MyAdapter(MainActivity mainActivity) {
 c=mainActivity;
 }

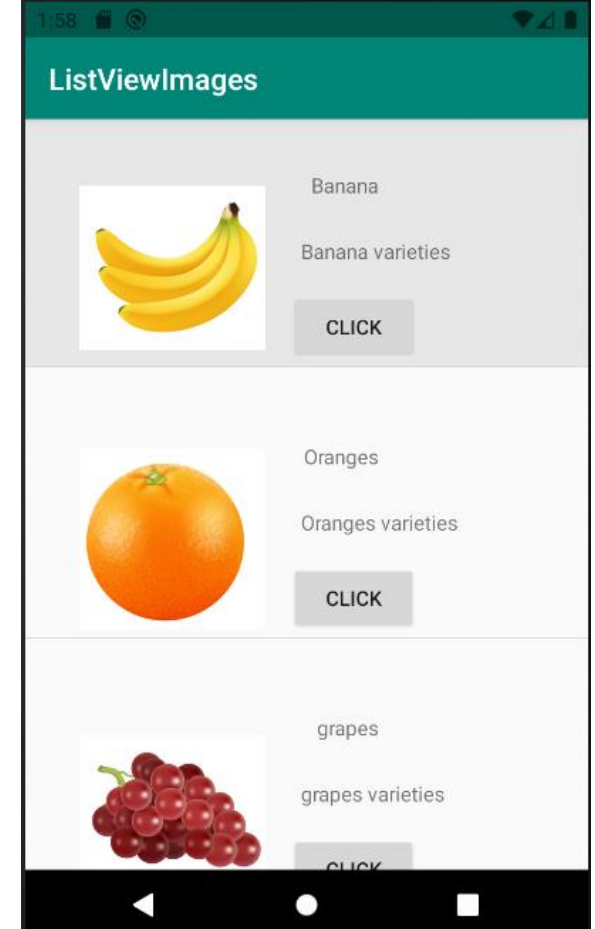
 @Override
 public int getCount() {
 return names.length;
 }

 @Override
 public Object getItem(int position) {
 return null;
 }

 @Override
 public long getItemId(int position) {
 return 0;
 }

 @Override
 public View getView(int position, View convertView, ViewGroup parent) {
 View view=LayoutInflater.from(c).inflate(R.layout.mylistitem,parent,false);
 TextView t1=view.findViewById(R.id.textView);
 TextView t2=view.findViewById(R.id.textView2);
 ImageView I1=view.findViewById(R.id.imageView);
 Button b1=view.findViewById(R.id.button);
 t1.setText(names[position]);
 t2.setText(desc[position]);
 I1.setImageResource(fruits[position]);
 b1.setText("Click");
 return view;
 }
}

```



## *Main\_activity.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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=".MainActivity">
 <ListView
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:id="@+id/listview"

 app:layout_constraintBottom_toBottomOf="parent"
 app:layout_constraintLeft_toLeftOf="parent"

 app:layout_constraintRight_toRightOf="parent"
 app:layout_constraintTop_toTopOf="parent"
 />

</androidx.constraintlayout.widget.ConstraintLayout>
>
```

## *Mylistitem.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"
 android:layout_width="match_parent"
 android:layout_height="match_parent">

 <ImageView
 android:id="@+id/imageView"
 android:layout_width="127dp"
 android:layout_height="134dp"
 android:layout_alignParentStart="true"
 android:layout_alignParentTop="true"
 android:layout_marginStart="37dp"
 android:layout_marginTop="50dp"
 app:srcCompat="@mipmap/ic_launcher" />

 <TextView
 android:id="@+id/textView"
 android:layout_width="wrap_content"
 android:layout_height="30dp"
 android:layout_alignParentTop="true"
 android:layout_alignParentEnd="true"
 android:layout_marginTop="51dp"
 android:layout_marginEnd="143dp"
 android:text="TextView" />

 <TextView
 android:id="@+id/textView2"
 android:layout_width="252dp"
 android:layout_height="30dp"
 android:layout_alignParentTop="true"
 android:layout_alignParentEnd="true"
 android:layout_marginTop="96dp"
 android:layout_marginEnd="-56dp"
 android:text="TextView" />

 <Button
 android:id="@+id/button"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:layout_alignParentTop="true"
 android:layout_alignParentEnd="true"
 android:layout_marginTop="133dp"
 android:layout_marginEnd="116dp"
 android:text="Button" />

</RelativeLayout>
```

# Spinner

- A **Spinner** is a widget that allows the user to select an item from a group.
- It displays one child at a time and lets the user pick among them.
- The items in the Spinner come from the Adapter associated with this view.
- It is similar to a dropdown list and will allow scrolling when the list exceeds the available vertical space on the screen

```

public class MainActivity extends AppCompatActivity {
 String countries[]={"Algeria", "Argentina", "Australia", "Brazil", "Cote d'Ivoire", "Cameroon",
"India","Chile", "Costa Rica", "Denmark", "England", "France", "Germany", "Ghana", "Greece", "Honduras",
"Italy", "Japan", "Netherlands", "New Zealand", "Nigeria", "North Korea", "Paraguay", "Portugal","Serbia",
"Slovakia", "Slovenia", "South Africa", "South Korea", "Spain", "Switzerland", "United States", "Uruguay" }};
 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);

 Spinner sp=findViewById(R.id.choice);
 sp.setAdapter(new ArrayAdapter<String>(this,android.R.layout.simple_dropdown_item_1line,countries));
 sp.setOnItemClickListener(new AdapterView.OnItemClickListener() {
 @Override
 public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
 String text=parent.getItemAtPosition(position).toString();
 Toast.makeText(getApplicationContext(),text,Toast.LENGTH_LONG).show();
 }
 @Override
 public void onNothingSelected(AdapterView<?> parent) {
 }
 });
 }
}

```

# Main\_activity.xml

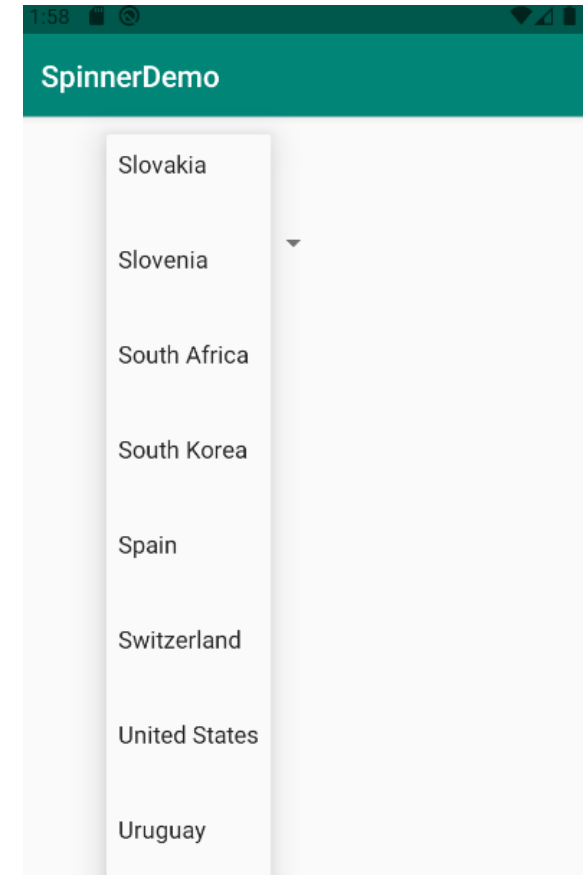
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
 xmlns:android="http://schemas.android.com/apk/res/android"
 xmlns:app="http://schemas.android.com/apk/res-auto"
 xmlns:tools="http://schemas.android.com/tools"
 android:layout_width="match_parent"
 android:layout_height="match_parent"
 tools:context=".MainActivity">

 <Spinner
 android:id="@+id/choice"
 android:layout_marginStart="60dp"
 android:layout_marginTop="60dp"
 app:layout_constraintStart_toStartOf="parent"
 app:layout_constraintTop_toTopOf="parent"
 android:layout_width="150dp"
 android:layout_height="50dp"

 >

 </Spinner>

</androidx.constraintlayout.widget.ConstraintLayout>
```



# AutocompleteTextView

- **AutoCompleteTextView** is an implementation of the **EditText** widget that will provide auto-complete suggestions as the user types.
- The suggestions are extracted from a collection of strings.



```
package com.example.autotextviewadptr;
import androidx.appcompat.app.AppCompatActivity;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.ListActivity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;
import android.widget.ListView;
import android.widget.TextView;
```

```
public class MainActivity extends Activity {
```

```
 @Override
```

```
 protected void onCreate(Bundle savedInstanceState) {
```

```
 super.onCreate(savedInstanceState);
```

```
 setContentView(R.layout.activity_main);
```

```
 AutoCompleteTextView t1=(AutoCompleteTextView)findViewById(R.id.editText);
```

```
 t1.setAdapter(new ArrayAdapter<String>(this,android.R.layout.simple_dropdown_item_1line,countries));
```

```
 }
```

```
 String countries[]={ "Algeria", "Argentina", "Australia", "Brazil", "Cote d'Ivoire", "Cameroon",
"India","Chile", "Costa Rica", "Denmark", "England", "France", "Germany", "Ghana", "Greece", "Honduras",
"Italy", "Japan", "Netherlands", "New Zealand", "Nigeria", "North Korea", "Paraguay", "Portugal","Serbia",
"Slovakia", "Slovenia", "South Africa", "South Korea", "Spain", "Switzerland", "United States", "Uruguay" };
}
```

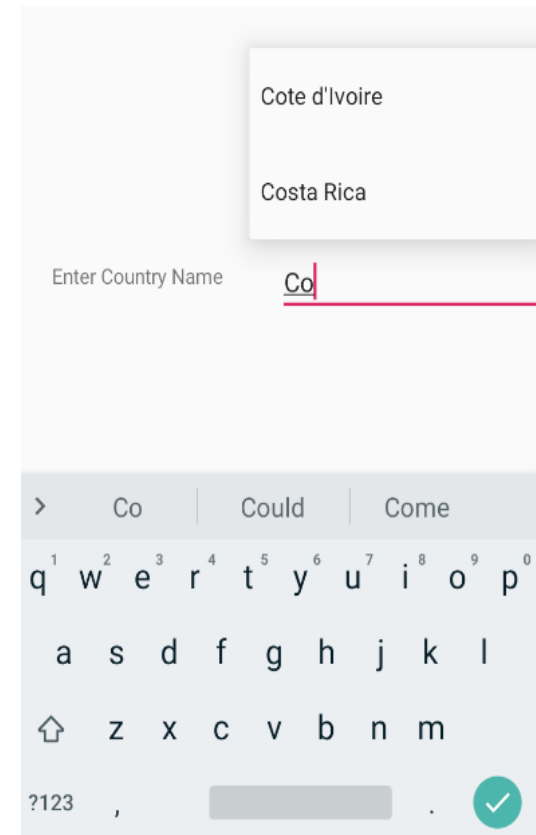
# Activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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=".MainActivity">

 <TextView
 android:id="@+id/textView"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:layout_marginStart="31dp"
 android:layout_marginTop="9dp"
 android:text="Enter Country Name"
 app:layout_constraintStart_toStartOf="parent"
 app:layout_constraintTop_toTopOf="@+id/editText" />

 <AutoCompleteTextView
 android:id="@+id/editText"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:layout_marginStart="40dp"
 android:layout_marginTop="168dp"
 android:ems="10"
 android:inputType="textPersonName"
 android:text="Name"
 app:layout_constraintStart_toEndOf="@+id/textView"
 app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```



# GridView

- A **GridView** displays items in a two-dimensional, scrolling grid.
- We have moderate control over the number and size of the columns and the number of rows is dynamically determined based on the number of items.
- The items are acquired from a **ListAdapter**.

- android:numColumns**

Specifies how many columns there are, or if we supply a value of `auto_fit`,

- Android will compute the number of columns based on the available space and the following properties.

- android:verticalSpacing** and **android:horizontalSpacing**

Indicate how much whitespace there should be between items in the grid.

- android:columnWidth**

Indicates how many pixel wide each column should be.

- android:stretchMode**

Indicates, for grids with `auto_fit` for `android:numColumns`,

```
package com.example.gridviewadpt;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import android.os.Bundle;
```

```
import android.widget.AdapterView;
```

```
import android.widget.GridView;
```

```
public class MainActivity extends AppCompatActivity {
```

```
 @Override
```

```
 protected void onCreate(Bundle savedInstanceState) {
```

```
 super.onCreate(savedInstanceState);
```

```
 setContentView(R.layout.activity_main);
```

```
 GridView gd=findViewById(R.id.grid1);
```

```
 gd.setAdapter(new ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,countries));
```

```
 }
```

```
 String countries[]={ "Algeria", "Argentina", "Australia", "Brazil", "Cote d'Ivoire", "Cameroon", "India", "Chile",
"Costa Rica", "Denmark", "England", "France", "Germany", "Ghana", "Greece", "Honduras", "Italy", "Japan",
"Netherlands", "New Zealand", "Nigeria", "North Korea", "Paraguay", "Portugal", "Serbia", "Slovakia", "Slovenia",
"South Africa", "South Korea", "Spain", "Switzerland", "United States", "Uruguay" };
}
```

GridViewAdpt	
Algeria	Argentina
Australia	Brazil
Cote d'Ivoire	Cameroon
India	Chile
Costa Rica	Denmark
England	France
Germany	Ghana
Greece	Honduras
Italy	Japan
Netherlands	New Zealand
Nigeria	North Korea

# Gridview with images

- User has to define the custom adapter.
- BaseAdapter is a common base class of a general implementation of an [Adapter](#) that can be used in [ListView](#), [GridView](#), Gallery etc.
- Base Adapter can be extended to create a custom Adapter for displaying a custom list items.

# BaseAdapter

```
public class CustomAdapter extends BaseAdapter
{ @Override
 public int getCount()
 {
 return 0;
 }
 @Override
 public Object getItem(int i)
 {
 return null;
 }
 @Override
 public long getItemId(int i)
 {
 return 0;
 }
 @Override public View getView(int i, View view, ViewGroup viewGroup)
 {
 return null;
 }
}
```

## activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<GridView 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:id="@+id/grid1"
 android:numColumns="2"
 android:layout_width="match_parent"
 android:layout_height="match_parent"
 tools:context=".MainActivity">
```

```
</GridView>
```



```
protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);
 GridView g1=findViewById(R.id.gridview);
 g1.setAdapter(new ImageAdapter(this));
}
```

```

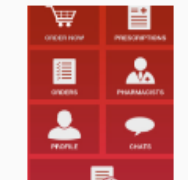
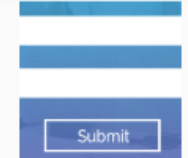
private class ImageAdapter extends BaseAdapter {
 final Context mContext;
 public ImageAdapter(MainActivity mainActivity) {
 mContext=mainActivity;
 }
 @Override
 public int getCount() {
 return img.length;
 }
 @Override
 public Object getItem(int position) {
 return null;
 }

 @Override
 public long getItemId(int position) {
 return 0;
 }
 @Override
 public View getView(int position, View convertView, ViewGroup parent) {

 ImageView i = new ImageView(mContext);
 i.setImageResource(img[position]);
 i.setLayoutParams(new GridView.LayoutParams(200, 200));
 i.setScaleType(ImageView.ScaleType.CENTER_CROP);
 i.setPadding(8, 8, 8, 8);
 return i;
 }
}

```

## GridViewImage



# Gallery

- A view that shows item in a central-locked,horizonatal scrolling list.
- The items in [Gallery](#) are added using [Adapter](#) just like in [ListView](#) or [GridView](#).
- Uses BasedAdapter.

# Main\_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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=".MainActivity">

 <Gallery
 android:id="@+id/gallery"
 android:layout_width="0dp"
 android:layout_height="251dp"
 app:layout_constraintBottom_toBottomOf="parent"
 app:layout_constraintEnd_toEndOf="parent"
 app:layout_constraintHorizontal_bias="0.0"
 app:layout_constraintLeft_toLeftOf="parent"
 app:layout_constraintRight_toRightOf="parent"
 app:layout_constraintStart_toStartOf="parent"
 app:layout_constraintTop_toTopOf="parent"
 app:layout_constraintVertical_bias="1.0" />

 <ImageView
 android:id="@+id/imageView"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:layout_marginStart="157dp"
 android:layout_marginTop="55dp"
 app:layout_constraintStart_toStartOf="parent"
 app:layout_constraintTop_toTopOf="parent"
 app:srcCompat="@mipmap/ic_launcher" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

```
public class MainActivity extends AppCompatActivity {
 int
 img[]={R.drawable.images,R.drawable.layout,R.drawable.login,R.drawable.music,R.draw
 able.login,R.drawable.images};

 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);
 Gallery g=findViewById(R.id.gallery);
 g.setAdapter(new MyAdapter(this));
 g.setOnItemClickListener(new AdapterView.OnItemClickListener() {
 @Override
 public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
 ImageView imageView=findViewById(R.id.imageView);
 imageView.setImageResource(img[position]);
 }
 });
 }
}
```

```

private class MyAdapter extends BaseAdapter {
 Context context;
 public MyAdapter(MainActivity mainActivity) {

 context=mainActivity;
 TypedArray a=context.obtainStyledAttributes(R.styleable.MyGallery);
 int im=a.getResourceId(R.styleable.MyGallery_android_galleryItemBackground,0);
 a.recycle();
 }

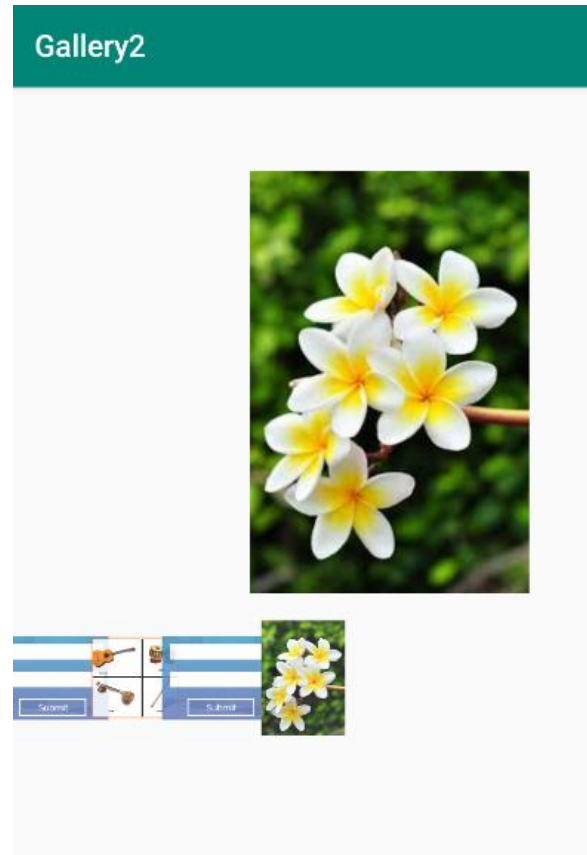
 @Override
 public int getCount() {
 return img.length;
 }

 @Override
 public Object getItem(int position) {
 return null;
 }

 @Override
 public long getItemId(int position) {
 return 0;
 }

 @Override
 public View getView(int position, View convertView, ViewGroup parent) {
 ImageView imageView=new ImageView(context);
 imageView.setImageResource(img[position]);
 imageView.setLayoutParams(new Gallery.LayoutParams(150,150));
 imageView.setPadding(20,20,20,20);
 imageView.setScaleType(ImageView.ScaleType.CENTER_CROP);
 return imageView;
 }
}

```



# Recycler View

- **Data**
- **RecyclerView** scrolling list for list items—[RecyclerView](#)
- **Layout** for one item of data—XML file
- **Layout manager** handles the organization of UI components in a view—[Recyclerview.LayoutManager](#)
- **Adapter** connects data to the RecyclerView—[RecyclerView.Adapter](#)
- **View holder** has view information for displaying one item—[RecyclerView.ViewHolder](#)

# Main\_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.recyclerview.widget.RecyclerView
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="wrap_content"
 android:id="@+id/recycler"
 tools:context=".MainActivity">

</androidx.recyclerview.widget.RecyclerView>
```

# my\_text\_view.xml

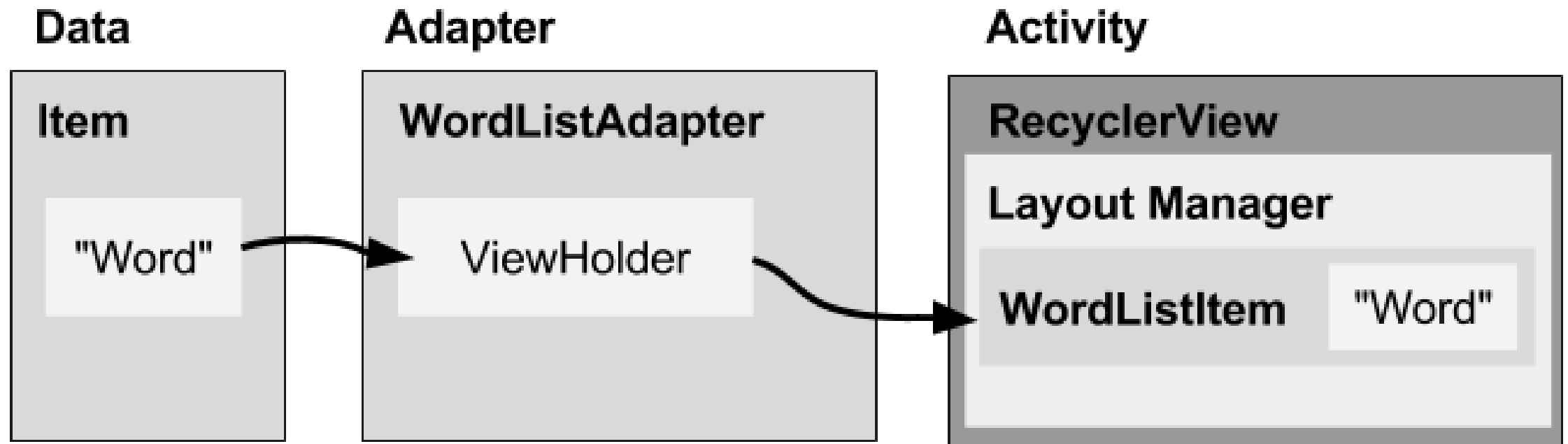
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
 android:layout_width="match_parent"
 android:layout_height="wrap_content">
 <TextView
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:id="@+id/textview1"/>

</LinearLayout>
```



# Components

## How components fit together overview



# What is a layout manager?

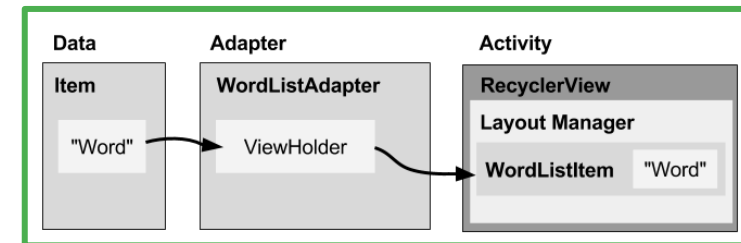
- All view groups have layout managers
- Positions item views inside a [RecyclerView](#).
- Reuses item views that are no longer visible to the user
- Built-in layout managers include [LinearLayoutManager](#), [GridLayoutManager](#), and [StaggeredGridLayoutManager](#)
- For RecyclerView, extend [RecyclerView.LayoutManager](#)

# What is an adapter?

- Helps incompatible interfaces work together, for example, takes data from a database [Cursor](#) and puts them as strings into a view
- Intermediary between data and view
- Manages creating, updating, adding, deleting item views as the underlying data changes
- [RecyclerView.Adapter](#)

# What is a view holder?

- Used by the adapter to prepare one view with data for one list item
- Layout specified in an XML resource file
- Can have clickable elements
- Is placed by the layout manager
- [RecyclerView.ViewHolder](#)



```
package com.example.recyclerview1;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Context;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.ViewGroup;
import android.widget.LinearLayout;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
 static final String countries[] = {"Algeria", "Argentina", "Austin", "Australia", "Brazil", "Cote d'Ivoire", "Cameroon",
 "Chile", "Costa Rica", "Denmark", "England", "France", "Germany", "Ghana", "Greece", "Honduras", "Italy", "Japan", "Netherlands",
 "New Zealand", "Nigeria", "North Korea", "Paraguay", "Portugal", "Serbia", "Slovakia", "Slovenia", "South Africa", "South Korea",
 "Spain", "Switzerland", "United States", "Uruguay"};

 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);
 RecyclerView recyclerView = findViewById(R.id.recycler);
 recyclerView.setLayoutManager(new LinearLayoutManager(this));
 MyAdapter myAdapter=new MyAdapter(this,countries);
 recyclerView.setAdapter(myAdapter);
 }
}
```

```

import androidx.recyclerview.widget.RecyclerView;
class MyAdapter extends RecyclerView.Adapter<MyAdapter.Myholder> {
 Context mycontext;
 String mycountries[];
 public MyAdapter(Context context,String countires[]) { /constructor to receive names and context
 mycontext=context;
 mycountries=countires;
 }
 @NonNull
 @Override
 public Myholder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
 LayoutInflater inflater=LayoutInflater.from(mycontext);
 View v=inflater.inflate(R.layout.my_text_view,parent,false);
 return new Myholder(v);
 }

 @Override
 public void onBindViewHolder(@NonNull Myholder holder, int position) {
 holder.textView.setText(mycountries[position]);
 }

 @Override
 public int getItemCount() {
 return mycountries.length;
 }

 public class Myholder extends RecyclerView.ViewHolder {
 TextView textView;
 public Myholder(@NonNull View itemView) {
 super(itemView);
 textView=itemView.findViewById(R.id.textview1);
 }
 }
}

```

# Recyclerview with images

```
package com.example.recyclerimage;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {
 final String languages[]={"c", "cpp", "java", "python", "kotlin", "html", "javascript"};
 final int images[]={R.drawable.c,R.drawable.cpp,R.drawable.java,R.drawable.python,
R.drawable.kotlin,R.drawable.html,R.drawable.javaccrpt};

 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);
 RecyclerView recyclerView=findViewById(R.id.recyclerview);
 recyclerView.setLayoutManager(new LinearLayoutManager(this));
 MyAdapter myAdapter=new MyAdapter(this, languages, images);
 recyclerView.setAdapter(myAdapter);
 }
}
```

```

class MyAdapter extends RecyclerView.Adapter<MyAdapter.MyHolder> {
 Context myContext;
 String languages[];
 int images[];
 MyAdapter(Context c, String l[], int i[])
 {
 myContext=c;
 languages=l;
 images=i;
 }
 @NonNull
 @Override
 public MyHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
 LayoutInflater inflater=LayoutInflater.from(myContext);
 View v=inflater.inflate(R.layout.myitem,parent,false);
 return new MyHolder(v);
 }
 @Override
 public void onBindViewHolder(@NonNull MyHolder holder, int position) {
 holder.i1.setImageResource(images[position]);
 holder.t1.setText(languages[position]);
 }
 @Override
 public int getItemCount() {
 return languages.length;
 }

 public class MyHolder extends RecyclerView.ViewHolder {
 TextView t1;
 ImageView i1;
 public MyHolder(@NonNull View itemView) {
 super(itemView);
 t1=itemView.findViewById(R.id.textView);
 i1=itemView.findViewById(R.id.imageView);
 }
 }
}

```



## Activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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=".MainActivity">

 <androidx.recyclerview.widget.RecyclerView
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:id="@+id/recyclerview"
 tools:ignore="MissingConstraints" />
</androidx.constraintlayout.widget.ConstraintLayout>
</>
```

## myitem.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout
 xmlns:android="http://schemas.android.com/apk/res/android"
 xmlns:app="http://schemas.android.com/apk/res-auto"
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 >

 <ImageView
 android:id="@+id/imageView"
 android:layout_width="wrap_content"
 android:layout_height="100dp"
 android:layout_weight="1"
 android:layout_gravity="left"
 app:srcCompat="@mipmap/ic_launcher" />

 <TextView
 android:id="@+id/textView"
 android:layout_width="wrap_content"
 android:layout_height="50dp"
 android:layout_weight="1"
 android:text="TextView"
 android:gravity="left"
 android:textSize="30sp"/>

</TableLayout>
```