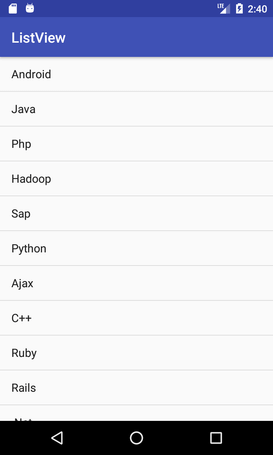
**Android Lectures:**

* **Thread:** (used as a luncher) Use for the multiple tasks that we want to run parallel to run thread we have to use run function after making its object

Thread th =new Thread() {  
 public void run() {  
 try {  
 *sleep*(5000);  
 } catch (  
 InterruptedException e) {  
 e.printStackTrace();  
 } finally {  
 Intent In = new Intent(Launcher.this, MainActivity.class);  
 startActivity(In);  
 }  
 }  
};  
th.start();

* **Calculator Code (Professional)**
* private static final String *TAG* = "error";  
  EditText tx;  
  Button[] btn=new Button[16];  
  String data;  
  String op;  
  ScriptEngineManager scm = new ScriptEngineManager();  
  ScriptEngine jsEngine = scm.getEngineByName("js");  
  int check=0;  
    
    
  @Override  
  protected void onCreate(Bundle savedInstanceState) {  
   super.onCreate(savedInstanceState);  
   setContentView(R.layout.*activity\_main*);  
   tx=findViewById(R.id.*text*);  
   tx.setText("");  
   for(int i=0 ; i<16 ;i++) {  
   String s\_b="btn"+(i+1);  
   int resIDb = getResources().getIdentifier(s\_b, "id", getPackageName());  
   btn[i] = findViewById(resIDb);  
    
    
   }  
    
   try {  
   for (int j = 0; j < btn.length; j++) {  
    
   final int finalId=j;  
   btn[j].setOnClickListener(v -> {  
   if(check==1)  
   {  
   tx.setText("");  
   check=0;  
   }  
   data=String.*valueOf*(tx.getText());  
   op=(String) btn[finalId].getText();  
   if(!op.equals("AC")) {  
   tx.setText(data += (String) btn[finalId].getText());  
   }  
   switch (op) {  
    
   case "AC":  
   if(data.length()>0) {  
   data = data.substring(0, data.length() - 1);  
   tx.setText(data);  
   }else{  
   tx.setText("");  
   }  
   break;  
    
   case "=":  
   try{  
   double res = (double)jsEngine.eval(data.replaceAll("[=]",""));  
   tx.setText(String.*valueOf*(res));  
   Toast.*makeText*(this,String.*valueOf*(res),Toast.*LENGTH\_SHORT*).show();  
   check=1;  
   }catch(Exception e){  
   Log.*i*("error",e.getCause().toString());  
   }  
    
    
   break;  
   }  
    
    
   });  
    
    
   }  
   }  
   catch (NullPointerException e)  
   {  
   Log.*i*(*TAG*, "onCreate: "+e);  
   }
* **List Array:**

**For simple text list array we use this code as Array list**

ListView Iv =(ListView) findViewById(R.id.*lst*);  
ArrayList<String> arry=new ArrayList<>();  
arry.add("android");  
arry.add("Kitkat");  
ArrayAdapter ad=new ArrayAdapter(this, android.R.layout.*simple\_list\_item\_1*,arry);  
Iv.setAdapter(ad);

Output------------------------------------------------------🡪

* **List View for image and text both based on Array list**

**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"  
 android:background="@drawable/pic2"  
 tools:context=".MainActivity">  
  
  
 <TextView  
 android:id="@+id/h1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginLeft="20dp"  
 android:fontFamily="cursive"  
 android:text="Food Mania"  
 android:textColor="@color/white"  
 android:textSize="40sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <ImageView  
 android:layout\_width="50dp"  
 android:layout\_height="50dp"  
 android:layout\_marginTop="8dp"  
 android:layout\_marginEnd="16dp"  
 android:layout\_marginRight="16dp"  
 android:src="@drawable/ic\_baseline\_shopping\_cart"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 />  
  
  
  
<ScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginTop="80dp"  
 >  
<androidx.constraintlayout.widget.ConstraintLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
 <EditText  
 android:id="@+id/search"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="20dp"  
 android:background="@drawable/search\_backg"  
 android:drawableStart="@drawable/ic\_baseline\_search\_24"  
 android:drawableLeft="@drawable/ic\_baseline\_search\_24"  
 android:drawablePadding="20dp"  
 android:fontFamily="sans-serif"  
 android:hint=" Search"  
 android:padding="15dp"  
 android:textColorHint="@color/white"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/h1" />  
  
 <TextView  
 android:id="@+id/h2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="32dp"  
 android:layout\_marginLeft="20dp"  
 android:layout\_marginTop="20dp"  
 android:fontFamily="@font/amaranth"  
 android:text="Featured Product:"  
 android:textColor="@color/white"  
 android:textSize="25sp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/search" />  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/rv"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 app:layout\_constraintTop\_toBottomOf="@+id/h2"  
 tools:layout\_editor\_absoluteX="11dp"  
 tools:layout\_editor\_absoluteY="213dp" />  
</androidx.constraintlayout.widget.ConstraintLayout>  
</ScrollView>  
</androidx.constraintlayout.widget.ConstraintLayout>

**Main Activity:**

ListView Iv =(ListView) findViewById(R.id.*rv*);  
P\_ArrayAdapter pd=new P\_ArrayAdapter(this,R.layout.*row*,arry);  
Iv.setAdapter(pd);

**Adapter class:**

public class P\_ArrayAdapter extends ArrayAdapter<Products> {  
 Context ct;  
 int res;  
 public P\_ArrayAdapter(@NonNull Context context, int resource, @NonNull ArrayList<Products> objects) {  
 super(context, resource,objects);  
 this.ct=context;  
 this.res=resource;  
 }  
  
 @Nullable  
 @Override  
 public Products getItem(int position) {  
 return super.getItem(position);  
 }  
  
 @NonNull  
 @Override  
 public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {  
 LayoutInflater layoutInflater=LayoutInflater.*from*(ct);  
 convertView =layoutInflater.inflate(res,parent,false);  
  
 ImageView imageView = (ImageView) convertView.findViewById(R.id.*imageView*);  
 TextView textView = (TextView) convertView.findViewById(R.id.*textView*);  
  
 imageView.setImageResource(getItem(position).getImgId());  
 textView.setText(getItem(position).getDescription());  
  
 return convertView;  
 }  
}

**Model Class: (To save data in class)**

public class Products {  
  
 private String product\_name;  
 private int image;  
 public Products(String p\_name, int image) {  
 this.product\_name = p\_name;  
 this.image = image;  
 }  
 public String getDescription() {  
 return product\_name;  
 }  
 public void setDescription(String product\_name) {  
 this.product\_name = product\_name;  
 }  
 public int getImgId() {  
 return image;  
 }  
 public void setImgId(int image) {  
 this.image = image;  
 }  
  
  
  
}

**Single row.xml**

<?xml version="1.0" encoding="utf-8"?>  
  
<androidx.cardview.widget.CardView 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/linear"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="260dp"  
 android:layout\_margin="5dp"  
 >  
<LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 >  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="200dp"  
 android:layout\_height="200dp"  
 android:src="@drawable/ic\_launcher\_background"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_marginBottom="4dp"  
 android:gravity="center"  
 android:padding="10dp"  
 android:text="product\_name"  
 android:textSize="22dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 tools:ignore="MissingConstraints" />  
  
</LinearLayout>  
  
  
</androidx.cardview.widget.CardView>

Output----------------------------------------🡪

* **Recycler View(based on Array List)**

**Mainactivity.java**

mRecyclerView= findViewById(R.id.*rv*);  
recycler\_pview rc=new recycler\_pview(this,db.getdata());  
mRecyclerView.setHasFixedSize(true);  
mRecyclerView.setLayoutManager(new GridLayoutManager(this,2,GridLayoutManager.*VERTICAL*,false));  
mRecyclerView.setAdapter(rc);

**Recycler Adapter:**

public class recycler\_pview extends RecyclerView.Adapter<recycler\_pview.mView> {  
  
 private static final String *TAG* = "CustomAdapter";  
 Context ct;  
 ArrayList<Products> marray;  
 public recycler\_pview(Context context , ArrayList<Products> objects ) {  
  
 ct=context;  
 this.marray=objects;  
 }  
  
 @NonNull  
 @Override  
 public mView onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view= LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*row*,parent,false);  
 return new mView(view);  
 }

**Product class: as above**

**Single row class : as above**

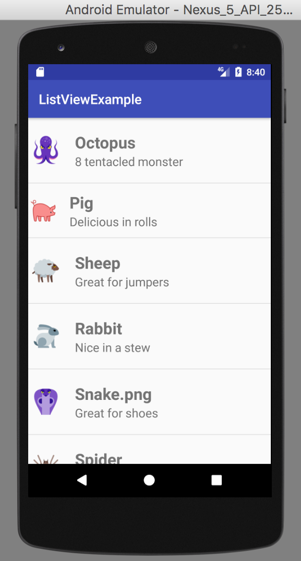
* **Recycler View (Based on array)**

**Adapter Class:**

public class recycler\_view extends RecyclerView.Adapter<recycler\_view.myView> {  
  
 private static final String *TAG* = "CustomAdapter";  
 Context ct;  
 String[] name ;  
 int[] image;  
 int[] rating ;  
 int[] price;  
 public recycler\_view(Context context , String n[], int img[], int rat[], int pri[] ) {  
 ct=context;  
 name=n;  
 image=img;  
 rating=rat;  
 price=pri;  
 }  
  
 @NonNull  
 @Override  
 public myView onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  
 View view= LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*row*,parent,false);  
 return new myView(view);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull myView holder, int position) {  
 String name\_= name[position];  
 int image\_= image[position];  
 holder.textView.setText(name\_);  
 holder.imageView.setImageResource(image\_);  
  
 }  
  
 @Override  
 public int getItemCount() {  
 return name.length;  
 }  
  
  
 public static class myView extends androidx.recyclerview.widget.RecyclerView.ViewHolder {  
 public ImageView imageView;  
 public TextView textView;  
 public CardView lLayout;  
 public myView(@NonNull View itemView) {  
 super(itemView);  
 this.imageView = (ImageView) itemView.findViewById(R.id.*imageView*);  
 this.textView = (TextView) itemView.findViewById(R.id.*textView*);  
 lLayout = (CardView) itemView.findViewById(R.id.*linear*);  
  
 }  
 }  
  
  
}

**MainActivity.java**

mRecyclerView= findViewById(R.id.*rv*);  
recycler\_pview rc=new recycler\_pview(this,name,image,rating,price);  
mRecyclerView.setHasFixedSize(true);  
mRecyclerView.setLayoutManager(new GridLayoutManager(this,1,GridLayoutManager.*VERTICAL*,false));  
mRecyclerView.setAdapter(rc);

**Single row as above**

**Fire Base data Base:**

**Tool->firebase connect**

**Single node no child:**

**Write Data:**

FirebaseDatabase database = FirebaseDatabase.*getInstance*();  
DatabaseReference myRef = database.getReference();  
myRef.setValue("Hello there");

**Read Data:**

**(Real time with addValueEventListner)**

**Single Read with addListenerForSingleValueE**

myRef.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 String data= snapshot.getValue(String.class);  
 tx.setText(data);  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
});

**Set or get child Node:**

FirebaseDatabase database = FirebaseDatabase.*getInstance*();  
DatabaseReference myRef = database.getReference("user");  
myRef.child("user1").setValue("Hello there");  
  
myRef.child("user1").addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot snapshot) {  
 String data= snapshot.getValue(String.class);  
 tx.setText(data);  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError error) {  
  
 }  
});

**On success and Failure listener:**

myRef.child("user1").setValue("Hello there").addOnSuccssListener….

**How to read node value with map:**

Map<String,Object> data= (Map<String,Object>) dataSnapshot.getValues();

Data.get(“Name”)

Data.get(“Age”);

**Update value:  
mRef.child(‘user1’).child(“Name”).setValue(data);**

**Work with Push Ids:**

**String key-mREf.push().getKey();**

**mRef.child(key).chid(“Name”).setvalue(data);**

**mRef.child(key).child(“Age”).setvalue(age);**

**Update Data**

**Map<String,Object> updatev=new HashMap<>();**

**Updatev.put(“/user1/Name”,name);**

**Updatev.put(“/pushid/Age”,age);**

**mRef.UpdateChilderen(Updatev);**

**Remove Data:**

**mRef.child(‘user1’).child(‘Name’).removeValue();**