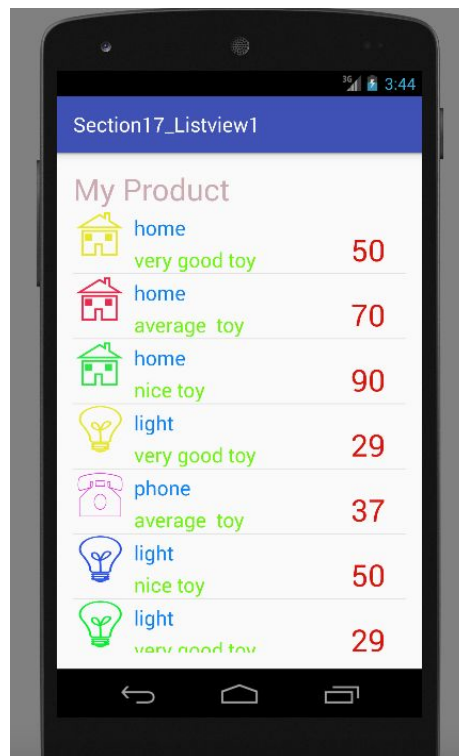


פרק 15 - ListView

ListView הוא אובייקט המורכב מתאים זהים. כל תא ב ListView יורכב מאותו layout

בסיסי:

נוכל לראות בדוגמה שלנו שכל תא מורכב מתמונה ושלושה טקסטים (כולל המספר):



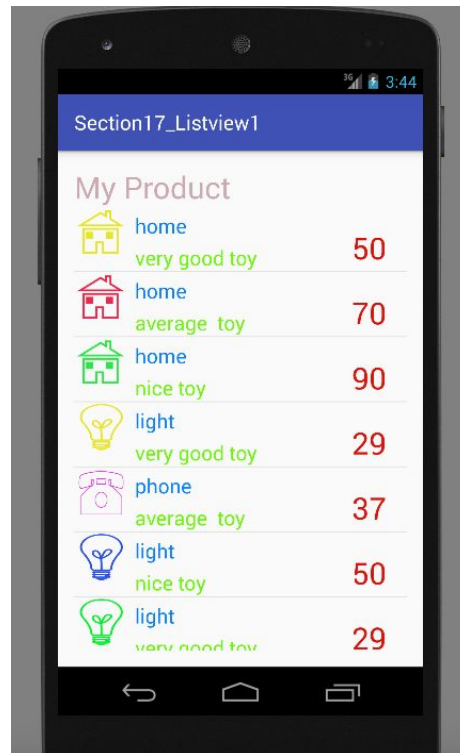
בשביל לבנות ListView נעקוב אחר השלבים הבאים:

1. ניצור layout ונעצב אותו כרצוננו. בהמשך נדחוף layout זה לכל שורה.
2. ניצור java class שמכיל את התכונות שעל כל תא ב - ListView
3. ניצור class adapter משלנו בשם כלשהו לדוגמה MyAdapter שיורש מ <code>ArrayAdapter<class>
4. נממש constructor
5. נממש את הפונקציה <code>getView</code>
6. ניצור מבנה נתונים בהתאם לרצוננו ב- ב - MainActivity לדוגמה <code>ArrayList</code> ונוסיף לו נתונים.
7. ניצור מופע של <code>ArrayAdapter</code> ונעביר לו את מבנה הנתונים.
8. נפעיל את הפונקציה <code>setAdapter</code> של ה-ListView ונעביר לה את ה- <code>ArrayAdapter</code> כפרמטר

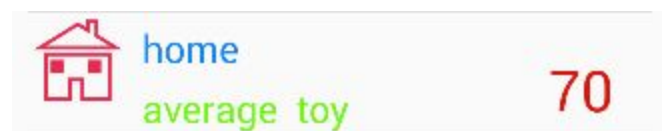
נריץ

ListView הוא אובייקט חשוב מאוד וניתן לראות אותו כמעט בכל אתר ואפליקציה.
לדוגמה: כשאנחנו נמצאים באימייל אנחנו רואים ListView של כל המיילים שלנו.

האפליקציה שנבנה תיראה כך:



כל custom cell (תא) ב ListView יראה כך:



נוכל לערוך כל custom cell ולמחוק תאים שאנחנו לא רוצים.

שלב 1 - עריכת layout:

נעצב Layout שיהיה תבנית לכל שורה.

נעמוד על תיקיית Layout ניצור Layout וניתן לו שם כלשהו. לדוגמה custom_layout.

ה- layout בדוגמה שלנו יכיל תמונה, 2 שדות טקסט לכותרת המוצר ומחיר המוצר.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >

    <ImageView
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:id="@+id/ivProduct"
    />

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
    >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/tvTitle"
            android:textSize="20dp"
            android:text="title"
            android:layout_marginTop="5dp"
            android:layout_marginLeft="10dp"
            android:textColor="#0e82f5"
        />

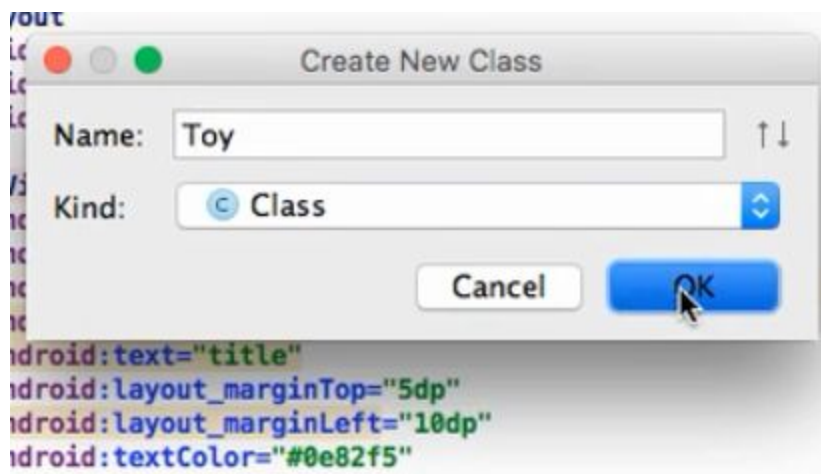
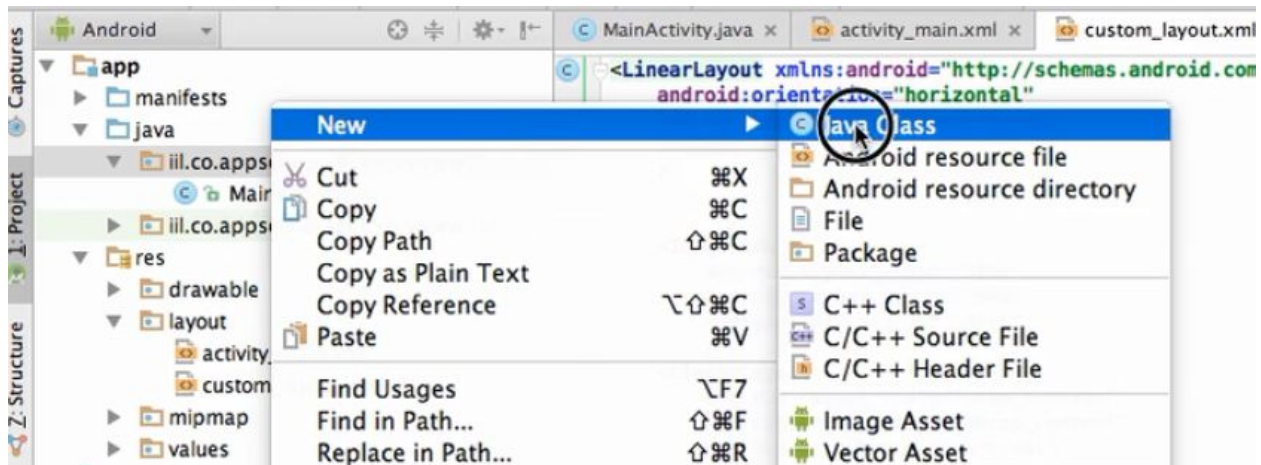
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
```

```
        android:id="@+id/tvSubTitle"
        android:textSize="20dp"
        android:text="subtitle"
        android:layout_marginTop="5dp"
        android:layout_marginLeft="10dp"
        android:textColor="#6ef50e"
    />
</LinearLayout>

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/tvPrice"
    android:textColor="#dc130c"
    android:textSize="30sp"
    android:text="50$"
    android:gravity="right"
    android:layout_marginRight="20dp"
    android:layout_marginTop="20dp"
/>

</LinearLayout>
```

שלב 2 - בניית class שמתאר את התא.



```
public class Toy {
```

```
    private int price;  
    private String title;  
    private String subTitle;  
    private Bitmap bitmap;
```

נצהיר על int למחיר, שני String
לטקסטים ו Bitmap לתמונה

```
    public Toy(int price, String title, String subTitle, Bitmap bitmap)  
    {
```

נבנה constructor מתאים

```
        this.price = price;  
        this.title = title;  
        this.subTitle = subTitle;  
        this.bitmap = bitmap;  
    }
```

```
public int getPrice() {  
    return price;  
}  
public void setPrice(int price) {  
    this.price = price;  
}  
public String getTitle() {  
    return title;  
}  
public void setTitle(String title) {  
    this.title = title;  
}  
public String getSubTitle() {  
    return subTitle;  
}  
public void setSubTitle(String subTitle) {  
    this.subTitle = subTitle;  
}  
  
public Bitmap getBitmap() {  
    return bitmap;  
}  
  
public void setBitmap(Bitmap bitmap) {  
    this.bitmap = bitmap;  
}  
  
}
```

Setters ו Getters מתאימים

שלב 3 - בניית ArrayAdapter

בשלב הזה נבנה ClassAdapter שיורש מ ArrayAdapter.

מה תפקיד ה Adapter?

תפקיד ה Adapter הוא לקחת רשימה של Toy (הקלאס שמתאר כל תא) ולשים אותם לתוך ה ListView, כלומר לשים אותם תא אחרי תא במסודר.

נבנה קודם כל קלאס שיורש מ ArrayAdapter גנרי של Toy:

```
public class ToyAdapter extends ArrayAdapter<Toy> {
```

לקלאס יהיו שתי תכונות:

```
Context context;
```

```
List<Toy> objects;
```

האקטיביטי שיצר את ה ListView

רשימה של התאים

נבנה גם constructor מתאים:

```
public ToyAdapter(Context context, int resource, int textViewResourceld, List<Toy> objects) {
```

```
    super(context, resource, textViewResourceld, objects);
```

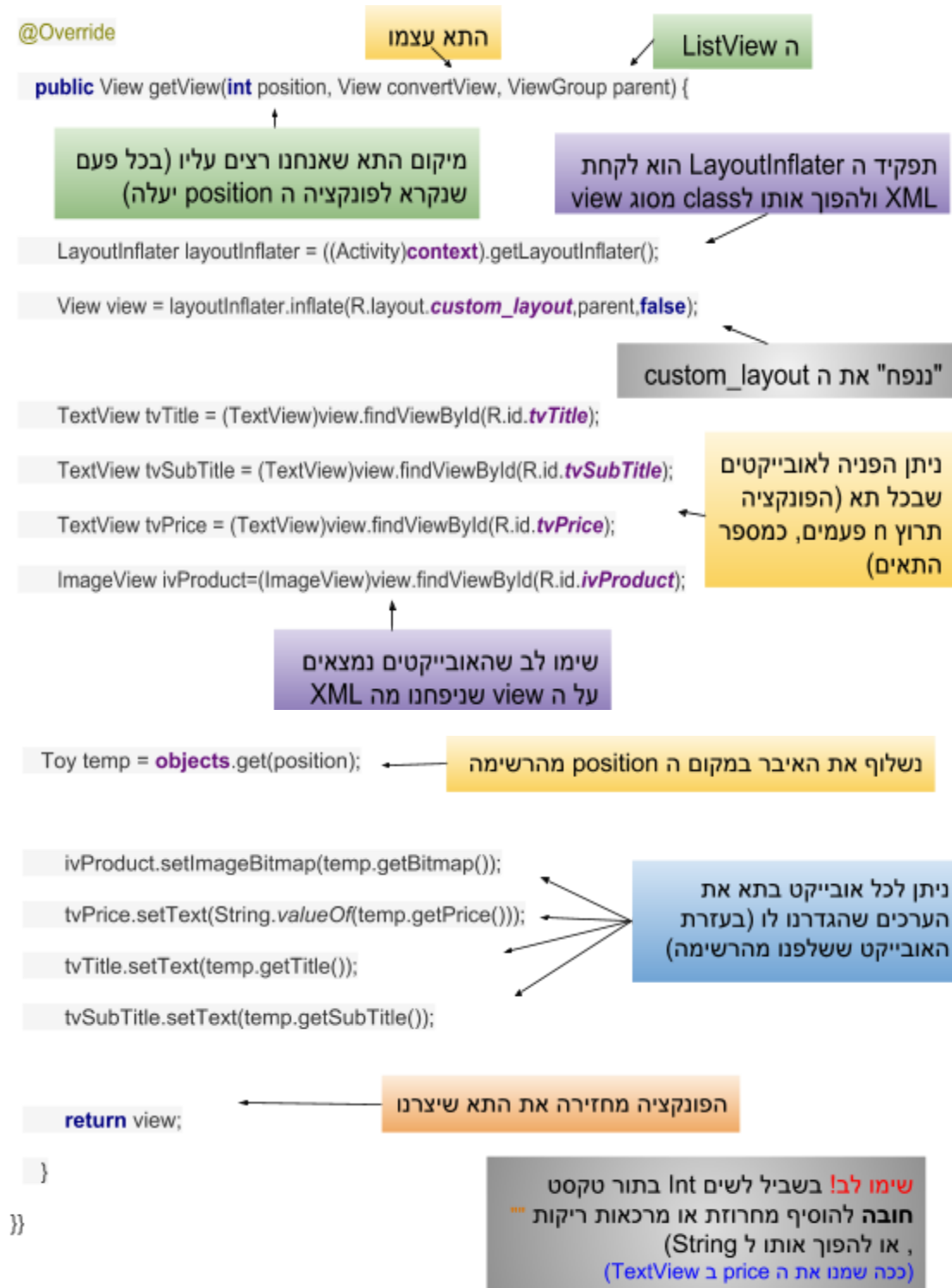
```
    this.context=context;
```

```
    this.objects=objects;
```

```
}
```

נממש את הפונקציה getView של ה Adapter:

ברגע שנבנה את ה ListView הפונקציה הזו תרוץ n פעמים (n גודל הרשימה)



שלב 4 - MainActivity

בשלב הזה נראה איך להשתמש בכל הקלאסים שבנינו ולחבר אותם ל ListView.
 בקובץ ה XML נבנה ListView:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:id="@+id/l"
    android:orientation="vertical"
    tools:context="iil.co.appschool.sec15_listview.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="ListView! Example" />

    <ListView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/lv"
    >

    </ListView>

</LinearLayout>
```

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

```
    ArrayList<Toy> toyList;
```

```
    ListView lv;
```

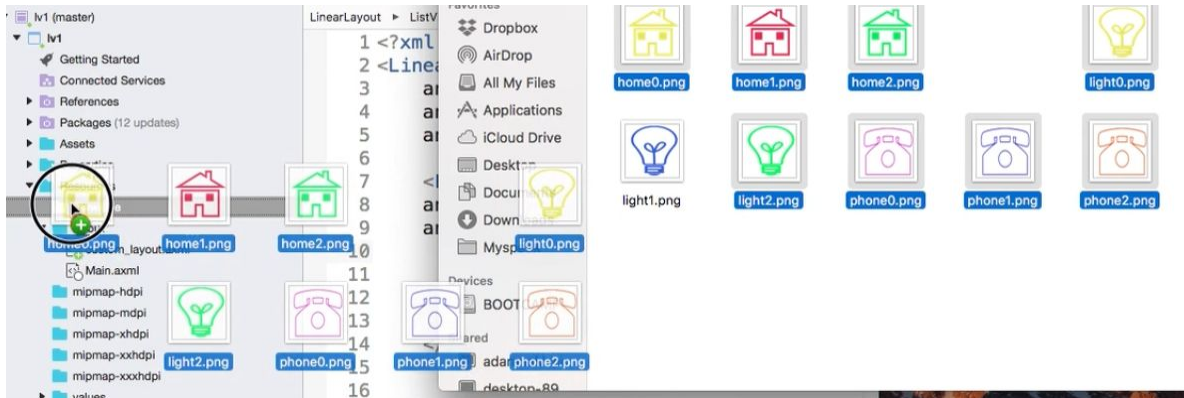
```
    ToyAdapter toyAdapter;
```

נצחיר על רשימה של Toy עם get
 ו set, בהמשך נכניס לה ערכים

נצחיר על Adapter ועל ListView

נוסיף תמונות מהמחשב לאפליקציה:

תוכלו לגרור את התמונות לתיקיה drawable:



ניצור bitmap לכל תמונה:

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

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

Bitmap home0 = BitmapFactory.decodeResource(getResources(), R.drawable.**home0**);

Bitmap home1 = BitmapFactory.decodeResource(getResources(), R.drawable.**home1**);

Bitmap home2 = BitmapFactory.decodeResource(getResources(), R.drawable.**home2**);

Bitmap light0 = BitmapFactory.decodeResource(getResources(), R.drawable.**light0**);

Bitmap light1 = BitmapFactory.decodeResource(getResources(), R.drawable.**light1**);

Bitmap light2 = BitmapFactory.decodeResource(getResources(), R.drawable.**light2**);

Bitmap phone0 = BitmapFactory.decodeResource(getResources(), R.drawable.**phone0**);

Bitmap phone1 = BitmapFactory.decodeResource(getResources(), R.drawable.**phone1**);

Bitmap phone2 = BitmapFactory.decodeResource(getResources(), R.drawable.**phone2**);

ניצור מופעים של Toy שנוסיף בהמשך לרשימה:

```
Toy t1 = new Toy(50,"home","very good toy",home0);  
Toy t2 = new Toy(70,"home","average toy",home1);  
Toy t3 = new Toy(90,"home","nice toy",home2);  
Toy t4 = new Toy(29,"light","very good toy",light0);  
Toy t5 = new Toy(37,"phone","average toy",light1);  
Toy t6 = new Toy(50,"light","nice toy",light2);  
Toy t7 = new Toy(29,"light","very good toy",phone0);  
Toy t8 = new Toy(37,"phone","average toy",phone1);  
Toy t9 = new Toy(50,"light","nice toy",phone2);
```

ניצר רשימה מסוג Toy ונוסיף את המופעים של Toy שיצרנו לרשימה:

```
toyList = new ArrayList<Toy>();  
toyList.add(t1);  
toyList.add(t2);  
toyList.add(t3);  
toyList.add(t4);  
toyList.add(t5);  
toyList.add(t6);  
toyList.add(t7);  
toyList.add(t8);  
toyList.add(t9);
```

ניצור Adapter וניתן לו את הרשימה:

```

toyAdapter=new ToyAdapter(this,0,0,toyList);

lv=(ListView)findViewById(R.id.lv);

lv.setAdapter(toyAdapter);

}

}

```

ניתן ל
Adapter של
ListView

תוכלו להריץ עכשיו את האפליקציה ולראות את ה ListView שבניתם :



סיכום כל השלבים שעשינו לבניית ListView:

1. בנינו custom_layout שיהיה העיצוב לכל תא (custom cell) ב ListView.
2. בנינו class Toy שישמור את הערכים שבהמשך נכניס לכל תא.
3. בנינו ToyAdapter שמקבל שתי תכונות: Context שהוא הפניה לאקטיביטי שבו ניצור את ה ListView ורשימה של Toy בשם objects.
4. ה ToyAdapter יירש מ ArrayAdapter ויממש את הפונקציה:

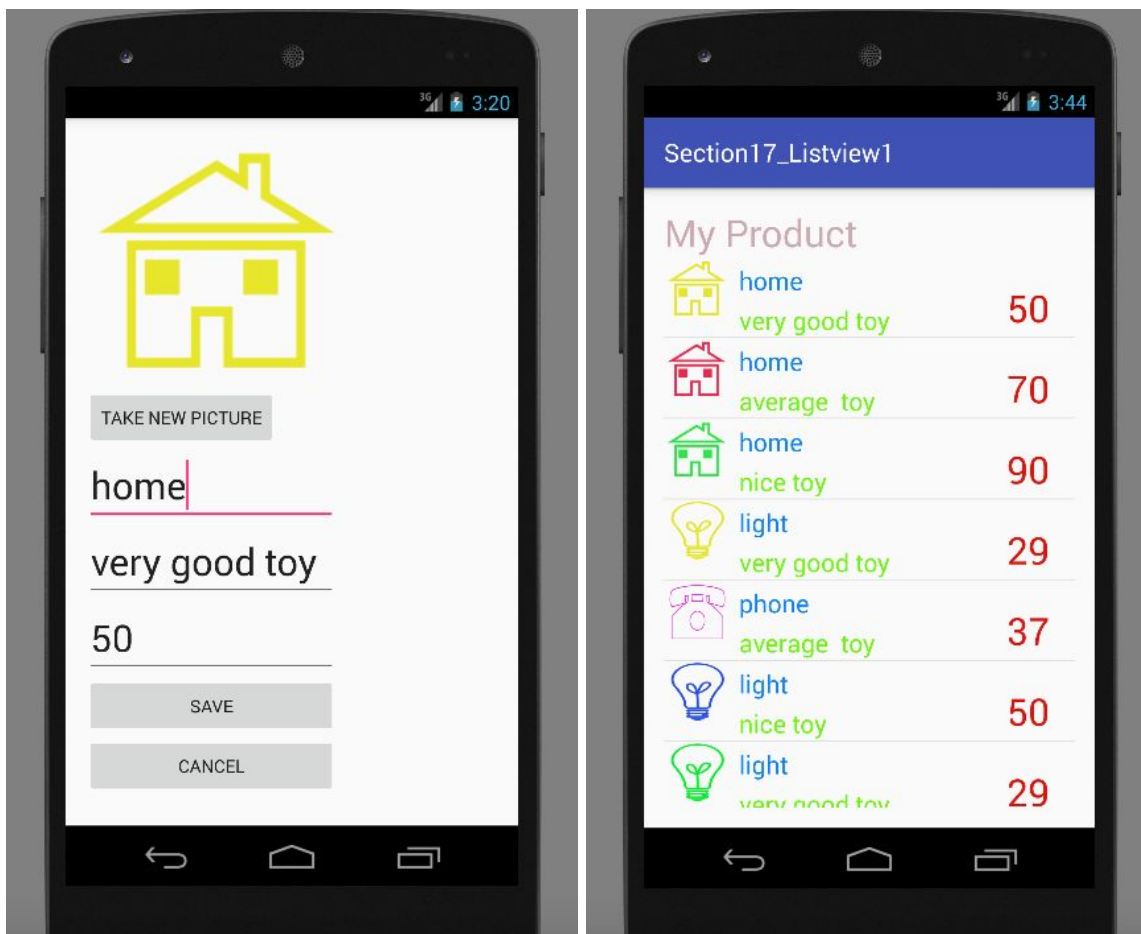
- getView

5. בפונקציה getView :

- המרנו את ה context לאקטיביטי על מנת להשתמש ב Inflater.
 - "ניפחנו" את ה XML שבנינו לכל תא וכך הפכנו את ה XML של כל תא ל View.
 - נתנו הפניות לכל האובייקטים שעל כל תא.
 - נתנו את הערכים המתאימים לכל אובייקט בתא.
6. ב MainActivity (האקטיביטי שבו יצרנו את ה ListView):
- העלנו תמונות
 - יצרנו מופעים של Toy
 - הוספנו את המופעים לרשימה
 - העברנו ל Adapter את הרשימה ו context כהפניה ל MainActivity
 - נתנו ל ListView שיצרנו הפניה ל ListView שב XML
 - הוספנו את ה Adapter ל ListView שיצרנו

עריכת הנתונים ב ListView:

בשביל לערוך את הנתונים באחד התאים שב ListView נעקוב אחרי השלבים הבאים:
 בדוגמה שלנו לחיצה על אחד התאים תיקח אותנו למסך Edit שבו נוכל לערוך את
 הערכים שבתא.



שלב 1 - יצירת מסך EditActivity ריק ועיצובו:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:orientation="vertical">
```

```
tools:context="il.co.appschool.section17_listview1.EditActivity">
```

```
<ImageView
```

```
    android:layout_width="200dp"
```

```
    android:layout_height="200dp"
```

```
    android:id="@+id/ivProduct"
```

```
<Button
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:id="@+id/btnPic"
```

```
    android:text="Take new picture"
```

```
<EditText
```

```
    android:layout_width="200dp"
```

```
    android:layout_height="wrap_content"
```

```
    android:id="@+id/etTitle"
```

```
    android:hint="title"
```

```
    android:textSize="30sp"
```

```
<EditText
```

```
    android:layout_width="200dp"
```

```
    android:layout_height="wrap_content"
```

```
    android:id="@+id/etSubtitle"
```

```
    android:hint="subtitle"
```

```
    android:textSize="30sp"
```

```
<EditText
```

```
    android:layout_width="200dp"
```

```
    android:layout_height="wrap_content"
```

```
    android:id="@+id/etPrice"
```

```
    android:hint="price"
```



```
android:textSize="30sp"
```

```
/>
```

```
<Button
```

```
    android:layout_width="200dp"
```

```
    android:layout_height="wrap_content"
```

```
    android:id="@+id/btnSave"
```

```
    android:text="save"
```

```
/>
```

```
<Button
```

```
    android:layout_width="200dp"
```

```
    android:layout_height="wrap_content"
```

```
    android:id="@+id/btnCancel"
```

```
    android:text="cancel"
```

```
/>
```

```
</LinearLayout>
```

שלב 2 - נבנה פונקציית האזנה ללחיצה על תא ב ListView ב MainActivity:

לשם כך נממש את הפונקציה:

```
lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
        lastSelected = toyAdapter.getItem(position);  
        Intent intent = new Intent(MainActivity.this, EditActivity.class);  
        intent.putExtra("title", lastSelected.getTitle());  
        intent.putExtra("sub", lastSelected.getSubTitle());  
        intent.putExtra("price", String.valueOf(lastSelected.getPrice()));  
        intent.putExtra("bitmap", Helper.bitmapToByteArray(lastSelected.getBitmap()));  
        startActivityForResult(intent, 0);  
    }  
});
```

ה ListView

התא

מיקום התא

נשמור את המופע של Toy שבתא עליו לחץ המשתמש

נייצר Intent למעבר למסך EditActivity

אל תשכחו שה position הראשון שלנו יהיה 0

lv.setOnItemClickListener(this)

ניתן ל ListView האזנה ללחיצה

שלב 3 - עריכת מסך EditActivity

```
public class EditActivity extends AppCompatActivity implements View.OnClickListener {
```

```
    EditText etTitle,etSubtitle,etPrice;
```

```
    Button btnSave,btnCancel;
```

```
    ImageView ivProduct;
```

```
    Bitmap bitmap;
```

```
    Button btnTakePic;
```

נצהיר על האובייקטים
שב EditActivity

@Override

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

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

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

ניתן הפניה לאובייקטים

```
    etTitle=(EditText)findViewById(R.id.etTitle);
```

```
    etPrice=(EditText)findViewById(R.id.etPrice);
```

```
    etSubtitle=(EditText)findViewById(R.id.etSubtitle);
```

```
    btnSave=(Button)findViewById(R.id.btnSave);
```

```
    btnCancel=(Button)findViewById(R.id.btnCancel);
```

```
    btnSave.setOnClickListener(this);
```

```
    btnCancel.setOnClickListener(this);
```

```
    ivProduct=(ImageView)findViewById(R.id.ivProduct);
```

```

btnTakePic=(Button)findViewById(R.id.btnPic);
btnTakePic.setOnClickListener(this);
//connect to intent if its edit mode
Intent intent= getIntent();
if(intent.getExtras()!=null)
{
    String title = intent.getExtras().getString("title");
    String subtitle = intent.getExtras().getString("sub");
    String price = intent.getExtras().getString("price");
    bitmap = Helper.byteArrayToBitmap(intent.getExtras().getByteArray("bitmap"));
    etTitle.setText(title);
    etSubtitle.setText(subtitle);
    etPrice.setText(price);
    ivProduct.setImageBitmap(bitmap);
}
}

```

בדיקה שיש ערכים

כשהמשתמש יגיע למסך EditActivity הערכים שיוצגו יהיו של המופע שהוא רוצה לשנות

שלב 5 - מימוש פונקציה onClick למקרה שהמשתמש לחץ על לקיחת תמונה ב Edit

@Override

```

public void onClick(View v)
{
    if(btnTakePic==v)//option 1 - take new picture
    {
        Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        startActivityForResult(intent,0);
    }
}

```

שלב 6 - הוספת onActivityResult ב EditActivity לחזרה מהמצלמה

@Override

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    super.onActivityResult(requestCode, resultCode, data);  
    if(requestCode==0)  
    {  
        if(resultCode==RESULT_OK)  
        {  
            bitmap = (Bitmap)data.getExtras().get("data");  
            if(bitmap!=null)  
                ivProduct.setImageBitmap(bitmap);  
        }  
    }  
}
```

כשנחזור מהמצלמה נשמור את
התמונה ונשנה את ה ImageView

שלב 7 - הוספת ל onClick האזנה לכפתור השמירה btnSave

```

else if(btnSave==v){//option 2 - save the data and go to first screen
    if(etPrice.getText().toString().length()>0&&
        etTitle.getText().toString().length()>0&&etSubtitle.getText().toString().length()>0
        &&bitmap!=null) {
        Intent intent = new Intent();
        intent.putExtra("price", etPrice.getText().toString());
        intent.putExtra("title", etTitle.getText().toString());
        intent.putExtra("sub", etSubtitle.getText().toString());
        intent.putExtra("bitmap", Helper.bitmapToByteArray(bitmap));
        setResult(RESULT_OK, intent);
        finish();
    }
    else
        Toast.makeText(this, "please fill all fields", Toast.LENGTH_LONG).show();
}
}
else if (btnCancel==v){//option 3 - cancel- and go to first screen
{
    setResult(RESULT_CANCELED,null);
    finish();
}
}

```

נשלף את הערכים שהשתמש
הזין ונשמור אותם ב
Extras של ה intent

נחזור ל MainActivity
עם RESULT_OK

```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(requestCode==0)
    {
        if(resultCode==RESULT_OK)
        {
            bitmap = (Bitmap)data.getExtras().get("data");
            if(bitmap!=null)
                ivProduct.setImageBitmap(bitmap);
        }
    }
}
}

```

שלב 8- נממש את האפשרות שחזרנו מהמסך EditActivity ב- onActivityResult ב- MainActivity:

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
    super.onActivityResult(requestCode, resultCode, data);
    if(requestCode==0)
    {
        if(resultCode==RESULT_OK)
        {
            String title = data.getExtras().getString("title");
            String subtitle = data.getExtras().getString("sub");
            String price = data.getExtras().getString("price");
            Bitmap bitmap=Helper.byteArrayToBitmap(data.getExtras().getByteArray("bitmap"));
            lastSelected.setPrice(Integer.valueOf(price));
            lastSelected.setTitle(title);
            lastSelected.setSubTitle(subtitle);
            lastSelected.setBitmap(bitmap);
            toyAdapter.notifyDataSetChanged();
            Toast.makeText(this,"data saved",Toast.LENGTH_LONG).show();
        }
        else if(resultCode==RESULT_CANCELED)
        {
            Toast.makeText(this,"user cancel",Toast.LENGTH_LONG).show();
        }
    }
}
```

נגיד ל Adapter לשים לב לשינויים שקרו בו

תוכלו להריץ עכשיו את האפליקציה ולשנות את הנתונים ב ListView שלכם (:

הוספת איבר ל ListView:

נוסיף כפתור להוספה ב MainActivity:

```
<Button
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:text="+"
    android:id="@+id/btnAdd"
/>
```

נצהיר על הכפתור:

```
Button btnAdd;
ListView lv;
```

ניתן לכפתור הפניה והאזנה ללחיצה.

נממש את פונקציית ההאזנה:

```
Intent intent = new Intent(this,EditActivity.class);
startActivityForResult(intent,1);
```

נוסיף לפונקציה onActivityResult את האפשרות שחזרנו ממצב הוספת פריט חדש:


```
if(requestCode==1)//come from add mode
{
    if(resultCode==RESULT_OK)
    {
        String title = data.getExtras().getString("title");
        String subtitle = data.getExtras().getString("sub");
        String price = data.getExtras().getString("price");
        Bitmap bitmap = Helper.byteArrayToBitmap(data.getExtras().getByteArray("bitmap"));
        Toy toy=new Toy(Integer.valueOf(price),title,subtitle,bitmap);
        toyAdapter.add(toy);
        toyAdapter.notifyDataSetChanged();
        Toast.makeText(this,"data saved",Toast.LENGTH_LONG).show();
    }
    else if(resultCode==RESULT_CANCELED)
    {
        Toast.makeText(this,"user cancel",Toast.LENGTH_LONG).show();
    }
}
```

עכשיו כשהמשתמש ילחץ על הכפתור, הוא יעבור ל `EditActivity`, שם הוא יוכל להגדיר איבר חדש ולבסוף להוסיף אותו לרשימה. נוכל להריץ ולראות את האיבר שהוספנו:

Redmi 5_Plus

9:43 99% 9:43

lv1

+		
	home	
	very good toy	50
	home	70
	average toy	
	home	90
	nice toy	
	light	
	very good toy	29
	phone	
	average toy	37
	light	
	nice toy	50
	phone	
	average toy	37
	light	
	nice toy	50
	phone	
	average toy	60

מחיקת איבר מ ListView:

לחיצה ארוכה על איבר ב ListView תמחק אותו.

ניתן ל ListView האזנה ללחיצה ארוכה:

נממש את הפונקציה:

```
lv.setOnItemLongClickListener(new AdapterView.OnItemClickListener() {
```

```
@Override
```

```
public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long
```

חובה לרשום כדי שה adapter יראה את השינוי שעשינו ברשימה

```
lastSelected=toyAdapter.getItem(position);
```

```
toyAdapter.remove(lastSelected);
```

```
toyAdapter.notifyDataSetChanged();
```

```
return false;
```

```
}
```

```
});
```