Android Introduction

appSchool.co.il

Android introduction

today:

1. example project
2. mainactivity
3. xml
4. manifast xml

example project

xml

LinearLayout is a view group (a subclass of ViewGroup) that lays out child views in either a vertical or horizontal orientation, as specified by the Android orientation attribute. Each child of a LinearLayout appears on the screen in the order in which it appears in the XML.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.
com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:orientation="vertical"
   android:background="@drawable/c"
    >
   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="you have 0 points"
      android:textSize="30dp"
      android:layout_gravity="center"
      android:gravity="center"
      android:id="@+id/txtdisplay"
                  />
```

```
   <Button
       android:layout_width="250dp"
      android:layout_height="wrap_content"
      android:text="add one point"
      android:textSize="15dp"
      android:layout_gravity="center"
      android:id="@+id/btnplus"
        />
      <Button
       android:layout_width="250dp"
      android:layout_height="wrap_content"
      android:text="minus one point"
      android:textSize="15dp"
      android:layout_gravity="center"
      android:id="@+id/btnminus"
            />
</LinearLayout>
```

LinearLayout

- Linear layout- vertical or horizontal
- match_parent = fill out to the size of the parent
- match_parent = just big enough to hold our own contents
- @string/hello_world = refer to data centralized in res/values/strings.xml.

Activity class

```
public class Androidlesson3Activity extends Activity {
    /** Called when the activity is first created. */
        int point;
        Button add;
        Button minus;
        TextView display;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        point=0;
        add=(Button)findViewById(R.id.btnplus);
        minus=(Button)findViewById(R.id.btnminus);

        display=(TextView)findViewById(R.id.txtdisplay);
        add.setOnClickListener(this);
        minus.setOnClickListener(this);

        @Override
        public void onClick(View v) {
                // TODO Auto-generated method stub
                if(v.getId()==R.id.btnplus)
                {
                point++;
                }
                else if( v.getId()==R.id.btnminus)
                {
                        point--;
                }
                display.setText("your totol point is " + point);
}
}
```

Activity class

```
public class Androidlesson3Activity extends Activity {
    /** Called when the activity is first created. */
        int point;
        Button add;
        Button minus;
        TextView display;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        point=0;
        add=(Button)findViewById(R.id.btnplus);
        minus=(Button)findViewById(R.id.btnminus);

        display=(TextView)findViewById(R.id.txtdisplay);
```

```
        add.setOnClickListener(new View.OnClickListener() {
          @Override
          public void onClick(View v) {
          // TODO Auto-generated method stub
          point++;
          display.setText("your totol point is " + point);
                                        }
                    });
        minus.setOnClickListener(new View.OnClickListener()
{
       @Override
      public void onClick(View v) {
          // TODO Auto-generated method stub
          point--;
          display.setText("your totol point is " + point);
          }
});
 }
}
```

android manifast file

```
<manifest xmlns:android="http://schemas.android.
com/apk/res/android"
    package="com.example.lessona_an2_example2"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="15" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/title_activity_main" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.
LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

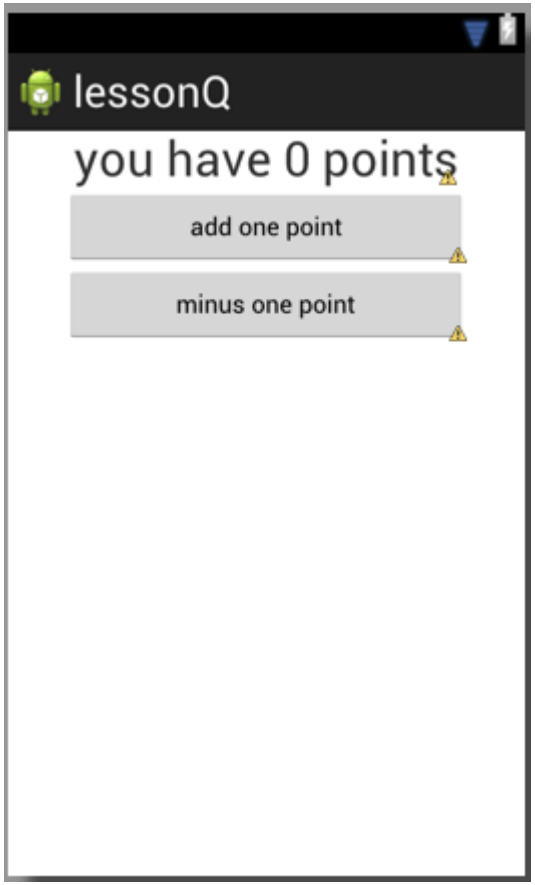
1. "@string/app_name" : name of the application.

2.The Activity "android:name :class name that implements the activity.

3.The “.” means that the class in the the current package

4. The launcher provides the app icon to launch this activity first.

example 1



Example 2

xml

LinearLayout is a view group (a subclass of ViewGroup) that lays out child views in either a vertical or horizontal orientation, as specified by the Android orientation attribute. Each child of a LinearLayout appears on the screen in the order in which it appears in the XML.

```
<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" >


  <EditText
      android:id="@+id/editText1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginTop="14dp"
      >
   </EditText>
  <Button
      android:id="@+id/button1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentLeft="true"
     android:text="Button" />
```

```
  <TextView
      android:id="@+id/textView1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentLeft="true"
      android:layout_below="@+id/button1"
      android:layout_marginTop="54dp"
      android:text="TextView" />


</LinearLayout>
```

Activity class

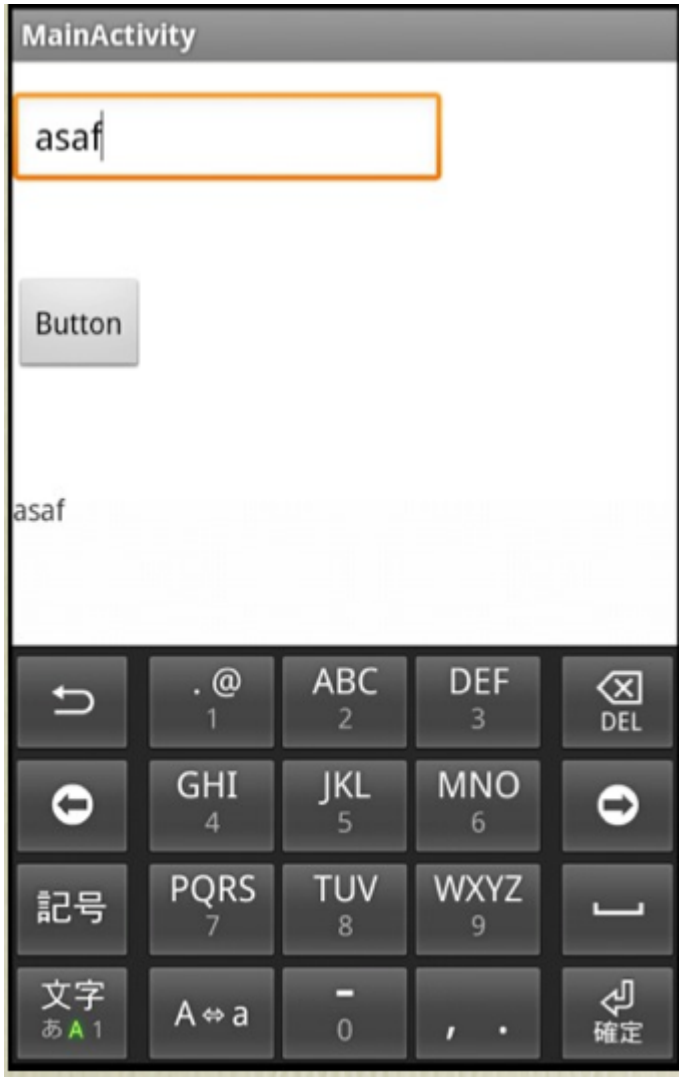
example code

```
public class MainActivity extends Activity {

    Button btn;
     TextView tv;
     EditText et;

@Override
   public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        et=(EditText) findViewById(R.id.editText1);
        btn=(Button) findViewById(R.id.button1);
        tv=(TextView) findViewById(R.id.textView1);

       btn.setOnClickListener(new View.OnClickListener() {

@Override

              public void onClick(View v) {
                    // TODO Auto-generated method stub
                    String info=et.getText().toString();
                    tv.setText(info);

} });
```

same example code

```
      public class MainActivity extends Activity {

    Button btn;
     TextView tv;
     EditText et;

@Override
   public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        et=(EditText) findViewById(R.id.editText1);
        btn=(Button) findViewById(R.id.button1);
        tv=(TextView) findViewById(R.id.textView1);

       btn.setOnClickListener(this) ;
@Override
                public void onClick(View v) {
                  // TODO Auto-generated method stub
                  String info=et.getText().toString();
                  tv.setText(info);

}
};
```

דוגמא נוספת



Home Work

1. כתב אפליקציה שמציגה 2 תיבות text ו-textviewמתחתיהם.לאחר שהמשתמש יזין את שמו ואת שם משפחתו לתיבות ה-text אזי שם ושם משפחתו יופיע על תיבת ה-text מתחת לכפתורים.
2. כתב אפליקציה שמכילה שני תיבות text ו-textview בינהם שכתוב עליו הסימן + ו-textview נוסף להצגת התוצאה.וכפתור submit.המשתמש יקיש שני מספרים וילחץ על כפתור submit ותופיע תוצאת המחשבון.
3. צור מסך המכיל על כפתורים את המספרים 1,2,3,4,5,6,7,8,9,0 ואת הפעולות כפל, חיבור,חיסור וחילוק והסימן שווה.המשתמש יקיש על מספר,פעולה ועוד מספר ולאחר מיכן על תוצאת שווה ויוצג לו תוצאת הפעולה על המספרים.
4. הוסף לתרגילים 2,1 ו-3 background לבחירתך.

יום ללא תכנות הוא יום מבוזבז!