



# Android השתלמות מתחילים Activity

אלון חימוביץ

# מסך קוד ה Java

```
activity_main.xml x MainActivity.java x
1 package com.example.myapplication;
2
3 import ...
4
5
6
7 public class MainActivity extends AppCompatActivity {
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13    }
14 }
```

המתודה שפועלת בעת טעינת המסך

מציג את ה layout על המסך

# מסך קוד ה Java - גרסה חדשה

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    // 1. Enables drawing edge-to-edge (under system bars)
    EdgeToEdge.enable(this);

    // 2. Sets the layout
    setContentView(R.layout.activity_settings);

    // 3. Adds window insets listener to handle system bars (status/nav)
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
        Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
        return insets;
    });
}
```

עיקר השינוי -  
מיקום הרכיבים משתנה בהתאם  
להופעת / ירידת החלון העליון  
status / nav bar

# חיבור ל Views

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="10"
    android:gravity="top"
    android:id="@+id/etMainText"
    android:hint="@string/write_some_text"

/>
```

לפנייה לרכיב ספציפי ב layout נדרש מזהה ייחודי - id

בקוד ה Java מגדירים הפנייה  
לאחר מכן מפנים אותה לאובייקט

```
// EditText לאובייקט מטיפוס
// NULL שימו לב בשלב זה
private EditText etMain;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    // קשור ההפנייה לאובייקט קיים מה XML
    etMain = findViewById(R.id.etMainText);
}
```

# חיבור לאירוע - Event

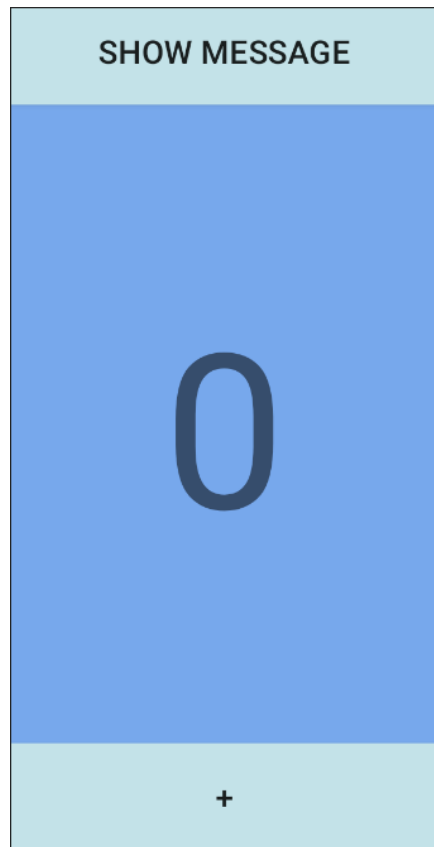
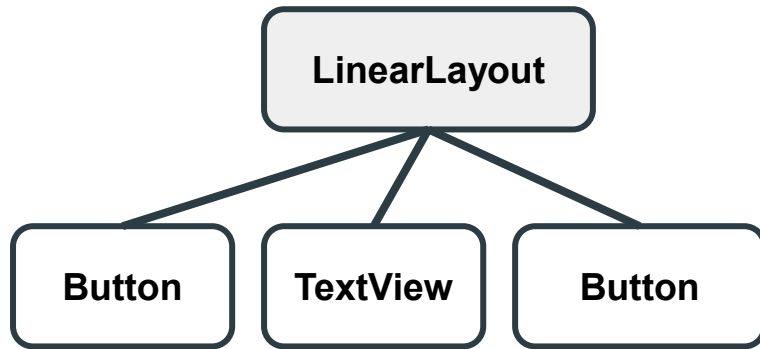
```
<Button  
    android:layout_width="match_parent"  
    android:layout_height="0dp"  
    android:layout_weight="3"  
    android:text="@string/click_button"  
    android:background="#4ED3E4"  
    android:onClick="ShowMessage"  
>
```

פעולה שתזומן בעת אירוע לחיצה

הפעולה בקוד ה Java

הקפצת הודעה על המסך

```
// לפעולה הזו מגיעים לאחר לחיצה על הכפתור  
public void ShowMessage(View view)  
{  
    // קוראים את תוכן הטקסט שהוקלד בתיבת הטקסט  
    String message = etMain.getText().toString();  
    // הצגה של ההודעה על המסך  
    Toast.makeText(context: this, text: "user entered " + message, Toast.LENGTH_LONG).show();  
}
```



# תרגיל Click Counter

- צרו LinearLayout שמכיל את ה Views הבאים:
- Button - כפתור עם טקסט "show"
- TextView - שמופיע בו המספר 0
- Button - כפתור עם טקסט "+"

# תרגיל Click Counter - אינטראקציה עם המשתמש

- ▶ ה `TextView` מכיל את מספר הלחיצות שנעשו על כפתור ה "+"
- ▶ `Click` על כפתור ה "+" מעלה את מספר הלחיצות ב 1
- ▶ לחיצה על כפתור ה `Show` - תציג `Toast` עם מספר הלחיצות עד כה:

You have clicked \_\_ times!

לפני כתיבה - חשוב להפריד בין ה `VIEW` לבין ייצוג הנתונים הנדרשים. (`Model`)  
הגדירו את המחלקה `Model`, אילו תכונות ופעולות נדרשות?

# תרגיל Click Counter פתרון באמצעות Codelab

[codelabs.developers.google.com/codelabs/android-training-layout-editor-part-a/index.html?index=..%2F..%2Fandroid-training#0](https://codelabs.developers.google.com/codelabs/android-training-layout-editor-part-a/index.html?index=..%2F..%2Fandroid-training#0) ►

<https://developer.android.com/codelabs/constraint-l>