

פרק 7 - Dialog

מה זה Dialog?

Dialog הוא מעין פופ - אפ שקופץ כאשר אנו לוחצים על משהו במסך.
הוא יכול להיות עיגול מסתובב שמראה על טעינה של משהו, או הודעה של "האם אתה בטוח שאתה רוצה להירשם?"
זה יכול להיות חלון של הרשמה או פרסומת ועוד.

מתי משתמשים ב dialog?

כמעט בכל אפליקציה שקיימת כיום יש לפחות Dialog אחד.
משתמשים ב dialog כשרוצים שיופיע משהו על המסך אבל לא לצאת מהמסך עצמו.

ישנם סוגים רבים של Dialog.

בפרק זה נלמד על:

Alert dialog

Custom dialog

Progress dialog

Date dialog

Time dialog

ה dialog הראשון שנלמד הוא custom dialog.

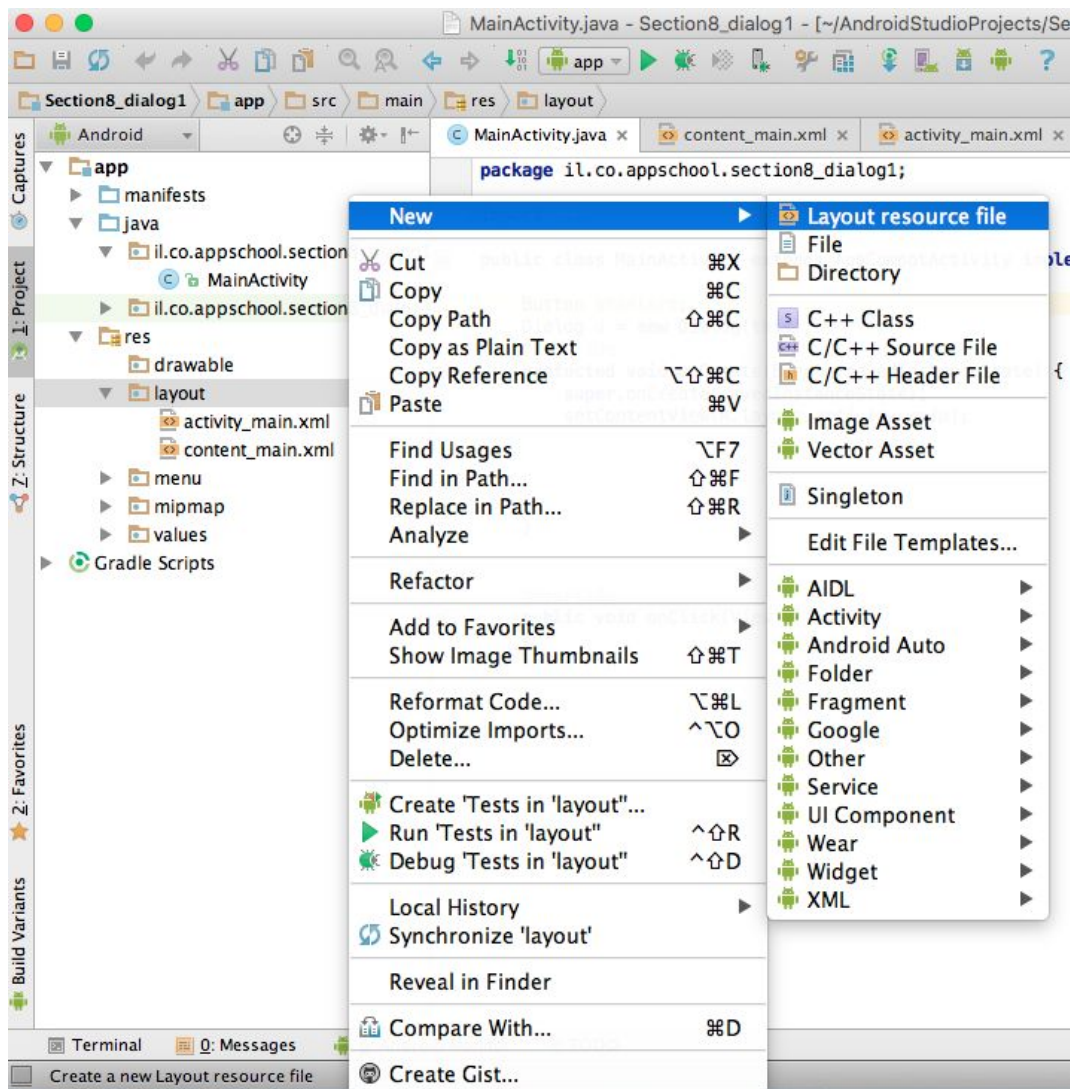
סדר פעולות יצירת custom dialog:

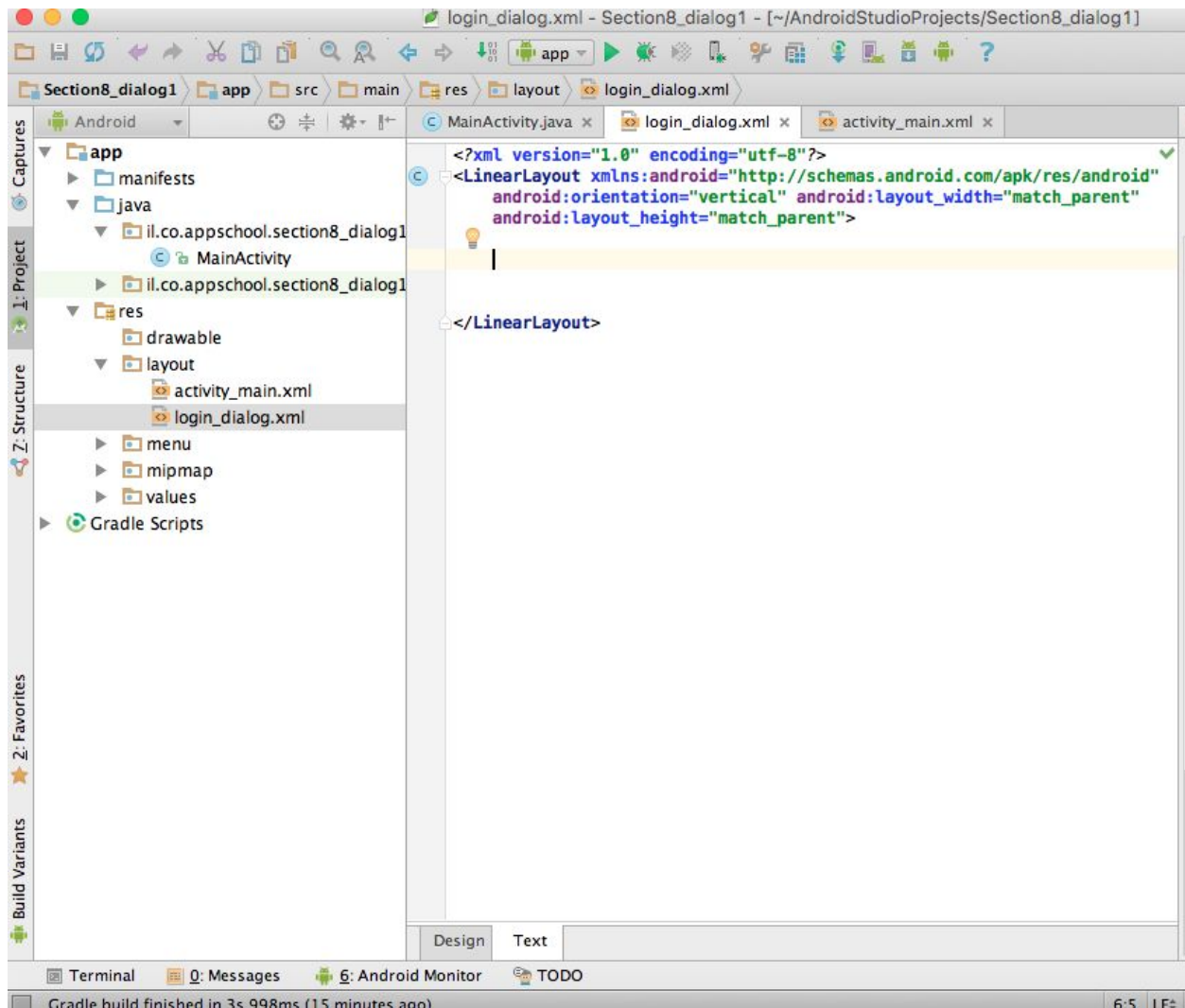
1. ניצור custom_layout ונעצב אותו
 2. נכריז על Dialog
 3. נעביר לו כפרמטר את ה- custom_layout שיצרנו.
 4. נקבל reference לאובייקטים על ה- dialog.
 5. נציג את ה- dialog באמצעות פונקציה d.show()
- לדוגמה לשימוש ב custom layout ניצור אפליקציה שדרך המסך כניסה נוכל ללחוץ על כפתור שיצור דיאלוג של חלון הרשמה.

שלב 1 - יצירת custom_layout ועיצובו

custom_layout הוא קובץ XML שמכיל את עיצוב ה dialog.

בשביל ליצור custom_layout נעקוב אחרי השלבים הבאים:





ה XML של ה custom_layout יראה כך:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
android:orientation="vertical"
```

```
android:layout_width="match_parent"
```

Copyright © 2019 appSchool. Powered by appSchool.co.il

```
android:layout_height="match_parent"
```

```
android:background="@drawable/a1">
```

```
<EditText
```

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

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

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

```
    android:hint="username"
```

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

```
<EditText
```

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

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

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

```
    android:hint="password"
```

```
    android:inputType="textPassword"
```

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

```
<Button
```

```

android:layout_width="200dp"

android:layout_height="wrap_content"

android:id="@+id/btnDialogLogin"

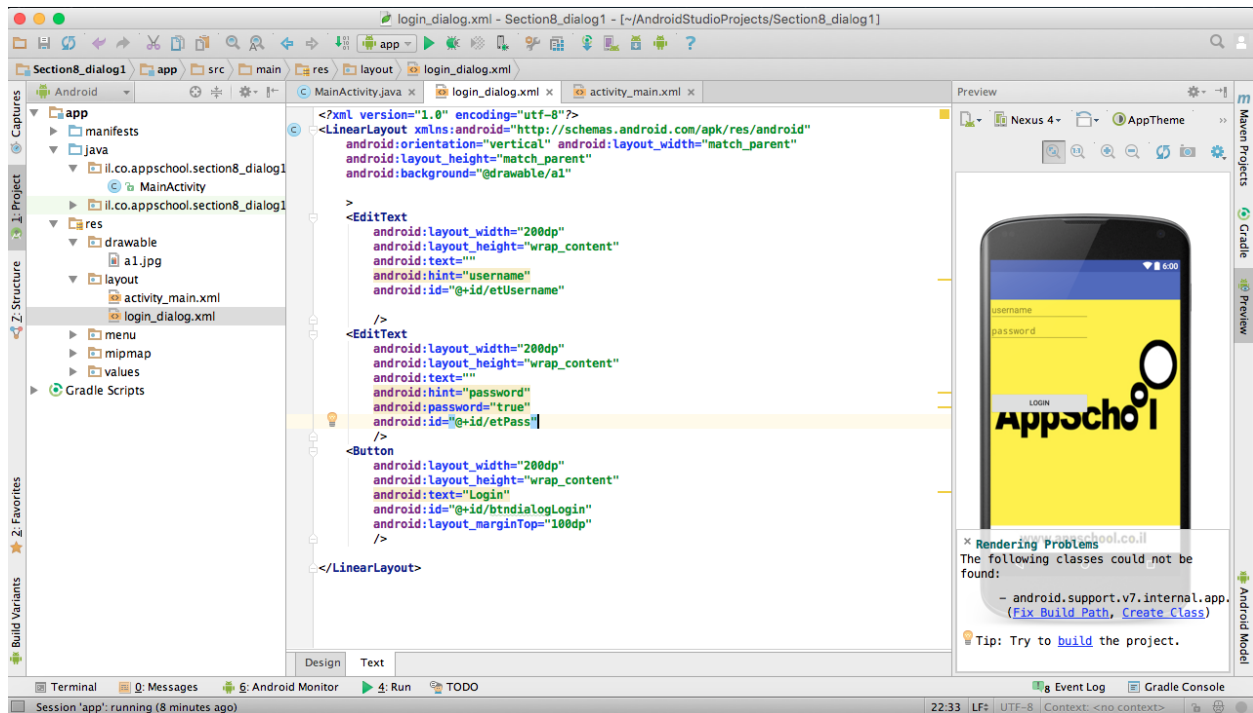
android:text="Login"

android:layout_marginTop="100dp"

/>

```

חלון הדיאלוג יראה כך:



נעצב גם את המסך כניסה (ה MainActivity):

```
<?xml version="1.0" encoding="utf-8"?>
```

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/activity_main"

android:layout_width="match_parent"

android:layout_height="match_parent"

android:orientation="vertical"

>

<TextView

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="Login!"

android:textSize="30sp"

android:layout_gravity="center"

/>

<Button

android:layout_width="200dp"

android:layout_height="wrap_content"

android:id="@+id/btnLogin"

```
android:text="Login"
```

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

```
/>
```

```
</LinearLayout>
```


שלב 2 - הכרזה על dialog

נעבור לקובץ ה JAVA של ה MainActivity.

נכריז על dialog.

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

```
    SharedPreferences sp;
```

```
    Dialog d;
```

הצהרות

```
    EditText etUserName, etPass;
```

```
    Button btnCustomLogin, btnLogin;
```

```
@Override
```

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

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

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

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

הפניה לכפתור ומתן האזנה

```
    btnLogin.setOnClickListener(this);
```

```
    sp=getSharedPreferences("details1",0);
```

יצירת תיקיה ב SharedPreferences

```
}
```

```
public void createLoginDialog()
```

```
{
```

```
    d= new Dialog(this);
```

ניצור דיאלוג חדש

פונקציה ליצירת הדיאלוג

```
    d.setContentView(R.layout.custom_layout);
```

ניתן לדיאלוג את ה layout שיצרנו

```
    d.setTitle("Login");
```

```
    d.setCancelable(true);
```

לחיצה מחוץ לדיאלוג
כשהוא פתוח סוגרת אותו

```
    etUserName=(EditText)d.findViewById(R.id.etUserName);
```

```
    etPass=(EditText)d.findViewById(R.id.etPassword);
```

ניתן הפניות לאובייקטים שעל
הדיאלוג.
שימו לב שבגלל שהם על הדיאלוג
נרשום d. (שם הדיאלוג)

```
    btnCustomLogin=(Button)d.findViewById(R.id.btnDialogLogin);
```

```
    btnCustomLogin.setOnClickListener(this);
```

```
    d.show();
```

נגיד לדיאלוג להופיע
אחרי שיצרנו אותו

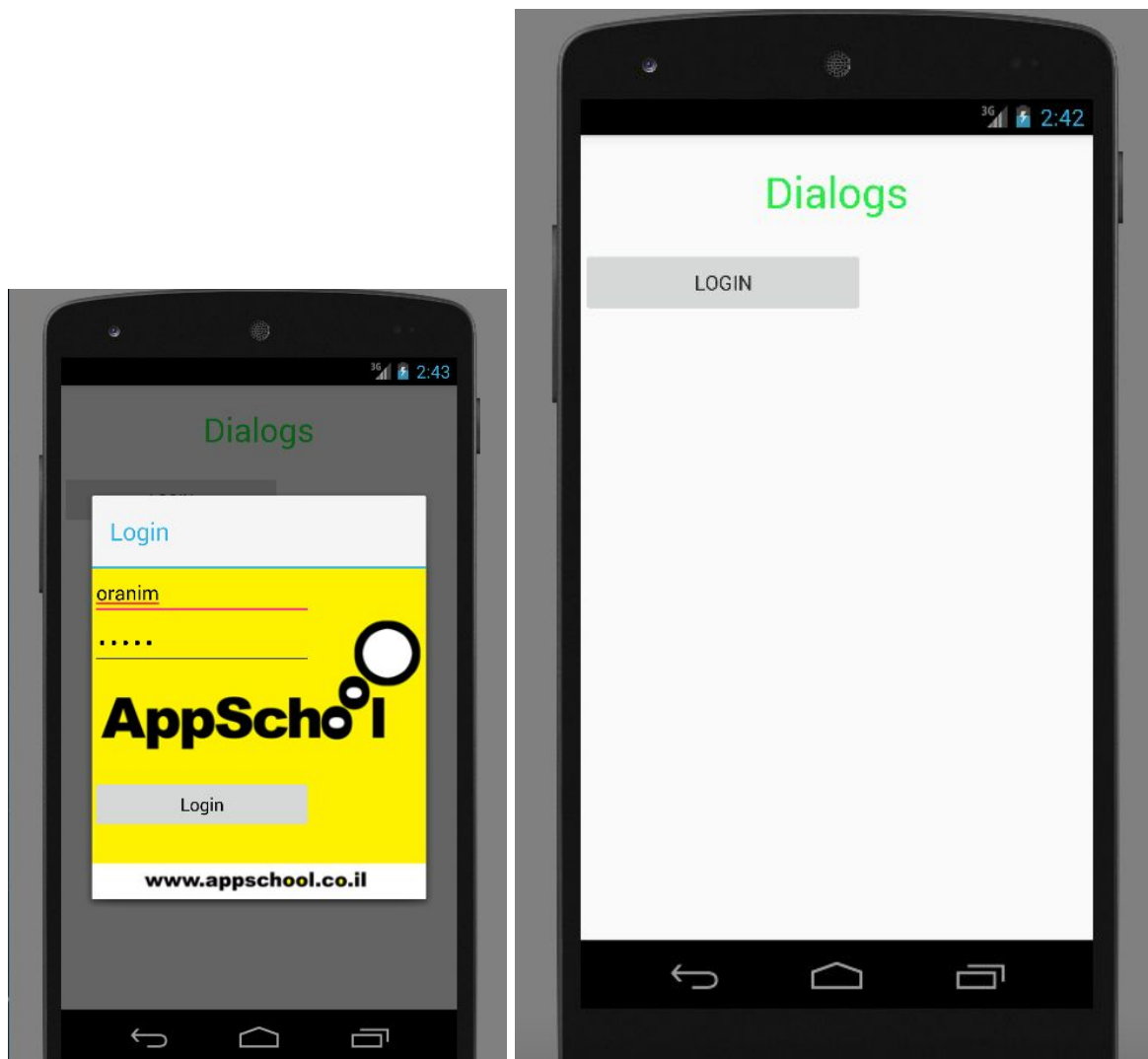
```
}
```

שימו לב שמתכנתים רבים שוכחים לכתוב d כשהם יוצרים אובייקט בתוך דיאלוג, ולכן האפליקציה קורסת להם.

```
@Override  
  
public void onClick(View v)  
{  
    if(v==btnLogin)  
    {  
        createLoginDialog();  
    }  
    else if(v==btnCustomLogin)  
    {  
        Toast.makeText(this,"username password saved",Toast.LENGTH_LONG).show();  
        d.dismiss();  
    }  
}  
}
```

← בלחיצה על btnCustomLogin
(הכפתור שבתוך הדיאלוג) יסגר הדיאלוג

הריצו את האפליקציה!



עד כה יצרנו אפליקציה שבלחיצה על כפתור פותחת דיאלוג, ובלחיצה על הכפתור שעל הדיאלוג סוגרת אותו.

החלק השני לאפליקציה שבנינו, יהיה להוסיף לה SharedPreferences, כך שבכל פעם שהמשתמש יכנס לדיאלוג ההתחברות, יופיעו לו הנתונים האחרונים ששמר.

ניצור קובץ שבו נשמור את הפרטים בשיטה של key ו value: (בפונקציית ה onCreate)

```
sp=getSharedPreferences("details1",0);
```

וכך בכל פעם שהמשתמש יגיע לדיאלוג ההרשמה יופיעו לו ב EditTextים הנתונים האחרונים שהזין:

```
String strname=sp.getString("username",null);
String strpass=sp.getString("pass",null );
if(strname!=null&&strpass!=null) {
    etUserName.setText(strname);
    etPass.setText(strpass);
}
```

נוסיף לפונקציית יצירת הדיאלוג קוד ששולף את הערכים שבשדות עם ה key pass ו username

אחרי שניתן הפניה ל EditTexts שבדיאלוג נכניס את הערכים ששלפנו מה SharedPreferences במקרה שהם לא null

```
@Override
public void onClick(View v)
{
    if(v==btnLogin)
    {
        createLoginDialog();
    }
    else if(v==btnCustomLogin)
    {
        SharedPreferences.Editor editor=sp.edit();
        editor.putString("username",etUserName.getText().toString());
        editor.putString("pass",etPass.getText().toString());
        editor.commit();

        Toast.makeText(this,"username password saved",Toast.LENGTH_LONG).show();
        d.dismiss();
    }
}
}
```

נוסיף לפונקציית ה onClick שבלחיצה על הכפתור לסגירת הדיאלוג נשמור ב ShredPreferences את הנתונים שהמשתמש הזין

- Alert Dialog

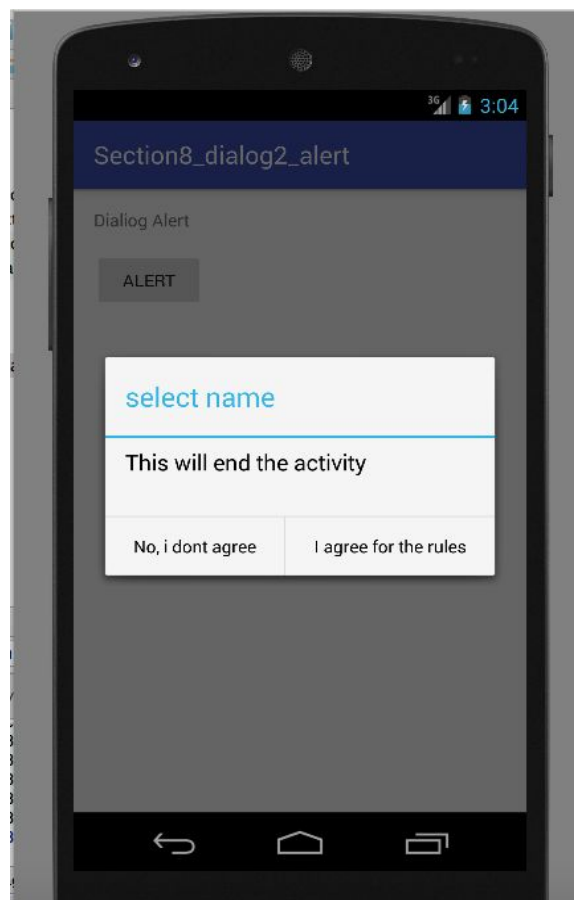
מה זה בעצם Alert Dialog?

Alert Dialog הוא דיאלוג הודעה שקופצת על המסך ומעדכנת את המשתמש.

דוגמה לשימוש בדיאלוג:

הודעה שקופצת על המסך כאשר המשתמש מעוניין לשמור נתונים, והיא שואלת: 'האם אתה בטוח שברצונך להמשיך?'.
המשתמש יכול לבחור באחת משתי האפשרויות: 'כן, אני רוצה להמשיך' או 'לא'.

נלמד איך לבנות Alert Dialog, ובדוגמה שלנו הוא יראה כך:



נתחיל:

נבנה לנו במסך הראשי (ה MainActivity) כפתור שלחיצה עליו תפתח לנו את הדיאלוג,

ו TextView קצר שרשום עליו "Hello World"

נבנה את קובץ ה XML:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />
    <Button
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:id="@+id/btnAlert"
        android:text="Alert"
        android:textSize="30sp" />
</LinearLayout>
```

נעבור לקובץ ה JAVA, ובו נצהיר על הכפתור שלחיצה עליו תראה לנו את הדיאלוג.

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    Button btnAlert;
```


ניתן הפניה לכפתור ואפשרות האזנה ללחיצה על כפתור:

@Override

```
protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);

    btnAlert=(Button)findViewById(R.id.btnAlert);

    btnAlert.setOnClickListener(this);

}
```

נממש את הפונקציה onClick:

@Override

```
public void onClick(View v) {
```

```
    if(v==btnAlert)
```

```
    {
```

```
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
```

```
        builder.setTitle("Select name");
```

```
        builder.setMessage("This is a message");
```

```
        builder.setCancelable(true);
```

```
        builder.setPositiveButton("I agree for the rules", new HandleAlertDialogListener());
```

```
        builder.setNegativeButton("No,i dont agree", new HandleAlertDialogListener());
```

```
        AlertDialog dialog=builder.create();
```

```
        dialog.show();
```

```
    }
```

```
}
```

נבנה את ה AlertDialog

על כל אחת מהאפשרויות בדיאלוג יהיה רשום טקסט בהתאם, ובחירה באחת מהאפשרויות תפעיל פונקציית onClick במחלקה שנממש בהמשך בשם HandleAlertDialogListener

ניצור את הדיאלוג עם ה builder שבנינו

הדיאלוג יופיע

לבסוף נממש את הפונקציה :HandleAlertDialogListener

```
public class HandleAlertDialogListener implements DialogInterface.OnClickListener
{
    @Override
    public void onClick(DialogInterface dialog, int which) {
        Toast.makeText(MainActivity.this, "u selected " + which, Toast.LENGTH_LONG).show();
    }
}
```

בבחירת אחת מהאפשרויות
יופיע Toast שמסביר באיזו
אפשרות בחר המשתמש

שימו לב שלא היינו חייבים ליצור
מחלקה ובמקום זה יכולנו לממש
את הפונקציה במחלקה שכבר יש
לנו (MainActivity) באמצעות this

:Progress Dialog

דיאלוג נוסף שנלמד בפרק זה הוא Progress Dialog.

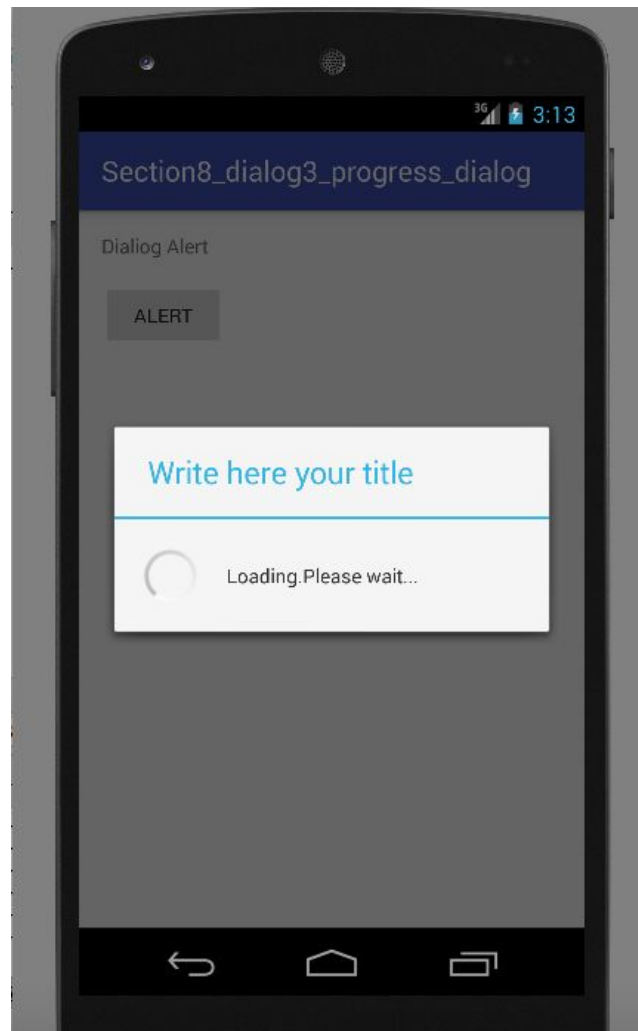
מה הוא Progress Dialog?

Progress Dialog הוא דיאלוג שמראה את ההתקדמות שלנו בביצוע פעולה מסוימת.

מתי משתמשים בו?

ניתן לראות בהרבה אתרים שכאשר מחכים למשהו שיטען, מופיע לנו עיגול מסתובב.
תוכלו לראות דיאלוג כזה ביוטיוב כשמחכים שהסרטון יטען, כשמחכים שסרט יטען ואפילו
בדפדפן שלכם לפעמים.

כך הוא נראה:



בשביל לראות איך להשתמש ב Progress Dialog נערוך את קובץ ה XML של ה MainActivity בדומה לדוגמאות הקודמות של הדיאלוגים:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
```

Copyright © 2019 appSchool. Powered by appSchool.co.il

```
android:id="@+id/activity_main"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!" />
<Button
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:id="@+id/btnProgress"
    android:text="progress"
    android:textSize="30sp" />
</LinearLayout>
```

נעבור לקובץ ה JAVA של ה MainActivity:

נצהיר על הדיאלוג ועל הכפתור שלחיצה עליו מראה את הדיאלוג:

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

```
Button btnProgress;
```

```
ProgressDialog p;
```

הצהרות

ניתן הפניה לכפתור וניתן לו אפשרות האזנה:

@Override

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

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

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

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

```
btnProgress.setOnClickListener(this);
```

```
}
```

הפניה ומתן האזנה

לבסוף, נערוך את הפונקציה onClick:

@Override

```
public void onClick(View v) {
```

```
p=ProgressDialog.show(this,"Write your title","Loading please wait...",true);
```

```
p.setCancelable(true);
```

```
p.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
```

```
p.setMessage("Loading...");
```

```
p.show();
```

```
}
```

```
}
```

ניצור את ה ProgressDialog

נוכל לבחור את סוג ה
ProgressDialog

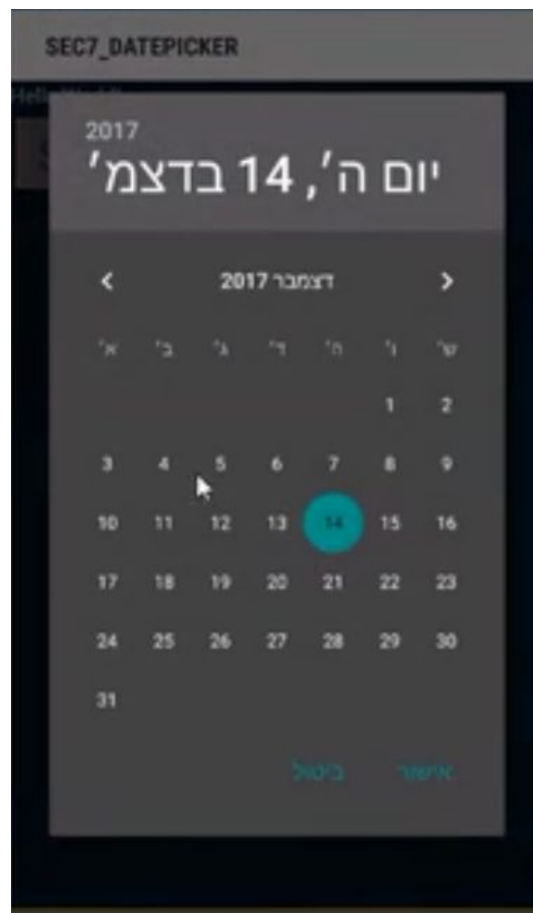
Date Dialog (Date Picker):

הדיאלוג הרביעי שנלמד בפרק זה נקרא Date Dialog.

Date Dialog נותן לנו אפשרות לבחור תאריך.

ניתן לראות שימוש ב Date Dialog ביומן.

כך יראה Date Dialog:



לדוגמה נבנה את ה XML מהדוגמאות הקודמות של הדיאלוגים (Button ו TextView).

```
?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />
    <Button
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:id="@+id/btnSelectDate"
        android:text="select date"
        android:textSize="30sp" />
</LinearLayout>
```

בקובץ ה JAVA של ה MainActivity נצהיר קודם כל על הכפתור שמראה לנו את הדיאלוג:

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    Button btnSelectDate;
```

ניתן לו הפניה ואפשרות להאזנה:

@Override

```
protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);

    btnSelectDate=(Button)findViewById(R.id.btnDate);

    btnSelectDate.setOnClickListener(this);

}
```

נממש את הפונקציה `OnClick`: (אל תשכחו לממש את הממשק שנותן לנו את הפונקציה בעזרת הנורה האדומה)

@Override

```
public void onClick(View v) {

    if(v==btnSelectDate)

    {

        Calendar systemCalender = Calendar.getInstance();

        int year = systemCalender.get(Calendar.YEAR);

        int month = systemCalender.get(Calendar.MONTH);

        int day = systemCalender.get(Calendar.DAY_OF_MONTH);

        DatePickerDialog datePickerDialog = new DatePickerDialog(this,new SetDate(),year,month,day);

        datePickerDialog.show();

    }

}
```

ניצור לוח שנה שיצביע על התאריך הנוכחי

נשלוף את היום, החודש והשנה מלוח השנה

בהמשך ניצור מחלקה בשם `SetDate` ונממש בה את הממשק והפונקציה להאזנה ל `DatePickerDialog`

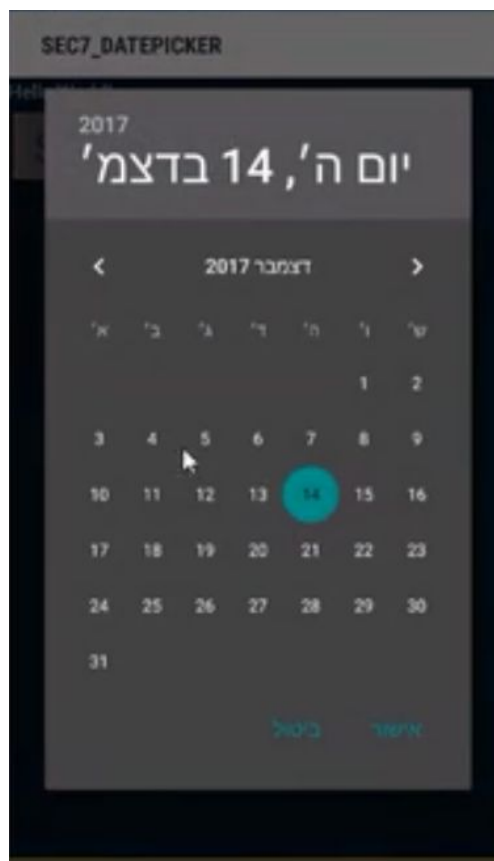
שימו לב! במקרה שמופיע לכם משתנה או אובייקט מהמערכת עם סימון אדום, עימדו עליו ותיפתח בצד שמאל נורה אדומה. תלחצו עליה ותממשו את מה שמוצע לכם. (יכול לקרות עם ה `DateTime`)

נממש את הפונקציה `onDateSet`:

```
public class SetDate implements DatePickerDialog.OnDateSetListener
{
    @Override
    public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {
        monthOfYear = monthOfYear + 1;
        String str = "You selected : " + dayOfMonth + "/" + monthOfYear + "/" + year;
        Toast.makeText(MainActivity.this, str, Toast.LENGTH_LONG).show();
        btnSelectDate.setText(str);
    }
}
```

תת מחלקה בקובץ
 MainActivity שמממשת
 את `OnDateSetListener`

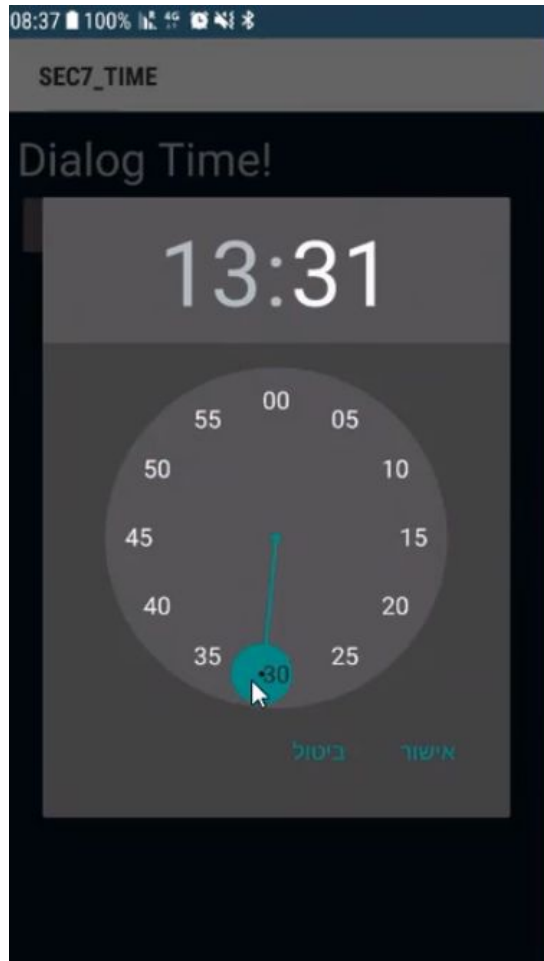
החודש שיישלף מהמערכת יהיה
 קטן ב 1, לכן תצטרכו להגדיל ב 1



Time Dialog (Time Picker):

הדיאלוג האחרון שנלמד להשתמש בו בפרק זה הוא Time Dialog.

Time Dialog הוא דיאלוג שנותן לנו אפשרות בחירת שעה.
ניתן לראות שימוש ב Time Dialog כשמכוונים שעון מעורר בטלפון.



נבנה את ה XML מהדוגמאות הקודמות:

```
<?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="match_parent"
```

```
    android:orientation="vertical"
```

```
    android:padding="10dp">
```

```
<TextView
```

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

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

```
    android:text="Dialog Time!"
```

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

```
<Button
```

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

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

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

```
    android:text="select time"
```

בקובץ ה JAVA נצהיר על הכפתור:

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
    Button btnSelectTime;
```

ניתן לו הפניה ואפשרות האזנה:

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    btnSelectTime=(Button)findViewById(R.id.btnTime);  
    btnSelectTime.setOnClickListener(this);  
}
```

נממש את הפונקציה onClick:

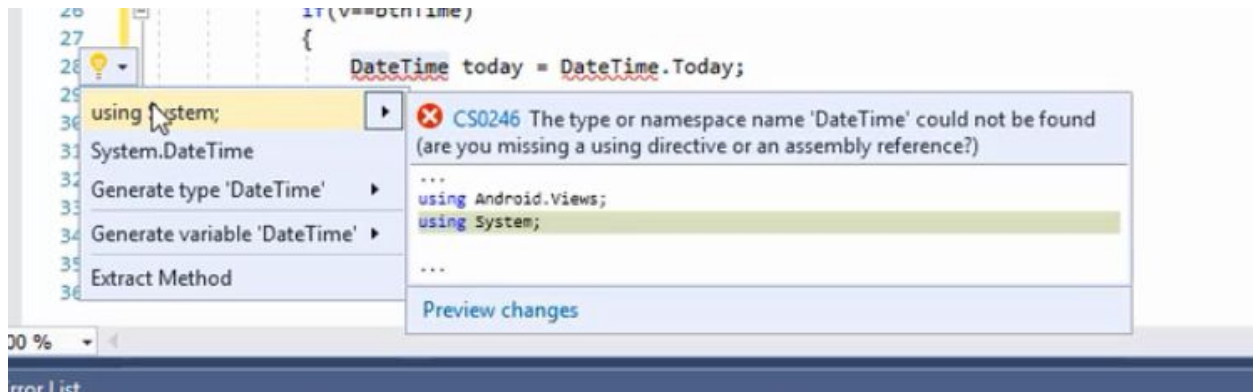
@Override

```
public void onClick(View v) {  
    if(v==btnSelectTime)  
    {  
        Calendar systemCalendar = Calendar.getInstance();  
        int hour = systemCalendar.get(Calendar.HOUR_OF_DAY);  
        int minute = systemCalendar.get(Calendar.MINUTE);  
        TimePickerDialog timePickerDialog = new TimePickerDialog(this, new SetYourTime(), hour, minute, true);  
        timePickerDialog.show();  
    }  
}
```

ניצור לוח שנה (מראה גם על השעה הנוכחית) ונשלוח ממנו את השעה והדקה הנוכחית

ניצור TimePickerDialog ונעביר לו את השעה והדקה ששלפנו. בהמשך נממש את התת מחלקה SetYourTime ונממש בה את הממשק והפונקציה של TimePickerDialog.

שימו לב! אם ה DateTime מופיעה עם סימון אדום, לחצו על הסימון ועל system.using.



לבסוף נממש את הפונקציה OnTimeSet ונקבע מה יקרה לאחר שהמשתמש יבחר שעה:

```
public class SetYourTime implements TimePickerDialog.OnTimeSetListener
{
    @Override
    public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
        String str = "Time is : " + hourOfDay + ":" + minute;
        btnSelectTime.setText(str);
    }
}
```

נשנה את הטקסט שיופיע
על הכפתור בהתאם

