

# פרק 11 - אחסון בקבצים

מערכת ההפעלה אנדרואיד תומכת בשני סוגי קבצים:

1. אחסון פנימי - שטח אחסון ששייך לאפליקציה בלבד.
2. אחסון חיצוני - שטח אחסון ששייך לכל האפליקציות.

בפרק זה נדון רק באחסון פנימי.

## אחסון פנימי:

1. כתיבה לקבצים בזיכרון הפנימי.
2. קבצים עם הרשאות גישה private **לא** יהיו נגישים למשתמש.
3. כאשר מסירים את האפליקציה מהמכשיר גם הקבצים יוסרו.

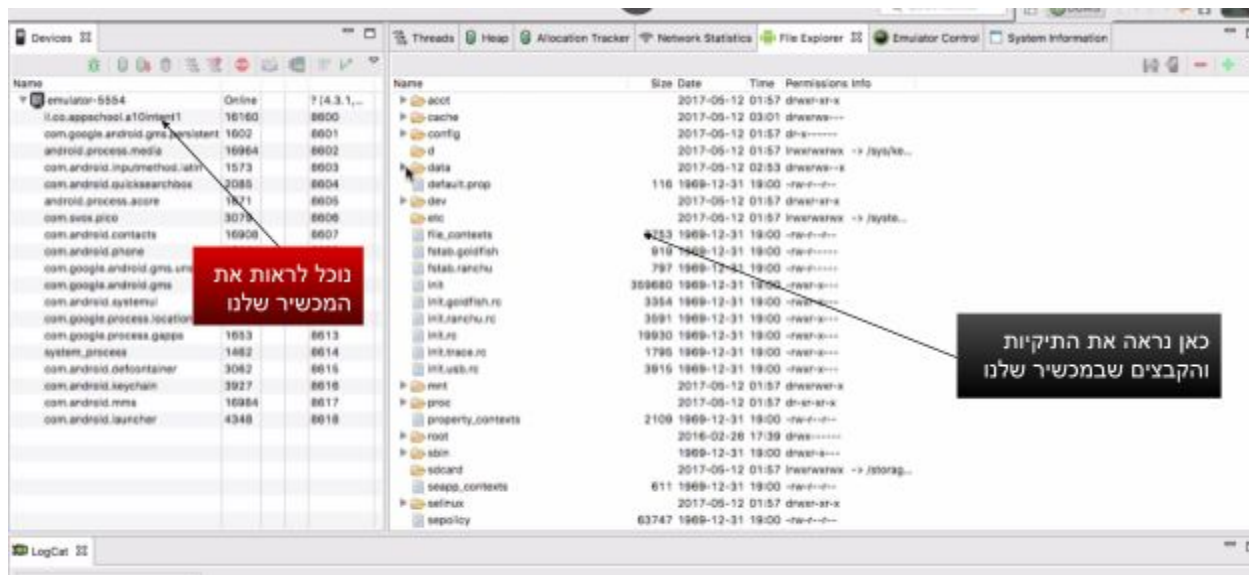
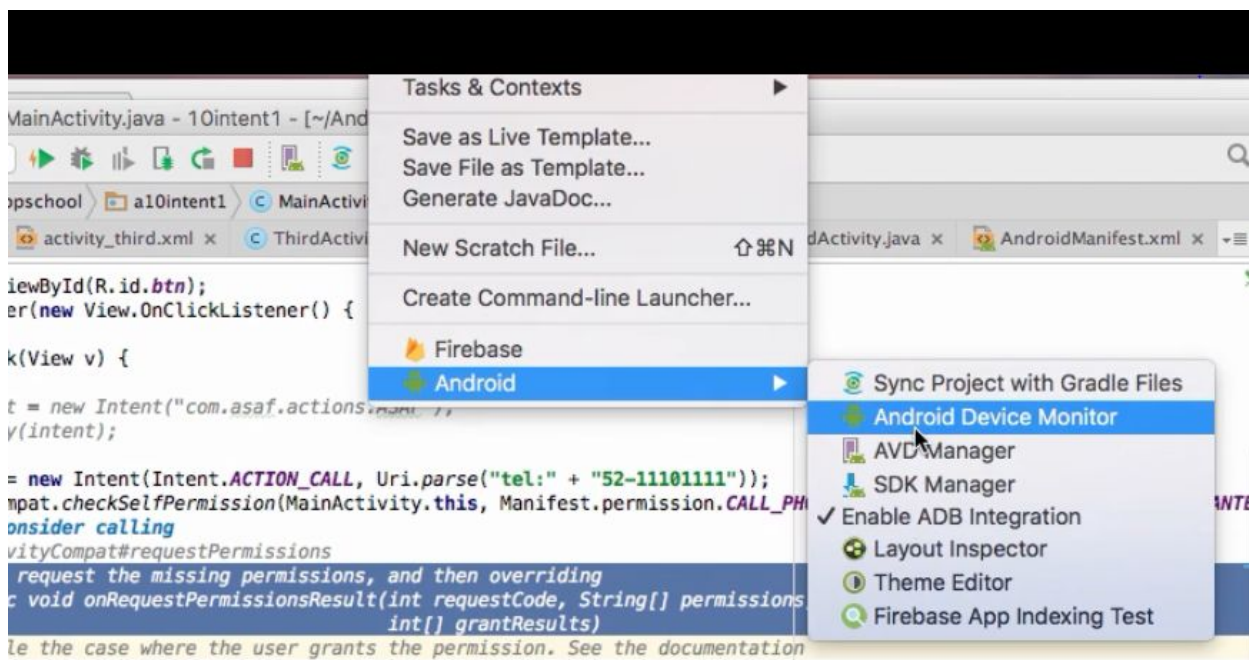
## פעולות בקבצים באנדרואיד

- getDir() - יוצרת או פותחת תיקיה בזיכרון הפנימי
- deleteFile() - מוחקת קובץ
- fileList() - מחזירה מערך של שמות הקבצים במכשיר
- getFileDir() - מחזירה את המיקום המוחלט שמאוחסן הקובץ כרגע

# DDMS



בשביל לראות את כל הקבצים של המכשיר ואת כל התוכן שלו נוכל להיכנס למערכת ה DDMS כך: (tools -> Android -> ADM)



בהמשך נראה איפה יישמר המידע שנשמור בטלפון.

## התחברות למערכת הקבצים של אנדרואיד

בשביל להתחבר למערכת הקבצים של אנדרואיד נשתמש באחת מהפונקציות הרצויות:

```
openFileInput("file1.txt");
```

הפונקציה מקבלת כפרמטר שם קובץ ומחזירה אובייקט מסוג `FileInputStream` שבעזרתו אנו יכולים לקרוא מהקובץ לתכנית.

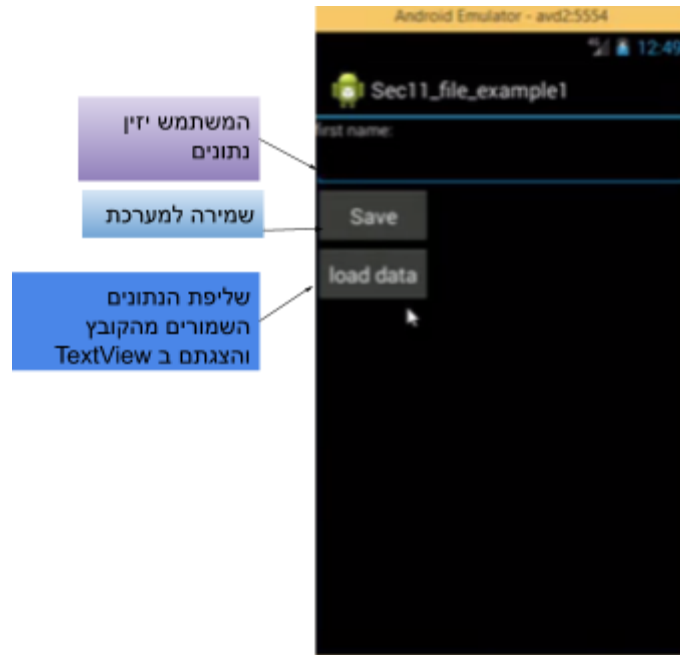
דוגמא :

```
FileInputStream fis = openFileInput("myfile.txt");
```

אותו הדבר לגבי הפונקציה **`openFileOutput`** שמאפשרת כתיבה לקובץ.

בדוגמה הראשונה לשימוש בקבצים נלמד איך לכתוב לקובץ שמור בטלפון את הפרטים שהמשתמש הזין.

נפתח אפליקציה חדשה ונעצב אותה:



```
<?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">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="first name:" />
    <EditText
        android:id="@+id/etTitle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10">
        <requestFocus />
    </EditText>
    <Button
        android:id="@+id/btnsave"
        android:layout_width="100dp"
        android:layout_height="50dp"
        android:text="Save" />
    <Button
        android:id="@+id/btnload"
        android:layout_width="100dp"
        android:layout_height="50dp"
```

```
        android:text="load data" />  
<TextView  
    android:id="@+id/textView1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="" />
```

ב MainActivity נצהיר על המשתנים שנשתמש בהם בתוכנית:

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

```
    Button btnsave, btnload;
```

```
    EditText et;
```

הצהרות

```
    TextView tv;
```

```
    FileOutputStream out;
```

```
    InputStream in;
```

```
    String str = null;
```

```
@Override
```

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

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

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

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

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

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

```
        tv = (TextView) findViewById(R.id.textView1);
```

```
        btnsave.setOnClickListener(this);
```

```
        btnload.setOnClickListener(this);
```

```
    }
```

הפניות לאובייקטים  
ומתן האזנה לכפתורים

## שלב 1 - כתיבה לקובץ:

@Override

```
public void onClick(View v) {
    if(v==btnsave)
    {
        try
        {
            str = et.getText().toString();
            out = openFileOutput("details1",MODE_PRIVATE);
            if(str!=null)
            {
                try {
                    out.write(str.getBytes(), 0, str.length());
                    out.close();
                }
            }
            Toast.makeText(this,out.getChannel().toString(),Toast.LENGTH_LONG).show();
        }
    }
}
```

במידה והמשתמש לחץ על btnSave

ננסה (כי לא תמיד נצליח) לשים את הפרטים שהמשתמש הזין בתוך ה string

נפתח את הקובץ userinfo מסוג txt (טקסט) (כיוון שהוא עדיין לא קיים, המערכת תיצור אותו)

הקובץ יהיה קריא לנו

מיקום המחזורת בקובץ יהיה בהתחלה (0)

אורך המחזורת שנשרשר יהיה באורך כל המחזורת שהמשתמש הזין

נסגור את הקובץ

Stream הוא הצינור שמקשר בין הקובץ שנפתח בטלפון לאפליקציה שלנו

catch (IOException e) {

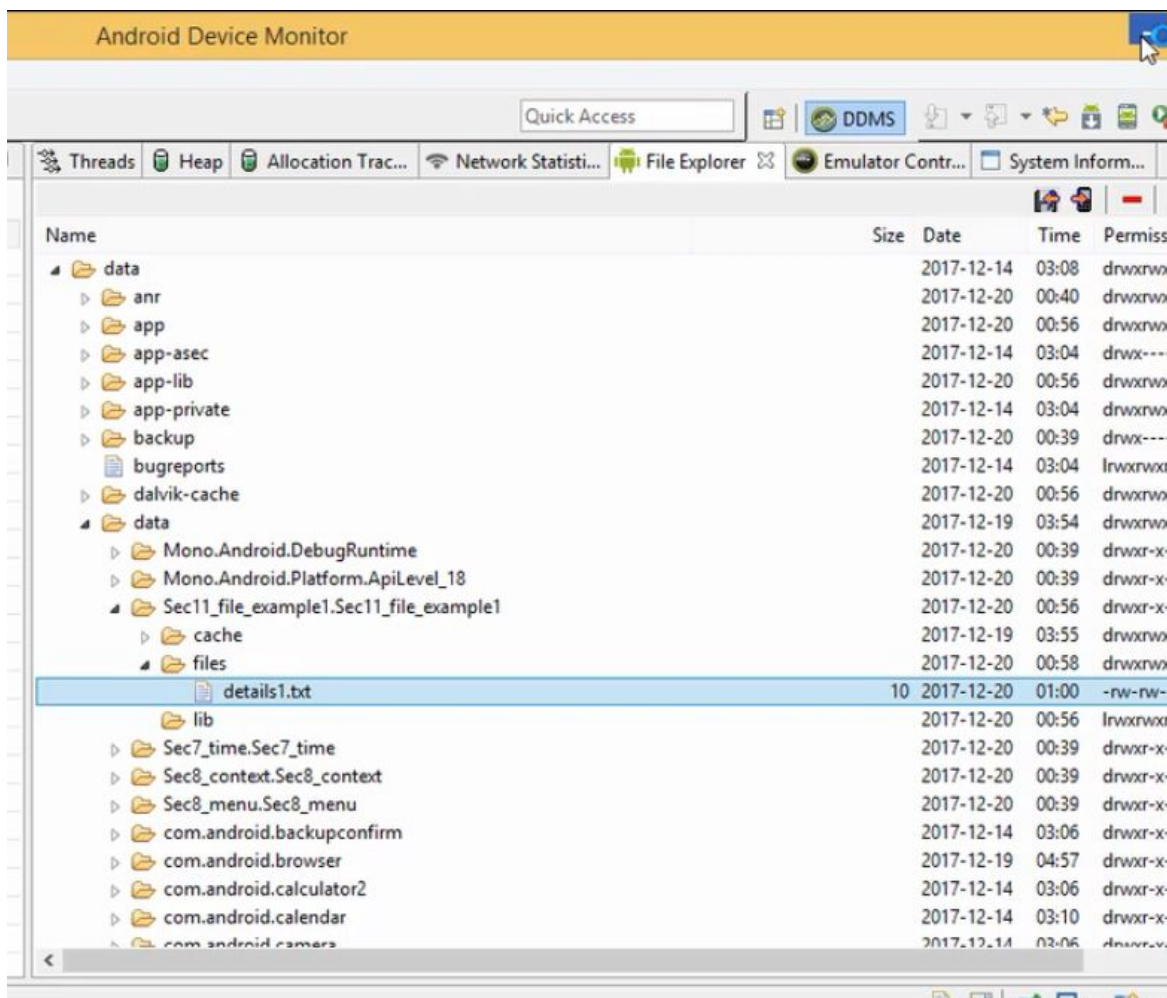
```
    e.printStackTrace();
}
}
} catch (FileNotFoundException e)
{
    e.printStackTrace();
}
}
}
```

לכל ניסיון try חובה שיהיה גם catch למקרה שהניסיון לא הצליח

במידה והגענו ל catch נציג את ההודעה שמודיעה מה הבעיה ב output



הריצו את האפליקציה ותוכלו לראות שהקובץ שיצרנו ב DDMS:  
 (שימו לב שאתם לא מריצים את האפליקציה על המכשיר שלכם, כי  
 אז לא תוכלו לראות ב DDMS)



## שלב 2 - קריאה מהקובץ:

```
else if(btnload==v)
```

```
{
```

```
try {
```

```
in=openFileInput("details1");
```

```
byte[]buffer=new byte[4096];
```

```
try {
```

```
in.read(buffer);
```

```
str=new String(buffer);
```

```
in.close();
```

```
if(str!=null)
```

```
tv.setText(str);
```

```
}
```

נצהיר על Stream שיקשר  
אותנו לתיקייה שיצרנו

נגדיר במערך את כמות  
הבייטים שנקרא מהקובץ

נשלוף מהקובץ מספר בייטים  
בגודל המערך שיצרנו

נהפוך את מערך הבייטים  
ששלפנו מהקובץ למחרוזת

נסגור את הstream

במידה והמחרוזת לא ריקה,  
נציג אותה ב TextView

```
catch (IOException e){  
    e.printStackTrace();  
}  
} catch (FileNotFoundException e)  
{  
    e.printStackTrace();  
}  
}  
}  
}
```

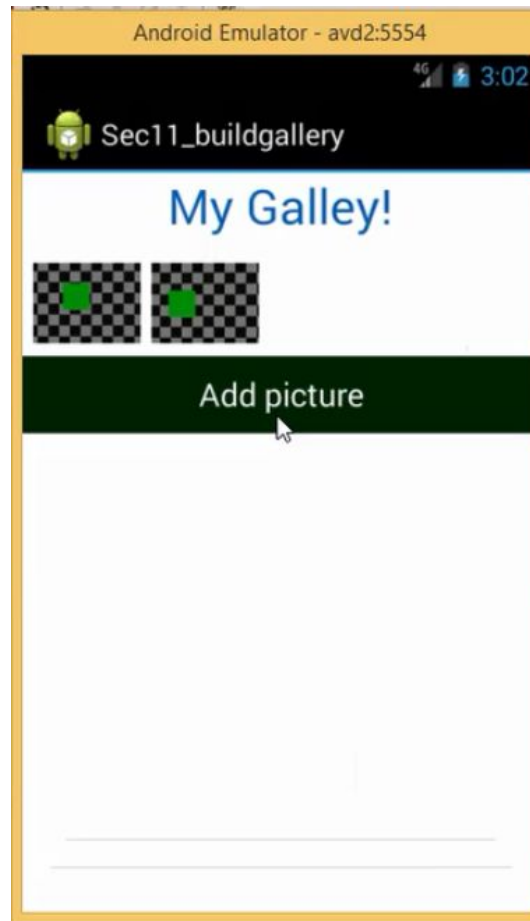
תוכלו להריץ את האפליקציה שלכם עכשיו (:

שימו לב שלמדנו עד עכשיו לשמור רק בזיכרון הפנימי של  
האפליקציה, לכן במידה ותמחקו את האפליקציה גם הקובץ עצמו  
ימחק.

דוגמה נוספת שנלמד היא בניית גלריית תמונות

כשנלחץ על כפתור ב MainActivity למצלמה.

התמונה שנצלם תישמר לנו בקובץ בזיכרון הפנימי של האפליקציה  
 ולאחר מכן תוצג במסך MainActivity בתוך HorizontalScrollView



שלב 1 - נערוך את קובץ ה XML של MainActivity

```
<?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" >
```

<TextView

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="first name:" />
```

<EditText

```
    android:id="@+id/etTitle"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10" >
    <requestFocus />
```

</EditText>

<Button

```
    android:id="@+id/btnsave"
    android:layout_width="100dp"
    android:layout_height="50dp"
    android:text="Save" />
```

<Button

```
    android:id="@+id/btnload"
    android:layout_width="100dp"
    android:layout_height="50dp"
    android:text="load data" />
```

<TextView

```
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="" />
```

</LinearLayout>

## שלב 2 - נצהיר על המשתנים בהם נשתמש במסך הנוכחי

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

```
    Button btnsave;
```

```
    Button btnload;
```

```
    EditText et;
```

```
    TextView tv;
```

```
    FileOutputStream out;
```

```
    InputStream in;
```

```
    String str = null;
```

## שלב 3 - ניתן הפניה לאובייקטים + ניתן לכפתורים האזנה ללחיצה

```
@Override
```

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

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

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

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

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

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

```
        tv = (TextView) findViewById(R.id.textView1);
```

```
        btnsave.setOnClickListener(this);
```

```
        btnload.setOnClickListener(this);
```

```
        logAllFiles();
```

```
    }
```

פונקציה שנמשך בהמשך.  
היא תיקרא בכל פעם  
שנפתח את המסך

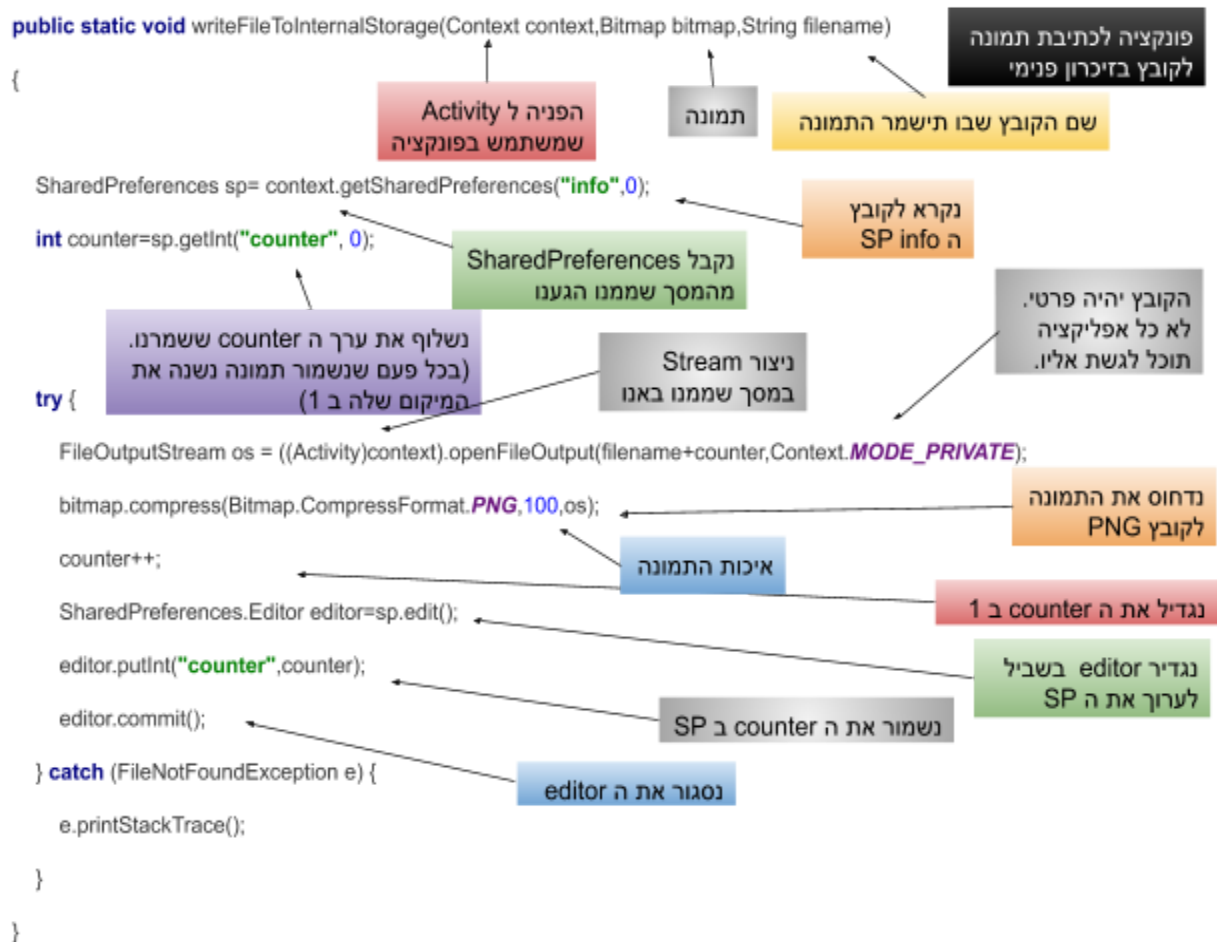
בשביל הנוחות של הקוד, נבנה קלאס נוסף.

הוא יהיה קלאס עזר כללי, שנוכל להיעזר בו מכל מסך.

```
public class HelperFile
```

```
}
```

נשים בו פונקציות סטטיות כלליות:



```
public static Bitmap readFromFileFromInternalStorage(Context context,String filename)
```

```
{
```

```
    Bitmap b=null;
```

```
    try {
```

```
        InputStream in = ((Activity)context).openFileInput(filename);
```

```
        b= BitmapFactory.decodeStream(in);
```

```
        in.close();
```

```
    } catch (IOException e) {
```

```
        e.printStackTrace();
```

```
    }
```

```
    return b;
```

```
}
```

פונקציה לקריאת נתונים  
מקובץ בזיכרון פנימי

הצהרה על התמונה  
ליתוכו נקרא את התמונה

הפניה למסך  
שממנו הגענו

שם הקובץ ממנו  
נקרא את התמונה

נעשה המרה כלפי מטה ל context, כדי  
שנוכל להשתמש בפונקציה  
OpenFileInput של האקטיבי.  
נפתח את הקובץ הנתון בעזרת Stream

נמיר את הנתונים  
שקראנו ל Bitmap

נחזיר את התמונה שקראנו

```
public static ArrayList<Bitmap>getAllFils(Context context)
```

```
{
```

```
    File mydir = context.getFilesDir();
```

```
    File lister = mydir.getAbsolutePath();
```

```
    ArrayList<Bitmap>arraylist=new ArrayList<Bitmap>();
```

```
    for (String list : lister.list()) {
```

```
        Toast.makeText(context,list,Toast.LENGTH_LONG).show();
```

```
        Bitmap b = readFromFileFromInternalStorage(context,list);
```

```
        arraylist.add(b);
```

```
    }
```

```
    return arraylist;
```

```
}
```

```
}
```

פונקציה שקוראת את כל  
הקבצים (נציג את כל התמונות)

ניצור רשימה של תמונות

עבור כל String ברשימה

נפעיל את הפונקציה שבנינו  
שקוראת מקובץ בזיכרון פנימי

נוסיף את  
התמונה לרשימה

נחזיר רשימה  
של כל התמונות



## נמשיך לעבוד עכשיו על MainActivity:

@Override

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

```
{
```

```
if(v==btnAdd)
```

כשהמשתמש ילחץ על btnAdd

```
{
```

```
Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
```

ניגש למסך למסך מצלמה

```
startActivityForResult(intent, 0);
```

```
}
```

```
}
```

@Override

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
```

```
super.onActivityResult(requestCode, resultCode, data);
```

```
if(resultCode==RESULT_OK&&requestCode==0)
```

```
{
```

```
Bitmap b= (Bitmap)data.getExtras().get("data");
```

```
HelperFile.writeFileToInternalStorage(this, b, "pic");
```

```
loadPicture();
```

לאחר שהמשתמש צילם תמונה  
ונחזור ממסך המצלמה:  
נשמור את התמונה ונקרא  
לפונקציה loadPicture  
שנממש בהמשך

```
}
```

```
}
```

