

# Android Files

[appSchool.co.il](http://appSchool.co.il)

# התחברות למערכת הקבצים של אנדרואיד

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

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

דוגמא :

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

אותו הדבר לגבי הפונקציה `openFileOutput` שמאפשרת כתיבה לקובץ.

# 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

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

# 1 אמצע

```
public class MainActivity extends Activity implements OnClickListener {  
    Button btnsave;  
    Button btnload;  
    EditText et;  
    TextView tv;  
    OutputStream out;  
    InputStream in;  
    String str=null;  
    @Override  
    public 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);  
    }  
}
```

# save to file

@Override

```
public void onClick(View v) {  
    // TODO Auto-generated method stub  
    switch (v.getId()) {  
        case R.id.btnsave:  
            try {  
                str=et.getText().toString();  
                out=openFileOutput("details1", MODE_PRIVATE|MDE_APPEND);  
                if(str!=null)O  
                out.write(str.getBytes(),0,str.getBytes().length);  
                out.close();  
            } catch (FileNotFoundException e1) {  
                // TODO Auto-generated catch block  
            } catch (IOException e1) {  
                // TODO Auto-generated catch block  
            }  
        }  
    }
```

# read from file

```
case R.id.btnload:
    try {
        in=openFileInput("details1");
        byte[]buffer=new byte[4096];
        in.read(buffer);
        str=new String(buffer);
    }
    catch (FileNotFoundException e)
    {
    }
    catch (IOException e) {
// TODO Auto-generated catch block
    }
```

```
finally
{
    try {
        in.close();
    }
    catch (IOException e)
    {
// TODO Auto-generated catch block
    }
    tv.setText(str);
    }
    break;
    default:
    break;
    }
}
```

# 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" >
    <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>
```



## 2 αμδι

```
public class MainActivity extends Activity implements
OnClickListener {
    Button btnsave;
    Button btnload;
    EditText et;
    TextView tv;
    OutputStream out;
    InputStream in;
    InputStreamReader insr;
    BufferedReader bufferreader;
    BufferedWriter bufferwriter;
    OutputStreamWriter opsr;
```

```
String str;
@Override
public 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);

}
```

# save to file

```
@Override
public void onClick(View v) {
// TODO Auto-generated method stub
switch (v.getId()) {
    case R.id.btnSave:
        try {
            out=openFileOutput("details1", MODE_PRIVATE);
            opsr=new OutputStreamWriter(out);
            bufferwriter=new BufferedWriter(opsr);
            bufferwriter.write(et.getText().toString());
            bufferwriter.flush();
        }
        catch (FileNotFoundException e1)
        {
            // TODO Auto-generated catch block
        } catch (IOException e1)
        {
            // TODO Auto-generated catch block
        }
}
```

```
finally
{
    try {
        out.close();
        //    opsr.close();
        //    bufferreader.close();
    }
    catch (IOException e)
    {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

break;
```

# read from file

```
case R.id.btnload:
try {
    in=openFileInput("details1");
    insr=new InputStreamReader(in);
    bufferreader=new BufferedReader(insr);
    StringBuffer sb=new StringBuffer();
    str=bufferreader.readLine();
    while(str!=null)
    {
        sb.append(str + "\n" );
        str=bufferreader.readLine();
    }
    tv.setText(sb.toString());
}
```

```
catch (FileNotFoundException e) {
    // TODO Auto-generated catch block
}
catch (IOException e)
{
    // TODO Auto-generated catch block
}
finally
{
    try {
        in.close();
        bufferreader.close();
        insr.close();
    }
catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}}
break;

default:
    break;
}
```

# 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" >
    <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>
```

# Resource file

# Resource file

ניתן לקרוא מ-resources אך לא ניתן לכתב לשם  
הקבצים לקריאה יהיו בדרך כלל בתיקיה res/raw

להלן דוגמא לפתיחת קובץ מ-resources :

```
InputStream is=this.getResource().openRawResource(R.  
raw.data)
```

```
private String streamToString(int id) {  
  
    InputStream file =getResources().openRawResource(id);  
  
    StringBuffer data =new StringBuffer();  
  
    DataInputStream dis =new DataInputStream(file);  
  
    String line =null;  
    try {  
  
        while ((line =dis.readLine()) !=null)  
            data.append(line +"\n");  
  
        dis.close();  
  
        file.close();  
    }  
    catch (IOException e) {  
    }  
    return data.toString();  
}
```

# External Storage

זיכרון חיצוני - sdcard - זיכרון קבוע בטלפון  
יש צורך בהרשאה :

```
<uses-permission android:name="android.  
permission.WRITE_EXTERNAL_STORAGE"/>
```



```
boolean mExternalStorageAvailable =false, mExternalStorageWriteable =false;

String state =Environment.getExternalStorageState();
if (Environment.MEDIA_MOUNTED.equals(state)) { //We can read and write the media
mExternalStorageAvailable =mExternalStorageWriteable =true;
}
else if (Environment.MEDIA_MOUNTED_READ_ONLY.equals(state)) {

//We can only read the media
mExternalStorageAvailable =true;
mExternalStorageWriteable =false;
}
else
{
//Something else is wrong. we can neither read nor write
mExternalStorageAvailable =mExternalStorageWriteable =false;
}
```

# פתיחה של קובץ

הפתיחה נעשית ע"י הפונקציה

`getExternalFilesDir|(string name)`

הקבצים שנוצרים לא נראים לעיני המשתמש

כל יישום יכול לקרוא מקובץ זה - לא מאובטח

כאשר נמחק את האפליקציה מהמכשיר - ימחקו גם  
הקבצים הנ"ל

# קבלת הנתיב לשורש הזיכרון החיצוני

```
Environment.getExternalStorageDirectory()
```

```
File file = new File(Environment.getExternalStorageDirectory(), "myfile.txt");
```

הקובץ יש ב- mnt/sdcard/myfile.txt

# Environment.getExternalStorageDir(string name)

```
File file =new File(getExternalFilesDir(Environment.DIRECTORY_MUSIC), "myfile.txt");
```

ניתן לקבל נתיב לתיקיות מיוחדות כגון :

DIRECTORY\_PICTURE  
DIRECTORY\_RINGTONES

יום ללא תכנות הוא יום מבוזבז!