



## الفصل الخامس:

### الأنشطة والجسور Activities and Intents

الصفحة	العنوان
3	1. دورة حياة النشاط Activity Lifecycle
4	التطبيقات Apps والذاكرة Memory والتخزين Storage
4	حالات النشاط
7	الإجرائية onCreate
7	الإجرائية onPause
9	الإجرائية onResume
10	الإجرائية onStop
11	الإجرائية onStart والإجرائية onRestart
12	الإجرائية onDestroy
13	اختبار حالة النشاط
14	2. الجسور Intents
15	الجسور Intents
17	الجسور Intents
18	إنشاء جسر Creating an Intent
19	الحصول على البيانات Extracting extra data
20	انتظار نتيجة Waiting for a result
20	إرجاع نتيجة Sending back a result
21	الحصول على النتيجة Grabbling the result
22	الجسر الضمني Implicit intent
22	تشغيل ملفات الصوت
23	3. مثال تعليمي
24	مثال تعليمي
25	الواجهة الأولى
30	الواجهة الثانية

## الكلمات المفتاحية:

التطبيقات Apps، الذاكرة Memory، التخزين Storage، نظام التسجيل Log، النشاط Activity، الجسر Intent.

## ملخص:

نعرض لمختلف الحالات التي يمر النشاط بها وإلى الأحداث الموافقة. ثم نعرض إلى آليات إنشاء التطبيقات ذات النشاطات المتعددة وكيفية التواصل بين هذه النشاطات باستخدام الجسور.

## أهداف تعليمية:

يتعرف الطالب في هذا الفصل على:

- حالات النشاط.
- أحداث النشاط.
- إنشاء جسر
- الحصول على البيانات
- انتظار نتيجة
- إرجاع نتيجة
- الحصول على النتيجة
- الجسر الضمني
- تشغيل ملفات الصوت.

## المخطط:

الأنشطة والجسور

- 3 وحدات (Learning Objects)

## 1. دورة حياة النشاط Activity Lifecycle

الأهداف التعليمية:

- دورة حياة النشاط

## التطبيقات Apps والذاكرة Memory والتخزين Storage

- تُخزّن التطبيقات والملفات في الجهاز ضمن القرص الداخلي والأقراص الأخرى الموصولة معه.
- يُمكنك عادةً معاينة التخزين على جهازك باستخدام:

### Settings → Storage

- عندما تقوم بتشغيل تطبيق، فإنه يتم تحميله في الذاكرة RAM أي تحميله من التخزين إلى الذاكرة.
- عندما تقوم بالخروج من تطبيق، يتم تحرير الذاكرة المشغولة من قبله أو يُمكن أن يبقى في الذاكرة.
- يُمكنك عادةً معاينة التطبيقات الموجودة في الذاكرة عن طريق:

### Settings → Apps → Running

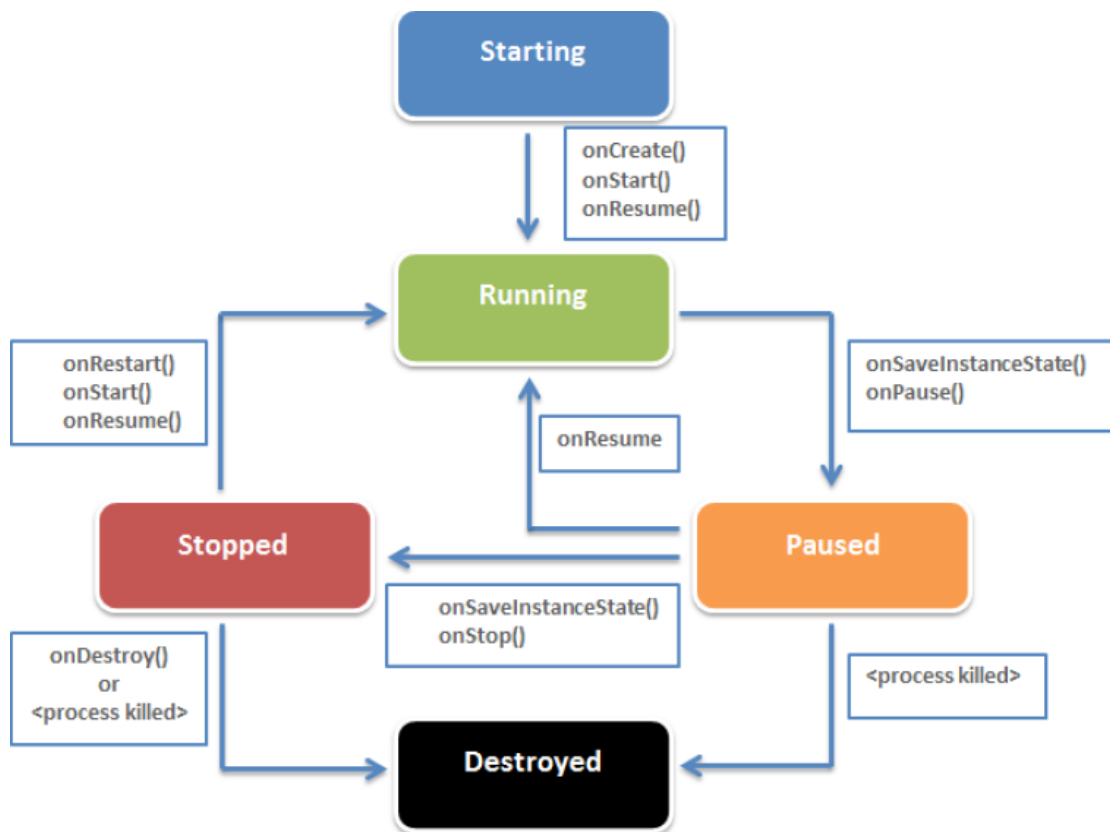


## حالات النشاط

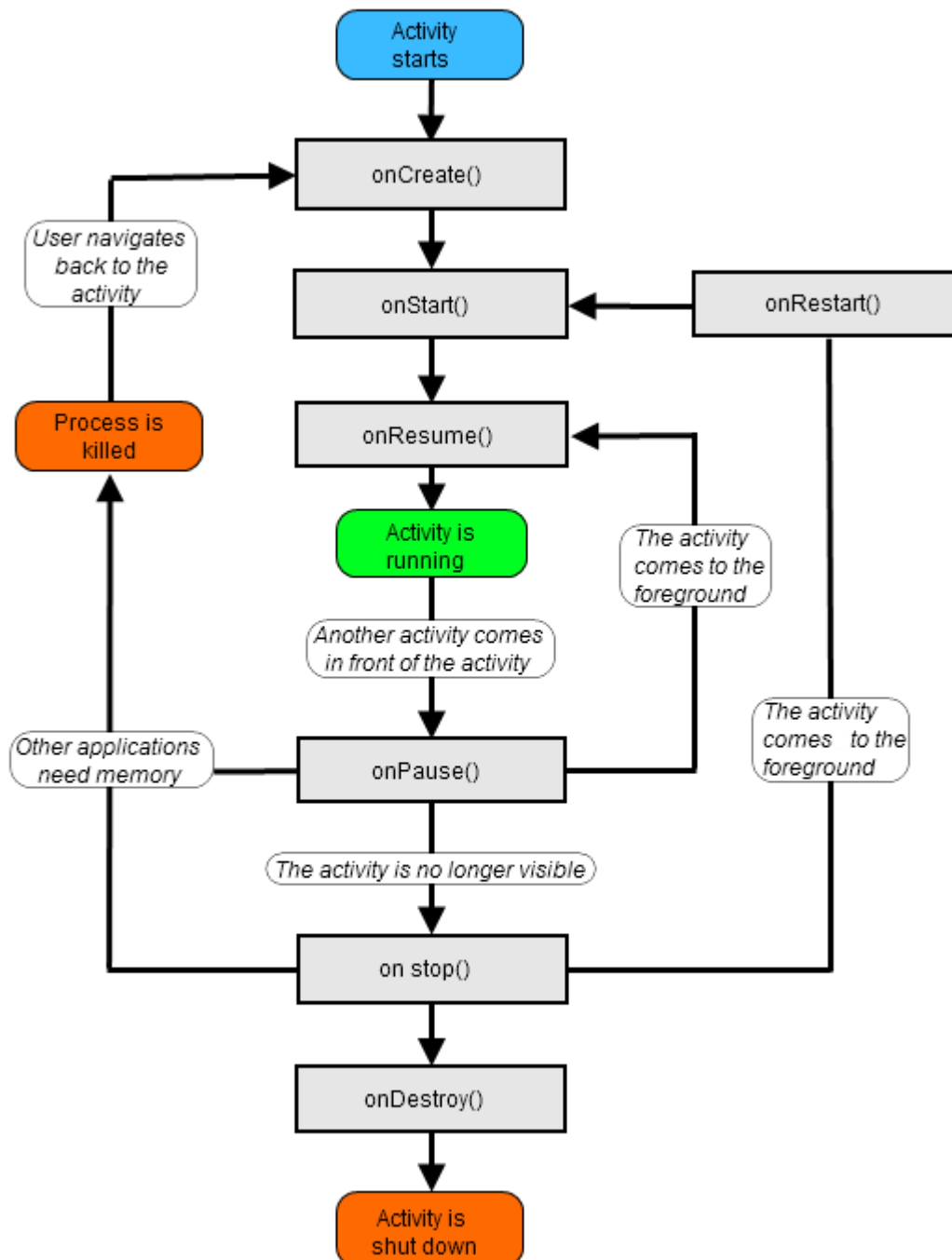
- يُمكن أن يكون النشاط في أحد الحالات التالية:

starting	In process of loading up, but not fully loaded.	في طور التحميل ولم ينته التحميل بعد.
running	Done loading and now visible on the screen.	بعد انتهاء التحميل وظهوره على الشاشة.
paused	Partially obscured or out of focus, but not shut down.	محبوب جزئياً أو ليس التركيز عليه، إنما غير متوقف.
stopped	No longer active, but still in the device's active memory.	ليس نشطاً ولكنه ما يزال في ذاكرة الجهاز النشطة.
destroyed	Shut down and no longer currently loaded in memory.	منهي وغير مُحمّل في الذاكرة.

- يتم تمثيل الانتقال بين الحالات عبر أحداث يمكن كتابة إجراءات لها في الكود: onCreate, onPause, onResume, onStop, onDestroy, ...
- يُبين المخطط التالي الأحداث والانتقال بين الحالات المختلفة للنشاط:

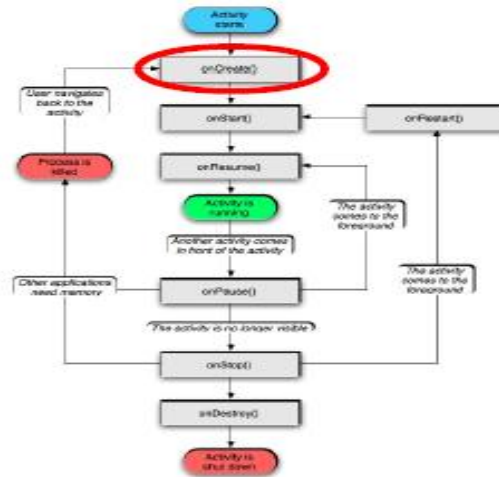


- كما يُمكن تمثيلها بالمخطط التالي:



## الإجرائية onCreate

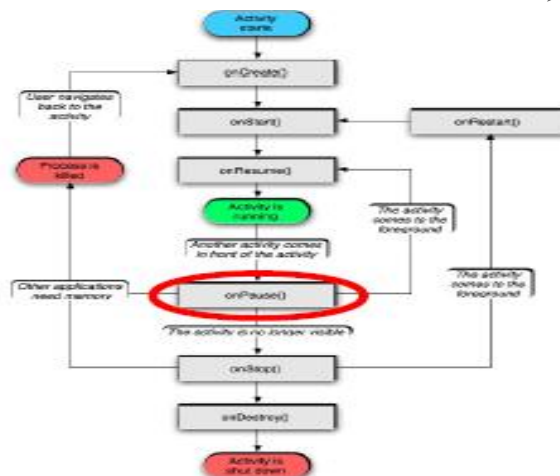
- يتم في هذه الإجرائية إنشاء وإعداد أغراض النشاط وتحميل المصادر الساكنة مثل الصور وإعداد القوائم.



```
public class FooActivity extends Activity {
    ...
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState); // always call super
        setContentView(R.layout.activity_foo); // set up layout
        any other initialization code; // anything else you need
    }
}
```

## الإجرائية onPause

- عند استدعاء الإجرائية onPause، يكون التطبيق مرئياً جزئياً ويكون متوقف مؤقتاً أو في طريق الإنهاء. يُمكن في هذه الإجرائية:



- إيقاف الإحياء Animations أو أي أعمال تستهلك المعالج CPU.
- حفظ التغييرات (Commit unsaved changes) مثل مسودة رسالة.

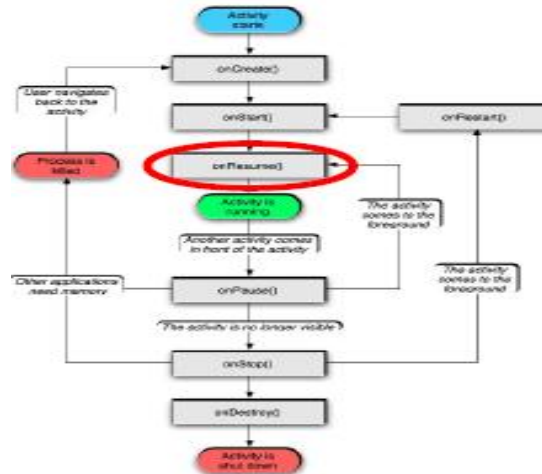


- تحرير موارد النظام (Release system resources) التي تستهلك حياة البطارية.

```
public void onPause() {  
    super.onPause(); // always call super  
    if (myConnection != null) {  
        myConnection.close(); // release resources  
        myConnection = null;  
    }  
}
```

## الإجرائية OnResume

- عند استدعاء الإجرائية OnResume، فإن النشاط يكون عائداً من حالة التوقف (Paused) إلى حالة العمل (Running) ثانيةً. كما أنها تُستدعى عند أول إنشاء وتحميل للنشاط. يُمكن في هذه الإجرائية:



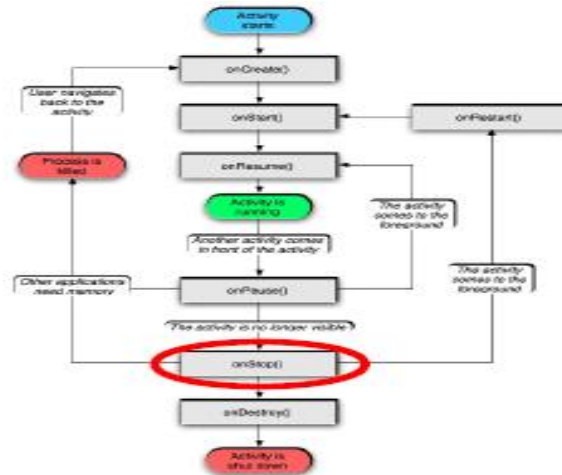
- تهيئة الموارد (Initialize resources) التي تم تحريرها في الإجرائية onPause.
- بدء/متابعة الإحياء (Animations) أو متابعة الأعمال التي يجب أن تعمل فقط عندما يكون النشاط ظاهراً على الشاشة.

```
public void onResume() {
    super.onPause(); // always call super

    if (myConnection == null) {
        myConnection = new ExampleConnect(); // init.resources
        myConnection.connect();
    }
}
```

## الإجرائية onStop

- تُستدعى الإجرائية onStop عندما لا يعود النشاط مرئي على الشاشة:



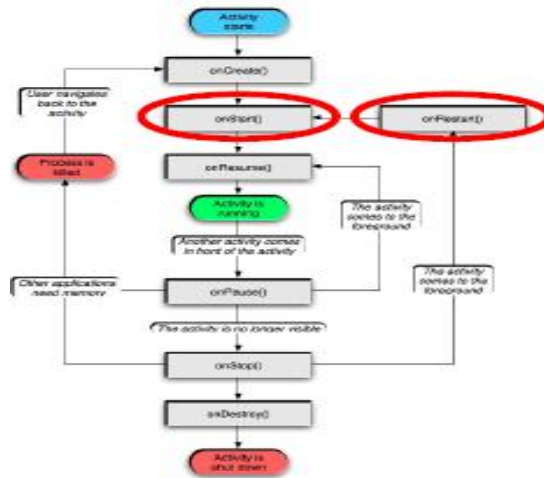
- إما أن المستخدم قام باختيار تطبيق آخر من نافذة التطبيقات الأخيرة Recent Apps.
- أو أن المستخدم قام بتشغيل نشاط جديد.
- أو أن المستخدم تلقى اتصال أثناء تشغيل النشاط.
- يُمكن أن يكون التطبيق مازال يعمل إلا أن النشاط لا يعمل.
- تُستدعى الإجرائية onPause دائماً قبل onStop.
- يتم في الإجرائية onStop عادةً القيام بمهام الإيقاف مثل الكتابة في قاعدة بيانات قبل الخروج.

```

public void onStop() {
    super.onStop(); // always call super
    ...
}
  
```

## الإجرائية OnStart والإجرائية OnRestart

- تُستدعى الإجرائية OnStart في كل مرة يبدأ النشاط.



- تُستدعى الإجرائية OnRestart بعد أن يكون النشاط متوقف (stopped) ثم بدأ لاحقاً (في جميع الحالات ما عدا أول مرة). تقوم عادةً الإجرائية OnRestart بفتح أي مصدر قامت الإجرائية OnStop بإغلاقه.

```
public void onStart() {
    super.onStart(); // always call super
    ...
}

public void onRestart() {
    super.onRestart(); // always call super
    ...
}
```

## الإجرائية OnDestroy

- تُستدعى الإجرائية OnDestroy بعد إنهاء التطبيق وإزالته من الذاكرة.



- لا يمكن التنبؤ تماماً متى سيتم استدعائها. ولذا يُفضل استخدام onPause أو onStop.

```

public void onDestroy() {
    super.onDestroy(); // always call super
    ...
}
  
```

## اختبار حالة النشاط

يُمكن استخدام نظام التسجيل LogCat لإظهار الرسائل الموافقة عندما تتغير حالة النشاط.

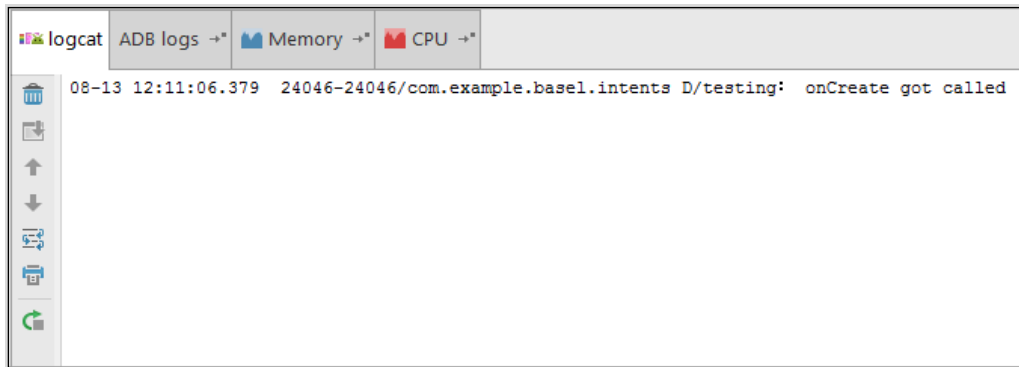
```
protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);

    Log.d("testing", "onCreate got called");

}
```



يكون لنظام التسجيل الطرق التالية:

Log.d("tag", "message")	debug message (for debugging)	رسالة تتبع
Log.e("tag", "message")	error message (fatal error)	رسالة خطأ
Log.i("tag", "message")	info message (low-urgency FYI)	رسالة معلومات
Log.v("tag", "message")	verbose message (rarely shown)	رسالة إعلام
Log.w("tag", "message")	warning message (non-fatal error)	رسالة تحذير
Log.wrf("tag", "message")	log stack trace of an exception	تسجيل تتبع المكس عند حصول استثناء

يُمكن للطريقة أن تقبل معامل استثناء إضافي كما يُبين المثال:

```
try { someCode(); }

    catch (Exception ex) {

        Log.e("error4", "something went wrong", ex);

    }
```

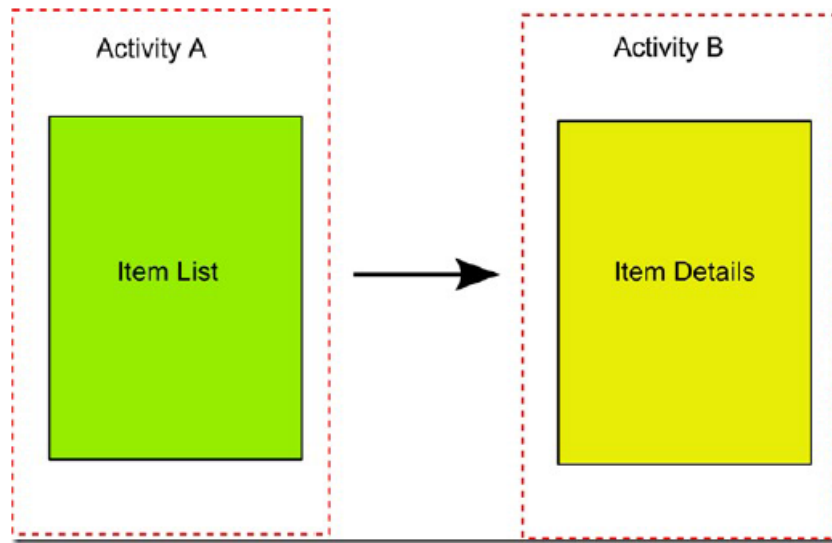
## 2. الجسور Intents

الأهداف التعليمية:

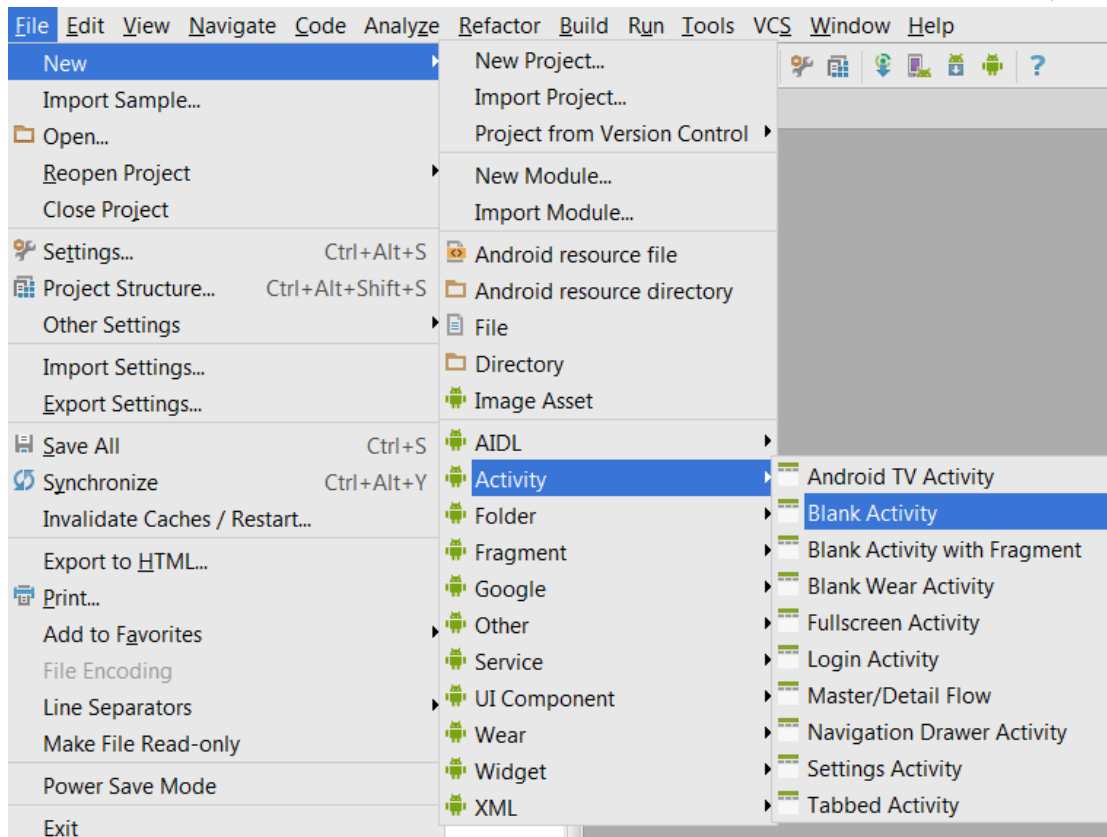
- الجسور Intents

## الجسور Intents

- يُمكن أن يكون للتطبيق العديد من النشاطات Activities. فمثلاً، في تطبيق "دفتر العناوين" (Address Book) يُمكن أن يحوي النشاط الرئيسي قائمة بالأشخاص وعند النقر على شخص معين، يتم الانتقال إلى نشاط آخر يُظهر بيانات هذا الشخص.
- يُمكن لنشاط A أن يُطلق نشاط آخر B تجاوباً مع حدث ما.
- يُمكن للنشاط A أن يُمرر بيانات للنشاط B.
- يُمكن للنشاط B أن يُعيد بدوره بيانات للنشاط A.



- يتم إضافة نشاط جديد من القائمة New:





- يؤدي إضافة النشاط الجديد إلى إضافة ملف XML. جديد في res/layouts وملف java. جديد في .src/java
- كما يتم تعديل الملف :AndroidManifest.xml

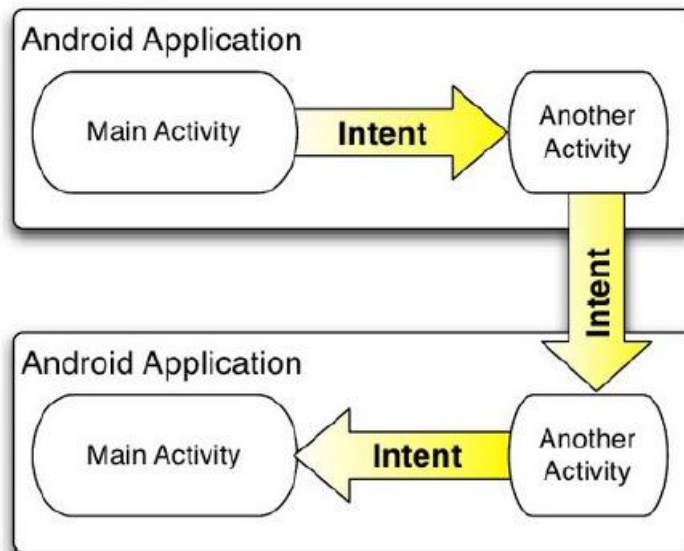
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.basel.intents" >

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >

        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".DetailsActivity"
            android:label="@string/title_activity_details" >
        </activity>
    </application>
</manifest>
```

## الجسور Intents

- يسمح الجسر بالانتقال من نشاط إلى نشاط ثاني.
- يُمكن للجسر أن يُخزّن بيانات لتمثيلها كمعاملات للنشاط الثاني.
- يُمكن للنشاط الثاني أن يُعيد بيانات للنشاط الأول.



## إنشاء جسر Creating an Intent

- لاستدعاء نشاط (عادةً استجابة لحدث معين)، قم بإنشاء غرض من الصف Intent مع تمرير النشاط في المعامل الثاني للبراني.
- ثم قم باستدعاء الطريقة startActivity مع تمرير الجسر المنشأ كمعامل لها. كما يُبين المثال التالي:

```
Intent intent = new Intent(this, ActivityName.class);  
startActivity(intent);
```

- إذا احتجت لتمرير معاملات أو بيانات إلى النشاط الثاني، يُمكنك استدعاء الطريقة putExtra على الجسر.
- تقوم الطريقة putExtra بتخزين البيانات كأزواج مفتاح/قيمة key/value.

```
Intent intent = new Intent(this, ActivityName.class);  
intent.putExtra("name1", value1);  
intent.putExtra("name2", value2);  
startActivity(intent);
```

## Extracting extra data الحصول على البيانات

- للحصول على البيانات في النشاط المُستدعى، يُمكنك الوصول إلى الجسر باستخدام الطريقة `.getIntent`
- لاستخراج البيانات من الجسر، يُمكن استخدام الطرق مثل:

`getStringExtra`, `getIntExtra`, `getStringExtra`,

```
public class SecondActivity extends Activity {  
  
    ...  
  
    public void onCreate(Bundle savedInstanceState) {  
  
        super.onCreate(savedInstanceState);  
  
        setContentView(R.layout.activity_second);  
  
        Intent intent = getIntent();  
  
        String extra = intent.getStringExtra("name");  
  
        ...  
  
    }  
  
}
```

## انتظار نتيجة Waiting for a result

- إذا أراد النشاط المستدعي الانتظار من أجل نتيجة من النشاط المُستدعى:
  - يجب استدعاء startActivityForResult عوضاً عن startActivity.
  - يجب تمرير معرف فريد إلى الإجرائية startActivityForResult يُمثل الإجراء المُنفذ (عادةً، يتم التصريح عن ثابت رقمي final int constant مع قيمة ما).
  - يجب كتابة الإجرائية onActivityResult والتي تُستدعى عند انتهاء النشاط الثاني: قم بفحص المعرف الفريد المُمرر كمعامل للإجرائية startActivityForResult.
  - يجب تعديل النشاط الثاني لإرجاع النتيجة المطلوبة.
  - استخدم الطريقة setResult والطريقة finish لإنهاء النشاط المُستدعى.

## إرجاع نتيجة Sending back a result

- يجب القيام بما يلي في النشاط الثاني المُستدعى:
  - إنشاء جسر للعودة.
  - تخزين أي بيانات مطلوبة في الجسر باستخدام setResult ومن ثم استدعاء finish.

```
public class SecondActivity extends Activity {
    ...

    public void myOnClick(View view) {
        Intent intent = new Intent();
        intent.putExtra("name", value);
        setResult(RESULT_OK, intent);
        finish(); // calls onDestroy
    }
}
```

## Grabbling the result الحصول على النتيجة

- تُبين التعليمات التالية كيفية الحصول على النتيجة في النشاط الأول المستدعي:

```
public class FirstActivity extends Activity {  
  
    private static final int REQ_CODE = 123; // MUST be 0-65535  
  
    public void myOnClick(View view) {  
  
        Intent intent = getIntent(this, SecondActivity.class);  
  
        startActivityForResult(intent, REQ_CODE);  
  
    }  
  
    protected void onActivityResult  
        (int requestCode, int resultCode, Intent intent) {  
  
        super.onActivityResult(requestCode, resultCode, intent);  
  
        if (requestCode == REQ_CODE) {  
  
            // came back from SecondActivity  
  
            String data = intent.getStringExtra("name");  
  
            Toast.makeText(this, "Got back: " + data,  
                Toast.LENGTH_SHORT).show();  
  
        }  
  
    }  
  
}
```

## الجسر الضمني Implicit intent

- يقوم الجسر الضمني باستدعاء تطبيق آخر دون تسميته مثل القيام باستدعاء المتصفح الافتراضي أو تحميل مشغل الموسيقى music player لتشغيل مقطوعة. كما تُبين الأمثلة التالية:

```
// make a phone call
Uri number = Uri.parse("tel:5551234");
Intent callIntent = new Intent(Intent.ACTION_DIAL, number);

// go to a web page in the default browser
Uri webpage = Uri.parse("http://www.stanford.edu/");
Intent webIntent = new Intent(Intent.ACTION_VIEW, webpage);

// open a map pointing at a given latitude/longitude (z=zoom)
Uri location = Uri.parse("geo:37.422219,-122.08364?z=14");
Intent mapIntent = new Intent(Intent.ACTION_VIEW, location);
```

## تشغيل ملفات الصوت



- ضع بعض ملفات الصوت (.wav, .mp3) ضمن المجلد `.app/src/main/res/raw`
- للتعامل مع ملف صوتي ضمن كود الجافا استخدم `R.raw.filename` (بدون كتابة اللاحقة).
- استخدم أسماء بسيطة للملفات لا تحوي سوى أحرف وأرقام.
- لتحميل الملفات الصوتية، استخدم الصف `MediaPlayer`:

```
MediaPlayer mp = MediaPlayer.create(this, R.raw.filename);
mp.start();
```

- من الطرق الأخرى للصف `MediaPlayer`:

`stop`, `pause`, `isLooping`, `isPlaying`, `getCurrentPosition`, `release`, `seekTo`, `setDataSource`, `setLooping`

### 3. مثال تعليمي

الأهداف التعليمية:

- مثال تعليمي.



## مثال تعليمي

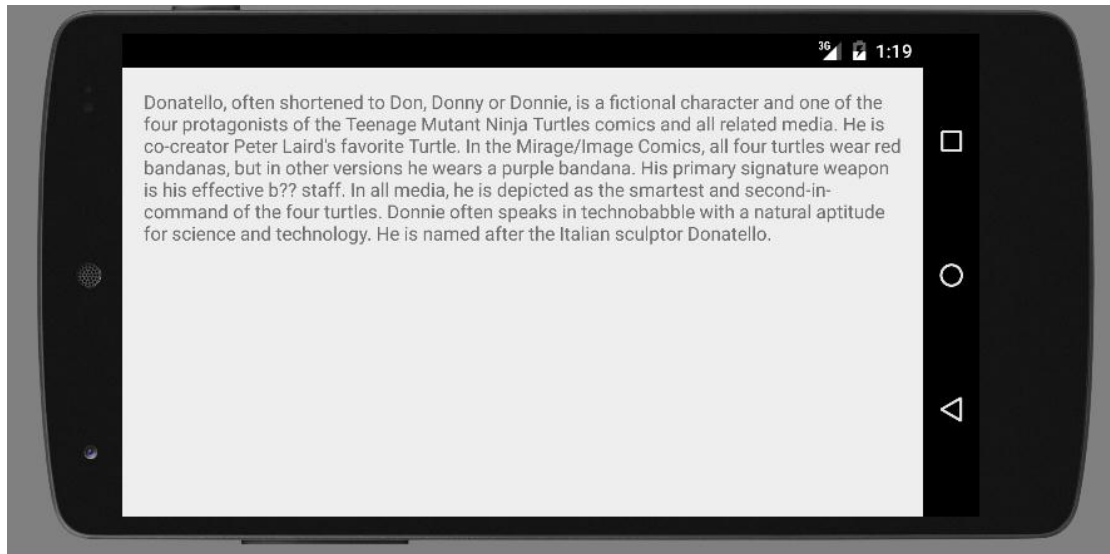
- عند إطلاق التطبيق يبدأ تشغيل مقطوعة موسيقية وتظهر الواجهة التالية:



- عند النقر على أحد أزرار الراديو تظهر صورة الشخصية الموافقة.



- عند النقر على صورة شخصية، يتم الانتقال إلى واجهة ثانية لعرض بيانات عن الشخصية:



## الواجهة الأولى

تحتوي الواجهة الأولى أربعة أزرار راديو و زر صورة.

يكون الملف activity\_main.xml للواجهة الأولى:

```
<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"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:gravity="top|center" android:orientation="vertical"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".MainActivity">

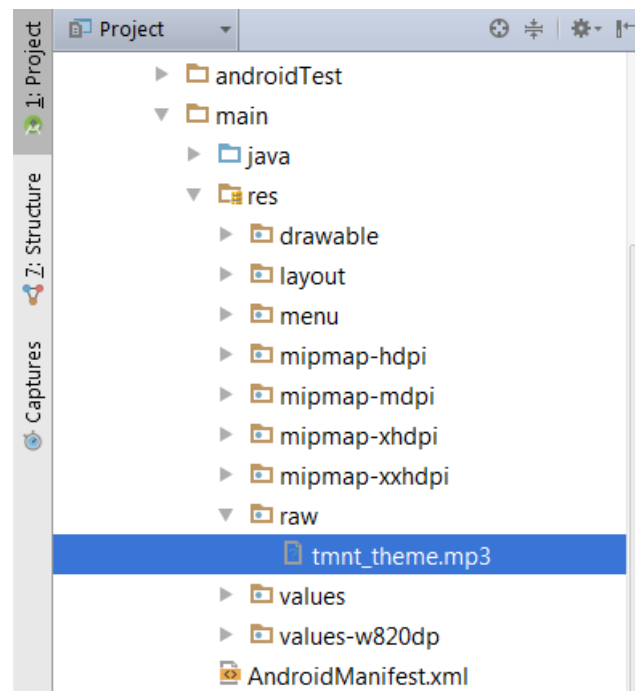
    <RadioGroup android:id="@+id/turtle_group"
        android:orientation="horizontal"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
        <RadioButton android:id="@+id/leo" android:onClick="pickTurtle"
            android:text="Leo" android:checked="true"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>

        <RadioButton android:id="@+id/mike" android:onClick="pickTurtle"
            android:text="Mike" android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
        <RadioButton android:id="@+id/don" android:onClick="pickTurtle"
            android:text="Don" android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>

        <RadioButton android:id="@+id/raph" android:onClick="pickTurtle"
            android:text="Raph" android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>
    </RadioGroup>

    <ImageButton android:id="@+id/turtle"
        android:onClick="onClickTurtleImage"
        android:src="@drawable/tmntleo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
</LinearLayout>
```

يحتوي المجلد res\raw الملف الصوتي tmnt\_theme.mp3



ويكون ملف الكود :MainActivity.java

```
package com.example.basel.intents;

import android.app.Activity;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.ImageButton;
import android.widget.RadioGroup;

public class MainActivity extends Activity {
    private MediaPlayer player;

    /*
     * Called when the activity first gets created.
     */
}
```

```
@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);
    player = MediaPlayer.create(this,
        R.raw.tmnt_theme);

    Log.d("testing", "onCreate got called");
}

public void onStart() {

    super.onStart();

    Log.d("testing", "onStart got called");
}

public void onResume() {

    super.onResume();

    if (player != null) {

        player.setLooping(true);

        player.start();

    }

    Log.d("testing", "onResume got called");
}

public void onPause() {

    super.onPause();

    player.pause();

    Log.d("testing", "onPause got called");
}
```

```
public void onStop() {  
    super.onStop();  
    Log.d("testing", "onStop got called");  
}  
  
public void onDestroy() {  
    super.onDestroy();  
    Log.d("testing", "onDestroy got called");  
}  
  
/*  
 * Called when the Details activity finishes running and comes back to here.  
 */  
  
@Override  
protected void onActivityResult  
    (int requestCode, int resultCode, Intent data) {  
    super.onActivityResult(requestCode, resultCode, data);  
  
}  
  
/*  
 * Called when the user clicks on the large TMNT image button.  
 * Loads the DetailsActivity for more information about that turtle.  
 */  
  
public void onClickTurtleImage(View view) {  
    Intent intent = new Intent(this, DetailsActivity.class);  
  
    RadioGroup group = (RadioGroup)  
        findViewById(R.id.turtle_group);
```

```
int id = group.getCheckedRadioButtonId();
intent.putExtra("turtle_id", id);
startActivity(intent);
}

/*
 * This method is called when the user chooses one of the turtle radio buttons.
 * In this code we set which turtle image is visible on the screen in the ImageView.
 */

public void pickTurtle(View view) {
    ImageButton img = (ImageButton)
        findViewById(R.id.turtle);

    if (view.getId() == R.id.Leo) {
        img.setImageResource(R.drawable.tmntleo);
    } else if (view.getId() == R.id.mike) {
        img.setImageResource(R.drawable.tmntmike);
    } else if (view.getId() == R.id.don) {
        img.setImageResource(R.drawable.tmntdon);
    } else if (view.getId() == R.id.raph) {
        img.setImageResource(R.drawable.tmntraph);
    }
}
}
```

## الواجهة الثانية

تحتوي الواجهة الثانية عنصر عرض نص فقط.

يكون الملف activity\_details.xml للواجهة الثانية:

```
<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"

    android:paddingLeft="@dimen/activity_horizontal_margin"

    android:gravity="top|center"

    android:orientation="vertical"

    android:paddingRight="@dimen/activity_horizontal_margin"

    android:paddingTop="@dimen/activity_vertical_margin"

    android:paddingBottom="@dimen/activity_vertical_margin"

    tools:context="com.example.stepp.layoutfun.DetailsActivity">

<TextView android:id="@+id/turtle_info"

    android:text=""

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"/>

</LinearLayout>
```

ويكون ملف الكود :DetailsActivity.java

```
package com.example.basel.intents;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
```

```
import android.widget.TextView;

public class DetailsActivity extends Activity {

    /*
     * Constant array of data about each of the four turtles.
     */

    private static final String[] TURTLE_DETAILS = {

        "Leonardo, or Leo, .....",

        "Michelangelo, Mike or Mikey ...",

        "Donatello, often shortened to Don. . .",

        "Raphael, or Raph, is a....."

    };

    /*
     * Called when the activity first gets created.
     */

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_details);

        // pull the turtle's ID out of the intent that the MainActivity used
        // to load me

        Intent intent = getIntent();

        int id = intent.getIntExtra("turtle_id", R.id.Leo);
```



```
String text = "";  
  
if (id == R.id.Leo) {  
    text = TURTLE_DETAILS[0];  
} else if (id == R.id.mike) {  
    text = TURTLE_DETAILS[1];  
} else if (id == R.id.don) {  
    text = TURTLE_DETAILS[2];  
} else { // if (id == R.id.raph)  
    text = TURTLE_DETAILS[3];  
}  
  
TextView tv = (TextView) findViewById(R.id.turtle_info);  
tv.setText(text);  
  
}  
}
```