



الفصل الثامن:

الكتل Fragments

الصفحة	العنوان
3	1. الكتل Fragments
4	Situation-Specific folders المجلدات الخاصة بالحالات
5	Portrait/Landscape التوضع الطولي والعرضي
5	Redundency مشكلة التكرار
6	الكتل Fragments
8	Fragment lifecycle دورة حياة الكتلة
10	Fragment Template قالب الكتلة
11	الكتل والنشاطات
12	مثال: التصريح عن حدث النقر
13	المعاملات المُمرة للنشاط و للكتلة
15	التواصل بين الكتل
16	الصفوف المشتقة من الكتل
17	2. مثال تعليمي
18	مثال تعليمي لاستخدام الكتل

الكلمات المفتاحية:

لكتل Fragments، المجلدات الخاصة بالحالات Situation-Specific Folders، التوضع الطولي Portrait، التوضع العرضي Landscape، دورة حياة الكتلة Fragment lifecycle، قالب الكتلة Fragment .template

ملخص:

نستعرض في هذا الفصل استخدام الكتل كطريقة للتقليل من التكرار عند كتابة تطبيقات يختلف سلوكها حسب حالة الجهاز.

أهداف تعليمية:

يتعرف الطالب في هذا الفصل على:

- دورة حياة الكتلة.
- قالب الكتل.
- التواصل بين الكتل.
- المعاملات المُمرة بين الكتل

المخطط:

الكتل Fragments

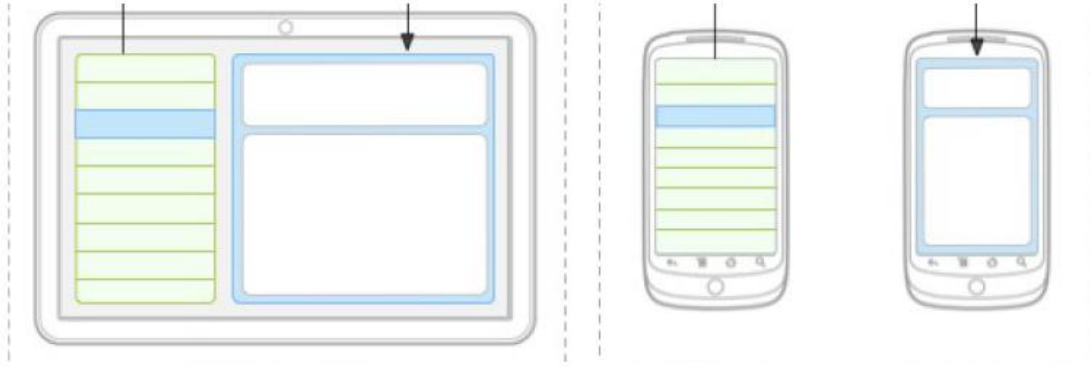
- 2 وحدة (Learning Objects)

1. الكتل Fragments

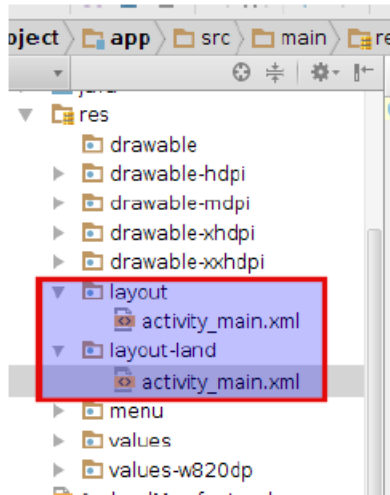
الأهداف التعليمية:

- استخدام الكتل

- يُمكن أن يكون للتطبيق تنسيقات مختلفة وذلك حسب الحالات المتعددة:
 - أجهزة مختلفة (هاتف، جهاز لوحي، ساعة).
 - أحجام شاشات مختلفة.
 - اتجاهات مختلفة طولية (portrait) أو عرضية (landscape).
 - إعدادات إقليمية مختلفة (لغة، بلد، ...).



المجلدات الخاصة بالحالات Situation-Specific folders

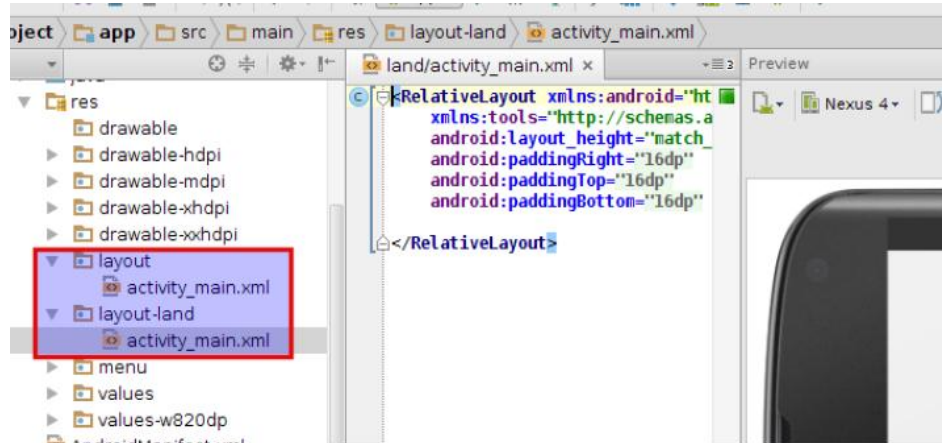


- تُستخدم لاحقات المجلدات (suffix) لإعلام التطبيق بمجموعة الخيارات المختلفة:
 - الاتجاه: للاتجاه الطولي (layout) أو الاتجاه العرضي (layout-land).
 - حجم الشاشة: صغير (small)، عادي (normal)، كبير (large)، كبير جداً (xlarge). (مثلاً layout-large).
 - كثافة الشاشة (مثلاً: drawable-hdpi): وتأخذ القيم التالية:

- xhdpi: 2.0 (twice as many pixels/dots per inch)
- hdpi: 1.5
- mdpi: 1.0 (baseline)
- ldpi: 0.75

التوضع الطولي والعرضي Portrait/Landscape

- لإنشاء نسق مختلف للتوضع العرضي:
 - قم بإنشاء مجلد في res باسم layout-land
 - ضع نسخة ثانية من ملف النسق (layout XML)
 - قم بالتعديلات التي تريدها على هذا الملف

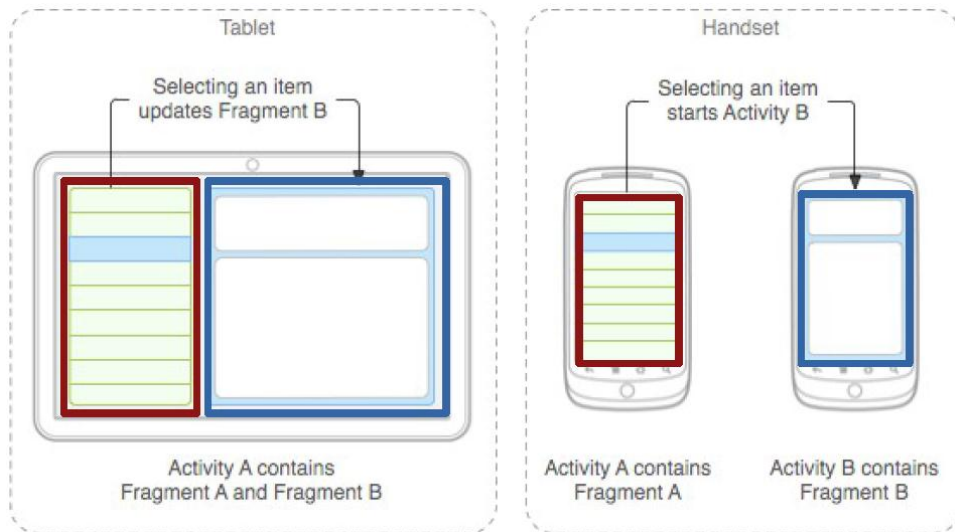


مشكلة التكرار Redundancy

- يؤدي النسق المتعلق بالحالة في الكثير من الحالات إلى تكرار نفس أقسام النسق أو الكود البرمجي الموافق.
- بشكل عام، قد تريد أن يتصرف الكود أيضاً وفق الحالة بشكل مختلف:
 - مثلاً في الحالة الطولية (portrait) قد تريد من زر أمر أن يفتح نشاط جديد (activity).
 - أما في الحالة العرضية (landscape) فتريد من هذا الزر إطلاق عرض جديد (view).

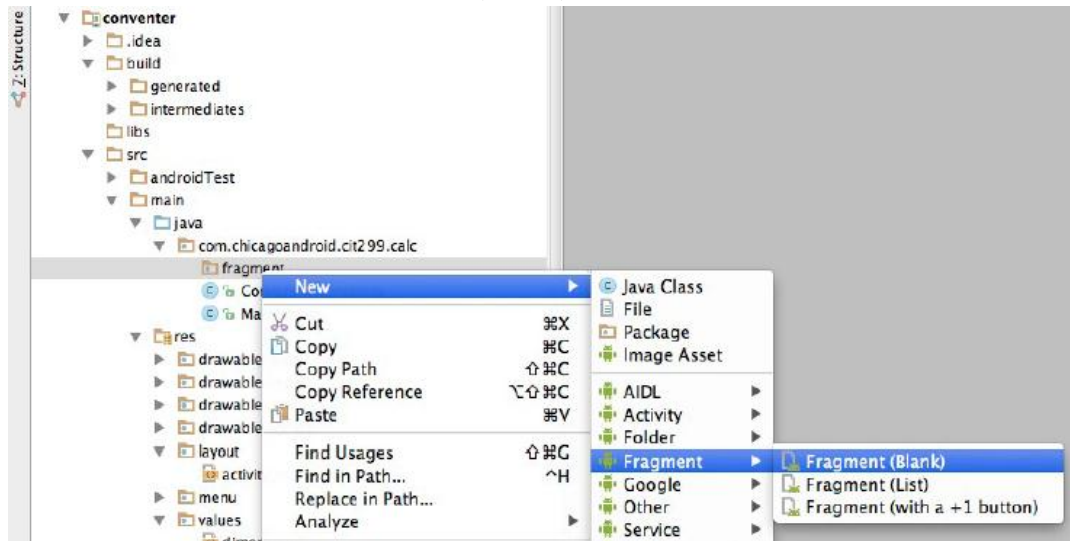
الكتل Fragments

- الكتلة (Fragment) هو جزء من الواجهة القابل للاستخدام في عدة نشاطات:
 - يُساعد في التعامل مع أجهزة وحجوم مختلفة.
 - يسمح بإعادة استخدام نفس الكتلة في نشاطات مختلفة.
- يُبين الشكل التالي مثلاً توضيحياً حيث يتألف النشاط A في حالة الجهاز اللوحي من الكتلة A والكتلة B. عند اختيار عنصر من الكتلة A، يتم تحديث الكتلة B (في نفس النشاط).
- أما التطبيق نفسه على جهاز هاتف فيتألف النشاط A من الكتلة A فقط. وعند اختيار عنصر من الكتلة A يتم فتح النشاط B والذي يحوي الكتلة B.



- لإنشاء كتلة في أندرويد ستديو، اختر:

New → Fragment → Fragment (blank)



- يُمكن في نشاط تضمين أكثر من كتلة:

```
<!-- activity_name.xml -->
```

```
<LinearLayout ...>  
<fragment ...  
  android:id="@+id/id1"  
  android:name="ClassName1"  
  tools:layout="@layout/name1" />  
<fragment ...  
  android:id="@+id/id2"  
  android:name="ClassName2"  
  tools:layout="@layout/name2" />  
</LinearLayout>
```

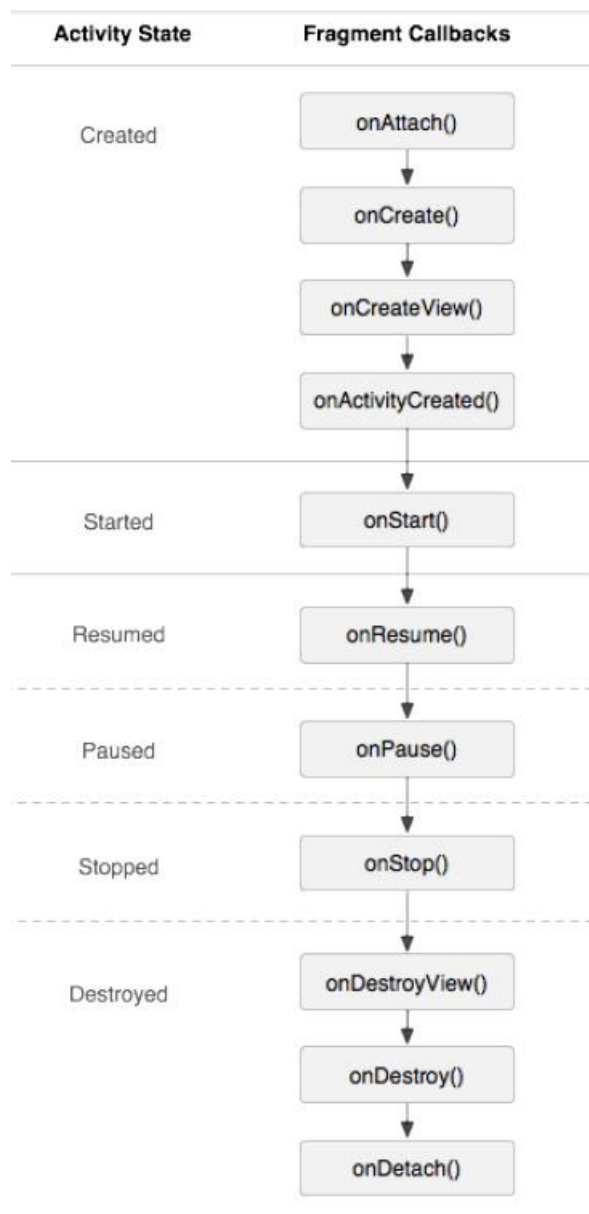


Fragment lifecycle دورة حياة الكتلة

- للكتلة دورة حياة وأحداث شبيهة بالنشاطات. من أهم الأحداث:

onAttach	to glue fragment to its surrounding activity	وضع الكتلة في النشاط
onCreate	when fragment is loading	تحميل الكتلة
onCreateView	method that must return fragment's root UI view	تُعيد عرض الكتلة الأساسي
onActivityCreated	method that indicates the enclosing activity is ready	جاهزية الكتلة
onPause	when fragment is being left/exited	الخروج من الكتلة
onDetach	just as fragment is being deleted	حذف الكتلة

- يُبين الشكل التالي استدعاءات أحداث الكتلة وفق حالة النشاط:



Fragment Template قالب الكتلة

- يكون للكتلة قالب التالي:

```
public class Name extends Fragment {  
  
    @Override  
  
    public View onCreateView(LayoutInflater inflater,  
  
        ViewGroup vg, Bundle bundle) {  
  
        // load the GUI layout from the XML  
  
        return inflater.inflate(R.layout.id, vg, false);  
  
    }  
  
    public void onActivityCreated(Bundle savedInstanceState) {  
  
        super.onActivityCreated(savedInstanceState);  
  
        // ... any other GUI initialization needed  
  
    }  
  
    // any other code (e.g. event-handling)  
  
}
```

الكتل والنشاطات

- يُشابه كود الكتلة كود النشاط مع بعض الاختلافات:
- لا يكون الوصول إلى العديد من طرق النشاط مباشرةً في الكتلة وإنما يتم عبر استدعاء الإجرائية

`.getActivity`

فمثلاً عوضاً عن:

```
Button b = (Button) findViewById(R.id.but);
```

نكتب:

```
Button b = (Button)
getActivity().findViewById(R.id.but);
```

- يُمكن أحياناً استخدام الإجرائية `getView` للوصول إلى نسق النشاط (activity layout).
- لا يُمكن ربط العناصر مع الأحداث في ملف XML وإنما يجب الربط في كود الجافا.
- للوصول إلى البيانات المُمَرَّة بواسطة الجسور `Intents`، يجب طلبها من النشاط المحيط بالكتلة.

مثال: التصريح عن حدث النقر

ليكن لدينا مثلاً في النشاط:

```
<Button android:id="@+id/b1"
android:onClick="onClickB1" ... />
```

وفي الكتلة:

```
Button android:id="@+id/b1" ... />
```

نكتب في الكود:

```
// in fragment's Java file

Button b = (Button) getActivity().findViewById(r.id.b1);
b.setOnClickListener(new View.OnClickListener() {
    @Override public void onClick(View view) {
        // whatever code would have been in onClickB1
    }
});
```

المعاملات المُمررة للنشاط و الكتلة

- يكون في النشاط الذي يقبل معاملات:

```
public class MainActivity extends Activity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

    }

}

public class Name extends Activity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.name);

        // extract parameters passed to activity from intent

        Intent intent = getIntent();

        int name1 = intent.getIntExtra("id1", default);

        String name2 = intent.getStringExtra("id2", "default");

        // use parameters to set up the initial state

        ...

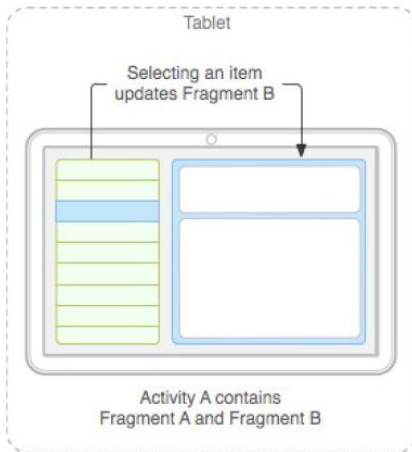
    }

    ...

}
```

- يكون في الكتلة التي تقبل معاملات:

```
public class Name extends Fragment {  
  
    @Override  
    public View onCreateView(LayoutInflater inflater,  
                             ViewGroup container, Bundle savedInstanceState) {  
        return inflater.inflate(R.layout.name, container, false);  
    }  
  
    @Override  
    public void onActivityCreated(Bundle savedInstanceState) {  
        super.onActivityCreated(savedInstanceState);  
  
        // extract parameters passed to activity from intent  
        Intent intent = getActivity().getIntent();  
        int name1 = intent.getIntExtra("id1", default);  
        String name2 = intent.getStringExtra("id2", "default");  
  
        // use parameters to set up the initial state  
        ...  
    }  
}
```



التواصل بين الكتل

- يُمكن أن يحوي نشاط مجموعة من الكتل.
- يُمكن للكتل أن تحتاج للتخاطب مع بعضها البعض.

- يُمكن استخدام الطريقة:

```
getFragmentManager().findFragmentById(R.id.id)
```

للوصول إلى أي كتلة عن طريق معرف الكتلة. كما يُبين المثال التالي:

```
Activity act = getActivity();
```

```
if (act.getResources().getConfiguration().orientation ==
    Configuration.ORIENTATION_LANDSCAPE) {
    // update other fragment within this same activity
    FragmentClass fragment = (FragmentClass)
    act.getFragmentManager().findFragmentById(R.id.id);
    fragment.methodName(parameters);
}
```


الصفوف المشتقة من الكتل

DialogFragment الصف

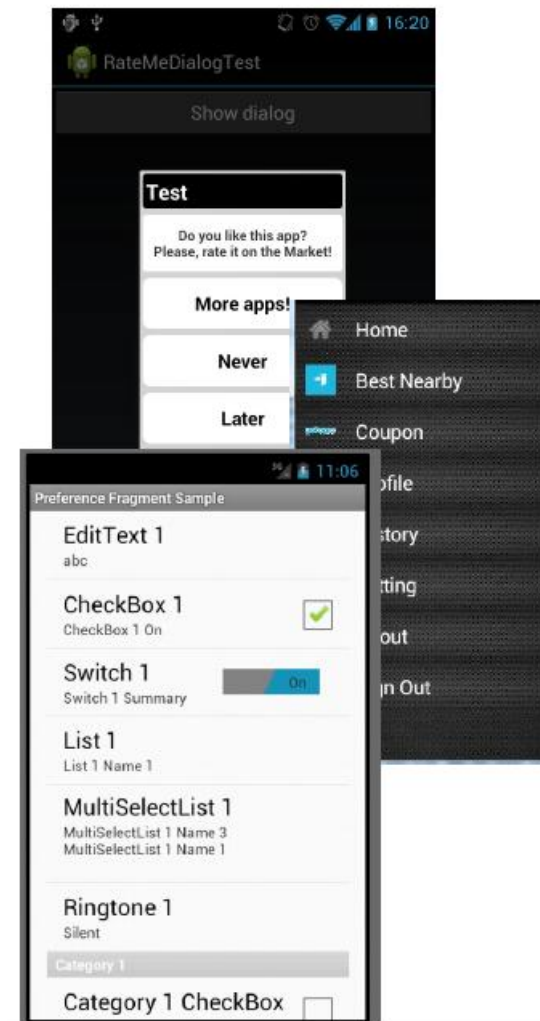
كتلة تظهر كصندوق حوار dialog box أعلى النشاط الحالي.

ListFragment الصف

يكون محتواها قائمة من العناصر.

PreferenceFragment الصف

كتلة يسمح محتواها بتغيير إعدادات التطبيق.



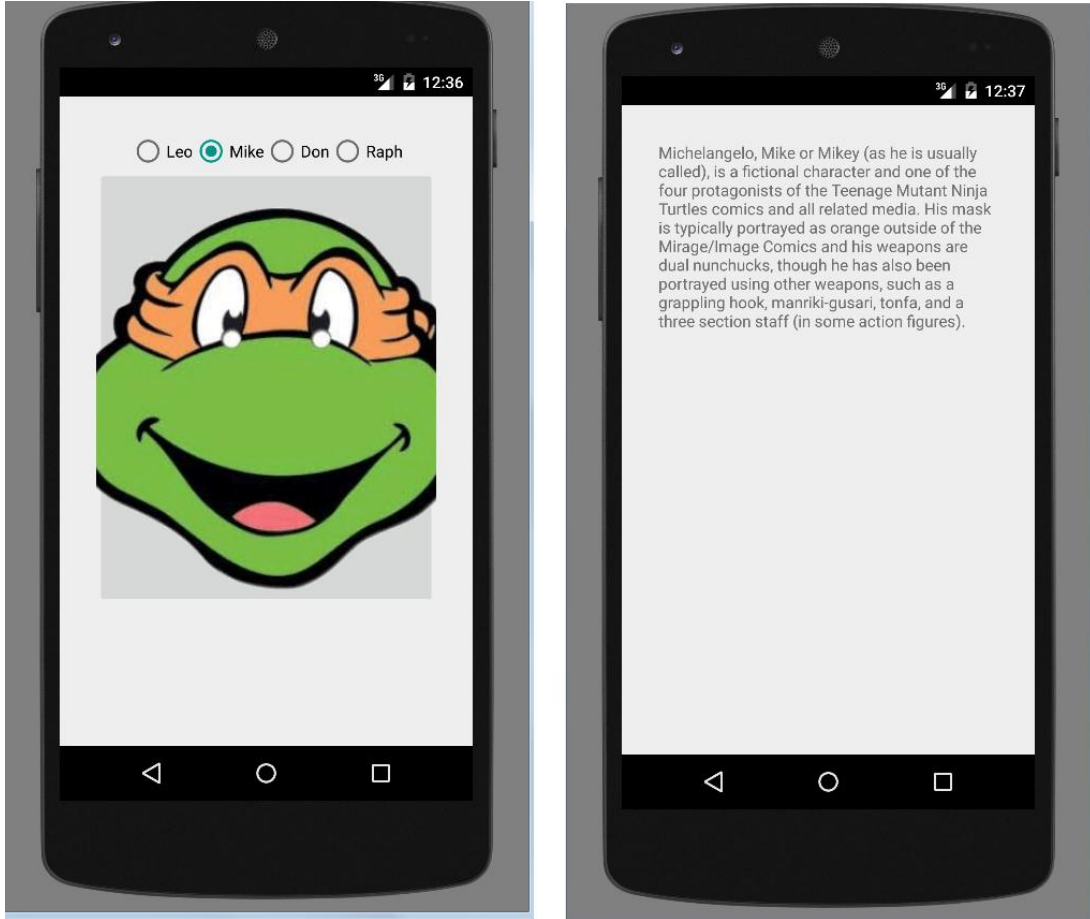
2. مثال تعليمي

الأهداف التعليمية:

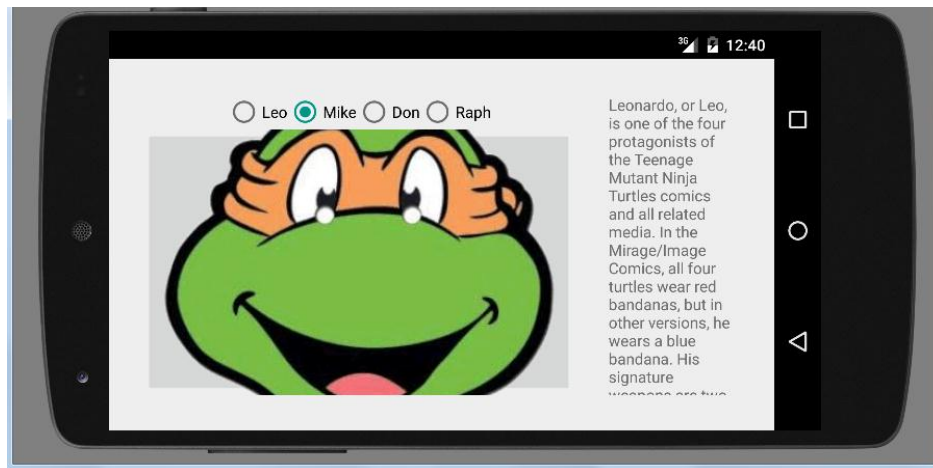
- مثال تعليمي

مثال تعليمي لاستخدام الكتل

- نقوم في المثال التعليمي التالي بتغيير سلوك التطبيق حسب اتجاه الجهاز طولي (Portrait) أم عرضي (Landscape).
- عند النقر على صورة أحد الشخصيات في حالة الاتجاه الطولي، يتم فتح نشاط جديد والانتقال إليه لعرض بيانات الشخصية.



- أما في كان الاتجاه عرضي (تذكر أنه لقلب اتجاه المحاكى، اضغط CTRL+F11)، فيتم عرض اللوحة عن الشخصية في نفس النشاط:



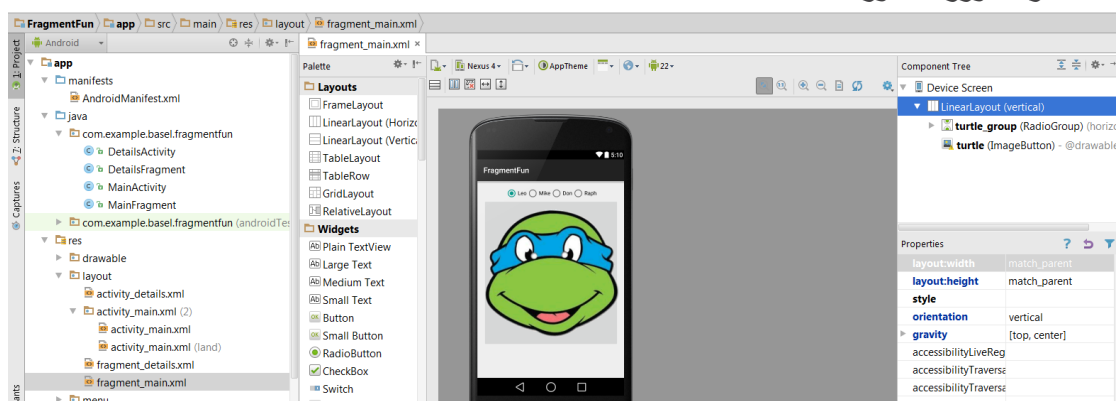
• مراحل العمل:

1. قم بإنشاء تطبيق جديد (يحتوي MainActivity.java و activity_main.xml).

2. قم بإضافة كتلة جديدة:

New → Fragment → Fragment(Blank)

وسمها fragment_main.xml، ثم اجعل لها التصميم التالي والذي يضم مجموعة أزرار الخيار الأربعة وزر الصورة:



و يكون الملف الموافق fragment_main.xml:

```
<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">
<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:text="Leo"

        android:checked="true"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"/>

    <RadioButton android:id="@+id/mike"

        android:text="Mike"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"/>

    <RadioButton android:id="@+id/don"

        android:text="Don"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"/>

    <RadioButton android:id="@+id/raph"

        android:text="Raph"

        android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"/>

</RadioGroup>

<ImageButton android:id="@+id/turtle"

        android:src="@drawable/tmnttleo"

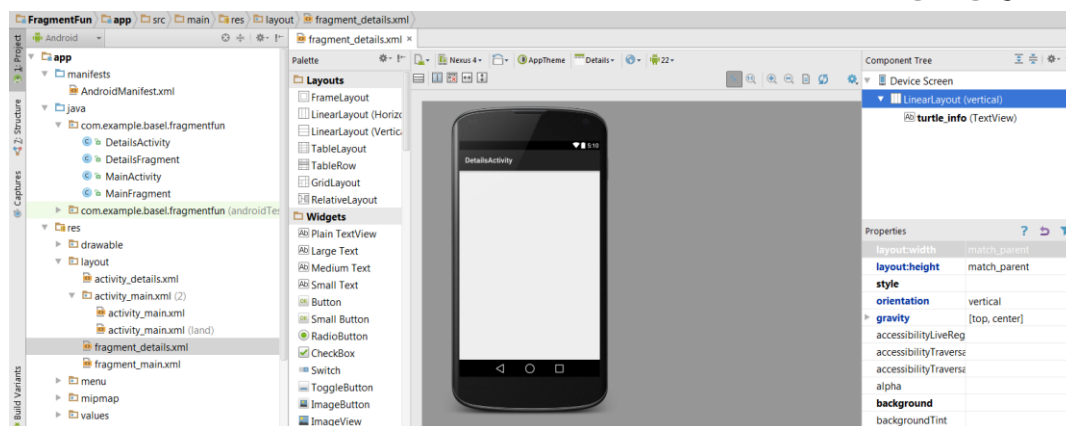
        android:layout_width="wrap_content"

        android:layout_height="wrap_content"/>

</LinearLayout>

```

3. قم بإضافة كتلة جديدة (سمّها fragment_details.xml) لها التصميم التالي والذي يحوي عنصر عرض نص :TextView



ويكون الملف الموافق fragment_details.xml:

```

<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.basel.fragmentfun.DetailsActivity">

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

        android:text="" android:layout_width="wrap_content"

        android:layout_height="wrap_content"/>

</LinearLayout>

```

4. قم بجعل النشاط الأساسي activity_main.xml يحوي الكتلة الأولى فقط:

```

<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">

<fragment android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:id="@+id/fragment1"

    android:name="com.example.basel.fragmentfun.MainFragment"

```

```
tools:layout="@layout/fragment_main"/>
```

```
</LinearLayout>
```

5. أنشئ نشاط جديد activity_details.xml واجعله يحوي الكتلة الثانية فقط:

```
<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.basel.fragmentfun.DetailsActivity">
```

```
<fragment android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:id="@+id/fragment2"
    android:name="com.example.basel.fragmentfun.DetailsFragment"

    tools:layout="@layout/fragment_details"/>
```

```
</LinearLayout>
```

6. قم بإنشاء مجلد جديد تحت المجلد res وسمّه layout-land وضع في داخله نسخة من الملف

activity_main.xml. قم بتعديل الملف (activity_main.xml(land بحيث يحوي على الكتلتين

معاً:

```
<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="horizontal"

        android:paddingRight="@dimen/activity_horizontal_margin"

        android:paddingTop="@dimen/activity_vertical_margin"

        android:paddingBottom="@dimen/activity_vertical_margin"

        tools:context=".MainActivity">

<fragment android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:id="@+id/fragment1"
        android:name="com.example.basel.fragmentfun.MainFragment"

        tools:layout="@layout/fragment_main"/>

<fragment android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:id="@+id/fragment2"
        android:name="com.example.basel.fragmentfun.DetailsFragment"

        tools:layout="@layout/fragment_details"/>

</LinearLayout>

```

7. اترك ملف الكود MainActivity.java بلا تعديل:

```
package com.example.basel.fragmentfun;
```

```
import android.app.Activity;

import android.support.v7.app.ActionBarActivity;

import android.os.Bundle;

import android.view.Menu;

import android.view.MenuItem;


public class MainActivity extends Activity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

    }

}
```

8. اترك ملف الكود DetailsActivity.java بلا تعديل:

```
package com.example.basel.fragmentfun;

import android.app.Activity;

import android.content.Intent;

import android.os.Bundle;

import android.support.v7.app.ActionBarActivity;

import android.view.Menu;

import android.view.MenuItem;

import android.widget.TextView;


public class DetailsActivity extends Activity {

    @Override
```

```

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_details);

}
}

```

9. قم بتعديل الملف MainActivity.java:

- يجب التصريح عن معالج الحدث للنقر على أحد أزرار الخيار ضمن الإجرائية :onActivityCreated

```

// attach event listener to radio button group

RadioGroup group = (RadioGroup)
getActivity().findViewById(R.id.turtle_group);

group.setOnCheckedChangeListener(new
RadioGroup.OnCheckedChangeListener() {

    @Override

    public void onCheckedChanged(RadioGroup group, int checkedId) {

        updateTurtleImage();

    }

});

```

- حيث تقوم الإجرائية updateTurtleImage() بوضع صورة الشخصية الموافقة في زر الصورة وذلك حسب زر الأمر الذي تمّ النقر عليه.
- يجب التصريح عن معالج الحدث على زر الصورة ضمن الإجرائية :onActivityCreated

```

// attach click event listener to turtle image button

ImageButton img = (ImageButton)
getActivity().findViewById(R.id.turtle);

img.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

```

```

        showDetailsAboutTurtle();
    }
});

```

- تقوم الإجرائية `showDetailsAboutTurtle()` بإظهار معلومات الشخصية كما يلي:
- في حال كان الاتجاه عرضي، يتم إظهار المعلومات في الكتلة الثانية الموجودة على نفس النشاط `(DetailsFragment)`.
- في حال كان الاتجاه طولي، يتم فتح نشاط جديد `(DetailsActivity)` وعرض المعلومات ضمنه.

```

package com.example.basel.fragmentfun;

import android.content.Intent;
import android.content.res.Configuration;
import android.os.Bundle;
import android.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageButton;
import android.widget.RadioGroup;

public class MainFragment extends Fragment {

    // these "request codes" are used to identify sub-activities that return results
    private static final int REQUEST_CODE_DETAILS_ACTIVITY = 1234;

    /*
     * This method initialize the fragment's layout.
     */

    @Override
    public View onCreateView

```

```
(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {

    return inflater.inflate
        (R.layout.fragment_main, container, false);
}

/*
 * This method is called after the containing activity is done being created.
 * This is a good place to attach event listeners and do other initialization.
 */
@Override

public void onActivityCreated(Bundle savedInstanceState) {

    super.onActivityCreated(savedInstanceState);

    // attach event listener to radio button group

    RadioGroup group = (RadioGroup)
        getActivity().findViewById(R.id.turtle_group);

    group.setOnCheckedChangeListener(new
        RadioGroup.OnCheckedChangeListener() {

        @Override

        public void onCheckedChanged(RadioGroup group, int checkedId) {

            updateTurtleImage();

        }

    });

    // attach click event listener to turtle image button

    ImageButton img = (ImageButton)
```

```
        getActivity().findViewById(R.id.turtle);
img.setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View v) {

        showDetailsAboutTurtle();

    }

});

}

/*
 * Shows more detailed information about the currently selected ninja turtle.
 * In portrait mode, this pops up a second activity.
 * In landscape mode, this updates the DetailsFragment within the same activity.
 */

public void showDetailsAboutTurtle() {

    RadioGroup group = (RadioGroup)
        getActivity().findViewById(R.id.turtle_group);

    int id = group.getCheckedRadioButtonId();

    if (getResources().getConfiguration().orientation ==
        Configuration.ORIENTATION_LANDSCAPE) {

        // show in same activity

        DetailsFragment frag = (DetailsFragment)
            getFragmentManager().findFragmentById(R.id.fragment2);

        frag.setTurtleId(id);

    } else {

        // launch details as its own activity
```

```
        Intent intent = new Intent(getActivity(),
            DetailsActivity.class);

        intent.putExtra("turtle_id", id);

        startActivityForResult(intent,
            REQUEST_CODE_DETAILS_ACTIVITY);
    }
}

/*
 * Updates which turtle image is showing
 * based on which radio button is currently checked.
 */

private void updateTurtleImage() {

    ImageButton img = (ImageButton)
        getActivity().findViewById(R.id.turtle);

    RadioGroup group = (RadioGroup)
        getActivity().findViewById(R.id.turtle_group);

    int checkedID = group.getCheckedRadioButtonId();

    if (checkedID == R.id.Leo) {

        img.setImageResource(R.drawable.tmntleo);

    } else if (checkedID == R.id.mike) {

        img.setImageResource(R.drawable.tmntmike);

    } else if (checkedID == R.id.don) {

        img.setImageResource(R.drawable.tmntdon);

    } else if (checkedID == R.id.raph) {

        img.setImageResource(R.drawable.tmntraph);
    }
}
```

```
    }

    if (getResources().getConfiguration().orientation ==
        Configuration.ORIENTATION_LANDSCAPE) {
        showDetailsAboutTurtle();
    }
}
}
```

10. قم بكتابة إجراءات الملف DetailsFragment.java

```
package com.example.basel.fragmentfun;

import android.content.Intent;
import android.os.Bundle;
import android.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

public class DetailsFragment extends Fragment {

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

    private static final String[] TURTLE_DETAILS = {
```


"Leonardo, or Leo, is one of the four protagonists of the Teenage Mutant Ninja Turtles comics and all related media. In the Mirage/Image Comics, all four turtles wear red bandanas, but in other versions, he wears a blue bandana. His signature weapons are two ninjato. Throughout the various media, he is often depicted as the eldest and leader of the four turtles, as well as the most disciplined. He is named after Leonardo da Vinci. In the 2012 series, he is the only turtle who harbors strong romantic affections for Karai, considering her his love interest.",

"Michelangelo, Mike or Mikey (as he is usually called), is a fictional character and one of the four protagonists of the Teenage Mutant Ninja Turtles comics and all related media. His mask is typically portrayed as orange outside of the Mirage/Image Comics and his weapons are dual nunchucks, though he has also been portrayed using other weapons, such as a grappling hook, manriki-gusari, tonfa, and a three section staff (in some action figures).",

"Donatello, often shortened to Don, Donny or Donnie, is a fictional character and one of the four protagonists of the Teenage Mutant Ninja Turtles comics and all related media. He is co-creator Peter Laird's favorite Turtle. In the Mirage/Image Comics, all four turtles wear red bandanas, but in other versions he wears a purple bandana. His primary signature weapon is his effective b?? staff. In all media, he is depicted as the smartest and second-in-command of the four turtles. Donnie often speaks in technobabble with a natural aptitude for science and technology. He is named after the Italian sculptor Donatello.",

"Raphael, or Raph, is a fictional character and one of the four protagonists of the Teenage Mutant Ninja Turtles comics and all related media. In the Mirage/Image Comics, all four turtles wear red bandanas over their eyes, but unlike his brothers in other versions, he is the only one who keeps the red bandana. Raphael wields twin sai as his primary weapon. (In the Next Mutation series, his sai stick together to make a staff-like weapon.) He is generally the most likely to experience extremes of emotion, and is usually depicted as being aggressive, sullen, maddened, and rebellious. The origin of Raphael's anger is not always fully explored, but in some incarnations appears to stem partly from the realization that they are the only creatures of their kind and ultimately alone. He also has a somewhat turbulent relationship with his older brother Leonardo because Leonardo is seen as the group's leader. Raphael is named after the 16th-century Italian painter Raphael. In 2011 Raphael placed 23rd on IGN's Top 100 Comic Book Heroes, a list that did not feature any of his brothers."

```
};
```

```
/*
```

```
 * This method initialize the fragment's layout.
```

```
*/
```

```
@Override
```

```
public View onCreateView(LayoutInflater inflater, ViewGroup
container,
```

```
        Bundle savedInstanceState) {

    return inflater.inflate(R.layout.fragment_details, container,
false);

}

/*
 * This method is called after the containing activity is done being created.
 * This is a good place to attach event listeners and do other initialization.
 */
@Override
public void onActivityCreated(Bundle savedInstanceState) {

    super.onActivityCreated(savedInstanceState);

    // pull out the turtle ID to show from the activity's intent
    Intent intent = getActivity().getIntent();

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

    setTurtleId(id);

}

/*
 * Sets the actively selected ninja turtle text based on the given resource ID.
 */

public void setTurtleId(int id) {

    int index;

    if (id == R.id.Leo) {

        index = 0;

    } else if (id == R.id.mike) {

        index = 1;

    }

}
```

```
    } else if (id == R.id.don) {  
        index = 2;  
    } else { // if (id == R.id.raph)  
        index = 3;  
    }  
  
    String text = TURTLE_DETAILS[index];  
    TextView tv = (TextView)  
        getActivity().findViewById(R.id.turtle_info);  
    tv.setText(text);  
}  
}
```