

## CHAPTER

## 04

# Listview, ListActivity IIละ: Spinner

## เนื้อหาในบทนี้

- ◆ การแสดงรายการข้อมูลด้วย ListView
- ◆ การแสดงรายการข้อมูลด้วย ListActivity
- ◆ การแสดงรายการข้อมูลที่เป็นตัวเลือกแบบ Multiple Choice
- ◆ การตรวจสอบสถานะของตัวเลือกทั้งหมดใน ListView
- ◆ การสร้าง ListView ที่เราออกแบบ Layout ของแต่ละข้อมูลเอง
- ◆ การสร้าง Custom Adapter เพื่อใช้งานกับ ListView
- ◆ การแสดงรายการข้อมูลด้วย Spinner
- ◆ การสร้าง Spinner ที่เราออกแบบ Layout ของแต่ละข้อมูลเอง

## การแสดงผลรายการข้อมูลด้วย ListView

ในการเขียนแอป เรามักต้องทำงานกับกลุ่มของข้อมูลที่มีลักษณะเป็นรายการ (List) อยู่บ่อยครั้ง ซึ่งแอนดรอยด์มีวิธหลายชนิดที่จะช่วยให้เราแสดงผลและทำงานกับข้อมูลเหล่านั้นได้โดยง่าย หนึ่งในนั้นก็คือ ListView

### ตัวอย่าง

ตัวอย่างนี้จะแสดงชื่อประเทศในภูมิภาคอาเซียนโดยใช้ ListView ซึ่งเมื่อคลิกชื่อประเทศก็จะแสดงข้อความใน TextView

#### 1 กำหนด Layout ของหน้าจอ

โปรเจ็ค ListViewDemo, ไฟล์ res\layout\activity\_\_main.xml

```
<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:id="@+id/text"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#66ff99"
        android:gravity="center"
        android:padding="5dp"
        android:text="@string/hello_world" />

    <ListView
        android:id="@+id/list_of_countries"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

#### 2 เพิ่มการประกาศตัวแปรอาร์เรย์ countries ในแอคทิวิตี

โปรเจ็ค ListViewDemo, ไฟล์ MainActivity.java

```
static final String[] countries = new String[] {
    "Brunei", "Cambodia", "Indonesia", "Laos", "Malaysia",
    "Myanmar (Burma)", "Philippines", "Singapore", "Thailand",
    "Vietnam" };
```

### 3 เพิ่มโค้ดในเมธอด onCreate ของแอคทิวิตี้

โปรเจ็ค ListViewDemo, ไฟล์ MainActivity.java

```
final TextView text = (TextView) findViewById(R.id.text);
ListView list = (ListView) findViewById(R.id.list_of_countries);

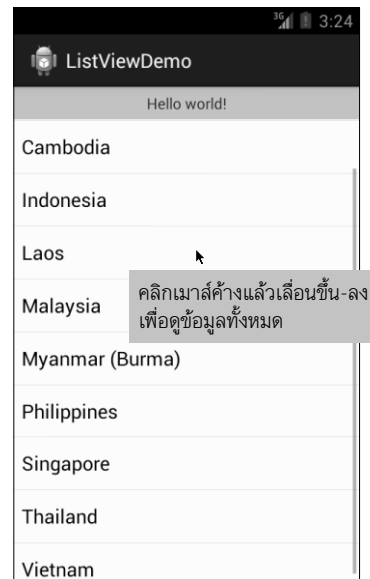
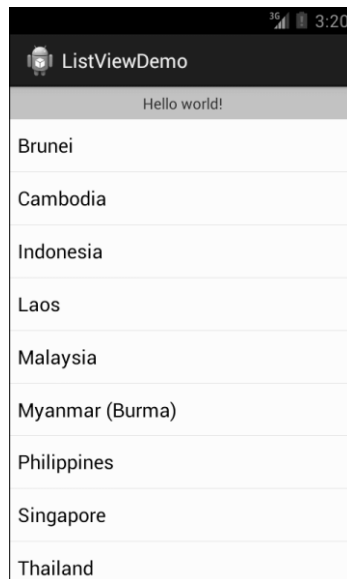
ArrayAdapter<String> adapter = new ArrayAdapter<String>(
    this, android.R.layout.simple_list_item_1, countries);
list.setAdapter(adapter);

list.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> av, View v, int position,
        long id) {
        String msg = "You have selected " + ((TextView) v).getText();
        msg += " at position " + String.valueOf(position);

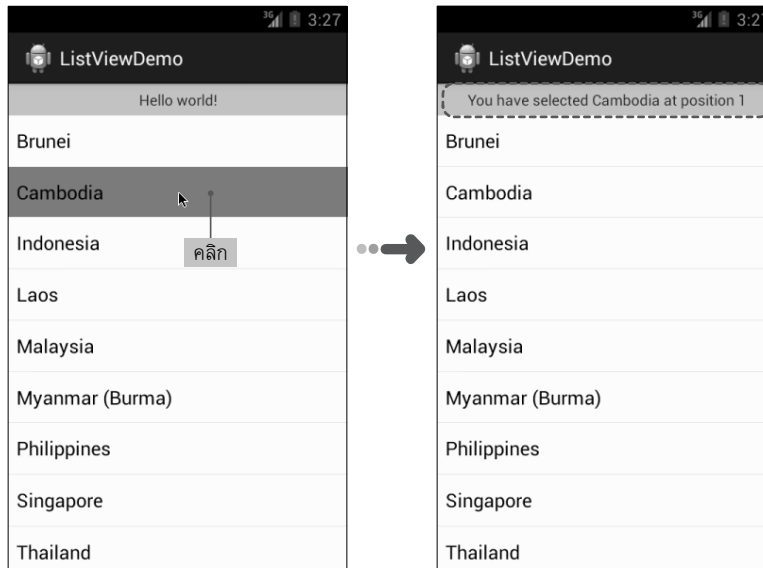
        text.setText(msg);
    }
});
```

### ผลการรัน

เมื่อรันจะได้หน้าจอที่มี TextView อยู่ด้านบน และพื้นที่ส่วนที่เหลือเป็น ListView โดยสามารถคลิกเมาส์ค้างที่ ListView เพื่อเลื่อนดูข้อมูล (ชื่อประเทศ) ที่แสดงออกมาไม่ครบ

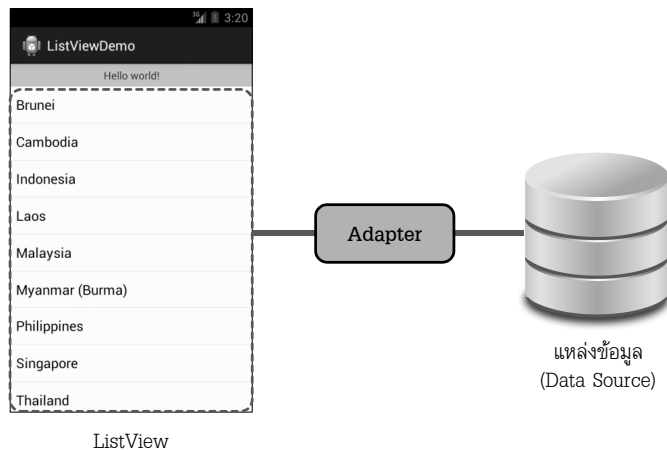


เมื่อคลิกชื่อประเทศจะแสดงข้อความใน TextView ที่บอกชื่อประเทศที่ถูกคลิก รวมถึงตำแหน่งของชื่อประเทศนั้นใน ListView (ชื่อแรกคือตำแหน่งที่ 0)



## คำอธิบาย

การแสดงผลข้อมูลใน ListView จะใช้ Adapter เป็นตัวกลางระหว่าง ListView กับแหล่งข้อมูล กล่าวคือ Adapter จะทำหน้าที่อ่านข้อมูลจากแหล่งข้อมูล แล้วป้อนข้อมูลนั้นให้แก่ ListView เพื่อแสดงผลออกมา ซึ่งแหล่งข้อมูลอาจเป็นอาร์เรย์ในหน่วยความจำ (เช่นตัวอย่างนี้), ฐานข้อมูล, ข้อมูลจากเน็ตเวิร์ก/อินเทอร์เน็ต หรืออื่นๆ



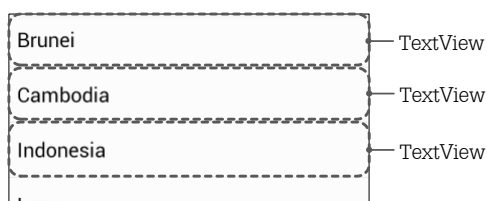
## สร้าง Adapter

ในที่นี้แหล่งข้อมูลของเราคืออาร์เรย์ ดังนั้นเราต้องสร้าง Adapter ชนิด ArrayAdapter ขึ้นมา เป็นตัวกลาง

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(
    this, android.R.layout.simple_list_item_1, countries);
```

คอนสตรัคเตอร์ของ ArrayAdapter มีพารามิเตอร์ 3 ตัว

- ◆ พารามิเตอร์ตัวที่ 1 ระบุคอนเท็กซ์
- ◆ พารามิเตอร์ตัวที่ 2 ระบุ Layout สำหรับไอเท็ม (ข้อมูล) หนึ่งๆใน ListView ซึ่งค่า android.R.layout.simple\_list\_item\_1 ที่เราระบุในที่นี้ คือการอ้างอิงไปยัง Layout ที่แอนดรอยด์เตรียมไว้ให้ ซึ่งเป็น Layout ที่มี TextView อยู่เพียงอันเดียว ดังนั้นชื่อประเทศแต่ละชื่อจะถูกแสดงอยู่ภายใน TextView ดังรูป



- ◆ พารามิเตอร์ตัวที่ 3 ระบุอาร์เรย์ที่จะแสดงข้อมูลออกมาใน ListView

## กำหนด Adapter ให้กับ ListView

หลังจากสร้าง Adapter เรียบร้อยแล้ว เราจะนำ Adapter นั้นมากำหนดให้กับ ListView โดยใช้เมธอด `setAdapter`

```
list.setAdapter(adapter);
```

## ระบุการทำงานเมื่อไอเท็มใน ListView ถูกคลิก

ขั้นตอนนี้จะต้อง Implement อินเทอร์เฟซ `AdapterView.OnItemClickListener` และระบุการทำงานที่ต้องการในเมธอด `onItemClick` จากนั้นจึงนำมากำหนดให้กับ ListView โดยใช้เมธอด `setOnItemClickListener`

```
list.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> av, View v, int position,
        long id) {
        String msg = "You have selected " + ((TextView) v).getText();
        msg += " at position " + String.valueOf(position);
```

```
        text.setText(msg);  
    }  
});
```

เมื่อผู้ใช้คลิกไอเท็มใน ListView แอนดรอยด์จะเรียกมายังเมธอด `onItemClick` พร้อมทั้งส่งผ่านค่ามายังพารามิเตอร์ทั้งสี่ ซึ่งพารามิเตอร์ตัวที่ 2 จะระบุถึงวิวของไอเท็มที่ถูกคลิก (วิวที่ใช้แสดงแต่ละข้อมูลใน ListView ซึ่งในที่นี้คือ TextView) และพารามิเตอร์ตัวที่ 3 จะบอกตำแหน่งของไอเท็มนั้นใน ListView (ไอเท็มแรกคือตำแหน่งที่ 0)

## การแสดงรายการข้อมูลด้วย ListView

โดยมากเมื่อนักพัฒนาต้องการแสดงรายการข้อมูล ก็มักจะสร้างหน้าจอที่มี ListView เพียงอันเดียวและแสดงผลเต็มจอ ด้วยเหตุนี้แอนดรอยด์จึงได้เตรียมแอกทिवิตีแบบพิเศษไว้ให้ นั่นคือ `ListActivity` ซึ่งเป็นซับคลาสของ `Activity` อีกทีหนึ่ง แอคทिवิตีชนิดนี้จะมี ListView มาให้ในตัวเลยช่วยให้เราแสดงรายการข้อมูลและจัดการอีเวนต์ได้สะดวกยิ่งขึ้น

### ตัวอย่าง

ตัวอย่างนี้จะทำงานคล้ายตัวอย่างที่แล้ว แต่ในหน้าจอจะมี ListView เพียงอย่างเดียว (ไม่มี TextView) ซึ่งเมื่อคลิกไอเท็มใน ListView จะแสดง Toast ออกมาแทน

- 1 แก้ไขคลาส `MainActivity` ให้สืบทอดจาก `ListActivity` แทนที่จะเป็น `Activity` และเพิ่มการประกาศอาร์เรย์ `countries` ภายในคลาส

โปรเจ็ค `ListActivityDemo`, ไฟล์ `MainActivity.java`

```
public class MainActivity extends ListActivity {  
  
    static final String[] countries = new String[] {  
        "Brunei", "Cambodia", "Indonesia", "Laos", "Malaysia",  
        "Myanmar (Burma)", "Philippines", "Singapore", "Thailand",  
        "Vietnam" };  
  
    ...  
}
```

- 2 ที่เมธอด `onCreate` ให้ลบบรรทัดที่เรียกเมธอด `setContentView` และเพิ่มโค้ดดังนี้

โปรเจ็ค `ListActivityDemo`, ไฟล์ `MainActivity.java`

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    //setContentView(R.layout.activity_main);
```

```

        ArrayAdapter<String> adapter = new ArrayAdapter<String>(
            this, android.R.layout.simple_list_item_1, countries);
        setListAdapter(adapter);
    }

```

### 3 เพิ่มเมธอด `onItemClickListener` ในแอกทิวิตี้

โปรเจ็ค `ListActivityDemo`, ไฟล์ `MainActivity.java`

```

@Override
protected void onItemClickListener(ListView l, View v, int position,
                                     long id) {
    String msg = "You have selected " + ((TextView) v).getText();
    msg += " at position " + String.valueOf(position);

    Toast.makeText(this, msg, Toast.LENGTH_SHORT).show();
}

```

## ผลการรัน



## คำอธิบาย

จุดที่แตกต่างเมื่อเทียบกับตัวอย่างที่แล้วคือ

- ◆ MainActivity สืบทอดจาก ListActivity ไม่ใช่ Activity
- ◆ ในเมธอด onCreate ไม่มีการเรียกเมธอด setContentView เพื่อกำหนด Layout ของหน้าจอ เนื่องจาก ListActivity จะเตรียม Layout มาให้แล้ว และเป็น Layout ที่มี ListView เพียงอย่างเดียว
- ◆ การกำหนด Adapter ให้กับ ListView จะใช้เมธอด setListAdapter ของ ListActivity แทนการใช้เมธอด setAdapter ของ ListView
- ◆ การระบุการทำงานเมื่อไอเท็มใน ListView ถูกคลิก จะทำได้โดย Override เมธอด OnItemClickListener ของ ListActivity แทนการ Implement อินเทอร์เฟซ AdapterView.OnItemClickListener และระบุการทำงานในเมธอด onItemClick

## ต้องการใช้ ListActivity และกำหนด Layout เองด้วย

ถ้าหากคุณต้องการใช้งาน ListActivity และขณะเดียวกันก็ต้องการกำหนด Layout เอง ก็สามารถทำได้เช่นกัน แต่มีเงื่อนไขคือ จะต้องสร้าง ListView ที่มี ID เป็น @android:id/list ไว้ภายใน Layout นั้น เช่น ถ้าจะออกแบบหน้าจอเหมือนตัวอย่างที่แล้ว (มี TextView อยู่ด้านบน) ให้กำหนด Layout ดังนี้

โปรเจ็ค ListActivityDemo, ไฟล์ res/layout/activity\_main.xml

```
<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:id="@+id/text"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#66ff99"
        android:gravity="center"
        android:padding="5dp"
        android:text="@string/hello_world" />

    <ListView
        android:id="@android:id/list"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```



จากนั้นในเมธอด onCreate ให้เรียก setContentView เพื่อกำหนด Layout ให้กับแอคทิวิตีตามปกติ แต่เมื่อต้องการเข้าถึง ListView ให้ใช้เมธอด getListView แทนการใช้เมธอด findViewById (ดูตัวอย่างในหัวข้อถัดไป)

สำหรับการผูก Adapter และระบบการทำงานเมื่อไอเท็มถูกคลิคนั้นสามารถใช้ตามตัวอย่างนี้ต่อไปได้ (ไม่ต้องแก้ไขกลับไปเหมือนตัวอย่างที่แล้ว) ซึ่งจะมีผลกับ ListView ที่มี ID เป็น @android:id/list โดยอัตโนมัติ

## การแสดงรายการข้อมูลที่เป็นตัวเลือกแบบ Multiple Choice

เราสามารถใช้ ListView ในการแสดงรายการตัวเลือกแบบ Multiple Choice ได้ ซึ่งหมายถึงรายการตัวเลือกที่อนุญาตให้ผู้เลือกได้หลายตัวเลือกพร้อมกัน ดังตัวอย่างต่อไปนี้

### ตัวอย่างและคำอธิบาย

ตัวอย่างนี้จะดัดแปลงจากตัวอย่างที่แล้ว โดยสมมติว่าต้องการแสดงหน้าจอให้ผู้เลือกชื่อประเทศต่างๆ ได้มากกว่า 1 ประเทศ ดังรูป ซึ่งแต่ละครั้งที่คลิกเลือก (select) หรือคลิกยกเลิก (deselect) ก็จะมี Toast ออกมา ตรงนี้จะทำให้คุณเห็นวิธีการเขียนโค้ดเพื่อตรวจสอบสถานะของตัวเลือกขณะเกิดอีเวนต์



- 1 แก้ไขคลาส MainActivity ให้สืบทอดจาก ListActivity แทนที่จะเป็น Activity และเพิ่มการประกาศอาร์เรย์ countries ภายในคลาส

โปรเจ็ค ListMultipleChoiceDemo, ไฟล์ MainActivity.java

```
public class MainActivity extends ListActivity {  
  
    static final String[] countries = new String[] {  
        "Brunei", "Cambodia", "Indonesia", "Laos", "Malaysia",  
        "Myanmar (Burma)", "Philippines", "Singapore", "Thailand",  
        "Vietnam" };  
  
    ...  
}
```

## 2 ที่เมธอด onCreate ให้ลบบรรทัดที่เรียกเมธอด setContentView และเพิ่มโค้ดดังนี้

โปรเจ็ค ListMultipleChoiceDemo, ไฟล์ MainActivity.java

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    //setContentView(R.layout.activity_main);

    ListView list = getListView(); ❶
    list.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE); ❷

    ArrayAdapter<String> adapter = new ArrayAdapter<String>(
        this, android.R.layout.simple_list_item_checked, countries); ❸
    setListAdapter(adapter); ❹
}
```

- ❶ ใช้เมธอด getListView เข้าถึง ListView ที่ ListActivity เตรียมมาให้
- ❷ ใช้เมธอด setChoiceMode กำหนดรูปแบบการเลือกของ ListView เป็นแบบ CHOICE\_MODE\_MULTIPLE คือเป็นตัวเลือกที่สามารถเลือกได้หลายตัวเลือกพร้อมกัน
- ❸ สร้าง ArrayAdapter โดยกำหนด Layout ของแต่ละไอเท็มใน ListView เป็นแบบ simple\_list\_item\_checked ซึ่งก็คือ CheckedTextView ที่มีข้อความอยู่ทางซ้าย และเครื่องหมายเช็คถูกอยู่ทางขวา
- ❹ กำหนด ArrayAdapter ให้กับ ListView

## 3 เพิ่มเมธอด onItemClick ในแอคทิวิตี เพื่อระบุการทำงานเมื่อผู้ใช้คลิกไอเท็มใน ListView

โปรเจ็ค ListMultipleChoiceDemo, ไฟล์ MainActivity.java

```
@Override
protected void onItemClick(ListView l, View v, int position, long id) {
    String msg;
    CheckedTextView check = (CheckedTextView) v; ❶

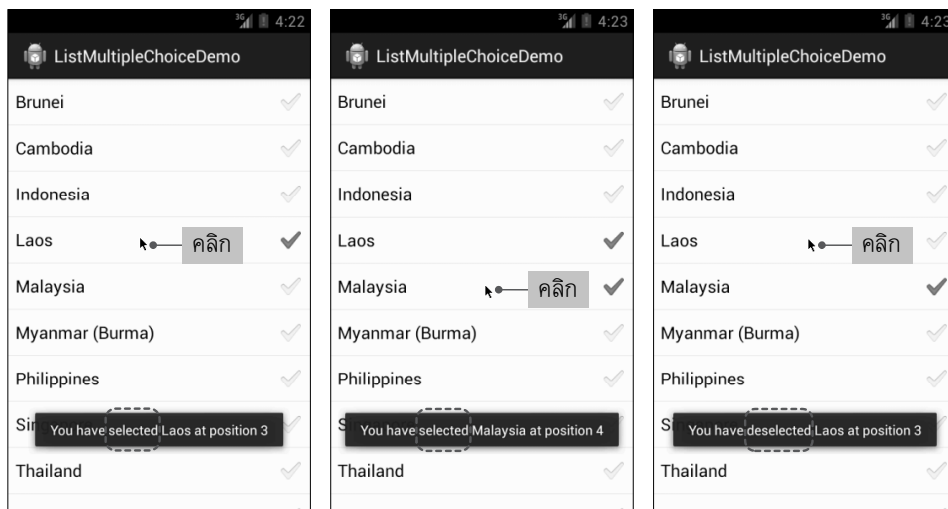
    if (check.isChecked()) { ❷
        msg = "You have selected " + check.getText();
    }
    else {
        msg = "You have deselected " + check.getText();
    }
    msg += " at position " + String.valueOf(position);

    Toast.makeText(this, msg, Toast.LENGTH_SHORT).show(); ❸
}
```

เราทราบแล้วว่า วิวของไอเท็มที่ถูกคลิกจะส่งผ่านมายังพารามิเตอร์ตัวที่สอง (v) ซึ่งในตัวอย่างนี้วิวของแต่ละไอเท็มคือ `CheckedTextView` ก่อนอื่นเราจึงแปลง (cast) ค่าของ v เป็นชนิดวิวดังกล่าว

- 1 จากนั้นตรวจสอบสถานะการเลือกของมันโดยใช้เมธอด `isChecked`
- 2 เพื่อสร้างข้อความเก็บไว้ในตัวแปร msg ให้เหมาะสม แล้วแสดงข้อความนั้นออกมาใน Toast
- 3

## ผลการรัน



## การตรวจสอบสถานะของตัวเลือกทั้งหมดใน ListView

ในตัวอย่างที่แล้ว เราตรวจสอบสถานะของตัวเลือกที่ทำให้เกิดอีเวนต์เท่านั้น แต่บางครั้งเราอาจปล่อยให้ผู้ใช้เลือกไปก่อน แล้วค่อยตรวจสอบสถานะของตัวเลือกทั้งหมดทีเดียว เช่น หลังจากผู้ใช้คลิกปุ่ม OK เป็นต้น

### ตัวอย่างและคำอธิบาย

เราจะปรับปรุงตัวอย่างที่แล้ว โดยเพิ่มปุ่มไว้ด้านบน ซึ่งเมื่อคลิกปุ่มนี้จะตรวจสอบสถานะของตัวเลือกทั้งหมดใน `ListView` แล้วแสดง Toast ออกมา

#### 1 กำหนด Layout ของหน้าจอ

โปรเจ็ค `ListMultipleChoiceDemo`, ไฟล์ `res\layout\activity__main.xml`

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

```

        android:orientation="vertical" >

        <ListView
            android:id="@android:id/list"
            android:layout_width="match_parent"
            android:layout_height="0dp"
            android:layout_weight="1" />

        <Button
            android:id="@+id/button"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Show selected items" />

    </LinearLayout>

```

## 2 เพิ่มโค้ดในเมธอด onCreate

โปรเจ็ค ListMultipleChoiceDemo, ไฟล์ MainActivity.java

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main); ❶

    ListView list = getListView();
    list.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);

    ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
        android.R.layout.simple_list_item_checked, countries);
    setListAdapter(adapter);

    Button button = (Button) findViewById(R.id.button);
    // ระบุการทำงานเมื่อปุ่มถูกคลิก
    button.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            ListView list = getListView();
            String msg = "Selected items:";

            for (int i = 0; i < list.getCount(); i++) {
                if (list.isItemChecked(i)) { ❷
                    msg += "\n" + list.getItemAtPosition(i); ❸
                }
            }
        }
    });
}

```

```

        Toast.makeText(MainActivity.this, msg, Toast.LENGTH_LONG)
            .show(); ❹
    }
    });
}

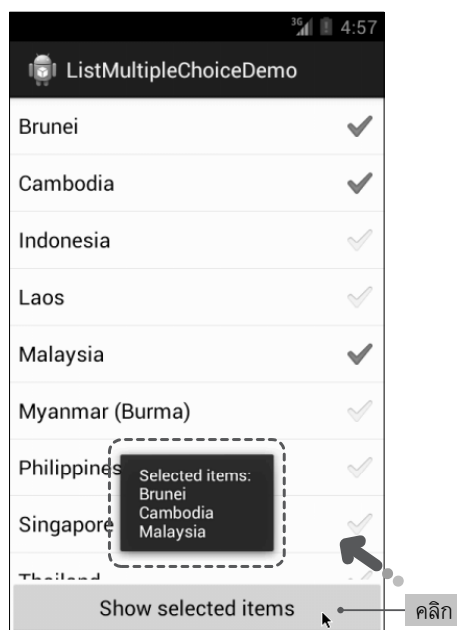
```

เราเรียกเมธอด `setContentView` เพื่อกำหนด Layout ของหน้าจอ ❶ เพราะก่อนหน้านี้เราไม่ได้กำหนด Layout เอง แต่ใช้ Layout ที่ `ListActivity` เตรียมไว้ให้ ซึ่งทั้งหน้าจอก็จะมี `ListView` เพียงอย่างเดียว

โค้ดส่วนที่เหลือที่เพิ่มเข้ามาคือการระบุการทำงานเมื่อปุ่มถูกคลิก โดยใช้เมธอด `isChecked` ตรวจสอบสถานะการเลือกของแต่ละไอเท็มใน `ListView` ❷ ถ้าไอเท็มใดมีสถานะเป็น `true` (ถูกเลือก) ก็จะใช้เมธอด `getItemAtPosition` อ่านข้อมูลของไอเท็มนั้นมาเก็บลงตัวแปร `msg` ❸ เพื่อแสดงเป็น Toast ออกมาในตอนท้าย ❹

## ผลการรัน

ลองคลิกเลือกชื่อประเทศต่างๆ เสร็จแล้ว  
คลิกปุ่ม Show selected items ข้างล่าง



## การสร้าง ListView ที่เราออกแบบ Layout ของแต่ละข้อมูลเอง

ที่ผ่านมาเรากำหนดรูปแบบการแสดงผลของไอเท็มใน ListView โดยใช้ Layout ที่แอนดรอยด์เตรียมมาให้ ได้แก่ `android.R.layout.simple_list_item_1` (TextView อันเดียว) และ `android.R.layout.simple_list_item_checked` (CheckedTextView อันเดียว) แต่นอกจากนี้ แอนดรอยด์ยังอนุญาตให้กำหนด Layout โดยใช้ Layout File ที่เราสร้างขึ้นเองได้ด้วย หรือเรียกว่า Custom ListView

### ตัวอย่างและคำอธิบาย

ตัวอย่างนี้จะแสดงชื่อประเทศใน ListView โดยมีรูปสัญลักษณ์ของอาเซียนอยู่หน้าชื่อทุกประเทศ

#### 1 กำหนด Layout สำหรับแต่ละไอเท็มใน ListView (ไฟล์ item.xml)

โปรเจ็ค CustomListViewDemo, ไฟล์ `res\layout\item.xml`

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_vertical" >

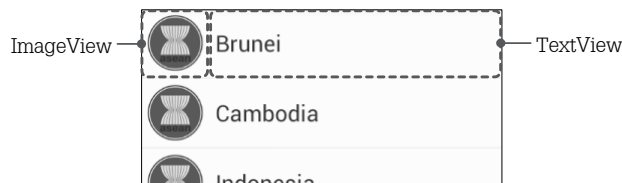
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:src="@drawable/asean_logo" />

    <TextView
        android:id="@+id/country_name"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="5dp"
        android:layout_weight="1"
        android:textSize="20sp" />

</LinearLayout>
```



asean\_logo.png



- 2 แก้ไขคลาส MainActivity ให้สืบทอดจาก ListActivity แทนที่จะเป็น Activity และเพิ่มการประกาศอาร์เรย์ countries ภายในคลาส

โปรเจ็ค CustomListViewDemo, ไฟล์ MainActivity.java

```
public class MainActivity extends ListActivity {

    static final String[] countries = new String[] {
        "Brunei", "Cambodia", "Indonesia", "Laos", "Malaysia",
        "Myanmar (Burma)", "Philippines", "Singapore", "Thailand",
        "Vietnam" };

    ...
}
```

- 3 ที่เมธอด onCreate ให้ลบบรรทัดที่เรียกเมธอด setContentView และเพิ่มโค้ดดังนี้

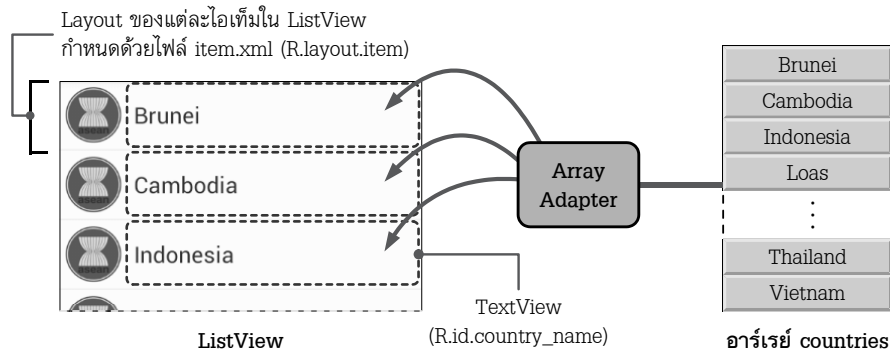
โปรเจ็ค CustomListViewDemo, ไฟล์ MainActivity.java

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    //setContentView(R.layout.activity_main);

    ArrayAdapter<String> adapter = new ArrayAdapter<String>(
        this, R.layout.item, R.id.country_name, countries);
    setListAdapter(adapter);
}
```

เมื่อต้องการกำหนด Layout ของแต่ละไอเท็มเอง จะต้องสร้าง ArrayAdapter โดยระบุพารามิเตอร์ 4 ตัว ซึ่งพารามิเตอร์ตัวที่สองจะระบุถึง Layout File (ไฟล์ item.xml) และพารามิเตอร์ตัวที่สามจะระบุถึงวิวใน Layout File นั้น ที่ใช้แสดงไอเท็มหนึ่งๆ

ArrayAdapter จะอ่านข้อมูลจากอาร์เรย์ countries แล้วใส่ข้อมูลลงใน TextView (R.id.country\_name) ภายใน Layout File ที่กำหนด (R.layout.item) ดังรูปหน้าถัดไป



### NOTE >>>

ความจริง ArrayAdapter จะต้อง Inflate Layout สำหรับแต่ละไอเท็มขึ้นมาจากไฟล์ item.xml ด้วย ก่อนที่จะใส่ข้อมูลลงใน TextView ที่เป็นส่วนหนึ่งของ Layout นั้น แล้วส่ง Layout ให้ ListView นำไปแสดงผล ซึ่งในหัวข้อถัดไปคุณจะได้เห็นการทำงานที่ว่านี้ เพราะเราจะสร้าง Adapter ในแบบของเราขึ้นมาเอง (Custom Adapter)

#### 4 เพิ่มเมธอด onItemClick ในแอคทิวิตี้ เพื่อระบุการทำงานเมื่อผู้ใช้คลิกไอเท็มใน ListView

โปรเจ็ค CustomListViewDemo, ไฟล์ MainActivity.java

```
@Override
protected void onItemClick(ListView l, View v, int position, long id) {
    TextView tv = (TextView) v.findViewById(R.id.country_name); ❶

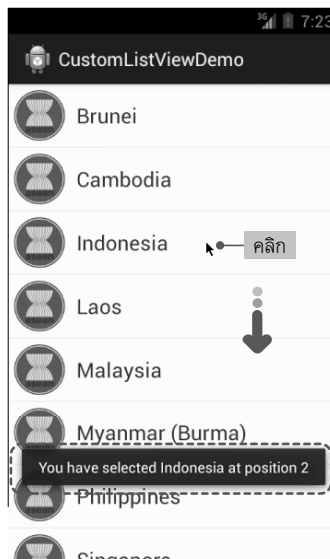
    String msg = "You have selected " + tv.getText(); ❷
    msg += " at position " + String.valueOf(position);

    Toast.makeText(this, msg, Toast.LENGTH_SHORT).show();
}
```

วิธีเขียนโค้ดในเมธอด onItemClick จะแตกต่างไปจากเดิมเล็กน้อย เนื่องจากค่าที่ส่งผ่านมายังพารามิเตอร์ v ไม่ใช่ TextView หรือ CheckedTextView แต่เป็น LinearLayout ที่บรรจุ ImageView กับ TextView ไว้ตามที่เรากำหนดในไฟล์ item.xml ดังนั้นจึงต้องเรียกใช้เมธอด findViewById บนพารามิเตอร์ v นี้เพื่อเข้าถึง TextView ให้ได้ก่อน ❶ แล้วค่อยใช้เมธอด getText อ่านชื่อประเทศมาอีกที ❷



## ผลการรับ



## การสร้าง Custom Adapter เพื่อใช้งานกับ ListView

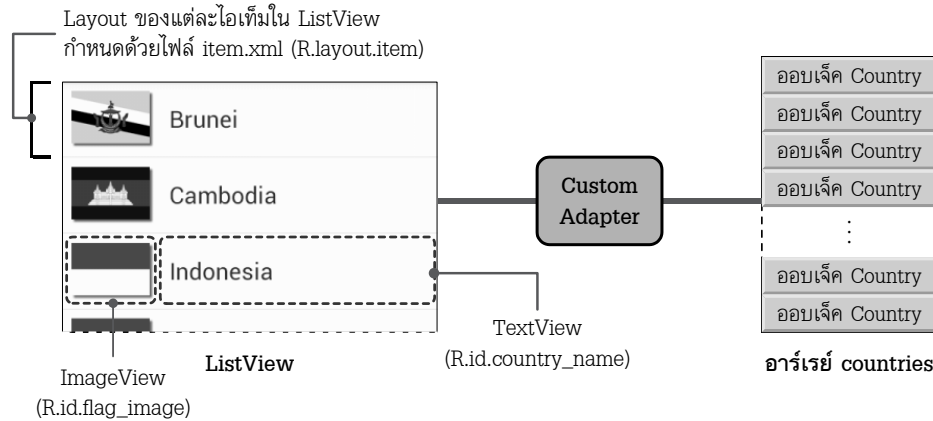
จากตัวอย่างที่ผ่านมา มีคำถามว่าถ้าหากต้องการแสดงรูปธงชาติของแต่ละประเทศแทนรูปสัญลักษณ์อาเซียนจะทำได้หรือไม่?

**ArrayAdapter** มีข้อจำกัดตรงที่มันจะอ่านข้อมูลจากอาร์เรย์แล้วใส่ข้อมูลลงในวิวหนึ่งๆภายใน Layout เท่านั้น ดังนั้นถ้าหากมีข้อมูลมากกว่า 1 ชิ้นที่จะใส่ลงในวิวมากกว่า 1 วิวสำหรับแต่ละไอเท็ม เช่น การแสดงทั้งชื่อประเทศและรูปธงชาติของแต่ละประเทศดังที่กล่าวข้างต้น กรณีอย่างนี้ **ArrayAdapter** จะทำไม่ได้ ทางออกคือเราต้องสร้าง Adapter ในแบบของเราขึ้นมาเอง (Custom Adapter)

### ตัวอย่างและคำอธิบาย

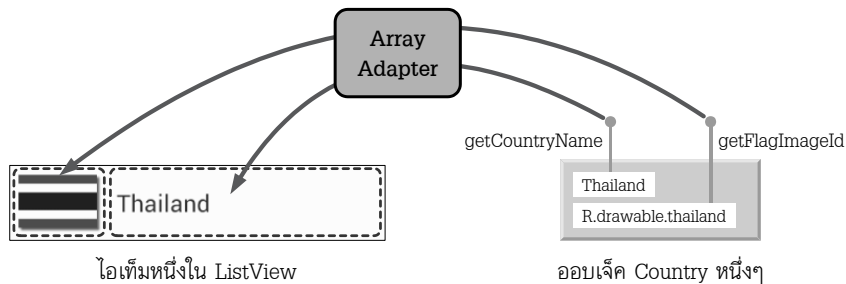
ตัวอย่างนี้จะสร้าง Custom Adapter โดยสืบทอดจาก **ArrayAdapter** เพื่อที่เราจะดัดแปลงการทำงานของมันให้มีการอ่านชื่อประเทศและรูปธงชาติจากอาร์เรย์ แล้วนำมากำหนดลงใน **TextView** และ **ImageView** ภายใน Layout ของแต่ละไอเท็ม ตามลำดับ

ก่อนอื่นขอให้ดูภาพอธิบายการทำงานของตัวอย่างนี้ ซึ่งจะช่วยให้คุณเข้าใจได้ดียิ่งขึ้น



ออบเจ็กต์ Country จะเก็บชื่อและ ID ของรูปธงชาติของประเทศหนึ่งๆไว้ ซึ่งได้ภายนอกสามารถอ่านข้อมูลทั้งสองจากออบเจ็กต์มาใช้งานได้โดยเรียกเมธอด `getCountryName` และ `getFlagImageId` ตามลำดับ

Custom Adapter จะ Inflate Layout ของแต่ละไอเท็มขึ้นมาจากไฟล์ item.xml แล้วอ่านข้อมูลทั้งสองจากออบเจ็กต์ Country หนึ่งๆมากำหนดลงใน TextView (R.id.country\_name) และ ImageView (R.id.flag\_image) ภายใน Layout นั้น ดังรูป



# 1 กำหนด Layout สำหรับแต่ละไอเท็มใน ListView (ไฟล์ item.xml)

โปรเจ็กต์ CustomAdapterDemo, ไฟล์ res/layout/item.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_vertical" >

    <ImageView
        android:id="@+id/flag_image"
        android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:layout_margin="8dp" />

<TextView
    android:id="@+id/country_name"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="5dp"
    android:layout_weight="1"
    android:textSize="20sp" />

</LinearLayout>

```

## 2 สร้างคลาสใหม่ชื่อ Country (คลิกขวาที่ชื่อโปรเจ็ค ► New ► Class) แล้วพิมพ์โค้ดดังนี้

โปรเจ็ค CustomAdapterDemo, ไฟล์ Country.java

```

package com.example.customadapter;

public class Country {
    private int m_flagImageId;    // ID ของรูปธงชาติ
    private String m_countryName; // ชื่อประเทศ

    // คอนสตรัคเตอร์
    public Country(int flagImageId, String countryName) {
        m_flagImageId = flagImageId;
        m_countryName = countryName;
    }

    // Getter Method สำหรับ ID ของรูปธงชาติ
    public int getFlagImageId() {
        return m_flagImageId;
    }

    // Setter Method สำหรับ ID ของรูปธงชาติ
    public void setFlagImageId(int flagImageId) {
        m_flagImageId = flagImageId;
    }

    // Getter Method สำหรับชื่อประเทศ
    public String getCountryName() {
        return m_countryName;
    }

    // Setter Method สำหรับชื่อประเทศ
    public void setCountryName(String countryName) {
        m_countryName = countryName;
    }
}

```

คลาส **Country** คือต้นแบบที่ใช้สร้างออบเจ็กต์ **Country** สำหรับเก็บชื่อและ ID ของรูปธงชาติของประเทศหนึ่งๆ การสร้างออบเจ็กต์ของคลาสนี้จะต้องระบุพารามิเตอร์ให้คอนสตรัคเตอร์ 2 ตัวคือ ID ของรูปธงชาติและชื่อประเทศ ซึ่งคอนสตรัคเตอร์จะนำค่าของพารามิเตอร์ทั้งสองไปเก็บลงฟิลด์ **m\_flagImageId** และ **m\_countryName** ของคลาส ตามลำดับ ฟิลด์ทั้งสองเป็น **private** ดังนั้นภายนอกจึงเข้าถึงโดยตรงไม่ได้ แต่จะต้องเข้าถึงผ่านทาง **Getter Method** และ **Setter Method** ที่คลาสเตรียมไว้เท่านั้น

### 3 สร้างคลาสใหม่ชื่อ **CountryAdapter** แล้วพิมพ์โค้ดดังนี้

โปรเจ็ค CustomAdapterDemo, ไฟล์ CountryAdapter.java

```
package com.example.customadapter;

import android.app.Activity;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;

public class CountryAdapter extends ArrayAdapter<Country> { ❶

    private Context context;
    private int itemLayoutId;
    private Country countries[] = null;

    // คอนสตรัคเตอร์
    public CountryAdapter(Context context, int itemLayoutId,
                          Country[] countries) {
        super(context, itemLayoutId, countries);

        this.itemLayoutId = itemLayoutId;
        this.context = context;
        this.countries = countries;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        LayoutInflater inflater = ((Activity) context).getLayoutInflater();
        View item = inflater.inflate(itemLayoutId, parent, false); ❸
```

```

        ImageView imgFlag =
            (ImageView) item.findViewById(R.id.flag_image);
        TextView txtCountryName =
            (TextView) item.findViewById(R.id.country_name);
    }
    Country country = countries[position];
    imgFlag.setImageResource(country.getFlagImageId());
    txtCountryName.setText(country.getCountryName());

    return item;
}

```

คลาส **CountryAdapter** นี้คือ Custom Adapter ของเรา ซึ่งสืบทอดจากคลาส **ArrayAdapter** ของแอนดรอยด์ และระบุว่า Adapter นี้จะทำงานกับข้อมูลชนิด **Country** (ออบเจ็กต์ **Country**) ❶

เราออกแบบให้คอนสตรัคเตอร์ของ **CountryAdapter** มีพารามิเตอร์ 3 ตัวเช่นเดียวกับ **ArrayAdapter** นั่นคือ คอนเท็กซ์ของแอป, ID ที่ระบุ Layout สำหรับไอเท็มหนึ่งๆ และอาร์เรย์ที่เก็บข้อมูลทั้งหมดไว้ คอนสตรัคเตอร์จะเก็บค่าทั้งสามนี้ลงในฟิลด์แบบ **private** ของคลาส ❷

ส่วนสำคัญที่สุดในคลาส **CountryAdapter** ก็คือการ Override เมธอด **getView** เพื่อให้ทำงานในแบบที่เราต้องการ ซึ่ง **ListView** จะเรียกมายังเมธอดนี้เพื่อรับวิวที่มีข้อมูลเรียบร้อยแล้วสำหรับแต่ละไอเท็มกลับไปแสดงผล และเรียกเมธอดนี้ซ้ำๆจนครบทุกไอเท็ม โดยพารามิเตอร์ **position** จะบอกให้รู้ว่าตอนนั้น **ListView** ต้องการไอเท็มที่ตำแหน่งใด ดังนั้นเราต้องเขียนโค้ดในเมธอด **getView** ให้สอดคล้องกับการทำงานดังกล่าว

ในที่นี้เรา Inflate วิวหรือ Layout ของไอเท็มขึ้นมาจากไฟล์ **item.xml** ❸ จากนั้นเข้าถึง **ImageView** และ **TextView** ภายในวิวนั้น ❹ ก่อนจะอ่านข้อมูลจากออบเจ็กต์ **Country** หนึ่งๆในอาร์เรย์ **countries** (ตาม **position** ที่ **ListView** ระบุมา) ❺ มากำหนดให้กับวิวทั้งสอง ❻ แล้วจึง return วิวของไอเท็มกลับไป ❼

- 4 แก้ไขคลาส **MainActivity** ให้สืบทอดจาก **ListActivity** แทนที่จะเป็น **Activity** และเพิ่มโค้ดดังนี้

โปรเจ็กต์ **CustomAdapterDemo**, ไฟล์ **MainActivity.java**

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    // setContentView(R.layout.activity_main);
}

```

```
// สร้างอาร์เรย์ countries ซึ่งมีสมาชิกเป็นออบเจ็ค Country ที่เก็บข้อมูลของแต่ละประเทศ
Country countries[] = new Country[] {
    new Country(R.drawable.brunei, "Brunei"),
    new Country(R.drawable.cambodia, "Cambodia"),
    new Country(R.drawable.indonesia, "Indonesia"),
    new Country(R.drawable.laos, "Laos"),
    new Country(R.drawable.malaysia, "Malaysia"),
    new Country(R.drawable.myanmar, "Myanmar (Burma)"),
    new Country(R.drawable.philippines, "Philippines"),
    new Country(R.drawable.singapore, "Singapore"),
    new Country(R.drawable.thailand, "Thailand"),
    new Country(R.drawable.vietnam, "Vietnam"), };

// สร้าง Adapter Object และนำไปกำหนดให้ ListView
CountryAdapter adapter = new CountryAdapter(this, R.layout.item,
                                             countries);

setListAdapter(adapter);
}
```

## 5 เพิ่มเมธอด onItemClick ในแอกทิวิตี้ เพื่อระบุการทำงานเมื่อผู้ใช้คลิกไอเท็มใน ListView

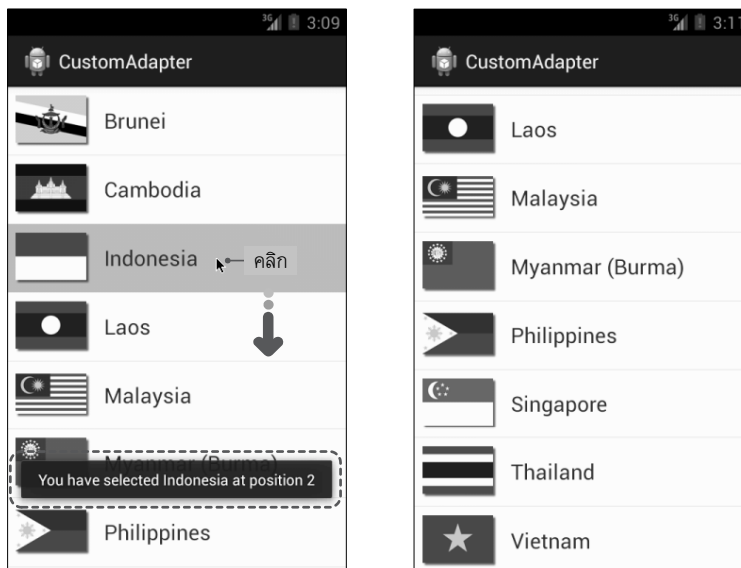
โปรเจ็ค CustomAdapterDemo, ไฟล์ MainActivity.java

```
@Override
protected void onListItemClick(ListView l, View v, int position, long id) {
    TextView tv = (TextView) v.findViewById(R.id.country_name);

    String msg = "You have selected " + tv.getText();
    msg += " at position " + String.valueOf(position);

    Toast.makeText(this, msg, Toast.LENGTH_SHORT).show();
}
```

## ผลการรับ



## ปรับปรุงประสิทธิภาพของ ListView

ตัวอย่างที่ผ่านมาถึงแม้จะทำงานได้สมบูรณ์ แต่ถ้าหาก ListView มีไอเท็มอยู่จำนวนมากจะทำให้สิ้นเปลืองทรัพยากรของระบบและส่งผลกระทบต่อประสิทธิภาพของแอป เนื่องจากจะมีการ Inflate วิวขึ้นมาจากไฟล์ item.xml สำหรับทุกไอเท็มใน ListView เช่นในที่นี้มี 10 ไอเท็ม (10 ประเทศ) ก็จะมี Inflate ขึ้นมา 10 วิว ซึ่งหากมี 100 ไอเท็มก็จะ Inflate 100 วิว ทั้งที่มีไอเท็มแค่จำนวนหนึ่งเท่านั้นที่ถูกแสดงออกมาในขณะหนึ่งๆ (ในตัวอย่างนี้คือ 7 ไอเท็ม)

เพื่อแก้ปัญหานี้ แอนดรอยด์จึงมีวิธีให้เรานำวิวของไอเท็มที่เคยสร้างไว้แล้วกลับมาใช้ซ้ำได้ ในกรณีที่มีไอเท็มนั้นไม่ถูกแสดงออกมา ให้คุณแก้ไขเมธอด `getView` ในคลาส `CountryAdapter` รวมทั้งเพิ่มคลาส `CountryHolder` ไว้ภายในคลาส `CountryAdapter` ดังนี้

โปรเจ็ค CustomAdapterDemo, ไฟล์ CountryAdapter.java

```
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    View item = convertView;
    CountryHolder holder = null;

    if (item == null) { ❶
        LayoutInflater inflater = ((Activity) context).getLayoutInflater();
        item = inflater.inflate(itemLayoutId, parent, false); ❷
    }
```

```

        holder = new CountryHolder(); ❸
        holder.imgFlag =
            (ImageView) item.findViewById(R.id.flag_image);
        holder.txtCountryName =
            (TextView) item.findViewById(R.id.country_name); ❹

        item.setTag(holder); ❺
    } else {
        holder = (CountryHolder) item.getTag(); ❻
    }

    Country country = countries[position];
    holder.imgFlag.setImageResource(country.getFlagImageId());
    holder.txtCountryName.setText(country.getCountryName());

    return item;
}

// Inner Class สำหรับการอ้างอิงของ ImageView และ TextView
static class CountryHolder {
    ImageView imgFlag;
    TextView txtCountryName;
}

```

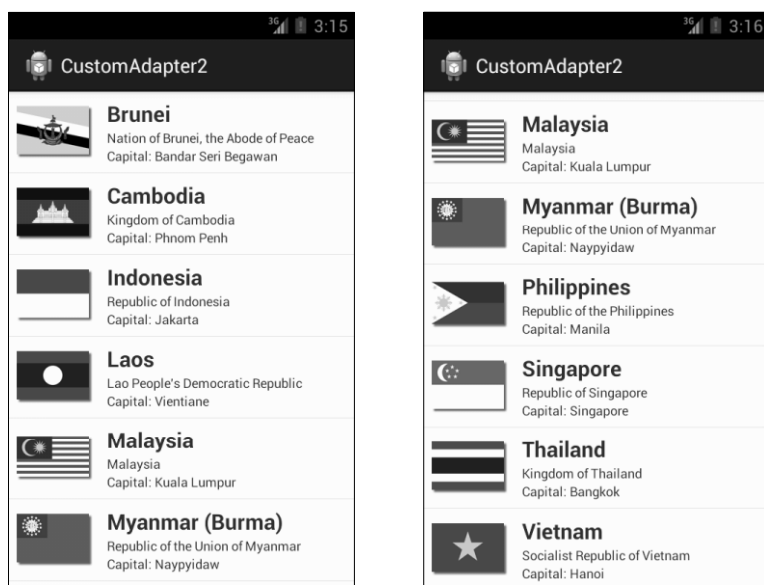
เมื่อ `ListView` เรียกมายังเมธอด `getView` มันจะส่งวิวของไอเท็มที่เราสามารถใช้ซ้ำได้มาทางพารามิเตอร์ `convertView` ดังนั้นก่อนอื่นเราจะตรวจสอบพารามิเตอร์นี้ ซึ่งถ้าเป็น `null` (ไม่มีวิวที่ใช้ซ้ำได้) ❶ เราจะ Inflate วิวจาก Layout File ตามปกติ ❷ แล้วสร้าง Holder object (ออบเจ็ค `CountryHolder`) ❸ เพื่ออ้างอิง `ImageView` และ `TextView` ภายในวิวที่เพิ่ง Inflate ขึ้นมา ❹ แล้วแนบ Holder object ไปกับวิวนั้นโดยใช้เมธอด `setTag` ❺

แต่ถ้าพารามิเตอร์ `convertView` ไม่เป็น `null` (มีวิวที่ใช้ซ้ำได้) เราจะใช้เมธอด `getTag` เข้าถึง Holder object ที่แนบมากับวิว ❻ เพื่อกำหนดข้อมูลลงใน `ImageView` และ `TextView` ที่ Holder object นั้นเก็บไว้ แทนที่จะ Inflate ขึ้นมาใหม่จาก Layout File

## Custom Adapter กับ Layout ที่ซับซ้อนยิ่งขึ้น

ด้วยหลักการที่อธิบายไปแล้ว การสร้าง `ListView` ที่มีความซับซ้อนดังรูปต่อไปนี้ก็ไม่ใช่เรื่องยากอีกต่อไป สิ่งที่คุณต้องทำเพิ่มโดยหลักๆคือ ออกแบบ Layout ของไอเท็ม (ไฟล์ `item.xml`) ให้มีวิวสำหรับแสดงข้อมูลที่เพิ่มเข้ามา (ในรูปเพิ่มชื่อที่เป็นทางการ กับชื่อเมืองหลวงของประเทศ), เพิ่มฟิลด์เก็บข้อมูลรวมถึง Setter/Getter Method ในคลาส `Country` และเพิ่มโค้ดใน `CountryAdapter` ที่จะอ่านข้อมูลจากออบเจ็ค `Country` มาใส่ลงในวิวที่เพิ่มมาใน Layout





ขอฝากเป็นการบ้านให้ผู้อ่านลองทำดูครับ สำหรับโค้ดที่สมบูรณ์ของตัวอย่างตามรูปข้างบนนี้อยู่ในโปรเจ็ค CustomAdapterDemo2 ในซอร์สโค้ด

## การแสดงรายการข้อมูลด้วย Spinner

Spinner คือวิวที่เอาไว้แสดงรายการข้อมูลเช่นเดียวกับ ListView แต่จะใช้พื้นที่หน้าจอน้อยกว่า โดยในสถานะปกติจะแสดงเฉพาะไอเท็มที่ถูกเลือก แต่หากเราคลิกที่ตัว Spinner ก็แสดงไอเท็มทั้งหมดออกมา (ถ้าแสดงไม่หมดก็สามารถเลื่อนดูได้เช่นเดียวกับ ListView) ลักษณะของมันจะเหมือนกับ Drop-down Listbox ที่พบใน Windows และระบบปฏิบัติการต่างๆบนพีซี



## ตัวอย่างและคำอธิบาย

ถ้าหากมีข้อมูลอยู่ในอาร์เรย์ การแสดงข้อมูลเหล่านั้นออกมาใน Spinner จะใช้ ArrayAdapter เป็นตัวกลางเช่นเดียวกับการแสดงข้อมูลใน ListView ดังตัวอย่างนี้

### 1 กำหนด Layout ของหน้าจอ

โปรเจ็ค SpinnerDemo, ไฟล์ res/layout/activity\_\_main.xml

```
<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="16dp" >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp" >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Select country: " />

        <Spinner
            android:id="@+id/list_of_countries"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1" />

    </LinearLayout>

    <Button
        android:id="@+id/show_selected_item_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginBottom="8dp"
        android:text="Show selected item" />

    <TextView
        android:id="@+id/text"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#66ff99"
        android:gravity="center"
```

```

        android:padding="5dp"
        android:text="@string/hello_world" />
</LinearLayout>

```

เราใช้ TextView (ID: text) เป็นตัวแสดงข้อความที่บอกให้รู้ว่าไอ้เพิ่มใดถูกเลือก โดยข้อความนี้จะมาจาก 2 อีเวนต์ คือ ตอนคลิกเลือกไอ้เพิ่มใน Spinner กับตอนคลิกปุ่ม (ซึ่งจะตรวจสอบว่าตอนนั้นไอ้เพิ่มใดใน Spinner ถูกเลือก แล้วแสดงข้อความออกมา)

## 2 เพิ่มการประกาศอาร์เรย์ countries ในแอกทิวิตี

โปรเจ็ค SpinnerDemo, ไฟล์ MainActivity.java

```

static final String[] countries = new String[] {
    "Brunei", "Cambodia", "Indonesia", "Laos", "Malaysia",
    "Myanmar (Burma)", "Philippines", "Singapore", "Thailand",
    "Vietnam" };

```

## 3 เพิ่มโค้ดในเมธอด onCreate ของแอกทิวิตี เพื่อสร้าง Adapter และผูกเข้ากับ Spinner รวมทั้งระบุการทำงานเมื่อไอ้เพิ่มใน Spinner ถูกเลือก

โปรเจ็ค SpinnerDemo, ไฟล์ MainActivity.java

```

final TextView text = (TextView) findViewById(R.id.text);
final Spinner spinner = (Spinner) findViewById(R.id.list_of_countries);

// สร้าง Adapter และนำไปกำหนดให้กับ Spinner
ArrayAdapter<String> adapter = new ArrayAdapter<String>(
    this, android.R.layout.simple_spinner_item, countries);
spinner.setAdapter(adapter);

// ระบุการทำงานเมื่อไอ้เพิ่มใน Spinner ถูกเลือก
spinner.setOnItemSelectedListener(
    new AdapterView.OnItemSelectedListener() {
        // เมธอดที่จะถูกเรียกเมื่อไอ้เพิ่มหนึ่งๆถูกเลือก
        @Override
        public void onItemSelected(AdapterView<?> av, View v, int position,
            long id) {
            String msg = "You have selected " + ((TextView) v).getText();
            msg += " at position " + String.valueOf(position);

            text.setText(msg);
        }

        // เมธอดที่จะถูกเรียกเมื่อไม่มีไอ้เพิ่มถูกเลือกเลย ในที่นี้เรา return กลับไปที่เดิมโดยไม่ทำอะไร
        @Override
        public void onNothingSelected(AdapterView<?> av) {

```

```
        return;  
    }  
});
```

ให้สังเกตว่าเรากำหนด Layout ของแต่ละไอเท็มใน Spinner โดยใช้วิวที่แอนดรอยด์เตรียมมาให้ อยู่แล้ว (android.R.layout.simple\_spinner\_item)

#### 4 เพิ่มโค้ดในเมธอด onCreate ของแอกทิวิตี เพื่อระบุการทำงานเมื่อปุ่มถูกคลิก

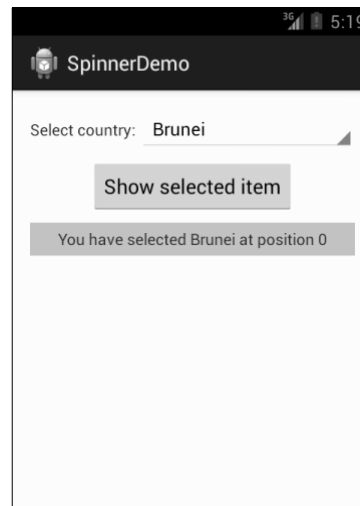
โปรเจ็ค SpinnerDemo, ไฟล์ MainActivity.java

```
Button button = (Button) findViewById(R.id.show_selected_item_button);  
button.setOnClickListener(new View.OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        String msg = "Selected item: ";  
        // ใช้เมธอด getSelectedItem อ่านข้อมูลของไอเท็มที่ถูกเลือก  
        msg += (String) spinner.getSelectedItem();  
        text.setText(msg);  
    }  
});
```

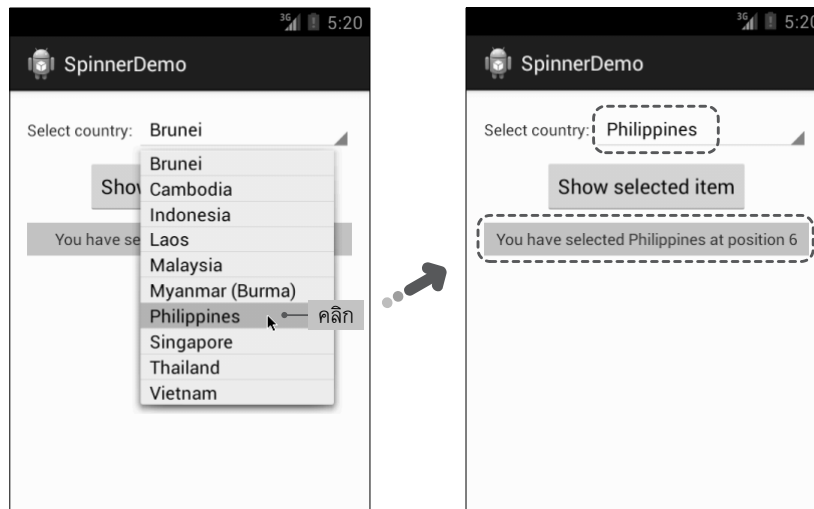
โค้ดในตัวอย่างนั้นนับว่าตรงไปตรงมา และใกล้เคียงกับตัวอย่าง ListView ก่อนหน้านี้ จึงไม่จำเป็นต้องอธิบายอะไรเพิ่มเติม เรามาดูผลการรันเลยดีกว่า

## ผลการรัน

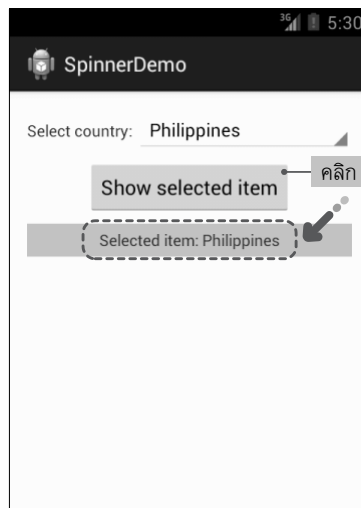
- 1 เมื่อรันขึ้นมา ไอเท็มแรกของ Spinner จะถูกเลือกอัตโนมัติ ทำให้ปรากฏข้อความใน TextView ดังแต่ต้น



## 2 ต่อไปจะลองเลือก Philippines



## 3 คลิกปุ่ม Show selected item



## การสร้าง Spinner ที่เราออกแบบ Layout ของแต่ละข้อมูลเอง

เช่นเคย แอนดรอยด์อนุญาตให้กำหนด Layout ให้กับแต่ละไอเท็มใน Spinner โดยใช้ Layout File ที่เราออกแบบเองได้ หรือเรียกว่า Custom Spinner นั้นเอง

### ตัวอย่างและคำอธิบาย

เราจะดัดแปลงตัวอย่างที่ผ่านมาให้แสดงรูปธงชาติของแต่ละประเทศใน Spinner ด้วย ดังนั้นจึงต้องสร้างทั้ง Custom Layout (ไฟล์ res/layout/item.xml) และ Custom Adapter (คลาส CountryAdapter) เหมือนที่ทำงานกับ ListView มาแล้วก่อนหน้านี้ (โปรเจ็ค CustomAdapterDemo)

#### 1 กำหนด Layout ของหน้าจอ

โปรเจ็ค CustomSpinnerDemo, ไฟล์ res/layout/activity\_\_main.xml

```
<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="16dp" >

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp" >

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="Select country: " />

        <Spinner
            android:id="@+id/list_of_countries"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1" />

    </LinearLayout>

    <Button
        android:id="@+id/show_selected_item_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginBottom="8dp"
        android:text="Show selected item" />
```

```

<TextView
    android:id="@+id/text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#66ff99"
    android:gravity="center"
    android:padding="5dp"
    android:text="@string/hello_world" />

</LinearLayout>

```

## 2 กำหนด Layout สำหรับแต่ละไอเท็มใน Spinner

โปรเจกต์ CustomSpinnerDemo, ไฟล์ res/layout/item.xml

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_vertical" >

    <ImageView
        android:id="@+id/flag_image"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="5dp" />

    <TextView
        android:id="@+id/country_name"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="3dp"
        android:layout_weight="1"
        android:textSize="16sp" />

</LinearLayout>

```

Layout นี้มีรูปธงชาติอยู่ทางซ้ายและชื่อประเทศอยู่ทางขวา เช่นเดียวกับตัวอย่าง ListView

## 3 สร้างคลาสใหม่ชื่อ Country แล้วพิมพ์โค้ดดังนี้

โปรเจกต์ CustomSpinnerDemo, ไฟล์ Country.java

```

package com.example.customspinnerdemo;

public class Country {
    private int m_flagImageId;
    private String m_countryName;

    public Country(int flagImageId, String countryName) {

```

```
        m_flagImageId = flagImageId;
        m_countryName = countryName;
    }

    public int getFlagImageId() {
        return m_flagImageId;
    }

    public void setFlagImageId(int flagImageId) {
        m_flagImageId = flagImageId;
    }

    public String getCountryName() {
        return m_countryName;
    }

    public void setCountryName(String countryName) {
        m_countryName = countryName;
    }
}
```

คลาส **Country** ในตัวอย่างนี้เหมือนในตัวอย่าง **ListView** ทุกประการ แตกต่างแค่บรรทัดแรกเท่านั้น

#### 4 สร้างคลาสใหม่ชื่อ **CountryAdapter** แล้วพิมพ์โค้ดดังนี้

โปรเจ็ค **CustomSpinnerDemo**, ไฟล์ **CountryAdapter.java**

```
package com.example.customspinnerdemo;

import android.app.Activity;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;

public class CountryAdapter extends ArrayAdapter<Country> {

    private Context context;
    private int itemLayoutId;
    private Country countries[] = null;

    public CountryAdapter(Context context, int itemLayoutId,
        Country[] countries) {
        super(context, itemLayoutId, countries);
    }
}
```



```
        this.context = context;
        this.itemLayoutId = itemLayoutId;
        this.countries = countries;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        LayoutInflater inflater = ((Activity) context).getLayoutInflater();
        View item = inflater.inflate(itemLayoutId, parent, false);

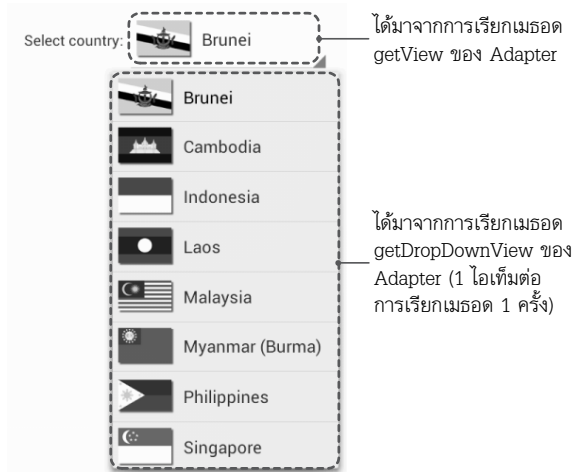
        ImageView imgFlag = (ImageView) item.findViewById(R.id.flag_image);
        TextView txtCountryName =
            (TextView) item.findViewById(R.id.country_name);

        Country country = countries[position];
        imgFlag.setImageResource(country.getFlagImageId());
        txtCountryName.setText(country.getCountryName());

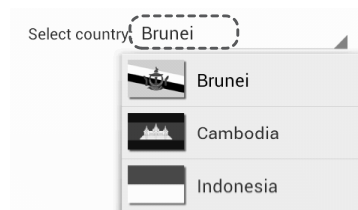
        return item;
    }

    @Override
    public View getDropDownView(int position, View convertView,
                                   ViewGroup parent) {
        return getView(position, convertView, parent);
    }
}
```

คลาส CountryAdapter ในตัวอย่างนี้ก็เหมือนตัวอย่าง ListView ทุกประการ ยกเว้นมีการ Override เมธอด `getDropDownView` เพิ่มอีก 1 เมธอด ซึ่ง Spinner จะเรียกมายังเมธอดนี้เมื่อต้องการรับวิวของแต่ละไอเท็มไปแสดงผลใน drop-down (เมื่อผู้ใช้คลิกที่ตัว Spinner เพื่อดูไอเท็มทั้งหมด) ส่วนเมธอด `getView` จะถูกเรียกเมื่อ Spinner ต้องการรับวิวไปแสดงที่ตัวมันเพื่อสื่อว่าไอเท็มใดถูกเลือกอยู่ตอนนั้น



ในที่นี้เราเขียนโค้ดในเมธอด `getDropDownView` ให้เรียกเมธอด `getView` อีกที เพื่อทำงานทุกอย่างแบบเดียวกับที่ `getView` ทำ ดังนั้นเมธอดทั้งสองจึง return วิวที่มี Layout แบบเดียวกันออกไป เราสามารถเขียนโค้ดให้แสดง Layout แบบหนึ่งในตัว Spinner แต่แสดง Layout อีกแบบหนึ่งใน drop-down ยกตัวอย่างเช่น ใน Spinner ให้แสดงเฉพาะชื่อประเทศ แต่ใน drop-down ให้แสดงรูปธงชาติและชื่อประเทศ ดังรูปถัดไป ทั้งหมดขึ้นอยู่กับเราว่าจะ return อะไรออกไปจากเมธอด `getView` และ `getDropDownView`



5. เพิ่มโค้ดในเมธอด `onCreate` ของแอกทิวิตี เพื่อเตรียมข้อมูลในอาร์เรย์ `countries`, สร้าง Adapter และผูกเข้ากับ Spinner

โปรเจ็ค CustomSpinnerDemo, ไฟล์ MainActivity.java

```
final TextView text = (TextView) findViewById(R.id.text);
final Spinner spinner = (Spinner) findViewById(R.id.list_of_countries);

Country countries[] = new Country[] {
    new Country(R.drawable.brunei, "Brunei"),
    new Country(R.drawable.cambodia, "Cambodia"),
    new Country(R.drawable.indonesia, "Indonesia"),
    new Country(R.drawable.laos, "Laos"),
    new Country(R.drawable.malaysia, "Malaysia"),
    new Country(R.drawable.myanmar, "Myanmar (Burma)"),
    new Country(R.drawable.philippines, "Philippines"),
    new Country(R.drawable.singapore, "Singapore"),
    new Country(R.drawable.thailand, "Thailand"),
```

```

        new Country(R.drawable.vietnam, "Vietnam"), });

CountryAdapter adapter = new CountryAdapter(this, R.layout.item,
                                              countries);
spinner.setAdapter(adapter);

```

## 6 เพิ่มโค้ดในเมธอด onCreate เพื่อระบุการทำงานเมื่อไอเท็มใน Spinner ถูกเลือก

โปรเจ็ค CustomSpinnerDemo, ไฟล์ MainActivity.java

```

spinner.setOnItemSelectedListener(
    new AdapterView.OnItemSelectedListener() {

        @Override
        public void onItemSelected(AdapterView<?> av, View v, int position,
                                     long id) {
            TextView tv = (TextView) v.findViewById(R.id.country_name);

            String msg = "You have selected " + tv.getText();
            msg += " at position " + String.valueOf(position);

            text.setText(msg);
        }

        @Override
        public void onNothingSelected(AdapterView<?> av) {
            return;
        }
    });

```

## 7 เพิ่มโค้ดในเมธอด onCreate เพื่อระบุการทำงานเมื่อปุ่มถูกคลิก

โปรเจ็ค CustomSpinnerDemo, ไฟล์ MainActivity.java

```

Button button = (Button) findViewById(R.id.show_selected_item_button);
button.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        String msg = "Selected item: ";
        /* ใช้เมธอด getSelectedItem เข้าถึงข้อมูลในไอเท็มที่ถูกเลือก ซึ่งข้อมูลนี้คือออบเจ็ค
           Country */
        Country selected = (Country) spinner.getSelectedItem();
        msg += selected.getCountryName();
        text.setText(msg);
    }
});

```

## ผลการรับ

