

User Interaction 1

التفاعل مع المستخدم

الوحدة الثامنة

ما لمقصود بالتفاعل مع المستخدم ؟

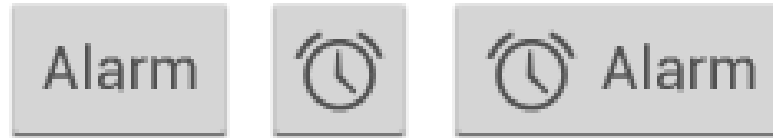
تفاعل المستخدم هي تجربة المستخدم الحقيقية بعد تفاعل واحد مع واجهة المستخدم. يبدأ

التفاعل بحدث تم إنشاؤه بواسطة مستخدم حقيقي عبر واجهة المستخدم في التطبيق. الأنواع

الشائعة لمثل هذه الأحداث هي النقر بالماوس واللمسات وأحداث لوحة المفاتيح.

الازرار - Buttons

يتكون الزر من نص أو رمز (أو كلاهما نص وأيقونة) التي توضح الإجراء الذي يحدث عندما يلمسه المستخدم.



بناءً على ما إذا كنت تريد زرًا به نص أو رمز أو كليهما ، يمكنك إنشاء الزر في XML File بثلاث طرق

- With text, using the `Button` class:

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_text"
    ... />
```

- With an icon, using the `ImageButton` class:

```
<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/button_icon"
    android:contentDescription="@string/button_icon_desc"
    ... />
```

- With text and an icon, using the `Button` class with the `android:drawableLeft` attribute:

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_text"
    android:drawableLeft="@drawable/button_icon"
    ... />
```

Within the `Activity` that hosts this layout, the following method handles the click event:

KOTLIN

JAVA

```
/** Called when the user touches the button */
public void sendMessage(View view) {
    // Do something in response to button click
}
```

For example, here's a layout with a button using `android:onClick`:

```
<?xml version="1.0" encoding="utf-8"?>
<Button xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/button_send"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_send"
    android:onClick="sendMessage" />
```

استجابة النقر على الزر

استخدام الميثود **OnClickListener**

تعريف الميثود : هو القيام بتسجيل رد اتصال ليتم استدعاؤه بعد النقر على الزر.
طريقة كتابة الميثود :

`public void setOnClickListener (View.OnClickListener)`

مثال :

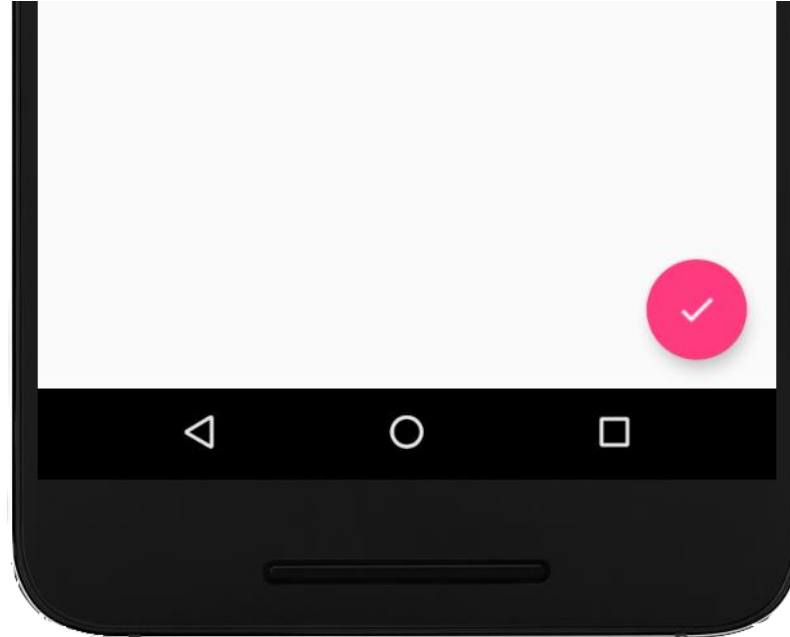
KOTLIN

JAVA

```
Button button = (Button) findViewById(R.id.button_send);  
button.setOnClickListener(new View.OnClickListener() {  
    public void onClick(View v) {  
        // Do something in response to button click  
    }  
});
```

زر الإجراء العائم (Floating Action Button)

زر الإجراء العائم (Floating Action Button) هو زر دائري يقوم بتشغيل الإجراء الأساسي في واجهة مستخدم تطبيقك.



You can then apply an `View.OnClickListener` to handle FAB taps. For example, the following code displays a `Snackbar` when the user taps the FAB:

KOTLIN

JAVA

```
FloatingActionButton fab = findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Snackbar.make(view, "Here's a Snackbar", Snackbar.LENGTH_LONG)
            .setAction("Action", null).show();
    }
});
```

The following code shows how the `FloatingActionButton` should appear in your layout file:

```
<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="end|bottom"
    android:src="@drawable/ic_my_icon"
    android:contentDescription="@string/submit"
    android:layout_margin="16dp" />
```

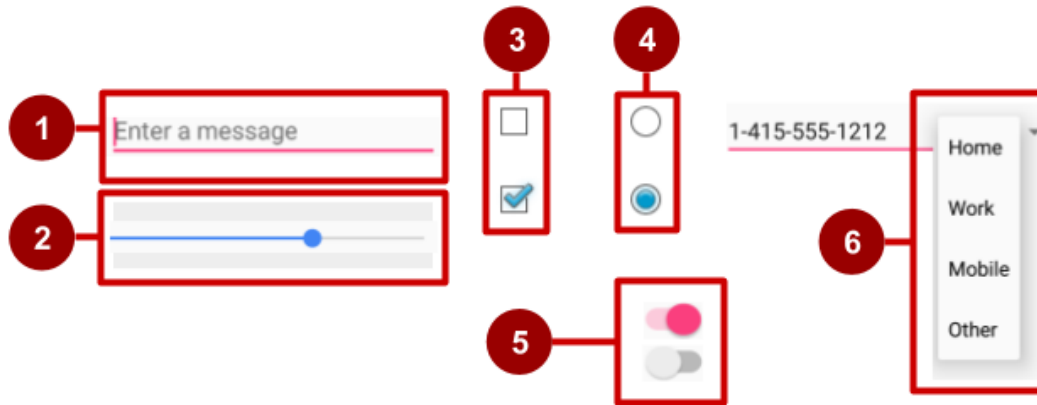

أدوات التحكم بالمدخلات

للكتابة الحرة نستخدم أداة النص القابل للتعديل مع لوحة المفاتيح Edit Text

للاختيارات نستخدم CheckBox, RadioButton, Spinner

للتحويل (تشغيل / إيقاف) نستخدم Toggle, Switch

اختيار قيمة من عدة قيم نستخدم SeekBar



كيفية عمل أدوات التحكم بالمدخلات

استخدم EditText لإدخال نص باستخدام لوحة المفاتيح

استخدم SeekBar لتمرير المؤشر إلى اليسار أو اليمين إلى أحد الإعدادات

استخدام CheckBox لاختيار أكثر من خيار واحد

استخدام Radio button يقوم المستخدم باختيار واحد فقط

استخدم مفتاح التبديل للتشغيل أو إيقاف التشغيل

استخدم Spinner لاختيار عنصر واحد من قائمة

EditText

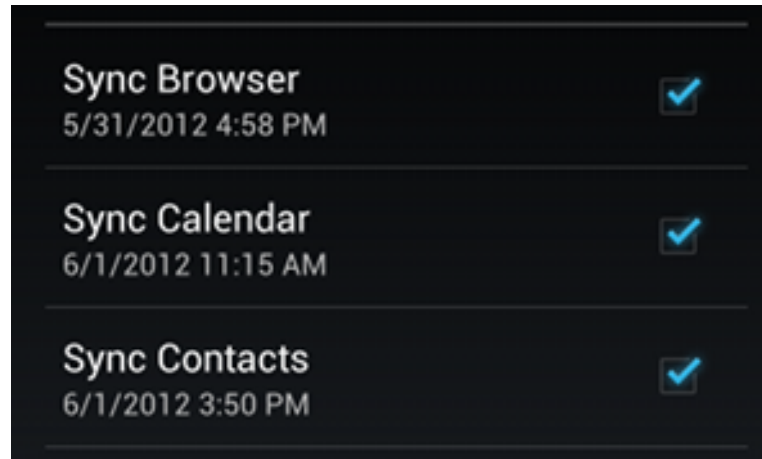
EditText هو عنصر تحكم واجهة مستخدم يستخدم للسماح للمستخدم بإدخال النص أو تعديله. أثناء

استخدام عنصر التحكم EditText في تطبيقات اندرويد الخاصة بنا ، نحتاج إلى تحديد نوع البيانات

التي يمكن أن يقبلها حقل النص باستخدام سمة `inputType`

CheckBox

تسمح مربعات الاختيار للمستخدم بتحديد خيار واحد أو أكثر من مجموعة. عادة ، يجب عليك تقديم كل خيار مربع اختيار في قائمة عمودية.



```
public void onCheckboxClicked(View view) {  
    // Is the view now checked?  
    boolean checked = ((CheckBox) view).isChecked();  
  
    // Check which checkbox was clicked  
    switch(view.getId()) {  
        case R.id.checkbox_meat:  
            if (checked)  
                // Put some meat on the sandwich  
            else  
                // Remove the meat  
            break;  
        case R.id.checkbox_cheese:  
            if (checked)  
                // Cheese me  
            else  
                // I'm lactose intolerant  
            break;  
        // TODO: Veggie sandwich  
    }  
}
```

For example, here are a couple `CheckBox` objects in a list:

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent">  
    <CheckBox android:id="@+id/checkbox_meat"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="@string/meat"  
        android:onClick="onCheckboxClicked"/>  
    <CheckBox android:id="@+id/checkbox_cheese"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="@string/cheese"  
        android:onClick="onCheckboxClicked"/>  
</LinearLayout>
```

Radio Button

تسمح أزرار الاختيار للمستخدم بتحديد خيار واحد من مجموعة.

ATTENDING?

☒ Yes ☐ Maybe ☐ No

KOTLIN

JAVA

```
public void onRadioButtonClicked(View view) {  
    // Is the button now checked?  
    boolean checked = ((RadioButton) view).isChecked();  
  
    // Check which radio button was clicked  
    switch(view.getId()) {  
        case R.id.radio_pirates:  
            if (checked)  
                // Pirates are the best  
                break;  
        case R.id.radio_ninjas:  
            if (checked)  
                // Ninjas rule  
                break;  
    }  
}
```

For example, here are a couple `RadioButton` objects:

```
<?xml version="1.0" encoding="utf-8"?>  
<RadioGroup xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="vertical">  
    <RadioButton android:id="@+id/radio_pirates"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="@string/pirates"  
        android:onClick="onRadioButtonClicked"/>  
    <RadioButton android:id="@+id/radio_ninjas"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="@string/ninjas"  
        android:onClick="onRadioButtonClicked"/>  
</RadioGroup>
```

Toggle Button

يتيح زر التبديل للمستخدم تغيير الإعداد بين حالتين.




```
<ToggleButton
    android:id="@+id/toggle1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="100dp"
    android:layout_marginTop="120dp"
    android:checked="true"
    android:textOff="OFF"
    android:textOn="ON"/>
```

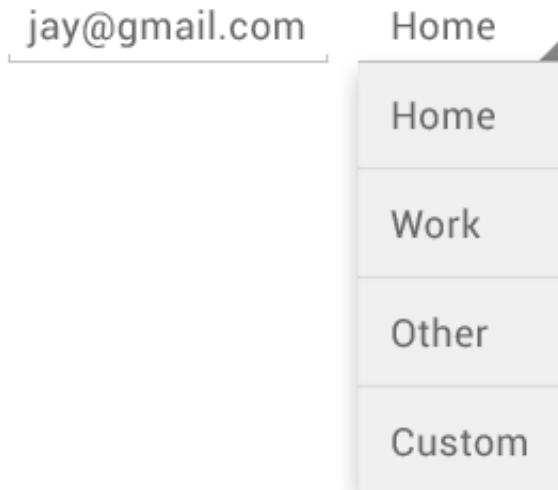
KOTLIN

JAVA

```
ToggleButton toggle = (ToggleButton) findViewById(R.id.togglebutton);
toggle.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
        if (isChecked) {
            // The toggle is enabled
        } else {
            // The toggle is disabled
        }
    }
});
```

Spinners

يوفر Spinners طريقة سريعة لتحديد قيمة واحدة من مجموعة. في الحالة الافتراضية ، يظهر المؤشر على القيمة المحددة حاليًا. يؤدي لمس المؤشر إلى عرض قائمة منسدلة تحتوي على جميع القيم الأخرى المتاحة ، والتي يمكن للمستخدم تحديد قيمة جديدة منها.



KOTLIN JAVA

```
Spinner spinner = (Spinner) findViewById(R.id.spinner);
// Create an ArrayAdapter using the string array and a default spinner layout
ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,
    R.array.planets_array, android.R.layout.simple_spinner_item);
// Specify the layout to use when the list of choices appears
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
// Apply the adapter to the spinner
spinner.setAdapter(adapter);
```

KOTLIN JAVA

```
public class SpinnerActivity extends Activity implements OnItemSelectedListener {
    ...

    public void onItemSelected(AdapterView<?> parent, View view,
        int pos, long id) {
        // An item was selected. You can retrieve the selected item using
        // parent.getItemAtPosition(pos)
    }

    public void onNothingSelected(AdapterView<?> parent) {
        // Another interface callback
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="planets_array">
        <item>Mercury</item>
        <item>Venus</item>
        <item>Earth</item>
        <item>Mars</item>
        <item>Jupiter</item>
        <item>Saturn</item>
        <item>Uranus</item>
        <item>Neptune</item>
    </string-array>
</resources>
```

```
<Spinner
    android:id="@+id/planets_spinner"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```

KOTLIN JAVA

```
Spinner spinner = (Spinner) findViewById(R.id.spinner);
spinner.setOnItemSelectedListener(this);
```