

Input events and Event listeners

سؤال, مناقشة goo.gl

3ي

Sala7.Ra4ad

Input events and Event listeners

<https://goo.gl/T7vwPf>

عايز افهم الـ Topic بطريقة مبسطة وموضحة يا شباب جزاكم الله خيراً

INPUT EVENTS & EVENT LISTENERS

Check out the article in the Instructor Notes about **Input Events** in Android.

Specifically, read through the introduction and skim through the **Event Listeners** section.

Write down a couple of sentences to summarize your understanding of **input events** and **event listeners**.

✓ Solved by mismail80 in post #3

في البداية خلينا نعرف معنى كلمة Event: وهو حدث أو تأثير ناتج من تفاعل المستخدم مع واجهة التطبيق والان دعنا نقرب الصورة بمثال من حياتنا اليومية ... فمثلا المصباح الموجود في غرفتك يتحكم فيه أحد الأزرار الموجوده على الحائط ... واذا أردت انارة المصباح فبالتأكيد يجب عليك الضغط على هذا الزرار وبالتالي تنير ا...

9 عايز اعرف ايجابية سؤال

2ي

aabed91
Tutor

بكل بساطة هي التعامل مع أي action ممكن يصير معك في الواجهة اللي انت شغال عليها. يعني مثلا لو عند زر والمستخدم ضغط عليه، ما هو الشيء اللي لازم يصير. لو عندي edit text والمستخدم بدا يكتب فيه، هل لازم استجيب انه النص بيتغير فيه ولا لا. لو عندي مثلا edit text وصار عليه focus او خلص المستخدم منه وراح focus عنه ايش المطلوب اعمل. لو المستخدم ضغط على اي حاجة في الواجهة ايش المفترض اعمل.

يعني باختصار كيف استجيب واتعامل مع الاشياء اللي بيعملها المستخدم في الواجهة

في البداية خلينا نعرف معنى كلمة **Event**: وهو حدث أو تأثير ناتج من تفاعل المستخدم مع واجهة التطبيق

والان دعنا نقرب الصورة بمثال من حياتنا اليومية ... فمثلا المصباح الموجود في غرفتك يتحكم فيه أحد الأزرار الموجوده على الحائط ... وإذا أردت انارة المصباح فبالأكيد يجب عليك الضغط على هذا الزرار وبالتالي تنير المصباح ... وهنا فإن الضغط على الزرار يعتبر **event** وهو ضروري ليقوم الزرار بوظيفته ... نفس المثال في التطبيقات ... اذا أردت أن تقوم بوظيفة مافي التطبيق ولكن بعد أن يقوم المستخدم بعمل أو حدث ما **event** فيجب أن تربط الوظيفة بهذا الحدث **event** ... فمثلا اذا كان التطبيق يحتوى على زرار وأردت أن يقوم المستخدم بالضغط عليه لينتقل للشاشة التالية ... فيجب ربط **event** الضغط بالكود الخاص بالانتقال للشاشة التالية ... وهنا تأتي أهمية ال **events**

وخلاصة الكلام السابق هو أن الأحداث **events** مسؤولة عن تنفيذ كود معين بعد تفاعل المستخدم مع أحد العناصر **views** في شاشة التطبيق. وبما أن معظم التفاعل بين المستخدم والتطبيق يتم من خلال شاشة اللمس الخاصة بالموبايل فإن معظم الأحداث **events** تقع تحت فئة **input events**

وبذلك ننتهي من الجزء الأول من السؤال 😊

أما الجزء الثاني وهو **Event listeners** ... في الكلام أعلاه ذكرنا أن بعد وقوع الحدث **event** يقوم التطبيق بتنفيذ كود ما ... مثل الزرار الذي ينقلنا الى الشاشة التالية .

وهنا تأتي أهمية ما يسمى بـ **listener** وهو دائما يكون عبارة عن **callback method** ويقوم المبرمج بكتابة الكود المطلوب تنفيذه بداخلها ... ليتم تنفيذ هذا الكود بعد وقوع الحدث **event**.

والآن دعنا نقوم بشرح المثال السابق بالكود المفصل

فنفترض عندنا زرار للانتقال للشاشة التالية فسوف يكون الكود كالتالي:

```
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="gotoNextActivity"
    android:text="Click me" />
```

نلاحظ الخاصية **onClick** وهي هنا تمثل الحدث **event** ... والحدث هنا هو **click** بمعنى الضغط على الزرار ... وقيمة الخاصية هنا هي الـ **listener** والتي ستنفذ عند الضغط على الزرار ... وسيكون الكود كالتالي:

```
public void gotoNextActivity(View v) {
    // هنا سيكتب الكود الخاص بالانتقال للشاشة التالية
}
```

أتمنى المعلومة تكون وصلت ... بالتوفيق

👉 ممكن شرح للدروس دي

تمام الكلام ده كله
ليه بقا استخدم في التطبيق ده الطريقة دي:

```
// Find the View that shows the family category
TextView family = (TextView) findViewById(R.id.family);
```

```
// Set a click listener on that View
family.setOnClickListener(new OnClickListener() {
    // The code in this method will be executed when the family category is cli
    @Override
    public void onClick(View view) {
        // Create a new intent to open the {@link FamilyActivity}
        Intent familyIntent = new Intent(MainActivity.this, FamilyActivity.class);

        // Start the new activity
        startActivity(familyIntent);
    }
});
```

وما استخدم الطريقة دي على طول اسهل واقل في الكود ؟:

```
public void openNumbersActivity(View view){
    Intent i = new Intent(this, NumbersActivity.class);
    startActivity(i);
}
```

باستخدام:

android:onClick="openNumbersActivity"

1ي

mismail80

أولا يجب أن تعلم أن الطريقتين سوف ينتج منهما نفس الشيء ... وهو تعريف click event للزرار ... ولا يوجد فرق بينهما من ناحية الأداء في سرعة التطبيق ... وفي الدرس لا يوجد سبب لاستعمال طريقة عن طريقة ... بسبب أننا في مرحلة التعلم ... وهم يحاولون التنويع بينهم لتصل لنا الفائدة أكثر.

ولكن قبل أن أشرح الفرق يجب التنويه بأن android:onClick تعمل من الاصدار API level 4 وما بعد ذلك ... وبذلك لا يمكن استعمالها مع الاصدارات الأقدم من ذلك.

والآن نشرح الفرق ... وسأذكرها في عدة نقاط:

1. اذا أردت تحديد اي طريقة أنسب فهذا يعتمد على مدى تعقيد وحجم التطبيق ... فمع التطبيقات البسيطة أفضل استخدام onClick وكلما زاد تعقيد التطبيق ومتطلباته أنصحك باستخدام OnClickListener.
2. في هذه النقطة سأشرح لماذا يفضل استخدام OnClickListener مع التطبيقات الكبيرة والمعقدة نسبيا ... كما نعلم أن ملفات الـ xml تمثل جانب الـ design في التطبيق وملفات الـ java تمثل الـ logic ... مع استعمال android:onClick نقوم بكتابة جزء من الـ logic داخل الـ design ... وعند استعمال OnClickListener نقوم بالفصل التام بين الـ design والـ logic وهذا أفضل كلما زاد حجم التطبيق وتعقيده ... ليسهل صيانة الكود في المستقبل وازافة تعديلات جديدة بكل سهولة.
3. وفي النقطة الثالثة سأشرح لك فائدة فصل الـ design عن الـ logic ... السبب الأول باستخدام OnClickListener تستطيع في أي وقت أن تغير سلوك ووظيفة الزرار أثناء تشغيل التطبيق في الـ run time ... بمعنى عند حدوث شرط معين يمكن أن تجعل الزرار أن يشير لـ method أخرى تقوم بوظيفة أخرى وعند انتهاء الشرط يمكن أن تجعل الزرار يشير الى وظيفته السابقة ... فستلاحظ هنا الديناميكية التي حصلنا عليها من فصل الـ design عن الـ logic ... سبب ثانى وهو يمكنك حذف أو تعطيل الـ event listener فى أى وقت بأن تجعل OnClickListener لا تفعل شئ بأن تجعلها تشير الى method خالية من الكود مثلا 😊
4. فائدة أخرى من فصل الـ design عن الـ logic ... وهي حالة متقدمة بعض الشيء ... عند استعمال onClick فإن الزرار سوف يبحث عن الـ method داخل الـ activity الحالية التي تحتوى على الزرار ... ولكن باستخدام OnClickListener يمكن أن توجه الزرار لـ Method أخرى موجودة فى activity أخرى ... وهذه فائدة كبيرة وخصوصا مع استعمال الـ Fragments التي تعتبر activity مصغرة يمكن اعادتها استخدامها فى أكثر من مكان داخل التطبيق.

سأكتفى بهذه النقاط لأن الموضوع يحتاج سرد أكثر من ذلك ... حاولت أوضحها لك بأبسط الطرق وبدون إطالة ... أنصحك بالبحث أكثر على الانترنت عن أمثلة عملية وتطبيقها لتفهم الموضوع بشكل أفضل ... بالتوفيق

Miwok - onClick Listener ☞

☞ ممكن شرح للدروس دي

1ي

Sala7.Ra4ad

شكراً جزيلاً على تعبك بجد وربنا يوفقك وتبقى من الأوائل يارب 😊

1ي

mismail80

العفو يا صديقي 😊 ... وان شاء الله موفقين جميعاً