

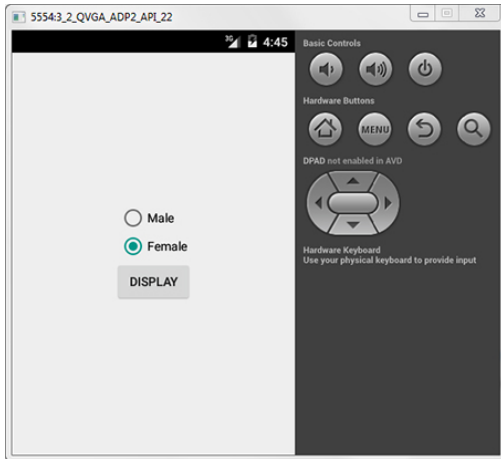
Slide 9.6: Hypertext: Java source code (cont.)

Slide 9.8: Radio buttons: layout file

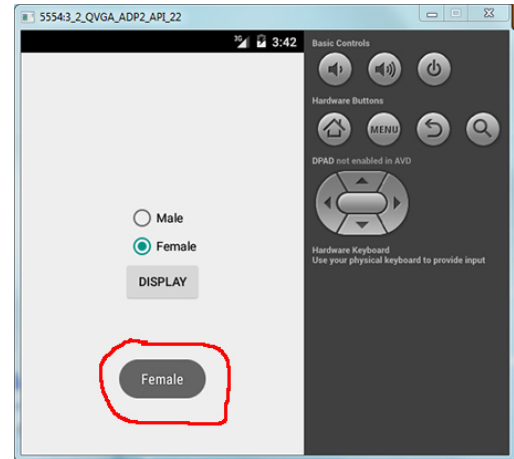
Home



Radio Buttons



Selecting
Female &
pushing
DISPLAY



[Radio buttons](#) allow the user to select one option from a set. To create each radio button option, create a [RadioButton](#) in your layout. However, because radio buttons are mutually exclusive, you must group them together inside a [RadioGroup](#). By grouping them together, the system ensures that only one radio button can be selected at a time.

Responding to Click Events

When the user selects one of the radio buttons, the corresponding `RadioButton` object receives an on-click event. To define the click event handler for a button, add the `android:onClick` attribute to the `RadioButton` element in your XML layout. The value for this attribute must be the name of the method you want to call in response to a click event. The activity hosting the layout must then implement the corresponding method. However, this example uses another approach.

MyApplication/app/src/main/AndroidManifest.xml

```

01 <?xml version="1.0" encoding="utf-8"?>
02 <manifest
03     xmlns:android = "http://schemas.android.com/apk/res/android"
04     package = "com.example.wenchen.myapplication" >
05     <application

```

```
06     android:icon    = "@mipmap/ic_launcher"
07     android:label    = "@string/app_name"
08     android:theme     = "@style/AppTheme" >
09     <activity
10         android:name    = ".MainActivity"
11         android:label    = "@string/app_name" >
12         <intent-filter>
13             <action      android:name = "android.intent.action.MAIN" />
14             <category    android:name = "android.intent.category.LAUNCHER" />
15         </intent-filter>
16     </activity>
17 </application>
18 </manifest>
```

[Slide 9.6: Hypertext: Java source code \(cont.\)](#)

[Slide 9.8: Radio buttons: layout file](#)

[Home](#)

