




---

## Lecture 6: Layouts, Menus, Views

# Create an Android Virtual Device

- Click the AVD Icon: 
- Window -> AVD Manager -> New
- Name & start the virtual device (this may take a minute or two)

Create new Android Virtual Device (AVD)

AVD Name:

Device:

Target:

CPU/ABI:

Keyboard:  Hardware keyboard present

Skin:  Display a skin with hardware controls

Front Camera:

Back Camera:

Memory Options: RAM:  VM Heap:

Internal Storage:

SD Card:

Size:

File:

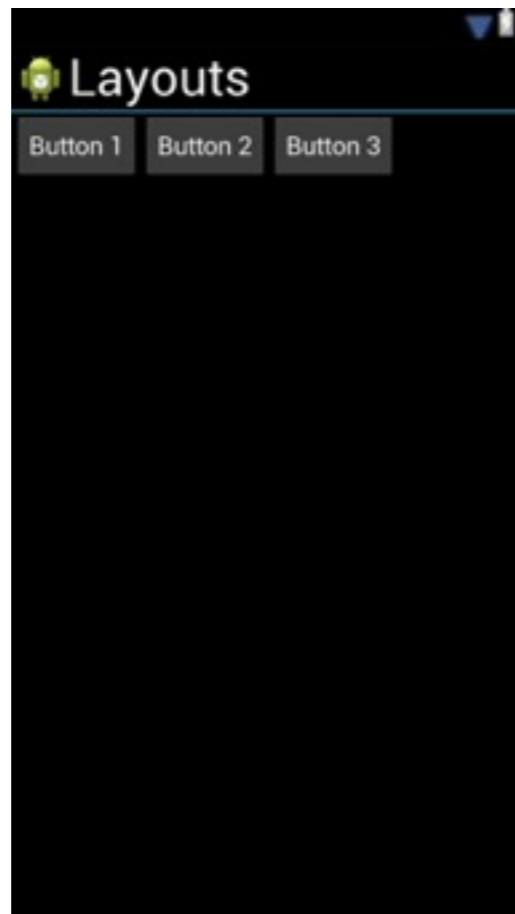
Emulation Options:  Snapshot  Use Host GPU

Override the existing AVD with the same name

**What is a layout?**

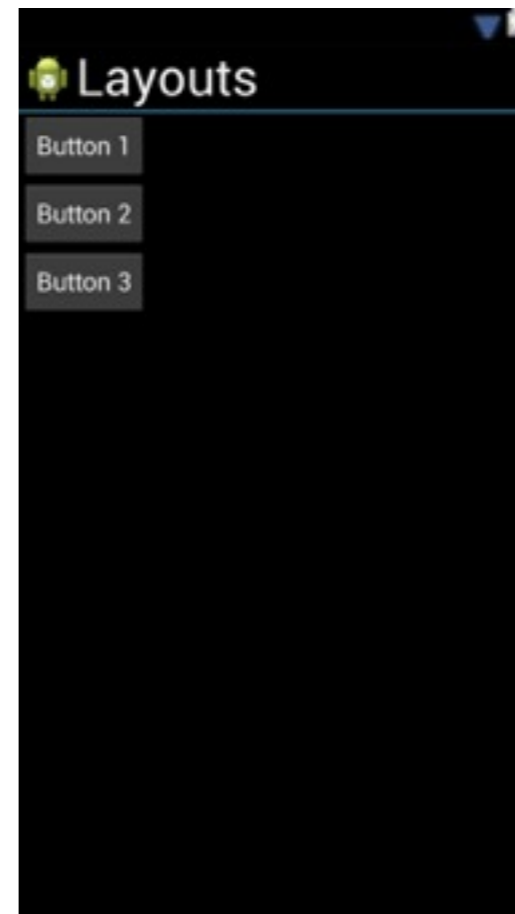
# LinearLayouts

---



Horizontal

*or*

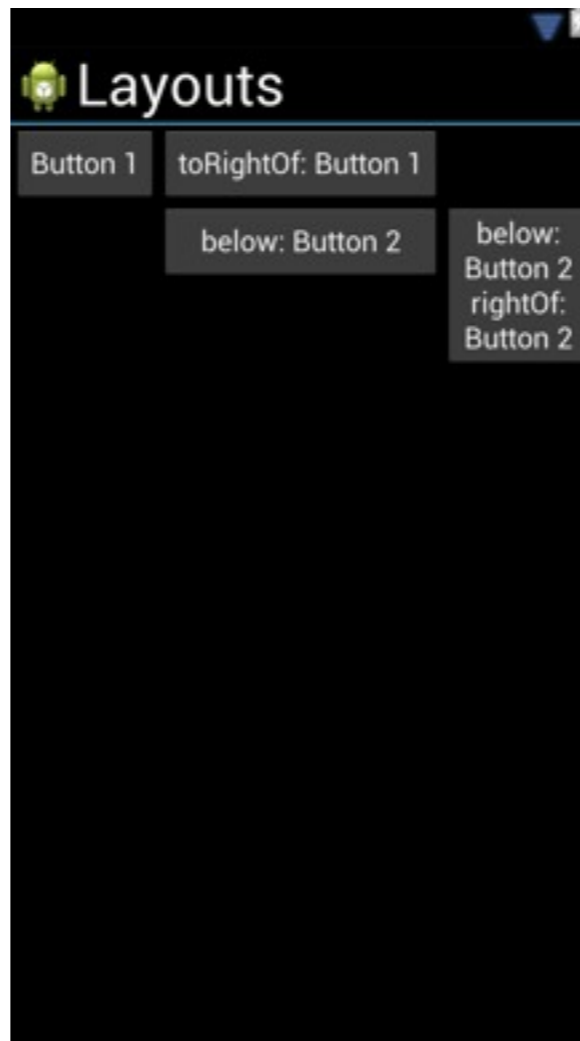


Vertical

Lay out controls in order,  
may display scrollbar

# RelativeLayouts

---

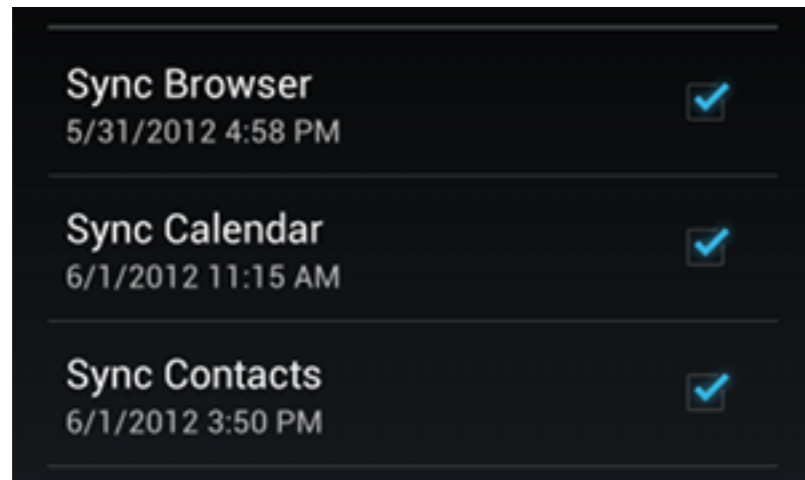


Lay out controls relative to each other

# Types of user controls



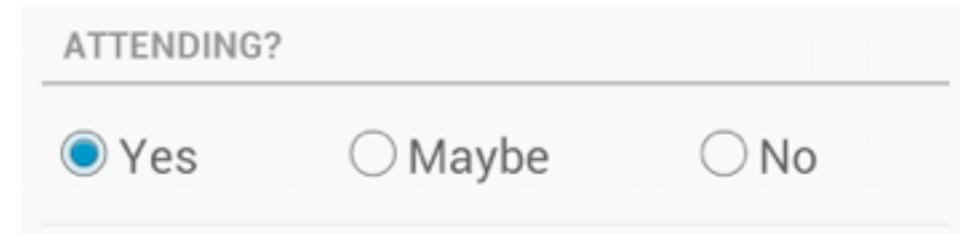
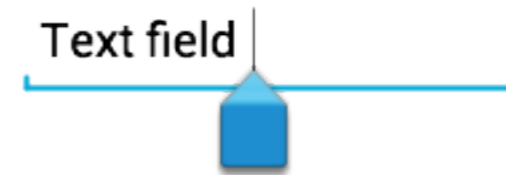
## Buttons



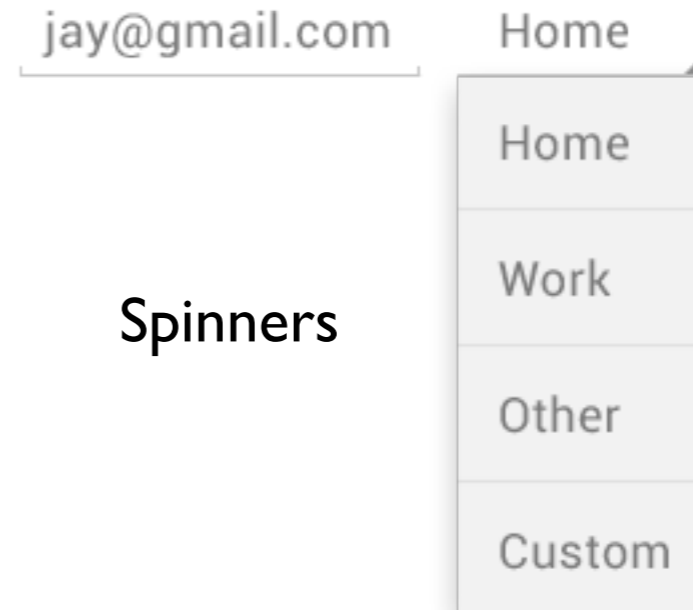
## Checkboxes



## Toggle Buttons



## Radio Buttons



## Spinners



# Other User Interface Components

---

- *TextView* – Display static text
- *WebView* – Display HTML/Websites
- *ImageView* – Display an image
- Composite views to display multiple items:
  - *ListView* – In a vertical list
  - *GridView* – In a (vertical) grid
  - *Gallery* – A horizontal sequential gallery (of images)

# Making interfaces in Eclipse

# Graphical Layout

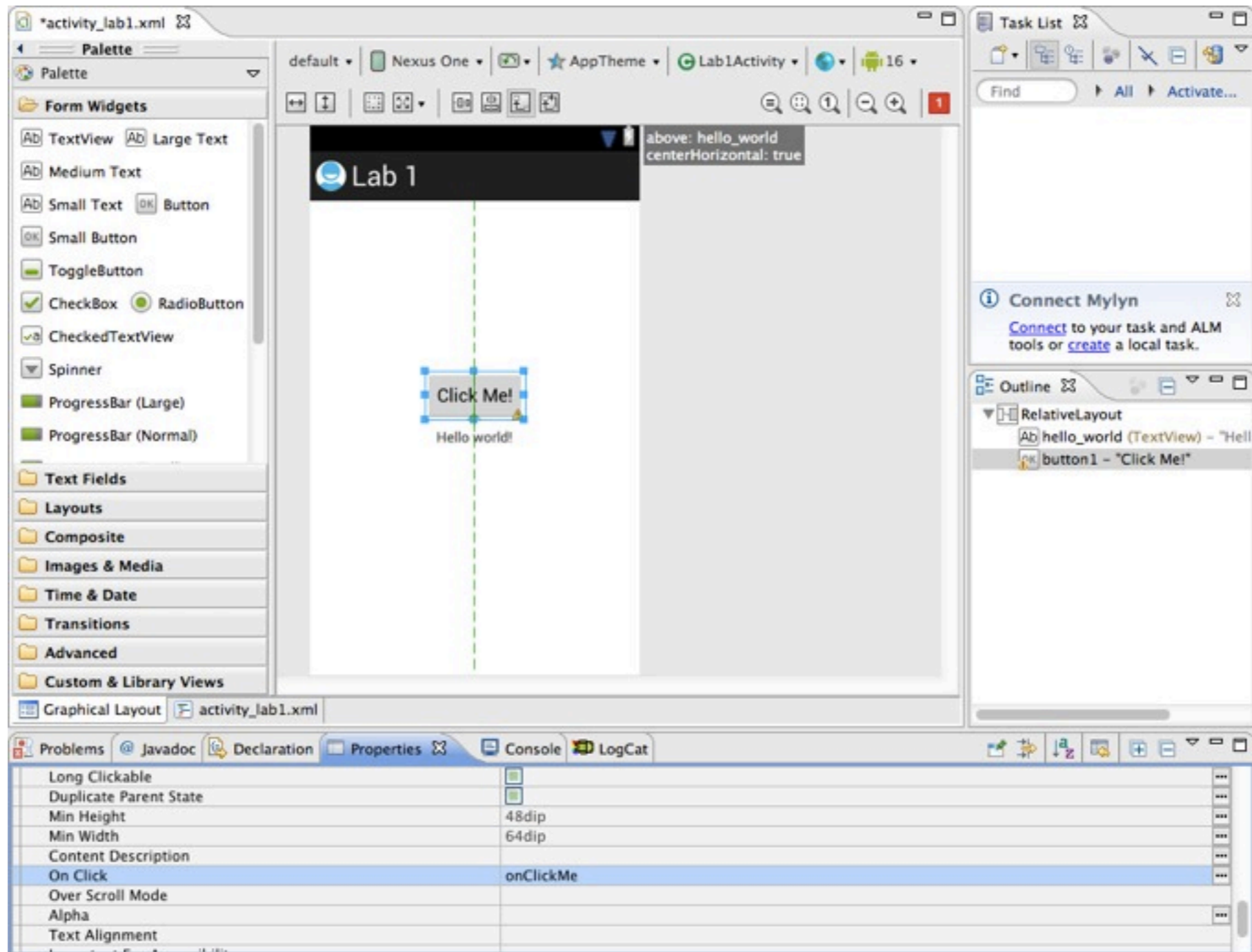
The screenshot displays the graphical layout editor for an Android activity. The interface is divided into several key sections:

- Palette (Left):** A list of UI controls categorized into Form Widgets, Text Fields, Layouts, Composite, Images & Media, Time & Date, Transitions, Advanced, and Custom & Library Views. A red arrow labeled "Controls" points to this area.
- Design View (Center):** A visual representation of the activity layout. It shows a black header bar with a blue back arrow and the text "Lab 1". Below the header, the text "Hello world!" is displayed on a white background.
- Outline (Right):** A hierarchical view of the layout structure. It shows a `RelativeLayout` container containing a `TextView` widget with the text "Hello world!". A red arrow labeled "Layout Structure" points to this area.
- Properties (Bottom):** A table-like view showing the properties of the selected `TextView` widget. A red arrow labeled "Properties" points to this area. The `layout_width` property is highlighted with a yellow tooltip.

Id	Value
Layout Parameters	{}
Gravity	
<b>layout_width</b>	match_parent
	match_parent
Specifies the basic width of the view. [dimension, enum]	
Background	
Padding Left	
Content Description	

# Using controls in your code

# A Simple Button



# A Simple Button: Code View

---

```
<RelativeLayout 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" >

    <TextView
        android:id="@+id/hello_world"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:padding="@dimen/padding_medium"
        android:text="@string/hello_world"
        tools:context=".Lab1Activity" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/hello_world"
        android:layout_centerHorizontal="true"
        android:onClick="onClickMe"
        android:text="Click Me!" />

</RelativeLayout>
```

# A Simple Button: Code View

---

```
<RelativeLayout 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" >
```

```
<TextView
```

```
  android:id="@+id/hello_world"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_centerHorizontal="true"
  android:layout_centerVertical="true"
  android:padding="@dimen/padding_medium"
  android:text="@string/hello_world"
  tools:context=".Lab1Activity" />
```

*[name the view]*

```
<Button
```

```
  android:id="@+id/button1"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_above="@+id/hello_world"
  android:layout_centerHorizontal="true"
  android:onClick="onClickMe"
  android:text="Click Me!" />
```

*[name the view]*

```
</RelativeLayout>
```

# A Simple Button: Code View

---

```
<RelativeLayout 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" >
```

*[expand to fill parent]*

```
<TextView
  android:id="@+id/hello_world"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_centerHorizontal="true"
  android:layout_centerVertical="true"
  android:padding="@dimen/padding_medium"
  android:text="@string/hello_world"
  tools:context=".Lab1Activity" />
```

*[minimum needed size]*

*[horizontally and vertically center]*

```
<Button
  android:id="@+id/button1"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_above="@+id/hello_world"
  android:layout_centerHorizontal="true"
  android:onClick="onClickMe"
  android:text="Click Me!" />
```

*[minimum needed size]*

*[horizontally center above hello\_world]*

```
</RelativeLayout>
```



# A Simple Button: Code View

---

```
<RelativeLayout 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" >
```

```
<TextView
    android:id="@+id/hello_world"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="true"
    android:padding="@dimen/padding_medium"
    android:text="@string/hello_world"
    tools:context=".Lab1Activity" />
```

```
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/hello_world"
    android:layout_centerHorizontal="true"
    android:onClick="onClickMe"
    android:text="Click Me!" />
```

*[name of the method to call when clicked]*

```
</RelativeLayout>
```

# A Simple Button: Code View

---

```
<RelativeLayout 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" >
```

```
<TextView
  android:id="@+id/hello_world"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_centerHorizontal="true"
  android:layout_centerVertical="true"
  android:padding="@dimen/padding_medium"
  android:text="@string/hello_world"
  tools:context=".Lab1Activity" />
```

*[reference a string resource]*

```
<Button
  android:id="@+id/button1"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_above="@+id/hello_world"
  android:layout_centerHorizontal="true"
  android:onClick="onClickMe"
  android:text="Click Me!" />
```

*[literal string of the button]*

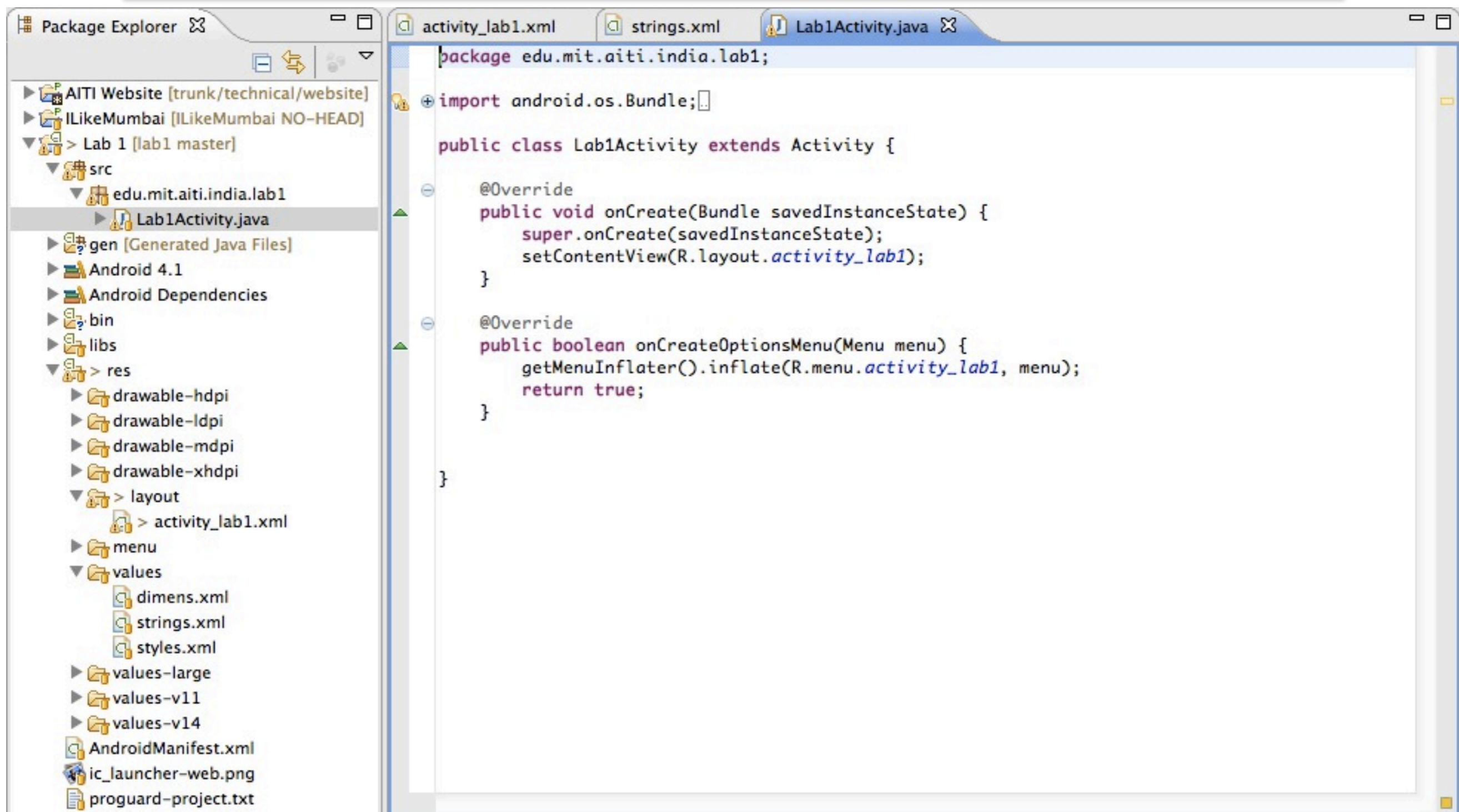
```
</RelativeLayout>
```

# String Resources

The screenshot shows the Android Studio interface. On the left, the Package Explorer displays the project structure, with the 'strings.xml' file selected under the 'res/values' directory. The main editor area shows the 'Android Resources (default)' view for 'strings.xml'. The 'Resources Elements' list contains four entries: 'app\_name (String)', 'hello\_world (String)', 'menu\_settings (String)', and 'title\_activity\_lab1 (String)'. The 'hello\_world (String)' entry is selected. To the right of the list are buttons for 'Add...', 'Remove...', 'Up', and 'Down'. The 'Attributes for hello\_world (String)' panel shows the 'Name' field containing 'hello\_world' and the 'Value\*' field containing 'Hello world!'.

**But what about the code?**

# The Code



The screenshot shows an IDE window with the Package Explorer on the left and the code editor on the right. The Package Explorer shows the project structure for 'Lab 1 [lab1 master]', including the 'src' directory, 'edu.mit.aiti.india.lab1' package, and 'Lab1Activity.java' file. The code editor displays the following Java code:

```
package edu.mit.aiti.india.lab1;

import android.os.Bundle;

public class Lab1Activity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_lab1);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_lab1, menu);
        return true;
    }

}
```

# The Code

```
package edu.mit.aiti.india.lab1;

import android.os.Bundle;

public class Lab1Activity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_lab1); [instantiate activity_lab1.xml]
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_lab1, menu);
        return true;
    }

}
```

# The Code:

## onClickMe()

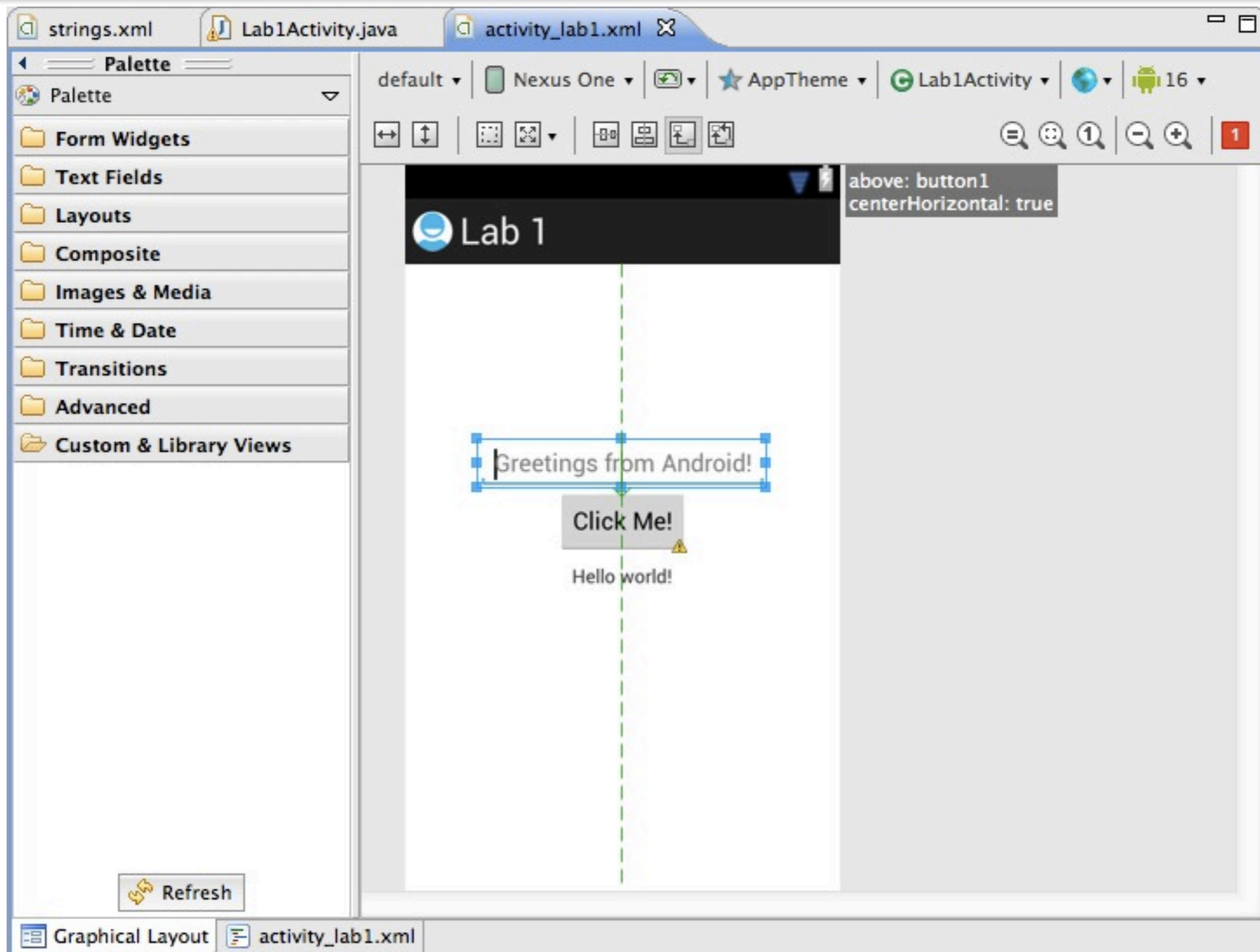
---

```
public void onClickMe(View button) {  
    /* get TextView @+id/hello_world */  
    TextView hello_world =  
        (TextView) findViewById(R.id.hello_world);  
  
    /* set its text to the string resource @string/greetings */  
    hello_world.setText(getString(R.string.greetings));  
}
```

# Capturing User Input: Adding an EditText



# Adding an EditText



# Adding an EditText: Code View

---

```
<EditText
    android:id="@+id/editText1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/button1"
    android:layout_centerHorizontal="true"
    android:ems="10"
    android:hint="@string/greetings_from_android"
    android:inputType="textNoSuggestions" >

    <requestFocus />
</EditText>
```

*[“placeholder” text]*  
*[workaround for bug in SDK]*

# Aside:

# EditText inputTypes

---

```
<EditText
    android:id="@+id/editText1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/button1"
    android:layout_centerHorizontal="true"
    android:ems="10"
    android:hint="@string/greetings_from_android"
    android:inputType="textNoSuggestions" >

    <requestFocus />
</EditText>
```



# Aside:

## EditText inputTypes

---

```
<EditText
    android:id="@+id/editText1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/button1"
    android:layout_centerHorizontal="true"
    android:ems="10"
    android:hint="@string/greetings_from_android"
    android:inputType="textNoSuggestions|textEmailAddress" >

    <requestFocus />
</EditText>
```



# Aside:

## EditText inputTypes

---

```
<EditText
    android:id="@+id/editText1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/button1"
    android:layout_centerHorizontal="true"
    android:ems="10"
    android:hint="@string/greetings_from_android"
    android:inputType="textNoSuggestions|phone" >

    <requestFocus />
</EditText>
```



# Aside:

## EditText inputTypes

---

```
<EditText
    android:id="@+id/editText1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/button1"
    android:layout_centerHorizontal="true"
    android:ems="10"
    android:hint="@string/greetings_from_android"
    android:inputType="textNoSuggestions|textPassword" >

    <requestFocus />
</EditText>
```



# The Code:

## onClickMe()

---

```
public void onClickMe(View button) {
    TextView hello_world =
        (TextView) findViewById(R.id.hello_world);
    EditText text_box =
        (EditText) findViewById(R.id.text_box);

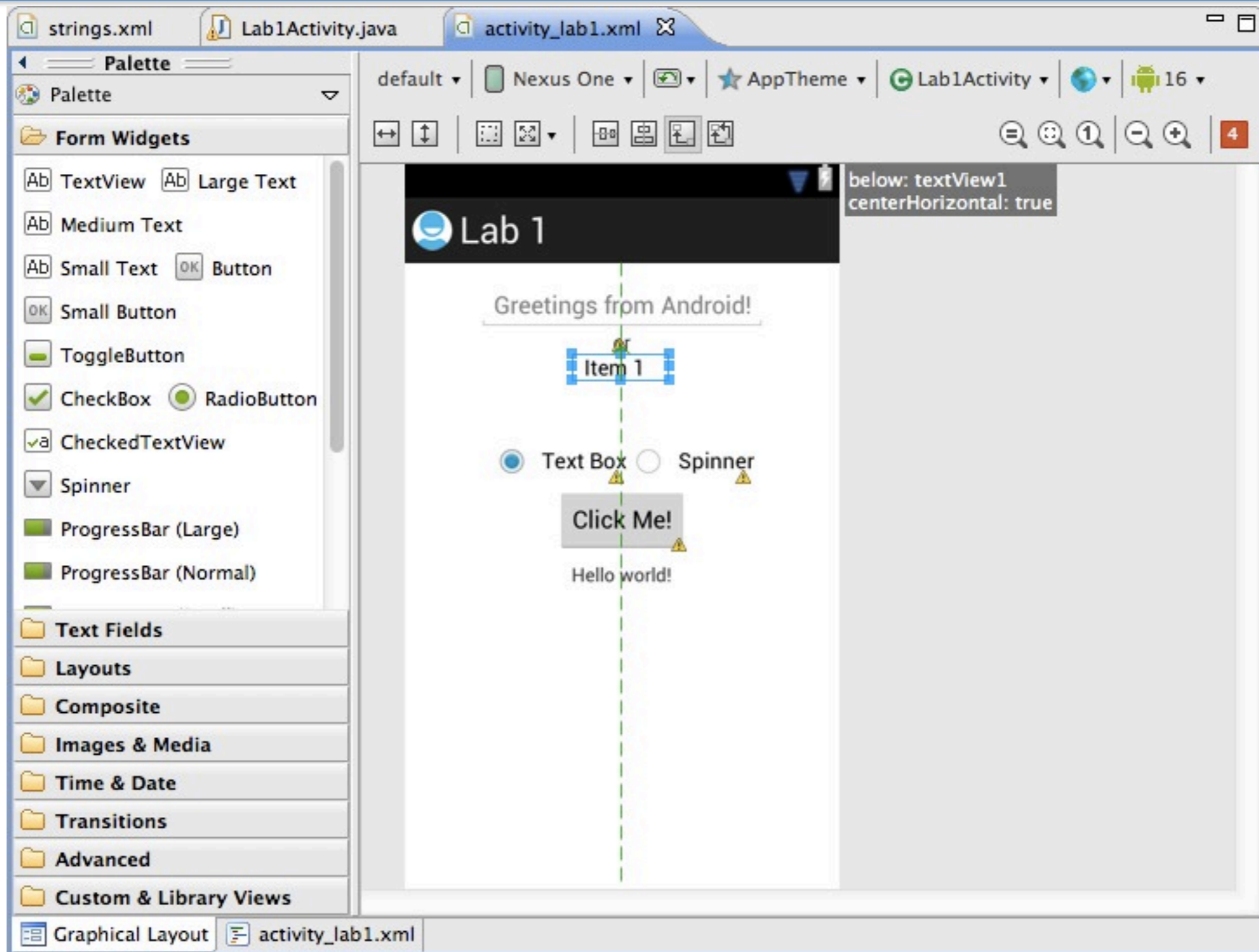
    if (text_box.getText().length() == 0) {
        hello_world.setText(getString(R.string.greetings));
    } else {
        /* set the text to the text in text_box if not empty */
        hello_world.setText(text_box.getText());
    }
}
```

# Adding an EditText: Demo



# Making choices: Radio Buttons and Spinners

# Radio Buttons and Spinners



# Radio Buttons and Spinners

## Code View

---

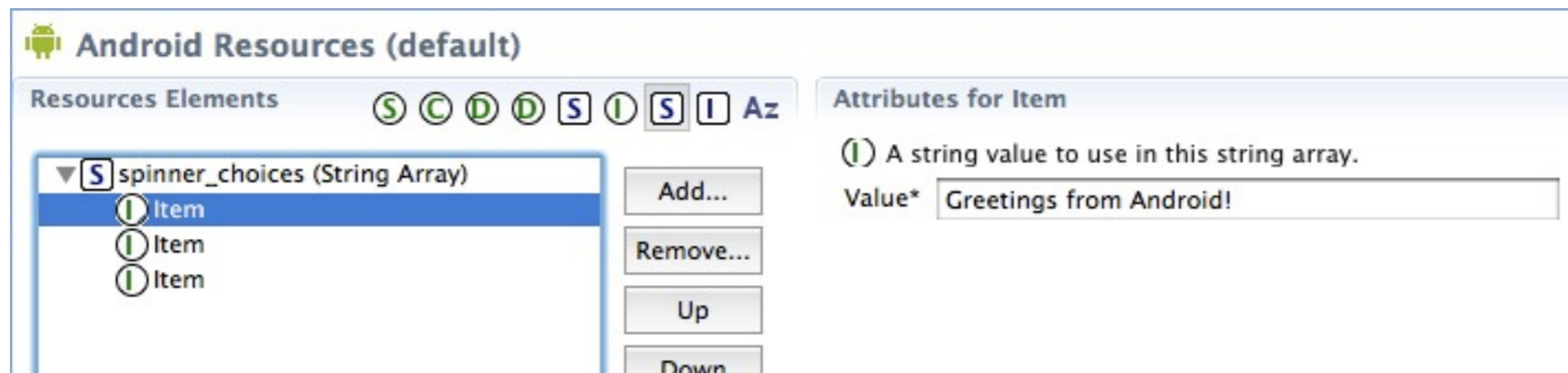
```
<RadioGroup android:id="@+id/chooseSource"  
    android:orientation="horizontal" >  
    <RadioButton android:id="@+id/radio0"  
        android:checked="true"  
        android:text="Text Box" />  
    <RadioButton android:id="@+id/radio1"  
        android:text="Spinner" />  
</RadioGroup>
```

*[This button is checked]*

```
<Spinner android:id="@+id/spinner"  
    android:entries="@array/spinner_choices" /> [The choices in the spinner]
```

# Radio Buttons and Spinners

## Resources



[or]

```
<resources>
```

```
<string name="app_name">Lab 1</string>
<string name="hello_world">Hello world!</string>
<string name="menu_settings">Settings</string>
<string name="title_activity_lab1">Lab 1</string>
<string name="greetings_from_android">Greetings from Android!</string>
<string-array name="spinner_choices">
    <item >Greetings from Android!</item>
    <item >Les salutations des Android!</item>
    <item >iSaludos desde Android!</item>
</string-array>
```

```
</resources>
```

# The Code:

## onClickMe()

---

```
public void onClickMe(View button) {
    RadioButton radio_text_box =
        (RadioButton) findViewById(R.id.radio0);

    if (radio_text_box.isChecked()) {
        /* as described earlier ... */
    } else {
        Spinner spinner = (Spinner) findViewById(R.id.spinner);
        /* set the text to the text in the spinner */
        hello_world.setText(
            spinner.getItemAtPosition(
                spinner.getSelectedItemPosition()
            ).toString());
    }
}
```