Accelerating Information Technology Innovation

http://aiti.mit.edu

India Summer 2012

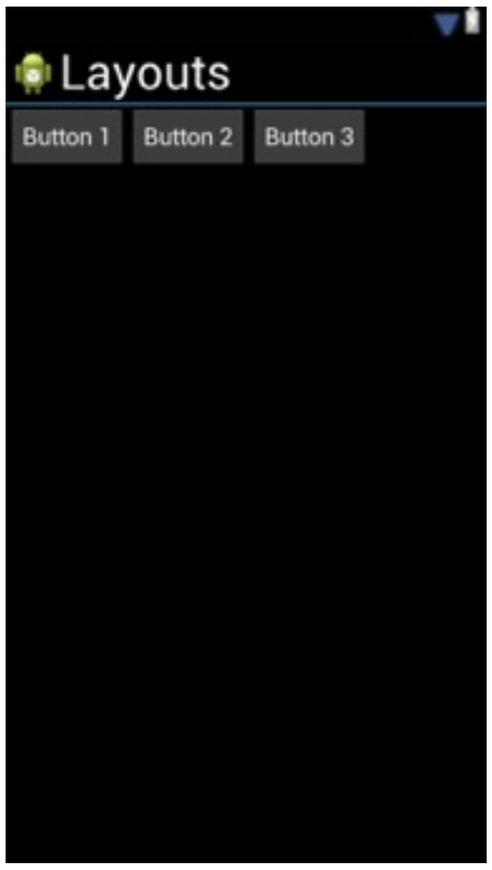
Lecture 2 – Android User Interface (Controls)

Overview

- Layouts
- Event "Listeners"
- Types of Android user controls
- Designing user interfaces in Eclipse
- How to use user controls in your code

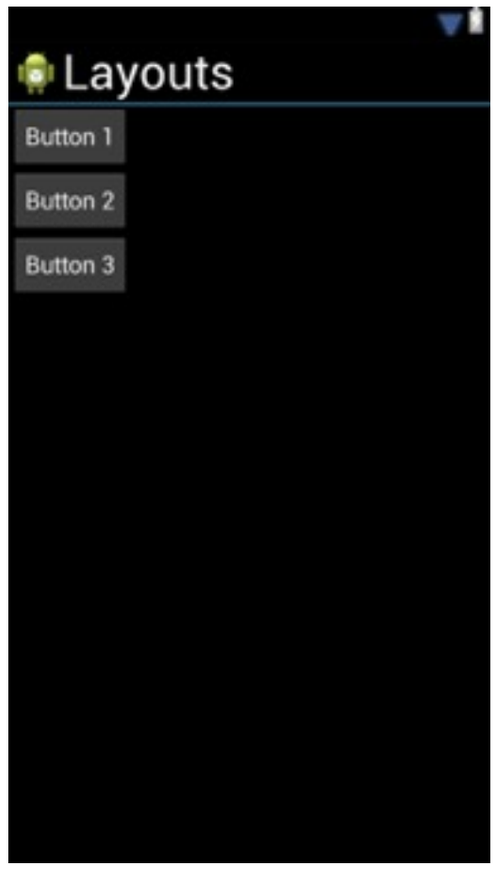
What is a layout?

LinearLayouts



Horizontal

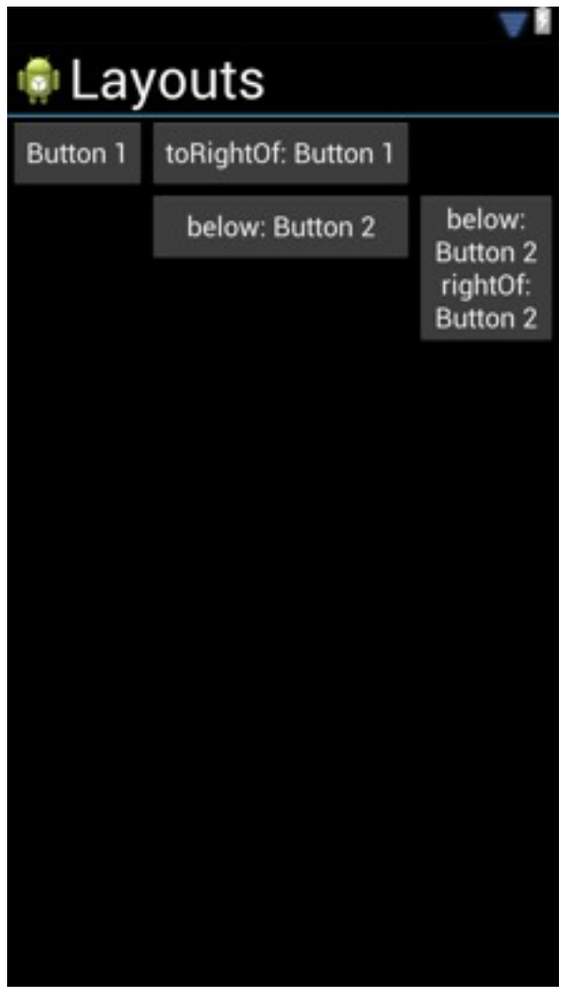
or



Vertical

Lay out controls in order,
may display scrollbar

RelativeLayouts



Lay out controls relative to each other

Event “Listeners”

Event Listeners

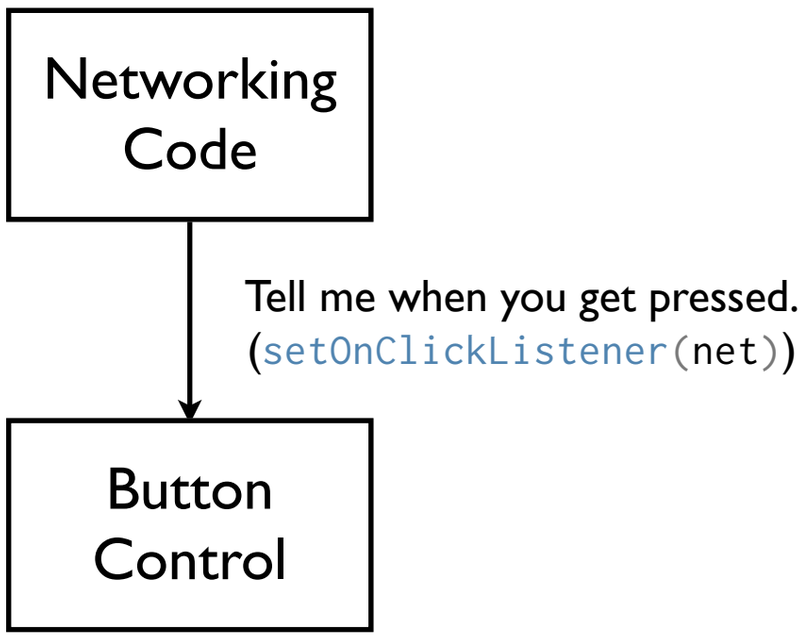
- Notify another object/call a method when an object does something
- May be used to communicate changes in state (e.g. when key pressed)
- Related Observer pattern notifies *multiple* objects

Networking Code

Button Control

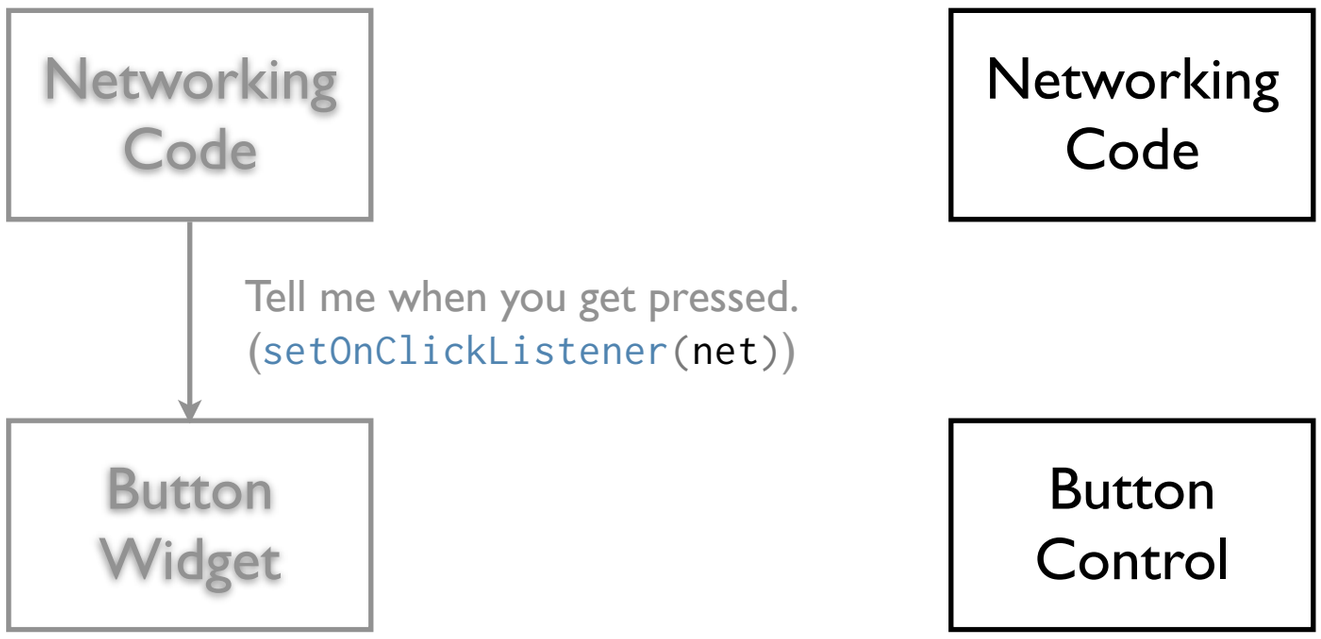
Event Listeners

- Notify another object/call a method when an object does something
- May be used to communicate changes in state (e.g. when key pressed)
- Related Observer pattern notifies *multiple* objects



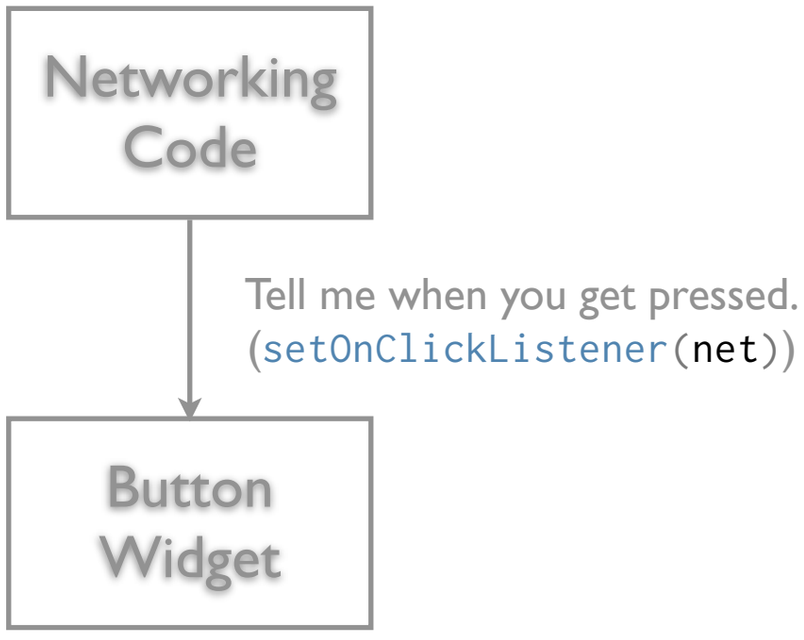
Event Listeners

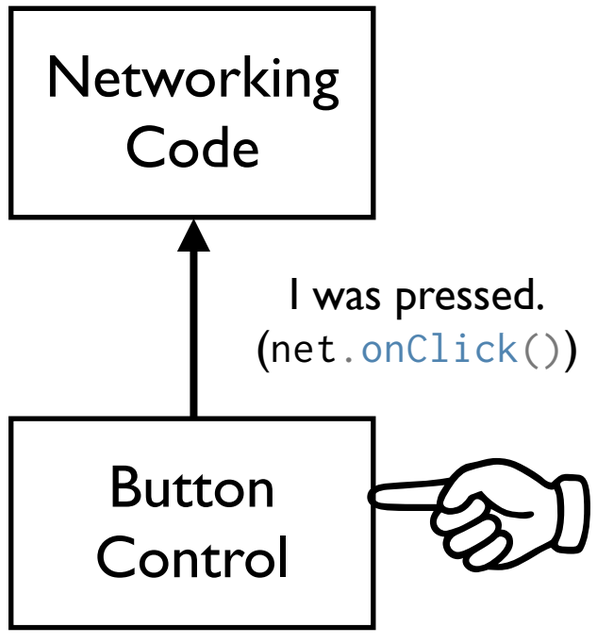
- Notify another object/call a method when an object does something
- May be used to communicate changes in state (e.g. when key pressed)
- Related Observer pattern notifies *multiple* objects



Event Listeners

- Notify another object/call a method when an object does something
- May be used to communicate changes in state (e.g. when key pressed)
- Related Observer pattern notifies *multiple* objects





Event Listeners

```
public interface OnClickListener {
    abstract void onClick(Position position);
}

public class Button {
    void setListener(OnClickListener x) { this.listener = x; }

    void click() {
        this.mousePosition = Mouse.readPosition();
        if (this.listener != null) {
            this.listener.onClick(this.mousePosition);
        }
    }
}
```

Event Listeners

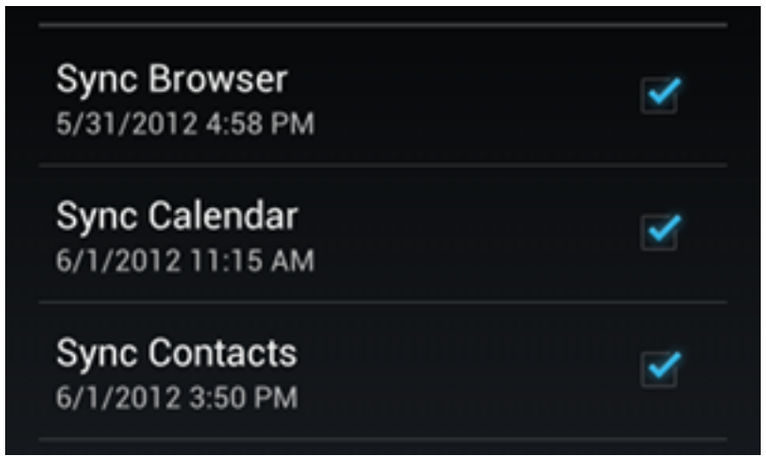
```
Button myButton = new Button();

myButton.setListener(new OnClickListener() {  // Anonymous!
    public void onClick(Position position) {
        System.out.println("Received click at "
                           + position.toString());
    }
});

// The following will print out "Received click at [position]"
myButton.click();
```

Note the anonymous class!

It could also be an instance of a concrete class implementing `OnClickListener`

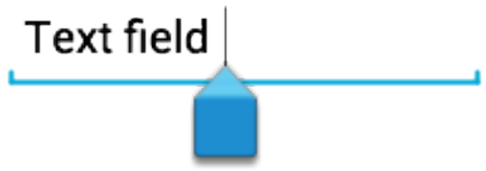
Types of user controls

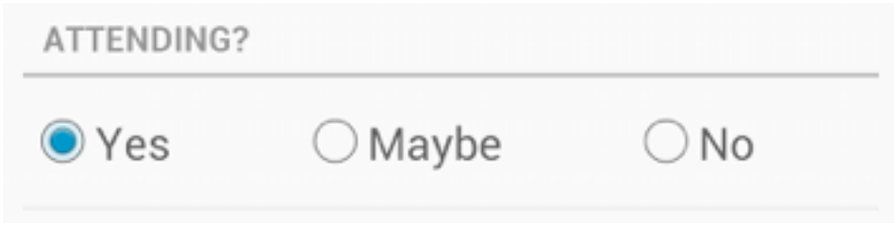
Buttons



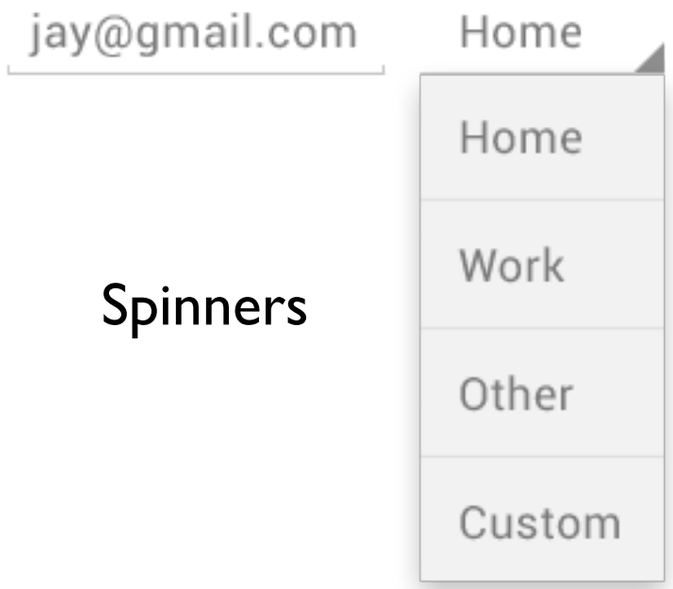
Checkboxes

Toggle Buttons





Radio Buttons



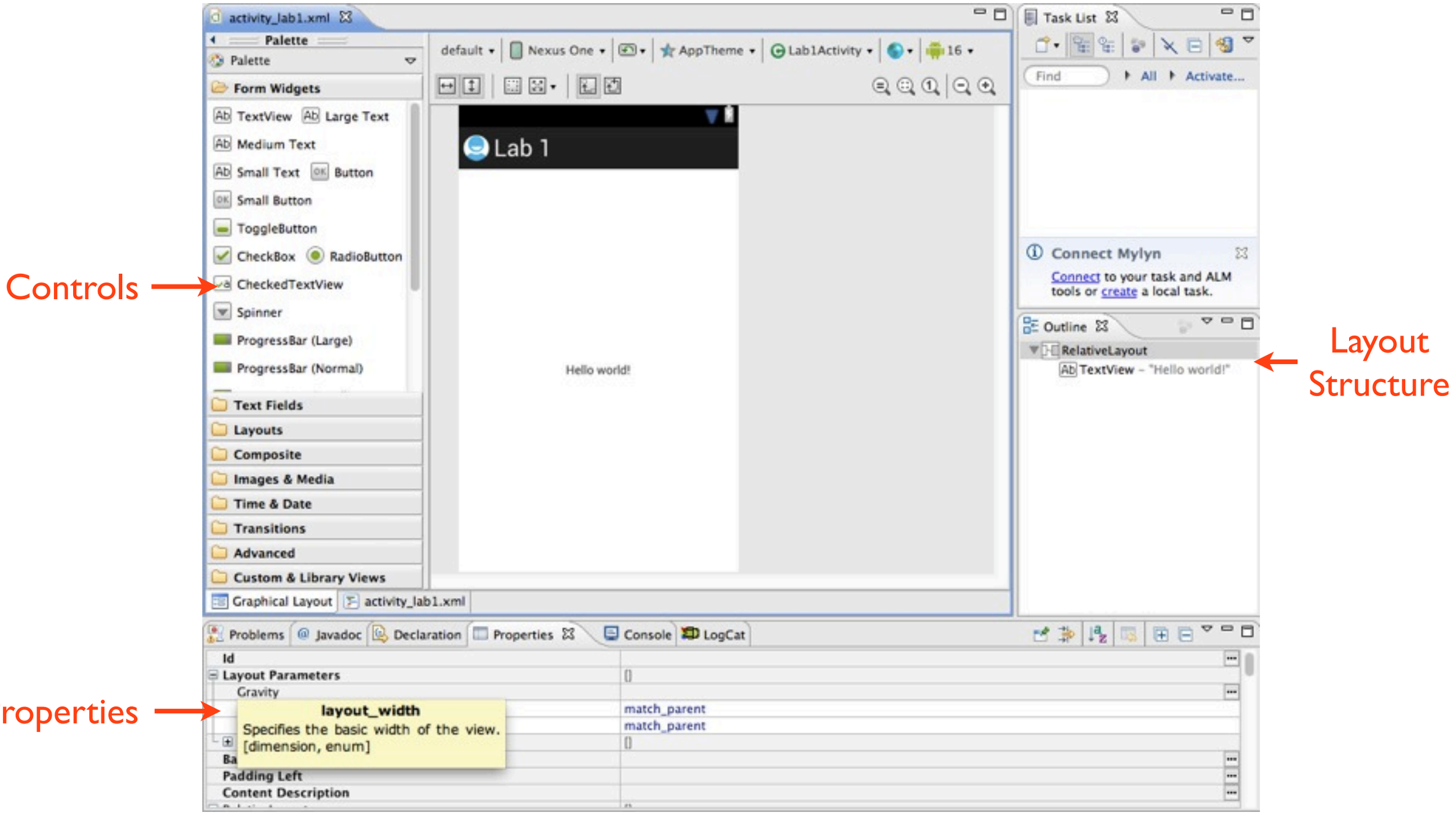
Spinners

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)

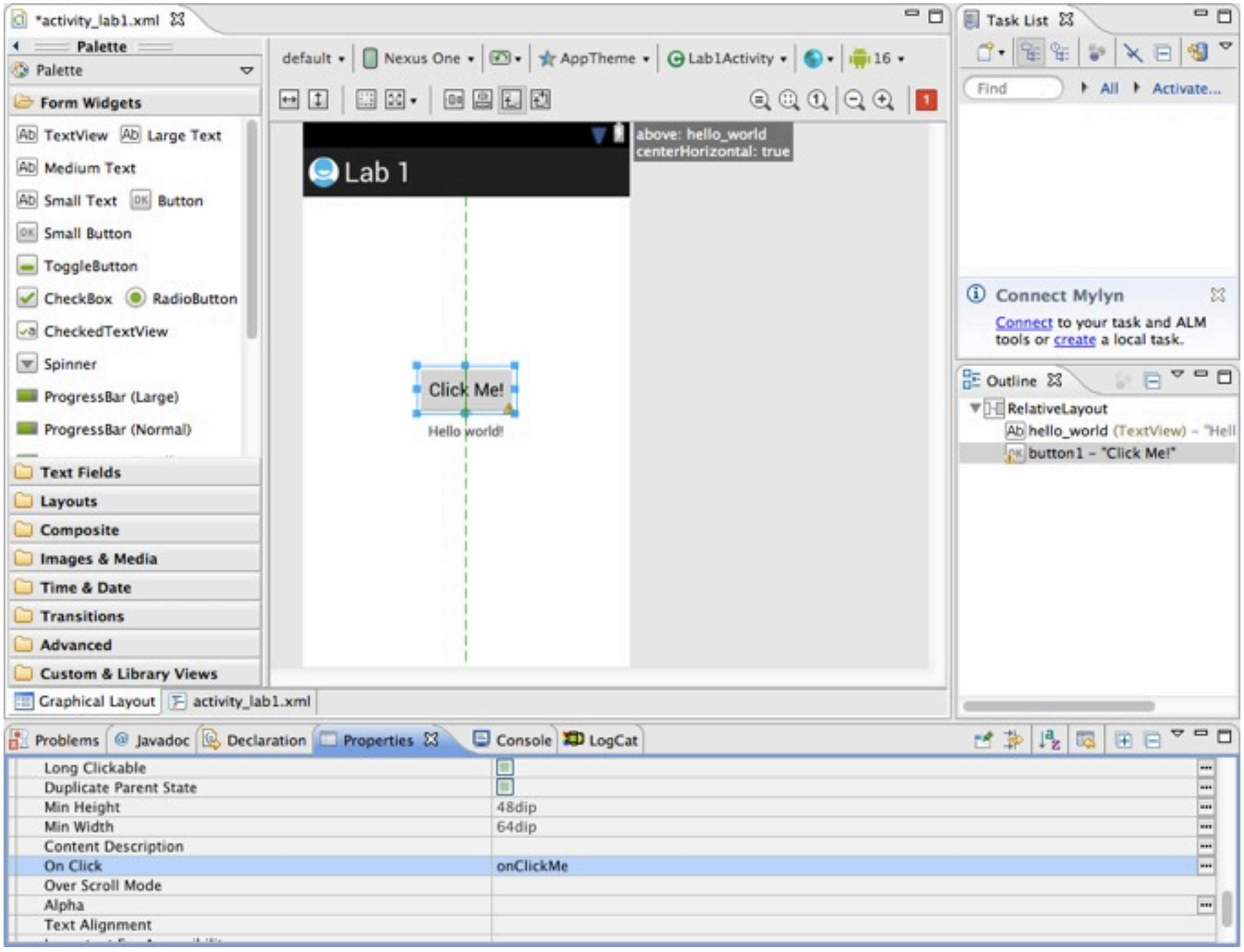
Making interfaces in Eclipse

Graphical Layout



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" />
    <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>
```

[name the view]

[name the view]

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>
```

[expand to fill parent]

[minimum needed size]

[horizontally and vertically center]

[minimum needed size]

[horizontally center above hello_world]

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>
```

[name of the method to call when clicked]

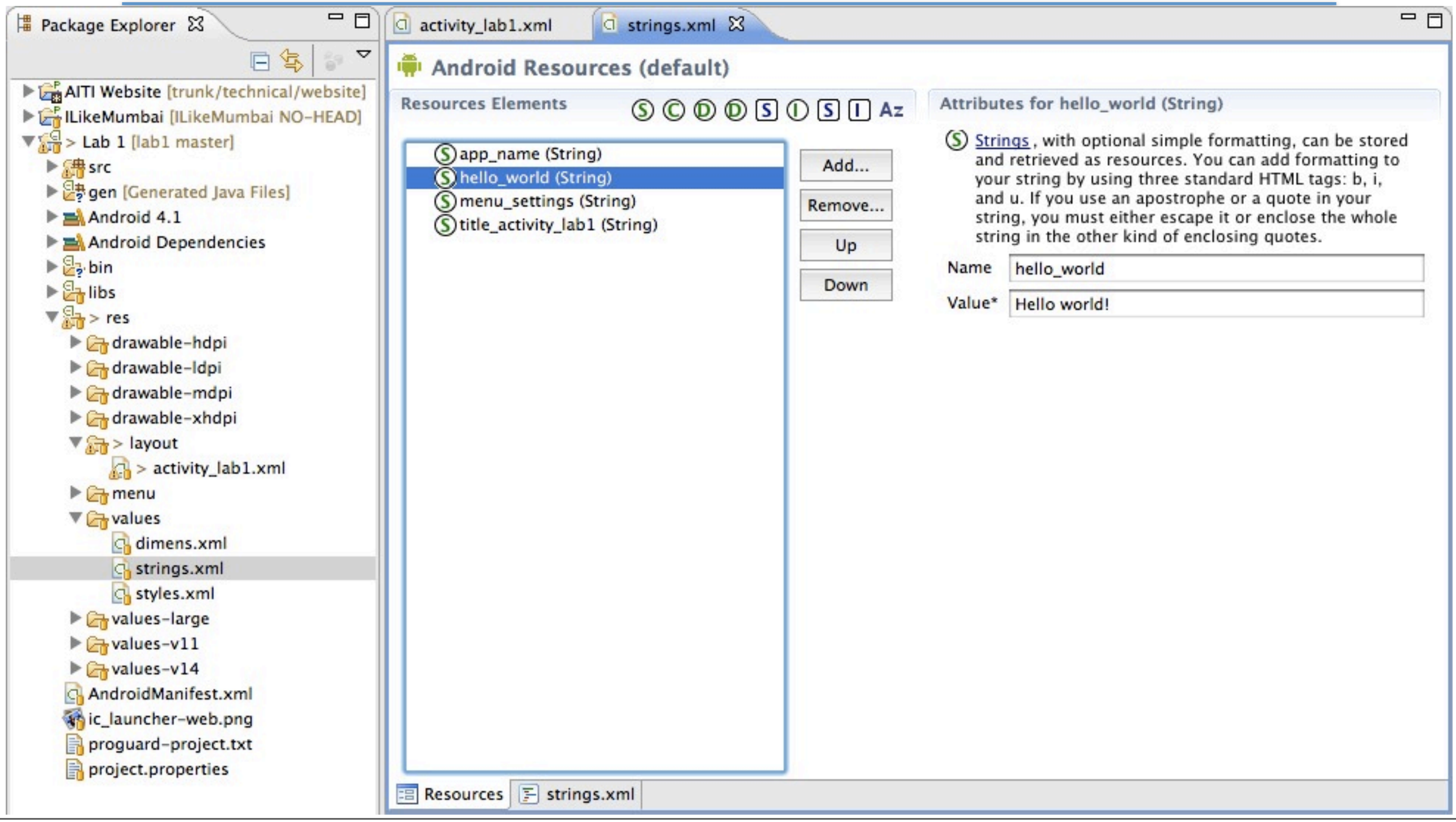
A Simple Button: Code View

```
<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>
```

[reference a string resource]

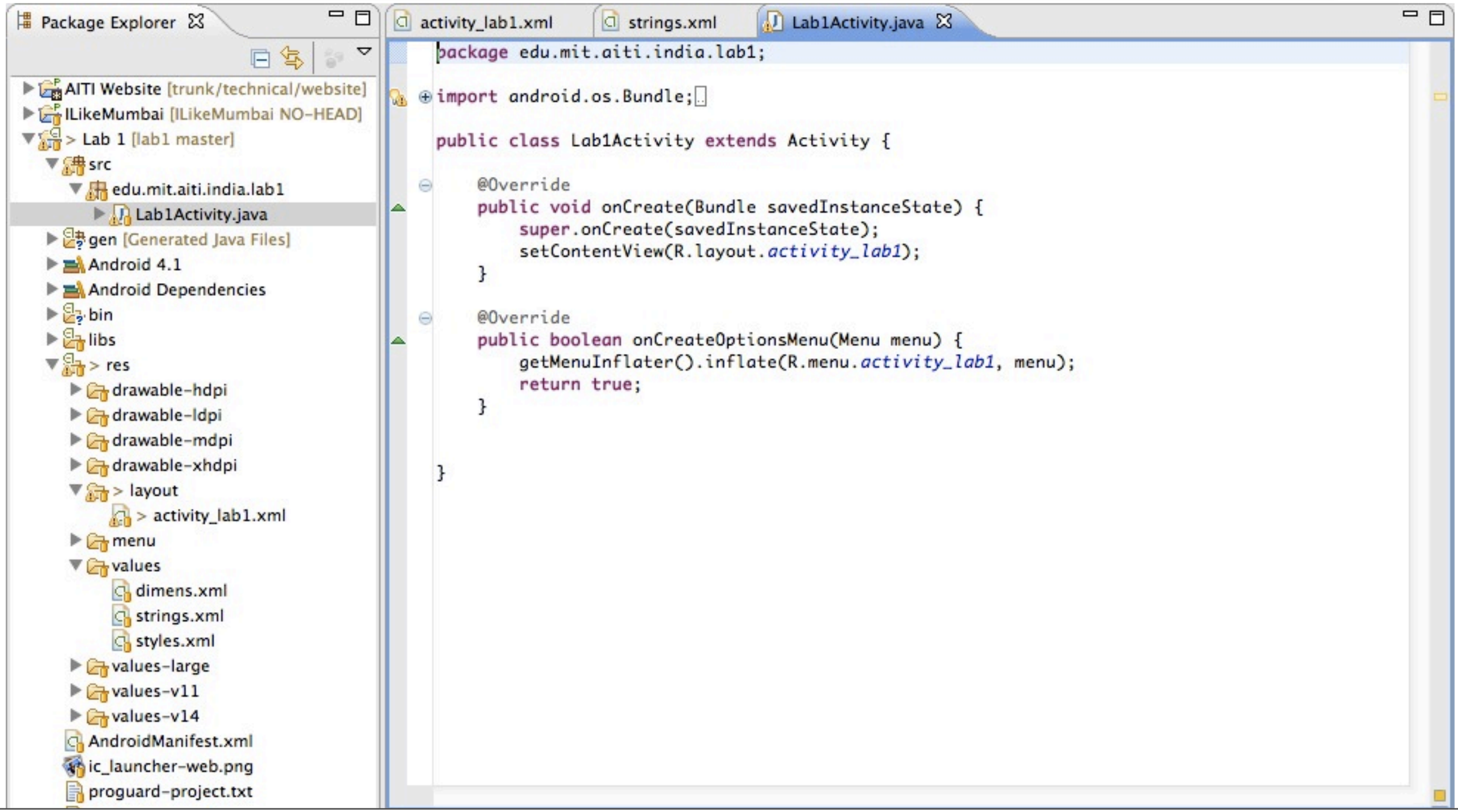
[literal string of the button]

String Resources

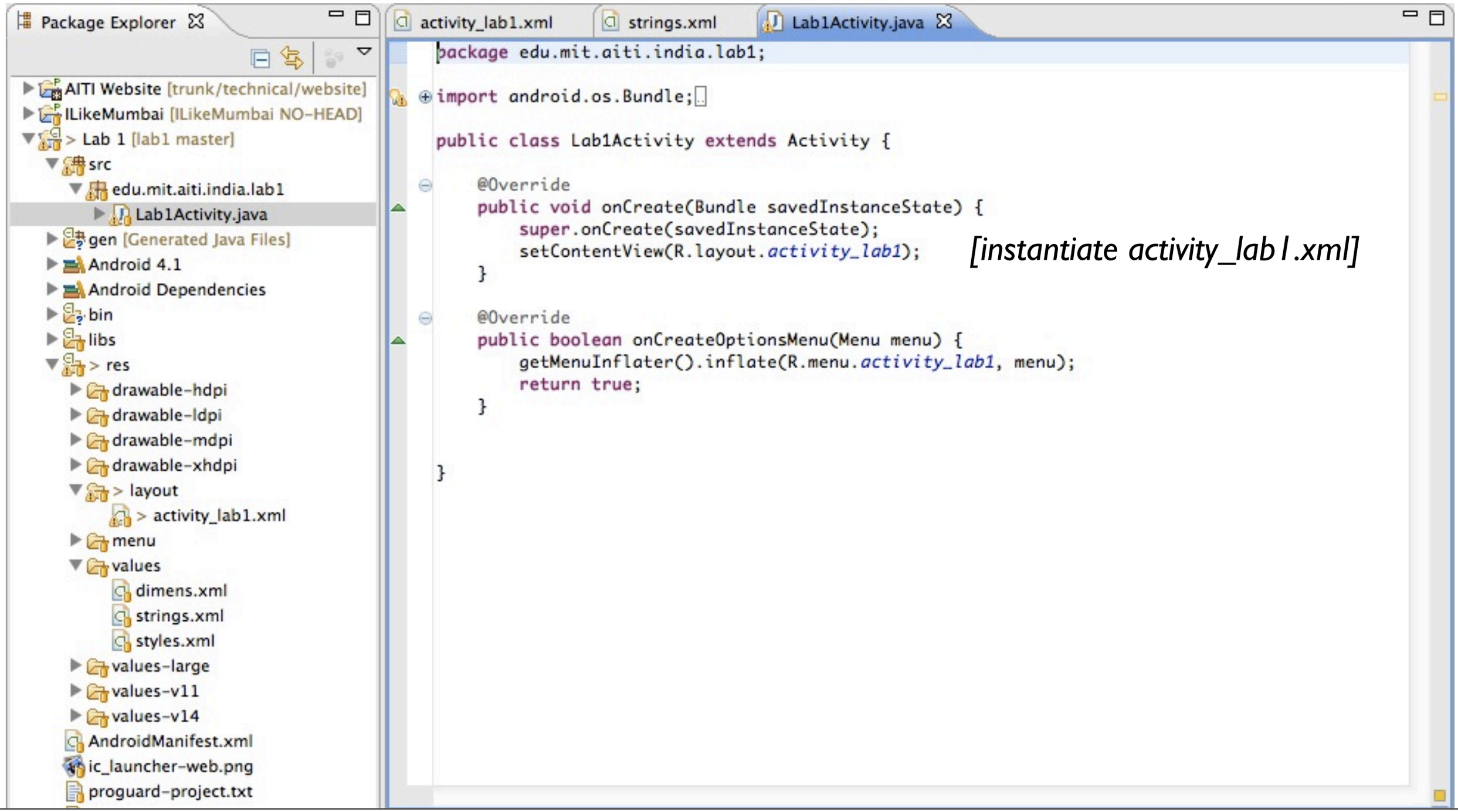


But what about the code?

The Code


Package Explorer
AITI Website [trunk/technical/website]
ILikeMumbai [ILikeMumbai NO-HEAD]
> Lab 1 [lab1 master]
src
edu.mit.aiti.india.lab1
Lab1Activity.java
gen [Generated Java Files]
Android 4.1
Android Dependencies
bin
libs
> res
drawable-hdpi
drawable-ldpi
drawable-mdpi
drawable-xhdpi
> layout
> activity_lab1.xml
menu
values
dimens.xml
strings.xml
styles.xml
values-large
values-v11
values-v14
AndroidManifest.xml
ic_launcher-web.png
proguard-project.txt
activity_lab1.xml
strings.xml
Lab1Activity.java
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



[instantiate activity_lab1.xml]

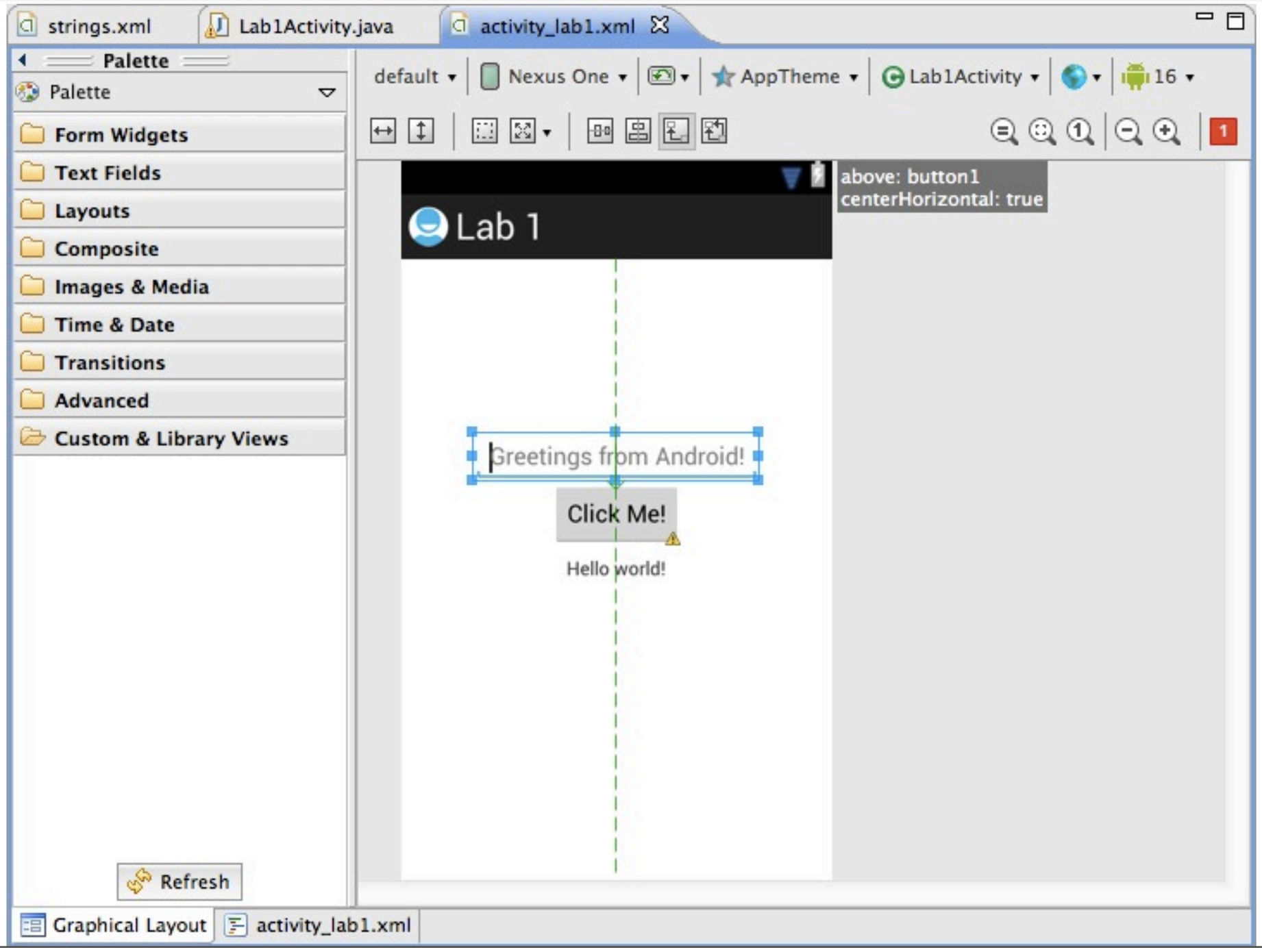
The Code:
onClickMe()

```
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));
}
```

A Simple Button: Demo

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>
```

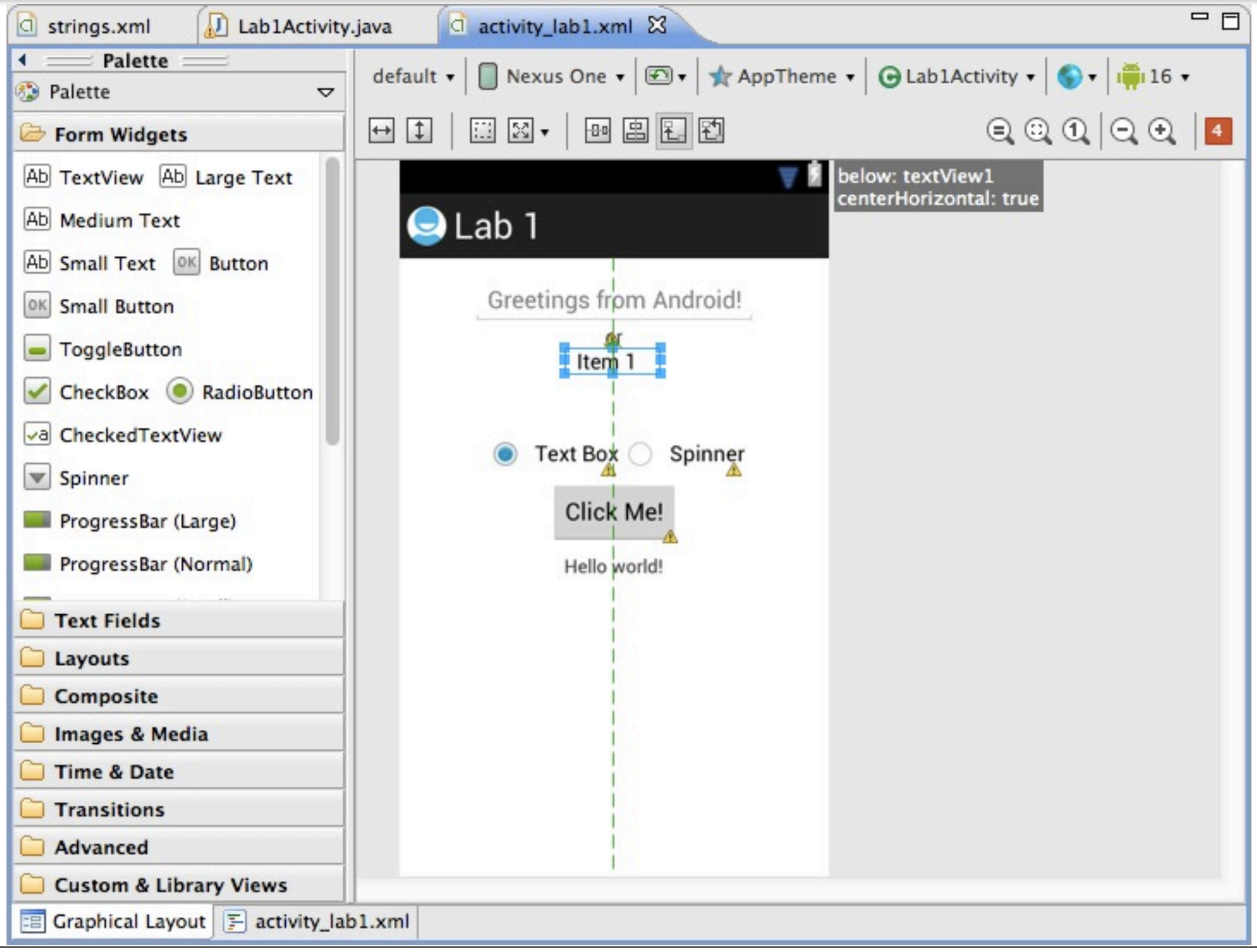
The Code:
onClickMe()

```
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



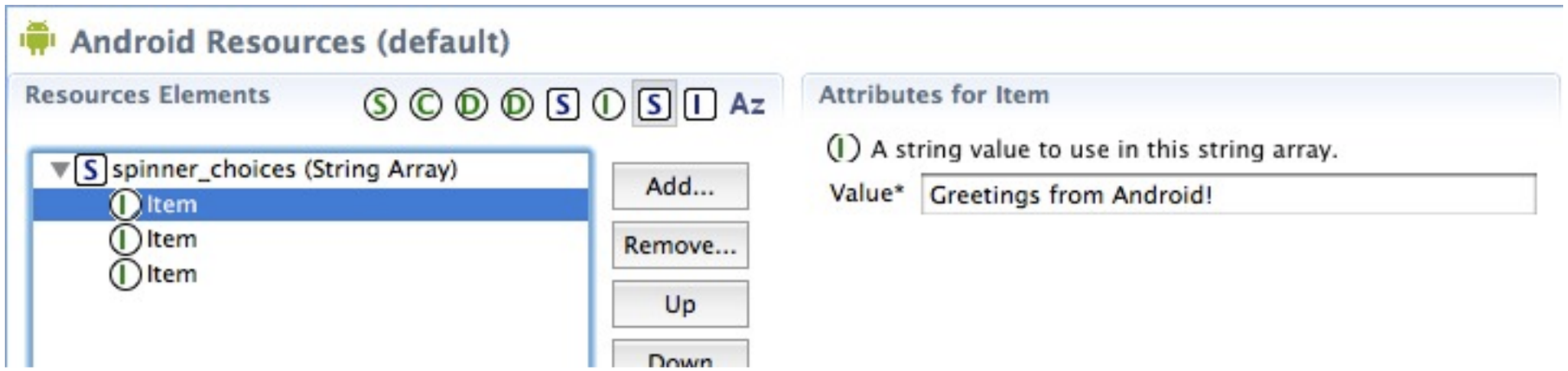
Radio Buttons and Spinners Code View

```
<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>

<Spinner android:id="@+id/spinner"
         android:entries="@array/spinner_choices" />
```

[This button is checked]

[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 >¡Saludos 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());
    }
}
```

Radio Buttons and Spinners Demo

Lab 1

<https://bitbucket.org/aiti_india/lab1>

Resources

- Android Developer Website "User Interface" <http://developer.android.com/guide/topics/ui/index.html>, especially:
  - "Layouts" <http://developer.android.com/guide/topics/ui/declaring-layout.html>
  - "Input Controls" <http://developer.android.com/guide/topics/ui/controls.html>
  - "Input Events" <http://developer.android.com/guide/topics/ui/ui-events.html>
- "Android User Interface Design" on MobileTuts <http://mobile.tutsplus.com/series/android-user-interface-design/>