

# EditText Field Conversion

Two EditText Objects

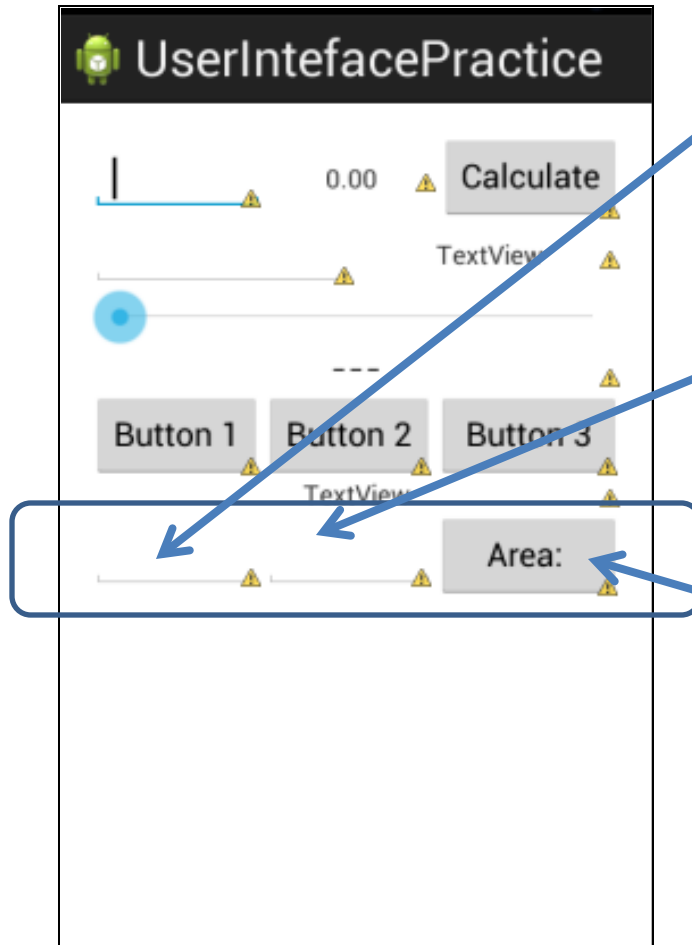
Button Object

Return Functions

Calculates and Displays the Area

Given User input of length and width

# EditText and Button XML



```
<EditText  
    android:id="@+id/editLength"  
    android:layout_weight = "0.33"  
    android:layout_height="wrap_content"  
/>  
  
<EditText  
    android:id="@+id/editWidth"  
    android:layout_weight = "0.33"  
    android:layout_height="wrap_content"  
/>  
  
<Button  
    android:id="@+id/buttonArea"  
    android:layout_weight = "0.33"  
    android:layout_height="wrap_content"  
    android:onClick = "areaButtonClick"  
    android:text="Area: " />
```

Remember: This is step 1; Create the Objects in XML for the User Interface

# Fields: Above onCreate()

```
34
35     // Fields for Area Formula
36     private EditText editLength;
37     private EditText editWidth;
38     private Button  buttonArea;
39
```

Remember: This is step 2; Create the Java fields in the Main Activity Class. The names of the objects (in blue and lower case) should match the XML object names to reduce confusion.

# Initialize and Bind to XML in onCreate()

```
67  
68     // Bind Area Formula objects to XML  
69     editLength = (EditText) findViewById(R.id.editLength);  
70     editWidth = (EditText) findViewById(R.id.editWidth);  
71     buttonArea = (Button) findViewById(R.id.buttonArea);  
72
```

Remember: This is step 3; Initialize and Bind the Java objects to the XML objects. The 'findViewById' command connects the Java object to the XML. Usually the Binding happens in the onCreate(). (Though, it can occur in other functions).

# Return Function that performs the Area Calculation

```
167
168     // Rectangle Area Function
169     public int area(int length, int width) {
170         int a = length * width;
171         return a;
172     }
173
```

Note: Additional functions in the Main Activity class are defined below the onCreate() function. The above example is a 'return' function that passes a data value back to the program. (In this example, it is an integer).

# Function to React to Button Click

```
173
174 // Function for the buttonArea
175 public void areaButtonClick(View v) {
176     Editable len = editLength.getText(); // Get Value from EditText
177     String length = String.valueOf(len); // Convert to a String
178     int l = Integer.valueOf(length); // Convert to an Integer
179
180     Editable wid = editWidth.getText(); // Get Value from EditText
181     String width = String.valueOf(wid); // Convert to a String
182     int w = Integer.valueOf(width); // Convert to an Integer
183
184     int a = area(l, w); // Call the function
185     buttonArea.setText(String.valueOf(a)); // Set the text
186 }
187
```

Note that the name of the function 'areaButtonClick' matches the android:onClick of the buttonArea object. This function is called when the buttonArea object is clicked. Remember, any function called by a button must have the (View v) in the parameter of the function.