

# CHAPTER 3

## Input, Variables, and Calculations

starting out with >>> **APP INVENTOR**  
FOR ANDROID



TONY GADDIS · REBECCA HALSEY

# Topics

- The *TextBox* Component
- Performing Calculations
- Storing Data with Variables
- Creating Blocks with Typeblocking
- The *Slider* Component
- Math Functions

# The `TextBox` Component

- The `TextBox` component is a rectangular area that can display text, and can also accept keyboard input.
- In the Designer, the `TextBox` is located in the User Interface section of the Palette.
- `TextBox` components are automatically given default names such as `TextBox1`.
- It is a good idea to change a component's default name to something meaningful.
- When the user types into a `TextBox` component, the text is stored in the component's `Text` property.

# The TextBox Component

Figure 3-1 Shows a screen from the example project. This is a summary of its components:

- *TableArrangement1* – A *TableArrangement* with one row and two columns.
- *Label1EnterYourName* – A label that displays the text *Enter your name:.*
- *TextBoxName* – A *TextBox* component for the user to enter his or her name.

# The TextBox Component

Figure 3-1 Shows a screen from the example project. This is a summary of its components:

- *ButtonReadInput* – A *Button* component that, when clicked, reads input that the user typed into the *TextBox* component, and displays the text in the *LabelOutput* component.
- *TableArrangement2* – A *TableArrangement* with one row and two columns.

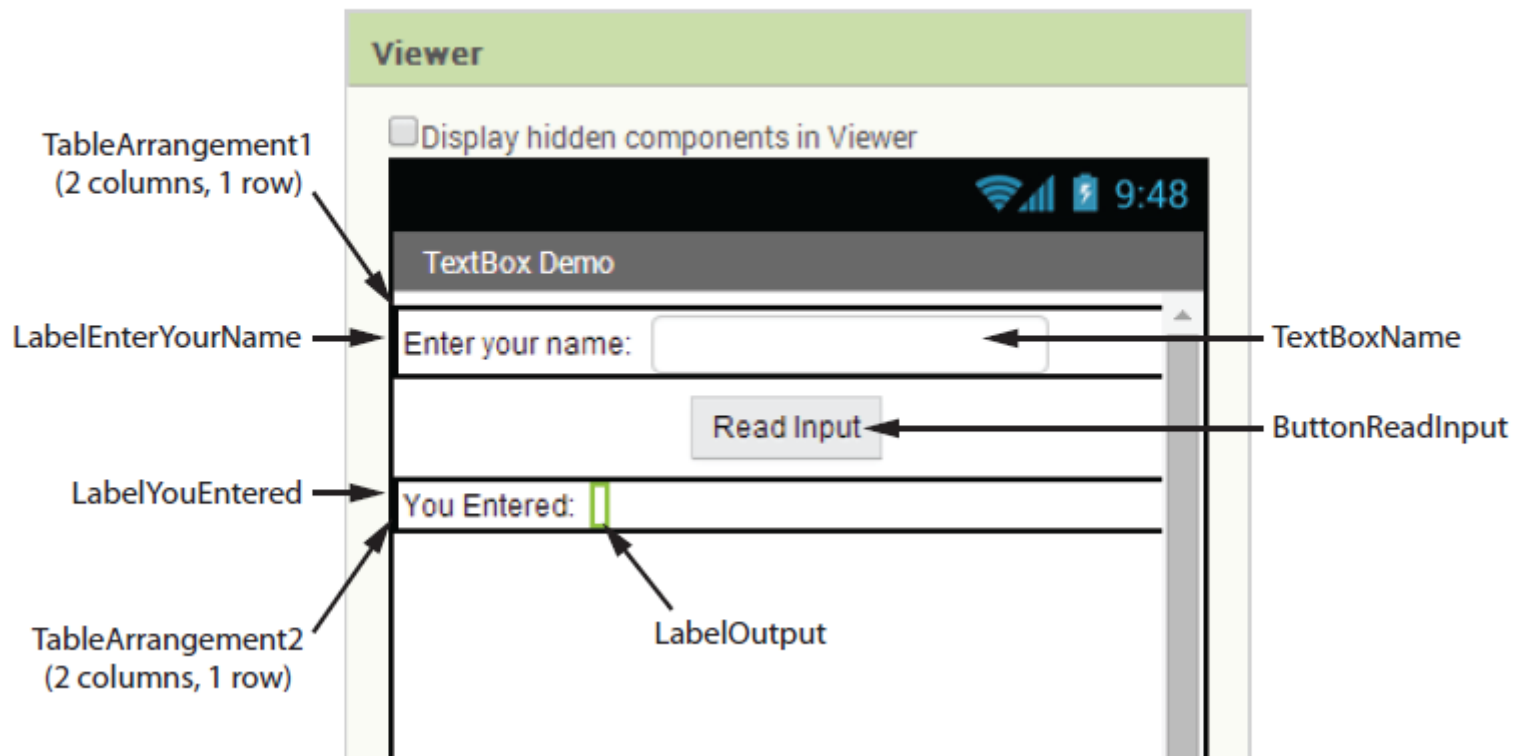
# The TextBox Component

Figure 3-1 Shows a screen from the example project. This is a summary of its components:

- *LabelYouEntered* – A Label that displays the text *You entered*.
- *LabelOutput* – A component that initially displays nothing when the user clicks the *ButtonReadInput* component. The text that the user entered into the *TextBox* name component is displayed in this label.

# The TextBox Component

**Figure 3-1** Example Project Using a TextBox Component (Source: MIT App Inventor 2)



# The TextBox Component

When the user clicks a *TextBox*, the emulator's virtual keyboard pops up on the screen.

Figure 3-2 The Example App Running in the Emulator (Source: MIT App Inventor 2)





# The TextBox Component

**Figure 3-3** The ButtonReadInput Click Event Handler (Source: MIT App Inventor 2)



The the *Click* event handler for the *ButtonReadInput* component is shown (Figure 3-3).

The blocks inside the event handler set the *LabelOutput* component's *Text* property.

# The TextBox Component

Figure 3-4 shows the app running in the emulator after the user has entered *Kathryn Smith*.

**Figure 3-4** The App after the User has Entered Input and Clicked the Button  
(Source: MIT App Inventor 2)



# The `TextBox` Component

## Other `TextBox` Properties

- *BackgroundColor* – Sets the `TextBox`'s background color.
- *Enabled* – If checked, the user is able to enter input into the `TextBox`.
- *FontBold*, *FontItalic*, and *FontSize* – Affect the font of the text displayed in the `TextBox`.
- *Hint* – Displays a hint for the user.
- *MultiLine* – If checked, the `TextBox` will allow the user to enter multiple lines of input.
- *NumbersOnly* – If checked, `TextBox` will only allow numbers to be entered.

# The `TextBox` Component

## Other `TextBox` Properties

- *TextAlignment* – Specifies how the text inside the `TextBox` is aligned. It may be set to *left*, *center*, or *right*.
- *TextColor* – Sets the color of the text displayed in the `TextBox`.
- *Visible* – Specifies whether the component is visible on the screen or hidden.
- *Width* and *Height* – Determines the control's width and height. May be set to *Automatic*, *Fill parent*, or a specific number of pixels.

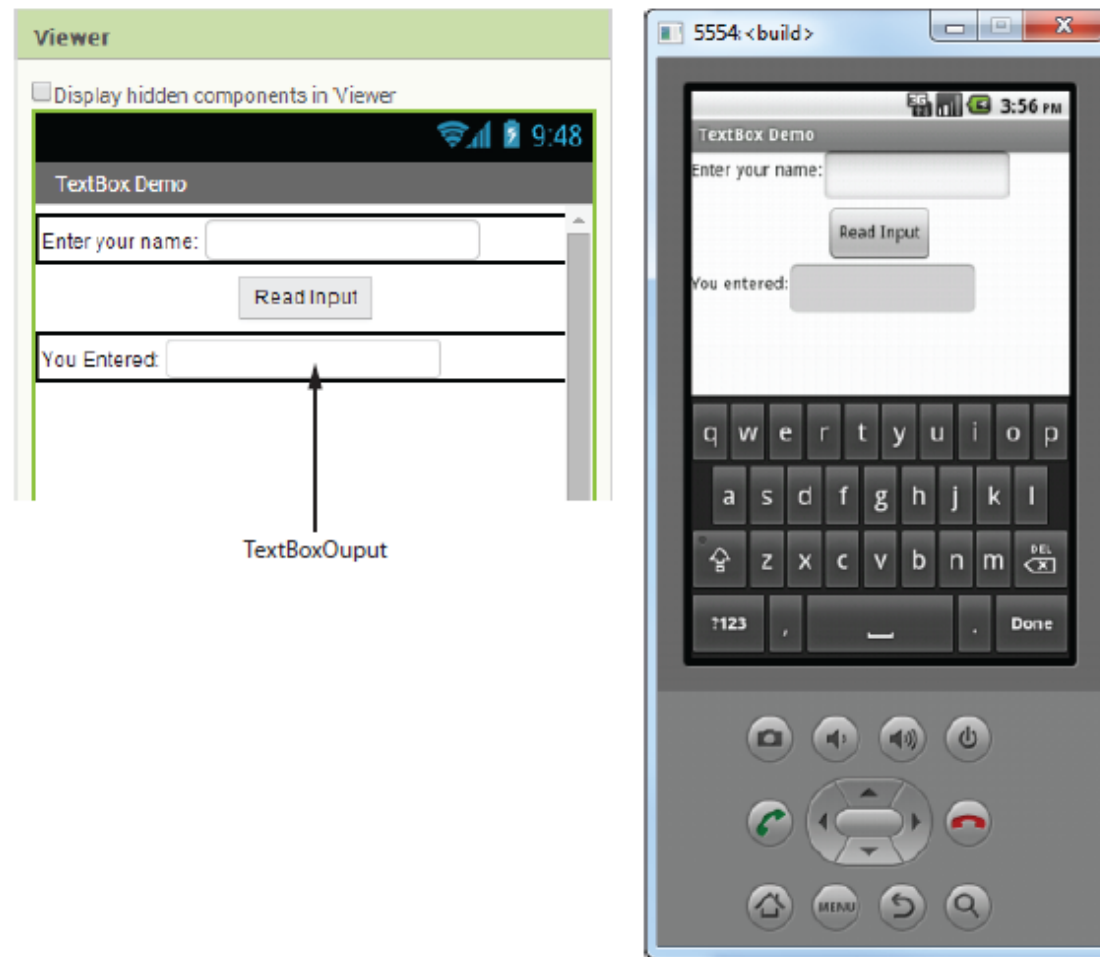
# The `TextBox` Component

## Using `TextBox` Components to Display Text

- `TextBox` components can also be used to display text.
- In Figure 3-5 the `TextBox` component appears clearly on the screen as a rectangular area.
- Sometimes it is helpful to the user to see the area on the screen where the output will be displayed.
- When using `TextBox` to display text (and not read input), it is a good idea to uncheck the component's `Enabled` property. That prevents the user from selecting it and entering input.

# The TextBox Component

**Figure 3-5** The Modified TextBoxDemo Project (Source: MIT App Inventor 2)



# The TextBox Component

## Using *TextBox* Components to Display Text

- If the *TextBoxDemo* displays its output in a *TextBox* instead of a *Label*, we need to modify the *Click* event handler for the *ButtonReadInput* component.
- Figure 3-6 shows the new event handler.

**Figure 3-6** The Modified Click Event Handler for the *ButtonReadInput* Component (Source: MIT App Inventor 2)

