

CHAPTER 3

Input, Variables, and Calculations

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

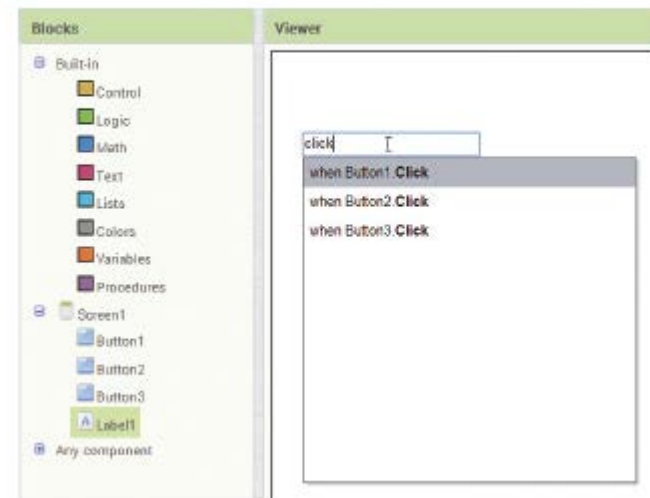


TONY GADDIS · REBECCA HALSEY

Creating Blocks with Typeblocking

- *Typeblocking* is a shortcut method for quickly creating blocks using the keyboard.
- In the Blocks Editor click anywhere in the workspace and type part of the name of the block that you want to create.

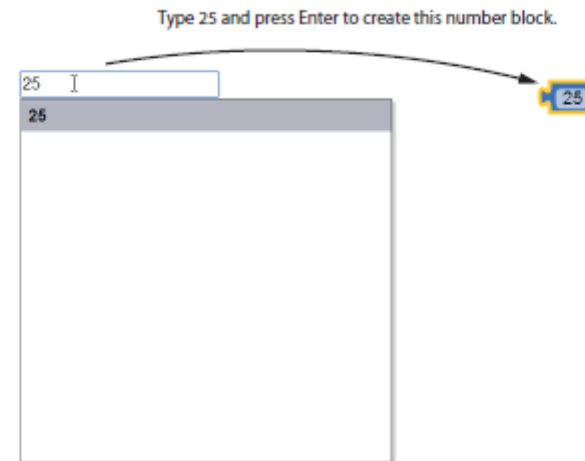
Figure 3-86 Creating a Click Event Handler with Typeblocking
(Source: MIT App Inventor 2)



Creating Blocks with Typeblocking

- Use Typeblocking to quickly create number blocks and text string blocks.
- Suppose you want to create a number block with the value 25.
- Click inside the workspace, type 25, press *Enter*.

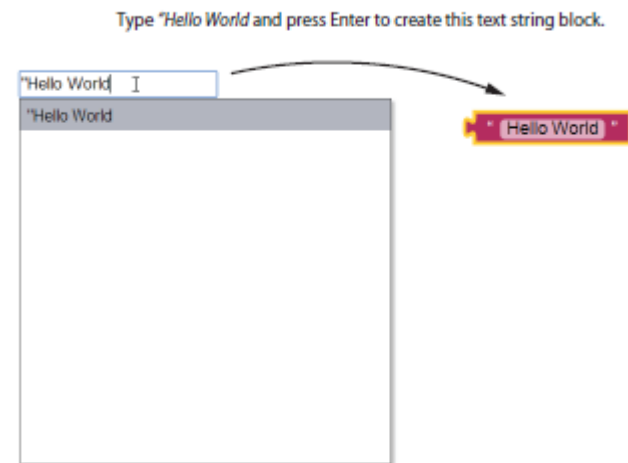
Figure 3-87 Creating a Number Block with Typeblocking (Source: MIT App Inventor 2)



Creating Blocks with Typeblocking

- To create a text string block, click inside the workspace, type a quotation mark, type the text you want to set as the block's value, press *Enter*.
- Do not type and ending quotation mark.

Figure 3-88 Creating a Text String Block with Typeblocking (Source: MIT App Inventor 2)



The Slider Component

The *Slider* component provides a visual way to adjust a value within a range of values.

In the Designer, you will find it in the Basic Palette.

Figure 3-89 A Slider Component (Source: MIT App Inventor 2)



- The *Slider* component has a *MinValue* property, and a *MaxValue* property that must be set to numeric values.
- By default the *MinValue* property is set to 10.0.
- By default *MaxValue* property is set to 50.0.

The Slider Component

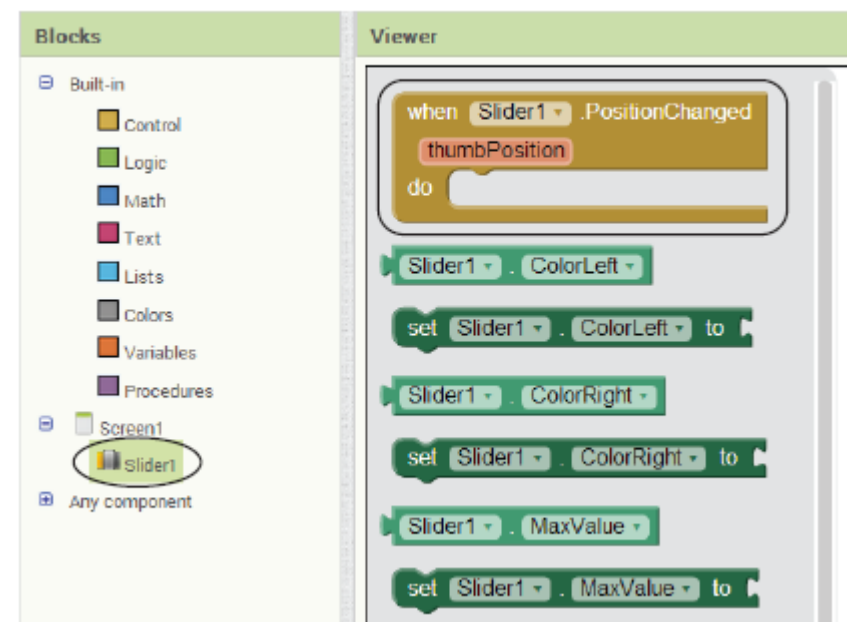
Here is a summary of the Slider components properties:

- *ColorLeft* – Specifies the color of the part of the horizontal track that is to the left of the thumb slider.
- *ColorRight* – Specifies the color of the part of the horizontal track that is to the right of the thumb slider.
- *MaxValue* – The *Slider* component's maximum value.
- *MinValue* – The *Slider* component's minimum value.
- *ThumbPosition* – The position of the thumb slider.
- *Visible* – Determines whether the component is visible on the screen.
- *Width* – The width of the component. It can be set to *Automatic*, *Fill parent*, or a specific number of pixels.

The Slider Component

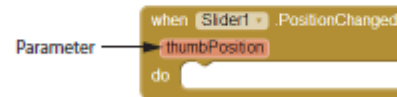
In the Blocks Editor, open the *Slider*, then select the block for the *PositionChanged* event handler.

Figure 3-90 Creating a PositionChanged Event Handler (Source: MIT App Inventor 2)



The Slider Component

Figure 3-91 The Slider Component's `PositionChanged` Event Handler
(Source: MIT App Inventor 2)

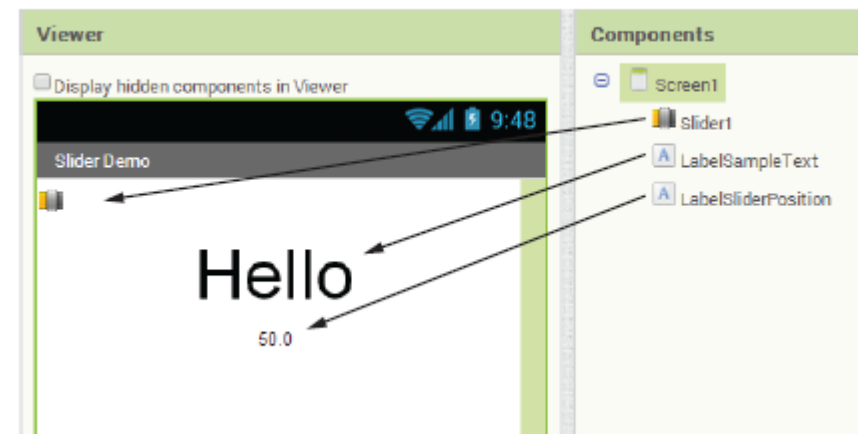


- Figure 3-91 shows an empty *PositionChanged* event handler.
- *thumbPosition* is a local variable known as a parameter variable. A parameter variable holds pieces of data passed to an event handler.
- The scope of the *thumbPosition* parameter variable is the *PositionChanged* event handler.
- Inside the *PositionChanged* event handler you can use, *get* and *set* blocks.

The Slider Component

Figure 3-92 shows the screen from the *SliderDemo* project.

Figure 3-92 The SliderDemo Project (Source: MIT App Inventor 2)



The Slider Component

Table 3-7 Component property settings (Source: Pearson Education, Inc.)

Component	Relevant Property Settings
Screen1	AlignHorizontal = <i>Center</i>
	Title = <i>Slider Demo</i>
Slider1	MaxValue = 100
	MinValue = 0
	ThumbPosition = 50
	Width = <i>Fill parent</i>
LabelSampleText	Text = <i>Hello</i>
	FontSize = 50
LabelSliderPosition	Text = <i>50.0</i>

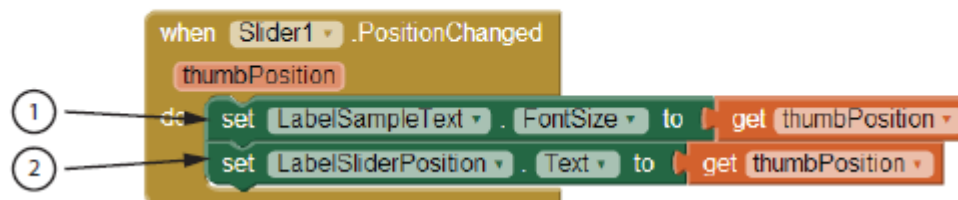
The Slider Component

Figure 3-95 shows the *Slider1.PositionChanged* event handler.

1. The first set of blocks sets the *LabelSampleText* component's *FontSize* to the value of the *thumbPosition* variable.
2. The second set of blocks sets the *LabelSliderPosition* component's *Text* property to the value of the *thumbPosition* variable.

Figure 3-95 The *Slider1.PositionChanged* Event Handler

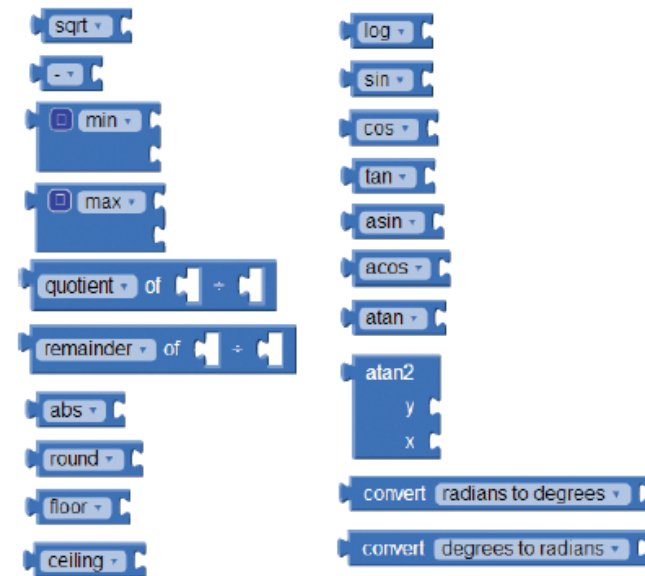
(Source: MIT App Inventor 2)



Math Functions

App Inventor provides numerous advanced math functions for complex calculations.

Figure 3-96 Math Functions (Source: MIT App Inventor 2)



Math Functions

Figure 3-97 shows an example use of the *sqrt* block.

Figure 3-98 shows another example. It sets the variable *MyVar* to the remainder of the 17 divided by 2.

Figure 3-97 Using the `sqrt` Function (Source: MIT App Inventor 2)



Figure 3-98 Using the remainder Function (Source: MIT App Inventor 2)

