

CHAPTER 4

Decision Blocks and Boolean Logic

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



TONY GADDIS · REBECCA HALSEY

Topics

- Introduction to Decision Blocks
- Relational Operators and the *if* Block
- The *if then else* block
- A First Look at Comparing Strings
- Logical Operators
- Nested Decision Blocks
- The *if then else if* block
- Working with Random Numbers
- The *Screen's Initialize* Event
- The *ListPicker* Component
- The *Checkbox* Component

Introduction to Decision Blocks

- Sometimes a program needs to "decide" whether or not to execute certain instructions. App Inventor provides three blocks for making decisions.
- So far you have worked with number values and text (string) values.

Introduction to Decision Blocks

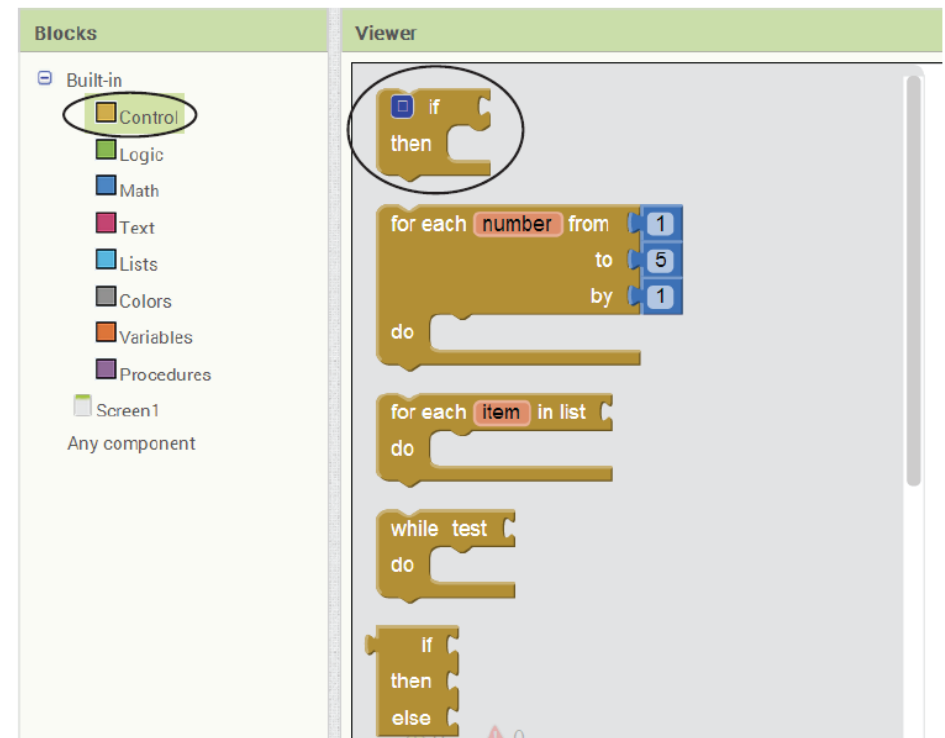
- Programs can also work with the values *true* and *false*.
- *true* and *false*, are known as *Boolean* values.
- *true* and *false* values are commonly used in decision making.

Introduction to Decision Blocks

The *if then* Block

App Inventor provides the *if then* block for making decisions.

Figure 4-1 The Decision Blocks (Source: MIT App Inventor 2)



Introduction to Decision Blocks

The *if then* Block

In Figure 4-2, notice that the *if then block* has two sockets: one for the *if* part, one for the *then* part.

Figure 4-2 The *if then* Block (Source: MIT App Inventor 2)

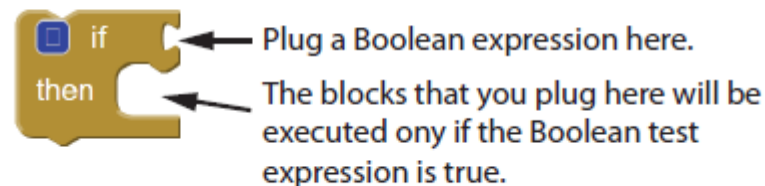
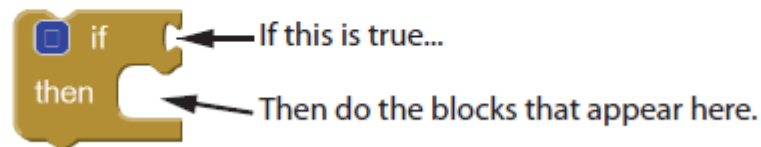


Figure 4-3 How to Think About the *if then* Block (Source: MIT App Inventor 2)

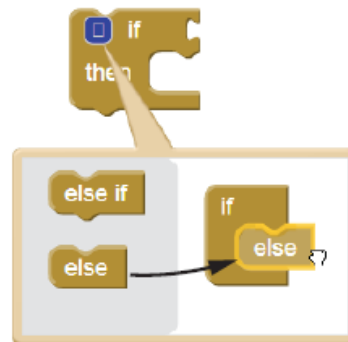


Introduction to Decision Blocks

The *if then else* Block

- The *if then* block is a mutator block.
- A mutator block has the ability to change in some way.
- Click the blue box that appears in the upper-left corner and the mutator bubble will appear.
- You can change the *if then* block to an *if then else* block.

Figure 4-5 Changing the *if then* Block to an *if then else* Block
(Source: MIT App Inventor 2)

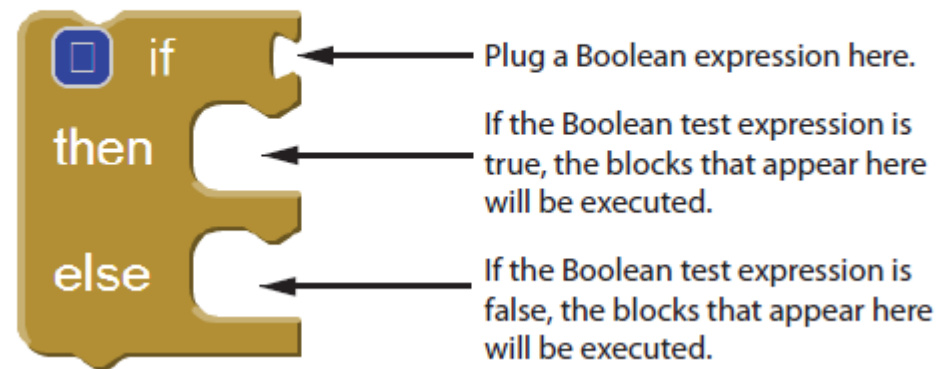


Introduction to Decision Blocks

The *if then else* Block

If the *Boolean* expression is *true*, the instructions that appear in the *then* sockets will be executed. If the *Boolean* expression is *false*, instructions that appear in the *else* socket will be executed.

Figure 4-6 The *if then else* Block (Source: MIT App Inventor 2)

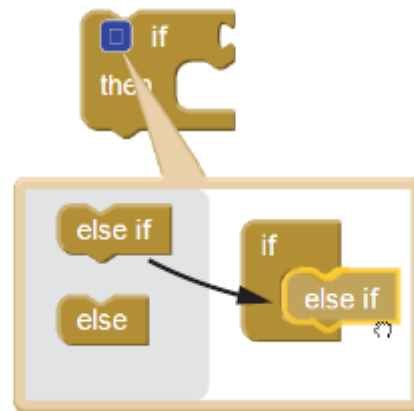


Introduction to Decision Blocks

The *if then else if* Block

Use the *if then* blocks mutator bubble to change the block into an *if then else if* block.

Figure 4-8 Changing an *if then* Block to an *if then else if* Block
(Source: MIT App Inventor 2)

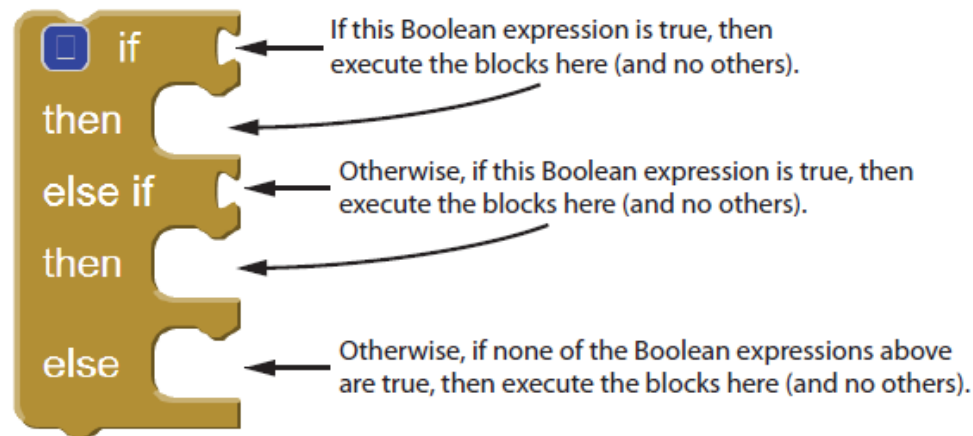


Introduction to Decision Blocks

The *if then else if* Block

- Drag the *else* block from the left side then insert it on the right side of the bubble.
- This creates an *if then else if* block like the one shown in Figure 4-10.

Figure 4-10 An *if then else if* Block (Source: MIT App Inventor 2)

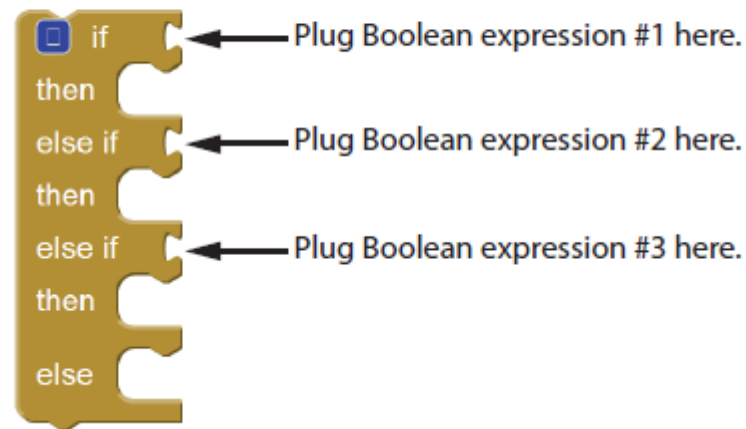


Introduction to Decision Blocks

The *if then else if* Block

Use the mutator bubble to add as many *else if* sections as you need.

Figure 4-11 An *if then else if* Block that Can Test Three Boolean Expressions
(Source: MIT App Inventor 2)

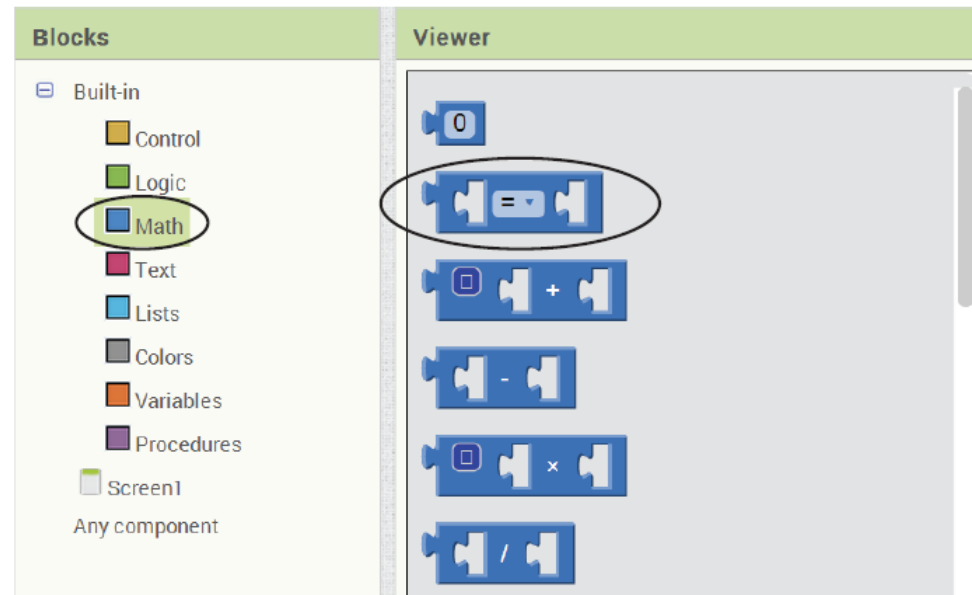


Relational Operators and the `if` Block

A *relational operator* determines whether a specific relationship exists between two values.

Access the relational operators in the *Math* drawer.

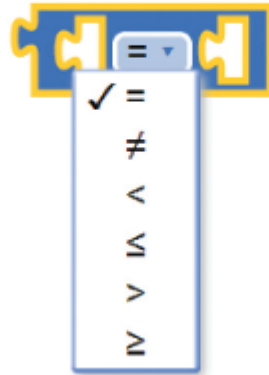
Figure 4-12 The Relational Operator Blocks (Source: MIT App Inventor 2)



Relational Operators and the if Block

Change it to any other relational operator by clicking on the down arrow (▼) .

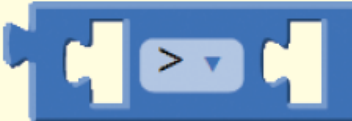
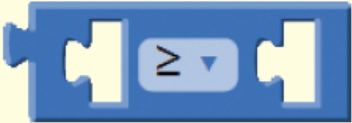

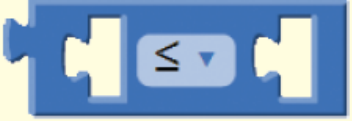
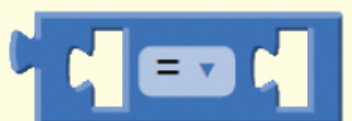
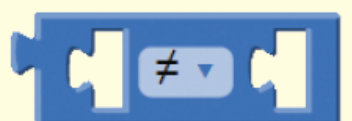
Figure 4-13 The Relational Operator Dropdown Menu (Source: MIT App Inventor 2)



Relational Operators and the if Block

Table 4-1 summarizes each of the relational operator blocks.

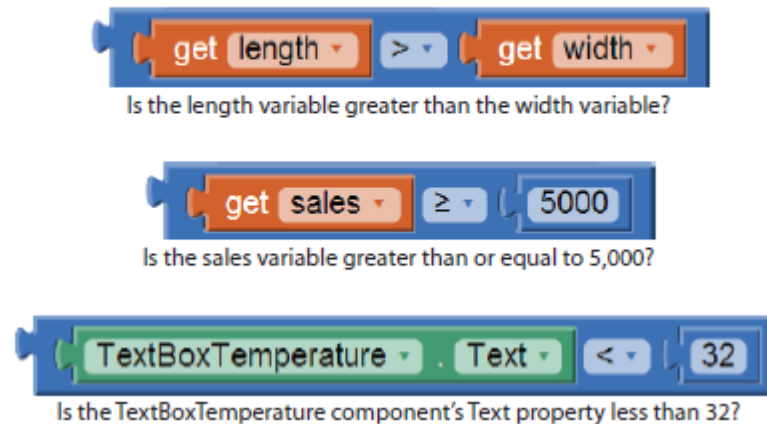
Table 4-1 The Relational Operator Blocks (Source: Pearson Education, Inc.)

Operator Block	Description
	This is the <i>greater than operator</i> . It returns true if the operand on the left is greater than the operand on the right. Otherwise, it returns false.
	This is the <i>greater than or equal to operator</i> . It returns true if the operand on the left is greater than or equal to the operand on the right. Otherwise, it returns false.
	This is the <i>less than operator</i> . It returns true if the operand on the left is less than the operand on the right. Otherwise, it returns false.
	This is the <i>less than or equal to operator</i> . It returns true if the operand on the left is less than or equal to the operand on the right. Otherwise, it returns false.
	This is the <i>equal to operator</i> . It returns true if the operand on the left is equal to the operand on the right. Otherwise, it returns false.
	This is the <i>not equal to operator</i> . It returns true if the operand on the left is not equal to the operand on the right. Otherwise, it returns false.

Relational Operators and the if Block

Figure 4-14 shows some examples of relational operators with operands plugged in.

Figure 4-14 Relational Operator Block Examples (Source: MIT App Inventor 2)



Relational Operators and the `if` Block

- The top example determines whether the *length* variable is greater than the *width* variable.
- The middle example determines whether the *sales* variable is greater than or equal to 5,000.
- The bottom example determines whether the *TextBox* temperature component's *Text* property contains a value that is less than 32.0.

Relational Operators and the `if` Block

- Figure 4-15 shows an example of a complete *if then* block.

Figure 4-15 Example `if then` Block (Source: MIT App Inventor 2)

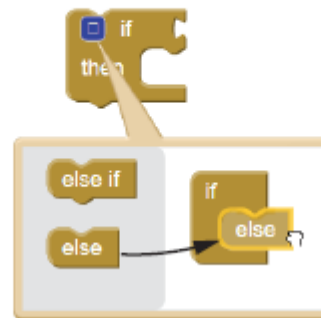


The *if then else* Block

- An *if then else* block will execute one set of blocks if it's Boolean expression is true or another set of blocks if its Boolean expression is false.
- Use the *if then* block's mutator bubble to convert the block into an *if then else* block.

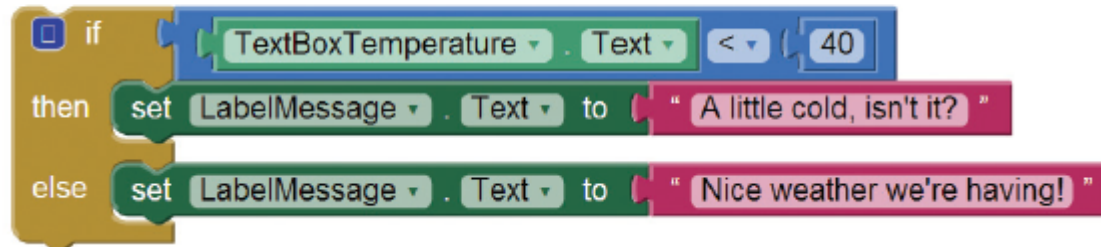
Figure 4-32 Changing the *if then* Block to an *if then else* Block

(Source: MIT App Inventor 2)



The if then else Block

Figure 4-34 Example of the if then else Block (Source: MIT App Inventor 2)



- The Boolean test expression uses the less than (<) operator to determine whether *TextBoxTemperature.Text* is less than 40.0.
- If the Boolean expression is true, then the text *A little cold, isn't it?* is assigned to *LabelMessage.Text*
- If the Boolean expression is false, the text *Nice weather we're having!* is assigned to *LabelMessage.Text*.