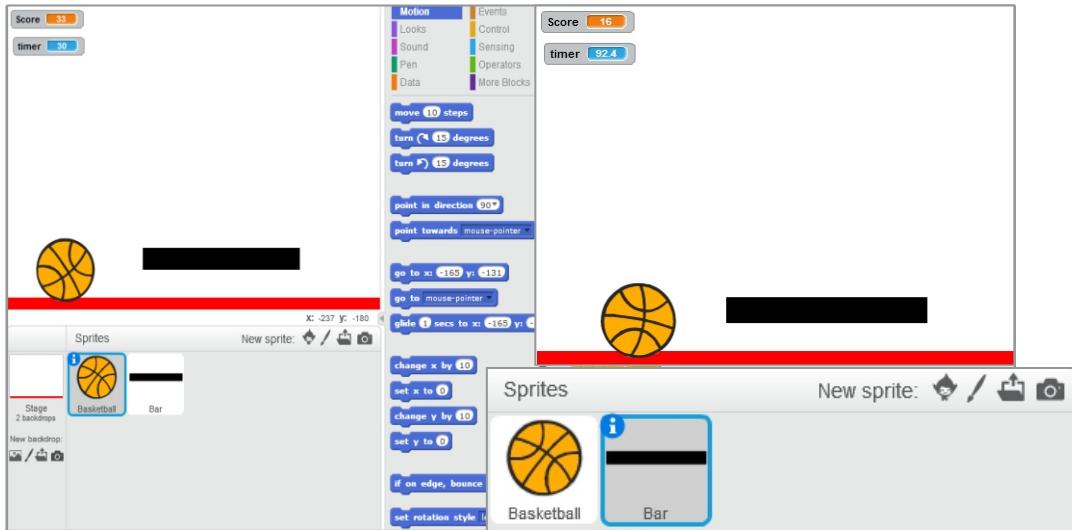


# Chapter 7



## Having Fun with Variables

After completing this chapter you will be able to know how to use the Control, Sensing and Operators Block and make the sprite:

- count from 1 up to 10 using the change block and by incrementing the variable by 1;
- display odd numbers by manipulating the assigned variable;
- creating string variable and display on stage;
- draw an octagon by using a Pen block and a variable to control how many lines and direction does the sprite take;
- recite a multiplication table using variables;
- interact with users through question & answer;
- password and pin code creation and interaction; and
- move and have fun with the Pong Game.

# Variables

To temporarily hold information and pass it to other blocks in your script, you can create variables. Each variable you create in Scratch has a unique name and can hold either a numeric value or a string of characters. A string can be an actual word or any combination of letters, numbers, and symbols, like in a password. For example, you could create a variable called `City` and give it the value `Boston`. Later in your script, you can set `City` to another value, such as `New York`. Variables come in handy whenever you need to use data values that change in a script. For example, suppose that you wanted the cat to count from 1 to 10.

You could create a script with a separate `say [ ] for [ ] secs` block for each of the 10 values (entering 1, 2, 3, and so on by hand), or you could create one variable that will hold the different values, one at a time, and then use a loop and a single `say [ ] for [ ] secs` block with the variable embedded. This is the power of a variable.

## Data Blocks and Creating Variables

To create a new variable in `Scratch`, you need to go to the block palette and select the **Data** blocks category (see Figure 7-1).

Once here, you click the **Make a Variable** button (see Figure 7-2) to open the **New Variable** window (see Figure 7-3).

Here (see Figure 7-3) you enter a name for the variable and select if you want the variable to be available for this sprite only or for all sprites. If you select **For this sprite only**, the variable is considered a `local` variable, meaning that it is only available to the sprite that was selected. If you chose **For all sprites**, the variable is considered a `global` variable. This means that the variable is available to all sprites in the specific project that it was created in. Click the **OK** button when ready. In Figure 7-4, I created a variable named `number`, which we'll use in the examples. A variable, whether local or global, is available per project only. So, you cannot create and use a variable in one project, and then reuse it in another project. You will need to re-create that variable in the second project.

After you create a variable, Scratch adds several other blocks of code to the Data category (see Figure 7-5). These new blocks enable you to work with your variables.

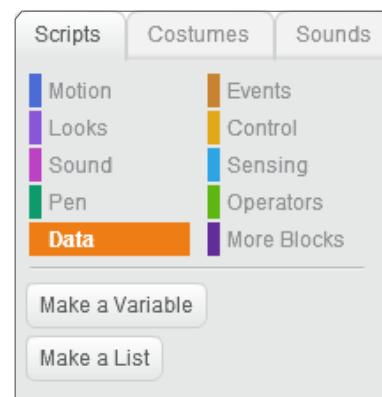


Figure 7-1. Go to the Data category to create a variable



Figure 7-2. Click Make a Variable

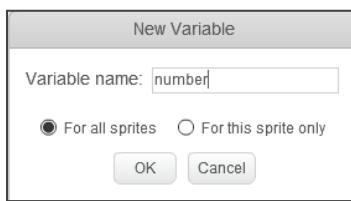


Figure 7-4. Give the variable a name

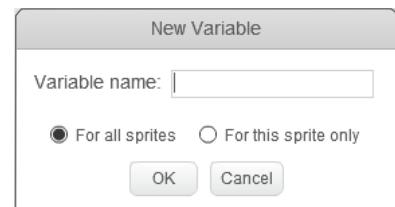


Figure 7-3. The New Variable dialog

The first block you see when you create a variable is a variable reporter block with the same name as your variable, such as `number`. It holds and reports the current value of the variable. To show a reporter window in the stage area that displays the current value of a variable, use the `show variable [number]` block and choose the desired variable's name from the pull-down menu. Likewise, you can hide the reporter window with the `hide variable [number]` block. To assign a specific value to your variable, use the `set [number] to [0]` block, choosing the desired variable name from the pull-down menu and enter a value in the field. Here, the variable called `number` is assigned the value 0. You can also embed a reporter block such as `answer` or an operator block such as `pick random (1 to 10)` in the block to assign the associated value to the variable. The `change [number] by [1]` block changes the value of the variable by the specified value. This block of code accepts numbers only. It will not work with strings. In this activity, the `number` variable is changed by 1, which means that 1 is added to the current value of the variable, and thus the new value for the variable is 2.

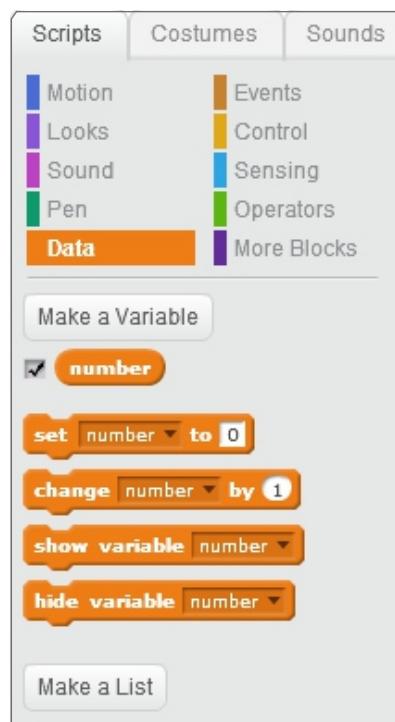


Figure 7-5. Variable blocks

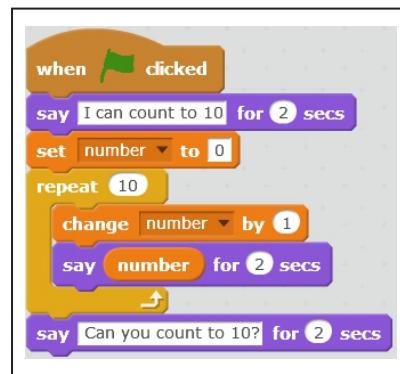
## Activities

Next, you will create some scripts that show how variables are used. By creating and running these scripts, you will get a better understanding of variables. Remember, after learning how they work, you can always change these scripts to your liking or adapt the techniques for other uses.

### Activity 7-1: Count to Ten

This script makes the sprite count from 1 to 10. Before you create Script 7-1, be sure to create a new variable in the Data category of the block palette and name it `number`. The script assigns 0 to `number`, and each time through the repeat loop, it adds 1 to the current value of the variable and displays the current value of `number`.

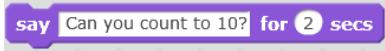
Script 7-1 starts running when the user clicks the green flag. The second block creates a speech bubble for the sprite that displays `I can count to 10` for 2 seconds. The third block assigns the value 0 to the `number` variable. Next is a repeat block that repeats the sequence of actions within it 10 times. The first block in the sequence changes the current value of the `number` variable by 1. The next block creates a speech bubble that displays the current value of the variable for 2 seconds. The last block creates a speech bubble that displays the text `Can you count to 10?` for 2 seconds. **Table 7-1** lists the blocks and describes the actions used in this activity.



Script 7-1. Count to ten



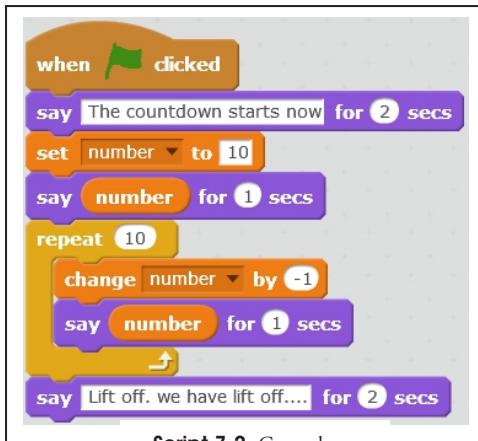
**Table 7-1. Code Blocks in Count to Ten**

Blocks	Actions
	Clicking the green flag activates the script. The green flag is the trigger to start the script running.
	The sprite gets a speech bubble that displays I can count to 10 for 2 seconds.
	Set the current value of the variable called <b>number</b> to 0.
	Repeat the actions represented by the blocks within this block 10 times.
	Add 1 to the current value of the <b>number</b> variable.
	The sprite gets a speech bubble that displays the specified text or value for 2 seconds.
	Hold and report the current value of the <b>number</b> variable.
	The sprite gets a speech bubble that displays Can you count to 10? for 2 seconds.

## Activity 7-2: Countdown

Remember to try to figure out what the script does before you create and run it. By now you should have enough knowledge to be able to read the script, block by block, and know what it does. (This is the same way how programmers read programs line by line, for example, when troubleshooting a program.) As you probably noticed, you need a **number** variable for this script. If necessary, create one now. Let's walk through the script to see if you figured it out.

Script 7-2 starts running when the user clicks the green flag. The next block creates a speech bubble that displays The countdown starts now for 2 seconds. The next block assigns the value 10 to the **number** variable. The fourth block creates a speech bubble that displays the current value of **number** for 1 second. Next, the repeat block repeats the sequence of actions within it 10 times. First, it changes the current value of the **number** variable by -1. Next, it creates a speech bubble that displays the current value of the **number** variable for 1 second. And then the loop repeats. The last block in this script creates a speech bubble that displays Liftoff.... for 2 seconds. **Table 7-2** lists the blocks and describes the actions used in this activity.



Script 7-2. Countdown



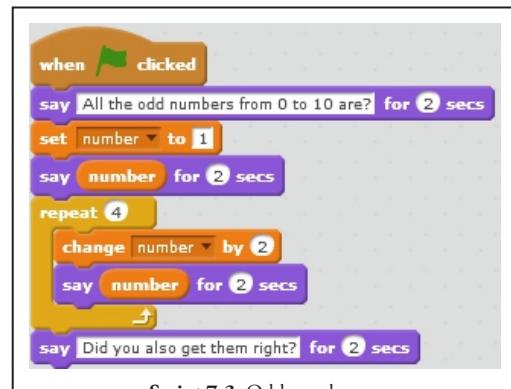
**Table 8-2 Code Blocks in Count Down**

Blocks	Actions
	Clicking the green flag activates the script. The green flag is the trigger to start the script running.
	The sprite gets a speech bubble that displays The countdown starts now for 2 seconds.
	Set the current value of the <b>number</b> variable to 10.
	The sprite gets a speech bubble that displays the specified text or value for 1 second.
	Hold and report the current value of the <b>number</b> variable.
	Repeat the actions represented by the blocks within this block 10 times.
	Subtract 1 from the current value of the <b>number</b> variable.
	The sprite gets a speech bubble that displays the specified text or value for 1 second.
	Hold and report the current value of the <b>number</b> variable.
	The sprite gets a speech bubble that displays Liftoff. We have liftoff.... for 2 seconds.

### Activity 7-3: Odd Numbers

The script assigns the value 1 to the **number** variable and displays this value. Each time through the repeat loop it adds 2 to the current value of the variable and displays the new value. The script displays the values 1, 3, 5, 7, and 9. Again, before you create this script, make sure that the **number** variable is created.

Script 7-3 also starts running when the user clicks the green flag. The next block creates a speech bubble that displays All the odd numbers from 0 to 10 are? for 2 seconds. The next block assigns the value 1 to the **number** variable. The next block creates a speech bubble and displays the current value of the **number** variable for 2 seconds. Next, the  block that repeats the sequence of actions within it 4 times: change the current value of the variable by 2 and create a speech bubble that displays the current value of the **number** variable for 2 seconds. When the loop is finished, the last block in the script creates a speech bubble that displays Did you also get them right? for 2 seconds. **Table 7-3** lists the blocks and describes the actions used in this activity.

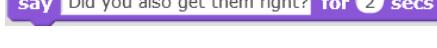


Script 7-3. Odd numbers



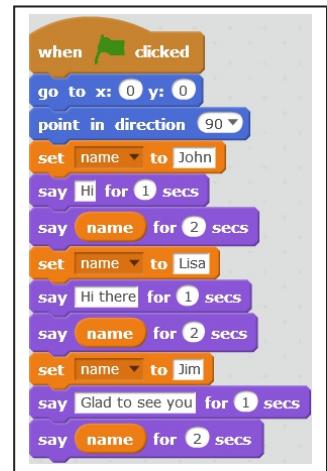
OUTPUT

**Table 7-3. Code Blocks in Odd Number**

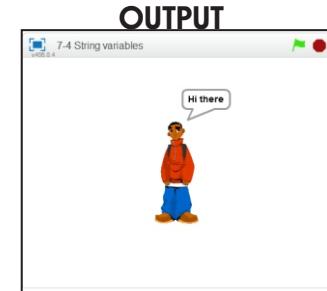
Blocks	Actions
	Clicking the green flag activates the script. The green flag is the trigger to start the script running.
	The sprite gets a speech bubble that displays All the odd numbers from 0 to 10 are? for 2 seconds.
	Set the current value of the number variable to 1.
	The sprite gets a speech bubble that displays the specified text or value for 2 seconds.
	Hold and report the current value of the number variable.
	Repeat the actions represented by the blocks within this block four times.
	Add 2 to the current value of the number variable.
	The sprite gets a speech bubble that displays the specified text or value for 2 seconds.
	Hold and report the current value of the number variable.
	The sprite gets a speech bubble that displays Did you also get them right? for 2 seconds.

## Activity 7-4: String Variable

This simple activity demonstrates that variables may contain a string of characters, such as a name. Also, notice how the same variable changes to hold different values as this script progresses. Before you create this script, you need to create the variable called **name**. Script 7-4 starts running when the user clicks the green flag. The next pair of blocks moves the sprite to the center of the stage (0, 0) and make it face to the right. The **set name to John** block assigns the value John to the **name** variable. The next block creates a speech bubble that displays Hi for 1 second. The next block creates a speech bubble that displays the current value of **name**, which is John, for 2 seconds. The next block assigns the value Lisa to the **name** variable. The next block creates a speech bubble that displays the text Hi there for 1 second. The next block creates a speech bubble that displays the current value of the **name** variable for 2 seconds; this time the value is Lisa. The **set name to Jim** block then assigns the value Jim to the **name** variable. The next block creates a speech bubble that displays Glad to see you for 1 second. The final block creates a speech bubble that displays Jim, the current value of the **name** variable, for 2 seconds. **Table 7-4** lists the blocks and describes the actions used in this activity.



Script 7-4. String variables



**Table 7-4. Code Blocks in String Variable**

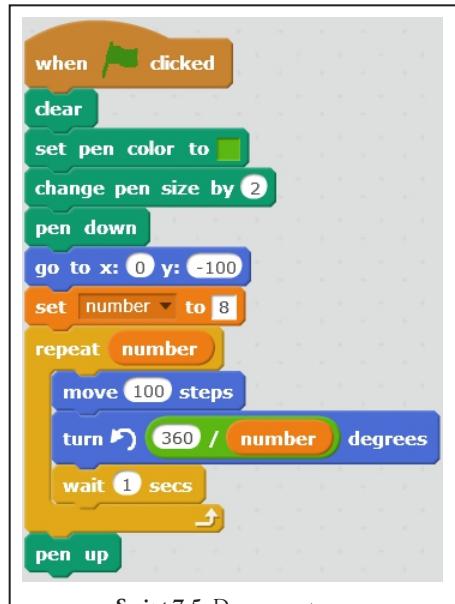
Blocks	Actions
	Clicking the green flag activates the script. The green flag is the trigger to start the script running.
	Move the sprite to the position where X = 0 and Y = 0, which is the center of the stage.
	Make the sprite face to the right.
	Set the current value of the variable called <b>name</b> to John .
	The sprite gets a speech bubble that displays Hi for 1 second.
	The sprite gets a speech bubble that displays the specified text or value for 2 seconds.
	Hold and report the current value of the <b>name</b> variable.
	Set the current value of the <b>name</b> variable to Lisa .
	Hold and report the current value of the <b>name</b> variable.
	The sprite gets a speech bubble that displays the specified text or value for 2 seconds.
	Hold and report the current value of the <b>name</b> variable.
	Set the current value of the <b>name</b> variable to Jim .
	The sprite gets a speech bubble that displays Glad to see you for 1 second.
	The sprite gets a speech bubble that displays the specified text or value for 2 seconds.
	Hold and report the current value of the <b>name</b> variable.

## Activity 7-5: Draw an Octagon

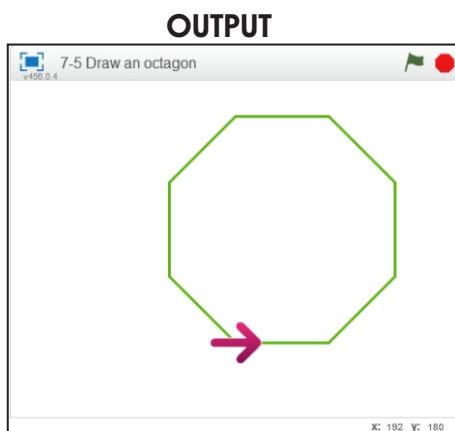
Remember activity 4-6 that drew an octagon in Chapter 4? This activity does the same thing using a **number** variable to determine the number of sides that the pen should draw and the angles between them. Because Script 8-5 sets the variable to 8, the script draws an octagon, but you could easily draw a square, a pentagon, or a circle with the same script. Just set the variable to 4, 5, or 36 instead. Before you give it a try, make sure that you have the **number** variable created. Script 7-5 starts when the user clicks the green flag. The second block clears the stage of any marks made by the pen or stamp. The third block sets the pen color, the fourth increase changes the pen size by 2, and the fifth puts the pen on the stage, ready to draw. The next block moves the sprite to the position (0, -100). Because the goal this time is to draw an octagon, which has 8 sides, the next block assigns the value 8 to the **number** variable. The **number** variable block is embedded in the **repeat** block, meaning the number of times the block repeats the sequenced of actions within it is equal to the value of **number**. In this case that's 8. The first block in the sequence moves the sprite 100 pixels in the direction that it's facing. The next block rotates the sprite a certain amount of degrees counterclockwise. Remember, the angles inside all polygons of four or more sides add up to 360 degrees, so an angle where two sides meet equals 360 divided by the number of sides. To calculate this information and pass it to the **turn  $\square$  degrees** block, you need to type **w360** in the left field of the **360 /  $\square$**  operator block, embed **number** in the right field, and then embed this combination in the **turn  $\square$  degrees** block. The next block pauses the script for 1 second, and then the action loops back to draw the next side. When the final pass through the loop completes, the script executes the last block, which lifts the pen off the stage so that if the sprite moves, the pen will not draw. Run the script to draw the octagon, and then try setting the **number** variable to 4 or 5 to draw a square or pentagon. For a real challenge, think about how you might adapt the script to automatically draw several shapes with different numbers of sides or draw a shape with the number of sides specified by user input. **Table 7-5** lists the blocks and describes the actions used in this activity.

**Table 7-5. Code Blocks in Draw an Octagon**

Blocks	Actions
	Clicking the green flag activates the script. The green flag is the trigger to start the script running.
	Remove all marks previously made by the pen or stamp.
	Set the pen color to lime color.
	Change the pen size to 2.
	The pen is on the stage and is ready to draw.



Script 7-5. Draw an octagon



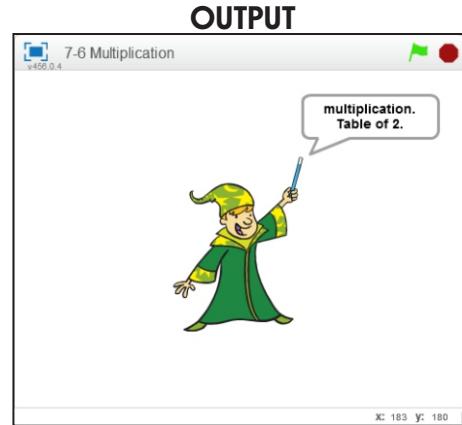
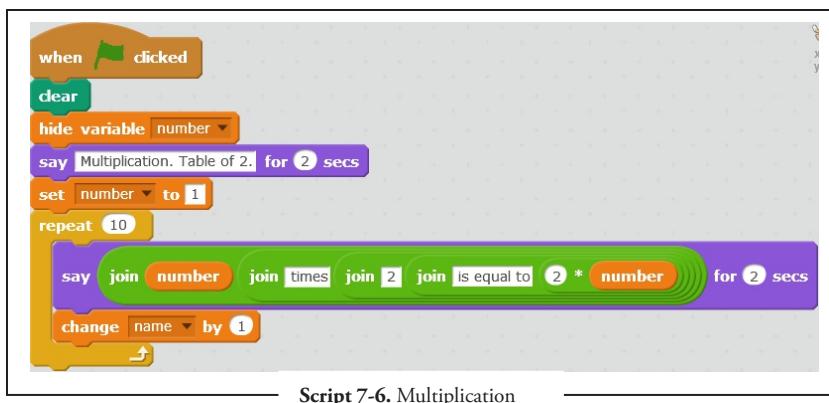
OUTPUT



- Move the sprite to coordinates (X = 0, Y = -100).
- Set the current value of the **number** variable to 8.
- Repeat the actions represented by the blocks within this block a certain number of times.
- Hold and report the current value of the **number** variable.
- Move the sprite 100 pixels in the direction that it's facing.
- Turn the sprite counterclockwise a certain number of degrees.
- Divide one value by another (360 / right field value) and report the result.
- Hold and report the current value of the **number** variable.
- The script waits 1 second. No actions are performed for 1 second.
- The pen is off the stage and cannot draw.
- 

## Activity 7-6: Multiplication

One variable and four **join [ ]** blocks make reciting a multiplication table easy. Before you create Script 7-6, however, make sure that the **number** variable is created.

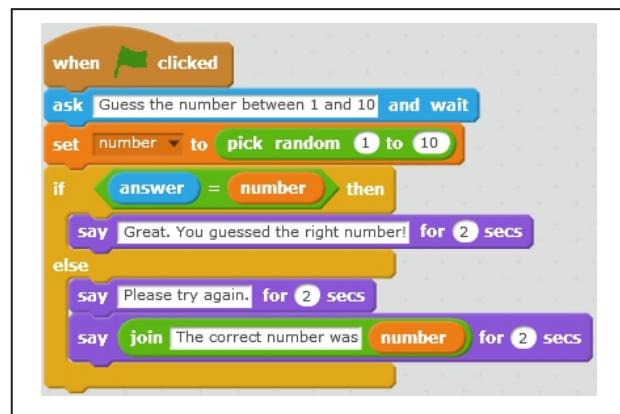


This script starts running when the user clicks the green flag. The second block clears the stage of any marks made by the pen or stamp. The next block hides the variable reporter window so it will not show up in the stage area. The next block creates a speech bubble that displays **Multiplication. Table of 2.** for 2 seconds. The next block assigns the value 1 to the **number** variable. Next there is a repeat block that repeats the sequence of actions within it 10 times. The first block in the sequence creates a speech bubble that for 2 seconds displays the current value of the variable, the text **times 2 is equal to**, and the value of 2 times the current value of the variable. To create this block, start by embedding **[number]** into **[2 \*]**, and then embed the combination into **[join [ ]]** block with the text **[is equal to]**. Keep embedding the **[join [ ]]** blocks and adding text until you have the block shown in Script 7-6.

Blocks	Actions
	Clicking the green flag activates the script. The green flag is the trigger to start the script running.
	Remove all marks previously made by the pen or stamp.
	Hide the variable's ( <b>number</b> ) monitor on the stage.
	The sprite gets a speech bubble that displays Multiplication. Table of 2. for 2 seconds.
	Set the current value of the <b>number</b> variable to 1.
	Repeat the actions represented by the blocks within this block 10 times.
	The sprite gets a speech bubble that displays the specified text or value for 2 seconds.
	Join the two values that have been specified in the block and report the result.
	Hold and report the current value of the <b>number</b> variable.
	Join the two values that have been specified in the block and report the result.
	Join the two values that have been specified in the block and report the result.
	Join the two values that have been specified in the block and report the result.
	Multiply two values and report the result.
	Hold and report the current value of the <b>number</b> variable.
	Add 1 to the current value of the <b>number</b> variable.

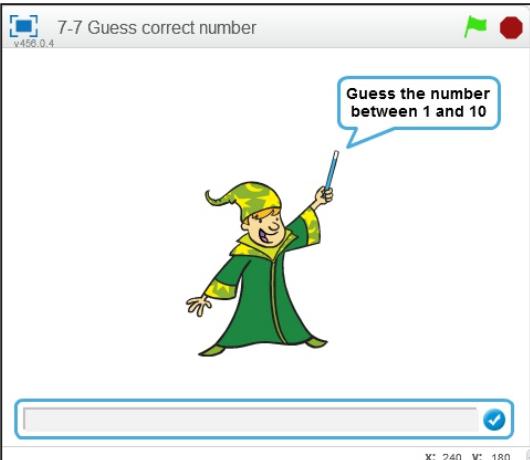
## Activity 7-7: Guess the Correct Number

The script in this activity creates a guessing game. The user needs to guess a number between 1 and 10 that is randomly chosen by the script. Depending on the user input **answer** different speech bubble displays. Before you build this script, make sure that a **number** variable is created. Script 7-7 starts running when the user clicks the green flag. The next block creates a speech bubble and displays the text Guess the number between 1 and 10, opens a user input field, and waits for the user's input. The next block assigns a random number between 1 and 10 to the **number** variable. You create this block by embedding **pick random 1 to 10** in the right field of **set number to 0**. Next, there is an If/ Then/Else conditional statement block that evaluates whether



Script 7-7. Guess correct number

## OUTPUT



the user input in **answer** is equal to the current value in **number**. If the condition is true, the block in the Then section creates a speech bubble that displays Great. You guessed the right number! for 2 seconds. If the condition is false, the two blocks in the Else section are executed. The first one creates a speech bubble that displays Please try again. for 2 seconds. The second block displays The correct number was and the current value of the variable for 2 seconds. **Table 7-7** lists the blocks and describes the actions used in this activity.

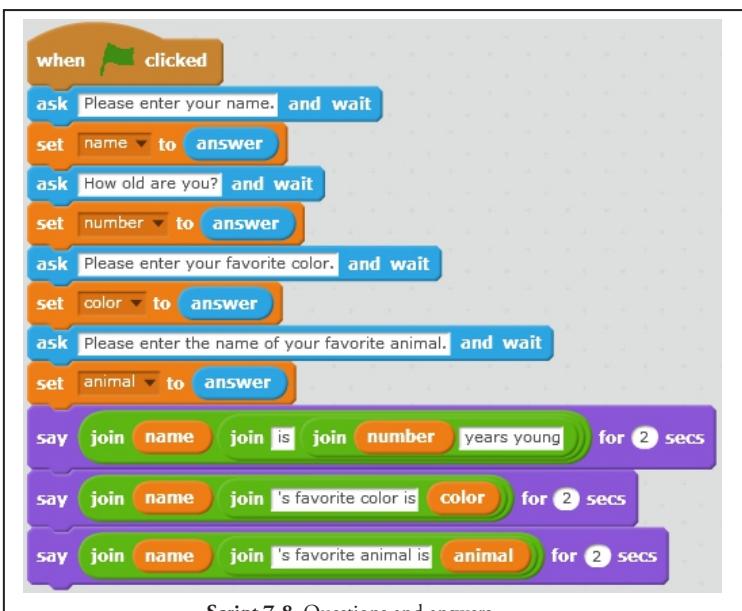
**Table 7-7. Code Blocks in Guess the Correct Number**

Blocks	Actions
	Clicking the green flag activates the script. The green flag is the trigger to start the script running.
	The sprite gets a speech bubble that displays Guess the number between 1 and 10 , opens user input field, and waits for user input.
	Set the current value of the number variable to the value that has been specified in the right field of the block.
	Pick a random value between the two specified numbers and report the result.
	Check if the condition is true. If the condition is true, execute the actions within it, before the word else . If the condition is false, execute the actions after the word else within the block.
	If the value on the left side is equal to the value on the right side, then this condition is true; otherwise, the condition is false.
	Hold and report the current user input value.
	Hold and report the current value of the <b>number</b> variable.
	The sprite gets a speech bubble that displays Great. You guessed the right number! for 2 seconds.
	The sprite gets a speech bubble that displays Please try again. for 2 seconds.
	The sprite gets a speech bubble that displays the specified text or value for 2 seconds.
	Join the two values that have been specified in the block and report the result.
	Hold and report the current value of the <b>number</b> variable.

## Activity 7-8: Questions and Answers

This script asks the user four questions, assigns the answers to four separate variables, and then combines the variable of the values with text in **say [ ] for 2 secs** blocks to respond to the user. With this technique you can embed user input into dialog for your sprite. In addition to the **number** and **name** variables that you used in previous examples, create variables named **color** and **animal**.

Script 7-8 starts running when the user clicks the green flag. The next block creates a speech bubble that displays the text Please enter your name, opens a user input field, and waits for the user's input. The next block assigns the user input as a value to the **name** variable. The next block creates a speech bubble that displays How old are you? , opens a user input field, and waits for the user's answer. The next block assigns the user input as a value to the **number** variable. The next block creates a speech bubble that displays Please enter your favorite color , opens a user input field, and waits for the user's input. The next block assigns the user input as a value to the variable called **color**. The next block creates a speech bubble that displays Please enter the name of your favorite animal , opens a user input field, and waits for the user's answer. The next block assigns the user input as a value to the variable called **animal**. The next three instructions in the script each use **join [ ]** blocks to combine variable values and text into **say [ ] for 2 secs** blocks to create 2-second speech bubbles. The first displays the value for the **name** variable, the text is , the value of the **number** variable, and the text years young. The second displays the value for the **name** variable together with the text's favorite color is and the value of the **color** variable. The third block's speech bubble displays the value for the **name** variable, the text's favorite animal is , and the value of the **animal** variable. **Table 7-8** lists the blocks and describes the actions used in this activity.



Script 7-8. Questions and answers



Table 7-8. Code Blocks in Questions and Answers

Blocks	Actions
	Clicking the green flag activates the script. The green flag is the trigger to start the script running.
	The sprite gets a speech bubble that displays Please enter your name. , opens user input field, and waits for user input.
	Set the current value of the <b>name</b> variable to the value that has been specified in the right field of the block.

**answer**

**ask** How old are you? **and wait**

**set** number **to**

**answer**

**ask** Please enter your favorite color. **and wait**

**set** color **to**

**answer**

**ask** Please enter the name of your favorite animal. **and wait**

**set** animal **to**

**answer**

**say**  **for**  **secs**

**join**

**name**

**join** is

**join**  years young

**number**

**say**  **for**  **secs**

**join**

**name**

**join** 's favorite color is

**color**

**say**  **for**  **secs**

**join**

**name**

**join** 's favorite animal is

**animal**

Hold and report the current user input value.

The sprite gets a speech bubble that displays How old are you? , opens user input field, and waits for user input.

Set the current value of the number variable to the value that has been specified in the right field of the block.

Hold and report the current user input value.

The sprite gets a speech bubble that displays Please enter your favorite color. , opens user input field, and waits for user input.

Set the current value of the variable called color to the value that has been specified in the right field of the block.

Hold and report the current user input value.

The sprite gets a speech bubble that displays Please enter the name of your favorite animal. , opens user input field, and waits for user input.

Set the current value of the variable called animal to the value that has been specified in the right field of the block.

Hold and report the current user input value.

The sprite gets a speech bubble that displays the specified text or value for 2 seconds.

Join the two values that have been specified in the block and report the result.

Hold and report the current value of the name variable.

Join the two values that have been specified in the block and report the result.

Join the two values that have been specified in the block and report the result.

Hold and report the current value of the number variable.

The sprite gets a speech bubble that displays the specified text or value for 2 seconds.

Join the two values that have been specified in the block and report the result.

Hold and report the current value of the name variable.

Join the two values that have been specified in the block and report the result.

Hold and report the current value of the color variable.

The sprite gets a speech bubble that displays the specified text or value for 2 seconds.

Join the two values that have been specified in the block and report the result.

Hold and report the current value of the name variable.

Join the two values that have been specified in the block and report the result.

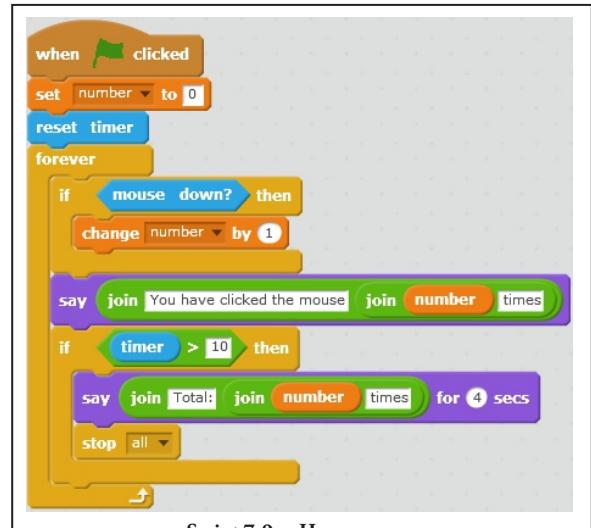
Hold and report the current value of the animal variable.

## Activity 7-9: How Many Mouse Clicks?

This script keeps track of the number of times that you click the mouse button in 10 seconds. After 10 seconds, the script displays the total number of times that you clicked the mouse button. For Script 7-9, you need the **number** variable. The script starts running when the user clicks the green flag. The next block assigns the value 0 to the **number** variable.

The next block resets the timer to 0.0. The next block loops the sequence of actions within it forever, or until the script is manually stopped. The first block within the forever block is an If/Then conditional statement block that checks whether the primary mouse button is clicked. If it is, then the condition is true and the **change number by 1** block adds 1 to the current value of the **number** variable. Next, the script goes to the **say** block, which creates a speech bubble that displays You have clicked the mouse , the current value of the **number** variable, and times. The following block is a second If/Then conditional statement.

This one checks whether the timer value is greater than 10. If it is, then the condition is true and the sequence of actions within the repeat block executes: the first block displays Total: , the value of the **number** variable, and the text times for 4 seconds. The second block stops all scripts in this project. If the timer is not greater than 10, the loop repeats again. **Table 7-9** lists the blocks and describes the actions used in this activity.



Script 7-9. How many

### OUTPUT

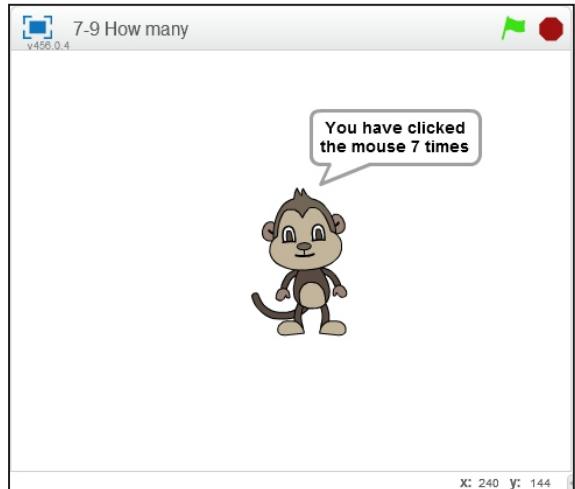


Table 7-9. Code Blocks in How Many Mouse Clicks?

Blocks	Actions
	Clicking the green flag activates the script. The green flag is the trigger to start the script running.
	Set the current value of the <b>number</b> variable to 0.
	Reset the timer value back to 0.0.
	Repeat all the instructions/blocks within this block forever.
	Check if the condition is true. If the condition is true, execute the actions within it. If the condition is false, skip to the next block.
	Check if the primary mouse button is clicked. If the primary mouse button is clicked, then the condition is true. If not, then the condition is false.

**change** **number** **by** **1**

Add 1 to the current value of the **number** variable.

**say**

The sprite gets a speech bubble that displays the specified text or value.

**join** **You have clicked the mouse**

Join the two values that have been specified in the block and report the result.

**join**  **times**

Join the two values that have been specified in the block and report the result.

**number**

Hold and report the current value of the **number** variable.

**if**  **then**

Check if the condition is true. If the condition is true, execute the actions within it. If the condition is false, skip to the next block.

**>** **10**

If the value on the left side is greater than 10, then this condition is true; otherwise, the condition is false.

**timer**

Hold and report the timer value.

**say**  **for** **4** **secs**

The sprite gets a speech bubble that displays the specified text or value for 4 seconds.

**join** **Total:**

Join the two values that have been specified in the block and report the result.

**join**  **times**

Join the two values that have been specified in the block and report the result.

**number**

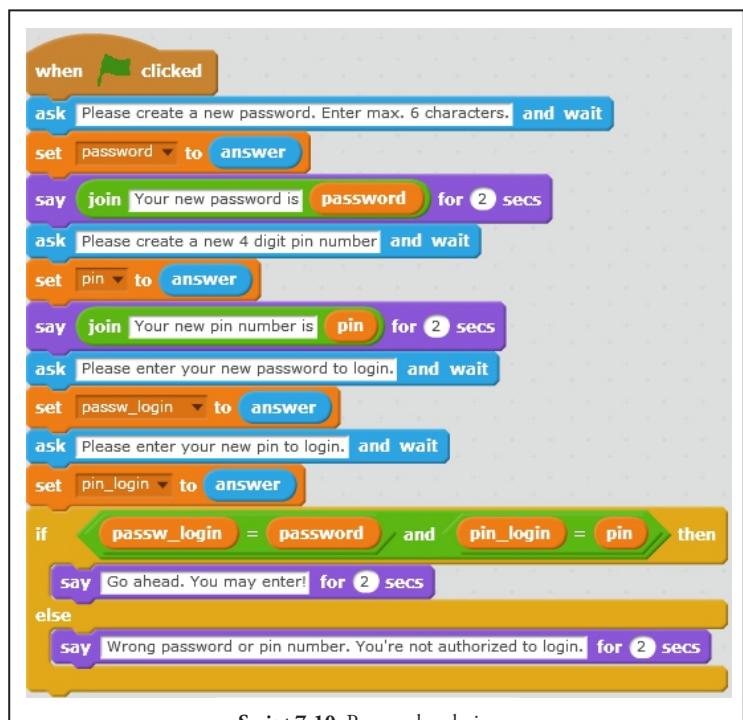
Hold and report the current value of the **number** variable.

**stop** **all**

Stop all scripts in this project.

## Activity 7-10: Password and Pin Code

Ever wonder what goes on behind the scenes when you log into a website? This activity will give you a glimpse! Script 7-10 asks the user to enter a password and pin code, and then assigns the user responses to variables called password and pin. The user is asked to log in with his or her new password and pin, and then the script saves those responses in variables named **pass\_login** and **pin\_login**. Finally, the script compares the values held in the two pairs of variables to ensure that they match. Before you create the script, be sure to create the variables **password**, **pin**, **passw\_login**, and **pin\_login**. The script starts running when the user clicks the green flag. The next block creates a speech bubble that displays Please create a new password. Enter max. 6 characters , opens a user input field, and waits for the user's input. The next block assigns the user input to the **password**

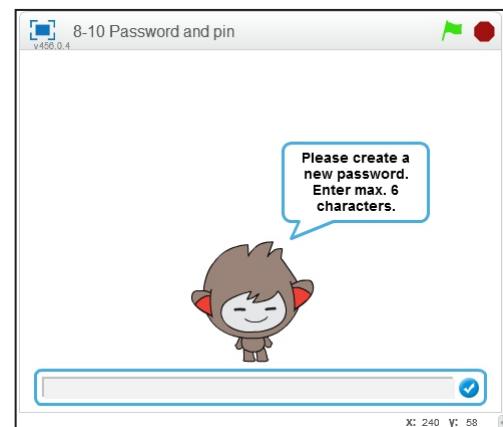


Script 7-10. Password and pin

variable. The next block creates a speech bubble that displays Your new password is and the value of the **password** variable for 2 seconds. The next block creates a speech bubble that displays Please create a new 4 digit pin number , opens a user input field, and waits for the user's input. The next block assigns the user input to the **pin** variable. The next block creates a speech bubble that displays Your new pin number is and the value of the **pin** variable for 2 seconds. The next block creates a speech bubble that displays Please enter your new password to log in, opens a user input field, and waits for the user's input. The next block assigns the user input to the **passw\_login** variable.

The next block creates a speech bubble that displays Please enter your new pin to log in, opens a user input field, and waits for the user's input. The next block assigns the user input to the **pin\_login**. The next block is an If/Then/Else conditional statement that evaluates whether the variable **passw\_login** variable is equal to the **password** variable and that the **pin\_login** variable is equal to the **pin** variable. If both are true, then the Then section is executed: the script creates a speech bubble that displays Go ahead. You may enter! for 2 seconds. If the condition is false, meaning one or both pairs of variables don't match, then the Else section is executed: the script creates a speech bubble that displays Wrong password or pin number. You're not authorized to log in. for 2 seconds. **Table 7-10** lists the blocks and describes the actions used in this activity.

## OUTPUT



**Table 7-10. Code Blocks in Password and Pin Code**

Blocks	Actions
	Clicking the green flag activates the script. The green flag is the trigger to start the script running.
	The sprite gets a speech bubble that displays Please create a new password. Enter max. 6 characters. , opens user input field, and waits for user input.
	Set the current value of the variable called <b>password</b> to the value that has been specified in the right field of the block.
	Hold and report the current user input value.
	The sprite gets a speech bubble that displays the specified text or value for 2 seconds.
	Join the two values that have been specified in the block and report the result.
	Hold and report the current value of the <b>password</b> variable.
	The sprite gets a speech bubble that displays Please create a new 4 digit pin number , opens user input field, and waits for user input.
	Set the current value of the variable called <b>pin</b> to the value that has been specified in the right field of the block.
	Hold and report the current user input value.

**say** [ ] **for** (2) **secs**

The sprite gets a speech bubble that displays the specified text or value for 2 seconds.

**join** [Your new pin number is] [ ]

Join the two values that have been specified in the block and report the result.

**pin**

**ask** [Please enter your new password to login.] **and wait**

Hold and report the current value of the **pin** variable.

The sprite gets a speech bubble that displays Please create a new password to log in., opens user input field, and waits for user input.

**set** [passw\_login] **to** [ ]

Set the current value of the variable called **passw\_login** to the value that has been specified in the right field of the block.

**answer**

**ask** [Please enter your new password to login.] **and wait**

Hold and report the current user input value.

The sprite gets a speech bubble that displays Please create a new pin to log in., opens user input field, and waits for user input.

**set** [pin\_login] **to** [ ]

Set the current value of the variable called **pin\_login** to the value that has been specified in the right field of the block.

**answer**

**if** [ ] **then**  
[ ]  
**else**  
[ ]

Hold and report the current user input value.

Check if the condition is true. If the condition is true, execute the actions within it, before the word **else**. If the condition is false, execute the actions after the word **else** within the block.

**and**

Check both conditions on each side of this block. If both conditions are true, then this condition is true. Any other combination results in this condition being false.

[ ] = [ ]

If the value on the left side is equal to the value on the right side, then this condition is true; otherwise, the condition is false.

**passw\_login**

Hold and report the current value of the **passw\_login** variable.

**password**

Hold and report the current value of the **password** variable.

[ ] = [ ]

If the value on the left side is equal to the value on the right side, then this condition is true; otherwise, the condition is false.

**pin\_login**

Hold and report the current value of the **pin\_login** variable.

**pin**

Hold and report the current value of the **pin** variable.

**say** [Go ahead. You may enter!] **for** (2) **secs**

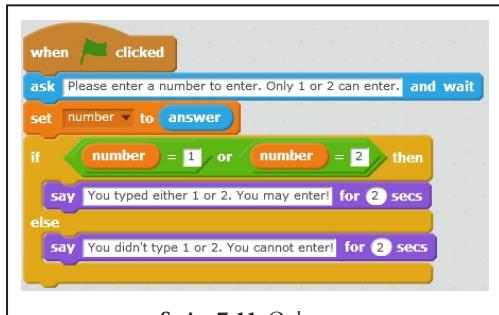
The sprite gets a speech bubble that displays Go ahead. You may enter! for 2 seconds.

**say** [Wrong password or pin number. You're not authorized to login.] **for** (2) **secs**

The sprite gets a speech bubble that displays Wrong password or pin number. You're not authorized to log in. for 2 seconds.

## Activity 7-11: Only One Correct Answer Required

Script 7-11 follows a pattern that should be familiar by now. Can you figure out what this script does before you create and run it? (Before creating Script 7-11, make sure that the **number** variable is created.) The script starts running when the user clicks the green flag. The next block creates a speech bubble that displays Please enter a number to enter. Only 1 or 2 can enter, opens a user input field, and waits for the user's input. The next block assigns the user input to the **number** variable. Next, the If/Then/Else conditional statement evaluates whether the **number** variable is equal to 1 or 2. If the variable holds either value, then result is true and the script executes the Then section. It creates a speech bubble that displays You typed either 1 or 2. You may enter! for 2 seconds. If the condition is false, the Else section is executed, meaning a speech bubble gets created with the following text for 2 seconds: You didn't type 1 or 2. You cannot enter!. Note that for this condition, just one side of the word **or** needs to be true for the condition to be true. If both sides are false, then the condition is false. **Table 7-11** lists the blocks and describes the actions used in this activity.



Script 7-11. Only one

### OUTPUT



Table 7-11. Code Blocks in Only One Correct Answer Required

Blocks	Actions
	Clicking the green flag activates the script. The green flag is the trigger to start the script running.
	The sprite gets a speech bubble that displays Please enter a number to enter. Only 1 or 2 can enter., opens user input field, and waits for user input.
	Set the current value of the <b>number</b> variable to the value that has been specified in the right field of the block.
	Hold and report the current user input value.
	Check if the condition is true. If the condition is true, execute the actions within it, before the word else. If the condition is false, execute the actions after the word else within the block.
	Check both conditions on each side of the block. If both conditions are true, then this condition is true. Any other combination results in this condition being false.
	If the value on the left side is equal to 1, then this condition is true; otherwise, the condition is false.
	Hold and report the current value of the <b>number</b> variable.
	The sprite gets a speech bubble that displays You typed either 1 or 2. You may enter!, for 2 seconds.
	The sprite gets a speech bubble that displays You didn't type 1 or 2. You cannot enter!, for 2 seconds.

## Activity 7-12: Pong Game

In this activity, you'll create a version of the Pong game (see Figure 7-6). The game's object is to hit the ball sprite with the paddle sprite (the black bar) as many times as possible during the 20-second time limit and to prevent the ball from hitting the red line at the bottom of the stage area (see Figure 7-7). The user controls the black bar but can move it from left to right only. The ball can bounce against any of the stage area edges. Each time the ball is hit with the black bar you score 1 point. The game stops after the time limit is reached or if the ball hits the red line at the bottom of the stage area. The goal of the game is to score as many points as possible within the time limit of 20 seconds.

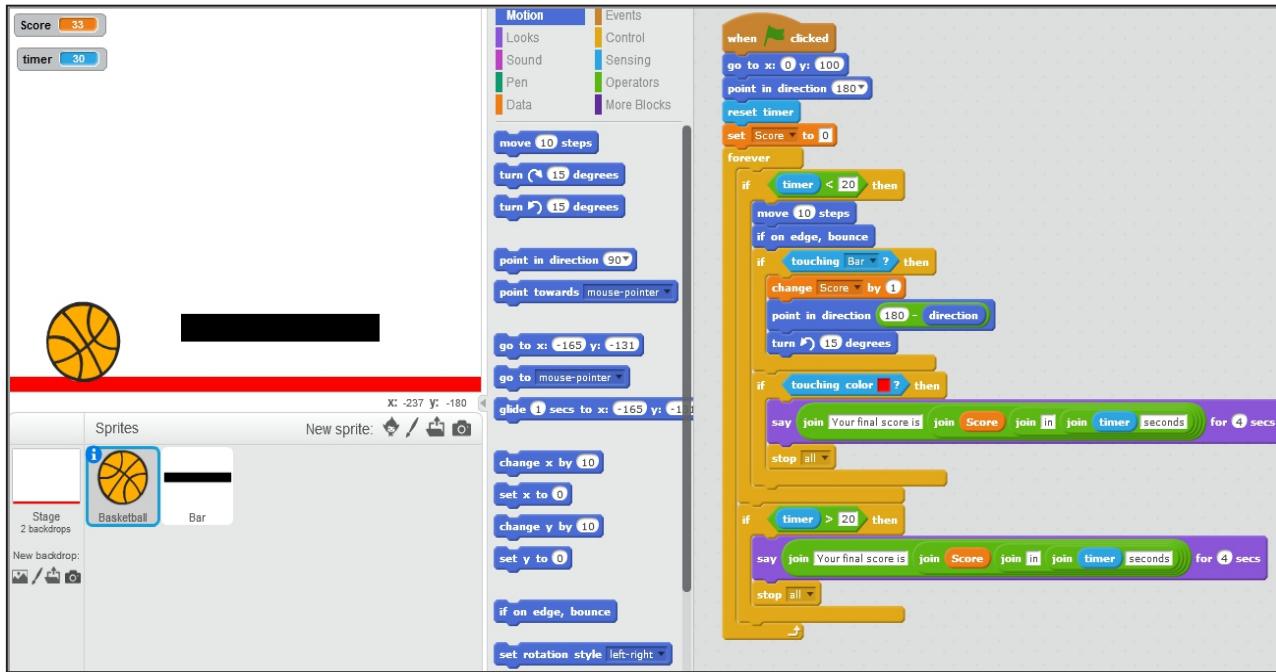


Figure 7-6. Pong game

For this script, you need to select a sprite from the Sprite Library to act as the ball. The sprite called **Basketball** works well (see Figure 7-8). You also need to draw a new sprite to be the paddle that hits the ball. Use Figure 7-9 as an example and call it **Bar**. You also need to create a backdrop with a red rectangle at the bottom of the stage area, as shown in Figure 7-10. (Refer to Chapter 2 if you need a reminder of how to draw a new sprite or backdrop.) Finally, before creating this script, you also need to create a variable called **Score**.

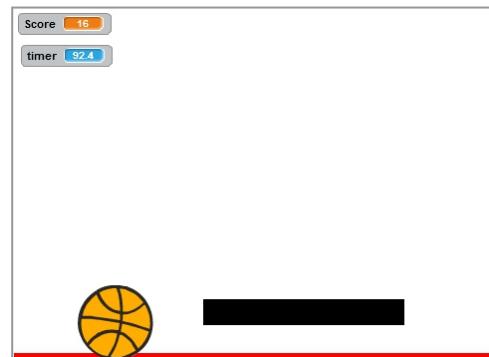


Figure 7-7. Pong stage

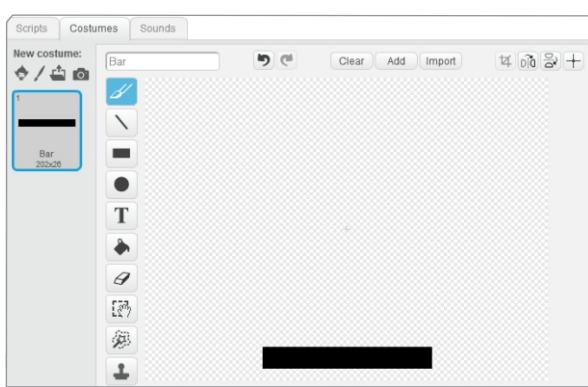


Figure 7-9. Bar



Figure 7-8. Pong sprites

Once all your elements are ready, you can start creating the script. Select the ball in the Sprites pane, because Script 7-11 is assigned to the ball sprite. The script starts running when the user clicks the green flag. The next block moves the sprite to the position (0, 100). The next block makes the sprite face down. The next block resets the timer to 0.0. The next block assigns the value 0 to the Score variable. The rest of the script resides in a forever block loop that repeats the sequence of actions within it forever, or until the script is stopped manually. The first block inside is an **if** block that checks whether the timer value is less than 20. If this condition is true, then it will execute the sequence of actions within it. The first block within it moves the sprite 10 pixels in the direction that it's facing. The next block bounces the sprite when if it hits any stage edge and makes it travel in the opposite direction. Next is another **if** block that checks whether the **Basketball** sprite is touching the sprite called **Bar**. If this condition is true, then the value 1 is added to the current value of the Score variable. The next block makes the **Basketball** sprite face in a direction that is calculated by subtracting the direction that the sprite is facing from 180. The next block rotates the **Basketball** sprite 15 degrees counterclockwise. Next is a third If/Then C block that checks whether the **Basketball** sprite is touching the color red in the stage area. The only red object in the stage area is the red rectangle at the bottom of the backdrop. If this condition is true, then a speech bubble gets created and displayed for 4 seconds. The text Your final score is , the value of the Score variable, the text in , the value of the timer, and the text seconds are displayed in the speech bubble. The last block in the third **if** stops all the scripts in this project. The fourth If/Then block checks whether the timer value is greater than 20. If this condition is true then a speech bubble gets created and displayed for 4 seconds. The text Your final score is , the value of the Score variable, the text in , the value of the timer, and the text seconds are displayed. The last block stops all the scripts in this project. **Table 7-12** lists the blocks and describes the actions used in this activity.

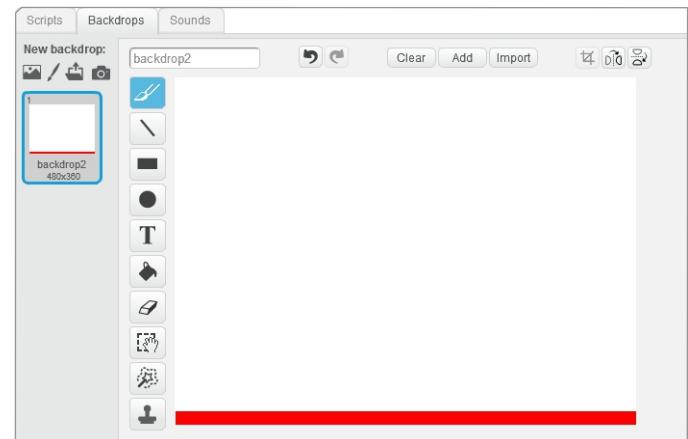
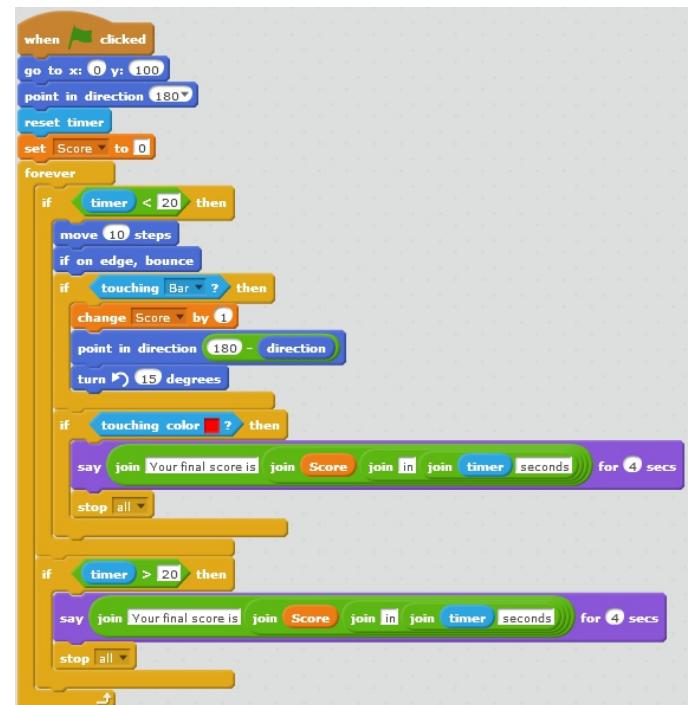
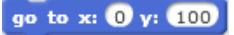
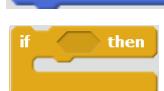
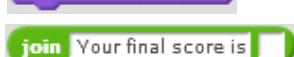


Figure 7-10. Red rectangle



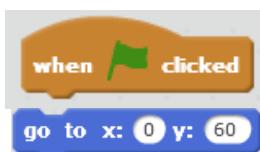
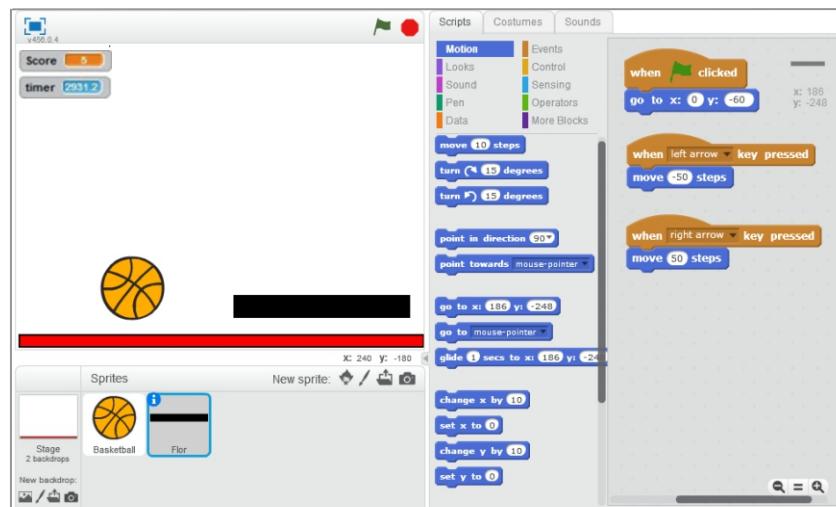
Script 7-12 Pong game ball script

**Table 7-12. Code Blocks in Pong Game Ball**

Blocks	Actions
	Clicking the green flag activates the script. The green flag is the trigger to start the script running.
	Move the sprite to coordinates (X = 0, Y = 100).
	Make the sprite face down.
	Reset the timer value back to 0.0.
	Set the current value of the variable called <b>Score</b> to 0.
	Repeat all the instructions/blocks within this block forever.
	Check if the condition is true. If the condition is true, execute the actions within it. If the condition is false, skip to the next block.
	If the value on the left side is less than 20, then this condition is true; otherwise, the condition is false.
	Hold and report the timer value.
	Move the sprite 10 pixels in the direction that it's facing.
	If the sprite reaches the edge of the stage, bounce in the opposite direction.
	Check if the condition is true. If the condition is true, execute the actions within it. If the condition is false, skip to the next block.
	Checks if the sprite is touching the sprite called <b>Bar</b> . If it is, then this condition is set to true.
	Add 1 to the current value of the <b>Score</b> variable.
	Make the sprite face in the specified direction.
	Subtract one value from another ( $180 - 10$ ) and report the result.
	Hold and report the direction value.
	Turn the sprite counterclockwise 15 degrees.
	Check if the condition is true. If the condition is true, execute the actions within it. If the condition is false, skip to the next block.
	Checks if the sprite is touching the color red. If it is, then this condition is set to true.
	The sprite gets a speech bubble that displays the specified text for 4 seconds.
	Join the two values that have been specified in the block and report the result.
	Join the two values that have been specified in the block and report the result.
	Hold and report the current value of the <b>Score</b> variable.

	Join the two values that have been specified in the block and report the result.
	Join the two values that have been specified in the block and report the result.
	Hold and report the timer value.
	Stop all scripts in this project.
	Check if the condition is true. If the condition is true, execute the actions within it. If the condition is false, skip to the next block.
	If the value on the left side is greater than 20, then this condition is true; otherwise, the condition is false.
	Hold and report the timer value.
	The sprite gets a speech bubble that displays the specified text for 4 seconds.
	Join the two values that have been specified in the block and report the result.
	Join the two values that have been specified in the block and report the result.
	Hold and report the current value of the Score variable.
	Join the two values that have been specified in the block and report the result.
	Join the two values that have been specified in the block and report the result.
	Hold and report the timer value.
	Stop all scripts in this project.

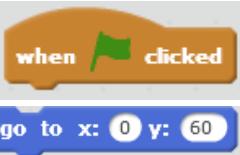
With the ball under control, you're ready to create the three scripts needed to make the **Bar** sprite move (see Figure 7-11). Select the **Bar** sprite to make it the current sprite, and then drag and snap together the following scripts. Script 7-13 moves the bar to a starting position at the beginning of the game. The script starts running when the green flag is clicked. The next block of code moves the **Bar** sprite to the position with coordinates (0, -60).



Script 7-13. Bar starting position

Script 7-11. Bar Control

**Table 7-13. Code Blocks in Bar Starting Position**

Blocks	Actions
	Clicking the green flag activates the script. The green flag is the trigger to start the script running.
	Move the sprite to coordinates (X = 0, Y = -60).

Script 7-14 moves the bar to the left whenever the left arrow key is pressed. Create Script 7-14 in the same script area as the Script 7-13 but beneath the previous script. This script starts when the left arrow key is pressed. The next block of code moves the **Bar** sprite 50 pixels to the left. **Table 7-14** lists the blocks and describes the actions used in this activity.



Script 7-14. Move bar left

**Table 7-14. Code Blocks in Move the Bar Sprite Left**

Blocks	Actions
	Pressing the left arrow key activates the script. The left arrow key is the trigger to start the script running.
	Move the sprite 50 pixels in the opposite direction that it's facing.

Script 7-15 moves the bar to the right whenever the right arrow key is pressed. Once again, create this script in the same script area but beneath the previous two. Script 8-15 starts when the right arrow key is pressed. The next block of code moves the **Bar** sprite 50 pixels to the right. **Table 7-15** lists the blocks and describes the actions used in this activity.



Script 7-15. Move bar to the right

**Table 7-15. Code Blocks in Move the Bar Sprite to the Right**

Blocks	Actions
	Pressing the right arrow key activates the script. The right arrow key is the trigger to start the script running.
	Move the sprite 50 pixels in the direction that it's facing.

Don't be concerned by the multiple scripts in this project. The whole project or game starts running when the green flag is clicked. That is the trigger to start the game. You control the bar with the left and right arrow keys.

## Summary

Variables are a very important topic when it comes to computer programming. All computer programming languages use variables, including Scratch. Variables make your programs and Scratch projects more flexible and dynamic. The key to remember is that a variable holds a value, but this value can change while the program runs. In the next chapter, you'll investigate another important topic called a list. A list is similar to a variable. The difference is that a list holds multiple values at one time and a variable holds only one value at a time. Remember to do the exercises at the end of each chapter. This way you'll put what you learned into practice, and as you probably know, "practice makes perfect."

# Snap Script

a Short hands-on activity

1. Create a script that asks the user to enter a magic word. If the user enters the correct magic word, the script displays a speech bubble with the text You may enter . If the user enters the wrong word, the script displays Please try again . You need to create a variable for this script.

## Output:



2. Create a script that makes the sprite count from 1 to 3. At the end of the script, create a speech bubble with the text Ready, steady, go! .

## Output:

