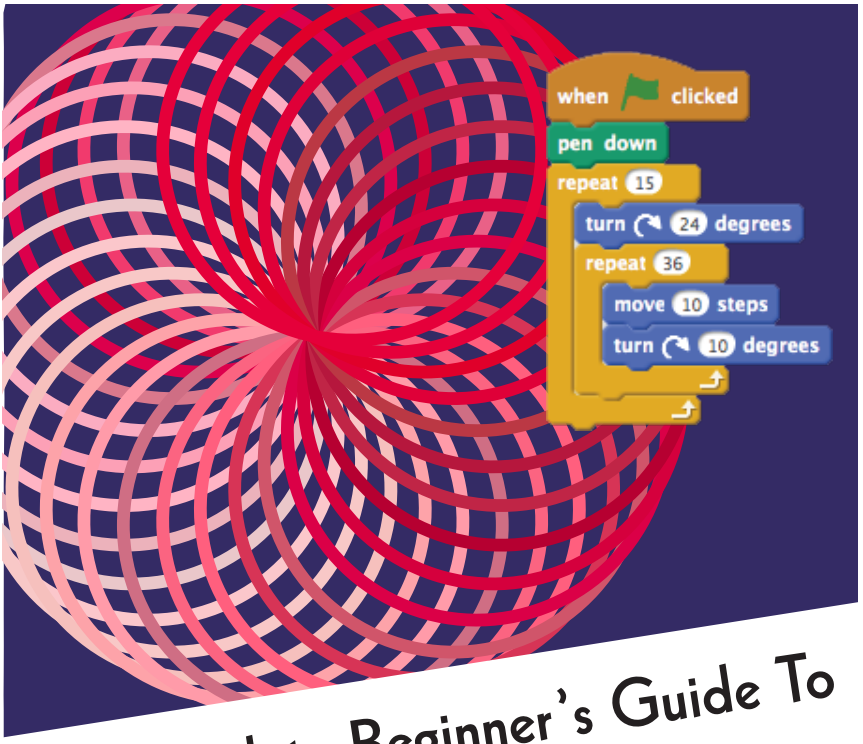


The Absolute Beginner's Guide To CODING

Using Scratch

Table Of Contents

Why Learn To Code?

01

All About Scratch

Have you ever wondered how websites are built, how video games are created, or how apps for smartphones are designed?

They were all made with a programming language, which is a special language that allows people to communicate with computers.

What is Scratch?

One of the many programming languages available is Scratch. Scratch is a visual programming language. People like to program with Scratch because it is easy to get started and quickly make the computer do complex things.

When you write a program in Scratch, you are making a list of instructions for the computer to follow. A computer's job is to follow instructions. In fact, a computer follows millions of instructions every second.

Your job is to learn how to communicate with the computer so it will follow your instructions. Don't worry, learning how to code is not hard at all, and you will soon be well on your way to creating your own computer programs.

Finding Scratch

Scratch is a free programming language that runs over the internet. You can program in scratch from your home computer by visiting their website,

https://scratch.mit.edu

Scratch requires Adobe Flash to run on Macs, Windows, and Linux. You can get Flash at,

https://get.adobe.com/flashplayer/

If Scratch does not work on your computer, you can use a very similar language called Snap!, which also runs over the internet, but does not require Flash.

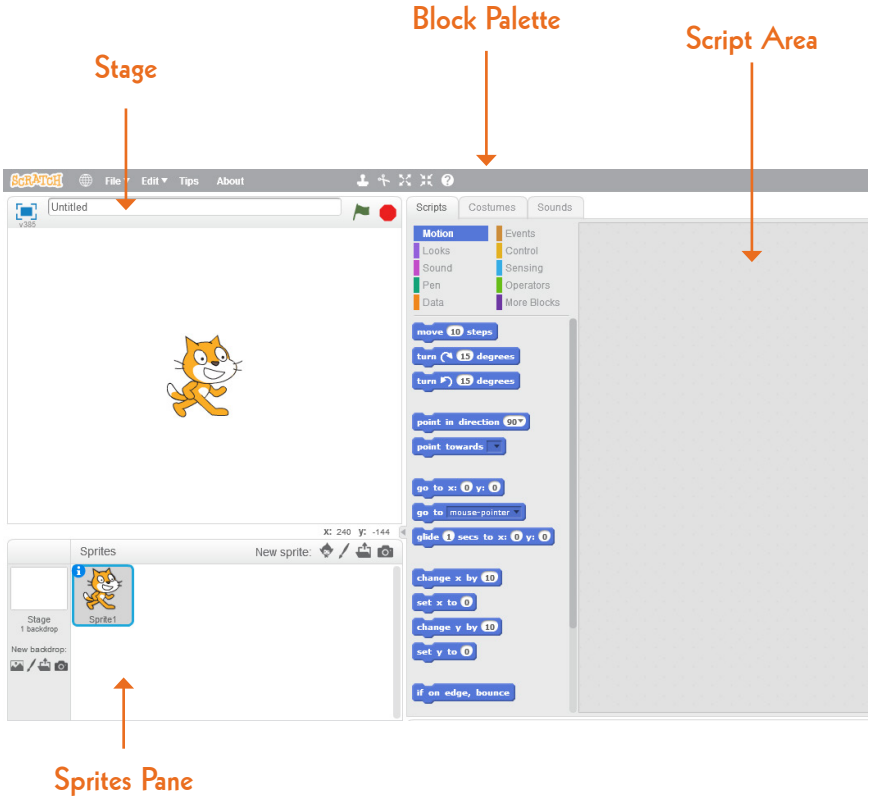
http://snap.berkeley.edu

Snap! is almost identical to Scratch and most of the programs you write in this book will work in either language.

Create Your First Program

02

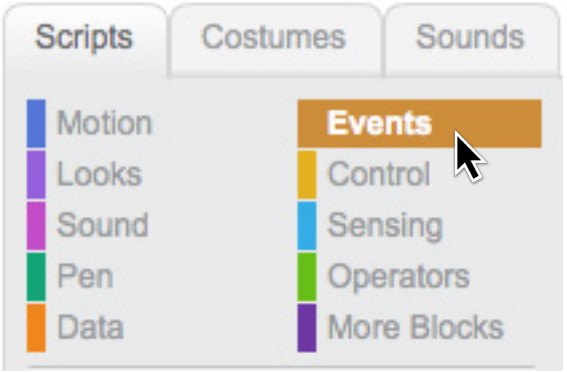
Go to the Scratch website, click the ***Create*** menu item at the top of the web page. When you first open Scratch you will see something similar to the sketch below.



One of the first thing you will notice is that there is a picture of a cat in the middle of the stage area. The cat is called a sprite. A sprite is an object that performs actions in the program. You can change your sprite's image by clicking on the ***Constumes Tab*** in the ***Block Palette***.

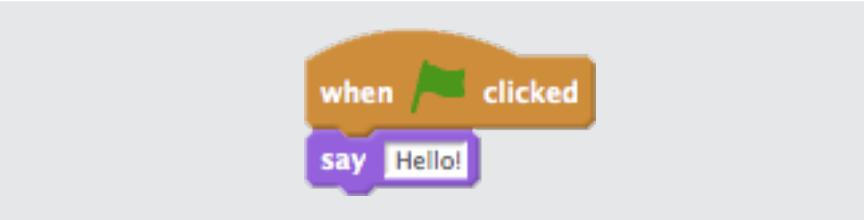
Making a program in Scratch is a snap because everything we want to tell the computer to do can be made by fitting together the command blocks. For our first program we will make the cat say "Hello World."

Step 1) The first thing we need is a way to start the program running. Click on ***Events*** in the block palette.

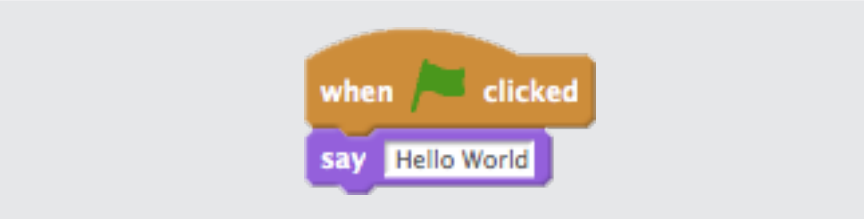


This will reveal a the event command blocks. We would like the program to start running whenever the green flag above the stage area is clicked. Drag the ***when flag clicked*** block into the script area.

Step 2) Next, click on ***Looks*** in the block palette. Drag the ***say*** block into the script area, connecting it to the ***when flag clicked*** block.

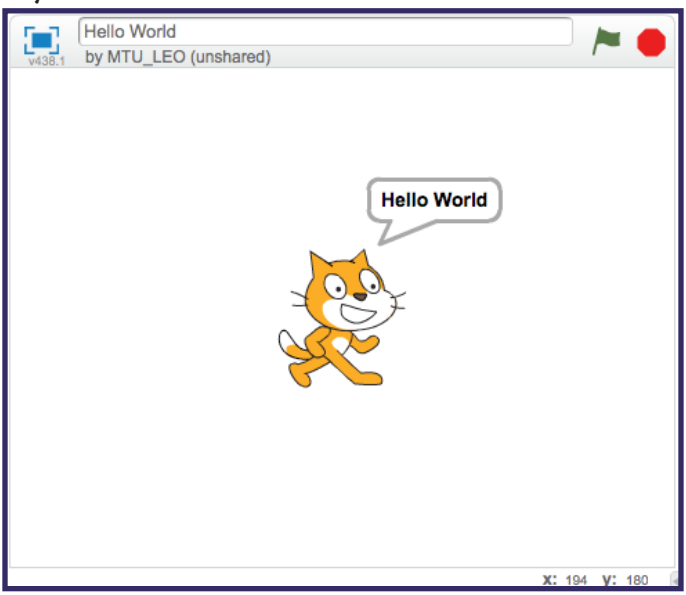


Step 3) Double click the word "Hello!" and change it to "Hello World".



That's it. You have written your first program in Scratch.

Running your program. To run your program, click on the green flag above the stage area. You should see your sprite say "Hello World."



Drawing a Square

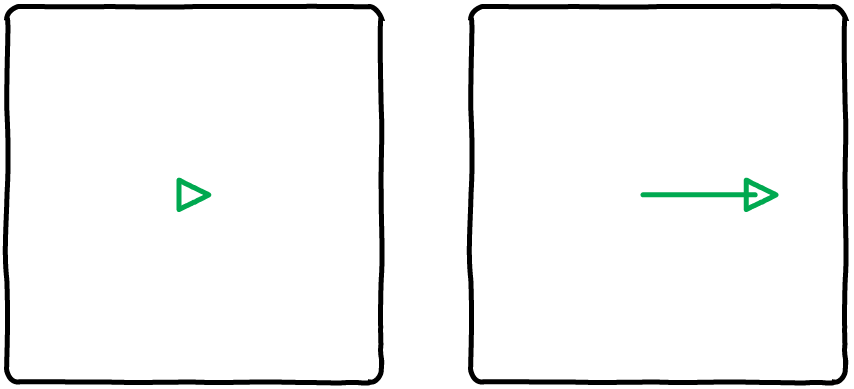
03

Drawing in Scratch is a lot like drawing with a pen and paper, except in order to draw something in Scratch, you have to tell the computer in which direction to draw the line and how far to draw the line.

By convention, we will use a sprite that looks like a green triangle when drawing as an aid to visibility. You can think of this triangle as a pen. Whichever direction the pen is pointing is the direction the line will be drawn. It is ok to continue using the cat sprite or you can pick your own shape.

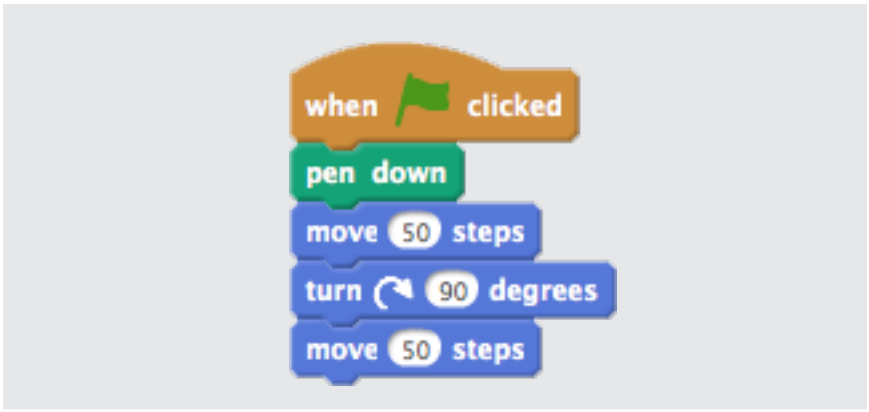
Step 1) The first thing we need to do is draw a straight line. To do this, we will need to put the pen down. ***Pen down*** is a command block in the ***Pen*** section of the block palette. Then we will need to move the sprite. ***Move*** is a command in the ***Motion*** section. Set move to 50 steps.

Step 3) When you click on the flag above the stage, the pen will move across the screen and there will be a line from where the pen started to where the pen ended.

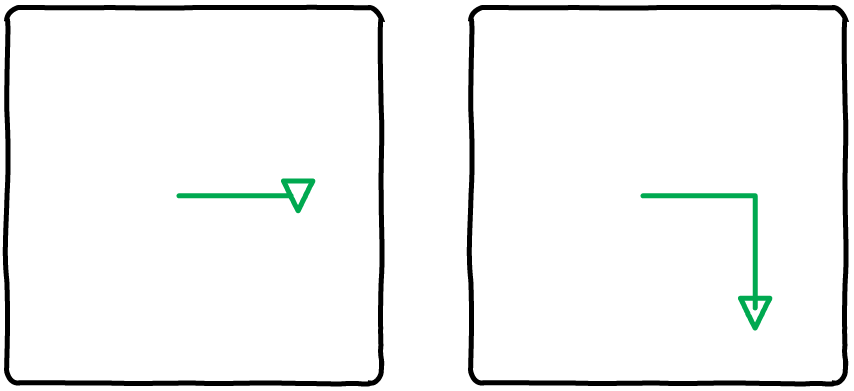


Step 4) Now we want to draw the side of the rectangle. To change the direction of pen use the ***turn clockwise*** block. Set the turn to 90 degrees. This tells the program to rotate the pen 90 degrees to the right. The pen will change direction and will now be facing down.

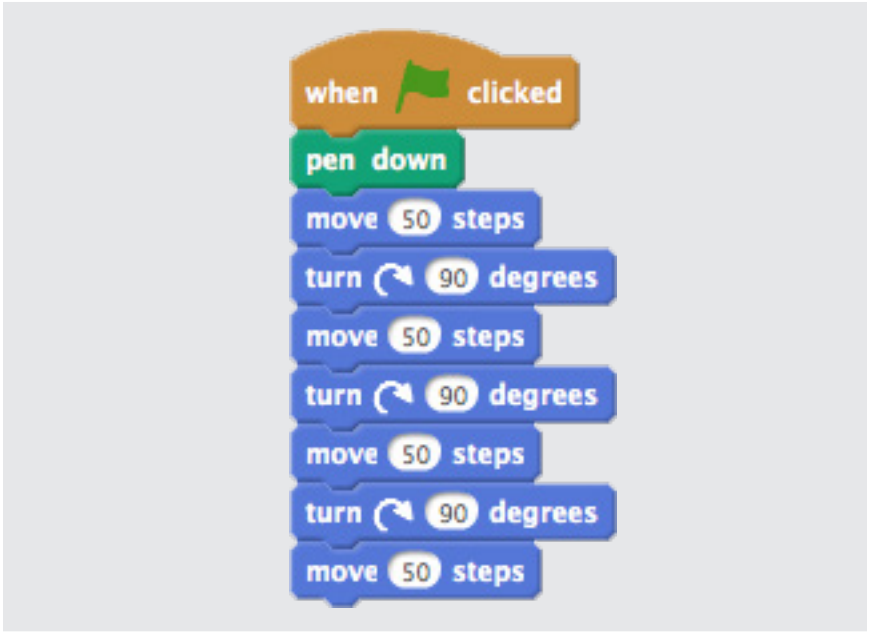
Next, draw the side if the square by connecting another ***move 50 steps*** block. Your program should look like this.



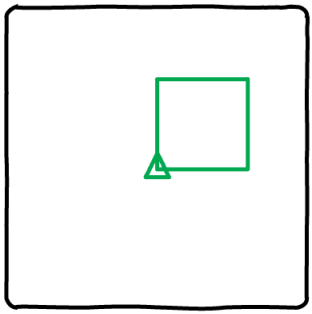
And when your program runs, it should draw two sides of a square like so:



Step 5) You can see where this is going. Snap together a couple more turnand move command blocks to complete your program.

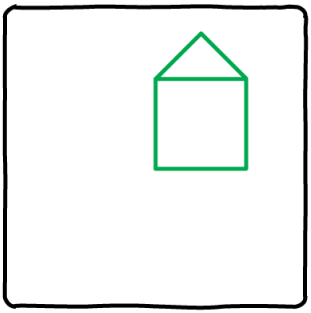


We manage to create a square! In future chapters we will learn how to draw triangles and circles.

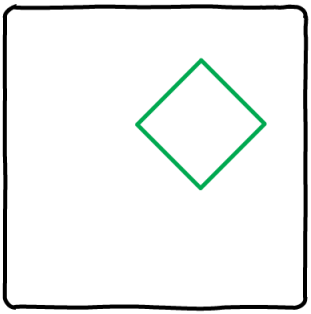


Ch 2: Practice Problems

1) Can you draw a house by adding a triangle on top of the rectangle we drew in this chapter?



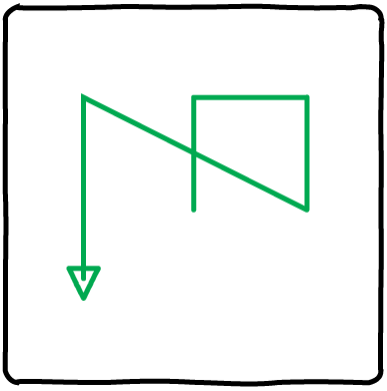
2) Try drawing a square that has been rotated 45 degrees. It is very similar to the rectangle that we drew in this chapter except the intial angle of the pen is 45 degrees instead of 90 degrees.

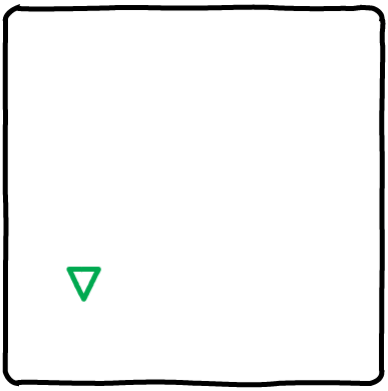


Deleting Your Drawing

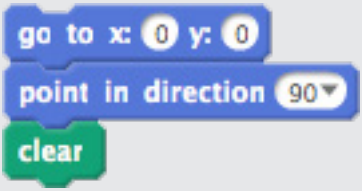
04

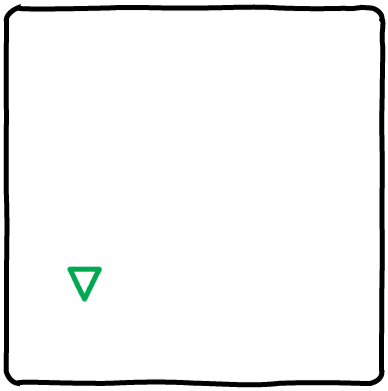
Clearing Your Drawing After you have experimented with drawing, you may want to clear the stage so you can start anew. The fastest way to clear the stage is to use the ***clear*** command from the ***Pen*** section of the block palette. (Notice the command are color-coded by section.) Drag the clear command to the script area and double click it

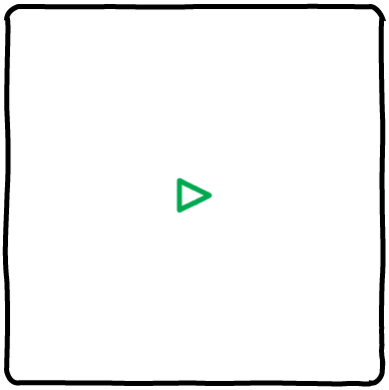




Return Home If you want to return your triangle to the middle of the screen, you can use the commands:

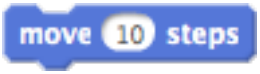
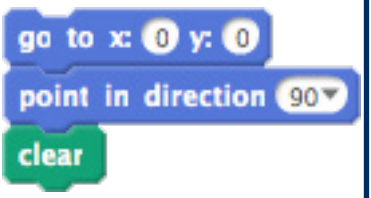
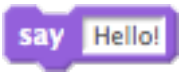






Common Tasks

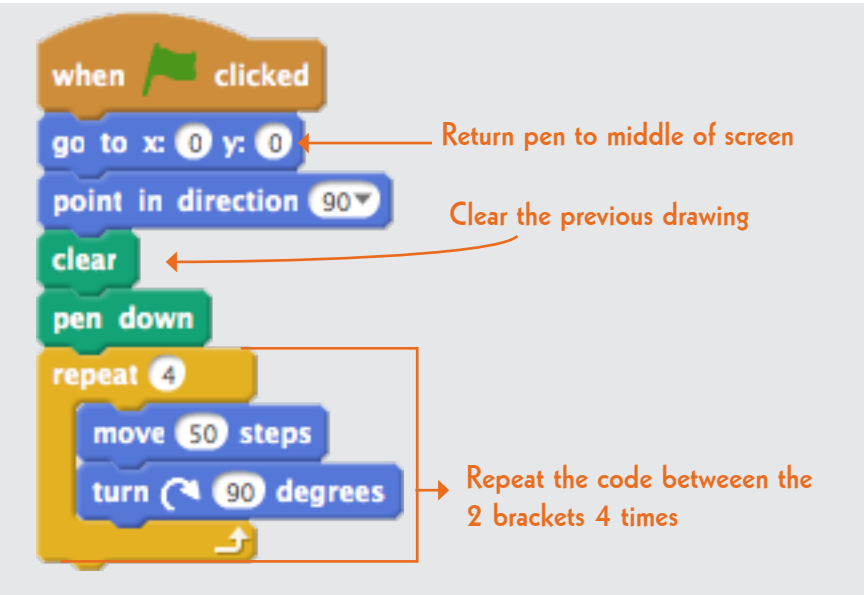
Common Commands

Command	The command does:
	Start a sequence of commands running when the flag above the stage is clicked.
	Go forward a number of steps (or backward for negative steps)
	Rotate pen to right
	Rotate pen to left
	Clear the drawing
	Return pen to the middle of screen
	Do not draw a line as sprite moves
	Draw a line as sprite moves
	Make the sprite say something in a speech bubble.

Repeat Command

05

You may have noticed that in the 2nd program we used the commands "move 50 steps" and "turn clockwise 90 degrees" four times each. There is a way to shorten this task so we only have to type each command once. We can do this by using the repeat command. Instead of dragging multiple copies of the same command in to the script area, we are going to wrap them in a repeat command:

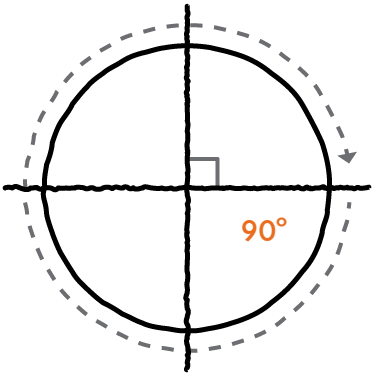


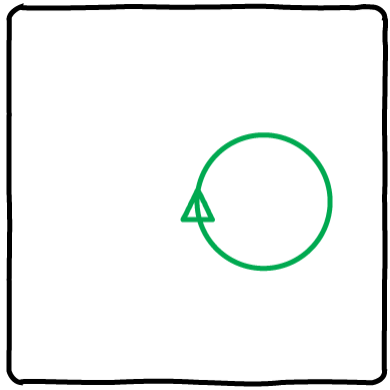
Drawing a circle on a computer is different from draw-

Drawing a Circle

ing a circle by hand. A computer can only draw straight lines, but we can create the *appearance* of a curved line by drawing a bunch of of tiny lines that, when connected together, look like a circle.

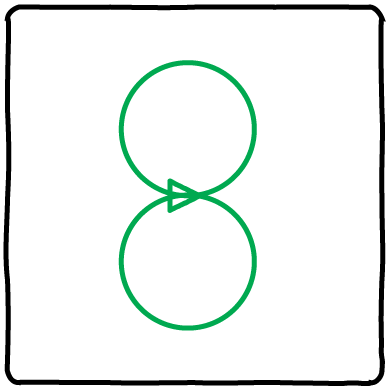
Since a circle has 360 degrees, we can create a circle by making 360 tiny lines, each one pixel long. After each line is drawn, we turn the pen 1 degree to the right.





Try the following:

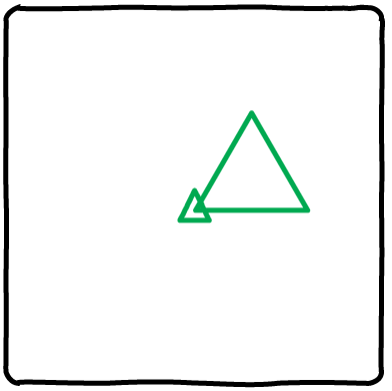
What if you wanted to draw a circle to the left? You could repeat the same commands except change the command "right" to the command "left."



Hint: To make the circle bigger we can change the length of line (For example: move 3).

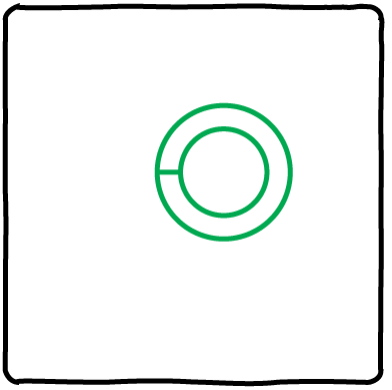
Drawing a Triangle

We can also use the repeat command to easily make an equilateral triangle (a triangle in which the length of all three sides are equal). Since each of the internal angles must also be equal we can create a triangle by rotating the pen 120 degrees after each line is drawn.

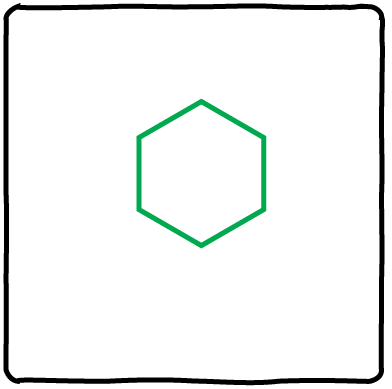


Ch 5: Practice Problems

1) Can you draw a small circle nested inside a big circle?



2) Try drawing a hexagon (a shape with 6 sides). To figure out the angles between the lines, divide 360 degrees by the number of sides.

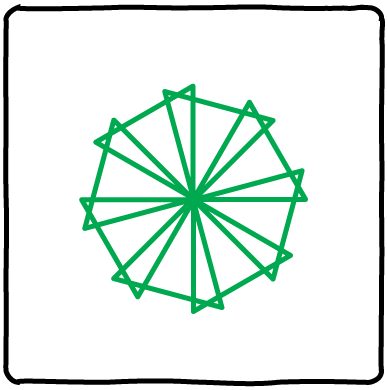


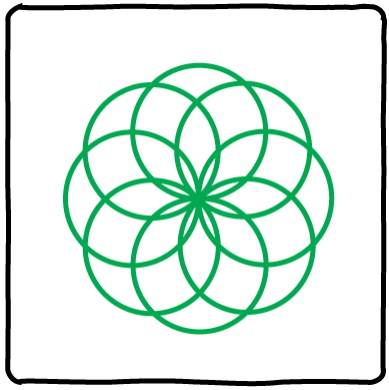
Nested Repeats

06

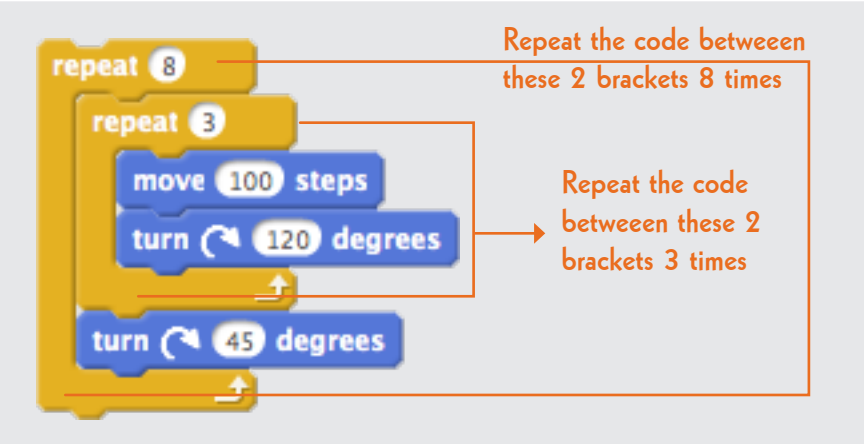
The really cool thing about repeat is that it allows us to draw the same shape as many times as we want with only a few lines of code.

In the last chapter we learned about using the repeat command. We can also nest a repeat command inside another repeat command. This allows us to build more complex shapes.





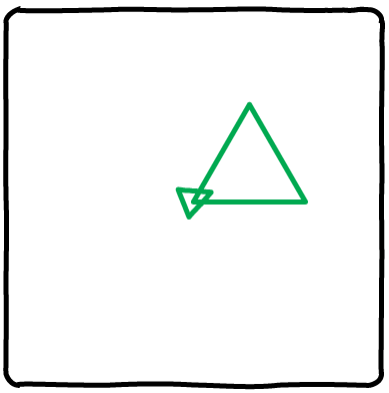
Nested loops can look a little complicated the first time you see them, so lets go through the code line by line for the drawing on the left.



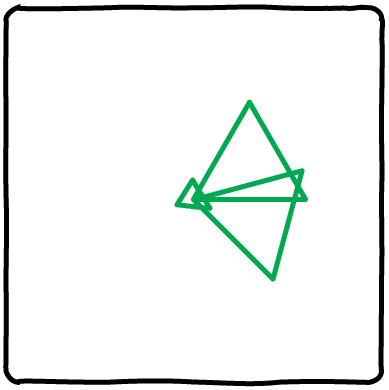
Step 1) First, Scratch does the inner repeat command repeat 3[move 100 steps, turn right 120 degrees]. This creates a triangle.

Step 2) Scratch then does the turn right 45 degrees. This turns the pen right by 45 degrees.

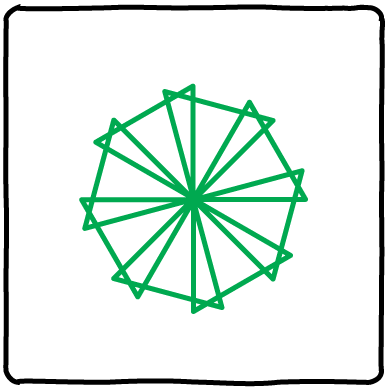
Step 3) After these two steps, Scratch has drawn a triangle. Scratch is now also ready to draw the next triangle. The screen looks like this:



Step 4) Scratch then does steps 1 and 2 again. The screen now looks like this:



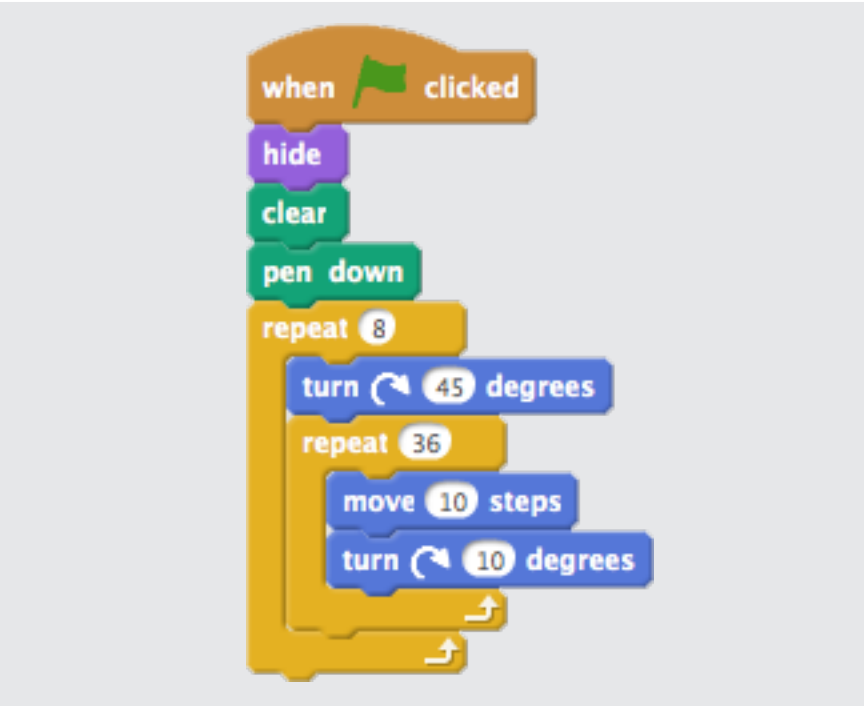
Step 5) Scratch continues to do steps 1 and 2 until it has done them 8 times. After the 8th time, Scratch stops. The final screen now looks like this:



Nested Repeats with Different Shapes

We can make lots of interesting shapes by repeating a simple shape over and over again. Other shapes that we could use as a base are circles, pentagons (shapes with 5 sides), octogons (shapes with 8 sides).

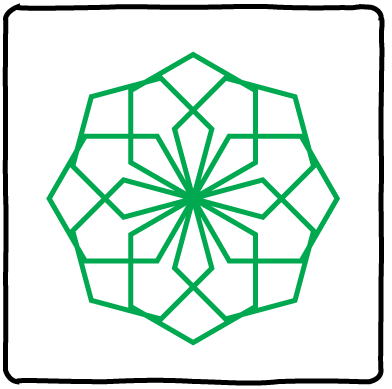
To make the shape above, a circle was drawn 8 different times. After each circle was drawn, the pen was rotated 45 degrees to the right. The code for the shape is as follows:



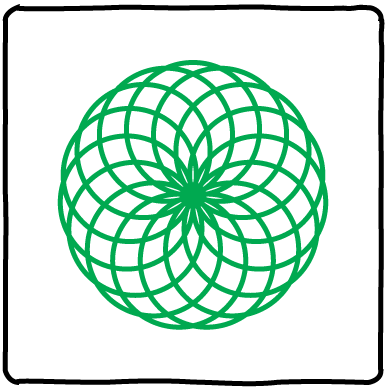
What would happen if you repeated the outer loop 36 times? Or if you changed the angle the pen rotates from 45 degrees to 10 degrees?

Ch 6: Practice Problems

1) Try to draw the shape below, which is just a repeat of a hexagon shape (a shape with 6 sides).



2) Edit the nested code for the circle shape in this chapter so there are more circles in the shape.



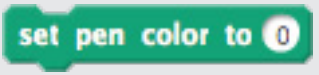
Making Colors

07

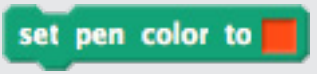
So far we have only been able to draw with the color black, but we can change the color of the pen to any color that we want.

Pen colors range from red to yellow to green to cyan to magenta and back to red.

The ***set pen color to*** block is a Pen block. The block takes a number between 0 and 199. A value of 200 is the same color as a 0 pen color value; in other words, if you change the pen color by 200, the color still looks the same.

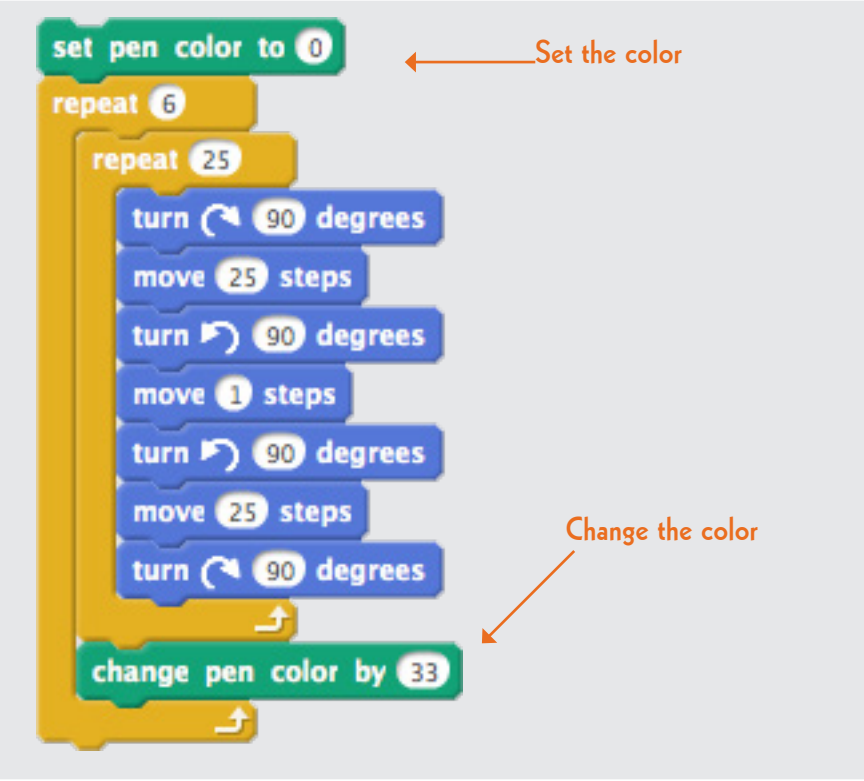


The easiest way to create a color is to use the color picker version of the ***set pen color to*** block. When you click in the color area, then click on a color anywhere else in the Scratch environment to use that color.



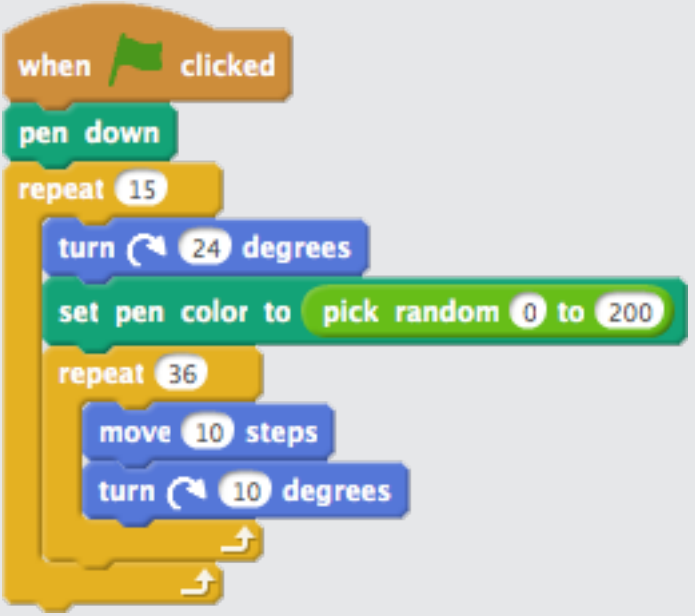
Common Colors			
Color	RGB Value	Color	RGB Value
RED	0	YELLOW	33
GREEN	66	CYAN	99
BLUE	132	MAGENTA	165

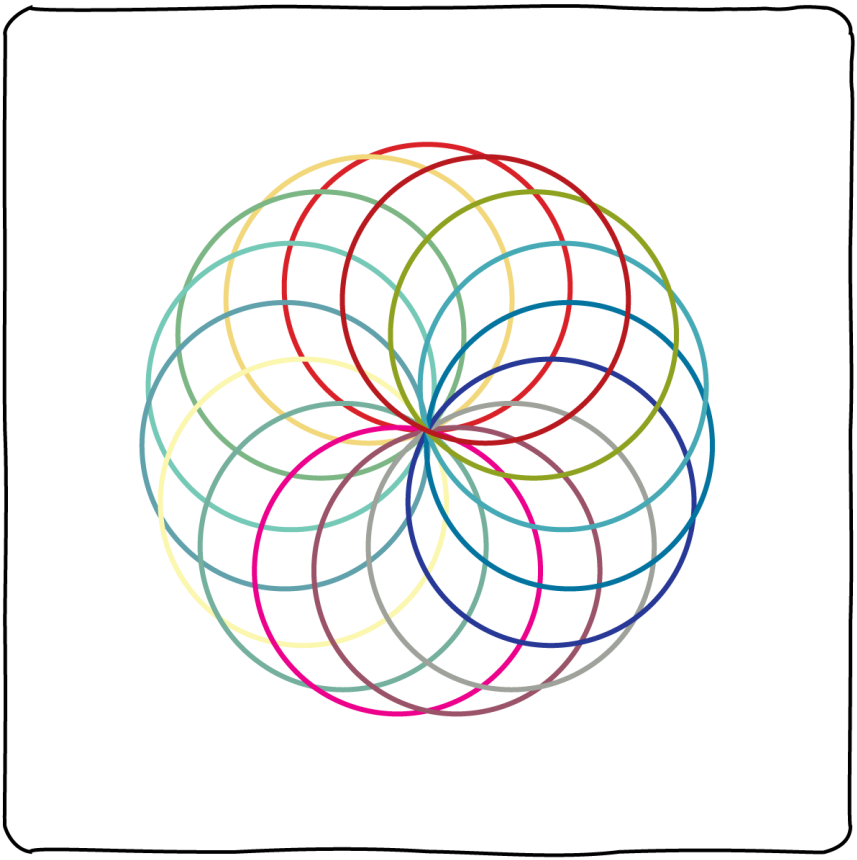
How To Use the Color Codes In the code example below, the color is set to red before the rectangle is drawn. Notice that the values in ***change pen color to*** are the same as the color values in the table above.



Random Colors To create random colors, we need to generate random values; there is an **Operator Block** for that allows us to pick a random number between two values.

The following code generates a new color every time one circle is drawn.



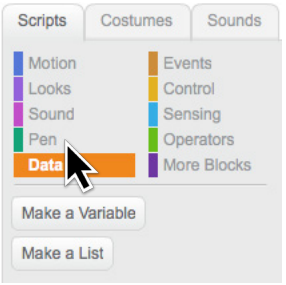


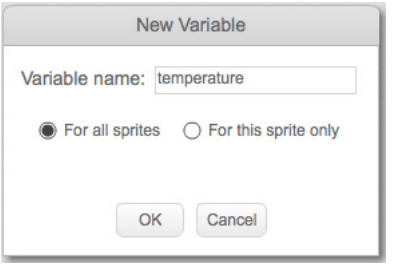
All About Variables

08

What is a Variable? Sometimes we want to be able to keep track of a value that might change as the code executes. In order to keep track of that value, we need to store it memory so we can access it again. Variables are a way for programmers to store values. It is helpful to think of a variable as a box that can only store one value. When a new value is put in the box, the old value is destroyed forever.

How to Create a Variable Go to the Data Section in the Block Palette and click the Make A Variable button. Then enter the name of the variable.





You will then see some new command blocks in the Data Section that can be used to set the value of your variable.

Changing the Value of a Variable At some point we will want to change the value stored in a variable. We can set the value to a specific number or we can perform some math on the variable already being stored.

Doing math is a little bit different in Scratch than what you remember from math class. Instead of using a symbol like + for addition, we use the word "sum."

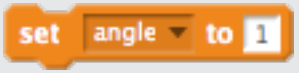
Math Commands

Command	Math	Command In Scratch
add	2 + 4	2 + 4
subtract	5 - 9	5 - 9
multiply	2 * 2	2 * 2
divide	8 / 4	8 / 4

Project: Make a Spiral

How variables work

When we first start the program, the value stored in 'angle is 0. As the program progresses, the we may want to change the value stored in the variable. To do this, we can use the ***set*** block to change the value of the variable.

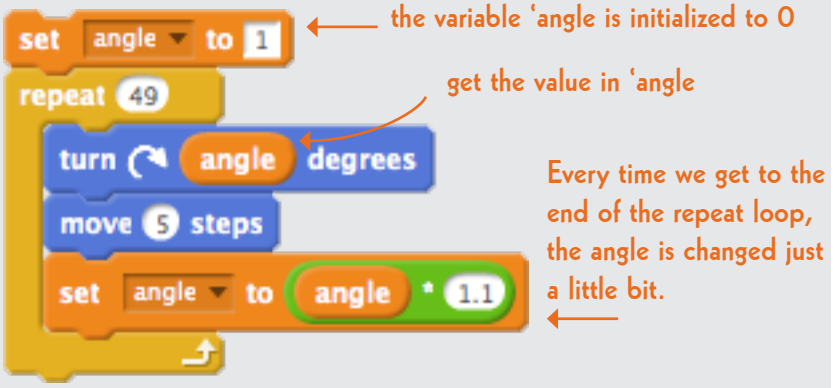


Getting the value of a variable

After we make a new variable, there is a new oval block with the variable name in the Data Block Section. This

variable block can be dragged onto any other block that takes a number as an argument. When a command block has a variable as an argument it substitutes the value of the variable for its argument.

When we draw a spiral, we need to adjust the angle of pen after each line segment is drawn. Now that we have learned about variables, it will be easy for us to draw a spiral because we can store the angle value in a variable.



Ch 8: Practice Problems

1) True or False. Are these two variable names the same: ‘myFirstVariable and ‘MYFIRSTVARIABLE?

2) What is the final value stored in w after the program executes?

The final value of w is: _____

3) What is the final value stored in w after the program executes?

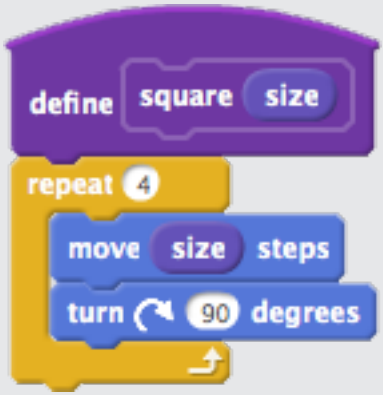
The final value of w is: _____

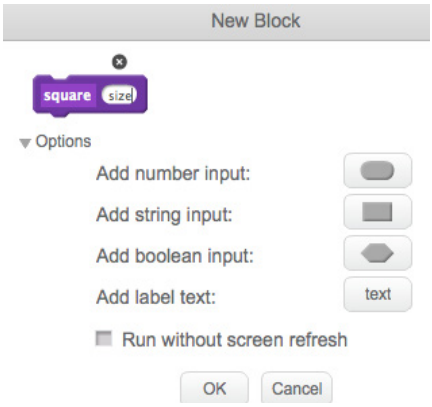
Awesome Functions

09

Why Use Functions? When we used the repeat command, it made drawing complex shapes really easy. Now we will learn about functions, which allow us to repeat the same code over and over again without having to rewrite the code. A function may sound an awful lot like the repeat command, but function allows us to adjust certain parts of the code based on our needs. With a function, we can use one block of code to draw different size squares!

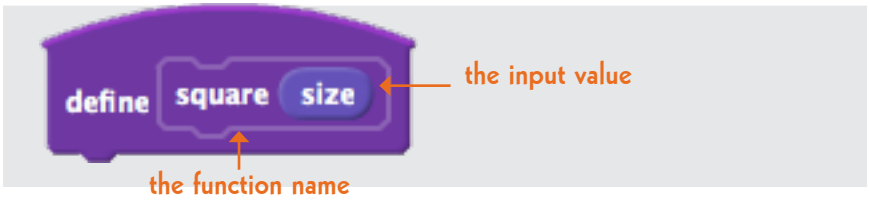
Formatting Functions Functions have to be written in a particular way. The format of a function is as follows:





To create a function, go to the **More Blocks Section** and click on the "***Make a Block***" button. Then enter a name for your function. If your function needs to use input values, click on one of the options and enter a name for the input.

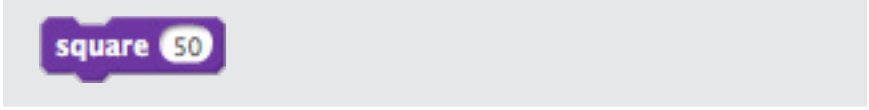
define Making a block will create a ***define*** block. The code attached to the define block comprises a function



square is the name of the function

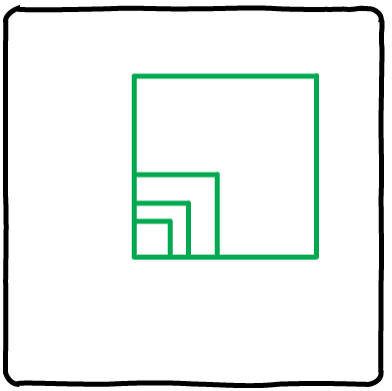
size says the command will have an input and in the function, its name will be "size." A function can have multiple inputs.

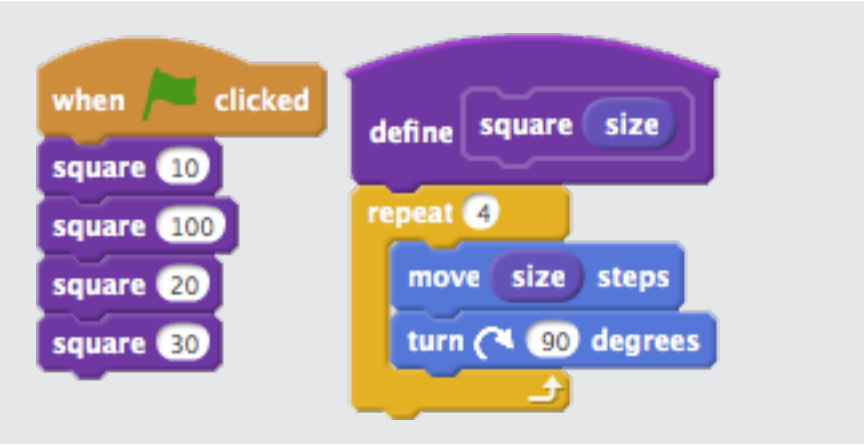
Calling A Function You can make a script attached to a define block and click "run", but nothing will happen. That is because we have to tell the computer to call the function. The way we call the function above is by using the following command block, which is created when we make the function.



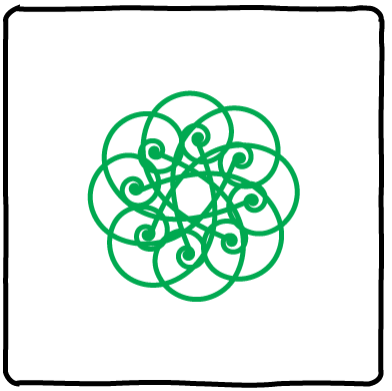
Project: Drawing Squares

Let's use a function to draw a bunch of different size squares. Can you draw some other size squares?

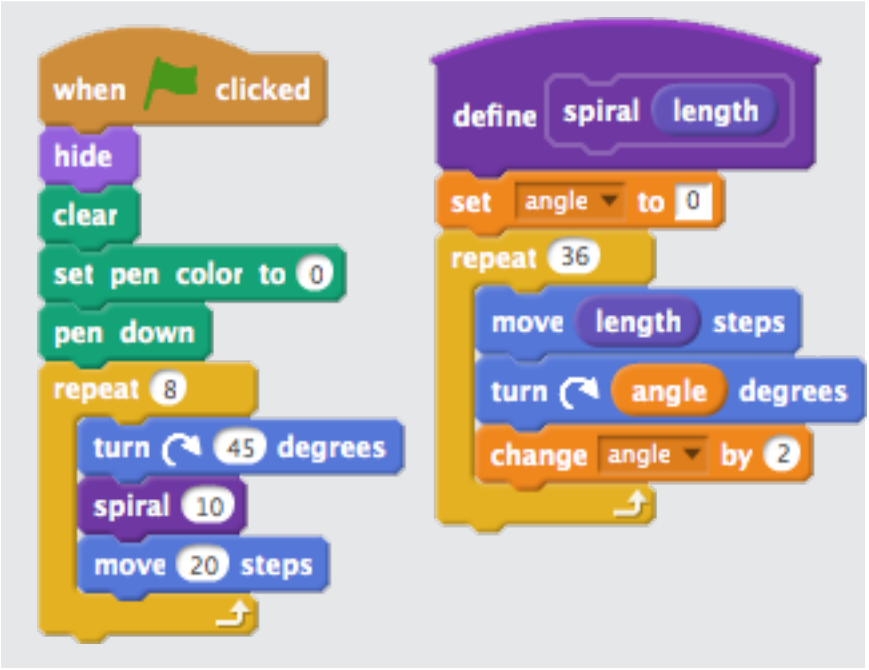




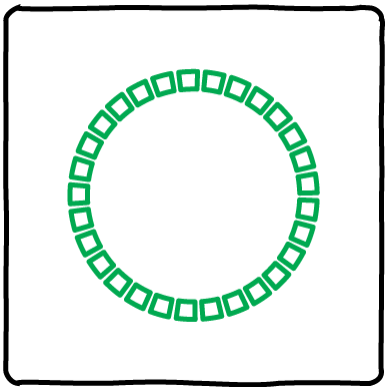
Project: Spiral Rose



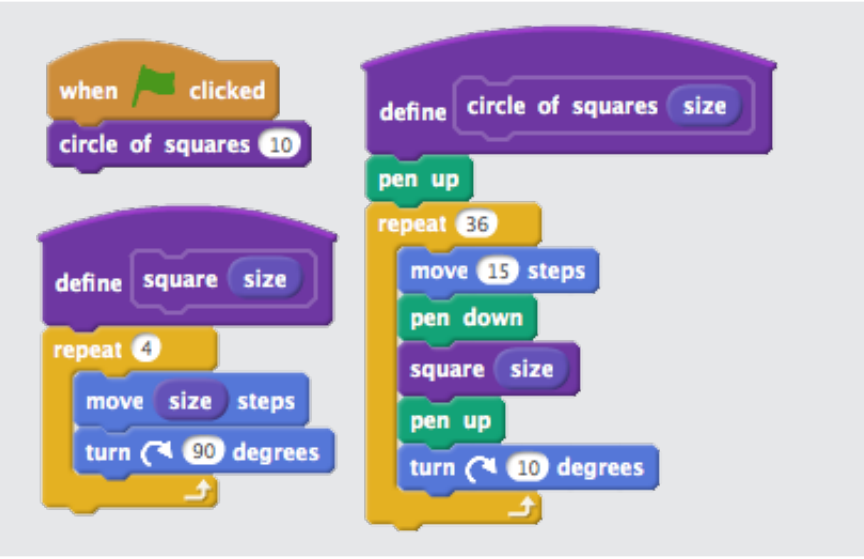
We can combine repeats and functions to make the spiral rose above.



Project: Circle of Squares

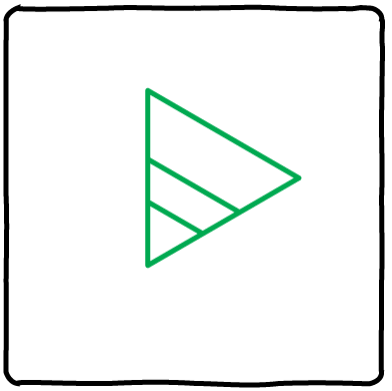


Lots of fun things can be created with functions like a circle made out of squares. The command penup is used to stop the pen from making marks while it is moving.

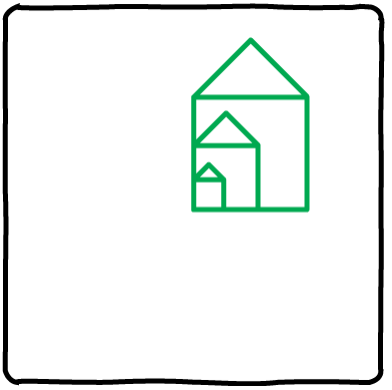


Ch 9: Practice Problems

1) Can you make a function that will help you draw different size triangles?



2) Make a function that will let you draw different size houses.

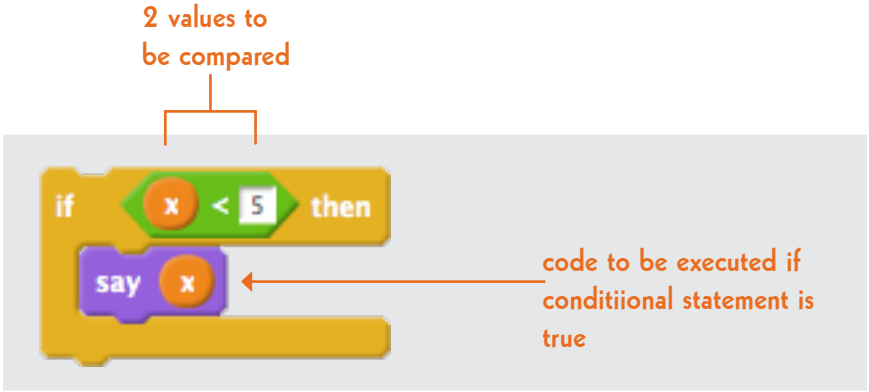


If Statements

10

What if you wanted your pen to do different things based on the current value of a variable? If the variable's value is greater than zero, the pen should rotate 45 degrees. If the variable's value is equal to zero, the pen should rotate 90 degrees. An easy way to program this is to use an if statement.

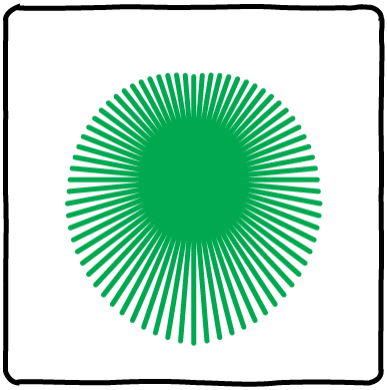
If Statement The if statement tells your program to execute a certain section of code only if a particular condition is true. If it is true, the code between the brackets will be executed. If it is false, Scratch will simply ignore the code between the brackets.



Conditional Statements Scratch lets you compare values to see if one is greater than the other, less than the other or equal to the other value.

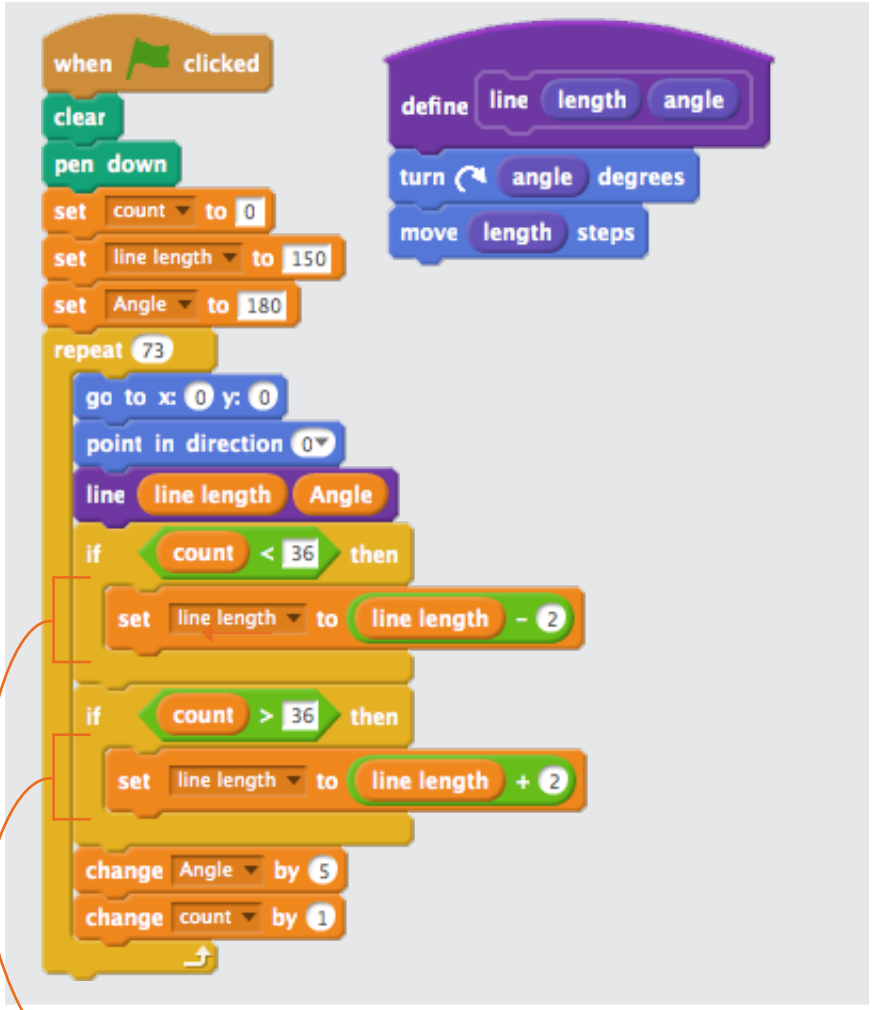
Conditional Statements		
Command	**Example**	**Command In Scratch**
equal	5 = 6	
less	10 < 3	
greater	8 > 5	

Project: Radiating Lines



To make the image to the left, we used two if statements to adjust the length of the lines.

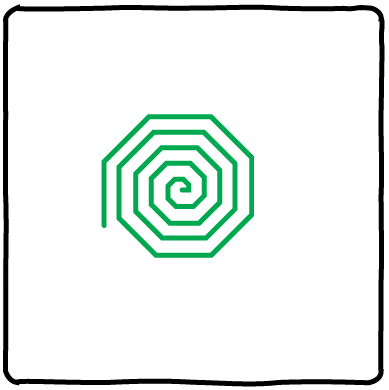
The repeat command is going to run 73 times. We are going to use a variable called 'count to keep track of what number repeat we are on.



if the count is less than 36, decrease the line length

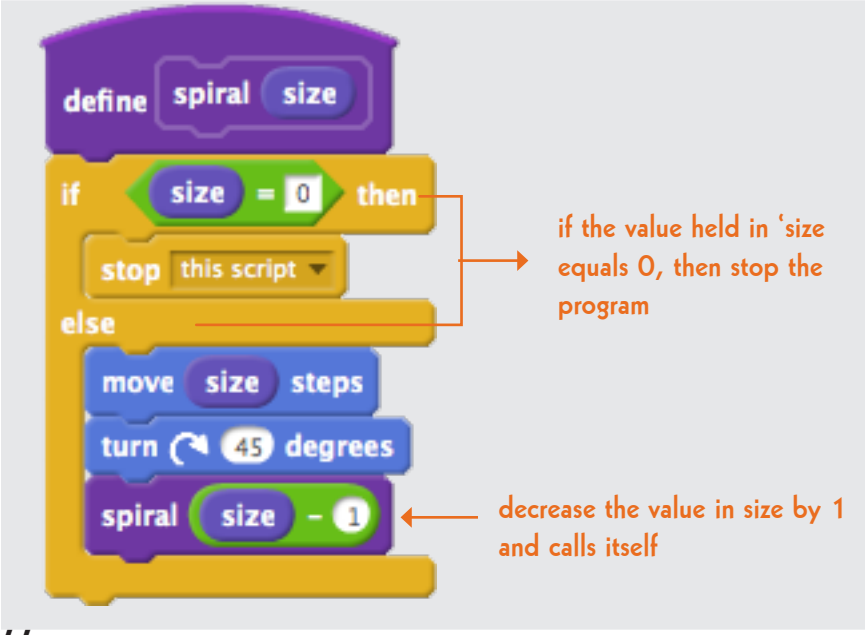
if the count is greater than 36, increase the line length

Project: Recursive Spiral

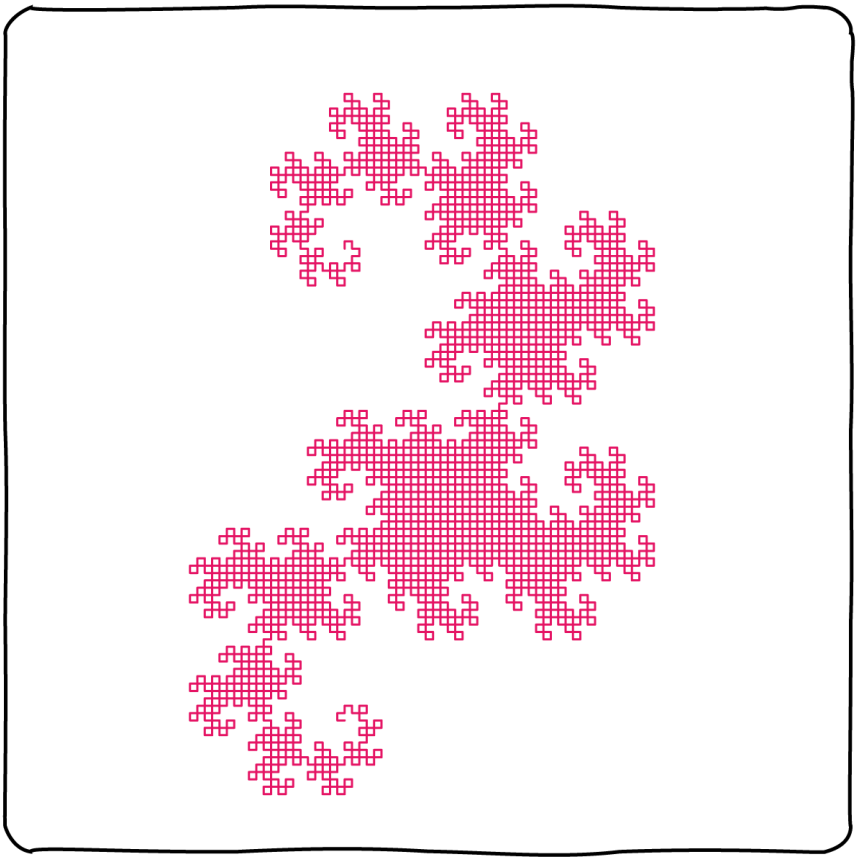


An interesting thing about functions is that they can call themselves. Whenever a function calls itself, it is called **recursion.** The only problem with this techniqe is that we need a way to stop the calling process or else the program will run forever!

If loops are helpful for stopping the program because we can tell the program to stop once a certain condition has been met. For example, we could initially call the loop with the variable ***size*** that has been initialized to 100. Each time the loop calls itself, the variable decreases in by 1. When the value of ***size*** reaches 0, the program will stop.

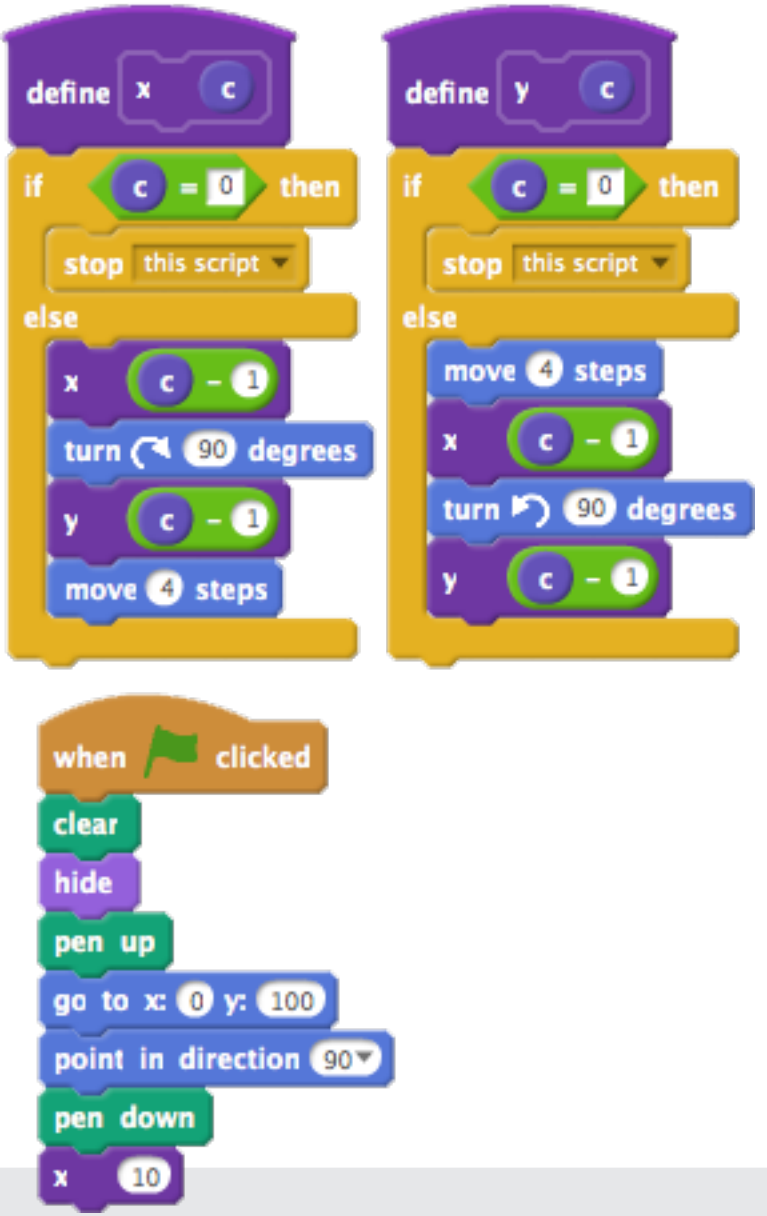


Project: Dragon Curve

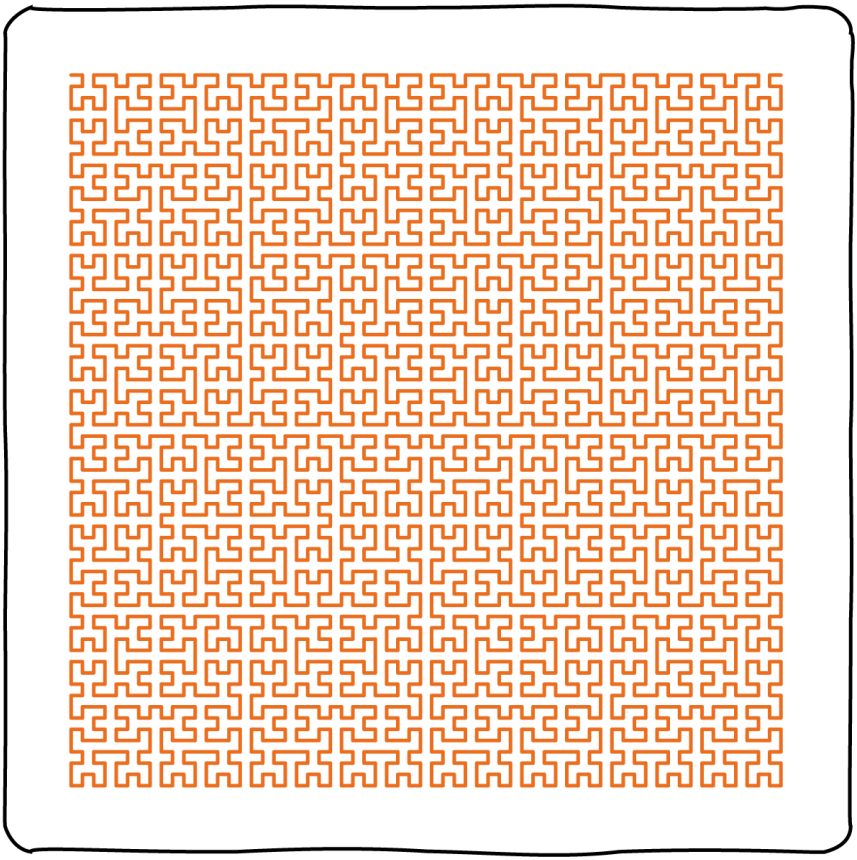


Programmers like recursion a lot because it lets them make really complicated drawings using relatively few lines of code. If we tried to draw the dragon curve above only by using functions and repeat loops, it would take us hours, and hundreds of lines of code, to complete.

Try experimenting with the code below by calling the function with different inputs (for example call the function x on the last line of code with x 6 or x 13).

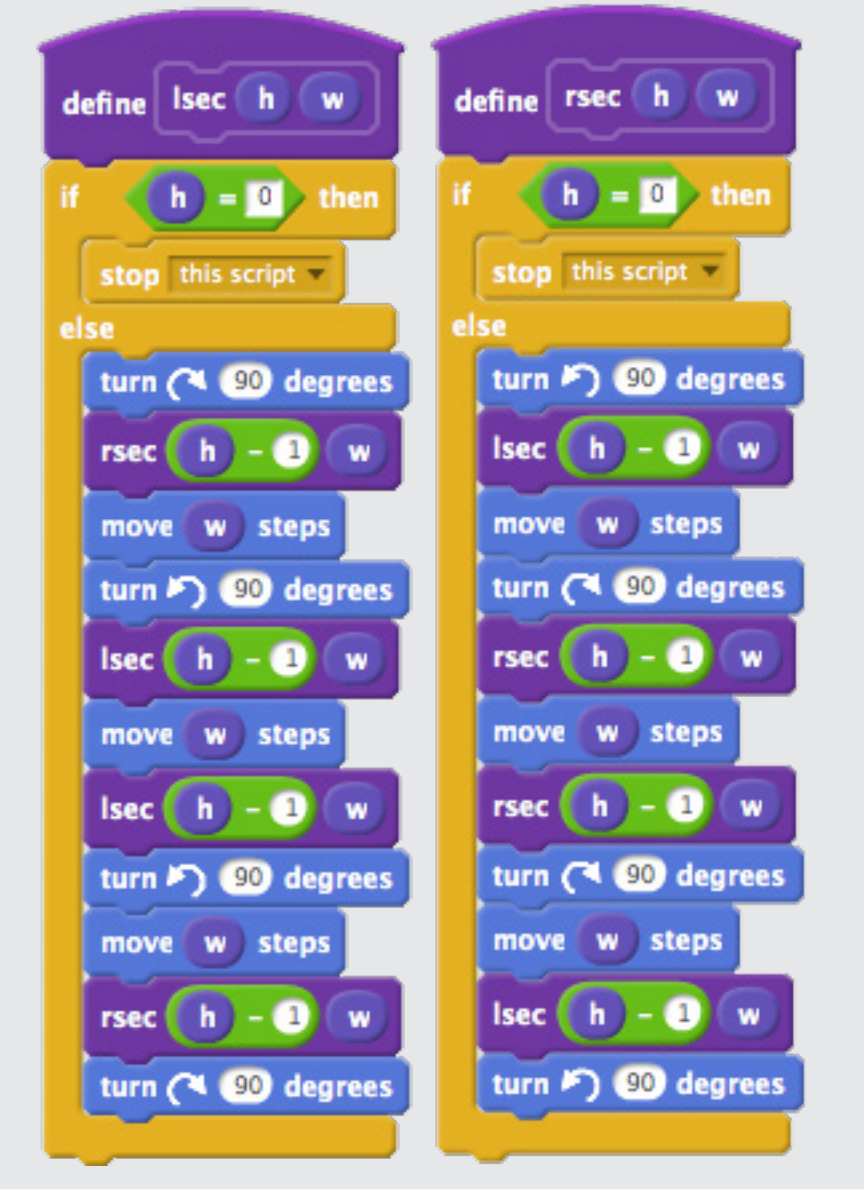

define x c
if c = 0 then
stop this script
else
x c - 1
turn 90 degrees
y c - 1
move 4 steps
define y c
if c = 0 then
stop this script
else
move 4 steps
x c - 1
turn 90 degrees
y c - 1
when clicked
clear
hide
pen up
go to x: 0 y: 100
point in direction 90
pen down
x 10

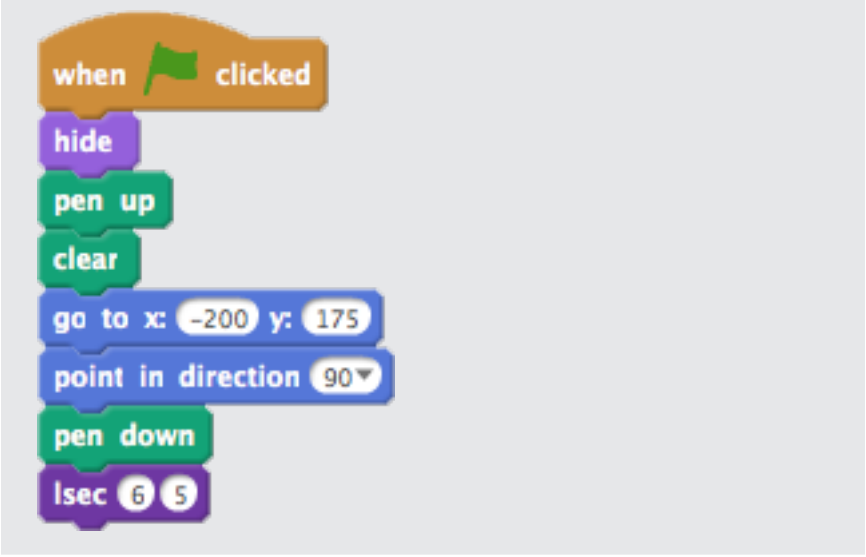
Project: Hilbert Curve



The labyrinth above is called the Hilbert curve. Like the dragon curve, it is really easy to draw with a recursive function.

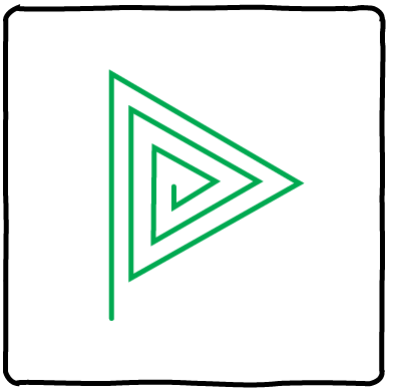
After you copy the code into Scratch and run the program, try experimenting with altering the code. For instance, try changing the inputs for the LSec function call. For example change the inputs in the last line of code to LSec 5 5 or LSec 3 3).


define lsec h w
if h = 0 then
stop this script
else
turn 90 degrees
rsec h - 1 w
move w steps
turn 90 degrees
lsec h - 1 w
move w steps
lsec h - 1 w
turn 90 degrees
move w steps
rsec h - 1 w
turn 90 degrees
define rsec h w
if h = 0 then
stop this script
else
turn 90 degrees
lsec h - 1 w
move w steps
turn 90 degrees
rsec h - 1 w
move w steps
rsec h - 1 w
turn 90 degrees
move w steps
lsec h - 1 w
turn 90 degrees

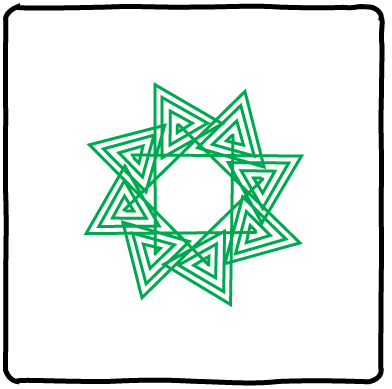

when clicked
hide
pen up
clear
go to x: -200 y: 175
point in direction 90
pen down
lsec 6 5

Ch 10: Practice Problems

1) Can you make a recursive spiral tirangle?



2) Use the spiral triangle code from above to make a star. Experiment with differnt angles and lengths to create differnt types of stars.



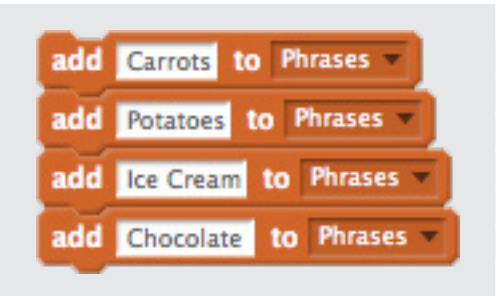
Making Lists

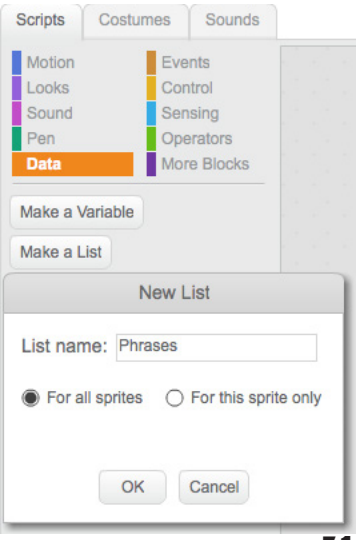
11

So far we have only used Scratch to manipulate numbers and variables one value at a time, but we can also store and manipulate lists of things.

Lists Remember when we learned about variables? Variables are a way to store one thing in memory. Sometimes we need to store multiple things in memory but we don't want to do a lot of tedious typing. Lists let us store many values in a single structure.

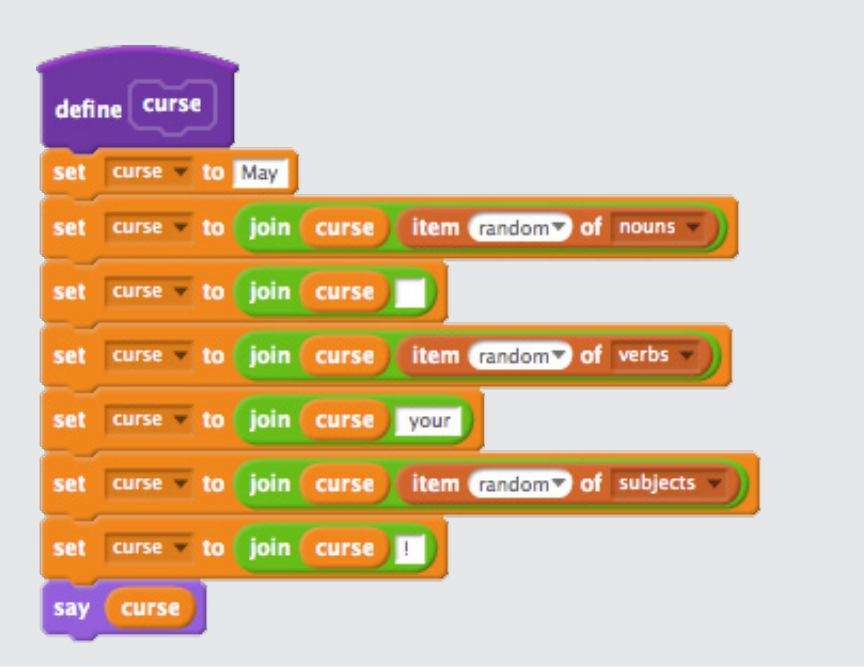
Making Lists Making lists is like making a variable. Go to the Data Section and click the "***Make List***" button then enter the name of the list. You can then use the ***add*** block to add items to the list.



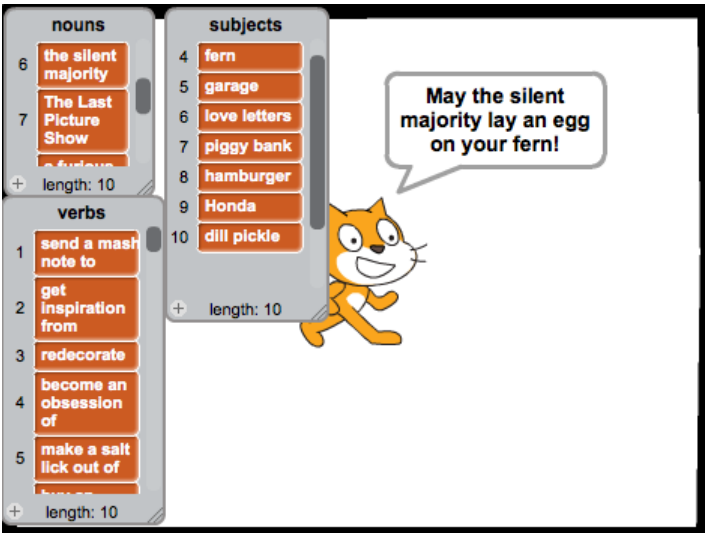


Project: Curses

This program is based on the work of Tom Dwyer and Margot Critchfield in 1978. It uses lists. What does it do?



Change this program up. Add more things to the lists. Take some things away. Change the pattern used to form the curse. Is the pattern for a blessing different from the pattern for a curse?



Project: Curses

Poetry

Many computer programs have been developed that generate poetry or music. Some of them use a technique similar to the curses program. These programs often have large lists of words that are arranged according to some predefined patterns.

For example, you might draw from lists in a pattern like this:

Title
Adjective Noun
Verb Noun
Noun Preposition Noun Verb Noun
  Ending Phrase

How could you make your poem rhyme? How could you link the Title and Ending Phrase to give your poem a sense of order and completion?

Ch 11: Practice Problems

1) Working in a group, modify your program so that is generates poetry instead of curses. Within in your group, select your three favorite computer generated poems.

2) Try making a program that generate haikus.

3) Make a Dadaist Poem in the style of Tristan Tzara:

a. Take a newspaper.
b. Choose an article as long as you are planning to make your poem.
c. Make a list containing each of the words that make up this article.
d. Make a poem by randomly choosing each word. Removing it after the word from the list after it is used.
e. The poem will be like you.

And here [is the computer] a writer, infinitely original and endowed with a sensibility that is charming though beyond the understanding of the vulgar.

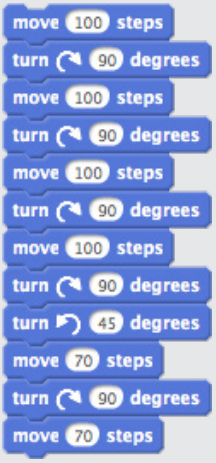
~Tristan Tzara

Problem Solutions

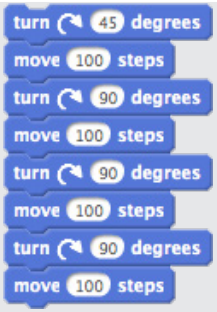
A

2: First Program

1) There are many ways to draw a house. The code below shows one way to draw a house by first drawing the rectangle and then adding a triangle to the top of the image.



2) A diamond can be drawn in many ways. The code below shows a simple way to draw a diamond.



5: Repeat Command

1) There are many ways to draw different size circles. The code below shows one way to draw two different size circles. The first repeat code draws the smaller inner circle and the second repeat code draws the bigger outer circle.

2) The easiest way to draw a hexagon is to draw 6 lines with an angle of 60 degrees between each line.

6: Nested Repeats

1) There are many ways to draw a shape consiting of hexagons. The code below describes one way to draw the shape.

2) The code below describes one way to draw a shape made out of many circles

8: All About Variables

1) False. since the computer thinks lower and upper case letters are different, ‘myFirstVariable and ‘MYFIRST-VARIABLE do not mean the same thing to the computer.

2)

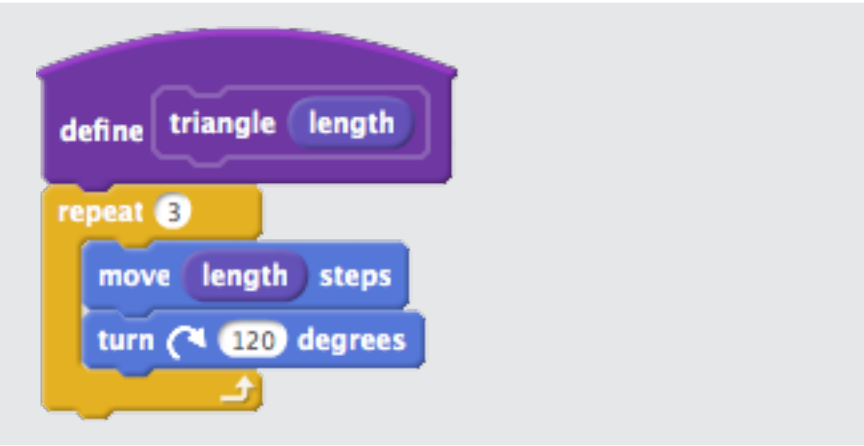
The final value of w is: 12

3)

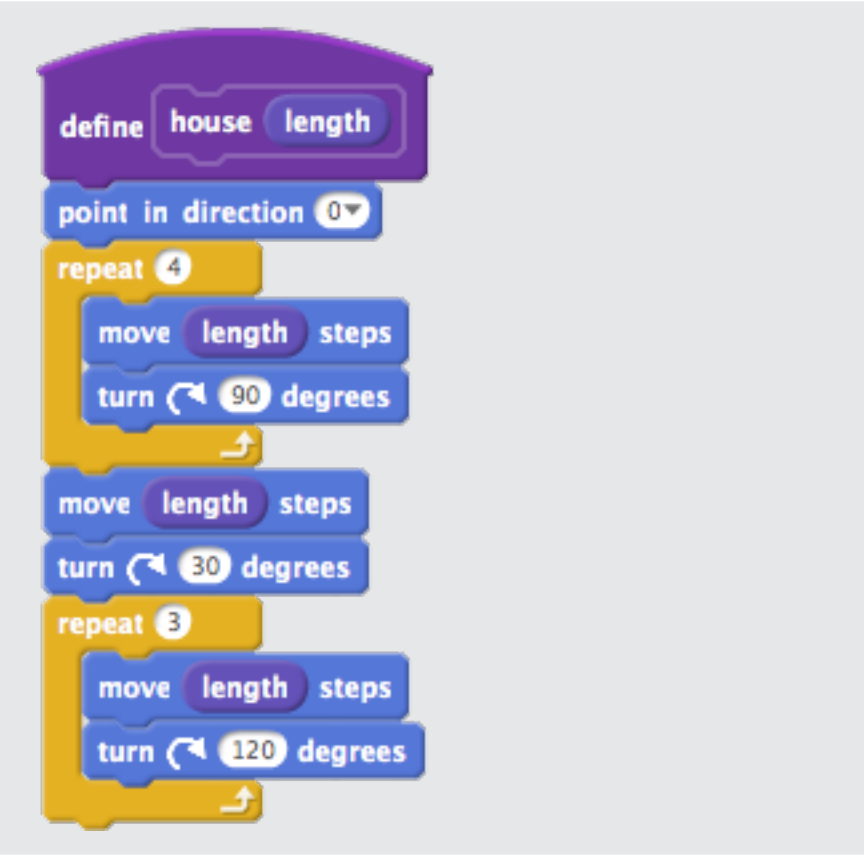
The final value of w is: 16

9: Awesome Functions

1) There are many ways create a function that draws triangles. Below is one way to do it.

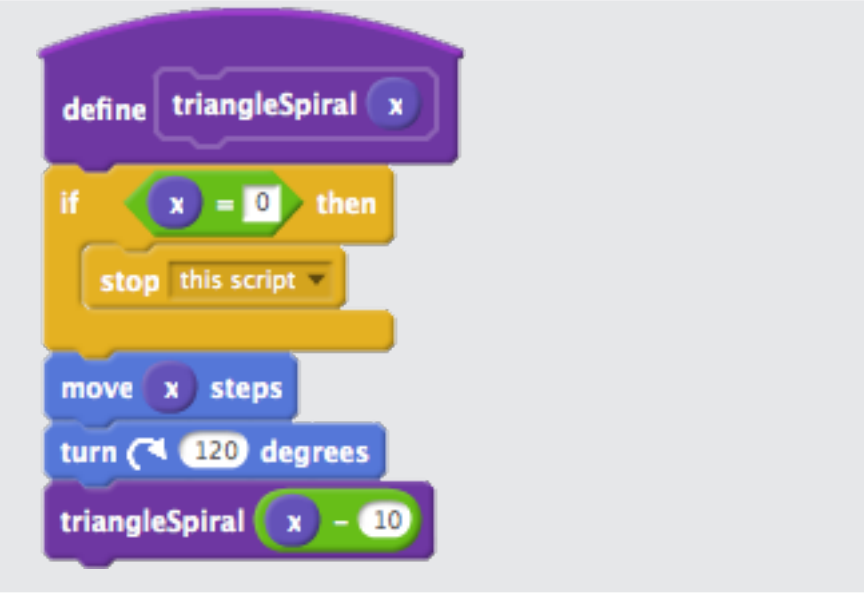


2) The code below shows you one way to draw a resizeable house.

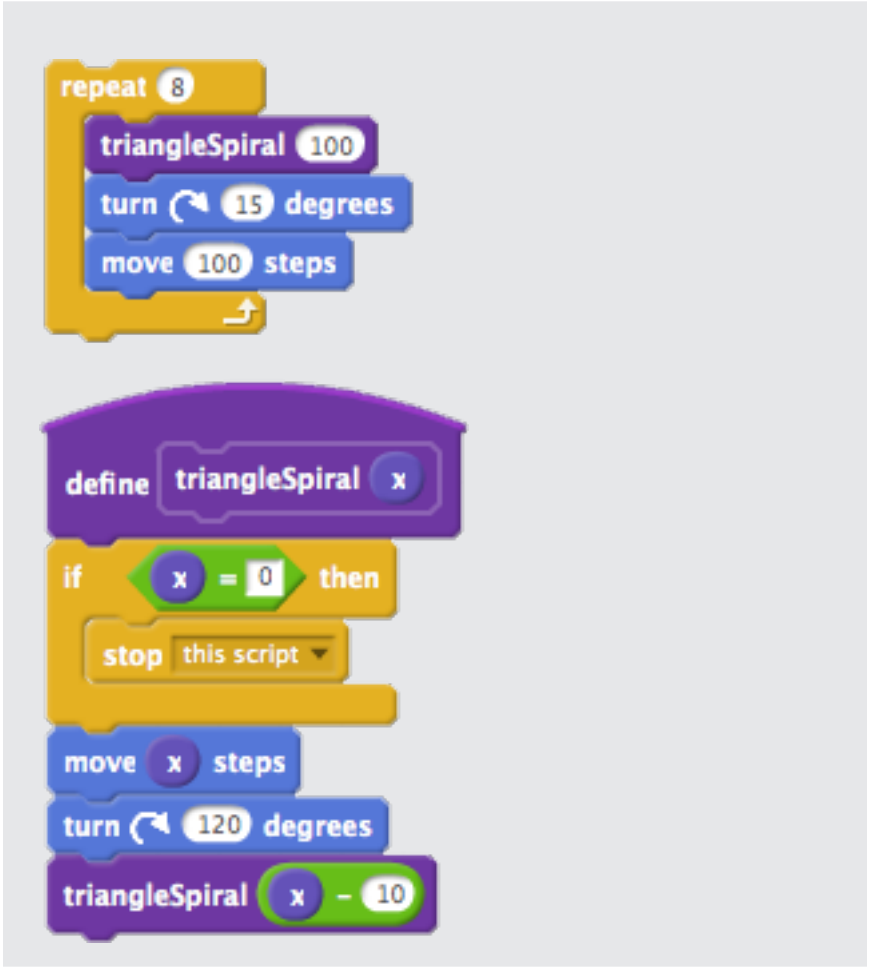


10: If Statements

1) There are many ways code a recursive triangle. Below is one way to do it.



2) To create the star in the picture provided, copy this code:



Resources

B

- Scratch Home Page http://www.scratch.mit.edu
- Snap! Home Page http://snap.berkeley.edu
- The Beauty and Joy of Computing http://bjc.berkeley.edu/website/curriculum.html
- Learn Scratch http://learnscratch.org
- Scratch for Budding Computer Scientists http://cs.harvard.edu/malan/scratch/index.php

Name:

Email:

Text and Illustrations By: Nicole Yarroch & Leo C. Ureel II