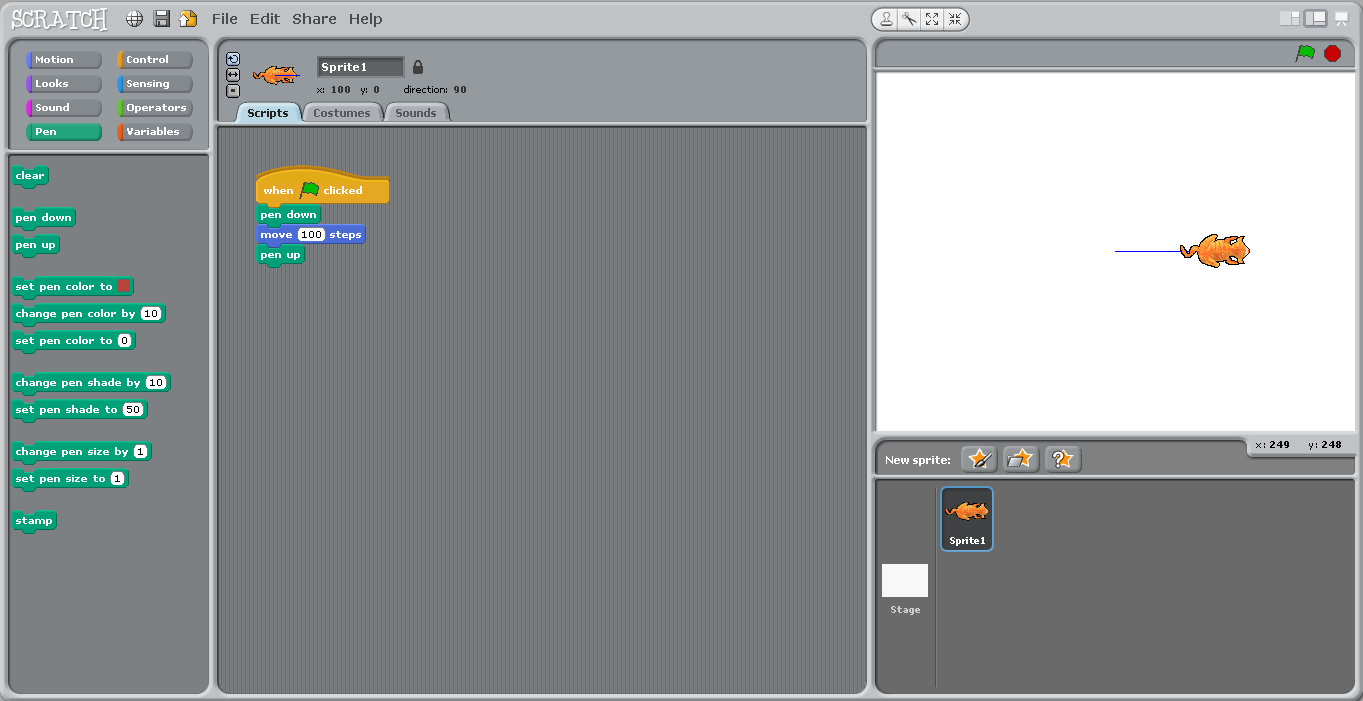
**Raspberry Pi Activity 3**

*Learning Objective:* Create shapes or line drawings using code

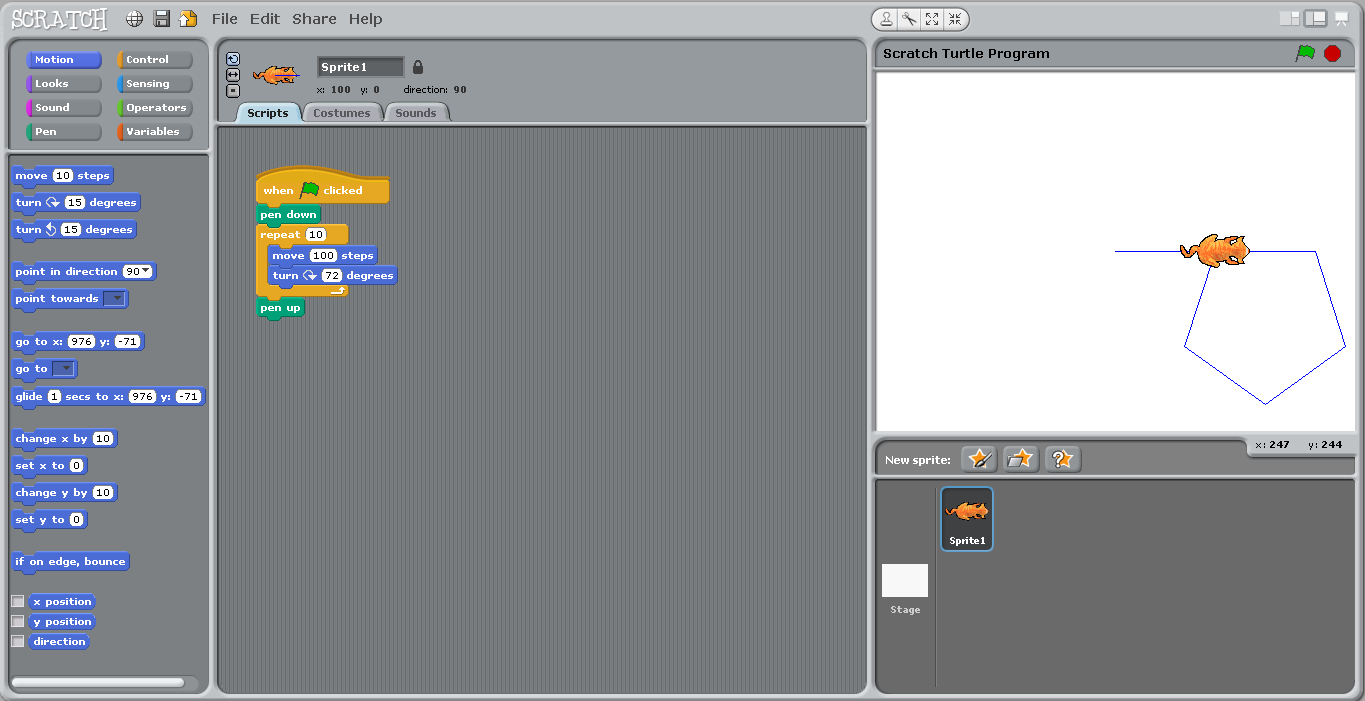
The module is called Turtle Graphics, and the sprite directing the cursor in Turtle Graphics is always called a turtle.

Modules are collections of useful code that perform specific functions.

***Task 1.*** To begin drawing a simple shape, you may find it easier to give instructions to the sprite if you have a bird’s-eye view. Select the bird’s-eye view of the Scratch cat from the Animals directory of the Sprites palette. Then, use the pen down and pen up blocks instruct the sprite to start and stop drawing.

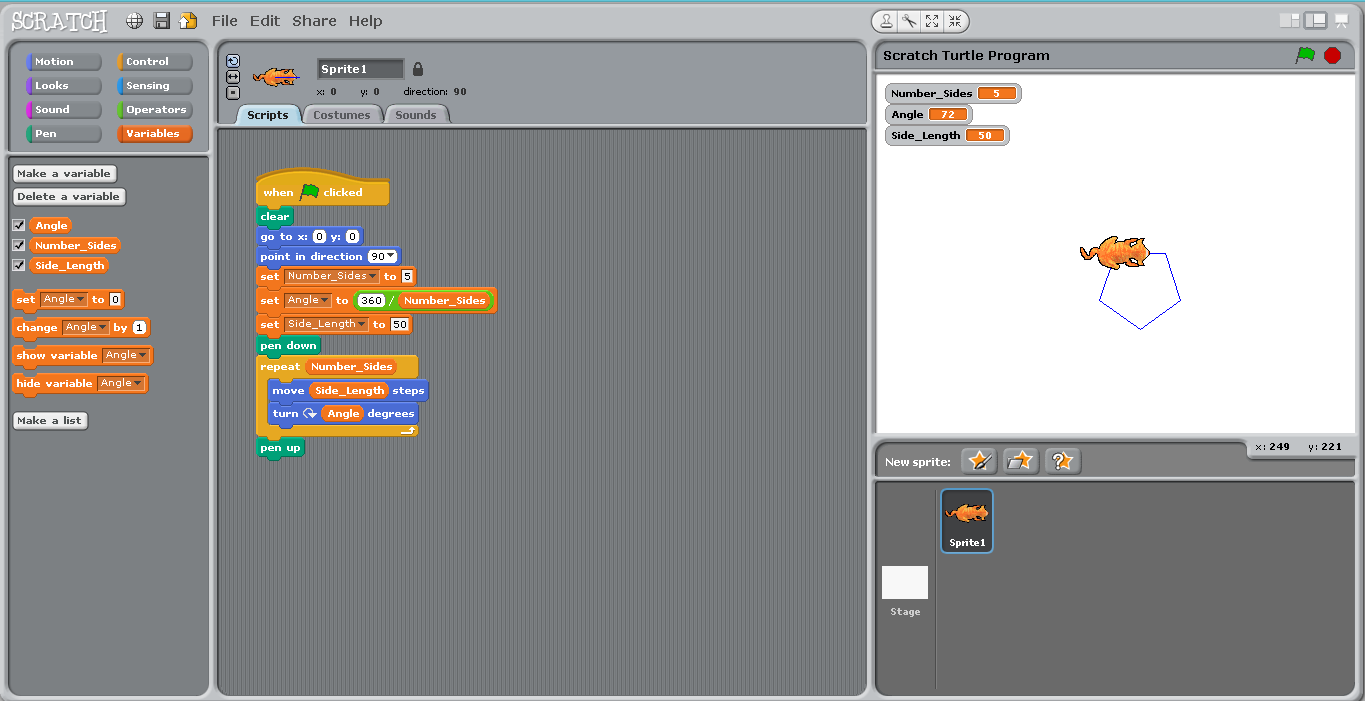


Next, add a repeat block (loop), the code can be iterated as many times as you want.

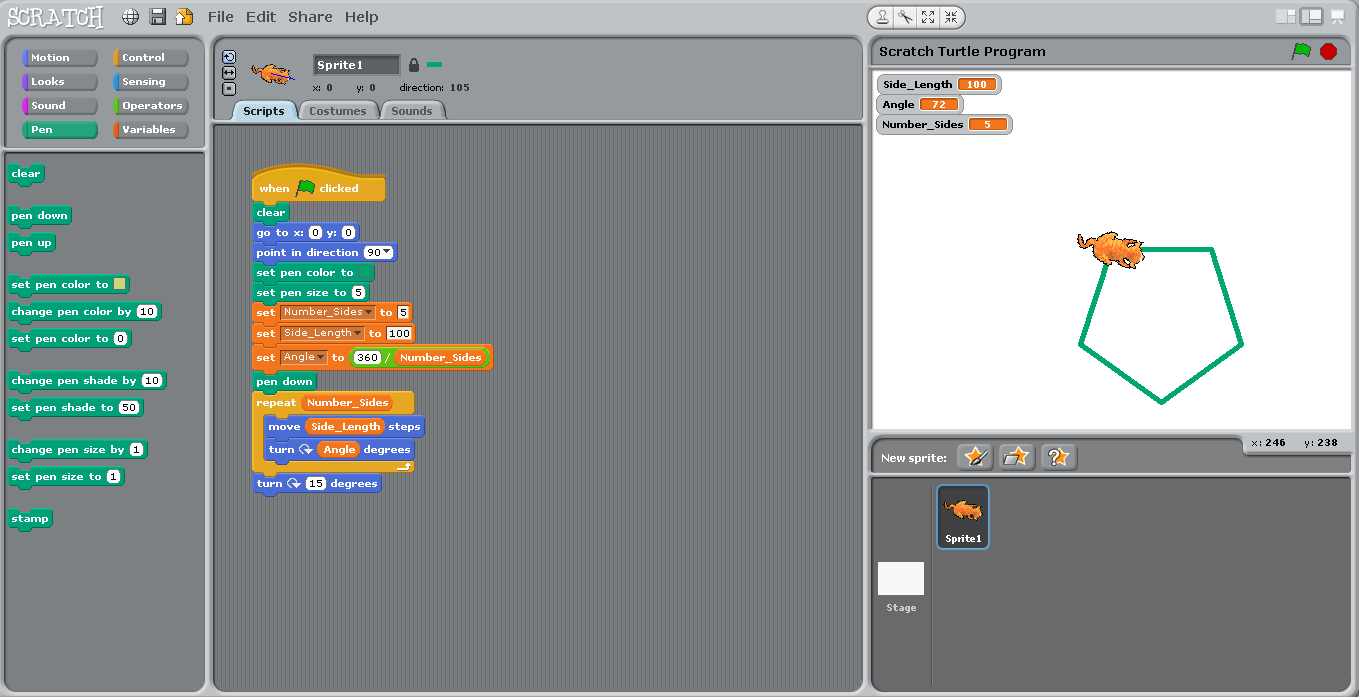


Finally, to ensure that you have a clear stage every time you run a script, add the clear block from the pen palette underneath the starting control green flag block. This block tells the program to remove the previous action before proceeding.

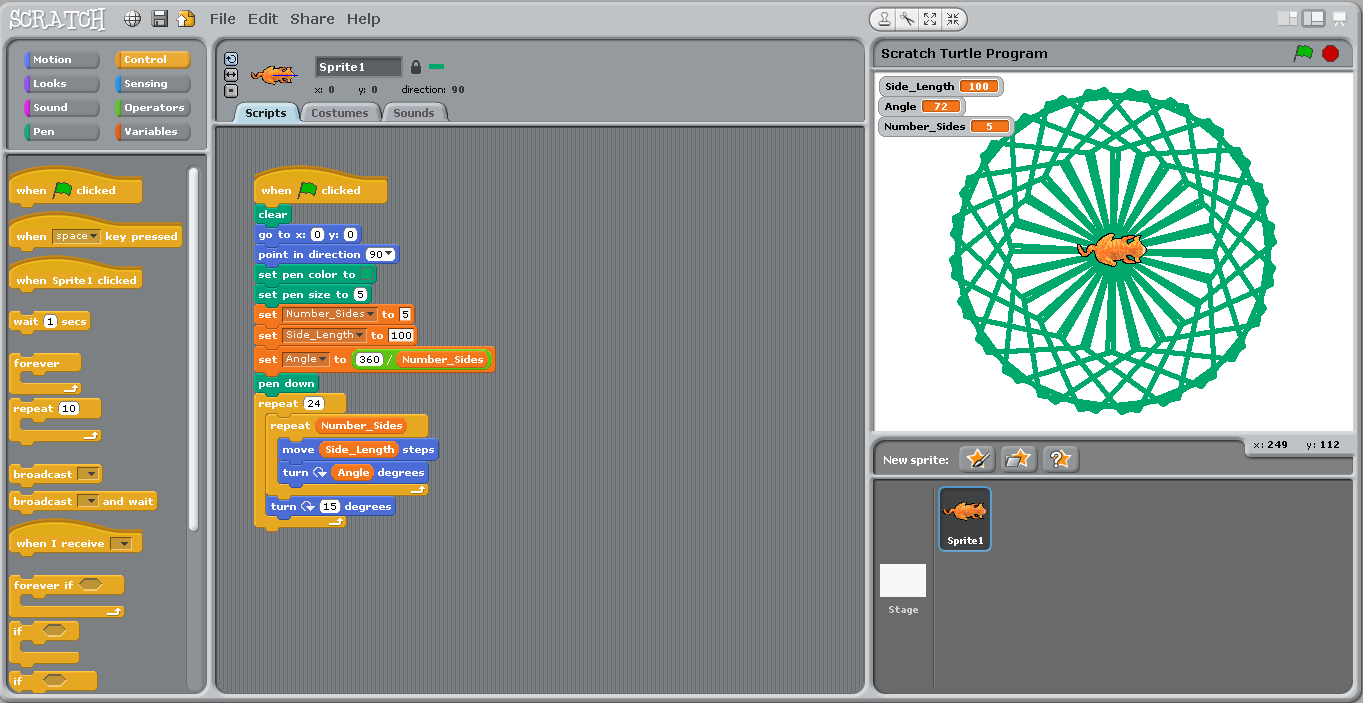
***Task 2.*** It is more logical to set variables for values that you wish to use several times in programming. This is illustrated in the screenshot below. Experiment with different values for each variable, Number of Sides, Angle and Side Length.



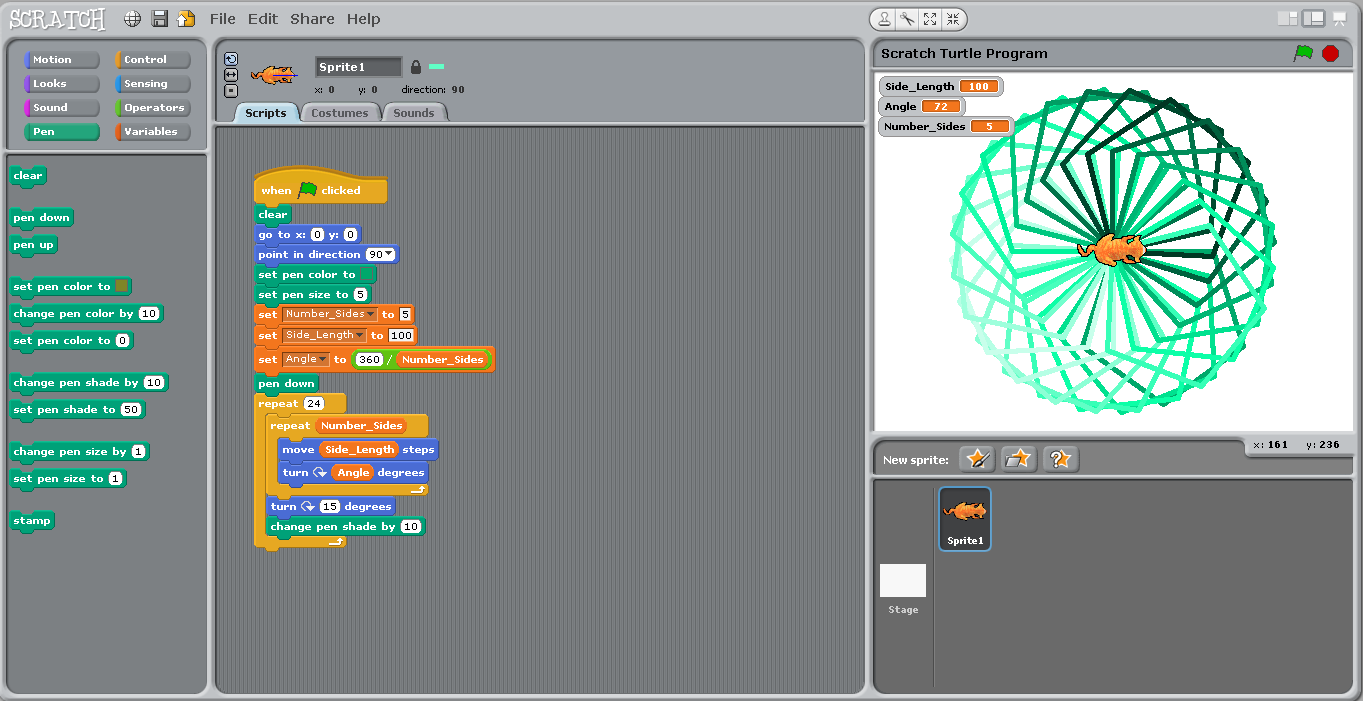
***Task 3.*** Now try changing the size and colour of the pen. An example is given by the screenshot below.



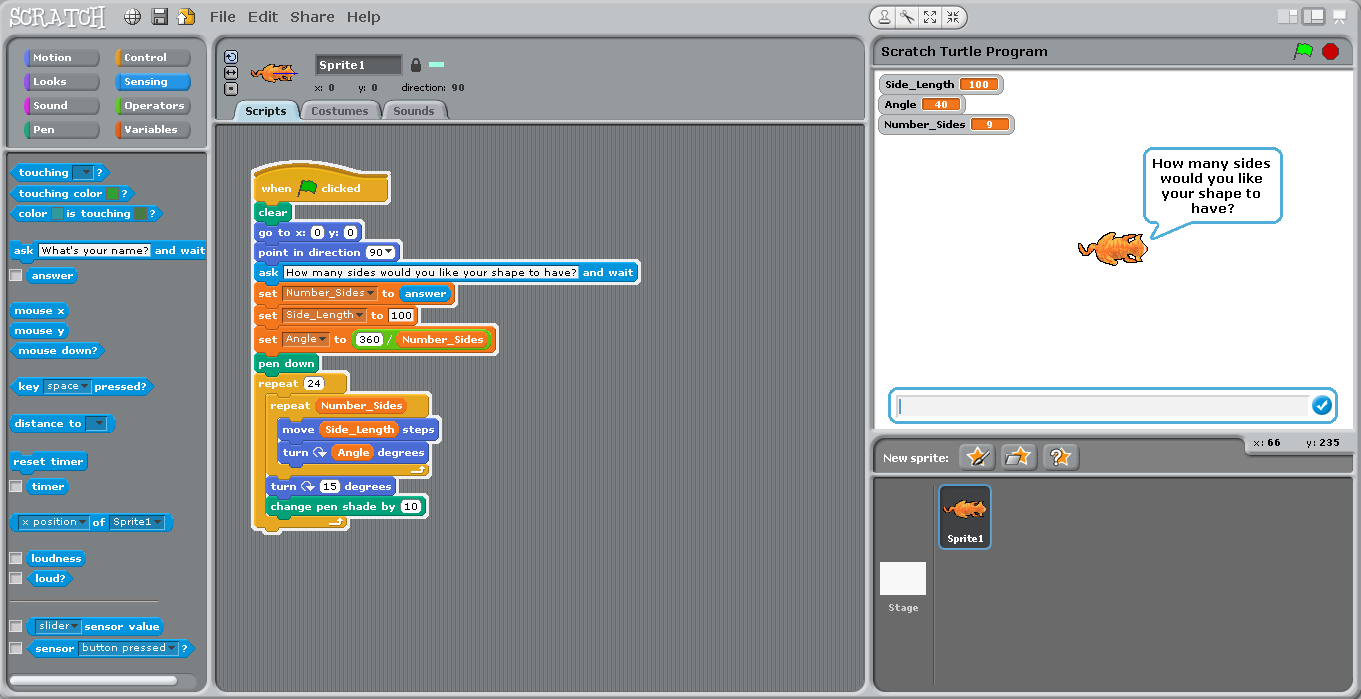
***Task 4.*** Have a go at creating a spiral pattern similar to the one illustrated.



***Task 5.*** Once you have created a spiral pattern, experiment with different pen shades.



***Task 6.*** Create a user input to determine the number of sides. (It is always more fun to allow a user to *interact* with a program that you have created).

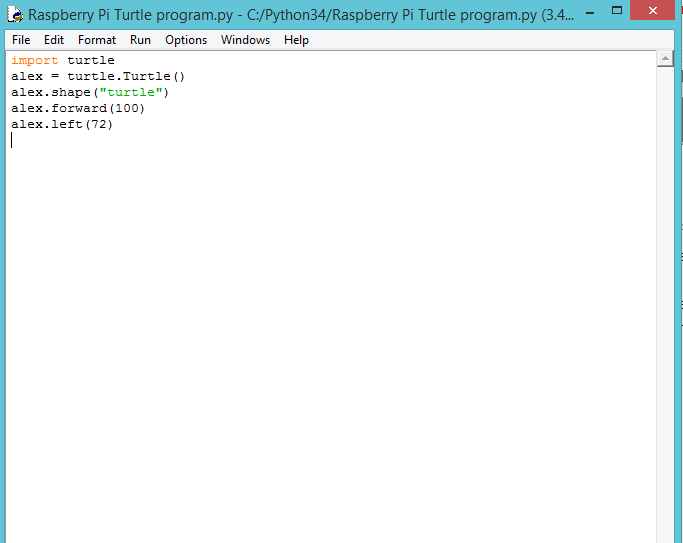


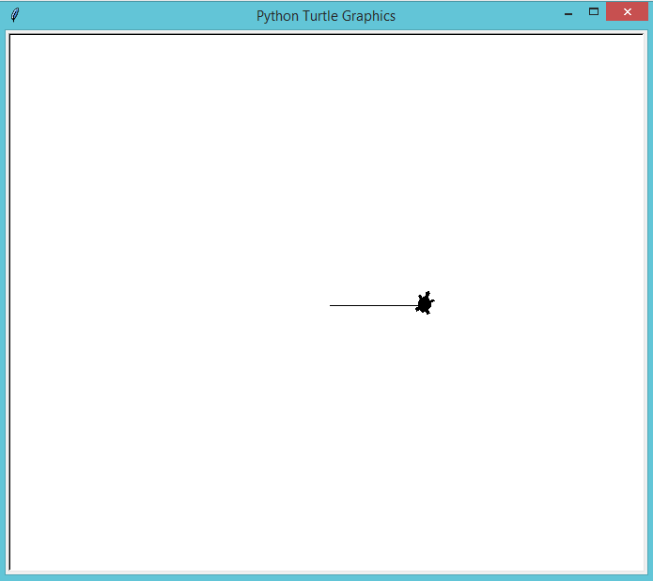
***Task 7.*** It’s now time to use Python Turtle Graphics. This is because Python includes a turtle module that you can use to create shapes and spirals in a similar way to Scratch.

Open the text editor by clicking File ⇨ New Window from the menu at the top of the Python interpreter ⇨ Save.

Note: The Python interpreter (or shell) interprets the code one line at a time. It makes more sense to use the text editor to type all your code before running it.

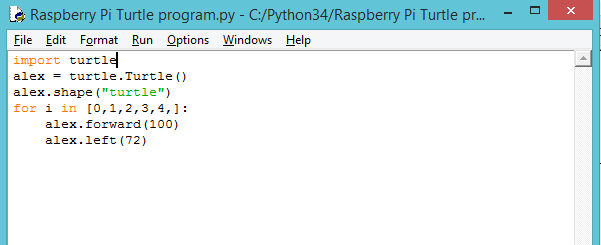
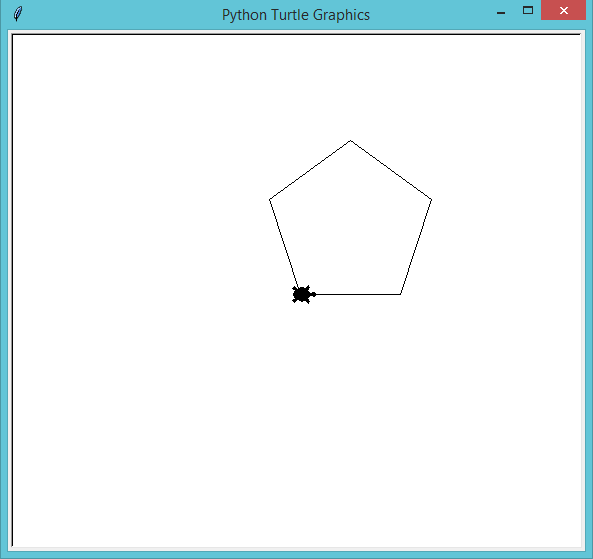
Follow the examples from the screenshots below to make your own shapes in Python similar to what you have done in Scratch.





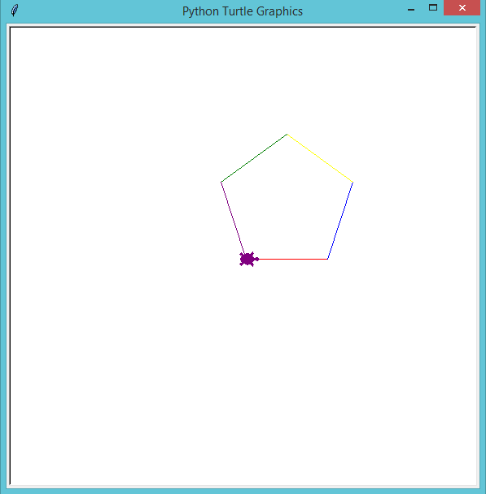
***Using “for” Loops and Lists***

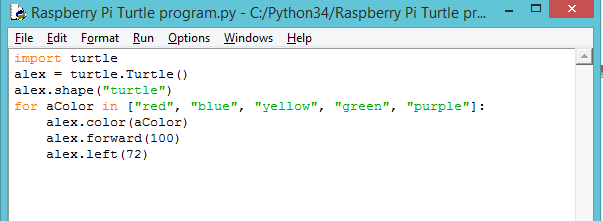
You can make the code efficient by writing the sequence once and then looping it five times. Each repeated instance of the looping is called an iteration.



The code says, “for each instance of (i) in the list, move alex forward 100 steps and then turn left 72 degrees. Numbered lists begin at 0 rather than 1.

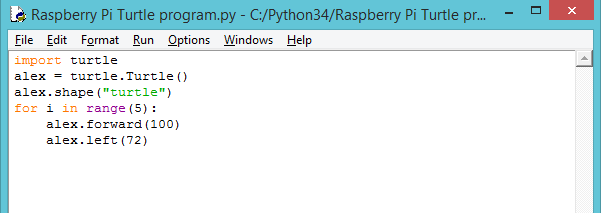
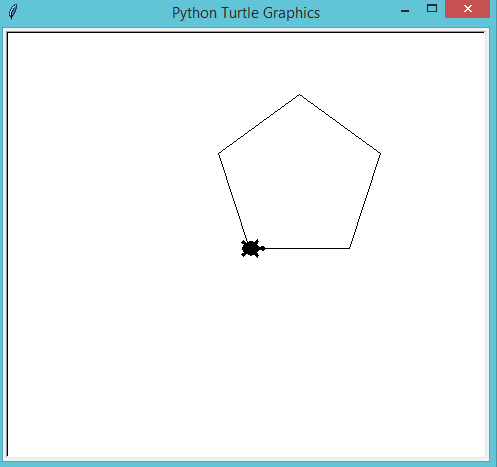
By using a loop to repeat the code, you have saved yourself a number of lines of code. By using iteration in your program you are thinking like a computer scientist.

Amend your Python pentagon code to look like the following, making sure to include the letter a before color. 



***The “range” Function***

A more efficient way of performing a loop is to use the range function instead of a list of numbers.

***Other Python Turtle Module Commands:***

Pen Up Pen Down - Just as in Scratch, the Python turtle module includes code for the pen up and pen down commands so that you can move the turtle cursor around the page without leaving a line.

alex.pendown()

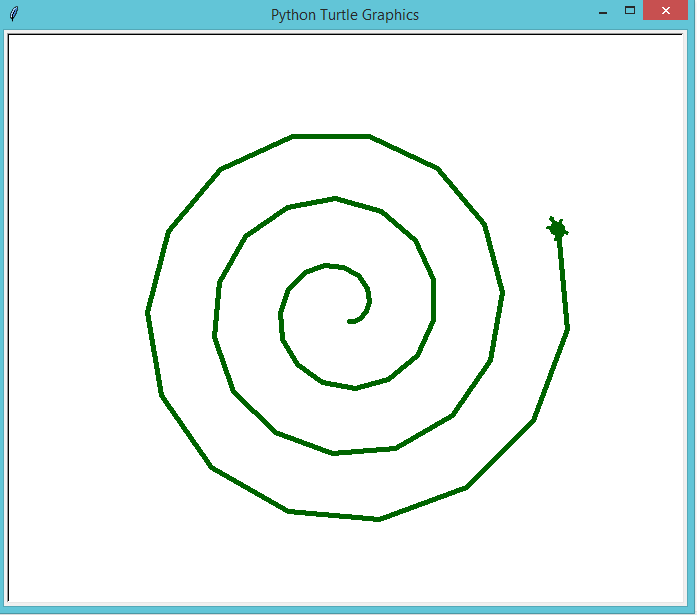
alex.forward(100)

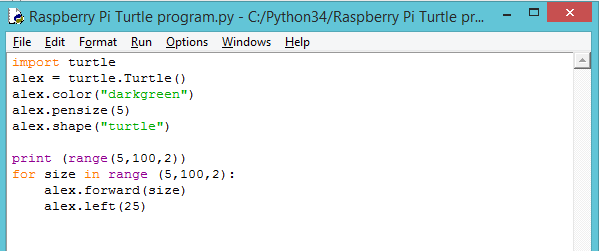
alex.penup()

Stamping – You can use .stamp to leave an imprint of the turtle cursor on the screen to form a pattern instead of, or as well as, usinh a pen line:

alex.stamp()

***The Spiral Turtle***

******



***The Spiral Turtle Stamp***

