CS1: Turtle Graphics

Week 2

From Chapter 3: <http://openbookproject.net/thinkcs/python/english3e/>

**Turtle graphics** is a term in computer graphics for a method of programming vector graphics using a relative cursor (the "turtle") upon a Cartesian plane.  We can move the turtle to create shapes and pictures.

The turtle has three attributes:

* + a location
  + an orientation
  + a pen, which has attributes such as color and width

The turtle moves with commands that are relative to its current position, such as "move forward 10 spaces" and "turn left 90 degrees".

In order to draw graphics using turtles, we must first “import” a module that contains helpful code. The statement is: **import** **turtle**

Really BIG Note!! Do not save any of your turtle programs with the name “turtle.py”. Why?

Create a new program and name it firstTurtle.py using the following instructions:

import turtle

wn= turtle.Screen() #Creates a screen for drawing

wn.bgcolor("blue") # Sets the background color of your screen

bepo=turtle.Turtle() #Creates a turtle, assigns it to the name bepo

bepo.forward(50) # Tells bepo to draw forward 50 pixels

bepo.left(90) # Tells bepo to turn left 90 degrees

bepo.forward(100) # Tells bepo to draw forward 100 pixels

wn.mainloop() # The program terminates when the user closes the window

**Your Turn:**

Make bepo draw a rectangle.

The dot notation in lines 2 and 4 means that both the **Turtle type** and the **Screen type** are defined in the module turtle that we imported.

For a turtle named bepo, it draws forward or backward by using commands bepo.forward( ) and bepo.backward( ), with the number of pixels to be moved inside the parentheses. The turtle’s direction is changed by using commands bepo.left( ) and bepo.right( ), with the number of degrees to turn left or right inside the parentheses. Turtle methods can use **positive** or **negative angles** or **distances**. Thus, the command bepo.forward(-100) will move bepo backward and the command bepo.left(-30) turns bepo to the right.

Question: What is the difference between bepo.left(90) and bepo.right(-90)?

A review of the previous program includes commands we use with Turtle graphics:

1. To set up the Screen for drawing:

wn = turtle.Screen() #Creates a screen for drawing

wn.bgcolor(“aquamarine1") # Sets the background color of your screen

…

wn.mainloop() #Program terminates when the user closes the window

1. To draw:

turtleOne = turtle.Turtle() #Creates a turtle, assigns it to the name turtleOne

turtleOne.shape(“turtle”) # Specifies a shape for the cursor

turtleOne.forward(50) # Tells turtleOne to move forward by 50 units

turtleOne.left(-120) # Tells turtleOne to turn left -90 degrees

turtleOne.backward(30) #The instruction turtleOne.back(30) also works

Possible colors are given in <http://www.tcl.tk/man/tcl8.4/TkCmd/colors.htm> , showing RGB color values (Red, Green, and Blue).

For turtle methods, see: <https://docs.python.org/3/library/turtle.html> . If you forget, just type “python methods” into Google and change the 2 that you find to a 3…

Changing pen values

A pen can be picked up and put down, allowing us to move from one location to another without continuously drawing. The commands are **penup()** and **pendown()**. We can set a pen size by using the command **pensize()**, and we can set a pen color by using the command **pencolor()**.

We can set speeds for drawing by using **speed()** with values between 1 (slowest) and 10 (fastest).

**Your Turn:**

Type the following commands into the Python Editor:

import turtle

wn = turtle.Screen()

wn.bgcolor("lightgreen") # Set the window background color

wn.title("Hello, Bepo!") # Set the window title

bepo = turtle.Turtle()

bepo.color("blue") # Tell bepo to change the color

bepo.pensize(3) # Tell bepo to set the pen width

bepo.forward(50)

bepo.left(120)

bepo.penup()

bepo.forward(50)

bepo.pendown()

bepo.backward(200)

wn.mainloop()

Modify your program so that before it creates the window, it prompts the user to enter the background color, the color of the turtle, the size of the pen, and the color of the pen.

To repeat an action 5 times, we would surround the action with the following statement:

for i in range(5) :

Notice the colon at the end of the ***for*** statement! …See sections 3.3 – 3.5 in your online text.

Make the following changes to your program:

import turtle

scrn1 = turtle.Screen()

scrn1.bgcolor("lightgreen")

scrn1.title(“Hello, yourName!”)

bepo = turtle.Turtle()

bepo.shape("turtle")

bepo.color("blue")

bepo.penup()

size = 20

for i in range(30):

bepo.stamp() # Leave an impression on the canvas

size = size + 3 # Increase the size on every iteration

bepo.forward(size) # Move bepo along

bepo.right(24) # ... and turn bepo

wn.mainloop()

What went wrong? Fix it…

How many times was the body of the loop executed?

How many turtle images did you see on the screen?

How many different turtle shapes can we use?

arrow, blank, circle, classic, square, triangle, turtle

What does the variable **size** affect?

**Your Turn:**

1. Make the first turtle stamp red and the rest white.
2. Now make the last turtle stamp blue.
3. Create 2 turtles. Use a ***for*** loop to make one turtle draw an equilateral triangle. Make the other turtle draw a rectangle. Both of these drawings should be on the same screen.

**For extra practice**: Work problems 6, 11, and 12 at the end of Chapter 3. Add a preface (documentation) to your code.