Lab Exercise #6

Assignment Overview

This lab exercise provides practice with file processing and with simple graphics using Python libraries.

You will work with a partner on this exercise during your lab session. Two people should work at one computer. Occasionally switch the person who is typing. Talk to each other about what you are doing and why so that both of you understand each step.

Part A: Reading Files

1. Predict what the Python program below will display.

```
file_name = "lab06.test.txt"
input_file = open( file_name, "r" )

for line in input_file:
    print( line )

input_file.close()
```

2. Download the program ("lab06.parta.py") and the input file ("lab06.test.txt"), then execute the program to check your prediction. Compare the output on the screen to the contents of the input file.

3. The program printed a blank line after each line of the input file. Revise the program to display each line of the input file without printing the extra blank lines. Consider using the "rstrip" string method to accomplish this task (http://docs.python.org/3.4/library/stdtypes.html#str.rstrip).

4. Revise the program to prompt the user to enter the name of the input file.

Part B: Writing Files

Develop a Python program (in the file named "lab06.partb.py") which prompts the user for the name of an output file, and then copies all subsequent lines into the output file until the user enters a period as the first character on a line.

a) The program will use mode "w" to open the file, and will close the file before halting.

b) Consider calling function "input" without an argument to read the lines from keyboard (after using function "input" to prompt the user to enter the name of the output file).

c) The output file should contain an exact copy of what the user entered, excluding the first line (with the name of the output file) and the last line (with the period).

★ **Demonstrate your completed program to your TA. On-line students should submit the completed program (named "lab06.partb.py") for grading via the CSE handin system.**

Part C: Turtle Graphics

Part of the *Logo* programming language, Turtle graphics (http://en.wikipedia.org/wiki/Turtle_graphics) is one of the oldest graphics programs. It is a 2D graphics package that uses a Cartesian coordinate system and a "turtle" which has a pen attached to its body. The turtle can move around the plane, drawing as it goes.

The Python "turtle" module is an implementation of the original package (see Appendix B of the text or the reference pages at http://docs.python.org/3.4/library/turtle.html).

To use turtle graphics in Python, you must first import the turtle module. You can then use the help function in IDLE to find out what methods this module includes and what they do. Type **import turtle** in the IDLE command window, and then type **help(turtle)** and scroll through the information. For more details, Google "Python 3 turtle."

Imagine the "turtle" sitting on a piece of paper. It has a pen, and it can either place the pen down on the paper or not. If the pen is down and the turtle moves, it leaves a line on the paper.

The following code segment draws a square, filled with red, starting at the present turtle location. The `time.sleep(5)` command waits 5 seconds before closing the turtle window.

```
import turtle
import time

turtle.down()
turtle.color(1.0, 0.0, 0.0)     # all red, no green, no blue
turtle.begin_fill()
for side in range(4):
    turtle.forward(100)
    turtle.left(90)
turtle.end_fill()
time.sleep(5)                   # wait 5 seconds
turtle.bye()                    # close the turtle window
```

The first thing you should do is try out some of the turtle commands by just typing them in the IDLE window. You can get a much better feel for how the whole thing works by just trying it.

After you have imported the turtle module, here are some commands to try:

`turtle.up(),turtle.down():` Set the pen state to be up (not drawing) or down (drawing)

`turtle.right(degrees), turtle.left(degrees):` Turn the direction that the turtle is facing. The amount of turn is indicated in degrees.

`turtle.forward(distance), turtle.backward(distance):` Move the turtle forward or backward the amount of distance indicated. Draws a line if the pen is down.

`turtle.goto(x,y)`: The goto method moves the turtle to a specified point, drawing a line if the pen is down. Note: The turtle always starts at the point (0,0).

`turtle.color(r,g,b)`, `turtle.color(s)`: The color method sets the color that the pen will hold for all drawing until the color is changed . There are two versions of this method. The first version takes three arguments, each a floating-point number between 0.0 and 1.0. The first argument is the amount of red, the second is the amount of green and the third is the amount of blue. The second version takes a Tk symbolic name (as a `string`), which denotes a color. Symbolic names are listed at http://www.tcl.tk/man/tcl8.4/TkCmd/colors.htm. For example, `"red"` denotes red; `"purple"` denotes purple; `"green"` denotes green.

`turtle.begin_fill()`, `turtle.end_fill()`: Use the command turtle.begin_fill() before you start drawing a figure. Draw the figure, then execute the command turtle.end_fill(). The figure drawn between the two fill commands will be filled with the present color setting.

`turtle.circle(radius)`: Draws a circle of radius `radius`. Drawing begins at the present turtle location and draws the circle tangent to the direction the turtle is heading in at the start of the method call; it does *not* start at the center of the circle.

`turtle.seth(degrees)`: Sets the direction the turtle is facing, which is called its heading ("seth" is short for "setheading"). The argument gives the angle at which to set it.

`turtle.bye()`: Close the turtle drawing window.

`time.sleep(seconds)`: Before you execute the `bye()` command above, you should `import time` and then execute `time.sleep(seconds)`. This will leave the window visible for the indicated number of seconds before the turtle window closes.

A sample Python program ("lab06.partc.py") is provided in the directory for this lab exercise.

Part D: Programming with Turtle Graphics

Develop a Python program (in the file named "lab06.partd.py") which draws a rectangle.

a) The program will prompt the user to enter the length of the two vertical sides.

b) The length of the two horizontal sides will be twice as long as the length of the two vertical sides.

c) The rectangle will be filled with the color green.

★ **Demonstrate your completed program to your TA. On-line students should submit the completed program (named "lab06.partd.py") for grading via the CSE handin system.**