Python : Nested loops using turtle graphics.

Introduction

This is a shortened and python centric version of my original nestedlopps.tex or nestedloops.pdf document

In programming, it is sometimes useful to be able to carry out the same task several times over. The two listings below compare how to perform exactly the same task. Listing 2 has some advantages

Listing 1

```
print("hello")
print("hello")
print("hello")
print("hello")
```

Listing 2

```
for x in range(0,4):
        print("hello")
```

- If you want to print hello, world instead then you change 1 line
- If you want to print more times then again its 1 line of code or in fact 1 part of that line.
- Shorter, therefore less typing
- Loops can be nested, which gives other advantages

In python the range refers to a start and end point, 0,4 repeats 4 times.

Listing 3

```
for x in range(0,3):
        print("hello, world")
```

Extention, try different numbers as both a start and end point in the range and see what happens.

Lets look into producing something using turtle graphics.

We can now look in to producing something that uses a nested loop, which essentially means loop within a loop.

Python : Nested loops using turtle graphics 2.

In Idle, enter, save and run the following code

Listing 4

```
import turtle
import time

for n in range(0, 4):
        turtle.forward(50)
        turtle.left(90)
time.sleep(5)
```

This simply produces a square. However it does so using 2 lines of turtle code, 1 to draw a line the other to turn 90 degrees, this happens 4 times. To produce the square.

Note: the indentation:

Now enter the following code. (or modify the code you entered above, if you do, save as a different filename.)

Listing 5

```
import turtle
import time

for x in range(0,72):
        turtle.left(5)
        for n in range(0, 4):
                turtle.forward(50)
                turtle.left(90)
time.sleep(5)
```

Again : note the indentation here:

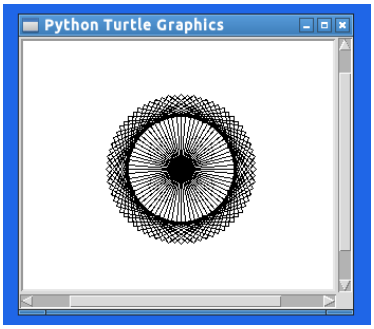

Figure 1: