Python : Using turtle graphics with loops, - with enhanced features.

Introduction

This activity will develop a program with a number of features, which will be explained as you develop, The main routine will be familar to anyone who has completed worksheets.pdf. This extends that program. Lines with # at the start are comments.

```
import turtle
import time
```

To begin with we import the turtle module and the time module.

```
#set file name
fname= raw_input("Filename (must include .eps ")
#fname="dial.eps"
```

I have added a feature so that the user can specify a file name so that the resulting turtle picture can be saved. The file name is stored as a variable fname.

```
sides = raw_input("number of sides to shape ")
sides = int(sides)
shapeangle = 360 / sides
```

Now we ask for the number of sides, The input produces a string so the next line converts this to an integer. Then the shape angle is calculated by taking 360 (degrees in a circle) and deviding by the number of sides. This is stored as a variable shapeangle.

```
for x in range(0,72):
        turtle.left(5)
        for n in range(0,sides):
                turtle.forward(150)
                turtle.left(shapeangle)

time.sleep(1)
```

This part of the program draws the shape, based on the number of sides and the required shapeangle. The time sleep simply adds a delay once finished.

```
ts = turtle.getscreen()
ts.getcanvas().postscript(file=fname)
```

+

We now need to grab the canvas as save this to the file, which we specified earlier.

```
print "Saved image to: ", fname
print "All done. Click image to exit."
turtle.exitonclick()
```

Finally, we output to the console to tell the user that the file has been saved, the file name, and then instruct the user to click on the image. Upon doing so the canvas is removed from the screen. The directory should have a .eps file with the resulting picture.

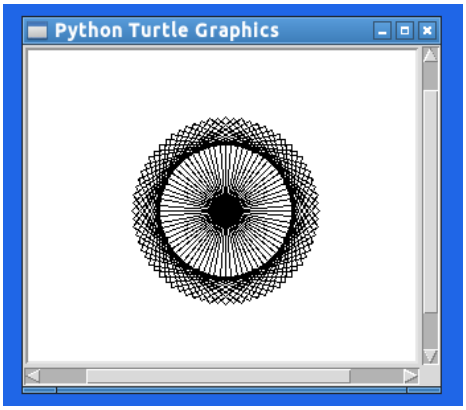

Figure 1: