Turtle Graphics

Today, as well as using Python, we are going to use a **turtle!** Specifically, the “turtle graphics” module for Python, which is used for drawing. Also, we will use **Idle,** which is a program that allows us to use Python *interactively* (the commands we type in do things immediately, instead of us having to run the tell Python to run the whole file). Try it by starting Idle (with the desktop icon) and type in:

```
>>> print “Hello!”
>>> 3+2
>>> a=5
>>> b=9
>>> a+b
>>> for i in range(10):
        print i, i*i
```

Note: You have to press <ENTER> twice after the last one. Can you think why?

This is useful if we want to try things out in Python without having to write a program in a file. Idle also has a text editor, similar to nano, if we do want to write a program.

1 *Starting Turtle*

First we need to tell Python that we want to use the Turtle functions by importing the module:

```
>>> import turtle
```

This may bring up another window with nothing in it – this is the window that we are going to draw in. If a window doesn't appear we can get it with:

```
>>> turtle.clearscreen()
```

This can also be used at any time to wipe out everything we have drawn and start again.

Now try typing this in ***but do not press <ENTER>*** (yet):

```
>>> turtle.
```

Instead, press the <TAB> key. Do you see that a drop-down menu has come up? Either type in or select from the list the word 'forward'. ***Still do not press <ENTER>.*** Instead, press '(' so that it reads:

```
>>> turtle.forward(
```

Do you see the help box? This is telling you what needs to go in the bracket, in this case just a single value that will be the distance. Try:

```
>>> turtle.forward(100)
```

And now you can press <ENTER>.

So, in Turtle Graphics, we draw by moving the Turtle (the pointer) to different places and a line appears along the path it takes.

2 Shapes

OK, so we can draw a line. How about something more interesting, like ... a **square!** OK, so it is not *a lot* more interesting yet. Here is how we can describe a square:

- Move forward 100
- Turn left 90 degrees
- Move forward 100
- Turn left 90 degrees
- Move forward 100
- Turn left 90 degrees
- Move forward 100
- Turn left 90 degrees

The last turn isn't really needed but it points the turtle back in the direction it started.

In Python, this is:

```
>>> turtle.clearscreen()
>>> turtle.forward(100)
>>> turtle.left(90)
>>> turtle.forward(100)
>>> turtle.left(90)
>>> turtle.forward(100)
>>> turtle.left(90)
>>> turtle.forward(100)
>>> turtle.left(90)
```

That's a lot of typing, and a lot of it is repeating the same things. What we really want to do is tell it to repeat the drawing of the line and the turning four times:

```
>>> turtle.clearscreen()
>>> for i in range(4):
        turtle.forward(100)
        turtle.left(90)
```

Remember to press <ENTER> twice at the end.

Challenge: A square is all very well, but can you draw a triangle? Or a pentagon? Or a dodecagon? (Hint – you need to change two of the numbers)

Challenge: Can you change it to make many-pointed stars instead? (Hint – use angles more than 90 degrees)

Challenge: (Harder) Can you make this into a function, so that you can draw a shape by just typing a single command? Hint – remember how to make functions with the 'def' command. It might start like:

```
>>> def draw_shape(sides, length, angle):
```

3 Jumping and Moving

It is going to be very difficult to draw more complicated things if a line appears every time the turtle moves. Fortunately, we can turn the pen off to move to a different place:

```
>>> turtle.clearscreen()
>>> turtle.forward(100)
>>> turtle.right(60)
>>> turtle.penup()
>>> turtle.forward(200)
>>> turtle.pendown()
>>> turtle.right(120)
>>> turtle.forward(100)
```

There are other ways to draw and move instead of just forward. There are 'turtle.backwards', 'turtle.circle' and 'turtle.goto' (can you guess what they do?) :

```
>>> turtle.clearscreen()
>>> turtle.penup()
>>> turtle.goto(-150, 150)
>>> turtle.pendown()
>>> turtle.circle(50)
>>> turtle.penup()
>>> turtle.goto(150, 150)
>>> turtle.pendown()
>>> turtle.circle(50)
>>> turtle.penup()
>>> turtle.goto(0, 100)
>>> turtle.pendown()
>>> turtle.goto(0, 0)
>>> turtle.goto(20, 0)
>>> turtle.penup()
>>> turtle.goto(-250, 0)
>>> turtle.pendown()
>>> turtle.goto(-160, -100)
>>> turtle.goto(160, -100)
>>> turtle.goto(250, 0)
```

4 Challenges

Can you make a better picture than mine? (Not hard!)

Hint – you might find it easier to put it in a file, like we did with other python programs, and then run it. In Idle you can do this with the 'New Window' option from the 'File' menu, and then when you are ready, use 'Run Module' from the 'Run' menu. Remember that the program has to start with this line:

```
import turtle
```

Try some other commands.

Remember at the start when you typed:

```
>>> turtle.
```

and then pressed <TAB>? A whole list of commands appeared! Try some of these to see what they do. See if you can change the pen colour or fill in shapes.

Use Python on your own computer!

You don't have to have to have a Raspberry Pi to use Python (but it helps). If you have a Windows computer you can install it with the instructions here:

http://docs.python.org/2/using/windows.html

Or if you have a Mac, try here:

http://docs.python.org/2/using/mac.html

Hint – There are two versions of Python at the moment, Python 2 and Python 3, and unfortunately, there are some important differences between the two. You will find it easier if you stick to using Python 2 for the moment.

5 *Special message for the Year 6 leavers ...*

(... and those in year 5 are allowed to read this as well!)

I hope you have enjoyed Code Club this year and wish you luck when you move to your new school.

If you want to carry on learning to code then visit:

www.codeclub.org.uk/further-resources

which has links to all kinds of online coding tools and some more cool groups of coders that you can join. If you go to the Manchester Coder Dojo I might see you there!