Turtle Geometry the Python Way

@SteveBattle
“The tradition of calling our display creatures ‘turtles’ started with Grey Walter, a neurophysiologist who experimented in Britain… with tiny robot creatures he called ‘tortoises’. These inspired the first turtles designed at MIT.” (1980)
Turtles and Tortoises

“Not even Dr. Grey Walter, their creator, can say what these mechanical tortoises are going to do next.”

Bristol based Inventor of the first autonomous robots in 1948.
BBC Buggy (1983)

The Buggy was tethered to the BBC Micro and used the ‘Logo’ language.
The Python Way

Start IDLE and start typing:

```python
from turtle import *
forward(100)
right(90)
forward(100)
```
angles

A 360° turn returns the turtle to 0°

The turtle starts off facing this way

A right angle going clockwise
Create a new file (Python module)

Save the file (but not as 'turtle')

Run
# turtle functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>forward(length)</code></td>
<td>move the turtle forward by <code>length</code> units</td>
</tr>
<tr>
<td><code>backward(length)</code></td>
<td>move the turtle backward by <code>length</code> units</td>
</tr>
<tr>
<td><code>left(angle)</code></td>
<td>turn the turtle left by <code>angle</code> degrees</td>
</tr>
<tr>
<td><code>right(angle)</code></td>
<td>turn the turtle right by <code>angle</code> degrees</td>
</tr>
<tr>
<td><code>penup()</code></td>
<td>lift the pen up</td>
</tr>
<tr>
<td><code>pendown()</code></td>
<td>put the pen down</td>
</tr>
<tr>
<td><code>done()</code></td>
<td>end the program</td>
</tr>
<tr>
<td><code>speed(s)</code></td>
<td>set the speed of the turtle to <code>s</code></td>
</tr>
<tr>
<td><code>shape(name)</code></td>
<td>set the shape of the turtle to <code>name</code></td>
</tr>
<tr>
<td><code>goto(x,y)</code></td>
<td>move the turtle to position <code>(x, y)</code></td>
</tr>
</tbody>
</table>

- `length` can be a positive or negative integer.
- `angle` must be a positive integer.
- `s` can be `'slow'`, `'fast'`, or `'fastest'`.
- `name` can be `'turtle'` or `'classic'`.
- `x` and `y` are integers representing the coordinates.
for loops

The loop repeats a fixed number of times (4).

Everything inside the loop is indented. Use TAB not SPACE.

slow down the output

```
from turtle import *
speed('slow')
pendown()
for i in range(4):
    forward(100)
    right(90)
done()
```
defining functions

Function definition

Note the double indentation

Call the function

'turtle' shape
function parameters

Add parameter ‘length’

Call ‘square’ with argument 100

def square(length):
    for i in range(4):
        forward(length)
        right(90)

square(100)
done()
polygons

```python
from turtle import *
def polygon(length, sides):
    for i in range(sides):
        forward(length)
        angle = 360/sides
        print(angle)
        right(angle)
speed('slow')
pendown()
polygon(100,3)
done()
```

The output from 'print' appears in the IDLE console

The angles of a polygon sum to 360°

Comma separated parameters

calculate angle and assign to a variable
The (interior) angles of a triangle sum to only 180˚.

The turtle turns through the exterior angles, which sum to 360˚.
What happens if we turn turtle through a multiple of $360^\circ$?

Try also heptagons:
- `star(100,7,2)`
- `star(100,7,4)`

Double the angle of a normal pentagon.
while loops

```python
from turtle import *

def spiral(length, sides, multiple, decrement):
    while length > 0:
        forward(length)
        right(multiple * 360/sides)
        length = length - decrement

speed('slow')
penup()
goto(-200, 200)
pendown()
spiral(200, 4, 1, 10)
done()
```

Exit the ‘while’ loop when the condition is false.

Try a non-integer multiple:
```
spiral(400, 3, 1.01, 10)
```
Recursion

Recursive functions call themselves.

The angle is close to 0 on the ‘straights’.

We can also get spiral motion by increasing the angle.

```python
from turtle import *

def inspiral(length, angle, increment):
    forward(length)
    right(angle)
    inspiral(length, (angle + increment) % 360, increment)

speed('slow')
pendown()
inspiral(50, 2, 20)
done()
```
The Snowflake Curve

Every side has smaller sides and so on, ad infinitum.
from turtle import *

def snowflake(length, level):
    for i in range(3):
        side(length, level)
        right(120)

def side(length, level):
    if level == 0:
        forward(length)
    else:
        side(length/3, level-1)
        left(60)
        side(length/3, level-1)
        right(120)
        side(length/3, level-1)
        left(60)
        side(length/3, level-1)

speed('fastest')
penup()
goto(-100, 100)
pendown()
snowflake(200, 4)
done()