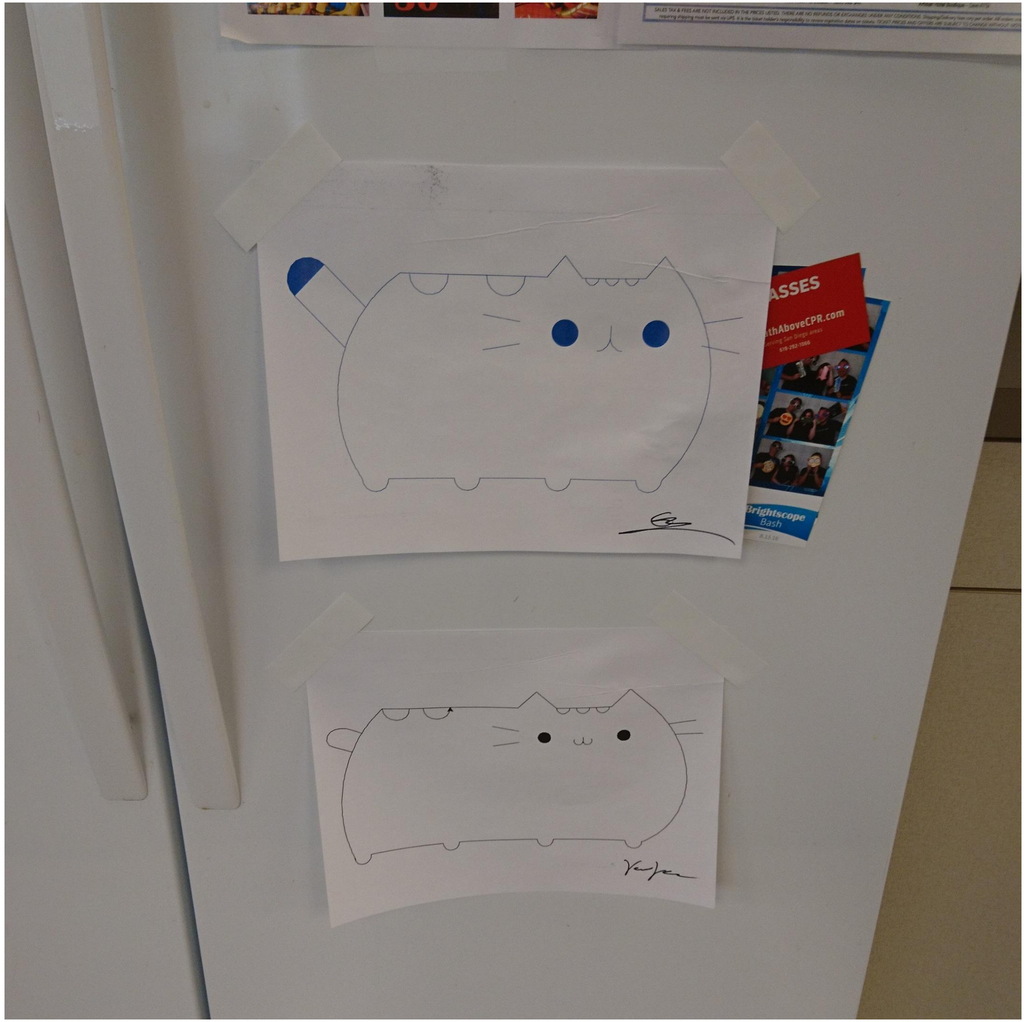
Recursion, Fractals, and the Python Turtle Module

Hayley Denbraver
@hayleydenb

europython
Edinburgh 23-29 July
2018

Hello!

Photo Source


ASSES
thAboveCPR.com
Brightscope
Bash

Pusheen.com

Recursion
Fractals
Python Turtle Module

Recursion

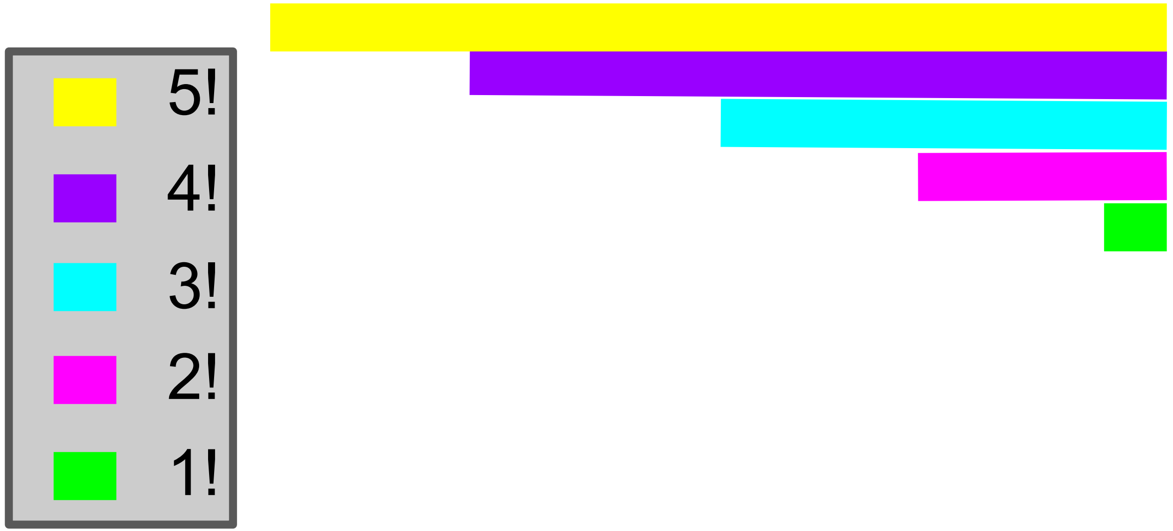
Recursion

Photo Credit

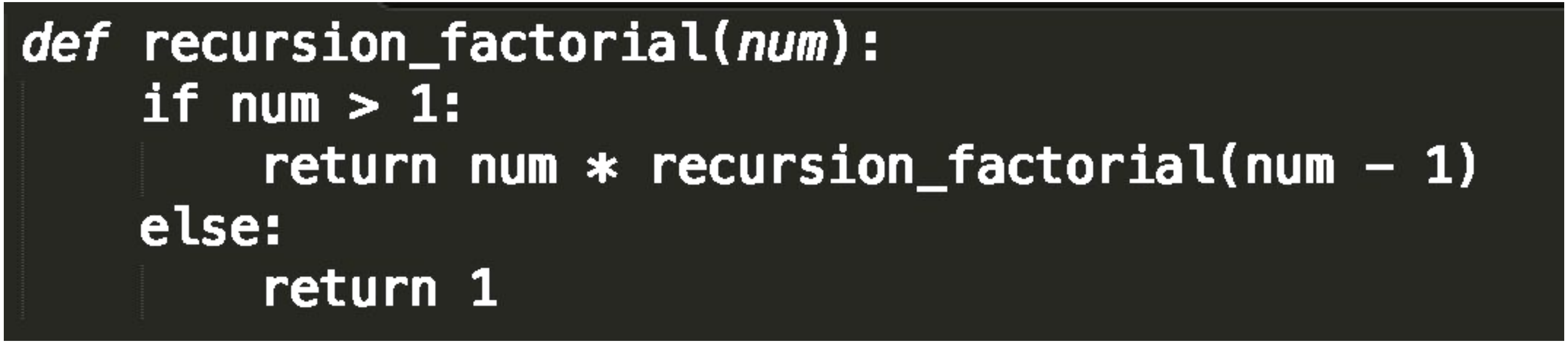
Recursion involves breaking a problem down into smaller and smaller subproblems until you get to a small enough problem that it can be solved trivially.

Factorials!

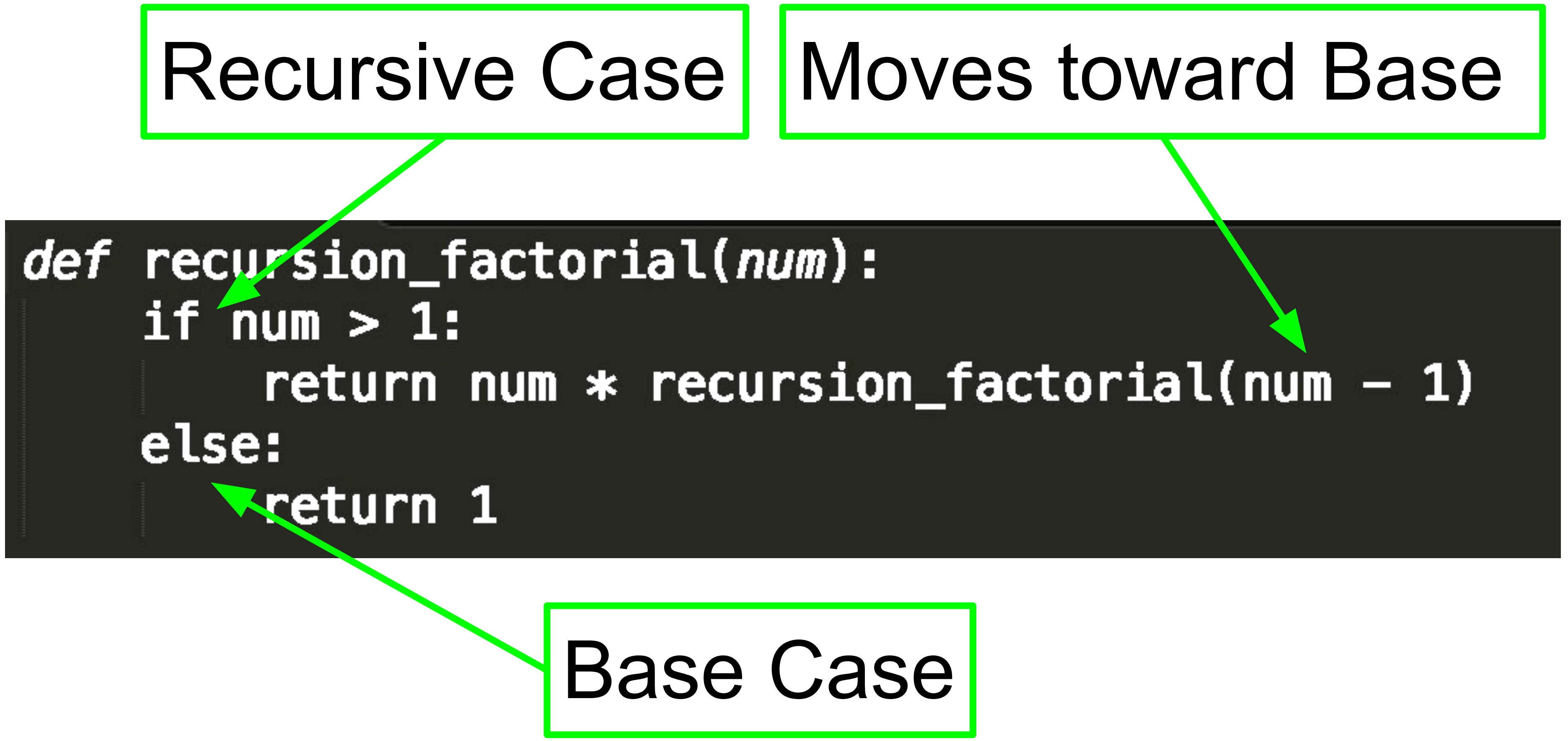
$$5! = 5 \times 4 \times 3 \times 2 \times 1$$



Recursive Version

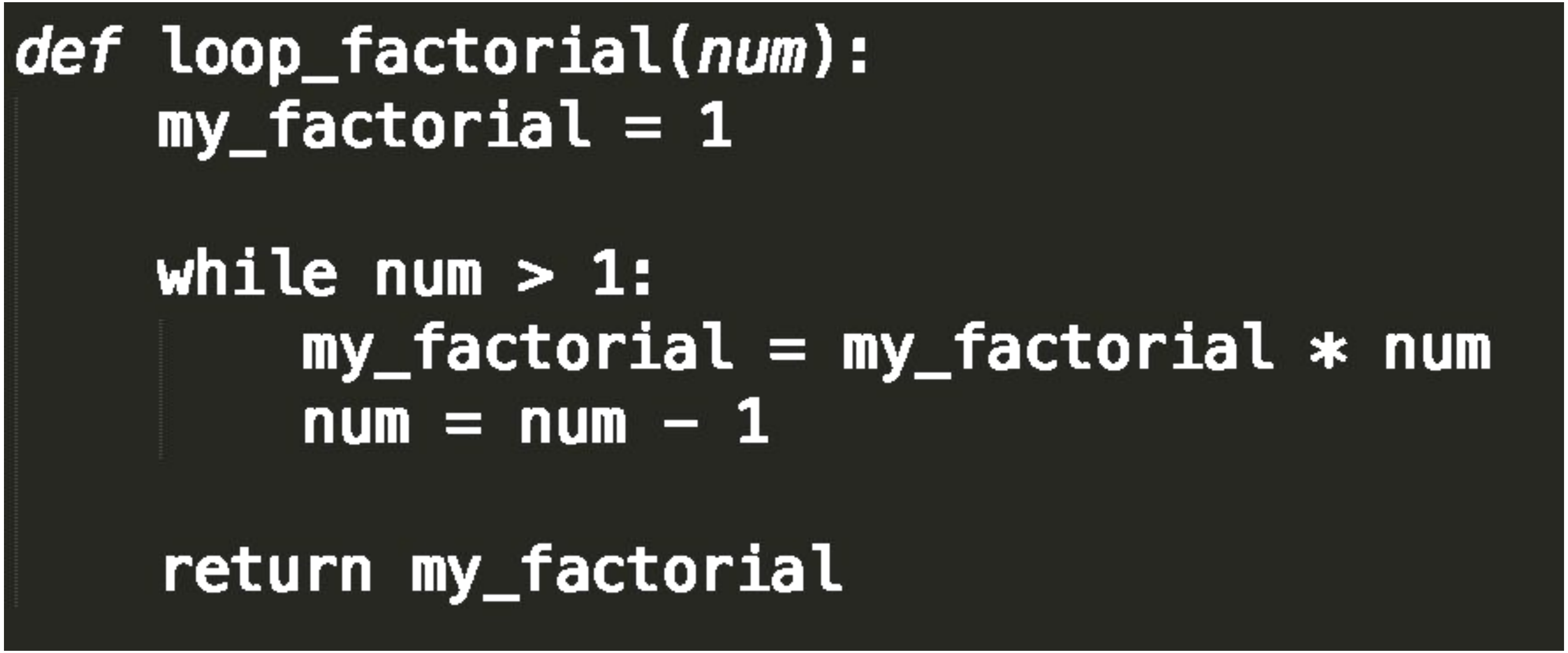


```python
def recursion_factorial(num):
    if num > 1:
        return num * recursion_factorial(num - 1)
    else:
        return 1
```



```
def recursion_factorial(num):
    if num > 1:
        return num * recursion_factorial(num - 1)
    else:
        return 1
```

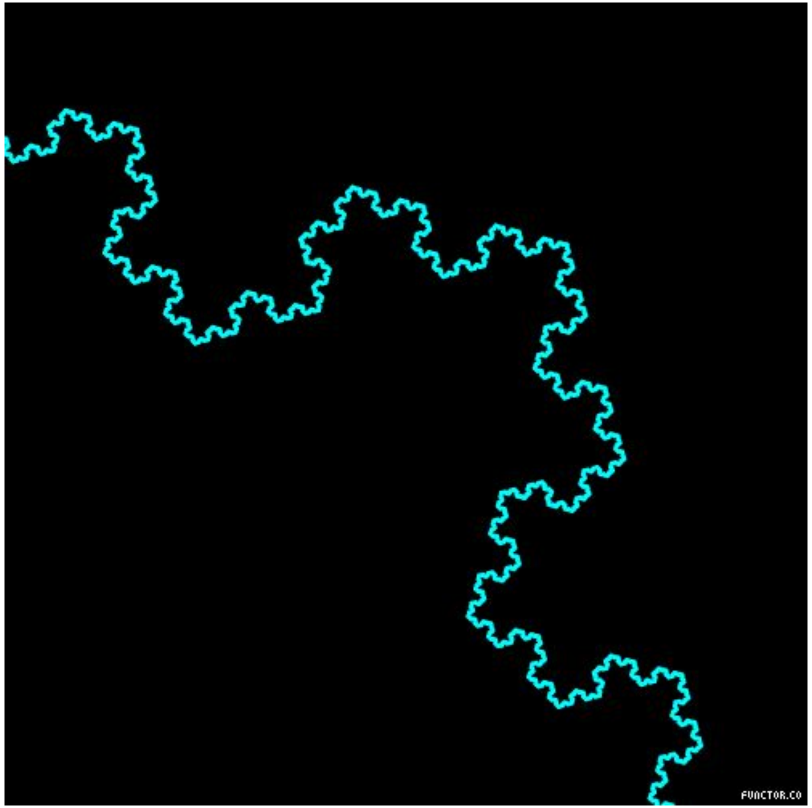
Loop Version



```
def loop_factorial(num):
    my_factorial = 1

    while num > 1:
        my_factorial = my_factorial * num
        num = num - 1

    return my_factorial
```

Fractals

Source

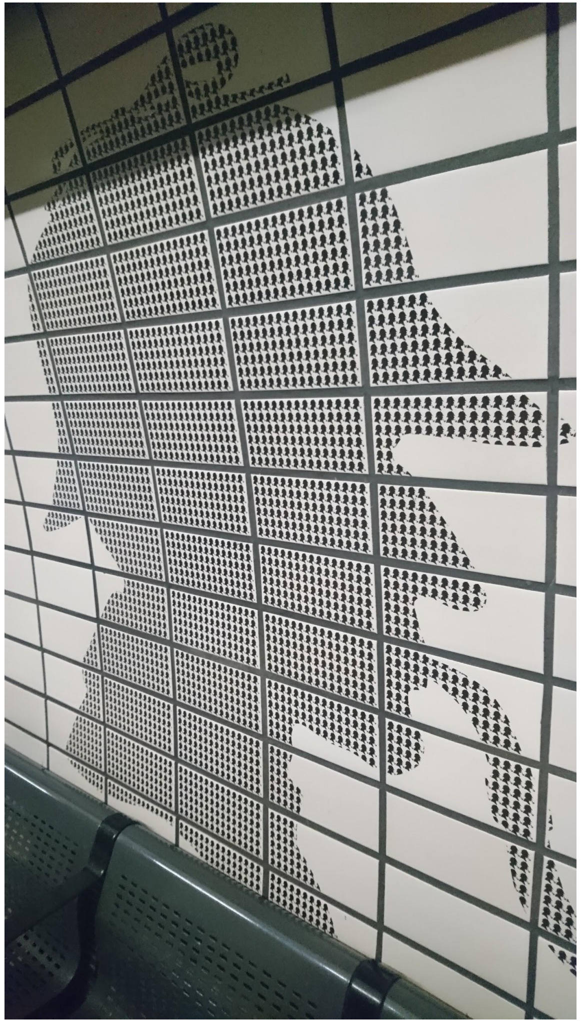


Source

Source

Source

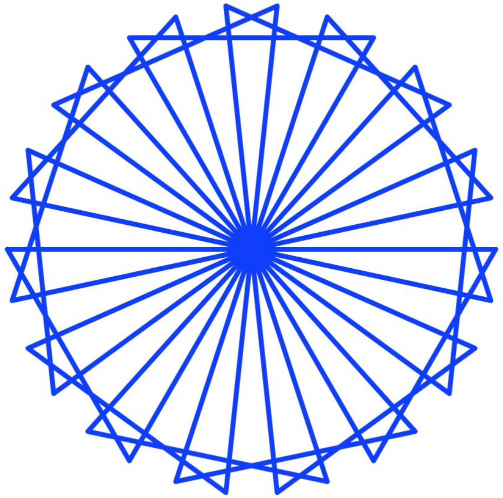
Encountering Fractals in the London Underground

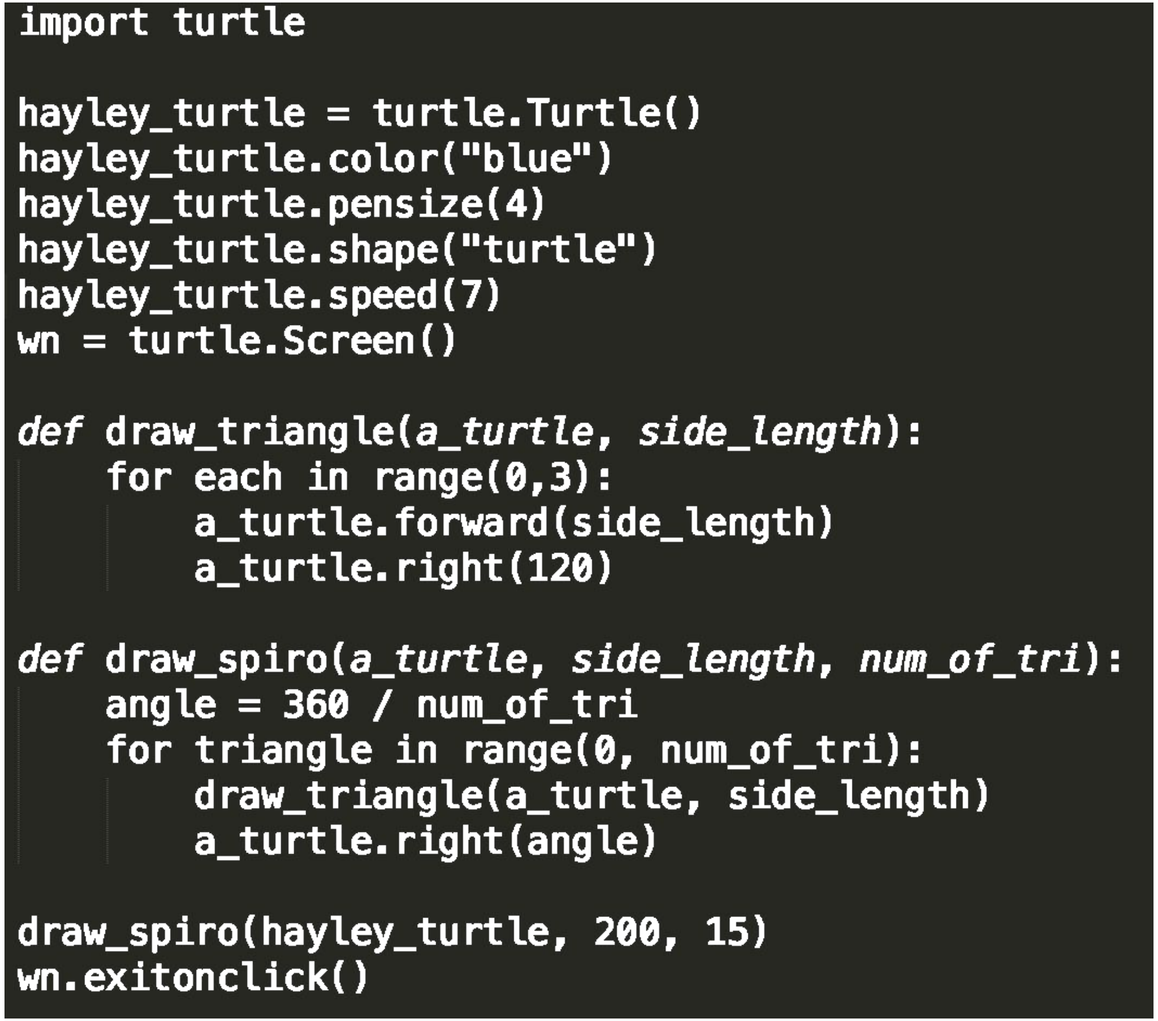


What’s Next?

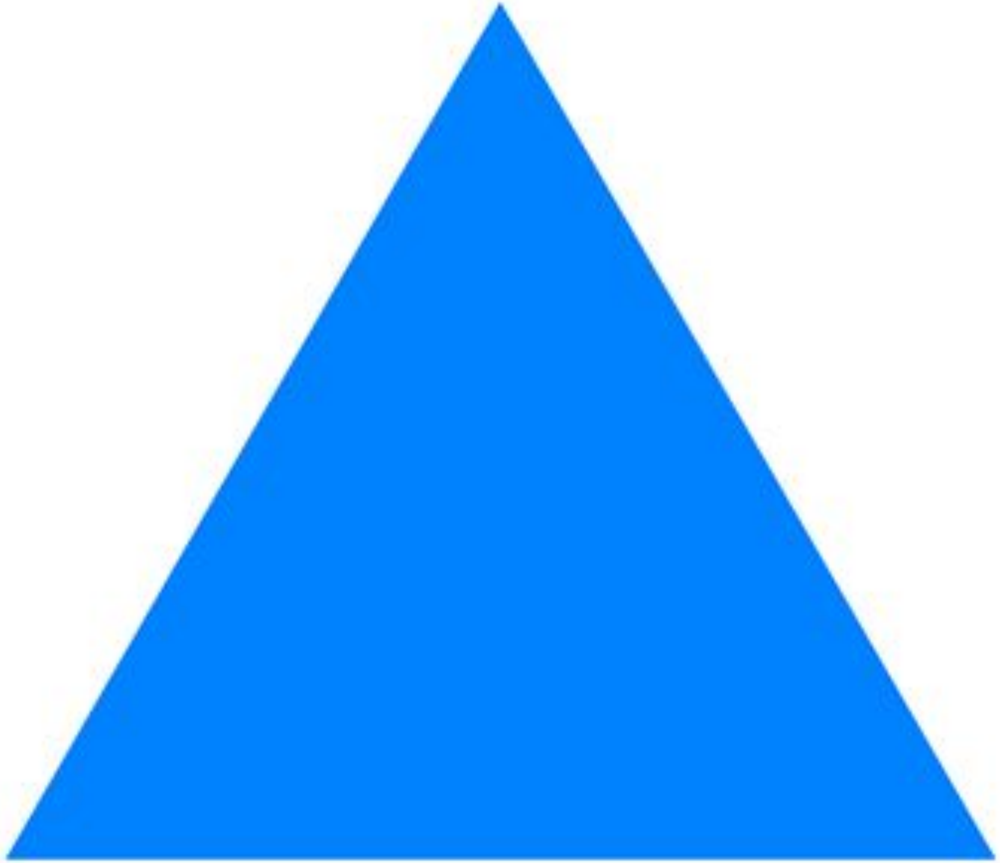
Source

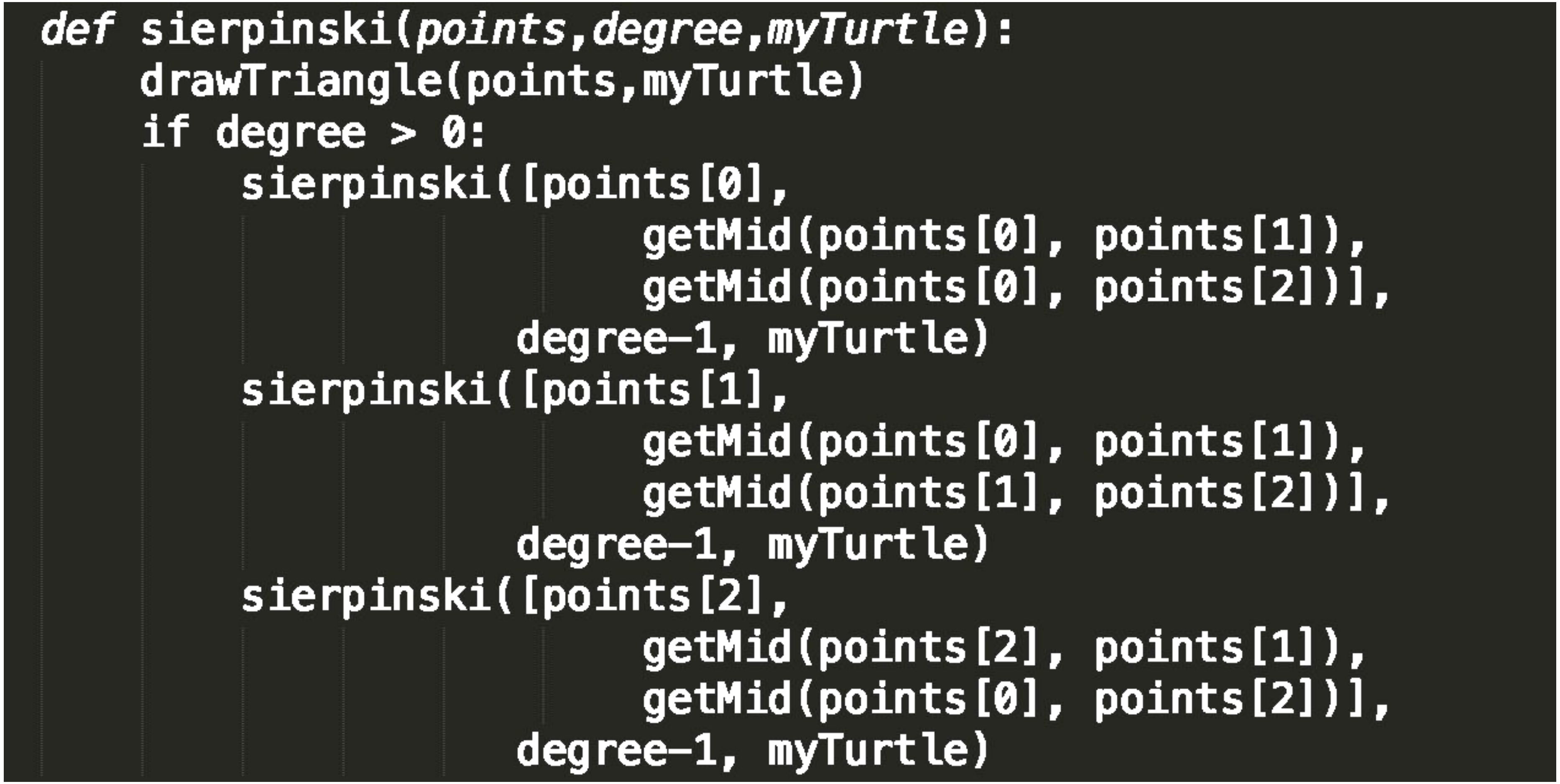
Source



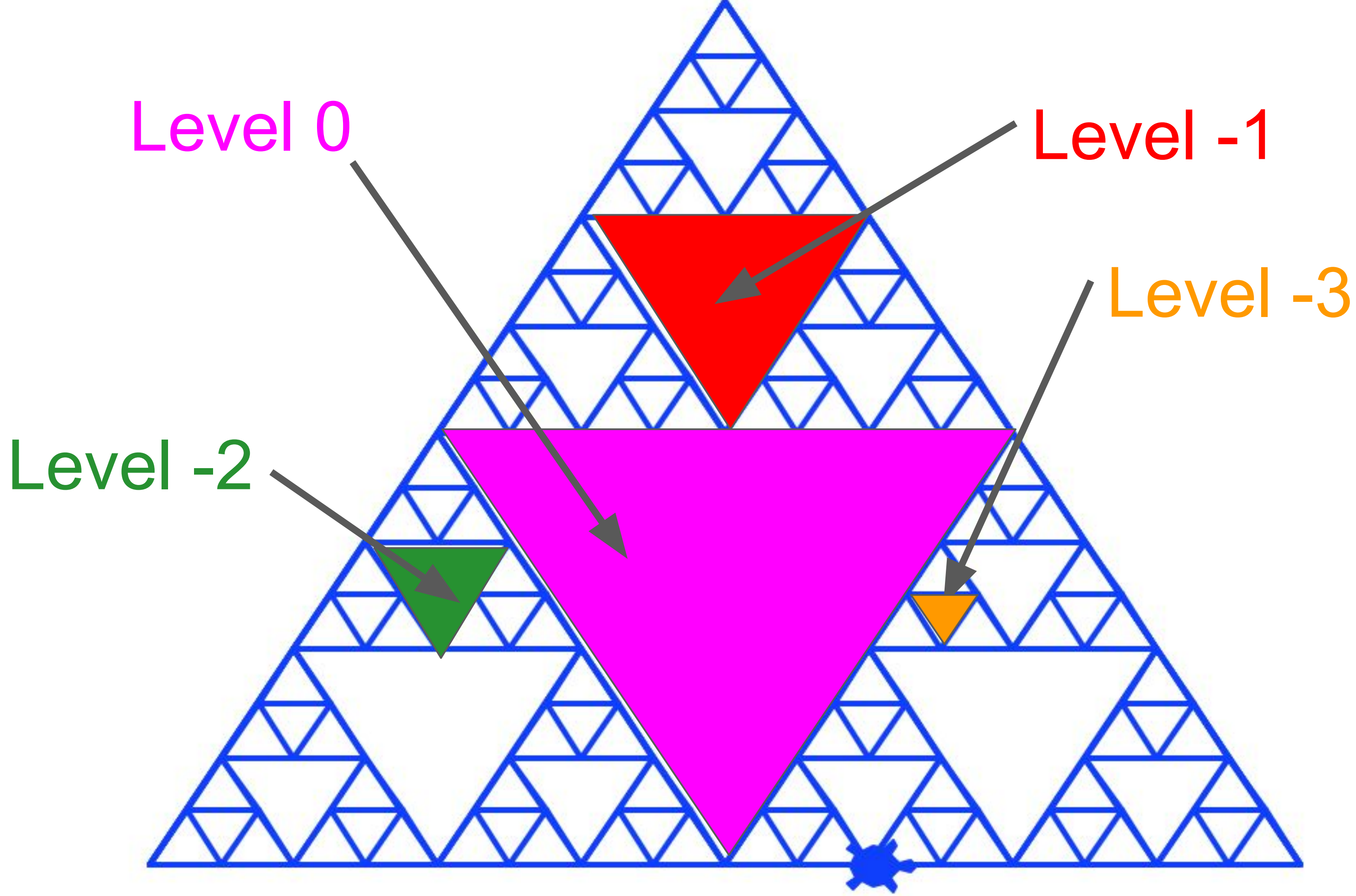


```
import turtle

hayley_turtle = turtle.Turtle()
hayley_turtle.color("blue")
hayley_turtle.pensize(4)
hayley_turtle.shape("turtle")
hayley_turtle.speed(7)
wn = turtle.Screen()

def draw_triangle(a_turtle, side_length):
    for each in range(0,3):
        a_turtle.forward(side_length)
        a_turtle.right(120)

def draw_spiro(a_turtle, side_length, num_of_tri):
    angle = 360 / num_of_tri
    for triangle in range(0, num_of_tri):
        draw_triangle(a_turtle, side_length)
        a_turtle.right(angle)

draw_spiro(hayley_turtle, 200, 15)
wn.exitonclick()
```





```python
def sierpinski(points,degree,myTurtle):
    drawTriangle(points,myTurtle)
    if degree > 0:
        sierpinski([points[0],
                        getMid(points[0], points[1]),
                        getMid(points[0], points[2])],
                   degree-1, myTurtle)
        sierpinski([points[1],
                        getMid(points[0], points[1]),
                        getMid(points[1], points[2])],
                   degree-1, myTurtle)
        sierpinski([points[2],
                        getMid(points[2], points[1]),
                        getMid(points[0], points[2])],
                   degree-1, myTurtle)
```


Level 0
Level -1
Level -3
Level -2

What Have We Learned?

Thank You!

Tweet Me your Python Turtle Creations!

They will all get retweets and scores out of 10

(Similar to the dog_rates twitter, all python turtle creations will get scores above 10/10 because all python turtle creations, like all dogs, are awesome)

@hayleydenb

Resources

Python 3 Turtle Module Docs

How to Think Like a Computer Scientist

Fractals

My Turtle Code