How to Multiply

Reviewing last week’s problem

Problem

Write a function to handle multiplication without the ‘*’ symbol

```
def multiply(num1, num2):
    return num1 * num2
```

Breakdown the problem

1. How do we multiply normally?
2. Is there something that allows us to **add** again and again ***n*** times?
3. Can we use an example? *2x100*

```
sum = 0
for i in range(100):
    sum += 2
print(sum)
```

Creating a Function

1. How can we make this more generic so we can reuse this as a multiplication function?
2. What are our parameters?

```
def multiply(num1, num2):
    sum = 0
    for i in range(num2):
        Sum += num1
    return sum
```

Teaching a turtle to draw

More fun with functions

What is turtle?

A module built-in to Python that allows you to control a robot turtle. See https://docs.python.org/3/library/turtle.html for more information.

Add:

```
from turtle import *
```

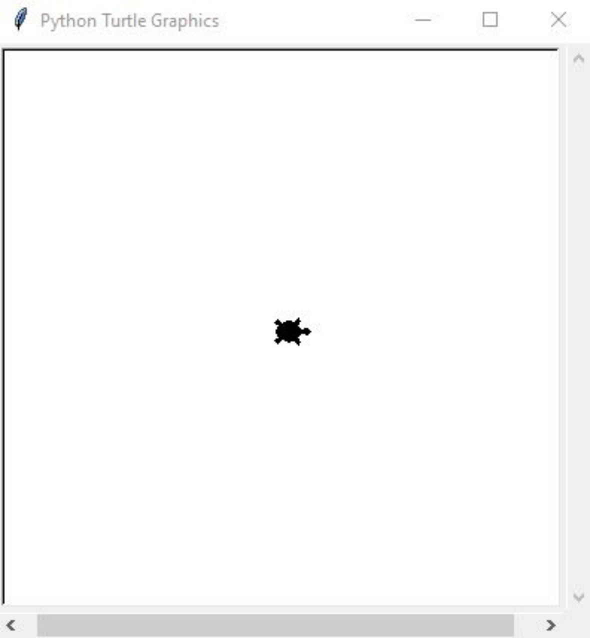
To gain access to the turtle module

Modules

- Self-contained package of code
- Allows you to reuse functionality that other people have created
- Python has many included that come with it (built in)
- Can download many others throughout the internet

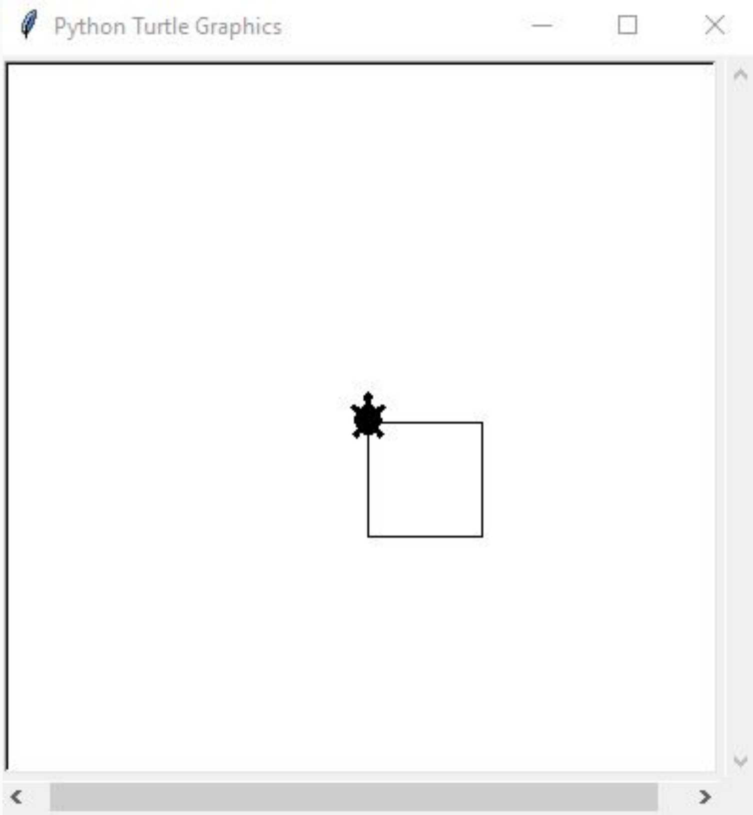
Adopting your turtle (set up)

```
from turtle import *

setup(400, 400)

shape(“turtle”)
```



Main Turtle functions

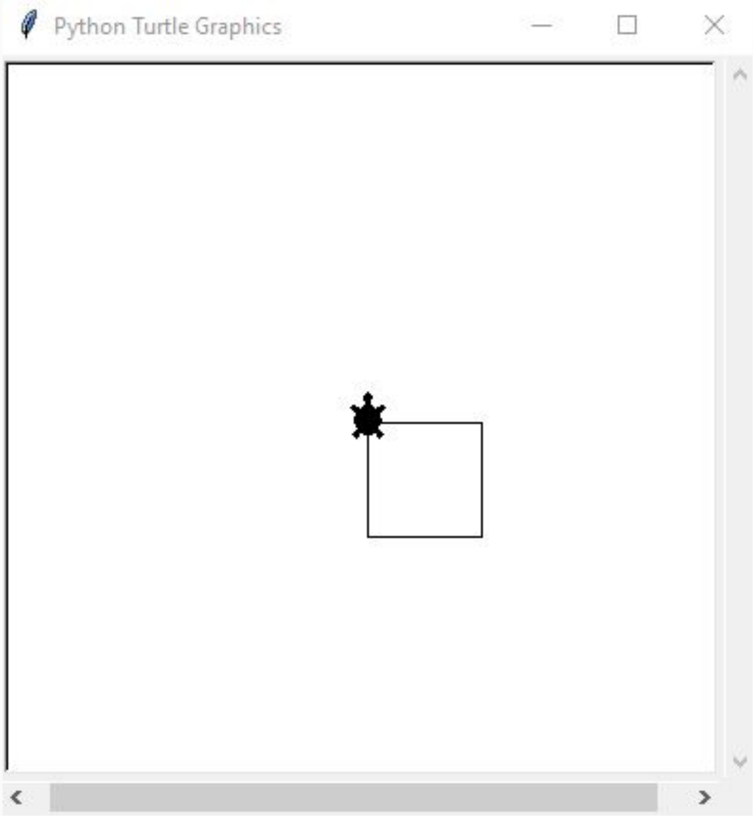
fd(dist)	Command the turtle to move forward by dist
bk(dist)	Command the turtle to move backward by dist
lt(angle)	Command the turtle to turn left by angle degrees
rt(angle)	Command the turtle to turn right by angle degrees
pu()	Command the turtle to raise its pen
pd()	Command the turtle to lower its pen
pensize(width)	Change the thickness of the pen the turtle is using
pencolor(color)	Change the color of the pen the turtle is using
home()	Move the turtle back to coordinates (0,0) the center of the screen
clear()	Clear the canvas, but don’t change the state of the turtle
reset()	Clear the canvas and reset the state of the turtle



Exercise: Draw a square with turtle

Drawing a Square

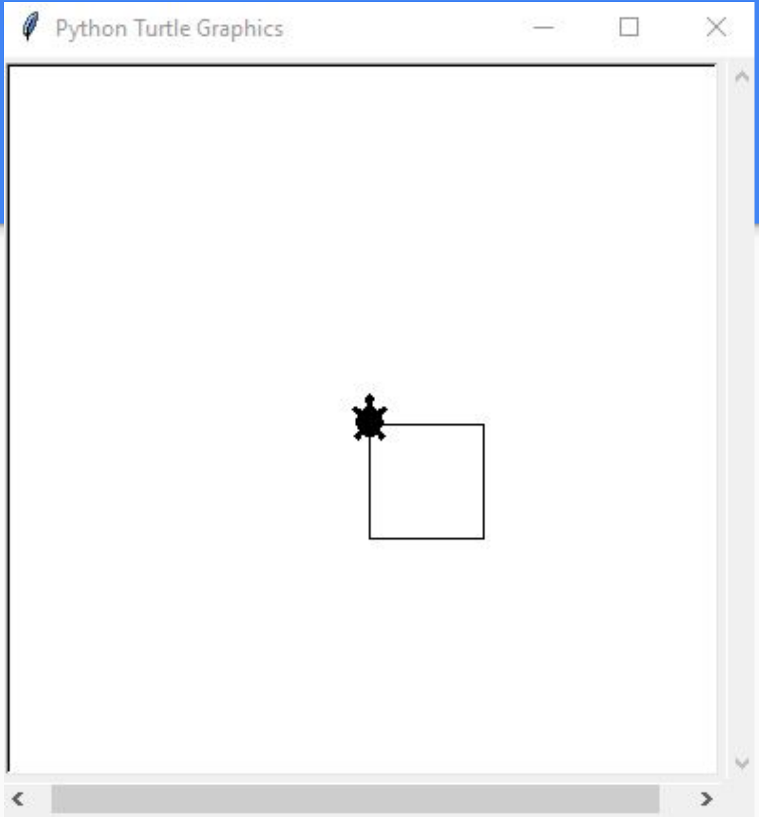
```
pd()
fd(60)
rt(90)
fd(60)
rt(90)
fd(60)
rt(90)
fd(60)
pu()
```



What if I want to reuse it?

Square Function

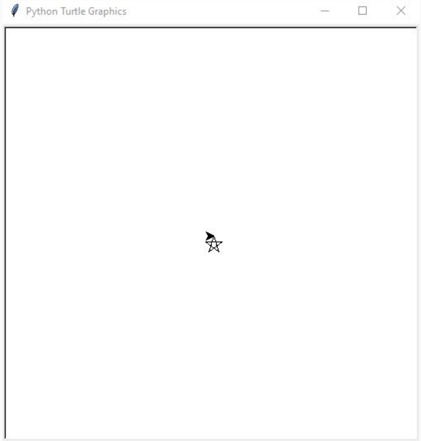
```
def drawSquare():
      pd()
      fd(60)
      rt(90)
      fd(60)
      rt(90)
      fd(60)
      rt(90)
      fd(60)
      pu()
```



What if I want my squares to be different sizes?

Star Function with Length parameter!

```
def drawSquare(length):
     pd()
     fd(length)
     rt(90)
     fd(length)
     rt(90)
     fd(length)
     rt(90)
     fd(length)
     pu()
```



Things to remember

- Functions allow you to break down your logic into small reusable pieces.
- Try to name your functions in a way that it’s clear what they will do
- If something is reusable or useful, it may belong in a function
- Sometimes a complex function may be better broken down into multiple functions

Project!

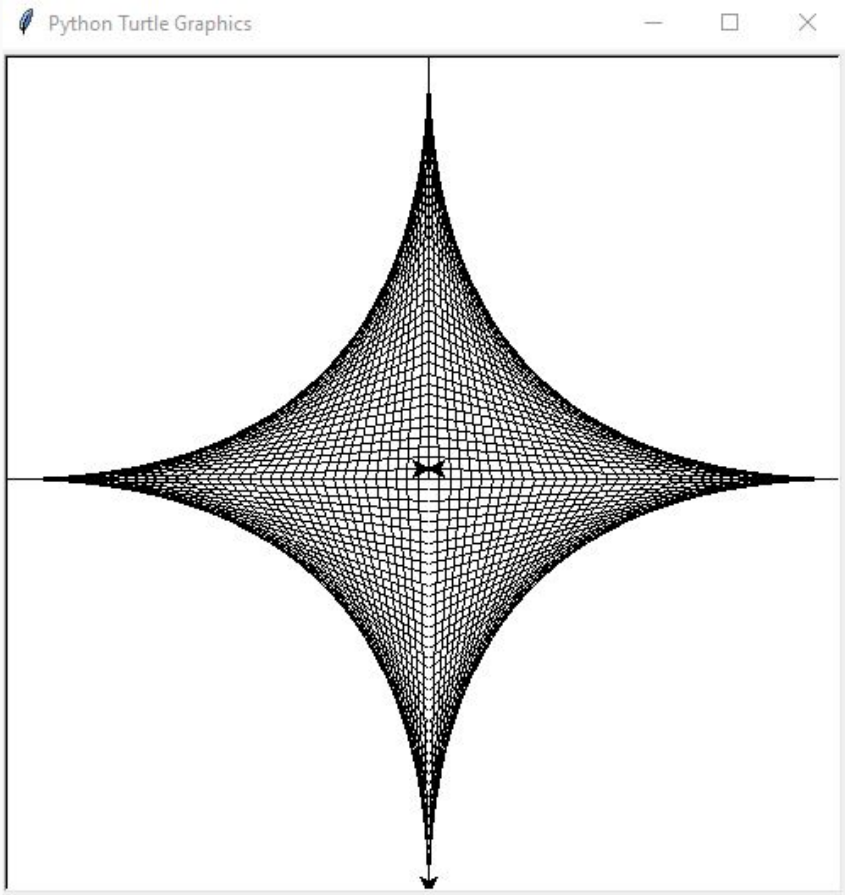
Teach your turtle to draw a shape!

Requirements:

- Have a shape that changes with at least two parameters (e.g. color, size, rotation)
- Draw 50 of that shape randomly on the window

Some ideas:

- Extend the star function and draw a night sky
- Make a fish function (or several) and make a fish tank



Moving turtle without drawing

Teleport your turtle

```
x = 10
y = 20
setheading(towards(x, y))
fd(distance(x,y))
```

Teleport function

```
def teleport(x, y):
    setheading(towards(x, y))
    fd(distance(x,y))
```

Random code

Generating random numbers in Python

At top of your file:

```
import random
```

Then when you need a random number you can do:

```
random.random()
```

Which will give you a random number between 0 - 1.0.

Random numbers

What if I want a number between 0 - 10000 ?

```
import random
```

```
random_number = random.random()* 10000
```

What about between 300 - 10000?

```
import random
```

```
random_number = 300 + (random.random()*(10000 - 300))
```