Folks, this is a brand new activity. If you encounter any issues/typos, please let Iris know!

Name:_________________________________ **Partners:** ___________________________

Python Activity 31: Drawing with Turtle

Learning Objectives

Students will be able to:

Content:

- Predict what **turtle** code will do

Process:

- Write code that draws line drawings

Prior Knowledge

- Python concepts: modules, functions

Critical Thinking Questions:

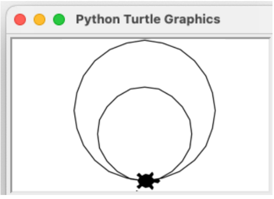
1. Examine the sample code below, which uses the pen-drawing module, `turtle`:

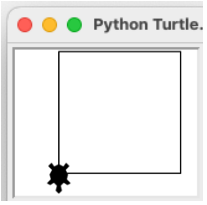
Sample Turtle Code

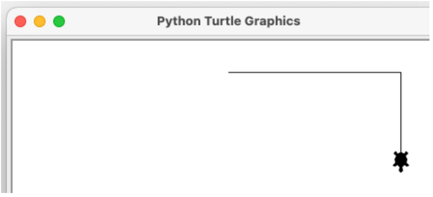
```
from turtle import *
setup(400, 400)

# Sample 1          Sample #2          Sample #3
forward(200)        forward(100)       circle(50)
right(90)           left(90)           circle(75)
forward(100)        forward(100)
                    left(90)
                    forward(100)
                    left(90)
                    forward(100)
```

a. Below is the output from these three code samples. Can you identify which output belongs to which code input?

Sample #____ Sample #____ Sample #____







b. Map the code on the left with what you think it does on the right:

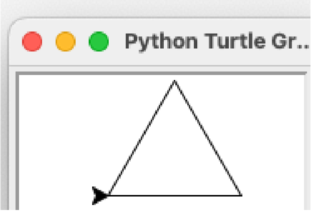
`from turtle import *`	change the color of the inside of our shapes
`setup(width, height)`	move turtle forward a given distance
`right(angle)`	turn left a given angle amount
`left(angle)`	draw a circle with specified radius
`forward(dist)`	import the turtle module so we can use its functions
`backward(dist)`	pull the pen up, so we don't draw
`circle(radius)`	turn right a given angle amount
`begin_fill()`	fills the shape after this command with a color
`fillcolor(color)`	create a window with given width & height
`end_fill()`	cease filling shapes with color
`down()`	move turtle backward a given distance
`up()`	put the pen down, so we draw

FYI: Forward, backward, left, and right are so commonly used in turtle that they have abbreviations: **`fd(..)`**, **`bk(..)`**, **`lt(..)`**, and **`rt(..)`**.

2. Examine the sample code below, and the output from a call to `mystery1(80,3)`:

Sample Turtle Code and Output

```
from turtle import *
setup(400, 400)

def mystery1(length, num_sides):
    for side in range(num_sides):
        fd(length)
        lt(360/ num_sides)
```



a. Trace through the loop in the `mystery1` function for `mystery1(80,3)`:

`length` `num_sides` `range(num_sides)` `side` `fd(length)` `lt(360/ num_sides)`

______ _________ __________________ _____ ____________ ________________

Output:

_____

_____

b. What might a call to `mystery1(80, 10)` draw? *(Hint: you may need to trace through the function again!)*

c. What might the `mystery1(length, num_sides)` function do?

____

Application Questions: Use the Python Interpreter to check your work.

1. Modify the `mystery1(length, num_sides)` function so that it takes a third parameter, `color`, and fills the shape it draws with that color. "purple" and "gold" are example color names that work in the turtle module.

```
from turtle import *
setup(400, 400)

def mystery2(length, num_sides, color):
     # set fill-color here

     # fill!

     for side in range(num_sides):
          fd(length)
          lt(360/num_sides)
     # cease filling!
```