

STEM Workshop: Graphics

Lesson 2: Python Turtle Graphics



Brought to you by the University of Maryland Balloon
Payload Program!

https://repl.it/languages/python_turtle

Introduction to Python Turtle

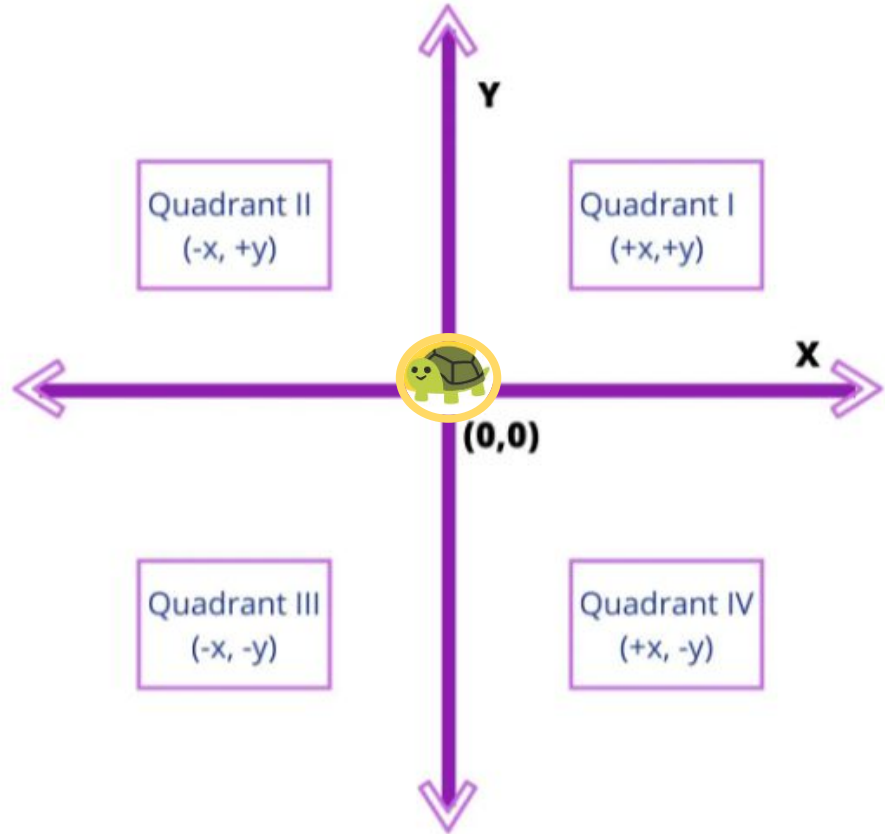


- Turtle is a built-in Python library
 - A library is a set of premade programs and functions so that you don't have to make them yourself from scratch
- Lets us create graphics on virtual canvas
- The pen/arrow used for drawing is called the "Turtle"
- Turtle has characteristics that you can change (e.g. direction, size, color, speed)
- Create the Python environment and initialize the turtle screen

```
1 import turtle
2 screen = turtle.Screen()
```

Turtle Screen

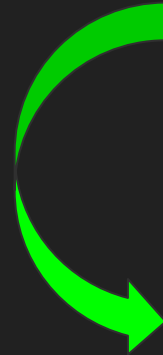
- It's like a canvas, to draw on
- Works like the coordinate plane/graph
- MATH!!!!
- Turtle/Pen starts at "home" (0, 0)



Turtle Controls

- 4 directions: forward, backward, left, right

```
turtle.forward()  
turtle.backward()  
turtle.left()  
turtle.right()
```



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



Drawing Shapes

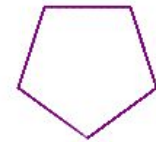
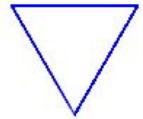
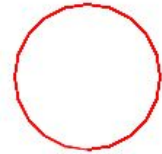
```
1 import turtle
2 Screen = turtle.Screen()
```

```
#circle
turtle.pu()#pen up
turtle.goto(-75,75)#goes to coordinate
turtle.pd()#pen down
turtle.color("red")#set pen color
turtle.circle(50,360)#radius of 50
```

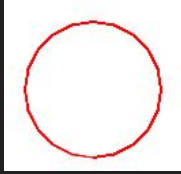
```
#triangle
turtle.pu()
turtle.goto(75,75)
turtle.pd()
turtle.color("blue")
turtle.circle(50, 360, 3)
```

```
#pentagon
turtle.pu()
turtle.goto(-75,-75)
turtle.pd()
turtle.color("purple")
turtle.circle(50,360,5)
```

```
#square
turtle.pu()
turtle.goto(75,-75)
turtle.pd()
turtle.color("yellow")
turtle.circle(50,360,4)
```



Circle



Ready to move the pen

Puts the pen down, ready to draw

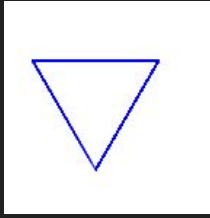
```
#circle
turtle.pu()#pen up
turtle.goto(-75,75)#goes to coordinate
turtle.pd()#pen down
turtle.color("red")#set pen color
turtle.circle(50,360)#radius of 50
```

Chooses where on the screen to draw

Sets drawing color

Draws a circle with radius 50 starting at the point (-75, 75)

Triangle

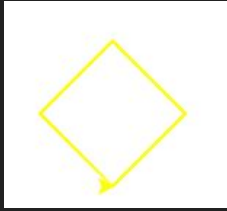


```
#triangle  
turtle.pu()  
turtle.goto(75,75)  
turtle.pd()  
turtle.color("blue")  
turtle.circle(50, 360, 3)
```

3 sides = Triangle

1. Picks the pen up
2. Moves the pen to desired coordinates
3. Puts the pen down at that spot, ready to draw
4. Chooses pen color (blue)
5. Draws triangle at (75, 75)

Square

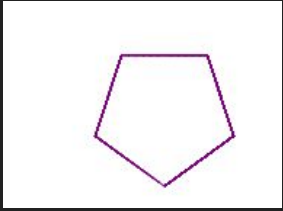


```
#square
turtle.pu()
turtle.goto(75,-75)
turtle.pd()
turtle.color("yellow")
turtle.circle(50,360,4)
```

4 sides = Square

1. Picks the pen up
2. Moves the pen to desired coordinates
3. Puts the pen down at that spot, ready to draw
4. Chooses pen color (yellow)
5. Draws Square at (75, -75)

Pentagon



```
#pentagon
turtle.pu()
turtle.goto(-75,-75)
turtle.pd()
turtle.color("purple")
turtle.circle(50,360,5)
```

5 sides = Pentagon

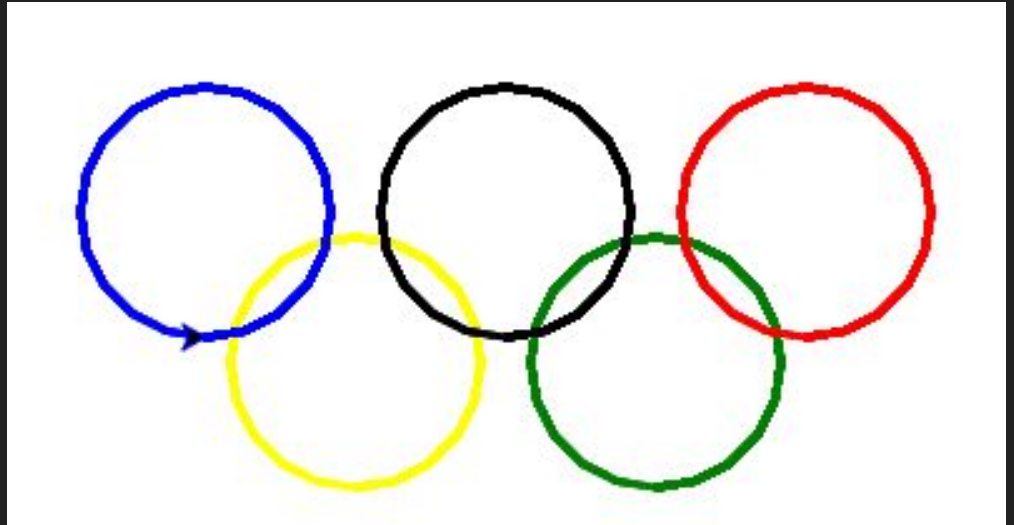
1. Picks the pen up
2. Moves the pen to desired coordinates
3. Puts the pen down at that spot, ready to draw
4. Chooses pen color (purple)
5. Draws Square at (-75, -75)

Olympics Logo

- Consider the following:
 - Shape
 - Color
 - Size
 - Location/position

Hint:

```
#circle  
turtle.pu()#pen up  
turtle.goto(-75,75)#goes to coordinate  
turtle.pd()#pen down  
turtle.color("red")#set pen color  
turtle.circle(50,360)#radius of 50
```



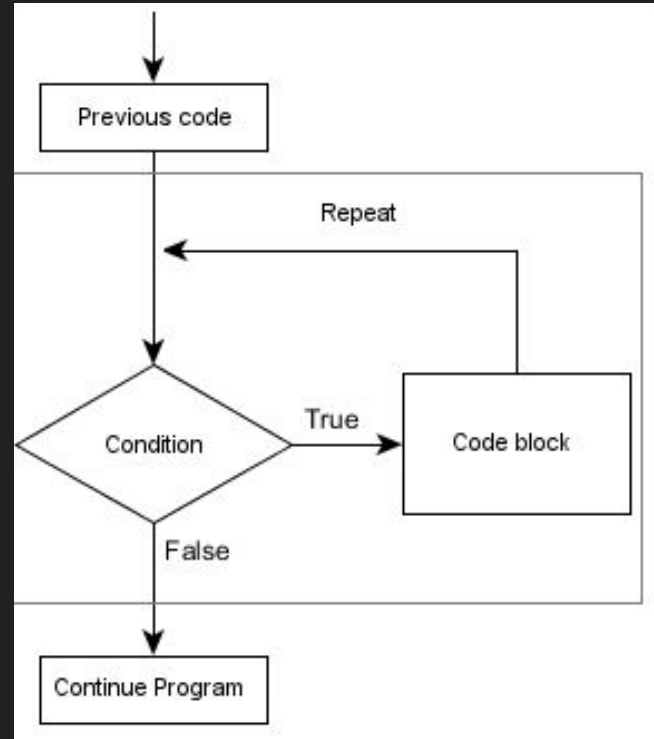
Olympics Logo Code

```
1  import turtle
2  screen = turtle.Screen
3
4  turtle.pensize(3)
5  turtle.pencolor("green")
6  turtle.circle(50)
7
8  turtle.penup()
9  turtle.setposition(-120, 0)
10 turtle.pendown()
11 turtle.pencolor("yellow")
12 turtle.circle(50)
```

```
13 turtle.penup()
14 turtle.setposition(60, 60)
15 turtle.pendown()
16 turtle.pencolor("red")
17 turtle.circle(50)
18
19 turtle.penup()
20 turtle.setposition(-60, 60)
21 turtle.pendown()
22 turtle.pencolor("black")
23 turtle.circle(50)
24
25 turtle.penup()
26 turtle.setposition(-180, 60)
27 turtle.pendown()
28 turtle.pencolor("blue")
29 turtle.circle(50)
```

Loops

- Loops: set of instructions that are continuously repeated *until something happens*
 - Types: 'for' loops, 'while' loops

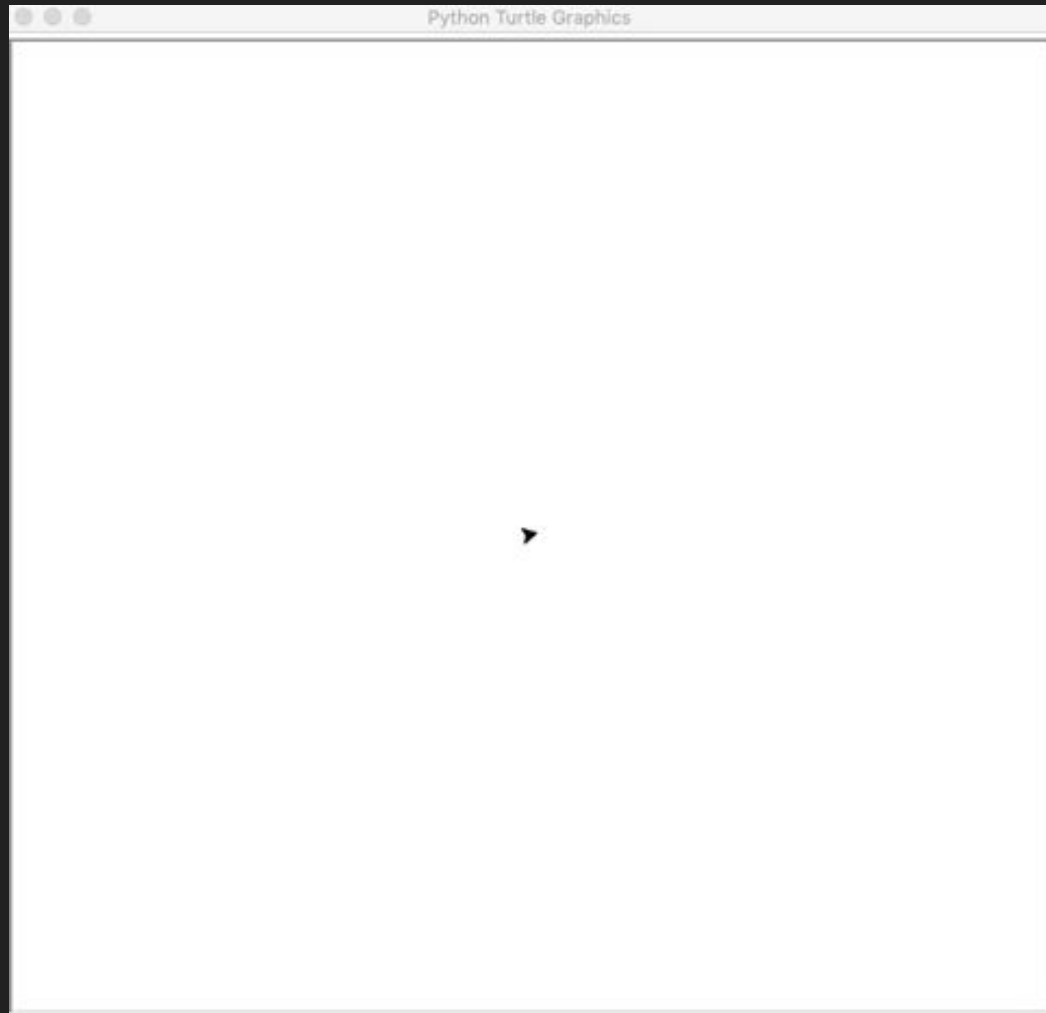


Looping Example

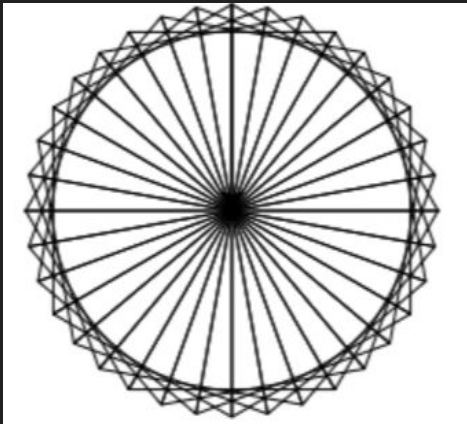
```
n = 10
while n <= 40:
    turtle.circle(n)
    n = n + 10
```

Trace Table:

| | |
|--------|-------------------|
| n = 10 | 10 <= 40 |
| n = 20 | 20 <= 40 |
| n = 30 | 30 <= 40 |
| n = 40 | 40 <= 40 Stops |



Spirographs

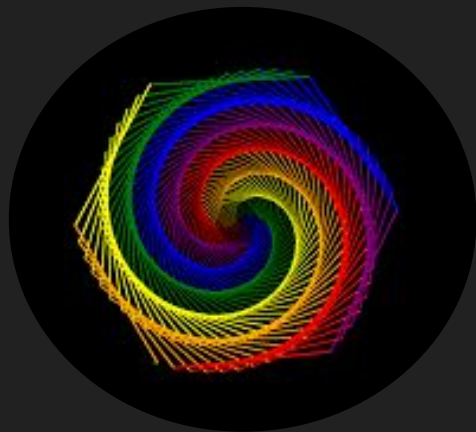
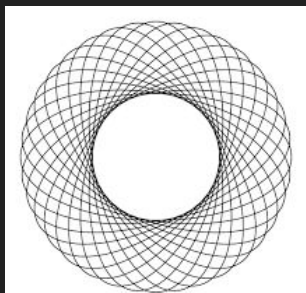
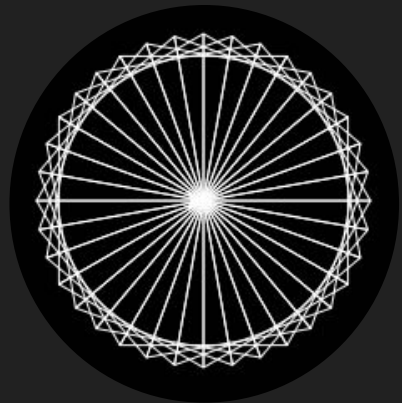


```
1 import turtle
2 screen= turtle.Screen()
3 turtle.speed(0)
4
5 a=0
6 while (a < 36):
7     turtle.forward (100)
8     turtle.left(120)
9     turtle.forward(100)
10    turtle.left(120)
11    turtle.forward(100)
12    turtle.left(130)
13    a = a+1
```

Challenge: Drawing Spirographs



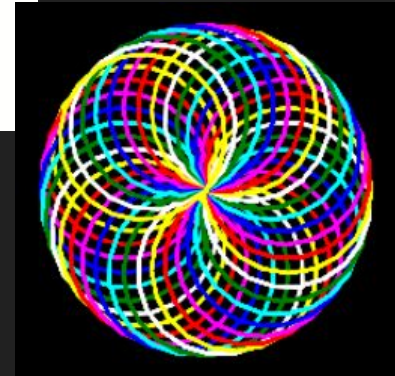
Create your own spirographs!



Spirographs Code

*There are **MANY** solutions, this code is just a possible one :)*

```
1 import turtle
2 Screen = turtle.Screen()
3
4 Screen.bgcolor('black')
5 turtle.pensize(2)
6 turtle.speed(10)
7
8 for i in range(6):
9     for color in ('red', 'magenta', 'blue', 'cyan', 'green', 'white', 'yellow'):
10         turtle.color(color)
11         turtle.circle(50)
12         turtle.left(10)
```



Other Repls:

Activity #1: Shapes

<https://repl.it/@EddieFang/Activity-1-Shapes#main.py>

Activity #2: Concentric Circles

<https://repl.it/@EddieFang/Turtle-Activity-2b-Concentric-Circles#main.py>

Activity #3: Solar System

<https://repl.it/@EddieFang/Turtle-Activity-3-Solar-System#main.py>

THANKS!

