

---

# Python with Turtle

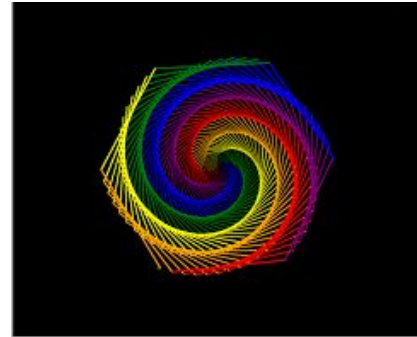
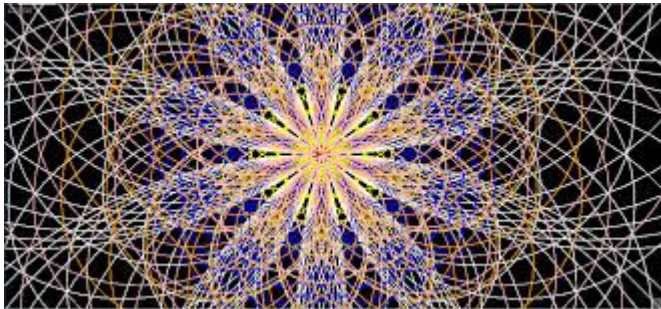
By Sarah, Liana, and Vivian



---

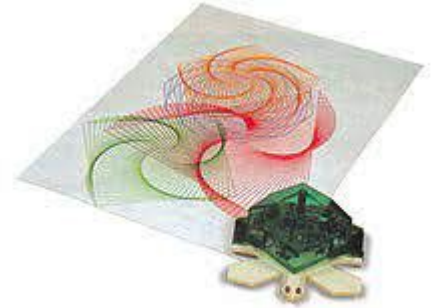
# What is Python with Turtle?

Python with Turtle is a Python library that enables you to create virtual drawings and shapes. The “turtle” is a the pen that you use to draw.

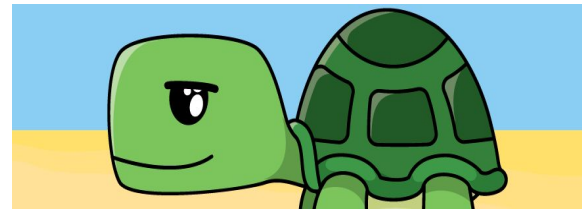


---

## History



- Part of the original Logo programming language developed by Wally Feurzig and Seymour Papert in 1966
- Turtles are a type of educational robots designed in the 1940s, that are typically designed low to the ground. Some turtles are specifically designed to have pens attached to the back of them so they can draw on large sheets of paper.
- It uses tkinter for the underlying graphics, it needs a version of Python installed with Tk support.



## A Basic Program on Turtle

- There is no programming language called 'turtle', so we use Python whenever we import the turtle library. Because of this, on our first line of code, we need to tell our computer we are using turtle:

```
>> import turtle
```

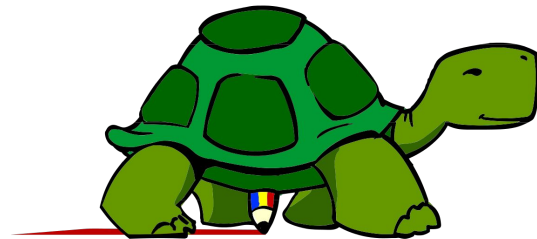
- Next, we need to name our turtle, and turtle will respond to whatever name you give it. We are just going to name it "alfred".

```
>> alfred = turtle.Turtle()
```

- After that, we can make alfred move forward, and we can do this by using the forward command in turtle. Whatever number you put in the parentheses is how many pixels alfred will move forward.

```
>> alfred.forward(40)
```

## A Basic Program on Turtle (Part 2)



- Similarly, we can have alfred move backwards.

```
>> alfred.backward(40)
```

- We can also increase the speed of alfred by using the following command:

```
>> alfred.speed(100)
```

- Now, we want to be able to control alfred with the arrow keys, so we will put down the following function to allow the screen to follow certain commands.

```
>> screen = turtle.Screen()
```



You can also give your screen a different name



## A Basic Program on Turtle (Part 3)

- Next, we need to create functions that are going to be in charge of alfred moving up, down, left, and right.

```
>> def moveUp():
```

```
    alfred.setheading(90)
```

```
    alfred.forward(50)
```

```
>>def moveDown():
```

```
    alfred.setheading(-90)
```

```
    alfred.forward(50)
```

```
>> def moveRight():
```

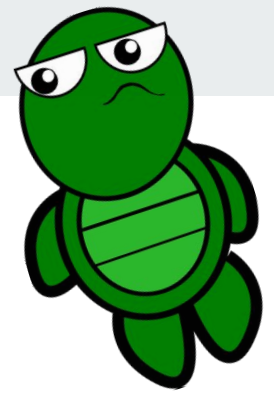
```
    alfred.setheading(0)
```

```
    alfred.forward(50)
```

```
>> def moveLeft():
```

```
    alfred.setheading(180)
```

```
    alfred.forward(50)
```



## A Basic Program on Turtle (Part 4)

- Now we need to make alfred move in each direction when the corresponding key is pressed, we can do this by using the following commands that call the functions we made earlier:

```
>>screen.onkey(moveUp,"Up")
```

```
>>screen.onkey(moveDown,"Down")
```

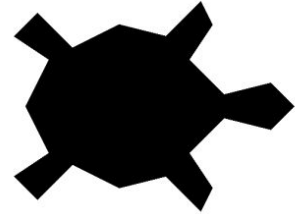
```
>>screen.onkey(moveRight,"Right")
```

```
>>screen.onkey(moveLeft,"Left")
```

- Next, we need to make the Screen listen to the commands we are giving it  
>>screen.listen()

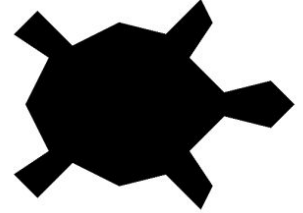


# Turtle Functions



- `turtle.Turtle()`
  - Makes a Turtle
- `turtle.Screen()`
  - Makes a Screen
- `.forward(int)`
  - Moves Turtle Forward
- `.backward(int)`
  - Moves Turtle Backward
- `.left(int)`
  - Turns Turtle Left
- `.right(int)`
  - Turns Turtle Right
- `.color(string or int,int,int).`
  - Changes Turtle Color
- `.penup().`
  - Enables the turtle to move around without drawing
- `.pendown().`
  - Enables drawing
- `.pensize(int)`
  - Changes Turtle Size
- `.speed(int)`
  - Changes Turtle Speed
- `.shape(string).`
  - Changes Turtle Shape
- `.fillcolor(string)`
  - Sets to fill in something drawn by the turtle with a color
- `.begin_fill().`
  - Starting spot for filling color
- `.end_fill().`
  - Ending spot for filling color
- `.xcor()`
  - Find X coordinate of Turtle
- `.ycor()`
  - Find Y coordinate of Turtle





## Turtle Functions Continued

- `.goto(int, int).`
  - Moves Turtle to a specific Spot
- `.circle(int)`
  - Draws a Circle
- `.setheading(int)`
  - Set orientation of the Turtle
- `.home().`
  - Send Turtle to 0, 0 and makes it go back to its starting orientation
- `.undo().`
  - Undo the Turtle last actions
- `.position()`
  - Returns Turtle's position
- `.towards(int, int)`
  - Returns the angle from the Turtle's position to the position given
- `.heading()`
  - Returns Turtle's current heading
- `.distance(int, int or turtle).`
  - Returns distance between turtle and given point or another turtle
- `.degrees(int)`
  - Set Number of Degrees for a Full Circle
- `.radians()`
  - Set Angle Measurement Units to Radians
- `.isdown()`
  - Check to see if Pen is Down or Up
- `.reset().`
  - Delete All of Turtle's Drawings, re-center Turtle, put everything back to default

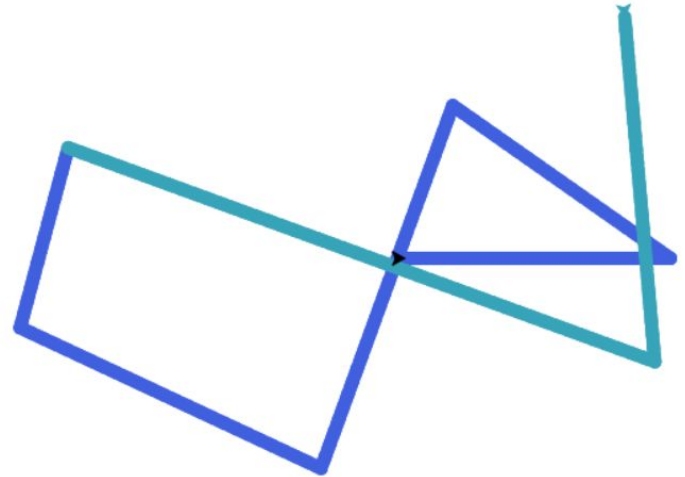
# Interactive #1

<https://repl.it/@silverhailwolf/FinalProjectFileA#main.py>

main.py

```
24 #This function turns the turtle to the left
25 def tcol():
26     r = random.randint(0, 255)
27     g = random.randint(0, 255)
28     b = random.randint(0, 225)
29     t.color(r, g, b)
30 #This changes the color of the turtle to a random color
31 def ufor():
32     u.forward(10)
33 def ubac():
34     u.backward(10)
35 def urig():
36     u.right(5)
37 def ulef():
38     u.left(5)
39 def ucol():
40     r = random.randint(0, 255)
41     g = random.randint(0, 255)
42     b = random.randint(0, 225)
43     u.color(r, g, b)
```

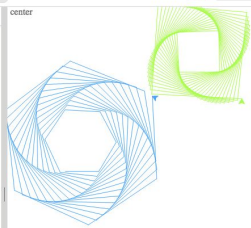
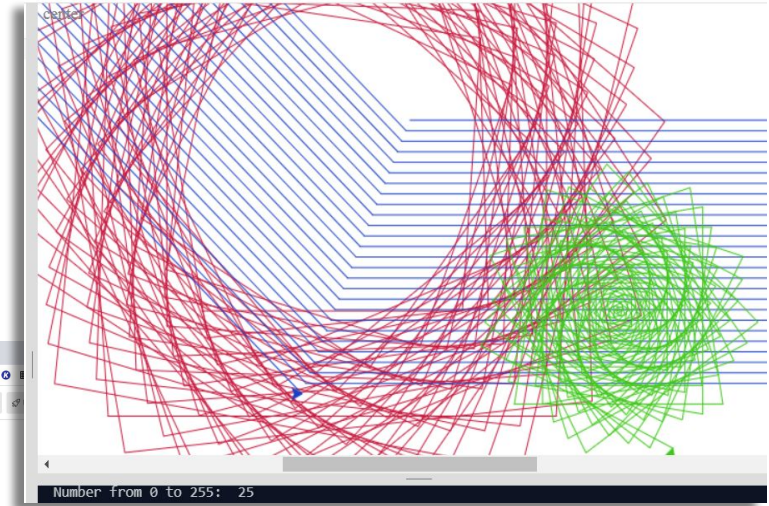
center



# Interactive #2

<https://repl.it/@silverhailwolf/FinalProjectFileB#main.py>

```
Repl.it - FinalProjectFileB
repl.it/silverhailwolf/FinalProjectFileB#main.py
@silverhailwolf/FinalProjectFileB
main.py
25 v.color(c, a, b)
26 #This changes the color of the turtles
27 t.up()
28 u.up()
29 v.up()
30 #This lifts up the turtle so that it doesn't leave a path
  behind it
31 t.goto(-200, -75)
32 u.goto(0, 100)
33 v.goto(200, -75)
34 #This moves the turtle to a spot on the screen
35 t.down()
36 u.down()
37 v.down()
38 #This puts the turtle back down
39 for i in range(d):
40     #That makes the turtle do through it d many times
41     t.forward(g)
42     u.forward(h)
43     v.forward(l)
44     #This moves the turtle forward
45     t.left(j)
46     u.left(k)
47     v.left(i)
48     #This turns the turtles to the left
49     g += 1
50     h += 1
51     l += 1
52     #This makes the length longer
```



```
Number from 0 to 255: 96
Number from 0 to 255: 182
Number from 0 to 255: 249
Number of Rounds: 300
First Length: 59
Second Length: 72
Third Length: 84
Degree One: 59
Degree Two: 91
Degree Three: 118
```



## Resources

- ([https://docs.python.org/3.3/library/turtle.html?highlight=turtle#:~:text=Turtle%20graphics%20is%20a%20popular,0\)%20in%20the%20x%2Dy%20plane.](https://docs.python.org/3.3/library/turtle.html?highlight=turtle#:~:text=Turtle%20graphics%20is%20a%20popular,0)%20in%20the%20x%2Dy%20plane.))
- Artemis Slides on Website:  
<http://www.bu.edu/lernet/artemis/lessons/Week3/Python%20with%20Turtle2020.pdf>
- <https://docs.python.org/3/library/turtle.html#turtle.setheading>
- MORE TURTLE FUNCTIONS: <https://docs.python.org/3/library/turtle.html>

**Thanks for  
listening!**

—