Turtle Programming in Python

Introduction | turtle module

"Turtle" is a Python feature like a drawing board, which lets us command a turtle to draw all over it! We can use functions like turtle.forward(...) and turtle.right(...) which can move the turtle around.Commonly used turtle methods are :

METHOD	PARAMETER	DESCRIPTION
Turtle()	None	Creates and returns a new tutrle object
forward()	amount	Moves the turtle forward by the specified amount
backward()	amount	Moves the turtle backward by the specified amount
right()	angle	Turns the turtle clockwise
left()	angle	Turns the turtle counter clockwise
penup()	None	Picks up the turtle's Pen
pendown()	None	Puts down the turtle's Pen
up()	None	Picks up the turtle's Pen
down()	None	Puts down the turtle's Pen
color()	Color name	Changes the color of the turtle's pen
fillcolor()	Color name	Changes the color of the turtle will use to fill a polygon
heading()	None	Returns the current heading
position()	None	Returns the current position
goto()	x, y	Move the turtle to position x,y
begin_fill()	None	Remember the starting point for a filled polygon
end_fill()	None	Close the polygon and fill with the current fill color
dot()	None	Leave the dot at the current position
stamp()	None	Leaves an impression of a turtle shape a the current location

METHOD	PARAMETER	DESCRIPTION
shape()	shapename	Should be 'arrow', 'classic', 'turtle' or 'circle'

Plotting using Turtle

To make use of the turtle methods and functionalities, we need to import turtle."turtle" comes packed with the standard Python package and need not be installed externally.The roadmap for executing a turtle program follows 4 steps:

1. Import the turtle module
2. Create a turtle to control.
3. Draw around using the turtle methods.
4. Run turtle.done().

So as stated above, before we can use turtle, we need to import it.We import it as :

```
from turtle import *
# or
import turtle
```

After importing the turtle library and making all the turtle functionalities available to us, we need to create a new drawing board(window) and a turtle. Let's call the window as wn and the turtle as skk. So we code as:

```
wn = turtle.Screen()
wn.bgcolor("light green")
wn.title("Turtle")
skk = turtle.Turtle()
```

Now that we have created the window and the turtle, we need to move the turtle. To move forward 100 pixels in the direction skk is facing, we code:

```
skk.forward(100)
```

We have moved skk 100 pixels forward, Awesome! Now we complete the program with the done() function and We're done!

```
turtle.done()
```

So, we have created a program that draws a line 100 pixels long. We can draw various shapes and fill different colors using turtle methods. There's plethora of functions and programs to be coded using the turtle library in python. Let's learn to draw some of the basic shapes.

Shape 1: Square

```
# Python program to draw square
# using Turtle Programming
import turtle
skk = turtle.Turtle()

for i in range(4):
    skk.forward(50)
    skk.right(90)

turtle.done()
```

Shape 2: Star

```
# Python program to draw star
# using Turtle Programming
import turtle

star = turtle.Turtle()

for i in range(50):
    star.forward(50)
```

```
    star.right(144)

turtle.done()
```

Shape 3: Hexagon

```
# Python program to draw hexagon
# using Turtle Programming
import turtle
polygon = turtle.Turtle()

num_sides = 6
side_length = 70
angle = 360.0 / num_sides

for i in range(num_sides):
    polygon.forward(side_length)
    polygon.right(angle)

turtle.done()
```

Visit pythonturtle.org to get a taste of Turtle without having python pre-installed. The shell in PythonTurtle is a full Python shell, and you can do with it almost anything you can with a standard Python shell. You can make loops, define functions, create classes, etc.
You can access these codes for wonderful turtle programs here

Some amazing Turtle Programs

1. Spiral Square Outside In and Inside Out

```
# Python program to draw
# Spiral Square Outside In and Inside Out
# using Turtle Programming
import turtle   #Outside_In
wn = turtle.Screen()
wn.bgcolor("light green")
wn.title("Turtle")
skk = turtle.Turtle()
skk.color("blue")

def sqrfunc(size):
    for i in range(4):
        skk.fd(size)
        skk.left(90)
        size = size-5

sqrfunc(146)
sqrfunc(126)
sqrfunc(106)
sqrfunc(86)
sqrfunc(66)
sqrfunc(46)
sqrfunc(26)
```

```
import turtle  #Inside_Out
wn = turtle.Screen()
wn.bgcolor("light green")
skk = turtle.Turtle()
skk.color("blue")

def sqrfunc(size):
    for i in range(4):
        skk.fd(size)
        skk.left(90)
        size = size + 5

sqrfunc(6)
sqrfunc(26)
sqrfunc(46)
sqrfunc(66)
```

```
sqrfunc(86)
sqrfunc(106)
sqrfunc(126)
sqrfunc(146)
```

Output:

2. User Input Pattern

```
# Python program to user input pattern
# using Turtle Programming
import turtle   #Outside_In
import turtle
import time
import random

print ("This program draws shapes based on the number you enter in a uniform pattern.")
num_str = input("Enter the side number of the shape you want to draw: ")
if num_str.isdigit():
    squares = int(num_str)

angle = 180 - 180*(squares-2)/squares

turtle.up

x = 0
y = 0
turtle.setpos(x, y)


numshapes = 8
for x in range(numshapes):
    turtle.color(random.random(), random.random(), random.random())
    x += 5
    y += 5
    turtle.forward(x)
    turtle.left(y)
    for i in range(squares):
        turtle.begin_fill()
        turtle.down()
        turtle.forward(40)
        turtle.left(angle)
        turtle.forward(40)
        print (turtle.pos())
        turtle.up()
        turtle.end_fill()

time.sleep(11)
turtle.bye()
```

3. Spiral Helix Pattern

```
# Python program to draw
# Spiral  Helix Pattern
```

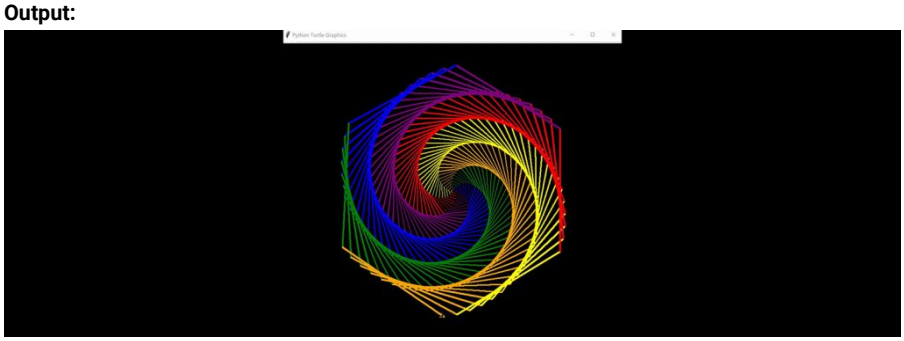
```
# using Turtle Programming

import turtle
loadWindow = turtle.Screen()
turtle.speed(2)

for i in range(100):
    turtle.circle(5*i)
    turtle.circle(-5*i)
    turtle.left(i)

turtle.exitonclick()
```

Output:

4. Rainbow Benzene

```
# Python program to draw
# Rainbow Benzene
# using Turtle Programming
import turtle
colors = ['red', 'purple', 'blue', 'green', 'orange', 'yellow']
t = turtle.Pen()
turtle.bgcolor('black')
for x in range(360):
    t.pencolor(colors[x%6])
    t.width(x/100 + 1)
    t.forward(x)
    t.left(59)
```

Output:



Trees using Turtle Programming

References:

- Turtle documentation for Python 3 and 2
- eecs.wsu.edu [PDF] !

This article is contributed by **Amartya Ranjan Saikia**. If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Recommended Posts:

How to write an empty function in Python - pass statement?
Operator Functions in Python | Set 2
Time Functions in Python | Set-2 (Date Manipulations)
Send mail from your Gmail account using Python
Print Single and Multiple variable in Python
Increment and Decrement Operators in Python
str() vs repr() in Python
Swap two variables in one line in C/C++, Python, PHP and Java
Generate all permutation of a set in Python
Class or Static Variables in Python
trunc() in Python
Division Operators in Python
Interesting facts about strings in Python | Set 1
When to use yield instead of return in Python?
How to split a string in C/C++, Python and Java?

Article Tags : Python

3

To-do Done

0

No votes yet.