

Turtle graphics using Python

Turtle is a Python library to draw graphics. After we import Turtle we can give commands like forward, backward, right, left etc. These commands will draw different shapes when we use them. When we combine these commands we can create many nice graphics. In the below example we will see some simple scenarios and then some complex ones where nice graphics are created.

Simple Turtle Commands

- **forward(10)** It moves the turtle (arrow) forward by 10 pixels.
- **backward(5)** It moves the turtle (arrow) backward by 5 pixels
- **right(35)** It moves the turtle (arrow) clockwise by an angle of 35 degrees.
- **left(55)** It moves the turtle (arrow) counter-clockwise by an angle of 55 degrees
- **goto(x,y)** It moves the turtle (arrow) to the position x, y
- **dot()** It creates a dot in the current position.
- **shape('circle')** It draws a circle shape.

Turtle is a special feature of Python. Using Turtle, we can easily draw in a drawing board.

First we import the turtle module. Then create a window, next we create turtle object and using turtle method we can draw in the drawing board.

Some turtle method

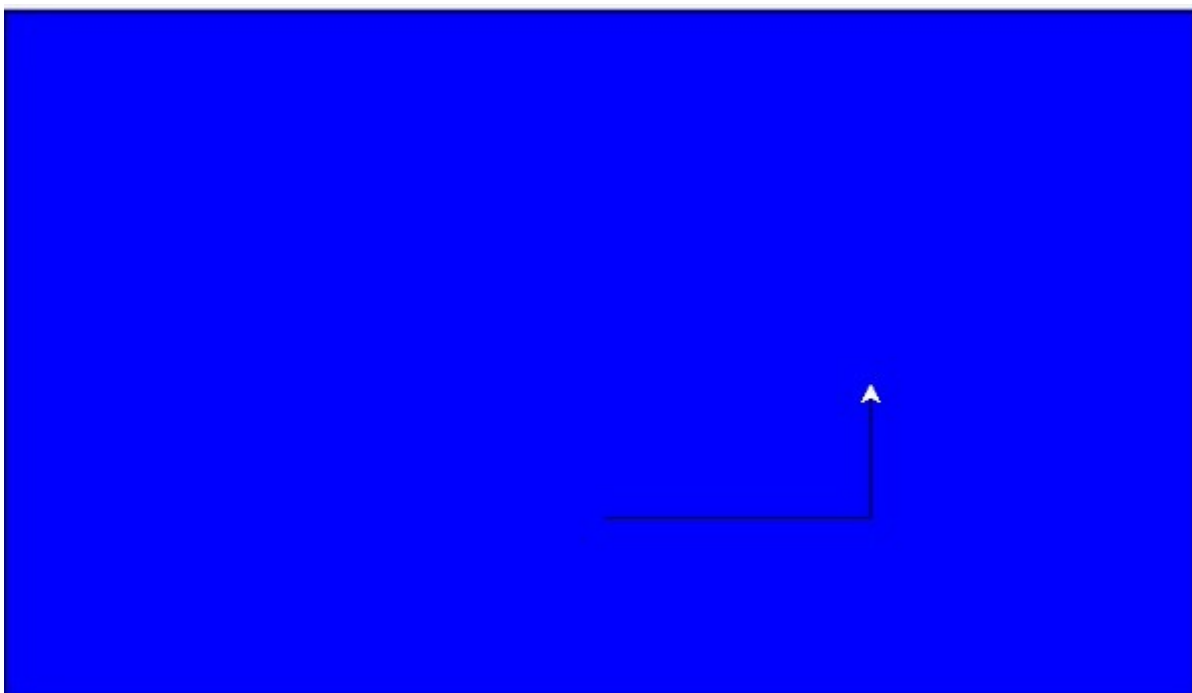
METHOD	PARAMETER	DESCRIPTION
Turtle()	None	It creates and returns a new turtle object
forward()	amount	It moves the turtle forward by the specified amount
backward()	amount	It moves the turtle backward by the specified amount
right()	angle	It turns the turtle clockwise
left()	angle	It turns the turtle counter clockwise
penup()	None	It 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	It returns the current heading
position()	None	It returns the current position

goto()	x, y	It moves the turtle to position x,y
begin_fill()	None	Remember the starting point for a filled polygon
end_fill()	None	It closes the polygon and fills with the current fill color
dot()	None	Leaves the dot at the current position
stamp()	None	Leaves an impression of a turtle shape at the current location
shape()	shapename	Should be 'arrow', 'classic', 'turtle' or 'circle'

Example code

```
# import turtle library
import turtle
my_window = turtle.Screen()
my_window.bgcolor("blue")    # creates a graphics window
my_pen = turtle.Turtle()
my_pen.forward(150)
my_pen.left(90)
my_pen.forward(75)
my_pen.color("white")
my_pen.pensize(12)
```

Output



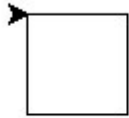
Draw a Square

Example code

```
# import turtle library
import turtle
```

```
my_pen = turtle.Turtle()
for i in range(4):
    my_pen.forward(50)
    my_pen.right(90)
turtle.done()
```

Output



Draw a star

Example code

```
# import turtle library
import turtle
my_pen = turtle.Turtle()
for i in range(50):
    my_pen.forward(50)
    my_pen.right(144)
turtle.done()
```

Output

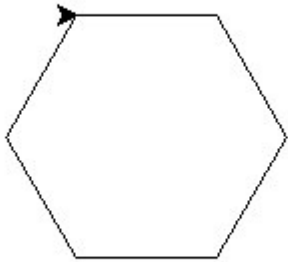


Draw a Hexagon

Example code

```
# import turtle library
import turtle
polygon = turtle.Turtle()
my_num_sides = 6
my_side_length = 70
my_angle = 360.0 / my_num_sides
for i in range(my_num_sides):
    polygon.forward(my_side_length)
    polygon.right(my_angle)
turtle.done()
```

Output

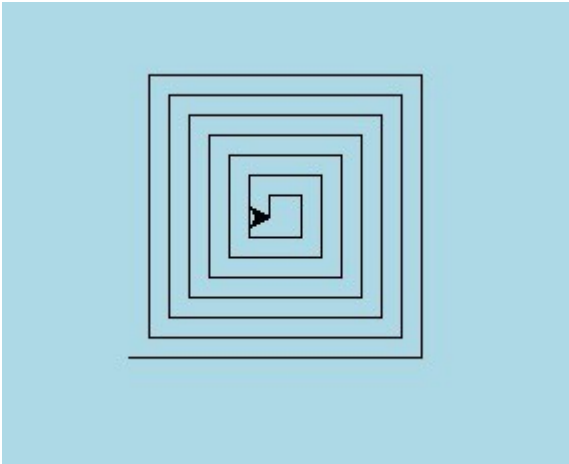


Draw a square inside another square box.

Example code

```
# import turtle library
import turtle
my_wn = turtle.Screen()
my_wn.bgcolor("light blue")
my_wn.title("Turtle")
my_pen = turtle.Turtle()
my_pen.color("black")
def my_sqfunc(size):
    for i in range(4):
        my_pen.fd(size)
        my_pen.left(90)
        size = size - 5
my_sqfunc(146)
my_sqfunc(126)
my_sqfunc(106)
my_sqfunc(86)
my_sqfunc(66)
my_sqfunc(46)
my_sqfunc(26)
```

Output

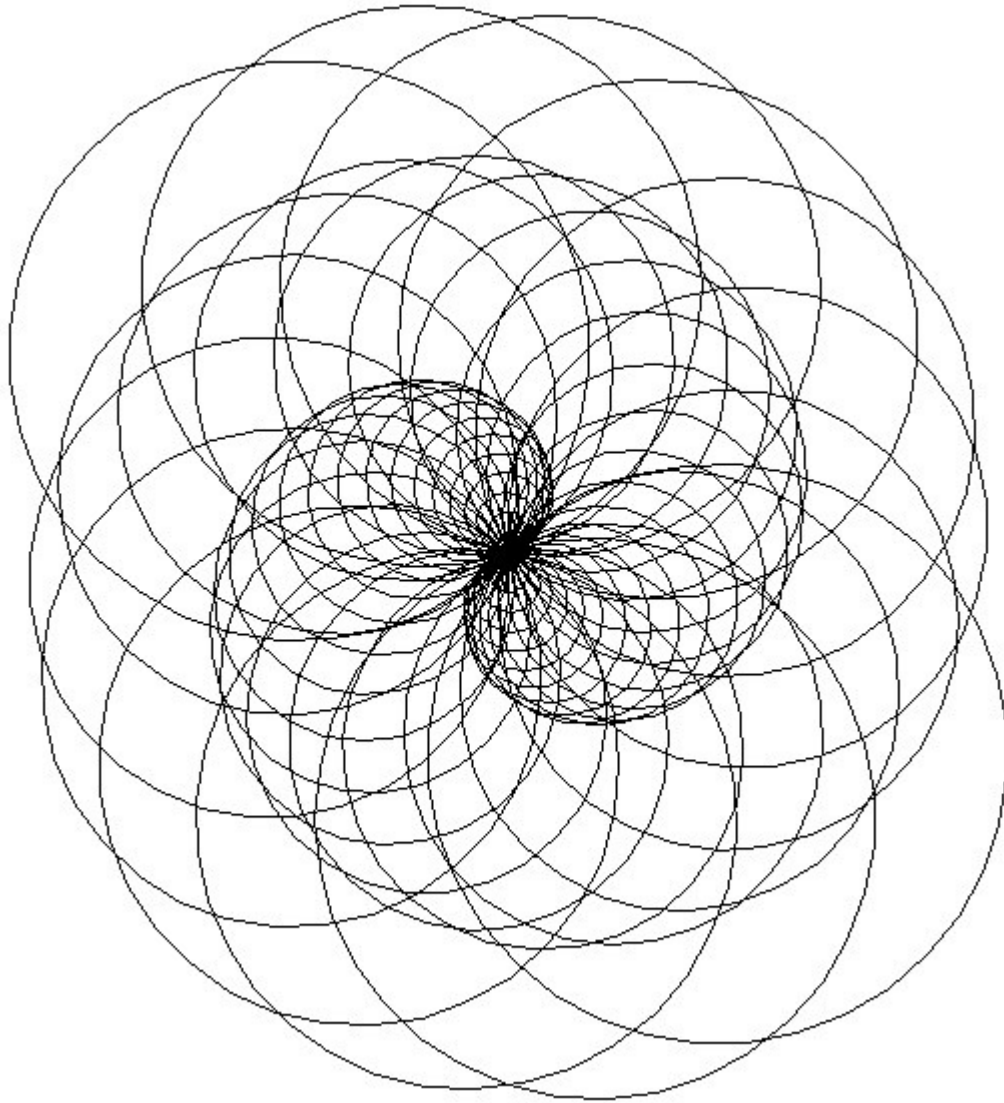


Drawing of another pattern

Example code

```
# import turtle library
import turtle
my_wn = turtle.Screen()
turtle.speed(2)
for i in range(30):
    turtle.circle(5*i)
    turtle.circle(-5*i)
    turtle.left(i)
turtle.exitonclick()
```

Output

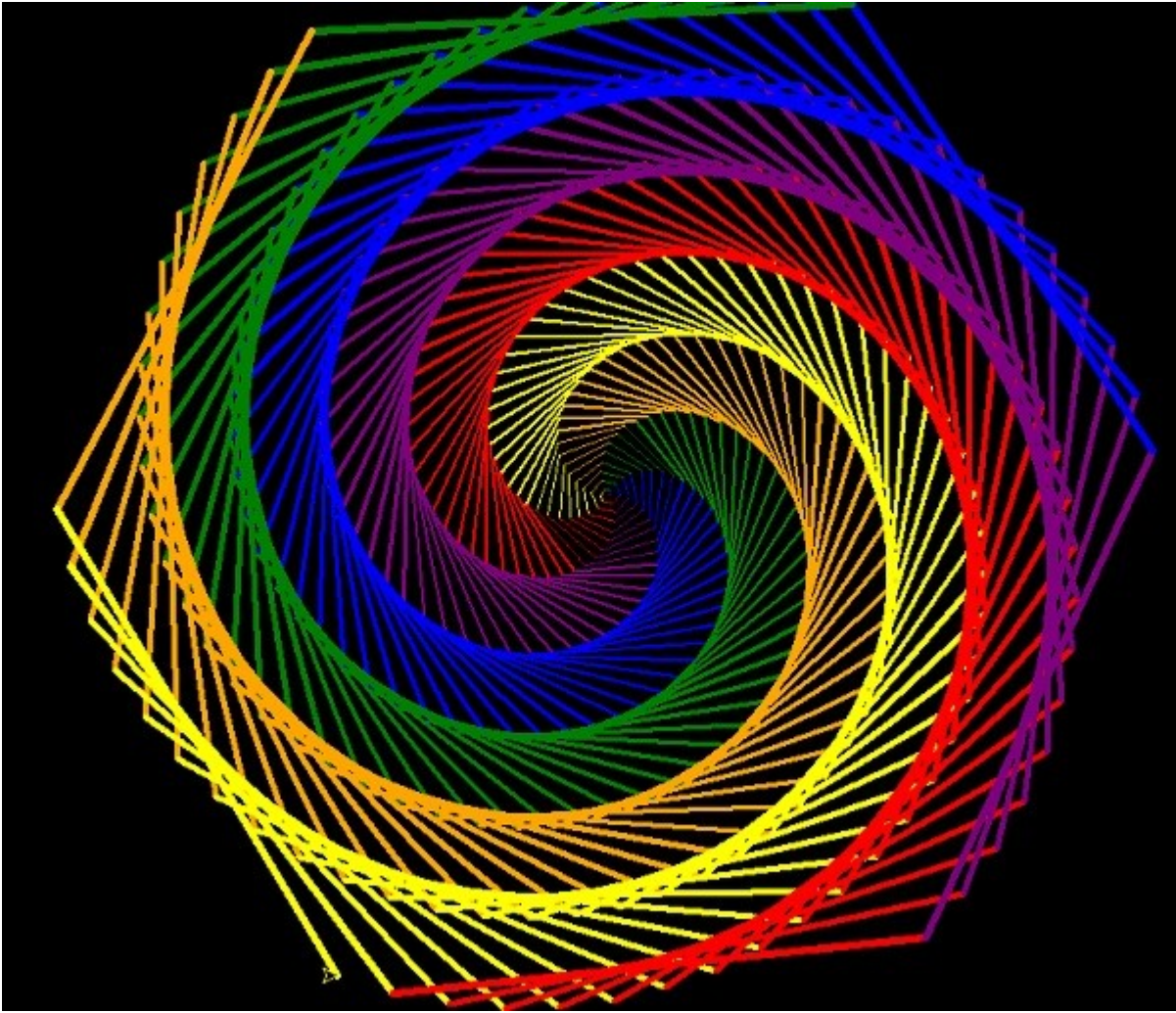


Drawing of another pattern

Example code

```
# import turtle library
import turtle
colors = [ "red","purple","blue","green","orange","yellow"]
my_pen = turtle.Pen()
turtle.bgcolor("black")
for x in range(360):
    my_pen.pencolor(colors[x % 6])
    my_pen.width(x/100 + 1)
    my_pen.forward(x)
    my_pen.left(59)
```

Output



Examples

Let's see some drawings using some simple commands.

Draw a Star

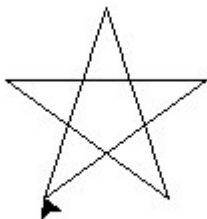
In the below program we draw a star. We choose appropriate steps to move the cursor forward and then right continuously to get this result.

Example

```
import turtle
star = turtle.Turtle()
for i in range(100):
    star.forward(100)
    star.right(144)
turtle.done()
```

Running the above code gives us the following result

Output



Draw Letter E

We follow a similar approach where the turtle moves in all four directions to create the English alphabet E.

Example

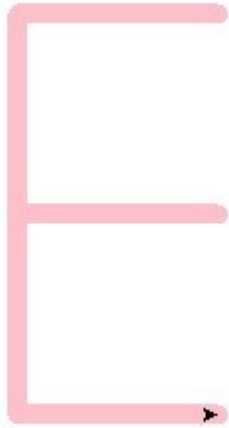
```
import turtle
t=turtle.Turtle()
t.penup()
t.setpos(-20,40)
t.pendown()
t.pensize(10)
t.pencolor("pink")
t.forward(100)
t.backward(100)
t.right(90)
t.forward(100)
t.left(90)
t.forward(100)
t.backward(100)
t.right(90)
t.forward(100)
t.left(90)
```



```
t.forward(100)
turtle.done()
```

Running the above code gives us the following result

Output



Multiple Squares

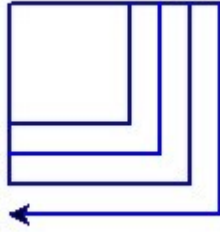
In the next example we see the drawing of multiple squares all starting from a common point. We use the usual simple commands to go forward, backward and then turn 90 degrees.

Example

```
import turtle
mult_square=turtle.Turtle()
def Multiple_Squares(length, colour):
    mult_square.pencolor(colour)
    mult_square.pensize(2)
    mult_square.forward(length)
    mult_square.right(90)
    mult_square.forward(length)
    mult_square.right(90)
    mult_square.forward(length)
    mult_square.right(90)
    mult_square.forward(length)
    mult_square.right(90)
    mult_square.setheading(360)
for i in range(60,120,15):
    Multiple_Squares(i,"blue")
turtle.done
```

Running the above code gives us the following result

Output



A spiral hexagon

This is a very interesting example where we use turtle to create a spiral structure. The final shape is a hexagon and there are various colours used in producing the sides of the hexagon.

Example

```
import turtle
colors = [ "pink","yellow","blue","green","white","red"]
sketch = turtle.Pen()
turtle.bgcolor("black")
for i in range(200):
    sketch.pencolor(colors[i % 6])
    sketch.width(i/100 + 1)
    sketch.forward(i)
    sketch.left(59)
```

Running the above code gives us the following result

