Python Graphics

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()
```

Multiple Squares

In the next example we see the drawing of multiple squares all starting from a common point. We sue 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
```

Problems

1.Load the appropriate image

2.Bring the turtle to the maze starting point

3.Figure out the commands to have your turtle find its way through the maze.

```
import turtle
turtle.showturtle()
turtle.shape("turtle")
# load the appropriate image
turtle.bgpic('maze1.png')
turtle.penup()
# bring the turtle to the starting point
turtle.goto(-70, 210)
turtle.pendown()
turtle.pencolor("red")
# write your code below
# press enter in the Terminal to exit the program
input()
```

Spiral

```
import turtle
length = 10
angle = 90
turtle.showturtle()
turtle.shape("turtle")
turtle.forward(length+length)
turtle.right(angle)
length = length + 10
turtle.forward(length+length)
turtle.right(angle)
length = length + 10
turtle.forward(length+length)
turtle.right(angle)
```

```
length = length + 10
turtle.forward(length+length)
turtle.right(angle)
length = length + 10
turtle.forward(length+length)
turtle.right(angle)
length = length + 10
turtle.forward(length+length)
turtle.right(angle)
length = length + 10
turtle.forward(length+length)
turtle.right(angle)
length = length + 10
turtle.forward(length+length)
turtle.right(angle)
length = length + 10
turtle.forward(length+length)
turtle.right(angle)
length = length + 10
turtle.forward(length+length)
turtle.right(angle)
length = length + 10
turtle.forward(length+length)
turtle.right(angle)
length = length + 10
turtle.forward(length+length)
turtle.right(angle)
length = length + 10
turtle.forward(length+length)
turtle.right(angle)
length = length + 10
```

```
turtle.forward(length+length)
turtle.right(angle)
length = length + 10
```

Zigzag

```
import turtle
length = 25
angle = 45
turtle.showturtle()
turtle.shape("turtle")
turtle.forward(length)
turtle.right(angle)
turtle.forward(length)
turtle.left(angle)
turtle.forward(length)
turtle.right(angle)
turtle.forward(length)
turtle.left(angle)
turtle.forward(length)
turtle.right(angle)
turtle.forward(length)
turtle.left(angle)
turtle.forward(length)
turtle.right(angle)
turtle.forward(length)
```

Loops

Loops are used when you have a block of code that you want to repeat.

A for loop is used when you have a block of code which you want to repeat a fixed number of times. The for loop iterates through the block of indented code.

Square

```
import turtle
turtle.showturtle()
turtle.shape("turtle")
for x in range(4):
    turtle.forward(50)
    turtle.right(90)
```

Spiral

```
import turtle
length = 10
angle = 90
turtle.showturtle()
turtle.shape("turtle")
for x in range(10):
    turtle.forward(length+length)
    turtle.right(angle)
    length = length+10
#Circle
import turtle
angle = 91
turtle.showturtle()
turtle.shape("turtle")
for x in range(100):
    turtle.circle(x)
    turtle.left(angle)
```

Triangle

```
import turtle
length = 100
angle = 120
turtle.showturtle()
turtle.shape("turtle")
for i in range(3):
    turtle.forward(length)
    turtle.right(angle)
```

Star1

```
import turtle
turtle.showturtle()
turtle.shape("turtle")
turtle.pencolor('green')
for x in range(13):
    turtle.forward(200)
    turtle.left(150)
```

Star2

```
import turtle
turtle.showturtle()
turtle.shape("turtle")
turtle.pencolor('purple')
for x in range(100):
    turtle.forward(200)
    turtle.left(175)
```

Spiral

```
import turtle
turtle.showturtle()
turtle.shape("turtle")
turtle.pencolor('pink')
for i in range(180):
   turtle.forward(100)
   turtle.right(30)
   turtle.forward(20)
   turtle.left(60)
   turtle.forward(50)
   turtle.right(30)
   turtle.penup()
   turtle.setposition(0, 0)
   turtle.pendown()
   turtle.right(2)
```

Drawing a square and a rectangle together :

As the heading suggests, we will be drawing a square and a rectangle in one single program altogether with the help of various functions of the turtle python library. The code for the following is given below :

```
Code :
# Python programme for drawing a square and a rectangle together in # Turtle - Python
import turtle
ttl = turtle.Turtle()

#SQUARE

for j in range(4):
  ttl.forward(60)
```

```
    ttl.left(90)
ttl.up()
ttl.goto(80,0)
ttl.down()

#RECTANGLE

ttl.forward(120)
ttl.left(90)
ttl.forward(80)
ttl.left(90)
ttl.forward(120)
ttl.left(90)
ttl.forward(80)
ttl.left(90)
```

Drawing Colour Filled Shapes in Turtle - Python

Python has a built-in module called turtle. It offers drawing with a cardboard screen and a turtle (pen). We must move the turtle in order to draw anything on the screen. There are other functions, such as forward() and reverse(), to move the turtle.

Turtle has three functions to fill the colours in the shapes it has drawn :

fillcolor() : This function aids in selecting the colour to fill the shape. It uses the colour name or hex value provided as an input parameter to fill the following closed topographical objects with the specified colour. Basic colour names include red, blue, green, and orange.The hex value of a colour is a string of hexadecimal numbers (beginning with "#"), such as #RGGRBG. The three hexadecimal numbers are R, G, and B.

begin_fill() : instructs Turtle that the selected colour must be used to fill any upcoming closed graphical objects.

End_fill() : instructs turtle to halt filling impending closed graphical objects.

#Python Program to draw a half-circle filled with colour in Turtle

```
import turtle
ttl = turtle.Turtle()

#Providing the colour to be filled
ttl.color("pink")

#Instructing to "begin" and "end" filling the "half-circle"
ttl.begin_fill()
ttl.circle(90,180)
ttl.end_fill()

#hiding the turtle after completing the drawing
ttl.hideturtle()
turtle.done()
```

#Python Program to draw a circle filled with colour in Turtle

```
import turtle
ttl = turtle.Turtle()

#Providing the colour to be filled
ttl.color("red")

#Instructing to "begin" and "end" filling the "circle"
ttl.begin_fill()
ttl.circle(90)
ttl.end_fill()

#hiding the turtle after completing the drawing
ttl.hideturtle()
turtle.done()
```

#Python Program to draw a square filled with colour in Turtle

```
import turtle
ttl = turtle.Turtle()

#Providing the color to be filled
ttl.color("yellow")

#Instructing to "begin" and "end" filling the "square"
ttl.begin_fill()

#for-loop to run 4 times to complete drawing each side of square
for j in range(4):
   ttl.forward(160)
   ttl.right(90)
ttl.end_fill()

#hiding the turtle after completing the drawing
ttl.hideturtle()
turtle.done()
```

#Python Program to draw a rectangle filled with colour in Turtle

```
import turtle
ttl = turtle.Turtle()

#Providing the color to be filled
ttl.color("green")

#Instructing to "begin" and "end" filling the "rectangle"
ttl.begin_fill()

#for-loop to run 2 times to complete drawing the sides of rectangle
```

```
for j in range(2):
    ttl.forward(160)
    ttl.right(90)
    ttl.forward(80)
    ttl.right(90)
ttl.end_fill()

#hiding the turtle after completing the drawing
ttl.hideturtle()
turtle.done()
```

#Python Program to draw a triangle filled with color in Turtle

```
import turtle
ttl = turtle.Turtle()

#Providing the color to be filled
ttl.color("blue")

#Instructing to "begin" and "end" filling the "triangle"
ttl.begin_fill()

#for-loop to run 3 times to complete drawing all sides of triangle
for j in range(3):
    ttl.forward(100)
    ttl.left(120)
ttl.end_fill()

#hiding the turtle after completing the drawing
ttl.hideturtle()
turtle.done()
```

#Python Program to draw a star filled with color in Turtle

```
import turtle
ttl = turtle.Turtle()


#Providing the color to be filled
ttl.color("magenta")


#Instructing to "begin" and "end" filling the "star"
ttl.begin_fill()


#for-loop to run 5 times to complete drawing a complete star
for j in range(5):
    ttl.forward(200)
    ttl.right(144)
ttl.end_fill()


#hiding the turtle after completing the drawing
ttl.hideturtle()
turtle.done()
```