

Simple Graphics using Turtle

```
In [ ]: #from turtle import Turtle
import turtle

t = turtle.Turtle()
turtle.done()
```

```
In [ ]: import turtle

t = turtle.Turtle()
t.width(2)          # For bolder lines
t.left(90)          # Turn to face north
t.forward(50)       # Draw a vertical line in black
turtle.done()
```

```
In [ ]: import turtle

t = turtle.Turtle()
t.left(90)          # Turn to face north
t.up()              # Prepare to move without drawing
t.forward(300)
turtle.done()
```

```
In [ ]: import turtle

t = turtle.Turtle()
t.setheading(0)     # Turn to face east
t.pencolor("red")   # make pen color red
t.down()            # Prepare to draw
t.forward(200)       # Draw a horizontal line in red
t.hideturtle()
turtle.done()
```

```
In [1]: import turtle

t = turtle.Turtle()

t.width(2)          # For bolder lines
t.left(90)          # Turn to face north
t.forward(50)       # Draw a vertical line in black
t.left(90)          # Turn to face west
t.up()              # Prepare to move without drawing
t.forward(10)       # Move to beginning of horizontal line
t.setheading(0)     # Turn to face east
t.pencolor("red")   # make pen color red
t.down()            # Prepare to draw
t.forward(20)       # Draw a horizontal line in red
t.hideturtle()      # Make the turtle invisible
turtle.done()
```

```
In [ ]: t.clear()
```

```
In [1]: #Draw a square
import turtle
t = turtle.Turtle()
t.goto(-200,-200)
t.forward(100)
t.left(90)
t.forward(100)
t.left(90)
t.forward(100)
t.left(90)
t.forward(100)
turtle.done()
```

```
In [1]: import turtle

t = turtle.Turtle()
def square(t, length):
    #Draws a square with the given length.
    for count in range(4):
        t.forward(length)
        t.left(90)

def hexagon(t, length):
    #Draws a hexagon with the given length.
    for count in range(6):
        t.forward(length)
        t.left(60)

def radialHexagons(t, n, length):
    #Draws a radial pattern of n hexagons with the given length.
    for count in range(n):
        hexagon(t, length)
        t.left(360 / n)

def radialpattern(t, n, length, shape):
    #Draws a radial pattern of n shapes with the given length.
    for count in range(n):
        shape(t, length)
        t.left(360 / n)

def main():

    t.pencolor("blue")
    t.hideturtle()
    square(t, 50)
    t.clear()

    # Embed a square in a hexagon
    hexagon(t, 50)
    t.clear() # Erase all drawings
    radialHexagons(t, 10, 50)
    t.clear()
    radialpattern(t,n=10,length = 50, shape = square)
    t.clear()
    radialpattern(t,n=10,length = 50, shape = hexagon)

main()
turtle.done()
```

```
In [1]: import turtle

t = turtle.Turtle()
def drawSquare(t, x, y, length):
    """Draws a square with the given turtle t, an upper-left
    corner point (x, y), and a side's length."""
    t.up()
    t.goto(x, y)
    t.setheading(90)
    t.down()
    for count in range(4):
        t.forward(length)
        t.left(90)

drawSquare(t,-200,-200,100)
turtle.done()
```

```
In [1]: import turtle

t = turtle.Turtle()
t.screen.bgcolor("orange")
x = t.screen.window_width() // 2
y = t.screen.window_height() // 2
print((-x, y), (x, -y))
turtle.done()
```

(-480, 405) (480, -405)

```
In [1]: """
File: circle.py
Project 7.1

Draws a circle.

1. The inputs are the coординates of the center point and the radius.

"""

import math
import turtle

t = turtle.Turtle()

def drawCircle(t, x, y, radius):
    """Draws a circle with the given center point and radius."""
    t.up()
    t.goto(x, y)
    t.setheading(90)
    t.down()
    for count in range(120):
        t.left(3)
        t.forward(2.0 * math.pi * radius / 120.0)

def main():
    """Allows the user to enter the center point and the radius."""
    #x = int(input("Enter the x coordinate of the center point: "))
    #y = int(input("Enter the y coordinate of the center point: "))
    #radius = int(input("Enter the radius: "))
    drawCircle(t, 10, 10, 50)

main()
turtle.done()
```

```
In [1]: import turtle

t = turtle.Turtle()
t.screen.bgcolor("yellow")
t.color("red")
t.circle(100)
turtle.done()
```

```
In [1]: import turtle

t = turtle.Turtle()
t.screen.bgcolor("yellow")
t.color("red")
t.pensize(10)
for angle in range(0,360,30):
    t.seth(angle)
    t.circle(100)
turtle.done()
```

```
In [1]: import turtle

t = turtle.Turtle()
t.color("red")
t.begin_fill()
t.circle(100)
t.end_fill()
t.hideturtle()
turtle.done()
```

```
In [1]: import turtle

t = turtle.Turtle()
t.color("red")
t.forward(100)
t.left(120)
t.forward(100)
t.left(120)
t.forward(100)
turtle.done()
```

```
In [ ]: import turtle

# Create a turtle object
my_turtle = turtle.Turtle()

# Move the turtle forward and turn right to draw the square
for i in range(4):
    my_turtle.forward(100)
    my_turtle.right(90)

# Close the turtle window on click
turtle.exitonclick()
```

```
In [1]: import turtle

# create turtle object
t = turtle.Turtle()

# draw star
for i in range(5):
    t.forward(200)
    t.right(144)

# hide turtle
t.hideturtle()

# keep window open until closed manually
turtle.done()
```

```
In [ ]: import turtle

# create turtle object
t = turtle.Turtle()

def star(t,length):
    for i in range(5):
        t.forward(length)
        t.right(144)
    t.hideturtle()

t.color("red")

star(t,200)

# keep window open until closed manually
turtle.done()
```

```
In [ ]:
```