Chapter 18 — Turtle Graphics

Introduction

In the early days of computing, a researcher named Grey Walter created a couple of simple robots that he commanded by telling them to move forward, backward, and to turn left and right. This idea was first used to teach children to draw simple graphics in the LOGO language.[11] This concept has been added to many languages over the years and was included in the early days of Python.

You can still think of the "turtle" as a little robot that drags a pen to draw lines as it moves. The turtle in Python can also adjust the width of the pen, draw with over 16 million different colors, and do many more things. This chapter will be a brief introduction to the turtle object. More can be found in the python documentation.

Objectives

Upon completion of this chapter's exercises, you should be able to:

- Describe the turtle's coordinate system.
- Draw polygons using basic turtle motion commands.
- Produce diagrams using colors, line width, and fill.
- Change the turtle's position to an absolute position on the drawing surface.
- Display text on the turtle's drawing surface.

Prerequisites

The material in this chapter only depends on material from Chapters 1-8. With slight modifications to the sample programs, only chapters 1-5 would be required.

Basic Turtle Motion

To use the turtle the first thing you must add to your program is the statement `import turtle`. This loads the turtle module.

turtle	Module
The turtle module allows you to draw images in a window. You will create turtles	

[11] https://social.technet.microsoft.com/wiki/contents/articles/29933.small-basic-the-history-of-the-logo-turtle.aspx

and will command them like little drawing robots. https://docs.python.org/3/library/turtle.html	

`variable = turtle.Turtle()`	Method of turtle
Creates a turtle object that will follow your commands. You may name your turtle by saving it into a. variable. Your program may have more than one turtle drawing on the screen. https://docs.python.org/3/library/turtle.html#turtle.Turtle	

`turtle.forward(distance_expr)`
`turtle.fd(distance_expr)`
`turtle.backward(distance_expr)`
`turtle.back(distance_expr)`
`turtle.bk(distance_expr)`
	Method of turtle

Move the turtle forward or backwards the specified distance based on the turtle's current direction. Distance may be an integer or floating-point number and can even be less than zero to move the turtle the opposite direction. https://docs.python.org/3/library/turtle.html#turtle.forward	

`turtle.right(angle_expr)`
`turtle.rt(angle_expr)`
`turtle.left(angle_expr)`
`turtle.lt(angle_expr)`
	Method of turtle

Rotate the turtle to the right or to the left by the angle specified. Angles may be integer or floating-point numbers and may be positive or negative numbers. The default turtle understands the angle in degrees (0-360). https://docs.python.org/3/library/turtle.html#turtle.right	

Now that we know the statements and methods to create a turtle and to make it move, you can see the turtle in action.

```
# a square with "moxy" the turtle
import turtle

moxy = turtle.Turtle()
```

```

for side in range(4):
    moxy.forward(10)
    moxy.right(90)
```

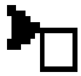


Illustration 23: The Turtle Draws a Square

Raising the Pen and Going Home

turtle.pendown()
turtle.pd()
turtle.down()
turtle.penup()
turtle.pu()
turtle.up()
	Method of turtle

Raise or lower the turtle's pen. When the pen is down a line will be drawn and when the pen is up the turtle will move without leaving a mark. https://docs.python.org/3/library/turtle.html#turtle.pendown	

turtle.home()	Method of turtle
Send the turtle back home. Usually this is the center if the canvas at position (0,0). If the pen is down, the path will be drawn. https://docs.python.org/3/library/turtle.html#turtle.home	

```
import turtle

moxy = turtle.Turtle()

l = 10
for side in range(40):
    if side % 3 ==0 :
        moxy.penup()
    else:
        moxy.pendown()
    moxy.forward(l)
    l = l + 3
    moxy.right(90)

moxy.pendown()
moxy.home()
```

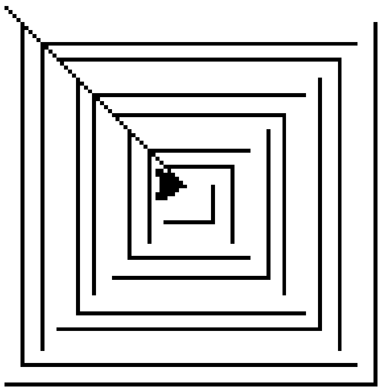


Illustration 24: The Turtle Draws an Odd Spiral

Colors and Line Thickness

We can tell the turtle what color and how wide if a line to draw.

```
turtle.color(color_expr)
turtle.color(r_expr, g_expr, b_expr)
turtle.color( ( r_color_expr, g_expr, b_expr ) )
```

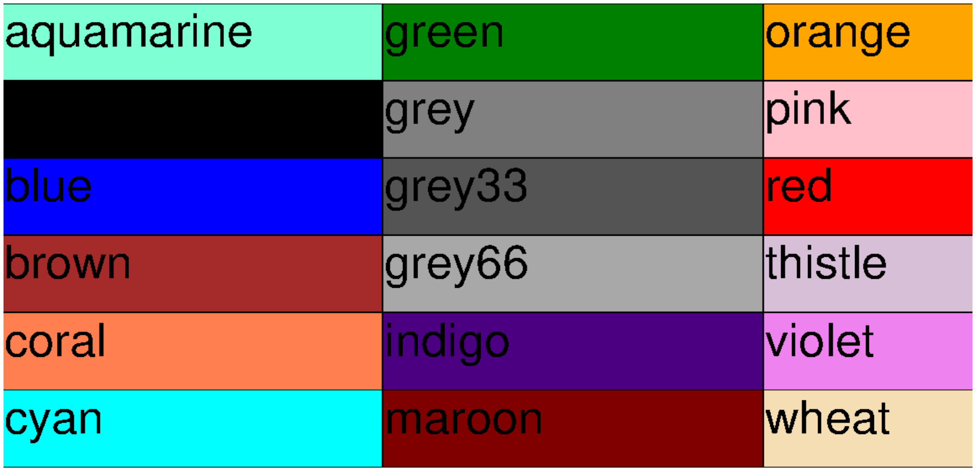
Method of turtle

Sets the turtle's color to the named color or the mix of red, green and blue. Using the color statement with only one color will set the turtle's pen and fill color to the same color. We will see an example of fill in the next section.

Python has hundreds of named colors, but they can be different depending on the system. Simple color names should always be available.

You may also specify a color numerically by mixing red, blue and green, The default range is 0 to 1 where 0 is the lack of the color, and 1 is full intensity. The RGB values may be passes as three values, or in a tuple.

https://docs.python.org/3/library/turtle.html#turtle.color
http://effbot.org/tkinterbook/tkinter-widget-styling.htm



aquamarine	green	orange
	grey	pink
blue	grey33	red
brown	grey66	thistle
coral	indigo	violet
cyan	maroon	wheat

Illustration 25: Common Turtle Color Names

```
turtle.pensize(width_expr)
turtle.width(width_expr)
```

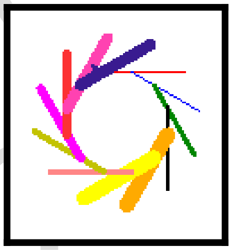
Method of turtle

Sets the width, in pixels, of the turtle's pen.

https://docs.python.org/3/library/turtle.html#turtle.pensize

```
import turtle
import random

fred = turtle.Turtle()

def line(ttl):
    ttl.forward(60)
    ttl.penup()
    ttl.backward(40)
    ttl.pendown()
    ttl.right(30)

fred.color("red")
line(fred)

fred.color("blue")
line(fred)

fred.color("green")
fred.pensize(2)
line(fred)

fred.color("black")
line(fred)

fred.color("orange")
fred.pensize(10)
line(fred)

fred.color("yellow")
line(fred)

fred.color(1.0, .5, .5)
fred.pensize(3)
line(fred)

fred.color(.75, .75, 0)
line(fred)

fred.color(1.0, 0, 1.0)
fred.pensize(5)
line(fred)
```

```

fred.color(random.random(), random.random(), random.random())
line(fred)

fred.color( ( 1.0, .235, .678) )
fred.pensize(7)
line(fred)

fred.color( ( .20, random.random(), .543) )
line(fred)
```



Drawing Shapes and Filling Them

`turtle.color(pen_color_expr, fill_color_expr)`
`turtle.pencolor(pen_color_expr)`
`turtle.fillcolor(fill_color_expr)`
	Method of turtle

The color method with two colors will set the pen color (line) and the fill color at the same time. The pen color and the fill color may be set separately. https://docs.python.org/3/library/turtle.html#turtle.color	

`turtle.begin_fill()`
`turtle.end_fill()`
	Method of turtle

If you wish to fill a shape, you must first `begin_fill` before you start the turtle drawing. Once the shape is closed, you may issue a `end_fill` and the enclosed space will be filled with the current fill color. https://docs.python.org/3/library/turtle.html#turtle.filling	

```
import turtle

sam = turtle.Turtle()
sam.color("blue","red")
sam.pensize(3)
sam.begin_fill()
for t in range(5):
    sam.forward(50)
    sam.right(180 - 180/5)
sam.end_fill()
```

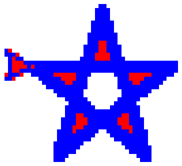


Illustration 26: Turtle Draws a Filled Star

Absolutely Moving the Turtle and Drawing Text

turtle.**goto(***x*,*y***)** *turtle*.**goto((** *x*,*y* **))** *turtle*.**setpos(***point or x*,*y***)** *turtle*.**setposition(***point or x*,*y***)**	Method of turtle
Send the turtle to a specific location on the screen. If the turtle's pen is down, a line to that point will be drawn. https://docs.python.org/3/library/turtle.html#turtle.goto	

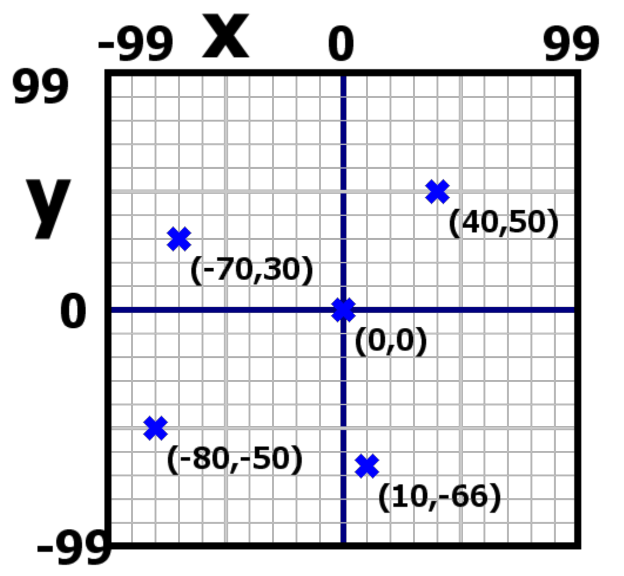
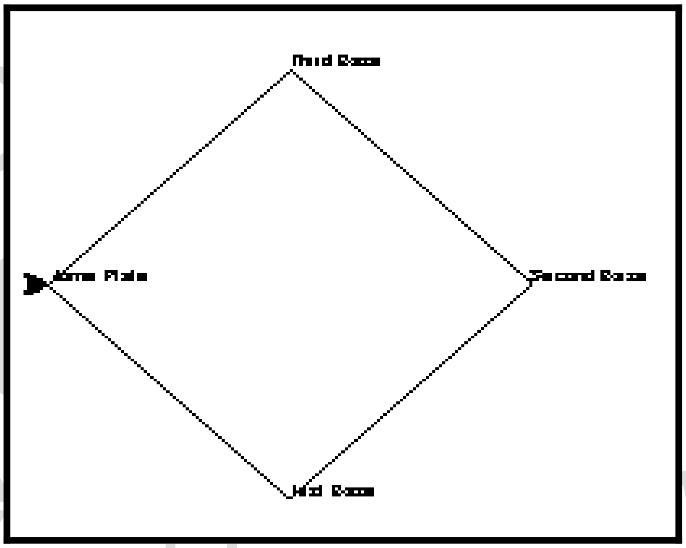


Illustration 27: Turtle Coordinates

turtle.**write**(*text*)	Method of turtle
Draw text at the turtle's current location. There are additional options to change the font and positioning that can be found in the documentation. https://docs.python.org/3/library/turtle.html#turtle.write	

```
import turtle

tom = turtle.Turtle()

# lift pen and go to home plate
tom.penup()
tom.goto(-100,0)
tom.pendown()
tom.write("Home Plate")
# go to first
tom.goto(0,-100)
tom.write("First Base")
# go to second
tom.goto(100,0)
tom.write("Second Base")
# go to third
tom.goto(0,100)
tom.write("Third Base")
# go back to home
tom.goto(-100,0)
```



Summary

Goes here

Important Terms

- backward
- begin_fill
- color
- coordinate
- degrees
- end_fill
- fillcolor
- forward
- goto
- home
- left
- pencolor
- pendown
- pensize
- penup
- right
- setposition
- turtle
- width
- write

Exercises

Here

Word Search

```
e h o m e f c e w o w r i t e r
c c o b t o o _ t c p s r r x i
s c o l o r o o o f l w g p m g
e e d n p w r e d n e f q e o h
t a a x e a d b e c f d a n e t
p g w n n r i a p e t e s s h o
o c i e c d n c d o e n f i a o
s z d w o p a k e d r d t z t e
i t t l l e t w g p k _ u e a r
t g h l o n e a r l e f r e v p
i o l n r d w r e t r i t e e e
o t j t d o f d e l f l l e q n
n o g q t w h c s l n l e k m u
o b e g i n _ f i l l w b r d p
u e e a o o f i l l c o l o r f
l c e _ l p m r l b n o s h c d
```

backward, begin_fill, color, coordinate, degrees, end_fill, fillcolor, forward, goto, home, left, pencolor, pendown, pensize, penup, right, setposition, turtle, width, write

References