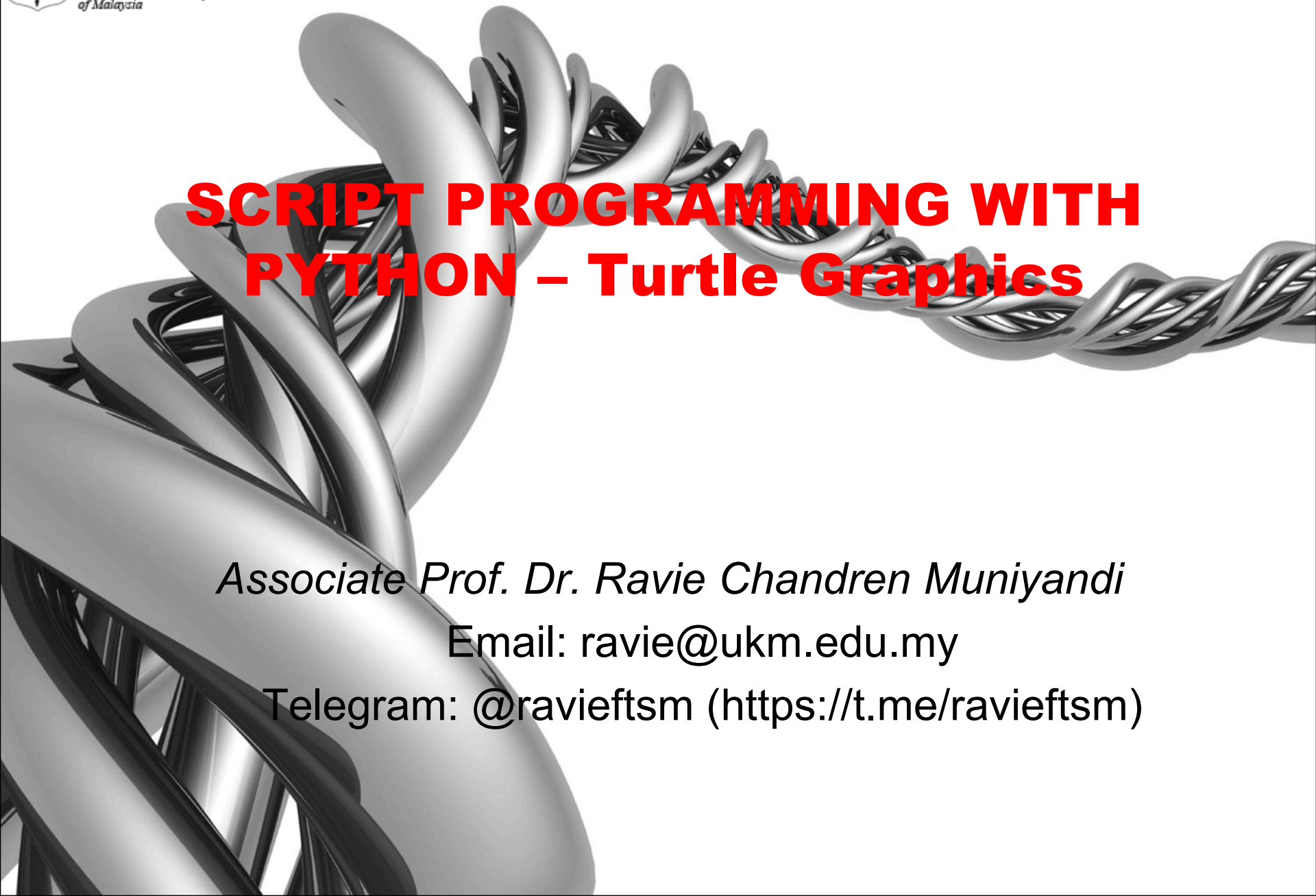


SCRIPT PROGRAMMING WITH PYTHON – Turtle Graphics

Associate Prof. Dr. Ravie Chandren Muniyandi
Email: ravie@ukm.edu.my
Telegram: @ravieftsm (https://t.me/ravieftsm)

Turtle Graphics

Coordinates

- Objects and methods from the module named "turtle."
- Given statements, window appears

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

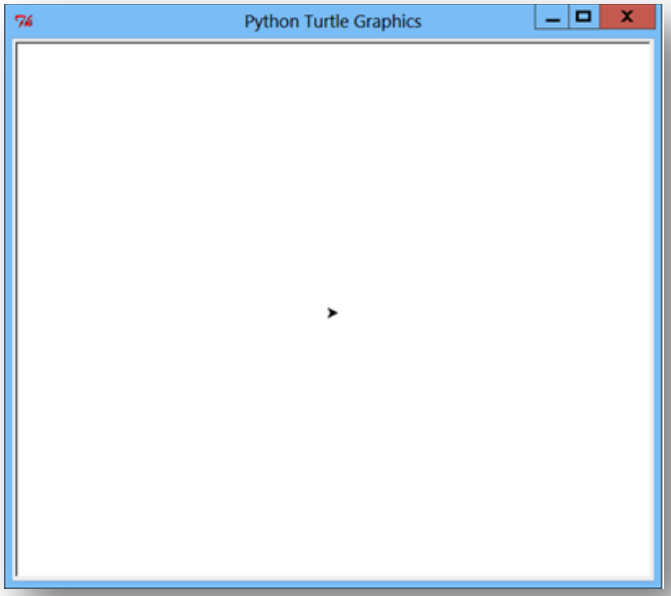


FIGURE 6.12 Turtle graphics window.

Coordinates

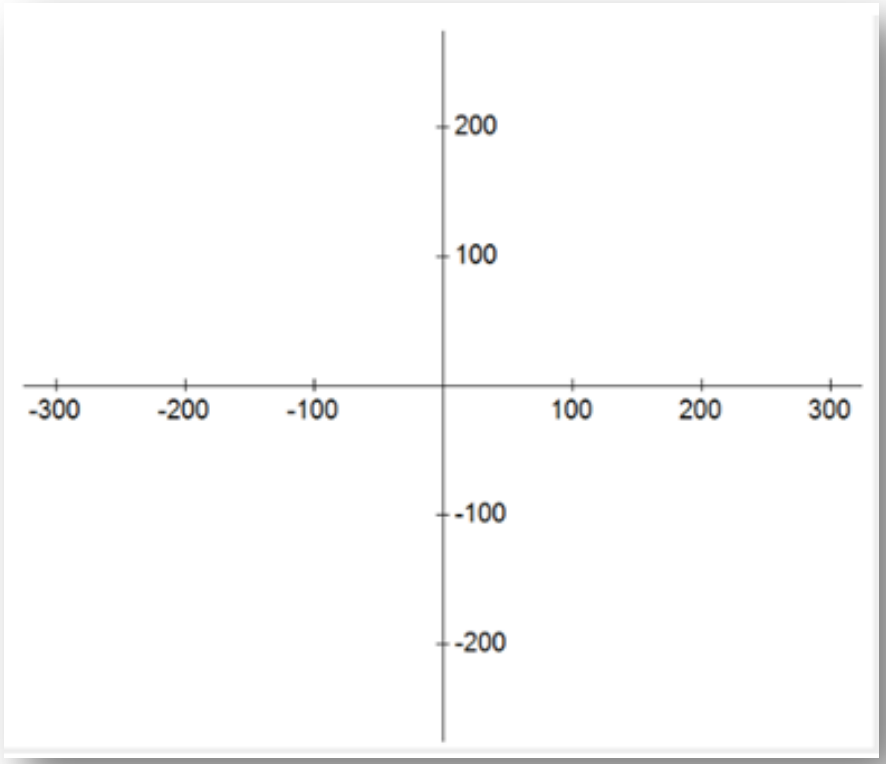
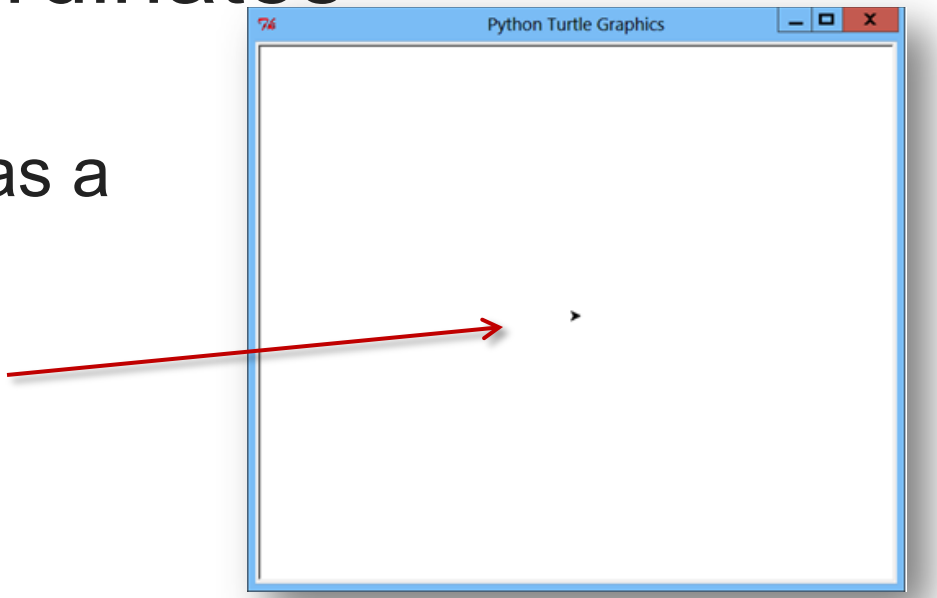


FIGURE 6.13 Coordinate system for canvas.

Coordinates



- Think of the chevron as a small turtle with a pen attached to its tail
- Python statements
  - Raise, lower “tail”
  - Change color
  - Move turtle in variety of ways

Functions from the *turtle* Module

- Functions provided
  - t.up( ), t.down( )
  - t.hideturtle( )
  - t.forward(distance), t.backward(distance)
  - t.goto(x,y)
  - t.pencolor(colorName)
  - t.setheading(deg)
  - t.left(deg), t.right(deg)
  - t.dot(diameter, colorName)

Rectangles

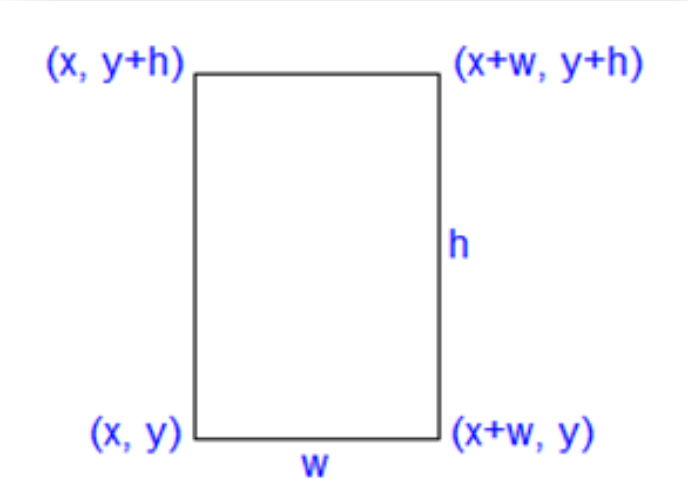


FIGURE 6.14 A general rectangle.

Rectangles

- Two possible functions to draw rectangles

```
import turtle

def drawRectangle(t, x, y, w, h, colorP="black"):
    ## Draw a rectangle with bottom-left corner (x, y),
    ## width w, height h, pencolor colorP.
    t.pencolor(colorP)
    t.up()
    t.goto(x , y)          # start at bottom-left corner of rectangle
    t.down()
    t.goto(x + w, y)       # draw line to bottom-right corner
    t.goto(x + w, y + h)   # draw line to top-right corner
    t.goto(x, y + h)       # draw line to top-left corner
    t.goto(x , y)          # draw line to bottom-left corner
```

Rectangles

- Two possible functions to draw rectangles

```
def drawRectangle2(t, x, y, w, h, colorP="black"):
    ## Draw a rectangle with bottom-left corner (x, y),
    ## width w, height h, pencolor colorP.
    t.pencolor(colorP)
    t.up()
    t.goto(x , y)          # start at bottom-left corner of rectangle
    t.down()
    for i in range(2):
        t.forward(w)       # draw horizontal side of rectangle
        t.left(90)         # rotate 90 degrees counterclockwise
        t.forward(h)       # draw vertical side of rectangle
```

Rectangles

- Example 1: Program draws a rectangle having a red border and a yellow interior.

```
def main():
    t = turtle.Turtle()
    t.hideturtle()
    drawRectangle(t, 0, 0, 100, 150, "red", "yellow")

def drawRectangle(t, x, y, w, h, colorP="black", colorF="white"):
    ## Draw a rectangle with bottom-left corner (x, y),
    ## width w, height h, pen color colorP, and fill color colorF.
    t.pencolor(colorP)
    t.fillcolor(colorF)
    t.up()
    t.goto(x , y)          # start at bottom-left corner of rectangle
```

Rectangles

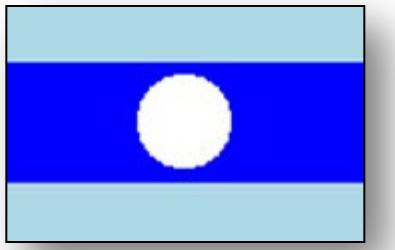
- Example 1:

```
    t.goto(x , y)           # start at bottom-left corner of rectangle
    t.down()
    t.begin_fill()
    t.goto(x + w, y)        # draw line to bottom-right corner
    t.goto(x + w, y + h)    # draw line to top-right corner
    t.goto(x, y + h)        # draw line to top-left corner
    t.goto(x , y)           # draw line to bottom-left corner
    t.end_fill()

main()
```

Flags

- Example 2: program draws flag shown on the right.



```
import turtle

def main():
    t = turtle.Turtle()
    t.hideturtle()
    # Draw the three stripes.
    drawFilledRectangle(t, 0, 0, 150, 25, "light blue", "light blue")
    drawFilledRectangle(t, 0, 25, 150 , 50, "blue", "blue")
    drawFilledRectangle(t, 0, 75, 150, 25, "light blue", "light blue")
    # Draw white dot. Center of flag is (75, 50). 40 = .8 * 50.
    drawDot(t, 75, 50, 40, "white")

def drawFilledRectangle(t, x, y, w, h, colorP="black", colorF="white"):
```

Flags

- Example 2:

```
def drawFilledRectangle(t, x, y, w, h, colorP="black", colorF="white"):
    ## Draw a filled rectangle with bottom-left corner (x, y),
    ## width w, height h, pen color colorP, and fill color colorF.
    t.pencolor(colorP)
    t.fillcolor(colorF)
    t.up()
    t.goto(x, y)          # bottom-left corner of rectangle
    t.down()
    t.begin_fill()
    t.goto(x + w, y)      # bottom-right corner of rectangle
    t.goto(x + w, y + h)  # top-right corner of rectangle
    t.goto(x, y + h)      # top-left corner of rectangle
```

Flags

- Example 2:

```
        t.begin_fill()
        t.goto(x + w, y)        # bottom-right corner of rectangle
        t.goto(x + w, y + h)    # top-right corner of rectangle
        t.goto(x, y + h)        # top-left corner of rectangle
        t.goto(x , y)           # bottom-left corner of rectangle
        t.end_fill()

    def drawDot(t, x, y, diameter, colorP):
        ## Draw a dot with center (x, y) and color colorP.
        t.up()
        t.goto(x, y)
        t.pencolor(colorP)
        t.dot(diameter)

    main()
```

Flags

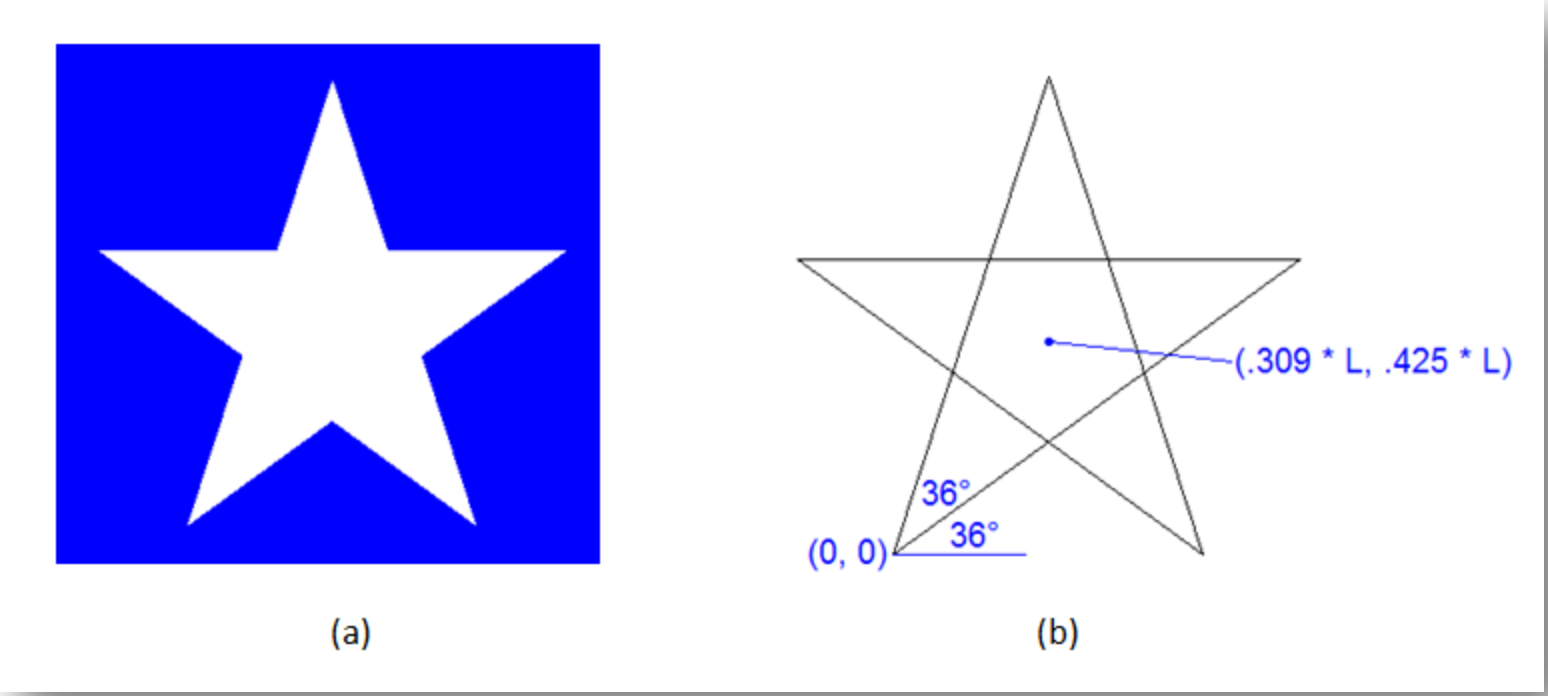


FIGURE 6.16 Five-Pointed Star

Flags

- Example 3: program draws the five-pointed star in Fig. 6.16 (b).

```
import turtle

def main():
    t = turtle.Turtle()
    t.hideturtle()
    lengthOfSide = 200
    drawFivePointStar(t, 0, 0, lengthOfSide)

def drawFivePointStar(t, x, y, lengthOfSide):
    # Drawing begins at (x, y) and moves in a north-east direction.
    t.up()
    t.goto(x, y)
    t.left(36)
    t.down()
    for i in range(5):
        t.forward(lengthOfSide)
        t.left(144)    # 144 = 180 - 36
main()
```

The *write* Method

- Given *s*, a string as argument in

```
t.write(s)
```

- Displays the string *s*
  - Bottom left corner of the string approximately at current position of the pen
- Other options

```
t.write(s, align="right")
```

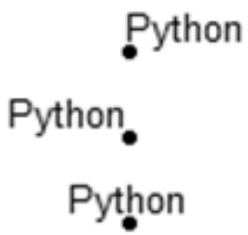
```
t.write(s, align="center")
```

The *write* Method

- Example 4: Program displays the word *Python* with different alignments.

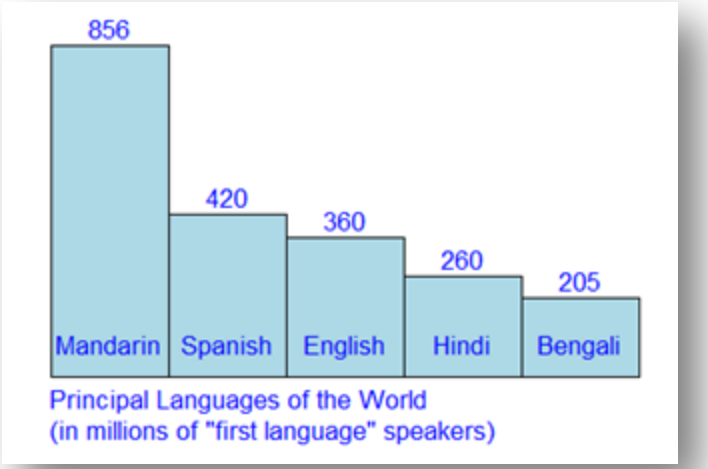
```
import turtle
t = turtle.Turtle()
t.hideturtle()
t.up()
t.goto(0, 60)
t.dot()
t.write("Python")
t.goto(0, 30)
t.dot()
t.write("Python", align="right")
t.goto(0, 0)
t.dot()
t.write("Python", align="center")
```

[Run]



Bar Charts

- Example 5: Program creates bar chart on the right.




```
import turtle

heights = [856, 420, 360, 260, 205]  # number of speakers for each language

def main():
    t = turtle.Turtle()
    t.hideturtle()
    for i in range(5):
        drawRectangle(t, -200 + (76 * i), 0, 76, heights[i] / 4,
                      "black", "light blue")
    displayText(t)

def drawRectangle(t, x, y, w, h, colorP="black", colorF="white"):
```

Bar Charts

- Example 5:

```
    displayText(t)

def drawRectangle(t, x, y, w, h, colorP="black", colorF="white"):
    ## Draw a rectangle with bottom-left corner (x, y), width w,
    ## height h, pen color colorP, and fill color colorF.
    t.pencolor(colorP)
    t.fillcolor(colorF)
    t.up()
    t.goto(x , y)          # bottom-left corner of rectangle
    t.down()
    t.begin_fill()
    t.goto(x + w, y)       # bottom-right corner of rectangle
    t.goto(x + w, y + h)   # top-right corner of rectangle
    t.goto(x, y + h)       # top-left corner of rectangle
    t.goto(x , y)          # bottom-left corner of rectangle
    t.end_fill()

def displayText(t):
    languages = ["Mandarin", "Spanish", "English",
```

Bar Charts

- Example 5:

```
        t.goto(x, y)            # bottom-left corner of rectangle
        t.end_fill()

    def displayText(t):
        languages = ["Mandarin", "Spanish", "English",
                     "Hindi", "Bengali"]
        t.pencolor("blue")
        t.up()
        for i in range(5):
            # Display number at top of rectangle.
            t.goto(-162 + (76 * i), heights[i] / 4)
            t.write(str(heights[i]), align="center",
                    font=("Ariel", 10, "normal"))
            # Display language.
            t.goto(-162 + (76 * i), 10)
```

Bar Charts

- Example 5:

```
            t.write(str(heights[i]), align="center",
                    font=("Ariel", 10, "normal"))
            # Display language.
            t.goto(-162 + (76 * i), 10)
            t.write(languages[i], align="center",
                    font=("Ariel", 10, "normal"))
        # Display title of bar chart.
        t.goto(-200, -25)
        t.write("Principal Languages of the World",
                font=("Ariel", 10, "normal"))
        t.goto(-200, -45)
        t.write('(in millions of "first language" speakers)',
                font=("Ariel", 10, "normal"))

main()
```

Line Charts

- Tabular data can be visually displayed it in a line chart.

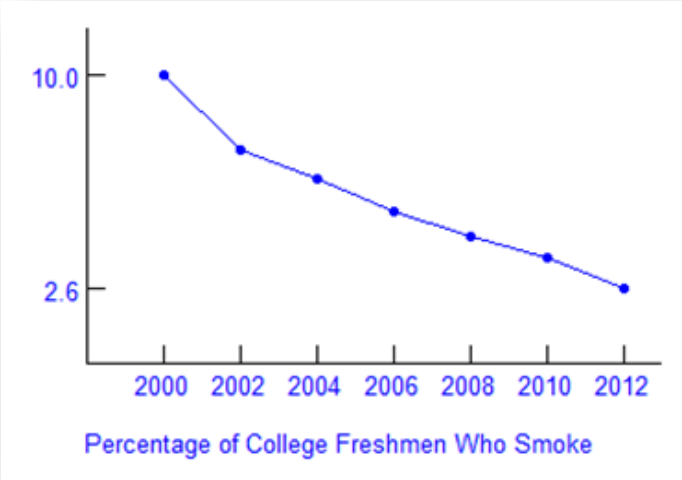
	2000	2002	2004	2006	2008	2010	2012
Percent	10.0	7.4	6.4	5.3	4.4	3.7	2.6

Source: Higher Education Research Institute.

Table 6.3 Percentage of college freshmen who smoke.

Line Charts

- Example 6: Program uses data in Table 6.3 to create the line chart on the right




```
import turtle

yValues = [10.0, 7.4, 6.4, 5.3, 4.4, 3.7, 2.6]  # percent for each year

def main():
    t = turtle.Turtle()
    t.hideturtle()
    drawLine(t, 0, 0, 300, 0)       # Draw x-axis.
    drawLine(t, 0, 0, 0, 175)       # Draw y-axis.
```

Line Charts

- Example 6:

```
        drawLine(t, 0, 0, 300, 0)       # Draw x-axis.
        drawLine(t, 0, 0, 0, 175)       # Draw y-axis.
        for i in range(6):
            drawLineWithDots(t, 40 + (40 * i), 15 * yValues[i],
                             40 + (40 * (i + 1)), 15 * yValues[i + 1], "blue")
        drawTickMarks(t)
        displayText(t)

    def drawLine(t, x1, y1, x2, y2, colorP="black"):
        ## Draw line segment from (x1, y1) to (x2, y2) having color colorP.
        t.up()
        t.goto(x1, y1)
        t.down()
        t.pencolor(colorP)
        t.goto(x2, y2)

    def drawLineWithDots(t, x1, y1, x2, y2, colorP="black"):
```

Line Charts

- Example 6:

```
    t.goto(x2, y2)

def drawLineWithDots(t, x1, y1, x2, y2, colorP="black"):
    ## Draw line segment from (x1, y1) to (x2, y2) having color
    ## colorP and insert dots at both ends of the line segment.
    t.pencolor(colorP)
    t.up()
    t.goto(x1, y1)  # beginning of line segment
    t.dot(5)
    t.down()
    t.goto(x2, y2)  # end of line segment
    t.dot(5)

def drawTickMarks(t):
```

Line Charts

- Example 6:

```
def drawTickMarks(t):
    ## Draw tick marks along x-axis.
    for i in range(1, 8):
        drawLine(t, 40 * i, 0, 40 * i , 10)
    # Draw tick mark on y-axis to indicate greatest value.
    drawLine(t, 0, 15 * max(yValues), 10, 15 * max(yValues))
    # Draw tick mark on y-axis to indicate least value.
    drawLine(t, 0, 15 * min(yValues), 10, 15 * min(yValues))

def displayText(t):
    t.pencolor("blue")
    t.up()
    # Display greatest y-value next to upper tick mark on y-axis.
    t.goto(-3, (15 * max(yValues)) - 10)
    t.write(max(yValues), align="right")
    # Display least y-value next to lower tick mark on y-axis.
    t.goto(-3, (15 * min(yValues)) - 10)
```

Line Charts

- Example 6:

```
        t.write(max(yValues), align="right")
        # Display least y-value next to lower tick mark on y-axis.
        t.goto(-3, (15 * min(yValues)) - 10)
        t.write(min(yValues), align="right")
        # Display the years below the tick marks on x-axis.
        x = 40
        for i in range(2000, 2013, 2):
            t.goto(x, -20)
            t.write(str(i), align="center")
            x += 40
        # Display title of graph.
        t.goto(0, -50)
        t.write("Percentage of College Freshmen Who Smoke")

main()
```


Thank you