**Outline**

Develop a better understanding of procedural sequencing by solving shape drawing challenges using the turtle environment.

**Objectives**

* Use correct terminology to describe programming concepts;
* Describe the types of data that computers can process and store (e.g., numbers, text);
* Explain the difference between constants and variables used in programming;
* Use variables, expressions, and assignment statements to store and manipulate numbers and text in a program

**Materials**

* Python Turtle Development Environment at: https://repl.it/
* PythonWorksheetII form the GitHub Repository
* Web links identified in the questions below

**Level 1: Drawing Basic Shapes With Python Turtle**

1. Open the document PythonWorksheetII from the class GItHub repository.   
   Read over “Part III” at the end of the PythonWorksheetII document.
2. Create an new Repl by selecting the “Python with Turtle” language / environment.
3. Begin all of your turtle programs with the following code to create a “pen”:

import turtle

myPen = turtle.Turtle()

1. Create a program to draw a red circle.
   1. Provide a listing of your program code below:

import turtle

myPen = turtle.Turtle()

myPen.color("red")

myPen.circle(90)

1. Create a program to draw any three of the shapes described in “Part III” of   
   the PythonWorksheetII document.
   1. Provide a listing of your program code below:
2. import turtle

myPen = turtle.Turtle()

myPen.color("green")

myPen.circle(21)

1. import turtle

myPen = turtle.Turtle()

myPen.color("green")

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.left(90)

1. import turtle

myPen = turtle.Turtle()

myPen.color("green")

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.right(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.right(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.right(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.right(90)

**Level 2: Using a Loop**

1. Google the keywords “Python Turtle Methods”.
   1. Explain how the “goto” method works and how you could use it when drawing repeated shapes.

Goto works with drawing repeated shapes by moving the turtle to position for X,Y.

* 1. List some other useful methods not listed in “Part III” at the end of the PythonWorksheetII document.
* Method: Heading Parameters: none Description: Returns the current heading
* Method: Position Parameters: none Description: Returns the current position

1. Create a repeating pattern on your screen. The pattern must meet the following requirements:
   1. The basic pattern must be made up of several individual Turtle methods (e.g. changes of colour, changes of direction, size, motion, etc.)
   2. The basic pattern must be repeated several times with a shift in starting position each time.
2. import turtle

myPen = turtle.Turtle()

myPen.color("green")

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.color("blue")

myPen.left(89)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.color("purple")

myPen.left(87)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.color("yellow")

myPen.left(85)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.color("blue")

myPen.left(83)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.color("blue")

myPen.left(81)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.color("blue")

myPen.left()

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.color("blue")

myPen.left(89)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

1. Use a Python Loop to create your repeating pattern
   1. The Loop may be a Counted Loop or a Conditional Loop
   2. The indented block of code for the loop should be your basic pattern.
2. Provide a listing of your repeating pattern loop below.

import turtle

myPen = turtle.Turtle()

myPen.speed(100000000000000000000)

for hi in "1234567":

myPen.color("green")

myPen.circle(60)

myPen.forward(100)

myPen.right(90)

myPen.forward(100)

myPen.right(90)

myPen.forward(100)

myPen.forward(100)

myPen.right(90)

myPen.forward(100)

myPen.right(90)

myPen.forward(100)

myPen.up()

myPen.forward(200)

myPen.down()

**Level 3: Defining a Function**

1. Google the keywords “Python Function Syntax”.
   1. Explain what the “def” keyword does

Keyword def marks the start of **function** header. This means that def keyword makes the function header.

* 1. Explain any special rules regarding the function name

The use of colon (:) is to end or stop the function header.

A colon (:) to mark the end of **function** header. This means the colon make the function header stop or end the sentence.

* 1. Explain what the parameters (or arguments) do

Parameter or arguments

Parameters (arguments) through which we pass values to a function. They are optional

* 1. Where should the colon “:” be placed

The colon (:) is placed in the end of a function to stop or end the function to complete it.

These colons will the function easier to do.

* 1. Explain how to write Python statements that make up the function body

One or more valid python statements that make up the function. Statements must have same indentation level (usually 4 space)

* 1. Explain the “return” statement

1. Provide an example of a simple function that uses one or more parameters.
   1. Write the function definition below
   2. Write some code to call the function below
2. Convert your basic pattern (from Level 2 above) into a function
3. The function name should be “my\_pattern”
4. The parameters should be the x and y starting position for your pattern
5. Your function does not need to use the “return” statement
6. Provide a listing of your function definition and repeating pattern loop below.