**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:

import turtle

myPen = turtle.Turtle()

myPen.color("red")

myPen.circle (90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.forward(90)

myPen.left(90)

myPen.forward(90)

myPen.forward(90)

import turtle

myPen = turtle.Turtle()

myPen.shape("arrow")

**Level 2: Using a Loop**

1. Google the keywords “Python Turtle Methods”.

Explain how the “goto” method works and how you could use it when drawing repeated shapes.

Is a statement found in many computer programming languages. It performs a one-way transfer of control to another line of code.

* 1. List some other useful methods not listed in “Part III” at the end of the PythonWorksheetII document.

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.

import turtle

myPen = turtle.Turtle()

myPen.color("blue")

myPen.circle(50)

for i in [0,1,2,3,4,5,6,7,8,10]:

myPen.circle(50)

myPen.forward(60)

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.circle(50)

for i in [0,1,2,3,4,5,6,7,8,10]:

myPen.circle(50)

**Level 3: Defining a Function**

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

It Defines the name of the variable that is going to be used to repeat the code. The value to be returned is given by the expression in the return statement.

* 1. Explain any special rules regarding the function name

Used to perform a single, related action.

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

A parameter is a variable in a method definition. When a method is called, the arguments are the data you pass into the method's parameters. Parameter is variable in the declaration of function. Argument is the actual value of this variable that gets passed to function.

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

When defining your function in the beginning of the program. To show that we are going to use this value in our expressions.

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

First, use the keyword def to declare the function and follow this up with the function name. Second, add parameters to the function: they should be within the parentheses of the function. Then, add statements that the functions should execute.

* 1. Explain the “return” statement

The return statement is used when a function is ready to return a value to its caller.

1. Provide an example of a simple function that uses one or more parameters.
   1. Write the function definition below

Functions are used to utilize code in more than one place in a program. A function in Python is defined by a def statement.

* 1. Write some code to call the function below

1. Convert your basic pattern (from Level 2 above) into a function
2. The function name should be “my\_pattern”
3. The parameters should be the x and y starting position for your pattern
4. Your function does not need to use the “return” statement

for p in range(0,+11):

yFrom=-10+p

xTo=p

1. Use a your basic pattern function and a Python Loop to create your repeating pattern
   1. The Loop may be a Counted Loop or a Conditional Loop
   2. Your function should be called from within the loop.
2. Provide a listing of your function definition and repeating pattern loop below.

import turtle

myPen = turtle.Turtle()

def Drawsquare(sidesize):

myPen.forward(sidesize)

myPen.left(90)

myPen.forward(sidesize)

myPen.left(90)

myPen.forward(sidesize)

myPen.left(90)

myPen.forward(sidesize)

Drawsquare(100)

Drawsquare(90)

Drawsquare(80)

Drawsquare(70)

Drawsquare(50)

Drawsquare(40)

Drawsquare(30)

Drawsquare(20)

Drawsquare(10)

Drawsquare(120)