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()
   ```

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

```
myPen.color("red")
myPen.circle(100)
```

5. Create a program to draw any three of the shapes described in "Part III" of the PythonWorksheetII document.
   a. Provide a listing of your program code below:

   Drawing a square:

```
myPen.forward(150)
myPen.left(90)
```

```
myPen.forward(150)
myPen.left(90)
myPen.forward(150)
myPen.left(90)
myPen.forward(150)
```

b)

```
myPen.color("red")
myPen.forward(50)
myPen.right(90)
myPen.forward(50)
myPen.left(90)
myPen.forward(25)
myPen.left(90)
myPen.forward(50)
myPen.right(90)
myPen.forward(50)
myPen.left(90)
myPen.forward(25)
myPen.left(90)
myPen.forward(50)
myPen.right(90)
myPen.forward(50)
myPen.left(90)
myPen.forward(25)
myPen.left(90)
myPen.forward(50)
myPen.right(90)
myPen.forward(50)
myPen.left(90)
myPen.forward(25)
```

c)

```
myPen.color("red")
myPen.forward(100)
myPen.left(90)
myPen.forward(100)
myPen.left(90)
myPen.forward(100)
myPen.left(90)
myPen.forward(100)
myPen.left(90)
myPen.forward(50)
myPen.color("blue")
myPen.circle(50)
```

d

```
myPen.up()
myPen.goto(100,50)
```

```
myPen.down()
myPen.left(45)
myPen.forward(100)
myPen.up()
myPen.goto(175,50)
myPen.down()
myPen.left(90)
myPen.forward(100)
```

Level 2: Using a Loop

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

Moves the turtle from the current position to the location x, y along the shortest path between the two locations (i.e. a direct line between the current position and (x,y). It can be used in a loop to move and repeat shapes.

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

Loops can be used to repeat these commands any number of times.

2. Create a repeating pattern on your screen. The pattern must meet the following requirements:
   a. The basic pattern must be made up of several individual Turtle methods (e.g. changes of colour, changes of direction, size, motion, etc.)
   b. The basic pattern must be repeated several times with a shift in starting position each time
3. Use a Python Loop to create your repeating pattern
   a. The Loop may be a Counted Loop or a Conditional Loop
   b. The indented block of code for the loop should be your basic pattern.
4. Provide a listing of your repeating pattern loop below.

```
for hello in [1,2,3,4]:
    myPen.up()
    myPen.goto(x,y)
    myPen.color("red")
    myPen.width(20)
    myPen.down()
    myPen.forward(150)
    myPen.left(90)
    myPen.color("blue")
    myPen.width(10)
    myPen.forward(150)
    myPen.goto(x+=1,y+=1)
```

Level 3: Defining a Function

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

Makes the start of a function header.

   b. Explain any special rules regarding the function name

- Can be any length
- Can be any combination of letters in lowercase, or uppercase, or digits, or an underscore,
- Cannot start with a digit
- Cannot be special symbols like ! @#%
- Keywords cannot be used as identifiers

c. Explain what the parameters (or arguments) do

"The term parameter (sometimes called formal parameter) is often used to refer to the variable as found in the function definition, while argument (sometimes called actual parameter) refers to the actual input supplied at function call." (Wikipedia).

"Parameter (Computer Programming)." *Wikipedia*, Wikimedia Foundation, 4 Nov. 2018, en.m.wikipedia.org/wiki/Parameter_(computer_programming).

d. Where should the colon ":" be placed

At the end of a function.

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

Define the python function's name follow by parenthesis with the parameters inside. Then put a colon, followed by whichever statements need to be executed when the function is called.

f. Explain the "return" statement

Return causes the function to stop executing and hand a value back to whatever called it.

2. Provide an example of a simple function that uses one or more parameters.
   a. Write the function definition below
   b. Write some code to call the function below

```
def DrawSquare(square,lemon):
    for size in [1,2,3,4]:
        myPen.forward(square)
        myPen.left(lemon)
DrawSquare(50,90)
```

3. Convert your basic pattern (from Level 2 above) into a function
4. The function name should be "my_pattern"
5. The parameters should be the x and y starting position for your pattern
6. Your function does not need to use the "return" statement

```
def my_pattern(x,y):
    for hello in [1,2,3,4]:
        myPen.up()
        myPen.goto(x,y)
        myPen.color("red")
        myPen.width(20)
        myPen.down()
        myPen.forward(150)
```

```
        myPen.left(90)
        myPen.color("blue")
        myPen.width(10)
        myPen.forward(150)
        myPen.goto(x+1,y+1)

my_pattern(20,20)
```

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

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

def myfunction(apples,oranges):
    myPen.up()
    myPen.goto(apples,oranges)
    myPen.down()
    myPen.circle(20)

for makingcircles in [1,2,3,4,5]:
    myfunction(20,100*makingcircles)
```