PALAWAN STATE UNIVERSITY
College of Sciences

CC2/L
Computer Programming I

Module 3

Python Modules & Turtles

Table of Contents

Content **Page**

Learning Objectives

After going through in this module, you should be able to:

- ✓ Express that Python is extensible through modules and turtles
- ✓ Import an entire module
- ✓ Construct programs using modules
- ✓ Construct multi-line program using the turtle framework
- ✓ Use methods and set attributes using dot notation
- ✓ Recognize the use of For Loop to draw common geometric shapes with the turtle

Overview

OVERVIEW HEADING (optional)

Introduction to Python Modules

A module is a file consisting of Python code.

Sample Python Code in Module

hello.py file:

```
def greetings(message):
    print("Hello" + message)
```

This code contains set of functions that you want to include in your application. It can be in the form of variables, functions or even class.

Remember that the filename becomes the module name.

Introduction to Python Modules

How to create a Module & Import it to another file?

To create a module, just save the code you want in a file with the file extension .py
(link to video)

Now we will create a module and import it into another file.

Step 1: create a python file: (hello.py)

hello.py file:

```
def greetings(message):
    print("Hello" + message)
```

Step 2: create a function to display python file for your hello.py

display_message.py file:

```
import hello

hello.greetings(" Welcome to Python Module")
```

Step 3
Import the file from function of **hello.py** using **IMPORT** keyword

Step 4: Call the greetings() from hello.py

Step 5: Run the display_message.py file

Output:

```
>>Hello Welcome to Python Module
```

Import in Module

Earlier, we have seen a simple module with a function. Now, let us define the import keyword. To use the module, you have to import it using the **import** keyword.

The import statement combines two operations; it searches for the named module, then binds the search results to a name in the local scope.

Syntax for importing more than 1 module:

```
import module1[, module2, ... moduleN]
```

Import in Module

Another way of importing when you only want to import some of the functionality from math module from math import pow.

Sample code in Importing math module

```
import math

print(math.pow(3,6))
print(math.sqrt(25))

for x in "test":
 print(x)
```

This code invokes function **pow** that is defined in module math.

module.function() e.g. math.pow(), math.sqrt()

Output

```
>>
729.0
5.0
t
e
s
t
```

Practice 01

Problem:

Write a module program that will compute for two variables.

Get the sum, difference, product, and quotient of the variables.

Create another module program that will call each function from the first module to compute variables.

Display output.

Solution 01

1. Create operation.py file

```
def add(x,y):
   print("sum ", x+y )

def diff(x, y):
       print("difference ", x - y
)

def product(x,y):
   print("product is ", x*y)

def quo(x,y):
   print("quotient is ", x/y)
```

2. Create display.py file

```
import operation

operation.add(4,3)
operation.diff(2,1)
operation.product(3,10)
operation.quo(5,2)
```

3. Output

```
>>
sum  7
difference  1
product is  30
quotient is  2.5
```

The Random Module

Python has many built-in modules that you can use. The random module is one of them.

Random module can be used to random number, selecting a random elements from a list, shuffle elements randomly, etc.

To generate a random floats, a method can be used. The random.random() method returns a float number between 0.0 to 1.0. This function does not need any arguments.

```
import random

digit = random.random()
print(digit)

>>30929486828694197
```

```
import random

digit = random.uniform(1,10)
print(digit)

>>7.701997218085037
```

random.uniform(1,10) will print any numbers with decimal from 1-10.

The Random Module

Python has many built-in modules that you can use. The random module is one of them.

Random module can be used to random number, selecting a random elements from a list, shuffle elements randomly, etc.

To generate a random floats, a method can be used. The random.random() method returns a float number between 0.0 to 1.0. This function does not need any arguments.

```
import random

digit = random.random()
print(digit)

>>30929486828694197
```

```
import random

digit = random.uniform(1,10)
print(digit)

>>7.701997218085037
```

random.uniform(1,10) will print any numbers with decimal from 1-10.

Random() methods

Some of the random() methods used:

Function	Description
seed()	Values produced will be deterministic, meaning, when the seed number is the same, the same sequence of values will be generated
randrange()	This can return random values between the specified limit and interval
randint()	This returns a random integer between the given limit
choice()	This returns a random number from a sequence
shuffle()	This randomly reorders the elements in a list
sample()	This returns randomly selected items from a sequence
uniform()	This returns floating point values between the given range

Random() methods

The following are sample fragment codes using **random methods**:

```
import random
random.choice('computer')
#output
>>> t

import random
random.choice([12,23,45,67,65,43])
#output
>>>45

import random
print(random.randint(2,9))
#output
>>>3

import random
for x in range(5):
   print(random.randrange(2,10))
#output
>>>
3
6
5
7
5
```

Sample Program

Task:

Solution:

Learning Check

INSERT LEARNING ACTIVITY HEADING (optional)

The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

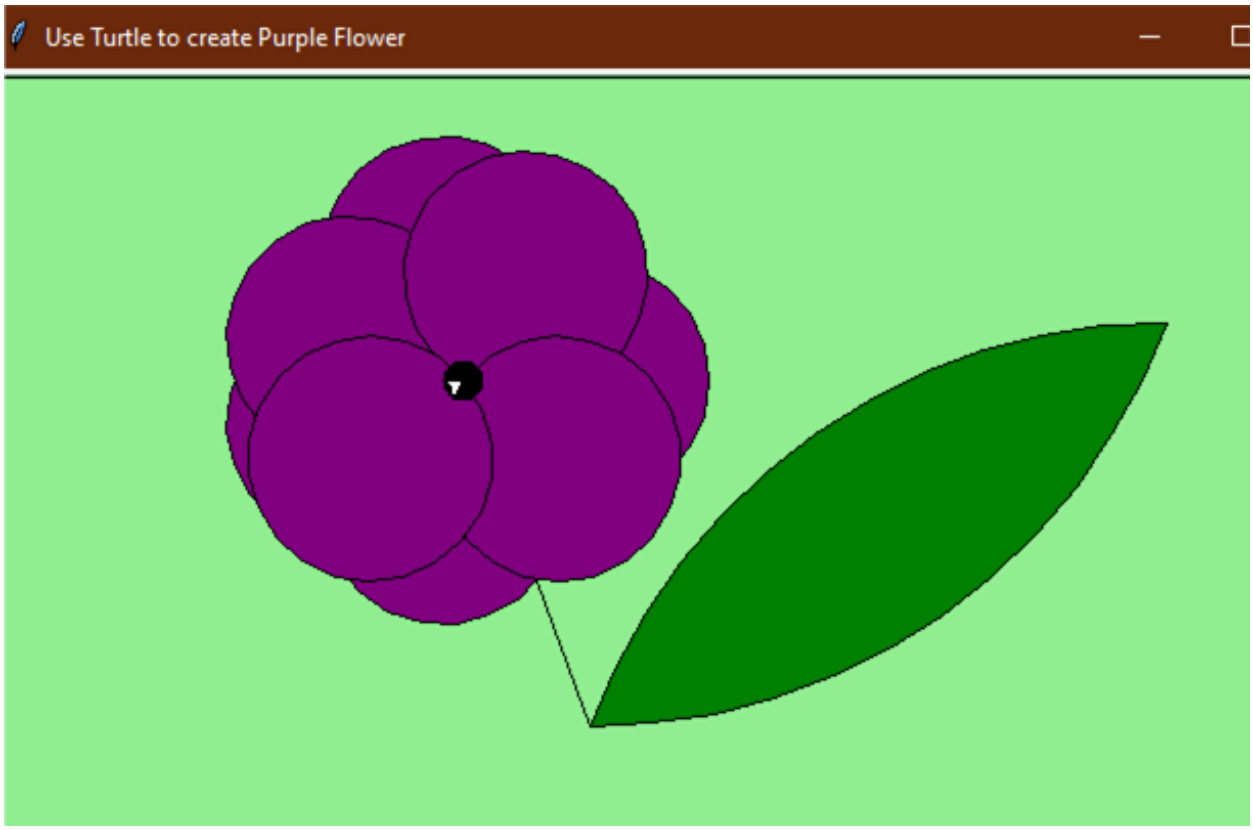
The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

Python Turtle

There are many modules in Python that provide compelling features. In this part of the module, you will learn how to create a turtle data object that can be used to draw pictures.

Turtle is a Python library that allows users to create designs, draw shapes and make pictures or images.

Sample Output: Turtle window



Discussion

Since Python uses objects, these objects have specific attributes and methods. Coding the python attributes and methods is used by placing a dot (.) written between them.

For example, if the dog is a class, then a dog named Rocky would be its instance/object.

```
class Dog:
Rocky = Dog( )
```

METHOD	DESCRIPTION
Turtle()	It creates and returns a new turtle object
forward()	It moves the turtle forward by the specified amount
backward()	It moves the turtle backward by the specified amount
right()	It turns the turtle clockwise
left()	It turns the turtle counter clockwise
penup()	It picks up the turtle's Pen
pendown()	Puts down the turtle's Pen
up()	Picks up the turtle's Pen
down()	Puts down the turtle's Pen
color()	Changes the color of the turtle's pen
fillcolor()	Changes the color of the turtle will use to fill a polygon
heading()	It returns the current heading
position()	It returns the current position
goto()	It moves the turtle to position x,y
begin_fill()	Remember the starting point for a filled polygon
end_fill()	It closes the polygon and fills with the current fill color
dot()	Leaves the dot at the current position
stamp()	Leaves an impression of a turtle shape at the current location
shape()	Should be 'arrow', 'classic', 'turtle' or 'circle

Discussion

Since Python uses objects, these objects have specific attributes and methods. Coding the python attributes and methods is used by placing a dot (.) written between them.

Learning Check

INSERT LEARNING ACTIVITY HEADING (optional)

The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

Evaluation

INSERT EVALUATION ACTIVITY HEADING (optional)

The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

Rubrics

COPY AND PASTE YOUR RUBRICS HERE

Reflection

INSERT DISCUSSION HEADING (optional)

The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

References

- *Python Reference*. (n.d.). https://www.w3schools.com/python/python_reference.asp
- Romano, F., & Kruger, H. (2021). *Learn Python Programming: An in-depth introduction to the fundamentals of Python, 3rd Edition* (3rd ed.). Packt Publishing.
- Rossum, G. V., & Team, P. D. (2020). *Learning Python: Crash Course Tutorial*. Medina Univ PR Intl.
- Sande, W., & Sande, C. (2019). *Hello World!: A complete Python-based computer programming tutorial with fun illustrations, examples, and hand-on exercises.* (3rd ed.). Manning.
- Lerner, R. M. (2020). *Python Workout: 50 ten-minute exercises* (1st ed.). Manning.
- *An Intro to Git and GitHub for Beginners (Tutorial)*. (n.d.). https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners
- *An Intro to Git and GitHub for Beginners (Tutorial)*. (n.d.). https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners