K. J. Somaiya Polytechnic, Mumbai-77

Batch No: C1

Enrollment No: FCOW19118

Experiment No.: 03

Experiment Name: Develop a program using built-in libraries.

Experiment No: 03

Experiment Name: Develop a program using built-in libraries.

Course Outcome:

O18RA72.1	**Understand basics of Python programming.**

Theory:

Built-in libraries in Python

1. NumPy

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python.

Why Use NumPy?

In Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy. Arrays are very frequently used in data science, where speed and resources are very important.

Why is NumPy Faster Than Lists?

NumPy arrays are stored at one continuous place in memory unlike lists, so processes can access and manipulate them very efficiently. This behavior is called locality of reference in computer science.This is the main reason why NumPy is faster than lists. Also it is optimized to work with latest CPU architectures.

Which Language is NumPy written in?

NumPy is a Python library and is written partially in Python, but most of the parts that require fast computation are written in C or C++.

2. Random

You can generate random numbers in Python by using random module. Python offers random module that can generate random numbers. These are pseudo-random number as the sequence of number generated depends on the seed. If the seeding value is same, the sequence will be the same. For example, if you use 2 as the seeding value, you will always see the following sequence.

import random random.seed(2) print(random.random()) print(random.random()) print(random.random())	0.9560342718892494 0.9478274870593494 0.05655136772680869

Python defines a set of functions that are used to generate or manipulate random numbers through the random module. Functions in the random module rely on a pseudo-random number generator function random(), which generates a random float number between 0.0 and 1.0. This particular type of functions is used in a lot of games, lotteries, or any application requiring a random number generation.

Random Number Operations

a. choice() : choice() is an inbuilt function in the Python programming language that returns a random item from a list, tuple, or string.

b. randrange(beg, end, step) : The random module offers a function that can generate random numbers from a specified range and also allowing rooms for steps to be included, called randrange().

c. random() : This method is used to generate a float random number less than 1 and greater or equal to 0.

d. seed() : Seed function is used to save the state of a random function so that it can generate some random numbers on multiple executions of the code on the same machine or on different machines (for a specific seed value). The seed value is the previous value number generated by the generator. For the first time when there is no previous value, it uses current system time.

e. shuffle() : It is used to shuffle a sequence (list). Shuffling means changing the position of the elements of the sequence. Here, the shuffling operation is in place.

3. Math

Python has a built-in module that you can use for mathematical tasks. The math module has a set of methods and constants.

Some of the Math Methods

Method	Description
math.acos()	Returns the arc cosine of a number
math.acosh()	Returns the inverse hyperbolic cosine of a number
math.asin()	Returns the arc sine of a number
math.asinh()	Returns the inverse hyperbolic sine of a number
math.atan()	Returns the arc tangent of a number in radians
math.atan2()	Returns the arc tangent of y/x in radians
math.atanh()	Returns the inverse hyperbolic tangent of a number
math.ceil()	Rounds a number up to the nearest integer
math.comb()	Returns the number of ways to choose k items from n items without repetition and order

Math Constants

Constant	Description
math.e	Returns Euler's number (2.7182...)
math.inf	Returns a floating-point positive infinity
math.nan	Returns a floating-point NaN (Not a Number) value
math.pi	Returns PI (3.1415...)
math.tau	Returns tau (6.2831...)

Some of the most popular mathematical functions are defined in the math module. These include trigonometric functions, representation functions, logarithmic functions, angle conversion functions, etc. In addition, two mathematical constants are also defined in this module.

Pie (π) is a well-known mathematical constant, which is defined as the ratio of the circumference to the diameter of a circle and its value is 3.141592653589793.

Another well-known mathematical constant defined in the math module is e. It is called Euler's number and it is a base of the natural logarithm. Its value is 2.718281828459045.

Code	Output
>>> import math >>>math.pi	3.141592653589793
>>>math.e	2.718281828459045

The math module contains functions for calculating various trigonometric ratios for a given angle. The functions (sin, cos, tan, etc.) need the angle in radians as an argument. We, on the other hand, are used to express the angle in degrees. The math module presents two angle conversion functions: degrees() and radians(), to convert the angle from degrees to radians and vice versa.

The math.log() method returns the natural logarithm of a given number. The natural logarithm is calculated to the base e.

The math.log10() method returns the base-10 logarithm of the given number. It is called the standard logarithm.

The math.exp() method returns a float number after raising e (math.e) to given number. In other words, exp(x) gives e**x. This can be verified by the exponent operator.

The math.pow() method receives two float arguments, raises the first to the second and returns the result. In other words, pow(4,4) is equivalent to 4**4.

The math.sqrt() method returns the square root of a given number.

The following two functions are called representation functions. The ***ceil()*** function approximates the given number to the smallest integer, greater than or equal to the given floating point number. The ***floor()*** function returns the largest integer less than or equal to the given number.

Implementation and Output

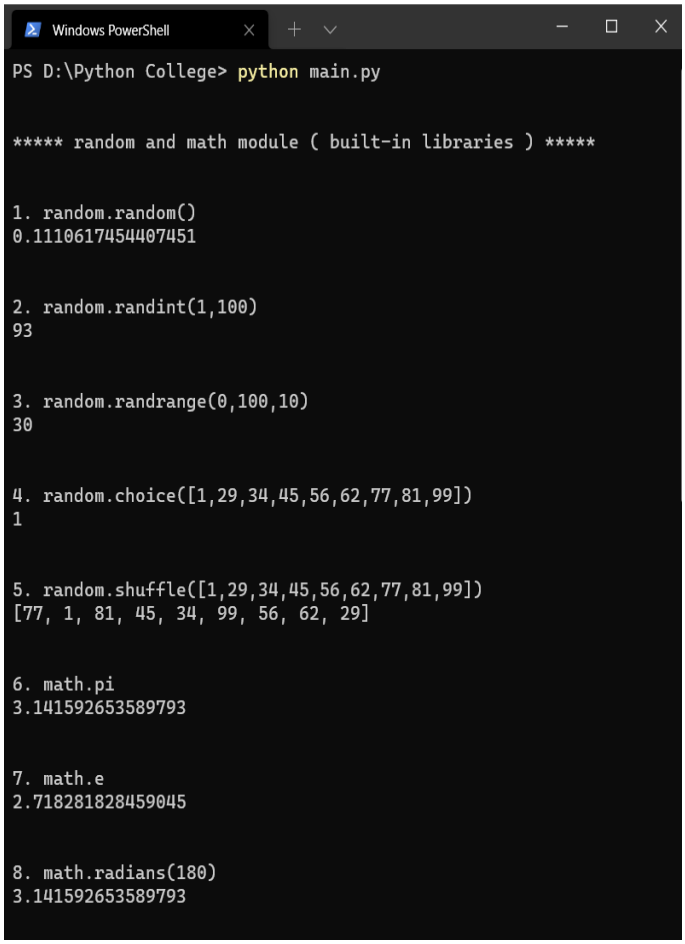
1. Built-in libraries in Python

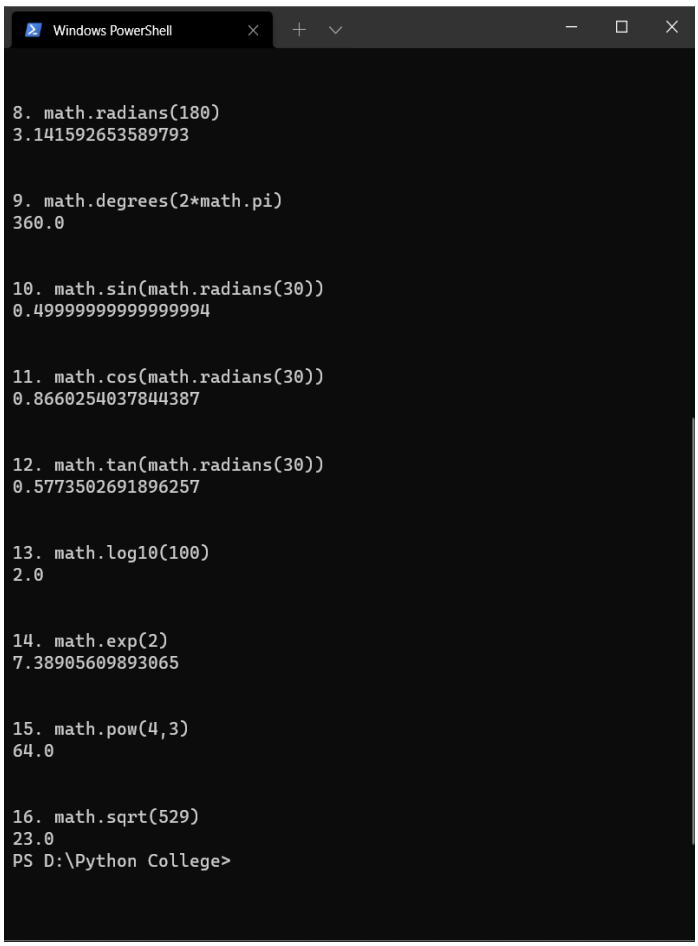
Program:

```
print("\n")
print("***** random and math module ( built-in libraries ) *****")
print("\n")
import random
print("1. random.random()")
print(random.random())
print("\n")
print("2. random.randint(1,100)")
print(random.randint(1,100))
print("\n")
print("3. random.randrange(0,100,10)")
print(random.randrange(0,100,10))
print("\n")
```

```
print("4. random.choice([1,29,34,45,56,62,77,81,99])")
print(random.choice([1,29,34,45,56,62,77,81,99]))
print("\n")
print("5. random.shuffle([1,29,34,45,56,62,77,81,99])")
numbers=[1,29,34,45,56,62,77,81,99]
random.shuffle(numbers)
print(numbers)
import math
print("\n")
print("6. math.pi")
print(math.pi)
print("\n")
print("7. math.e")
print(math.e)
print("\n")
print("8. math.radians(180)")
print(math.radians(180))
print("\n")
print("9. math.degrees(2*math.pi)")
print(math.degrees(2*math.pi))
print("\n")
print("10. math.sin(math.radians(30))")
print(math.sin(math.radians(30)))
print("\n")
print("11. math.cos(math.radians(30))")
print(math.cos(math.radians(30)))
print("\n")
print("12. math.tan(math.radians(30))")
print(math.tan(math.radians(30)))
print("\n")
print("13. math.log10(100)")
print(math.log10(100))
print("\n")
print("14. math.exp(2)")
print(math.exp(2))
print("\n")
print("15. math.pow(4,3)")
print(math.pow(4,3))
print("\n")
print("16. math.sqrt(529)")
print(math.sqrt(529))
```

Output:





Conclusion: Thus, we have developed a program using built-in libraries.