Chapter -04 Using Python Libraries

Working with some Standard Library Modules:

Other than *built-in functions*, *standard library* also provides some *modules* having functionality for *specialized actions*. Let us learn to use some modules such as *random and string modules of Python's standard library.*

1. Using Random Module:

Python has a *module* namely *random* that provides *random- number generators*. A *random* number in simple words means – *a number generated by chance, i.e. randomly.*

In order to use *random number generators* in Python program, first we have to *import module random* using the *import command*,

e.g.

import random

Some most *common random number generator* functions in *random module* are:

i) ***random()*** **:**

It returns a *random floating* point number ***N*** in the range ***[0.0, 1.0] i.e., 0.0<=N<1.0***. The number generated with random() will always be less than 1.0. ***(Only lower range –limit is inclusive)***. It generates a *floating point* number.

ii) ***randint(a, b):***

It returns a *random integer* ***N*** in the range ***(a,b), i.e. a <=N <=b (both range limits are inclusive).*** It generates *a floating point* number.

iii) ***random.uniform(a, b):***

It returns a *random floating* point number ***N*** such that ***a <= N <= b for a <=b and b <= N <= a for b < a.***

iv) ***random.randrange(stop):***

It returns a *randomly* selected element from range ***(0 to stop)*** *(both range limits are inclusive)*

v) ***random.randrange(start, stop[, step]):***

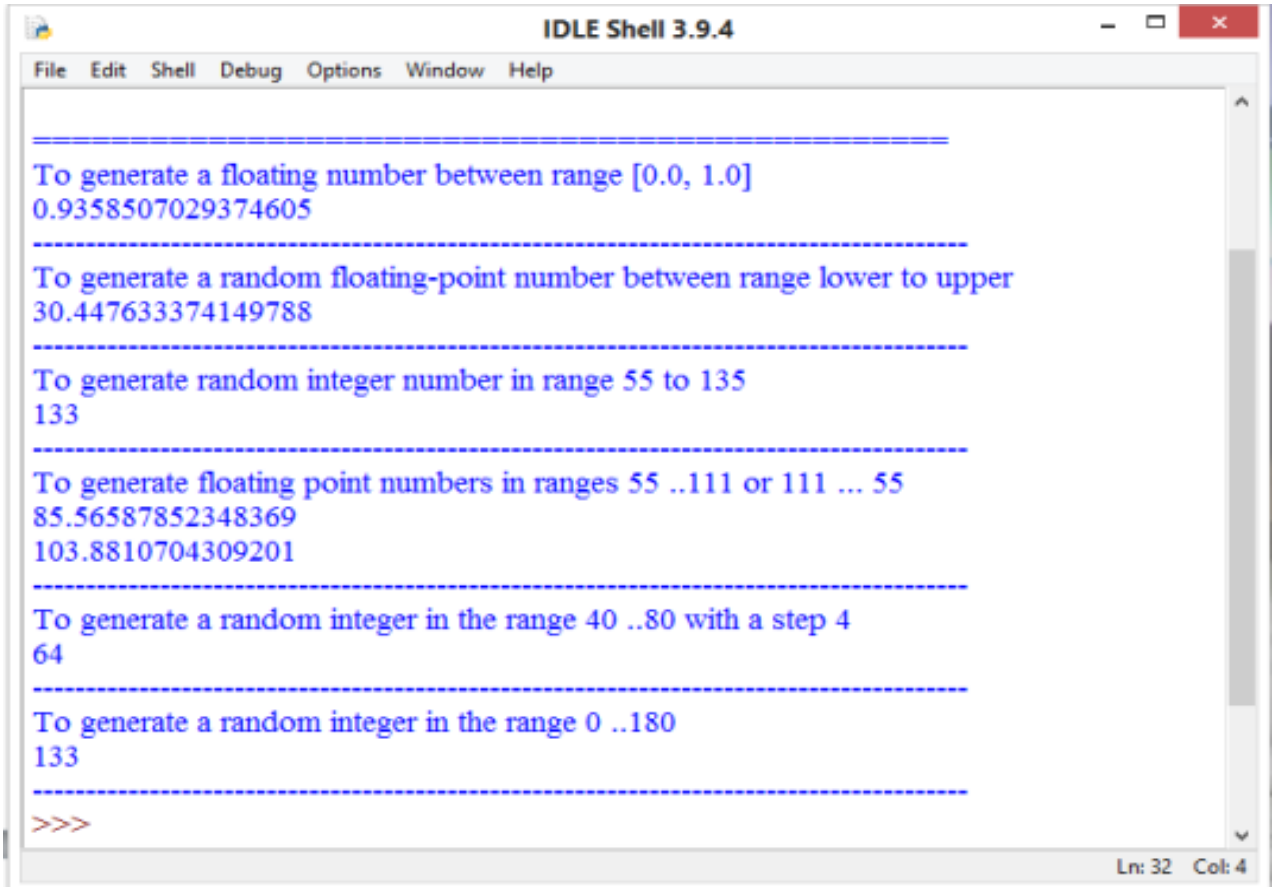
It returns a *randomly* selected element from ***range (start, stop, and step)***

Practical Implementation -1

#Program demonstrates the use of random module. import random

```
import random
print("\n==============================================")
print("To generate a floating number between range [0.0, 1.0]")
print(random.random())
print("-----------------------------------------------------------------------------------")
print("To generate a random floating-point number between range lower to upper")
print(random.random()*(35-15)+15)
print("-----------------------------------------------------------------------------------")
print("To generate random integer number in range 55 to 135")
print(random.randint(55,135))
print("-----------------------------------------------------------------------------------")
print("To generate floating point numbers in ranges 55 ..111 or 111 ... 55")
print(random.uniform(55,111))
print(random.uniform(111,55))
print("-----------------------------------------------------------------------------------")
print("To generate a random integer in the range 40 ..80 with a step 4")
print(random.randrange(40,80,4))
print("-----------------------------------------------------------------------------------")
print("To generate a random integer in the range 0 ..180")
print(random.randrange(180))
print ("-----------------------------------------------------------------------------------")
```

Output:



2. Using String Module:

Python has a module namely *string* that comes with many *constants and classes*. It offers useful ***utility functions and constants*.** Before using any ofthe constants/functions defined in the string module, one must have to import it using an import statement:

import string

Some useful ***constants*** defined in the string module are being listed below:

ii)	***string.ascii_letters***	It returns a string containing all the collection of ASCII letters.
iii)	***string.ascii_lowercase***	It returns a string containing all the lowercase ASCII letters.
iv)	***string.ascii_uppercase***	It returns all the uppercase ASCII letters.
v)	***string.digits***	It returns a string containing all the digits Python allows.
vi)	***String.hexdigits***	It returns a string containing all hexadecimal digits Python allows.
vii)	***String.octdigits***	It returns a string containing all the octal digits Python allows.
viii)	***string.punctuation***	It returns a string of ASCII characters which are considered punctuation characters.

The *string module* also offers a utility function ***capwords():***

capwords(<str>,sep ='None')	It capitalize each word separately and all leading, trailing whitespaces will be removed and inside whitespace characters **(e.g '\n' and spaces)** will be replaced with a single space.

Practical Implementation -1

 # Program demonstrates the use of string module.

```
import string
print("\n==================================================")
print("Constants defined in the string module")
print("string.ascii_letters ->",string.ascii_letters)
print("string.ascii_lowercase ->",string.ascii_lowercase)
print("string.ascii_uppercase ->",string.ascii_uppercase)
print("string.digit        ->",string.digits)
print("string.hexdigits ->",string.hexdigits)
print("string.octdigits ->",string.octdigits)
print("string.punctuation ->",string.punctuation)
print("==================================================")
print("Utility function Capwords:-")
Str = "God made the Earth;\nMan made confining countries"
print("Normal string is ->",Str)
print("string.capwords(Str) ->",string.capwords(Str))
print("==================================================")
```

Output:

```
File  Edit  Shell  Debug  Options  Window  Help
==================================================
Constants defined in the string module
string.ascii_letters -> abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
string.ascii_lowercase -> abcdefghijklmnopqrstuvwxyz
string.ascii_uppercase -> ABCDEFGHIJKLMNOPQRSTUVWXYZ
string.digit            -> 0123456789
string.hexdigits -> 0123456789abcdefABCDEF
string.octdigits -> 01234567
string.punctuation -> !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
==================================================
Utility function Capwords:-
Normal string is -> God made the Earth;
Man made confining countries
string.capwords(Str) -> God Made The Earth; Man Made Confining Countries
==================================================
>>>
                                                  Ln: 29  Col: 4
```