Python Programming

RGM College of Engineering & Technology (Autonomous)

Department of Computer Science & Engineering

Academic Year : 2020-2021

MODULES - 3

Guido Van Rossum

Learning Mantra

If you really strong in the basics, then remaining things will become so easy.

Agenda:

1. Working with math module
2. Working with ‘random’ module
3. Example Programs

10.Working with math module

- Python provides inbuilt module **math**.
- This module defines several functions which can be used for mathematical operations.

The main important functions are:

1. sqrt(x)
2. ceil(x)
3. floor(x)
4. fabs(x)
5. log(x)
6. sin(x)
7. tan(x)

Eg:

```
from math import *
print(sqrt(4))
print(ceil(10.1))
print(floor(10.1))
print(fabs(-10.6))      # Ignore sign, consider only value (float absolute function)
print(fabs(10.6))       # Ignore sign, consider only value
```

Output:

```
2.0
11
10
10.6
10.6
```

Note:

- We can find help for any module by using **help()** function.

Eg:

```
import math
help(math)
```

Output:

```
Help on built-in module math:

NAME
    math

DESCRIPTION
    This module is always available.  It provides access to the
    mathematical functions defined by the C standard.

FUNCTIONS
```

11. Working with ‘random’ module

- This module defines several functions to generate random numbers.
- We can use these functions while developing games, in cryptography and to generate random numbers on fly for authentication. (For example, OTPs).

1. random() function:

- This function always generate some float value between 0 and 1 (not inclusive).

Eg:

```
from random import random
# import random
for i in range(10):
    print(random())
```

```
0.5298687885975731
0.1959883975809048
0.7856765556766167
0.621345638337082
0.00173970627339004 04
0.05368238913099 1326
0.6870134702620266
0.01240050300491 4687
0.9615995552319757
0.5501802331038093
```

2. randint() function:

- This function is used to generate random integers between two given numbers (inclusive).

Eg:

```
from random import *
for i in range(10):
    print(randint(1,10))
```

```
4
5
3
9
4
5
6
10
6
6
```

3. uniform():

- It returns random float values between 2 given numbers (not inclusive).

Eg:

```
from random import *
for i in range(10):
    print(uniform(1,10))
```

```
4.219659431824531
5.058564769299971
7.526648575704248 5
4.190749282077976
3.8464174832123033
7.043381178043777
9.30504048393276
4.41712781745301
3.9935721537809465
8.274960415539518
```

Note:

- random() ===>in between 0 and 1 (not inclusive) ➔ float
- randint(x,y) ==>in between x and y (inclusive) ➔ int
- uniform(x,y) ==> in between x and y (not inclusive) ➔float

4. randrange([start],stop,[step])

- returns a random number from range start<= x < stop
- start argument is optional and default value is 0
- step argument is optional and default value is 1

For example,

- randrange(10)-->generates a number from 0 to 9
- randrange(1,11)-->generates a number from 1 to 10
- randrange(1,11,2)-->generates a number from 1,3,5,7,9

Eg:

```
from random import *
for i in range(10):
    print(randrange(10))
```

```
5
8
6
1
2
6
2
9
6
1
```

Eg:

```
from random import *

for i in range(10):

    print(randrange(1,11))
```

```
8
5
5
6
6
8
3
10
10
4
```

Eg:

```
from random import *

for i in range(10):

    print(randrange(1,11,2))
```

```
1
9
1
3
7
7
7
7
1
5
```

5. choice() function:

- It won't return random number.
- It will return a random object from the given list or tuple.
- Always the argument for this function is any indexable sequence. (i.e., Set is not supported).

Eg:

```
from random import *
list=["Sunny","Bunny","Chinny","Vinny",“binny"]
for i in range(10):
        print(choice(list))
```

Output:

Chinny
Chinny
binny
Sunny
Chinny
Vinny
Chinny
Bunny
Sunny
Bunny

Eg:

```
from random import *
list=("Sunny","Bunny","Chinny","Vinny",“binny“)
for i in range(10):
        print(choice(list))
```

Output:

Bunny
Bunny
binny
Bunny
Sunny
Vinny
Sunny
Chinny
Bunny
Chinny

Eg:

```
from random import *
list={"Sunny","Bunny","Chinny","Vinny",“binny"}
for i in range(10):                    #Set object is not support indexing
    print(choice(list))
```

```
TypeError: 'set' object is not subscriptable
```

Eg:

```
from random import *
list=["Sunny","Bunny","Chinny","Vinny",“binny"]
for i in range(10):
        print(choice('karthi'))
```

Output:

```
r
a
t
t
h
r
a
a
h
r
```

12. Example programs

Q 1. Write a Python program to generate a six digit random number as One Time Password (OTP).

Way 1:

```
from random import *
for i in range(10):
    print(randint(0,9),randint(0,9),randint(0,9),randint(0,9),randint(0,9),randint(0,9),sep=' ')
```

```
792314
871530
780895
157332
796357
502414
826712
785971
309040
015443
```

Eg:

```
from random import *
for i in range(10):
    print(randint(0,9),randint(0,9),randint(0,9),randint(0,9),randint(0,9),randint(0,9),sep=' ')
```

```
161446
197083
758751
996540
851466
666187
700286
719132
372328
136409
```

Way 2:

```
from random import *
for i in range(10):
    for x in range(6):
        print(randint(0,9),end='')        # Correct version
    print()
```

```
899095
829710
031807
656661
053638
290606
805589
240823
108752
356289
```

Eg:

```
from random import *
for i in range(10):
    for x in range(6):
        print(randint(0,9),end='')        # Correct version
    print()
```

```
738483
300653
071471
666878
872723
287566
540940
676690
328498
360878
```

Way 3:

```
from random import *
for i in range(10):
    print(randint(000000,999999),sep=' ')
```

```
962601
889203
384254
393560
633705
103495
985154
107436
149361
240032
```

Eg:

```
from random import *
for i in range(10):
    print(randint(000000,999999),sep=‘ ')
```

```
682096
266601
510098
805968
203852
838713
744178
568925
830862
538721
```

Eg:

```
from random import *
for i in range(10):    # Some times it may give wrong output also.
    print(randint(000000,999999),sep=' ')
```

```
347289
624504
70817
916648
627988
848795
66449
415174
989153
872602
```

Way 4:

```
from random import *
for i in range(10):
    print(randint(100000,999999),sep='')
```

```
869252
909810
596249
585590
346792
498318
326801
788542
835508
960551
```

Eg:

```
from random import *
for i in range(10):
    print(randint(100000,999999),sep='')
```

```
572571
692732
462218
234897
442399
480218
278091
182737
396578
244616
```

Flaw: It won't generate OTP which start from 0.

Q 2. Write a Python program to generate a random password of 6 length.

Within the OTP --

- 1,3,5 positions are alphabets.
- 2,4,6 positions are digits.

Way 1:

```
from random import *
for i in range(10):
    print(chr(randint(65,90)),randint(0,9),chr(randint(65,90)),randint(0,9),chr(randint(65,90)), randint(0,9))
```

```
R2I6T5
F4S6I6
W5B8K5
M3U7L6
C3A6M3
O5L7W3
T5W7X4
S3I7W1
L5C9A9
J9H4W4
```

Eg:

```
from random import *
for i in range(10):
    print(chr(randint(65,90)),randint(0,9),chr(randint(65,90)),randint(0,9),chr(randint(65,90)), randint(0,9))
```

```
B9D7J9
X0S2M0
S6R7Y6
B2B1B0
O4G4I0
J6U2P2
L6C6B3
C6C5K9
M1Q7N3
I3Y2F7
```

Way 2:

```
from random import *
for i in range(10):
    for x in range(1,7):
        if x%2 == 1:
            print(chr(randint(65,90)),end='')
        else:
            print(randint(0,9),end='')
    print()
```

```
P3H1U3
V1Q1A7
R4Z2G9
I3Y4H8
B8K7V9
N4X7S0
R5L3S2
O5A3Y8
M0D7J7
H7V2Z0
```

Eg:

```
from random import *
for i in range(10):
    for x in range(1,7):
        if x%2 == 1:
            print(chr(randint(65,90)),end='')
        else:
            print(randint(0,9),end='')
    print()
```

```
I4O0C6
E1K2K9
K9J6X8
W9I3Z9
Z0A1X9
T6S9M7
L6E8W7
A1X8J2
A7M6A3
R4R1E7
```

Any question?

If you try to practice programs yourself, then you will learn many things automatically

Spend few minutes and then enjoy the study

Thank You