

Random Numbers in Python

Riya Jacob K
Dept of BCA
2020 -21

What is a Random Number?

- Random number does NOT mean a different number every time.
- Random means something that can not be predicted logically.

Pseudo Random and True Random

- Computers work on programs, and programs are definitive set of instructions. So it means there must be some algorithm to generate a random number as well.
- If there is a program to generate random number it can be predicted, thus it is not truly random.
- Random numbers generated through a generation algorithm are called *pseudo random*.

- Can we make truly random numbers?
- Yes. In order to generate a truly random number on our computers we need to get the random data from some outside source. This outside source is generally our keystrokes, mouse movements, data on network etc.
- We do not need truly random numbers, unless its related to security (e.g. encryption keys) or the basis of application is the randomness (e.g. Digital roulette wheels).
- Here we will discussed about pseudo random numbers.

Generate Random Number

- NumPy (NumPy is a python library) offers **the random module** to work with random numbers.

- Example

- Generate a random integer from 0 to 100:

```
from numpy import random  
x = random.randint(100)  
print(x)
```

- Output

43

Generate Random Float

- The random module's rand() method returns a random float between 0 and 1.
- Example

Generate a random float from 0 to 1:

```
from numpy import random  
x = random.rand()  
print(x)
```

- Output

0.4140522484659195

Generate Random Array

- In NumPy we work with arrays, and you can use the two methods from the above examples to make random arrays.

Integers

- The randint() method takes a size parameter where you can specify the shape of an array.

Example

- Generate a 1-D array containing 5 random integers from 0 to 100:

```
from numpy import random
x=random.randint(100, size=(5))
print(x)
```

Output

```
[2 5 6 14 92]
```


Generate 2D Integer Random Array

- Example

Generate a 2-D array with 3 rows, each row containing 5 random integers from 0 to 100:

```
from numpy import random  
x = random.randint(100, size=(3, 5))  
print(x)
```

- Output

```
[[90 99 11 30 34]  
 [66 40 63 36 37]  
 [63 35 89 51 58]]
```

Generate 1D Float Random Array

- The rand() method also allows you to specify the shape of the array.
- Example

Generate a 1-D array containing 5 random floats:

```
from numpy import random  
x = random.rand(5)  
print(x)
```

Output

```
[0.4305005 0.1667810 0.9989659 0.4566901  
0.3199066]
```

Generate 2D Float Random Array

- Example

Generate a 2-D array with 3 rows, each row containing 5 random numbers:

```
from numpy import random
```

```
x = random.rand(3, 5)
```

```
print(x)
```

- Output

- ```
[[0.14252791 0.44691071 0.59274288 0.73873487 0.22082345]
 [0.00484242 0.36294206 0.88507594 0.56948479 0.15075563]
 [0.69195833 0.75111379 0.92780785 0.57986471 0.6203633]]
```

# Generate Random Number From Array

- The choice() method allows you to generate a random value based on an array of values.
- The choice() method takes an array as a parameter and randomly returns one of the values.

## Example

- Return one of the values in an array:

```
from numpy import random
x = random.choice([3, 5, 7, 9])
print(x)
```

## Output

7

- The choice() method also allows you to return an array of values.
- Add a size parameter to specify the shape of the array.

- Example

- Generate a 2-D array that consists of the values in the array parameter (3, 5, 7, and 9):

```
from numpy import random
x = random.choice([3, 5, 7, 9], size=(3, 5))
print(x)
```

- Output

```
[[5 9 7 5 9]
 [3 7 7 9 7]
 [3 7 9 9 5]]
```

# Random Module-Functions

| Method               | Description                                                       |
|----------------------|-------------------------------------------------------------------|
| <u>seed()</u>        | Initialize the random number generator                            |
| <u>getstate()</u>    | Returns the current internal state of the random number generator |
| <u>setstate()</u>    | Restores the internal state of the random number generator        |
| <u>getrandbits()</u> | Returns a number representing the random bits                     |
| <u>randrange()</u>   | Returns a random number between the given range                   |
| <u>randint()</u>     | Returns a random number between the given range                   |
| <u>choice()</u>      | Returns a random element from the given sequence                  |
| <u>choices()</u>     | Returns a list with a random selection from the given sequence    |
| <u>shuffle()</u>     | Takes a sequence and returns the sequence in a random order       |
| <u>sample()</u>      | Returns a given sample of a sequence                              |
| <u>random()</u>      | Returns a random float number between 0 and 1                     |

|                     |                                                                                                                                                             |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <u>uniform()</u>    | Returns a random float number between two given parameters                                                                                                  |
| <u>triangular()</u> | Returns a random float number between two given parameters, you can also set a mode parameter to specify the midpoint between the two other parameters      |
| betavariate()       | Returns a random float number between 0 and 1 based on the Beta distribution (used in statistics)                                                           |
| expovariate()       | Returns a random float number between 0 and 1, or between 0 and -1 if the parameter is negative, based on the Exponential distribution (used in statistics) |
| gammavariate()      | Returns a random float number between 0 and 1 based on the Gamma distribution (used in statistics)                                                          |
| gauss()             | Returns a random float number between 0 and 1 based on the Gaussian distribution (used in probability theories)                                             |
| lognormvariate()    | Returns a random float number between 0 and 1 based on a log-normal distribution (used in probability theories)                                             |
| normalvariate()     | Returns a random float number between 0 and 1 based on the normal distribution (used in probability theories)                                               |



`vonmisesvariate()` Returns a random float number between 0 and 1 based on the von Mises distribution (used in directional statistics)

`paretovariate()` Returns a random float number between 0 and 1 based on the Pareto distribution (used in probability theories)

`weibullvariate()` Returns a random float number between 0 and 1 based on the Weibull distribution (used in statistics)

Thank You