



Notas de Aula #10: Biblioteca RANDOM.PY (Funções e Métodos)

A biblioteca **RANDOM.PY** fornece métodos que permitem:

- i) A geração de números aleatórios;
- ii) Embaralhamento de listas, e;
- iii) Sorteio de um valor dentro de uma determinada sequência.

Para se usar essa biblioteca, devemos importar para o código a biblioteca `random.py` através do comando `import`.

```
import random
```

RANDINT()

Retorna um valor **N** do tipo inteiro (*integer*) aleatório, onde: **start** <= **N** <= **stop**.

O argumento **start** deverá ser sempre menor que o argumento **stop** (**start** < **stop**).

Sintaxe:

```
random.randint(start, stop)
```

No exemplo a seguir serão impressos 5 números inteiros entre 1 e 99

```
1 import random
2
3 for contador in range(1, 6):
4     print(random.randint(1, 99))
```



RANDRANGE()

Retorna um valor **N** do tipo inteiro (*integer*) aleatório, onde: **start** <= **N** < **stop**.

O argumento **start** deverá ser sempre menor que o argumento **stop** (**start** < **stop**). Caso o argumento **start** não seja informado, será assumido o valor 0 (zero).

Sintaxe:

```
random.randrange([start,] stop[, step])
```

No exemplo a seguir serão impressos 5 números inteiros entre 1 e 99.

```
1 import random
2
3 for contador in range(1, 6):
4     print(random.randrange(1, 99))
```

No próximo exemplo serão impressos 5 números ímpares inteiros entre 1 e 100.

```
1 import random
2
3 for contador in range(1, 6):
4     print(random.randrange(1, 100, 2))
```

RANDOM()

Retorna um valor **N** do tipo ponto flutuante (*float*) dentro do intervalo **[0.0, 1.0)** (fechado em 0.0 e aberto em 1.0).

Sintaxe:

```
random.random()
```

No exemplo a seguir serão impressos 5 números do tipo ponto flutuante (*float*) entre 0.0 e 1.0, sendo que o 1.0 não entrará no sorteio.

```
1 import random
2
3 for contador in range(1, 6):
4     print(random.random())
```



UNIFORM()

Retorna um valor **N** do tipo ponto flutuante (*float*) aleatório, onde:

- i) **start** <= **N** <= **stop** quando **start** <= **stop**, ou;
- ii) **stop** <= **N** <= **start** quando **stop** <= **start**.

Sintaxe:

```
random.uniform(start, stop)
```

No exemplo a seguir serão impressos 5 números do tipo ponto flutuante (*float*) entre 1 e 99 (inclusive).

```
1 import random
2
3 print('Gerando 5 números decimais aleatórios entre 1 e 99:')
4 for contador in range(1, 6):
5     print(random.uniform(1, 99))
```

CHOICE()

Retorna um elemento **N** aleatório dentro de uma **lista**.

Se argumento **lista** for uma lista vazia será disparado um erro do tipo `IndexError`.

Sintaxe:

```
random.choice(lista)
```

Vejamos o exemplo a seguir.

```
1 import random
2
3 lista_aleatorios = []
4
5 for contador in range(1, 6):
6     lista_aleatorios.append(random.uniform(1,99))
7
8 print(lista_aleatorios)
9
10 print(random.choice(lista_aleatorios))
```

Na linha 3 está sendo instanciada uma lista vazia, que será populada na linha 6 por 5 valores *float*.

Na linha 8 está sendo impressa a lista gerada completa.

Na linha 10 está sendo impresso aleatoriamente um dos valores da lista populada.