



# RANDOM( )

Essa biblioteca permite que geremos números ou que seja feita escolhas pseudoaleatória, com base em certas regras, abaixo será listado algumas das funções que podem ser usadas dentro do código

## Gerador de Valores

### UNIFORM

Dentro dessa função é possível a obtenção de valores de pontos flutuantes estabelecidos por um determinado intervalo ao chamar a função



**Nesse caso, é incluído o valor máximo passado como parâmetro**

#### Exemplo

```
import random

random.uniform(0,15)
```

### RANDRANGE

#### Exemplo

```
import random

random.randrange(0, 15, 2)
```

Nesse caso estou passando que o valor aleatório será múltiplo de 2

Tem basicamente as mesma características da função acima, porém a maior diferença está no parâmetro `sleep`, que permite controlar a qual valor o número gerado será múltiplo



**Nesse caso não é incluído o valor máximo fornecido**

---

## RANDOM

Comando mais básico para geração de números aleatórios, porém diferente dos demais, não recebe parâmetro e retorna um valor entre **0 e 1** (Não incluso o 1), **logo o valor retornado é formado para um ponto flutuante**

### Exemplo

```
import random

random.random()
```

---

## RANDINT

### Exemplo

```
import random

random.randint(0, 20)
```

Retorna um valor aleatório inteiro, com um intervalo definido nos parâmetros



**Nesse caso ambos os parâmetros são considerados como possíveis resultados**

---

## Manipulação de listas

Dentro da manipulação podemos observar algumas funções que nos ajudam a escolher de forma aleatória certos elementos de uma lista, bem como mudar a ordem dos mesmos dentro da lista

## CHOICE

```
import random

lista = ['A', 'B', 'C', 'D']
```

Esse conceito, é passado uma lista como parâmetro para a função e ela se encarrega de fazer a escolha de um dos itens de forma aleatória

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

- Será retornado um dos elementos da lista como resultado

## CHOICES

Dentro desse conceito, é observado que podemos tanto retornar apenas um valor como no exemplo acima, como uma sub-lista com base na lista passada como parâmetro, o diferencial dessa função se encontra em relação a dois parâmetros 1- Weights e 2 - K

1. Nesse parâmetro, podemos passar certos pesos para cada item da lista, que de certa forma influencia na escolha do mesmo
  - a. Weights - Por padrão quando é mencionado esse parâmetro apenas com seu próprio nome, estamos passando pesos de forma normal, deixando para o programa fazer o acumulo do mesmo



**Podemos dizer que cada valor representa de forma separada a probabilidade de cada valor ser selecionado**

```
import random
```

```
lista = ['A', 'B', 'C', 'D']
```

```
peso = [10, 5, 25, 60]
```

```
random.choicea(lista, weights = peso, k=3)
```

- b. cum\_weights - Esse parâmetro você já passa o peso de forma acumulativa, ou seja não necessitando de processamento a mais para o cálculo do peso



**Aqui o peso corresponde a um intervalo que corresponde da diferença do peso anterior com o peso comparado, ou seja o valor B = 15 não quer dizer que é maior que A = 10 (Pois entre 0 e 10 temos 10 unidades, já entre 15 e 10 temos 5 unidades)  
- Quanto maior o intervalo maiores as chances de escolha**

```
import random

lista = ['A', 'B', 'C', 'D']
peso_acumulado = [10, 15, 40, 100]

random.choicea(lista, weights = peso_acumulado, k=3)
```

Em resumo quando criamos um peso acumulado, estamos definindo um intervalo, ou seja se um intervalo for definido entre 10 e 25 se for gerado o valor 18 será retornado o item corresponde ao intervalo, por isso quanto maior o intervalo maiores as chances

2. K → Dentro desse parâmetros é estabelecido a quantidade de itens que a sub-lista terá 3 itens como mencionado no código acima