



MATRIZ

É uma variável multidimensional, onde os valores dos índices determinam a posição onde será armazenado o dado. Iremos trabalhar mais com matrizes bidimensionais, onde o primeiro índice indica a linha e o segundo a coluna.

V=

4	7	2	5	3
---	---	---	---	---

M=

3	8	1	5
0	2	4	7
2	5	9	3

T=

				6	3	8	1
			7	3	0	2	5
3	8	1	5	9	4	0	3
0	2	4	7	1	5		
2	5	9	3				

MATRIZ: EXEMPLOS

Também é possível inicializar o vetor no momento de sua declaração.

Sintaxe: **nome_matriz = [[valores_linha1], [valores_linha2],
[valores_linha3]]**

Sendo que todos os elementos devem ser separados por vírgula e serem todos do mesmo tipo. A seguir temos a declaração do “vetor_exemplo” com os valores atribuídos.

Para entender melhor, considere que seja necessário armazenar 6 elementos, considerando uma matriz com dimensão 2x3.

matriz_exemplo = [[9, 21, 17], [13, 7, -5]]

O primeiro colchete interno representa os elementos da primeira linha da matriz e o segundo colchete representa os elementos da 2 linha da matriz.

EXEMPLO: MATRIZ ATRIBUINDO E LISTANDO

Inicializando uma matriz de inteiros e listando os dados armazenados.

```
matriz_exemplo = [[9, 21, 17], [13, 7, -5]]  
print(matriz_exemplo)
```

```
C:\PythonCurso\venv\Scripts\python.exe C:/PythonCurso/exemplo32.py  
[[9, 21, 17], [13, 7, -5]]
```

```
Process finished with exit code 0
```

Como estão armazenados os números:

matriz_exemplo [0] [0] = 9

matriz_exemplo [0] [1] = 21

matriz_exemplo [0] [2] = 17

matriz_exemplo [1] [0] = 13

matriz_exemplo [1] [1] = 7

matriz_exemplo [1] [2] = -5

EXEMPLO: MATRIZ – ENTRADA E EXIBIÇÃO

Solicitar ao usuário para digitar 6 números que devem ser armazenados em uma matriz 3 x 2. O programa deverá exibir todas as informações da matriz.

```
1  matriz = []
2  quantidade_linhas = 3
3  quantidade_colunas = 2
4  for i in range(quantidade_linhas):
5      linha = []
6      for j in range(quantidade_colunas):
7          elemento = input("Digite um número na posição [0][1]: ".format(i, j))
8          linha.append(elemento)
9      matriz.append(linha[:])
10
11 print(matriz)
```

RESULTADO: MATRIZ – ENTRADA E EXIBIÇÃO

```
C:\PythonCurso\venv\Scripts\python.exe C:/PythonCurso/exemplo33.py
Digite um número na posição [0][1]: 21
Digite um número na posição [0][1]: 54
Digite um número na posição [0][1]: 09
Digite um número na posição [0][1]: 78
Digite um número na posição [0][1]: -95
Digite um número na posição [0][1]: 127
[['21', '54'], ['09', '78'], ['-95', '127']]
```

Armazenar o nome, idade e salário de 3 funcionários de uma empresa. Algumas informações devem ser exibidas no final:

- Todos os dados da pessoa com menor idade
- Nome da pessoa com maior salário

EXEMPLO: MATRIZ EMPRESA

```
1      funcionarios = []
2      quantidade = 3
3      maior_salario = 0
4      maior_idade = 0
5
6      for i in range(quantidade):
7          linha = [input('\n Digite o nome do(a) funcionário(a): '),
8                  int(input(' Digite a idade: ')),
9                  float(input(' Digite o salário: '))]
10
11         funcionarios.append(linha)
12
13     for i in range(quantidade):
14         if funcionarios[i][2] > funcionarios[maior_salario][2]:
15             maior_salario = i
16
17         if funcionarios[i][1] > funcionarios[maior_idade][1]:
18             maior_idade = i
19
20     print('\n Funcionário com maior salário: ', funcionarios[maior_salario])
21     print('\n Funcionário mais velho: ', funcionarios[maior_idade])
```

RESULTADO: MATRIZ - EMPRESA

```
C:\PythonCurso\venv\Scripts\python.exe C:/PythonCurso/exemplo35.py
```

```
Digite o nome do(a) funcionário(a): João
```

```
Digite a idade: 54
```

```
Digite o salário: 1254.99
```

```
Digite o nome do(a) funcionário(a): Carmen
```

```
Digite a idade: 60
```

```
Digite o salário: 1954.54
```

```
Digite o nome do(a) funcionário(a): Angela
```

```
Digite a idade: 33
```

```
Digite o salário: 2500.00
```

```
Funcionário com maior salário: ['Angela', 33, 2500.0]
```

```
Funcionário mais velho: ['Carmen', 60, 1954.54]
```


EXEMPLO: MATRIZ ENTRADA DE DADOS

No exemplo anterior, para a entrada de dados, foi utilizada a seguinte estrutura:

```
for i in range(quantidade):  
    linha = [input('\n Digite o nome do(a) funcionário(a): '),  
             int(input(' Digite a idade: ')),  
             float(input(' Digite o salário: '))]  
  
    funcionarios.append(linha)
```

Podemos também usar a seguinte estrutura para armazenar os dados em uma matriz:

```
for i in range(quantidade):  
    linha = []  
    linha.append(eval(input('\n Digite o nome do(a) funcionário(a): ')))  
    linha.append(int(input(' Digite a idade: ')))  
    linha.append(float(input(' Digite o salário: ')))  
  
    funcionarios.append(linha)
```

EXEMPLO: MATRIZ ALEATÓRIA

Armazenar em uma matriz 5 x 7, números inteiros aleatórios de 1 a 100.

*Usar **random**: biblioteca que contem funções para trabalhar com geração de números aleatórios*

```
print(sorteados[i][j], end=" ")
```

O parâmetro **end** é um parâmetro que serve para indicar ao final do print o que deve ser acrescentado, onde "separador" pode ser um espaço, um x ou qualquer string. A função **print** por padrão, acrescenta uma quebra de linha.

EXEMPLO: MATRIZ ALEATÓRIA

C:\PythonCurso\venv\Scripts\python.exe C:/PythonCurso/exemplo36.py

```
[61] [7] [70] [16] [70] [13] [55]  
[37] [10] [10] [13] [61] [13] [82]  
[10] [73] [25] [13] [85] [58] [94]  
[7] [64] [91] [37] [1] [4] [88]  
[25] [85] [97] [31] [88] [64] [97]
```

```
1  import random  
2  
3  
4  random.seed()  
5  
6  sorteados = []  
7  for i in range(5):  
8      linha = []  
9      for j in range(7):  
10         linha.append([random.randrange(1, 100, 3)])  
11  
12     sorteados.append(linha)  
13  
14     for i in range(5):  
15         print('\n')  
16         for j in range(7):  
17             print(sorteados[i][j], end=" ")
```

FUNÇÃO random

random.random() Retorna o próximo número de ponto flutuante aleatório no intervalo de 0.0 a 1.0

random.randint(a , b) Retorna um número inteiro entre **a** e **b**

random.randrange(a) Retorna um inteiro de **0** até **a**, inclusive

random.randrange(a, b, 2) Retorna número aleatório entre **a** e **b**, com intervalo de 2 números

random.choice('abcdefghij') Seleciona um dos elementos aleatoriamente

itens = [1, 2, 3, 4, 5, 6, 7]

random.shuffle(itens) Embaralha os itens aleatoriamente

random.seed (semente) gera o mesmo número aleatório a cada vez que usado por exemplo, um *randint* na sequência.

EXEMPLO: random

```
1 import random
2
3 n = random.random()
4 print('Número de ponto flutuante aleatório no intervalo de 0.0 a 1.0: ', n)
5
6 n = random.randint(1, 20)
7 print('Número inteiro entre a e b: ', n)
8
9 n = random.randrange(50)
10 print('Número inteiro de 0 até 50, inclusive: ', n)
11
12 n = random.randrange(0, 10, 2)
13 print('número aleatório entre a e b, com intervalo de 2 números: ', n)
14
15 n = random.choice('abcdefghij')
16 print('Seleciona um dos elementos do vetor aleatoriamente: ', n)
17
18 itens = [1, 2, 3, 4, 5, 6, 7]
19 random.shuffle(itens)
20 print(itens)
21
22 random.seed(20)
23 n = random.randint(1, 50)
24 print('Primeira vez com semente 20, gerado o número: ', n)
25
26 random.seed(20)
27 n = random.randint(1, 50)
28 print('Segunda vez com semente 20, gerado o número: ', n)
```

EXEMPLO: random - RESULTADO

```
C:\PythonCurso\venv\Scripts\python.exe C:/PythonCurso/exemplo37.py
Número de ponto flutuante aleatório no intervalo de 0.0 a 1.0: 0.4141666024751156
Número inteiro entre a e b: 5
Número inteiro de 0 até 50, inclusive: 34
número aleatório entre a e b, com intervalo de 2 números: 8
Seleciona um dos elementos do vetor aleatoriamente: d
[2, 7, 5, 1, 4, 3, 6]
Primeira vez com semente 20, gerado o número: 47
Segunda vez com semente 20, gerado o número: 47

Process finished with exit code 0
```