



Matriz



**INSTITUTO
FEDERAL**

Paraíba

Campus
João Pessoa



Definição de Matriz

- Matrizes são estruturas bidimensionais (tabelas) com linhas e colunas muito importantes na matemática, utilizadas por exemplo para a resolução de sistemas de equações e transformações lineares.
- Exemplo: matriz $M_{2 \times 3}$, 2 linhas e 3 colunas

$$M = \begin{bmatrix} 5 & 3 & 2 \\ 7 & 6 & 1 \end{bmatrix}$$

≡ Uso de Matriz em Programação

- Em programação, estamos interessados na sua utilização como forma de representar e organizar dados (estrutura de dados).
 - Exemplo: representação de um sistema de reservas de poltronas em um teatro.
- É representada como um Array Bidimensional, onde uma dimensão são as linhas e a outra dimensão são as colunas.
 - O acesso a cada posição da matriz é feito através de índices (linha e coluna).
- Exemplo:

Coluna

M =

0	5	3	2
1	7	6	1

Linha

$M_{0,0} = 5$
 $M_{0,1} = 3$
 $M_{0,2} = 2$
 $M_{1,0} = 7$
 $M_{1,1} = 6$
 $M_{1,2} = 1$

≡ Representação de Matriz em Python

- Em Python, uma matriz 2x3 é implementada como um **vetor de vetores**

$$M_{2 \times 3} = \begin{bmatrix} 5 & 3 & 2 \\ 7 & 6 & 1 \end{bmatrix}$$

- Representação da matriz M em Python:

$$M = [[5, 3, 2], [7, 6, 1]]$$

A matriz é um vetor com 2 elementos, onde cada elemento é um vetor com 3 elementos.

≡ Referenciando elementos

- Representação da matriz M em Python:

`M = [[5, 3, 2], [7, 6, 1]]`

- Referenciando elementos na matriz:

`M[linha][coluna]`

`M[0][0] = 5` `M[1][0] = 7`

`M[0][1] = 3` `M[1][1] = 6`

`M[0][2] = 2` `M[1][2] = 1`

Coluna `[0 1 2]` `[0 1 2]`
`M = [[5,3,2], [7,6,1]]`
Linha `[0 1]`



Criando uma matriz

- Criando uma matriz de ordem 2x3 (ou seja, 2 linhas e 3 colunas) com **valores predefinidos**:



```
matriz = [[1,2,3],[4,5,6]]
```

```
print(matriz)
```



```
[[1, 2, 3], [4, 5, 6]]
```

≡ Criando uma matriz

- Criando uma matriz de ordem 2x3 com **valores nulos**:

```
▶ nlin = 2 # número de linhas da matriz  
ncol = 3 # número de colunas da matriz  
  
matriz = [None]*nlin  
  
for i in range(nlin):  
    matriz[i] = [None]*ncol  
  
print(matriz)
```

```
↪ [[None, None, None], [None, None, None]]
```

- Outra maneira (mais prática):

```
▶ nlin = 2 # número de linhas da matriz  
ncol = 3 # número de colunas da matriz  
  
matriz = [ [None]*ncol for i in range(nlin) ]  
  
print(matriz)
```

```
↪ [[None, None, None], [None, None, None]]
```

≡ Lendo os elementos da matriz

- Criada a matriz, é possível percorrê-la e atribuir novos valores:

```
▶ nlin = 2 # número de linhas da matriz
ncol = 3 # número de colunas da matriz

matriz = [[None]*ncol for i in range(nlin)]

print('Matriz antes da leitura:')
print(matriz)

print('\nDigite os elementos da matriz:')
for i in range(nlin):
    for j in range(ncol):
        matriz[i][j] = int(input())

print('\nMatriz após a leitura:')
print(matriz)
```

```
↳ Matriz antes da leitura:
[[None, None, None], [None, None, None]]

Digite os elementos da matriz:
1
2
3
4
5
6

Matriz após a leitura:
[[1, 2, 3], [4, 5, 6]]
```


≡ Imprimindo a matriz

- Como já foi mostrado, uma matriz pode ser impressa com um simples *print(matriz)*.
- Dessa maneira, a matriz é mostrada na forma de "vetor de vetores".



```
matriz = [[1,2,3],[4,5,6]]
```

```
print(matriz)
```



```
[[1, 2, 3], [4, 5, 6]]
```

≡ Imprimindo no formato de matriz

- Entretanto, é possível imprimir a matriz no **formato de matriz**, ou seja, em linhas e colunas.

```
▶ nlin = 2  
ncol = 3  
m = [[1,22,3],[44,5,66]]  
  
for i in range(nlin):  
    for j in range(ncol):  
        print(f'{m[i][j]:4}',end=' ')  
    print()
```

```
↳ 1  22  3  
   44  5  66
```

≡ Exemplo 1

- Ler uma matriz A, de ordem 2x3 contendo inteiros, gerar uma matriz B onde cada elemento de B corresponderá ao dobro do respectivo elemento de A.

```
▶ # Definição do número de linhas e de colunas
nlin = 2
ncol = 3
# Criação das matrizes A e B
a = [[None]*ncol for i in range(nlin)]
b = [[None]*ncol for i in range(nlin)]
# Leitura da matriz A
print('Digite os elementos da matriz A:')
for i in range(nlin):
    for j in range(ncol):
        a[i][j] = int(input(f'A[{i}][{j}]: '))
# Cálculo da matriz B
for i in range(nlin):
    for j in range(ncol):
        b[i][j] = a[i][j] * 2
# Exibição das matrizes
print(f'\nA = {a}')
print(f'\nB = {b}')
```

↳ Digite os elementos da matriz A:

A[0][0]: 1

A[0][1]: 2

A[0][2]: 3

A[1][0]: 4

A[1][1]: 5

A[1][2]: 6

A = [[1, 2, 3], [4, 5, 6]]

B = [[2, 4, 6], [8, 10, 12]]

≡ Exemplo 2

- Ler uma matriz A de ordem M x N (obs: M e N serão lidos) contendo inteiros, calcular e exibir a soma de todos os seus elementos.

```
▶ # Leitura da ordem da matriz
print('Entre com a ordem da matriz:')
nlin = int(input('Nº de linhas: '))
ncol = int(input('Nº de colunas: '))
# Criação da matriz com valores nulos
m = [[None]*ncol for i in range(nlin)]
# Leitura da matriz
print('\nEntre com os elementos da matriz:')
for i in range(nlin):
    for j in range(ncol):
        m[i][j] = int(input(f'm[{i}][{j}]: '))
# Cálculo da soma dos elementos da matriz
s = 0
for i in range(nlin):
    for j in range(ncol):
        s += m[i][j]
# Exibição do resultado
print(f'\nSoma = {s}')
```

```
↳ Entre com a ordem da matriz:
Nº de linhas: 2
Nº de colunas: 3

Entre com os elementos da matriz:
m[0][0]: 1
m[0][1]: 2
m[0][2]: 3
m[1][0]: 4
m[1][1]: 5
m[1][2]: 6

Soma = 21
```

≡ Exemplo 3

- Gerar e imprimir uma matriz quadrada de ordem N (N será lido), onde cada elemento corresponderá a soma de seus índices. Obs: a matriz deve ser impressa no formato de matriz.

```
# Leitura da ordem da matriz
n = int(input('Digite a ordem da matriz: '))

# Criação da matriz com valores nulos
m = [[None]*n for i in range(n)]

# Cálculo dos elementos da matriz
for i in range(n):
    for j in range(n):
        m[i][j] = i + j

# Impressão da matriz
print('\nMatriz:')
for i in range(n):
    for j in range(n):
        print(f'{m[i][j]:4}',end='')
    print()
```

☞ Digite a ordem da matriz: 5

Matriz:

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

≡ Exemplo 4

- Dada uma matriz de inteiros, determinar o maior valor da matriz. Obs: a ordem da matriz será lida e os elementos serão gerados com valores aleatórios entre 1 e 20.

```
import random

# Leitura da ordem da matriz
nlin = int(input('Nº de linhas: '))
ncol = int(input('Nº de colunas: '))

# Criação da matriz com valores nulos
m = [[None]*ncol for i in range(nlin)]

# Preencimento da matriz com valores aleatórios
for i in range(nlin):
    for j in range(ncol):
        m[i][j] = random.randint(1,20)
```

```
# Exibição da matriz
print('\nMatriz:')
for i in range(nlin):
    for j in range(ncol):
        print(f'{m[i][j]:4}',end='')
    print()

# Cálculo do maior valor da matriz
maior = m[0][0]
for i in range(nlin):
    for j in range(ncol):
        if m[i][j] > maior:
            maior = m[i][j]
print(f'\nMaior = {maior}')
```

```
➡ Nº de linhas: 3
   Nº de colunas: 5

Matriz:
   15   4  10   7  12
   13   4   2   2  16
    9  15  11  20   9

Maior = 20
```

- **import random** – importa a biblioteca **random**, que contém funções para gerar números aleatórios.
- **randint(x,y)** – gera um número aleatório entre **x** e **y**.

≡ Exemplo 5

- Gerar uma matriz (3 x 5) com valores inteiros aleatórios (entre 1 e 10), calcular a soma de cada linha e a soma de cada coluna.

```
import random

# Inicialização das variáveis
nlin = 3
ncol = 5
m = [[None]*ncol for i in range(nlin)]

# Preenchimento da matriz com valores aleatórios
for i in range(nlin):
    for j in range(ncol):
        m[i][j] = random.randint(1,10)

# Exibição da matriz
print('Matriz:')
for i in range(nlin):
    for j in range(ncol):
        print(f'{m[i][j]:4}',end='')
    print()
```

```
# Soma das linhas
print('\nSoma de cada linha:')
for i in range(nlin):
    s = 0
    for j in range(ncol):
        s += m[i][j]
    print(f'{s:4}')

# Soma das colunas
print('\nSoma de cada coluna:')
for j in range(ncol):
    s = 0
    for i in range(nlin):
        s += m[i][j]
    print(f'{s:4}',end='')
print()
```

```
Matriz:
 7  2  5  5  2
 4  7  6  3  4
 7  2  9  1  9

Soma de cada linha:
21
24
28

Soma de cada coluna:
18 11 20  9 15
```