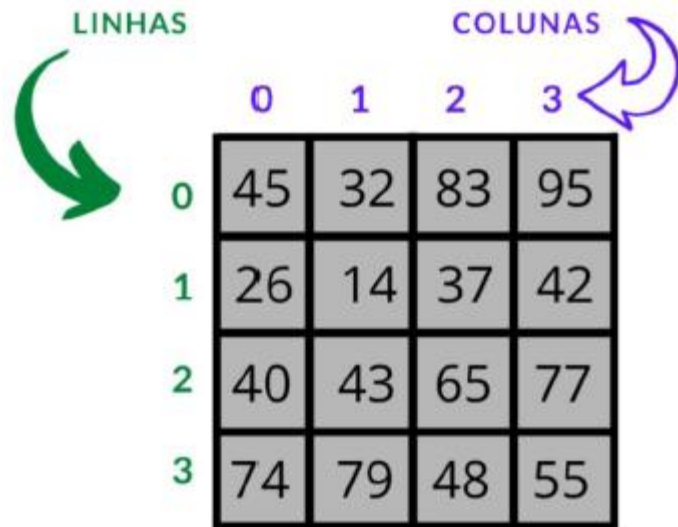




# MATRIZES



The diagram shows a 4x4 matrix with row indices (0-3) on the left and column indices (0-3) on top. A green arrow labeled 'LINHAS' points to the row indices, and a purple arrow labeled 'COLUNAS' points to the column indices. The matrix contains the following values:

	0	1	2	3
0	45	32	83	95
1	26	14	37	42
2	40	43	65	77
3	74	79	48	55

- São estruturas de dados bidimensionais com  $m$  linhas e  $n$  colunas ( $m \times n$ ). Cada elemento em uma matriz é identificado por um par de índices que indicam sua posição na matriz.
- Exemplo: O elemento 14 está na posição (1, 1).

```
25     print(matriz[3][2])
```

# MATRIZES

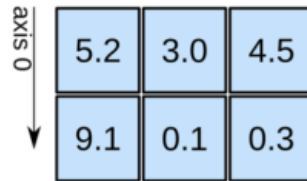
1D array



axis 0

shape: (4,)

2D array

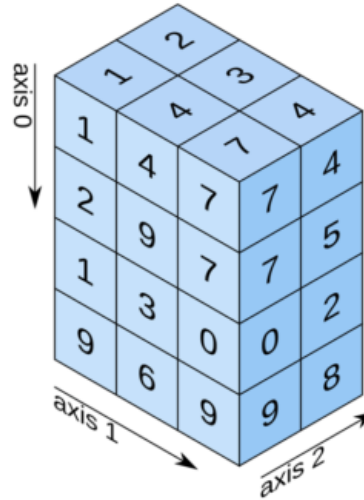


axis 0

axis 1

shape: (2, 3)

3D array



axis 0

axis 1

axis 2




shape: (4, 3, 2)

- As matrizes são amplamente usadas em programação para armazenar e manipular grandes conjuntos de dados de forma eficiente.
- Por exemplo:  
Armazenamento de uma imagem digital, onde cada elemento representa um pixel na imagem..

# APLICAÇÃO

- O retorno de uma consulta em um banco de dados é parecido como uma matriz onde temos linhas e colunas contendo informações.

10 • `select * from pedido;`

Result Grid   Filter Rows:  Edit: 

	id_pedido	id_cliente	status	data_pedido
▶	1	6	Pendente	2022-05-06
	2	6	Finalizado	2022-05-08
	3	3	Finalizado	2022-06-10
	4	4	Pendente	2022-06-28
	5	4	Finalizado	2022-07-27
	6	2	Pendente	2022-07-25
	7	8	Finalizado	2022-05-12
	8	9	Finalizado	2022-06-11
	9	10	Finalizado	2022-06-10

# PYTHON

- Em Python, uma matriz pode ser representada como uma lista de listas. Onde um elemento da lista contém uma linha da matriz, que por sua vez corresponde a uma coluna da matriz.

linhas da matriz

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

```
print(matriz[0][0])
```

para imprimir o  
primeiro elemento da  
matriz, ou seja: 1

colunas da matriz

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

0 1 2

0  
1

# INICIALIZAR UMA MATRIZ EM PYTHON COM ZEROS

```
19  #-----#
20  l = int(input("Digite a quantidade de linhas da matriz: "))
21  c = int(input("Digite a quantidade de colunas da matriz: "))
22
23  i = 0
24  matriz = []
25  while i < l:
26      linha = []
27      j = 0
28      while j < c:
29          linha.append(0)
30          j = j + 1
31      matriz.append(linha)
32      i = i + 1
```

# ENTRADA DE DADOS EM MATRIZ COM PYTHON

```
19  #-----#
20  l = int(input("Digite a quantidade de linhas da matriz: "))
21  c = int(input("Digite a quantidade de colunas da matriz: "))
22
23  i = 0
24  matriz = []
25  while i < l:
26      linha = []
27      j = 0
28      while j < c:
29          valor = int(input("Digite um número: "))
30          linha.append(valor)
31          j = j + 1
32      matriz.append(linha)
33      i = i + 1
```

# IMPRIMIR UMA MATRIZ

- Existem várias formas de imprimir uma matriz, a mais simples e de melhor visualização é imprimir linha após linha.

```
36     i = 0
37     while i < len(matriz):
38         print(matriz[i])
39         i += 1
```



# EXERCÍCIOS

1. Utilizando a estrutura de repetição While crie uma matriz 3 x 3 com os seguintes valores, após reescreva o código utilizando for.

```
[1, 2, 3]
```

```
[4, 5, 6]
```

```
[7, 8, 9]
```

# EXERCÍCIOS

2. Utilizando a estrutura de repetição **While** crie um programa que inicialize uma matriz 4 x 4, o usuário deve digitar os dados de entrada de cada elemento da matriz. Durante a execução o programa deve informar qual linha e coluna o usuário está preenchendo. Ao final imprima a matriz preenchida pelo usuário.

ENTRADA:

```
Lin-0 Col-2:10
Entrada de dados
Lin-0 Col-3:32
Entrada de dados
Lin-1 Col-0:40
Entrada de dados
Lin-1 Col-1:50
Entrada de dados
Lin-1 Col-2:
```

SAÍDA:

```
['12', '11', '10', '32']
['40', '50', '44', '10']
['99', '12', '15', '16']
['18', '19', '20', '22']
```

# EXERCÍCIOS

3. Faça um programa que leia uma matriz de 3 x 3 elementos, multiplique cada elemento de cada linha por 5 e armazene a multiplicação em uma nova matriz. Exemplo:

```
matriz = [[1,2,3],[4,5,6],[7,8,9]]  
nova_matriz = []
```

```
print(matriz[0])  
print(matriz[1])  
print(matriz[2])
```

```
print(nova_matriz[0])  
print(nova_matriz[1])  
print(nova_matriz[2])
```

ENTRADA:

```
[1, 2, 3]  
[4, 5, 6]  
[7, 8, 9]
```

SAÍDA:

```
[5, 10, 15]  
[20, 25, 30]  
[35, 40, 45]
```

# EXERCÍCIOS

4. Crie um programa que leia uma lista de 9 posições e converta em uma matriz de 3x3:

ENTRADA:

```
numeros = [10, 20, 30, 10, 20, 30, 10, 20, 30]
```

SAÍDA:

```
[10, 20, 30]  
[10, 20, 30]  
[10, 20, 30]
```

# EXERCÍCIOS

5. Crie um programa que inicialize uma matriz 8 x 8 no formato de uma rifa, preenchida com números 1. Em seguida solicite ao usuário a entrada da linha e coluna que deseja marcar um X e imprima novamente a matriz com a alteração realizada. Neste exercício estamos simulando a escolha de uma posição em uma rifa, caso necessário, você pode alimentar a rifa com números em sequência.

