

code001

June 4, 2024

```
[ ]: #Importing packages  
import numpy as np  
import pandas as pd
```

```
[ ]: #Working with data  
lista = [1,2,3,4,5,6,7,8,9]  
#Llevando una lista de python a numpy  
lista2 = np.array(lista)  
  
#Otra manera de hacer una lista  
#a =np.array([1,2,3])  
#print(f"a = {a}\nType = {type(a)}")  
  
#Imprimiendo las listas  
print(f"lista = {lista}\nType = {type(lista)}\n")  
print(f"lista2 = {lista2}\nType = {type(lista2)}")
```

```
lista = [1, 2, 3, 4, 5, 6, 7, 8, 9]  
Type = <class 'list'>
```

```
lista2 = [1 2 3 4 5 6 7 8 9]  
Type = <class 'numpy.ndarray'>
```

```
[ ]: #Creando objetos de multiples dimensiones  
matriz = [[1,2,3],[4,5,6],[7,8,9]]  
matriz2 = np.array(matriz)  
  
#Imprimiendo las matrices  
print(f"matriz = {matriz}\nType = {type(matriz)}\n")  
print(f"matriz2 = {matriz2}\nType = {type(matriz2)}")
```

```
matriz = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  
Type = <class 'list'>
```

```
matriz2 = [[1 2 3]  
 [4 5 6]  
 [7 8 9]]  
Type = <class 'numpy.ndarray'>
```

```
[ ]: #Indexing lista
print(f"lista[0] = {lista[0]}\nlista[1] = {lista[1]}\n")
print(f'lista[3] + lista[7] = {lista[3]} + {lista[7]} = {lista[3]+lista[7]}')
```

```
lista[0] = 1
lista[1] = 2
```

```
lista[3] + lista[7] = 4 + 8 = 12
```

```
[ ]: #Indexing matrices
print(f'matriz[0] = {matriz[0]}\n')
print(f'matriz[1] = {matriz[1]}\n')
print(f'matriz[2] = {matriz[2]}\n')

#Indexing por filas y columnas solo se puede con numpy
print(f'matriz2[0,1] = {matriz2[0,1]}')
```

```
matriz[0] = [1, 2, 3]
```

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

```
matriz[2] = [7, 8, 9]
```

```
matriz2[0,1] = 2
```

```
[ ]: #Slicing
print(f'matriz[0:3] = {matriz[0:3]}\n')
print(f'matriz2[0:3] = {matriz2[0:3]}\n')

#slicing
print(f'lista[:3] = {lista[:3]}')
print(f'matriz2[0::3] = {matriz2[0::3]}\n')

#También esto nos arrojará un error
#matriz[1:,0:2]
#TypeError: list indices must be integers or slices, not tuple

#Como se puede ver la forma de acceder correctamente es
print(f'matriz2[1:,0:2] = {matriz2[1:,0:2]}')
#Hay que recordar que el slicing no toma en cuenta el segundo dígito después de ↵
↵ los 2 puntos
```

```
matriz[0:3] = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

```
matriz2[0:3] = [[1 2 3]
[4 5 6]
[7 8 9]]
```

```

lista[::3] = [1, 4, 7]
matriz2[0::3] = [[1 2 3]]

matriz2[1:,0:2] = [[4 5]
[7 8]]

```

```

[ ]: #Haciendo una matriz de 3 dimensiones en python
matres =_
↪ [[1,2,3],[4,5,6],[7,8,9]],[[10,11,12],[13,14,15],[16,17,18]],[[19,20,21],[22,23,24],[25,26,27]]
print(f'Matriz de n3 en python = {matres}\n')

#Haciendo una matriz de 3 dimensiones en numpy
matres2 = np.array(matres)
print(f'Matriz de n3 en numpy = \n{matres2}')

```

Matriz de n3 en python = [[1, 2, 3], [4, 5, 6], [7, 8, 9]], [[10, 11, 12], [13, 14, 15], [16, 17, 18]], [[19, 20, 21], [22, 23, 24], [25, 26, 27]]]

Matriz de n3 en numpy =

```

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

[[10 11 12]
[13 14 15]
[16 17 18]]

[[19 20 21]
[22 23 24]
[25 26 27]]]

```

```

[ ]: #Accediendo a los elementos de numpy
print(f'matres2[1] = \n{matres2[1]}\n')

print(f'matres2[1][2] = {matres2[1][2]}\n')

print(f'matres2[1][1][2] = {matres2[1][1][2]}')

```

```

matres2[1] =
[[10 11 12]
[13 14 15]
[16 17 18]]

```

```
matres2[1][2] = [16 17 18]
```

```
matres2[1][1][2] = 15
```