

Questão 1

```
import numpy as np
x = np . array ( [ 1 , 2 , 3 , 4 , 5 , 6 ] )

print("a) x[5] = ",x[5]) # posição 5 do vetor
print("b) x[1:4] = ",x[1:4]) # da posição 1 a 3
print("c) x[1:] = ",x[1:]) # todos apos a posição 1
print("d) x[:-1] = ",x[:-1]) # todos exceto o ultimo
print("e) x[5:1:-1] = ",x[5:1:-1]) # da posição 5 a 1 com passo -1, invertido
print("f) x[[1, 5, 2, 1, 1]] = ",x[[1, 5, 2, 1, 1]]) #colocando em um novo vetor os valore

a) x[5] = 6
b) x[1:4] = [2 3 4]
c) x[1:] = [2 3 4 5 6]
d) x[:-1] = [1 2 3 4 5]
e) x[5:1:-1] = [6 5 4 3]
f) x[[1, 5, 2, 1, 1]] = [2 6 3 2 2]
```

Questão 2

[+ Código](#)
[+ Texto](#)

```
import numpy as np
M = np.array([ [ 10 , 2 , 10 , 5 ] , [ 2 , 5 , 1 , 6 ] , [ 2 , 4 , 8 , 10 ] , [ 4 , 10 , 3 , 5 ] ])
print(M)

M[:,0]=1
M[3,:]=1
print("\n")
print(M)

[[10  2 10  5]
 [ 2  5  1  6]
 [ 2  4  8 10]
 [ 4 10  3  5]]

[[ 1  2 10  5]
 [ 1  5  1  6]
 [ 1  4  8 10]
 [ 1  1  1  1]]
```

Questão 3

```
import numpy as np
A = np.array([ 1 , 2 , 3 ]) .reshape(( 1 , 3 ))
B = np.array([ 3 , 2 , 1 ]) .reshape(( 1 , 3 ))
M = np.array([[ 4 , 5 , 6 ],[ 6 , 5 , 3 ]])

result=A+B #se somou item a item
result.shape
```

```

print("a")""Resultado: ", result,"Shape: ", result.shape )

result=A+M #cada linha da matriz foi somada com o vetor
result.shape
print("b")""Resultado: ", result,"Shape: ", result.shape )

result=A.T #Transposição do vetor
result.shape
print("c")""Resultado: ", result,"Shape: ", result.shape )

result=A.T+B #matriz 3x3 da soma da vetor coluna A com o vetor linha B
result.shape
print("d")""Resultado: ", result,"Shape: ", result.shape )

result=M - np.vstack([A, B]) #a subtração da matriz formada pela junção de A e B subtraindo
result.shape
print("e")""Resultado: ", result,"Shape: ", result.shape )

#result= np.vstack([A, B.T]) #Erro, pois os itens devem ter as mesmas dimensões
result.shape
print("f")""Resultado: ", result,"Shape: ", result.shape )

result=np.vstack([A, B]) #matriz formada pela junção de A e B
result.shape
print("g")""Resultado: ", result,"Shape: ", result.shape )

result=M - 3 #se subtrai 3 de todos os valores de M
result.shape
print("h")""Resultado: ", result,"Shape: ", result.shape )

result=A*B #multiplicação de "matrizes" feitas em cima do vetor
result.shape
print("i")""Resultado: ", result,"Shape: ", result.shape )

result=np.matmul(A, B.T) #soma dos valores obtidos com a multiplicação de matrizes
result.shape
print("j")""Resultado: ", result,"Shape: ", result.shape )

```

```

a)Resultado: [[4 4 4]] Shape: (1, 3)
b)Resultado: [[5 7 9]
 [7 7 6]] Shape: (2, 3)
c)Resultado: [[1]
 [2]
 [3]] Shape: (3, 1)
d)Resultado: [[4 3 2]
 [5 4 3]
 [6 5 4]] Shape: (3, 3)
e)Resultado: [[3 3 3]
 [3 3 2]] Shape: (2, 3)
f)Resultado: [[3 3 3]
 [3 3 2]] Shape: (2, 3)
g)Resultado: [[1 2 3]
 [3 2 1]] Shape: (2, 3)
h)Resultado: [[1 2 3]
 [3 2 0]] Shape: (2, 3)

```

```
i)Resultado: [[3 4 3]] Shape: (1, 3)
j)Resultado: [[10]] Shape: (1, 1)
```

Questão 4

```
M = np.array([[ 2 , 10 , 7 , 6 ],[ 3 , 12 , 25 , 9 ]])
print(M)
print("\n")
M = np.append(M,[[30 , 21 , 19 , 1]],axis=0)
print(M)
```

```
B = np.append(M[0][1:3],M[1][1:3],axis=0)
B = B.reshape((2,2))
print(B)
```

```
[[ 2 10  7  6]
 [ 3 12 25  9]]
```

```
[[ 2 10  7  6]
 [ 3 12 25  9]
 [30 21 19  1]]
[[10  7]
 [12 25]]
```

Questão 5

```
x = np.arange(31,75,2)
print(x)
```

```
[31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67 69 71 73]
```

Questão 6

```
x = np.random.randint(100, size=10)
print(x)
index = np.argmax(x)
print(index)
x[index]=x[index]**2
print(x)
```

```
[69 53 78 77 21 54 99 72 22 31]
6
[ 69  53  78  77  21  54 9801  72  22  31]
```

✓

0s

conclusão: 16:24

×