

In [1]:

```
!pip install numpy
```

Collecting numpy

Downloading numpy-1.21.2-cp39-cp39-win\_amd64.whl (14.0 MB)

Installing collected packages: numpy

Successfully installed numpy-1.21.2

WARNING: You are using pip version 21.2.3; however, version 21.2.4 is available.

You should consider upgrading via the 'D:\curso-data-science\venv\Scripts\python.exe -m pip install --upgrade pip' command.

## Introdução ao Numpy

In [4]:

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

Out[4]:

```
array([[1, 2, 3],
       [4, 5, 6]])
```

## Tipos

In [5]:

```
type(matriz)
```

Out[5]:

```
numpy.ndarray
```

In [6]:

```
matriz.dtype.name
```

Out[6]:

```
'int32'
```

## Dimensões do DataFrame

In [7]:

```
matriz.shape
```

Out[7]:

```
(2, 3)
```

In [8]:

```
matriz.size
```

Out[8]:

6

### Criando valores (passo a passo, espaçados ou randômicos)

In [9]:

```
np.arange(0, 30,3)
```

Out[9]:

```
array([ 0,  3,  6,  9, 12, 15, 18, 21, 24, 27])
```

In [10]:

```
np.linspace(0, 10,4)
```

Out[10]:

```
array([ 0.          ,  3.33333333,  6.66666667, 10.          ])
```

In [11]:

```
np.random.randint(0,30,5)
```

Out[11]:

```
array([24, 11, 16,  3,  9])
```

In [12]:

```
matriz2 = np.random.randint(0,30,6).reshape(3,2)  
matriz2
```

Out[12]:

```
array([[ 3,  5],  
       [17,  8],  
       [19,  3]])
```

### Operações com Numpy

#### maior número

In [18]:

```
matriz.max()
```

Out[18]:

6

#### menor número

In [14]:

```
matriz.min()
```

Out[14]:

1

**posição do número, lembrando que começa no 0**

In [15]:

```
matriz.argmax()
```

Out[15]:

5

In [16]:

```
matriz.argmin()
```

Out[16]:

0

**Média da matriz**

In [20]:

```
matriz.mean()
```

Out[20]:

3.5

**Desvio padrão**

In [21]:

```
matriz.std()
```

Out[21]:

1.707825127659933

In [23]:

```
matriz3 = np.array([[2,4,6],[8,7,1]])  
matriz3
```

Out[23]:

```
array([[2, 4, 6],  
       [8, 7, 1]])
```

In [24]:

```
matriz3.shape
```

Out[24]:

```
(2, 3)
```

In [26]:

```
matriz
```

Out[26]:

```
array([[1, 2, 3],  
       [4, 5, 6]])
```

In [27]:

```
matriz3
```

Out[27]:

```
array([[2, 4, 6],  
       [8, 7, 1]])
```

### Somando matriz com o shape igual

In [28]:

```
matriz + matriz3
```

Out[28]:

```
array([[ 3,  6,  9],  
       [12, 12,  7]])
```

### Multiplicar matriz

In [29]:

```
matriz * matriz3
```

Out[29]:

```
array([[ 2,  8, 18],  
       [32, 35,  6]])
```

### Elevado

In [30]:

```
matriz3 ** 2
```

Out[30]:

```
array([[ 4, 16, 36],  
       [64, 49,  1]], dtype=int32)
```

### Seleção e alteração de valores

In [34]:

```
matriz3
```

Out[34]:

```
array([[2, 4, 6],
       [8, 7, 1]])
```

In [31]:

```
matriz3[1][1]
```

Out[31]:

```
7
```

In [32]:

```
matriz3[matriz3 > 4]
```

Out[32]:

```
array([6, 8, 7])
```

In [33]:

```
matriz3[1]
```

Out[33]:

```
array([8, 7, 1])
```

In [36]:

```
matriz3[1][matriz3[1] > 3]
```

Out[36]:

```
array([8, 7])
```

In [38]:

```
matriz3[1][0] = 16
matriz3
```

Out[38]:

```
array([[ 2,  4,  6],
       [16,  7,  1]])
```

In [40]:

```
matriz3 = np.insert(matriz3, 1, [0,0,1], axis=0)
matriz3
```

Out[40]:

```
array([[ 2,  4,  6],
       [ 0,  0,  1],
       [ 0,  0,  1],
       [16,  7,  1]])
```

In [41]:

```
matriz3 = np.delete(matriz3, 1, axis=0)
```

In [42]:

```
matriz3
```

Out[42]:

```
array([[ 2,  4,  6],
       [ 0,  0,  1],
       [16,  7,  1]])
```

In [43]:

```
matriz4 = matriz3
matriz4
```

Out[43]:

```
array([[ 2,  4,  6],
       [ 0,  0,  1],
       [16,  7,  1]])
```

In [45]:

```
matriz3 = np.delete(matriz3, 2, axis=0)
```

In [46]:

```
matriz3
```

Out[46]:

```
array([[2, 4, 6],
       [0, 0, 1]])
```

In [ ]: