

Boas-vindas

Esteja confortável, pegue uma água
e se acomode em um local tranquilo
que já começamos.

Como você **chega?**

1



2



3



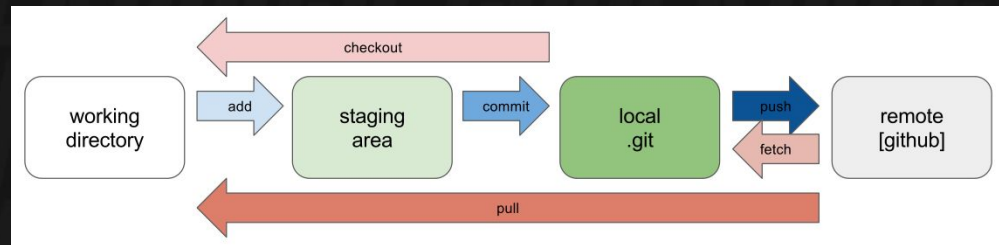
Esta aula será

- **gravada**

Resumo

da aula anterior

- ✓ Git x GitHub
- ✓ Comandos mais usados:
 - git pull
 - git add .
 - git commit -m "adicionado exercícios 3"
 - git push origin main



Perguntas?

Aula 07. PYTHON

Introdução a Numpy e Pandas

Objetivos da aula

- Conhecer as bibliotecas Numpy e Pandas
- Criação de arrays e ndarrays
- Compreender indexação e operações com numpy
- Geração de valores aleatórios
- Criação de dataframes e series

NumPy

NumPy

"NumPy" é uma abreviação de "Numeric Python".

É uma biblioteca de código aberto, que **fornece suporte para trabalhar com arrays e matrizes multidimensionais, bem como funções matemáticas para operações nesses arrays.**



NumPy – Arrays

Criar arrays multidimensionais, como `np.array()`.
Os arrays NumPy podem ter qualquer número de dimensões, permitindo que sejam usados para armazenar dados em matrizes e tensores.

```
import numpy as np
```

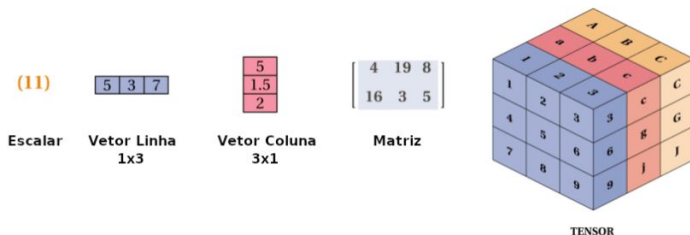
```
np.array([1,2,3,4])
```

```
✓ 0.5s
```

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

NumPy – Ndarrays

Tensores são uma extensão de matrizes para mais de duas dimensões. Tensor pode ter qualquer número de dimensões.



```
matriz = np.array([[1.0,2],[3,4]])  
matriz
```

✓ 0.0s

```
array([[1., 2.],  
       [3., 4.]])
```

```
tensor = np.array([[[1,2],[3,4]],[[5,6],[7,8]]])  
tensor
```

✓ 0.0s

```
array([[[1, 2],  
        [3, 4]],  
       [[5, 6],  
        [7, 8]]])
```

NumPy – Ndarrays

dtype é o atributo de um array que retorna o tipo de dados dos elementos do array.

shape é o atributo que fornece o tamanho do array ao longo de cada dimensão.

```
print(matriz.dtype)  
print(tensor.dtype)
```

✓ 0.0s

float64
int32

```
print(matriz.shape)  
print(tensor.shape)
```

✓ 0.0s

(2, 2)
(2, 2, 2)

NumPy – Criação de Nddarray

```
ones = np.ones((3,3))
```

```
ones
```

✓ 0.0s

```
array([[1., 1., 1.],  
       [1., 1., 1.],  
       [1., 1., 1.]])
```

```
full = np.full((3,2),42)
```

```
full
```

✓ 0.0s

```
array([[42, 42],  
       [42, 42],  
       [42, 42]])
```

```
zeros = np.zeros((2,3))
```

```
zeros
```

✓ 0.0s

```
array([[0., 0., 0.],  
       [0., 0., 0.]])
```

NumPy – Valores aleatórios

As principais formas são:

numpy.random.rand: gera valores aleatórios no intervalo $[0, 1)$.

numpy.random.randint: gera valores inteiros aleatórios em um intervalo especificado em uma forma especificada.

```
np.random.rand(2, 3)
```

✓ 0.0s

```
array([[0.45490846, 0.1245827 , 0.76373337],  
       [0.87837161, 0.2226538 , 0.36051527]])
```

```
np.random.randint(0, 10, size=(2, 3))
```

✓ 0.0s

```
array([[8, 2, 9],  
       [2, 5, 6]])
```

NumPy – Indexação

A indexação em um **array NumPy** é semelhante à indexação em uma lista ou vetor, com algumas diferenças.

```
a = np.array([[1, 2, 3], [4, 5, 6]])
print(a[0, 0]) # (linha 0, coluna 0)
print(a[0, 2]) # (linha 0, coluna 2)
print(a[-1, -1]) # (linha -1, coluna -1)
```

✓ 0.0s

1
3
6

```
print("Indexação 1:")
print(a[0:2, 1:3])
print("Indexação 2:")
print(a[:, 0:2])
print("Indexação 3:")
print(a[1, :])
```

4] ✓ 0.0s

Indexação 1:

[[2 3]

[5 6]]

Indexação 2:

[[1 2]

[4 5]]

Indexação 3:

[4 5 6]

NumPy - Operações

As principais operações ou agregações:

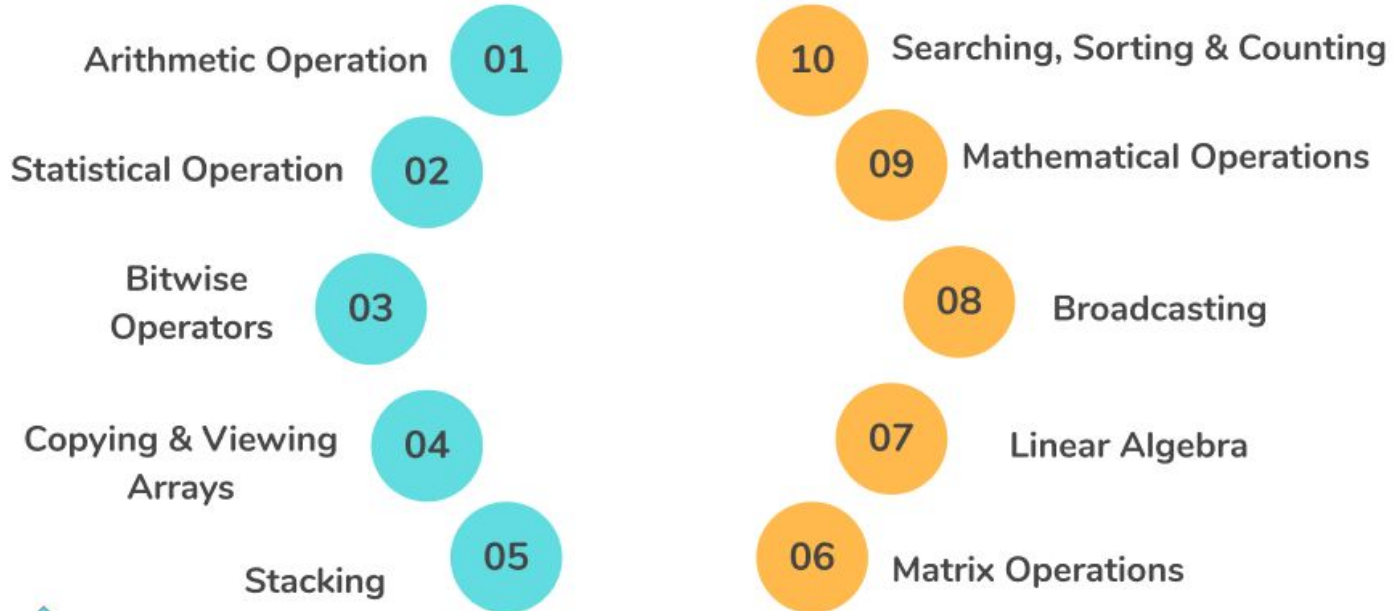
```
a = np.array([[1, 2, 3], [4, 5, 6]])  
a  
✓ 0.0s  
array([[1, 2, 3],  
       [4, 5, 6]])
```

```
print("sum:      ",a.sum())  
print("max:      ",a.max())  
print("min:      ",a.min())  
print("media:    ",a.mean())  
print("std:      ",a.std())  
print("var:      ",a.var())
```

✓ 0.0s

```
sum:      21  
max:      6  
min:      1  
media:    3.5  
std:      1.707825127659933  
var:      2.9166666666666665
```

Uses Of NumPy



Introdução a Pandas

Pandas?



Pandas

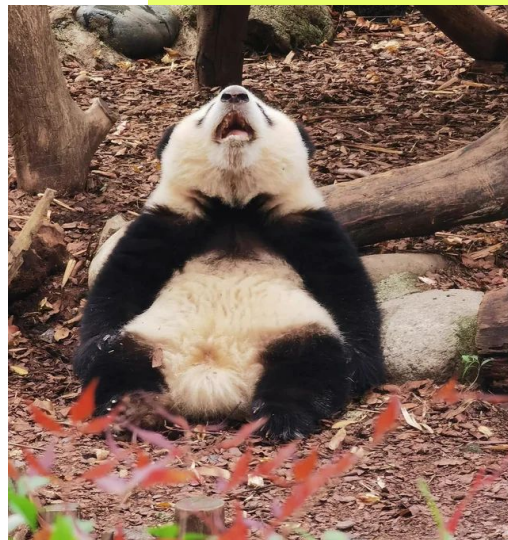
É uma biblioteca de código aberto que é usada para análise e manipulação de dados. Ela fornece estruturas de dados eficientes e de alta performance, juntamente com ferramentas para ler e escrever dados em diferentes formatos, como CSV, Excel, SQL, entre outros.



Pandas e NumPy

A biblioteca Pandas é construída sobre a biblioteca NumPy, o que significa que **as estruturas de dados do Pandas são criadas a partir de arrays NumPy**. Isso permite que as operações do NumPy sejam aplicadas às estruturas de dados do Pandas.

Portanto Pandas é uma extensão da biblioteca NumPy que adiciona recursos adicionais para análise e manipulação de dados.





Break

5 minutos e voltamos!





Break

10 minutos e voltamos!



Pandas – Serie

A **Series** é uma estrutura de dados unidimensional que pode armazenar diferentes tipos de dados.

A principal diferença entre uma Series e um array NumPy é que uma Series tem um índice associado a cada valor.

```
arr = np.array(range(10,20))  
arr
```

✓ 0.0s

```
array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19])
```

```
pd.Series(arr)
```

✓ 0.0s

0	10
1	11
2	12
3	13
4	14
5	15
6	16
7	17
8	18
9	19

dtype: int32

Pandas - DataFrame

DataFrame é uma estrutura de dados **bidimensional** que pode armazenar dados de diferentes tipos em diferentes colunas.

Cada coluna ou linha pode ser relacionada a uma Serie, portanto DataFrame é a junção de várias Series.

```
import pandas as pd
mat = np.array([[1, 2, 3], [4, 5, 6]])

df = pd.DataFrame(mat)
df
```

✓ 1.8s

	0	1	2
0	1	2	3
1	4	5	6

```
df = pd.DataFrame(mat, columns=['col1', 'col2', 'col3'])
df
```

✓ 0.0s

	col1	col2	col3
0	1	2	3
1	4	5	6

Biblioteca Pandas

Nas próximas aulas focaremos na Biblioteca Pandas e suas funcionalidades





Colocando em prática

Duração: 30 minutos



Colocando em prática

1. Crie um array NumPy com 10 elementos inteiros aleatórios no intervalo $[0, 100]$ e exiba os valores e o tipo de dado do array.
2. Crie uma matriz de 5×5 com valores aleatórios de $[0,1)$ e encontre o valor mínimo, máximo e a média da matriz.
3. Crie um array com 10 elementos aleatórios no intervalo $[0, 1]$. Em seguida, crie um novo array com os elementos do primeiro array multiplicados por 10 e convertidos para inteiro.
4. Crie um array de duas dimensões com shape $(3, 3)$ e valores aleatórios inteiros no intervalo $[0, 9]$. Substitua todos os elementos da segunda linha por -1 .
5. Crie um DataFrame com as seguintes informações:
 - ✓ Coluna "fruta": "Banana", "Maça", "Pera"
 - ✓ Coluna "preco": 7.90, 10.20, 11.80
 - ✓ Coluna "quantidade": 12, 3, 4

Perguntas?

Como foi a aula?

1

Que bom

O que foi super legal na aula e podemos sempre trazer para as próximas?

2

Que pena

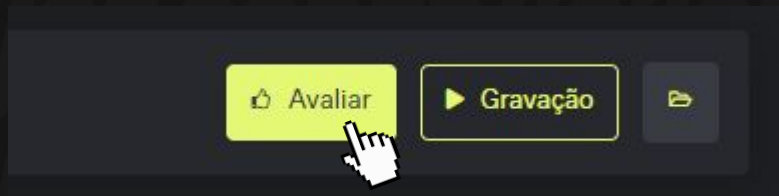
O que você acha que não funcionou bem e precisamos melhorar?

3

Que tal

Qual sugestão deveríamos tentar em próximas aulas?

O que você achou da aula?



Seu feedback vale pontos para o Top 10!! 🕶️



Deixe sua opinião!

1. Acesse a plataforma
2. Vá na aula do dia
3. Clique em **Avaliar**

Resumo

da aula de hoje

- ✓ Biblioteca Numpy
- ✓ Arrays e ndarrays
- ✓ Indexação
- ✓ Operações
- ✓ Valores aleatórios
- ✓ Biblioteca Pandas
- ✓ Dataframes
- ✓ Series



Ainda quer saber mais?
Recomendamos o
seguinte material



MATERIAL AMPLIADO

Recursos multimídia

Pandas vs NumPy

- ✓ [Pandas Vs NumPy: What's The Difference? | Interview Bit](#)



**Obrigado por estudar
conosco!**