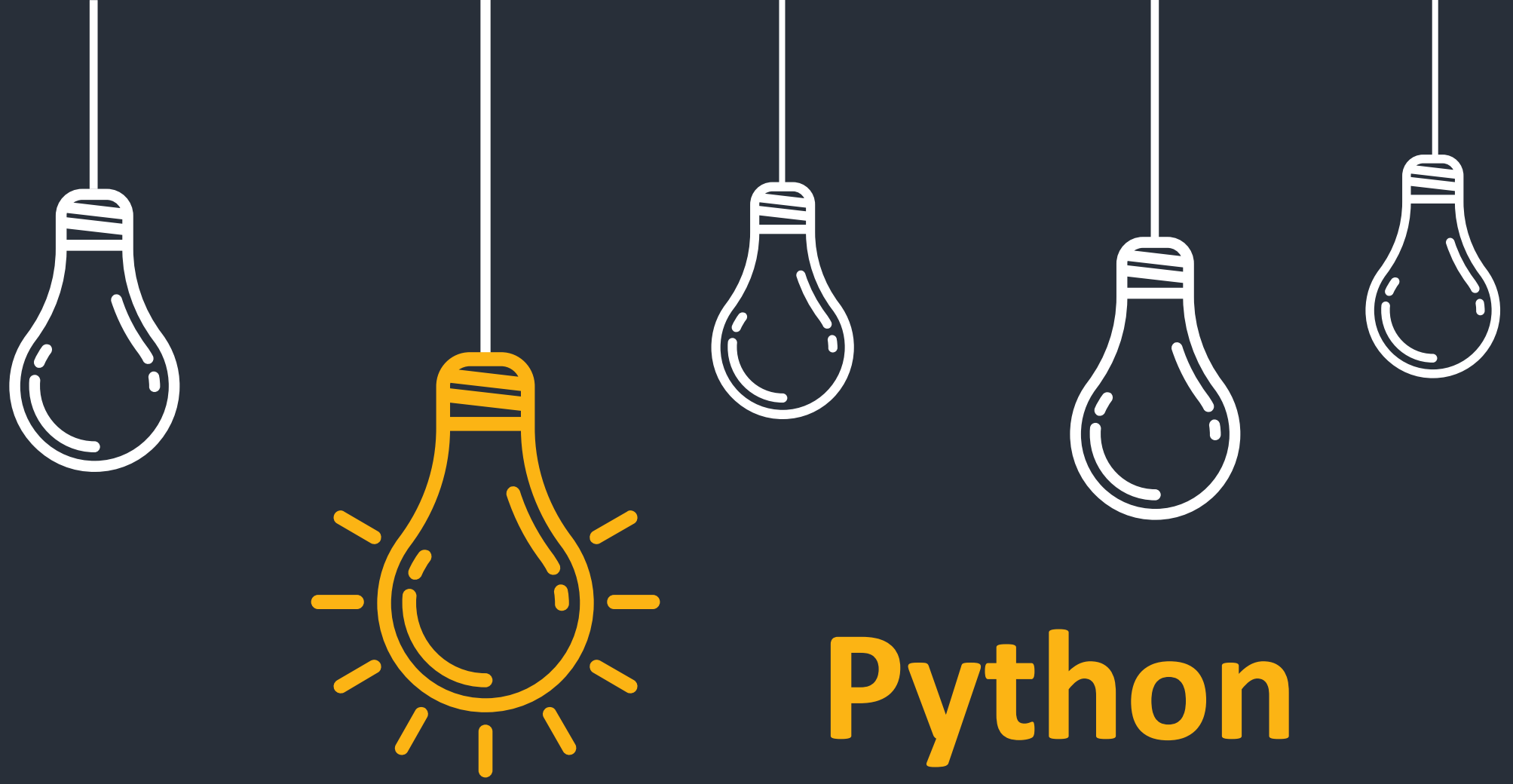

Python NumPy

Prof. Fernando F Ferrera

Python = Pacotes

- **pandas**: a fast, powerful, flexible, and easy-to-use open-source data analysis and manipulation tool. It's known as a fast, efficient, and easy-to-use tool for data analysis and manipulation.
- **NumPy**: a Python library that provides a multidimensional array object, an assortment of routines for fast operations on arrays, and much more.
- **TensorFlow**: an end-to-end open-source platform for machine learning. It has a comprehensive ecosystem of tools, libraries, and community resources that lets researchers and developers easily build and deploy ML-powered applications.

NumPy = Numerical Python

```
import modulo
import modulo.algo
import modulo.algo as malg
from modulo import *
from modulo import algo
from modulo import algo.item as ait
```

Import Numpy as np

V=np.array([,,,...],[,,,...],...[,,,...])

Criação de tensores


```
import numpy as np
```

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

```
v.shape = (2,2)
print(v)
```

```
v = np.array([1,2,3,4]).reshape(2,2)
print(v)
```

```
v = np.array([1,2,3,4,5,6,7,8])
v = v.reshape(2, -1)
print(v)
```

```
v = np.array(range(50)).reshape(2,5,5)

print('Shape = ', v.shape)
print('Número de dimensões = ', v.ndim)
print('Número de elementos = ', v.size)
print('Tensor v = \n', v)
```

As funções zeros, ones e diag

```
V = np.zeros((3,3))
print('V = \n', V)
U = np.ones((3,3))
print('U = \n', U)
D = np.diag([10, 10, 10])
print('D = \n', D)
```

```
V =
 [[0. 0. 0.]
  [0. 0. 0.]
  [0. 0. 0.]]
U =
  [[1. 1. 1.]
  [1. 1. 1.]
  [1. 1. 1.]]
D =
  [[10  0  0]
  [ 0 10  0]
  [ 0  0 10]]
```

As funções arange e linspace permitem a criação de sequências

```
v = np.arange(0, 5, 0.5)

u = np.linspace(0, 5, 10)

print("v =", v)
print("u =", u)
```

Agora, vejam essa função flatten

```
v = np.array(range(50)).reshape(2,5,5)

print(v.flatten())
```

Indexação de Tensores

```
v = np.array(range(50)).reshape(2,5,5)
print(v[1])
```

Resposta ?

```
v = np.array([10, 20, 30, 40]).reshape(2,2)
print(v[0,1])
```

Resposta ?

Tensor de rank 3:

```
v = np.arange(8).reshape(2,2,2)
print("v =\n", v)
print("\n v[0,0,1] =", v[0,0,1])
```

Resposta ?

Sub Tensores


```
A = np.arange(15).reshape(3,5)
print(A)

subA = A[0:2, 2:4]
print(subA)
```

Podemos passar uma lista:

```
u = np.array([2.0, 3.5, 4.0, -10.1])
v = u[[2,3]]
print(v)

A = np.arange(10).reshape(2,5)
B = A[:, [2,4]]
Print(B)
```

Operações sobre Tensores

```
v = np.array([10,20,30])
u = np.array([2,2,2])
w = u+v
print(w)

w = u*v
print(w)

x = u/v
print("x =", x)

y = w**2
print(w)
```

Funções Específicas

```
x = np.arange(10) media = x.mean()
 menor_valor = np.min(x)
arg_max = np.argmax(x)
print("Média =", media)
print("Menor valor =", menor_valor)
print("Arg max =", arg_max)
```

```
A = np.array([10, 30, 40, 20]).reshape(2,2)
menor_colunas = A.min(axis=0)
```

Funções Específicas: vetores

produto interno (ou produto escalar) pode ser realizada pela função dot:

```
w = np.dot(u, v)
print("w =", w)

x = u.dot(v)
print("x =", x)
```

Funções Específicas: matrizes

produto interno (ou produto escalar) pode ser realizada pela função dot:

```
A = np.ones((2,2))
B = 10*np.ones((2,2))
C= np.dot(A,B)
print(C)
```

```
C = A @ B
print(C)
```

```
u = np.arange(5)
v = np.exp(u)
print(v)
```

```
v = np.sin(u)
print(v)
```

Funções Específicas: matrizes

Operadores Lógios:

```
u = np.array([-1, 2, -3])

w = u[u < 0]
print(w)

print("u =", u)
u[u < 0] = 0
print("u =", u)
```

```
u = np.arange(5)
v = np.exp(u)
print(v)

v = np.sin(u)
print(v)
```

Funções Específicas: matrizes

```
A = np.arange(4).reshape(2,2)
print("Transposta de A =\n", A.transpose())
print("Transposta de A =\n", A.T)
```

```
A = np.array([10, 20, 30, 40]).reshape(2,2)
inv_A = np.linalg.inv(A)
print(inv_A)

det_A = np.linalg.det(A)
```

Solução de sistemas lineares


```
A = np.array([10, 20, 30, 40]).reshape(2,2)

b = np.array([5,10])

x = np.linalg.solve(A, b)

print(x)
```

Créditos:


```
@misc{pitombeira-neto,
  author="Pitombeira Neto, Anselmo Ramalho",
  title="Introdução ao Numerical Python (Numpy)",
  year=2020,
  url=http://www.opl.ufc.br/pt/post/numpy/,
  note = "[Online; accessed 2021-05-07]"
}
```