



Python - NumPy



Multibancas



Reta Final

Professor Rogerão Araújo

www.professorgabrielpacheco.com.br

<https://linktr.ee/professorgabrielpacheco>



Introdução e conceituação

Conceituação



Numerical Python

É uma biblioteca Python usada para trabalhar com arrays

Visa fornecer um objeto de matriz mais rápido que as listas tradicionais do Python

Conceituação

É um pacote fundamental

Fornece

Para a computação científica em Python

Estruturas de dados para armazenar e manipular grandes matrizes e vetores multidimensionais

Funções e operações matemáticas para trabalhar com esses dados

Onde é utilizado

Ciência de dados

Aprendizado de
máquina

Processamento de
imagens

Física

Engenharia

Finanças

Muitas outras
disciplinas que lidam
com dados numéricos
e computacionais

ndarray



É um objeto representa um array

Descreve o formato de cada elemento na matriz

Pacote NumPy

Multidimensional

Homogêneo

De itens de tamanho fixo

A ordem de byte

Quantos bytes ocupa na memória

Se é um número inteiro, um número de ponto flutuante ou outro tipo

É construído em torno desse objeto

Exemplo 1

- Código:
 - import **numpy**
 - **arrei** = **numpy.array**([10, 12, 14])
 - print(arrei)
 - print(type(arrei))
 - print(arrei.dtype)
 - print(arrei.size)
 - print(arrei.ndim)
 - print(arrei.shape)
- Resultado da execução:
 - [10 12 14]
 - <class 'numpy.ndarray'>
 - int64
 - 3
 - 1
 - (3,)

Exemplo 2

- Código:
 - import **numpy** as **np**
 - linguagens = ["Python", "Java", "PHP", "C"]
 - arrei = **np.array**(linguagens)
 - print(arrei)
 - print(type(arrei))
 - print(arrei.dtype)
 - print(arrei.size)
 - print(arrei.ndim)
 - print(arrei.shape)
- Resultado da execução:
 - ['Python' 'Java' 'PHP' 'C']
 - <class 'numpy.ndarray'>
 - <U6
 - 4
 - 1
 - (4,)

Exemplo 3

- Código:
 - import **numpy** as **np**
 - lista = [[1, 2, 3, 4], [15, 16, 17, 18]]
 - arrei = **np.array**(lista)
 - print(arrei)
 - print(type(arrei))
 - print(arrei.dtype)
 - print(arrei.size)
 - print(arrei.ndim)
 - print(arrei.shape)
- Resultado da execução:
 - [[1 2 3 4]
 - [15 16 17 18]]
 - <class 'numpy.ndarray'>
 - int64
 - 8
 - 2
 - (2, 4)

Exemplo 4

- Código:
 - import **numpy** as **np**
 - lista = [[1, 2, 3, 4], [15, 16, 17, 18]]
 - arrei = **np.array**(lista)
 - print(arrei[arrei < 5])
 - print(arrei[arrei > 5])
- Resultado da execução:
 - [1 2 3 4]
 - [15 16 17 18]



Questões de concursos

[UFMT 2021 UFMT – Técnico de Tecnologia da Informação] A respeito da linguagem python, analise as afirmativas.

- [I] Possui bibliotecas como pandas e numpy.
- [II] Disponibiliza documentação na Internet.
- [III] É uma linguagem de baixo nível.

Questões de concursos

[UFMT 2021 UFMT – Técnico de Tecnologia da Informação] A respeito da linguagem python, analise as afirmativas.

- **[I] Possui bibliotecas como pandas e numpy.**
- **[II] Disponibiliza documentação na Internet.**
- [III] É uma linguagem de ~~baixo~~ **alto** nível.



Questões de concursos

[UFMT 2021 UFMT – Técnico de Tecnologia da Informação] Está correto o que se afirma em

- [A] I e III, apenas.
- [B] I e II, apenas.
- [C] II e III, apenas.
- [D] III, apenas.

Questões de concursos

[UFMT 2021 UFMT – Técnico de Tecnologia da Informação] Está correto o que se afirma em

- [A] I e III, apenas.
- **[B] I e II, apenas.**
- [C] II e III, apenas.
- [D] III, apenas.



Questões de concursos

[FUNDATEC 2022 AGERGS – Técnico Superior Engenheiro de Dados]

Analise o código Python a seguir:

- `import numpy as np`
- `series = [[23,45,12,679], [14,48,69,38]]`
- `new_series = np.array(series)`
- `print(new_series.ndim)`
- `print(new_series.shape)`

Comentários

- Código:
 - `import numpy as np`
 - `series = [[23, 45, 12, 679], [14, 48, 69, 38]]`
 - `new_series = np.array(series)`
 - `print(new_series)`
 - `print(type(new_series))`
 - `print(new_series.dtype)`
 - `print(new_series.size)`
 - `print(new_series.ndim)`
 - `print(new_series.shape)`
- Resultado da execução:
 - `[[23 45 12 679]`
 - `[14 48 69 38]]`
 - `<class 'numpy.ndarray'>`
 - `int64`
 - `8`
 - `2`
 - `(2, 4)`



Questões de concursos

[FUNDATEC 2022 AGERGS – Técnico Superior Engenheiro de Dados] Qual é a saída correta?

- [A] 2 (2, 4)
- [B] 2 (4, 2)
- [C] 4 (2)
- [D] 2 (4)
- [E] 4 (2,4)



Questões de concursos

[FUNDATEC 2022 AGERGS – Técnico Superior Engenheiro de Dados] Qual é a saída correta?

- [A] 2 (2, 4)
- [B] 2 (4, 2)
- [C] 4 (2)
- [D] 2 (4)
- [E] 4 (2,4)

Questões de concursos

[CESGRANRIO 2021 Banco do Brasil – Agente de Tecnologia] Considere o código Python a seguir.

- `import numpy as np`
- `a = np.array ([[1,2,3],[4,5,6],[7,8,9]])`
- `print (a[a>5])`

Comentários

- Código:
 - `import numpy as np`
 - `a = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])`
 - `print(a)`
 - `print(a.size)`
 - `print(a.ndim)`
 - `print(a.shape)`
 - `print(a[a>5])`
- Resultado da execução:
 - `[[1 2 3]`
 - `[4 5 6]`
 - `[7 8 9]]`
 - `9`
 - `2`
 - `(3, 3)`
 - `[6 7 8 9]`

Questões de concursos

[CESGRANRIO 2021 Banco do Brasil – Agente de Tecnologia] O que será exibido após esse código ser executado?

- [A]
 - [[False False False]
 - [False False True]
 - [True True True]]
- [B]
 - [[False False False]
 - [False False 6]
 - [7 8 9]]

Questões de concursos

[CESGRANRIO 2021 Banco do Brasil – Agente de Tecnologia] O que será exibido após esse código ser executado?

- [C]
 - [[]
 - [6]
 - [7 8 9]]
- [D]
 - [False False False False False 6 7 8 9]
- [E]
 - [6 7 8 9]



Questões de concursos

[CESGRANRIO 2021 Banco do Brasil – Agente de Tecnologia] O que será exibido após esse código ser executado?

- [E]
– [6 7 8 9]



Indexação e fatiamento de arrays

Indexação de um array

Pode-se acessar um elemento de array referindo-se ao seu número de índice

Utiliza-se a indexação numérica (array[n])

Os índices em arrays NumPy começam com 0

Informando qual a posição (n)

Índice 0

Índice 1

Índice 2

Índice n - 1

Seu índice

Primeiro elemento

Segundo elemento

Terceiro elemento

Último elemento

Indexação de um array

Pode-se também usar valores negativos para representar os índices de um array

Índice -1

Índice -2

Índice -3

Último
elemento

Penúltimo
elemento

Antepenúltimo
elemento

Exemplo 1

- Código:
 - import **numpy**
 - **arrei** = **numpy.array**([10, 12, 14])
 - print(arrei[0])
 - print(arrei[1])
 - print(arrei[2])
 - print(arrei[1] + arrei[2])
- Resultado da execução:
 - 10
 - 12
 - 14
 - 26

Exemplo 2

- Código:
 - import **numpy** as **np**
 - linguagens = ["Python", "Java", "PHP", "C"]
 - arrei = **np.array**(linguagens)
 - print(arrei[-1])
 - print(arrei[-2])
 - print(arrei[-3])
 - print(arrei[-4])
 - print(arrei[-1] + arrei[-2])
- Resultado da execução:
 - C
 - PHP
 - Java
 - Python
 - CPHP

Exemplo 3

- Código:
 - import **numpy** as **np**
 - lista = [[1, 2, 3, 4], [15, 16, 17, 18]]
 - arrei = **np.array**(lista)
 - print(arrei[0, 0])
 - print(arrei[1, 2])
 - print(arrei[-1, -3])
 - print(arrei[-2, -2])
- Resultado da execução:
 - 1
 - 17
 - 16
 - 3

Fatiando um array

`array[início:fim:passo]`

início

É um número inteiro especificando em qual posição

Iniciar

fim

É um número inteiro especificando em qual posição

Parar

passo

É um número inteiro especificando a incrementação

É opcional

Padrão é 1

Exemplos

- Código:
 - import **numpy** as **np**
 - linguagens = ["Python", "Java", "PHP", "C"]
 - arrei = **np.array**(linguagens)
 - print(arrei[0:3])
 - print(arrei[:])
 - print(arrei[:4])
 - print(arrei[0:])
 - print(arrei[::-1])
 - print(arrei[:,2])
 - print(arrei[-1:-4:-1])
 - print(arrei[-1:-4:-2])
- Resultado da execução:
 - ['Python' 'Java' 'PHP']
 - ['Python' 'Java' 'PHP' 'C']
 - ['Python' 'Java' 'PHP' 'C']
 - ['Python' 'Java' 'PHP' 'C']
 - ['C' 'PHP' 'Java' 'Python']
 - ['Python' 'PHP']
 - ['C' 'PHP' 'Java']
 - ['C' 'Java']



Operações

Operações aritméticas

- Código:
 - import **numpy** as **np**
 - lista1 = [1, 2, 3, 4]
 - lista2 = [15, 16, 17, 18]
 - print(**np.add**(lista1, lista2))
 - print(**np.subtract**(lista1, lista2))
 - print(**np.multiply**(lista1, lista2))
 - print(**np.divide**(lista1, lista2))
 - print(**np.power**(lista1, lista2))
- Resultado da execução:
 - [16 18 20 22]
 - [-14 -14 -14 -14]
 - [15 32 51 72]
 - [0.06666667 0.125 0.17647059
0.22222222]
 - [
1 65536 129140163
68719476736]

Operações aritméticas

- Código:
 - import **numpy** as **np**
 - lista1 = [1, 2, 3, 4]
 - lista2 = [15, 16, 17, 18]
 - # mod() e remainder() retornam o resto da divisão.
 - print(**np.mod**(lista1, lista2))
 - print(**np.remainder**(lista1, lista2))
- Resultado da execução:
 - [1 2 3 4]
 - [1 2 3 4]

Operações aritméticas

- Código:
 - import **numpy** as **np**
 - lista1 = [1, 2, 3, 4]
 - lista2 = [15, 16, 17, 18]
 - lista3 = [-1, -2, 3, 4]
 - # divmod() retorna o quociente e o mod.
 - print(**np.divmod**(lista1, lista2))
 - # absolute() retorna os valores absolutos dos elementos.
 - print(**np.absolute**(lista3))
- Resultado da execução:
 - (array([0, 0, 0, 0]), array([1, 2, 3, 4]))
 - [1 2 3 4]

Operações aritméticas

- Código:
 - import **numpy** as **np**
 - **arrei1** = **np.array**([1, 2, 3, 4])
 - **arrei2** = **np.array**([15, 16, 17, 18])
 - **print**(**arrei1** + **arrei2**)
 - **print**(**arrei1** - **arrei2**)
 - **print**(**arrei1** * **arrei2**)
 - **print**(**arrei1** / **arrei2**)
 - **print**(**arrei1** ** **arrei2**)
 - **print**(**arrei1** % **arrei2**)
- Resultado da execução:
 - [16 18 20 22]
 - [-14 -14 -14 -14]
 - [15 32 51 72]
 - [0.06666667 0.125 0.17647059
0.22222222]
 - [
1 65536 129140163
68719476736]
 - [1 2 3 4]

Sumarização

- Código:
 - import **numpy** as **np**
 - lista1 = [1, 2, 3, 4]
 - lista2 = [15, 16, 17, 18]
 - # sum() retorna a soma todos os números.
 - print(**np.sum**([lista1, lista2]))
 - # Se houver axis = 1, será retornado um array com a soma de cada lista.
 - print(**np.sum**([lista1, lista2], axis = 1))
 - # cumsum() retorna soma parciais.
 - print(**np.cumsum**(lista1))
- Resultado da execução:
 - 76
 - [10 66]
 - [1 3 6 10]

Diferenças

- Código:
 - import **numpy** as **np**
 - lista = [10, 15, 25, 5]
 - # diff() retorna a diferença discreta.
 - # $15 - 10 = 5$
 - # $25 - 15 = 10$
 - # $5 - 25 = -20$
 - print(**np.diff**(lista))
- Resultado da execução:
 - [5 10 -20]

Diferenças

- Código:
 - import **numpy** as **np**
 - lista = [10, 15, 25, 5]
 - # diff() retorna a diferença discreta.
 - # 15 - 10 = 5
 - # 25 - 15 = 10
 - # 5 - 25 = -20
 - # 10 - 5 = 5
 - # -20 - 10 = -30
 - print(**np.diff**(lista, n = 2))
- Resultado da execução:
 - [5 -30]

Produtos

- Código:
 - import **numpy** as **np**
 - lista1 = [1, 2, 3, 4]
 - lista2 = [15, 16, 17, 18]
 - # $1 * 2 * 3 * 4 = 24$
 - print(**np.prod**(lista1))
 - # $1 * 2 * 3 * 4 * 15 * 16 * 17 * 18 = 1762560$
 - print(**np.prod**([lista1, lista2]))
- Resultado da execução:
 - 24
 - 1762560

Produtos

- Código:
 - import **numpy** as **np**
 - lista1 = [1, 2, 3, 4]
 - lista2 = [15, 16, 17, 18]
 - # Se houver axis = 1, será retornado um array com o produto de cada lista.
 - print(**np.prod**([lista1, lista2], axis = 1))
 - # cumprod() retorna produtos parciais.
 - print(**np.cumprod**(lista1))
- Resultado da execução:
 - [24 73440]
 - [1 2 6 24]



Questões de concursos

[CESGRANRIO 2021 Banco do Brasil – Agente de Tecnologia] Considere o código Python a seguir.

- `import numpy as np`
- `valorAplicado = np.array([5000, 6000, 7000, 8000])`
- `taxaJuros = np.array([1, 2, 3, 4])`
- `resultado = valorAplicado * taxaJuros`

Comentários

- Código:
 - `import numpy as np`
 - `valorAplicado = np.array([5000, 6000, 7000, 8000])`
 - `taxaJuros = np.array([1, 2, 3, 4])`
 - `resultado = valorAplicado * taxaJuros`
 - `print(resultado)`
- Resultado da execução:
 - `[5000 12000 21000 32000]`



Questões de concursos

[CESGRANRIO 2021 Banco do Brasil – Agente de TecnologiaAo executar o código acima, o valor final da variável resultado será

- [A]
 - 70000
- [B]
 - 260000
- [C]
 - [5000 12000 21000 32000]

Questões de concursos

[CESGRANRIO 2021 Banco do Brasil – Agente de TecnologiaAo executar o código acima, o valor final da variável resultado será

- [D]
 - [[5000 10000 15000 20000]
 - [6000 12000 18000 24000]
 - [7000 14000 21000 28000]
 - [8000 16000 24000 32000]]

Questões de concursos

[CESGRANRIO 2021 Banco do Brasil – Agente de TecnologiaAo executar o código acima, o valor final da variável resultado será

- [E]
 - [[5000 6000 7000 8000]
 - [10000 12000 14000 16000]
 - [15000 18000 21000 24000]
 - [20000 24000 28000 32000]]



Questões de concursos

[CESGRANRIO 2021 Banco do Brasil – Agente de TecnologiaAo executar o código acima, o valor final da variável resultado será

- **[C]**
 - **[5000 12000 21000 32000]**



Funções estatísticas

Média, mediana e moda

- Código:
 - import **numpy** as **np**
 - from **scipy** import **stats** as **st**
 - velocidades = [99, 86, 87, 88, 111, 86, 103, 87, 94, 78, 77, 85, 86]
 - x = **np.mean**(velocidades)
 - y = **np.median**(velocidades)
 - z = **st.mode**(velocidades)
 - print(x)
 - print(y)
 - print(z)
- Resultado da execução:
 - 89.76923076923077
 - 87.0
 - ModeResult(mode=array([86]), count=array([3]))

Média

- Código:
 - import **numpy** as **np**
 - from **scipy** import **stats** as **st**
 - lista = [[1, 2], [2, 2], [3, 3]]
 - arrei = **np.array**(lista)
 - print(arrei)
 - # Retorna a média do conjunto todo.
 - print(**np.mean**(arrei))
 - # 0: retorna a média de cada coluna.
 - print(**np.mean**(arrei, axis = 0))
 - # 1: retorna a média de cada linha.
 - print(**np.mean**(arrei, axis = 1))
- Resultado da execução:
 - [[1 2]
 - [2 2]
 - [3 3]]
 - 2.1666666666666665
 - [2. 2.33333333]
 - [1.5 2. 3.]

Mediana

- Código:
 - import **numpy** as **np**
 - from **scipy** import **stats** as **st**
 - lista = [[1, 2], [2, 2], [3, 3]]
 - arrei = **np.array**(lista)
 - print(arrei)
 - # Retorna a mediana do conjunto todo.
 - print(**np.median**(arrei))
 - # 0: retorna a mediana de cada coluna.
 - print(**np.median**(arrei, axis = 0))
 - # 1: retorna a mediana de cada linha.
 - print(**np.median**(arrei, axis = 1))
- Resultado da execução:
 - [[1 2]
 - [2 2]
 - [3 3]]
 - 2.0
 - [2. 2.]
 - [1.5 2. 3.]

Moda

- Código:
 - import **numpy** as **np**
 - from **scipy** import **stats** as **st**
 - lista = [[1, 2], [2, 2], [3, 3]]
 - arrei = **np.array**(lista)
 - print(arrei)
 - # Retorna a moda de cada coluna.
 - print(**st.mode**(arrei))
 - # 0: retorna a moda de cada c0luna.
 - print(**st.mode**(arrei, axis = 0))
 - # 1: retorna a moda de cada l1nha.
 - print(**st.mode**(arrei, axis = 1))
- Resultado da execução:
 - [[1 2 2]
 - [2 2 3]
 - [2 3 3]]
 - ModeResult(mode=array([[2, 2, 3]]), count=array([[2, 2, 2]]))
 - ModeResult(mode=array([[2, 2, 3]]), count=array([[2, 2, 2]]))
 - ModeResult(mode=array([[2],
 - [2],
 - [3]]), count=array([[2],
 - [2],
 - [2]]))

Variância e desvio padrão

- Código:
 - import **numpy** as **np**
 - velocidades = [99, 86, 87, 88, 111, 86, 103, 87, 94, 78, 77, 85, 86]
 - x = **np.mean**(velocidades)
 - y = **np.var**(velocidades)
 - z = **np.std**(velocidades)
 - print(x)
 - print(y)
 - print(z)
- Resultado da execução:
 - 89.76923076923077
 - 85.71597633136093
 - 9.258292301032677

Percentil

- Código:
 - import **numpy** as **np**
 - idades = [5, 31, 43, 48, 50, 41, 7, 11, 15, 39, 80, 82, 32, 2, 8, 6, 25, 36, 27, 61, 31]
 - a = **np.mean**(idades)
 - b = **np.median**(idades)
 - c = **st.mode**(idades)
 - print(a)
 - print(b)
 - print(c)
- Resultado da execução:
 - 32.38095238095238
 - 31.0
 - ModeResult(mode=array([31]), count=array([2]))

Percentil

- Código:
 - import **numpy** as **np**
 - idades = [5, 31, 43, 48, 50, 41, 7, 11, 15, 39, 80, 82, 32, 2, 8, 6, 25, 36, 27, 61, 31]
 - d = **np.percentile**(idades, 25)
 - e = **np.percentile**(idades, 50)
 - f = **np.percentile**(idades, 75)
 - print(d)
 - print(e)
 - print(f)
- Resultado da execução:
 - 11.0
 - 31.0
 - 43.0

Questões de concursos

[CESPE/CEBRASPE 2021 BANESE – Técnico Bancário III – Área de Informática – Desenvolvimento] No que se refere ao pacote NumPy do Python, julgue o item subsequente.

- Considerando a matriz de dados para resolução da questão, o código a seguir retorna as medianas dos valores encontrados na primeira coluna, {1,2,3}, e na segunda coluna, {2,2,3}.

$$\begin{pmatrix} 1 & 2 \\ 2 & 2 \\ 3 & 3 \end{pmatrix}$$

- import numpy
- y = numpy.array([[1, 2], [2, 2], [3, 3]])
- numpy.median(y, axis=0)

Comentários

- Código:
 - `import numpy`
 - `y = numpy.array([[1, 2], [2, 2], [3, 3]])`
 - `# Retorna a mediana do conjunto todo.`
 - `print(numpy.median(y))`
 - `# 0: retorna a mediana de cada coluna.`
 - `print(numpy.median(y, axis = 0))`
 - `# 1: retorna a mediana de cada linha.`
 - `print(numpy.median(y, axis = 1))`
- Resultado da execução:
 - `2.0`
 - `[2. 2.]`
 - `[1.5 2. 3.]`

Questões de concursos

[CESPE/CEBRASPE 2021 BANESE – Técnico Bancário III – Área de Informática – Desenvolvimento] No que se refere ao pacote NumPy do Python, julgue o item subsequente.

- Considerando a matriz de dados para resolução da questão, o código a seguir retorna as medianas dos valores encontrados na primeira coluna, {1,2,3}, e na segunda coluna, {2,2,3}.
 - import numpy
 - y = numpy.array([[1, 2], [2, 2], [3, 3]])
 - numpy.median(y, axis=0)
 - Gabarito: **CERTO**.

Questões de concursos

[CESPE/CEBRASPE 2022 TRT 8ª Região – Analista Judiciário – Tecnologia da Informação] No módulo NumPy, utilizado no aprendizado de máquina do Python 3,

- [A] o método `deviation()` permite encontrar o desvio padrão em uma matriz.
- [B] o método `percentile()` permite encontrar o percentil em uma matriz.
- [C] o método `half()` permite calcular a média em uma matriz.
- [D] o método `mean()` permite encontrar a mediana em uma matriz.
- [E] a propriedade `type()` retorna o tipo de dados do array de uma matriz.

Questões de concursos

[CESPE/CEBRASPE 2022 TRT 8ª Região – Analista Judiciário – Tecnologia da Informação] No módulo NumPy, utilizado no aprendizado de máquina do Python 3,

- [A] o método ~~deviation()~~ **std()** permite encontrar o desvio padrão em uma matriz.
- **[B] o método percentile() permite encontrar o percentil em uma matriz.**
- [C] o método ~~half()~~ **mean()** permite calcular a média em uma matriz.
- [D] o método ~~mean()~~ **median()** permite encontrar a mediana em uma matriz.
- [E] a propriedade **dtype**~~()~~ retorna o tipo de dados do array de uma matriz.