Atividade 3

August 23, 2022

1 Exercícios NumPy

Rhenan Dias - GU3009254

1.0.1 1 Exercícios NumPy

Importe NumPy como np

[ ]:
```
import numpy as np
```

Crie uma matriz de 10 zeros

[ ]:
```
np.zeros(10)
```

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

Crie uma matriz de 10 ones

[ ]:
```
np.ones(10)
```

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

Crie uma matriz de 10 cincos

[ ]:
```
np.full(10, 5)
```

[ ]:
```
array([5, 5, 5, 5, 5, 5, 5, 5, 5, 5])
```

Crie um array de inteiros de 10 até 50

[ ]:
```
np.arange(10, 51, 1)
```

[ ]:
```
array([10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26,
       27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43,
       44, 45, 46, 47, 48, 49, 50])
```

Crie um array dos numeros pares de 10 até 50

[ ]:
```
np.arange(10, 51, 2)
```

[ ]:
```
array([10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42,
       44, 46, 48, 50])
```

Criei uma matriz 3x3 com valores variando de 0 até 8

[ ]:
```
np.arange(0, 9).reshape(3, 3)
```

[ ]:
```
array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]])
```

Crie uma matriz identidade 3x3

[ ]:
```
np.identity(3)
```

[ ]:
```
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

Use NumPy para gerar números aleatórios entre 0 e 1

[ ]:
```
np.random.rand(1)
```

[ ]:
```
array([0.11413791])
```

Use Numpy para gerar um array de 25 números aleatórios tirados de uma distribuição normal

[ ]:
```
np.random.randn(25)
```

[ ]:
```
array([ 1.93155488,  0.61612077, -1.65548752, -2.0694288 , -0.00742863,
        2.415924  ,  0.65559223,  0.93842254, -1.0595365 ,  0.11281739,
       -1.70670812, -2.15749325,  0.83807689, -0.0112006 ,  0.31248313,
       -0.96131357,  1.51369729,  1.06936538, -1.17432876, -1.15098099,
       -0.66780422, -0.87928613, -0.99295714, -0.2542706 ,  0.21236484])
```

Crie a seguinte matriz:

[ ]:
```
np.arange(0.01, 1.01, 0.01)
```

[ ]:
```
array([0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1 , 0.11,
       0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.2 , 0.21, 0.22,
       0.23, 0.24, 0.25, 0.26, 0.27, 0.28, 0.29, 0.3 , 0.31, 0.32, 0.33,
       0.34, 0.35, 0.36, 0.37, 0.38, 0.39, 0.4 , 0.41, 0.42, 0.43, 0.44,
       0.45, 0.46, 0.47, 0.48, 0.49, 0.5 , 0.51, 0.52, 0.53, 0.54, 0.55,
       0.56, 0.57, 0.58, 0.59, 0.6 , 0.61, 0.62, 0.63, 0.64, 0.65, 0.66,
       0.67, 0.68, 0.69, 0.7 , 0.71, 0.72, 0.73, 0.74, 0.75, 0.76, 0.77,
       0.78, 0.79, 0.8 , 0.81, 0.82, 0.83, 0.84, 0.85, 0.86, 0.87, 0.88,
       0.89, 0.9 , 0.91, 0.92, 0.93, 0.94, 0.95, 0.96, 0.97, 0.98, 0.99,
```

```
       1.  ])
```

Crie um array de tamanho 20 igualmente espaçado entre 0 e 1.

[ ]:
```
np.linspace(0, 1, 20)
```

[ ]:
```
array([0.        , 0.05263158, 0.10526316, 0.15789474, 0.21052632,
       0.26315789, 0.31578947, 0.36842105, 0.42105263, 0.47368421,
       0.52631579, 0.57894737, 0.63157895, 0.68421053, 0.73684211,
       0.78947368, 0.84210526, 0.89473684, 0.94736842, 1.        ])
```

1.0.2 1.1 Indexação Numpy e Seleção

Agora você receberá algumas matrizes e será solicitado a replicar as saídas resultantes da matriz:

[ ]:
```
mat = np.arange(1,26).reshape(5,5)
mat
```

[ ]:
```
array([[ 1,  2,  3,  4,  5],
       [ 6,  7,  8,  9, 10],
       [11, 12, 13, 14, 15],
       [16, 17, 18, 19, 20],
       [21, 22, 23, 24, 25]])
```

[ ]:
```
mat[2:, 1:,]
```

[ ]:
```
array([[12, 13, 14, 15],
       [17, 18, 19, 20],
       [22, 23, 24, 25]])
```

[ ]:
```
mat[3, 4]
```

[ ]:
```
20
```

[ ]:
```
mat[0:3, 1:2]
```

[ ]:
```
array([[ 2],
       [ 7],
       [12]])
```

[ ]:
```
mat[4, 0:5]
```

[ ]:
```
array([21, 22, 23, 24, 25])
```

[ ]:
```
mat[3:5, 0:5]
```

[ ]:
```
array([[16, 17, 18, 19, 20],
       [21, 22, 23, 24, 25]])
```

1.0.3 1.1.1 Agora faça o seguinte

Obter a soma de todos os valores no "mat"

[ ]:
```
np.sum(mat)
```

[ ]:
```
325
```

Obter o desvio padrão dos valores em mat

[ ]:
```
np.std(mat)
```

[ ]:
```
7.211102550927978
```

Obter a soma de todas as colunas em mat

[ ]:
```
np.sum(mat, axis=0)
```

[ ]:
```
array([55, 60, 65, 70, 75])
```