

Task-6

 $x = np.zeros((3, n), dtype=int)$

08:00

NumPy:

 $x = np.ones((4, 5))$

09:00

- in Python as np

 $x = np.full((4, 3), 5)$

- take less time

 $\Rightarrow [[5, 5, 5]]$

المصفوفة

10:00

ex. nD array (x)

 $\begin{bmatrix} 5 & 5 & 5 \\ 5 & 5 & 5 \\ 5 & 5 & 5 \end{bmatrix}$

- Time: 0.09 ثانية

 $x = np.eye(5)$

Diagonal = 1

5x5

11:00

- Create array

 $x = np.diag([1, 2, 3, 4, 5])$

12:00

 $x = np.array([1, 2, 3])$

01:00

 $x.dtype \Rightarrow int64$ $x.shape \Rightarrow (5,)$ $y = np.arange([0, 1, 2, 3, 4])$ $y.shape \Rightarrow (5,)$ $y.size \Rightarrow 5$ $x = np.arange([0, 1, 2, 3, 4], dtype=int64)$

print(x)

print('dtype:', x.dtype)

 $\Rightarrow [0, 1, 2, 3, 4]$

dtype: int64

 $x = np.array([1, 2, 3])$

np.save('my_array', x)

 $y = np.load('my_array.npy')$ print(y) $\Rightarrow [1, 2, 3]$ $x = np.arange(10)$

02:00

 $\Rightarrow [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]$ $x = np.arange(1, 14, 3)$

03:00

1 4 7 10 13

 $x = np.linspace(0, 25, 10)$

04:00

 $[0, 2.7777777777777777, 5.555555555555555, 8.333333333333333, 11.111111111111111, 13.888888888888889, 16.666666666666667, 19.444444444444443, 22.222222222222225, 25.0]$ $x = np.linspace(0, 25, 10, endpoint=False)$

05:00

 $[0, 2.5, 5, 7.5, 10, 12.5, 15, 17.5, 20, 22.5]$ $x = np.arange(20)$

06:00

 $x = np.reshape(x, (4, 5))$ $\begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 & 9 \\ 10 & 11 & 12 & 13 & 14 \\ 15 & 16 & 17 & 18 & 19 \end{bmatrix}$

07:00

 $x = np.reshape(x, (10, 2))$ $\begin{bmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \\ 6 & 7 \\ 8 & 9 \\ 10 & 11 \\ 12 & 13 \\ 14 & 15 \\ 16 & 17 \\ 18 & 19 \end{bmatrix}$

08:00

٢٩ كيهك ١٧٣٨ ق

08:00

- $Y = \text{np.arange}(20, \text{reshape}(1, 2))$ - $X = \text{np.linspace}(0, 25, 10, \text{endpoint}=\text{False}, \text{reshape}(5, 2))$

09:00

$$\begin{pmatrix} 0 & 5 \\ 10 & 15 \end{pmatrix}$$

10:00

- $X = \text{np.vander}, \text{vander}((3, 3))$ - $X = \text{np.vander}, \text{vanderint}(4, 15, (3, 2))$

11:00

$$\begin{bmatrix} 0 & 5 \\ 12 & 14 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 5 \\ 12 & 14 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 5 \\ 12 & 14 \end{bmatrix}$$

12:00

- $X = \text{np.random.normal}(0, 0.1, \text{size}=(100, 100))$

01:00

$$\begin{matrix} \text{pr} & X, \text{mean} \end{matrix}$$

02:00

$$X, \text{std}$$

$$X, \text{max}$$

$$X, \text{min}$$

03:00

$$\text{Posit} : (X > 0), \text{sum}()$$

$$\text{neg} : (X < 0), \text{sum}()$$

04:00

- $X = \text{np.array}((1, 2, 3))$

05:00

$$X[0] \rightarrow 1 \quad X[-1] = 3$$
- $X[3] = 20$ هو خارج الحدود

06:00

- $X = \text{np.arange}(1, 10, \text{reshape}(3, 3))$ - $\text{np.delete}(Y, 5, \text{axis}=0)$ دليل 5

07:00

- $W = \text{np.insert}(Y, 1, [4, 5, 6], \text{axis}=0)$

08:00

$$\text{صف الثاني جديد}$$
- $V = \text{np.insert}(Y, 1, 5, \text{axis}=1)$

8

Saturday
January

٥ جماد آخر ١٤٤٣ هـ

السبت
يناير

٢٠٢٢



٣٠ كيهك ١٧٣٨ ق

- z = [3 4

5 6)

z = np.vstack((x, y))

C 12

3 4

5 6

w = np.hstack((x, x.reshape(2, 1)))

[3 4]

5 6 2)

x = [1 2]

جافه حشمت

- x = np.arange(1, 20).reshape((4, 5))

z = x[1:4, 2:5] → 8 9 16

= x[1, 2:5] → 13 14 15

z = x[1, 2] all row index = 2

all 3 8 13 18

z = x[1, 2:3] → 3 8

x = np.array([1, 2, 3, 4, 5])

y = [6, 7, 2, 8, 9]

⇒ np.intersect1d(x, y)

= [2 4]

⇒ np.setdiff1d(x, y)

= [1 3 5]

⇒ np.union1d(x, y)

= 1 2 3 4 5 6 7 8

* - np.random.randn(1, 1, size=(4,))

C 7 2 4 5 9 3 8 7 9 1

np.sort(x) → 2 3 4 5 9

x → 1 2 3 4 5 6 7 8

np.sort(np.unique(x))

1 2 3 4 5 7 8 9

np.random.randn(1, 1, size=(5, 5))

- np.sort(x, axis=0)

- print(x) = print(np.sort(x, axis=0))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))

np.random.randn(1, 1, size=(5, 5))