

# Data Analysis Tools with Numpy - Exercises

ให้ Import NumPy Library ให้อยู่ในตัวแปร np

```
In [1]: 1 import numpy as np
```

จงสร้าง zeros array ที่มีสมาชิก 10 ตัว

```
In [4]: 1 a = np.zeros(10)
        2 a
```

```
Out[4]: array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

จงสร้าง ones array ที่มีสมาชิก 10 ตัว

```
In [5]: 1 b = np.ones(10)
        2 b
```

```
Out[5]: array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

จงสร้าง array บรรจุ เลข 5 จำนวน 10 ตัว

```
In [8]: 1 c = np.full(10, 5)
        2 c
```

```
Out[8]: array([5, 5, 5, 5, 5, 5, 5, 5, 5, 5])
```

จงสร้าง array บรรจุจำนวนเต็ม ที่มีค่าตั้งแต่ 10 - 50

```
In [10]: 1 d = np.arange(10, 51)
        2 d
```

```
Out[10]: 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])
```

จงสร้าง array บรรจุจำนวนเต็มคู่ ที่มีค่าตั้งแต่ 10 - 50

```
In [11]: 1 e = np.arange(10, 51, 2)
          2 e
```

```
Out[11]: array([10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42,
               44, 46, 48, 50])
```

**จงสร้าง Matrix ขนาด 3x3 บรรจุมหาชิกตัวเลขจำนวนเต็ม ที่มีค่าตั้งแต่ 0 ถึง 8**

```
In [13]: 1 f = np.arange(0, 9).reshape(3, 3)
          2 f
```

```
Out[13]: array([[0, 1, 2],
               [3, 4, 5],
               [6, 7, 8]])
```

**จงสร้าง identity matrix ที่มีขนาด 3x3**

```
In [14]: 1 g = np.eye(3)
          2 g
```

```
Out[14]: array([[1., 0., 0.],
               [0., 1., 0.],
               [0., 0., 1.]])
```

**จงใช้ NumPy เพื่อสร้างจำนวนสุ่มตัวเลข (random) ที่มีค่าระหว่าง 0 ถึง 1**

```
In [16]: 1 from numpy import random as rd
          2 rd.rand(1)
```

```
Out[16]: array([0.91444219])
```

**จงใช้ NumPy เพื่อสร้าง array จำนวนสุ่มตัวเลข (random) 25 ค่า ที่มีค่ากระจายตัวแบบ standard normal distribution**

```
In [41]: 1 np.random.standard_normal(25)
```

```
Out[41]: array([ 0.0363809 , -1.55119901,  1.15428445, -0.01382339,  0.3254269 ,
                 0.87389326, -0.51897591,  0.17660467, -0.64441773,  0.92969058,
                -0.57526329,  0.13641662, -0.87485842, -0.34536432,  0.69794488,
                -0.77187005, -0.91667344,  1.80763489, -0.65312835, -0.09077742,
                -1.03593639,  0.2561931 , -1.01226624,  0.69680724,  0.59291322])
```

**จงสร้าง Matrix ต่อไปนี้**

```
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 ]],
```

```
In [22]: 1 h = np.arange(0.01, 1.01, .01).reshape(10, 10)
        2 h
```

```
Out[22]: 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.  ]])
```

จงสร้าง array ด้วยคำสั่ง linspace () ที่มีจำนวนสมาชิก 20 ตัว ที่มีค่าเรียงตั้งแต่ 0 ถึง1

```
In [25]: 1 i = np.linspace(0, 1, 20)
        2 i
```

```
Out[25]: 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.          ])
```

## Numpy Indexing and Selection

ให้นำ matrices (mat) ที่กำหนดให้ ใช้ประยุกต์เพื่อตอบคำถามข้อ a-h

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

```
Out[26]: 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]])
```

a.

จงสร้าง Matrix ให้มีค่าดังภาพ จาก matrices (mat) ที่กำหนดให้

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

```
In [28]: 1 mat[ 2 : , 1 : ]
```

```
Out[28]: array([[12, 13, 14, 15],
               [17, 18, 19, 20],
               [22, 23, 24, 25]])
```

**b.**

จงสร้าง Matric ให้มีค่าดังภาพ จาก matrices (mat) ที่กำหนดให้

20

```
In [33]: 1 mat[[3], [4]]
```

```
Out[33]: array([20])
```

**c.**

จงสร้าง Matric ให้มีค่าดังภาพ จาก matrices (mat) ที่กำหนดให้

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

```
In [30]: 1 mat[ 0 : 3 , [1]]
```

```
Out[30]: array([[ 2],
               [ 7],
               [12]])
```

**d.**

จงสร้าง Matric ให้มีค่าดังภาพ จาก matrices (mat) ที่กำหนดให้

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

```
In [36]: 1 mat[[4], : ]
```

```
Out[36]: array([21, 22, 23, 24, 25])
```

**e.**

```
In [37]: 1 # WRITE CODE HERE THAT REPRODUCES THE OUTPUT OF THE CELL BELOW
          2 # BE CAREFUL NOT TO RUN THE CELL BELOW, OTHERWISE YOU WON'T
          3 # BE ABLE TO SEE THE OUTPUT ANY MORE
          4
```

```
Out[37]: array([[16, 17, 18, 19, 20],
               [21, 22, 23, 24, 25]])
```

```
In [38]: 1 mat[3 : , : ]
```

```
Out[38]: array([[16, 17, 18, 19, 20],
               [21, 22, 23, 24, 25]])
```

**f. ให้ทำการ sum ทุกค่าใน mat**

```
In [35]: 1 mat.sum()
```

```
Out[35]: 325
```

**g. จงหาค่า standard deviation ของค่าที่อยู่ใน mat**

```
In [39]: 1 mat.std()
```

```
Out[39]: 7.211102550927978
```

**h. ให้ทำการ sum ทุกๆ columns ใน mat**

```
In [40]: 1 mat.sum(axis = 0)
```

```
Out[40]: array([55, 60, 65, 70, 75])
```

**----- ภาพนามยปัญญา ปัญญาที่เกิดจากการลงมือทำ! -----**