

Numpy

เรียนรู้การใช้งาน Python ร่วมกับ Numpy

ติดตามผู้เชี่ยวชาญ ผ่านช่องทางยูทูป



https://www.youtube.com/channel/UCQ1r_4x-P-fETLIU4pqf98w



<https://www.facebook.com/KongRuksiamTutorial/>

หรือสแกน QR CODE



เรียนเนื้อหา Numpy ได้ที่

<https://bit.ly/35HUECe>

List & Array (ndarray)

- Array สมาชิกใน Array ต้องมีชนิดข้อมูลเหมือนกัน
- List สมาชิกมีชนิดข้อมูลต่างกันได้
- Array ขนาดที่แน่นอนเปลี่ยนแปลงขนาดไม่ได้
- List มีขนาดที่ยืดหยุ่นกว่า

Array คือการนำข้อมูลมาอยู่ในกลุ่ม
เดียวกัน โดยสมาชิกภายใน Array
ต้องมี**ชนิดข้อมูลเหมือนกัน**

1D Array

3	2
---	---

2D Array

1	0	1
3	4	1

3D Array

1	7	9
5	9	3
7	9	9

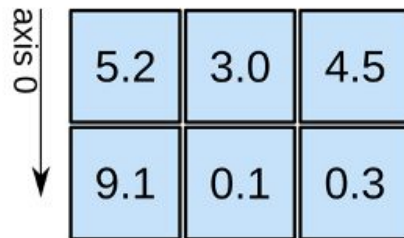
1D array



axis 0 →

shape: (4,)

2D array

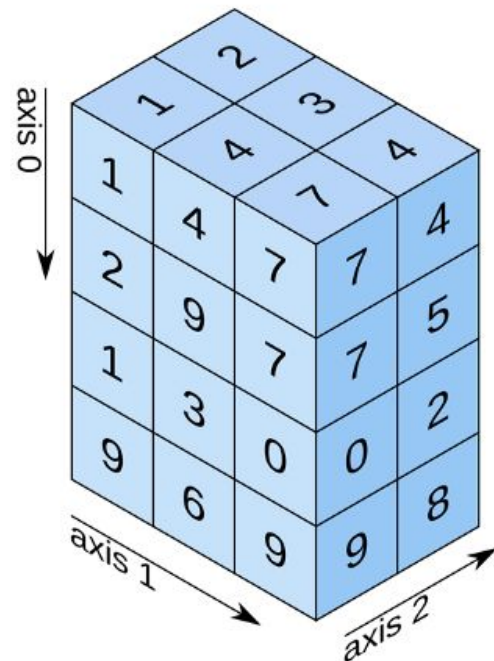


axis 0 ↓

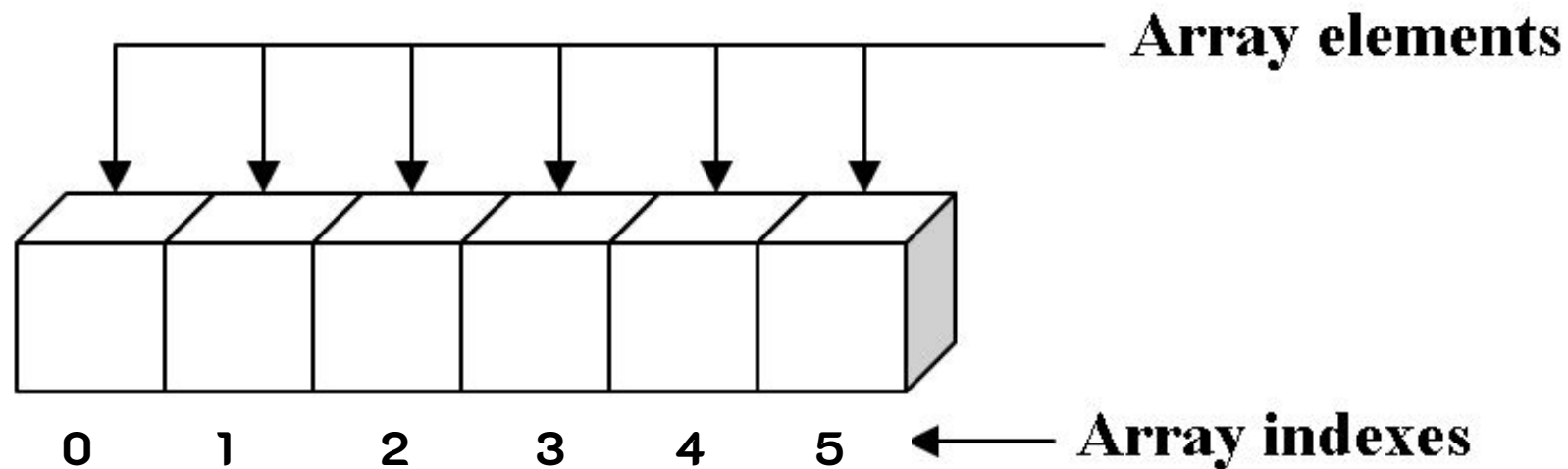
axis 1 →

shape: (2, 3)

3D array



shape: (4, 3, 2)



One-dimensional array with six elements

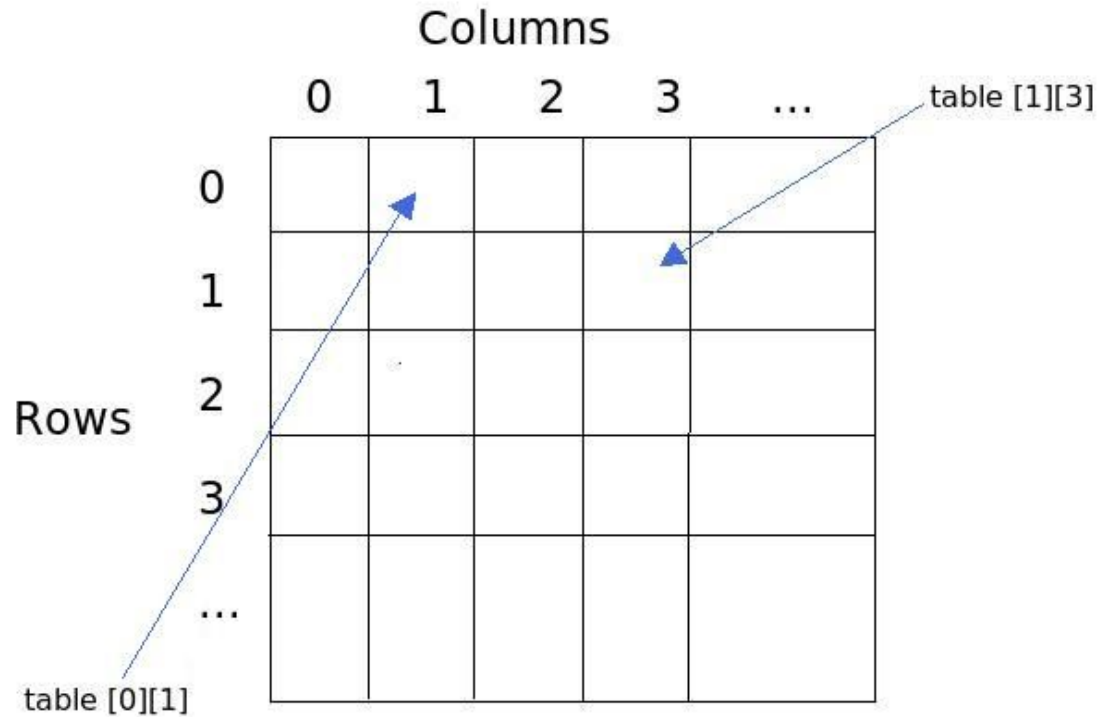
Length = 5

`a`	`b`	`c`	`d`	`e`
-----	-----	-----	-----	-----

index --> [0] [1] [2] [3] [4]

[-5] [-4] [-3] [-2] [-1] <-- negative index

A 2-Dimensional Array: table



Col 0

Col 1

ROW 0

1

2

ROW 1

3

4

ROW 2

5

6

-2

-1

Col 0

Col 1

ROW 0

-3

1

2

ROW 1

-2

3

4

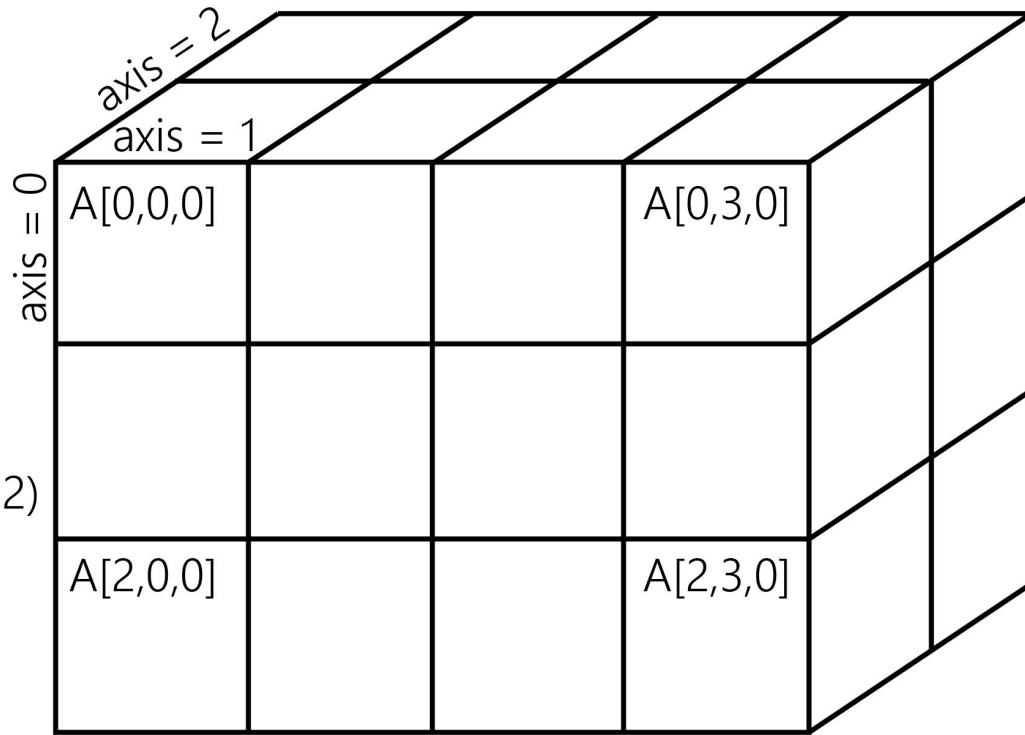
ROW 2

-1

5

6

ndarray
ndim = 3
shape = (3, 4, 2)



ชนิดข้อมูลหลักๆ ที่ใช้ใน Numpy

1. Integer
2. Float
3. String
4. Boolean
5. Complex
6. Object

Array 2 มิติ =
Matrix

การเข้าถึงข้อมูลสมาชิกใน Array 2 มิติ

```
A=np.array([-2,5,6],[5,2,7])
```

A[แถวที่ , คอลัมน์ที่]

$$A = \begin{bmatrix} -2 & 5 & 6 \\ 5 & 2 & 7 \end{bmatrix}$$

3 columns

2 rows

การเข้าถึงข้อมูลสมาชิกใน Array 2 มิติ

Rows	Columns		
	0	1	2
0	10	5	3
1	2	1	9
2	11	6	7
3	0	4	3

	Column 1	Column 2	Column 3	Column 4
Row 1	x[0][0]	x[0][1]	x[0][2]	x[0][3]
Row 2	x[1][0]	x[1][1]	x[1][2]	x[1][3]
Row 3	x[2][0]	x[2][1]	x[2][2]	x[2][3]

เมตริกซ์จัตุรัส (Square Matrix)

คือ จำนวนแถวและคอลัมน์เท่ากัน

เมตริกซ์ศูนย์ (Zero Matrix)

คือ สมาชิกทุกตัวเป็น 0

เมตริกซ์เอกลักษณ์ (Identity Matrix)

สมาชิกเส้นทแยงมุมมีค่าเป็น 1
ที่เหลือเป็น 0

Board Casting

- ขนาดและมิติของ Array 2 ตัวไม่สอดคล้องกัน

```
x= np.array([1,2,3,4,5,6])
```

```
y= np.array([1,2])
```

```
x+y
```

```
x = np.array([[1,2],[3,4],[5,6]])
```

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

```
x+y
```

ขนาดของ Array ต่างกัน / มิติก็ต่างกัน

Board Casting

- ขนาดเท่ากัน แต่มิติ Array 2 ตัวไม่เท่ากัน

```
x = np.array([[1,2],[3,4],[5,6]])  
y = np.array([1,2])  
x+y
```

ทำงานได้โดยการ Broadcast Array ที่มีขนาดเล็กกว่าถูกทำซ้ำ
เช่น array y มีขนาด 1 มิติจะถูก broadcast ไปเป็น array 2 มิติ
เพื่อให้สอดคล้องกับ array x เทียบมิติจากขวาไปซ้าย

การดำเนินการคณิตศาสตร์ในกรณี Array ขนาดไม่เท่ากัน

```
x = np.array([[1,2],[3,4],[5,6]]) # (3, 2)
y = np.array([2]) # (1, )
x+y
```

$$\begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 2 & 2 \\ 2 & 2 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} = \begin{bmatrix} 2+1 & 2+2 \\ 2+3 & 2+4 \\ 2+5 & 2+6 \end{bmatrix}$$

การดำเนินการคณิตศาสตร์ในกรณี Array ขนาดไม่เท่ากัน

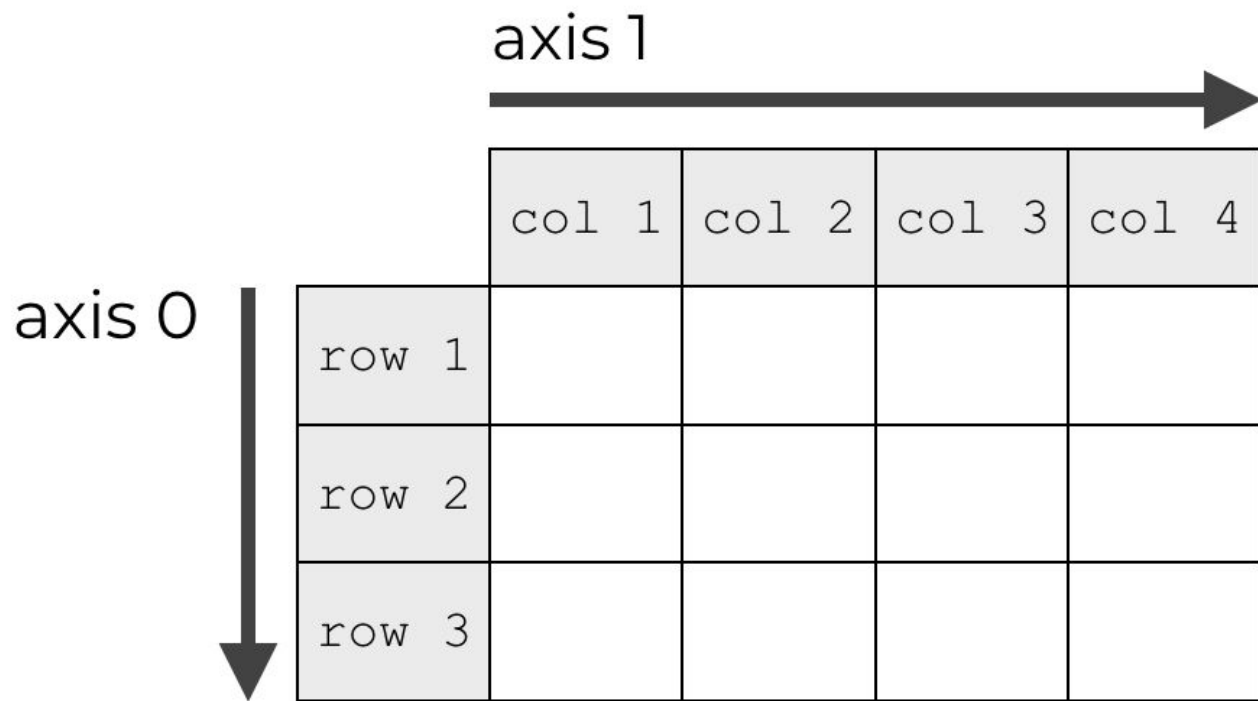
```
x = np.array([[1,2],[3,4],[5,6]])# (3, 2)
y = np.array([10,20]) # (2,1)
x+y
```

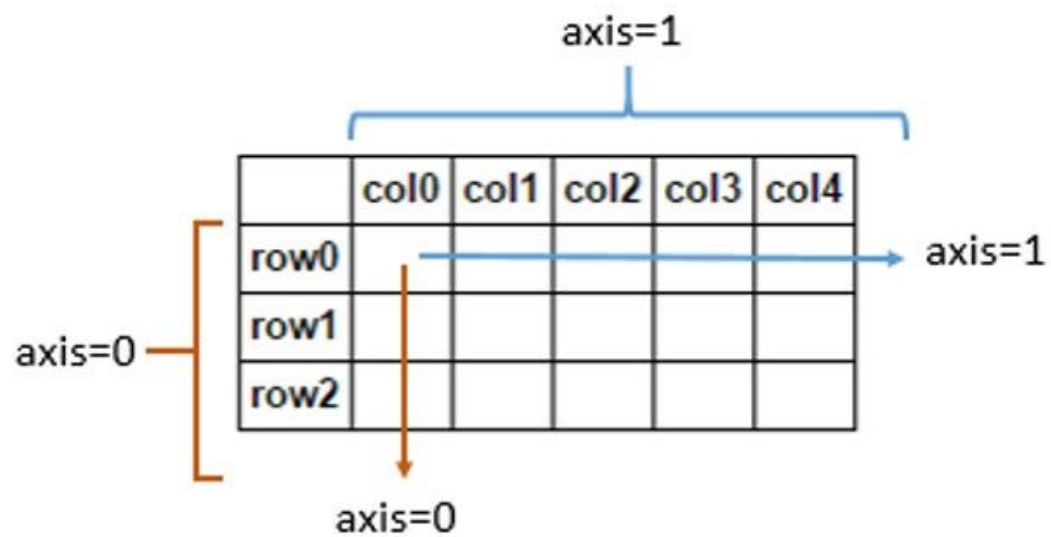
$$\begin{bmatrix} 10 & 20 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} = \begin{bmatrix} 10 & 20 \\ 10 & 20 \\ 10 & 20 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} = \begin{bmatrix} 10 + 1 & 20 + 2 \\ 10 + 3 & 20 + 4 \\ 10 + 5 & 20 + 6 \end{bmatrix}$$

การดำเนินการคณิตศาสตร์ในกรณี Array ขนาดไม่เท่ากัน

```
x = np.array([[1,2],[3,4],[5,6]])# (3, 2)
y = np.array([[10],[20],[30]])# (3, 1)
x+y
```

$$\begin{bmatrix} 10 \\ 20 \\ 30 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} = \begin{bmatrix} 10 & 10 \\ 20 & 20 \\ 30 & 30 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} = \begin{bmatrix} 10 + 1 & 10 + 2 \\ 20 + 3 & 20 + 4 \\ 30 + 5 & 30 + 6 \end{bmatrix}$$





dot product

$$\begin{array}{c} \vec{a}_1 \rightarrow \\ \vec{a}_2 \rightarrow \end{array} \begin{array}{c} \vec{b}_1 \quad \vec{b}_2 \\ \downarrow \quad \downarrow \end{array} \begin{array}{c} A \quad B \end{array} \cdot \begin{array}{c} \begin{bmatrix} 3 & 3 \\ 5 & 2 \end{bmatrix} \\ C \end{array} = \begin{array}{c} \begin{bmatrix} \vec{a}_1 \cdot \vec{b}_1 & \vec{a}_1 \cdot \vec{b}_2 \\ \vec{a}_2 \cdot \vec{b}_1 & \vec{a}_2 \cdot \vec{b}_2 \end{bmatrix} \\ C \end{array}$$