

# Numpy In Python

Treepop Wisanwanichthan



Software Park Thailand  
</Code Camp>

# ແບະນຳຜູ້ສອບ

- ຕຣີກພ ວິຄາລາວນິຈຮັງມູ
- ປະວັດຕາການສຶກຫາ
  - ປະລິບປະງາດ ສາຂາວິທະຍາຄາສົດ ເອກວິທະຍາຄາສົດຄອມພິວເຕອົງແລະຄນືຕຄາສົດ (Distinction) From University of New South Wales, Australia
  - ນັກເຮັດວຽກ Thai-Australian Scholarship



# Outline

## 1. แนะนำ Numpy

- Numpy คืออะไร
- Numpy ใช้งานประเภทไหน

## 2. Installing Numpy ติดตั้ง numpy

- ใช้ Conda ติดตั้ง numpy
- ติดตั้งแบบเลือกเวอร์ชัน
- ถอนการติดตั้ง, อัพเดท numpy

## 3. Numpy และ Array เมื่องต้น

- แนะนำ Built-in Array ใน Python
- การใช้งาน Built-in Array ฉบับย่อ
- แนะนำ numpy array และฟังก์ชันเบื้องต้น

# Outline

## 4. Numpy array และ List

- เขียนโปรแกรมการแปลงเซนติเมตรเป็นนิ้ว
- โดยเปรียบเทียบ List, List Comprehension, Map Lambda และ Numpy Array
- เปรียบเทียบความเร็ว Python List กับ Numpy Array

## 5. การใช้งาน numpy

- array 1D, array 2D
- Vector, Matrix
- พัฒนาชั้นต่างๆใน numpy

## 6. Operations and Indexing การดำเนินการและการใช้ Index

- การดำเนินการใน numpy array
- การแก้ไข/เปลี่ยนแปลง สมาชิกใน array 1D, array 2D
- การเข้าถึงสมาชิกจาก index รวมถึงการใช้ slicing

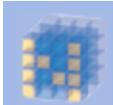
# Intro to Numpy

# ແລະທຳ numpy

# ແນະນຳ Numpy



Software Park Thailand  
</Code Camp>



## NumPy

[NumPy.org](https://www.numpy.org)

## NumPy

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- a powerful N-dimensional array object
- sophisticated (broadcasting) functions
- tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

NumPy is licensed under the [BSD license](#), enabling reuse with few restrictions.

# แนะนำ Numpy

## นัมพี้

จากวิกิพีเดีย สารานุกรมเสรี



บทความนี้อาจต้องการพิสูจน์อักษร ในด้านไวยากรณ์ รูปแบบการเขียน การเรียนเรียง คุณภาพ หรือการสะกด คุณสามารถช่วยพัฒนาบทความได้



บทความนี้ไม่มีการอ้างอิงจากแหล่งที่มาใด กรุณารายปรับปรุงบทความนี้ โดยเพิ่มการอ้างอิงแหล่งที่มาที่แน่ เช่น ก็อปปี้เนื้อความที่ไม่มีแหล่งที่มาอาจถูกคัดค้านหรือลบออก

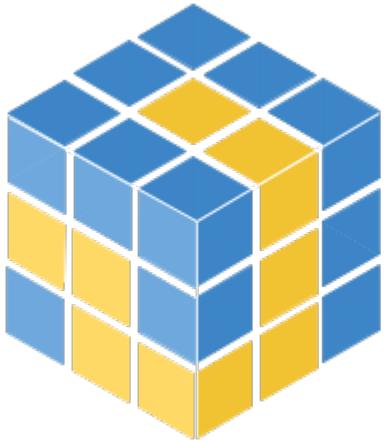
นัมพี้ เป็นส่วนขยายของภาษา Python เพื่อจัดการ **เมตริกซ์** หรืออาร์เรย์ คล้ายมิติร่วมกับฟังก์ชันทางคณิตศาสตร์ที่ทำงานบนอาร์เรย์เหล่านี้

นอกจากนั้นคลังซอฟต์แวร์โอเพนซอร์สและเสรีนี้มีฟังก์ชันหลายอย่างที่ทำให้ทำได้ โดยเฉพาะอย่างยิ่งในการสร้างอาร์เรย์โดยตรงจากไฟล์ หรือเพื่อบันทึกอาร์เรย์ในไฟล์ เพื่อจัดการเวลาเตอร์ เมทริกซ์ และพหุนาม นัมพี้เป็นฐานเพิ่มเติมของ SciPy ซึ่งเป็นコレคชันของห้องสมุด Python เกี่ยวกับการคำนวณทางวิทยาศาสตร์

# ແບບ Numpy



Software Park Thailand  
/>Code Camp>



# NumPy

# Installing Numpy

# ติดตั้ง numpy

# ติดตั้ง Numpy



Software Park Thailand  
</Code Camp>

	Anaconda Prompt			
m2w64-gcc-libs-core	5.3.0	7		
m2w64-gmp	6.1.0	2		
m2w64-libwinpthread-git	5.0.0-4634.697f757	2		
markdown	3.1.1	py36_0		
markupsafe	1.1.1	py36he774522_0		
matplotlib	1.5.3	pypi_0	pypi	
mccabe	0.6.1	py36_1		
menuinst	1.4.16	py36he774522_0		
mistune	0.8.4	py36he774522_0		
mkl	2018.0.3	1		
msgpack	0.6.2	pypi_0	pypi	
msgpack-numpy	0.4.4.3	py_0		
msgpack-python	0.5.6	py36he980bc4_1		
msys2-conda-epoch	20160418	1		
multidict	4.5.2	pypi_0	pypi	
murmurhash	1.0.2	pypi_0	pypi	
navigator-updater	0.2.1	py36_0		
nbconvert	5.4.1	pypi_0	pypi	
nbformat	4.4.0	py36h3a5bc1b_0		
nbsphinx	0.4.2	pypi_0	pypi	
networkx	1.11	pypi_0	pypi	
nltk	3.4.4	py36_0		
notebook	6.0.1	py36_0		
numpy	1.17.1	pypi_0	pypi	
oauth2client	4.1.3	pypi_0	pypi	
olefile	0.46	py36_0		
opencv-python	4.0.0.21	pypi_0	pypi	
openssl	1.1.1d	he774522_0		
packaging	18.0	pypi_0	pypi	
pandas	0.24.2	py36ha925a31_0		

- เช็คเวอร์ชัน

## Conda list

# ติดตั้ง Numpy



Software Park Thailand  
</Code Camp>

## - ติดตั้ง numpy

### Conda install numpy

```
Anaconda Prompt - conda install numpy

## Package Plan ##

environment location: C:\Users\tree3\Anaconda3

added / updated specs:
- numpy

The following packages will be downloaded:

      package          build
ca-certificates-2020.1.1           0      165 KB
h5py-2.10.0                         py36h5e291fa_0    992 KB
openssl-1.1.1d                      he774522_3     5.7 MB
pip-20.0.2                           py36_1        1.9 MB
tqdm-4.42.0                          py_0        56 KB
                                           Total:    8.8 MB

The following NEW packages will be INSTALLED:

h5py          pkgs/main/win-64::h5py-2.10.0-py36h5e291fa_0
keras-base    pkgs/main/win-64::keras-base-2.2.4-py36_0
pip           pkgs/main/win-64::pip-20.0.2-py36_1
tqdm          pkgs/main/noarch::tqdm-4.42.0-py_0

The following packages will be UPDATED:

ca-certificates          2019.8.28-0 --> 2020.1.1-0
openssl                  1.1.1d-he774522_0 --> 1.1.1d-he774522_3

Proceed ([y]/n)?
```

# ติดตั้ง Numpy



Software Park Thailand  
</Code Camp>

ถอนการติดตั้ง

- Conda uninstall numpy

ติดตั้งแบบระบุเวอร์ชัน

- Conda install numpy==1.17.1

อัพเดทเวอร์ชันเป็นเวอร์ชันล่าสุด

- Conda update numpy

# ติดตั้ง Numpy



Software Park Thailand  
</Code Camp>

https://pypi.org/project/numpy/#history

Search projects

Help    Donate    Log in    Register

numpy 1.18.1

pip install numpy   

Latest version   

Released: Jan 7, 2020

NumPy is the fundamental package for array computing with Python.

Navigation

- Project description
- Release history**
- Download files

Project links

- Homepage
- Bug Tracker
- Documentation
- Source Code
- Download

Release history

THIS VERSION    1.18.1 Jan 7, 2020

1.18.0 Dec 22, 2019

1.18.0rc1 PRE-RELEASE Dec 6, 2019

1.17.5 Jan 1, 2020

[Release notifications](#)

เว็บไซต์เช็คเวอร์ชันก่อนๆ

# Numpy และ Array เบื้องต้น



# Numpy และ Array เบื้องต้น

Numpy เป็น package ที่ไว้สร้าง/ดำเนินการ array และ matrix โดยเฉพาะ

Array เป็น Data Structure ประเภทหนึ่งที่จะต้องเก็บข้อมูลประเภทเดียวกันเท่านั้น



# Numpy และ Array เบื้องต้น

ข้อควรรู้:

Array เป็น data structure ยอดฮิตใน Java, C/C++ etc  
แต่ไม่ยอดฮิตและไม่ได้ถูกพูดถึงมากใน python

เวลาที่คนที่ไม่ได้เขียน python พูดถึง array ใน python  
ส่วนใหญ่จะหมายถึง List ซึ่งเป็นที่นิยมใน python มากกว่า



# Numpy และ Array เบื้องต้น

Array สามารถสร้างโดยไม่ใช้ numpy ได้  
แต่ต้อง import array ใน python



```
1 import array as arr
2 x = arr.array('d', [1.1, 2.2, 3.3])
3
4 print(x)
```

```
array('d', [1.1, 2.2, 3.3])
```



# Numpy และ Array เบื้องต้น

## Array ไม่สามารถบรรจุข้อมูลต่างชนิดได้

```
▶ 1 import array as arr
  2 x = arr.array('d', [1.1, 2.2, 'String'])
  3
  4 print(x)
```

**TypeError**

Traceback (most recent call last)

```
<ipython-input-7-bfc21966525e> in <module>
      1 import array as arr
----> 2 x = arr.array('d', [1.1, 2.2, 'String'])
      3
      4 print(x)
```

**TypeError:** must be real number, not str



# Numpy และ Array เบื้องต้น

```
▶ 1 import array as arr
  2 x = arr.array('d', [1.1, 2.2, 3.3])
  3
  4 print(x)
```

array('d', [1.1, 2.2, 3.3])

**'d'** คือรหัสการเก็บข้อมูล = float

**'i'** = int



# Numpy และ Array เบื้องต้น

สามารถเข้าถึงข้อมูลได้เหมือน List



```
1 import array as arr
2 x = arr.array('d', [1.1, 2.2, 3.3])
3
4 print(x[0])
```

1.1



# Numpy และ Array เบื้องต้น

**append()** = เพิ่มข้อมูลตัวเดียว

**extend()** = เพิ่มข้อมูลหลายตัว

```
▶ 1 import array as arr  
2 x = arr.array('d', [1.1, 2.2, 3.3])  
3  
4 print(x[0])
```

1.1

```
▶ 1 x.append(10.0)  
2 print(x)
```

array('d', [1.1, 2.2, 3.3, 10.0])

```
▶ 1 x.extend([15.5,16.6,17.7])  
2 print(x)
```

array('d', [1.1, 2.2, 3.3, 10.0, 15.5, 16.6, 17.7])



# Numpy และ Array เบื้องต้น

ข้อควรรู้:

Array ที่เป็น built-in function ใน python ไม่เป็นที่นิยม  
เนื่องจาก ไม่มีความยืดหยุ่นเหมือน list และช้า

ดังนั้น ในการคำนวณทางคณิตศาสตร์ วิทยาศาสตร์ เช่น Machine Learning จะนิยมใช้ array ใน numpy ซึ่งเร็วกว่ามาก



# Numpy และ Array เบื้องต้น

ใช้ np.array([ ]) ในการประกาศใช้ array

```
▶ 1 import numpy as np  
2  
3 x = np.array([1,2,3,4,5])  
4 print(x)
```

[1 2 3 4 5]

```
▶ 1 print(type(x))
```

<class 'numpy.ndarray'>



# Numpy และ Array เเบ่งตัว

ข้อดีของ numpy คือมีฟังก์ชันคำนวณทาง  
คณิตที่หลากหลาย  
 เช่น **sum** เพื่อหาผลรวม

```
1 import numpy as np
2
3 # ให้ v = ความเร็วของรถแต่ละคัน
4 v1 = 10
5 v2 = 20
6 v3 = 30
7 v4 = 40
```

```
1 # ต้องการหาผลรวมของสมาชิกภายใน
2 v = [v1,v2,v3,v4]
3 print((v1+v2+v3+v4))
4
5 # ถ้าเป็น numpy array
6 # ใช้ sum ได้เลย
7 va = np.array([v1,v2,v3,v4])
8 print(np.sum(va))
```

100  
100



# Numpy และ Array เบื้องต้น

```
1 a= [1,2,3]
2 b= [3,4,5]
3
4 a*b
```

ในลิสเราไม่สามารถเอาสมาชิกมา  
ดำเนินการได้โดยตรง

```
-----
TypeError                                 Traceback (most recent call
last)
<ipython-input-11-ae78e5e974fd> in <module>
      2 b= [3,4,5]
      3
----> 4 a*b
```

**TypeError:** can't multiply sequence by non-int of type 'list'



# Numpy และ Array เบื้องต้น

นำ array 2 array มา  
ดำเนินการโดยตรง

```
▶ 1 print(va)
  2 # ให้ t เป็นระยะเวลาที่รถแต่ละคันใช้
  3 t1 = 4.66
  4 t2 = 2.86
  5 t3 = 10.31
  6 t4 = 7.72
  7
  8 ta = np.array([t1,t2,t3,t4])
  9 print(ta)
 10
 11 # หาระยะทางที่รถแต่ละคันทำได้
 12
 13 s = va*ta
 14 print(s)
```

```
[10 20 30 40]
[ 4.66  2.86 10.31  7.72]
[ 46.6  57.2 309.3 308.8]
```

# Numpy และ Array เบื้องต้น

Mean = หาค่าเฉลี่ย

Std = หาค่าเบี่ยงเบนมาตรฐาน



1

```
print(np.mean(va))
```

25.0



1

```
print(np.std(va))
```

11.180339887498949



# Numpy array ถ้า List



# Numpy array และ List

```
► 1 # ตัวอย่างการใช้งาน numpy array เปรียบเทียบกับ List
  2 # โดยใช้แปลงหน่วยเซนติเมตรเป็นนิ้ว
  3 # 1 เซนติเมตร = 0.3937 นิ้ว
  4
  5 import numpy as np
  6
  7 cent = [175, 185, 168, 177]
```

```
► 1 # ใช้ List ธรรมดा
  2 inch = []
  3
  4 for i in cent:
  5     inch.append(i/2.54)
  6
  7 print(inch)
```

[68.89763779527559, 72.83464566929133, 66.14173228346456, 69.68503937007874]



# Numpy array และ List

```
▶ 1 # ใช้ List comprehension
 2 inch2 = [i/2.54 for i in cent]
 3 print(inch)
```

```
[68.89763779527559, 72.83464566929133, 66.14173228346456, 69.68503937007874]
```

```
▶ 1 # ใช้ map กับ Lambda
 2 # ใช้ map เพื่อจับสมาชิกแต่ละตัวในลิส
 3 inch3 = list(map(lambda x:x/2.54, cent))
 4 print(inch3)
```

```
[68.89763779527559, 72.83464566929133, 66.14173228346456, 69.68503937007874]
```



# Numpy array และ List



```
1 # ใช้ numpy array สะดวกกว่าเยอะ
2 inch4 = np.array(cent) / 2.54
3 print(inch4)
```

```
[68.8976378 72.83464567 66.14173228 69.68503937]
```

การเรียกแต่ละตัวใน **array** ไปหารหรือดำเนินการได้โดยตรงเรียกว่า  
**Vectorization** ซึ่งจะมีในภาษา R, Matlab

จะไม่มีใน Java, C ซึ่งต้องเขียนโค้ดที่ซับซ้อนกว่านี้ในการดำเนินการแบบ  
**vectorization**



# Numpy และ Array เบื้องต้น

Numpy array เเร็วกว่า List  
จริงไหม?

ทดสอบโดยใช้  
`Time.time()` ลองจับเวลา

# Numpy array และ List

ทำงานชิ้นเดียว กัน

Numpy array ใช้ 0.2 วินาที  
List comprehension ใช้ 37.67 วินาที

```
1 temp = np.arange(100000000)
```

```
1 # ใช้ numpy array
2 temp = np.arange(100000000)
3 import time
4 time1 = time.time()
5 x = temp**2
6 time2 = time.time()
7 print(time2-time1)
```

0.2034602165222168

```
1 # ใช้ List comprehension
2 import time
3 time1 = time.time()
4 [x ** 2 for x in range(100000000)]
5 time2 = time.time()
6 print(time2-time1)
```

37.67916536331177

# การใช้งาน numpy

# การใช้งาน numpy



Software Park Thailand  
/>Code Camp>

## การประกาศใช้ numpy array 1D (Vector)

```
▶ 1 import numpy as np  
2  
3 list1 = [1,3,5,7]  
4  
5 arr = np.array(list1)  
6 print(type(arr))  
7 print(arr)
```

```
<class 'numpy.ndarray'>  
[1 3 5 7]
```

# การใช้งาน numpy



Software Park Thailand  
/>Code Camp<

## การประกาศใช้ numpy array 2D (Matrix)



```
1 # numpy array 2D
2
3 list2 = [[1,3,10],[5,7,20],[9,11,30]]
4
5 arr2 = np.array(list2)
6 print(arr2)
```

```
[[ 1   3 10]
 [ 5   7 20]
 [ 9 11 30]]
```

# การใช้งาน numpy

## Vector และ Matrix คืออะไร?

Scalar

24

Vector

$$\begin{bmatrix} 2 & -8 & 7 \end{bmatrix}$$

row

or  
column

$$\begin{bmatrix} 2 \\ -8 \\ 7 \end{bmatrix}$$

Matrix

$$\begin{bmatrix} 6 & 4 & 24 \\ 1 & -9 & 8 \end{bmatrix}$$

row(s) × column(s)

อ่านเพิ่มเติม: <https://th.betweenmates.com/difference-between-vector-and-matrix-566600>



# การใช้งาน numpy

## np.arange()

สร้าง array ที่มีเลขมาให้

```
1 np.arange(10)
```

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
1 np.arange(0,10)
```

```
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
1 np.arange(0,20,2)
```

```
array([ 0,  2,  4,  6,  8, 10, 12, 14, 16, 18])
```

# การใช้งาน numpy



Software Park Thailand  
</Code Camp>

## np.zeros()

สร้าง array ที่มีแต่ 0

```
1 np.zeros(3)
```

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

```
1 np.zeros((3,3))
```

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

# การใช้งาน numpy



Software Park Thailand  
</Code Camp>

## np.ones()

สร้าง array ที่มีแต่ 1

```
1 np.ones(4)
```

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

```
1 np.ones((4,3))
```

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



# การใช้งาน numpy

```
1 np.linspace(0,1,5)
```

```
array([0. , 0.25, 0.5 , 0.75, 1. ])
```

```
1 np.linspace(0,10,8)
```

```
array([ 0.          ,  1.42857143,  2.85714286,  4.28571429,  5.71428571,
       7.14285714,  8.57142857, 10.        ])
```

## np.linspace()

สร้าง array ที่มีระยะห่างระหว่างกัน  
(interval) เท่ากัน

# การใช้งาน numpy



Software Park Thailand  
/>Code Camp>

`np.eye()`

สร้าง array ที่เป็น identity matrix

1 `np.eye(3)`

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

อ่านเพิ่มเติม: <https://www.mathpaper.net/index.php/en/4/779-identity-matrix>

# การใช้งาน numpy



Software Park Thailand  
</Code Camp>

## np.random.randint()

สร้าง array เพื่อ random เลข integer ขึ้นมา

```
1 x = np.random.randint(5)  
2 x
```

2

```
1 np.random.randint(1,20,5)
```

```
array([13, 17, 8, 15, 6])
```

```
1 np.random.randint(1,20,(2,2))
```

```
array([[ 3, 16],  
       [18, 19]])
```

# การใช้งาน numpy

## np.random.rand()

สร้าง array เพื่อ random float ช่วง [0,1] ขึ้นมา

```
1 x = np.random.rand(3)  
2 x
```

```
array([0.74595886, 0.51913262, 0.97924939])
```

```
1 type(x[1])
```

```
numpy.float64
```

```
1 np.random.rand(3,2)
```

```
array([[0.23890592, 0.92054281],  
       [0.17686122, 0.89782939],  
       [0.68649088, 0.45677965]])
```

# การใช้งาน numpy



Software Park Thailand  
/>Code Camp>

## arr.reshape()

เปลี่ยนโครงสร้างของ array

```
1 temp = np.random.randint(1,20,9)
2 temp
```

```
array([12,  7, 17, 16,  2,  2, 10, 17, 11])
```

```
1 temp.reshape(3,3)
```

```
array([[12,  7, 17],
       [16,  2,  2],
       [10, 17, 11]])
```

# การใช้งาน numpy



Software Park Thailand  
</Code Camp>

```
1 temp = np.random.randint(1,20,9)
2 temp
```

```
array([12,  7, 17, 16,  2,  2,  2, 10, 17, 11])
```

```
1 temp.reshape(3,2)
```

ระวังโครงสร้างของ array  
อาจจะไม่ match

```
ValueError
nt call last)
<ipython-input-109-08bacad722f2> in <module>
----> 1 temp.reshape(3,2)
```

Traceback (most rece

**ValueError:** cannot reshape array of size 9 into shape (3,2)



# การใช้งาน numpy

arr.T

คือการ **Transpose** เปลี่ยนແຕວ

กับคอลัมพ์ของทุกສมาชิกใน

**matrix**

```
1 temp2 = temp.reshape(3,3)  
2 temp2
```

```
array([[12, 7, 17],  
       [16, 2, 2],  
       [10, 17, 11]])
```

```
1 temp2.T
```

```
array([[12, 16, 10],  
       [ 7,  2, 17],  
       [17,  2, 11]])
```

อ่านเพิ่มเติม: <https://www.mathpaper.net/index.php/en/4/774-transpose-of-matrix>

# การใช้งาน numpy

Transposing a 2x3 matrix to create a 3x2 matrix

$$\begin{bmatrix} 6 & 4 & 24 \\ 1 & -9 & 8 \end{bmatrix}^T = \begin{bmatrix} 6 & 1 \\ 4 & -9 \\ 24 & 8 \end{bmatrix}$$



# Operations and Indexing

# การดำเนินการและใช้ index

# Operations and Indexing

```
1 x = np.random.randint(1,20,8)  
2 x
```

```
array([10, 14, 10, 18, 19, 6, 3, 2])
```

## arr.max()

```
1 x.max()
```

```
19
```

การหาค่าสูงสุด

```
1 np.max(x)
```

```
19
```



# Operations and Indexing

## arr.min()

การหาค่าต่ำสุด

```
1 | x = np.random.randint(1,20,8)
2 | x
array([19,  7,  1,  4, 15,  4,  1, 10])
```

```
1 | x.min()
```

1



# Operations and Indexing

## arr.argmax()

การหา index ของค่าสูงสุด

```
1 | x = np.random.randint(1,20,8)
2 | x
```

```
array([ 4, 16, 14, 19,  6, 17,  8, 16])
```

```
1 | x.argmax()
```

3



# Operations and Indexing

## arr.argmin()

การหา index ของต่ำสุด

```
1 x = np.random.randint(1,20,8)
2 x
```

```
array([ 4, 16, 14, 19, 6, 17, 8, 16])
```

```
1 x.argmin()
```

```
0
```

# Operations and Indexing

## arr.sort()

การเรียงค่าน้อยไปมาก

```
1 x = np.random.randint(1,20,8)  
2 x
```

```
array([ 4, 16, 14, 19, 6, 17, 8, 16])
```

```
1 x.sort()  
2 x
```

```
array([ 4, 6, 8, 14, 16, 16, 17, 19])
```



# Operations and Indexing

`arr.sort() + [::-1]`

การเรียงค่ามากไปน้อย

```
1 x = np.random.randint(1,20,8)  
2 x
```

```
array([ 4, 16, 14, 19,  6, 17,  8, 16])
```

```
1 x.sort()
```

```
array([ 4,  6,  8, 14, 16, 16, 17, 19])
```

```
1 x[::-1]
```

```
array([19, 17, 16, 16, 14,  8,  6,  4])
```

# Operations and Indexing

## arr.shape()

การเช็ค shape ของ array

```
1 arr1 = np.random.randint(1,20,5)  
2 arr1
```

```
array([12, 17, 12, 15, 7])
```

```
1 arr1.shape
```

```
(5,)
```

```
1 arr2 = np.random.randint(1,20,(3,3))  
2 arr2
```

```
array([[13, 15, 15],  
       [ 2,  9, 16],  
       [ 9, 14,  7]])
```

```
1 arr2.shape
```

```
(3, 3)
```

# Operations and Indexing

ใน array 1D

การเข้าถึงสมาชิกใช้วิธีเดียวกันกับ

List

```
1 arr1 = np.random.randint(1,99,10)  
2 arr1
```

```
array([ 3, 79, 19, 47, 96, 21, 6, 79, 28, 89])
```

```
1 arr1[1]
```

```
79
```

```
1 arr1[0:3]
```

```
array([ 3, 79, 19])
```

```
1 arr1[:5]
```

```
array([ 3, 79, 19, 47, 96])
```

```
1 arr1[7:]
```

```
array([79, 28, 89])
```

# Operations and Indexing

ใน array 1D

สามารถแก้ไขสมาชิก โดยใช้วิธี  
เดียวกันกับ List

```
1 arr1 = np.random.randint(1,99,10)  
2 arr1
```

```
array([31, 14, 51, 89, 18, 71, 60, 29, 23, 47])
```

```
1 arr1[0] = 111  
2 arr1
```

```
array([111, 14, 51, 89, 18, 71, 60, 29, 23, 47])
```

```
1 arr1[8:] = 99  
2 arr1
```

```
array([111, 14, 51, 89, 18, 71, 60, 29, 99, 99])
```



# Operations and Indexing

ใน array 2D

การเข้าถึงสมาชิกใช้วิธีคล้ายกับ

Nested List

```
1 arr2d = np.random.randint(1,99,(3,3))  
2 arr2d
```

```
array([[ 4, 52, 66],  
       [97, 73, 98],  
       [45, 36, 79]])
```

```
1 arr2d[1]
```

```
array([97, 73, 98])
```

```
1 arr2d[2][2]
```

79

```
1 arr2d[2,2]
```

79



# Operations and Indexing

ใน array 2D  
สามารถใช้ slicing ได้เช่นกัน

```
1 arr2d = np.random.randint(1,99,(3,3))  
2 arr2d
```

```
array([[ 4, 52, 66],  
       [97, 73, 98],  
       [45, 36, 79]])
```

```
1 arr2d[:2,2]
```

```
array([66, 98])
```

```
1 arr2d[1,:2]
```

```
array([97, 73])
```

```
1 arr2d[:3,:2]
```

```
array([[ 4, 52],  
       [97, 73],  
       [45, 36]])
```



# Operations and Indexing

## Conditional Selection

การเลือกแบบมีเงื่อนไข

ประยุกต์ใช้ร่วมกับ Boolean  
True, False



# Operations and Indexing

```
1 arr1 = np.random.randint(1,99,10)
2 arr1
```

```
array([70, 96, 45, 52, 72, 52, 96, 17, 13, 9])
```

```
1 arr1 > 50
```

```
array([ True,  True, False,  True,  True,  True,  True, False, False,
       False])
```

```
1 arr1[arr1>50]
```

```
array([70, 96, 52, 72, 52, 96])
```

## Conditional Selection

การเลือกแบบมีเงื่อนไข

ใน array 1D



# Operations and Indexing

## Conditional Selection

การเลือกแบบมีเงื่อนไข

ใน array 2D

```
1 arr2d = np.random.randint(1,99,(3,3))  
2 arr2d
```

```
array([[81, 7, 2],  
       [72, 5, 42],  
       [71, 57, 64]])
```

```
1 boolArr = arr2d > 65  
2 boolArr
```

```
array([[ True, False, False],  
       [ True, False, False],  
       [ True, False, False]])
```

```
1 arr2d[boolArr]
```

```
array([81, 72, 71])
```

# Quizzz

# Quizzz

1. จงเขียน array ที่มีแต่เลข 0 แบบ vector 10 ตัว และ แบบ Matrix 4x4
2. จงเขียน array ที่มีแต่เลข 1 แบบ vector 5 ตัว และ แบบ Matrix 3x3
3. จงเขียน array ที่มีเป็นการ random ตัวเลข 20 ตัว แบบ float ค่าอยู่ในช่วง 10-99 และวน้ำแต่ละเลขยกกำลังสอง
4. ทำเหมือนข้อ 3 แต่เขียนแบบ List comprehension
5. นำข้อ 3 และ 4 มาเปรียบเวลา กัน โดยใช้ฟังก์ชัน `time.time()` เพื่อจับเวลา แสดงผลลัพธ์ในรูปแบบมิลลิวินาที
6. จงเขียน array ที่ใช้ฟังก์ชันใน numpy ได้เลข 10 ถึง 1000

7. จงเขียน array ที่ใช้พังก์ชันใน numpy ได้เลขคู่ 10 ถึง 2000
8. จงเขียน array ที่ใช้พังก์ชันใน numpy ได้เลขคี่ 12 ถึง 900
9. จงสร้าง identity matrix แบบ 4x4
10. จงหาว่า array ที่มีสมาชิก 100 ตัว เริ่มจาก 5 ถึง 25 มีผลต่างแต่ละตัวเท่ากับเท่าใด
11. จงเขียนพังก์ชัน random ตัวเลขแบบ float ในช่วง 0-1 ขึ้นมา 1 ตัว
12. จงเขียนพังก์ชันที่ random ตัวเลขแบบ int ในช่วง 50-100 โดยให้อยู่ในรูป Matrix 4x4 และ Transpose



# Quizzz

13. ให้ array ที่กำหนด จงเขียนโค้ดที่เลือกเฉพาะ 8,9,13,14
14. ให้ array ที่กำหนด จงเขียนโค้ดที่เลือกเฉพาะ 18,19,20
15. ให้ array ที่กำหนด จงเขียนโค้ดที่เลือกเฉพาะ 16,17,21,22
16. ให้ array ที่กำหนด จงเขียนโค้ดที่เลือกทั้งหมด 3
17. จงเขียนโค้ดที่ทำให้ได้ array นี้  
[[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]]

```
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]])
```

18. จงเขียนโปรแกรมสุ่มเลขจำนวนเต็มในช่วง 10-100 ออกมาอยู่ใน matrix 5x5 และใช้ **If-else** เช็คทีละตัวว่าตัวใดบ้างที่มีค่ามากกว่า 50 และปรินต์ออกมา
19. จงเขียนโปรแกรมสุ่มเลขจำนวนเต็มในช่วง 10-100 ออกมาอยู่ใน matrix 5x5 และใช้ **conditional selection** เช็คว่าตัวใดบ้างมีค่ามากกว่า 40 นำมาใส่ **List** ที่สร้างใหม่ และปรินต์ออกมา
20. เขียนโปรแกรมรับค่าจากผู้ใช้ให้ป้อนเลข 9 ตัว แต่ละตัวต้องไม่ซ้ำกันและมีค่าอยู่ในช่วง 10-100 นำเลข 9 ตัวนั้นมาทำเป็น matrix 3x3 และ **transport** และหาเลขอยู่ตำแหน่งที่ (2,2)

# lesson.end()