



<http://vku.udn.vn>

ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN
Vietnam - Korea University of Information and Communication Technology

Numpy

Python cho AI



VUI MỪNG

- Giới thiệu về Numpy
- Lập chỉ mục mảng Numpy
- Hoạt động mảng Numpy
- Phát sóng
- Xử lý dữ liệu



VUI MỪNG

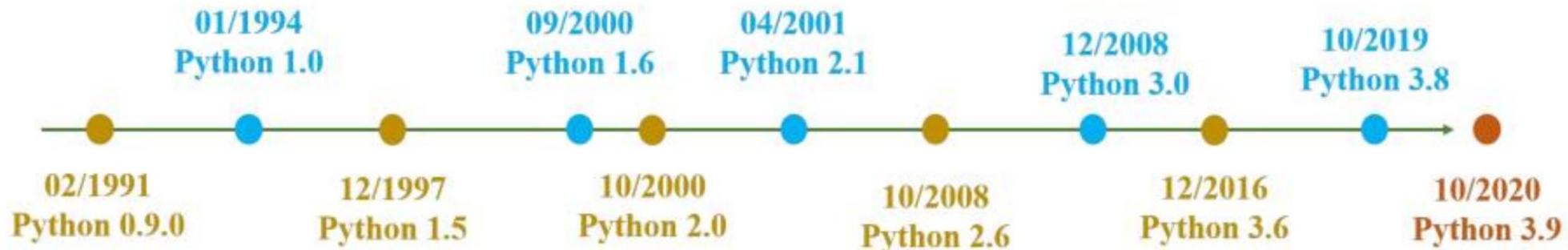
- Giới thiệu về Numpy

- Lập chỉ mục mảng Numpy
- Hoạt động mảng Numpy
- Phát sóng
- Xử lý dữ liệu



Giới thiệu

❖ Numpy is a Python library



Ý tưởng từ 1980s



Được đặt tên theo
nhóm hài Monty Python



Bắt đầu cài đặt
từ 12/1989



Guido van Rossum



228,855 python packages
(PyPI)



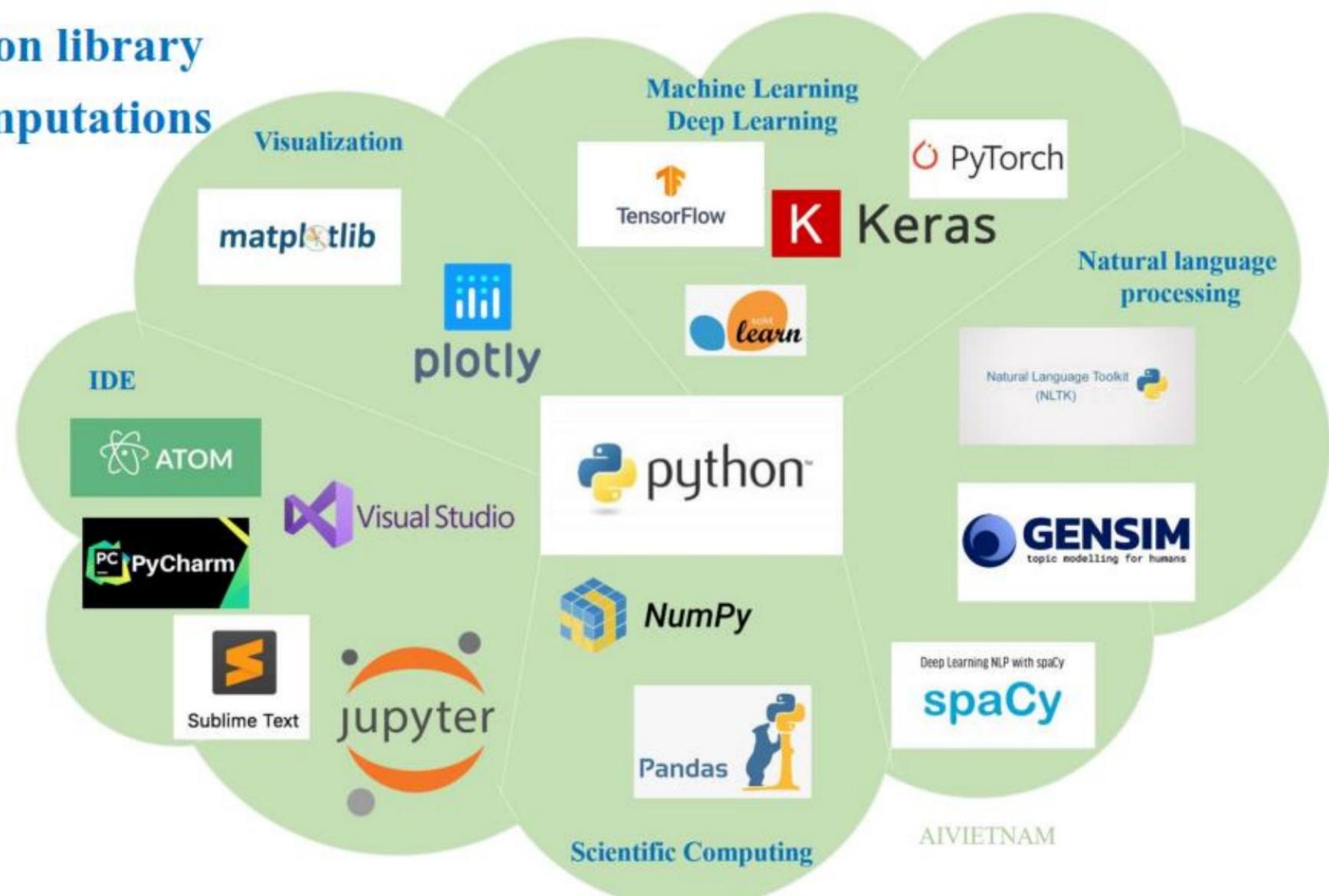
Hỗ trợ rất mạnh cho
Data Science và Machine Learning

AIVIETNAM



Giới thiệu

- ❖ Numpy is a Python library
- ❖ For scientific computations





Giới thiệu

❖ Numpy array ↔ Tensor in Tensorflow and Pytorch



NumPy



PyTorch



TensorFlow

```

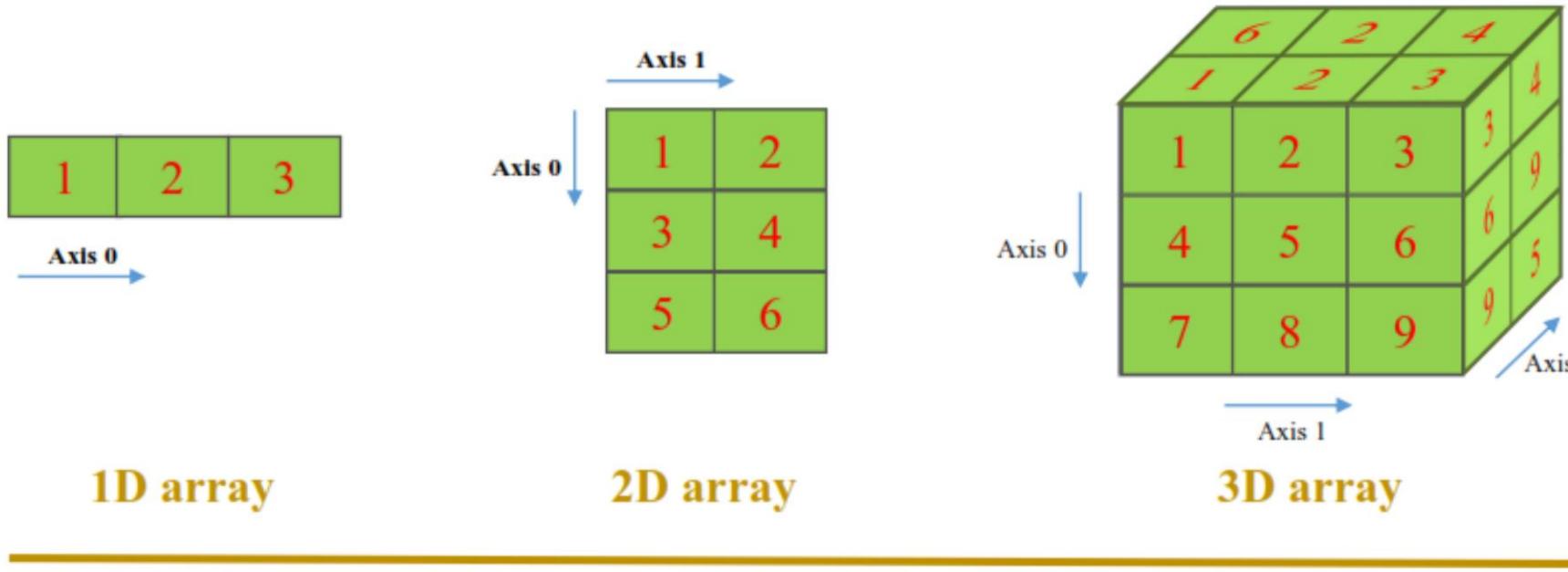
1 import tensorflow as tf
2 import numpy as np
3
4 # create a numpy array
5 arr_np = np.arange(10)
6 print(arr_np)
7 print(arr_np.shape)
8 print('-----')
9
10 # convert from numpy array to tensor
11 arr_tf = tf.convert_to_tensor(arr_np)
12 print(arr_tf)
13 print(arr_tf.shape)
14 print('-----')
15
16 # convert from tensor to numpy array
17 arr_np_back = arr_tf.numpy()
18 print(arr_np_back)
19 print(arr_np_back.shape)

[0 1 2 3 4 5 6 7 8 9]
(10,)

-----
tf.Tensor([0 1 2 3 4 5 6 7 8 9], shape=(10,), dtype=int64)
(10,)

-----
[0 1 2 3 4 5 6 7 8 9]
(10,)
```

❖ Numpy arrays are multi-dimensional arrays

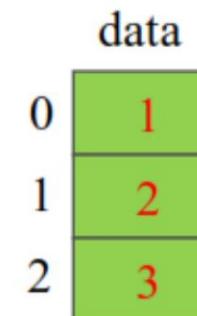




Giới thiệu

❖ Create Numpy array ❖ From List

```
arr_np = np.array(python_list)
```



data[0]

1

data[1]

2

```

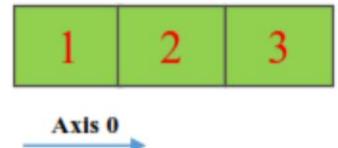
1 # aivietnam.ai
2 # tạo ndarray từ list
3
4 import numpy as np
5
6 # tạo list
7 l = list(range(1, 4))
8
9 # tạo ndarray
10 data = np.array(l)
11
12 print(data)
13 print(data[0])
14 print(data[1])

```

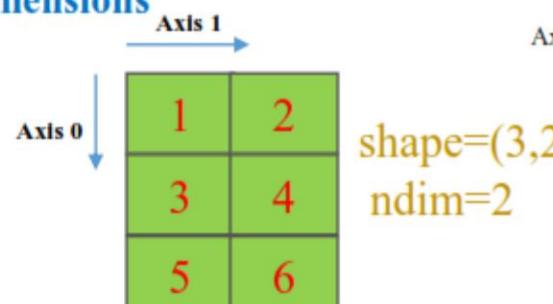
```
[1 2 3]
1
2
```

❖ Common attributes

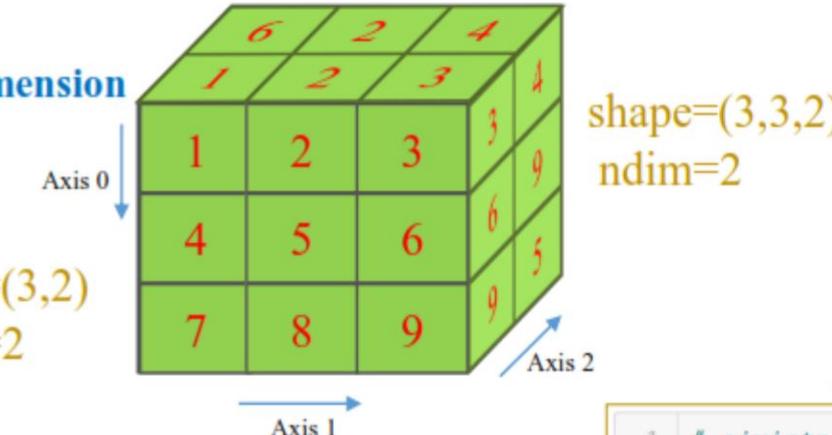
- ❖ **dtype:** data type
- ❖ **shape:** return a tuple of #elements in each dimension
- ❖ **ndim:** return #dimensions



shape=(3,)
ndim=1



shape=(3,2)
ndim=2



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

```

1 # aivietnam.ai
2 # tạo ndarray từ list
3
4 import numpy as np
5
6 # tạo list
7 list1D = [1,2,3]
8
9 # tạo ndarray
10 data = np.array(list1D)
11
12 print(data)
13 print(data.shape)

```

[1 2 3]
(3,)

...

```

1 # aivietnam.ai
2 # tạo ndarray từ list
3
4 import numpy as np
5
6 # tạo list
7 list2D = [[1,2],[3,4],[5,6]]
8
9 # tạo ndarray
10 data = np.array(list2D)
11
12 print(data)
13 print(data.shape)

```

[[1 2]
[3 4]
[5 6]]
(3, 2)

```

1 # aivietnam.ai
2 # tạo ndarray từ list
3
4 import numpy as np
5
6 # tạo list
7 list3D = [[[1,6], [2,2], [3,4], [4,7], [5,2], [6,9], [7,7], [8,2], [9,5]]]
8
9 # tạo ndarray
10 data = np.array(list3D)
11
12 print(data)
13 print(data.shape)
14
15 print(data.dtype)

```

(3, 3, 2)

dtype example

```

1 # aivietnam.ai
2 # tạo ndarray từ list
3
4 import numpy as np
5
6 # tạo ndarray
7 data1 = np.array([1,2,3])
8 print(data1.dtype)
9
10 data2 = np.array([1.,2.,3.])
11 print(data2.dtype)
12
13 data3 = np.array([1,2,3], dtype=np.int64)
14 print(data3.dtype)

```

int32
float64
int64



.Giới thiệu

❖ Update an element

index	0	1	2
data =	1	2	3

data[0] = 8

data =	8	2	3
--------	---	---	---

```

1 # aivietnam.ai
2 # thay đổi giá trị phần tử
3
4 import numpy as np
5
6 # tạo list
7 l = list(range(1, 4))
8
9 # tạo ndarray
10 data = np.array(l)
11 print(data)
12
13 data[0] = 8
14 print(data)

```

[1 2 3]
[8 2 3]



Giới thiệu

❖ Create Numpy array

zeros() function

0	1	2
0	0	0
1	0	0

```

1 # aivietnam.ai
2 # Tạo một numpy array
3 # với tất cả phần tử là 0
4
5 import numpy as np
6
7 # shape: 2 dòng, 3 cột
8 arr = np.zeros((2,3))
9 print(arr)

[[0. 0. 0.]
 [0. 0. 0.]]

```

ones() function

0	1	2
0	1	1
1	1	1

```

1 # aivietnam.ai
2 # Tạo một numpy array với
3 # tất cả phần tử là 1
4
5 import numpy as np
6
7 # numpy.ones(shape)
8 # shape: 2 dòng, 3 cột
9 arr = np.ones((2,3))
10 print(arr)

[[1. 1. 1.]
 [1. 1. 1.]]

```

full() function

0	1	2
0	9	9
1	9	9

```

1 # aivietnam.ai
2 # Tạo một numpy array với tất
3 # cả phần tử là hằng số fill_value
4
5 import numpy as np
6
7 # numpy.full(shape, fill_value)
8 # shape: 2 dòng, 3 cột
9 arr = np.full((2,3), 9)
10 print(arr)

[[9 9 9]
 [9 9 9]]

```



Giới thiệu

❖ Create Numpy array

arange() function

```
0   1   2   3   4
arr1 = [ 0 | 1 | 2 | 3 | 4 ]
          0   1   2
arr2 = [ 0 | 2 | 4 ]
```

```
1 # aivietnam.ai
2 import numpy as np
3
4 # np.arange(start=0, stop, step=1)
5 arr1 = np.arange(5)
6 print(arr1)
7
8 arr2 = np.arange(0, 5, 2)
9 print(arr2)
```

[0 1 2 3 4]
[0 2 4]

eye() function

	0	1	2
0	1	0	0
1	0	1	0
2	0	0	1

```
1 # aivietnam.ai
2 # Tạo một numpy array với đường chéo là số 1
3 # số 0 được điền vào những ô phần tử còn lại
4
5 import numpy as np
6
7 # numpy.eye(N)
8 # shape: 3 dòng, 3 cột
9 arr = np.eye(3)
10 print(arr)
```

[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]

random() function

	0	1	2
0	0.574	0.682	0.704
1	0.806	0.844	0.799

```
1 # aivietnam.ai
2 # Tạo một numpy array với
3 # giá trị ngẫu nhiên
4
5 import numpy as np
6
7 # np.random.random(size)
8 # shape: 2 dòng, 3 cột; với
9 # phần tử có giá trị ngẫu nhiên
10 arr = np.random.random((2,3))
11 print(arr)
```

[[0.57488062 0.68266312 0.70438569]
 [0.80661973 0.84413356 0.79905247]]



Giới thiệu

Some important functions

where() function

arr =	0	1	2	3	4
arr<3 =	T	T	T	F	F
out =	0	1	2	6	8

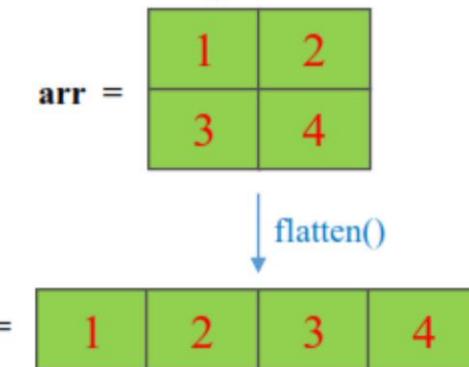
```

1 # aivietnam.ai
2 import numpy as np
3
4 # create an array
5 arr = np.arange(5)
6 print(arr)
7
8 # condition
9 condition = arr < 3
10 out = np.where(condition, arr, arr**2)
11
12 print(condition)
13 print(out)

```

```
[0 1 2 3 4]
[ True  True  True False False]
[0 1 2 6 8]
```

flatten() function



```

1 # aivietnam.ai
2 import numpy as np
3
4 arr = np.array([[1,2], [3,4]])
5 out = arr.flatten()
6
7 print(arr)
8 print(out)

```

```
[[1 2]
 [3 4]]
[1 2 3 4]
```



.Giới thiệu

Some important functions

reshape() function

data

1	2	3
4	5	6

data_rs

1	2
3	4
5	6

```

1 # aivietnam.ai
2 import numpy as np
3
4 # tạo list
5 l = [[1,2,3],
6     [4,5,6]]
7
8 # tạo ndarray
9 data = np.array(l)
10 print('data\n', data)
11 print('data shape\n', data.shape)
12
13 # reshape
14 data_rs = np.reshape(data, (3,2))
15 print('data_rs\n', data_rs)
16 print('data_rs shape\n', data_rs.shape)

```

```

data
[[1 2 3]
 [4 5 6]]
data shape
(2, 3)
data_rs
[[1 2]
 [3 4]
 [5 6]]
data_rs shape
(3, 2)

```



VUI MỪNG

- Giới thiệu về Numpy
- Lập chỉ mục mảng Numpy
- Hoạt động mảng Numpy
- Phát sóng
- Xử lý dữ liệu

❖ Slicing

`arr[for_axis_0, for_axis_1, ...]`

‘:’: get all the elements

‘**a:b**’: get the elements from ath to bth

		0	1
0	1	2	
1	3	4	
2	5	6	

		0	1
0	1	2	
1	3	4	
2	5	6	

		0	1
0	1	2	
1	3	4	
2	5	6	

		0	1
0	1	2	
1	3	4	
2	5	6	

		0	1
0	1	2	
1	3	4	
2	5	6	

❖ Slicing

	0	1	2
0	1	2	3
1	5	6	7

b_{arr} = a_{arr}[:, 1:3]

	0	1
0	2	3
1	6	7

```

1 # aivietnam.ai
2 import numpy as np
3
4 # Khởi tạo numpy array a_arr
5 a_arr = np.array([[1,2,3],
6                   [5,6,7]])
7
8 # Sử dụng slicing để tạo mảng b_arr
9 # bằng cách lấy tất cả các dòng và cột 1,2
10 b_arr = a_arr[:, 1:3]
11
12 print(a_arr)
13 print(b_arr)

```

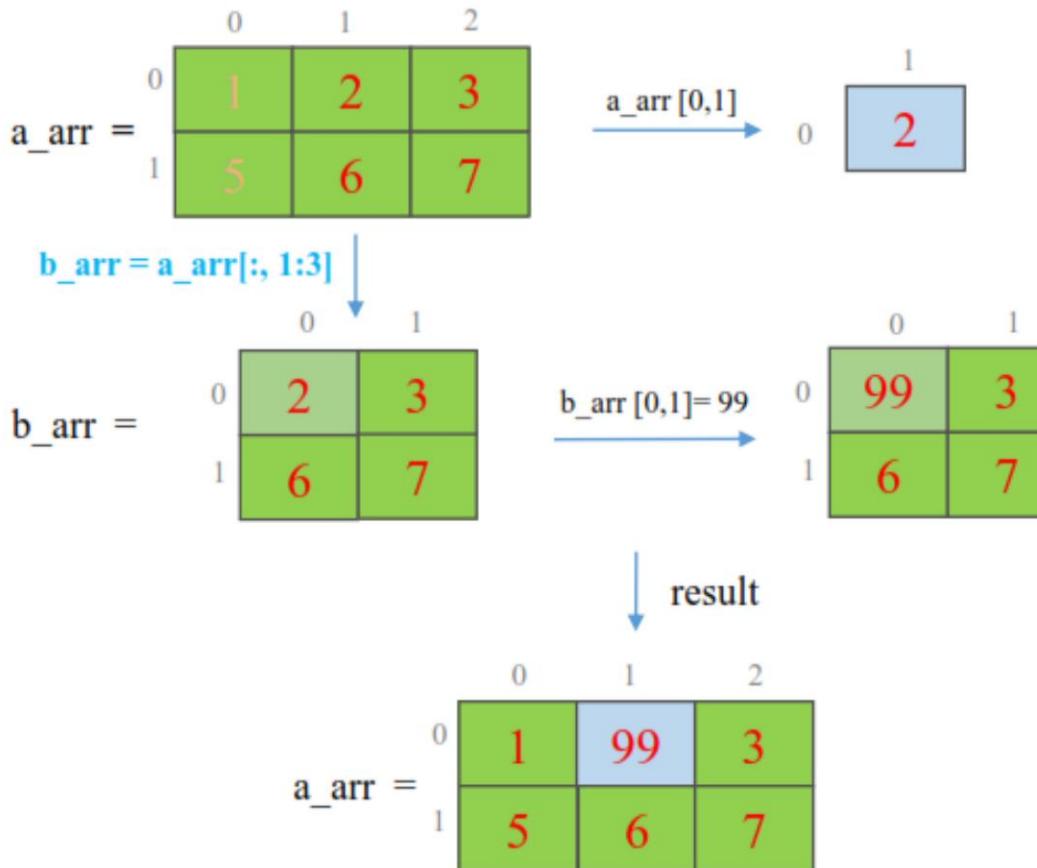
```

[[1 2 3]
 [5 6 7]]
[[2 3]
 [6 7]]

```

❖ Slicing

❖ Mutable



```

1 # aivietnam.ai
2 import numpy as np
3
4 # Khởi tạo numpy array a_arr
5 a_arr = np.array([[1,2,3],
6                   [5,6,7]])
7 print('a_arr \n', a_arr)
8
9 # Sử dụng slicing để tạo mảng b_arr
10 b_arr = a_arr[:, 1:3]
11 print('b_arr \n', b_arr)
12
13 print('before changing \n', a_arr[0, 1])
14 b_arr[0, 0] = 99
15 print('after changing \n', a_arr[0, 1])

```

```

a_arr
[[1 2 3]
 [5 6 7]]
b_arr
[[2 3]
 [6 7]]
before changing
2
after changing
99

```



❖ Get a row

```
arr = [[1, 2, 3],
       [5, 6, 7],
       [9, 10, 11]]
```

```
row_m1 = arr[1].shape(3, )
```

```
row_m2 = arr[1:2, :].shape(1, 3)
```

.Lập chỉ mục mảng

```

1 # aivietnam.ai
2 import numpy as np
3
4 # Tạo một numpy array có shape (3, 3) với giá trị
5 # [[ 1  2  3]
6 # [ 5  6  7]
7 # [ 9 10 11]]
8 arr = np.array([[1, 2, 3],
9                 [5, 6, 7],
10                [9,10,11]])
11
12 # Hai cách truy cập dữ liệu ở dòng index=1
13 # cách 1: số chiều giảm
14 row_m1 = arr[1, :]
15
16 # cách 2: số chiều được giữ nguyên
17 row_m2 = arr[1:2, :]
18
19 print(row_m1, row_m1.shape)
20 print(row_m1, row_m2.shape)

```

```
[5 6 7] (3, )
[5 6 7] (1, 3)
```



❖ Get a column

```

arr = [[1, 2, 3],
       [5, 6, 7],
       [9, 10, 11]]
```

	0	1	2
0	1	2	3
1	5	6	7
2	9	10	11


```

col_m1 = [2, 6, 10] (3,)
```

	0	1	2
0	2	6	10
1			
2			


```

col_m2 = [2, 6, 10] (3, 1)
```

	0
0	2
1	6
2	10

Lập chỉ mục mảng

```

1 # aivietnam.ai
2 import numpy as np
3
4 # Tạo một numpy array có shape (3, 3) với giá trị
5 arr = np.array([[1,2,3],
6                 [5,6,7],
7                 [9,10,11]])
8
9 # Hai cách truy cập dữ liệu ở cột index=1 của mảng
10 # cách 1: số chiều giảm
11 col_m1 = arr[:, 1]
12
13 # cách 2: số chiều được giữ nguyên
14 col_m2 = arr[:, 1:2]
15
16 print(col_m1, col_m1.shape)
17 print(col_m2, col_m2.shape)

[ 2  6 10] (3, )
[[ 2]
 [ 6]
[10]] (3, 1)
```

❖ Using Lists as indices

```
arr = [[0, 1, 2], [3, 4], [5, 6]]  
  
arr[[0, 1, 2], [0, 1, 0]] = [1, 4, 5]  
  
arr[[0, 0], [1, 1]] = [2, 2]
```

```
1 # aivietnam.ai  
2 import numpy as np  
3  
4 # tạo arr  
5 arr = np.array([[1, 2],  
6                 [3, 4],  
7                 [5, 6]])  
8  
9 # lấy giá trị vị trí (0,0), (1,1) và (2,0)  
10 out1 = arr[[0, 1, 2], [0, 1, 0]]  
11 print('out1:\n', out1)  
12  
13 # Có thể truy xuất tới 1 phần tử nhiều hơn 1 lần  
14 out2 = arr[[0, 0], [1, 1]]  
15 print('out2:\n', out2)  
  
out1:  
[1 4 5]  
out2:  
[2 2]
```

❖ Boolean indices

arr =

	0	1
0	1	2
1	3	4
2	5	6

arr > 2 =

	0	1
0	F	F
1	T	T
2	T	T

arr[arr>2] =

0	1	2	3
3	4	5	6

```

1 # aivietnam.ai
2 import numpy as np
3
4 arr = np.array([[1,2],
5                 [3, 4],
6                 [5, 6]])
7 print(arr)
8
9 # Tim các phần tử lớn hơn 2
10 bool_idx = (arr > 2)
11 print(bool_idx)

```

```

[[1 2]
 [3 4]
 [5 6]]
[[False False]
 [ True  True]
 [ True  True]]

```

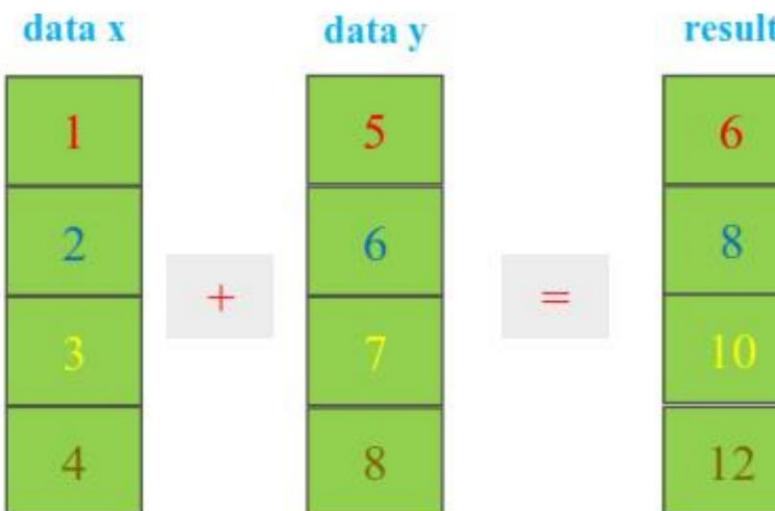


VUI MỪNG

- Giới thiệu về Numpy
- Lập chỉ mục mảng Numpy
- Các thao tác mảng Numpy
- Phát sóng
- Xử lý dữ liệu



❖ Addition



Hoạt động mảng

```

1 # aivietnam.ai
2 import numpy as np
3
4 x = np.array([1,2,3,4])
5 y = np.array([5,6,7,8])
6
7 print('data x \n', x)
8 print('data y \n', y)
9
10 # Tổng của 2 mảng
11 print('method 1 \n', x + y)
12 print('method 2 \n', np.add(x, y))

```

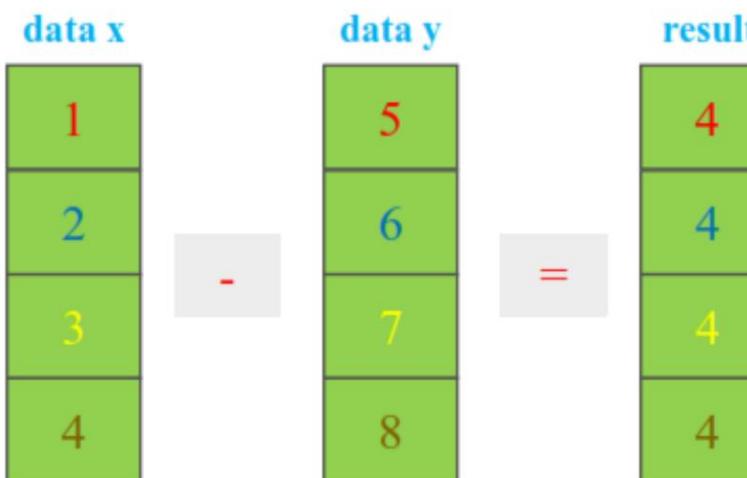
```

data x
[1 2 3 4]
data y
[5 6 7 8]
method 1
[ 6  8 10 12]
method 2
[ 6  8 10 12]

```



❖ Subtraction



.Hoạt động mảng

```

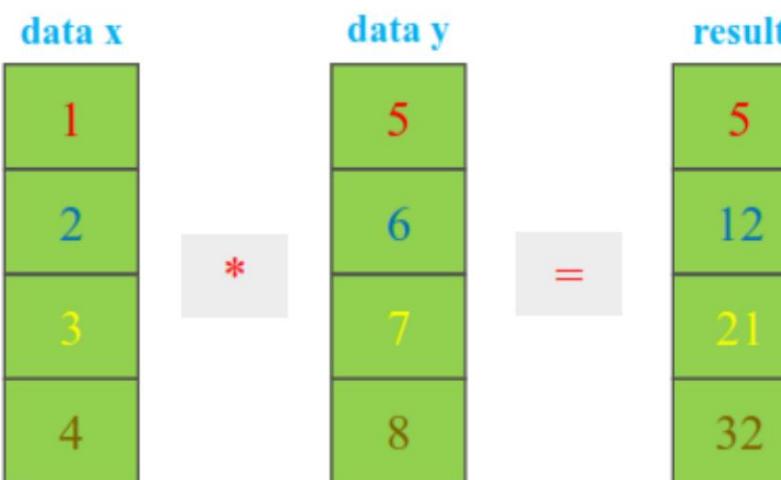
1 # aivietnam.ai
2 import numpy as np
3
4 x = np.array([5,6,7,8])
5 y = np.array([1,2,3,4])
6
7 print('data x \n', x)
8 print('data y \n', y)
9
10 # Hiệu 2 mảng
11 print('method 1 \n', x - y)
12 print('method 2 \n', np.subtract(x, y))
  
```

```

data x
[5 6 7 8]
data y
[1 2 3 4]
method 1
[4 4 4 4]
method 2
[4 4 4 4]
  
```

.Hoạt động mảng

❖ Multiplication



```

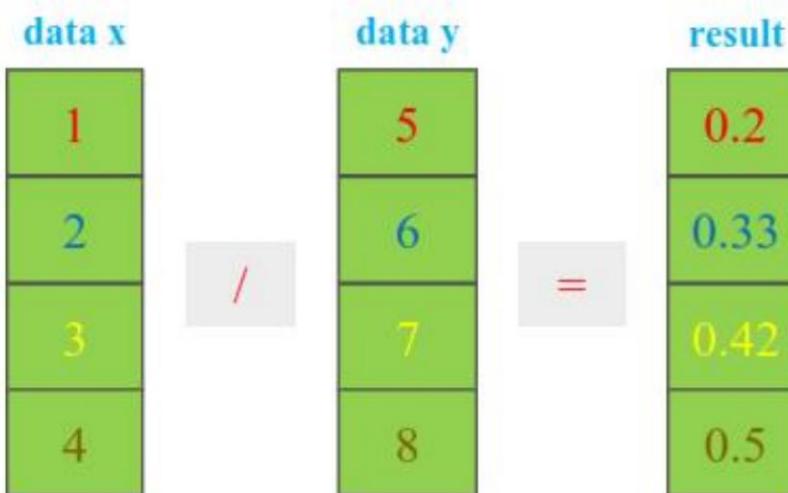
1 # aivietnam.ai
2 import numpy as np
3
4 x = np.array([1,2,3,4])
5 y = np.array([5,6,7,8])
6
7 print('data x \n', x)
8 print('data y \n', y)
9
10 # Tích các phần tử tương ứng giữa x và y
11 print('method 1 \n', x*y)
12 print('method 2 \n', np.multiply(x, y))
  
```

```

data x
[1 2 3 4]
data y
[5 6 7 8]
method 1
[ 5 12 21 32]
method 2
[ 5 12 21 32]
  
```



❖ Division



.Hoạt động mảng

```

1 # aivietnam.ai
2 import numpy as np
3
4 x = np.array([5, 6, 7, 8])
5 y = np.array([1, 2, 3, 4])
6
7 print('data x \n', x)
8 print('data y \n', y)
9
10 # Phép chia các từng phần tương ứng x cho y
11 print('method 1 \n', x / y)
12 print('method 2 \n', x // y)
13 print('method 3 \n', np.divide(x, y))

```

```

data x
[5 6 7 8]
data y
[1 2 3 4]
method 1
[5.          3.          2.33333333 2.          ]
method 1
[5 3 2 2]
method 2
[5.          3.          2.33333333 2.          ]

```



.Hoạt động mảng

❖ Square root

data
1
2
3
4

result
1.0
1.4
1.7
2.0

sqrt(data) =

```

1 # aivietnam.ai
2 import numpy as np
3
4 data = np.array([1,2,3,4])
5
6 print('data \n', data)
7
8 # Căn bậc 2 từng phần tử trong data
9 print('sqrt \n', np.sqrt(data))

```

```

data
[1 2 3 4]
sqrt
[1.          1.41421356 1.73205081 2.        ]

```



.Hoạt động mảng

❖ Inner product

$$\begin{matrix} v \\ \begin{matrix} 1 \\ 2 \end{matrix} \end{matrix} \cdot \begin{matrix} w \\ \begin{matrix} 2 \\ 3 \end{matrix} \end{matrix} = \begin{matrix} \text{result} \\ 8 \end{matrix}$$

```

1 # aivietnam.ai
2 import numpy as np
3
4 v = np.array([1, 2])
5 w = np.array([2, 3])
6
7 # Tính inner product giữa v và w
8 print('method 1 \n', v.dot(w))
9 print('method 2 \n', np.dot(v, w))

```

method 1
8

method 2
8

.Hoạt động mảng

❖ Vector-matrix multiplication

$$\begin{matrix} & \text{x} \\ \begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} & \bullet & \begin{matrix} \text{v} \\ \begin{matrix} 1 \\ 2 \end{matrix} \end{matrix} & = & \begin{matrix} \text{result} \\ \begin{matrix} 5 \\ 11 \end{matrix} \end{matrix} \end{matrix}$$

$$\begin{matrix} & \text{v} \\ \begin{matrix} 1 & 2 \end{matrix} & \bullet & \begin{matrix} & \text{x} \\ \begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix} \end{matrix} & = & \begin{matrix} \text{result} \\ \begin{matrix} 7 \\ 10 \end{matrix} \end{matrix} \end{matrix}$$

```

1 # aivietnam.ai
2 import numpy as np
3
4 X = np.array([[1,2],
5               [3,4]])
6 v = np.array([1,2])
7
8 print('matrix X \n', X)
9 print('vector v \n', v)
10
11 # phép nhân giữa ma trận và vector
12 print('method 1: X.dot(v) \n', X.dot(v))
13 print('method 1: v.dot(X) \n', v.dot(X))
14 #print('\n method 2: X.dot(v) \n', np.dot(X, v))
15 #print('\n method 2: v.dot(X) \n', np.dot(v, X))

matrix X
[[1 2]
 [3 4]]
vector v
[1 2]
method 1: X.dot(v)
[ 5 11]
method 1: v.dot(X)
[ 7 10]

```



.Hoạt động mảng

❖ Matrix-matrix multiplication

$$\begin{array}{c}
 \text{X} \\
 \begin{array}{|c|c|} \hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array}
 \end{array}
 \cdot
 \begin{array}{c}
 \text{Y} \\
 \begin{array}{|c|c|} \hline 2 & 3 \\ \hline 2 & 1 \\ \hline \end{array}
 \end{array}
 =
 \begin{array}{c}
 \text{result} \\
 \begin{array}{|c|c|} \hline 6 & 5 \\ \hline 14 & 13 \\ \hline \end{array}
 \end{array}$$

$$\begin{array}{c}
 \text{Y} \\
 \begin{array}{|c|c|} \hline 2 & 3 \\ \hline 2 & 1 \\ \hline \end{array}
 \end{array}
 \cdot
 \begin{array}{c}
 \text{X} \\
 \begin{array}{|c|c|} \hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array}
 \end{array}
 =
 \begin{array}{c}
 \text{result} \\
 \begin{array}{|c|c|} \hline 11 & 16 \\ \hline 5 & 8 \\ \hline \end{array}
 \end{array}$$

```

1 # aivietnam.ai
2 import numpy as np
3
4 X = np.array([[1,2],
5 [3,4]])
6 Y = np.array([[2,3],
7 [2,1]])
8
9 # Phép nhân giữa hai ma trận
10 print('method 1 \n', X.dot(Y))
11 print('method 1 \n', Y.dot(X))
12 #print('method 2 \n', np.dot(X, Y))
13 #print('method 2 \n', np.dot(Y, X))

```

method 1

```
[[ 6  5]
 [14 13]]
```

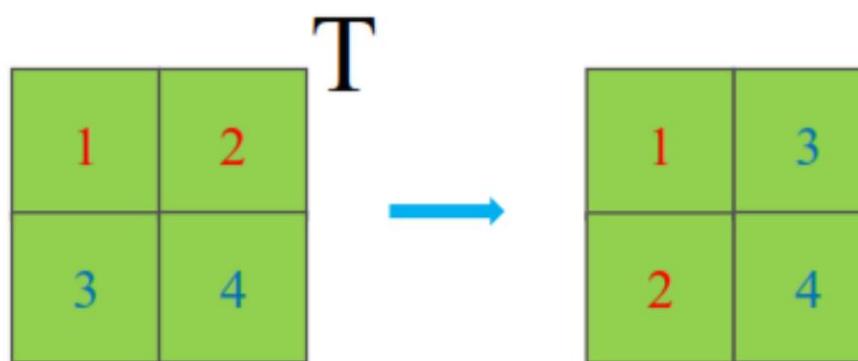
method 1

```
[[11 16]
 [ 5  8]]
```



.Hoạt động mảng

❖ Transpose



```

1 # aivietnam.ai
2 import numpy as np
3
4 X = np.array([[1,2],
5               [3,4]])
6 print(X)
7
8 #chuyển vị
9 print(X.T)

```

```

[[1 2]
 [3 4]]
[[1 3]
 [2 4]]

```

.Hoạt động mảng

❖ Summation

data	
Axis 0	
	Axis 1
1	2
3	4

sum(data)
10

sum(data, axis=0)	
4	6

sum(data, axis=1)	
3	7

```

1 # aivietnam.ai
2 import numpy as np
3
4 X = np.array([[1,2],
5             [3,4]])
6
7 # Tổng các phần tử của mảng
8 print(np.sum(X))
9
10 # Tính tổng theo từng cột
11 print(np.sum(X, axis=0))
12
13 # Tính tổng theo từng dòng
14 print(np.sum(X, axis=1))

```

10
[4 6]
[3 7]



.Hoạt động mảng

❖ Max and min

data

1
2
3

.max() =

3

data

1
2
3

.min() =

1

```

1 # aivietnam.ai
2 import numpy as np
3
4 data = np.array([1, 2, 3])
5
6 print(data.max())
7 print(data.min())

```

3

1

1	2
3	4
5	6

.max(axis=0) =

5	6
---	---

1	2
3	4
5	6

.min(axis=0) =

1	2
---	---

1	2
3	4
5	6

.max(axis=1) =

2	4	6
---	---	---

1	2
3	4
5	6

.min(axis=1) =

1	3	5
---	---	---



VUI MỪNG

- Giới thiệu về Numpy
- Lập chỉ mục mảng Numpy
- Hoạt động mảng Numpy
- Phát thanh truyền hình
- Xử lý dữ liệu

❖ Vector and a scalar

data
 0 1 2
 *  = **result**


data
 0 1 2
 -  = **result**


```

1 # aivietnam.ai
2 import numpy as np
3
4 # create data
5 data = np.array([1,2,3])
6 factor = 2
7
8 # broadcasting
9 result_multiplication = data*factor
10 result_minus = data - factor
11
12 print(data)
13 print(result_multiplication)
14 print(result_minus)

```

```

[1 2 3]
[2 4 6]
[-1 0 1]

```

❖ Matrix and vector

X

0	1	2
1	2	3
2	4	5
3	7	8
4	10	11

+

v

1	0	1
---	---	---

↓

Y

0	1	2
1	2	4
2	5	5
3	8	10
4	11	11

+

1	0	1
1	0	1
1	0	1
1	0	1

=

0	1	2
1	2	4
2	5	7
3	8	10
4	11	13

```

1 # aivietnam.ai
2 import numpy as np
3
4 X = np.array([[1, 2, 3],
5 [4, 5, 6],
6 [7, 8, 9],
7 [10, 11, 12]])
8 v = np.array([1, 0, 1])
9
10 Y = X + v
11 print(Y)

```

```

[[ 2  2  4]
 [ 5  5  7]
 [ 8  8 10]
 [11 11 13]]

```



VUI MỪNG

- Giới thiệu về Numpy
- Lập chỉ mục mảng Numpy
- Hoạt động mảng Numpy
- Phát sóng
- Xử lý dữ liệu



Xử lý dữ liệu

❖ Text data

❖ IRIS data

sepal_length	sepal_width	petal_length	petal_width	species
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3	1.4	0.2	Iris-setosa
4.7	3.2	1.3	0.2	Iris-setosa
4.6	3.1	1.5	0.2	Iris-setosa
5	3.6	1.4	0.2	Iris-setosa
5.4	3.9	1.7	0.4	Iris-setosa
4.6	3.4	1.4	0.3	Iris-setosa
5	3.4	1.5	0.2	Iris-setosa
4.4	2.9	1.4	0.2	Iris-setosa
4.9	3.1	1.5	0.1	Iris-setosa



Xử lý dữ liệu

❖ Text data

❖ IRIS data

col 0 col 1 col 2 col 3 col 4

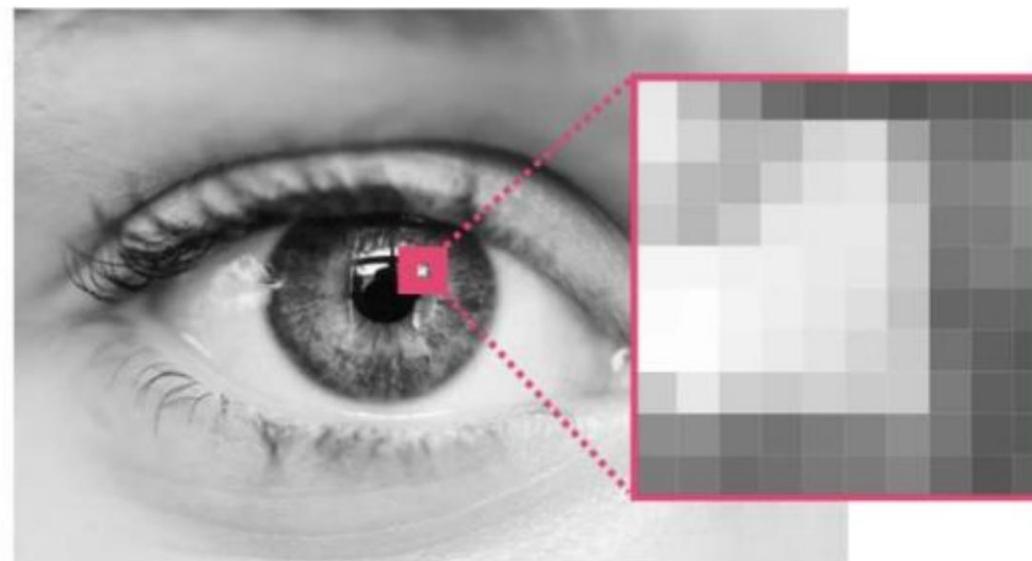
sepal_length	sepal_width	petal_length	petal_width	species
--------------	-------------	--------------	-------------	---------

```

1 # aivietnam.ai
2 # Đọc file IRIS.csv
3
4 import numpy as np
5 import numpy.core.defchararray as np_f
6
7 # lấy các đặc trung và lưu vào biến X
8 X = np.genfromtxt('IRIS.csv', delimiter=',',
9                     dtype='float', usecols=[0,1,2,3],
10                    skip_header=1)
11 print(X.shape)
12
13 # lấy species và lưu vào biến y
14 y = np.genfromtxt('IRIS.csv', delimiter=',',
15                     dtype='str', usecols=4, skip_header=1)
16
17 # thay chuỗi bằng số
18 categories = np.unique(y)
19 for i in range(categories.size):
20     y = np_f.replace(y, categories[i], str(i))
21
22 # đưa về kiểu float
23 y = y.astype('float')
24 print(y)

(150, 4)
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.
 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2. 2.
```

❖ Grayscale images



(Height, Width)

Pixel p = scalar

$0 \leq p \leq 255$

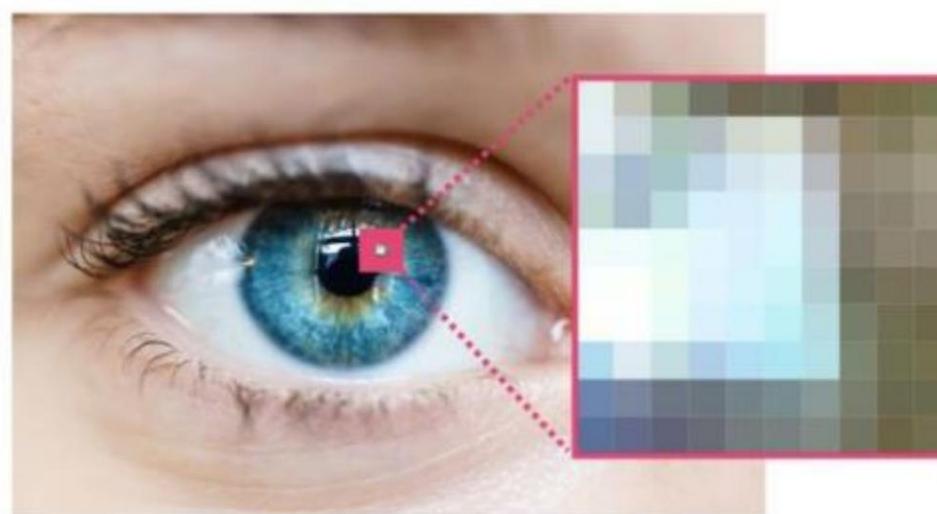
230	194	147	108	90	98	84	96	91	101
237	206	188	195	207	213	163	123	116	128
210	183	180	205	224	234	188	122	134	147
198	189	201	227	229	232	200	125	127	135
249	241	237	244	232	226	202	116	125	126
251	254	241	239	230	217	196	102	103	99
243	255	240	231	227	214	203	116	95	91
204	231	208	200	207	201	200	121	95	95
144	140	120	115	125	127	143	118	92	91
121	121	108	109	122	121	134	106	86	97

Resolution: #pixels
Resolution = Height \times Width



Dữ liệu hình ảnh

❖ Color images



(Height, Width, channel)

RGB color image

$$\text{Pixel } p = \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

$$0 \leq r,g,b \leq 255$$

.Xử lý dữ liệu

233	188	137	96	90	95	63	73	73	82
237	202	159	120	105	110	88	107	112	121
226	191	147	110	101	112	98	123	110	119
221	191	176	182	203	214	169	144	133	145
185	160	161	184	205	223	186	137	147	161
181	174	189	207	206	215	194	136	142	151
246	237	237	231	208	206	192	122	143	144
254	254	241	224	199	192	181	99	122	117
239	248	232	207	187	182	184	110	114	110
193	215	193	167	158	164	181	114	112	111
113	119	110	111	113	123	135	120	108	106
93	97	91	103	107	111	122	112	104	114

Resolution: #pixels
Resolution = Height x Width

<http://jalammar.github.io/visual-numpy/>



Phân loại hình ảnh

Fashion-MNIST dataset

Grayscale images

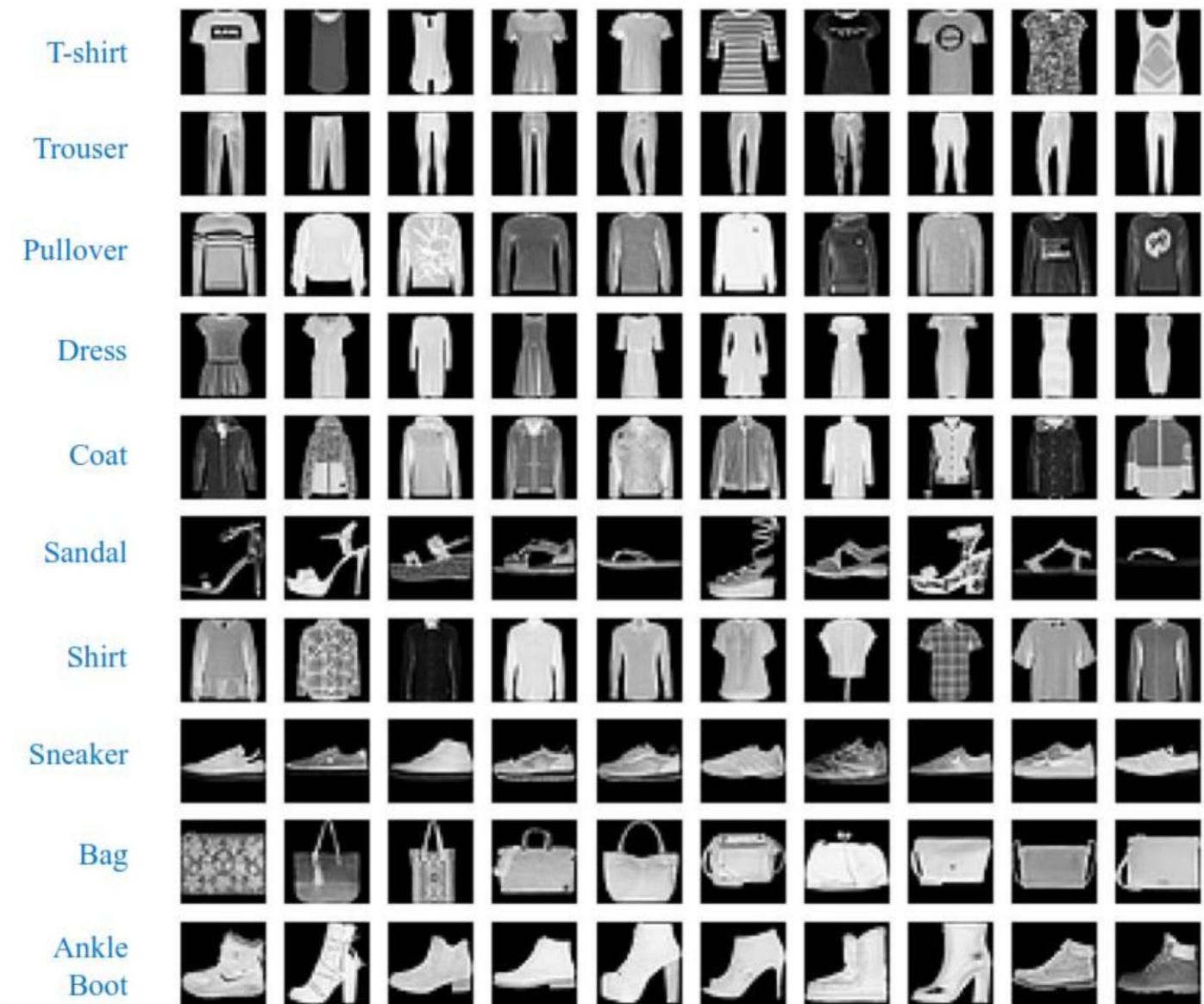
Resolution=28x28

Training set: 60000 samples

Testing set: 10000 samples

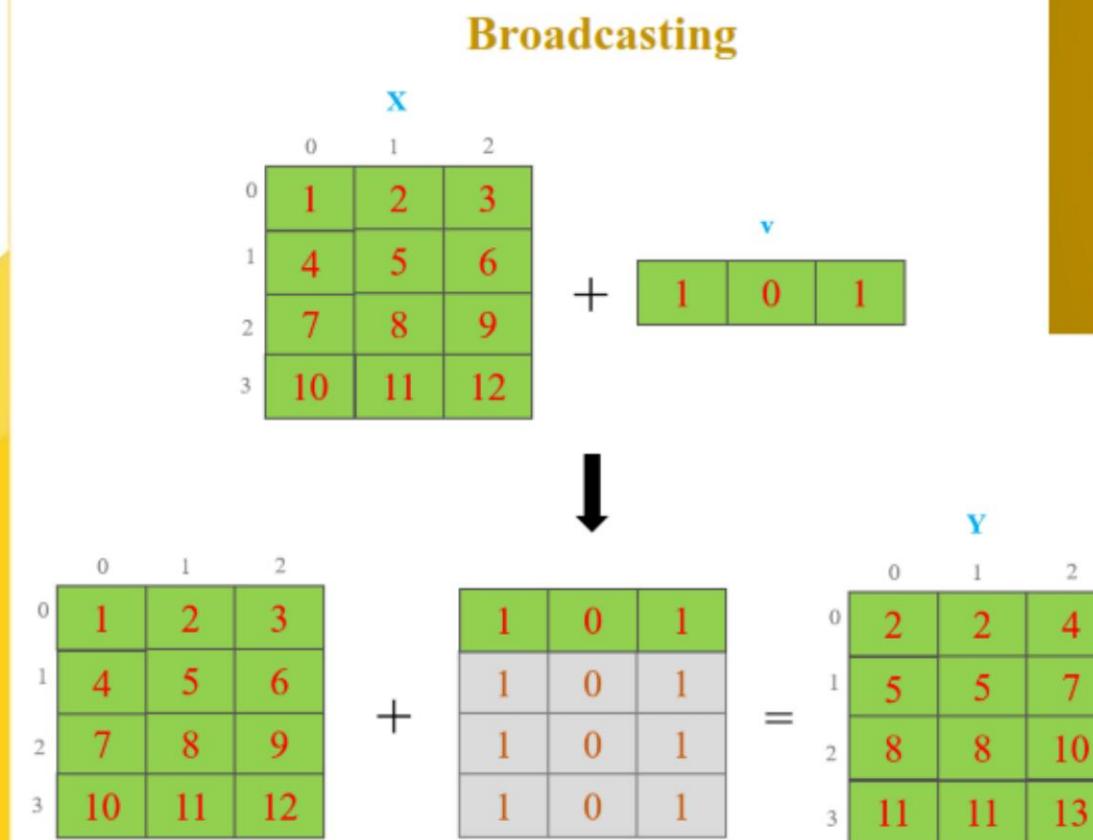


Xử lý dữ liệu





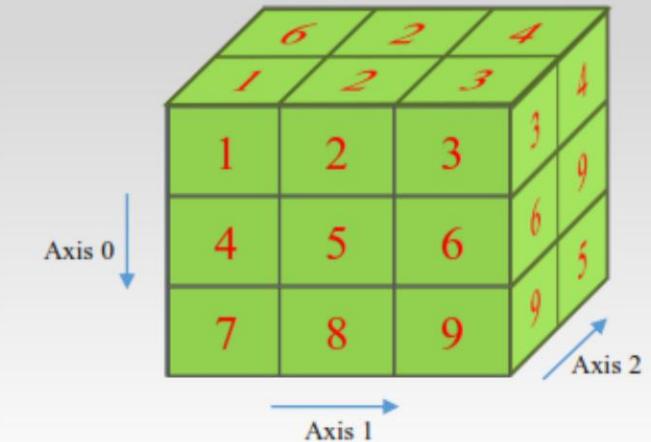
Bản tóm tắt



Indexing

data	data[0, 1]	data[1: 3]	data[0: 1, 0]	data[:, :]
0 1	0 1	0 1	0 1	0 1
1 3 4	1 2	3 4	3 4	3 4
2 5 6	5 6	5 6	5 6	5 6

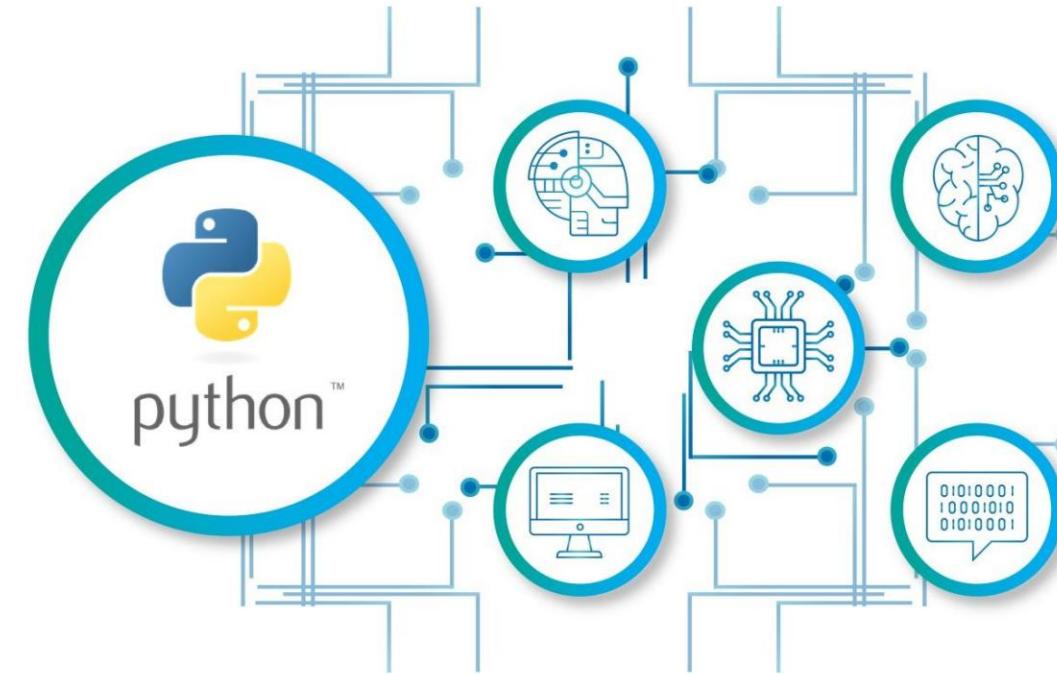
Multi-dimension Array







Python cho AI



Cảm ơn You...!