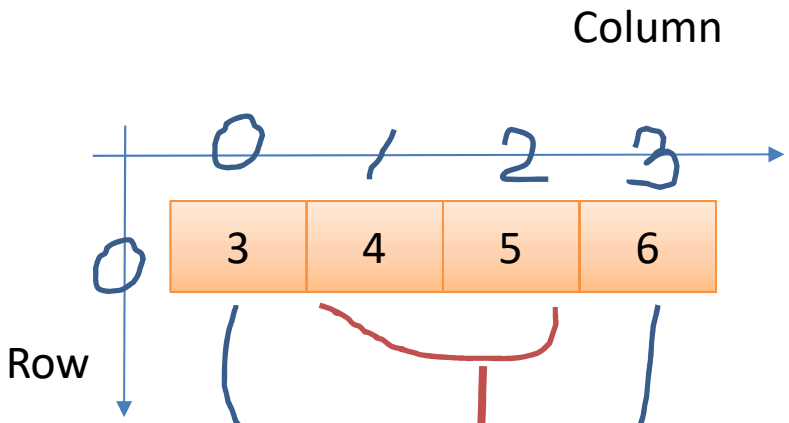
Numpy : 處理array 的套件(package)

- image data :
  - Two-dimensional array of numbers
- Sound data :
  - One-dimensional arrays of intensity versus time
- NumPy (numerical Python) :
  - ndarray object (N-dimensional array, or just array)
  - import numpy as np （使用別名)
- Ndarray attibutes
  - Ndim
  - Shape
  - Size
  - dtype

NumPy

- Arrays must be homogeneous
- One-dimensional array is roughly equal to a Python list
  - import numpy as np
  - s=np.array([1,2,3]) # 一維array
- Why use ndarrays instead of Python list ?
  - Arrays are fast, size efficient

Ndarray: 用list 創造一維array




```
my_first_array=np.array([3,4,5,6])

print(my_first_array)
print(type(my_first_array))
print(my_first_array.ndim)
print(my_first_array.shape)
print(my_first_array.size)
```

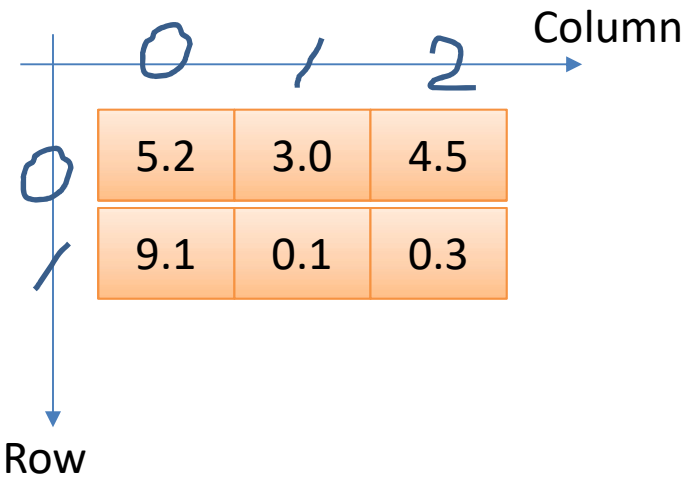
```
[3 4 5 6]
<class 'numpy.ndarray'>
1
(4,)
4
```

```
my_first_array[0]
3

my_first_array[3]
6

my_first_array[1:3]
array([4, 5])
```

Ndarray: 用雙層list 創造二維array



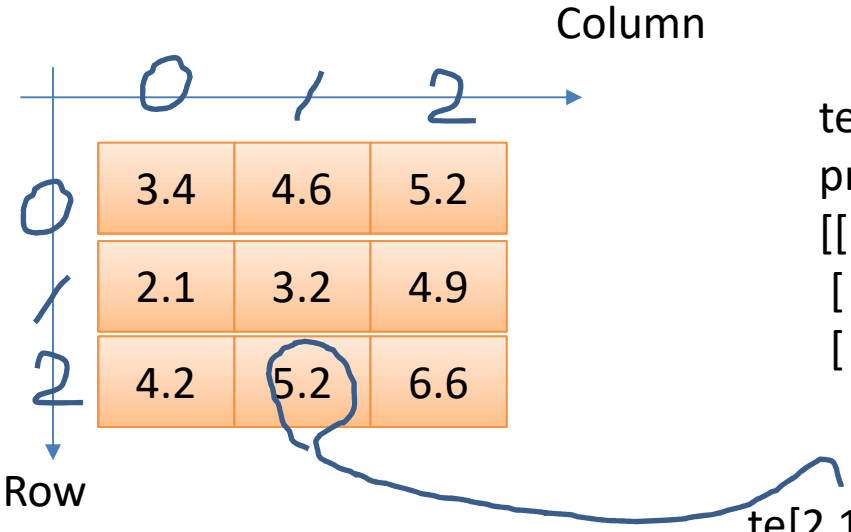
2-D array
Shape (2 rows, 3 columns)
Size: 6
每個元素 data type 是 float

```
row0=[5.2, 3.0, 4.5]
row1=[9.1 ,0.1 ,0.3]

ary_2D=np.array(  [ row0 , row1 ]  )
print(ary_2D)
print(ary_2D.dtype)
print(ary_2D.ndim)
print(ary_2D.shape)
print(ary_2D.size)
[[ 5.2  3.   4.5]
 [ 9.1  0.1  0.3]]
float64
2
(2, 3)
6
```

list of list

Ndarray: 用雙層list 創造二維array




```
te=np.array([[3.4,4.6,5.2],[2.1,3.2,4.9],[4.2,5.2,6.6]])
print(te)
[[ 3.4  4.6  5.2]
 [ 2.1  3.2  4.9]
 [ 4.2  5.2  6.6]]
```

te[2,1]

2-D array
Shape (3 rows, 3 columns)
Size: 9
dtype:

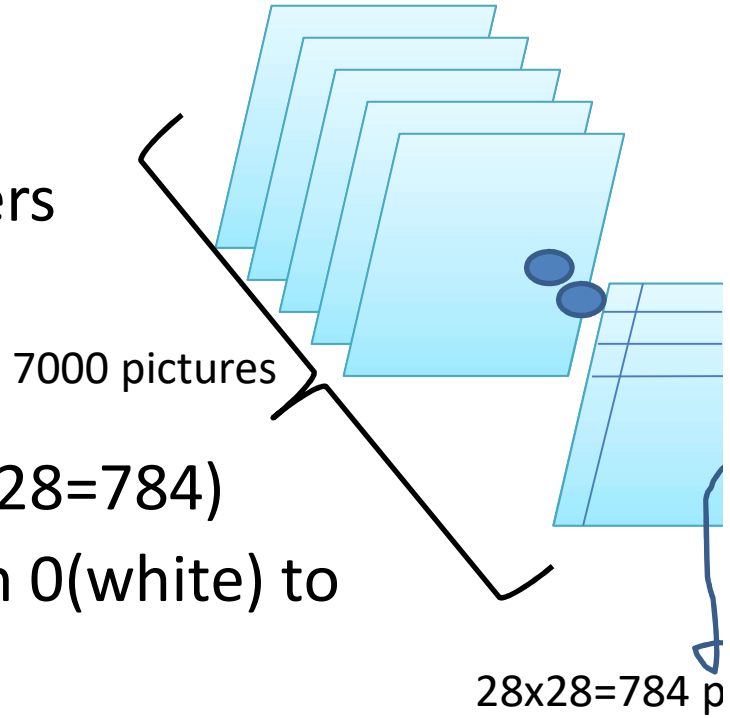
```
print(te[0:2])
[[ 3.4  4.6  5.2]
 [ 2.1  3.2  4.9]]
```

```
print(te[0:2,0:1])
[[ 3.4]
 [ 2.1]]
```

```
tet=te[:  , 2:3]
print(tet)
[[ 5.2]
 [ 4.9]
 [ 6.6]]
```

Ex : how to store 70000 pictures

- Images:
  - Two-dimensional array of numbers
- How to store image data
  - 70000 pictures
  - Each picture is 28x28 pixels (28x28=784)
  - Each pixel has only intensity from 0(white) to black(255)
  - Treat each picture as a one-dim array of size 784
  - Store the total 70000 picture as two-dim array, each row corresponding to a picture



Ex : how to store 70000 pictures ?