Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

Bài 2: Thư viện Numpy

Phòng Lập Trình

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

2023

Nội dung

- Giới thiệu
- Mảng một chiều
- Mảng nhiều chiều

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Giới thiệu

❑ NumPy (Numerical Python)

- Thư viện xử lý và tính toán hiệu quả trên mảng dữ liệu số (ndarray).
- Tích hợp sẵn nhiều hàm toán học, thống kê.
- Tối ưu hóa bộ nhớ và xử lý khi dữ liệu lớn.
- Là nền tảng của nhiều thư viện khác: pandas, SciPy, Scikit-learn, PyTorch,...
- Cài đặt: **`pip install numpy`**
- Sử dụng: **`import numpy as np`**

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Giới thiệu

- So sánh list và numpy array

list

```
print(my_list)
```

```
['Trần Tuấn Anh', 19, 8.5, True]
```

numpy array

```
print(my_arr)
```

```
['Trần Tuấn Anh' '19' '8.5' 'True']
```

Kiểu dữ liệu mỗi phần tử?

=> Cho phép lưu trữ nhiều kiểu dữ liệu khác nhau

=> Đồng bộ về **cùng** 1 kiểu dữ liệu

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Giới thiệu

- So sánh list và numpy array

list

```
# Tạo 1 list số nguyên từ 0-999999
```

Kích thước bộ nhớ của biến lst_num ?

numpy array

```
# Tạo mảng NumPy array với cùng dữ liệu
```

Kích thước bộ nhớ của biến arr_num?

List chiếm dung lượng bộ nhớ nhiều hơn hẳn Numpy Array

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Giới thiệu

- So sánh list và numpy array

Phép toán số học?

list

```
# Nhân 2 cho từng phần tử
```

=> Sử dụng vòng lặp

```
lst_cal[:5]
```

```
[0, 2, 4, 6, 8]
```

numpy array

```
# Nhân 2 cho từng phần tử
```

=> Tính toán trực tiếp

```
arr_cal[:5]
```

```
array([0, 2, 4, 6, 8])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Giới thiệu

- So sánh list và numpy array

Tốc độ xử lý?

list

```
start_time = time.time()
lst_result = [x ** 2 for x in lst_num]
end_time = time.time()
t0 = end_time - start_time
print(f"Thời gian : {t0:.6f} seconds")
lst_result[:5]
```

```
Thời gian : 0.225517 seconds
[0, 1, 4, 9, 16]
```

numpy array

```
start_time = time.time()
arr_result = arr_num ** 2
end_time = time.time()
t1 = end_time - start_time
print(f"Thời gian: {t1:.6f} seconds")
arr_result[:5]
```

```
Thời gian: 0.002773 seconds
array([ 0,  1,  4,  9, 16])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Nội dung

- Giới thiệu
- Mảng một chiều
- Mảng nhiều chiều

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng một chiều

- **Tạo mảng một chiều**
  - Cú pháp tạo mảng: `numpy.array(object, dtype)`
  - Trong đó:
    - object: đối tượng dạng danh sách (list, tuple, array)
    - dtype: kiểu dữ liệu của các phần tử trong mảng (tùy chọn)

```
arr1 = np.array([120, 85, 230, 150, 60])
arr1
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng một chiều

- **Thuộc tính**

Thuộc tính	Ý nghĩa
dtype	Kiểu dữ liệu của các phần tử trong mảng numpy
size	Số phần tử
shape	Giá trị kiểu bộ (tuple) cho biết số chiều, kích thước của numpy array
ndim	Số chiều của numpy array
itemsize	Kích thước phần tử trong mảng numpy
nbytes	Kích thước của mảng numpy

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng một chiều

- Tạo mảng bằng các hàm tiện ích

Tên hàm	Ý nghĩa
np.zeros(n)	Tạo mảng gồm n phần tử có giá trị là 0
np.ones(n)	Tạo mảng gồm n phần tử có giá trị là 1
np.full(n,fill_value)	Tạo mảng n phần tử có giá trị là fill_value
np.arange([start,] stop[, step,])	Tạo mảng số nguyên từ start đến stop (không bao gồm stop)
np.linspace(start, stop, n)	Tạo mảng n phần tử cách đều trong khoảng [start, stop]
np.random.rand(n)	Tạo mảng gồm n số ngẫu nhiên trong [0, 1)
np.random.randint(low, high,size=n)	Tạo mảng gồm n số nguyên ngẫu nhiên từ low đến nhỏ hơn high

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng một chiều

- **Tạo mảng từ tập tin csv, txt**

```
np.genfromtxt(fname [, delimiter])
```

Ví dụ: Tạo mảng chứa chiều cao (inches) của các VĐV bóng chày từ file heights.txt

```
heights = np.genfromtxt('heights.txt', delimiter=',')
```

heights.txt - Notepad

File Edit Format View Help

74, 74, 72, 72, 73, 69, 69, 71, 76, 71, 73, 73, 74, 74, 69, 70, 73, 75, 78, 79, 76, 74,
76, 72, 71, 75, 77, 74, 73, 74, 78, 73, 75, 73, 75, 75, 74, 69, 71, 74, 73, 73, 76, 74,
74, 70, 72, 77, 74, 70, 73, 75, 76, 76, 78, 74, 74, 76, 77, 81, 78, 75, 77, 75, 76, 74,
72, 72, 75, 73, 73, 73, 70, 70, 70, 76, 68, 71, 72, 75, 75, 75, 75, 68, 74, 78, 71,
73, 76, 74, 74, 79, 75, 73, 76, 74, 74, 73, 72, 74, 73, 74, 72, 73, 69, 72, 73, 75, 75,

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng một chiều

- **Truy xuất phần tử trong mảng**
  - Index của các phần tử

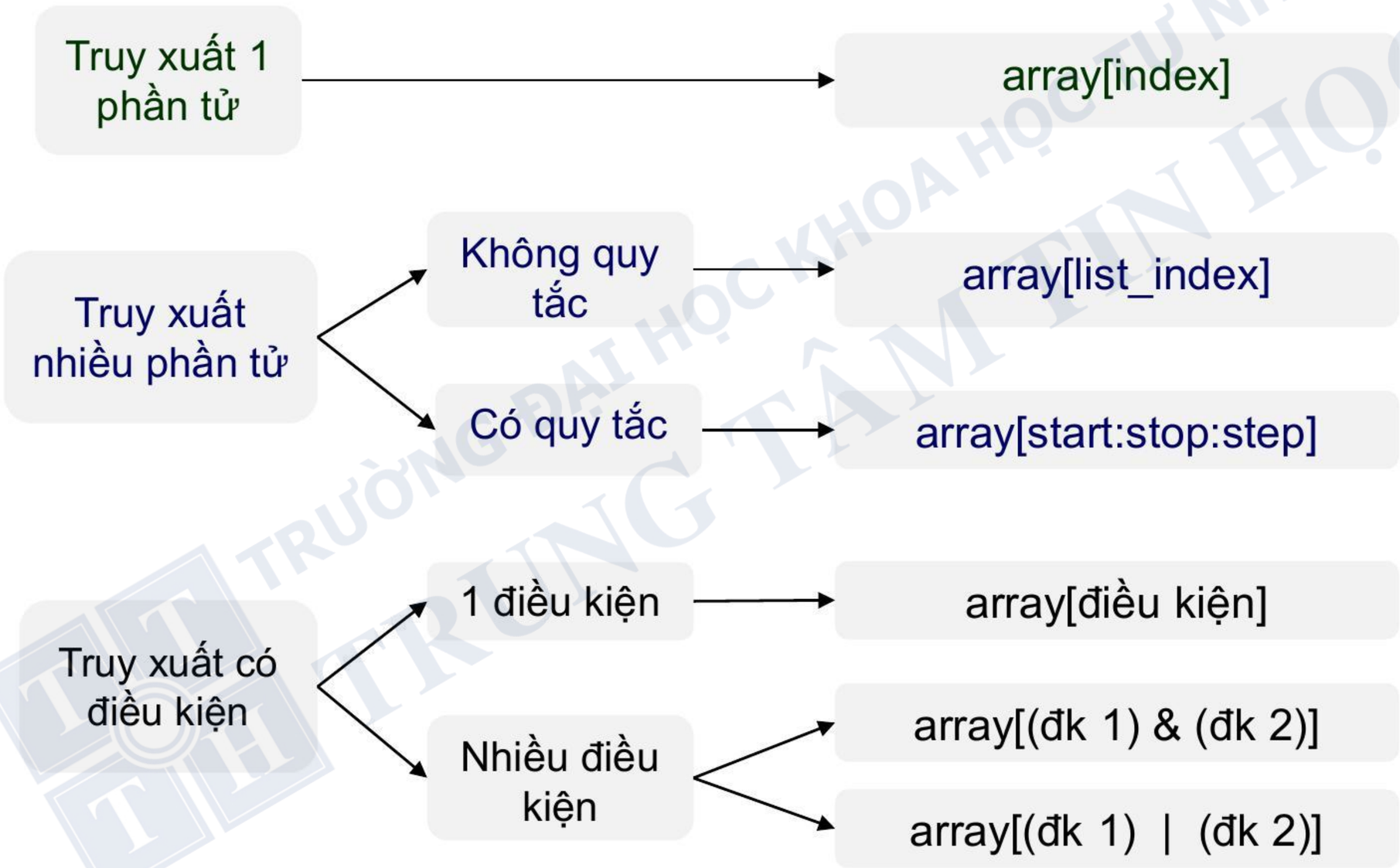
```
array[index]
```

Index:	-5	-4	-3	-2	-1
array([	120,	85,	230,	150,	60])
Index:	0	1	2	3	4

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng một chiều

- **Truy xuất phần tử trong mảng**



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng một chiều

Truy xuất phần tử trong mảng

- Truy xuất 1 phần tử: `array[index]`

```
array([120,  85, 230, 150,  60])
```

- Truy xuất nhiều phần tử theo danh sách index:

```
array[list_index]
```

```
array([120,  85, 230, 150,  60])
```

- Truy xuất nhiều phần tử theo slice

```
array[start:stop:step]
```

```
array([120,  85, 230, 150,  60])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng một chiều

- **Truy xuất phần tử trong mảng**
  - Truy xuất theo 1 điều kiện

```
array[điều_kiện]
```

```
array([120,  85, 230, 150,  60])
# Hiển thị phần tử có các giá trị >100
arr1[arr1 > 100]
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng một chiều

❑ Truy xuất phần tử trong mảng

- Truy xuất theo nhiều điều kiện

```
array[(đk 1) & (đk 2)]
array[(đk 1) | (đk 2)]
array[~(đk 1)]
```

arr1: `array([120,  85, 230, 150,  60])`

```
# Hiển thị các phần tử có giá trị từ 100 đến 200
arr1[(arr1 >= 100) & (arr1 <= 200)] # and
```

```
array([120, 150])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng một chiều

Truy xuất phần tử trong mảng

- Truy xuất theo điều kiện với np.where()

```
idx = np.where(điều kiện)
```

Trả về các index tương ứng các phần tử thỏa điều kiện

Truy xuất phần tử: `array[idx]`

arr1: `array([120, 85, 230, 150, 60])`

```
# Index của các phần tử có giá trị < 100?
idx = np.where(arr1< 100)
idx
```

```
(array([1, 4], dtype=int64),)
```

```
# Hiển thị các phần tử có giá trị < 100
arr1[idx]
```

```
array([85, 60])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TRUNG TÂM TIN HỌC

Mảng một chiều

❑ Cập nhật dữ liệu mảng

- Thêm phần tử vào cuối mảng

```
np.concatenate([array,[value]])
```

arr1: `array([120,  85, 230, 150,  60])`

```
# Thêm giá trị vào cuối mảng arr1
arr1 = np.concatenate ([arr1, [200]])
arr1
```

```
array([120,  85, 230, 150,  60, 200])
```

```
# thêm nhiều giá trị vào cuối mảng arr1
arr1= np.concatenate([arr1, [170, 95, 210]])
arr1
```

```
array([120,  85, 230, 150,  60, 200, 170,  95, 210])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng một chiều

❑ Cập nhật dữ liệu mảng

- Chèn phần tử vào mảng 1 chiều

```
np.insert(arr1d, position[s], value[s])
```

arr1: `array([120,  85, 230, 150,  60])`

```
# Chèn 1 phần tử vào mảng
arr4 = np.insert(arr1, 4, 100)
arr4
```

```
array([120,  85, 230, 150, 100,  60])
```

```
# Chèn nhiều phần tử vào mảng
arr4 = np.insert(arr1, 5, [80, 70])
arr4
```

```
array([120,  85, 230, 150,  60,  80,  70])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM TRUNG TÂM TIN HỌC

Mảng một chiều

❑ Cập nhật dữ liệu mảng

- Xóa phần tử: np.delete

```
np.delete(arr1d, position[s])
```

arr1: `array([120,  85, 230, 150,  60])`

```
# Xóa phần tử ở vị trí 1
np.delete(arr1, 1)
```

```
array([120, 230, 150,  60])
```

```
arr1
```

```
array([120,  85, 230, 150,  60])
```

```
# Xóa phần tử ở vị trí 0,2,3
np.delete(arr1, [0,2,3])
```

```
array([85, 60])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng một chiều

❑ Cập nhật dữ liệu mảng

- Sửa giá trị phần tử trong mảng

Bước 1: truy xuất

Bước 2: gán giá trị mới

```
array([120,  85, 230, 150,  60])
```

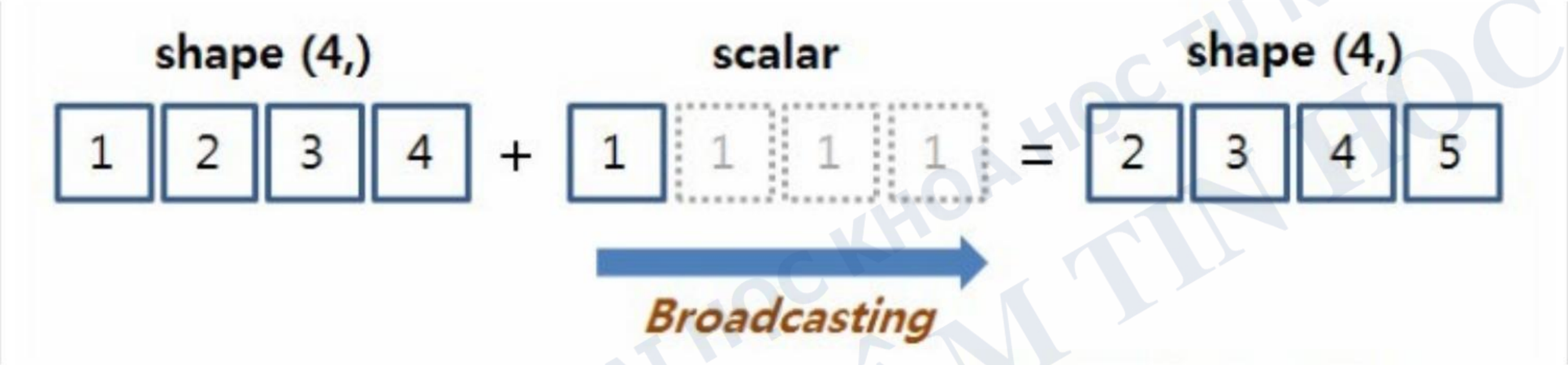
```
# Sửa giá trị ở vị trí 2
arr1[2] = 180
arr1
```

```
array([120,  85, 180, 150,  60])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng một chiều

- Các phép toán số học




```
arr1
```

```
array([120,  85, 230, 150,  60])
```

```
arr1 * 3
```

```
array([360, 255, 690, 450, 180])
```

```
arr_a
```

```
array([110,  50,  60,  80, 100])
```

```
arr_b
```

```
array([ 90,  65,  75,  85, 105])
```

```
arr_a + arr_b
```

```
array([200, 115, 135, 165, 205])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng một chiều

❑ Các hàm toán học

Tên hàm	Ý nghĩa
np.square(arr)	Tính bình phương của các phần tử trong mảng arr.
np.sqrt(arr)	Tính căn bậc hai của các phần tử trong mảng arr.
np.cbrt(arr)	Tính căn bậc ba của các phần tử trong mảng arr.
np.abs(arr)	Trả về giá trị tuyệt đối của các phần tử trong mảng arr.
np.round(arr, n)	Làm tròn các phần tử của mảng arr đến n chữ số thập phân.

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng một chiều

- **Các hàm thống kê**

Hàm	Ý nghĩa
np.sum(arr)	Tính tổng
np.min(arr)	Tìm giá trị nhỏ nhất
np.max(arr)	Tìm giá trị lớn nhất
np.mean(arr)	Tính trung bình
np.median(arr)	Tính trung vị (trung bình giữa)
np.std(arr)	Tính độ lệch chuẩn (standard deviation)
np.var(arr)	Tính phương sai (variance)
np.quantile(arr)	Tính phân vị

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng một chiều

- **Sắp xếp mảng**
  - **`np.sort(arr)`:** sắp xếp mảng tăng dần

```
# Sắp xếp tăng dần
np.sort(arr1)
```

```
array([ 60,  85, 120, 150, 230])
```

  - **`np.sort(arr)[::-1]`:** sắp xếp mảng giảm dần

```
# Sắp xếp giảm dần
np.sort(arr1)[::-1]
```

```
array([230, 150, 120,  85,  60])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng một chiều

- Chuyển đổi kiểu dữ liệu

```
array.astype()
```

```
arr1
```

```
array([120,  85, 230, 150,  60])
```

```
arr1 = arr1.astype(np.float16)
arr1
```

```
array([120.,  85., 230., 150.,  60.], dtype=float16)
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Nội dung

- Giới thiệu
- Mảng một chiều
- **Mảng nhiều chiều**

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng nhiều chiều

- **Tạo mảng bằng np.array()**
  - Tạo mảng nhiều chiều

Cú pháp tạo mảng: `numpy.array(object, dtype)`

```
arr2 = np.array([[120, 85, 230, 150, 60],
                 [90, 65, 75, 85, 105],
                 [110, 75, 85, 100, 90],
                 [70, 50, 60, 80, 100]])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng nhiều chiều

- **Thuộc tính**

Thuộc tính	Ý nghĩa
dtype	Kiểu dữ liệu của các phần tử trong mảng numpy
size	Số phần tử
shape	Giá trị kiểu bộ (tuple) cho biết số chiều, kích thước của numpy array
ndim	Số chiều của numpy array
itemsize	Kích thước phần tử trong mảng numpy
nbytes	Kích thước của mảng numpy

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng nhiều chiều

- Tạo mảng bằng các hàm tiện ích

Tên hàm	Ý nghĩa
np.zeros(shape)	Tạo mảng nhiều chiều kích thước shape, có giá trị là 0
np.ones(shape)	Tạo mảng nhiều chiều kích thước shape, có giá trị là 1
np.full(shape, fill_value)	Tạo mảng nhiều chiều kích thước shape, có giá trị fill_value
np.random.rand(shape)	Tạo mảng nhiều chiều số ngẫu nhiên trong [0, 1) có kích thước shape
np.random.randint(low, high, shape)	Tạo mảng nhiều chiều kích thước shape, là số nguyên ngẫu nhiên từ low đến nhỏ hơn high

shape: Giá trị kiểu tuple (n,m,...), cho biết kích thước mảng nhiều chiều

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TRUNG TÂM TIN HỌC

Mảng nhiều chiều

❑ Truy xuất phần tử trong mảng

- Truy xuất 1 phần tử → array[index_dòng, index_cột]
- Truy xuất nhiều phần tử
  - Không quy tắc → array[list_index_dòng, list_index_cột]
  - Có quy tắc → array[start:stop:step, start:stop:step]
- Truy xuất có điều kiện
  - 1 điều kiện → array[điều kiện]
  - Nhiều điều kiện
    - → array[(đk 1) & (đk 2)]
    - → array[(đk 1) | (đk 2)]

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng nhiều chiều

- **Truy xuất phần tử trong mảng**
  - Truy xuất 1 phần tử

Mảng 2 chiều: **`array[index dòng, index cột]`**

```
array([[120,  85, 230, 150,  60],
       [ 90,  65,  75,  85, 105],
       [110,  75,  85, 100,  90],
       [ 70,  50,  60,  80, 100]])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng nhiều chiều

- **Truy xuất phần tử trong mảng**
  - Truy xuất nhiều phần tử theo danh sách index

Mảng 2 chiều: **`array[list_index_dòng, list_index_cột]`**

```
array([[120,  85, 230, 150,  60],
       [ 90,  65,  75,  85, 105],
       [110,  75,  85, 100,  90],
       [ 70,  50,  60,  80, 100]])
```

  - Truy xuất nhiều phần tử theo slice

Mảng 2 chiều: **`array[start:stop:step, start:stop:step]`**

```
array([[120,  85, 230, 150,  60],
       [ 90,  65,  75,  85, 105],
       [110,  75,  85, 100,  90],
       [ 70,  50,  60,  80, 100]])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TRUNG TÂM TIN HỌC

Mảng nhiều chiều

- **Truy xuất phần tử trong mảng**
  - Truy xuất theo 1 điều kiện `array[điều_kiện]`

arr2:

```
array([[120,  85, 230, 150,  60],
       [ 90,  65,  75,  85, 105],
       [110,  75,  85, 100,  90],
       [ 70,  50,  60,  80, 100]])
```

```
# Hiển thị các phần tử có các giá trị <70
arr2[arr2 < 70]
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TRUNG TÂM TIN HỌC

Mảng nhiều chiều

Truy xuất phần tử trong mảng

- Truy xuất theo nhiều điều kiện

```
array[(đk 1) & (đk 2)]
array[(đk 1) | (đk 2)]
array[~(đk 1)]
```

arr2:

```
array([[120,  85, 230, 150,  60],
       [ 90,  65,  75,  85, 105],
       [110,  75,  85, 100,  90],
       [ 70,  50,  60,  80, 100]])
```

```
# Hiển thị các phần tử có giá trị từ 100 đến 200
arr2[(arr2 > 100) & (arr2 < 200)] # and
```

```
array([120, 150, 105, 110])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng nhiều chiều

❑ Truy xuất phần tử trong mảng

- Truy xuất theo điều kiện với np.where()

```
idx = np.where(điều kiện)
```

Trả về các index tương ứng các phần tử thỏa điều kiện

Truy xuất phần tử: `array[idx]`

arr2:

```
array([[120,  85, 230, 150,  60],
       [ 90,  65,  75,  85, 105],
       [110,  75,  85, 100,  90],
       [ 70,  50,  60,  80, 100]])
```

```
# Index của các phần tử có giá trị  < 70?
idx = np.where(arr2 < 70)
idx
```

```
(array([0, 1, 3, 3], dtype=int64), array([4, 1, 1, 2], dtype=int64))
```

```
# Hiển thị các phần tử có giá trị < 70
arr2[idx]
```

```
array([60, 65, 50, 60])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng nhiều chiều

❑ Cập nhật dữ liệu mảng

- Thêm phần tử vào cuối mảng

```
np.concatenate([arr2d, array_to_concat], axis=0)#thêm dòng
np.concatenate([arr2d, array_to_concat], axis=1)#thêm cột
```

```
# Thêm 1 dòng vào arr2
row_append = [[75, 80, 95, 110, 160]]
arr3 = np.concatenate([arr2, row_append], axis=0)
arr3
```

```
array([[120,  85, 230, 150,  60],
       [ 90,  65,  75,  85, 105],
       [110,  75,  85, 100,  90],
       [ 70,  50,  60,  80, 100],
       [ 75,  80,  95, 110, 160]])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng nhiều chiều

❑ Cập nhật dữ liệu mảng

- Chèn phần tử vào mảng nhiều chiều

```
np.insert(arr2d, position[s], new_row[s], axis=0)
np.insert(arr2d, position[s], new_col[s], axis=1)
```

```
# Chèn 1 dòng vào arr2
new_row = np.array([ 180,  57,  58,  68, 90])
arr5 = np.insert(arr2, 4, new_row, axis=0)
arr5
```

```
array([[120,  85, 230, 150,  60],
       [ 90,  65,  75,  85, 105],
       [110,  75,  85, 100,  90],
       [ 70,  50,  60,  80, 100],
       [180,  57,  58,  68,  90]])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng nhiều chiều

❑ Cập nhật dữ liệu mảng

- Xóa phần tử:

```
np.delete(arr2d, position[s], axis=0) # xóa dòng
np.delete(arr2d, position[s], axis=1) # xóa cột
```

```
array([[120,  85, 230, 150,  60],
       [ 90,  65,  75,  85, 105],
       [110,  75,  85, 100,  90],
       [ 70,  50,  60,  80, 100]])
```

```
# Xóa cột 1 và 3 trong arr2
np.delete(arr2, [1,3], axis=1)
```

```
array([[120, 230,  60],
       [ 90,  75, 105],
       [110,  85,  90],
       [ 70,  60, 100]])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng nhiều chiều

Cập nhật dữ liệu mảng

- Sửa giá trị phần tử trong mảng

  Bước 1: truy xuất

  Bước 2: gán giá trị mới

```
# Sửa giá trị ở vị trí dòng 2, cột 4 là 90
arr2[2,4] = 90
arr2
```

```
array([[120,  85, 230, 150,  60],
       [ 90,  65,  75,  85, 105],
       [110,  75,  85, 100,  90],
       [ 70,  50,  60,  80, 100]])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng nhiều chiều

- Thực hiện phép toán số học từng phần trên mảng

```
arr1
```

```
array([120,  85, 230, 150,  60])
```

```
arr2
```

```
array([[120,  85, 230, 150,  60],
       [ 90,  65,  75,  85, 105],
       [110,  75,  85, 100,  90],
       [ 70,  50,  60,  80, 100]])
```

```
arr1 + arr2
```

```
array([[240, 170, 460, 300, 120],
       [210, 150, 305, 235, 165],
       [230, 160, 315, 250, 150],
       [190, 135, 290, 230, 160]])
```

```
arr1 - arr2
```

```
array([[  0,   0,   0,   0,   0],
       [ 30,  20, 155,  65, -45],
       [ 10,  10, 145,  50, -30],
       [ 50,  35, 170,  70, -40]])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng nhiều chiều

❑ Các hàm toán học

Tên hàm	Ý nghĩa
np.square(arr)	Tính bình phương của các phần tử trong mảng arr.
np.sqrt(arr)	Tính căn bậc hai của các phần tử trong mảng arr.
np.cbrt(arr)	Tính căn bậc ba của các phần tử trong mảng arr.
np.abs(arr)	Trả về giá trị tuyệt đối của các phần tử trong mảng arr.
np.round(arr, n)	Làm tròn các phần tử của mảng arr đến n chữ số thập phân.

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng nhiều chiều

Các hàm thống kê cơ bản

Hàm	Ý nghĩa
np.sum(arr, axis)	Tính tổng
np.min(arr, axis)	Tìm giá trị nhỏ nhất
np.max(arr, axis)	Tìm giá trị lớn nhất
np.mean(arr, axis)	Tính trung bình
np.median(arr, axis)	Tính trung vị (trung bình giữa)
np.std(arr, axis)	Tính độ lệch chuẩn (standard deviation)
np.var(arr, axis)	Tính phương sai (variance)
np.quantile(arr, axis)	Tính phân vị

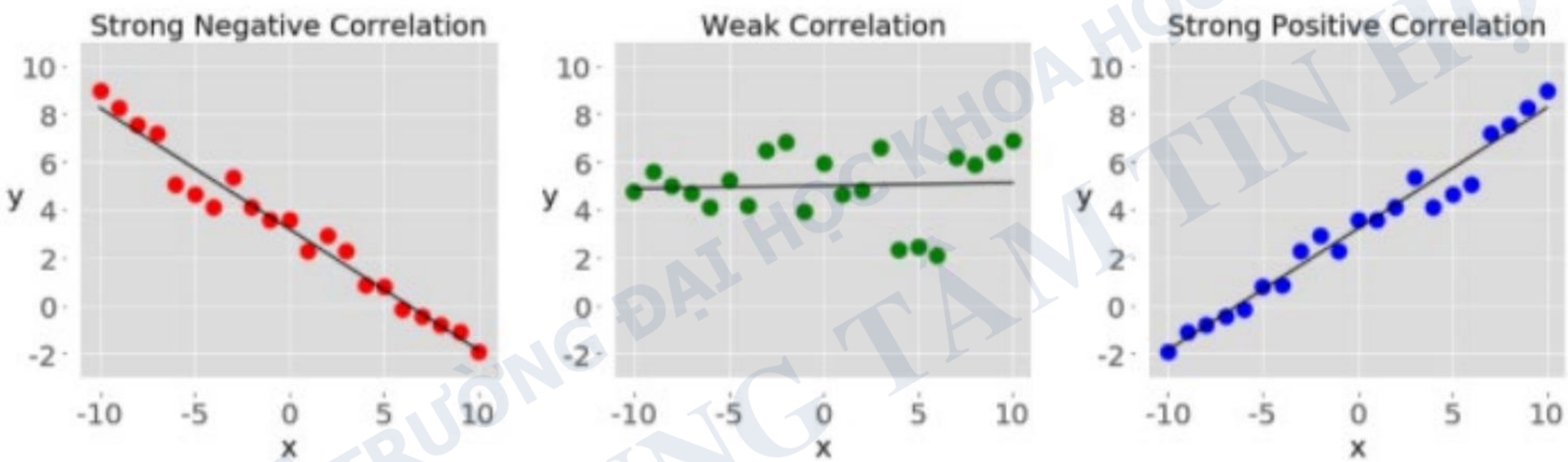
Mảng 2 chiều:
- axis=None (mặc định): tính trên toàn bộ mảng giá trị
- axis=0: Thống kê các dòng trong cột
- axis=1: Thống kê các cột trong dòng

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng nhiều chiều

- Tính hệ số tương quan Pearson: **`np.corrcoef()`**

Đo lường mức độ tương quan giữa 2 mảng (giá trị [-1, 1])




```
arr_a
```

```
array([ 90,  65,  75,  85, 105])
```

```
arr_b
```

```
array([110,  75,  85, 100,  90])
```

```
np.corrcoef(arr_a, arr_b)
```

```
array([[1.        , 0.56130962],
       [0.56130962, 1.        ]])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng nhiều chiều

- **Sắp xếp mảng**
  - `np.sort(arr, axis=None)`: sắp xếp mảng tăng dần

```
np.sort(arr2, axis = 0) # sắp xếp theo cột
```

```
array([[ 70,  50,  60,  80,  60],
       [ 90,  65,  75,  85,  90],
       [110,  75,  85, 100, 100],
       [120,  85, 230, 150, 105]])
```

```
np.sort(arr2, axis = 1) # sắp xếp theo dòng
```

```
array([[ 60,  85, 120, 150, 230],
       [ 65,  75,  85,  90, 105],
       [ 75,  85,  90, 100, 110],
       [ 50,  60,  70,  80, 100]])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TT TÂM TRUNG TIN HỌC

Mảng nhiều chiều

- **Biến đổi mảng**
  - Thay đổi kích thước: **`np.reshape()`**

```
arr1
```

```
array([120,  85, 230, 150,  60,  90])
```

```
arr1.reshape(2,3)
```

```
array([[120,  85, 230],
       [150,  60,  90]])
```

```
arr2
```

```
array([[120,  85, 230, 150,  60],
       [ 90,  65,  75,  85, 105],
       [110,  75,  85, 100,  90],
       [ 70,  50,  60,  80, 100]])
```

```
arr2.reshape(5,4)
```

```
array([[120,  85, 230, 150],
       [ 60,  90,  65,  75],
       [ 85, 105, 110,  75],
       [ 85, 100,  90,  70],
       [ 50,  60,  80, 100]])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng nhiều chiều

❑ Biến đổi mảng

- Chuyển vị: **`np.transpose() hoặc arr.T`**

```
arr2
```

```
array([[120,  85, 230, 150,  60],
       [ 90,  65,  75,  85, 105],
       [110,  75,  85, 100,  90],
       [ 70,  50,  60,  80, 100]])
```

```
np.transpose(arr2)
```

```
array([[120,  90, 110,  70],
       [ 85,  65,  75,  50],
       [230,  75,  85,  60],
       [150,  85, 100,  80],
       [ 60, 105,  90, 100]])
```

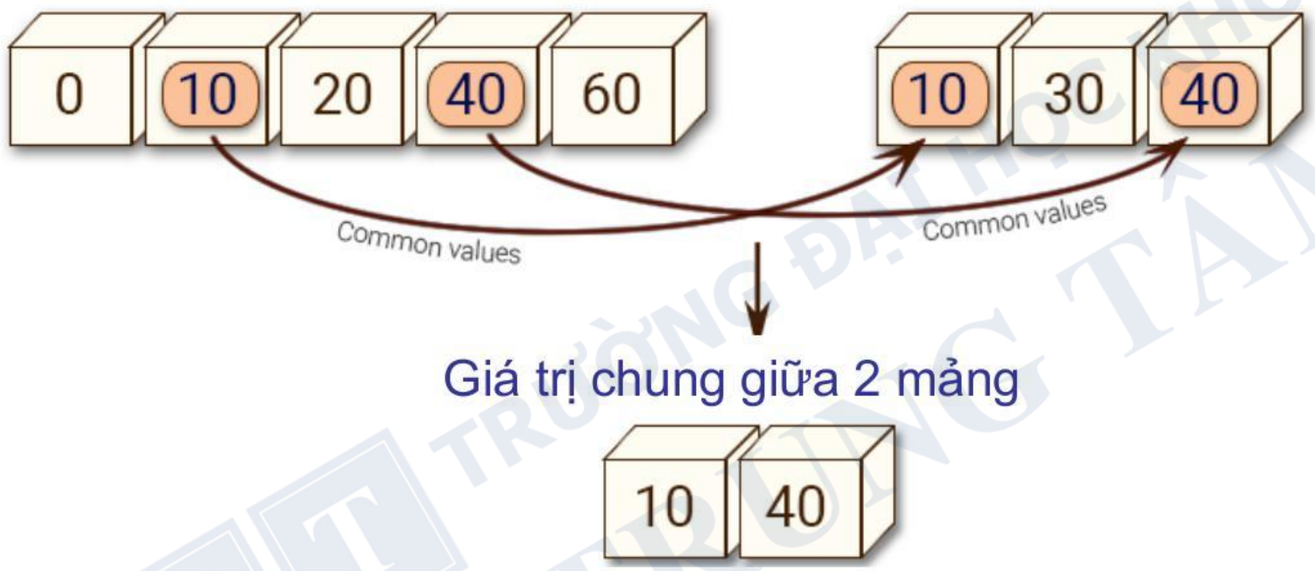
```
arr2.T
```

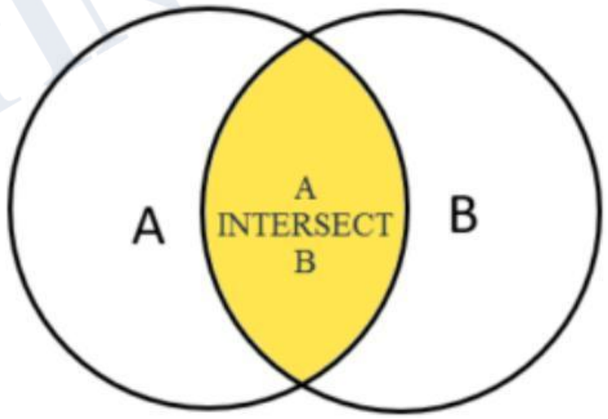
```
array([[120,  90, 110,  70],
       [ 85,  65,  75,  50],
       [230,  75,  85,  60],
       [150,  85, 100,  80],
       [ 60, 105,  90, 100]])
```

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng nhiều chiều

- **Kết hợp mảng**
  - Sử dụng `np.intersect1d()`
    - Tìm các phần tử chung giữa hai mảng.

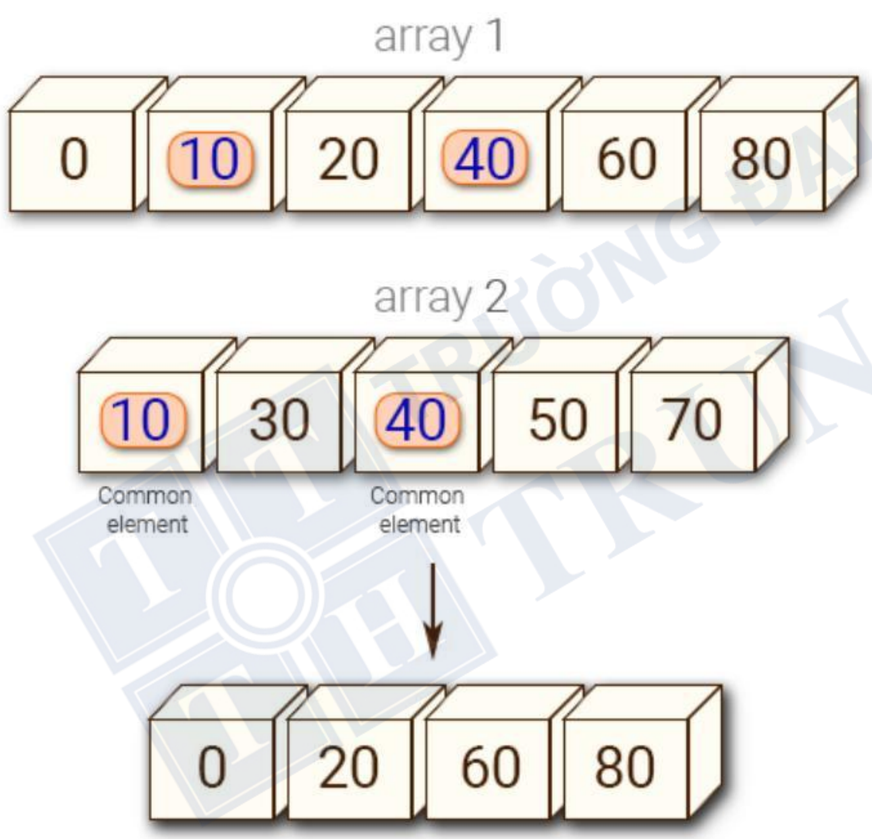


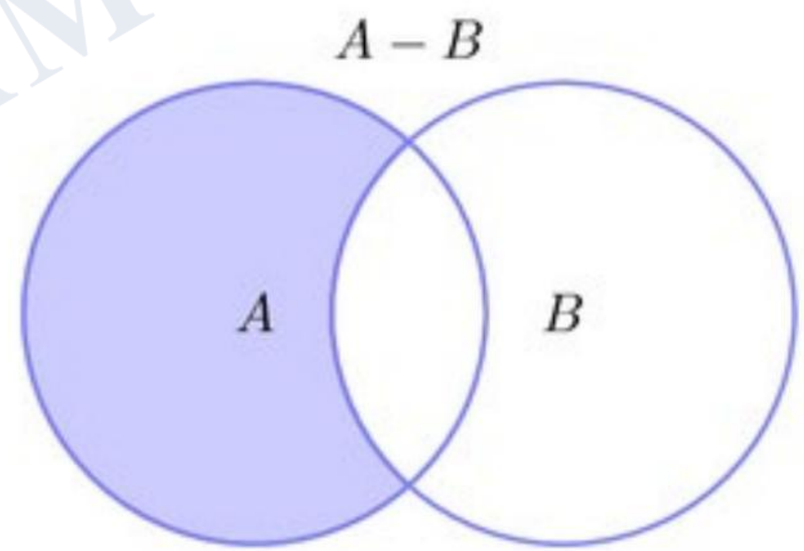


TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

Mảng nhiều chiều

- **Kết hợp mảng**
  - Sử dụng **`np.setdiff1d()`**
    - Xác định các phần tử duy nhất trong một mảng mà không có trong mảng khác.





TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
TRUNG TÂM TIN HỌC