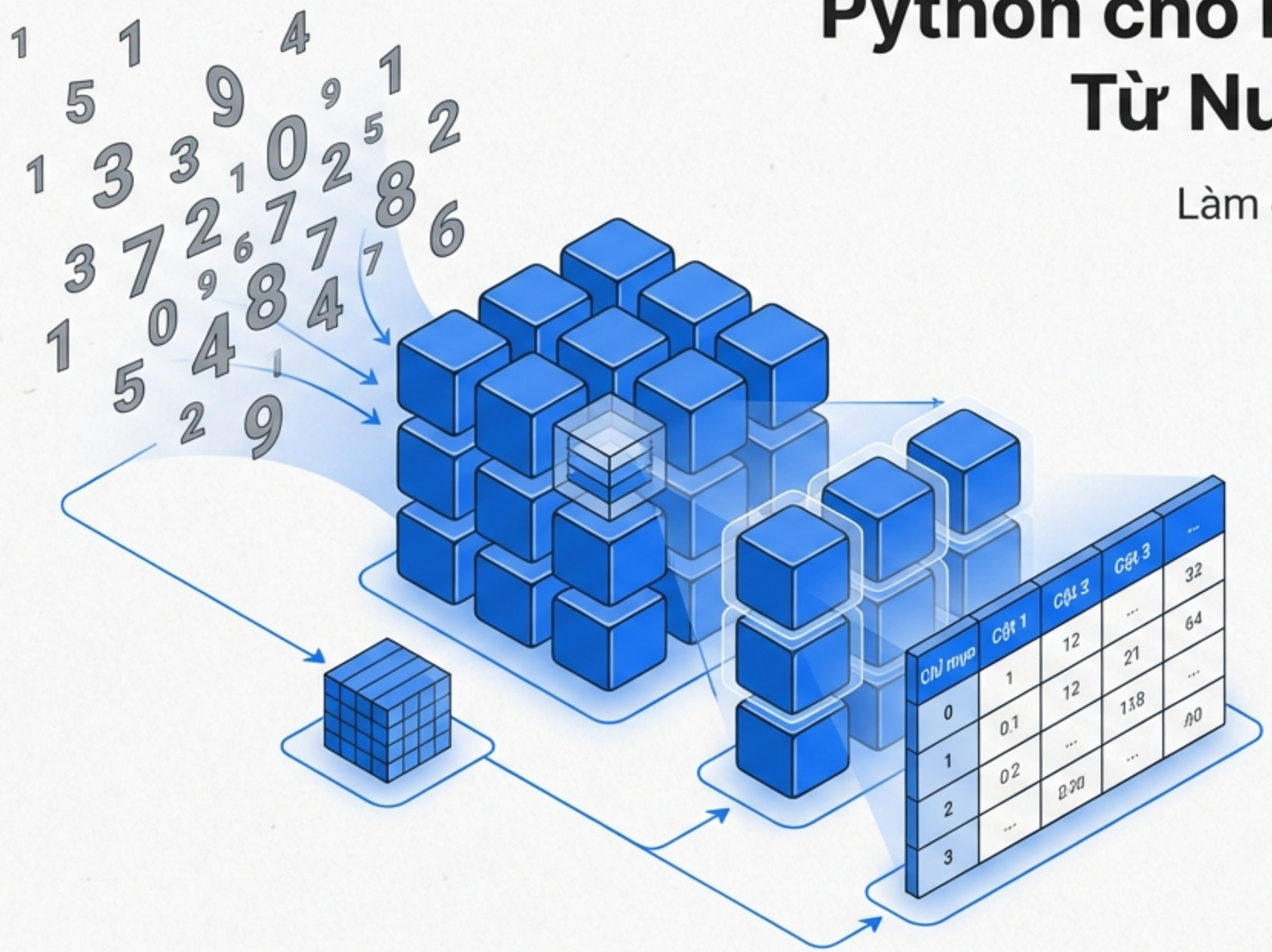
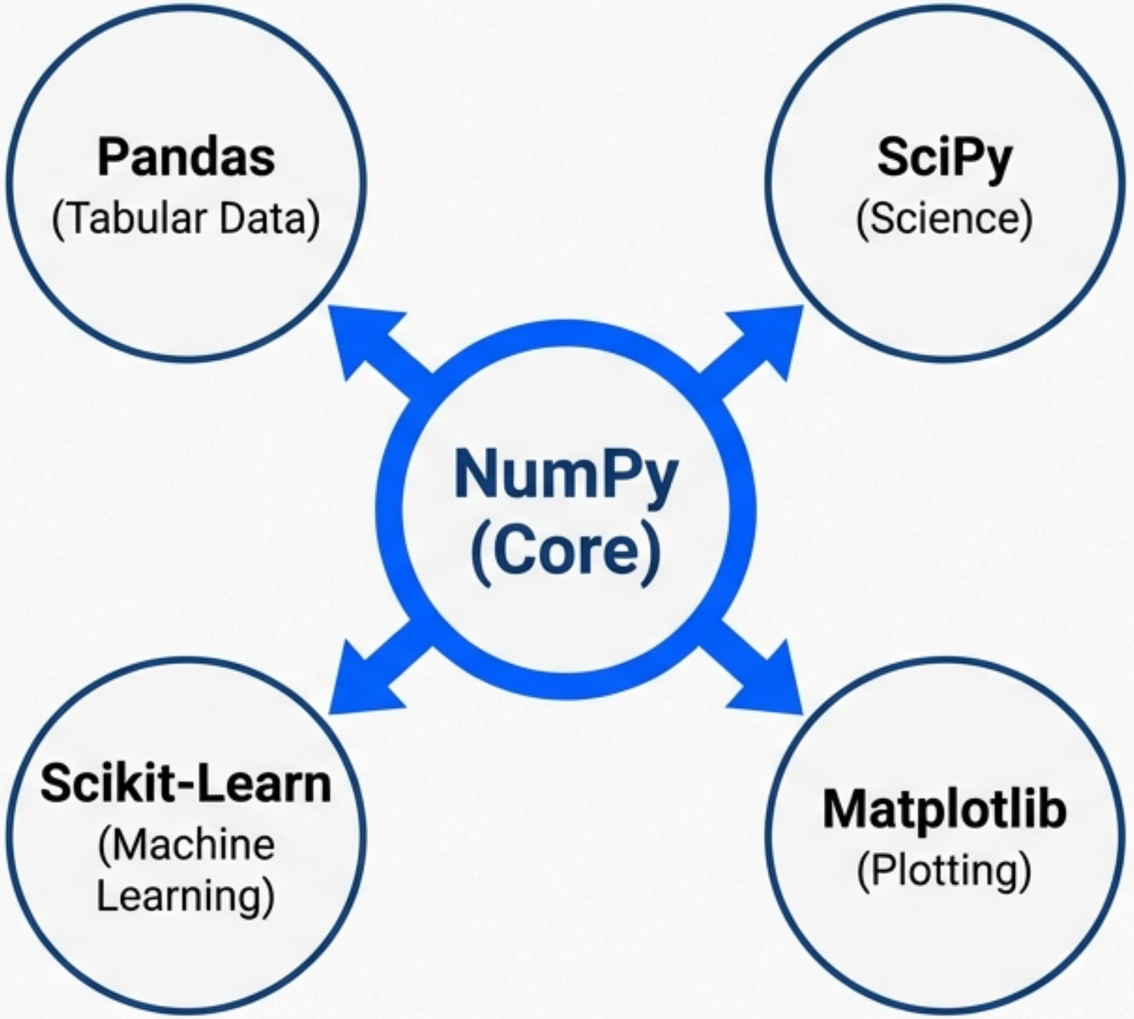
Python cho Khoa học Dữ liệu: Từ NumPy đến Pandas

Làm chủ cấu trúc mảng hiệu năng cao và thao tác dữ liệu bảng.



Tài liệu tự học & Tra cứu. Tổng hợp kiến thức cốt lõi về tính toán số học và xử lý dữ liệu.

Hệ sinh thái NumPy: Trái tim của Tính toán Số học



Tính năng cốt lõi

- **ndarray:** Mảng N-chiều hiệu quả
- **Vectorization:** Tính toán song song, không vòng lặp
- **Tools:** Đại số tuyến tính, Fourier, Số ngẫu nhiên

⚠ CẢNH BÁO

```
import numpy as np
from numpy import *
```

CẢNH BÁO: Tránh import tất cả (*) để ngăn chặn xung đột tên hàm (ví dụ: min, max).

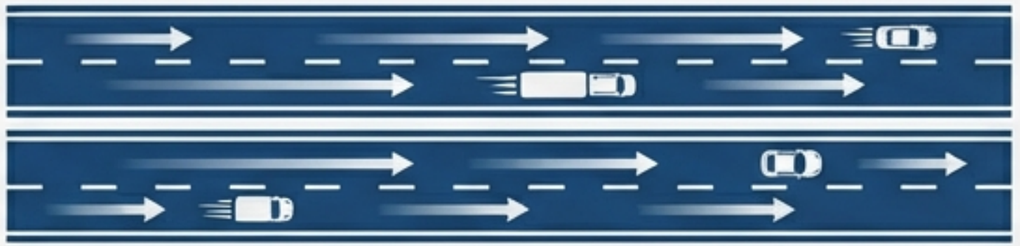
Cuộc đua tốc độ: Python List vs. NumPy Array

Python List (Chậm)

```
my_list = list(range(1000000))
# Phải lặp qua từng phần tử (Loop)
for _ in range(10):
    my_list2 = [x * 2 for x in my_list]
```

Xử lý tuần tự, tốn bộ nhớ.

NumPy ndarray (Nhanh)

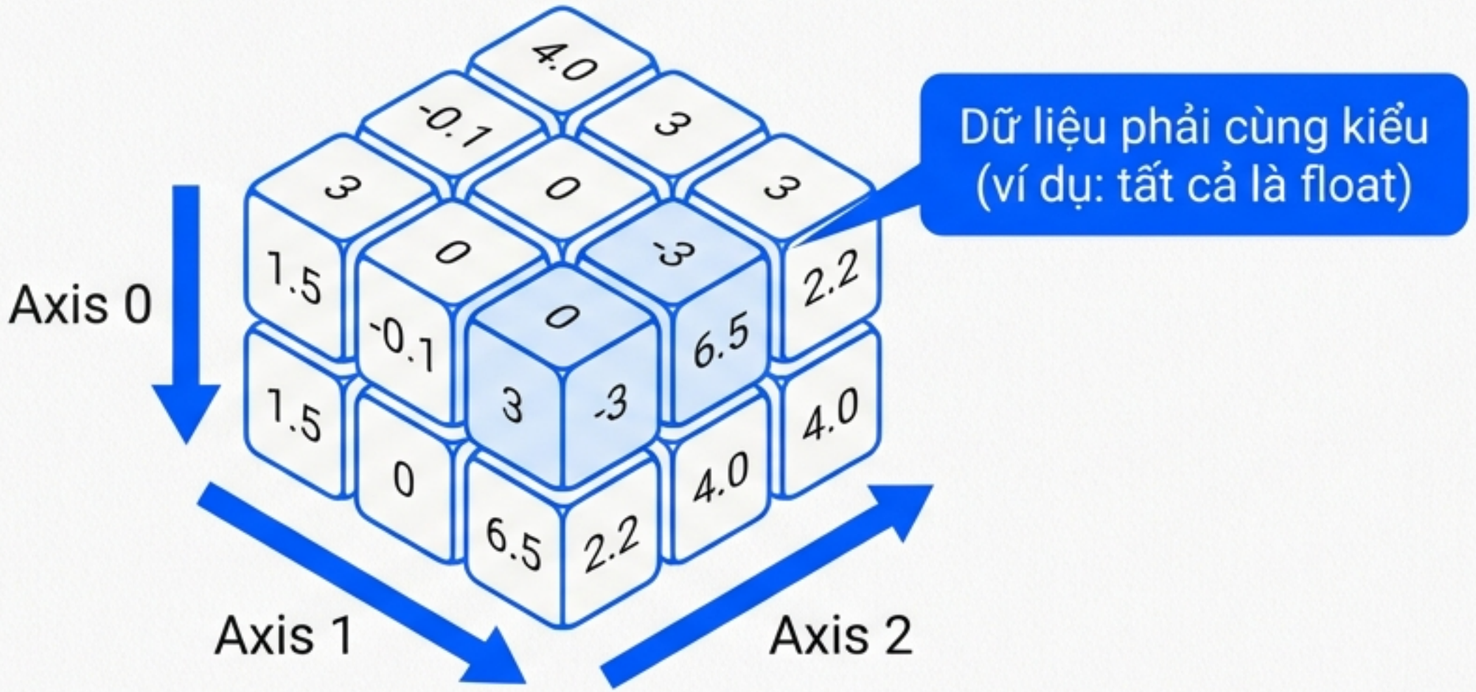


```
my_arr = np.arange(1000000)
# Vectorization (Không vòng lặp)
for _ in range(10):
    my_arr2 = my_arr * 2
```

Xử lý trên khối bộ nhớ liền kề (Contiguous Memory).

Giải phẫu đối tượng ndarray

Thùng chứa đa chiều cho dữ liệu đồng nhất (Homogeneous Data).



..shape

Kích thước các chiều.
Ví dụ: (2, 3)

..dtype

Kiểu dữ liệu.
Ví dụ: float64, int32

..ndim

Số chiều của mảng.
Ví dụ: 2

```
data = np.array([[1.5, -0.1, 3], [0, -3, 6.5]])
# shape: (2, 3) | dtype: float64
```

Khởi tạo Mảng & Ép kiểu Dữ liệu

Creation Toolkit

Hàm	Mô tả
`np.array(seq)`	Chuyển list/tuple thành mảng
`np.arange(n)`	Tạo dải số (giống range)
`np.zeros, np.ones`	Mảng toàn 0 hoặc 1
`np.eye`	Ma trận đơn vị N x N

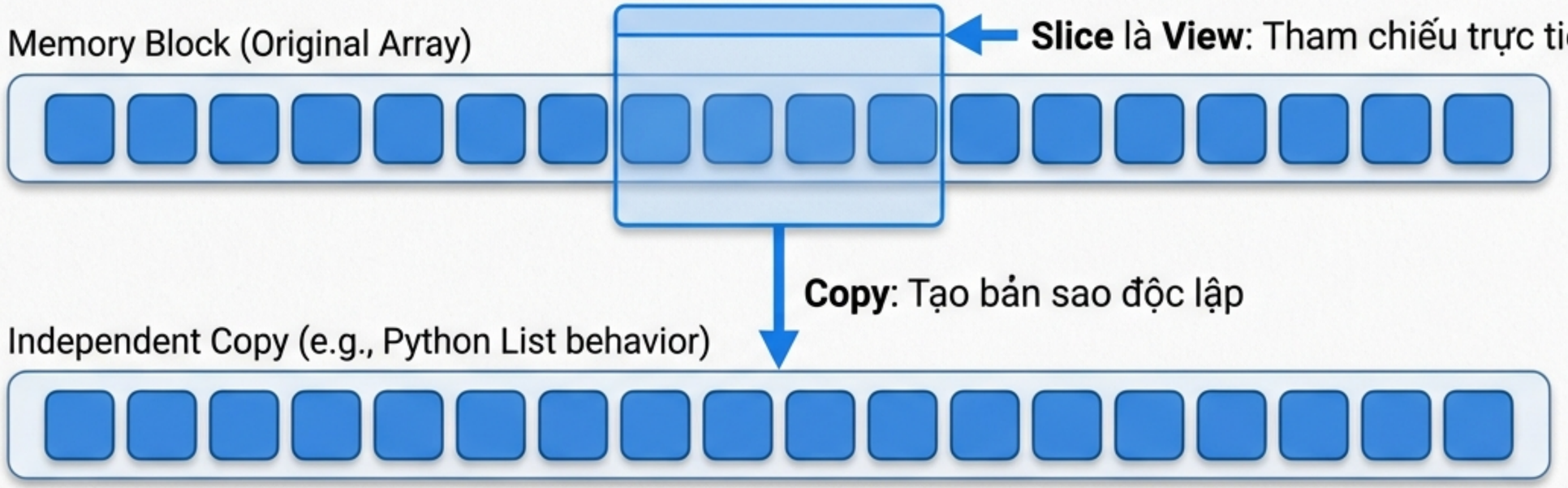
Phương thức .astype()

```
arr = np.array([3.7, -1.2, 0.5])
arr_int = arr.astype(np.int32)
# Kết quả: [3, -1, 0] (Mất phần thập phân)
```

LƯU Ý:

.astype() luôn tạo ra **bản sao mới (Copy)** trong **bộ nhớ**, ngay cả khi kiểu dữ liệu không đổi.

Indexing & Slicing: View vs. Copy




```
arr = np.arange(10)
slice_arr = arr[5:8]
slice_arr[1] = 9999      # Thay đổi trên slice
print(arr)               # Mảng gốc BỊ THAY ĐỔI!
```

LƯU Ý: Muốn sao chép dữ liệu thực sự? Hãy dùng `.copy()`

Truy xuất Nâng cao: Boolean & Fancy Indexing

Roboto Regular: Các phương thức này **luôn tạo bản sao (COPY), không phải View.**

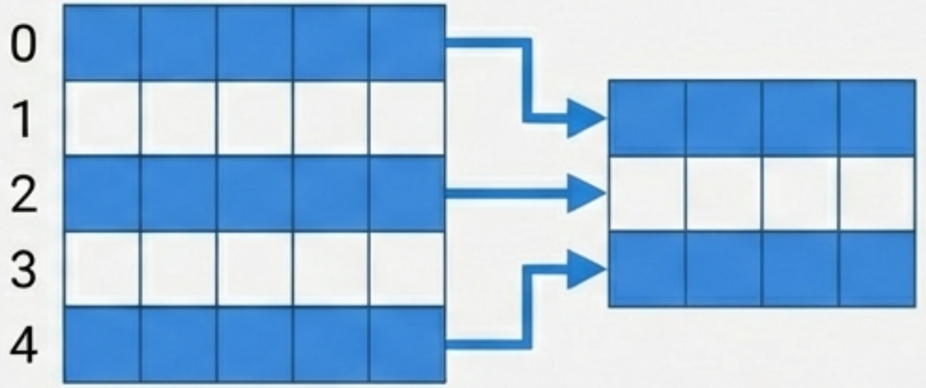
Boolean Indexing

```
data[names == "Bob"]
```

Lọc dữ liệu theo điều kiện logic.

Sử dụng `&` (và), `|` (hoặc), `~` (phủ định). Không dùng `and`/`or`.

Fancy Indexing




```
arr[[4, 3, 0]]
```

Dùng mảng số nguyên để chọn hàng/cột theo thứ tự cụ thể.

Sức mạnh Véc-tơ hóa & UFuncs

Roboto Regular: Thực hiện phép toán trên toàn bộ mảng. Quên đi vòng lặp 'for'.

Unary (1 mảng):

- `sqrt`
- `exp`
- `abs`
- `isnan`

Binary (2 mảng):

- `add`
- `maximum`
- `power`

Feature Spotlight: `np.where`

```
# Cú pháp: np.where(condition, x, y)
# Nếu arr > 0 trả về 2, ngược lại trả về -2

result = np.where(arr > 0, 2, -2)
```

→ Logic điều kiện cực nhanh cho dữ liệu lớn.

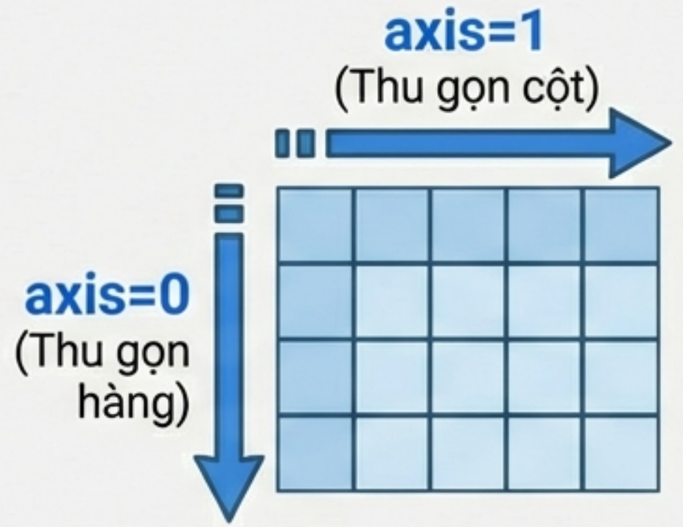
Thống kê & Tập hợp Dữ liệu

Roboto Regular: Các phương thức này thực hiện trên toàn bộ mảng hoặc dọc theo một trục (axis).

Key Methods Table

Method	Mô tả
`sum`	Tổng / Trung bình.
`mean`	Tổng / Trung bình.
`std`	Độ lệch chuẩn / Phương sai.
`var`	Độ lệch chuẩn / Phương sai.
`min`	Giá trị nhỏ nhất / lớn nhất.
`max`	Giá trị nhỏ nhất / lớn nhất.
`cumsum`	Tổng tích lũy.

Axis Concept



Set Operations

`np.unique(x)`	Trả về các giá trị duy nhất (đã sắp xếp).
`np.isin(x, y)`	Kiểm tra phần tử **x** có nằm trong **y** không.

Đại số Tuyến tính & Số Ngẫu nhiên

Linear Algebra

$$A \cdot B$$

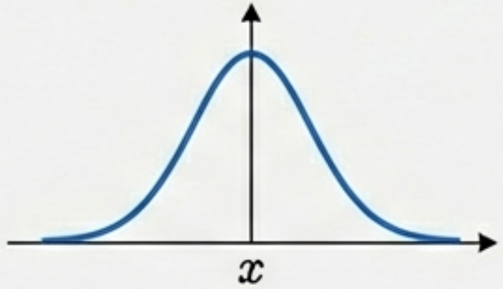
Nhân ma trận (Dot product):

`np.dot(x, y)` or `x @ y`

Other key functions on Fira code:

- `inv`: Nghịch đảo
- `qr`: Phân rã
- `T`: Chuyển vị

Random (numpy.random)



Sinh dữ liệu giả lập tốc độ cao.

`standard_normal()`: Phân phối chuẩn.

`permutation()`: Hoán vị ngẫu nhiên.

`default_rng(seed)`: Generator để tái lập kết quả.

Nhập môn Pandas: Khi dữ liệu cần “Nhãn”

Raw Data (NumPy)

```
[[1.2, 3.4, 5.6],
 [7.8, 9.0, 1.2],
 [3.4, 5.6, 7.8]]
```

Tiến hóa thành

Labeled Data (Pandas)

Index	Col1	Col2	Col3
A	1.2	3.4	5.6
B	7.8	9.0	1.2
C	3.4	5.6	7.8

Core Concepts

- **Pandas (Panel Data)**
  Tối ưu cho dữ liệu bảng (tabular) và hỗn hợp (heterogeneous).
- **Convention**
  `import pandas as pd`

	Value
X	1.2
Y	7.8
Z	3.4

Series (1D)

	C1	C2
R1	1.2	3.4
R2	7.8	9.0
R3	3.4	7.8

DataFrame (2D)

Series: Mảng Một Chiều có Định danh

Anatomy of a Series

Index	Values
a	4
b	7
c	-5

Key Feature: Alignment

Tự động căn chỉnh dữ liệu theo Index.

```
s1 = pd.Series([1], index=['a'])
s2 = pd.Series([2], index=['a'])
# s1 + s2 = 3 (tại index 'a')
```

Missing Data

Xử lý dữ liệu thiếu (NaN).

```
pd.isna(obj)
```

DataFrame: Bảng tính Mạnh mẽ

Create
Tạo

`pd.DataFrame(data)`
(From Dict of Lists)

Preview
Xem trước

`.head()`
(Xem 5 dòng đầu)

Index	state	year	pop
0	Ohio	2000	1.5
1	Nevada	2001	1.7
2	California	2002	3.6

Select Column
Chọn Cột

`frame["state"]` hoặc
`frame.state`

Modify
Chỉnh sửa

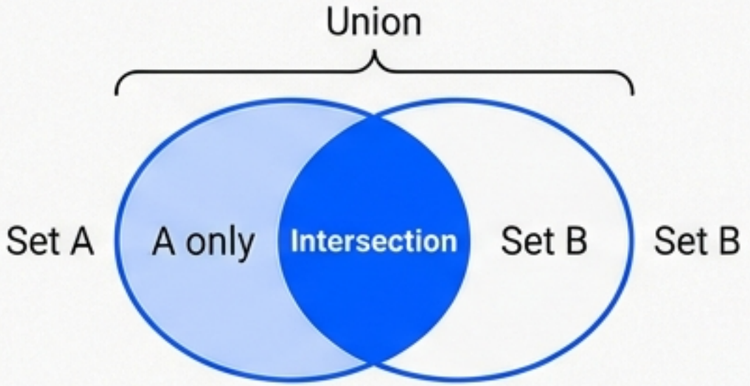
`frame["new_col"] = value`
Thêm cột
`del frame["col"]`
Xóa cột

```
data = {"state": ["Ohio", "Nevada"], "year": [2000, 2001]}
frame = pd.DataFrame(data)
```

Đối tượng Index & Tương tác Dữ liệu

Index Properties

Index giống như một tập hợp (Set) nhưng cố định (Immutable).



Immutable **Set-like** **Duplicate-allowed**

Selection with .loc

Truy xuất chính xác theo Nhãn (Label)

```
frame.loc["Ohio"]
```

Chọn dữ liệu hàng có nhãn là “Ohio”.

Dùng `.T` để chuyển vị (Transpose) hàng thành cột.

Tổng kết & Kiểm tra Kiến thức

NumPy	Pandas
Tính toán số học	Phân tích bảng
Mảng đa chiều	Xử lý dữ liệu thực tế
Dữ liệu đồng nhất	Dữ liệu hỗn hợp

1. **NumPy dùng làm gì?**
   Tính toán khoa học, mảng N-chiều.

2. **Pandas dùng làm gì?**
   Phân tích dữ liệu bảng.

3. **Matplotlib?**
   Trực quan hóa (Vẽ biểu đồ).

4. **Scikit-learn?**
   Học máy (Machine Learning).

Nắm vững NumPy là bước đệm bắt buộc để thành thạo Pandas.