Chapter 2
Data Type

Tất cả đều là Object

Mọi thứ trong Python đều là **đối tượng (object)**, có:

- **identity**: địa chỉ trong bộ nhớ (dùng `id()` để kiểm tra)
- **type**: kiểu dữ liệu cố định, xác định các phương thức, phép toán hỗ trợ (dùng `type()` để kiểm tra)
- **value**: nội dung hoặc trạng thái nội tại kể cả khi kiểu đó mutable hoặc immutable

Mutable vs Immutable: Một số đối tượng có thể thay đổi nội dung, một số khác thì không; ***nhưng bản chất vẫn là object***.

Mutable vs Immutable

Mutable: là kiểu dữ liệu có thể **thay đổi nội dung** sau khi đã được tạo.
Ví dụ: `list` , `dict` , `set`

Immutable: là kiểu dữ liệu **không thể thay đổi nội dung** sau khi đã tạo.
Ví dụ: `int` , `float` , `str` , `tuple`

Khi thay đổi:

Mutable → sửa đổi trên chỗ cũ (cùng địa chỉ bộ nhớ)
Immutable → tạo ra một bản sao mới (địa chỉ mới)

Các Built-in Types chính của Python

Numeric Types (số học) - immutable

`int` : số nguyên không giới hạn kích thước (chỉ hạn chế bởi bộ nhớ).

`float` : số dấu phẩy động (double precision ~64-bit IEEE-754).

`complex` : số phức, dạng `a + bj` , real & imag đều float.

Text Type - immutable

`str` : chuỗi Unicode, hỗ trợ ký tự đa ngôn ngữ

Sequence Types

`list` (mutable): danh sách tuần tự, có thể chứa nhiều kiểu khác nhau.

`tuple` (immutable): tương tự list nhưng không thể thay đổi, phù hợp làm key dict hoặc bảo toàn thứ tự.

`range` (immutable): dãy số, thường dùng trong vòng `for`.

Mapping Type

`dict` (mutable): ánh xạ key → value; keys phải là kiểu hashable (immutable), value có thể bất kỳ kiểu nào

Set Types

`set` (mutable): tập hợp không theo thứ tự, các phần tử duy nhất, kiểu hashable.

`frozenset` (immutable): phiên bản bất biến của set.

Binary Types

`bytes` (immutable): chuỗi byte, thường dùng cho dữ liệu nhị phân.

`bytearray` (mutable): phiên bản mutable của bytes.

`memoryview`: view không copy dữ liệu nhị phân để thao tác

Boolean & Special

`bool` (immutable): chỉ gồm hai giá trị `True` hoặc `False` (con là subclass của `int`, với `True` = 1, `False` = 0)

`NoneType`: kiểu của `None`, biểu thị "không có giá trị"

`NotImplementedType`: giá trị `NotImplemented`, dùng trong operator overloading khi không xử lý được operand

`EllipsisType`: biểu tượng `...`, dùng trong indexing (thường NumPy)

Bảng tổng hợp nhanh

Nhóm	Kiểu	Mutable?	Ghi chú
Numeric	`int`, `float`, `complex`	Không	Immutable
Textual	`str`	Không	Unicode
Sequence	`list`	Có	Mutable
	`tuple`, `range`	Không	Tuples immutable, range immutable
Mapping	`dict`	Có	Key must be hashable
Set	`set`, `frozenset`	set mutable, frozenset immutable	Không trùng phần tử
Binary	`bytes`, `bytearray`, `memoryview`	bytes immutable, bytearray mutable, memoryview view only	Dữ liệu nhị phân
Boolean/Special	`bool`, `NoneType`, `NotImplementedType`, `EllipsisType`	Không	Giá trị đặc biệt

Tại sao kiểu dữ liệu lại quan trọng

Cho phép Python **kiểm tra tính đúng hợp lệ**, ví dụ: `+` chỉ có nghĩa với numeric hoặc str, chứ không với list và dict cùng lúc.

Python dùng **dấu này** (`instance.__len__` , `__bool__` , `__hash__` ,...) để kiểm tra truthiness, hashability, khả năng is, v.v.

Dynamic typing: bạn không phải khai báo kiểu, Python sẽ **gán kiểu khi chạy** dựa vào giá trị gán vào biến — kiểu có thể thay đổi theo assignment sau đó

Ví dụ

```
x = 5              # tạo object int(5), immutable
y = [1, 2, 3]      # tạo list object, mutable

x = "hello"        # biến x giờ tham chiếu tới object str
y.append(4)        # thay đổi nội dung list y (mutable)

d = {'a': 1, 2: 'two'}  # tạo dict mapping int & str keys
```

Mỗi đối tượng như `5`, `"hello"`, `[1,2,3]` đều có type, identity, value.
Khi bạn gán lại `x = "hello"`, Python không đổi nội dung object trước, mà **đổi biến x tham chiếu** tới object mới.

Tóm lại

Python xử lý dữ liệu qua **đối tượng**, mỗi object có **identity, type, và value**.

Các kiểu dữ liệu built-in gồm numeric, sequence, mapping, set, binary, boolean và một số kiểu đặc biệt như None, NotImplemented, Ellipsis.

Có thể phân biệt rõ ràng giữa **mutable và immutable**, ảnh hưởng rất nhiều đến cách hoạt động của ngôn ngữ.

Python là **dynamic typed**, tức là không khai báo kiểu trước, kiểu gán vào biến theo giá trị tại runtime.

cheers

cảm ơn

thank you!

muchas gracias

dziękuję

danke