

DATA TYPE, I/O, OPERATOR

1 Data type

1.1 Data type là gì?

Ta có thể xem Data type như một loại đồ vật. Giống như việc "táo" là một loại trái cây, "xe" là phương tiện giao thông thì trong Python, mỗi dữ liệu đều thuộc một "loại" nào đó.

Python cần biết: Thứ này là số? Là chữ? Là hộp chứa đồ?

1.2 Kiểu số - Numbers

- **int - số nguyên:** ta thường dùng số nguyên để đếm. Ví dụ: số học sinh, số bước đi
- **float - số thập phân:** dùng để đo lường chiều cao, cân nặng,... Đối với Python, nếu ta khai báo là 2.0 thay vì 2, chương trình sẽ hiểu đó là loại float, không phải int.
- **complex - số phức:** chủ yếu cho toán cao cấp

1.3 Kiểu chữ - Text

- **str/string - chuỗi ký tự:** là chữ, đặt trong dấu nháy, kể cả khi có số bên trong dấu nháy đó vẫn là chữ.

1.4 Kiểu đúng sai

- **bool/boolean:** có hai giá trị là **True** và **False**. Dùng để trả lời cho câu hỏi "Có không?", "Đúng hay sai?"

1.5 Kiểu "hộp đựng đồ"

- **list - danh sách:** giống như hộp bút, có thể thay đổi, nhưng đặc biệt hộp bút này có thứ tự.

```
1 mylist = ["apple", "banana", "cherry"]
```

- **tuple - danh sách khóa:** cũng là một chiếc hộp nhưng bị niêm phong lại, có thứ tự và không thay đổi được thành phần bên trong.

```
1 mytuple = ("apple", "banana", "cherry")
```

- **set - tập hợp:** giống như túi bi, không quan tâm thứ tự nhưng cũng không sửa được.

```
1 myset = {"apple", "banana", "cherry"}
```

- **dict - từ điển:** một cuốn sổ tra cứu gồm hai phần: key và value.

```
1 {  
2     "name": "An",  
3     "age": 10  
4 }
```

2 I/O

2.1 Input & Output

I/O là một cách ta nói chuyện với máy tính, với output là phần cho máy nói ra màn hình còn input lại là phần máy tính nghe người dùng nói.

2.2 Type casting

Có nhiều cách ép kiểu trong Python

```
1 int("5")      # 5  
2 float("3.2")  # 3.2  
3 str(10)       # "10"
```

3 Operator

3.1 Toán học

Toán tử	Ý nghĩa	Ví dụ
+	Cộng	$3 + 2 = 5$
-	Trừ	$5 - 3 = 2$
*	Nhân	$4 * 2 = 8$
/	Chia	$5 / 2 = 2.5$
//	Chia lấy phần nguyên	$5 // 2 = 2$
%	Chia lấy phần dư	$5 \% 2 = 1$
**	Lũy thừa	$2 ** 3 = 8$

Table 1: Toán tử toán học trong Python

3.2 So sánh

Toán tử	Ý nghĩa	Ví dụ
==	Bằng nhau	$5 == 5 \rightarrow \text{True}$
!=	Khác nhau	$5 != 3 \rightarrow \text{True}$
>	Lớn hơn	$5 > 3 \rightarrow \text{True}$
<	Nhỏ hơn	$2 < 4 \rightarrow \text{True}$
>=	Lớn hơn hoặc bằng	$5 >= 5 \rightarrow \text{True}$
<=	Nhỏ hơn hoặc bằng	$3 <= 5 \rightarrow \text{True}$

Table 2: Toán tử so sánh trong Python

3.3 Logic

Toán tử	Ý nghĩa	Ví dụ
and	Và (cả hai đúng)	$(5 > 3) \text{ and } (2 > 1) \rightarrow \text{True}$
or	Hoặc (một trong hai đúng)	$(5 > 3) \text{ or } (2 < 1) \rightarrow \text{True}$
not	Phủ định	$\text{not } (5 > 3) \rightarrow \text{False}$

Table 3: Toán tử logic trong Python

3.4 Gán, Membership, So sánh đối tượng

Đối với phép gán trong Python cũng tương tự như C++

```
1 x = 5
2 x += 1    # x = x + 1
3 x *= 2
```

Thành viên (membership) là cách Python kiểm tra một đối tượng có nằm trong đối tượng khác hay không

```
1 "a" in "apple"    # True
```

Ngoài ra ta còn một cách so sánh nâng cao hơn, nếu "==" là so sánh xem có giống giá trị không thì "is" là để xem có cùng "đồ vật" trong bộ nhớ hay không

```
1 a is b
```