

Computer Science Basic

BÀI 3. KIỂU DỮ LIỆU BOOLEAN VÀ CẤU TRÚC ĐIỀU KIỆN

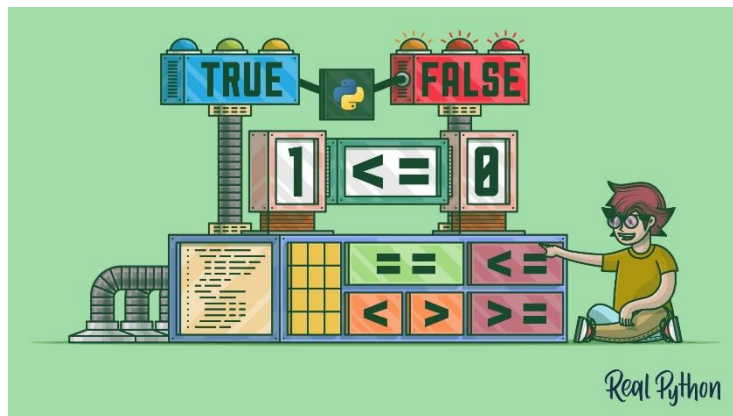
1. Kiểu Dữ Liệu Boolean

Trong Python, kiểu dữ liệu **boolean** chỉ bao gồm 2 giá trị: **True** và **False**.

Các giá trị **boolean** thường thể hiện tính **đúng** hoặc **sai** của một mệnh đề nào đó. Ví dụ:

- Phép so sánh $1 < 2$ có kết quả **True**
- Phép so sánh $1 > 2$ có kết quả **False**

Kiểu dữ liệu này được đặt tên theo nhà toán học và logic học người Anh, **George Boole**.



Ảnh: [Real Python](#)

PHÉP TOÁN SO SÁNH

Phép toán so sánh cho phép ta kiểm tra quan hệ trước sau giữa hai giá trị bất kỳ. Kết quả của một phép so sánh trong Python là một giá trị **boolean**.

- Giá trị này là **True** nếu biểu thức so sánh là **đúng**.
- Giá trị này là **False** nếu biểu thức so sánh là **sai**.

| Phép toán | Toán tử | Ví dụ |
|---------------------------|--------------------|----------------------------------|
| So sánh bằng | <code>==</code> | <code>1 == 2</code> (= False) |
| So sánh khác | <code>!=</code> | <code>1 != 2</code> (= True) |
| So sánh lớn hơn | <code>></code> | <code>1 > 2</code> (= False) |
| So sánh bé hơn | <code><</code> | <code>1 < 2</code> (= True) |
| So sánh lớn hơn hoặc bằng | <code>>=</code> | <code>1 >= 2</code> (= False) |
| So sánh bé hơn hoặc bằng | <code><=</code> | <code>1 <= 2</code> (= True) |

Đối với **string**, phép so sánh thực hiện trên từng ký tự, từ trái sang phải và xác định kết quả theo thứ tự bảng chữ cái.

Đối với **boolean**, Python định nghĩa `False < True`.

Ví dụ: `"abc" < "acb"` (= True)
`False >= True` (= False)

Chú ý: Tránh nhầm lẫn phép gán (`=`) và phép so sánh bằng (`==`).

PHÉP TOÁN LOGIC

Phép toán logic cho phép ta kết hợp các giá trị **boolean** trong một biểu thức logic. Kết quả của một phép toán logic là một giá trị **boolean**.

| Phép toán | Toán tử | Độ ưu tiên | Ví dụ | Diễn giải |
|-----------|------------------|------------|---------------------------------------|--|
| Phép và | <code>and</code> | 2 | <code>True and False</code> (= False) | Đúng khi toán hạng trái và phải đúng |
| Phép hoặc | <code>or</code> | 1 | <code>True or False</code> (= True) | Đúng khi toán hạng trái hoặc phải đúng |
| Phép đổi | <code>not</code> | 3 | <code>not True</code> (= False) | Đúng khi toán hạng sai |

Khi lập trình, ta thường dùng phép toán logic để kiểm tra:

- Có phải tất cả biểu thức đều đúng: phép **and**
- Có phải có ít nhất một biểu thức đúng: phép **or**
- Có phải biểu thức sai: phép **not**

Ví dụ: Biểu thức sau trả về **True** nếu biến **num** có giá trị từ 1 đến 10:

```
(num >= 1) and (num <= 10)
```

Chú ý: Mọi toán tử so sánh đều có độ ưu tiên **cao hơn** toán tử logic, do đó:

```
(num >= 1) and (num <= 10) ⇔ num >= 1 and num <= 10
```

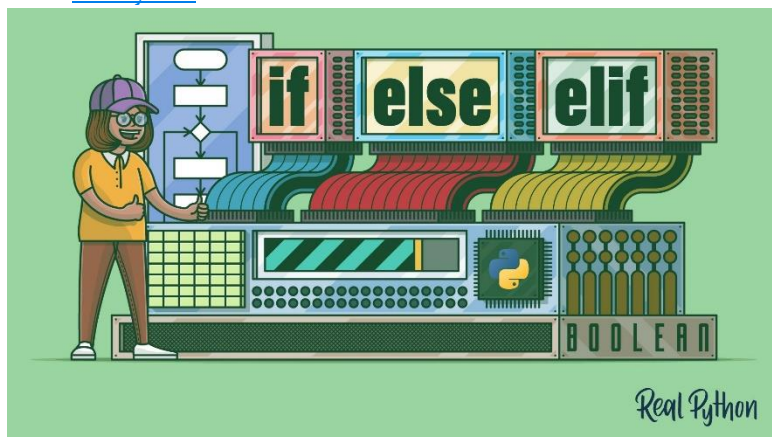
2. Cấu Trúc Điều Kiện

Cấu trúc điều kiện cho phép ta quyết định hành động nào được thực hiện, dựa trên các **điều kiện** cho trước.

Ví dụ: Người dùng được truy cập vào tài khoản với điều kiện là tên đăng nhập và mật khẩu được nhập đúng.

Trong Python, cấu trúc điều kiện được thể hiện bằng các từ khóa **if**, **else**, **elif**.

Ảnh: [Real Python](#)



CẤU TRÚC IF...

Cấu trúc if bao gồm hai thành phần: **điều kiện** và **hành động**.

- **Điều kiện** là một *biểu thức logic*, có kết quả là một giá trị *boolean*.
- **Hành động** là một đoạn code, bao gồm một hoặc nhiều câu lệnh.

Cấu trúc này mang ý nghĩa:

- Nếu điều kiện **đúng**, thực hiện hành động.
- Nếu điều kiện **sai**, bỏ qua hành động.

Cú pháp của cấu trúc if:

```
if <điều kiện>:
    <câu lệnh 1>
    <câu lệnh 2>
    ...
    <câu lệnh n>

<các câu lệnh tiếp theo>
```

Chú ý:

- Các câu lệnh trong cấu trúc **if** đều phải được **viết thật vào** và **ngang hàng nhau** để Python xác định câu lệnh nào nằm trong **if** và câu lệnh nào không.
- Phần thật vào thường là 2 khoảng trắng, 4 khoảng trắng hoặc 1 kí tự tab.

Ví dụ: Cho biến **num** chứa một giá trị số do người dùng nhập vào. Ta in ra thông báo lỗi nếu số có giá trị âm.

```
if num < 0:
    print('Invalid Input!')
```

CẤU TRÚC IF... ELSE...

Cấu trúc if else bao gồm:

- **Điều kiện**
- **Hành động nếu điều kiện đúng**
- **Hành động nếu điều kiện sai**

Cú pháp của cấu trúc if else:

```
if <điều kiện>:
    <các câu lệnh nếu điều kiện đúng>
else:
    <các câu lệnh nếu điều kiện sai>

<các câu lệnh tiếp theo>
```

Ví dụ: Cho biến **num** chứa một giá trị số do người dùng nhập vào. Ta kiểm tra **num** có phải số dương hay không và thông báo kết quả.

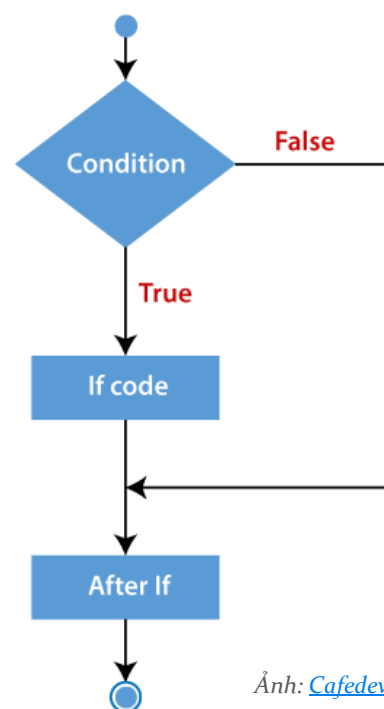
```
if num > 0:
    print(f'{num} is positive')
else:
    print(f'{num} is not positive')
```

Sử dụng:

```
>>> Input number: 10
>>> 10 is positive

>>> Input number: -2
>>> -2 is not positive
```

Luồng chạy của cấu trúc if



Ảnh: [Cafedev](#)

CẤU TRÚC IF... ELIF...

Cấu trúc if elif về bản chất là **cấu trúc if** được lồng trong **cấu trúc if else**.

Ví dụ: Với biến **num** người dùng nhập vào, ta kiểm tra **num** là số dương hay số âm.

| Cấu trúc if else lồng nhau | Cấu trúc if elif |
|--|--|
| <pre>if num > 0: print(f'{num} is positive') else: if num < 0: print(f'{num} is negative')</pre> | <pre>if num > 0: print(f'{num} is positive') elif num < 0: print(f'{num} is negative')</pre> |

Mọi cấu trúc **if elif** đều có thể được viết dưới dạng các cấu trúc **if else** lồng nhau. Tuy nhiên, từ khóa **elif** đơn giản hóa đoạn code cần viết, giúp người lập trình dễ đọc, dễ sửa lỗi.

Cùng ví dụ trên, ta có thể thêm từ khóa **else** vào cuối cấu trúc **if elif** để xử lý trường hợp **num** đúng bằng 0.

```
if num > 0:
    print(f'{num} is positive')
elif num < 0:
    print(f'{num} is negative')
else:
    print(f'{num} is zero')
```

3. Phụ Lục: Bảng Kết Quả Các Phép Toán Logic

| x | y | x and y | x or y | not x |
|----------|----------|----------------|---------------|--------------|
| False | False | False | False | True |
| False | True | False | True | True |
| True | False | False | True | False |
| True | True | True | True | False |