

CHƯƠNG 3: THỰC THI CÓ ĐIỀU KIỆN

3.1 Biểu thức Boolean (Boolean Expression)

- Luôn trả về giá trị kiểu bool: True hoặc False
- Toán tử quan hệ (Comparison Operators): `==`, `!=`, `>`, `<`, `>=`, `<=`
- True/False trong Python: True và False thuộc kiểu bool. Không phải chuỗi.

3.2 Toán tử logic (Logical Operators)

- `and`: True khi cả 2 đều True
- `or`: True khi ít nhất 1 đúng
- `not`: Đảo ngược giá trị Boolean

3.3 Câu lệnh rẽ nhánh đơn (if statement):

- Dùng để thực hiện lệnh khi điều kiện đúng.
- Cấu trúc:

```
if condition:
    statement
```
- Dòng điều kiện kết thúc bằng dấu :
- Lệnh trong thân phải thụt dòng (indent).

3.4 Rẽ nhánh hai chiều (if - else)

Có hai trường hợp, đúng chạy nhánh 1, sai chạy nhánh 2
if condition:

```
...
else:
    ...
```

3.8 Đánh giá ngắn mạch (Short-circuit Evaluation) & Mẫu bảo vệ (Guardian Pattern)

- Python sẽ ngừng đánh giá khi đã biết kết quả.
 - `x >= 2 and y != 0 and (x/y) > 2` Nếu `x < 2` → dừng luôn, không tính `(x/y)`
 - Tránh lỗi chia cho 0
- Đây gọi là guardian pattern (mẫu bảo vệ)

3.5 Rẽ nhánh đa chiều / Điều kiện chuỗi (chained conditional – if-elif-else)

- Chỉ một nhánh được thực thi, nhánh đầu tiên thỏa điều kiện.
- ```
if x < y:
 print('x is less than y')
elif x > y:
 print('x is greater than y')
else:
 print('x and y are equal')
```

## 3.6 Điều kiện lồng nhau (nested conditional):

- If nằm trong if khác.
- ```
if x == y:
    print("bằng nhau")
else:
    if x < y:
        print("x nhỏ hơn y")
    else:
        print("x lớn hơn y")
```

3.7 Xử lý ngoại lệ (Exception Handling) – try / except

- Dùng để chương trình không bị dừng khi xảy ra lỗi.
- ```
try:
 giá_trị = float(đầu_vào)
 kết_quả = 100 / giá_trị
except:
 print("Đầu vào không hợp lệ hoặc chia cho 0")
```

## 3.9 Debugging – Gỡ lỗi

Một vài lỗi hay gặp:

- `SyntaxError`: Sai cú pháp
- `IndentationError`: Sai thụt dòng
- `RuntimeError`: Lỗi khi chạy (vd: chia cho 0)
- Error message chỉ ra nơi phát hiện lỗi, không phải nơi gây lỗi