

CHƯƠNG 5: CẤU TRÚC VÒNG LẶP

1. CÁC LOẠI VÒNG LẶP

- Vòng lặp while (Lặp vô hạn/không xác định):

`n = 5`

`while n > 0:`

`print(n) n = n - 1 # Biến lặp thay đổi để thoát vòng lặp print('Blastoff!')`

- Vòng lặp for (Lặp xác định):

`for i in [5, 4, 3, 2, 1]:`

`print(i) # i lần lượt là 5, 4, 3, 2, 1`

2. ĐIỀU KHIỂN LƯỠNG TRONG VÒNG LẶP

- `break`: "Phá vỡ" vòng lặp ngay lập tức. Nhảy ra khỏi vòng lặp và chạy câu lệnh kế tiếp bên ngoài.
- Ví dụ: Dùng để thoát khi người dùng nhập 'done'.
- `continue`: "Bỏ qua" phần còn lại của lần lặp hiện tại. Nhảy về đầu vòng lặp để bắt đầu lần lặp mới.
- Ví dụ: Bỏ qua các dòng bắt đầu bằng dấu # (comment).

3. CÁC MẪU THUẬT TOÁN VÒNG LẶP (LOOP IDIOMS)

- Đếm (Counting): `count = 0` trước vòng lặp, `count = count + 1` bên trong.
- Tính tổng (Summing): `total = 0` trước vòng lặp, `total = total + value` bên trong.
- Tìm giá trị lớn nhất/nhỏ nhất (Max/Min):
- Dùng `None` để đánh dấu trạng thái "chưa có giá trị nào".

VĐ:

`smallest = None`
`for value in [9, 41, 12, 3]:`

`if smallest is None:`

`smallest = value # Gán giá trị đầu tiên`

`elif value < smallest:`

`smallest = value # Cập nhật nếu tìm thấy số nhỏ hơn`

`print('Smallest:', smallest)`

Tìm kiếm (Filtering): Dùng biến cờ (flag) boolean (`found = False`), bật lên `True` khi tìm thấy.

4. TOÁN TỬ IS VÀ IS NOT

- `is`: So sánh sự tương đồng tuyệt đối (cùng một đối tượng trong bộ nhớ). Mạnh hơn `==`.
- Thường dùng để kiểm tra: `if x is None` hoặc `if x is True`.
- Lưu ý: Hạn chế dùng `is` với số nguyên hoặc chuỗi, chỉ nên dùng với Booleans hoặc `None`.