

Chapter 1

Syntax

Thụt dòng (Indentation) — Không có `{}` hay `begin-end`

Python dùng thụt dòng (indentation) để xác định khối lệnh (**block**), không dùng dấu `{}` như C/Java.

```
if True:
    print("Yes")      # nằm trong if
print("Done")        # nằm ngoài if
```

Lỗi thường gặp: Sai khoảng trắng hoặc trộn tab/spaces.

Dấu hai chấm `:` báo hiệu bắt đầu khối lệnh

```
def say_hi():  
    print("Hi") # bắt buộc thụt vào vì có :
```

Áp dụng cho:

- `if`, `for`, `while`
- `def`, `class`
- `try`, `except`, `finally`
- `with`
- `match` (Python 3.10+)

Không cần khai báo kiểu biến — Dynamically Typed

```
x = 10      # x là int  
x = "ten"   # giờ x là string
```

Python tự suy ra kiểu biến tại thời điểm chạy.

Biến được gán bằng dấu `=`

```
x = 5  
y = x + 2
```

Không có var, let, int x = 5; như C-style.

Gọi hàm = viết tên + dấu **()**

```
print("Hello")  
result = add(2, 3)
```

Comment bắt đầu bằng

Đây là ghi chú

Không có `main()` bắt buộc, nhưng có thể dùng

```
if __name__ == "__main__":  
    main()
```

Dùng để phân biệt chạy trực tiếp hay import từ nơi khác.

Mỗi dòng là một câu lệnh (statement)

```
x = 5  
print(x)
```

Có thể viết nhiều lệnh trên 1 dòng bằng `;`, nhưng không nên dùng:

```
x = 1; y = 2
```

Danh sách các cú pháp điều khiển

Câu lệnh	Ý nghĩa
if , elif , else	điều kiện
for , while	vòng lặp
break , continue	điều khiển vòng lặp
try , except	xử lý lỗi
def , return	định nghĩa và trả về hàm
class , self	lớp và đối tượng
import , from	nhập module

Danh sách các cú pháp điều khiển

Ví dụ

```
def greet(name):           # định nghĩa hàm
    if name:
        print(f"Hello, {name}") # in chuỗi
    else:
        print("Hello, stranger")

greet("Vinh")              # gọi hàm
```

Cốt lõi Python syntax chỉ xoay quanh

- Thụt dòng để tạo khối
- `:` để mở đầu khối
- Gán `=`, gọi hàm `()`
- Không khai báo kiểu
- Không dấu ngoặc `{}`, không kết thúc bằng `end`

cheers

cảm ơn

thank you!

muchas gracias

dziękuję

danke