CHAPTER 4

FUNCTIONS

KHÁI NIỆM HÀM

Hàm là một khối mã có thể tái sử dụng, nhận input (arguments), xử lý và trả về output.

Ví dụ: def square(x): return x * x

ĐỊNH NGHĨA VÀ GỌI HÀM

Dùng def để tạo hàm và gọi hàm với đối số.

Ví dụ:

```
def greet(name): print("Hello", name)
greet('Alice')
```

ARGUMENTS& PARAMETERS

Parameter là biến trong định nghĩa hàm, Argument là giá trị được truyền khi gọi hàm.

Ví dụ:

```
def greet(lang): print("Hello in", lang)
greet('English')
```

HÀM CÓ TRẢ VỀ (FRUITFUL)

Hàm dùng return để trả về kết quả và kết thúc quá trình

Ví dụ:

```
def add(a, b): return a + b
result = add(2, 3)  # result = 5
```

HÀM KHÔNG TRẢ VỀ (NON-FRUITFUL)

Hàm không có return chỉ thực hiện hành động như in ra màn hình.

Ví dụ:

```
def say_hello(): print("Hello")
say_hello()  # In ra "Hello"
```

HÀM VỚI NHIỀU THAM SỐ

Hàm có thể nhận nhiều tham số để xử lý dữ liệu.

Ví dụ:

```
def multiply(a, b, c): return a * b * c
result = multiply(2, 3, 4)  # result = 24
```

CHUYỂN ĐỔI KIỂU DỮ LIỆU

Dùng int(), float(), str() để chuyển kiểu khi cần. Trộn int và float sẽ tự động chuyển đổi, nhưng chuyển string không hợp lệ sẽ gây lỗi.

Ví dụ:

```
x = float(99) / 100  # x = 0.99
```

TỔ CHỨC CHƯƠNG TRÌNH BẰNG HÀM

Chia nhỏ chương trình thành các hàm giúp tránh lặp code và dễ quản lý.

Ví dụ:

```
def calculate_area(radius): return 3.14 * radius ** 2
area = calculate_area(5)  # area = 78.5
```