CHƯƠNG 3: FUNCTIONS

Functions

- Function là khối mã có thể tái sử dụng.
- Gồm: tên hàm, tham số (parameters), thân hàm, và giá trị trả về (nếu có).
- Có 2 loại:
  - Built-in functions: print(), input(), type(), int(), float(), max(), min(), ...
  - User-defined functions: hàm do bạn tự viết bằng def.

Function Built-in

- max(x): trả về phần tử lớn nhất.
- min(x): trả về phần tử nhỏ nhất.
- type(x): trả về kiểu dữ liệu.
- int(), float(): ép kiểu số.
- int("123") → 123,
- float("9.5") → 9.5

def – call

- Cú pháp:

```
def function_name(parameters):
    statement(s)
```

- ví dụ:

```
def thing():
    print("Hello")
    print("Fun")
```

- Gọi hàm:

```
thing()
```

Type conversion

- int(x), float(x), str(x)
- Tự động chuyển integer → float khi có phép tính hỗn hợp.
- ví dụ:

```
print(float(99) / 100)  # 0.99
```

- nhiều tham số:

```
def addtwo(a, b):
    return a + b
```

Tham số & Đối số

- Argument: giá trị truyền vào khi gọi hàm.
- Parameter: biến “chứa” argument khi hàm chạy.
- ví dụ:

```
def greet(lang):   # lang là parameter
    if lang == 'es':
        print('Hola')
greet('es')        # 'es' là argument
```

Nguyên tắc viết hàm tốt

- Chia chương trình thành đoạn nhỏ có ý nghĩa.
- Không lặp lại code (“Don’t Repeat Yourself”).
- Dùng hàm để tổ chức và tái sử dụng logic.
- Biến hàm thành thư viện để dùng lâu dài.

Fruitful function

- Dùng **return** để trả kết quả về cho câu lệnh gọi hàm.
- ví dụ:

```
def addtwo(a, b):
    return a + b
x = addtwo(3, 5)   # x = 8
```

- **return** kết thúc hàm ngay lập tức.

Void (non-fruitful) Functions

- Không có return hoặc return không giá trị.
- Ví dụ: hàm chỉ dùng để in, ghi file, thực hiện hành động.

```
def print_lyrics():
    print("I'm a lumberjack")
```