

TUẦN 28 - THỨ 2 BUỔI SÁNG

FUNCTIONS CƠ BẢN - PARAMETERS - RETURN VALUES

Thời gian: 09:00 - 11:30 (2 tiếng 30 phút)

PHẦN 1: GIỚI THIỆU VỀ FUNCTIONS (09:00 - 09:30)

1.1 Function là gì?

- **Function (hàm)** là một khối code có thể tái sử dụng
- Giúp tổ chức code tốt hơn, tránh lặp lại code
- Giống như các function đã dùng: `print()`, `len()`, `input()`, `type()`
- Function như một "công cụ" để thực hiện một nhiệm vụ cụ thể

1.2 Tại sao cần Functions?

Ví dụ KHÔNG dùng function:

```
# Tính diện tích 3 hình chữ nhật
dai1, rong1 = 5, 3
dien_tich1 = dai1 * rong1
print(f"Diện tích 1: {dien_tich1}")

dai2, rong2 = 10, 4
dien_tich2 = dai2 * rong2
print(f"Diện tích 2: {dien_tich2}")

dai3, rong3 = 7, 6
dien_tich3 = dai3 * rong3
print(f"Diện tích 3: {dien_tich3}")
```

Dùng function:

```
def tinh_dien_tich(dai, rong):
    return dai * rong

print(f"Diện tích 1: {tinh_dien_tich(5, 3)}")
```

```
print(f"Diện tích 2: {tinh_dien_tich(10, 4)}")
print(f"Diện tích 3: {tinh_dien_tich(7, 6)}")
```

1.3 Cú pháp cơ bản:

```
def ten_function():
    """Docstring mô tả function (tùy chọn)"""
    # Code thực hiện
    pass
```

Quy tắc đặt tên function:

- Sử dụng chữ thường và dấu gạch dưới
- Tên có ý nghĩa, mô tả chức năng
- Ví dụ: `tinh_tong()`, `kiem_tra_chan_le()`, `in_thong_tin()`

```
In [1]: # TODO 1: Tạo function đầu tiên
def chao_hoi():
    print("Xin chào! Tôi là function đầu tiên")
    print("Rất vui được gặp bạn!")

# Gọi function
chao_hoi()
```

Xin chào! Tôi là function đầu tiên
Rất vui được gặp bạn!

```
In [2]: # TODO 2: Tạo function tính toán đơn giản
def tinh_tong_2_so():
    so_1 = 10
    so_2 = 20
    tong = so_1 + so_2
    print(f"Tổng của {so_1} và {so_2} là: {tong}")

# Gọi function
tinh_tong_2_so()
```

Tổng của 10 và 20 là: 30

PHẦN 2: PARAMETERS - THAM SỐ (09:30 - 10:15)

2.1 Parameter vs Argument

- Parameter:** Biến trong định nghĩa function
- Argument:** Giá trị thực tế truyền vào khi gọi function

```
def chao_hoi(ten): # 'ten' là parameter
    print(f"Xin chào {ten}")
```

```
chao_hoi("Minh") # "Minh" là argument
```

2.2 Các loại Parameters:

1) Positional Parameters (Tham số vị trí)

```
def gioi_thieu(ten, tuoi, que_quan):
    print(f"Tên: {ten}, Tuổi: {tuoi}, Quê: {que_quan}")
```

```
gioi_thieu("Nam", 20, "Hà Nội") # Thứ tự quan trọng
```

2) Keyword Parameters (Tham số từ khóa)

```
gioi_thieu(tuoi=25, ten="Hoa", que_quan="HCM") # Thứ tự không quan trọng
```

3) Default Parameters (Tham số mặc định)

```
def chao_hoi(ten, ngon_ngu="Tiếng Việt"):
    print(f"Xin chào {ten} bằng {ngon_ngu}")
```

```
chao_hoi("An") # Dùng giá trị mặc định
```

```
chao_hoi("Bob", "English") # Ghi đè giá trị mặc định
```

```
In [3]: # TODO 3: Function với 1 parameter
def chao_ten(ten):
    print(f"Xin chào {ten}!")
    print(f"Rất vui được gặp bạn, {ten}!")

# Gọi function với argument
chao_ten("Minh")
chao_ten("Hoa")
chao_ten("Python")
```

```
Xin chào Minh!
Rất vui được gặp bạn, Minh!
Xin chào Hoa!
Rất vui được gặp bạn, Hoa!
Xin chào Python!
Rất vui được gặp bạn, Python!
```

```
In [4]: # TODO 4: Function với nhiều parameters
def tinh_tong(so_1, so_2):
    tong = so_1 + so_2
    print(f"Tổng của {so_1} và {so_2} là: {tong}")

# Gọi function
tinh_tong(5, 3)
tinh_tong(10, 25)
tinh_tong(100, 200)
```

Tổng của 5 và 3 là: 8
 Tổng của 10 và 25 là: 35
 Tổng của 100 và 200 là: 300

```
In [5]: # TODO 5: Function tính diện tích hình chữ nhật
def tinh_dien_tich(chieu_dai, chieu_rong):
    dien_tich = chieu_dai * chieu_rong
    print(f"Diện tích hình chữ nhật {chieu_dai} x {chieu_rong} = {dien_tich}")

# Test function
tinh_dien_tich(5, 3)
tinh_dien_tich(10, 8)
tinh_dien_tich(2.5, 4.2)
```

Diện tích hình chữ nhật $5 \times 3 = 15$
 Diện tích hình chữ nhật $10 \times 8 = 80$
 Diện tích hình chữ nhật $2.5 \times 4.2 = 10.5$

PHẦN 3: RETURN VALUES - GIÁ TRỊ TRẢ VỀ (10:30 - 11:15)

3.1 Return là gì?

- **Return** trả về kết quả từ function
- Có thể lưu kết quả vào biến để sử dụng sau
- Function kết thúc ngay khi gặp `return`
- Nếu không có `return`, function trả về `None`

3.2 Các loại Return:

1) Return một giá trị:

```
def binh_phuong(x):
    return x * x
```

```
ket_qua = binh_phuong(5) # ket_qua = 25
```

2) Return nhiều giá trị (tuple):

```
def tinh_chu_vi_dien_tich(dai, rong):
    chu_vi = 2 * (dai + rong)
    dien_tich = dai * rong
    return chu_vi, dien_tich
```

```
cv, dt = tinh_chu_vi_dien_tich(5, 3)
```

3) Return có điều kiện:

```
def kiem_tra_diem(diem):
    if diem >= 8:
        return "Giỏi"
    elif diem >= 6.5:
        return "Khá"
    elif diem >= 5:
        return "Trung bình"
    else:
        return "Yếu"
```

4) Return sớm (Early Return):

```
def chia_so(a, b):
    if b == 0:
        return "Lỗi: Không thể chia cho 0"
    return a / b
```

```
In [6]: # TODO 6: Function trả về giá trị
def cong_2_so(a, b):
    ket_qua = a + b
    return ket_qua

# Sử dụng return value
tong = cong_2_so(5, 3)
print(f"Kết quả: {tong}")

# Có thể dùng trực tiếp
print(f"10 + 20 = {cong_2_so(10, 20)}")
```

Kết quả: 8
10 + 20 = 30

```
In [7]: # TODO 7: Function tính toán cơ bản
def nhan_2_so(a, b):
    return a * b

def chia_2_so(a, b):
    if b != 0:
        return a / b
    else:
        return "Không thể chia cho 0"

# Test các function
print(f"5 x 6 = {nhan_2_so(5, 6)}")
print(f"10 / 2 = {chia_2_so(10, 2)}")
print(f"10 / 0 = {chia_2_so(10, 0)}")
```

5 x 6 = 30
10 / 2 = 5.0
10 / 0 = Không thể chia cho 0

```
In [8]: # TODO 8: Function tính chu vi và diện tích
def tinh_chu_vi_hinh_vuong(canh):
    chu_vi = canh * 4
    return chu_vi
```

```
def tinh_dien_tich_hinh_vuong(canh):
    dien_tich = canh * canh
    return dien_tich

# Sử dụng functions
canh = 5
chu_vi = tinh_chu_vi_hinh_vuong(canh)
dien_tich = tinh_dien_tich_hinh_vuong(canh)

print(f"Hình vuông cạnh {canh}:")
print(f"Chu vi: {chu_vi}")
print(f"Diện tích: {dien_tich}")
```

Hình vuông cạnh 5:

Chu vi: 20

Diện tích: 25

4. FUNCTION EXAMPLES (VÍ DỤ THỰC TẾ)

Kết hợp kiến thức đã học: variables, operators, conditions, loops

```
In [9]: # TODO 9: Function kiểm tra số chẵn lẻ
def kiểm_tra_chan_le(so):
    if so % 2 == 0:
        return "chẵn"
    else:
        return "lẻ"

# Test function
for i in range(1, 6):
    ket_qua = kiểm_tra_chan_le(i)
    print(f"Số {i} là số {ket_qua}")
```

Số 1 là số lẻ

Số 2 là số chẵn

Số 3 là số lẻ

Số 4 là số chẵn

Số 5 là số lẻ

```
In [10]: # TODO 10: Function tính giai thừa
def tinh_giai_thua(n):
    if n < 0:
        return "Không thể tính giai thừa số âm"
    elif n == 0 or n == 1:
        return 1
    else:
        ket_qua = 1
        for i in range(2, n + 1):
            ket_qua = ket_qua * i
        return ket_qua

# Test function
print(f"5! = {tinh_giai_thua(5)}")
```

```
print(f"0! = {tinh_giai_thua(0)}")
print(f"(-3)! = {tinh_giai_thua(-3)}")
```

5! = 120

0! = 1

(-3)! = Không thể tính giai thừa số âm

```
In [11]: # TODO 11: Function tìm số lớn nhất trong list
def tim_so_lon_nhat(danh_sach):
    if len(danh_sach) == 0:
        return "Danh sách rỗng"

    so_lon_nhat = danh_sach[0]
    for so in danh_sach:
        if so > so_lon_nhat:
            so_lon_nhat = so

    return so_lon_nhat

# Test function
so_list = [3, 1, 4, 1, 5, 9, 2, 6]
print(f"Danh sách: {so_list}")
print(f"Số lớn nhất: {tim_so_lon_nhat(so_list)}")
```

Danh sách: [3, 1, 4, 1, 5, 9, 2, 6]

Số lớn nhất: 9

```
In [12]: # TODO 12: Function đếm ký tự trong string
def dem_ky_tu(chuoi, ky_tu):
    so_lan = 0
    for char in chuoi:
        if char == ky_tu:
            so_lan += 1
    return so_lan

# Test function
van_ban = "Hello Python Programming"
print(f"Văn bản: {van_ban}")
print(f"Số lần xuất hiện chữ 'o': {dem_ky_tu(van_ban, 'o')}")
print(f"Số lần xuất hiện chữ 'P': {dem_ky_tu(van_ban, 'P')}")
```

Văn bản: Hello Python Programming

Số lần xuất hiện chữ 'o': 3

Số lần xuất hiện chữ 'P': 2

```
In [13]: # TODO 13: Function với default parameters
def gioi_thieu_hoc_sinh(ten, tuoi, lop="12A1", truong="THPT ABC"):
    print(f"Học sinh: {ten}")
    print(f"Tuổi: {tuoi}")
    print(f"Lớp: {lop}")
    print(f"Trường: {truong}")
    print("-" * 30)

# Test với các cách gọi khác nhau
gioi_thieu_hoc_sinh("Minh", 17)
gioi_thieu_hoc_sinh("Hoa", 16, "11B2")
gioi_thieu_hoc_sinh("Nam", 18, "12C3", "THPT XYZ")
```

Học sinh: Minh
 Tuổi: 17
 Lớp: 12A1
 Trường: THPT ABC

Học sinh: Hoa
 Tuổi: 16
 Lớp: 11B2
 Trường: THPT ABC

Học sinh: Nam
 Tuổi: 18
 Lớp: 12C3
 Trường: THPT XYZ

```
In [14]: # TODO 14: Function return nhiều giá trị
def phan_tich_so(n):
    """
    Phân tích một số: chẵn/lẻ, dương/âm/0, có phải số nguyên tố không
    """
    # Kiểm tra chẵn lẻ
    chan_le = "chẵn" if n % 2 == 0 else "lẻ"

    # Kiểm tra dương âm
    if n > 0:
        duong_am = "dương"
    elif n < 0:
        duong_am = "âm"
    else:
        duong_am = "bằng 0"

    # Kiểm tra số nguyên tố (đơn giản)
    la_nguyen_to = False
    if n > 1:
        la_nguyen_to = True
        for i in range(2, int(n**0.5) + 1):
            if n % i == 0:
                la_nguyen_to = False
                break

    return chan_le, duong_am, la_nguyen_to

# Test function
test_numbers = [7, 12, -5, 0, 2]
for num in test_numbers:
    chan_le, duong_am, nguyen_to = phan_tich_so(num)
    print(f"Số {num}: {chan_le}, {duong_am}, {'là' if nguyen_to else 'không là'} số
```

Số 7: lẻ, dương, là số nguyên tố
 Số 12: chẵn, dương, không là số nguyên tố
 Số -5: lẻ, âm, không là số nguyên tố
 Số 0: chẵn, bằng 0, không là số nguyên tố
 Số 2: chẵn, dương, là số nguyên tố

PHẦN 4: BÀI TẬP THỰC HÀNH (11:15 - 11:30)

Mục tiêu: Áp dụng tất cả kiến thức đã học về Functions

```
In [15]: # BÀI TẬP 1: Tạo function tính điểm trung bình (CƠ BẢN)
def tinh_diem_trung_binh(diem_list):
    """
    Tính điểm trung bình từ danh sách điểm
    Input: list các điểm số
    Output: điểm trung bình (float)
    """
    if len(diem_list) == 0:
        return 0

    tong_diem = sum(diem_list)
    so_mon = len(diem_list)
    diem_tb = tong_diem / so_mon
    return round(diem_tb, 2)

# Test function
diem_hoc_sinh = [8, 7, 9, 6, 8]
print(f"Điểm các môn: {diem_hoc_sinh}")
print(f"Điểm trung bình: {tinh_diem_trung_binh(diem_hoc_sinh)}")

# Test thêm
print(f"Điểm TB của [10, 9, 8]: {tinh_diem_trung_binh([10, 9, 8])}")
print(f"Điểm TB của []: {tinh_diem_trung_binh([])}")
```

Điểm các môn: [8, 7, 9, 6, 8]

Điểm trung bình: 7.6

Điểm TB của [10, 9, 8]: 9.0

Điểm TB của []: 0

```
In [16]: # BÀI TẬP 2: Tạo function kiểm tra năm nhuận (TRUNG BÌNH)
def kiem_tra_nam_nhuan(nam):
    """
    Kiểm tra năm nhuận theo quy tắc:
    - Chia hết cho 4 VÀ
    - (Không chia hết cho 100 HOẶC chia hết cho 400)
    """
    if nam % 4 != 0:
        return False
    elif nam % 100 != 0:
        return True
    elif nam % 400 == 0:
        return True
    else:
        return False

# Test function
test_nam = [2020, 2021, 2024, 1900, 2000, 1600, 1700]
for nam in test_nam:
    ket_qua = kiem_tra_nam_nhuan(nam)
    print(f"Năm {nam}: {'Là năm nhuận' if ket_qua else 'Không là năm nhuận'}")

# Giải thích thêm
```

```
print("\n--- Giải thích ---")
print("2020: 4✓, 100X → Nhuận")
print("1900: 4✓, 100✓, 400X → Không nhuận")
print("2000: 4✓, 100✓, 400✓ → Nhuận")
```

Năm 2020: Là năm nhuận
 Năm 2021: Không là năm nhuận
 Năm 2024: Là năm nhuận
 Năm 1900: Không là năm nhuận
 Năm 2000: Là năm nhuận
 Năm 1600: Là năm nhuận
 Năm 1700: Không là năm nhuận

--- Giải thích ---
 2020: 4✓, 100X → Nhuận
 1900: 4✓, 100✓, 400X → Không nhuận
 2000: 4✓, 100✓, 400✓ → Nhuận

In [17]: *# BÀI TẬP 3: Tạo function tính tiền điện (NÂNG CAO)*

```
def tinh_tien_dien(so_kwh):
    """
    Tính tiền điện theo bậc thang:
    - Bậc 1 (0-50 kwh): 1,500 VND/kwh
    - Bậc 2 (51-100 kwh): 2,000 VND/kwh
    - Bậc 3 (101-200 kwh): 2,500 VND/kwh
    - Bậc 4 (>200 kwh): 3,000 VND/kwh
    """
    if so_kwh <= 0:
        return 0

    tong_tien = 0

    # Bậc 1: 0-50 kwh
    if so_kwh <= 50:
        tong_tien += so_kwh * 1500
    else:
        tong_tien += 50 * 1500
        so_kwh_con_lai = so_kwh - 50

    # Bậc 2: 51-100 kwh
    if so_kwh_con_lai <= 50:
        tong_tien += so_kwh_con_lai * 2000
    else:
        tong_tien += 50 * 2000
        so_kwh_con_lai -= 50

    # Bậc 3: 101-200 kwh
    if so_kwh_con_lai <= 100:
        tong_tien += so_kwh_con_lai * 2500
    else:
        tong_tien += 100 * 2500
        so_kwh_con_lai -= 100

    # Bậc 4: >200 kwh
    tong_tien += so_kwh_con_lai * 3000
```

```

    return int(tong_tien)

# Test function với giải thích chi tiết
test_kwh = [30, 80, 150, 250]
for kwh in test_kwh:
    tien = tinh_tien_dien(kwh)
    print(f"Sử dụng {kwh:3d} kwh → Tiền điện: {tien:,.} VND")

# Ví dụ chi tiết cho 150 kwh
print("\n--- Chi tiết tính 150 kwh ---")
print("Bậc 1: 50 kwh × 1,500 = 75,000 VND")
print("Bậc 2: 50 kwh × 2,000 = 100,000 VND")
print("Bậc 3: 50 kwh × 2,500 = 125,000 VND")
print("Tổng cộng: 300,000 VND")

```

Sử dụng 30 kwh → Tiền điện: 45,000 VND
 Sử dụng 80 kwh → Tiền điện: 135,000 VND
 Sử dụng 150 kwh → Tiền điện: 300,000 VND
 Sử dụng 250 kwh → Tiền điện: 575,000 VND

--- Chi tiết tính 150 kwh ---
 Bậc 1: 50 kwh × 1,500 = 75,000 VND
 Bậc 2: 50 kwh × 2,000 = 100,000 VND
 Bậc 3: 50 kwh × 2,500 = 125,000 VND
 Tổng cộng: 300,000 VND

In [18]: # BÀI TẬP 4: Tạo function tìm ước chung lớn nhất - UCLN (NÂNG CAO)

```

def tim_ucln(a, b):
    """
    Tìm ước chung lớn nhất bằng thuật toán Euclid
    Nguyên lý: UCLN(a,b) = UCLN(b, a%b) cho đến khi b = 0
    """
    # Đảm bảo a, b là số dương
    a, b = abs(a), abs(b)

    print(f"Tìm UCLN({a}, {b}):")

    while b != 0:
        du = a % b
        print(f" {a} = {b} × {a//b} + {du}")
        a, b = b, du

    print(f" Kết quả: UCLN = {a}")
    return a

# Test function
print("=" * 30)
tim_ucln(48, 18)
print("=" * 30)
tim_ucln(100, 25)
print("=" * 30)
tim_ucln(17, 13)

```

```

=====
Tìm UCLN(48, 18):
    48 = 18 × 2 + 12
    18 = 12 × 1 + 6
    12 = 6 × 2 + 0
    Kết quả: UCLN = 6
=====
Tìm UCLN(100, 25):
    100 = 25 × 4 + 0
    Kết quả: UCLN = 25
=====
Tìm UCLN(17, 13):
    17 = 13 × 1 + 4
    13 = 4 × 3 + 1
    4 = 1 × 4 + 0
    Kết quả: UCLN = 1

```

Out[18]: 1

```

In [19]: # BÀI TẬP 5: Function tính Lương nhân viên (TỔNG HỢP)
def tinh_luong_nhan_vien(luong_co_ban, so_gio_lam, gio_tang_ca=0, he_so_tang_ca=1.5)
    """
    Tính lương nhân viên:
    - Lương cơ bản theo giờ
    - Tăng ca (nếu có) với hệ số nhân
    - Thuế thu nhập (10% nếu > 5 triệu)
    """

    # Tính Lương chính
    luong_chinh = luong_co_ban * so_gio_lam

    # Tính Lương tăng ca
    luong_tang_ca = luong_co_ban * gio_tang_ca * he_so_tang_ca

    # Tổng Lương trước thuế
    tong_luong = luong_chinh + luong_tang_ca

    # Tính thuế (10% nếu > 5 triệu)
    thue = 0
    if tong_luong > 5000000:
        thue = tong_luong * 0.1

    # Lương thực nhận
    luong_thuc_nhan = tong_luong - thue

    return {
        'luong_chinh': luong_chinh,
        'luong_tang_ca': luong_tang_ca,
        'tong_luong': tong_luong,
        'thue': thue,
        'luong_thuc_nhan': luong_thuc_nhan
    }

# Test function
nv1 = tinh_luong_nhan_vien(50000, 160, 20) # 50k/giờ, 160h thường, 20h tăng ca
nv2 = tinh_luong_nhan_vien(30000, 160, 0)  # 30k/giờ, chỉ giờ thường

```

```
print("NHÂN VIÊN 1:")
for key, value in nv1.items():
    print(f" {key.replace('_', ' ').title()}: {value:,.0f} VND")

print("\nNHÂN VIÊN 2:")
for key, value in nv2.items():
    print(f" {key.replace('_', ' ').title()}: {value:,.0f} VND")
```

NHÂN VIÊN 1:

Luong Chinh: 8,000,000 VND
Luong Tang Ca: 1,500,000 VND
Tong Luong: 9,500,000 VND
Thue: 950,000 VND
Luong Thuc Nhan: 8,550,000 VND

NHÂN VIÊN 2:

Luong Chinh: 4,800,000 VND
Luong Tang Ca: 0 VND
Tong Luong: 4,800,000 VND
Thue: 0 VND
Luong Thuc Nhan: 4,800,000 VND

TỔNG KẾT BÀI HỌC (11:15 - 11:30)



NHỮNG ĐIỀU ĐÃ HỌC HÔM NAY:

1. Giới thiệu Functions (09:00-09:30)

- Hiểu được function là gì và tại sao cần dùng
- Cú pháp cơ bản: `def ten_function():`
- Quy tắc đặt tên function
- So sánh code có và không có function






2. Parameters - Tham số (09:30-10:15)

- Phân biệt Parameter vs Argument
- **Positional Parameters:** Thứ tự quan trọng
- **Keyword Parameters:** Gọi theo tên
- **Default Parameters:** Giá trị mặc định

3. Return Values (10:30-11:15)

- Return một giá trị
- Return nhiều giá trị (tuple)
- Return có điều kiện
- Early return pattern

4. Bài tập thực hành đã làm:

-  Tính điểm trung bình
-  Kiểm tra năm nhuận
-  Tính tiền điện theo bậc thang
-  Tìm UCLN bằng thuật toán Euclid
-  Tính lương nhân viên (tổng hợp)

CÁC NGUYÊN TẮC QUAN TRỌNG:

1. Single Responsibility Principle

- Một function chỉ làm một việc cụ thể
- Dễ hiểu, dễ test, dễ bảo trì

2. Đặt tên có ý nghĩa

- `tinh_diem_trung_binh()` thay vì `calculate()`
- `kiem_tra_nam_nhuan()` thay vì `check()`

3. Sử dụng Docstring

```
def my_function(param):  
    """Mô tả ngắn gọn function làm gì"""  
    return result
```

4. Error Handling

- Kiểm tra input hợp lệ
- Xử lý trường hợp đặc biệt (chia cho 0, list rỗng...)

BÀI TẬP VỀ NHÀ:

1. **Viết function tính BMI** và phân loại sức khỏe
2. **Function chuyển đổi nhiệt độ** ($C \leftrightarrow F \leftrightarrow K$)
3. **Function tính số ngày trong tháng** (xét năm nhuận)
4. **Function mã hóa Caesar cipher** đơn giản

BUỔI CHIỀU SẼ HỌC:

- **Scope & Lifetime** của biến trong function
- **Lambda functions** (anonymous functions)
- **Built-in functions** nâng cao
- **Math library** và các module hữu ích

 **Chúc mừng! Bạn đã hoàn thành phần Functions cơ bản!**