

# Bài 28.2: Viết Functions Thực Tế, Math Library & Utility Functions

## Mục tiêu bài học

- Viết các functions thực tế để giải quyết vấn đề cụ thể
- Sử dụng thư viện Math của Python
- Tạo các utility functions hữu ích
- Áp dụng kiến thức đã học (variables, data types, operators, loops, conditions, lists, dictionaries)

## Kiến thức cần có

- Functions cơ bản (đã học buổi sáng)
- Variables và data types
- Operators và conditions
- Loops (for, while)
- Lists và dictionaries

## 1. Ôn tập Functions cơ bản

Trước khi viết functions phức tạp, hãy ôn lại cấu trúc function cơ bản:

```
In [5]: # Cấu trúc function cơ bản
def ten_function(tham_so):
    # Xử lý Logic
    return ket_qua

# Ví dụ đơn giản
def chao_hoi(ten):
    return f"Xin chào {ten}!"

# Gọi function
ket_qua = chao_hoi("Minh")
print(ket_qua)
```

Xin chào Minh!



### TODO 1: Ôn tập function cơ bản

Viết một function `tinh_tong(a, b)` để tính tổng hai số và test function này:

```
In [6]: # TODO 1: Viết function tinh_tong
def tinh_tong(a, b):
```

```
# Viết code ở đây
pass

# Test function
print(tinh_tong(5, 3)) # Kết quả mong đợi: 8
print(tinh_tong(10, -2)) # Kết quả mong đợi: 8
```

None

None

## 2. Thư viện Math trong Python

Python có thư viện `math` tích hợp sẵn với nhiều functions toán học hữu ích:

```
In [7]: import math

# Các functions math phổ biến
print("Căn bậc hai của 16:", math.sqrt(16)) # 4.0
print("2 mũ 3:", math.pow(2, 3)) # 8.0
print("Làm tròn lên:", math.ceil(4.3)) # 5
print("Làm tròn xuống:", math.floor(4.7)) # 4
print("Giá trị tuyệt đối:", math.fabs(-5)) # 5.0
print("Pi:", math.pi) # 3.141592653589793
print("e:", math.e) # 2.718281828459045
```

Căn bậc hai của 16: 4.0

2 mũ 3: 8.0

Làm tròn lên: 5

Làm tròn xuống: 4

Giá trị tuyệt đối: 5.0

Pi: 3.141592653589793

e: 2.718281828459045



### TODO 2: Thực hành với Math library

Viết function `tinh_hinh_tron(ban_kinh)` để tính diện tích và chu vi hình tròn:

```
In [8]: import math

# TODO 2: Viết function tính diện tích và chu vi hình tròn
def tinh_hinh_tron(ban_kinh):
    # Công thức:
    # Diện tích = pi * r^2
    # Chu vi = 2 * pi * r

    # Viết code ở đây
    pass

# Test function
dien_tich, chu_vi = tinh_hinh_tron(5)
print(f"Bán kính 5: Diện tích = {dien_tich:.2f}, Chu vi = {chu_vi:.2f}")
```

```

-----
TypeError                                Traceback (most recent call last)
Cell In[8], line 13
     10     pass
     12 # Test function
--> 13 dien_tich, chu_vi = tinh_hinh_tron(5)
     14 print(f"Bán kính 5: Diện tích = {dien_tich:.2f}, Chu vi = {chu_vi:.2f}")

TypeError: cannot unpack non-iterable NoneType object

```

### 3. Functions thực tế - Xử lý dữ liệu

Viết functions để xử lý dữ liệu thực tế như tính toán thống kê cơ bản:

```

In [ ]: # Function tính trung bình của một list
def tinh_trung_binh(danh_sach):
    if len(danh_sach) == 0:
        return 0
    return sum(danh_sach) / len(danh_sach)

# Function tìm giá trị lớn nhất và nhỏ nhất
def tim_min_max(danh_sach):
    if len(danh_sach) == 0:
        return None, None
    return min(danh_sach), max(danh_sach)

# Test functions
diem_so = [8.5, 7.2, 9.1, 6.8, 8.9, 7.5]
print(f"Điểm trung bình: {tinh_trung_binh(diem_so):.2f}")
min_val, max_val = tim_min_max(diem_so)
print(f"Điểm thấp nhất: {min_val}, Điểm cao nhất: {max_val}")

```

#### TODO 3: Viết function thống kê

Viết function `thong_ke_diem(danh_sach_diem)` trả về dictionary chứa thống kê đầy đủ:

```

In [ ]: # TODO 3: Viết function thống kê điểm
def thong_ke_diem(danh_sach_diem):
    # Trả về dictionary với các key:
    # - 'trung_binh': điểm trung bình
    # - 'cao_nhat': điểm cao nhất
    # - 'thap_nhat': điểm thấp nhất
    # - 'so_luong': số lượng điểm
    # - 'tong_diem': tổng điểm

    # Viết code ở đây
    pass

# Test function
diem_test = [8.5, 7.2, 9.1, 6.8, 8.9, 7.5, 9.5]
ket_qua = thong_ke_diem(diem_test)
print("Thống kê điểm:")

```

```
for key, value in ket_qua.items():
    print(f" {key}: {value}")
```

## 4. Utility Functions - Xử lý chuỗi

Viết các functions tiện ích để xử lý chuỗi:

```
In [ ]: # Function đếm từ trong chuỗi
def dem_tu(chuoi):
    return len(chuoi.split())

# Function viết hoa chữ cái đầu mỗi từ
def viet_hoa_chu_dau(chuoi):
    return chuoi.title()

# Function xóa khoảng trắng thừa
def xoa_khoang_trang_thua(chuoi):
    return ' '.join(chuoi.split())

# Test functions
text = " xin chào các bạn "
print(f"Chuỗi gốc: '{text}'")
print(f"Số từ: {dem_tu(text)}")
print(f"Viết hoa chữ đầu: '{viet_hoa_chu_dau(text)}'")
print(f"Xóa khoảng trắng thừa: '{xoa_khoang_trang_thua(text)}'")
```

### TODO 4: Viết function xử lý chuỗi

Viết function `xu_ly_ho_ten(ho_ten)` để chuẩn hóa họ tên:

```
In [ ]: # TODO 4: Viết function chuẩn hóa họ tên
def xu_ly_ho_ten(ho_ten):
    # Yêu cầu:
    # 1. Xóa khoảng trắng thừa ở đầu và cuối
    # 2. Xóa khoảng trắng thừa giữa các từ
    # 3. Viết hoa chữ cái đầu mỗi từ
    # 4. Trả về chuỗi đã chuẩn hóa

    # Viết code ở đây
    pass

# Test function
test_names = [
    " nguyễn văn a ",
    "trần thị b",
    " lê văn c "
]

for name in test_names:
    print(f"Gốc: '{name}' -> Chuẩn hóa: '{xu_ly_ho_ten(name)}'")
```

## 5. Functions thực tế - Validation (Kiểm tra dữ liệu)

Viết functions để kiểm tra tính hợp lệ của dữ liệu:

```
In [ ]: # Function kiểm tra email đơn giản
def kiem_tra_email(email):
    return '@' in email and '.' in email

# Function kiểm tra số điện thoại
def kiem_tra_so_dien_thoai(sdt):
    # Xóa khoảng trắng và dấu gạch nối
    sdt = sdt.replace(' ', '').replace('-', '')
    # Kiểm tra độ dài và chỉ chứa số
    return len(sdt) >= 10 and sdt.isdigit()

# Function kiểm tra tuổi hợp lệ
def kiem_tra_tuoi(tuoi):
    return isinstance(tuoi, int) and 0 <= tuoi <= 120

# Test functions
print("Kiểm tra email:")
print(f" 'test@gmail.com': {kiem_tra_email('test@gmail.com')}")
print(f" 'invalid-email': {kiem_tra_email('invalid-email')}")

print("\nKiểm tra SĐT:")
print(f" '0123456789': {kiem_tra_so_dien_thoai('0123456789')}")
print(f" '0123-456-789': {kiem_tra_so_dien_thoai('0123-456-789')}")

print("\nKiểm tra tuổi:")
print(f" 25: {kiem_tra_tuoi(25)}")
print(f" -5: {kiem_tra_tuoi(-5)}")
```



### TODO 5: Viết function validation

Viết function `kiem_tra_thong_tin_sinh_vien(thong_tin)` để kiểm tra thông tin sinh viên:

```
In [ ]: # TODO 5: Viết function kiểm tra thông tin sinh viên
def kiem_tra_thong_tin_sinh_vien(thong_tin):
    # thong_tin là dictionary với keys: 'ho_ten', 'tuoi', 'email', 'sdt'
    # Trả về dictionary với kết quả kiểm tra:
    # {
    #     'hop_le': True/False,
    #     'loi': [] # danh sách lỗi nếu có
    # }

    # Quy tắc kiểm tra:
    # - Họ tên không được rỗng
    # - Tuổi từ 16-40
    # - Email hợp lệ
    # - SĐT hợp lệ
```

```

# Viết code ở đây
pass

# Test function
sinh_vien_1 = {
    'ho_ten': 'Nguyễn Văn A',
    'tuoi': 20,
    'email': 'nguyenvana@gmail.com',
    'sdt': '0123456789'
}

sinh_vien_2 = {
    'ho_ten': '',
    'tuoi': 15,
    'email': 'invalid-email',
    'sdt': '123'
}

print("Kiểm tra sinh viên 1:")
ket_qua_1 = kiem_tra_thong_tin_sinh_vien(sinh_vien_1)
print(f"    Hợp lệ: {ket_qua_1['hop_le']}")
print(f"    Lỗi: {ket_qua_1['loi']}")

print("\nKiểm tra sinh viên 2:")
ket_qua_2 = kiem_tra_thong_tin_sinh_vien(sinh_vien_2)
print(f"    Hợp lệ: {ket_qua_2['hop_le']}")
print(f"    Lỗi: {ket_qua_2['loi']}")

```

## 6. Utility Functions - Làm việc với Lists

Viết các functions tiện ích để xử lý lists:

```

In [ ]: # Function lọc số chẵn
def loc_so_chan(danh_sach):
    return [num for num in danh_sach if num % 2 == 0]

# Function xóa phần tử trùng lặp
def xoa_trung_lap(danh_sach):
    return list(set(danh_sach))

# Function tìm phần tử xuất hiện nhiều nhất
def phan_tu_xuat_hien_nhieu_nhat(danh_sach):
    if not danh_sach:
        return None

    dem = {}
    for item in danh_sach:
        dem[item] = dem.get(item, 0) + 1

    return max(dem, key=dem.get)

# Test functions
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
mixed_list = [1, 2, 2, 3, 3, 3, 4, 5]

```

```
print(f"Số chẵn: {loc_so_chan(numbers)}")
print(f"Xóa trùng lặp: {xoa_trung_lap(mixed_list)}")
print(f"Xuất hiện nhiều nhất: {phan_tu_xuat_hien_nhieu_nhat(mixed_list)}")
```

## TODO 6: Viết function xử lý danh sách

Viết function `phan_tich_danh_sach(danh_sach)` để phân tích danh sách số:

```
In [ ]: # TODO 6: Viết function phân tích danh sách
def phan_tich_danh_sach(danh_sach):
    # Trả về dictionary với thông tin:
    # - 'so_chan': danh sách số chẵn
    # - 'so_le': danh sách số lẻ
    # - 'so_duong': danh sách số dương
    # - 'so_am': danh sách số âm
    # - 'tong': tổng các số
    # - 'trung_binh': trung bình cộng

    # Viết code ở đây
    pass

# Test function
test_list = [-5, -2, 0, 1, 2, 3, 4, 6, 8, 9]
ket_qua = phan_tich_danh_sach(test_list)

print("Phân tích danh sách:")
for key, value in ket_qua.items():
    print(f" {key}: {value}")
```

## 7. Functions thực tế - Tính toán tài chính

Viết functions để tính toán tài chính cơ bản:

```
In [ ]: # Function tính lãi suất đơn giản
def lai_suat_don_gian(tien_goc, lai_suat, thoi_gian):
    lai = tien_goc * lai_suat * thoi_gian / 100
    return tien_goc + lai

# Function tính lãi suất kép
def lai_suat_kep(tien_goc, lai_suat, thoi_gian):
    return tien_goc * (1 + lai_suat/100) ** thoi_gian

# Function tính thuế thu nhập
def tinh_thue_thu_nhap(thu_nhap):
    if thu_nhap <= 5000000:
        return 0
    elif thu_nhap <= 10000000:
        return (thu_nhap - 5000000) * 0.05
    else:
        return 250000 + (thu_nhap - 10000000) * 0.1
```

```
# Test functions
print("Tính lãi suất:")
print(f"    Lãi đơn (10M, 5%, 2 năm): {lai_suat_don_gian(10000000, 5, 2):,.0f} VND")
print(f"    Lãi kép (10M, 5%, 2 năm): {lai_suat_kep(10000000, 5, 2):,.0f} VND")

print("\nTính thuế thu nhập:")
print(f"    Thu nhập 8M: {tinh_thue_thu_nhap(8000000):,.0f} VND")
print(f"    Thu nhập 15M: {tinh_thue_thu_nhap(15000000):,.0f} VND")
```



## TODO 7: Viết function tính toán tài chính

Viết function `tinh_tong_chi_phi_vay(tien_vay, lai_suat, so_thang)` tính tổng chi phí vay trả góp:

```
In [ ]: # TODO 7: Viết function tính chi phí vay trả góp
def tinh_tong_chi_phi_vay(tien_vay, lai_suat_thang, so_thang):
    # Tính theo công thức trả góp đều (annuity)
    # Số tiền trả hàng tháng = P * r * (1+r)^n / ((1+r)^n - 1)
    # Trong đó:
    # P = số tiền vay
    # r = lãi suất hàng tháng (decimal)
    # n = số tháng

    # Trả về dictionary với:
    # - 'tien_tra_hang_thang': số tiền trả hàng tháng
    # - 'tong_tien_tra': tổng số tiền phải trả
    # - 'tong_lai': tổng tiền lãi

    # Viết code ở đây
    pass

# Test function
ket_qua = tinh_tong_chi_phi_vay(100000000, 1.5, 12) # Vay 100M, lãi 1.5%/tháng, 12
print("Thông tin vay:")
print(f"    Trả hàng tháng: {ket_qua['tien_tra_hang_thang']:,.0f} VND")
print(f"    Tổng tiền trả: {ket_qua['tong_tien_tra']:,.0f} VND")
print(f"    Tổng tiền lãi: {ket_qua['tong_lai']:,.0f} VND")
```

## 8. Tạo thư viện Functions riêng

Tập hợp các functions thành một "thư viện" để sử dụng lại:

```
In [ ]: # Tạo class chứa các utility functions
class MathUtils:
    @staticmethod
    def tinh_hinh_tron(ban_kinh):
        import math
        dien_tich = math.pi * ban_kinh ** 2
        chu_vi = 2 * math.pi * ban_kinh
        return dien_tich, chu_vi

    @staticmethod
```



```
def tinh_hinh_vuong(canh):
    dien_tich = canh ** 2
    chu_vi = canh * 4
    return dien_tich, chu_vi

# Test utility class
dt, cv = MathUtils.tinh_hinh_tron(5)
print(f"Hình tròn r=5: DT={dt:.2f}, CV={cv:.2f}")

dt, cv = MathUtils.tinh_hinh_vuong(4)
print(f"Hình vuông cạnh=4: DT={dt}, CV={cv}")
```

## 9. Built-in Functions nâng cao

Khám phá các built-in functions hữu ích trong Python:

```
In [1]: # Built-in functions với lists
numbers = [1, 5, 3, 9, 2, 8, 4]

print("Built-in functions cơ bản:")
print(f" len(): {len(numbers)}")
print(f" sum(): {sum(numbers)}")
print(f" min(): {min(numbers)}")
print(f" max(): {max(numbers)}")
print(f" sorted(): {sorted(numbers)}")

# enumerate() - đánh số thứ tự
fruits = ['apple', 'banana', 'orange']
print("\nEnumerate:")
for i, fruit in enumerate(fruits):
    print(f" {i}: {fruit}")

# zip() - ghép lists
names = ['An', 'Bình', 'Chi']
ages = [20, 21, 19]
print("\nZip:")
for name, age in zip(names, ages):
    print(f" {name}: {age} tuổi")

# any() và all()
conditions = [True, False, True]
print(f"\nany(): {any(conditions)}") # True nếu có ít nhất 1 True
print(f"\nall(): {all(conditions)}") # True nếu tất cả đều True
```

Built-in functions cơ bản:

```
len(): 7
sum(): 32
min(): 1
max(): 9
sorted(): [1, 2, 3, 4, 5, 8, 9]
```

Enumerate:

```
0: apple
1: banana
2: orange
```

Zip:

```
An: 20 tuổi
Binh: 21 tuổi
Chi: 19 tuổi
```

any(): True

all(): False

```
In [4]: for packed in zip(names, ages):
        print(packed)
```

```
('An', 20)
('Binh', 21)
('Chi', 19)
```



## TODO 8: Thực hành Built-in Functions

Viết function sử dụng các built-in functions để phân tích dữ liệu:

```
In [ ]: # TODO 8: Function phân tích điểm thi
def phan_tich_diem_thi(danh_sach_diem):
    # Sử dụng các built-in functions để phân tích
    # Trả về dictionary với thông tin thống kê

    if not danh_sach_diem:
        return {'loi': 'Danh sách rỗng'}

    return {
        'so_hoc_sinh': len(danh_sach_diem),
        'diem_cao_nhat': max(danh_sach_diem),
        'diem_thap_nhat': min(danh_sach_diem),
        'tong_diem': sum(danh_sach_diem),
        'diem_trung_binh': round(sum(danh_sach_diem) / len(danh_sach_diem), 2),
        'co_ai_rot': any(diem < 5 for diem in danh_sach_diem),
        'tat_ca_deu_dat': all(diem >= 5 for diem in danh_sach_diem),
        'so_hoc_sinh_dat': sum(1 for diem in danh_sach_diem if diem >= 5),
        'top_3_cao_nhat': sorted(danh_sach_diem, reverse=True)[:3]
    }

# Test function
diem_lop = [8.5, 6.2, 4.5, 9.1, 7.8, 5.5, 3.2, 8.9, 7.1, 6.8]
thong_ke = phan_tich_diem_thi(diem_lop)
```

```
print("Thông kê điểm thi:")
for key, value in thong_ke.items():
    print(f" {key.replace('_', ' ').title()}: {value}")
```

## 10. Bài tập tổng hợp cuối buổi

Áp dụng tất cả kiến thức đã học để tạo một hệ thống quản lý đơn giản:

### TODO 9: Hệ thống quản lý điểm sinh viên

Tạo các functions để quản lý điểm sinh viên:

```
In [ ]: # TODO 9: Hệ thống quản lý điểm sinh viên
import math

# Database đơn giản (list of dictionaries)
sinh_vien_db = []

def them_sinh_vien(ma_sv, ten, diem_list):
    """Thêm sinh viên mới vào database"""
    # Viết code ở đây
    pass

def tinh_diem_trung_binh_sv(ma_sv):
    """Tính điểm trung bình của một sinh viên"""
    # Viết code ở đây
    pass

def xep_loai_hoc_luc(diem_tb):
    """Xếp loại học lực dựa trên điểm trung bình"""
    # Xuất sắc: >= 9, Giỏi: >= 8, Khá: >= 6.5, Trung bình: >= 5, Yếu: < 5
    # Viết code ở đây
    pass

def tim_sinh_vien_gioi_nhat():
    """Tìm sinh viên có điểm trung bình cao nhất"""
    # Viết code ở đây
    pass

def thong_ke_lop():
    """Thông kê tổng quan về lớp"""
    # Viết code ở đây
    pass

# Test hệ thống
print("=== THÊM SINH VIÊN ===")
them_sinh_vien("SV001", "Nguyễn Văn A", [8, 7, 9, 8.5])
them_sinh_vien("SV002", "Trần Thị B", [6, 7, 5, 8])
them_sinh_vien("SV003", "Lê Văn C", [9, 9.5, 8.5, 9])

print("\n=== THỐNG KÊ LỚP ===")
thong_ke_lop()
```

```
print("\n=== SINH VIÊN GIỎI NHẤT ===")
sv_gioi_nhat = tim_sinh_vien_gioi_nhat()
if sv_gioi_nhat:
    print(f"Sinh viên giỏi nhất: {sv_gioi_nhat['ten']} - Điểm TB: {sv_gioi_nhat['di']")
```

## Các lỗi thường gặp

1. **Quên return:** Function không trả về giá trị
2. **Tham số sai:** Truyền sai kiểu dữ liệu
3. **Không xử lý exception:** Không kiểm tra input
4. **Tên function không rõ ràng:** Khó hiểu mục đích

## 11. Tổng kết và Bài tập về nhà

### ✓ Những gì đã học buổi chiều:

1. **Functions thực tế:** Viết functions giải quyết vấn đề cụ thể
2. **Math library:** Sử dụng các functions toán học có sẵn
3. **Utility functions:** Tạo các functions tiện ích tái sử dụng
4. **Validation functions:** Kiểm tra tính hợp lệ của dữ liệu
5. **Built-in functions:** Sử dụng các functions có sẵn của Python

### 🎯 Bài tập về nhà:

1. **Hoàn thành tất cả TODO** trong notebook này (TODO 1-9)
2. **Viết function tính BMI** và phân loại sức khỏe
3. **Tạo calculator đơn giản** sử dụng functions
4. **Viết game đoán số** sử dụng functions
5. **Tạo menu điều khiển** cho hệ thống quản lý sinh viên

### 📁 Chuẩn bị cho buổi sau (Thứ 3 buổi sáng):

- **Function scope:** Phạm vi biến trong functions
- **Default parameters:** Tham số mặc định
- **Lambda functions:** Functions ẩn danh đơn giản