

WHILE LOOP, BREAK, CONTINUE VÀ MINI PROJECT PREPARATION

1. WHILE LOOP CƠ BẢN

1.1 While loop là gì?

- While loop là vòng lặp lặp lại một khối code **khi điều kiện còn đúng**
- Khác với for loop (biết trước số lần lặp), while loop có thể lặp **không xác định số lần**
- Cú pháp cơ bản:

```
while điều_kiện:
    # Code được thực hiện
    # Phải có cách để thay đổi điều kiện
```

```
In [ ]: # Ví dụ 1: Đếm từ 1 đến 5
count = 1
while count <= 5:
    print(f"Số thứ {count}")
    count += 1 # Quan trọng: phải tăng count để tránh vòng lặp vô hạn

print("Kết thúc vòng lặp")
```

```
In [ ]: # Ví dụ 2: Nhập số đến khi đúng
number = 0
while number != 42:
    number = int(input("Nhập số 42 để thoát: "))
    if number != 42:
        print("Sai rồi, thử lại!")

print("Chính xác! Bạn đã nhập đúng số 42")
```

1.2 So sánh While vs For

| While Loop | For Loop |
|------------------------------|-------------------------------|
| Không biết trước số lần lặp | Biết trước số lần lặp |
| Dựa vào điều kiện | Dựa vào sequence/range |
| Có thể vòng lặp vô hạn | Luôn kết thúc |
| Phù hợp với input validation | Phù hợp với danh sách, số đếm |

2. BREAK STATEMENT

2.1 Break là gì?

- `break` được dùng để **thoát khỏi vòng lặp ngay lập tức**
- Khi gặp `break`, chương trình nhảy ra khỏi vòng lặp và tiếp tục code phía sau
- Dùng trong cả `while` và `for` loop

```
In [ ]: # Ví dụ 1: Tìm số đầu tiên chia hết cho 7
number = 1
while True: # Vòng Lặp vô hạn
    if number % 7 == 0:
        print(f"Số đầu tiên chia hết cho 7 là: {number}")
        break # Thoát khỏi vòng Lặp
    number += 1

print("Đã tìm thấy và thoát vòng lặp")
```

```
In [ ]: # Ví dụ 2: Menu đơn giản
while True:
    print("\n=== MENU ===")
    print("1. Chào hỏi")
    print("2. Tính toán")
    print("3. Thoát")

    choice = input("Chọn (1-3): ")

    if choice == "1":
        print("Xin chào!")
    elif choice == "2":
        a = int(input("Nhập số a: "))
        b = int(input("Nhập số b: "))
        print(f"Tổng: {a + b}")
    elif choice == "3":
        print("Tạm biệt!")
        break # Thoát menu
    else:
        print("Lựa chọn không hợp lệ!")
```

3. CONTINUE STATEMENT

3.1 Continue là gì?

- `continue` được dùng để **bỏ qua phần còn lại của lần lặp hiện tại**
- Nhảy ngay lên đầu vòng lặp để kiểm tra điều kiện tiếp theo
- Khác với `break` (thoát hoàn toàn), `continue` chỉ bỏ qua lần lặp hiện tại

```
In [ ]: # Ví dụ 1: In các số lẻ từ 1 đến 10
number = 0
while number < 10:
    number += 1
    if number % 2 == 0: # Nếu Là số chẵn
        continue
```

```

        continue # Bỏ qua, không in
    print(f"Số lẻ: {number}")

```

```

In [ ]: # Ví dụ 2: Xử lý danh sách với điều kiện
numbers = [1, -2, 3, -4, 5, -6, 7]
i = 0
positive_sum = 0

while i < len(numbers):
    if numbers[i] < 0:
        print(f"Bỏ qua số âm: {numbers[i]}")
        i += 1
        continue # Bỏ qua số âm

    positive_sum += numbers[i]
    print(f"Cộng số dương: {numbers[i]}")
    i += 1

print(f"Tổng các số dương: {positive_sum}")

```

4. CÁC PATTERN THƯỜNG GẶP

4.1 Input Validation Pattern

```

In [ ]: # Pattern 1: Validation với while True
def get_positive_number():
    while True:
        try:
            number = int(input("Nhập số dương: "))
            if number > 0:
                return number
            else:
                print("Số phải lớn hơn 0!")
        except ValueError:
            print("Vui lòng nhập số nguyên!")

# Sử dụng
valid_number = get_positive_number()
print(f"Số hợp lệ: {valid_number}")

```

4.2 Game Loop Pattern

```

In [ ]: # Pattern 2: Game đoán số
import random

def guessing_game():
    target = random.randint(1, 100)
    attempts = 0
    max_attempts = 7

    print("=== GAME ĐOÁN SỐ ===")
    print(f"Tôi đã nghĩ ra một số từ 1-100. Bạn có {max_attempts} lần đoán!")

```

```

while attempts < max_attempts:
    try:
        guess = int(input(f"Lần đoán {attempts + 1}: "))
        attempts += 1

        if guess == target:
            print(f"🎉 Chúc mừng! Bạn đã đoán đúng sau {attempts} lần!")
            break
        elif guess < target:
            print("📉 Số bạn đoán nhỏ hơn!")
        else:
            print("📈 Số bạn đoán lớn hơn!")

        if attempts == max_attempts:
            print(f"😞 Hết lượt! Số đúng là {target}")

    except ValueError:
        print("Vui lòng nhập số nguyên!")
        attempts -= 1 # Không tính lần nhập sai

# Chạy game
guessing_game()

```

5. LƯU Ý QUAN TRỌNG

5.1 Tránh vòng lặp vô hạn

```

In [ ]: # ❌ SAI: Vòng Lặp vô hạn
# count = 1
# while count <= 5:
#     print(count)
#     # Quên tăng count -> vòng Lặp vô hạn!

# ✅ ĐÚNG: Luôn có cách thay đổi điều kiện
count = 1
while count <= 5:
    print(count)
    count += 1 # Quan trọng!

```

5.2 Sử dụng break và continue hiệu quả

```

In [ ]: # Ví dụ: Tìm và xử lý dữ liệu
data = [1, 2, -1, 4, 0, 6, -3, 8]
processed = []
i = 0

while i < len(data):
    current = data[i]

    # Dừng khi gặp số 0
    if current == 0:

```

```

        print("Gặp số 0, dừng xử lý")
        break

    # Bỏ qua số âm
    if current < 0:
        print(f"Bỏ qua số âm: {current}")
        i += 1
        continue

    # Xử lý số dương
    processed.append(current * 2)
    print(f"Xử lý {current} -> {current * 2}")
    i += 1

print(f"Kết quả: {processed}")

```

6. BÀI TẬP THỰC HÀNH

Bài tập 1: Máy tính đơn giản

```

In [ ]: # TODO: Hoàn thành máy tính với menu
def simple_calculator():
    print("=== MÁY TÍNH ĐƠN GIẢN ===")

    while True:
        print("\nChọn phép tính:")
        print("1. Cộng (+)")
        print("2. Trừ (-)")
        print("3. Nhân (*)")
        print("4. Chia (/)")
        print("5. Thoát")

        choice = input("Lựa chọn (1-5): ")

        if choice == "5":
            print("Tạm biệt!")
            break

        if choice in ["1", "2", "3", "4"]:
            # TODO: Hoàn thành code tính toán
            pass
        else:
            print("Lựa chọn không hợp lệ!")

    # Gọi hàm để test
    # simple_calculator()

```

Bài tập 2: Quản lý danh sách số

```

In [ ]: # TODO: Tạo chương trình quản lý danh sách số
def number_manager():
    numbers = []

```

```

while True:
    print(f"\nDanh sách hiện tại: {numbers}")
    print("1. Thêm số")
    print("2. Xóa số")
    print("3. Tính tổng")
    print("4. Tìm số lớn nhất")
    print("5. Thoát")

    choice = input("Chọn: ")

    # TODO: Hoàn thành các chức năng
    if choice == "1":
        # Thêm số vào danh sách
        pass
    elif choice == "2":
        # Xóa số khỏi danh sách
        pass
    elif choice == "3":
        # Tính tổng
        pass
    elif choice == "4":
        # Tìm số lớn nhất
        pass
    elif choice == "5":
        break
    else:
        print("Lựa chọn không hợp lệ!")

# number_manager()

```

7. CHUẨN BỊ CHO MINI PROJECT

7.1 Ý tưởng Mini Project: Calculator nâng cao

Dự án sẽ kết hợp tất cả kiến thức đã học:

- Variables và Data types
- Operators và Conditions
- Loops (for và while)
- Functions (sẽ học tuần sau)

7.2 Tính năng Calculator

1. **Menu chính:** Sử dụng while loop
2. **Validation:** Kiểm tra input hợp lệ
3. **Tính toán:** Các phép toán cơ bản + nâng cao
4. **Lịch sử:** Lưu các phép tính đã thực hiện
5. **Thoát:** Sử dụng break

```

In [ ]: # Template cho Mini Project Calculator
def advanced_calculator():
    print("🧮 === CALCULATOR NÂNG CAO === 🧮")
    history = [] # Lưu lịch sử tính toán

    while True:
        print("\n" + "="*40)
        print("1. Phép tính cơ bản (+, -, *, /)")
        print("2. Phép tính nâng cao (^, sqrt, %)")
        print("3. Xem lịch sử")
        print("4. Xóa lịch sử")
        print("5. Thoát")
        print("="*40)

        choice = input("Chọn chức năng (1-5): ")

        if choice == "1":
            # TODO: Phép tính cơ bản
            print("Chức năng đang phát triển...")

        elif choice == "2":
            # TODO: Phép tính nâng cao
            print("Chức năng đang phát triển...")

        elif choice == "3":
            # TODO: Hiển thị lịch sử
            if history:
                print("\n📋 Lịch sử tính toán:")
                for i, record in enumerate(history, 1):
                    print(f"{i}. {record}")
            else:
                print("Chưa có lịch sử tính toán")

        elif choice == "4":
            # TODO: Xóa lịch sử
            history.clear()
            print("✅ Đã xóa lịch sử")

        elif choice == "5":
            print("👋 Cảm ơn bạn đã sử dụng Calculator!")
            break

        else:
            print("❌ Lựa chọn không hợp lệ! Vui lòng chọn 1-5")
            continue

    # Uncomment để chạy
    # advanced_calculator()

```

8. TÓM TẮT BÀI HỌC

🎯 Kiến thức đã học:

1. **While Loop:** Lặp khi điều kiện còn đúng
2. **Break:** Thoát khỏi vòng lặp ngay lập tức
3. **Continue:** Bỏ qua lần lặp hiện tại, tiếp tục lần lặp tiếp theo
4. **Patterns:** Input validation, Game loop, Menu system



Lưu ý quan trọng:

- Luôn có cách thay đổi điều kiện để tránh vòng lặp vô hạn
- Sử dụng break để thoát khi cần
- Sử dụng continue để bỏ qua xử lý không cần thiết
- Validation input để tránh lỗi



Chuẩn bị buổi sau:

- Hoàn thành Mini Project Calculator
- Kết hợp tất cả kiến thức đã học
- Chuẩn bị cho phần Functions tuần tới