Tuần 27 - Thứ 2: Python Data Structures -List cơ bản

Thời gian: 09:00-12:00

Giáo viên: Nguyễn Mạnh Duy

Mục tiêu học tập:

- Hiểu khái niệm List trong Python
- Nắm vững cách tạo và truy cập List
- Thành thạo List indexing và slicing
- Áp dụng các thao tác cơ bản với List

1. List là gì?

List là một cấu trúc dữ liệu trong Python cho phép lưu trữ nhiều giá trị trong một biến duy nhất.

Đặc điểm của List:

- Có thứ tự (ordered): Các phần tử có vị trí cố định
- Có thể thay đổi (mutable): Có thể thêm, xóa, sửa phần tử
- Cho phép trùng lặp: Có thể chứa các giá trị giống nhau
- Đa kiểu dữ liệu: Có thể chứa nhiều kiểu dữ liệu khác nhau

```
In [1]: # Ví dụ về List
    numbers = [1, 2, 3, 4, 5]
    names = ["An", "Bình", "Cường"]
    mixed = [1, "Hello", 3.14, True]
    empty_list = []

    print("Danh sách số:", numbers)
    print("Danh sách tên:", names)
    print("Danh sách hỗn hợp:", mixed)
    print("Danh sách rỗng:", empty_list)

Danh sách số: [1, 2, 3, 4, 5]
    Danh sách tên: ['An', 'Bình', 'Cường']
    Danh sách hỗn hợp: [1, 'Hello', 3.14, True]
    Danh sách rỗng: []
```

2. Cách tạo List

2.1 Tạo List trực tiếp

```
In [2]: # Tạo List với dấu ngoặc vuông []
fruits = ["táo", "cam", "chuối", "xoài"]
scores = [85, 92, 78, 96, 88]

print("Danh sách trái cây:", fruits)
print("Danh sách điểm:", scores)

Danh sách trái cây: ['táo', 'cam', 'chuối', 'xoài']
Danh sách điểm: [85, 92, 78, 96, 88]
```

2.2 Tạo List bằng hàm list()

```
In [3]: # Tạo list từ string
    letters = list("HELLO")
    print("Chữ cái:", letters)

# Tạo list từ range
    numbers = list(range(1, 6))
    print("Số từ 1-5:", numbers)

# Tạo list rỗng
    empty = list()
    print("List rỗng:", empty)

Chữ cái: ['H', 'E', 'L', 'L', '0']
    Số từ 1-5: [1, 2, 3, 4, 5]
    List rỗng: []
```

3. Truy cập phần tử trong List

3.1 List Indexing (Chi muc)

- Python sử dụng chỉ mục bắt đầu từ 0
- Chỉ mục âm để truy cập từ cuối list

```
In [4]: colors = ["đỏ", "xanh", "vàng", "tím", "cam"]

# Chỉ mục dương (từ đầu)
print("Phần tử đầu tiên:", colors[0]) # đỏ
print("Phần tử thứ hai:", colors[1]) # xanh
print("Phần tử thứ ba:", colors[2]) # vàng

# Chỉ mục âm (từ cuối)
print("Phần tử cuối cùng:", colors[-1]) # cam
print("Phần tử áp cuối:", colors[-2]) # tím

# Hiển thị chỉ mục và giá trị
```

```
print("\nChi mục và giá trị:")
for i in range(len(colors)):
    print(f"Index {i}: {colors[i]}")

Phần tử đầu tiên: đổ
Phần tử thứ hai: xanh
Phần tử thứ ba: vàng
Phần tử cuối cùng: cam
Phần tử ap cuối: tím

Chỉ mục và giá trị:
Index 0: đổ
Index 1: xanh
Index 2: vàng
Index 3: tím
Index 4: cam
```

3.2 List Slicing (Cắt lát)

```
Cú pháp: list[start:stop:step]
```

- start : vị trí bắt đầu (bao gồm)
- stop : vị trí kết thúc (không bao gồm)
- step: bước nhảy (mặc định là 1)

```
In [5]: numbers = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
        # Slicing cơ bản
        print("Toàn bô list:", numbers)
        print("Từ index 2 đến 5:", numbers[2:6]) # [2, 3, 4, 5]
        print("Từ đầu đến index 4:", numbers[:5]) # [0, 1, 2, 3, 4]
        print("Từ index 5 đến cuối:", numbers[5:]) # [5, 6, 7, 8, 9]
        # Slicing với step
        print("Các số chẵn:", numbers[::2])
                                                  # [0, 2, 4, 6, 8]
        print("Các số lẻ:", numbers[1::2])
                                                    # [1, 3, 5, 7, 9]
        print("Đảo ngược list:", numbers[::-1])
                                                  # [9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
       Toàn bộ list: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
       Từ index 2 đến 5: [2, 3, 4, 5]
       Từ đầu đến index 4: [0, 1, 2, 3, 4]
       Từ index 5 đến cuối: [5, 6, 7, 8, 9]
       Các số chẵn: [0, 2, 4, 6, 8]
```

4. Thao tác cơ bản với List

Đảo ngược list: [9, 8, 7, 6, 5, 4, 3, 2, 1, 0]

4.1 Độ dài của List

Các số lẻ: [1, 3, 5, 7, 9]

```
In [6]: subjects = ["Toán", "Lý", "Hóa", "Sinh", "Văn"]
```

```
print("Số môn học:", len(subjects))
print("List có rỗng không?", len(subjects) == 0)

Số môn học: 5
List có rỗng không? False
```

4.2 Kiểm tra phần tử có tồn tại

```
In [7]: fruits = ["táo", "cam", "chuối", "xoài"]

print("'táo' có trong list?", "táo" in fruits) # True
print("'dưa' có trong list?", "dưa" in fruits) # False
print("'cam' không trong list?", "cam" not in fruits) # False

'táo' có trong list? True
'dưa' có trong list? False
'cam' không trong list? False
```

4.3 Nối List (Concatenation)

```
In [8]: list1 = [1, 2, 3]
    list2 = [4, 5, 6]
    list3 = [7, 8, 9]

# Nối với toán tử +
    combined = list1 + list2
    print("Nối 2 list:", combined)

# Nối nhiều list
    all_numbers = list1 + list2 + list3
    print("Nối 3 list:", all_numbers)

Nối 2 list: [1, 2, 3, 4, 5, 6]
    Nối 3 list: [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

4.4 Lặp lại List (Repetition)

```
In [9]: # Lặp lại với toán tử *
pattern = [1, 0]
repeated = pattern * 5
print("Lặp lại pattern:", repeated)

# Tạo List với giá trị giống nhau
zeros = [0] * 10
print("10 số 0:", zeros)

Lặp lại pattern: [1, 0, 1, 0, 1, 0, 1, 0]
10 số 0: [0, 0, 0, 0, 0, 0, 0, 0, 0]
```

4.5 So sánh List

```
In [10]: list_a = [1, 2, 3]
    list_b = [1, 2, 3]
    list_c = [1, 2, 4]
```

```
list_d = [3, 2, 1]
print("list_a == list_b:", list_a == list_b) # True
print("list_a == list_c:", list_a == list_c) # False
print("list_a < list_c:", list_a < list_c) # True (so sánh từng phần tử)

list_a == list_b: True
list_a == list_c: False
list_a < list_c: True</pre>
```

5. Các hàm built-in thường dùng với List

```
In [11]: scores = [85, 92, 78, 96, 88, 91]
         print("Danh sách điểm:", scores)
         print("Điểm cao nhất:", max(scores))
         print("Điểm thấp nhất:", min(scores))
         print("Tổng điểm:", sum(scores))
         print("Điểm trung bình:", sum(scores) / len(scores))
         # Sắp xếp (không thay đổi list gốc)
         print("Sắp xếp tăng dần:", sorted(scores))
         print("Sắp xếp giảm dần:", sorted(scores, reverse=True))
         print("List gốc không đổi:", scores)
        Danh sách điểm: [85, 92, 78, 96, 88, 91]
        Điểm cao nhất: 96
        Điểm thấp nhất: 78
        Tổng điểm: 530
        Điểm trung bình: 88.333333333333333
        Sắp xếp tăng dần: [78, 85, 88, 91, 92, 96]
        Sắp xếp giảm dần: [96, 92, 91, 88, 85, 78]
        List gốc không đổi: [85, 92, 78, 96, 88, 91]
```

6. Ví dụ thực tế

6.1 Quản lý danh sách học sinh

```
In [12]: # Danh sách học sinh Lớp 12A1
students = ["Nguyễn Văn An", "Trần Thị Bình", "Lê Văn Cường", "Phạm Thị Dung"]

print("=== DANH SÁCH LỚP 12A1 ===")
print(f"Sĩ số lớp: {len(students)} học sinh")
print("\nDanh sách học sinh:")

for i in range(len(students)):
    print(f"{i+1:2d}. {students[i]}")

# Kiểm tra học sinh
name_to_check = "Lê Văn Cường"
if name_to_check in students:
    position = students.index(name_to_check) + 1
    print(f"\n{name_to_check} có trong danh sách (số thứ tự: {position})")
```

```
else:
    print(f"\n{name_to_check} không có trong danh sách")

=== DANH SÁCH LỚP 12A1 ===
Sĩ số lớp: 4 học sinh

Danh sách học sinh:

1. Nguyễn Văn An
2. Trần Thị Bình
3. Lê Văn Cường
4. Phạm Thị Dung

Lê Văn Cường có trong danh sách (số thứ tự: 3)
```

6.2 Phân tích điểm số

```
In [13]: # Điểm kiểm tra của lớp
         test_scores = [8.5, 7.2, 9.1, 6.8, 8.9, 7.5, 9.3, 8.1, 7.8, 8.7]
         print("=== PHÂN TÍCH ĐIỂM KIỂM TRA ===")
         print(f"Số học sinh: {len(test_scores)}")
         print(f"Điểm cao nhất: {max(test_scores)}")
         print(f"Điểm thấp nhất: {min(test_scores)}")
         print(f"Điểm trung bình: {sum(test_scores)/len(test_scores):.2f}")
         # Phân Loại điểm
         excellent = [score for score in test_scores if score >= 9.0]
         good = [score for score in test_scores if 8.0 <= score < 9.0]</pre>
         average = [score for score in test_scores if 7.0 <= score < 8.0]</pre>
         below_average = [score for score in test_scores if score < 7.0]
         print(f"\nPhân loại:")
         print(f"Xuất sắc (≥9.0): {len(excellent)} học sinh")
         print(f"Gioi (8.0-8.9): {len(good)} hoc sinh")
         print(f"Khá (7.0-7.9): {len(average)} học sinh")
         print(f"Trung binh (<7.0): {len(below_average)} hoc sinh")</pre>
        === PHÂN TÍCH ĐIỂM KIỂM TRA ===
        Số học sinh: 10
        Điểm cao nhất: 9.3
        Điểm thấp nhất: 6.8
        Điểm trung bình: 8.19
        Phân loại:
        Xuất sắc (≥9.0): 2 học sinh
        Giỏi (8.0-8.9): 4 học sinh
        Khá (7.0-7.9): 3 học sinh
        Trung bình (<7.0): 1 học sinh
```

7. Bài tập thực hành

Bài tập 1: Tạo và truy cập List

```
In [14]: # TODO: Tạo list chứa tên 5 môn học yêu thích của bạn
favorite_subjects = []

# TODO: In ra môn học đầu tiên và cuối cùng

# TODO: In ra 3 môn học đầu tiên

# TODO: Kiểm tra xem "Toán" có trong danh sách không
```

Bài tập 2: Thao tác với List số

```
In [15]: # TODO: Tạo list chứa điểm của 10 học sinh (từ 1.0 đến 10.0)
student_scores = []
# TODO: Tính điểm trung bình
# TODO: Tìm điểm cao nhất và thấp nhất
# TODO: Đếm có bao nhiêu học sinh đạt điểm >= 8.0
```

8. Tóm tắt

Những điều đã học:

- 1. List là cấu trúc dữ liệu có thứ tự, có thể thay đổi
- 2. **Indexing**: Truy cập phần tử bằng chỉ mục (bắt đầu từ 0)
- 3. **Slicing**: Cắt lát list với cú pháp [start:stop:step]
- 4. Thao tác cơ bản: nối (+), lặp (*), kiểm tra (in), độ dài (len)
- 5. Hàm built-in: max(), min(), sum(), sorted()

Chuẩn bi cho buổi sau:

- Ôn tập các thao tác cơ bản với List
- Thực hành các bài tập đã cho
- Chuẩn bị cho việc học List methods (append, remove, ...)

9. List Methods co ban (Preview)

9.1 Thêm phần tử vào List

```
In [16]: # Giới thiệu sơ qua các methods sẽ học chi tiết buổi sau
fruits = ["táo", "cam"]
print("Ban đầu:", fruits)

# append(): thêm 1 phần tử vào cuối
fruits.append("chuối")
print("Sau khi append:", fruits)
```

```
# insert(): thêm phần tử vào vị trí chỉ định
fruits.insert(1, "xoài")
print("Sau khi insert:", fruits)

print("\n=> Buổi sau sẽ học chi tiết về các methods này!")

Ban đầu: ['táo', 'cam']
Sau khi append: ['táo', 'cam', 'chuối']
Sau khi insert: ['táo', 'xoài', 'cam', 'chuối']

=> Buổi sau sẽ học chi tiết về các methods này!
```

10. Lỗi thường gặp khi làm việc với List

10.1 IndexError - Truy cập index không tồn tại

```
In [17]: numbers = [1, 2, 3, 4, 5]
         print("List có:", len(numbers), "phần tử")
         print("Index hợp lệ: 0 đến", len(numbers)-1)
         # Cách tránh lỗi
         index = 10
         if index < len(numbers):</pre>
             print(f"Phần tử tại index {index}:", numbers[index])
         else:
             print(f"Index {index} không hợp lệ!")
         # Sử dụng negative index an toàn
         if len(numbers) > 0:
             print("Phần tử cuối:", numbers[-1])
         else:
             print("List rong!")
        List có: 5 phần tử
        Index hợp lệ: 0 đến 4
        Index 10 không hợp lệ!
        Phần tử cuối: 5
```

10.2 Slicing an toàn

```
In [18]: data = [1, 2, 3, 4, 5]

# Slicing không gây lỗi ngay cả khi index vượt quá
print("data[2:10]:", data[2:10]) # Chỉ lấy đến hết list
print("data[10:20]:", data[10:20]) # Trả về list rỗng
print("data[-100:2]:", data[-100:2]) # Bắt đầu từ đầu list

# Copy toàn bộ list an toàn
copy_list = data[:] # Hoặc data.copy()
print("Bản copy:", copy_list)
```

```
data[2:10]: [3, 4, 5]
data[10:20]: []
data[-100:2]: [1, 2]
Bån copy: [1, 2, 3, 4, 5]
```

11. Ứng dụng thực tế nâng cao

11.1 Hệ thống quản lý thư viện đơn giản

```
In [19]: # Dữ liệu thư viện - sử dụng nhiều list song song
         book_ids = ["B001", "B002", "B003", "B004", "B005"]
         book_titles = [
             "Lập trình Python cơ bản",
             "Khoa học dữ liệu với Python",
             "Machine Learning thực hành",
             "Web Development với Django"
             "Computer Vision với OpenCV"
         book_authors = ["Nguyễn Văn A", "Trần Thị B", "Lê Văn C", "Phạm Thị D", "Hoàng Văn
         book_years = [2020, 2021, 2019, 2022, 2023]
         book available = [True, False, True, True, False] # Có sẵn để mươn
         print("=== HP THONG THU VIPN ===")
         print(f"Tổng số sách: {len(book_titles)}")
         # Hiển thị thông tin chi tiết
         def display library():
             print("\n" + "="*80)
             print(f"{'ID':<6} {'Tên sách':<30} {'Tác giả':<15} {'Năm':<6} {'Trạng thái':<12</pre>
             print("="*80)
             for i in range(len(book_ids)):
                  status = "Có sẵn" if book_available[i] else "Đã mượn"
                  print(f"{book_ids[i]:<6} {book_titles[i]:<30} {book_authors[i]:<15} {book_y</pre>
         display library()
         # Thống kê
         available count = sum(book available) # Đếm True values
         borrowed_count = len(book_available) - available_count
         print(f"\nThống kê:")
         print(f"- Sách có sẵn: {available_count} cuốn")
         print(f"- Sách đã mượn: {borrowed count} cuốn")
         print(f"- Tỷ lệ mượn: {borrowed_count/len(book_available)*100:.1f}%")
```

```
=== HỆ THỐNG THƯ VIỆN ===
Tổng số sách: 5
```

- Tỷ lê mượn: 40.0%

______ TD Tên sách Tác giả Năm Trạng thái ______ Nguyễn Văn A B001 Lập trình Python cơ bản 2020 Có sẵn B002 Khoa học dữ liệu với Python Trần Thị B 2021 Đã mươn 2019 Có sẵn Machine Learning thực hành Lê Văn C B003 B004 Pham Thi D 2022 Có sẵn Web Development với Django B005 Computer Vision với OpenCV Hoàng Văn E 2023 Đã mượn Thống kê: - Sách có sẵn: 3 cuốn - Sách đã mượn: 2 cuốn

11.2 Phân tích dữ liệu bán hàng

```
In [20]: # Dữ Liệu bán hàng 7 ngày
         days = ["Thứ 2", "Thứ 3", "Thứ 4", "Thứ 5", "Thứ 6", "Thứ 7", "Chủ nhật"]
         daily sales = [1200000, 1500000, 1800000, 2100000, 2500000, 3200000, 2800000] # VN
         daily_customers = [25, 32, 38, 45, 52, 68, 58]
         print("=== PHÂN TÍCH BÁN HÀNG TUẦN ===")
         # Báo cáo hàng ngày
         print("\nBáo cáo chi tiết:")
         print("-" * 55)
         print(f"{'Ngày':<12} {'Doanh thu':<15} {'Khách hàng':<12} {'TB/khách':<15}")</pre>
         print("-" * 55)
         for i in range(len(days)):
             avg_per_customer = daily_sales[i] / daily_customers[i]
             print(f"{days[i]:<12} {daily_sales[i]:,<15} {daily_customers[i]:<12} {avg_per_c</pre>
         # Thống kê tổng quan
         total_sales = sum(daily_sales)
         total_customers = sum(daily_customers)
         avg daily sales = total sales / len(daily sales)
         best_day_index = daily_sales.index(max(daily_sales))
         worst_day_index = daily_sales.index(min(daily_sales))
         print("\n=== THONG KÊ TONG QUAN ===")
         print(f"Tổng doanh thu tuần: {total sales:,} VND")
         print(f"Tổng số khách: {total_customers} người")
         print(f"Doanh thu TB/ngày: {avg_daily_sales:,.0f} VND")
         print(f"Ngày bán chạy nhất: {days[best_day_index]} ({daily_sales[best_day_index]:,}
         print(f"Ngày bán ít nhất: {days[worst_day_index]} ({daily_sales[worst_day_index]:,}
         # Phân tích xu hướng
         weekday_sales = daily_sales[:5] # Thứ 2 đến Thứ 6
         weekend_sales = daily_sales[5:] # Thứ 7 và Chủ nhật
         print(f"\nPhân tích xu hướng:")
```

```
print(f"Doanh thu ngày thường: {sum(weekday_sales):,} VND (TB: {sum(weekday_sales)/
 print(f"Doanh thu cuối tuần: {sum(weekend_sales):,} VND (TB: {sum(weekend_sales)/le
=== PHÂN TÍCH BÁN HÀNG TUẦN ===
Báo cáo chi tiết:
           Doanh thu
Ngày
                            Khách hàng
                                        TB/khách
Thứ 2
            1200000,,,,,,, 25
                                         48,000
Thứ 3
          1500000,,,,,,,, 32
                                        46,875
            1800000,,,,,,,, 38
                                         47,368
Thứ 4
Thứ 5
            2100000,,,,,,,, 45
                                        46,667
Thứ 6
            2500000,,,,,,, 52
                                        48,077
            3200000,,,,,,, 68
Thứ 7
                                        47,059
Chủ nhật
            2800000,,,,,,, 58
                                        48,276
=== THỐNG KÊ TỔNG QUAN ===
Tổng doanh thu tuần: 15,100,000 VND
Tổng số khách: 318 người
Doanh thu TB/ngày: 2,157,143 VND
Ngày bán chạy nhất: Thứ 7 (3,200,000 VND)
Ngày bán ít nhất: Thứ 2 (1,200,000 VND)
Phân tích xu hướng:
Doanh thu ngày thường: 9,100,000 VND (TB: 1,820,000)
Doanh thu cuối tuần: 6,000,000 VND (TB: 3,000,000)
```

12. Bài tập thực hành trong lớp (30 phút)

Bài tập 3: Quản lý điểm thi toàn trường

```
In [21]: # Dữ Liệu điểm thi 3 môn của 20 học sinh
         student_names = [
             "Nguyễn Văn An", "Trần Thị Bình", "Lê Văn Cường", "Phạm Thị Dung", "Hoàng Văn E
             "Vũ Thị Giang", "Đặng Văn Hải", "Bùi Thị Lan", "Ngô Văn Minh", "Lý Thị Nga",
             "Tạ Văn Phong", "Đinh Thị Quỳnh", "Hồ Văn Sơn", "Trương Thị Thảo", "Lưu Văn Tùn
             "Phan Thị Uyên", "Đỗ Văn Vinh", "Chu Thị Xuân", "Mai Văn Yên", "Kiều Thị Zung"
         math_scores = [8.5, 7.2, 9.1, 6.8, 8.9, 7.5, 9.3, 8.1, 7.8, 8.7,
                       6.9, 9.5, 8.2, 7.4, 8.8, 9.0, 7.6, 8.4, 9.2, 7.9]
         physics_scores = [7.8, 8.1, 8.7, 7.0, 9.2, 8.3, 8.9, 7.7, 8.5, 7.9,
                          7.2, 9.1, 8.0, 7.8, 8.6, 8.8, 7.5, 8.2, 9.0, 8.1]
         chemistry_scores = [8.2, 7.5, 9.0, 6.9, 8.7, 7.8, 9.1, 8.0, 7.6, 8.5,
                            7.1, 9.3, 8.3, 7.7, 8.9, 8.6, 7.4, 8.1, 9.2, 7.8]
         print("=== HỆ THỐNG QUẢN LÝ ĐIỂM THI TOÀN TRƯỜNG ===")
         print(f"Số học sinh: {len(student names)}")
         print("Các môn thi: Toán, Lý, Hóa")
         # TODO: Học sinh thực hành các yêu cầu sau:
```

```
print("\n=== YÊU CÂU THỰC HÀNH ===")
 print("1. Tính điểm trung bình 3 môn của từng học sinh")
 print("2. Tìm học sinh có điểm TB cao nhất")
 print("3. Tìm học sinh có điểm Toán cao nhất")
 print("4. Đếm số học sinh đạt điểm giỏi (TB ≥ 8.0) mỗi môn")
 print("5. Tính điểm TB của toàn trường mỗi môn")
 print("6. Tìm 5 học sinh có điểm TB cao nhất")
 # Gợi ý bắt đầu:
 print("\n=== GOI Ý BẮT ĐẦU ===")
 print("# Ví dụ tính điểm TB học sinh đầu tiên:")
 first_student_avg = (math_scores[0] + physics_scores[0] + chemistry_scores[0]) / 3
 print(f"{student_names[0]}: {first_student_avg:.2f}")
 print("\n# Bây giờ hãy làm cho tất cả học sinh...")
=== HỆ THỐNG QUẨN LÝ ĐIỂM THI TOÀN TRƯỜNG ===
Số học sinh: 20
Các môn thi: Toán, Lý, Hóa
=== YÊU CẦU THỰC HÀNH ===
1. Tính điểm trung bình 3 môn của từng học sinh
2. Tìm học sinh có điểm TB cao nhất
3. Tìm học sinh có điểm Toán cao nhất
4. Đếm số học sinh đạt điểm giỏi (TB ≥ 8.0) mỗi môn
5. Tính điểm TB của toàn trường mỗi môn
6. Tìm 5 học sinh có điểm TB cao nhất
=== GƠI Ý BẮT ĐẦU ===
# Ví dụ tính điểm TB học sinh đầu tiên:
Nguyễn Văn An: 8.17
# Bây giờ hãy làm cho tất cả học sinh...
```

Bài tập 4: Phân tích dữ liệu nhiệt độ

```
In [22]: # Nhiệt độ hàng ngày trong tháng (30 ngày)
         daily temperatures = [
             28, 29, 27, 30, 32, 31, 29, 28, 30, 33,
             34, 32, 31, 29, 28, 30, 31, 33, 35, 34,
             32, 30, 29, 31, 33, 34, 32, 30, 28, 29
         print("=== PHÂN TÍCH DỮ LIỆU NHIỆT ĐỘ THÁNG ===")
         print(f"Số ngày quan sát: {len(daily_temperatures)}")
         # TODO: Thực hành phân tích dữ liệu
         print("\n=== NHIÊM VU PHÂN TÍCH ===")
         print("1. Tìm nhiệt độ cao nhất và thấp nhất")
         print("2. Tính nhiệt độ trung bình của tháng")
         print("3. Đếm số ngày có nhiệt độ > 30 độ")
         print("4. Tìm 10 ngày đầu và 10 ngày cuối của tháng")
         print("5. Tính nhiệt độ TB của từng tuần (7 ngày)")
         print("6. Tìm những ngày có nhiệt độ bằng nhiệt độ TB")
         # Mẫu giải một phần:
```

```
print("\n=== DEMO GIÁI ===")
 max_temp = max(daily_temperatures)
 min temp = min(daily temperatures)
 avg_temp = sum(daily_temperatures) / len(daily_temperatures)
 print(f"Nhiệt độ cao nhất: {max temp}°C")
 print(f"Nhiệt độ thấp nhất: {min_temp}°C")
 print(f"Nhiệt độ trung bình: {avg_temp:.1f}°C")
 # Tìm vị trí nhiệt độ cao nhất
 max_day = daily_temperatures.index(max_temp) + 1
 print(f"Ngày nóng nhất: ngày thứ {max_day}")
 print("\n=== HÃY TIẾP TỤC CÁC YÊU CẦU KHÁC! ===")
=== PHÂN TÍCH DỮ LIỆU NHIỆT ĐỘ THÁNG ===
Số ngày quan sát: 30
=== NHIỆM VỤ PHÂN TÍCH ===
1. Tìm nhiệt độ cao nhất và thấp nhất
2. Tính nhiệt độ trung bình của tháng
3. Đếm số ngày có nhiệt độ > 30 độ
4. Tìm 10 ngày đầu và 10 ngày cuối của tháng
5. Tính nhiệt độ TB của từng tuần (7 ngày)
6. Tìm những ngày có nhiệt độ bằng nhiệt độ TB
=== DEMO GIAI ===
Nhiệt độ cao nhất: 35°C
Nhiệt độ thấp nhất: 27°C
Nhiệt độ trung bình: 30.7°C
Ngày nóng nhất: ngày thứ 19
=== HÃY TIẾP TỤC CÁC YÊU CẦU KHÁC! ===
Số ngày quan sát: 30
=== NHIỆM VỤ PHÂN TÍCH ===
1. Tìm nhiệt độ cao nhất và thấp nhất
2. Tính nhiệt độ trung bình của tháng
3. Đếm số ngày có nhiệt độ > 30 độ
4. Tìm 10 ngày đầu và 10 ngày cuối của tháng
5. Tính nhiệt độ TB của từng tuần (7 ngày)
6. Tìm những ngày có nhiệt độ bằng nhiệt độ TB
=== DEMO GIAI ===
Nhiệt độ cao nhất: 35°C
Nhiệt độ thấp nhất: 27°C
Nhiệt độ trung bình: 30.7°C
Ngày nóng nhất: ngày thứ 19
=== HÃY TIẾP TỤC CÁC YÊU CẦU KHÁC! ===
```

13. Mini Quiz - Kiểm tra hiểu biết (15 phút)

Câu hỏi trắc nghiệm

```
In [23]: # Mini Quiz về List
         print("=== MINI QUIZ: LIST CO BAN ===")
         print("\nCâu 1: Cho list: numbers = [10, 20, 30, 40, 50]")
         print("Kết quả của numbers[1:4] là gì?")
         print("A) [10, 20, 30, 40]")
         print("B) [20, 30, 40]")
         print("C) [20, 30, 40, 50]")
         print("D) [10, 20, 30]")
         print("\nCâu 2: Cách nào sau đây để tạo list chứa 5 số 0?")
         print("A) [0, 0, 0, 0, 0]")
         print("B) [0] * 5")
         print("C) list([0, 0, 0, 0, 0])")
         print("D) Tất cả đều đúng")
         print("\nCâu 3: Cho list: data = ['a', 'b', 'c', 'd']")
         print("Để lấy phần tử cuối cùng, ta dùng:")
         print("A) data[4]")
         print("B) data[-1]")
         print("C) data[len(data)]")
         print("D) data[3]")
         # Đáp án sẽ được giải thích sau
         print("\n=> Thảo luận nhóm 5 phút rồi cùng giải!")
        === MINI QUIZ: LIST CO BAN ===
        Câu 1: Cho list: numbers = [10, 20, 30, 40, 50]
        Kết quả của numbers[1:4] là gì?
        A) [10, 20, 30, 40]
        B) [20, 30, 40]
        C) [20, 30, 40, 50]
        D) [10, 20, 30]
        Câu 2: Cách nào sau đây để tạo list chứa 5 số 0?
        A) [0, 0, 0, 0, 0]
        B) [0] * 5
        C) list([0, 0, 0, 0, 0])
        D) Tất cả đều đúng
        Câu 3: Cho list: data = ['a', 'b', 'c', 'd']
        Để lấy phần tử cuối cùng, ta dùng:
        A) data[4]
        B) data[-1]
        C) data[len(data)]
        D) data[3]
        => Thảo luận nhóm 5 phút rồi cùng giải!
```

Đáp án và giải thích

```
In [24]: print("=== ĐÁP ÁN VÀ GIẢI THÍCH ===")
# Câu 1
```

numbers = [10, 20, 30, 40, 50]

```
result1 = numbers[1:4]
 print(f"Câu 1: {result1}")
 print("Đáp án: B) [20, 30, 40]")
 print("Giải thích: Slicing [1:4] lấy từ index 1 đến 3 (không bao gồm 4)")
 # Câu 2
 method1 = [0, 0, 0, 0, 0]
 method2 = [0] * 5
 method3 = list([0, 0, 0, 0, 0])
 print(f"\nCâu 2: Tất cả đều tạo ra: {method2}")
 print("Đáp án: D) Tất cả đều đúng")
 print("Giải thích: Ba cách đều tạo ra cùng kết quả")
 # Câu 3
 data = ['a', 'b', 'c', 'd']
 print(f"\nCâu 3: data[-1] = {data[-1]}")
 print("Đáp án: B) data[-1]")
 print("Giải thích: Index âm -1 luôn trỏ đến phần tử cuối")
 print(f"(data[4] se loi IndexError vi chi co index 0-3)")
=== ĐÁP ÁN VÀ GIẢI THÍCH ===
Câu 1: [20, 30, 40]
Đáp án: B) [20, 30, 40]
Giải thích: Slicing [1:4] lấy từ index 1 đến 3 (không bao gồm 4)
Câu 2: Tất cả đều tạo ra: [0, 0, 0, 0, 0]
Đáp án: D) Tất cả đều đúng
Giải thích: Ba cách đều tạo ra cùng kết quả
Câu 3: data[-1] = d
Đáp án: B) data[-1]
Giải thích: Index âm -1 luôn trỏ đến phần tử cuối
(data[4] sẽ lỗi IndexError vì chỉ có index 0-3)
Câu 1: [20, 30, 40]
Đáp án: B) [20, 30, 40]
Giải thích: Slicing [1:4] lấy từ index 1 đến 3 (không bao gồm 4)
Câu 2: Tất cả đều tạo ra: [0, 0, 0, 0, 0]
Đáp án: D) Tất cả đều đúng
Giải thích: Ba cách đều tạo ra cùng kết quả
Câu 3: data[-1] = d
Đáp án: B) data[-1]
Giải thích: Index âm -1 luôn trỏ đến phần tử cuối
(data[4] sẽ lỗi IndexError vì chỉ có index 0-3)
 https://play.kahoot.it/v2/lobby?quizId=c2351c83-848b-4d11-9ab6-91911574bd79
```

14. Tổng kết và chuẩn bị buổi sau

Kiến thức đã học trong 3 tiếng

```
In [25]: print("=== TONG KÉT BUOI HOC ===")
         print("\n☑ ĐÃ HỌC:")
         print("1. Khái niệm List - cấu trúc dữ liệu có thứ tự, có thể thay đổi")
         print("2. Cách tao List: [], list(), range()")
         print("3. Indexing: truy cập phần tử bằng chỉ mục [0, 1, 2...] và [-1, -2...]")
         print("4. Slicing: cắt lát list với [start:stop:step]")
         print("5. Thao tác cơ bản: nối (+), lặp (*), kiểm tra (in), độ dài (len)")
         print("6. Ham built-in: max(), min(), sum(), sorted()")
         print("7. Ứng dụng thực tế: quản lý dữ liệu, phân tích thống kê")
         print("8. Debug: xử lý IndexError, slicing an toàn")
         print("\n♂ MỤC TIÊU ĐẠT ĐƯỢC:")
         print("- Hiểu bản chất của List trong Python")
         print("- Áp dụng được vào bài toán thực tế")
         print("- Tránh được các lỗi cơ bản")
         print("\n ■ BUỔI SAU SẼ HỌC:")
         print("- List methods: append(), remove(), pop(), insert()")
         print("- List comprehension co ban")
         print("- Nested lists (list trong list)")
         print("- Sorting và filtering nâng cao")
         print("\n ♠ BÀI TẬP VỀ NHÀ:")
         print("1. Hoàn thành bài tập 1 và 2 chưa làm xong")
         print("2. Tự tạo 1 ứng dụng quản lý danh sách (VD: sách, phim, nhạc...)")
         print("3. Luyên tập slicing với các pattern khác nhau")
         print("4. Đọc trước về List methods")
```

=== TỔNG KẾT BUỔI HỌC ===

☑ ĐÃ HỌC:

- 1. Khái niệm List cấu trúc dữ liệu có thứ tự, có thể thay đổi
- 2. Cách tạo List: [], list(), range()
- 3. Indexing: truy cập phần tử bằng chỉ mục [0, 1, 2...] và [-1, -2...]
- 4. Slicing: cắt lát list với [start:stop:step]
- 5. Thao tác cơ bản: nối (+), lặp (*), kiểm tra (in), độ dài (len)
- 6. Ham built-in: max(), min(), sum(), sorted()
- 7. Ứng dụng thực tế: quản lý dữ liệu, phân tích thống kê
- 8. Debug: xử lý IndexError, slicing an toàn

- Hiểu bản chất của List trong Python
- Áp dụng được vào bài toán thực tế
- Tránh được các lỗi cơ bản

BUỔI SAU SẼ HỌC:

- List methods: append(), remove(), pop(), insert()
- List comprehension cơ bản
- Nested lists (list trong list)
- Sorting và filtering nâng cao

♠ BÀI TẬP VỀ NHÀ:

- 1. Hoàn thành bài tập 1 và 2 chưa làm xong
- 2. Tự tạo 1 ứng dụng quản lý danh sách (VD: sách, phim, nhạc...)
- 3. Luyện tập slicing với các pattern khác nhau
- 4. Đọc trước về List methods