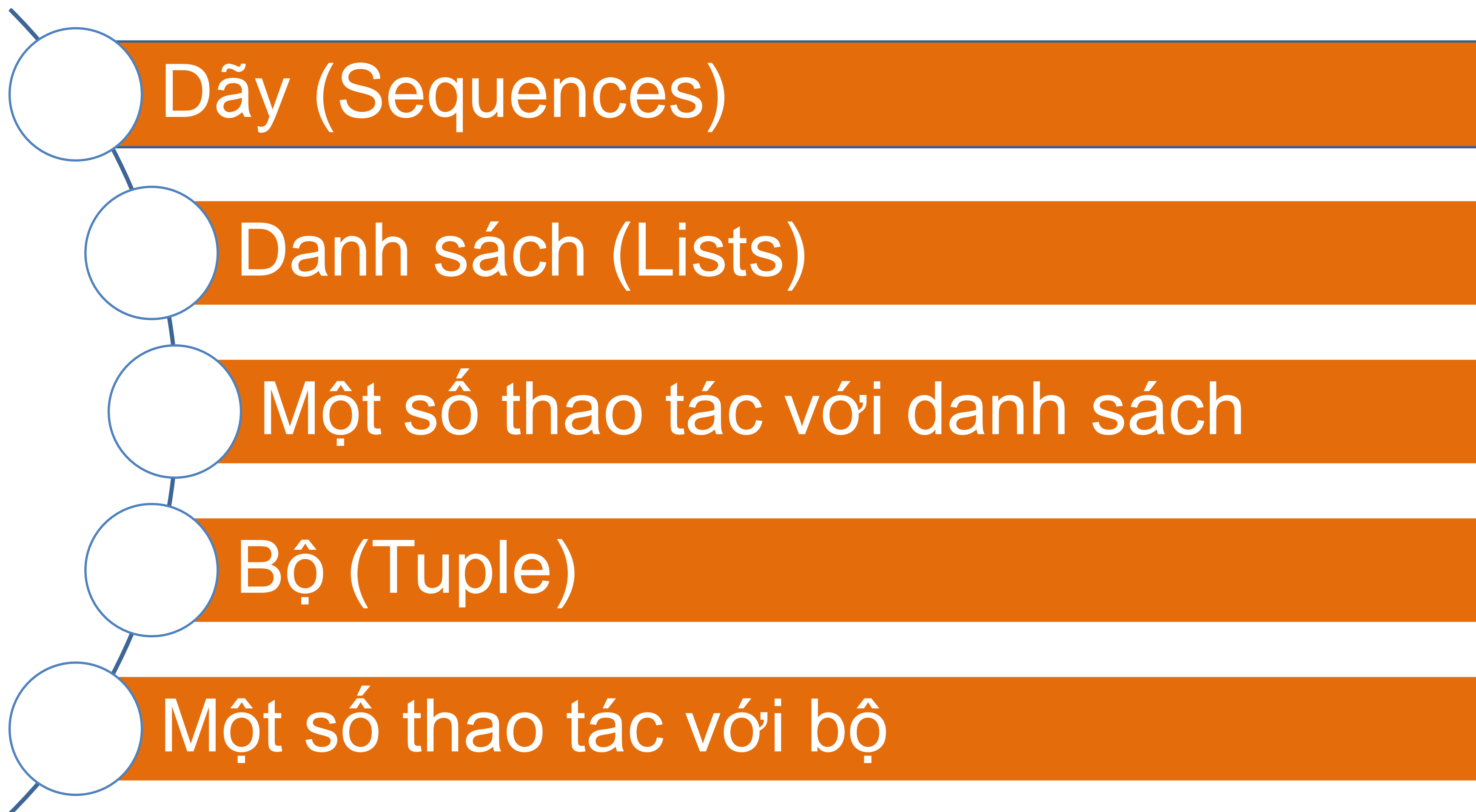




LẬP TRÌNH PYTHON

Chương 7: Lists and Tuples

Giảng viên: Đặng Hải Phong



Dãy (Sequences) là gì?

- Dãy là một tập hợp các phần tử được lưu trữ theo thứ tự, mỗi phần tử đều có một chỉ mục xác định.
- Gồm các kiểu dữ liệu sau:
 - Chuỗi (String): Chuỗi các ký tự
 - Danh sách (List) : Tập hợp các phần tử có thể thay đổi (mutable)
 - Bộ giá trị (Tuple) : Tập hợp các phần tử không thể thay đổi (immutable)

Đặc điểm của Dãy

- Có thứ tự: Các phần tử được sắp xếp theo một thứ tự xác định và có thể truy cập qua chỉ mục.
- Chỉ mục: Dãy có chỉ mục bắt đầu từ 0, cho phép truy cập phần tử cụ thể.
- Cắt (Slicing): Có thể trích xuất một phần dãy bằng cú pháp cắt (slicing).

Giới thiệu về Danh sách

- Định nghĩa danh sách: Một bộ sưu tập có thứ tự, có thể thay đổi
- Cách tạo danh sách trong Python
- Chỉ mục danh sách và cách truy cập phần tử
- Ví dụ: `my_list = [1, 2, 3, 4]`

Cắt Danh sách

- Khái niệm cắt: Trích xuất các phần của danh sách
- Cú pháp: `list[start:end]`
- Chỉ mục âm và cắt danh sách
- Ví dụ: `my_list[1:3]` # Kết quả: `[2, 3]`

Tìm Phần tử với Toán tử in

- Kiểm tra phần tử có trong danh sách bằng in
- Ví dụ:
 - fruits = ['banana', 'apple', 'tomato']
 - 'apple' in fruits
- Trả về True hoặc False

Phương thức và Hàm cho Danh sách

- Các phương thức danh sách phổ biến: *.append()*, *.remove()*, *.sort()*, *.reverse()*
- Các hàm có sẵn: *len()*, *max()*, *min()*, *sum()*
- Ví dụ: `my_list.append(5)`

Sao chép Danh sách

- Sử dụng phương thức `copy()`
- Tránh những sai lầm phổ biến khi sao chép danh sách
- Ví dụ: `list_copy = original_list.copy()`

Xử lý Danh sách

- Duyệt qua danh sách bằng vòng lặp for
- Thay đổi các phần tử trong danh sách
- Ví dụ:

```
for item in my_list:  
    print(item)
```

Biểu thức Danh sách

- Định nghĩa và cú pháp của biểu thức danh sách
- Sử dụng biểu thức danh sách để viết mã ngắn gọn
- Ví dụ: `[x**2 for x in range(5)]` # Kết quả: `[0, 1, 4, 9, 16]`

Danh sách Hai chiều

- Định nghĩa danh sách hai chiều (danh sách của các danh sách)
- Truy cập phần tử trong danh sách hai chiều
- Ví dụ:

```
matrix = [[1, 2], [3, 4], [5, 6]]
```

```
print(matrix[1][0]) # Kết quả: 3
```

Bộ giá trị - Tuple

- Định nghĩa về bộ giá trị: Một tập hợp có thứ tự và không thể thay đổi
- Cú pháp tạo bộ giá trị
- Khi nào nên sử dụng bộ giá trị so với danh sách
- Ví dụ: `my_tuple = (1, 2, 3)`

Một số thao tác với Tuple

- Truy cập Phần tử trong Tuple:
 - Phần tử trong Tuple được truy cập bằng chỉ mục, bắt đầu từ 0.
 - Ví dụ:

```
my_tuple = ('apple', 'banana', 'cherry')  
print(my_tuple[1]) # Kết quả: 'banana'
```


Một số thao tác với Tuple

- Cắt Tuple (Tuple Slicing):
 - Giống như danh sách, Tuple có thể được cắt để lấy một phần của dữ liệu.
 - Cú pháp: `tuple[start:end]`
 - Ví dụ:

```
my_tuple = (10, 20, 30, 40, 50)  
print(my_tuple[1:4]) # Kết quả: (20, 30, 40)
```

Một số thao tác với Tuple

- Kiểm tra sự tồn tại của Phần tử trong Tuple
 - Dùng toán tử in để kiểm tra phần tử có tồn tại trong Tuple hay không.
 - Ví dụ:

```
my_tuple = (1, 2, 3, 4, 5)
```

```
print(3 in my_tuple) # Kết quả: True
```

Một số thao tác với Tuple

- Nối Tuple (Concatenation):
 - Có thể nối hai Tuple lại với nhau bằng toán tử +.
 - Ví dụ:

```
tuple1 = (1, 2)
tuple2 = (3, 4)
combined_tuple = tuple1 + tuple2
print(combined_tuple) # Kết quả: (1, 2, 3, 4)
```

Một số thao tác với Tuple

- Nhân Tuple (Tuple Repetition):
 - Nhân một Tuple với một số nguyên để lặp lại các phần tử của Tuple.
 - Ví dụ:

```
my_tuple = ('A', 'B')  
repeated_tuple = my_tuple * 3  
print(repeated_tuple) # Kết quả: ('A', 'B', 'A', 'B', 'A', 'B')
```

Một số thao tác với Tuple

- Đếm Phần tử trong Tuple (Count):
 - Dùng phương thức `.count()` để đếm số lần xuất hiện của một phần tử trong Tuple.
 - Ví dụ:

```
my_tuple = (1, 2, 2, 3, 4, 2)
```

```
print(my_tuple.count(2)) # Kết quả: 3
```

Một số thao tác với Tuple

- Tìm chỉ số Phần tử trong Tuple (Index):
 - Phương thức `.index()` trả về vị trí của phần tử đầu tiên khớp với giá trị tìm kiếm.
 - Ví dụ:

```
my_tuple = (10, 20, 30, 40)  
print(my_tuple.index(30)) # Kết quả: 2
```


Lợi ích của Tuple

- **Hiệu suất:** Tuple nhanh hơn danh sách vì không thể thay đổi.
- **Bảo mật:** Tuple có thể giúp bảo vệ dữ liệu không bị thay đổi trong quá trình thực thi.
- **Dùng làm khóa trong từ điển:** Tuple có thể được sử dụng làm khóa vì không thể thay đổi.

Chuyển Tuple thành Danh sách

- Dùng hàm `list()` để chuyển một Tuple thành một danh sách.
- Ví dụ:

```
my_tuple = (1, 2, 3)  
my_list = list(my_tuple)  
print(my_list) # Kết quả: [1, 2, 3]
```

Chuyển Danh sách thành Tuple

- Dùng hàm `tuple()` để chuyển một danh sách thành Tuple.
- Ví dụ:

```
my_list = [1, 2, 3]
```

```
my_tuple = tuple(my_list)
```

```
print(my_tuple) # Kết quả: (1, 2, 3)
```

Câu hỏi Ôn tập

- Tuple khác gì so với danh sách (List)?
- Làm thế nào để nối hai Tuple lại với nhau?
- Tại sao Tuple lại hữu ích hơn trong một số tình huống nhất định?

Một số ví dụ minh họa

- Thực hiện và giải thích các ví dụ sau
- `Chuongtrinh7-1.py` → `Chuongtrinh7-28.py`

HỎI ĐÁP





Trân trọng cảm ơn!