



CHƯƠNG 3: DANH SÁCH (LIST)

VŨ HOÀI THU'
CNTT1. PTIT

NỘI DUNG

- ☐ Định nghĩa danh sách
- ☐ Thay đổi, thêm và xóa các phần tử
- ☐ Tổ chức danh sách
- ☐ Tránh lỗi chỉ mục
- ☐ Lặp qua toàn bộ danh sách
- ☐ Lập danh sách số
- ☐ Làm việc với một phần của danh sách
- ☐ Tuples

ĐỊNH NGHĨA DANH SÁCH

- ❑ Trong Python, dấu ngoặc vuông ([]) chỉ định một danh sách và các phần tử trong danh sách được phân tách bởi dấu phẩy (,)
- ❑ Ví dụ: Một danh sách chứa các loại xe đạp khác nhau.

```
bicycles = ['trek', 'cannondale', 'redline', 'specialized']  
print(bicycles)
```



```
['trek', 'cannondale', 'redline', 'specialized']
```

TRUY CẬP VÀO CÁC PHẦN TỬ TRONG DANH SÁCH

- ❑ Để truy cập một phần tử trong danh sách, hãy viết tên của danh sách, theo sau là các chỉ mục được đặt trong dấu ngoặc vuông.
- ❑ Ví dụ: Hãy lấy ra chiếc xe đạp đầu tiên trong danh sách bicycles

```
bicycles = ['trek', 'cannondale', 'redline', 'specialized']  
print(bicycles[0])
```

→ trek

- ❑ Python chỉ trả về phần tử đó mà không có dấu ngoặc vuông.

ĐÁNH CHỈ MỤC

- ❑ Chỉ mục của một danh sách bắt đầu từ 0

Index	0	1	2	3
bicycles	trek	cannondale	redline	specialized
Reversed index	-4	-3	-2	-1

- ❑ Để truy cập phần tử thứ tư trong danh sách, ta yêu cầu phần tử có chỉ

mục 3

`print(bicycles[3])` → specialized

- ❑ Để truy cập phần tử cuối cùng trong danh sách, ta dùng chỉ mục -1

`print(bicycles[-1])` → specialized

SỬ DỤNG GIÁ TRỊ RIÊNG LẺ TỪ DANH SÁCH

- ❑ Có thể sử dụng các giá trị riêng lẻ từ một danh sách giống như cách ta làm với bất kỳ biến nào khác.
- ❑ Ví dụ: Ta có thể sử dụng f-string để tạo ra một thông báo dựa trên giá trị từ một danh sách

```
bicycles = ['trek', 'cannondale', 'redline', 'specialized']  
message = f"My first bicycle was a {bicycles[0].title()}."  
print(message)
```



My first bicycle was a Trek.

THAY ĐỔI, THÊM VÀ XÓA CÁC PHẦN TỬ

- ❑ Hầu hết các danh sách sẽ là động, nghĩa là ta sẽ xây dựng một danh sách và sau đó thêm và xóa các phần tử khỏi nó khi chương trình chạy.
- ❑ Ví dụ, ta có thể tạo một trò chơi, trong đó người chơi phải bắn hạ quái vật trên bầu trời. Ta có thể lưu trữ nhóm người ngoài hành tinh ban đầu trong một danh sách và sau đó xóa một người ngoài hành tinh trong danh sách mỗi khi quái vật bị bắn hạ.

THAY ĐỔI PHẦN TỬ TRONG DANH SÁCH

- ❑ Để thay đổi một phần tử, sử dụng tên của danh sách, theo sau bằng chỉ mục của phần tử mong muốn thay đổi, sau đó gán giá trị.

```
motorcycles = ['honda', 'yamaha', 'suzuki']  
print(motorcycles)
```

→ ['honda', 'yamaha', 'suzuki']

```
motorcycles[0] = 'ducati'  
print(motorcycles)
```

→ ['ducati', 'yamaha', 'suzuki']

THÊM PHẦN TỬ VÀO DANH SÁCH

- ❑ Để thêm phần tử vào cuối danh sách, sử dụng phương thức `append()`

```
motorcycles = ['honda', 'yamaha', 'suzuki']  
motorcycles.append('ducati')  
print(motorcycles)
```

→ ['honda', 'yamaha', 'suzuki', 'ducati']

- ❑ Dùng `append` để tạo một danh sách động

```
motorcycles = []  
motorcycles.append('honda')  
motorcycles.append('yamaha')  
motorcycles.append('suzuki')  
print(motorcycles)
```

→ ['honda', 'yamaha', 'suzuki']

THÊM PHẦN TỬ VÀO DANH SÁCH (t)

- ❑ Thêm một phần tử mới ở bất kỳ vị trí nào trong danh sách bằng cách sử dụng phương thức insert()

```
motorcycles = ['honda', 'yamaha', 'suzuki']  
motorcycles.insert(0, 'ducati')  
print(motorcycles)
```

→ ['ducati', 'honda', 'yamaha', 'suzuki']

- ❑ Phương thức insert() tạo ra một khoảng trống tại vị trí 0 và lưu giá trị “ducati” vào vị trí đó. Hành động này dịch chuyển tất cả các phần tử còn lại một vị trí sang phải.

XÓA PHẦN TỬ KHỎI MỘT DANH SÁCH

- ❑ Xóa phần tử sử dụng lệnh `del` (Khi biết vị trí của các phần tử trong danh sách)

```
motorcycles = ['honda', 'yamaha', 'suzuki']  
  
print(motorcycles)  
  
del motorcycles[0]  
  
print(motorcycles)
```

→ `['honda', 'yamaha', 'suzuki']`
`['yamaha', 'suzuki']`

- ❑ Trong ví dụ trên, chúng ta không thể truy cập vào phần tử bị xóa khỏi danh sách sau khi sử dụng lệnh `del`.

XÓA PHẦN TỬ SỬ DỤNG PHƯƠNG THỨC POP

- ❑ Phương thức pop() loại bỏ phần tử cuối cùng trong danh sách, nhưng cho phép làm việc với phần tử đó sau khi loại bỏ nó.

```
motorcycles = ['honda', 'yamaha', 'suzuki']  
print(motorcycles)  
popped_motorcycle = motorcycles.pop()  
print(motorcycles)  
print(popped_motorcycle)
```



```
['honda', 'yamaha', 'suzuki']  
['honda', 'yamaha']  
suzuki
```

```
motorcycles = ['honda', 'yamaha', 'suzuki']  
last_owned = motorcycles.pop()  
print(f"The last motorcycle I owned was a {last_owned.title()}.")
```



The last motorcycle I owned was a Suzuki.

XÓA PHẦN TỬ TỪ BẤT KỲ VỊ TRÍ NÀO TRONG DANH SÁCH

- ❑ Sử dụng `pop()` để xóa một mục khỏi bất kỳ vị trí nào trong danh sách bằng cách bao gồm chỉ mục của mục mà ta muốn xóa:

```
motorcycles = ['honda', 'yamaha', 'suzuki']  
  
first_owned = motorcycles.pop(0)  
  
print(f"The first motorcycle I owned was a {first_owned.title()}.")
```



The first motorcycle I owned was a Honda.

- ❑ Khi muốn xóa một mục khỏi danh sách và không sử dụng mục đó theo bất kỳ cách nào, hãy sử dụng lệnh `del`. Nếu muốn sử dụng phần tử khi xóa nó, hãy sử dụng phương thức `pop()`

XÓA PHẦN TỬ BẰNG GIÁ TRỊ

- ❑ Nếu ta chỉ biết giá trị của phần tử muốn xóa, có thể sử dụng phương thức `remove()`

```
motorcycles = ['honda', 'yamaha', 'suzuki', 'ducati']  
print(motorcycles)  
motorcycles.remove('ducati')  
print(motorcycles)
```



```
['honda', 'yamaha', 'suzuki', 'ducati']  
['honda', 'yamaha', 'suzuki']
```

BÀI TẬP

- ❑ Bài tập 1: Cho một danh sách bất kỳ. Viết chương trình thực hiện các yêu cầu sau:
 - ✓ Xóa phần tử cuối cùng của danh sách
 - ✓ Thêm một giá trị bất kỳ vào vị trí thứ 4 của danh sách
 - ✓ Thay đổi giá trị của phần tử thứ nhất bằng “Python”

- ❑ Bài tập 2: Cho một danh sách bất kỳ. Viết chương trình hoán đổi phần tử đầu tiên và vị trí cuối cùng của danh sách.

TỔ CHỨC DANH SÁCH

- ❑ Thông thường, danh sách sẽ được tạo theo một thứ tự không thể biết trước, bởi vì không thể kiểm soát thứ tự mà người dùng cung cấp dữ liệu của họ.
- ❑ Vấn đề là ta luôn cần trình bày thông tin theo một trật tự nào đó.
- ❑ Đôi khi, ta cần giữ nguyên thứ tự ban đầu của dữ liệu, đôi khi lại cần thay đổi thứ tự theo lúc đầu.
- ❑ Python cung cấp một số cách khác nhau để thay đổi thứ tự tùy theo tình huống

SẮP XẾP DANH SÁCH VĨNH VIỄN VỚI PHƯƠNG THỨC SORT()

- ❑ Phương thức sort() của Python giúp sắp xếp danh sách:

```
cars = ['bmw', 'audi', 'toyota', 'subaru']  
  
cars.sort()  
  
print(cars)
```

→ ['audi', 'bmw', 'subaru', 'toyota']

- ❑ Sắp xếp danh sách này theo thứ tự bảng chữ cái ngược lại bằng cách chuyển đổi số reverse = True vào phương thức sort()

```
cars = ['bmw', 'audi', 'toyota', 'subaru']  
  
cars.sort(reverse=True)  
  
print(cars)
```

→ ['toyota', 'subaru', 'bmw', 'audi']

SẮP XẾP DANH SÁCH TẠM THỜI VỚI PHƯƠNG THỨC SORTED()

- ❑ Hàm `sorted()` cho phép hiển thị danh sách theo một thứ tự cụ thể nhưng không ảnh hưởng đến thứ tự thực tế của danh sách.

```
cars = ['bmw', 'audi', 'toyota', 'subaru']  
print("Here is the original list:")  
print(cars)  
print("\nHere is the sorted list:")  
print(sorted(cars))
```



Here is the original list:
['bmw', 'audi', 'toyota', 'subaru']

Here is the sorted list:
['audi', 'bmw', 'subaru', 'toyota']

- ❑ Hàm **`sorted()`** cũng có thể chấp nhận đối số **`reverse = True`** nếu ta muốn hiển thị danh sách theo thứ tự bảng chữ cái ngược lại

IN DANH SÁCH THEO THỨ TỰ NGƯỢC

- ❑ Để đảo ngược thứ tự ban đầu của danh sách, ta có thể sử dụng phương thức `reverse()`

```
cars = ['bmw', 'audi', 'toyota', 'subaru']  
  
print(cars)  
  
cars.reverse()  
  
print(cars)
```

→
 ['bmw', 'audi', 'toyota', 'subaru']
 ['subaru', 'toyota', 'audi', 'bmw']

- ❑ Lưu ý rằng, phương thức `reverse()` không sắp xếp theo thứ tự `alphabet()`, nó chỉ đơn thuần đảo ngược thứ tự của các phần tử trong danh sách hiện tại.

TÌM ĐỘ DÀI CỦA DANH SÁCH

- ❑ Tìm độ dài của danh sách bằng cách sử dụng hàm len()

```
cars = ['bmw', 'audi', 'toyota', 'subaru']  
  
print(len(cars))
```

4

- ❑ *Chú ý: Python đếm các phần tử trong danh sách bắt đầu bằng một, vì vậy ta sẽ không gặp phải bất kỳ lỗi nào khi xác định độ dài của danh sách*

TRÁNH LỖI CHỈ MỤC

- ❑ Giả sử danh sách có 3 phần tử, ta yêu cầu in ra phần tử thứ tư, Python sẽ thông báo lỗi chỉ mục

```
motorcycles = ['honda', 'yamaha', 'suzuki']  
print(motorcycles[3])
```

```
-----  
IndexError                                Traceback (most recent call last)  
~\AppData\Local\Temp\ipykernel_1496\3878205628.py in <module>  
      1 motorcycles = ['honda', 'yamaha', 'suzuki']  
----> 2 print(motorcycles[3])  
  
IndexError: list index out of range
```

- ❑ Lỗi chỉ mục có nghĩa là Python không thể kết xuất một mục tại chỉ mục được yêu cầu. Nếu lỗi chỉ mục xảy ra trong chương trình, hãy thử điều chỉnh chỉ mục đang được yêu cầu một đơn vị.

TRÁNH LỖI CHỈ MỤC

- ❑ Bất cứ khi nào ta muốn truy cập phần tử cuối cùng trong danh sách, hãy sử dụng chỉ mục (-1)

```
1 motorcycles = ['honda', 'yamaha', 'suzuki']  
2 print(motorcycles[-1])
```

suzuki

- ❑ Tiếp cận này chỉ xảy ra lỗi trong trường hợp duy nhất đó là khi danh sách trống

```
1 motorcycles = []  
2 print(motorcycles[-1])
```

IndexError Traceback (most recent call last)

~\AppData\Local\Temp\ipykernel_1496\3288811120.py in <module>

```
1 motorcycles = []  
----> 2 print(motorcycles[-1])
```

IndexError: list index out of range

BÀI TẬP

- ❑ Bài tập 3: Cho một danh sách gồm các số. Viết chương trình xóa các phần tử chia hết cho 2 trong danh sách.
- ❑ Bài tập 4: Cho một danh sách gồm các phần tử có giá trị bất kỳ. Tính tổng và trung bình cộng của các phần tử có giá trị là các số có trong danh sách.

BÀI TẬP(t)

- ❑ Bài tập 5: Cho một danh sách gồm các số. Tính bình phương các phần tử trong danh sách và sắp xếp theo thứ tự giảm dần..
- ❑ Bài tập 6: Cho một danh sách bất kỳ. Tính số lượng các giá trị duy nhất có trong danh sách.

Ví dụ: array = [1, "a", 34, "a", "b", 1, "c"]

=> Kết quả: 5