

Học xong bài này, em sẽ:

- ✓ Nhận biết được sơ lược cấu trúc của kiểu dữ liệu mảng trong các ngôn ngữ lập trình bậc cao.
- ✓ Mô tả được kiểu danh sách trong Python có cấu trúc giống như kiểu mảng.
- ✓ Viết được câu lệnh trong Python để khởi tạo và truy cập tới các phần tử của danh sách.
- ✓ Sử dụng được một số hàm xử lý danh sách thường dùng.



Có nhiều bài toán thực tế cần giải quyết mà trong đó dữ liệu có được ở dạng một bản liệt kê tuần tự (thường gọi là danh sách). Ví dụ: Từ danh sách kết quả một cuộc thi, hãy đưa ra danh sách những người đỗ trong kì thi đó. Em hãy đưa thêm ví dụ.

1 Kiểu dữ liệu danh sách

Nhiều khi chúng ta cần lưu trữ nhiều phần tử dữ liệu cùng với nhau thành một dãy mà trong dãy đó thứ tự của mỗi phần tử dữ liệu là quan trọng. Với những dãy dữ liệu như thế ta có thể truy cập, xem hoặc thay đổi được một phần tử của dãy khi biết vị trí của nó trong dãy.

Bảng 1. Nhiệt độ cao nhất ở Hà Nội của các ngày trong tuần

Ngày	Thứ Hai	Thứ Ba	Thứ Tư	Thứ Năm	Thứ Sáu	Thứ Bảy	Chủ nhật
Nhiệt độ °C	38	36	37	36	38	38	37

Nhiều ngôn ngữ lập trình bậc cao cho phép sử dụng kiểu dữ liệu theo cấu trúc như vậy, gọi là kiểu mảng. Thay vì dùng nhiều biến riêng lẻ chứa các đại lượng cùng một kiểu dữ liệu, ta có thể dùng một biến kiểu mảng chứa cả dãy các đại lượng đó.

Trong Python có kiểu dữ liệu danh sách (list) để lưu trữ dãy các đại lượng có thể ở các kiểu dữ liệu khác nhau và cho phép truy cập tới mỗi phần tử của dãy theo vị trí (chỉ số) của phần tử đó. Khi tất cả các phần tử trong danh sách đều có cùng một kiểu dữ liệu thì danh sách đó tương ứng với mảng ở nhiều ngôn ngữ lập trình bậc cao khác. Các phần tử trong danh sách của Python được đánh chỉ số bắt đầu từ 0.

Ví dụ 1. Thay vì dùng sáu biến kiểu kí tự để lưu trữ tên sáu bạn (*Hình 1a*), có thể dùng một biến kiểu danh sách (*Hình 1b*).

```
>>> friend1 = 'Anh Hồng'  
>>> friend2 = 'Minh Hằng'  
>>> friend3 = 'Tuyết Nga'  
>>> friend4 = 'Tuân Thành'  
>>> friend5 = 'Anh Quân'  
>>> friend6 = 'Thùy Anh'
```

```
>>> friends = ['Anh Hồng', 'Minh Hằng',  
'Tuyết Nga', 'Tuân Thành', 'Anh Quân',  
'Thùy Anh']  
>>> type(friends)  
<class 'list'>
```

Các phần tử trong danh sách phải cách nhau bởi dấu “,”

Hình 1a. Dùng sáu biến để lưu trữ tên sáu bạn

Hình 1b. Dùng một biến kiểu danh sách



1

- 1) Với gợi ý từ *Ví dụ 1*, em hãy viết câu lệnh Python để tạo ra một biến kiểu danh sách lưu trữ được dữ liệu cho ở *Bảng 1*.
- 2) Viết câu lệnh in ra phần tử thứ ba của danh sách được tạo ở yêu cầu 1.
- 3) Dùng hàm `type()` kiểm tra lại kiểu dữ liệu của biến vừa tạo ra.
- 4) Dùng hàm `len()` để biết kích thước của danh sách (độ dài hay số phần tử của danh sách).

Khởi tạo danh sách

Có nhiều cách khởi tạo danh sách, ba cách trong số các cách đó là:

- Dùng phép gán, ví dụ: `ds = [1, 1, 2, 3, 5, 8]`
- Dùng câu lệnh lặp `for` gán giá trị trong khoảng cho trước, ví dụ:
`ds = [i for i in range(6)]`
Kết quả: `ds = [0, 1, 2, 3, 4, 5]`
- Khởi tạo danh sách số nguyên hay thực từ dữ liệu nhập vào:

```
a = [int(i) for i in input().split()]
```

Câu lệnh cho phép nhập một dãy số nguyên trên cùng một dòng

```
File Edit Format Run Options Window Help  
  
print ("Nhập một danh sách gồm các số nguyên")  
a = [int(i) for i in input().split()]  
print (a)
```

```
File Edit Shell Debug Options Window Help  
  
Nhập một danh sách gồm các số nguyên  
12 24 9 11 7  
[12, 24, 9, 11, 7]  
>>>
```

Hình 2a. Một chương trình nhập danh sách các số nguyên và in ra danh sách đó

Hình 2b. Kết quả chạy chương trình ở Hình 2a khi nhập năm số nguyên, các số cách nhau một hay một số dấu cách

Truy cập đến phần tử trong danh sách

Nếu tên danh sách và chỉ số của phần tử, chỉ số cần đặt trong cặp dấu ngoặc vuông. Chỉ số có thể là một biểu thức số học.

Trong *Ví dụ 1*, với danh sách friends, friends[5] là phần tử thứ ba trong danh sách và có giá trị là 'Thuý Anh' (*Hình 3*).

```
>>> friends [0] ← Yêu cầu cho biết phần tử đầu tiên của danh sách friends  
'Anh Hồng'  
>>> friends [5] ← Yêu cầu cho biết phần tử ở vị trí thứ sáu của danh sách friends  
'Thuý Anh'
```

Hình 3. Xem một số phần tử của danh sách

② Một số hàm và thao tác xử lý danh sách



2

Hãy hình dung, nhóm em dùng một danh sách trong Python để lưu trữ và quản lý danh sách các bạn trong Câu lạc bộ Lập trình của lớp em. Trong tình huống ấy, nhóm em mong muốn Python cung cấp sẵn những công cụ nào ở dạng hàm để dễ thực hiện được việc quản lý danh sách câu lạc bộ?

Bảng 2 sau đây giới thiệu một số hàm Python cung cấp để người lập trình xử lý danh sách nhanh chóng, thuận lợi. Ngoài ra, còn có nhiều hàm khác nữa có thể dùng trong xử lý danh sách mà người lập trình có thể dễ dàng tra cứu và tìm hiểu.

Bảng 2. Một số hàm xử lý danh sách trong Python

Hàm xử lý danh sách	Ý nghĩa
<code>a.append(x)</code>	Bổ sung phần tử x vào cuối danh sách a.
<code>a.pop(i)</code>	Xoá phần tử đứng ở vị trí i trong danh sách a và đưa ra phần tử này.
<code>a.insert(i,x)</code>	Bổ sung phần tử x vào trước phần tử đứng ở vị trí i trong danh sách a. <code>a.insert(0,x)</code> sẽ bổ sung x vào đầu danh sách.
<code>a.sort()</code>	Sắp xếp các phần tử của danh sách a theo thứ tự không giảm.

Ví dụ 2. Hình 3 minh họa chương trình Python sử dụng một số hàm để xử lý danh sách.

```
File Edit Shell Debug Options Window Help
>>> friends = ['Mai', 'Minh', 'Nga', 'Anh', 'Giang', 'Lan']
>>> friends.append('Hoa')
>>> friends[6]
'Hoa'
>>> friends.pop(2)
'Nga'
>>> friends.insert(0, 'Phan')
>>> friends[0]
'Phan'
>>> friends.sort()
>>> print(friends)
['Anh', 'Giang', 'Hoa', 'Lan', 'Mai', 'Minh', 'Phan']
```

Hình 3. Một chương trình xử lý danh sách

Ghép các danh sách thành một danh sách

Phép “+” được dùng để ghép nối hai danh sách.

Ví dụ 3. Chương trình ở *Hình 4* thực hiện ghép hai danh sách.

```
File Edit Shell Debug Options Window Help
>>> a = [1, 2, 3]
>>> b = ["Hồng", "Cúc", "Lan", "Mai"]
>>> c = a + b
>>> print(c)
[1, 2, 3, 'Hồng', 'Cúc', 'Lan', 'Mai']
```

Hình 4. Chương trình ghép nối hai danh sách

Duyệt các phần tử trong danh sách theo thứ tự lưu trữ

Gọi a là một danh sách, câu lệnh duyệt danh sách có dạng:

for i in a:
Các câu lệnh xử lý

Ví dụ 4. Hình 5 minh họa chương trình và kết quả duyệt danh sách bằng câu lệnh **for**.

```
File Edit Format Run Options Window Help
a = [4,1,4,2,2,5]
for i in a:
    print(i*i)
```

```
16
1
16
4
4
25
>>>
```

Hình 5. Chương trình duyệt danh sách bằng câu lệnh for



Bài 1. Đọc chương trình sau đây và cho biết kết quả in ra màn hình. Em hãy soạn thảo và chạy chương trình để kiểm tra dự đoán của em.

```
File Edit Format Run Options Window Help
ds = [int(i) for i in input().split()]
sonho = 0
for i in ds:
    if i <= 100: sonho = sonho + 1
print(sonho)
```

Bài 2. Bạn Thanh muốn tính trung bình cộng của nhiệt độ trung bình các ngày trong tuần. Thanh đã viết được đoạn chương trình nhập từ bàn phím nhiệt độ trung bình của bảy ngày trong tuần vào một danh sách (*Hình 6*). Em hãy giúp bạn Thanh viết tiếp những câu lệnh còn thiếu vào chỗ trống để máy tính đưa ra màn hình kết quả cần có.

```
File Edit Format Run Options Window Help
nh_d = [float(i) for i in input().split()]
tb = 0
.....
print("Nhiệt độ trung bình:", tb/7)
```

Hình 6. Chương trình tính nhiệt độ trung bình của bảy ngày trong tuần



Camera đặt cạnh trạm thu phí đường cao tốc ghi nhận nhiều thông tin, trong đó có mảng số nhận dạng loại ô tô đi qua. Mỗi loại ô tô được mã hoá thành một số nguyên dương. Cho dãy số, mỗi số là mã hoá về loại của một ô tô đi qua trạm thu phí. Em hãy viết chương trình nhập dãy số mã hoá xe vào từ bàn phím và đưa ra màn hình số loại xe khác nhau đã được nhận dạng.

Ví dụ:

INPUT	OUTPUT
2 4 2 5 4 5 2 5 4 5 5	3



Trong các câu sau đây, những câu nào đúng?

- 1) Trong các ngôn ngữ lập trình bậc cao đều có kiểu dữ liệu để lưu trữ một dãy hữu hạn các phần tử.

- 1) Trong ngôn ngữ lập trình Python, dữ liệu kiểu danh sách là một dãy hữu hạn các phần tử cho phép truy cập đến từng phần tử của nó.
- 2) Python bắt buộc các phần tử của một danh sách phải có cùng một kiểu dữ liệu.
- 3) Phải khởi tạo một danh sách trong Python bằng phép gán trong chương trình, không thể nhập các phần tử của danh sách từ bàn phím.
- 4) Python chỉ cung cấp những hàm sau đây để xử lý danh sách: `append()`, `pop()`, `insert()`, `sort()`, `clear()`.

Tóm tắt bài học

- ✓ Các ngôn ngữ lập trình bậc cao đều cung cấp kiểu dữ liệu cho phép lưu trữ một dãy hữu hạn các phần tử và các chương trình con có sẵn để xử lý dữ liệu thuộc kiểu này.
- ✓ Trong Python, list là kiểu dữ liệu có cấu trúc dùng để nhóm một tập dữ liệu thành một dãy giá trị được đánh số và có thể truy cập đến từng giá trị.
- ✓ Có thể khởi tạo cho list trong Python bằng cách gán trực tiếp hoặc nhập giá trị các phần tử vào từ thiết bị vào chuẩn.
- ✓ Python cung cấp nhiều thao tác hữu dụng trên list, một số hàm thông dụng là: `len()`, `append()`, `pop()`, `insert()`, `sort()`.

BÀI TÌM HIỂU THÊM

NHẬP DANH SÁCH TỪ FILE

Có thể nhập một danh sách từ file như chương trình ở *Hình 7*. Nếu danh sách cần xử lý rất dài thì chúng ta sẽ thấy được rõ tính ưu việt của việc nhập dữ liệu từ file. Ví dụ chương trình dưới đây nhập danh sách các số nguyên từ file input.txt và ghi tổng các số nguyên đó ra file output.txt.

```

File Edit Format Run Options Window Help
import sys
fi = open("input.txt", encoding = "utf-8")
fo = open("output.txt", "w", encoding = "utf-8")
sys.stdin = fi
sys.stdout = fo
a = [int(i) for i in input().split()]
print(sum(a))
fo.close()

```

Hình 7. Minh họa chương trình nhập danh sách từ file