

BÀI 1**Kiểu mảng và cấu trúc mảng**

Học xong bài này, em sẽ:

- ✓ Trình bày được cấu trúc dữ liệu mảng một chiều.
- ✓ Biết và sử dụng được một số hàm có sẵn trong Python để thao tác với biến kiểu mảng.



Khi lập trình Python, nếu cần xử lý một dãy số thì em dùng kiểu dữ liệu gì?

1 Biến mảng và cấu trúc mảng

Xét bài toán phân tích kết quả học tập cuối năm của một lớp, ví dụ: Lớp 11A, có 45 học sinh. Đầu vào là bảng điểm tổng kết của tất cả học sinh trong lớp, có các cột *Họ và tên, Điểm Toán, Điểm Ngữ văn, Điểm Tin học,...* Cần viết chương trình máy tính cho biết các kết quả như: điểm trung bình mỗi môn học, điểm cao nhất từng môn học, họ và tên học sinh đạt được điểm cao nhất đó,...

Mỗi cột điểm môn học là một dãy số thực gồm 45 số. Hầu hết các ngôn ngữ lập trình bậc cao đều có sẵn kiểu dữ liệu mảng (`array`) phù hợp để chứa một dãy số nguyên hay dãy số thực có độ dài định trước.

a) Khai báo mảng một chiều

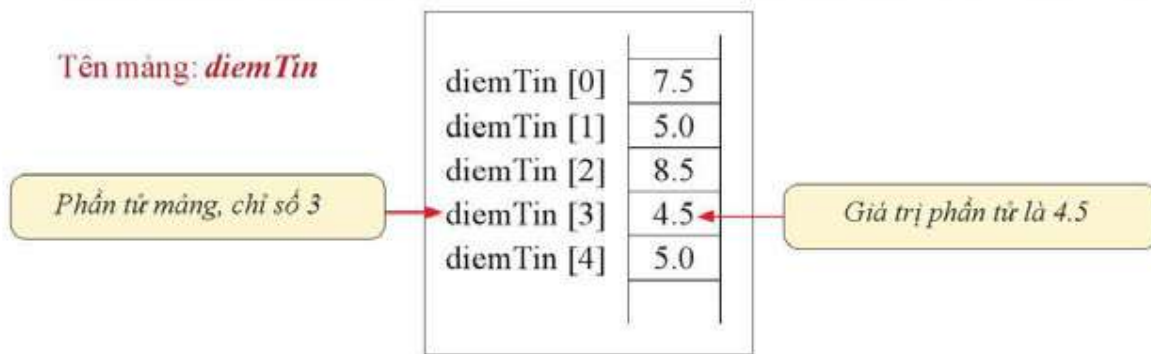
Khai báo tức là cung cấp đủ các thông tin: *tên biến mảng, kiểu dữ liệu, kích thước*. Ví dụ:

- Tên biến mảng: *diemTin*.
- Kiểu dữ liệu của mảng: Số thực.
- Kích thước: 45.

b) Tổ chức mảng một chiều

Trong bộ nhớ, mảng một chiều được lưu trữ thành một khối các ô nhớ liên kế liên tục, có dung lượng bằng tích *kích thước \times độ dài kiểu dữ liệu* (Hình 1). Ví dụ, nếu để lưu trữ một số thực (`float`) cần dùng 32 bit (4 byte) thì mảng *A* gồm 10 phần tử trong bộ nhớ sẽ chiếm 40 byte.

Mảng có kích thước n thì các phần tử mảng được đánh chỉ số tuần tự từ 0 đến $n - 1$.



Hình 1. Phần đầu của mảng *diemTin* (5 phần tử) trong bộ nhớ

Có thể hình dung bộ nhớ RAM là một dây bit rất dài, chia thành nhiều ô nhớ liên nhau. Mỗi ô nhớ được đánh số gọi là địa chỉ truy cập. Tùy theo cách tổ chức bộ nhớ cho chương trình máy tính, một ô nhớ có thể dài 1 byte, 2 byte hay 4 byte.

Một số nguyên (integer) có thể chiếm 1 byte hoặc 2 byte. Một số thực (float hay double) có thể chiếm 4 byte hoặc 8 byte. Để dễ minh họa với mảng số thực, ta coi một ô nhớ chứa vừa đúng một số thực.



Mảng: một cấu trúc dữ liệu gồm các phần tử có cùng kiểu dữ liệu, được lưu trữ thành một khối nhiều ô nhớ liên tiếp trong bộ nhớ.

c) Truy cập ngẫu nhiên

Các thông tin có trong khai báo mảng sẽ được máy tính dùng để xác định độ lớn phần bộ nhớ dành cho một biến mảng. Nó cũng cho phép tìm được vị trí chính xác của từng phần tử trong mảng khi biết chỉ số tương ứng.

Ví dụ, nếu mảng *diemTin* được lưu trữ trong bộ nhớ bắt đầu từ địa chỉ A thì phần tử mảng có chỉ số i là *diemTin*[i] sẽ được lưu trữ tại ô nhớ cách vị trí bắt đầu của mảng đúng i ô nhớ. Địa chỉ ô nhớ chứa *diemTin*[3] sẽ là $A + 3$. Máy tính tìm ngay được địa chỉ này. Nếu mảng rất dài, ví dụ gồm 10 000 phần tử thì máy tính cũng tìm ngay được địa chỉ phần tử chỉ số i bất kì theo cách tính trên.



Thời gian thực hiện hằng số: một câu lệnh máy, một thuật toán, một chương trình được coi là có thời gian thực hiện hằng số nếu thời gian thực hiện đều không vượt quá một hằng số cho trước, không phụ thuộc kích thước dữ liệu đầu vào.

Mảng được sử dụng nhiều vì thời gian truy cập đọc giá trị hay gán giá trị mới cho một phần tử bất kì (đã cho biết chỉ số) là hằng số.

2) Mảng một chiều trong Python

Cú pháp khai báo mảng một chiều trong Python như sau:

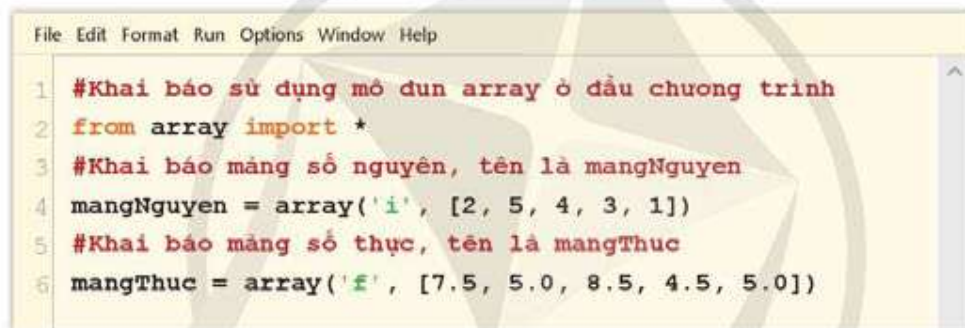
- Khai báo sử dụng mô đun **array** ở đầu chương trình.
- Khai báo biến kiểu mảng theo mẫu dưới đây:

```
mang_1 = array('i', [...])  
mang_2 = array('f', [...])
```

Trong đó:

- Ký tự 'i' là viết tắt của **integer**; ký tự 'f' là viết tắt của **float**.
- Thay cho dấu "..." ở dòng thứ nhất là một danh sách các số nguyên trong **mang_1**.
- Thay cho dấu "..." ở dòng thứ hai là một danh sách các số thực trong **mang_2**.

Hình 2 là một ví dụ khai báo mảng trong Python kèm giải thích câu lệnh.



```
File Edit Format Run Options Window Help  
1 #Khai báo sử dụng mô đun array ở đầu chương trình  
2 from array import *  
3 #Khai báo mảng số nguyên, tên là mangNguyen  
4 mangNguyen = array('i', [2, 5, 4, 3, 1])  
5 #Khai báo mảng số thực, tên là mangThuc  
6 mangThuc = array('f', [7.5, 5.0, 8.5, 4.5, 5.0])
```

Hình 2. Ví dụ khai báo mảng trong Python

Dùng danh sách Python làm mảng

Có thể dùng kiểu danh sách của Python làm mảng một chiều. Không những thế, kiểu danh sách linh hoạt hơn nhiều và có thêm một số hàm (chính xác hơn là các phương thức) mà kiểu mảng không áp dụng được.

Danh sách dùng làm mảng được khai báo và sử dụng như một danh sách Python thông thường.



1

Em hãy khám phá các phép toán cơ sở với mảng trong Python, sao chép lại và chạy thử các câu lệnh ở Hình 3 và Hình 4; thêm dần từng dòng lệnh, sau đó thực hiện các công việc sau:

- 1) Đoán trước kết quả và chạy chương trình để kiểm tra.
- 2) Xem kết quả và cho biết có sự tương tự giữa mảng với danh sách hay không.

```
File Edit Format Run Options Window Help
1 from array import *
2 mangNguyen = array('i', [2, 5, 4, 3, 1])
3 mangThuc = array('f', [7.5, 5.0, 8.5, 4.5, 5.0])
4 dsNguyen = [2, 5, 4, 3, 1]
5 dsThuc = [7.5, 5.0, 8.5, 4.5, 5.0]
6 #Đọc giá trị; gán giá trị cho phần tử mảng và danh sách tương tự nhau
7 print(dsNguyen[1])
8 print(mangNguyen[2])
9 mangThuc[3] = 7.0
10 print(mangThuc [3])
11 print(dsThuc)
12 mangNguyen[3] = 3.5 #Bảo lỗi sai kiểu phần tử
13 dsNguyen[3] = 3.5
14 mangThuc[3] = 8
15 print(mangThuc[3]) #In ra 8.0
```

Hình 3. Ví dụ xử lý mảng một chiều

```
File Edit Format Run Options Window Help
1 #Hàm sorted() để sắp xếp, mặc định theo thứ tự tăng dần
2 #Dãy số kết quả là bản sao khác; dãy số ban đầu giữ nguyên
3 print (sorted(mangThuc))
4 print (mangThuc)
5 #Phương thức sort() không áp dụng được cho mảng, sẽ báo lỗi
6 mangThuc.sort()
7 #Hàm tolist() chuyển mảng thành danh sách, sau đó mới áp dụng sort
8 ds = mangThuc.tolist()
9 ds.sort()
10 print(ds)
```

Hình 4. Minh họa về hàm không áp dụng được cho mảng

Có thể dùng kiểu mảng hay kiểu danh sách của Python để biểu diễn mảng một chiều.

③ Một số hàm gộp và hàm phân tích thống kê

Các hàm gộp max, min, sum đã có sẵn và có thể sử dụng ngay cho kiểu mảng cũng như kiểu danh sách. Python có một số hàm phân tích thống kê áp dụng cho kiểu mảng và kiểu danh sách các số. Để sử dụng các hàm trong mô đun này cần khai báo mô đun **statistics** ở đầu chương trình theo cú pháp như sau:

```
from statistics import *
```



Em hãy khám phá các hàm gộp và hàm phân tích thống kê (tham khảo *Hình 5*).
Thử áp dụng từng hàm với mảng và với danh sách rồi quan sát kết quả.

#Các hàm phân tích thống kê

```
print(mean(mangThuc))
```

```
print(median(dsThuc))
```

```
print(mode(mangThuc))
```

...

| Tên hàm | Chức năng |
|---------|--|
| mean | Trả về trung bình cộng các phần tử |
| median | Trả về trung vị |
| mode | Trả về giá trị xuất hiện nhiều lần nhất trong dãy số |

Hình 5. Một số hàm phân tích thống kê



Câu 1. Hãy kể tên một số hàm Python áp dụng được cho cả danh sách và mảng.

Câu 2. Hãy kể tên một hàm Python áp dụng cho danh sách nhưng không áp dụng cho mảng.



Cho *diemTin* là dãy điểm tổng kết môn Tin học của lớp 11A. Lập trình đưa ra màn hình các phân tích thống kê: điểm cao nhất, điểm thấp nhất, điểm trung bình, trung vị,... và dãy điểm sắp xếp theo thứ tự từ cao xuống thấp.

Gợi ý: Hàm *sorted* có lựa chọn sắp thứ tự giảm dần.

Cú pháp:

```
sorted(biến_mảng, reverse=True)
```



Câu 1. Khai báo mảng là cung cấp những thông tin gì?

Câu 2. Nói “Thời gian thực hiện câu lệnh là hằng số” có nghĩa là gì?

Tóm tắt bài học

- ✓ Mảng là một tập hợp các phần tử có cùng kiểu dữ liệu, được lưu trữ thành một khối nhiều ô nhớ liên kế trong bộ nhớ. Các phần tử mảng được đánh chỉ số tuần tự và có thể truy cập ngẫu nhiên với thời gian hằng số.
- ✓ Trong Python, mảng và danh sách có nhiều điểm tương tự; có thể dùng danh sách thay cho mảng.