

LIST trong Python

Một số vấn đề nâng cao

TS. Trần Quang Quý

e-mail: tqquy@ictu.edu.vn

Contact: 0818981166(zalo)

Trường Đại học Công nghệ Thông tin và
Truyền thông Thái Nguyên

Tháng 09 năm 2021



Khởi tạo danh sách nâng cao

Thông thường để khởi tạo một danh sách trong Python, chúng ta có thể khai báo trực tiếp và nhập từng phần tử cho LIST. Tuy nhiên đối với các phần tử có tính chất giống nhau hoặc lặp lại thì chúng ta nên sử dụng tới vòng lặp for. Ví dụ:

```
list1 = [ ]  
for i in range(100): list1.append(i)  
list2 = [i for i in range(6)]  
list3 = [i*i for i in range(6)]  
list4 = [0 for i in range(6)]  
list5 = [4*i for i in range(6)]
```

Như vậy, chúng ta có cú pháp chung là:

```
listname = [expression for i in dataset]
```



Tạo một list có điều kiện

Ngoài ra chúng ta cũng có thể kết hợp với câu lệnh điều kiện if khi khởi tạo một danh sách thỏa mãn một điều kiện nào đó của đầu bài. Ví dụ:

```
list1 = [i for i in range(6) if i%2 == 0]  
list2 = [2*i+1 for i in range(6) if i%2 == 0]  
list3 = [i*i for i in (1,2,3) if i%2 == 0]
```



Sử dụng vòng lặp for kết hợp

Mỗi phần tử của một danh sách có thể là một danh sách con khác, khi đó để tạo danh sách, chúng ta có thể sử dụng các vòng lặp for lồng nhau. Ví dụ:

```
list1 = [[i,j] for i in (1,2,3) for j in range(2)]
```

```
list2 = [[i,j] for i in (1,2,3) for j in range(2) if  $i + j == 2$ ]
```



Một số phương thức khác

- Phương thức: `danhsach.count(x)` đếm số lần xuất hiện của phần tử `x` có trong danh sách.
- Phương thức: `danhsach.remove(x)` sẽ xóa phần tử `x` xuất hiện đầu tiên có trong danh sách. Nếu trong danh sách không có phần tử `x` thì trình dịch sẽ báo lỗi
- Phương thức: `danhsach.pop()` - xóa phần tử cuối cùng của danh sách.
- Phương thức: `del danhsach[index]` - xóa phần tử có chỉ số là `index` trong list.
- Phương thức: `del danhsach[i:j]` - xóa các phần tử có chỉ số trong đoạn $i, i + 1, \dots, j - 1$ trong danh sách.



Một số phương thức khác

- Phương thức `del danh_sach[i,j,t]` - xóa các phần tử trong danh sách có chỉ số $i, i + t, i + 2t, \dots, i + lt$ với $i + lt < j$
- Phương thức `danh_sach.clear()` - xóa tất cả các phần tử khỏi danh sách, khi đó danh sách trở thành danh sách rỗng (số phần tử bằng 0).
- Phương thức `danh_sach.insert(i,x)` - Chèn phần tử có giá trị là x vào vị trí có chỉ số i trong danh sách.
- Phương thức `danh_sach.sort()` - Sắp xếp danh sách theo thứ tự tăng dần, để sắp xếp danh sách theo thứ tự giảm dần, chúng ta thêm tham số `reverse = True`.
- Phương thức `sorted(list)` - Trả về kết quả là một danh sách mới đã được sắp xếp từ danh sách ban đầu.
- Phương thức `danh_sach.reverse()` - Đảo ngược thứ tự các phần tử có trong danh sách. Đôi khi chúng ta có thể sử dụng câu lệnh `danh_sach[::-1]` để đảo ngược thay cho `reverse`.



Ma trận - Danh sách hai chiều

Trong Python, các phần tử của danh sách có thể là một danh sách con khác. Một ma trận hay một mảng hai chiều được biểu diễn như một danh sách mà mỗi phần tử của nó là một danh sách biểu diễn các phần tử trên cùng một dòng.

Với một ma trận gồm 3 dòng, 3 cột như dưới đây được biểu diễn bằng một danh sách mà các phần tử của nó là:

`[[1,2,3],[2,3,4],[3,4,5]]`

1	2	3
2	3	4
3	4	5



Truy cập vào các phần tử của ma trận

- Các dòng và các cột trong ma trận được đánh chỉ số bắt đầu từ 0. Để truy cập đến một phần tử nào đó trong ma trận, chúng ta thực hiện: `danhsach[indexrow][indexcolumn]`.
- Để thực hiện in các phần tử của ma trận, chúng ta sử dụng 2 vòng lặp for lồng nhau.

Bài tập áp dụng: Viết chương trình trong Python nhập vào một ma trận gồm có m dòng và n cột. In ra ma trận được nhập vào, hiển thị ra màn hình.



Danh sách của danh sách

- Một ma trận hai chiều như trên có thể coi là trường hợp riêng đặc biệt của danh sách, trong đó các phần tử của danh sách là các danh sách khác. Mỗi phần tử trong danh sách là các danh sách mà số phần tử của nó có thể không bằng nhau. Sử dụng hai vòng lặp for để in ra các phần tử của danh sách. Xem ví dụ trên Colabgooogle.
- Python cho phép so sánh hai danh sách bằng cách lần lượt thực hiện so sánh các phần tử có chỉ số 0 đến hết danh sách. Khi gặp hai phần tử tương ứng khác nhau thì danh sách nào chứa phần tử lớn hơn thì danh sách đó sẽ lớn hơn.



Các bài tập áp dụng

- ❶ **Xoay vòng danh sách:** Cho một danh sách nhập từ bàn phím, hãy xoay vòng danh sách đã cho bằng cách chuyển phần tử cuối cùng đến vị trí đầu tiên của danh sách. Ví dụ: $[1,2,3,4,5]$ xoay vòng thành $[5,1,2,3,4]$.
- ❷ **Dãy các phần tử liên tiếp bằng 1 dài nhất:** Hãy tạo một danh sách gồm có n phần tử nhập từ bàn phím chỉ chứa số 0 và số 1. Sử dụng hàm $\text{randint}(0,1)$; Tìm dãy các phần tử liên tiếp nhau có giá trị đều bằng số 1 và có nhiều phần tử nhất. Ví dụ: $[0,1,1,1,1,0,1]$, dãy này gồm các phần tử có giá trị bằng 1 nhiều nhất là 4 phần tử.
- ❸ **Bài toán thống kê:** Hãy tạo một danh sách A có n số nguyên bằng cách sử dụng hàm sinh ngẫu nhiên. Các số có giá trị thuộc đoạn $[0,1000]$. Hãy thống kê xem mỗi phần tử trong danh sách A xuất hiện bao nhiêu lần, thực hiện tính toán và in ra màn hình.



Bài tập áp dụng - Tiếp

- **Gộp hai danh sách:** Cho hai danh sách là list1 và list2 có các phần tử đã được sắp xếp theo thứ tự tăng dần. Hãy gộp hai danh sách này thành một danh sách mới là list3 sao cho các phần tử trong list3 cũng được sắp xếp tăng dần. Ví dụ:
list1 = [1,1,2,9]
list2 = [1,4,4,5]
list3 = [1,1,1,2,4,4,5,9]
- **Tổng đường chéo chính phụ ma trận:** Nhập vào một ma trận A gồm n dòng và n cột (ma trận vuông cấp n). Hãy tính tổng tất cả các phần tử nằm trên đường chéo chính và phụ của ma trận này. Ví dụ, với $n = 5$ chúng ta có ma trận sau:



Tổng các phần tử trên đường chéo chính là: $1+1+1+1+1=5$;

Tổng các phần tử trên đường chéo phụ là: $4+4+1+4+4=17$

1	0	0	0	4
0	1	0	4	0
0	0	1	0	0
0	4	0	1	0
4	0	0	0	1



Ma trận chuyển vị - Transpose matrix

Xét một ma trận A gồm có m dòng và n cột, có dạng như sau:

a_{11}	a_{12}	..	a_{1n}
...
a_{m1}	a_{m2}	..	a_{mn}

Ma trận chuyển vị của ma trận A được ký hiệu là A^t gồm n dòng và m cột, trong đó $A_{ij}^T = A_{ji}$. Yêu cầu: Hãy nhập vào ma trận A và in ra ma trận chuyển vị A^T . Ví dụ:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 9 & 8 \end{bmatrix}$$

, ma trận chuyển vị

$$A^T = \begin{bmatrix} 1 & 2 \\ 2 & 9 \\ 3 & 8 \end{bmatrix}$$

