

COMPREHENSIONS

KHOA-TRAN-VIET¹

MỤC LỤC

1	[] List comprehension	2
1.1	List comprehension cơ bản	2
1.2	List Comprehension lồng nhau	3
1.3	List Comprehension với if..else	4
2	{ } Set comprehension	5
3	{ } Dictionary comprehension	7

¹ Khoa Công nghệ Thông tin, Đại học Khoa học Huế

1 [] LIST COMPREHENSION

1.1 List comprehension cơ bản

Python có thể tạo ra các danh sách linh hoạt chỉ bằng một câu lệnh, List comprehension với cú pháp sau:

```
[expression for var in iterable if_clause]
```

Về cơ bản, List Comprehension có ba thành phần chủ yếu:

- <expression>: Gọi là biểu thức cho giá trị trả về và tùy chọn nghĩa là có thể không cần thiết. Tuy nhiên đây chính là điểm chính của List Comprehension.
- <for var in iterable>: iterable trong Python là một đối tượng cho phép bạn duyệt qua các phần tử của nó với vòng lặp for. Như vậy vòng lặp lấy các phần tử của đối tượng iterable. Nếu có expression nó sẽ áp dụng theo thứ tự 1-1 cho từng phần tử được lấy ra.
- <if _clause>: Biểu thức logic dùng để lọc các phần tử, mục này cũng tùy chọn.

Ví dụ 1: Tạo danh sách có 10 số bình phương đầu tiên.

Command Line

```
>>> lst = [x*x for x in range(1,11)]
>>> lst
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

Ví dụ 2: Tạo danh sách chứa các số bình phương đầu tiên nhỏ hơn 1000 là số lẻ.

Command Line

```
>>> lst = [x*x for x in range(100) if (x%2 !=0 and x**2 <1000)]
>>> lst
[1, 9, 25, 49, 81, 121, 169, 225, 289, 361, 441, 529, 625, 729, 841, 961]
```

Ví dụ 3: Xóa ký tự trắng của các từ trong một danh sách cho trước. Sử dụng List Comprehension để xóa các ký tự trắng thừa của các chuỗi trong một danh sách.

Ví dụ: colors = [' red', ' green ', 'blue '], sau khi xóa L = ['red', 'green', 'blue']

Sử dụng hàm strip() để cắt ký tự trắng thừa bên phải và bên trái chuỗi.

Command Line

```
>>> colors = [' red', ' green ', 'blue ']
>>> L = [color.strip() for color in colors]
>>> print(L)
['red', 'green', 'blue']
```

Bài tập 1 Liệt kê bội số

Cho ba số nguyên dương a, b, c và số nguyên dương n . Hãy liệt kê các bội số của ba số trên trong đoạn từ 1 đến n

Dữ liệu vào: - Gồm một dòng duy nhất chứa bốn số nguyên n, a, b, c thỏa $0 < a, b, c < n \leq 10^6$.

Kết quả: - Ghi ra các bội số của a, b, c trên cùng 1 dòng. Giữa các số cách nhau một dấu cách. Nếu bài toán không có nghiệm thì ghi số -1.

Ví dụ:

Input

30 4 2 3

Output

12 24

```

1 n, a, b, c = map(int, input().split())
2 multiples = [x for x in range(1, n+1) if ((x%a == 0) and (x%b == 0) and (x%c == 0))]
3 if not multiples:
4     print(-1)
5 else:
6     for x in multiples:
7         print(x, end=" ")

```

Thuật toán 1: Boiso

1.2 List Comprehension lồng nhau

Cú pháp:

```

[expression for var in iterable]
↑
[expression for var in iterable for var in iterable]

```

Tương tự như trên nhưng với cú pháp lồng nhau này giúp ta có thể tạo ra các danh sách lồng nhau (nhiều chiều). Biểu thức ban đầu có thể thay thế bằng một cú pháp của List Comprehension.

Bài tập 2 Tính ma trận chuyển vị

Cho ma trận $A_{n,m}$. Lập trình in ma trận chuyển vị của A .

Dữ liệu vào:

- Dòng thứ nhất chứa hai số nguyên m, n là số dòng và cột của ma trận thỏa $1 \leq m, n \leq 100$
- m dòng tiếp theo chứa n số nguyên là các phần tử $a_{i,j}$ của ma trận.

Dữ liệu ra:

- In ra ma trận A sau khi đã chuyển vị.

Ví dụ:

Input:

```
3 4
1 2 3 4
5 6 7 8
9 10 11 12
```

Output:

```
1 5 9
2 6 10
3 7 11
4 8 12
```

Lời giải:

```
1 m,n = map(int, input().split())
2 A = [list(map(int, input().split())) for i in range(m)]
3 B = [[row[i] for row in A] for i in range(n)]
4 # hoặc
5 B = [[A[j][i] for j in range(m)] for i in range(n)]
6 for i in range(n):
7     for j in range(m):
8         print(B[i][j], end=' ')
9     print()
```

Thuật toán 2: maTranChuyenVi

1.3 List Comprehension với if..else

Ta có thể sử dụng câu lệnh if..else để sinh các đối tượng cần lưu trữ trong list theo điều kiện rẽ nhánh ở phần expression. Xét ví dụ đoạn code sau:

Command Line

```
>>> L=[(i, 'Even') if i%2==0 else (i, 'Odd') for i in range(1,6)]
>>> L
[(1, 'Odd'), (2, 'Even'), (3, 'Odd'), (4, 'Even'), (5, 'Odd')]
```

Bài tập 3 Biến đổi ma trận

Cho ma trận vuông $A_{n \times n}$. Lập trình in ma trận A sau khi biến đổi các phần tử về 0 trừ các phần tử trên đường chéo chính được bình phương về giá trị.

Dữ liệu vào:

- Dòng thứ nhất chứa số nguyên n là số dòng và cột của ma trận thỏa $1 \leq n \leq 100$
- n dòng tiếp theo chứa n số nguyên là các phần tử $a_{i,j}$ của ma trận.

Dữ liệu ra:

- In ra ma trận A sau khi đã thay đổi.

Ví dụ:

Input:

3

1 2 3

5 6 7

9 10 8

Output:

1 0 0

0 36 0

0 0 64

Lời giải:

```

1 n = int(input())
2 A = [list(map(int, input().split())) for i in range(n)]
3 B = [[A[i][j]*A[i][j] if i == j else 0 for i in range(n)] for j in range(n)]
4 for i in range(n):
5     for j in range(n):
6         print(B[i][j], end=' ')
7     print()

```

Thuật toán 3: bienDoiMaTran

2 {} SET COMPREHENSION

Hoàn toàn tương tự như List Comprehence, điểm khác biệt duy nhất là các phần tử lưu vào tập hợp (set) là đôi một phân biệt nhau và cú pháp chỉ thay đổi dấu [] thành dấu {}.

Ví dụ 4: Tạo tập các số lẻ trong khoảng 1 đến 100.

Command Line

```

>>> S = {x for x in range(100) if (x%2 !=0)}
>>> print(S)
1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53,
55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99

```

Bài tập 4 Sàng số nguyên tố

Tạo một tập hợp chứa các số nguyên tố nhỏ hơn hoặc bằng số nguyên n cho trước.

Dữ liệu vào:

- Dòng duy nhất chứa số nguyên n thỏa $2 \leq n \leq 10^6$

Dữ liệu ra:

- In ra tập số nguyên tố.

Ví dụ:

Input:

10

Output:

2, 3, 5, 7

Lời giải:

```
1 sieve = lambda n: set(range(2,n)) - set({i for q in range(2,int(n**0.5)) for i in range(q*2,
    n,q)})
2 n = int(input())
3 print(sieve(n))
```

Thuật toán 4: SievePrime

Bài tập 5 Từ điển

Cho một xâu văn bản chứa các từ có thể có ký tự hoa, dấu chấm, dấu phẩy. Hãy tạo một tập hợp chứa các từ phân biệt xuất hiện trong xâu văn bản trên.

Dữ liệu vào:

- Dòng duy nhất chứa xâu văn bản.

Dữ liệu ra:

- In ra tập các từ khác nhau của xâu sau khi chuyển về ký tự thường.

Ví dụ:

Input:

The cat in the hat had two sidekicks, thing one and thing two.

Output:

and

cat

had

hat

in

one

sidekicks

the

thing

two

Lời giải:

```
1 str = str(input())
2 unique_words = {word for word in str.lower().replace('.', ' ').replace(',', ' ').split()}
3 for word in unique_words:
4     print(word)
```

Thuật toán 5: tuDien

3 {} DICTIONARY COMPREHENSION

Đây là một cấu trúc đặc biệt khác với hai cấu trúc trên, kiểu dữ liệu Dictionary (từ điển) trong Python là một tập hợp các cặp key-value không có thứ tự, có thể thay đổi và lập chỉ mục (truy cập phần tử theo chỉ mục). Mỗi cặp key-value được xem như là một item. Kiểu dữ liệu này hoàn toàn tương tự lớp map trong thư viện STL của C++.

Key và value được phân biệt bởi dấu hai chấm (:). Các item phân biệt nhau bởi dấu phẩy (,). Python có thể tạo ra từ điển linh hoạt chỉ bằng một câu lệnh Dictionary comprehension với cú pháp sau:

```
{key:value for var in iterable}
```

<i>key: value</i>	<i>var</i>	<i>iterable</i>
<i>Key & value can be any expression & evaluated once for each item in iterable</i>	<i>It takes items from an iterable one by one</i>	<i>It's a collection of objects (like a list, tuple etc.)</i>

Về cơ bản, Dictionary Comprehension hoàn toàn tương tự List Comprehension, điểm khác biệt là các phần tử được sinh ra và lưu vào từ điển là một cặp hai giá trị là khóa:giá trị (key:value).

Ví dụ 5: Ký tự 'A' có mã ASCII là 65, 'B' là 66, tương tự cho đến 'Z'. Thay vì phải xử lý số là 65 cho 'A' ta lập một từ điển mà khóa cho ký tự 'A' là 0 và tiếp tục tăng 1 cho các ký tự còn lại.

Command Line

```
>>> D= {i:chr(65+i) for i in range(26)}
>>> print(D)
0: 'A', 1: 'B', 2: 'C', 3: 'D', 4: 'E', 5: 'F', 6: 'G', 7: 'H', 8: 'I', 9: 'J', 10: 'K', 11: 'L', 12: 'M', 13: 'N', 14: 'O', 15: 'P', 16: 'Q', 17: 'R', 18: 'S', 19: 'T', 20: 'U', 21: 'V', 22: 'W', 23: 'X', 24: 'Y', 25: 'Z'
```

Bạn nào đã sử dụng qua hai hàm Vlookup(), HLookup() trong Excel chắc hẳn đã hiểu rõ công dụng của cấu trúc này.

Bài tập 6 In tháng bằng tiếng Anh

<http://oj.hueuni.edu.vn/practice/problem/347/details>

Lời giải:

```
1 n = int(input())
2 month = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December']
3 switch = {i+1:month[i] for i in range(12)}
4 print(switch[n] if n in switch else 'invalid Month number!')
```

Thuật toán 6: printMonth

Ứng dụng của ba cấu trúc trên rất rộng rãi trong lập trình với ngôn ngữ Python. Bài viết chỉ trình bày một ý nhỏ về cách tạo dựng linh hoạt ba cấu trúc đó bằng khái niệm gọi là Comprehension.

Chúc vui khi đọc.