20. List, Set, Dictionary Comprehension

List Comprehension

Trong toán học, ví dụ, các bình phương của các số tự nhiên được tạo ra bởi {x^2 | X ∈ ℕ} hoặc tập các số nguyên phức {(x, y) | X ∈ ℤ ∧ y ∈ ℤ}.

List Comprehension là cách dễ dàng để đinh nghĩa hay tạo ra một list các phần tử trong python. Nó là một thay thế cho lambda, map(), filter(), reduce(). Đối với nhiều người list comprehensive dễ nắm bắt hơn.

Trong phạm vi bài này tôi sẽ dùng CMD thay cho IDE để thực hiện các ví dụ cho bài viết.

Ví dụ 1: Trong bài trước ta có ví dụ về chuyển đổi giữa đồng vnd và usd. Ví dụ dưới đây là cách thức thay thế sử dụng map() và lambda trong ví dụ đó.

```
>>> vnd = [20e6,50e6,70e7]

>>> usd = [float(x)/22000 for x in vnd]

>>> usd

[909.0909090909091, 2272.7272727272725, 31818.18181818182]

>>> [x * 22000 for x in usd]

[20000000.0, 49999999.99999999, 700000000.0]

>>>
```

Ví dụ 2: Tìm một bộ ba số Pythagoras gồm ba số nguyên dương a, b, và c, sao cho a**2 + b**2 = c**2 (**2 là toán tử bình phương). Khi đó ta viết bộ ba đó là (a, b, c), và bộ ba ai cũng biết là (3, 4, 5) với điều kiện a,b,c < 20.

```
>>> [(a,b,c) for a in range(1,20) for b in range(1,20) for c in range(1,20) if a**2 + b**2 == c**2]

[(3, 4, 5), (4, 3, 5), (5, 12, 13), (6, 8, 10), (8, 6, 10), (8, 15, 17), (9, 12, 15), (12, 5, 13), (12, 9, 15), (15, 8, 17)]

>>>
```

Set Comprehension và Dictionary Comprehension

Ví dụ 2: set comprehension

```
>>> import random

>>> {random.randint(0,100) for i in range(10)}

set([96, 33, 43, 38, 73, 23, 85, 55, 25, 61])

>>>
```

Ví dụ 3: dictionary compresension

```
>>> {key:{random.randint(0,100) for i in range(10)} for key in range(10)}

{0: set([97, 34, 72, 42, 45, 46, 19, 86, 57, 91]), 1: set([33, 1, 7, 43, 12, 50, 55, 29]), 2: set([33, 4, 69, 41, 78, 50, 83, 20, 86, 59]), 3: set([96, 33, 38, 74, 46, 14, 47, 62, 42, 30]), 4: set([64, 83, 7, 1

>>>
```

Kết luận

List comprehension là một cách diễn đạt khác thay thế cho việc sử dụng lambda, filter(), reduce() hay map() như đã giới thiệu trong bài học trước. Các bạn có thể sử dụng cách nào cũng được nếu bạn nắm chắc và hiểu về chúng. List comprehension có thể chứa những phần tử có cùng giá trị mà đôi khi chúng ta không cần dùng đến. Trong trường hợp đó Set comprehension lại là một sự lựa chọn thích hợp hơn để tạo ra một tập các phần tử đơn nhất. Nên tùy vào yêu cầu mỗi bài toán, các bạn cần lựa chọn các cách diễn đạt sao cho phù hợp và tối ưu nhất.