

Генератори масивів. Функції `map()` та `lambda`

Генератори списків

В мові програмування Python для створення та заповнення елементами масивів є значне спрощення. Для цих цілей використовують генератори списків (*list comprehension*).

Генератор списків дозволяє швидко створити та заповнити список на базі іншого масива.

Генератор списку записується у квадратних дужках та має наступні частини:

```
>>> a = [1,2,3]
>>> b = [i+10 for i in a if i<=2]
```

- 1) Дія, що виконується з кожним елементом перед записом у список. (`i+10`)
- 2) Назва змінної. (`for i`)
- 3) Початковий (базовий) масив. (`in a`)
- 4) Можлива наявність умови. (`if i<=2`)

Наявність умови не обов'язкова.

Як ітог ми отримуємо не змінени список `a` та створений на його основі новий список `b`.

```
>>> a
[1, 2, 3]
>>> b
[11, 12]
```

Також як базовий масив може використовуватися *діапазон*.

```
>>> b = [i+10 for i in range(6)]
>>> b
[10, 11, 12, 13, 14, 15]
```

Так як списки майже однакові з кортежами, то після використання генератора списку результат можна конвертувати у кортеж.

```
>>> b = tuple([i+10 for i in range(6)])
>>> b
(10, 11, 12, 13, 14, 15)
```

Генератори словників та множин

Для створення заповнених словників та множин також є генератори. Принцип їх написання не відрізняється від генераторів списків, але квадратні дужки замінюються на фігурні.

```
>>> a = [1,2,3,4,5]
>>> b = {i:i*2 for i in a}
>>> b
{1: 2, 2: 4, 3: 6, 4: 8, 5: 10}
```

Для створення словників потрібно вказувати пару ключ-елемент. Вони вказуються через двокрапку.

```
>>> a = [1,1,2,2,3,4,5,5,5]
>>> b = {i for i in a}
>>> b
{1, 2, 3, 4, 5}
```

Створення множини повністю аналогічне до створення списку.

Функції `map()` та `lambda`

Для швидкої обробки вже існуючих списків існує функція `map()`. В її параметри вводяться функція, яка застосовується на кожен елемент масиву послідовно, і, власне, сам список, що оброблюється. Для простіших дій можна використовувати *анонімну функцію* `lambda`.

Приклад функції `lambda`:

```
lambda x: x*2
```

В данному випадку функція приймає `x` та повертає його подвійне значення.

Приклад `map()`:

```
>>> a = [1,2,3,4,5]
>>> b = list(map(lambda x: x*2, a))
>>> b
[2, 4, 6, 8, 10]
```

Функція `map()` повертає об'єкт, тому для його коректного читання потрібно конвертувати результат у список.

Контрольні питання:

1. Для чого призначений генератор списків?
2. Як зробити генератор списків?
3. Як отримати кортеж, використовуючи генератор списків?
4. Як зробити генератор словників?
5. Як використовувати функцію `map()`?