

7.2. Кортежі

Кортеж (tuple) - це незмінна структура даних, яка за своєю будовою дуже схожа на список. Інколи навіть кажуть, що кортеж – це незмінюваний список. Так само, як і список, кортеж може містити елементи різних типів. Кортеж записується, як перелік елементів, розділених комою та взятих в круглі дужки: (1, 3, 5, 'Hello').

Існує кілька причин, коли варто використовувати кортежі замість списків. Першою причиною є можливість захисту даних від випадкової зміни (захист від дурня). Якщо ми отримали набір даних, і є необхідність опрацювати його без зміни даних, то це як раз той випадок, коли доцільно використати кортеж.

Другою причиною є те, що кортежі в пам'яті займають менший об'єм у порівнянні зі списками.

```
>>> lst=[1, 2, 3]
>>> tpl=(1, 2, 3)
>>> print (lst.__sizeof__())
32
>>> print (tpl.__sizeof__())
24
```

Третьою причиною є приріст продуктивності, який пов'язаний з тим, що кортежі працюють швидше, ніж списки (наприклад, операції перебору елементів). Четвертою причиною є можливість використання кортежів в якості ключа у словнику.

```
>>> d = {(1, 1, 1) : 1}
>>> d = {[1, 1, 1] : 1}
Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    d = {[1, 1, 1] : 1}
TypeError: unhashable type: 'list'
```

7.2.1. Задання кортежів

Для задання порожнього кортежу можна скористатися однією з наступних команд:

```
>>> a=()
>>> b=tuple()
```

Задання кортежу з наперед заданим набором елементів:

```
>>> a=(1, 3, 5, 'Hello')
```

Окрім того, будь-який набір різних об'єктів, розділених комами та не виділених будь-якими дужками (тобто ні квадратними, ні фігурними, ні круглими) за замовчуванням буде вважатися кортежем:

```
>>> b=1, 3, 5, 'Hello'
>>> print(b)
(1, 3, 5, 'Hello')
```

Проте створення кортежу з одного елемента має певні особливості. Так, ввівши команду `a = (3)`, отримаємо в змінній `a` лише число, а не кортеж

```
>>> a=(3)
>>> print(a)
3
```

Для створення кортежу з одного елемента можна записати:

```
>>> a=(3,)
>>> print(a)
(3,)
```

Створення кортежів з інших структур даних

Кортеж можна отримати з елементів об'єкта, що може ітеруватися (діапазон, рядок, словник, множина, кортеж, файл і т.д.), використавши функцію `tuple([iterable])`:

```
>> b=tuple(range(1,10,2))
>>> print(b)
(1, 3, 5, 7, 9)
>>> b=tuple('Hello')
>>> print(b)
('H', 'e', 'l', 'l', 'o')
```

Об'єднання кортежів

Нові кортежі можуть бути створені методом конкатенації (об'єднання) декількох кортежів. Для цього використовується перевизначена операція додавання («+»), яка використовується як операція конкатенації кортежів.

```
>>> b=(1, 3, 5, 7, 9)
>>> b=b + (10,)
>>> print(b)
(1, 3, 5, 7, 9, 10)
```

Багаторазове повторення елементів

Аналогічно до перевизначеної операції («+») для кортежів, в мові Python перевизначена і операція множення «*». Якщо виконати операцію «*» кортежу *b* на ціле число *n*, то в результаті буде отриманий кортеж, що складається з *n* повторень кортежу *b*:

```
>>> b=(0,)*5
>>> print(b)
(0, 0, 0, 0, 0)
```

7.2.2. Виконання дій над кортежами та їхніми елементами

Доступ до елементів кортежу

Доступ до елементів кортежу здійснюється аналогічно доступу до елементів списку – за їхніми індексами. Тобто для того, щоб звернутися до елемента кортежу, необхідно вказати ім'я змінної кортежу та в квадратних дужках індекс необхідного елемента (ім'я_кортежу[індекс]).

```
>>> a=(1, 3, 5, 7)
>>> print(a[0])
1
```

Те ж саме стосується і зрізів кортежів. Задати зріз можна одним з двох варіантів:

```
tuple[start: stop].
tuple[start: stop: step].
```

```
>>> print(a[1:3])
(3, 5)
```

Можна перевірити приналежність деякого елемента до кортежу, використовуючи оператор `in` (значення `in` ім'я_кортежу).

```
>>> a=(1, 3, 5, 7)
>>> 3 in a
True
```

```
>>> 3 in a
False
```

Але, як вже було сказано - змінювати елементи кортежу не можна. При спробі змінити чи вилучити елемент кортежу виникне виняток `TypeError`.

```
>>> b=(1, 2, 3)
>>> b[1]=15
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    b[1]=15
TypeError: 'tuple' object does not support item
      assignment
>>> del b[1]
Traceback (most recent call last):
  File "<pyshell#49>", line 1, in <module>
    del b[1]
TypeError: 'tuple' object doesn't support item
deletion
```

Вилучення кортежів

Як було зазначено, вилучити окремий елемент з кортежу неможливо, але можна видалити кортеж повністю, скориставшись командою `del` (`del ім'я_кортежу`).

```
>>> a=(1, 3, 5, 7)
>>> del a
>>> print(a)
Traceback (most recent call last):
  File "<pyshell#27>", line 1, in <module>
    print(a)
NameError: name 'a' is not defined
```

Перетворення кортежів у список і назад

На основі існуючого кортежу можна створити список, як і навпаки - зі списку можна створити кортеж. Тобто, використовуючи кортеж для захисту