

Кортежі

Розділ 10

Python для всіх
www.py4e.com



Кортежі схожі на списки

Кортеж – це ще один тип послідовності, який функціонує подібно до списку – вони мають елементи, які індексуються, починаючи з 0

```
>>> x = ('Glenn', 'Sally', 'Joseph')
```

```
>>> print(x[2])
```

```
Joseph
```

```
>>> y = ( 1, 9, 2 )
```

```
>>> print(y)
```

```
(1, 9, 2)
```

```
>>> print(max(y))
```

```
9
```

```
>>> for iter in y:  
...     print(iter)
```

```
...
```

```
1
```

```
9
```

```
2
```

```
>>>
```

Але...Кортежі «незмінні»

На відміну від **списку**, після створення кортежу ви **не можете змінити** його вміст — подібно до рядка

```
>>> x = [9, 8, 7]
>>> x[2] = 6
>>> print(x)
>>> [9, 8, 6]
>>>
```

```
>>> y = 'ABC'
>>> y[2] = 'D'
Traceback: 'str'
object does
not support item
Assignment
>>>
```

```
>>> z = (5, 4, 3)
>>> z[2] = 0
Traceback: 'tuple'
object does
not support item
Assignment
>>>
```

Що не можна робити з кортежами

```
>>> x = (3, 2, 1)
```

```
>>> x.sort()
```

```
Traceback:
```

```
AttributeError: 'tuple' object has no attribute 'sort'
```

```
>>> x.append(5)
```

```
Traceback:
```

```
AttributeError: 'tuple' object has no attribute 'append'
```

```
>>> x.reverse()
```

```
Traceback:
```

```
AttributeError: 'tuple' object has no attribute 'reverse'
```

```
>>>
```

Казка про дві послідовності

```
>>> l = list()
>>> dir(l)
['append', 'count', 'extend', 'index', 'insert', 'pop',
'remove', 'reverse', 'sort']
```

```
>>> t = tuple()
>>> dir(t)
['count', 'index']
```

Кортежі більш ефективні

- Оскільки Python не потрібно створювати структури кортежів, які можна модифікувати, вони простіші та ефективніші з боку використання пам'яті та продуктивності, ніж списки
- Тому в нашій програмі, коли ми створюємо «тимчасові змінні», ми надаємо перевагу кортежам перед списками

Кортежі та присвоювання

- Ми також можемо помістити **кортеж** у **лівій частині** інструкції присвоювання
- Ми навіть можемо опустити дужки

```
>>> (x, y) = (4, 'fred')
>>> print(y)
fred
>>> (a, b) = (99, 98)
>>> print(a)
99
```

Кортежі та словники

Метод `items()` у
словниках
повертає список
кортежів (ключ,
значення)

```
>>> d = dict()
>>> d['csev'] = 2
>>> d['cwen'] = 4
>>> for (k,v) in d.items():
...     print(k, v)
...
csev 2
cwen 4
>>> tups = d.items()
>>> print(tups)
dict_items([('csev', 2), ('cwen', 4)])
```


Кортежі можна порівнювати

Оператори порівняння працюють з **кортежами** та іншими послідовностями. Якщо перші елементи однакові, Python переходить до наступного елемента і так далі, поки не знайде елементи, які відрізняються

```
>>> (0, 1, 2) < (5, 1, 2)
True
>>> (0, 1, 2000000) < (0, 3, 4)
True
>>> ( 'Jones', 'Sally' ) < ( 'Jones', 'Sam' )
True
>>> ( 'Jones', 'Sally' ) > ( 'Adams', 'Sam' )
True
```

Сортування списків кортежів

- Ми можемо скористатися можливістю сортування списку **кортежів**, щоб отримати відсортовану версію словника
- Спочатку ми сортуємо словник за ключем за допомогою методу **items()** та функції **sorted()**

```
>>> d = {'a':10, 'c':22, 'b':1}
>>> d.items()
dict_items([('a', 10), ('c', 22), ('b', 1)])
>>> sorted(d.items())
[('a', 10), ('b', 1), ('c', 22)]
```

Використання sorted()

Ми можемо зробити це безпосередньо за допомогою вбудованої функції `sorted`, яка отримує послідовність як параметр і повертає відсортовану послідовність

```
>>> d = {'a':10 , 'b':1, 'c':22}
>>> t = sorted(d.items())
>>> t
[('a', 10), ('b', 1), ('c', 22)]
>>> for k, v in sorted(d.items()):
...     print(k, v)
...
a 10
b 1
c 22
```

Сортування за значенням замість ключа

- Якби ми могли створити список кортежів виду (значення-ключ), ми могли б сортувати за значенням
- Ми робимо це за допомогою циклу `for`, який створює список кортежів

```
>>> c = {'a':10, 'b':1, 'c':22}
>>> tmp = list()
>>> for k, v in c.items():
...     tmp.append( (v, k) )
...
>>> print(tmp)
[(10, 'a'), (1, 'b'), (22, 'c')]
>>> tmp = sorted(tmp, reverse=True)
>>> print(tmp)
[(22, 'c'), (10, 'a'), (1, 'b')]
```

10 найпоширеніших слів

```
fhand = open('romeo.txt')
counts = {}
for line in fhand:
    words = line.split()
    for word in words:
        counts[word] = counts.get(word, 0) + 1

lst = []
for key, val in counts.items():
    newtup = (val, key)
    lst.append(newtup)

lst = sorted(lst, reverse=True)

for val, key in lst[:10]:
    print(key, val)
```

Коротша версія

```
>>> c = {'a':10, 'b':1, 'c':22}
```

```
>>> print( sorted( [ (v,k) for k,v in c.items() ] ) )
```

```
[(1, 'b'), (10, 'a'), (22, 'c')]
```

Генератор списку створює динамічний список. У цьому випадку ми створюємо список з перевернутих кортежів, а потім сортуємо його

<http://wiki.python.org/moin/HowTo/Sorting>

Підсумки

- Синтаксис кортежів
- Незмінність
- Порівняння
- Сортування
- Кортежі в інструкціях присвоювання
- Сортування словників за ключем або значенням

Права власності / Застереження



Авторські права на ці слайди з 2010 року належать Чарльзу Северенсу (www.dr-chuck.com) зі Школи інформації Мічиганського університету та захищені ліцензією Creative Commons Attribution 4.0. Будь ласка, збережіть цей фінальний слайд у всіх копіях документа, щоб відповідати вимогам ліцензії щодо посилань на джерела. При повторній публікації матеріалів, якщо щось зміните, додайте ім'я та організацію до переліку співавторів нижче.

Першоджерело: Чарльз Северенс, Школа інформації Мічиганського університету

Переклад: платформа Prometheus