

## Практическая работа №7. Кортежи (tuple)

- использование кортежей
- основные приемы работы с кортежами
- преобразования кортежа в список и обратно.

### Что такое кортеж (tuple) в Python?

Кортеж (tuple) – это неизменяемая структура данных, которая по своему подобию очень похожа на список (Список - это изменяемый тип данных).

Допустим, есть список `a = [1, 2, 3]` и если мы хотим *заменить второй элемент*, находящийся в позиции 1 на 15, то можно это сделать, напрямую обратившись к элементу списка.

```
>>> a = [1, 2, 3]
>>> print(a)
[1, 2, 3]
>>> a[1] = 15
>>> print(a)
[1, 15, 3]
```

*В кортеже мы не можем производить такие операции, т.к. элементы его изменять нельзя.*

```
>>> b = (1, 2, 3)
>>> print(b)
(1, 2, 3)
>>> b[1] = 15
Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    b[1] = 15
TypeError: 'tuple' object does not support item assignment
```

### Зачем нужны кортежи в Python?

- 1- обезопасить данные от случайного изменения.
- 2- увеличение производительности выполнения программы, который связан с тем, что кортежи работают быстрее из-за своего размера, чем списки
- 3- кортежи можно использовать в качестве ключа в словарях.
- 4- экономия места в памяти ПК, отводимое для переменной типа кортеж.

*Убедимся в том, что кортежи занимают меньший объем по сравнению со списками.*

```
>>> lst = [10, 20, 30]
>>> tpl = (10, 20, 30)
```

```
>>> print(lst.__sizeof__())
32
>>> print(tpl.__sizeof__())
24
```

## ОСНОВНЫЕ ПРИЕМЫ РАБОТЫ С КОРТЕЖАМИ

### 1. Создание кортежей

Для создания пустого кортежа можно воспользоваться одной из следующих команд.

```
>>> a = ()
>>> print(type(a))
<class 'tuple'>
>>> b = tuple()
>>> print(type(b))
<class 'tuple'>
```

Кортеж с заданным содержанием создается также как список, только вместо квадратных скобок используются *круглые*.

```
>>> a = (1, 2, 3, 4, 5)
>>> print(type(a))
<class 'tuple'>
>>> print(a)
(1, 2, 3, 4, 5)
```

При желании можно воспользоваться функцией *tuple()*.

```
>>> a = tuple((1, 2, 3, 4))
>>> print(a)
(1, 2, 3, 4)
```

### 2. Доступ к элементам кортежа

Доступ к элементам кортежа осуществляется также как к элементам списка – через *указание индекса*. Но, как уже было сказано – **изменять элементы кортежа нельзя!**

```
>>> a = (1, 2, 3, 4, 5)
>>> print(a[0])
1
>>> print(a[1:3])
(2, 3)
>>> a[1] = 3
Traceback (most recent call last):
  File "<pyshell#24>", line 1, in <module>
    a[1] = 3
```

TypeError: 'tuple' object does not support item assignment

### 3. Удаление кортежей

Удалить *отдельные элементы из кортежа невозможно*.

```
>>> a = (1, 2, 3, 4, 5)
>>> del a[0]
Traceback (most recent call last):
  File "<pyshell#26>", line 1, in <module>
    del a[0]
TypeError: 'tuple' object doesn't support item deletion
```

*Но можно удалить кортеж целиком.*

```
>>> del a
>>> print(a)
Traceback (most recent call last):
  File "<pyshell#28>", line 1, in <module>
    print(a)
NameError: name 'a' is not defined
```

### 4. Преобразование кортежа в список и обратно

На базе кортежа можно создать список, и наоборот.

Для превращения списка в кортеж достаточно передать его в качестве аргумента функции tuple().

```
>>> lst = [1, 2, 3, 4, 5]
>>> print(type(lst))
<class 'list'>
>>> print(lst)
[1, 2, 3, 4, 5]
>>> tpl = tuple(lst)
>>> print(type(tpl))
<class 'tuple'>
>>> print(tpl)
(1, 2, 3, 4, 5)
```

Обратная операция также является корректной.

```
>>> tpl = (2, 4, 6, 8, 10)
>>> print(type(tpl))
<class 'tuple'>
>>> print(tpl)
(2, 4, 6, 8, 10)
>>> lst = list(tpl)
>>> print(type(lst))
```

```
<class 'list'>  
>>> print(lst)  
[2, 4, 6, 8, 10]
```

**Задание:**

- 1) Отработать Пайтоне все методы по теме
- 2) Переписать в конспект на память
- 3) Оформить в виде отчета в Ворде(скрины)
- 4) Сохранить отчет в PDFформате с титульным листом
- 5) Отправить до 07.04 до 9-00

**С уважением Баян Е**