



КОРТЕЖИ

Занятие №5.2

Проверка пройденного на занятии №5

1. Что такое коллекции?
2. Какие есть коллекционные типы данных ?
3. Дайте определение спискам.
4. Как объявить список?
5. Расскажите про варианты создания списков.
6. Как можно обратиться к элементу списка?
7. Индекс может быть отрицательным?
8. Как добавить элемент в список на указанную позицию?
9. Как можно объединить списки?
10. Расскажите о вариантах копирования списков.
11. Может ли быть список вложенным?

Проверка домашнего задания

Дан список `list=[15,48,'hello',6,19,'world']`.

Все числа этого списка проверить на чётность. Если число чётное, то посчитать сумму его цифр. Если нечётное, то заменить его на 1 в списке.

Все слова: посчитать количество гласных и согласных.

Вывести всё на экран.

Решение

```
list = [15, 48, 'hello', 6, 19, 'world']
l = 0
h = 0
d = 0

for i in list:
    if type(i) is int:
        if i % 2 == 0:
            i = str(i)
            for k in i:
                k = int(k)
                l += k
            print(i, "Сумма цифр: ", l, "\n")
        else:
            index = list.index(i)
            list[index] = 1
    elif type(i) is str:
        for r in i:
            if r in "aeoiu":
                h += 1
            else:
                d += 1
        print(i, "\nКоличество гласных: ", h)
        print("Количество согласных: ", d, "\n")
        h = 0
        d = 0

print(list)
```

План занятия

- 1) Кортежи (tuple) в Python.
- 2) Кортежи и списки.
- 3) Операции с кортежами.

Кортежи (tuple) в Python – это те же списки за одним исключением. Кортежи неизменяемые структуры данных. Так же как списки они могут состоять из элементов разных типов, перечисленных через запятую. Кортежи заключаются в круглые, а не квадратные скобки.

Обратите внимание на запятую (,) в объявлении кортежа a. Если ее не указать при создании объекта с одним элементом? Python предположит, что вы по ошибке добавили лишнюю пару скобок (это ни на что не влияет), но тип данных в таком случае — это не кортеж. Поэтому важно не забывать использовать запятую при объявлении кортежа с одним элементом.

```
a = (1, 2, 3, 4, 5, 6)

print(a)
```

Зачем нужны кортежи, если есть списки?

- Кортеж защищен от изменений, как намеренных (что плохо), так и случайных (что хорошо).
- Меньший размер.

```
a = (1, 2, 3, 4, 5, 6)
b = [1, 2, 3, 4, 5, 6]

print(a.__sizeof__())

print(b.__sizeof__())
```

```
72
```

```
136
```

```
Process finished with exit code 0
```

Из кортежа можно извлекать элементы и брать срезы:

```
a = (1, 2, 3, 4, 5, 6)
print(a[0:3])
print(a[:3])
print(a[1:])
print(a[2::2])
print(a[::-2])
```


Однако изменять его элементы нельзя.

```
a[1] = 11  
TypeError: 'tuple' object does not support item assignment
```

Также у типа **tuple** нет методов для добавления и удаления элементов.

Возникает резонный вопрос. Зачем в язык программирования был введен этот тип данных, по - сути представляющий собой неизменяемый список? Дело в том, что иногда надо защитить список от изменений. Преобразовать же кортеж в список, если это потребуется, как и выполнить обратную операцию легко с помощью встроенных в Python функций **list()** и **tuple()**:

```
a = (10, 2.13, "square", 89, 'C')
b = [1, 2, 3]
c = list(a)
d = tuple(b)
```

Кортежи могут содержать списки, также как списки быть вложенными в другие списки.

```
nested = (1, "do", ["param", 10, 20])
```

Как вы думаете, можем ли мы изменить список ["param", 10, 20] вложенный в кортеж nested?

Список изменяем, кортеж – нет. Если вам кажется, что нельзя, то вам кажется неправильно. На самом деле можно:

```
nested = (1, "do", ["param", 10, 20])  
nested[2][1] = 15  
print(nested)
```

Примечание. Выражения типа `nested[2][1]` используются для обращения к вложенным объектам. Первый индекс указывает на позицию вложенного объекта, второй – индекс элемента внутри вложенного объекта. Так в данном случае сам список внутри кортежа имеет индекс 2, а элемент списка 10 – индекс 1 в списке.

Объединение кортежей

```
x = (1, 2, 3, 4)
y = (5, 6, 7, 8)

# Объединение двух кортежей для формирования нового кортежа
z = x + y
print(z)
```

Разрешается объединять только определенные типы данных. Так, попытка соединить кортеж и список закончится ошибкой.

Умножение кортежей

Операция умножения приводит к тому, что кортеж повторяется несколько раз.

```
x = (1, 2, 3, 4)  
z = x * 2  
print(z)
```

Функции кортежей

В отличие от списков у кортежей нет методов, таких как `append()`, `remove()`, `extend()`, `insert()` или `pop()` опять-таки из-за их неизменяемости. Но есть другие:

`count()` и `len()`

`count()` возвращает количество повторений элемента в кортеже.

```
a = (1, 2, 3, 4, 5, 5)

print(a.count(5), len(a))
```

Функции кортежей

Функция **max()** возвращает самый большой элемент последовательности, а **min()** — самый маленький.

Эти функции можно использовать и для кортежей со строками.

```
a = (1, 2, 3, 4, 5, 5)
print('max ', max(a), 'min ', min(a))
```


Задание № 1

Создайте кортеж из случайных 10 чисел. Найдите его максимальный минимальный элемент.

Решение

```
import random

a = [random.randint(0, 100) for i in range(10)]
b = tuple(a)
print(b)
print('max ', max(a), 'min ', min(a))
```

Задание № 2

Заполните один кортеж десятью случайными целыми числами от 0 до 5 включительно. Также заполните второй кортеж числами от -5 до 0. Объедините два кортежа с помощью оператора +, создав тем самым третий кортеж. С помощью метода кортежа `count()` определите в нем количество нулей. Выведите на экран третий кортеж и количество нулей в нем.

Решение

```
import random
a = tuple([random.randint(0, 5) for _ in range(10)])
b = tuple([random.randint(-5, 0) for _ in range(10)])
c = a + b
print(c, '\n Количество нулей:', c.count(0))
```

Задание № 3

Вывести данные кортежа без скобок, через запятую.

Обязательно: элементы кортежа – строки.

Решение

```
a = ('one', 'two', 'three')
```

```
c = ','.join(a)
```

```
print(c)
```

Задание № 4

Даны два кортежа:

$A = (13, 5, 43, 49, 67, 32, 12, 98, 6, 10, 34, 20, 55, 68, 14, 60, 14)$

$B = (53, 21, 4, 23, 76, 3, 43, 12, 54, 342, 21)$

Необходимо определить:

- 1) Сумма элементов какого из кортежей больше и вывести соответствующее сообщение на экран (Сумма больше в кортеже - ..)
- 2) Вывести на экран порядковые номера минимальных элементов этих кортежей

Решение

```
A = (13, 5, 43, 49, 67, 32, 12, 98, 6, 10, 34, 20, 55, 68, 14, 60, 14)
B = (53, 21, 4, 23, 76, 3, 43, 12, 54, 342, 21)
s_A = sum(A)
s_B = sum(B)
if s_A > s_B:
    print("Сумма больше в кортеже - A")
else:
    print("Сумма больше в кортеже - B")

print('min A', min(A), 'Номер - ', A.index(min(A)))
print('min B', min(B), 'Номер - ', B.index(min(B)))
```


Закрепление темы

1. Создайте кортеж с цифрами от 0 до 9 и посчитайте сумму
2. Выведите статистику частности букв в кортеже

```
long_word = ('т', 'т', 'а', 'и', 'и', 'а', 'и',  
            'и', 'и', 'т', 'т', 'а', 'и', 'и',  
            'и', 'и', 'и', 'т', 'и')
```

3. Допишите скрипт для расчета средней температуры
Постарайтесь посчитать количество дней на основе week_temp.
Так наш скрипт сможет работать с данными за любой период

```
week_temp = (26, 29, 34, 32, 28, 26, 23)  
sum_temp =  
days =  
mean_temp = sum_temp / days  
print(int(mean_temp))
```

Домашнее задание

Задание №1

Найти самое длинное слово в строке.

Задание №2

Преобразовать текст в список слов с удалением знаков препинания.