Кортежи

Основы Python

План урока

- Особенности кортежей
- Отличие кортежей от списков
- Примеры задач

КОРТЕЖИ

Это коллекция, которая является итерируемым объектом, то есть по кортежу можно:

- Итерироваться
- Обращаться к элементам по индексам
- Брать срезы

Создание кортежа

```
ооо код

a = (2, -34, 13)←

print(a, type(a))
```

000	вывод		
>>>	(2, -34,	13) <class 'tuple'=""></class>	

В переменную а записываем кортеж из чисел: 2, -34, 13

Создание кортежа

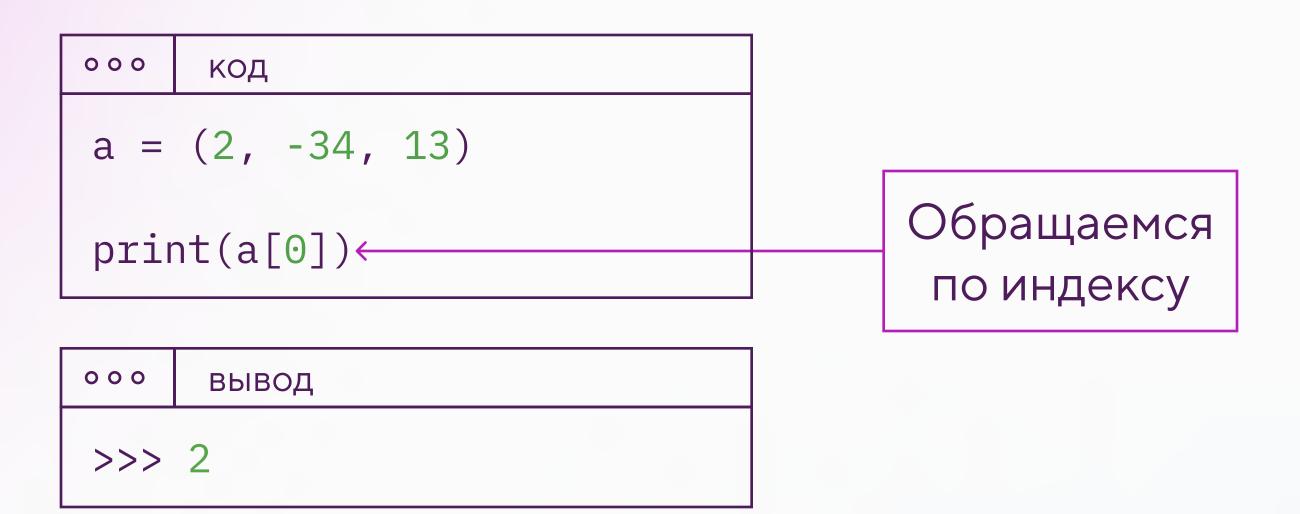
```
ооо код

a = (2, -34, 13)

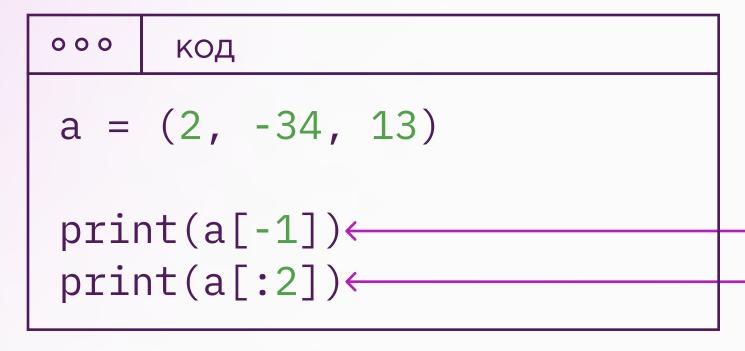
print(a, type(a))←
```

Выводим в консоль и печатаем тип

Обращение к элементу кортежа



Обращение к элементу кортежа



Используем отрицательный индекс

ooo вывод
>>> 13
>>> (2, -34)

Используем срез

ID элементов кортежа

```
ооо код
a = (3, 0, -1)
b = (3, 0, -1)
print(id(a), id(b))
```

```
      ооо
      вывод

      >>> 139729186887104
      139729186887104
```

Неизменяемость кортежей

```
ооо код

a = (3, 0, -1)

try:
    a[1] = 5
except Exception as e:
    print(e)
print(a)
```

```
ooo вывод

>>> 'tuple' object does
not support item assignment
>>> (3, 0, -1)
```

Множественное присваивание

```
ooo код
a = (3, 0, -1)
x, y, z = a
print(x, y, z)
```

```
ooo вывод
>>> 3 0 -1
```

Если первый элемент кортежа А **больше** элемента кортежа Б, то кортеж А считается **больше**, чем кортеж Б, и наоборот

```
ооо код

tuple_a = (5, 10, 15)

tuple_b = (2, 4, 6)
```

```
5 > 2
tuple_a > tuple_b
```

Если первый элемент кортежа А **больше** элемента кортежа Б, то кортеж А считается **больше**, чем кортеж Б, и наоборот

```
ооо код

tuple_c = (3, 1, 5)

tuple_d = (3, 7, 9)
```

```
7 > 1
tuple_d > tuple_c
```

Если первый элемент кортежа А **больше** элемента кортежа Б, то кортеж А считается **больше**, чем кортеж Б, и наоборот

```
ооо код

tuple_e = (5, 6, 7)

tuple_f = (5, 6, 7)
```

```
tuple_e = tuple_f
```

ооо код a = (1, 4) b = (3, 9)print(a < b, a == b, a > b)

Выводим значения сравнения

••• Вывод

>>> True False False

```
ооо код

a = (1, 4)
b = (3, 9)

print(a < b, a == b, a > b)
```

```
ooo Вывод
>>> True False False
```

```
ооо код
a = (1, 4)
b = (3, 9)

print(a < b, a == b, a > b)
```

```
ooo Вывод
>>> True False False
```

```
1 < 3
a < b
>>> True
```

```
ооо код

a = (4, 9)
b = (4, 11)

print(a < b, a == b, a > b)
```

```
°°° Вывод
>>> True False False
```

```
9 < 11
a < b
>>> True
```

```
ooo код

a = (4, 9)
b = (4, 9)

print(a < b, a == b, a > b)
```

```
ooo Вывод
>>> False True False
```

```
(4, 9) = (4, 9)
a = b
>>> True
```

Пример

```
000
     КОД
age_name = [
(12, 'Peter'),
(14, 'Anna'),
(10, 'Steven'),
(27, 'Jack'),
(20, 'Garry'),
(14, 'Bob'),
(27, 'Alice'),
(14, 'Elizabeth'),
(30, 'Amelia'),
                          Имя человека
Возраст человека
```

METOД SORT

Сортирует элементы в порядке возрастания

Пример

```
000
     КОД
age_name = [
    (12, 'Peter'),
    (14, 'Anna'),
    (10, 'Steven'),
    (27, 'Jack'),
    (20, 'Garry'),
    (14, 'Bob'),
    (27, 'Alice'),
    (14, 'Elizabeth'),
    (30, 'Amelia'),
age_name.sort()
for age, name in age_name:
    print(age, name)
```

Пример

```
ooo вывод

>>> 10 Steven

>>> 12 Peter

>>> 14 Anna

>>> 14 Bob

>>> 14 Elizabeth

>>> 20 Garry

>>> 27 Alice

>>> 30 Amelia
```