Кортежи
Tuple

Кортежи похожи на списки

Кортежи — еще один вид последовательности, которая очень похожа на список. Элементы кортежа индексируются, начиная с 0

```
>>> x = ('Гленн', 'Салли', 'Джозеф')
>>> print(x[2])
Джозеф
>>> y = ( 1, 9, 2 )
>>> print(y)
(1, 9, 2)
>>> print(max(y))
9
```

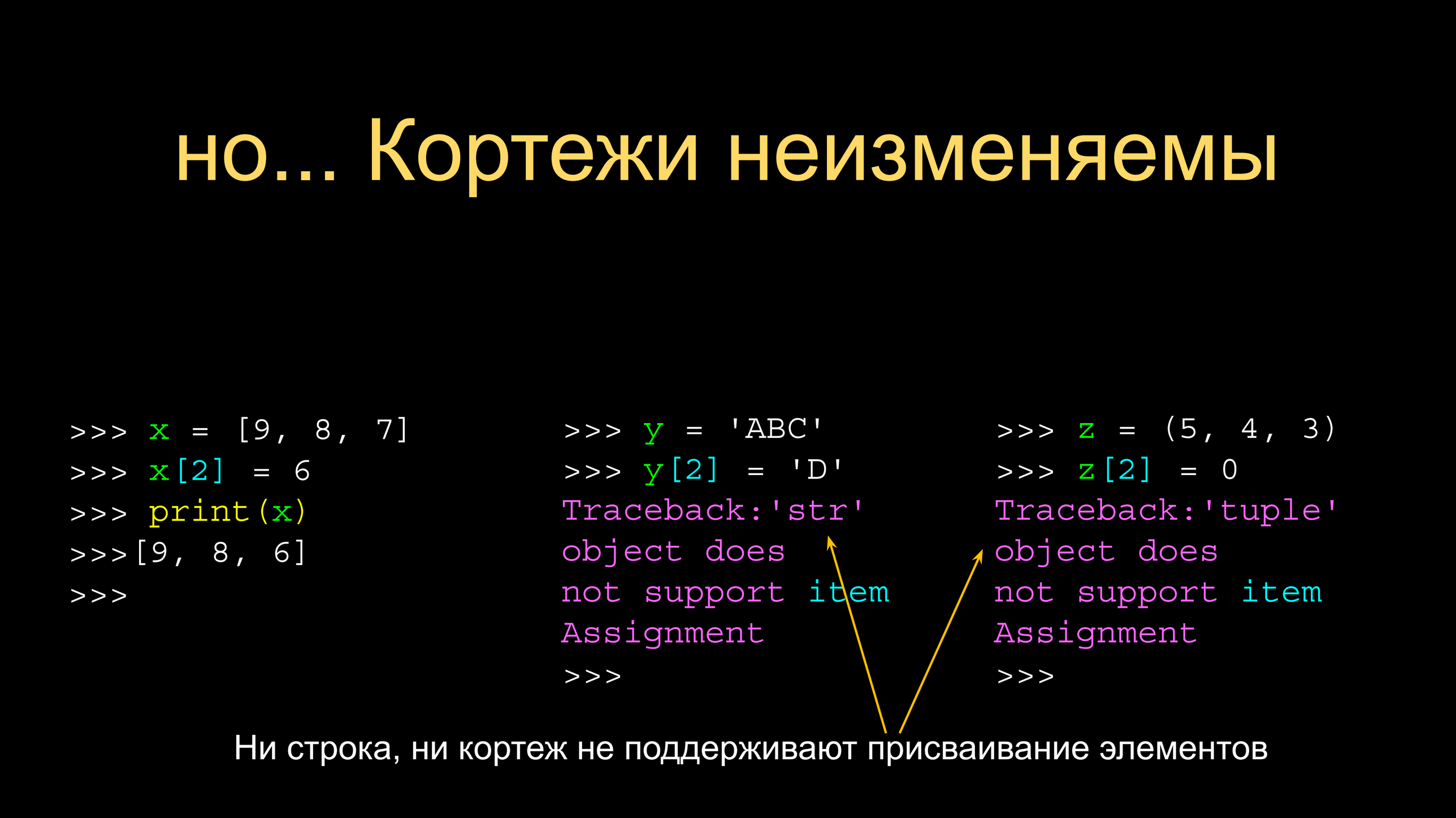
```
>>> for iter in y:
...     print(iter)
...
1
9
2
>>>
```

но… Кортежи неизменяемы

```
>>> 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
>>>
```



Ни строка, ни кортеж не поддерживают присваивание элементов

Что нельзя сделать с кортежем

```
>>> 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'
>>>
```

Кортеж не сортируется

В кортеж нельзя добавить элемент

Нельзя изменить порядок элементов

Списки и Кортежи

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

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

Кортежи более эффективны

- Поскольку в Пайтон кортежи являются неизменяемыми, они гораздо проще и эффективнее с точки зрения использования памяти и производительности, чем списки
- Поэтому в программах при создании «временных переменных» мы отдаем предпочтение кортежам, а не спискам

Кортежи и присваивание

- Можно поместить кортеж в левую часть операции присваивания
- Можно опустить круглые скобки:

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

Кортежи и Словари

Метод items() , примененный к словарю, возвращает список кортежей (ключ, значение)

```
>>> d = dict()
>>> d[‘red'] = 2
>>> d[‘green'] = 4
>>> for (a,b) in d.items():
...     print(a, b)
...
red 2
green 4
>>> tups = d.items()
>>> print(tups)
dict_items([(‘red', 2), (‘green', 4)])
```

Кортежи сравнимы

Операторы сравнения работают с кортежами и другими последовательностями. Если первые элементы равны, Пайтон переходит к следующему элементу, и так далее, пока не найдет элементы, которые отличаются.

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

Сортировка списков кортежей

- Подвергнув сортировке список кортежей, мы можем получить отсортированную версию словаря
- Сначала сортируем словарь по ключу, используя метод items(), а затем применяем функцию sorted()

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

Использование sorted()

Можно сделать еще проще: используем встроенную функцию sorted(), которая принимает последовательность в качестве параметра и возвращает отсортированную последовательность

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


>>> for i, j in sorted(d.items()):
...     print(i, j)
...
a 10
b 1
c 22
```

Сортировка по Значениям, а не по Ключам

- Если бы мы могли получить список кортежей в виде - (значение, ключ), то смогли бы отсортировать содержимое по значению
- Мы можем сделать это при помощи цикла for, который создает список кортежей

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

# k - key(ключ)
# v - value(значение)
```

10 самых распространенных слов

```
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)
```

Короткая версия

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

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

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

Генератор списка создает динамический список.
В данном случае мы создаем список инвертированных кортежей (значение, ключ), а затем сортируем его.

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

Резюме

- Синтаксис кортежей
- Неизменяемость
- Сопоставимость
- Сортировка

- Кортежи в операторах присваивания
- Сортировка словарей по ключу или по значению