



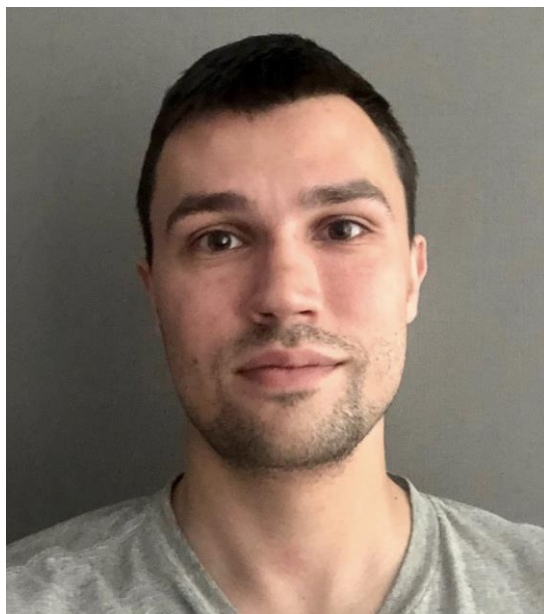
# Python Starter

Множества и кортежи



# Python Starter

## Introduction



Бондаренко Кирилл

Data scientist & Python developer



<https://www.linkedin.com/in/kirill-bond/>



<https://medium.com/@bond.kirill.alexandrovich>



# Python Starter

Тема урока

## Множества и кортежи

# Python Starter

## План урока

1. Понятие множеств
2. Методы множеств
3. Понятие кортежей
4. Методы кортежей

# Python Starter

## Понятие множеств

**Множества** – это структура данных, не содержащая повторяющихся элементов. Множества не поддерживают индексацию, поэтому доступ к элементам возможен перебором.

Синтаксис множеств:

**{1, 4, 2, 3}**

# Python Starter

## Методы множеств

**clear()** – очищает множество.

```
>>> my_set = {1, 4, 2, 3}
>>> my_set.clear()
>>> my_set
set()
```

**pop()** – удаляет первый элемент множества.

```
>>> my_set = {1, 4, 2, 3}
>>> my_set.pop()
1
>>>
```

# Python Starter

## Методы множеств

**discard(elem)** – удаляет элемент из множества.

```
>>> my_set = {1, 4, 2, 3}
>>> my_set.clear()
>>> my_set
set()
```

**remove(elem)** – удаляет элемент из множества. Если такого элемента нет, то выдает ошибку.

```
>>> my_set = {1, 4, 2, 3}
>>> my_set.pop()
1
>>>
```

# Python Starter

## Методы множеств

**add(elem)** – добавляет элемент во множество.

```
>>> my_set = {1, 4, 2, 3}
>>> my_set.add(100)
>>> my_set
{1, 2, 3, 100, 4}
>>> 
```



# Python Starter

## Методы множеств

**union(another\_set)** – объединяет множества.

```
>>> first_set = {1,2,3}
>>> second_set = {4}
>>> first_set.union(second_set)
{1, 2, 3, 4}
>>>
```

**intersection(another\_set)** – находит пересечение двух множеств.

```
>>> first_set = {1,2,3}
>>> second_set = {4, 3}
>>> first_set.intersection(second_set)
{3}
>>>
```

# Python Starter

## Методы множеств

**difference(another\_set)** – разница множеств.

```
>>> first_set = {1,2,3}
>>> second_set = {4, 3}
>>> first_set.difference(second_set)
{1, 2}
>>> 
```

elem **in set\_name** – находится ли элемент в множестве.

```
>>> my_set = {1, 4, 2, 3}
>>> 4 in my_set
True
>>> 
```

# Python Starter

## Понятие кортежей

**Кортеж** (tuple) – это неизменяемый список.

Синтаксис кортежей:

**(1, 2, 3)**

**(1,)**

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

# Python Starter

## Методы кортежей

Кортежам свойственны все методы списков, не изменяющие его. Очень часто встречается случай использования кортежей для того, чтобы поменять элементы местами. Для этого достаточно прописать:

**a, b = b, a**

```
>>> a = 1
>>> b = 2
>>> a, b = b, a
>>>
>>> a
2
>>>
>>> b
1
>>>
```

# Python Starter

## Методы кортежей

Также, при прохождении по списку кортежей, цикл **for** может принимать столько значений, сколько элементов в tuple.

Синтаксис:

**for** elem1, elem2 **in** [(1,2), (2,4)]:

...

```
>>> for elem1, elem2 in [(1,2), (2,4)]:  
...     print(elem1, elem2)  
...  
(1, 2)  
(2, 4)  
>>>
```

# Проверка знаний

TestProvider.com



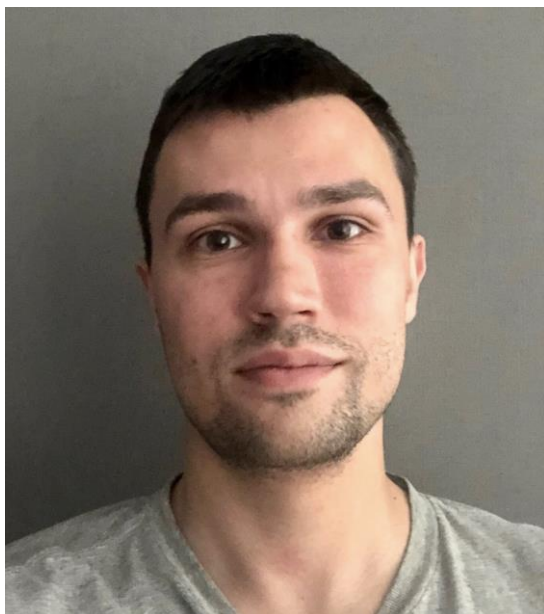
Проверьте как Вы усвоили данный материал на [TestProvider.com](http://TestProvider.com)

TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.

# Python Starter

Спасибо за внимание! До новых встреч!



**Бондаренко Кирилл**

Data scientist & Python developer



# Информационный видеосервис для разработчиков программного обеспечения

