Python SET(Множества)

План

0. Источники

1. Свойства Python коллекций

2. Python Set

3. Генерирование множества (Сode)

4. Получить єлемент множества (СODE)

5. Добовление єлементов Множества (СODE)

6. Удаление єлементов множества (СODE)

7. Копирование множества (СODE)

8. Обновление другими множествами (СODE)

9. Обеденение двух и более множеств (СODE)

10. Пересечение множеств (СODE)

11. Отличие множеств (СODE)

12. Проверка на отличие множеств (СODE)

13. Встроенние методи множеств (СODE)

0. Источники.

https://docs.python.org/3/library/stdtypes.html#sethttps://docs.python.org/3/c-api/set.html

[https://docs.python.org/3/library/stdtypes.html#frozenset](https://docs.python.org/3/library/stdtypes.html" \l "frozenset)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

<https://realpython.com/python-sets/>

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

https://www.programiz.com/python-programming/set

https://pythonworld.ru/tipy-dannyx-v-python/mnozhestva-set-i-frozenset.html

1. Свойства Python коллекций

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| collection data type | ordered | mutable | duplicated | indexed |
| List | + | + | + | + |
| Tuple | + | - | + | + |
| Set | - | + | - | -(but iterated) |
| Dictionary | - | + | - | + |

2. Python Set

Множество в python - "контейнер", содержащий не повторяющиеся элементы в случайном порядке.

Vожества имеет тот же литерал, что и [словарь](https://pythonworld.ru/tipy-dannyx-v-python/slovari-dict-funkcii-i-metody-slovarej.html), но пустое множество с помощью литерала создать нельзя.

Python предоставляет для этого встроенный тип набора. Наборы отличаются от других типов объектов уникальными операциями, которые с ними можно выполнять.

Встроенный в Python тип набора имеет следующие характеристики:

* Наборы неупорядочены. Неупорядоченный означает, что элементы в наборе не имеют определенного порядка. Элементы набора могут появляться в разном порядке каждый раз, когда вы их используете, и на них нельзя ссылаться по индексу или ключу.
* Элементы множества уникальны. Повторяющиеся элементы не допускаются. (СODE)
* Сам набор может быть изменен. После создания набора вы не можете изменять его элементы, но можете добавлять новые элементы.
* Eлементы, содержащиеся в наборе, должны быть неизменного типа - хешируемие. (СODE)

3. Генерирование множества (Сode)

* Use a comma-separated list of elements within braces: {'jack', 'sjoerd'}
* Use a set comprehension: {c for c in 'abracadabra' if c not in 'abc'}
* Use the type constructor: set(), set('foobar'), set(['a', 'b', 'foo'])

4. Получить єлемент множества (СODE)

Вы не можете получить доступ к элементам в наборе, ссылаясь на индекс или ключ.

Но вы можете перебирать элементы набора, используя цикл for, или спрашивать, присутствует ли указанное значение в наборе, используя ключевое слово in.

5. Добовление єлементов Множества (СODE)

6. Удаление єлементов множества (СODE)

7. Копирование множества (СODE)

8. Обновление другими множествами (СODE)

**x1.update(x2[, x3 ...])**

**x1 |= x2 [| x3 ...]**

**x1.intersection\_update(x2[, x3 ...])**

**x1 &= x2 [& x3 ...]**

**x1.difference\_update(x2[, x3 ...])**

**x1 -= x2 [| x3 ...]**

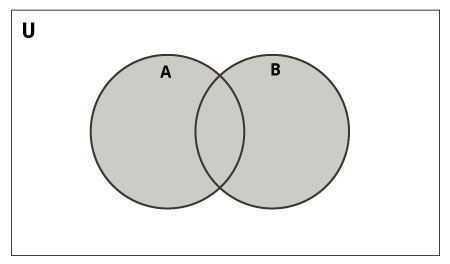
**x1.symmetric\_difference\_update(x2)**

**x1 ^= x2**

9. Обеденение двух и более множеств (СODE)

**x1.union(x2[, x3 ...])**

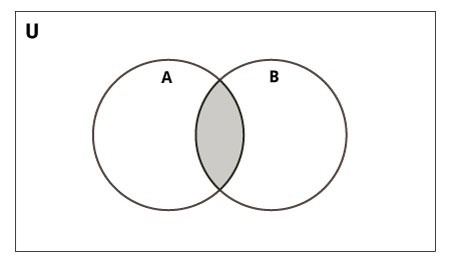
**x1 | x2 [| x3 …]**



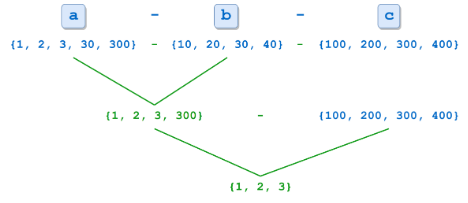
10. Пересечение множеств (СODE)

**x1.intersection(x2[, x3 ...])**

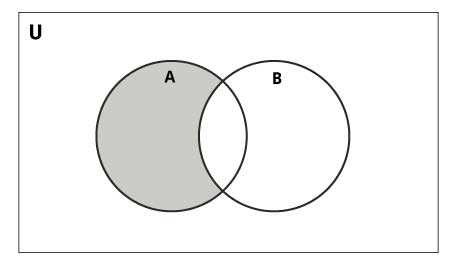
**x1 & x2 [& x3 ...]**

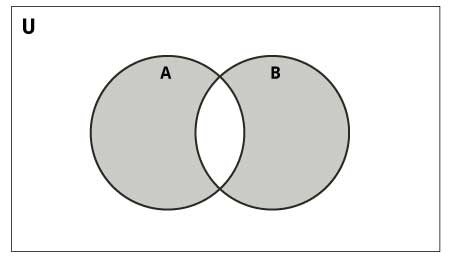


11. Отличие множеств (СODE)

**x1.difference(x2[, x3 ...])**

**x1 - x2 [- x3 ...]**



**x1.symmetric\_difference(x2)**

**x1 ^ x2 [^ x3 ...]**

12. Проверка на отличие множеств (СODE)

**x1.isdisjoint(x2)**

**x1.issubset(x2)**

**x1 <= x2**

**x1.issuperset(x2)**

**x1 >= x2**

13. Встроенние методи множеств (СODE)

* [len()](https://www.programiz.com/python-programming/methods/built-in/len)
* all() метод
* any() метод
* [enumerate()](https://www.programiz.com/python-programming/methods/built-in/enumerate)
* [max()](https://www.programiz.com/python-programming/methods/built-in/max)
* [min()](https://www.programiz.com/python-programming/methods/built-in/min)
* [sorted()](https://www.programiz.com/python-programming/methods/built-in/sorted)
* [sum()](https://www.programiz.com/python-programming/methods/built-in/sum)