



# Python 3 для начинающих

Книги Самоучитель Курсы Категории ↓ Все статьи

## Множества (set и frozenset)



### Онлайн-тренажер Python 3 для начинающих

Теория без воды. Задачи с автоматической проверкой. Подсказки на русском языке. Работает в любом современном браузере.

НАЧАТЬ БЕСПЛАТНО

Доброго времени суток! Сегодня я расскажу о работе с множествами в python, операциях над ними и покажу примеры их применения.

### Что такое множество?

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

Создаём множества:

```
>>> a = set()
>>> a
set()
>>> a = set('hello')
>>> a
{'h', 'o', 'l', 'e'}
>>> a = {'a', 'b', 'c', 'd'}
>>> a
{'b', 'c', 'a', 'd'}
>>> a = {i ** 2 for i in range(10)} # генератор множеств
>>> a
{0, 1, 4, 81, 64, 9, 16, 49, 25, 36}
>>> a = {} # А так нельзя!
>>> type(a)
<class 'dict'>
```

Как видно из примера, множества имеет тот же литерал, что и словарь, но пустое множество с помощью литерала создать нельзя.

Множества удобно использовать для удаления повторяющихся элементов:

```
>>> words = ['hello', 'daddy', 'hello', 'mum']
>>> set(words)
{'hello', 'daddy', 'mum'}
```

Поиск ...

#### Свежее

- Модуль csv - чтение и запись CSV файлов
- Создаём сайт на Django, используя хорошие практики. Часть 1: создаём проект
- Онлайн-обучение Python: сравнение популярных программ

#### Категории

- Книги о Python
- GUI (графический интерфейс пользователя)
- Курсы Python
- Модули
- Новости мира Python
- NumPy
- Обработка данных
- Основы программирования
- Примеры программ
- Типы данных в Python
- Видео
- Python для Web
- Работа для Python-программистов

#### Полезные материалы

- Сделай свой вклад в развитие сайта!
- Самоучитель Python
- Карта сайта
- Отзывы на книги по Python
- Реклама на сайте

С множествами можно выполнять множество операций: находить объединение, пересечение...

- `len(s)` - число элементов в множестве (размер множества).
- `x in s` - принадлежит ли `x` множеству `s`.
- `set.isdisjoint(other)` - истина, если `set` и `other` не имеют общих элементов.
- `set == other` - все элементы `set` принадлежат `other`, все элементы `other` принадлежат `set`.
- `set.issubset(other)` или `set <= other` - все элементы `set` принадлежат `other`.
- `set.issuperset(other)` или `set >= other` - аналогично.
- `set.union(other, ...)` или `set | other | ...` - объединение нескольких множеств.
- `set.intersection(other, ...)` или `set & other & ...` - пересечение.
- `set.difference(other, ...)` или `set - other - ...` - множество из всех элементов `set`, не принадлежащие ни одному из `other`.
- `set.symmetric_difference(other)`; `set ^ other` - множество из элементов, встречающихся в одном множестве, но не встречающиеся в обоих.
- `set.copy()` - копия множества.

И операции, непосредственно изменяющие множество:

- `set.update(other, ...)`; `set |= other | ...` - объединение.
- `set.intersection_update(other, ...)`; `set &= other & ...` - пересечение.
- `set.difference_update(other, ...)`; `set -= other | ...` - вычитание.
- `set.symmetric_difference_update(other)`; `set ^= other` - множество из элементов, встречающихся в одном множестве, но не встречающиеся в обоих.
- `set.add(elem)` - добавляет элемент в множество.
- `set.remove(elem)` - удаляет элемент из множества. `KeyError`, если такого элемента не существует.
- `set.discard(elem)` - удаляет элемент, если он находится в множестве.
- `set.pop()` - удаляет первый элемент из множества. Так как множества не упорядочены, нельзя точно сказать, какой элемент будет первым.
- `set.clear()` - очистка множества.

## frozenset

Единственное отличие `set` от `frozenset` заключается в том, что `set` - изменяемый тип данных, а `frozenset` - нет. Примерно похожая ситуация с [списками](#) и [кортежами](#).

Мы в соцсетях

Подпишись на обновления [по RSS](#) или по почте!

Подписаться!

```
>>> a = set('qwerty')
>>> b = frozenset('qwerty')
>>> a == b
True
>>> True
```

```
True
>>> type(a - b)
<class 'set'>
>>> type(a | b)
<class 'set'>
>>> a.add(1)
>>> b.add(1)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
AttributeError: 'frozenset' object has no attribute 'add'
```

Ошибка в тексте?  
Выделите ее мышкой!  
И нажмите:

Ctrl + Enter

S skillbox.ru РЕКЛАМА · 16+

## Курс «Профессия Data Scientist». 9 проектов в портфолио

Узнать больше



practicum.yandex.ru РЕКЛАМА · 16+ Я

## Станьте специалистом по Data Science на курсе от Яндекса

Обучаем специалистов в Data Science с нуля. 20 часов практики – бесплатно.

Узнать больше

Для вставки кода на Python в комментарий заключайте его в теги `<pre><code>`  
`class="python3">Ваш код</code></pre>`

51 Комментариев

 Войти ▼

Присоединиться к обсуждению...

ВОЙТИ С ПОМОЩЬЮ

ИЛИ ЧЕРЕЗ DISQUS 

Имя



4

Поделиться

Лучшие Новые Старые**Умеренный центрист**

3 месяца назад

Я так понял это как коллекции в JS ?

0 0 Ответить • Поделиться ›

**MisterKoK**

→ Умеренный центрист

месяц назад

Да они даж называются одинаково.  
Основная особенность, что там  
уникаьные элементы. =)

0 0 Ответить • Поделиться ›

**KAnun**

8 месяцев назад

Имба документации

0 0 Ответить • Поделиться ›

**ff ff**

2 года назад

Множество ошибок в статье:

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

В случайном порядке? серьезно?

2. В коде:

```
a = set('hello')
```

```
a
```

```
{'h', 'o', 'l', 'e'} ==> это не верно. Будет выведено:
```

```
{'e', 'h', 'l', 'o'}. Это не случайный порядок. Они
```

распределены так в соответствии с таблицей  
ascii

1 11 Ответить • Поделиться ›

**www578**

→ ff ff

6 месяцев назад

хетел сумничать, не получилось)))

0 0 Ответить • Поделиться ›

**blade**

→ ff ff

2 года назад

вы хоть ЭТОТ код запускали, всегда  
рандомно выдает, тем более без  
привязки к каким-то там таблицам

5 1 Ответить • Поделиться ›

© 2012-2023 Python 3 для начинающих

**Anton**

→ blade

— 🚩

год назад edited

всегда рандомно выдает

```
>>> set("hello")
{'l', 'h', 'e', 'o'}
>>> set("hello")
{'l', 'h', 'e', 'o'}
>>> set("hello")
{'l', 'h', 'e', 'o'}
>>> set("hello")
{'l', 'h', 'e', 'o'}
>>> set("hello")
{'l', 'h', 'e', 'o'}
>>> set("hello")
{'l', 'h', 'e', 'o'}
>>> set("hello")
{'l', 'h', 'e', 'o'}
```

0 2 Ответить • Под

**Виктор  
Попов**

— 🚩

→ Anton

год назад

У тебя же код выдает  
упорядоченные  
объекты, и где ты  
увидел, что всегда  
рандомно выдает?

0 1 Ответит

**Сергей  
Адвокат**

— 🚩

→ Anton

год назад

Проверил, тож всегда  
рендомно

1 0 Ответит

**Виктор  
Попов**

— 🚩

→ Сергей

Адвокат

год назад

Обнови python до  
версии 3.7.

1 0 Ответит

**Товкач  
Богдан**

— 🚩

→ Сергей

Адвокат

год назад

рандомное  
расположения  
относительно  
процеса или  
внесения изменений  
если перезагрузить  
процес, то получим  
снова новые

расположения

1 0 Ответит

**Пух**

2 года назад

что такое set непосредственно? я думал что вместе set идет название множества, но тогда как работает строчка `s1 = set.intersection(*sas)` ?

0 0 Ответить • Поделиться ›

**Emak**

2 года назад

как удалить несколько элементов за раз ?

0 0 Ответить • Поделиться ›

**Lilo KINO**

→ Emak

2 года назад edited

методом for in?

```
a=set('qwerty228')
for x in a.copy():
    _if x=='q' or x==2 or x=='t':
        __a.discard(x)
print(a)
```

1 0 Ответить • Поделиться ›

**Лексий**

3 года назад

Как удалить повторяющиеся элементы без использования `set()`?

0 0 Ответить • Поделиться ›

**Svit Niky**

→ Лексий

3 года назад

Если ты не используешь `set()` из-за невозможности сохранить порядок элементов, то можно сделать так:

```
data = sorted(set(data), key=lambda d:
data.index(d))
```

1 0 Ответить • Поделиться ›

**ID ABSENT**

→ Лексий

3 года назад edited

Отсортируй и если совпадает с прошлым удаляй. Или более медленный метод (при большем числе элементов) - проверяй на количество и удаляй, если больше одного. Метод `count()`.

1 0 Ответить • Поделиться ›

**TruelsFalse**

4 года назад edited

Можно создать пустое множество так:

```
>>emptyset = {1,2}&{3,4}
>>emptyset
set()
```

Или `emptyset = set()`

Верный ли это подход?

0 1 Ответить • Поделиться ›

**Юрий Ерлин**

→ TruelsFalse

7 месяцев назад

я бы использовал более развернутый вариант emptyset =  
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20}&  
{21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40}

1 0 Ответить • Поделиться ›

**evgen**

→ TruelsFalse

2 года назад

первый вариант нужно использовать как можно чаще

0 0 Ответить • Поделиться ›

**illaaa**

→ TruelsFalse

4 года назад

Оба варианта верны, но второй более прост для восприятия

1 0 Ответить • Поделиться ›

**Комуто Херовато**

5 лет назад

Почему не говорится, что метод pop() возвращает удалённый элемент из множества? Иначе, если элемент не был найден, то метод генерирует исключение TypeError.

1 0 Ответить • Поделиться ›

**Artem123**

5 лет назад

Пустое множество можно вполне создать с помощью литерала {}.

0 0 Ответить • Поделиться ›

**Михаил Свечкин**

→ Artem123

4 года назад edited

Тут надо распаковку использовать, тогда да. Иначе – то, что сказал dmitryoboukhov

```
>>> a, b = {}, {}  
>>> a, b  
({}, {})
```

0 0 Ответить • Поделиться ›

**Dmitry Oboukhov**

→ Artem123

4 года назад

это множество будет содержать один элемент - пустой кортеж

0 0 Ответить • Поделиться ›

**Mega Reflex (DigitalLife)**

6 лет назад edited

По умолчанию тип множества set, т.е. по умолчанию они изменяемые. ))))

0 1 Ответить • Поделиться ›

**matsuhito**



6 лет назад edited

на основе чего сравниваются множества  
к примеру `a = {1,2,3} < {2,3,4,5}`, где "а" это False  
почему?

0 0 Ответить • Поделиться ›

**Sour Source** → matsuhito

6 лет назад edited

```
a = {1,2,3} < {2,3,4,5} # a вернёт False
b = a == {2,3,4,5} # b вернёт True
```

1 1 Ответить • Поделиться ›

**13** → matsuhito

6 лет назад edited

Если ты хочешь их сравнить по  
количеству объектов внутри - можно  
вот так:

```
>>> a = set([1,2])
>>> b = set([1,2,3,4])
>>> len(a) < len(b)
True
```

с Хабра:

```
>>> a = set([1,2,3,4])
>>> b = set([3,4,5,6])
>>> a | b # Объединение
{1, 2, 3, 4, 5, 6}
>>> a & b # Пересечение
{3, 4}
>>> a < b # Подмножества
False
>>> a - b # Разница
{1, 2}
>>> a ^ b # Симметрическая разность
{1, 2, 5, 6}
```

2 0 Ответить • Поделиться ›

**matsuhito**

→ 13

6 лет назад

нет просто хотел узнать что  
вот этот конкретный пример  
на основании чего  
сравнивается? почему это  
False `{1,2,3} < {2,3,4,5}` ?

0 0 Ответить • Под

**sergovoy**

matsuhito

6 лет

назад

Наверное потому, что  
эти два множества  
несравнимы

1 0 Ответ

**matsuhito**

sergovoy

6 лет

назад



почему?

17 0 Ответы

**Dzmitry  
Maliuzhenets**

matsuhito

6 лет

назад

Из документации

<https://docs.python.org/3.5...>

set &lt; other

Test whether the set is  
a proper subset of  
other, that is, set <=  
other and set != other.

Множество {1,2,3} не  
является

подмножеством  
множества {2,3,4,5}.

Следовательно,  
результат выражения  
False

5 0 Ответы

**Михаил Максимов**

6 лет назад

что такое other в примерах?

0 0 Ответить • Поделиться ›

**Andre Havryliuk**

→ Михаил Максимов

6 лет назад

Ещё одно множество

0 0 Ответить • Поделиться ›

**example**

7 лет назад edited

можно ли вывести множество в порядке  
возрастания или в порядке убывания, если это  
множество чисел?

1 0 Ответить • Поделиться ›

**Vlad**

→ example

6 лет назад

А мне кажется, что всё-таки можно:

```
a = {1, 19, 2, 45, 54}
a.sort() # AttributeError: 'set' obj
```