



# Множества (set) в python

## Что такое множество?

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

Создаём множества:

```
a = set()
# set()

a = set('hello')
# {'h', 'o', 'l', 'e'}

a = {'a', 'b', 'c', 'd'}
# {'b', 'c', 'a', 'd'}

a = {i ** 2 for i in range(10)} # генератор множеств
# {0, 1, 4, 81, 64, 9, 16, 49, 25, 36}

a = {} # А так нельзя!
# type(a)
# <class 'dict'>
```



Множества удобно использовать для удаления повторяющихся элементов:

```
words = ['hello', 'daddy', 'hello', 'mum']
# {'hello', 'daddy', 'mum'}
```

## Методы set()

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.

## Ссылки по типу данных set

- Ролик на youtube [Хеш-таблица – Самая Популярная Структура Данных](#) понятно объясняющий принцип хешей.

Теги: `python` `programming` `types` `set` `frozenset`

 [Отредактировать эту страницу](#)

*Последнее обновление **4 сент. 2023 г.** от **Stavis***