**МНОЖЕСТВО**

**1. Создание множеств.**

Множество — составной тип данных, представляющий собой несколько значений (элементов множества) под одним именем.

Создание пустого множества:

animals = set()

Создание заполненного множества:

animals = {'cat', 'dog', 'fox', 'elephant'}

Элементы в множестве Python не упорядочены!

При выводе будут выводиться в разной последовательности.

Множество может содержать разные типы данных:

animals\_and\_numbers = {'cat', 5, 'dog', 3, 'fox', 12, 'elephant', 4}

print(animals\_and\_numbers)

Может ли элемент входить в множество несколько раз?

**Важно!**

Итак, у множеств есть три ключевые особенности:

* Порядок элементов в множестве не определен
* Элементы множеств — строки и/или числа
* Множество не может содержать одинаковых элементов

## 2. Операции над множеством

Простейшая операция — **вычисление числа элементов** множества.

Для этого служит функция len.

my\_set = {'a', 'b', 'c'}

n = len(my\_set) # => 3

Очень часто необходимо **обойти все элементы** множества в цикле.

my\_set = {'a', 'b', 'c'}

for elem in my\_set:

print(elem)

Чтобы **проверить наличие элемента** в множестве, можно воспользоваться уже знакомым оператором in:

if elem in my\_set:

print('Элемент есть в множестве')

else:

print('Элемента нет в множестве')

**Добавление элемента** в множество делается при помощи add:

new\_elem = 'e'

my\_set.add(new\_elem)

**Удаление элемента:**

Для этого есть сразу три метода:

**discard** (удалить заданный элемент, если он есть в множестве, и ничего не делать, если его нет),

**remove** (удалить заданный элемент, если он есть, и породить ошибку KeyError, если нет)

**pop**. Метод pop удаляет некоторый элемент из множества и возвращает его как результат. Порядок удаления при этом неизвестен.

my\_set = {'a', 'b', 'c'}

my\_set.discard('a') # Удалён

my\_set.discard('hello') # Не удалён, ошибки нет

my\_set.remove('b') # Удалён

print(my\_set) # В множестве остался один элемент 'c'

my\_set.remove('world') # Не удалён, ошибка KeyError

Метод pop удаляет из множества случайный элемент и возвращает его значение:

my\_set = {'a', 'b', 'c'}

print('до удаления:', my\_set)

elem = my\_set.pop()

print('удалённый элемент:', elem)

print('после удаления:', my\_set)

**Очистить** множество от всех элементов можно методом clear:

my\_set.clear()

## 3. Операции над двумя множествами

Есть четыре операции, которые из двух множеств делают новое множество: объединение, пересечение, разность и симметричная разность.

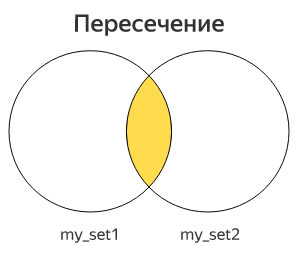


Объединение двух множеств включает в себя все элементы, которые есть хотя бы в одном из них. Для этой операции существует метод union:

union = my\_set1.union(my\_set2)

Или можно использовать оператор |:

union = my\_set1 | my\_set2

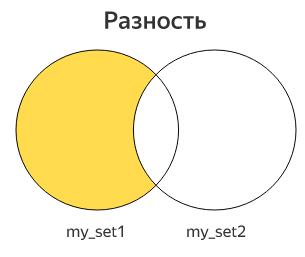


Пересечение двух множеств включает в себя все элементы, которые есть в обоих множествах:

intersection = my\_set1.intersection(my\_set2)

или аналог:

intersection = my\_set1 & my\_set2

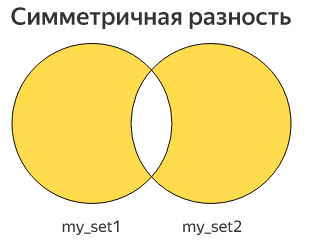


Разность двух множеств включает в себя все элементы, которые есть в первом множестве, но которых нет во втором:

diff = my\_set1.difference(my\_set2)

Или аналог:

diff = my\_set1 - my\_set2



Симметричная разность двух множеств включает в себя все элементы, которые есть только в одном из этих множеств:

symm\_diff = my\_set1.symmetric\_difference(my\_set2)

Или аналогичный вариант:

symm\_diff = my\_set1 ^ my\_set2

## 4. Сравнение множеств

Все операторы сравнения множеств, а именно: ==, <, >, <=, >=, возвращают True, если сравнение истинно, и False — в противном случае.

Множества считаются равными, если они содержат одинаковые наборы элементов.

**Подмножество и надмножество**

Подмножество — некоторая выборка элементов множества, которая может быть как меньше множества, так и совпадать с ним, на что указывают символы «<» и «=» в операторе <=. Наоборот, надмножество включает все элементы некоторого множества и, возможно, какие-то еще.

s1 = {'a', 'b', 'c'}

print(s1 <= s1) # True

s2 = {'a', 'b'}

print(s2 <= s1) # True

s3 = {'a'}

print(s3 <= s1) # True

s4 = {'a', 'z'}

print(s4 <= s1) # False

**Решение задач на множества:**

1. **Поездка.**

Саша едет на автобусе и, чтобы развеять скуку, придумал себе игру. Он смотрит в окно и записывает все номера домов на тетрадный листок. На середине пути он берёт новый листок и продолжает записывать номера. В конце пути он хочет узнать, какие номера встретились ему в обоих списках, и выписывает их на третий листок. Если таких номеров нет, то Саша записывает слово EMPTY.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| 10  23  31  5  10  49  31 | 10  31 |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| 1  2  3  10  20  30  40 | EMPTY |

### Пример 3

| Ввод | Вывод |
| --- | --- |
| 5  9  7  10  5  4  7  9 | 5  7  9 |

1. **Города.**

Ира и Оля играют в города. Они очень любят эту игру, знают много городов и к концу игры забывают, какие уже называли. На вас возложена почётная задача вести запись игры и напоминать девочкам, если какой-то город уже был назван.

Выходные данные:

В первой строке записано число названных городов N. Затем идут N строк с названиями городов и ещё одна строка с новым только что названым городом.

Входные данные:

Слово OK, если такого города ещё не было названо, и TRY ANOTHER, если город уже был назван.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| 3  Москва  Нью-Йорк  Лондон  Париж | OK |

### Пример 1

| Ввод | Вывод |
| --- | --- |
| 8  Лондон  Париж  Москва  Вашингтон  Берлин  Вена  Мадрид  Рим  Мадрид | TRY ANOTHER |

1. **Языки.**

Каждый ученик в классе изучает либо английский, либо немецкий, либо оба этих языка.  
У классного руководителя есть списки учеников, изучающих каждый из языков. Напишите программу, которая позволит классному руководителю быстро выяснить, сколько учеников изучает только один язык.

Входные данные:

В первых двух строках указывается количество учеников, изучающих английский и немецкий языки (M и N). Затем идут M строк — фамилии учеников, которые изучают английский язык; и N строк с фамилиями учеников, изучающих немецкий. Гарантируется, что среди учеников нет однофамильцев.

Выходные данные:

Количество учеников, которые изучают только один язык. Если таких не окажется, в строке вывода нужно написать NO.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| 3  2  Иванов  Петров  Васечкин  Иванов  Михайлов | 3 |

1. **Языки в.2**

Каждый ученик в классе изучает либо английский, либо немецкий, либо оба этих языка. У классного руководителя были списки учеников, изучающих каждый из языков. Для проведения классного мероприятия он напечатал списки и разрезал их пофамильно и разложил полученные листочки две отдельные кучки. Вдруг из окна подул ветер и перемешал листочки в обеих кучах. Помогите классному руководителю выяснить, сколько учеников в классе изучают только один язык.

Входные данные:

В первых двух строках указывается количество учеников, изучающих английский и немецкий языки (M и N). Затем идут M+N строк с фамилиями учеников в произвольном порядке. Гарантируется, что среди учеников нет однофамильцев.

Выходные данные:

Количество учеников, которые изучают только один язык. Если таких не окажется, в строке вывода нужно написать NO.

### Пример 1

| Ввод | Вывод |
| --- | --- |
| 3  2  Иванов  Петров  Васечкин  Иванов  Михайлов | 3 |

### Пример 2

| Ввод | Вывод |
| --- | --- |
| 3  3  Иванов  Петров  Васечкин  Иванов  Петров  Васечкин | NO |