Множества

Вы уже знаете, что...

Когда в Python какой-то тип хранит в себе несколько чего-нибудь, то его называют **коллекцией.** Например – строка.

Если элементы коллекции стоят в каком-то определённом порядке, то такую коллекцию называют **упорядоченной.** Например – строка,

У упорядоченной коллекции каждый элемент имеет свой номер – **индекс.** С помощью индекса можно обращаться к элементам коллекции.

Это упорядоченные коллекции:

```
s = 'Лицей БГУ'
numbers = [10, 20, 30, 40, 50]
what_is_that = (s, numbers)
```

Множество – это коллекция

Множество – это типа такой мешок. В него можно:

– положить что-то, чтобы оно там лежало

– убрать что-то, чтобы оно там не лежало

– посмотреть, лежит ли там что-то определённое

Множество в Python имеет тип `set`

```
vowels = {'a', 'e', 'i', 'o', 'u', 'y'}     # так можно создать множество элементов
empty_set = set()                           # так можно создать пустое множество

print(vowels)                       # {'e', 'u', 'i', 'a', 'o', 'y'}
print(empty_set)                    # set()
print(len(vowels), len(empty_set))  # 6 0
```

Множество – это коллекция

Множество можно получить из другой коллекции с помощью функции `set()`

```
word = "коллекция"
letters = set(word)
print(letters)  # {'о', 'ц', 'я', 'к', 'е', 'л', 'и’}
```

Обратите внимание:

– Сет хранит только уникальные элементы

– Порядок элементов в сете не определён
(при каждом запуске этого кода, эти символы будут печататься в другом порядке)

Множество – это коллекция

– С помощью оператора in можно проверить, содержится ли элемент в сете.

– По элементам сета можно пройтись циклом for.

```
vowels = {'a', 'e', 'i', 'o', 'u', 'y'}
letter = input("Введите латинскую букву: ")     # Введите латинскую букву: L
if letter.lower() in vowels:
    print("Гласная буква")
else:
    print("Согласная буква")                    # Согласная буква


for letter in vowels:
    print(letter, end='')       # eyiauo
```

Методы множества

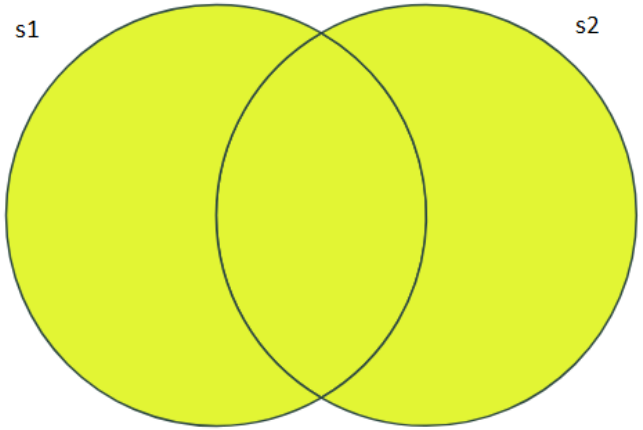
Метод	Описание	Пример	Вывод print(s)
set.add(e)	**Добавить** элемент во множество	s = set() s.add(1)	{1}
set.remove(e)	**Удалить** элемент множества.	s = {1, 2, 3} s.remove(2)	{1, 3}
set.discard(e)	**Удалить** элемент, если он **принадлежит** множеству	s = {1, 2, 3} s.discard(2)	{1, 3}
set.pop()	**Вернуть** и **удалить** произвольный элемент множества	s = {1, 2, 3} x = s.pop()	{1, 3}
set.clear()	**Очистить** множество, удалив все его элементы	s = {1, 2, 3} s.clear()	set()

Множество хорошо подходит в условиях, когда:

- вы работаете с **уникальными элементами**
- вы планируете **добавлять** и **удалять** элементы из коллекции
- порядок элементов не важен
- важна информация, **присутствует ли** что-то в множестве или нет, но не важно в каком конкретно месте

Операции над множествами

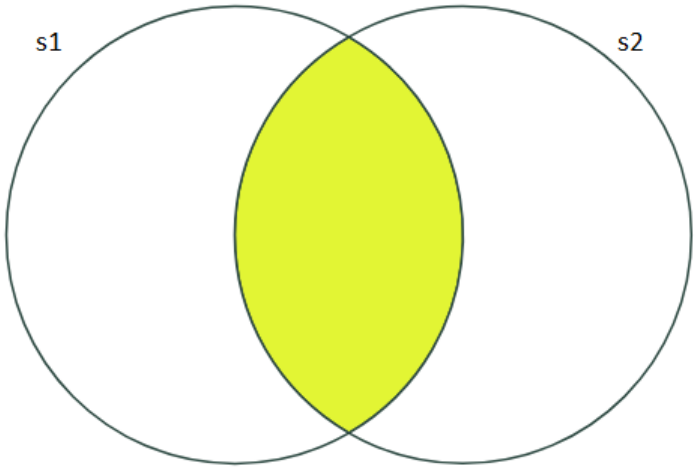
Объединение множеств. Возвращает множество, состоящее из элементов всех объединяемых множеств. Обозначается | либо union(...)




```
s1 = {1, 2, 3}
s2 = {3, 4, 5}
s_union = s1 | s2
s_union = s1.union(s2)  # второй способ
print(s_union)  # {1, 2, 3, 4, 5}
```

Операции над множествами

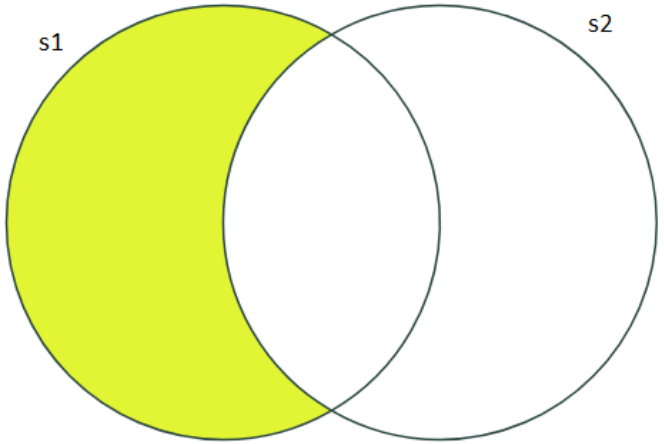
Пересечение множеств. Возвращает множество, состоящее из общих элементов пересекаемых множеств. Обозначается & либо intersection(...)




```
s1 = {1, 2, 3}
s2 = {3, 4, 5}
s_intersection = s1 & s2
s_intersection = s1.intersection(s2)    # второй способ
print(s_intersection)   # {3}
```

Операции над множествами

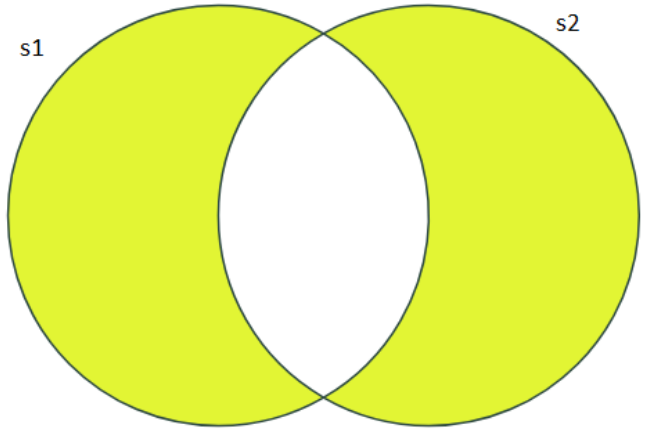
Разность множеств. Возвращает множество, состоящее из элементов первого множества, которые не встречаются во втором. Обозначается - либо difference(...)




```
s1 = {1, 2, 3}
s2 = {3, 4, 5}
s_dif = s1 - s2
s_dif = s1.difference(s2)   # второй способ
print(s_dif)    # {1, 2}
```

Операции над множествами

Симметрическая разность множеств. Возвращает множество, состоящее из элементов первого множества, которые не встречаются во втором. Обозначается ^ либо symmetric_difference(...)




```
s1 = {1, 2, 3}
s2 = {3, 4, 5}
s_sym_dif = s1 ^ s2
s_sym_dif = s1.symmetric_difference(s2)     # второй способ
print(s_sym_dif)    # {1, 2, 4, 5}
```

Пример

Определить, какие гласные буквы встречаются в строке 'Лицей БГУ'

```
vowels = set('уеыаоэяиюё')
s = input('Введите строку: ')                    # Введите строку: Лицей БГУ
print('Гласные буквы:', set(s.lower()) & vowels)  # {'и', 'у', 'е’}
```

более умный способ:

```
print('Гласные буквы: ' + ', '.join(set(s.lower()) & vowels))     # и, у, е
```