алгоритмика

Модуль 7. Урок 2.

Множества

Методические указания

Во время урока откройте презентацию в режиме Просмотр (ctrl+enter)**.**

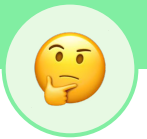
Так ученики увидят только свои слайды. В презентации есть методические слайды, которые нужны только преподавателю. Они отмечены иконкой «глаз».

Новая тема

Задача:

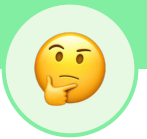
написать программу, которая хранит список различных книг, прочитанных одним классом.

Новая тема

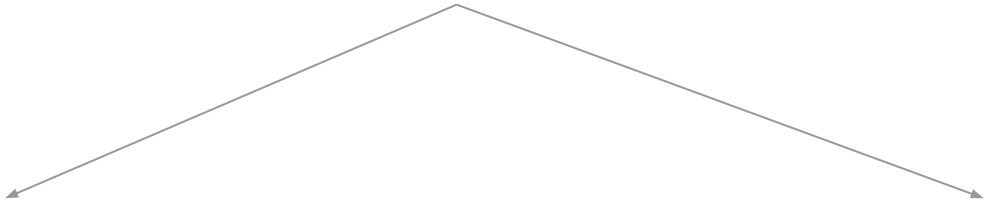


С помощью каких инструментов Python можно запомнить прочитанные книги?

Новая тема



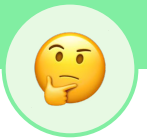
Хранение прочитанных книг



Список книг

Словарь с книгами

Новая тема



По какому алгоритму будет работать программа, хранящая прочитанные книги в списке?

Новая тема

Возможная схема программы со списками:

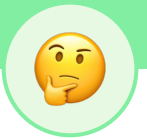
- *Пользователь вводит название книги.*
- *Программа просматривает список, есть ли книга с таким названием.*
- *Если такой книги нет, то она добавляется в список.*

Новая тема

Возможная схема программы со словарями:

- *Пользователь вводит название книги.*
- *Программа сразу определяет, есть ли книга в словаре среди ключей (оператор in).*
- *Если такой книги нет, то создаётся новая пара «ключ-элемент».*

Новая тема



В чём преимущества и недостатки каждого способа?

Новая тема

Программа со списками:

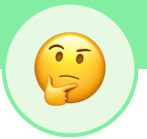
книги могут дублироваться (неэкономный расход памяти).

Новая тема

Программа со словарями:

**нужно хранить лишние данные,
кроме названия книги
(не эффективно).**

Новая тема



🙄

Какая структура данных решила бы задачу наиболее эффективно?

Новая тема

Для базы данных книг нам нужна:

структура данных,

хранящая элементы только по 1 разу,

порядок расположения элементов и их номера не важны.

Такой инструмент есть в Python!

Определение

Множество —
это неупорядоченный набор элементов.

Каждый элемент встречается в множестве только **1 раз.**

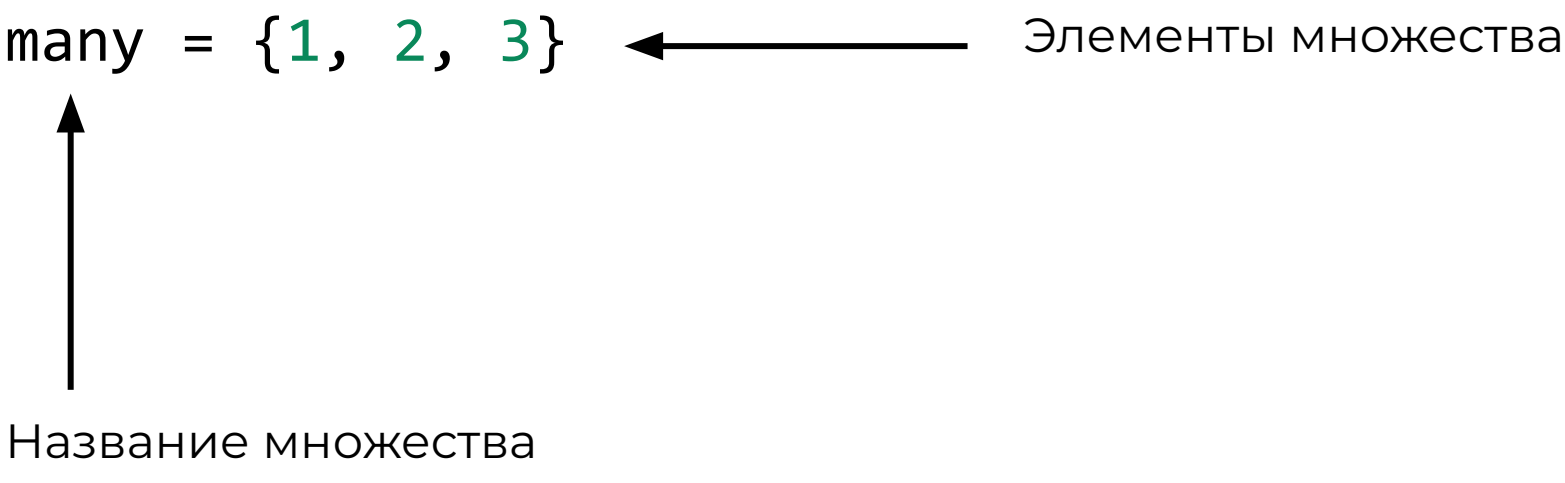
Создание множества

```
many = set()
```

Команда, создающая пустое множество

Название множества

Создание множества



Пример:

Программа

```
many = {1, 2, 3, 3, 2, 1}
print(many)
```

Вывод

{1, 2, 3}

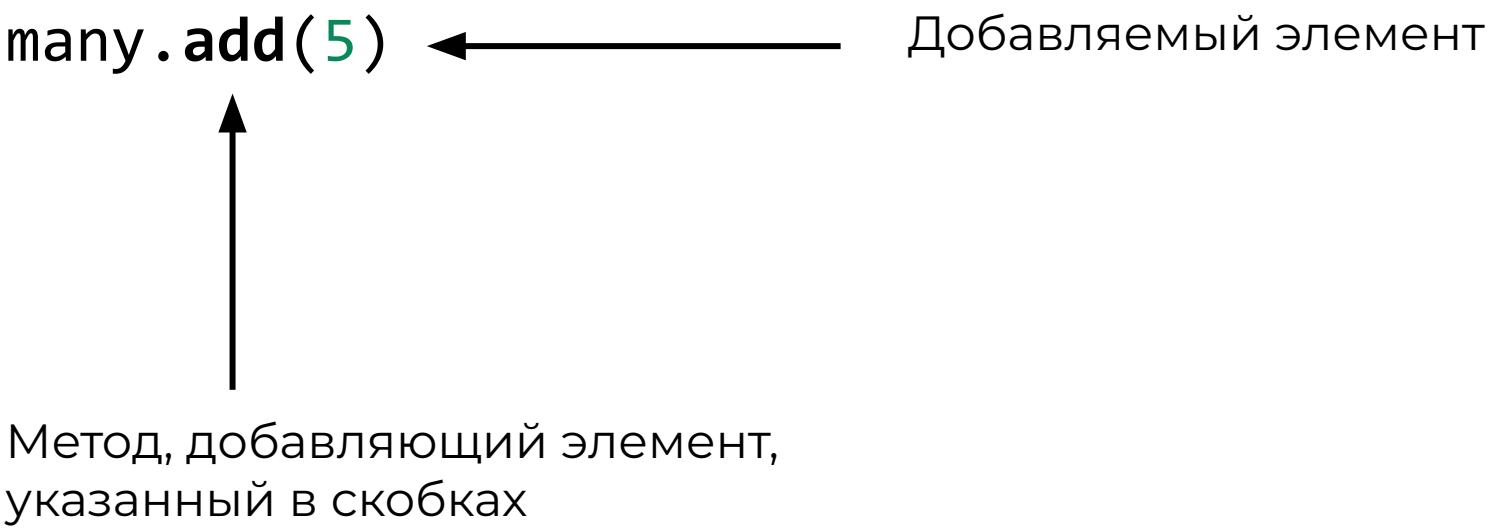
Элементы множества не повторяются

Вывод

Множества удобно использовать, если:

- 😉 **важно знать входит элемент во множество или нет;**
- 💥 **другие свойства элемента (например, порядковый номер) не важны.**

Добавление элемента во множество



Пример:

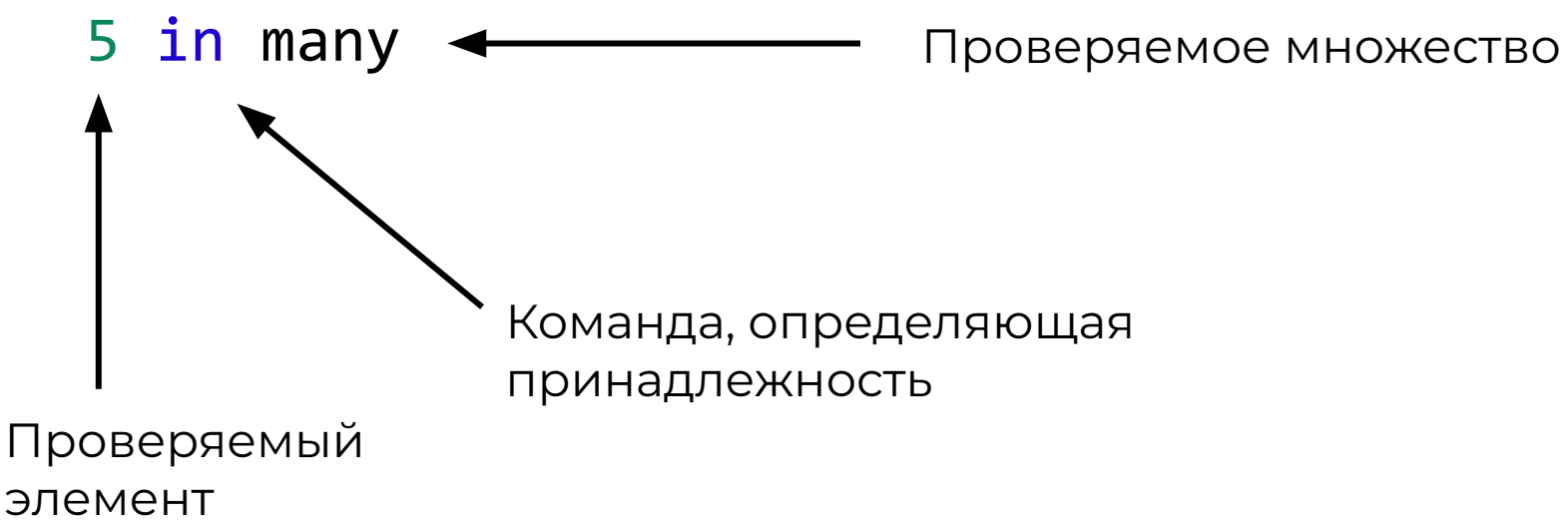
Программа

```
many = {1, 2, 3}
many.add(5)
print(many)
```

Вывод

```
{1, 2, 3, 5}
```

Принадлежит ли элемент множеству



Пример:

Программа

```
many = {1, 2, 3, 5}
if 5 in many:
    print(“Yes!”)
```

Вывод

```
Yes
```

Демонстрация

Примеры и программа, хранящая названия прочитанных книг.

Работаем на платформе

1 - заходим
на платформу.

2 - выбираем
задание
«Множества:
задачи».

Перерыв

Повторение

Делимся на команды по 3 или 4 человека.

Каждая команда выбирает капитана.

Ход игры:

🙄 **Команда читает вопрос на экране.**

☝️ **Команда придумывает ответ (2 минуты).**

🗽 **Капитан записывает ответ на листе.**

Что такое список?

Назови 2 любых отличия списка от множества.

Какая это структура данных?

```
phones={‘Ваня’: 9067345, ‘Коля’: 916224}
```

Что делает метод `add()`?

Что делает оператор `in`?

Что напечатает программа?

```
numbers = [1, 2, 3, 0]
print(numbers[0])
numbers.remove(1)
print(numbers)
```

Повторение

Сдай лист с ответом учителю.

Получи лист другой команды.

Отметь правильные ответы и посчитай баллы.

А теперь ответы!

Список — это упорядоченный набор элементов.

Повторение

- **Элементы в списке упорядочены, во множестве нет.**
- **Каждый элемент списка имеет номер (индекс), а во множестве нет.**
- **Элементы в списке могут повторяться.**

Это словарь

```
phones={‘Ваня’: 9067345, ‘Коля’: 916224}
```

Ключ

Значение

Метод add() **добавляет элемент в множество.**

Оператор `in` определяет, входит ли элемент в словарь или множество.

Повторение

Программа:

```
numbers = [1, 2, 3, 0]
print(numbers[0])
numbers.remove(1)
print(numbers)
```

Вывод:

```
1
[2, 3, 0]
```

Подведём итоги

Работаем на платформе

1 - заходим
на платформу.

2 - выбираем
задание
«Структуры данных:
задачи».

До скорых встреч!