алгоритмика

Международная школа
программирования для детей

Модуль 2. Урок 6.

Словари и множества

алгоритмика

Повторим

Сегодня на занятии:

- Словари и множества: сложно звучит — легко запоминается.
- Поиск элементов по ключам (нет, не дверным).
- Проверка принадлежности элемента множеству.
- Словари и множества, а в чём отличия от списков?

Демонстрация

(перебор элементов по индексам)

алгоритмика

Что делать, если нужны другие индексы?

Ключ —

уникальный идентификатор, с помощью которого можно получить доступ к конкретному элементу словаря.

Формат записи элементов в словарь

451: '451 градус по Фаренгейту'

Уникальный ключ (идентификатор) элемента

Значение, привязанное к ключу

Доступ к элементу через ключ

```
dictionary = {451: '451 градус по Фаренгейту',
20000: 'Двадцать тысяч льё под водой',
10: 'Собачье сердце', 1840: 'Герой нашего
времени', 12: 'Алиса в стране чудес'}
print(dictionary[451])
```

Ключ элемента

Обращение к элементу по ключу

Вывод:

```
'451 градус по Фаренгейту'
```

Вывод элемента, соответствующего ключу

Формат команды для добавления элемента в заполненный словарь

```
dictionary[1] = 'one'
```

Добавление элемента в заполненный словарь

```
dictionary = {451: '451 градус по Фаренгейту',
20000: 'Двадцать тысяч льё под водой', 10:
'Собачье сердце', 1840: 'Герой нашего
времени', 12: 'Алиса в стране чудес'}
dictionary[1836] = 'Ревизор'
```

Имеющиеся в словаре элементы

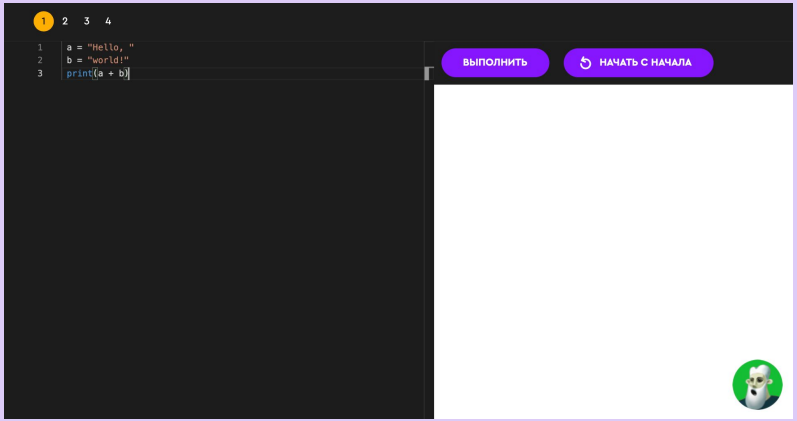
Ключ добавляемого элемента

Значение

Добавление элемента в словарь

Заходим на платформу

Словари



Задание на платформе

Итог первой половины урока

алгоритмика

Давайте отдохнём!

Демонстрация

(множество)

Свойство множества — уникальность элементов

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

Записано 6 элементов, 3 из которых дублируются

Вывод:

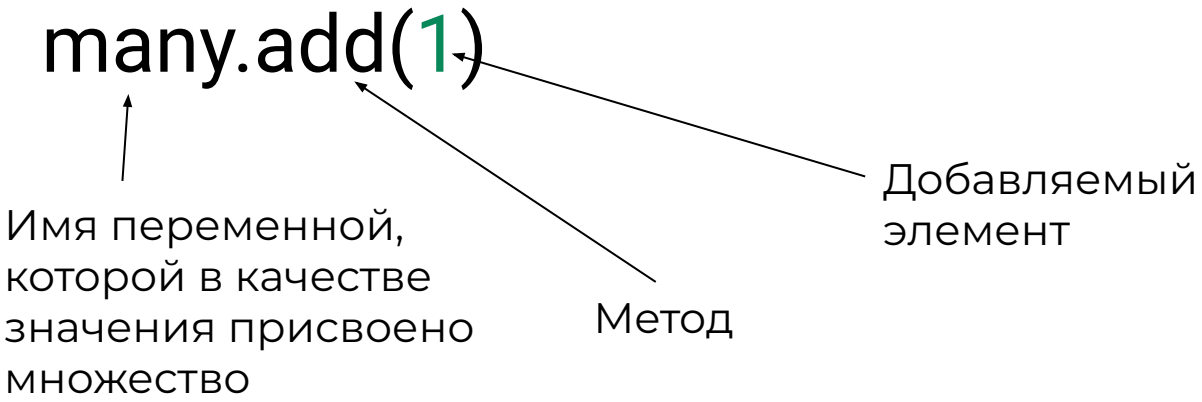
```
{1, 2, 3}
```

Выводится 3 элемента, без дубликатов

Демонстрация

(уникальность элементов в множестве (строки))

Формат записи команды для добавления элемента в множество



Демонстрация

(создание пустого множества и добавление элементов)

Демонстрация

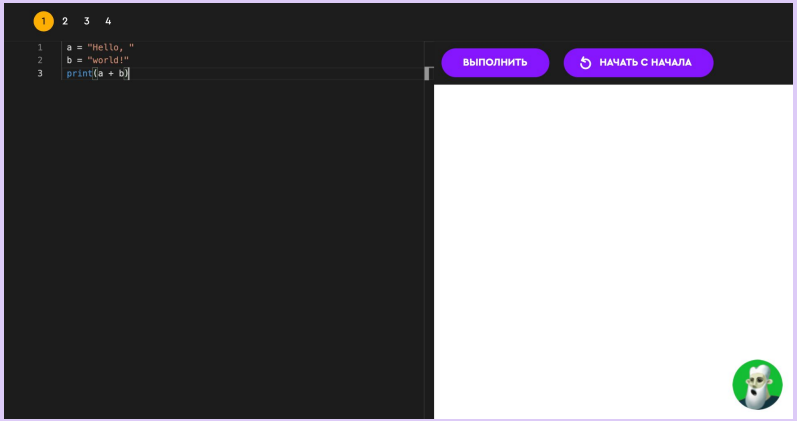
(создание пустого множества и добавление элементов (правильная программа))

Проверка принадлежности элемента множеству

```
i in many
```

Заходим на платформу

Множества



Задание на платформе

алгоритмика

Как прошло занятие?

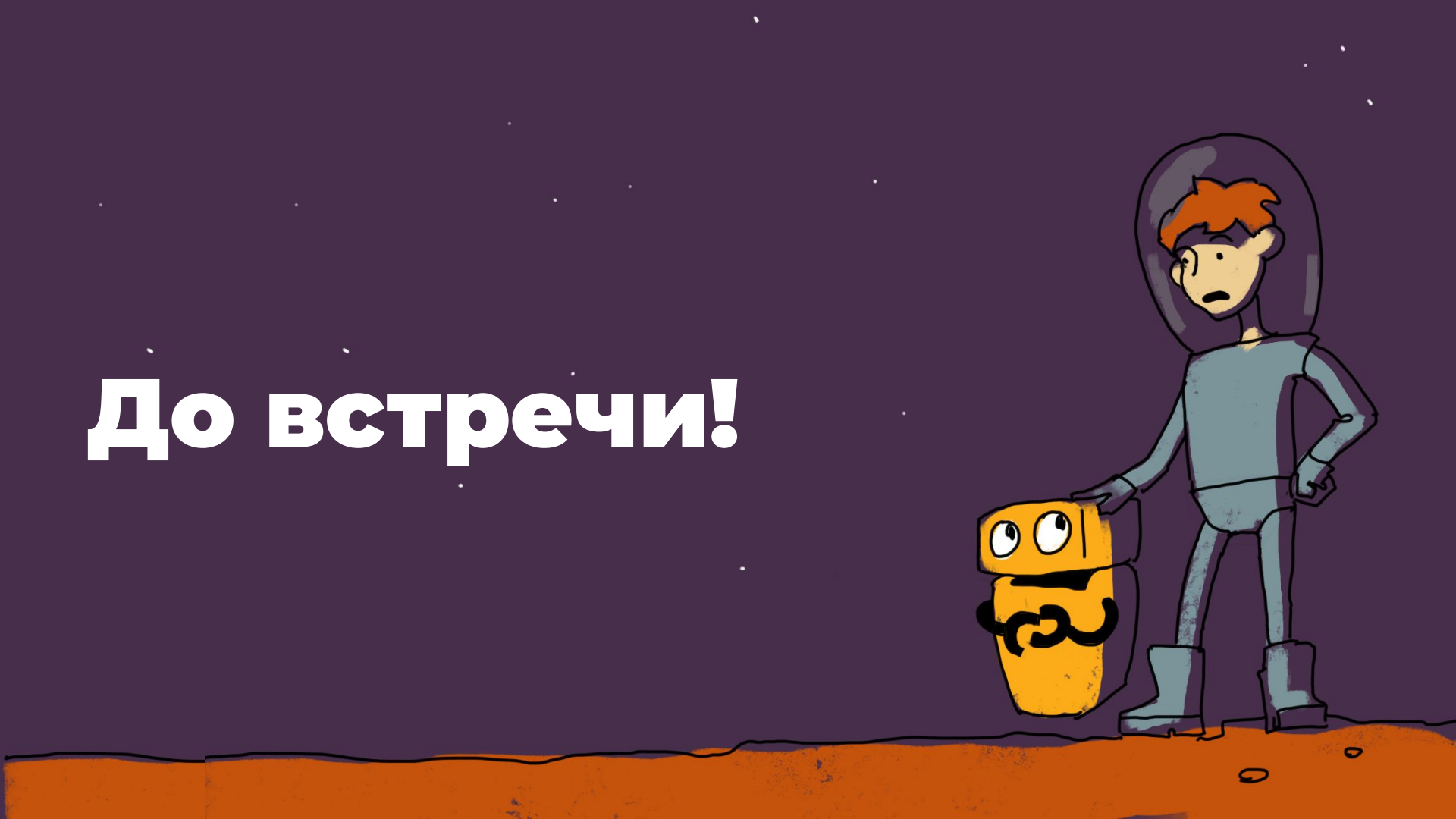
Проверь себя

- Что такое словарь?
- Что такое множество?
- Что такое ключ в словаре?
- Что означает уникальность элементов в множестве?
- Как добавить элементы в словарь?
- Как добавить элементы в множество?
- Как проверить принадлежность элемента множеству?

На следующем занятии:

- Проект «Шифр» — много раз решали задачи, пора испробовать самим!

До встречи!



Операции с множествами (1 часть)

A.union(B)	Возвращает множество, являющееся объединением множеств **A** и **B**.
A.update(B)	Добавляет в множество **A** все элементы из множества **B**.
A.intersection(B)	Возвращает множество, являющееся пересечением множеств **A** и **B**.
A.intersection_update(B)	Оставляет в множестве **A** только те элементы, которые есть в множестве **B**.
A.difference(B)	Возвращает элементы, входящие в **A**, но не входящие в **B**.
A.difference_update(B)	Удаляет из множества **A** все элементы, входящие в **B**.

Теория

Операции с множествами (2 часть)

A.symmetric_difference(B)	Возвращает элементы, входящие в **A** или в **B**, но не в оба из них одновременно.
A.issubset(B)	Возвращает true, если **A** является подмножеством **B**.
A.issuperset(B)	Возвращает true, если **B** является подмножеством **A**.
A < B	Эквивалентно **A <= B** and **A != B**
A > B	Эквивалентно **A >= B** and **A != B**

Теория

Быстрый поиск элемента в словаре

Как вы уже знаете, в словаре всегда хранятся значения, которым присвоен уникальный ключ. По нему мы можем проверить, входит ли элемент в словарь или нет. А для этого программисты придумали специальный оператор - in.

Оператор in —

оператор для работы с ключами в словаре. Проверяет наличие значения в словаре по его ключу (если ключ есть в словаре, значит и значение есть в словаре - оператор возвращает True, иначе - False).

Оператор in - применение

Вернёт True

```
dictionary = {1: "one", 2: "two", 3: "three"}
print(1 in dictionary)
```

Вернёт False

```
dictionary = {1: "one", 2: "two", 3: "three"}
print(4 in dictionary)
```