

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)

**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])
```

Вывод:

'451 градус по Фаренгейту'

Ключ элемента

Обращение к
элементу по
ключу

Вывод элемента,
соответствующего
ключу

Формат команды для добавления элемента в заполненный словарь

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

Имя переменной,
которой, в качестве
значения, присвоен
словарь

Уникальный ключ

Значение

Добавление элемента в заполненный словарь

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

← Имеющиеся в
словаре
элементы

```
dictionary[1836] = 'Ревизор'
```

↑
Ключ добавляемого
элемента

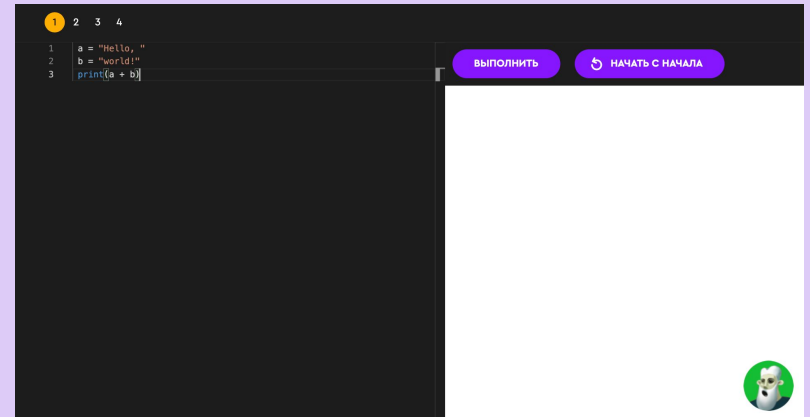
↑
Значение

← Добавление элемента в
словарь

Заходим на платформу



Словари



The image shows a code editor window with a dark theme. The code is as follows:

```
1 a = "Hello, "  
2 b = "world!"  
3 print(a + b)
```

At the top of the editor, there are four numbered tabs: 1 (highlighted in yellow), 2, 3, and 4. On the right side of the editor, there are two buttons: "ВЫПОЛНИТЬ" (Execute) and "НАЧАТЬ С НАЧАЛА" (Start from the beginning). In the bottom right corner of the editor, there is a small circular icon of a white robot head.

Задание на платформе

Итог первой половины урока



алгоритмика

Давайте отдохнём!

Демонстрация

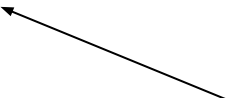
(множество)

Свойство множества — уникальность элементов

```
many = {1, 2, 3, 1, 2, 3}
```

```
print(many)
```

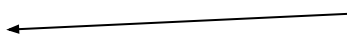
Записано 6 элементов, 3 из
которых дублируются



Вывод:

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

Выводится 3 элемента, без
дубликатов



Демонстрация

(уникальность элементов в множестве (строки))

Формат записи команды для добавления элемента в множество

`many.add(1)`

↑
Имя переменной,
которой в качестве
значения присвоено
множество

Метод

Добавляемый
элемент

Демонстрация

(создание пустого множества и добавление элементов)

Демонстрация

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

Проверка принадлежности элемента множеству

`i in many`

Проверяемый
элемент

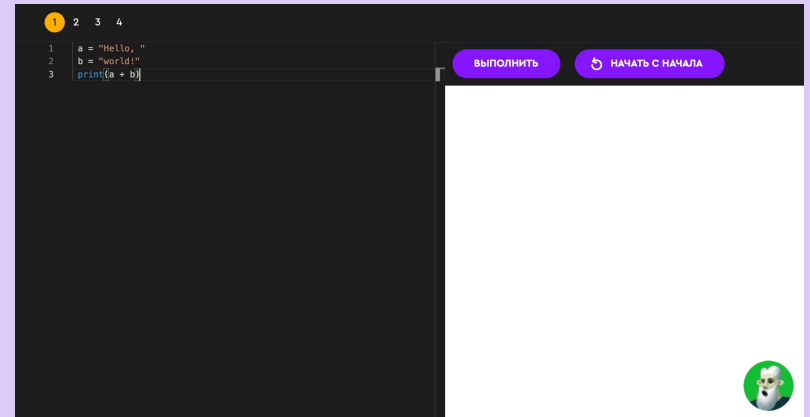
Ключевое
слово

Имя переменной, которой в качестве значения
присвоено множество

Заходим на платформу



Множества



The image shows a code editor interface with a dark theme. The code is as follows:

```
1 a = "Hello, "  
2 b = "world!"  
3 print(a + b)
```

At the top of the editor, there are four tabs labeled 1, 2, 3, and 4. Tab 1 is active and highlighted with a yellow dot. On the right side of the editor, there are two buttons: "ВЫПОЛНИТЬ" (Execute) and "НАЧАТЬ С НАЧАЛА" (Start from the beginning). The right side of the editor is currently blank, indicating that the code has not been executed yet. In the bottom right corner of the editor, there is a small circular icon of a white robot head on a green background.

Задание на платформе

алгоритмика

Как прошло занятие?

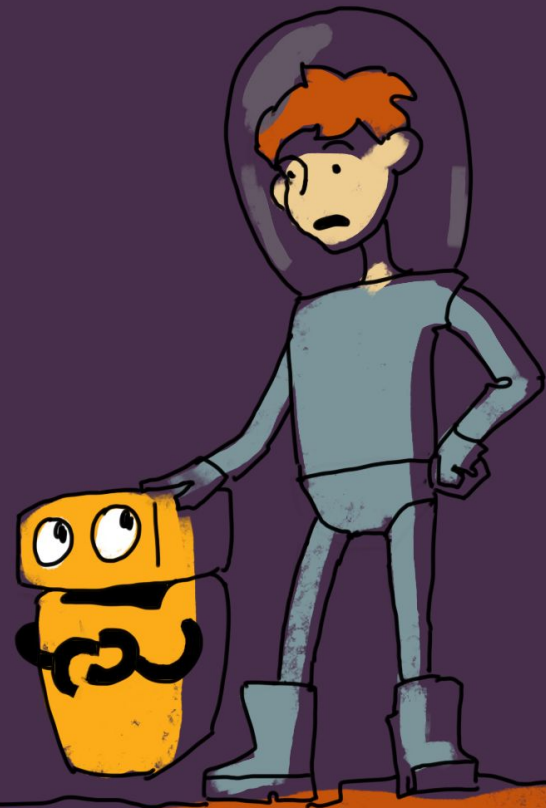
Проверь себя

- 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 .

Операции с множествами

(2 часть)

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

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)
```