



Программирование  
на Python

Презентация занятия

# Изучение возможностей и синтаксиса Python: Контейнеры. Часть 2.

9 занятие



**инжинириум**<sup>®</sup>

МГТУ им. Н.Э. Баумана

2019

# СОДЕРЖАНИЕ

## 1. ВВЕДЕНИЕ. ОРГАНИЗАЦИОННАЯ ИНФОРМАЦИЯ

- Тема занятия
- Цели и задачи занятия
- Результаты занятия
- Материалы для преподавателя
- Материалы для ученика
- Тайминг проведения занятия

## 2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

- Ассоциативные массивы
- Особенности ассоциативных массивов
- Структура множества

## 3. ПРАКТИЧЕСКАЯ ЧАСТЬ

- Инициализация словаря
- Методы обработки словарей
- Неупорядоченные контейнеры
- Сортировка словаря



## Тема: Изучение возможностей и синтаксиса Python: Контейнеры. Часть 2.

### Хеш-таблица

#### 1.1 Структура данных - ассоциативный массив

Ассоциативный массив - структура данных, похожая на массив, отличающаяся отсутствием порядка (упорядоченных индексов) с произвольной индексацией.

#### 1.2 Для чего он нам нужен?

Ассоциативный массив - самая удобная вариация хранилища.

#### 1.3 В чем его особенность

Особенностями ассоциативных массивов является их неупорядоченность и , как следствие, несортируемость.



## Тема: Изучение возможностей и синтаксиса Python: Контейнеры. Часть 2.

Словарь — неупорядоченная структура данных, которая позволяет хранить пары «ключ — значение». Ключом может являться в принципе любой неизменяемый тип данных. Значением же конкретного ключа может быть что угодно.  
Создание словаря

```
dictionary = {'персона': 'человек',  
             'марафон': 'гонка бегунов длиной около 26 миль',  
             'бежать': 'двигаться со скоростью'}  
d = dict(short='dict', long='dictionary')  
d = dict.fromkeys(['a', 'b'], 100)
```



## Тема: Изучение возможностей и синтаксиса Python: Контейнеры. Часть 2.

### Редактирование словаря

Для получения значения конкретного ключа используются квадратные скобки [].

```
d = {1: 2, 2: 4, 3: 9}
```

```
d[1] = 4
```

```
# {1: 4, 2: 4, 3: 9}
```

```
d[4] = 4 ** 2
```

```
d['1']
```

Traceback (most recent call last):

```
File "", line 1, in
```

```
  d['1']
```

```
KeyError: '1'
```



## Тема: Изучение возможностей и синтаксиса Python: Контейнеры. Часть 2.

### Методы словаря

`dict.items()` - возвращает пары (ключ, значение).

`dictionary.items()`

[('персона', 'человек'),

('марафон', 'гонка бегунов длиной около 26 миль'),

('бежать', 'двигаться со скоростью')]

`dict.keys()` - возвращает ключи в словаре.

`dictionary.keys()`

['персонаж', 'марафон', 'бежать']

`dict.values()` - возвращает значения в словаре.



## Тема: Изучение возможностей и синтаксиса Python: Контейнеры. Часть 2.

Итерация по словарю

Вы можете провести итерацию по каждому ключу в словаре.

```
for key in story_count:
```

```
    print(key)
```

В примере кода ниже цикл `for` использует метод `items()` для получения пары «ключ — значение» на каждую итерацию.

```
for key, value in dictionary.items():
```

```
    print(key, value)
```

```
('персона', 'человек')
```

```
('марафон', 'гонка бегунов длиной около 26 миль')
```

```
('бежать', 'двигаться со скоростью')
```



## Тема: Изучение возможностей и синтаксиса Python: Контейнеры. Часть 2.

Общее о контейнерах

- 1 Список - это массив, изменяемый, упорядоченный.
- 2 Кортеж - это массив, неизменяемый, упорядоченный.
- 3 Множество - это объект, изменяемый, неупорядоченный, хранит только уникальные значения.
- 4 Словарь - это объект, изменяемый, неупорядоченный, хранит множество пар ключ:значение.

В чем особенность множества

Множества в Python - объект, хранящий исключительно уникальные значения элементов, добавленных в множество.

