

Лекция 8

Введение в структуры данных

Коллекции. Общие сведения

Коллекции - это специальные хранилища объектов.

Коллекции предоставляют множество методов для удобной работы:

- Добавление нового объекта в коллекцию
- Удаление объекта из коллекции
- Сортировка объектов в коллекции
- Обращение к объекту коллекции
- ...

а также методы, специфичные для типа коллекции.

Работа с коллекциями

- Количество элементов: `len()`
- Операторы `in`, `not in`
- Обход коллекции в цикле `for`
- `min()`, `max()` (элементы коллекции должны быть одного типа!)
- `sorted()`

Списки, массивы, связанные списки

- **Массив** - это упорядоченная коллекция объектов одинакового типа.
- **Список** - это упорядоченная коллекция объектов любого типа.
- Список в языке Python реализован на основе динамических массивов языка Си.
- **Связный список** - коллекция элементов, каждый из которых содержит как собственные данные, так и одну или две ссылки на следующий и/или предыдущий узел списка.

Словари

- Словарь - неупорядоченная коллекция, доступ в которой осуществляется по ключу.

Пример:

```
{  
    "Russia": "Moscow",  
    "Finland": "Helsinki",  
}
```

Обход словаря

```
countries = {"Russia": "Moscow", "Finland": "Helsinki", "USA": "New York",  
"USA": "Washington, D.C."}
```

➤ Доступ к элементу осуществляется по ключу:

```
print(countries["Russia"])
```

➤ Элементы можно перебрать в цикле:

```
for country in countries:
```

```
    print(country)
```

DEMO

Изменяемость

- Словарь - изменяемый объект.
- Ключи словаря должны быть неизменяемыми уникальными объектами.
- Значения словаря могут быть изменяемыми неуникальными объектами.

Генераторы словарей

```
res = {x: ord(x) for x in 'QwErTyQ'}
```

```
res = {}
```

```
for x in 'QwErTyQ':
```

```
    res.update({x: ord(x)})
```

Преобразование типов

- `init_list = [`
 `['key_1', 1],`
 `['key_2', 2],`
 `['key_3', 3]`
 `]`
- `dict(init_list) # { 'key_1': 1, 'key_2': 2, 'key_3': 3 }`

Преобразование типов

- `init_str = "{ 'key_1': 1, 'key_2': 2, 'key_3': 3 }"`
- `eval(init_str)` # ! **НЕбезопасная функция !**
- `import ast`
`ast.literal_eval(init_str)`

Многомерные словари

```
subjects = {  
    "groups": {  
        "303": ["Computer Science", "Programming"],  
        "310": ["History"]  
    },  
    "classrooms": {  
        "Computer Science": "10",  
        "Programming": "20",  
        "History": "30",  
    }  
}
```

Возможные ошибки

```
groups = {"901": "Иванов", "902": "Петрова"}
```

Нельзя менять размер словаря во время итерирования словаря с помощью цикла.

```
for i in groups:
```

```
    groups["903"] = "Антонова"
```

```
for i in groups:
```

```
    groups.pop("901")
```

DEMO