

## Лекция 5

### Работа с коллекциями: списки, кортежи и словари

#### Введение

Коллекции данных в Python представлены несколькими основными структурами: списками, кортежами и словарями. Эти структуры данных являются фундаментальными элементами языка и обеспечивают мощные средства для хранения, управления и манипуляции разнообразными данными. В данной лекции мы рассмотрим каждую из этих структур, их особенности, синтаксис и методы, а также примеры их эффективного использования в программировании на Python.

#### 1. Списки

Список в Python — это упорядоченная изменяемая коллекция объектов различных типов. Списки являются одними из наиболее часто используемых инструментов в Python из-за их гибкости и простоты использования.

##### 1.1 Создание и доступ к элементам

Создать список можно, разместив элементы в квадратных скобках, разделенные запятыми. Доступ к элементам списка осуществляется по индексу, начинающемуся с нуля:

```
python
Копировать код
my_list = [1, "Hello", 3.14]
print(my_list[1]) # Вывод: Hello
```

##### 1.2 Методы списков

Списки в Python поддерживают множество методов для манипуляции данными:

- `append(x)` — добавляет элемент в конец списка;
- `insert(i, x)` — вставляет элемент на указанную позицию;
- `remove(x)` — удаляет первый элемент в списке, который равен `x`;
- `pop([i])` — удаляет элемент в позиции `i` и возвращает его;
- `clear()` — очищает список;
- `sort()` — сортирует элементы в списке.

## 1.3 Итерация по списку

Итерация по списку может выполняться с помощью цикла `for`:

```
python
Копировать код
for element in my_list:
    print(element)
```

## 2. Кортежи

Кортеж (`tuple`) — это неизменяемая упорядоченная коллекция объектов. Как и списки, кортежи могут содержать элементы различных типов. Основное отличие заключается в том, что кортежи нельзя изменять после создания.

### 2.1 Создание и доступ к элементам

Кортежи создаются путем размещения элементов в круглых скобках:

```
python
Копировать код
my_tuple = (1, "Hello", 3.14)
print(my_tuple[1]) # Вывод: Hello
```

### 2.2 Операции с кортежами

Хотя кортежи и неизменяемы, они поддерживают некоторые операции:

- Конкатенация;
- Повторение;
- Мембершип (`in` и `not in`);
- Индексирование и срезы.

### 2.3 Применение кортежей

Кортежи часто используются для защиты данных, которые не должны изменяться в программе, и как элементы словарей для ключей.

## 3. Словари

Словарь в Python — это неупорядоченная коллекция элементов в формате ключ-значение. Словари идеально подходят для задач, где элементы коллекции должны быть индексированы с помощью уникальных идентификаторов.

### 3.1 Создание и доступ к элементам

Словари создаются с помощью фигурных скобок, где каждый элемент состоит из пары ключ:значение:

```
python
Копировать код
my_dict = {'name': 'Alice', 'age': 25}
print(my_dict['name']) # Вывод: Alice
```

### 3.2 Методы словарей

Словари в Python обладают множеством полезных методов:

- `get(key)` — возвращает значение ключа, но если его нет, возвращает `None`;
- `keys()` — возвращает ключи в словаре;
- `values()` — возвращает значения в словаре;
- `items()` — возвращает пары (ключ, значение);
- `update(other)` — обновляет словарь, добавляя пары из другого словаря.

### 3.3 Итерация по словарю

Итерация по словарю может быть выполнена по ключам, значениям или по парам ключ-значение:

```
python
Копировать код
for key, value in my_dict.items():
    print(f"{key}: {value}")
```

### Заключение

Коллекции в Python предоставляют программистам мощные и гибкие структуры данных для решения широкого спектра задач. Знание особенностей и методов работы со списками, кортежами и словарями позволяет строить эффективные и оптимизированные программы. Надлежащее использование этих структур способствует повышению качества кода и его поддерживаемости.