

Кортежи (Tuple)

Список так же может быть неизменяемым, в этом случае он называется кортеж (tuple). Кортеж использует меньше памяти, чем список. Кортеж может содержать объекты, которые можно изменить.

Функция tuple() берет в качестве аргумента строку или список и превращает его в кортеж:

```
>>> tuple('abc')
('a', 'b', 'c')
```

Множества (Set)

Множества — неупорядоченная коллекция уникальных элементов. Множества поддерживают итерацию, добавление и удаление объектов и т.д. Индексация и срезы не поддерживаются.

```
>>> s = set('abcde')
```

Множества имеют встроенные функции:

- add() — добавление элемента;
- remove() — удаление элемента;
- clear() — очистка множества;
- pop() — удаление первого элемента.

Со множествами можно проводить операции пересечения, объединения, разности и так далее.

Словари (Dict)

После списков словарь является самым гибким встроенным типом. Если список — это упорядоченная коллекция, то словарь — неупорядоченная. Основные особенности словарей:

1. Доступ осуществляется по ключу, а не по индексу. По аналогии со списком, в словаре можно получить доступ к элементам в цикле по ключам.
2. Значения словаря хранятся в неотсортированном порядке, более того, ключи могут храниться не в том порядке, в котором они добавляются.
3. По аналогии со списками, словарь может хранить вложенные словари. Словарь может хранить в качестве значений объекты любого типа. Ключ в словаре — неизменяемый тип, может быть строкой, целым или вещественным числом, либо кортежем, состоящим из указанных типов.
4. Словари, как и списки, хранят ссылки на объекты, а не сами объекты.

1. Что такое словарь

Словарь — это ассоциативный массив, или неупорядоченное множество пар ключ: значение, где ключи уникальны.

Пара фигурных скобок {} создает пустой словарь (не множество!). В отличие от последовательностей, доступ к элементам словаря производится по ключу, а не по индексу, ключ может быть любого типа, ключ не допускает изменений.

Основные операции над словарем — сохранение с заданным ключом и извлечение по нему значения. Также можно удалить пару key: value с помощью инструкции del.

Метод keys() для словаря возвращает список всех используемых ключей в произвольном порядке; для сортировки списка можно применить метод sort(). Добавление нового объекта в словарь не

требует предварительных проверок: если ранее ключу уже соответствовало некоторое значение, оно будет перезаписано.

Создать словарь можно несколькими способами:

- `D = {'name': 'Ivan', 'age': 18}`
- Или:
`D = {}`
`D['name'] = 'Ivan'`
`D['age'] = 18`
- С помощью функции `dict()` — ключи при этом должны быть строками.
- С помощью `fromkeys()` — создает словарь по списку ключей с пустыми значениями:
`D = {}.fromkeys(['name', 'age'], 0)`
- С помощью конструктора:
`d = dict((x, x**2) for x in range(10))`

2. Функции/методы словаря

- `dict()` — создание словаря;
- `len()` — возвращает число пар;
- `clear()` — удаляет все значения из словаря;
- `copy()` — создает псевдокопию словаря;
- `deepcopy()` — создает полную копию словаря;
- `items()` — возвращает список значений;
- `keys()` — возвращает список ключей;
- `fromkeys()` — создает словарь по заданным ключам с пустыми значениями:

```
>>> {}.fromkeys(['name', 'age'])
{'age': None, 'name': None}
```

Можно все значения заполнить по умолчанию:

```
>>> {}.fromkeys(['name', 'age'], 123)
{'age': 123, 'name': 123}
```
- `get()` — получает значение по ключу, в случае отсутствия дает `None`;
- `pop()` — извлекает значение по ключу с последующим удалением;
- `popitem()` — извлекает произвольное значение с последующим удалением;
- `update()` — изменяет значение по ключу;
- `values()` — возвращает список значений;
- `del` — оператор удаляет пару ключ: значение по ключу;