

## Словари

Обычные списки (массивы) представляют собой набор пронумерованных элементов, то есть для обращения к какому-либо элементу списка необходимо указать его номер. Номер элемента в списке однозначно идентифицирует сам элемент. Но идентифицировать данные по числовым номерам не всегда оказывается удобно. Например, маршруты поездов в России идентифицируются численно-буквенным кодом (число и одна буква), также численно-буквенным кодом идентифицируются авиарейсы, то есть для хранения информации о рейсах поездов или самолетов в качестве идентификатора удобно было бы использовать не число, а текстовую строку.

Структура данных, позволяющая идентифицировать ее элементы не по числовому индексу, а по произвольному, называется *словарем* или *ассоциативным массивом*. Соответствующая структура данных в языке Питон называется `dict`.

Создание словаря:

```
d = dict()
# или так
d = { }
```

Рассмотрим простой пример использования словаря. Заведем словарь `Capitals`, где индексом является название страны, а значением – название столицы этой страны. Это позволит легко определять по строке с названием страны ее столицу.

```
# Создадим пустой словарь Capitals
Capitals = dict()
```

```
# Заполним его несколькими значениями
Capitals['Russia'] = 'Moscow'
Capitals['Ukraine'] = 'Kiev'
Capitals['USA'] = 'Washington'
```

```
Countries = ['Russia', 'France', 'USA', 'Russia']
```

```
For country in Countries:
```

```
    # Для каждой страны из списка проверим, есть ли она в словаре Capitals
```

```
    if country in Capitals:
```

```
        print('Столица страны ' + country + ': ' + Capitals[country])
```

```
    else:
```

```
        print('В базе нет страны с названием ' + country)
```

Пример словаря с ключами-кортежами. В качестве ключа используются координаты, а в качестве значения – название города.

```
cities = {  
    (55.75, 37.5): 'Москва',  
    (59.8, 30.3): 'Санкт-Петербург',  
    (54.32, 48.39): 'Ульяновск'  
}  
  
print(cities[(55.75, 37.5)])  
cities[(53.2, 50.15)] = 'Самара'
```

Значениями в словаре, в отличие от ключей, могут быть объекты любого типа – числа, строки, кортежи, списки и даже другие словари.

### Работа с элементами словаря

A[key]	Получение значения элемента по ключу, если элемента с заданным ключом нет в словаре, то возникает исключение <code>KeyError</code>
A.get(key)	Другой способ определения значения по ключу. Если элемента с ключом <code>get</code> нет в словаре, то возвращается значение <code>None</code> .
A.get(key, val)	Метод возвращает значение <code>val</code> , если элемент с ключом <code>key</code> отсутствует в словаре.
in и not in	Проверка принадлежности элемента словарю
A[key] = value	Для добавления нового элемента в словарь нужно просто присвоить ему какое-то значение
del A[key]	Удаление элемента из словаря. Операция генерирует исключение <code>KeyError</code> , если такого ключа в словаре нет
A.keys()	Получение списка всех ключей словаря
A.values()	Получение всех значений словаря
for key, val in A.items():	Перебрать элементы словаря A так, чтобы в переменной <code>key</code> оказывался ключ, а в <code>value</code> – соответствующее ему значение