

Тема 3.10 СЛОВАРИ

Словарь в Python (dict) - неупорядоченная последовательность произвольного типа с доступом по ключу. Их иногда ещё называют ассоциативными массивами или хеш-таблицами.

Элементами такой коллекции выступают пары объектов, каждая из которых включает в себя **ключ** и **значение**. Это позволяет идентифицировать ее элементы не по числовому индексу (как списки), а по произвольному, т.е. в качестве идентификатора можно использовать и число, и текстовую строку. Значениями в словаре, в отличие от ключей, могут быть объекты любого типа – числа, строки, кортежи, списки и даже другие словари.

В Python словари пишутся в фигурных скобках {}.

Создание словаря:

1) присвоить свободной переменной произвольное количество пар объектов. Элементы необходимо поместить в фигурные скобки, а между ключом и значением должен стоять символ двоеточия.

```
d = {} # создание пустого словаря
thisdict = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964
}
print(thisdict) # {'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
```

2) использовать **конструктор dict()**. В таком случае быть ключом может только строка, как это показано в следующем примере кода.

```
d = dict() # создание пустого словаря
thisdict = dict(brand="Ford", model="Mustang", year=1964)
print(thisdict) # {'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
```

Обратите внимание, ключевые слова не являются строками, а вместо двоеточия для задания используется «равно».

Для того чтобы ввести в коллекцию новую пару объектов необходимо указать новый ключ в квадратных скобках, а также соответствующее ему значение.

```
thisdict = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964
}
thisdict["color"] = "red"
print(thisdict) # {'brand': 'Ford', 'model': 'Mustang', 'year': 1964, 'color': 'red'}
```

Как и с другими наборами данных, производить заполнение словарей можно при помощи генераторов.

```
a = {a: a * a for a in range(5)}
print(a) # {0: 0, 1: 1, 2: 4, 3: 9, 4: 16}
```

Для работы со словарями доступны функции, меняющие их содержимое и выполняющие различные операции над ними. Его можно конвертировать в другие типы данных, например, в строку (потребуется функция `str`).

```
a = {1: "one", 2: "two", 3: "three"}
b = str(a)
print(b)      # {1: 'one', 2: 'two', 3: 'three'}
```

В Python словари могут быть **вложенными**, то есть выступать частью другого более крупного словаря.

```
a = {
    "First": {
        1: "one",
        2: "two",
        3: "three"
    },
    "Second": {
        4: "four",
        5: "five"
    }
}
print(a) # {'First': {1: 'one', 2: 'two', 3: 'three'}, 'Second': {4: 'four', 5: 'five'}}
```

В примере, создается словарь **a**, включающий в себя два других словаря (First и Second), т.е. в свою очередь, содержат несколько пар ключей и значений.

В Python существует набор встроенных методов, с помощью которых можно работать со словарями (таблица 1).

Таблица 1 – Методы словаря

Метод	Значение
<code>len()</code>	Получение размера (возвращает число пар)
<code>clear()</code>	Удаляет все элементы из словаря
<code>copy()</code>	Делает копию словаря
<code>fromkeys()</code>	Возвращает словарь с указанными ключами и значениями
<code>get()</code>	Возвращает значение по ключу
<code>items()</code>	Возвращает список, содержащий tuple для каждой пары ключ-значение
<code>keys()</code>	Возвращает список, содержащий ключи словаря
<code>pop()</code>	Удаляет элементы по ключу и возвращает его
<code>popitem()</code>	Удаляет последнюю пару ключа со значением

Метод	Значение
setdefault()	Задаёт значение по ключу. Если ключа нет в словаре, добавляет его с указанным значением или None
update()	Обновляет словарь, добавляя пары ключ-значение (объединение словарей)
values()	Возвращает список всех значений в словаре
del()	Удаляет пару по ключу или весь словарь
in	Проверяет наличие значения по ключу
eval()	Конвертация строки в словарь

На примере программы — телефонной книги покажем некоторые операции со словарями:

```
telbook = {'sasha': "32-11-4", 'vanya': '44-65-99'} # Объявляем словарь
telbook['fedya'] = '22-47-32' # добавляем новый объект в словарь
print telbook # Выводим все значения словаря
{'vanya': '44-65-99', 'fedya': '22-47-32', 'sasha': '32-11-4'}
print(len(telbook)) # определяет размер словаря
3
print telbook['vanya'] # Выводим номер значения 'vanya'
44-65-99
del telbook['sasha'] # удаляем значение 'sasha'
print telbook # смотрим, что получилось
{'vanya': '44-65-99', 'fedya': '22-47-32'}
print telbook.keys() # Выводим значения по ключу
['vanya', 'fedya']
print telbook.has_key('vanya') # проверяем, есть ли в словаре значение 'vanya'
True
```