Словари

Словарь в Python (тип данных dict) — это неупорядоченная коллекция элементов; каждый элемент словаря состоит из пары ключ:значение.

С помощью словарей можно быстро находить данные по ключу, как в телефонной книге, и структурировать и организовывать данные, чтобы быстро и эффективно получать доступ к нужной информации по её метке (ключу). Например, можно использовать словарь для хранения информации о студентах, где ключами могут быть их имена, а значениями — их возрасты или оценки.

Синтаксис

- Словари в Python записываются в фигурных скобках {}.
- Между ключом и значением ставится двоеточие.
- В пределах словаря каждый ключ уникален.
- Элементы словаря разделяются запятой.

Как объявить словарь

1 Литерально:

```
empty_dict = {}
print(empty_dict)
# Вывод в терминал: {}
print(type(empty_dict))
# Вывод в терминал: <class 'dict'>
```

2 С помощью функции dict().

В качестве аргумента функции передаётся список кортежей, каждый из которых должен состоять из двух элементов; эти кортежи будут преобразованы в элементы словаря.

```
movies = [('Матрица', 4.7), ('Трон', 3.8)]
movies_dict = dict(movies)
print(movies_dict)
# Вывод в терминал: {'Матрица': 4.7, 'Трон': 3.8}
```

3 С помощью метода dict.fromkeys().

Метод принимает в качестве обязательного аргумента последовательность элементов. Значения элементов будут ключами словаря.

Вторым, необязательным аргументом можно передать любой объект, который будет значением для всех ключей словаря. По умолчанию он равен None.

```
movies = dict.fromkeys(['Матрица', 'Хакеры', 'Трон', 'Кибер'], 4.8) print(movies)
# Вывод в терминал: {'Матрица': 4.8, 'Хакеры': 4.8, 'Трон': 4.8, 'Кибер': 4.8}
```

4 С помощью функции zip().

Это функция-упаковщик. Она принимает в качестве аргументов итерируемые объекты, а возвращает объект типа zip — это коллекция кортежей. Каждый из кортежей содержит элементы исходных коллекций с одинаковыми индексами. Напечатать zip-объект невозможно: содержимое упаковано.

```
movie_ratings = [4.7, 5.0, 4.3, 4.0]
movies = ['Матрица', 'Хакеры', 'Трон', 'Кибер']
movies_info = zip(movies, movie_ratings)
```

5 С помощью конструкции dict comprehensions.

Эта конструкция создаёт словарь на основе значений какой-то исходной последовательности, пробегаясь по ней циклом, получая её элементы и создавая ключи и значения элементов словаря. Синтаксис конструкции:

```
<имя_словаря> = {
    <ключ>:<значение> for <переменная_цикла> in <имя_исходной_последовательности>
}
```

Работа со словарями

Обращение к элементам словаря Возможно только по ключу:

```
movies = {
    'Трон': 3.8,
    'Кибер': 2.5,
    'Пятая власть': 4.1,
}

print(movies['Кибер'])
# Вывод в терминал: 2.5
```

Проверка наличия ключа в словаре Выполняется через оператор in:

```
movies = {
     Трон':  {'rating': '3.8', 'review': 'Смотреть можно'},
    'Кибер':  {'rating': '2.5', 'review': 'Так себе киношечка'},
}

movie = 'Трон'
if movie in movies:
    print('Информация об этом фильме доступна')
else:
    print('Информация о фильме отсутствует!')
```

Возврат значения по ключу: метод get()

Если запрошенный ключ не найден — вернётся None или значение, переданное вторым аргументом.

```
movies = {
   'Трон':  {'rating': 3.8, 'review': 'Смотреть можно'},
   'Кибер':  {'rating': 2.5, 'review': 'Так себе киношечка'},
   'Пятая власть':  {'rating': 4.1, 'review': 'Смотреть можно'}, }
# Вызов метода get() с одним обязательным параметром.
print(movies.get('Хоббит')) # Вывод в
терминал: None

# Вызов метода get() с двумя аргументами.
# Второй аргумент - значение, которое вернётся, если ключ в словаре не найден. print(movies.get('Хатико', 'Такого фильма нет в словаре'))
# Вывод в терминал: Такого фильма нет в словаре

print(movies.get('Трон'))
# Вывод в терминал: {'rating': 3.8, 'review': 'Смотреть можно'}
```

Добавление нового элемента в словарь

Новому ключу словаря присваивается значение:

```
movies = {
    'Матрица': 4.7,
    'Хакеры': 4.3,
    'Трон': 3.8,
}

movies['Сеть'] = 4.3
print(movies)
# Вывод в терминал: {'Матрица': 4.7, 'Хакеры': 4.3, 'Трон': 3.8, 'Сеть': 4.3}
```

Объединение словарей: метод update()

```
movies = {
    'Матрица': 4.7,
    'Хакеры': 4.3,
    'Трон': 3.8,
}

new_movies = {
    'Сеть': 4.1,
    '23': 4.3,
}

movies.update(new_movies)
```

Удаление элемента из словаря: оператор del

```
movies = {
    'Матрица': 4.7,
    'Хакеры': 5.0,
    'Трон': 3.8,
}

del movies['Трон']
```

Возврат значения удаляемого элемента: метод pop() Принимает два аргумента pop(key, default):

- key — ключ элемента, который нужно удалить из словаря;
- необязательный аргумент default — значение, которое вернёт метод в случае, если переданный ключ не найден в словаре.

```
movies = {
    'Трон': 3.8,
    'Кибер': 2.5,
    'Пятая власть': 4.1,
}
movie_pop = movies.pop('Сеть', 'Фильм не найден')
print(movie_pop)
# Вывод в терминал: Фильм не найден

movie_pop = movies.pop('Трон')
print(movie_pop)
# Вывод в терминал: 3.8
```

Получение ключей и значений элементов словаря: методы keys() и values()

Метод keys() возвращает объект dict_keys — это последовательность, хранящая ключи словаря.

Метод values() возвращает объект dict_values: это последовательность, элементами которой будут значения словаря.

```
movies = {
    'Трон': 3.8,
    'Кибер': 2.5,
    'Пятая власть': 4.1,
}
keys = movies.keys()
print(keys)
# Вывод в терминал: dict_keys(['Трон', 'Кибер', 'Пятая власть'])
values = movies.values()
print(values)
# Вывод в терминал: dict_values([3.8, 2.5, 4.1])
```

Перебор словаря: цикл for

```
movies = {
    'Трон': 3.8,
    'Кибер': 2.5,
}

for movie_name in movies:
    print(movie_name)

# Трон
# Кибер
```

Получение из словаря пары «ключ:значение»: метод items()

Преобразует словарь в объект dict_items — это коллекция кортежей, каждый из которых первым элементом содержит ключ, а вторым — значение элемента словаря.

```
movies = {
    'Трон': 3.8,
    'Кибер': 2.5,
}

print(movies.items())
# dict_items([('Трон', 3.8), ('Кибер', 2.5)])
```

Очистка и копирование словаря: метод clear()

```
movies = {
    'Трон': 3.8,
    'Кибер': 2.5,
}

movies.clear()
```

Копирование словаря: метод copy()

```
movies = {
    'Матрица': 4.7,
    'Хакеры': 4.3,
}
print(id(movies))
# Вывод в терминал: 1781869204544

movies_copy = movies.copy()
print(id(movies_copy))
# Вывод в терминал: 1781869204608
```