

Методы словаря dict и поддерживаемые операции



code.top-academy.ru

РЕКЛАМА

Бесплатные курсы программирования для студентов СПО

Разработка игр на Python. Построение сайтов. Обучим любой из профессий Бесплатно!

[Узнать больше](#)

[Справочник по языку Python3.](#) / Методы словаря dict и поддерживаемые операции

Что из себя представляет [тип данных словарь dict](#) можно посмотреть в разделе "[Встроенные типы данных](#)"

Здесь представлены [операции, которые поддерживают словари](#) и, следовательно, пользовательские типы словарей также должны их поддерживать.

[Представления словарей dict.keys\(\), dict.values и dict.items в Python](#)

Объекты, возвращаемые методами словарей `dict.keys()`, `dict.values()` и `dict.items()` являются объектами-представлениями. Они обеспечивают динамическое представление записей словаря. Другими словами, при изменении словаря представление будет отража

[Исходный словарь для представления dictview.mapping в Python](#)

Атрибут представления словаря `dictview.mapping` возвращает `types.MappingProxyType`, который оборачивает исходный словарь, на который ссылается представление.

[Получение списка ключей словаря list\(dict\) в Python](#)

Список ключей словаря можно получить в результате преобразования словаря `dict` в список, используя встроенный класс `list()`. Метод `dict.keys()` возвращает список-представление всех ключей, используемых в словаре `dict`.

[Количество элементов в словаре len\(dict\) в Python](#)

Операции `len(dict)` возвращает количество элементов в словаре `dict`. Функция `len(dictview)` так же возвратит количество элементов в словаре, если `dictview` является представлением словаря `dict`.

[Доступ к значению словаря по ключу dict\[key\] в Python](#)

Операция `dict[key]` вернет элемент словаря `dict` с ключом `key`. Вызывает исключение `KeyError`, если ключ `key` отсутствует в словаре.

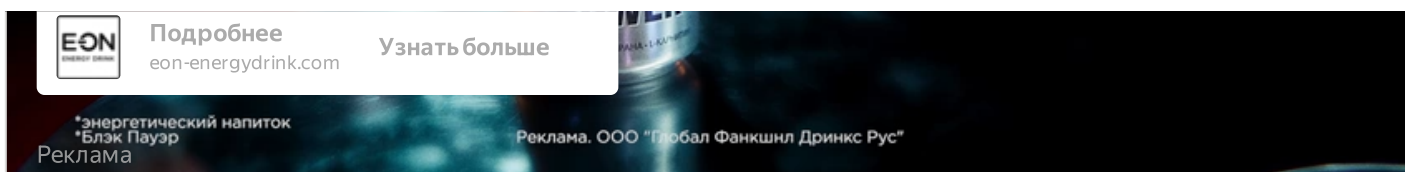
[Добавление/изменение значения словаря по ключу key в Python](#)

Операция `d[key] = value` добавит в словарь `dict` новый элемент - пару ключ-значение. Если в словаре существует ключ `key` то эта операция присвоит ключу `key` новое значение `value`.

[Удаление значения словаря по ключу в Python](#)

Операция `del dict[key]` удалит элемент словаря `dict`, соответствующий ключу `key`. Операция удаления бросает исключение `KeyError`, если ключ `key` отсутствует в словаре `dict`.





[Проверка наличия/отсутствия ключа key в словаре dict в Python](#)

Представленные операции проверки вхождения ключа key в словарь dict вернут True, если у словаря dict ЕСТЬ ключ key, если НЕТ то False или наоборот.

[Проверка наличия/отсутствия значения value в словаре Python](#)

Операция `val in dict.values()` вернет True, если в словаре dict любой из ключей имеет значение val, если такое значение не найдено то вернет False.

[Проверка наличия/отсутствия пары \(key, value\) в словаре dict в Python](#)

Операция `key_val in dict.items()` вернет True, если в словаре dict ЕСТЬ проверяемая пара key_val ключ-значение, если пара не найдена то вернет False.

[Итерирование по ключам и значениям словаря Python](#)

Операция `iter(dict)` вернет итератор по ключам словаря dict. Это операция является ссылкой на итератор, созданный из списка-представления ключей ярлык для `iter(dict.keys())`.

[Метод dict.clear\(\) в Python. Очистить словарь](#)

Метод `dict.clear()` производит удаление всех элементов из словаря dict.

[Метод dict.copy\(\) в Python, копия словаря](#)

Метод `dict.copy()` вернет мелкую копию словаря dict.

[Метод dict.fromkeys\(\) в Python, словарь с ключами по умолчанию](#)

Метод `dict.fromkeys()` встроенного класса dict() создает новый словарь с ключами из последовательности iterable и значениями, установленными в value.

[Метод dict.get\(\) в Python, значение по умолчанию если ключа нет](#)

Метод `dict.get()` возвращает значение для ключа key, если ключ находится в словаре, если ключ отсутствует то вернет значение default. Если значение default не задано и ключ key не найден, то метод вернет значение None.

[Метод dict.items\(\) в Python, список кортежей](#)

Метод `dict.items()` возвращает новый список-представление пар элементов словаря dict. Другими словами возвращает список кортежей вида (key, value), состоящий из элементов словаря.

[Метод dict.keys\(\) в Python, список ключей словаря](#)

Метод `dict.keys()` возвращает новый список-представление всех ключей dict_keys, содержащихся в словаре dict. Все изменения, такие как удаление или добавление ключей в словаре сразу отражаются на этом представлении.

[Метод dict.values\(\) в Python, список значений словаря](#)

Метод `dict.values()` возвращает новый список-представление всех значений dict_values, содержащихся в словаре dict. Все изменения, такие как удаление, изменение или добавление значений в словаре сразу отражаются на этом представлении.

[Метод dict.pop\(\) в Python](#)

Метод `dict.pop()` вернет значение ключа key, а также удалит его из словаря dict. Если ключ не найден, то вернет значение по умолчанию default.

[Метод dict.popitem\(\) в Python, получить пару ключ/значение](#)

Метод `dict.popitem()` удалит и вернет пару (key, value) из словаря. Пары возвращаются в порядке LIFO (последним пришёл - первым ушёл).

[Метод dict.setdefault\(\) в Python, получает/вставляет значение ключа](#)

Метод `dict.setdefault()` возвращает значение ключа key находящегося в словаре dict. Если ключа нет, то вставит его в словарь с указанным значением.

[Метод dict.update\(\) в Python, обновление/дополнение словаря](#)

Метод `dict.update()` обновляет словарь `dict` с помощью пар ключ-значение из `other`, перезаписывая существующие ключи. Если ключ в словаре отсутствует, то он добавляется. Метод ничего не возвращает.

[Объединение двух словарей в новый словарь Python](#)

Выражение `{**dict1, **dict2}` создает **новый словарь** с объединенными ключами и значениями из `dict1` и `dict2`, которые оба должны быть словарями.

[Сортировка словаря по значению и/или ключу в Python](#)

Сортировка словарей производится при помощи встроенной функцией `sorted()` и происходит немного сложнее чем сортировка списков или кортежей. В словарях необходимо сортировать список кортежей (`key, val`) получаемых методом `dict.items()`, а затем обратно преобразовывать в словарь.

[Обратный порядок/реверс словаря `reversed\(dict\)` в Python](#)

Операция `reversed(dict)` возвращает обратный итератор по ключам словаря `dict`. Является ссылкой на операцию `reversed(dict.keys())`.

[Генератор словаря и его использование в Python](#)

Для создания словарей из произвольных ключей и значений можно использовать генераторы словарей. При помощи генератора словаря можно создать словарь из списка с каким-то значением по умолчанию для значений ключей, тем самым воспроизвести метод `dict.fromkeys()`.

[Фильтр словаря по ключам и/или значениям в Python](#)

Отфильтровать ключи и значения словаря Python можно двумя способами. Первый - при помощи выражения генератора словаря, а второй с помощью встроенной функции `filter()`.

[Словарь как фабрика функций в Python](#)

Так как функции в Python являются объектами, то можно заменить значения ключей словаря функциями и возвращать их, если ключи совпадают.

ХОЧУ ПОМОЧЬ
ПРОЕКТУ

