



Python 3 для начинающих

Книги Самоучитель Курсы Категории ↓ Все статьи

Словари (dict) и работа с ними. Методы словарей



Онлайн-тренажер Python 3 для начинающих

Теория без воды. Задачи с автоматической проверкой. Подсказки на русском языке. Работает в любом современном браузере.

НАЧАТЬ БЕСПЛАТНО

Сегодня я расскажу о таком типе данных, как **словари**, о работе со словарями, операциях над ними, методах, о генераторах словарей.

Словари в Python - неупорядоченные коллекции произвольных объектов с доступом по ключу. Их иногда ещё называют ассоциативными массивами или хеш-таблицами.

Чтобы работать со словарём, его нужно создать. Сделать это можно несколькими способами. Во-первых, с помощью литерала:

```
>>> d = {}
>>> d
{}
>>> d = {'dict': 1, 'dictionary': 2}
>>> d
{'dict': 1, 'dictionary': 2}
```

Во-вторых, с помощью функции **dict**:

```
>>> d = dict(short='dict', long='dictionary')
>>> d
{'short': 'dict', 'long': 'dictionary'}
>>> d = dict([(1, 1), (2, 4)])
>>> d
{1: 1, 2: 4}
```

В-третьих, с помощью метода **fromkeys**:

```
>>> d = dict.fromkeys(['a', 'b'])
>>> d
{'a': None, 'b': None}
>>> d = dict.fromkeys(['a', 'b'], 100)
>>> d
{'a': 100, 'b': 100}
```

В-четвертых, с помощью генераторов словарей, которые очень похожи на **генераторы списков**.

Поиск ...

Свежее

- [Модуль csv - чтение и запись CSV файлов](#)
- [Создаём сайт на Django, используя хорошие практики. Часть 1: создаём проект](#)
- [Онлайн-обучение Python: сравнение популярных программ](#)

Категории

- [Книги о Python](#)
- [GUI \(графический интерфейс пользователя\)](#)
- [Курсы Python](#)
- [Модули](#)
- [Новости мира Python](#)
- [NumPy](#)
- [Обработка данных](#)
- [Основы программирования](#)
- [Примеры программ](#)
- [Типы данных в Python](#)
- [Видео](#)
- [Python для Web](#)
- [Работа для Python-программистов](#)

Полезные материалы

- [Сделай свой вклад в развитие сайта!](#)
- [Самоучитель Python](#)
- [Карта сайта](#)
- [Отзывы на книги по Python](#)
- [Реклама на сайте](#)

Мы в соцсетях

```
>>> d = {a: a ** 2 for a in range(7)}
>>> d
{0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36}
```

Теперь попробуем добавить записей в словарь и извлечь значения ключей:

```
>>> d = {1: 2, 2: 4, 3: 9}
>>> d[1]
2
>>> d[4] = 4 ** 2
>>> d
{1: 2, 2: 4, 3: 9, 4: 16}
>>> d['1']
Traceback (most recent call last):
  File "", line 1, in
    d['1']
KeyError: '1'
```

Как видно из примера, присвоение по новому ключу расширяет словарь, присвоение по существующему ключу перезаписывает его, а попытка извлечения несуществующего ключа порождает исключение. Для избежания исключения есть специальный метод (см. ниже), или можно [перехватывать исключение](#).

Что же можно еще делать со словарями? Да то же самое, что и с другими объектами: [встроенные функции](#), [ключевые слова](#) (например, [циклы for](#) и [while](#)), а также специальные методы словарей.

Методы словарей

dict.clear() - очищает словарь.

dict.copy() - возвращает копию словаря.

classmethod **dict.fromkeys(seq[, value])** - создает словарь с ключами из seq и значением value (по умолчанию None).

dict.get(key[, default]) - возвращает значение ключа, но если его нет, не бросает исключение, а возвращает default (по умолчанию None).

dict.items() - возвращает пары (ключ, значение).

dict.keys() - возвращает ключи в словаре.

dict.pop(key[, default]) - удаляет ключ и возвращает значение. Если ключа нет, возвращает default (по умолчанию бросает исключение).

dict.popitem() - удаляет и возвращает пару (ключ, значение). Если словарь пуст, бросает исключение KeyError. Помните, что словари неупорядочены.

dict.setdefault(key[, default]) - возвращает значение ключа, но если его нет, не бросает исключение, а создает ключ со значением default (по умолчанию None).

dict.update([other]) - обновляет словарь, добавляя пары (ключ, значение) из other. Существующие ключи перезаписываются. Возвращает None (не новый словарь!).

dict.values() - возвращает значения в словаре.

Ошибка в тексте?

Выделите ее мышкой!

И нажмите:



Для вставки кода на Python в комментарий закрывайте его в теги `<pre><code class="python3">Ваш код</code></pre>`

Подпишись на обновления [по RSS](#) или по почте!

Подписаться!

120 Комментариев

 Войти ▾

Присоединиться к обсуждению...

ВОЙТИ С ПОМОЩЬЮ

ИЛИ ЧЕРЕЗ DISQUS  26

Поделиться

Лучшие Новые Старые**Андрей Валерьевич**

год назад

```
{1: 2, 2: 4, 3: 9, 4: 16}
```

```
>>> d['1']
```

Traceback (most recent call last):

File "", line 1, in

почему ошибка? в чем отличие с верхней строкой? видим же ключ 1

0 0 Ответить **werg567og**

→ Андрей Валерьевич

8 месяцев назад

болбес, у тебя нет ключа '1', у тебя есть ключ 1

2 1 Ответить **Арсений**

→ Андрей Валерьевич

год назад

Подозреваю что в словаре ключ 1 имеет тип данных integer, а в запросе d['1'] спрашивается о значении под ключом с типом данных string. '1' не равно 1.

1 0 Ответить **Alexey Shipunov**

2 года назад edited

Нужна помощь начинающему

Есть словарь 234 в нем ещё 2 словаря - на выходе надо получить

количество обоих словарей и сумму

54 27860 - это получилось отсюда (22*510+32*520)

```
store = {
'234': [
{'quantiti': 22, 'price': 510}, {'quantiti': 32, 'price': 520}
],
```

0 0 Ответить **Obida**

→ Alexey Shipunov

10 месяцев назад

```
store = {"234" : [{"quantiti": 22, "price": 510}, {"quantiti": 32, "price": 520} ]
}
```

```
result = store["234"][0]["quantiti"] * store["234"][0]["price"] + store["234"]
[1]["quantiti"] * store["234"][1]["price"]
```

```
print(result) # output 27860
```

0 0 Ответить **Alexey Shipunov**

→ Alexey Shipunov

2 года назад

Решение уже найдено мало ли кому пригодиться вот полный код задания

т.к. в помощи просил по одной строке (не умел написать цикл)

```
goods = {
'Лампа': '123',
'Стол': '234',
'Диван': '345',
'Стул': '456',
}
```

```

store = {'123': [{'quantiti': 27, 'price': 42}],
'234': [{'quantiti': 22, 'price': 510}, {'quantiti': 32, 'price': 520}],
'345': [{'quantiti': 2, 'price': 1200}, {'quantiti': 1, 'price': 1150}],
'456': [{'quantiti': 50, 'price': 100}, {'quantiti': 12, 'price': 95}, {'quantiti': 43,
'price': 97}, ]}

# In[2]:
#создаю цикл для key_g - проходится по каждому ключу словаря,
а values_g по значению внутри
#тоже самое делаю для store, чтобы сравнить ключ магазина и
значение в goods

for key_g, values_g in goods.items():
for key_s, values_s in store.items():
if values_g == key_s: #пишу условие, если значение goods = ключу
store, то:
for i in range(len(values_s)): #создаю цикл который работает от 0 до
длины values_s = 4
quantiti = 0 # обнуляю каждый раз значения
price = 0 # обнуляю каждый раз значения
for v in values_s: # еще один цикл для того чтобы вытащить, что
внутри values_s
quantiti += v["quantiti"] # все значения внутри с quantiti складываю и
записываю в quantiti
price += v["quantiti"] * v["price"] # все значения внутри с quantiti и
price умножаю
print(f'{key_g} {quantiti} шт, стоимостью {price}') # ну и наконец
вывожу все в приятной форме

# In[3]:
goods.items()
# In[4]:
len(store.values())
# In[5]:
store.values()

```

2 0 Ответить



Hulitolku → Alexey Shipunov

2 года назад

Сумма 54 откуда берется?

0 0 Ответить



Alexey Shipunov → Hulitolku

2 года назад

22+32

0 0 Ответить



Сергей Шахов

2 года назад

```
def apply_discounts(products, stocks):
```

```
"""
```

Напишите функцию apply_discounts(), которая снижает цену продуктов в словаре products на указанный в словаре stocks процент.

Функция должна вернуть результирующий словарь, ключи которого — товары, а значения — новые цены.

Если продукта из словаря products нет в словаре stocks, то его необходимо пропустить. Цены округлите до второго знака после запятой.

```
"""
```

```

new_products = {}
for a in products:
if a in stocks:
products[a] = round(((products[a]/100)*(100-int(stocks[a][2]))),2)
new_products.update(products)
return new_products

```

```

if __name__ == '__main__':
products={
"Oranges (packaged)": 114.99,
"Candy (Rotfront)": 280.00,
"Boiled sausage": 199.99,
"Juice J7 (orange)": 119.99,
"Trout (Seven Seas)": 399.99
}

```

stocks = {
 "Boiled sausage": "33%",
 "Juice J7 (orange)": "12%",
 "Trout (Seven Seas)": "18%"
}

new_products = apply_discounts(products, stocks)
 print(new_products)

Добрый день код не проходит тест в Codeboard, кто может подсказать?

0 0 Ответить



s_t_r_a_j → Сергей Шахов

2 года назад edited

```
def apply_discounts():
    products={
        "Oranges (packaged)": 114.99,
        "Candy (Rotfront)": 280.00,
        "Boiled sausage": 199.99,
        "Juice J7 (orange)": 119.99,
        "Trout (Seven Seas)": 399.99
    }
    stocks = {
        "Boiled sausage": "33%",
        "Juice J7 (orange)": "12%",
        "Trout (Seven Seas)": "18%"
    }
    new_products = {}
    for a in products:
        if a in stocks:
            products[a] = round(((products[a]/100)*(100-int(stocks[a][:2]))),2)
            new_products.update(products)
    return new_products
print(apply_discounts())
```

Только учитывайте табулатуру в коде, здесь знаки табуляции не отображаются

0 0 Ответить



Igor Kutsevalov

2 года назад

Можно ли пройти по словарю циклом while?

1 0 Ответить



Alex Saminsky → Igor Kutsevalov

2 года назад

Да, перебор делается функцией
 next(x) - Возвращает следующий элемент итератора.

0 0 Ответить



maldinixus maldinixus → Igor Kutsevalov

2 года назад

да

0 0 Ответить



Ник

2 года назад

подскажите, а если у меня есть словарь, значением в котором является список, как я могу вызвать или изменить конкретный элемент списка, к примеру {"Вова": [3, 5, 4, 3]} как я могу вызвать и поменять оценку 4 на 5?

0 0 Ответить



Dmitriy Makhnev → Ник

2 года назад

как удалить ненужные значения?

0 0 Ответить



Ник → Ник

2 года назад

всем спасибо, нашел решение сам, может кому пригодится:
 dict = {"Вова": [3, 5, 4, 3]}

`print(dict["Вова"][2])` выведет на экран значение 4, так как возьмет из слова dict, ключ "Вова" и соответствующее ему значение под номером 2

1 0 Ответить



Krazy Deni X

2 года назад

как достать 1 ключ из словаря

0 0 Ответить



Ник

→ Krazy Deni X

2 года назад

`list(res.keys())[0]`

0 0 Ответить



Алексей Иванов

→ Krazy Deni X

2 года назад

`dict.keys()` с помощью цикла `for` берешь нужный

```
_dict = {
    'Name': 'noname',
    'Age': '100',
    'IQ': '0'
}
for key in _dict.keys():
    if key == 'IQ':
        print(_dict.get(key))
```

2 0 Ответить



Mishel Mishels

2 года назад

Объедините два словаря в один?

`dic1 = {1: 10, 2: 20}`

`dic2 = {3: 30, 4: 40}`

как напишите код два словаря в один?

0 0 Ответить



Viktor Viktor

→ Mishel Mishels

5 месяцев назад

`print(dic1 | dic2)`

0 0 Ответить



Andy Andy

→ Mishel Mishels

2 года назад

`dic1 = {1: 10, 2: 20}`

`dic2 = {3: 30, 4: 40}`

`dic1.update(dic2)`

`print(dic1)` #проверяем объединение словарей

1 0 Ответить



Ghz

→ Andy Andy

2 года назад

`dic1 = {1: 10, 2: 20}`

`dic2 = {3: 30, 4: 40}`

`dic3 = {**dic1, **dic2}`

0 0 Ответить



Mishel Mishels

→ Mishel Mishels

2 года назад

`dict_num =(dic1, dic2)`

`print(dict_num)`

правильно ответ?

0 0 Ответить



induxa

→ Mishel Mishels

2 года назад

Правильно но тогда он не объединится в один, а просто

будут два в одном

получится так:

`((1: 10, 2: 20), (3: 30, 4: 40))`

{1: 10, 2: 20, 3: 30, 4: 40}

если по заданию тебе так надо то можно так, но если два

в одном должны быть то выглядит должно быть так:

{1: 10, 2: 20, 3: 30, 4: 40}

0 0 Ответить



Исламбек Токтомушов

3 года назад

Дан словарь, удалите из него все элементы со значениями None.

Например:

a = {'a': None, 'b': 1, 'c': 2, 'd': None, 'e': 3} -> {'b': 1, 'c': 2, 'e': 3}

1 0 Ответить



Исламбек Токтомушов

→ Исламбек Токтомушов

3 года назад

не могу решить ,хэлп

1 0 Ответить



Alex Kr

→ Исламбек Токтомушов

2 года назад

a = {i: j for i, j in a.items() if j!=None }

2 0 Ответить



Рома Одесса

→ Исламбек Токтомушов

3 года назад

мой вариант

a = {'a': None, 'b': 1, 'c': 2, 'd': None, 'e': 3}

for j in a.copy():

if a[j] == None:

a.pop(j)

1 0 Ответить



Orozbek Sambaev

→ Рома Одесса

3 года назад

thanks a bunch!

1 0 Ответить



Илья Касьянов

→ Исламбек Токтомушов

3 года назад edited

b = {}

for k, v in a.items():

if v != None:

b.update({k: v})

print(b)

1 0 Ответить



Evgenij Jobs

3 года назад

С версии 3.7 словарь - упорядоченный набор ключ-значений.

1 0 Ответить



Шелколапа

3 года назад

А как добавить в уже существующий словарь ключ и значение к нему?

0 0 Ответить



Xanday Channel

→ Шелколапа

3 года назад

name_of_dict[key] = value

0 1 Ответить



Гаджидада Магомедов

3 года назад

Иван решил создать самый большой словарь в мире, которая принимает

неограниченное количество параметров «ключ: значение и обновляет

созданный им словарь, состоящий всего из одного элемента со

значением.

0 0 Ответить ↗



Гаджидада Магомедов

→ Гаджидада Магомедов

— 🚩

3 года назад

Как это сделать?

0 0 Ответить ↗



Alexander Andreev

— 🚩

3 года назад

Либо я вообще куку, либо этот метод `dict.update([other])` работает {только с такими} скобками.

0 0 Ответить ↗



Andy Andy

→ Alexander Andreev

— 🚩

2 года назад

со словарями работает `update`. С множествами - `union`

0 0 Ответить ↗



Igor Lunov

— 🚩

3 года назад

Здравствуйте. Благодарю за информацию. Небольшое замечание `dict.popitem()` удаляет и возвращает ПОСЛЕДнюю пару (ключ, значение)

0 0 Ответить ↗



Haya Dvora Sharon

— 🚩

4 года назад

А у меня вот это почему-то не работает:

```
d = dict(short='dict', long='dictionary')
```

`TypeError: 'dict' object is not callable`

0 0 Ответить ↗



Alexander Andreev

→ Haya Dvora Sharon

— 🚩

3 года назад

edited

тоже не работает

0 0 Ответить ↗



Екатерина Фролова

→ Haya Dvora Sharon

— 🚩

4 года назад

проверьте, не определили ли вы раньше по коду `dict` как переменную с чем-то - он очевидно ругается на это

2 0 Ответить ↗



Данил Самодуров

— 🚩

5 лет назад

Здравствуйте. При применении метода `.values()` возвращается объект типа `<class 'dict_values'>`. Хотелось бы узнать как извлечь этот объект в виде списка. Знаю, что на версии Python 2 он изначально был списком, а теперь нет. Моя задача заключается в том, чтобы взять среднее из списка значений словаря.

0 0 Ответить ↗



Константин Щербаков

→ Данил Самодуров

— 🚩

4 года назад

`list(d.values())` - список всех значенк списка `d`

1 0 Ответить ↗



lol kek

→ Константин Щербаков

— 🚩

4 года назад

```
val = d.values()
print(val.__dict__)
```

0 0 Ответить ↗



Данил Самодуров

→ Константин Щербаков

— 🚩

4 года назад

edited



Это именно то, что я попробовал в первую очередь – не помогло. Помогло вот это: [i for i in dic.values()]

0 0 Ответить



Аватар

Комментарий был удален.



Вячеслав Евсеев

➔ Guest



5 лет назад