The results are in! See what nearly 90,000 developers picked as their most loved, dreaded, and desired coding languages and more in the 2019 Developer Survey.

Python add new item to dictionary [duplicate]

<div style="float:right">Ask Question</div>

459

This question already has an answer here:
[Add new keys to a dictionary?](#) *15 answers*

★

101

I want to add an item to an existing dictionary in python. For example, this is my dictionary:

```
default_data = {
            'item1': 1,
            'item2': 2,
 }
```

I want to add new item such that:

```
default_data = default_data + {'item3':3}
```

How to achieve this?

`python`   `dictionary`   `items`

edited Jun 21 '16 at 14:25

Chaitanya Bapat
327   1   8   22

asked Jun 20 '11 at 19:07

brsbilgic
4,379   8   50   82

marked as duplicate by Shai, Ber, alecxe, FallenAngel, SysDragon
Jun 11 '13 at 8:12

This question has been asked before and already has an answer. If those answers do not fully address your question, please ask a new question.

12   [docs.python.org/tutorial/datastructures.html#dictionaries](#) –
Ignacio Vazquez-Abrams Jun 20 '11

3　　@machine yearning:
　　　meta.stackexchange.com/questions/5
　　　280/embrace-the-non-googlers –
　　　Fred Larson Jun 20 '11 at 19:15 ✎

　　　I apologize, I did not realize the depth
　　　of policy on this type of matter.
　　　Thanks for the link, it was informative.
　　　– machine yearning Jun 20 '11 at
　　　19:21

　　　@machine yearning: Good find. I've
　　　voted to close as an exact duplicate.
　　　– Fred Larson Jun 20 '11 at 19:32

3 Answers

```
default_data['item3'] = 3
```

1035

Easy as py.

Another possible solution:

```
default_data.update({'item3': 3})
```

which is nice if you want to insert
multiple items at once.

answered Jun 20 '11 at 19:08

Chris Eberle
38k　9　68　104

14　This response is more useful than
　　that at the duplicate post! +1 for
　　improving on something simple! –
　　machine yearning Jun 20 '11 at
　　19:29

1　Sorry for the thread necro, but is
　　there any reason to prefer one
　　method over the other when adding
　　one item? – Warrick Feb 26 '13 at
　　14:01

4　@Warrick there's absolutely no
　　difference except for personal taste.
　　Personally I find the first to be a little
　　more intuitive for just one item. –
　　Chris Eberle Feb 26 '13 at 19:22

　　which is faster? – user3067923 Jul 8
　　'17 at 20:59

1　@user3067923 This is pure
　　conjecture but I *imagine* that the first
　　one would be marginally faster since
　　it's mutating the dict in place,
　　whereas the second one has to
　　create a temporary dict, then
　　mutate, then garbage collect the
　　temporary dict. I'd need to
　　benchmark it to say definitively. –
　　Chris Eberle Sep 6 '17 at 23:35

It can be as simple as:

66

```
default_data['item3'] = 3
```

As [Chris' answer](#) says, you can use [update](#) to add more than one item. An example:

```
default_data.update({'item4': 4, '
```

Please see the docs about [dictionaries as data structures](#) and [dictionaries as built-in types](#).

edited May 23 '17 at 12:18

Community ♦
1   1

answered Jun 20 '11 at 19:09

GreenMatt
13.7k   6   42   70

---

1   Useful for knowing that you can do multiple items in an update. – [tisaconundrum](#) Sep 24 '17 at 14:10

It occurred to me that you may have actually be asking how to implement the `+` operator for dictionaries, the following seems to work:

23

```
>>> class Dict(dict):
...     def __add__(self, other):
...         copy = self.copy()
...         copy.update(other)
...         return copy
...     def __radd__(self, other):
...         copy = other.copy()
...         copy.update(self)
...         return copy
...
>>> default_data = Dict({'item1':
>>> default_data + {'item3': 3}
{'item2': 2, 'item3': 3, 'item1':
>>> {'test1': 1} + Dict(test2=2)
{'test1': 1, 'test2': 2}
```

Note that this is more overhead then using `dict[key] = value` or `dict.update()`, so I would recommend against using this solution unless you intend to create a new dictionary anyway.

2   If you don't want to implement your
     own operators, you can also do
     `dict(default_data.items() +`
     `{'item3': 3}.items())` — Pakman
     Aug 23 '13 at 13:56 ✎

1   @Pakman's comment doesn't work in
     Python3 (see:
     stackoverflow.com/questions/1336151
     0/…) — Prof Feb 21 '16 at 21:15