PYTHON SNIPPETS

Updating Python Dictionaries

Phil Best

2 min read · Aug 12

In this week's Python snippet post, we're going to take a look at the `update` method for Python dictionaries. There are three main ways to use `update` method, which allows us to extend a dictionary with new keys and values.

The first way we're going to look at is using another dictionary object. We can call the `update` method on a dictionary, and pass in this second dictionary object as an argument. The dictionaries will then get merged into a single dictionary object:

```
student = {
    "name": "Rolf",
    "age": 15
}

grades = {
    "english": "A",
    "maths": "B",
    "science": "A*"
}

student.update(grades)
# {'name': 'Rolf', 'age': 15, 'english': 'A', 'maths': 'B', 'science
```

Note that this is an in-place operation, so the original dictionary is modified by this operation. If you need to preserve the original dictionaries, it's therefore a good idea to make a copy.

The second option for using the `update` method is passing in an iterable containing key value pairs. These pairs are in turn stored in some kind of iterable object, such a tuple or list.

```
student = {
    "name": "Rolf",
    "age": 15
}

grades = [("english", "A"), ("maths", "B"), ("science", "A*")]

student.update(grades)
# {'name': 'Rolf', 'age': 15, 'english': 'A', 'maths': 'B', 'science
```

As you can see, we end up with the exact same result.

The final way to extend a dictionary with `update` is to use keyword arguments.

```
student = {
    "name": "Rolf",
    "age": 15
}

student.update(english="A", maths="B", science="A*")
# {'name': 'Rolf', 'age': 15, 'english': 'A', 'maths': 'B', 'science
```

Here each keyword represents a key, and the value is whatever we assigned to that keyword. Once again, the result is identical.

One thing to keep in mind when it comes to using the `update` method with any of these syntax options is that if a key already exists, this will not raise an exception. Instead, the value of that key will be updated. In other words, the old values will be replaced.

Wrapping up

That's it for this one! As you can see, the `update` method is happy to take data in many different forms, which makes it really versatile. It's also a method that we often have use for, so I hope you can experiment with the different ways of using `update` in your own code.

If you're looking to upgrade your Python skills even further, you might want to take a look at our Complete Python Course! We recently released a major update, and we're improving the course every day. We'd love to have you!

If you like our content, I'd also recommend signing up to our mailing list below. Not only will this let you stay up to date with all our content, we also share discount code with our subscribers so that they can get the best deals on all our courses.

Phil Best

I'm a freelance developer, mostly working in web development. I also create course content for Teclado!

Read More

Assignment expressions in Python →

Python's namedtuples →

Python's divmod Function →

→ See all 26 posts

FLASK

Handling the next URL when logging in with Flask

Learn how to redirect users to their intended destination after logging in, instead of a default "profile" page every time. Simple, but can have a big impact in user experience!

Jose Salvatierra
5 min read · Aug 15

Like what you see? Enter your e-mail to hear when new posts come out!

E-mail*

Name*

Receive our newsletter and get notified about new posts we publish on the site, as well as occasional special offers.

Sign Up