PYTHON SNIPPETS

Python Deques

Phil Best
2 min read · Jun 24

Deques, or double ended queues, are a handy collection type from the `collections` module that allow for highly efficient `append` and `pop` operations from either end of the collection. They also come with a very useful `rotate` method used for taking an item from one end of the deque, and appending it to the other end.

The `collections` module is part of Python's Standard Library, but we still need to import it to make use of deques.

```
from collections import deque
```

From here, we can create a deque by passing in an iterable to the `deque` class. This could be a list, a set, or even a string.

Note that to create a deque object containing a single string, we have to pass that string in as an element in another iterable object, otherwise each character will be a separate element in the resulting

```
oops = deque("abc")  # deque(['a', 'b', 'c'])
```

Once we have a deque containing the items we want to work with, we can use many of the same methods available to us when using a regular list. This includes `append`, `count`, `copy`, `clear`, and several others. We do, however, also gain access to some new methods specific to deques.

The first method of note is `appendleft`. This works just like `append`, but places the new element at index `0`. `appendleft` has a counterpart in the form of `popleft`, which functions exactly like `pop(0)` for the standard list. One limitations of deques is that their `pop` method doesn't accept any arguments, and always pops the final item in the collection.

```
base = deque([1, 2, 3])

x = base.pop()  # 3
base.appendleft(x)  # deque([3, 1, 2])

y = base.popleft()  # 3
base.append(y)  # deque([1, 2, 3])
```

Another very interesting method available to us when using deques is `rotate`. `rotate` allows us to `pop` an item from one end of the deque and `append` it to the opposite end.

```
base = deque([1, 2, 3])
base.rotate()  # deque([3, 1, 2])
```

In the example above, 3 was popped from the right and appended to the left.

We're not limited to a single rotation, however, and we can control the

degree and direction of the rotation by passing in a number and associated sign as an argument. By default, a deque rotates to the right, but providing a negative rotation value will cause it to rotate left.

```
base = deque([1, 2, 3, 4, 5])

# rotates base 2 steps to the left
base.rotate(-2)  # deque([3, 4, 5, 1, 2])

# rotates base 3 steps to the right
base.rotate(3)   # deque([5, 1, 2, 3, 4])
```

Wrapping up

And that's it for our brief highlight of deques! If you want to learn more about deques, you can read about them in the official documentation, along with all the other cool collection types in the `collections` module.

We also cover stuff like this in our Complete Python Course, so be to check it out if you want to upgrade your Python skills even further.

If you enjoyed this little post, share it with your friends, and be sure to

Share this

date with all our content!

Phil Best

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

Read More

Python Snippets

Assignment expressions in Python →

Python's namedtuples →

Python's divmod Function →

→ See all 26 posts

CODING INTERVIEW PROBLEMS

Coding Interview Problems: Rotating a List

In this post we tackle another common coding interview problem: rotating a list. Avoid common pitfalls and impress at your next coding interview.

Phil Best
9 min read · Jun 27

Share this