Gmail **Michael Anderson <mganderson87@gmail.com>**

Python Morsels Exercise: with_previous

2 messages

Trey Hunner <trey@truthful.technology> Mon, Jun 4, 2018 at 7:00 AM
To: mganderson87@gmail.com

Hi! 😊

This week I want you to write a function that accepts a sequence (a list for example) and returns a new iterable (anything you can loop over) that includes a tuple of each item and the previous item (the item just before it). The first "previous item" should be None.

For example:

```
>>> with_previous("hello")
[('h', None), ('e', 'h'), ('l', 'e'), ('l', 'l'), ('o', 'l')]
>>> with_previous([1, 2, 3])
[(1, None), (2, 1), (3, 2)]
```

There are three optional bonuses for this exercise.

For the first bonus, make sure you accept any iterable as an argument, not just a sequence (which means you can't use index lookups in your answer). ✔

Here's an example with a generator expression, which is a lazy iterable:

```
>>> with_previous(n**2 for n in [1, 2, 3])
[(1, None), (4, 1), (9, 4)]
```

For the second bonus, I want you to return a lazy iterator (for example a generator) from your with_previous function instead of a list. ✔

This should allow your with_previous function to accept infinitely long iterables. If your function returns an iterator, this should work:

```
>>> next(with_previous([1, 2, 3]))
(1, None)
```

As a third bonus, I want you to allow your with_previous function to accept an optional fillvalue keyword-only argument (defaulting to None). ✔

This should allow your function to work like this:

```
>>> list(with_previous([1, 2, 3], fillvalue=0))
[(1, 0), (2, 1), (3, 2)]
```

But this new argument should be allowed as a keyword argument. This should raise an error:

```
>>> list(with_previous([1, 2, 3], 0))
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: with_previous() takes 1 positional argument but 2 were
given
```

Automated tests for this week's exercise can be found here. You'll need to write your function in a module named with_previous.py next to the test file. To run the tests you'll run "python test_with_previous.py" and check the output for "OK". You'll see that there are some "expected failures" (or "unexpected successes" maybe). If you'd like to do the bonus, you'll want to comment out a line of code in the tests file to test it properly (there's a comment noting which line).

Michael Anderson <mganderson87@gmail.com> Mon, Jun 4, 2018 at 12:28 PM
To: Jeury Mejia <jeurymejia@gmail.com>

[Quoted text hidden]