Python Morsels problem: compact

1 message

trey@pythonmorsels.com <trey@pythonmorsels.com> 29 October 2018 at 15:03
To: jason.tokarz@gmail.com

Hi! ✨

For this week's exercise 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) with adjacent duplicate values removed.

For example:

```
>>> compact([1, 1, 1])
[1]
>>> compact([1, 1, 2, 2, 3, 2])
[1, 2, 3, 2]
>>> compact([])
[]
```

There are two bonuses for this exercise.

I recommend solving the exercise without the bonuses first and then attempting each bonus separately.

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

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

```
>>> compact(n**2 for n in [1, 2, 2])
[1, 4]
```

As a second bonus, make sure you return an iterator (for example a generator) from your compact function instead of a list. ✔

This should allow your compact function to accept infinitely long iterables (or other lazy iterables).

Automated tests for this week's exercise can be found here. You'll need to write your function in a module named compact.py next to the test file. To run the tests you'll run "python test_compact.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 the noted lines of code in the tests file to test them properly.
