

---

## Python Morsels problem: get\_earliest

1 message

---

**trey@pythonmorsels.com** <trey@pythonmorsels.com>  
To: jason.tokarz@gmail.com

15 October 2018 at 16:04

Hey!

Your first exercise is a somewhat silly one. I'd like you to compare date strings, but allow invalid dates while comparing them.

Make sure you read all the way to the end of this email, because I've linked to some automated tests to help you ensure you've solved this exercise correctly.

I want you to write a function that takes two strings representing dates and returns the string that represents the earliest point in time. The strings are in the US-specific MM/DD/YYYY format... just to make things harder. Note that the month, year, and day will always be represented by 2, 4, and 2 digits respectively.

Your function should work like this:

```
>>> get_earliest("01/27/1832", "01/27/1756")
"01/27/1756"
>>> get_earliest("02/29/1972", "12/21/1946")
"12/21/1946"
>>> get_earliest("02/24/1946", "03/21/1946")
"02/24/1946"
>>> get_earliest("06/21/1958", "06/24/1958")
"06/21/1958"
```

There's a catch though. Your exercise **should work with invalid month and date combinations**. What I mean by that is that dates like 02/40/2006 should be supported. By that I mean 02/40/2006 is before 03/01/2006 but after 02/30/2006 (dates don't rollover at all). I'm adding this requirement so you can't rely on Python's datetime module.

There are many ways to solve this one. See if you can figure out the clearest and most idiomatic way to solve this exercise. ✨

If you complete the main exercise, there's also a bonus for you to attempt: allow the function to accept any number of arguments and return the earliest date string of all provided. ✓

So if you complete the bonus, this should work:

```
>>> get_earliest("02/24/1946", "01/29/1946", "03/29/1945")
"03/29/1945"
```

I've written some tests to make it easier to ensure your code functions as expected. You can [download the test file here](#). You'll need to write your function in a file named `earliest.py` next to where you've saved that test file. To run the tests you'll run `python test_earliest.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 to test them properly. You'll see that noted in the test file.

You'll receive some answers and links to resources explaining ways to solve this exercise within a few days. Don't peek at the answers before attempting to solve this on your own.

---

You can also [view the problem statement for this exercise](#) on the Python Morsels website.

To make sure you keep getting these emails, please add [trey@pythonmorsels.com](mailto:trey@pythonmorsels.com) to your address book.

If you want to unsubscribe from Python Morsels [click here](#)