

Documentation Exercises

Paul Ganssle

  @pganssle

This talk on Github: [pganssle-talks/doc-exercises-lightning](https://github.com/pganssle-talks/doc-exercises-lightning)

New users are not experts in your library



New users are documentation experts



ganssle.io



@pganssle

Mutually beneficial arrangement?



Documentation Exercises

Simple exercises or problem sets demonstrating a problem that can be solved by your project...

Documentation Exercises

Simple exercises or problem sets demonstrating a problem that can be solved by your project...

...to be completed *only using your documentation.*

Exercises

It is often useful to work through some examples in order to understand how a module works; on this page, there are several exercises of varying difficulty that you can use to learn how to use `dateutil`.

If you are interested in helping improve the documentation of `dateutil`, it is recommended that you attempt to complete these exercises with no resources *other than dateutil's documentation*. If you find that the documentation is not clear enough to allow you to complete these exercises, open an issue on the [dateutil issue tracker](#) to let the developers know what part of the documentation needs improvement.

Next Monday meeting

A team has a meeting at 10 AM every Monday and wants a function that tells them, given a `datetime.datetime` object, what is the date and time of the next Monday meeting? This is probably best accomplished using a [relativedelta] (<https://dateutil.readthedocs.io/en/stable/relativedelta.html>).



Next Monday meeting

A team has a meeting at 10 AM every Monday and wants a function that tells them, given a `datetime.datetime` object, what is the date and time of the next Monday meeting? This is probably best accomplished using a [relativedelta] (<https://dateutil.readthedocs.io/en/stable/relativedelta.html>).

```
# ----- YOUR CODE ----- #
from dateutil import relativedelta

def next_monday(dt):
    <<YOUR CODE HERE>>

# ----- #

from datetime import datetime
from dateutil import tz

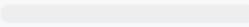
NEXT_MONDAY_CASES = [
    ... # Skipped test cases for space reasons
]

def test_next_monday_1():
    for dt_in, dt_out in NEXT_MONDAY_CASES:
        assert next_monday(dt_in) == dt_out

if __name__ == "__main__":
    test_next_monday_1()
    print('Success!')
```

dateutil experience

 **Reworded wkst documentation** ✓ documentation rrule

#844 opened on Dec 7, 2018 by SavenR  0 of 3 

 **#623: added documentation for rrulestr function** ✓ documentation rrule

#762 by prdickson was merged on Jun 20, 2018  2 of 2   2.8.0

 **Improve relativedelta documentation** ✓ documentation relativedelta

#759 by andrewcbennett was merged on Jun 21, 2018  2 of 2   2.8.0

 **Relativedelta doc typo** ✓ documentation relativedelta

#758 by nherriot was merged on Jun 18, 2018  2.8.0

 **Improved docstrings for count and until parameters** ✓ documentation

#755 by lucaferocino was merged on Jun 8, 2018  2 of 2 

 **change the order of cache and byester in rrule docstring** ✓ documentation rrule

#695 by rmahajan14 was merged on Apr 22, 2018  Master

 **Tuple documentation fix** ✓ documentation

#673 by ElliotJH was merged on Apr 16, 2018  Master

dateutil experience

Tuple documentation fix #673 Edit

Merged pganssle merged 4 commits into `dateutil:master` from `ElliotJH:tuple-documentation-fix` on Apr 16, 2018

Conversation 4 Commits 4 Checks 0 Files changed 3 +38 -26

Changes from all commits File filter... Clear filters Jump to... 0 / 3 files viewed Review changes

> 3 AUTHORS.md Viewed

▼ 4 changelog.d/673.doc.rst Viewed

```
... @@ -0,0 +1,4 @@  
1 +Updated the documentation for relativedelta. Removed references to  
2 +tuple arguments for weekday, explained effect of weekday(_, 1) and  
3 +better explained the order of operations that relativedelta  
4 +applies. Fixed by @kvn219 @huangy22 and @ElliotJH (gh pr #673)
```

Why you should do this?

- Great way to onboard people at sprints
- Signals to your users that you value their experience.
- Signals that they can contribute *from day 1*.



Why you should do this?

- Great way to onboard people at sprints
- Signals to your users that you value their experience.
- Signals that they can contribute *from day 1*.

I'm convinced - what else can I do?

Why you should do this?

- Great way to onboard people at sprints
- Signals to your users that you value their experience.
- Signals that they can contribute *from day 1*.

I'm convinced - what else can I do?

Spread the word!