monthday Documentation

Release 0.9.0

Hong Minhee

November 28, 2015

Contents

This package provides the `MonthDay` value type for dealing dates without year. It is useful for dealing with birthdays, or anniversaries. Works on Python 2.6, 2.7, 3.2–3.5, PyPy, PyPy3.

```
>>> from monthday import *
>>> aug_4 = MonthDay(8, 4)
>>> aug_4
monthday.MonthDay(8, 4)
>>> aug_4.date(1988)
datetime.date(1988, 8, 4)
>>> list(aug_4.dates(range(2013, 2016)))
[datetime.date(2013, 8, 4),
 datetime.date(2014, 8, 4),
 datetime.date(2015, 8, 4)]
>>> from datetime import date
>>> MonthDay.from_date(date(2015, 12, 25))
monthday.MonthDay(12, 25)
```

It's available on PyPI:

```
$ pip install monthday
```

Written by Hong Minhee, and distributed under LGPLv3 or later. Find the source code from the GitHub repository.

CHAPTER 1

monthday — Date without year

class monthday.**MonthDay**(*month*, *day*)

Date without year. Useful for birthdays, or anniversaries.

Parameters

- **month** (numbers.Integral) – a month number, from 1 to 12
- **day** (numbers.Integral) – a day of the month, from 1 to 31 (or 30, or 29)

Raises ValueError if month or date is out of valid range

month

(numbers.Integral) The month number, from 1 to 12.

day

(numbers.Integral) The day of the month, from 1 to 31.

date(*year*)

Get a date by combining the given year with it.

```
>>> MonthDay(12, 25).date(2015)
datetime.date(2015, 12, 25)
```

It may raise ValueError if February 29 is tried to be combined with a non-leap year e.g.:

```
>>> feb_29 = MonthDay(2, 29)
>>> feb_29.date(2012)
datetime.date(2012, 2, 29)
>>> feb_29.date(2013)
Traceback (most recent call last):
  ...
ValueError: since 2013 is not a leap year,
            monthday.MonthDay(2, 29) can't be combined with 2013
```

Parameters **year** (numbers.Integral) – a year to combine with

Returns a datetime.date with the given year

Return type datetime.date

Raises ValueError when year is not a leap year while it's MonthDay(2, 29)

dates(*years*, *error_invalid_dates=True*)

Get dates by combining the given years with it.

```
>>> list(MonthDay(8, 4).dates(range(1988, 1992)))
[datetime.date(1988, 8, 4), datetime.date(1989, 8, 4),
 datetime.date(1990, 8, 4), datetime.date(1991, 8, 4)]
```

It may raise `ValueError` if there happen to be any invalid dates in the result, e.g. Feburary 29 for non-leap years:

```
>>> feb_29 = MonthDay(2, 29)
>>> list(feb_29.dates(range(2011, 2017)))
Traceback (most recent call last):
  ...
ValueError: since 2010 is not a leap year,
            monthday.MonthDay(2, 29) can't be combined with 2010
```

If you want to simply ignore these invalid dates in the result, set `error_invalid_dates` to `False` e.g.:

```
>>> list(feb_29.dates(range(2011, 2017), error_invalid_dates=False))
[datetime.date(2012, 2, 29), datetime.date(2016, 2, 29)]
```

But the result length might be shorter than the input `years` list.

If you want to match the length of the input and the result, set `error_invalid_dates` to `None` — it will replace invalid dates in the result with `None` values e.g.:

```
>>> list(feb_29.dates(range(2011, 2017), error_invalid_dates=None))
[None, datetime.date(2012, 2, 29),
 None, None, None, datetime.date(2016, 2, 29)]
```

Parameters

- **years** (`Iterable`) – years to combine with
- **error_invalid_dates** (`bool`, `type(None)`) – if set to `True`, raise `ValueError` for invalid dates. if set to `False`, just ignore invalid dates — the result length might be shorter than the input `years` list. if set to `None`, fill `None` values instead of invalid dates — the result length must be the same to the input `year` list. `True` by default

Returns `datetime.date` values with the given `years`. the order corresponds to the input `years`' order

Return type `Iterable`

Raises

- **ValueError** – if `error_invalid_dates` is set to `True` and there happend to be any invalid dates in the result
- **TypeError** – if `years` is not iterable of integers

classmethod from_date(*date*)

Get only *MonthDay* from the given `date`.

Parameters **date** (`datetime.date`, `datetime.datetime`) – the date or date/time

Returns *MonthDay* without `date`'s `year`

Return type *MonthDay*

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

m

D

F

M