
python*exampleDocumentation*

Release 0.1.1

Sylvain Corlay

Jun 03, 2020

CONTENTS:

1	API Reference	3
2	Similar Libraries	5
3	Installation (Python)	7
4	Indices and tables	9
	Python Module Index	11
	Index	13



API REFERENCE

class daylight.Sunclock (*self*: daylight.Sunclock, latitude: float, longitude: float, tz_offset: float = 0) → None

Constructs a Sunclock object

Parameters

- **latitude** (float) – latitude of the location
- **longitude** (float) – longitude of the location
- **tz_offset** (float) – in hours, east is positive. i.e. IST is +5.5

irradiance (*self*: daylight.Sunclock, datetime: numpy.ndarray[int64]) → object

Calculates the irradiance level for given datetime.

This is a vectorized function.

Parameters **datetime** (int or array_like) – unix timestamp (in UTC, seconds)

Examples

```
>>> import daylight
>>> sun = daylight.Sunclock(17.3859, 78.4867, 5.5)
>>> sun.irradiance(1590050435)
0.882753920406182
>>> sun.irradiance([1590010200, 1590024600])
array([-0.56570521,  0.28650605])
```

solar_noon (*self*: daylight.Sunclock, date: numpy.ndarray[int64]) → object

Calculates the solar noon time for given date.

This is a vectorized function.

Parameters **date** (int or array_like) – unix timestamp (in UTC, seconds)

Examples

```
>>> import daylight
>>> sun = daylight.Sunclock(17.3859, 78.4867, 5.5)
>>> sun.solar_noon(1589999400)
1590043354
>>> sun.solar_noon([1590010200, 1590024600])
array([1590043355, 1590043355])
```

sunrise (*self*: `daylight.Sunclock`, *date*: `numpy.ndarray[int64]`) → object
Calculates the sunrise time for given date.

This is a vectorized function.

Parameters *date* (*int* or *array_like*) – unix timestamp (in UTC, seconds)

Examples

```
>>> import daylight
>>> sun = daylight.Sunclock(17.3859, 78.4867, 5.5)
>>> sun.sunrise(1589999400)
1590019961
>>> sun.sunrise([1590010200, 1590024600])
array([1590019959, 1590019957])
```

sunset (*self*: `daylight.Sunclock`, *date*: `numpy.ndarray[int64]`) → object
Calculates the sunset time for given date.

This is a vectorized function.

Parameters *date* (*int* or *array_like*) – unix timestamp (in UTC, seconds)

Examples

```
>>> import daylight
>>> sun = daylight.Sunclock(17.3859, 78.4867, 5.5)
>>> sun.sunset(1589999400)
1590066748
>>> sun.sunset([1590010200, 1590024600])
array([1590066751, 1590066754])
```


SIMILAR LIBRARIES

2.1 PyEphem

GitHub: <https://github.com/brandon-rhodes/pyephem>

PyEphem is, comparatively, a much more sophisticated and full-blown astronomical engine daylight, on the other hand, is focused on much smaller subset of problems

2.2 pvlib-python

GitHub: <https://github.com/pvlib/pvlib-python>

Aimed at photovoltaic energy system development, it includes many things which you might not need, so if your use cases is to just compute brightness levels of the day - it could be an overkill

Additionally, it is written in pure python, compared to daylight which benefits from the portability and efficiency of a C++ based implementation

2.3 solarpy

GitHub: <https://github.com/aqreed/solarpy>

- Can calculate for any orientation of surface plane
- Seems to only consider latitude when computing irradiance levels
- Written in pure python

daylight or libdaylight is a library which enables you to answer daylight related questions like:

- Irradiance - How bright is it outside, given a particular time and location?
- What is the time of sunrise/solarnoon/sunset, given a particular time and location?

For basic usage via Python or C APIs, check [README](#) on our GitHub repository.

INSTALLATION (PYTHON)

```
pip install git+https://github.com/adonmo/daylight
```


INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

d

daylight, 3

INDEX

D

`daylight`
 module, 3

I

`irradiance()` (*daylight.Sunclock method*), 3

M

module
 daylight, 3

S

`solar_noon()` (*daylight.Sunclock method*), 3

`Sunclock` (*class in daylight*), 3

`sunrise()` (*daylight.Sunclock method*), 4

`sunset()` (*daylight.Sunclock method*), 4