python-networkdays

Release 0.1

github.com/cadu-leite

Jun 25, 2022

CONTENTS:

CHAPTER

ONE

NETWORKDAYS

class networkdays.networkdays.**JobSchedule**(*project_duration_hours*, *workhours_per_day*, *date_start*, *networkdays=None*)

days()

job_workdays()
list workdays for a given job duration

Returns workday datetime.date list

Return type list

months(*year=None*)
return a weeks *iterATOR*

Parameters **year** (*None, optional*) – Description

Returns Description

Return type TYPE

weeks(*year=None*, *month=None*)
return an *interator* for ISO format see https://docs.python.org/3/library/datetime.html#datetime.date.isocalendar)

Parameters

- **year** (*None, optional*) – filter per year
- **month** (*None, optional*) – filter per month

Returns weeks iso numbers based

Return type iter

years()
Its not duration

class networkdays.networkdays.**Networkdays**(*date_start*, *date_end=None*, *holidays={}*, *weekdaysoff={6, 7}*)

holidays()

networkdays()
NetWorkDays like Excel Networkdays function. given 2 dates, the return will the number of days between dates, minus holidays, e week days off (ex.: saturday and sunday).

The *weekdaysoff* is a per week ISO days list where Monday is 1 and sunday is 7. The holidays may be any single date, datetime.date object, in a year.

Parameters

- **date_start** (*datetime.date*) – initial date
- **date_end** (*datetime.date*) – end date, or if none, is the last day of the date_start year.
- **holidays** (*sipytho net*) – list of datetime object, indicating days off.
- **weekdaysoff** (*set*) – list of weekdays not working, default is Saturday and Sunday {6,7}.

Returns list of work days.

ex.:

networkdays(datetime.date(2020,1,1), datetime.date(2020,2,31), holiday=datetime.date(2020,1,1), weekdaysoff={6,7}

)

weekends ( )

- Business days calendar.
- JobSchedule on business days.

Tip: Just Python built-in libs, no dependencies

Networkdays: Return working days between two dates exclude weekends and holidays.

- just like spreadsheets *networkdays* function
- exclude Holidays
- Exclude "days off" per week.

Job schedule: Calculate the period for a given job hours, based on *Networkdays*.

Table of Contents

CHAPTER

TWO

INSTALLATION

python-networkdays can be installed from PyPI using pip

```
pip install python-networkdays
```

Tip: note that the package name is different from the importable name

Page on Pypi: https://pypi.org/project/python-networkdays/

There is no dependencies.

CHAPTER

THREE

FEATURES

- Return a list of business days between 2 dates.
- Exclude weekends by default
- Custom “days off” may be informed as list like {1,2,3,4,5,6,7}, where 1 is Monday default is {6,7} = (Sat, Sun).
- How many business days between two dates.
- How many days off, including holidays and weekends.
- Return a list of business days for a given number of hours
- Return a list of Years, months or weeks for a given number of hours
- **No Pandas or NumPy dependencies**

CHAPTER

FOUR

EXAMPLES

4.1 Networkdays.networkdays()

```
In [1]: from networkdays import networkdays

In [2]: import datetime

In [3]: HOLIDAYS = { datetime.date(2020, 12, 25) }  # define a Holidays list

# initiate  class::`networkdays.Networkdays`
In [4]: days = networkdays.Networkdays(
            datetime.date(2020, 12, 15),  # start date
            datetime.date(2020, 12, 31),  # end date
            HOLIDAYS  # list of Holidays
        )

In [5]: days.networkdays()  # return a list os workdays
Out[5]:
[datetime.date(2020, 12, 15),
 datetime.date(2020, 12, 16),
 datetime.date(2020, 12, 17),
 datetime.date(2020, 12, 18),
 datetime.date(2020, 12, 21),
 datetime.date(2020, 12, 22),
 datetime.date(2020, 12, 23),
 datetime.date(2020, 12, 24),
 datetime.date(2020, 12, 28),
 datetime.date(2020, 12, 29),
 datetime.date(2020, 12, 30),
 datetime.date(2020, 12, 31)]

In [6]: days.weekends()  # list os Weekends (default = Saturday ans Sunday)
Out[6]:
[datetime.date(2020, 12, 19),
 datetime.date(2020, 12, 20),
 datetime.date(2020, 12, 26),
 datetime.date(2020, 12, 27)]

In [7]: days.holidays()
Out[7]: [datetime.date(2020, 12, 25)] # list of holidays
```

4.2 Networkdays.jobschedule()

```
# jobSchedule
import datetime
from networkdays import networkdays

# Distribute the 600 hrs of effort, starting on december 1, 2020 working 8hrs per day.
jobschedule = networkdays.JobSchedule(600, 8, datetime.date(2020, 12, 1),␣
↪networkdays=None)
job_dates = jobschedule.job_workdays()

# print results ...
print(f'''

bussines days:          {jobschedule.bussines_days}
calendar days:          {jobschedule.total_days}
starts - ends:          {jobschedule.prj_starts} - {jobschedule.prj_ends}

years:                  {list(jobschedule.years())}
months:                 {list(jobschedule.months())}
weeks (ISO):            {list(jobschedule.weeks())}

days:
    {list(jobschedule.days())[:2]} ...\n ...{list(jobschedule.days())[-2:]}

Works days dates on january:
    {list(jobschedule.days())[:2]} ...\n ...{list(jobschedule.days())[-2:]}
''')
```

```
bussines days:          54
calendar days:          73 days, 0:00:00
starts - ends:          12/01/20 - 02/12/21

years:                  [2020, 2021]
months:                 [12, 1, 2]
weeks (ISO):            [49, 50, 51, 52, 53, 1, 2, 3, 4, 5, 6]

days:
    [datetime.date(2020, 12, 1), datetime.date(2020, 12, 2)] ...
 ...[datetime.date(2021, 2, 11), datetime.date(2021, 2, 12)]

Works days dates on january:
    [datetime.date(2020, 12, 1), datetime.date(2020, 12, 2)] ...
 ...[datetime.date(2021, 2, 11), datetime.date(2021, 2, 12)]
```

CHAPTER

FIVE

OTHER SIMILAR PROJECTS

When I start to code, I did check for some similar projects.

I knew about python-dateutil, a great project I use for years... I'd like something more straightforward or simpler.

After to publish the python-networkdays on PyPi I found some others 8(

- workdays : A 5 years old project, looks the same as **networkdays_**
- timeboard : A more complex but powerful project
- python-dateutil is great, powerful but even more complex.
- python-bizdays : Quick simple and direct ...

I will try to keep this list updated...

CHAPTER

SIX

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

n

INDEX