pyluach

Release 2.2.0.post1

MS List

Mar 01, 2023

CONTENTS

CHAPTER

ONE

PYLUACH

Pyluach is a Python package for dealing with Hebrew (Jewish) calendar dates.

1.1 Features

- Conversion between Hebrew and Gregorian dates
- Finding the difference between two dates
- Finding a date at a given duration from the given date
- Rich comparisons between dates
- Finding the weekday of a given date
- Finding the weekly Parsha reading of a given date
- Getting the holiday occuring on a given date
- Generating html and text Hebrew calendars

1.2 Installation

Use `pip install pyluach`.

1.3 Documentation

Documentation for pyluach can be found at https://readthedocs.org/projects/pyluach/.

1.4 Examples

```
>>> from pyluach import dates, hebrewcal, parshios

>>> today = dates.HebrewDate.today()
>>> lastweek_gregorian = (today - 7).to_greg()
>>> lastweek_gregorian < today
    True
>>> today - lastweek_gregorian
7
>>> greg = dates.GregorianDate(1986, 3, 21)
>>> heb = dates.HebrewDate(5746, 13, 10)
>>> greg == heb
True

>>> purim = dates.HebrewDate(5781, 12, 14)
>>> purim.hebrew_day()
''
>>> purim.hebrew_date_string()
'  '
>>> purim.hebrew_date_string(True)
'  '

>>> rosh_hashana = dates.HebrewDate(5782, 7, 1)
>>> rosh_hashana.holiday()
'Rosh Hashana'
>>> rosh_hashana.holiday(hebrew=True)
' '
>>> (rosh_hashana + 3).holiday()
None

>>> month = hebrewcal.Month(5781, 10)
>>> month.month_name()
'Teves'
>>> month.month_name(True)
''
>>> month + 3
Month(5781, 1)
>>> for month in hebrewcal.Year(5774).itermonths():
...     print(month.month_name())
Tishrei Cheshvan ...

>>> date = dates.GregorianDate(2010, 10, 6)
>>> parshios.getparsha(date)
[0]
>>> parshios.getparsha_string(date, israel=True)
'Beraishis'
>>> parshios.getparsha_string(date, hebrew=True)
''
>>> new_date = dates.GregorianDate(2021, 3, 10)
>>> parshios.getparsha_string(new_date)
'Vayakhel, Pekudei'
```

(continues on next page)

(continued from previous page)

```
>>> parshios.getparsha_string(new_date, hebrew=True)
', '
```

1.5 Contact

For questions and comments please raise an issue in github or contact me at simlist@gmail.com.

1.6 License

Pyluach is licensed under the MIT license.

1.6.1 dates module

The dates module implements classes for representing and manipulating several date types.

Contents

Note: All instances of the classes in this module should be treated as read only. No attributes should be changed once they're created.

---

class pyluach.dates.**Rounding**(*value*, *names=None*, ***, *module=None*, *qualname=None*, *type=None*, *start=1*, *boundary=None*)

Bases: Enum

Enumerator to provide options for rounding Hebrew dates.

This provides constants to use as arguments for functions. It should not be instantiated.

PREVIOUS_DAY

If the day is the 30th and the month only has 29 days, round to the 29th of the month.

NEXT_DAY

If the day is the 30th and the month only has 29 days, round to the 1st of the next month.

EXCEPTION

If the day is the 30th and the month only has 29 days, raise a ValueError.

class pyluach.dates.**BaseDate**

Bases: ABC

BaseDate is a base class for all date types.

It provides the following arithmetic and comparison operators common to all child date types.

Operation	Result
d2 = date1 + int	New date `int` days after date1
d2 = date1 - int	New date `int` days before date1
int = date1 - date2	Positive integer equal to the duration from date1 to date2
date1 > date2	True if date1 occurs later than date2
date1 < date2	True if date1 occurs earlier than date2
date1 == date2	True if date1 occurs on the same day as date2
date1 != date2	True if `date1 == date2` is False

Any subclass of `BaseDate` can be compared to and diffed with any other subclass date.

abstract property jd

Return julian day number.

Returns
The Julian day number at midnight (as `n.5`).

Return type
float

abstract to_heb()

Return Hebrew Date.

Return type
HebrewDate

weekday()

Return day of week as an integer.

Returns
An integer representing the day of the week with Sunday as 1 through Saturday as 7.

Return type
int

isoweekday()

Return the day of the week corresponding to the iso standard.

Returns
An integer representing the day of the week where Monday is 1 and and Sunday is 7.

Return type
int

shabbos()

Return the Shabbos on or following the date.

Returns
self if the date is Shabbos or else the following Shabbos as the same date type as called from.

Return type
JulianDay, *GregorianDate*, or *HebrewDate*

Examples

```
>>> heb_date = HebrewDate(5781, 3, 29)
>>> greg_date = heb_date.to_greg()
>>> heb_date.shabbos()
HebrewDate(5781, 4, 2)
>>> greg_date.shabbos()
GregorianDate(2021, 6, 12)
```

fast_day(*hebrew=False*)

Return name of fast day of date.

Parameters

hebrew (*bool, optional*) – `True` if you want the fast day name in Hebrew letters. Default is `False`, which returns the name transliterated into English.

Returns

The name of the fast day or `None` if the date is not a fast day.

Return type

str or None

festival(*israel=False*, *hebrew=False*, *include_working_days=True*, *prefix_day=False*)

Return name of Jewish festival of date.

This method will return all major and minor religous Jewish holidays not including fast days.

Parameters

- **israel** (*bool, optional*) – `True` if you want the holidays according to the Israel schedule. Defaults to `False`.
- **hebrew** (*bool, optional*) – `True` if you want the festival name in Hebrew letters. Default is `False`, which returns the name transliterated into English.
- **include_working_days** (*bool, optional*) – `True` to include festival days on which melacha (work) is allowed; ie. Pesach Sheni, Chol Hamoed, etc. Default is `True`.
- **prefix_day** (*bool, optional*) – `True` to prefix multi day festivals with the day of the festival. Default is `False`.

Returns

The name of the festival or `None` if the given date is not a Jewish festival.

Return type

str or None

Examples

```
>>> pesach = HebrewDate(2023, 1, 15)
>>> pesach.festival(prefix_day=True)
'1 Pesach'
>>> pesach.festival(hebrew=True, prefix_day=True)
' '
>>> shavuos = HebrewDate(5783, 3, 6)
>>> shavuos.festival(israel=True, prefix_day=True)
'Shavuos'
```

holiday(*israel=False*, *hebrew=False*, *prefix_day=False*)

Return name of Jewish holiday of the date.

The holidays include the major and minor religious Jewish holidays including fast days.

Parameters

- **israel** (*bool, optional*) – `True` if you want the holidays according to the Israel schedule. Defaults to `False`.
- **hebrew** (*bool, optional*) – `True` if you want the holiday name in Hebrew letters. Default is `False`, which returns the name transliterated into English.
- **prefix_day** (*bool, optional*) – `True` to prefix multi day holidays with the day of the holiday. Default is `False`.

Returns

The name of the holiday or `None` if the given date is not a Jewish holiday.

Return type

str or None

Examples

```
>>> pesach = HebrewDate(2023, 1, 15)
>>> pesach.holiday(prefix_day=True)
'1 Pesach'
>>> pesach.holiday(hebrew=True, prefix_day=True)
' '
>>> taanis_esther = HebrewDate(5783, 12, 13)
>>> taanis_esther.holiday(prefix_day=True)
'Taanis Esther'
```

class pyluach.dates.**CalendarDateMixin**(*year*, *month*, *day*, *jd=None*)

Bases: `object`

CalendarDateMixin is a mixin for Hebrew and Gregorian dates.

Parameters

- **year** (*int*) –
- **month** (*int*) –
- **day** (*int*) –

year

Type

int

month

Type

int

day

Type

int

jd

The equivalent Julian day at midnight.

Type
float

tuple()

Return date as tuple.

Returns
A tuple of ints in the form `(year, month, day)`.

Return type
tuple of ints

dict()

Return the date as a dictionary.

Returns
A dictionary in the form `{'year': int, 'month': int, 'day': int}`.

Return type
dict

replace(*year=None*, *month=None*, *day=None*)

Return new date with new values for the specified field.

Parameters

- **year** (*int, optional*) –
- **month** (*int, optional*) –
- **day** (*int, optional*) –

Returns

- *CalendarDateMixin* – Any date that inherits from CalendarDateMixin (`GregorianDate`, ``HebrewDate`).
- *Raises*
- *ValueError* – Raises a `ValueError` if the new date does not exist.

class pyluach.dates.**JulianDay**(*day*)

Bases: *BaseDate*

A JulianDay object represents a Julian Day at midnight.

Parameters
day (*float or int*) – The julian day. Note that Julian days start at noon so day number 10 is represented as 9.5 which is day 10 at midnight.

day

The Julian Day Number at midnight (as *n*.5)

Type
float

property jd

Return julian day.

Return type
float

static from_pydate(*pydate*)

Return a *JulianDay* from a python date object.

Parameters
pydate (*datetime.date*) – A python standard library `datetime.date` instance

Return type
JulianDay

static today()

Return instance of current Julian day from timestamp.

Extends the built-in `datetime.date.today()`.

Returns
A JulianDay instance representing the current Julian day from the timestamp.

Return type
JulianDay

> **Warning:** Julian Days change at noon, but pyluach treats them as if they change at midnight, so at midnight this method will return `JulianDay(n.5)` until the following midnight when it will return `JulianDay(n.5 + 1)`.

to_greg()

Convert JulianDay to a Gregorian Date.

Returns
The equivalent Gregorian date instance.

Return type
GregorianDate

Notes

This method uses the Fliegel-Van Flandern algorithm.

to_heb()

Convert to a Hebrew date.

Returns
The equivalent Hebrew date instance.

Return type
HebrewDate

to_pydate()

Convert to a datetime.date object.

Returns
A standard library `datetime.date` instance.

Return type
datetime.date

fast_day(*hebrew=False*)

Return name of fast day of date.

Parameters
hebrew (*bool, optional*) – `True` if you want the fast day name in Hebrew letters. Default is `False`, which returns the name transliterated into English.

Returns
The name of the fast day or `None` if the date is not a fast day.

Return type
str or None

festival(*israel=False*, *hebrew=False*, *include_working_days=True*, *prefix_day=False*)

Return name of Jewish festival of date.

This method will return all major and minor religous Jewish holidays not including fast days.

Parameters

- **israel** (*bool, optional*) – `True` if you want the holidays according to the Israel schedule. Defaults to `False`.
- **hebrew** (*bool, optional*) – `True` if you want the festival name in Hebrew letters. Default is `False`, which returns the name transliterated into English.
- **include_working_days** (*bool, optional*) – `True` to include festival days on which melacha (work) is allowed; ie. Pesach Sheni, Chol Hamoed, etc. Default is `True`.
- **prefix_day** (*bool, optional*) – `True` to prefix multi day festivals with the day of the festival. Default is `False`.

Returns
The name of the festival or `None` if the given date is not a Jewish festival.

Return type
str or None

Examples

```
>>> pesach = HebrewDate(2023, 1, 15)
>>> pesach.festival(prefix_day=True)
'1 Pesach'
>>> pesach.festival(hebrew=True, prefix_day=True)
' '
>>> shavuos = HebrewDate(5783, 3, 6)
>>> shavuos.festival(israel=True, prefix_day=True)
'Shavuos'
```

holiday(*israel=False*, *hebrew=False*, *prefix_day=False*)

Return name of Jewish holiday of the date.

The holidays include the major and minor religious Jewish holidays including fast days.

Parameters

- **israel** (*bool, optional*) – `True` if you want the holidays according to the Israel schedule. Defaults to `False`.
- **hebrew** (*bool, optional*) – `True` if you want the holiday name in Hebrew letters. Default is `False`, which returns the name transliterated into English.

- **prefix_day** (*bool, optional*) – `True` to prefix multi day holidays with the day of the holiday. Default is `False`.

Returns
The name of the holiday or `None` if the given date is not a Jewish holiday.

Return type
str or None

Examples

```
>>> pesach = HebrewDate(2023, 1, 15)
>>> pesach.holiday(prefix_day=True)
'1 Pesach'
>>> pesach.holiday(hebrew=True, prefix_day=True)
' '
>>> taanis_esther = HebrewDate(5783, 12, 13)
>>> taanis_esther.holiday(prefix_day=True)
'Taanis Esther'
```

`isoweekday()`

Return the day of the week corresponding to the iso standard.

Returns
An integer representing the day of the week where Monday is 1 and and Sunday is 7.

Return type
int

`shabbos()`

Return the Shabbos on or following the date.

Returns
self if the date is Shabbos or else the following Shabbos as the same date type as called from.

Return type
JulianDay, *GregorianDate*, or *HebrewDate*

Examples

```
>>> heb_date = HebrewDate(5781, 3, 29)
>>> greg_date = heb_date.to_greg()
>>> heb_date.shabbos()
HebrewDate(5781, 4, 2)
>>> greg_date.shabbos()
GregorianDate(2021, 6, 12)
```

`weekday()`

Return day of week as an integer.

Returns
An integer representing the day of the week with Sunday as 1 through Saturday as 7.

Return type
int

class pyluach.dates.**GregorianDate**(*year*, *month*, *day*, *jd=None*)

Bases: *BaseDate*, *CalendarDateMixin*

A GregorianDate object represents a Gregorian date (year, month, day).

This is an idealized date with the current Gregorian calendar infinitely extended in both directions.

Parameters

- **year** (*int*) –
- **month** (*int*) –
- **day** (*int*) –
- **jd** (*float, optional*) – This parameter should not be assigned manually.

year

Type
int

month

Type
int

day

Type
int

> **Warning:** Although B.C.E. dates are allowed, they should be treated as approximations as they may return inconsistent results when converting between date types and using arithmetic and comparison operators!

strftime(*fmt*)

Return formatted date.

Wraps `datetime.date.strftime()` method and uses the same format options.

Parameters
fmt (*str*) – The format string.

Return type
str

property jd

Return the corresponding Julian day number.

Returns
The Julian day number at midnight.

Return type
float

classmethod from_pydate(*pydate*)

Return a *GregorianDate* instance from a python date object.

Parameters
pydate (*datetime.date*) – A python standard library `datetime.date` instance.

Return type
GregorianDate

`static today()`

Return a GregorianDate instance for the current day.

This static method wraps the Python standard library's date.today() method to get the date from the timestamp.

Returns
The current Gregorian date from the computer's timestamp.

Return type
GregorianDate

`is_leap()`

Return if the date is in a leap year

Returns
True if the date is in a leap year, False otherwise.

Return type
bool

`to_jd()`

Convert to a Julian day.

Returns
The equivalent JulianDay instance.

Return type
JulianDay

`to_heb()`

Convert to Hebrew date.

Returns
The equivalent HebrewDate instance.

Return type
HebrewDate

`to_pydate()`

Convert to a standard library date.

Returns
The equivalent datetime.date instance.

Return type
datetime.date

`dict()`

Return the date as a dictionary.

Returns
A dictionary in the form `{'year': int, 'month': int, 'day': int}`.

Return type
dict

fast_day(*hebrew=False*)

Return name of fast day of date.

Parameters
hebrew (*bool, optional*) – `True` if you want the fast day name in Hebrew letters. Default is `False`, which returns the name transliterated into English.

Returns
The name of the fast day or `None` if the date is not a fast day.

Return type
str or None

festival(*israel=False*, *hebrew=False*, *include_working_days=True*, *prefix_day=False*)

Return name of Jewish festival of date.

This method will return all major and minor religous Jewish holidays not including fast days.

Parameters

- **israel** (*bool, optional*) – `True` if you want the holidays according to the Israel schedule. Defaults to `False`.
- **hebrew** (*bool, optional*) – `True` if you want the festival name in Hebrew letters. Default is `False`, which returns the name transliterated into English.
- **include_working_days** (*bool, optional*) – `True` to include festival days on which melacha (work) is allowed; ie. Pesach Sheni, Chol Hamoed, etc. Default is `True`.
- **prefix_day** (*bool, optional*) – `True` to prefix multi day festivals with the day of the festival. Default is `False`.

Returns
The name of the festival or `None` if the given date is not a Jewish festival.

Return type
str or None

Examples

```
>>> pesach = HebrewDate(2023, 1, 15)
>>> pesach.festival(prefix_day=True)
'1 Pesach'
>>> pesach.festival(hebrew=True, prefix_day=True)
' '
>>> shavuos = HebrewDate(5783, 3, 6)
>>> shavuos.festival(israel=True, prefix_day=True)
'Shavuos'
```

holiday(*israel=False*, *hebrew=False*, *prefix_day=False*)

Return name of Jewish holiday of the date.

The holidays include the major and minor religious Jewish holidays including fast days.

Parameters

- **israel** (*bool, optional*) – `True` if you want the holidays according to the Israel schedule. Defaults to `False`.

- **hebrew** (*bool, optional*) – `True` if you want the holiday name in Hebrew letters. Default is `False`, which returns the name transliterated into English.
- **prefix_day** (*bool, optional*) – `True` to prefix multi day holidays with the day of the holiday. Default is `False`.

Returns
The name of the holiday or `None` if the given date is not a Jewish holiday.

Return type
str or None

Examples

```
>>> pesach = HebrewDate(2023, 1, 15)
>>> pesach.holiday(prefix_day=True)
'1 Pesach'
>>> pesach.holiday(hebrew=True, prefix_day=True)
' '
>>> taanis_esther = HebrewDate(5783, 12, 13)
>>> taanis_esther.holiday(prefix_day=True)
'Taanis Esther'
```

isoweekday()

Return the day of the week corresponding to the iso standard.

Returns
An integer representing the day of the week where Monday is 1 and and Sunday is 7.

Return type
int

replace(*year=None*, *month=None*, *day=None*)

Return new date with new values for the specified field.

Parameters

- **year** (*int, optional*) –
- **month** (*int, optional*) –
- **day** (*int, optional*) –

Returns

- *CalendarDateMixin* – Any date that inherits from CalendarDateMixin (`GregorianDate`, ``HebrewDate`).
- *Raises*
- *ValueError* – Raises a `ValueError` if the new date does not exist.

shabbos()

Return the Shabbos on or following the date.

Returns
self if the date is Shabbos or else the following Shabbos as the same date type as called from.

Return type
JulianDay, *GregorianDate*, or *HebrewDate*

Examples

```
>>> heb_date = HebrewDate(5781, 3, 29)
>>> greg_date = heb_date.to_greg()
>>> heb_date.shabbos()
HebrewDate(5781, 4, 2)
>>> greg_date.shabbos()
GregorianDate(2021, 6, 12)
```

tuple()

Return date as tuple.

Returns
A tuple of ints in the form `(year, month, day)`.

Return type
tuple of ints

weekday()

Return day of week as an integer.

Returns
An integer representing the day of the week with Sunday as 1 through Saturday as 7.

Return type
int

class pyluach.dates.**HebrewDate**(*year*, *month*, *day*, *jd=None*)

Bases: *BaseDate*, *CalendarDateMixin*

A class for manipulating Hebrew dates.

The following format options are available similar to strftime:

Format	Example	Meaning
%a	Sun	Weekday as locale's abbreviated name
%A	Sunday	Weekday as locale's full name
%w	1	Weekday as decimal number 1-7 Sunday-Shabbos
%d	07	Day of the month as a 0-padded 2 digit decimal number
%-d	7	Day of the month as a decimal number
%B	Iyar	Month name transliterated into English
%m	02	Month as a 0-padded 2 digit decimal number
%-m	2	Month as a decimal number
%y	82, 01	Year without century as a zero-padded decimal number
%Y	5782	Year as a decimal number
%*a		Weekday as a Hebrew numeral
%*A		Weekday name in Hebrew
%*d	,	Day of month as Hebrew numeral
%*-d	,	Day of month without gershayim
%*B		Name of month in Hebrew
%*y		Year in Hebrew numerals without the thousands place
%*Y	'	Year in Hebrew numerals with the thousands place
%%	%	A literal '%' character

Example

```
>>> date = HebrewDate(5783, 1, 15)
>>> f'Today is {date:%a - %*-d %*B, %*y}'
'Today is Thu -  , "'
```

Parameters

- **year** (*int*) – The Hebrew year.
- **month** (*int*) – The Hebrew month starting with Nissan as 1 (and Tishrei as 7). If there is a second Adar in the year it is has a value of 13.
- **day** (*int*) – The Hebrew day of the month.
- **jd** (*float, optional*) – This parameter should not be assigned manually.

year

Type
int

month

The Hebrew month starting with Nissan as 1 (and Tishrei as 7). If there is a second Adar it has a value of 13.

Type
int

day

The day of the month.

Type
int

Raises
ValueError – If the year is less than 1, if the month is less than 1 or greater than the last month, or if the day does not exist in the month a `ValueError` will be raised.

property jd

Return the corresponding Julian day number.

Returns
The Julian day number at midnight.

Return type
float

static from_pydate(*pydate*)

Return a *HebrewDate* from a python date object.

Parameters
pydate (*datetime.date*) – A python standard library `datetime.date` instance

Return type
HebrewDate

`static today()`

Return HebrewDate instance for the current day.

This static method wraps the Python standard library's `date.today()` method to get the date from the timestamp.

Returns
The current Hebrew date from the computer's timestamp.

Return type
HebrewDate

> **Warning:** Pyluach treats Hebrew dates as if they change at midnight. If it's after nightfall but before midnight, to get the true Hebrew date do `HebrewDate.today() + 1`.

`to_jd()`

Convert to a Julian day.

Returns
The equivalent JulianDay instance.

Return type
JulianDay

`to_greg()`

Convert to a Gregorian date.

Returns
The equivalent GregorianDate instance.

Return type
GregorianDate

`to_pydate()`

Convert to a standard library date.

Returns
The equivalent datetime.date instance.

Return type
datetime.date

`to_heb()`

Return Hebrew Date.

Return type
HebrewDate

`month_name`(*hebrew=False*)

Return the name of the month.

Parameters
hebrew (*bool, optional*) – `True` if the month name should be in Hebrew characters. Default is `False` which returns the month name transliterated into English.

Return type
str

hebrew_day(*withgershayim=True*)

Return the day of the month in Hebrew letters.

Parameters

withgershayim (*bool, optional*) – Default is `True` which includes a geresh with a single character and gershayim between two characters.

Returns

The day of the month in Hebrew letters.

Return type

str

Examples

```
>>> date = HebrewDate(5782, 3, 6)
>>> date.hebrew_day()
''
>>> date.hebrew_day(False)
''
>>> HebrewDate(5783, 12, 14).hebrew_day()
''
```

hebrew_year(*thousands=False*, *withgershayim=True*)

Return the year in Hebrew letters.

Parameters

- **thousands** (*bool*) – `True` to prefix the year with a letter for the thousands place, ie. ‘’. Default is `False`.
- **withgershayim** (*bool, optional*) – Default is `True` which includes a geresh after the thousands place if applicable and a gershayim before the last character of the year.

Return type

str

hebrew_date_string(*thousands=False*)

Return a Hebrew string representation of the date.

The date is in the form `f'{day} {month} {year}'`.

Parameters

thousands (*bool*) – `True` to have the thousands include in the year. Default is `False`.

Return type

str

Examples

```
>>> date = HebrewDate(5781, 9, 25)
>>> date.hebrew_date_string()
'  '
>>> date.hebrew_date_string(True)
'  '
```

add(*years=0*, *months=0*, *days=0*, *adar1=False*, *rounding=Rounding.NEXT_DAY*)

Add years, months, and days to date.

Parameters

- **years** (*int, optional*) – The number of years to add. Default is 0.
- **months** (*int, optional*) – The number of months to add. Default is 0.
- **days** (*int, optional*) – The number of days to add. Default is 0.
- **adar1** (*bool, optional*) – True to return a date in Adar Aleph if *self* is in a regular Adar and after adding the years it's leap year. Default is `False` which will return the date in Adar Beis.
- **rounding** (*Rounding, optional*) – Choose what to do if self is the 30th day of the month, and there are only 29 days in the destination month. *Rounding.NEXT_DAY* to return the first day of the next month. *Rounding.PREVIOUS_DAY* to return the last day of the month. *Rounding.EXCEPTION* to raise a ValueError. Default is *Rounding.NEXT_DAY*.

Return type
HebrewDate

Note: This method first adds the *years*. If the starting month is Adar and the destination year has two Adars, it chooses which one based on the *adar1* argument, then it adds the *months*. If the starting day doesn't exist in that month it adjusts it based on the *rounding* argument, then it adds the *days*.

Examples

```
>>> date = HebrewDate(5783, 11, 30)
>>> date.add(months=1)
HebrewDate(5783, 1, 1)
>>> date.add(months=1, rounding=Rounding.PREVIOUS_DAY)
HebrewDate(5783, 12, 29)
```

dict()

Return the date as a dictionary.

Returns
A dictionary in the form `{'year': int, 'month': int, 'day': int}`.

Return type
dict

fast_day(*hebrew=False*)

Return name of fast day of date.

Parameters
hebrew (*bool, optional*) – `True` if you want the fast day name in Hebrew letters. Default is `False`, which returns the name transliterated into English.

Returns
The name of the fast day or `None` if the date is not a fast day.

Return type
str or None

festival(*israel=False*, *hebrew=False*, *include_working_days=True*, *prefix_day=False*)

Return name of Jewish festival of date.

This method will return all major and minor religous Jewish holidays not including fast days.

Parameters

- **israel** (*bool, optional*) – `True` if you want the holidays according to the Israel schedule. Defaults to `False`.
- **hebrew** (*bool, optional*) – `True` if you want the festival name in Hebrew letters. Default is `False`, which returns the name transliterated into English.
- **include_working_days** (*bool, optional*) – `True` to include festival days on which melacha (work) is allowed; ie. Pesach Sheni, Chol Hamoed, etc. Default is `True`.
- **prefix_day** (*bool, optional*) – `True` to prefix multi day festivals with the day of the festival. Default is `False`.

Returns
The name of the festival or `None` if the given date is not a Jewish festival.

Return type
str or None

Examples

```
>>> pesach = HebrewDate(2023, 1, 15)
>>> pesach.festival(prefix_day=True)
'1 Pesach'
>>> pesach.festival(hebrew=True, prefix_day=True)
' '
>>> shavuos = HebrewDate(5783, 3, 6)
>>> shavuos.festival(israel=True, prefix_day=True)
'Shavuos'
```

holiday(*israel=False*, *hebrew=False*, *prefix_day=False*)

Return name of Jewish holiday of the date.

The holidays include the major and minor religious Jewish holidays including fast days.

Parameters

- **israel** (*bool, optional*) – `True` if you want the holidays according to the Israel schedule. Defaults to `False`.
- **hebrew** (*bool, optional*) – `True` if you want the holiday name in Hebrew letters. Default is `False`, which returns the name transliterated into English.

- **prefix_day** (*bool, optional*) – `True` to prefix multi day holidays with the day of the holiday. Default is `False`.

Returns
The name of the holiday or `None` if the given date is not a Jewish holiday.

Return type
str or None

Examples

```
>>> pesach = HebrewDate(2023, 1, 15)
>>> pesach.holiday(prefix_day=True)
'1 Pesach'
>>> pesach.holiday(hebrew=True, prefix_day=True)
' '
>>> taanis_esther = HebrewDate(5783, 12, 13)
>>> taanis_esther.holiday(prefix_day=True)
'Taanis Esther'
```

isoweekday()

Return the day of the week corresponding to the iso standard.

Returns
An integer representing the day of the week where Monday is 1 and and Sunday is 7.

Return type
int

replace(*year=None*, *month=None*, *day=None*)

Return new date with new values for the specified field.

Parameters

- **year** (*int, optional*) –
- **month** (*int, optional*) –
- **day** (*int, optional*) –

Returns

- *CalendarDateMixin* – Any date that inherits from CalendarDateMixin (`GregorianDate`, ``HebrewDate`).
- *Raises*
- *ValueError* – Raises a `ValueError` if the new date does not exist.

shabbos()

Return the Shabbos on or following the date.

Returns
self if the date is Shabbos or else the following Shabbos as the same date type as called from.

Return type
JulianDay, *GregorianDate*, or *HebrewDate*

Examples

```
>>> heb_date = HebrewDate(5781, 3, 29)
>>> greg_date = heb_date.to_greg()
>>> heb_date.shabbos()
HebrewDate(5781, 4, 2)
>>> greg_date.shabbos()
GregorianDate(2021, 6, 12)
```

subtract(*years=0*, *months=0*, *days=0*, *adar1=False*, *rounding=Rounding.NEXT_DAY*)

Subtract years, months, and days from date.

Parameters

- **years** (*int, optional*) – The number of years to subtract. Default is 0.
- **months** (*int, optional*) – The number of months to subtract. Default is 0.
- **days** (*int, optional*) – The number of days to subtract. Default is 0.
- **adar1** (*bool, optional*) – True to return a date in Adar Aleph if *self* is in a regular Adar and the destination year is leap year. Default is `False` which will return the date in Adar Beis.
- **rounding** (Rounding, *optional*) – Choose what to do if self is the 30th day of the month, and there are only 29 days in the destination month. *Rounding.NEXT_DAY* to return the first day of the next month. *Rounding.PREVIOUS_DAY* to return the last day of the month. *Rounding.EXCEPTION* to raise a ValueError. Default is *Rounding.NEXT_DAY*.

Return type
HebrewDate

Note: This method first subtracts the *years*. If the starting month is Adar and the destination year has two Adars, it chooses which one based on the *adar1* argument, then it subtracts the *months*. If the starting day doesn't exist in that month it adjusts it based on the *rounding* argument, then it subtracts the *days*.

---

tuple()

Return date as tuple.

Returns
A tuple of ints in the form `(year, month, day)`.

Return type
tuple of ints

weekday()

Return day of week as an integer.

Returns
An integer representing the day of the week with Sunday as 1 through Saturday as 7.

Return type
int

1.6.2 hebrewcal module

The hebrewcal module contains Hebrew calendar related classes and functions.

It contains classes for representing a Hebrew year and month, functions for getting the holiday or fast day for a given date, and classes adapting `calendar` classes to render Hebrew calendars.

Contents

exception pyluach.hebrewcal.**IllegalMonthError**(*month*)

Bases: `ValueError`

An exception for an illegal month.

Subclasses `ValueError` to show a message for an invalid month number for the Hebrew calendar. Mimics `calendar.IllegalMonthError`.

Parameters
month (*int*) – The invalid month number

add_note()

Exception.add_note(note) – add a note to the exception

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception pyluach.hebrewcal.**IllegalWeekdayError**(*weekday*)

Bases: `ValueError`

An exception for an illegal weekday.

Subclasses `ValueError` to show a message for an invalid weekday number. Mimics `calendar.IllegalWeekdayError`.

Parameters
month (*int*) – The invalid month number

add_note()

Exception.add_note(note) – add a note to the exception

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

class pyluach.hebrewcal.**Year**(*year*)

Bases: object

A Year object represents a Hebrew calendar year.

It provided the following operators:

Operation	Result
year2 = year1 + int	New `Year` `int` days after year1.
year2 = year1 - int	New `Year` `int` days before year1.
int = year1 - year2	`int` equal to the absolute value of the difference between year2 and year1.
bool = year1 == year2	True if year1 represents the same year as year2.
bool = year1 > year2	True if year1 is later than year2.
bool = year1 >= year2	True if year1 is later or equal to year2.
bool = year1 < year2	True if year 1 earlier than year2.
bool = year1 <= year2	True if year 1 earlier or equal to year 2.

Parameters
year (*int*) – A Hebrew year.

year

The hebrew year.

Type
int

leap

True if the year is a leap year else false.

Type
bool

monthscount()

Return number of months in the year.

Return type
int

itermonths()

Yield Month instance for each month of the year.

Yields
Month – The next month in the Hebrew calendar year as a `Month` instance beginning with Tishrei through Elul.

iterdays()

Yield integer for each day of the year.

Yields
int – An integer beginning with 1 for the the next day of the year.

iterdates()

Iterate through each Hebrew date of the year.

Yields
pyluach.dates.HebrewDate – The next date of the Hebrew calendar year starting with the first of Tishrei.

classmethod from_date(*date*)

Return Year object that given date occurs in.

Parameters
date (BaseDate) – Any subclass of BaseDate.

Return type
Year

classmethod from_pydate(*pydate*)

Return Year object from python date object.

Parameters
pydate (*datetime.date*) – A python standard library date object

Returns
The Hebrew year the given date occurs in

Return type
Year

year_string(*thousands=False*)

Return year as a Hebrew string.

Parameters
thousands (*bool, optional*) – True to prefix the year with the thousands place. Default is False.

Examples

```
>>> year = Year(5781)
>>> year.year_string()

>>> year.year_string(True)
```

class pyluach.hebrewcal.**Month**(*year*, *month*)

Bases: object

A Month object represents a month of the Hebrew calendar.

It provides the same operators as a *Year* object.

Parameters

- **year** (*int*) –
- **month** (*int*) – The month as an integer starting with 7 for Tishrei through 13 if necessary for Adar Sheni and then 1-6 for Nissan - Elul.

year

The Hebrew year.

Type
int

month

The month as an integer starting with 7 for Tishrei through 13 if necessary for Adar Sheni and then 1-6 for Nissan - Elul.

Type
int

`classmethod from_date(date)`

Return Month object that given date occurs in.

Parameters
date (BaseDate) – Any subclass of BaseDate.

Returns
The Hebrew month the given date occurs in

Return type
Month

`classmethod from_pydate(pydate)`

Return Month object from python date object.

Parameters
pydate (*datetime.date*) – A python standard library date object

Returns
The Hebrew month the given date occurs in

Return type
Month

`month_name(hebrew=False)`

Return the name of the month.

Replaces *name* attribute.

Parameters
hebrew (*bool, optional*) – *True* if the month name should be written with Hebrew letters and False to be transliterated into English using the Ashkenazic pronunciation. Default is *False*.

Return type
str

`month_string(thousands=False)`

Return month and year in Hebrew.

Parameters
thousands (*bool, optional*) – True to prefix year with thousands place. Default is False.

Returns
The month and year in Hebrew in the form `f'{month} {year}'`.

Return type
str

`starting_weekday()`

Return first weekday of the month.

Returns
The weekday of the first day of the month starting with Sunday as 1 through Saturday as 7.

Return type
int

`iterdates()`

Iterate through the Hebrew dates of the month.

Yields
pyluach.dates.HebrewDate – The next Hebrew date of the month.

`molad()`

Return the month's molad.

Returns
A dictionary in the form `{weekday: int, hours: int, parts: int}`

Return type
dict

Note: This method does not return the molad in the form that is traditionally announced in the shul. This is the molad in the form used to calculate the length of the year.

See also:

molad_announcement
The molad as it is traditionally announced.

`molad_announcement()`

Return the month's molad in the announcement form.

Returns a dictionary in the form that the molad is traditionally announced. The weekday is adjusted to change at midnight and the hour of the day and minutes are given as traditionally announced. Note that the hour is given as in a twenty four hour clock ie. 0 for 12:00 AM through 23 for 11:00 PM.

Returns

A dictionary in the form:

```
{
    weekday: int,
    hour: int,
    minutes: int,
    parts: int
}
```

Return type
dict

`pyluach.hebrewcal.to_hebrew_numeral(num, thousands=False, withgershayim=True)`

Convert int to Hebrew numeral.

Function useful in formatting Hebrew calendars.

Parameters

- **num** (*int*) – The number to convert
- **thousands** (*bool, optional*) – True if the hebrew returned should include a letter for the thousands place ie. '' for five thousand. Default is `False`.
- **withgershayim** (*bool, optional*) – `True` to include a geresh after a single letter and double geresh before the last letter if there is more than one letter. Default is `True`.

Returns
The Hebrew numeral representation of the number.

Return type
str

class pyluach.hebrewcal.**HebrewCalendar**(*firstweekday=1*, *hebrewnumerals=True*, *hebrewweekdays=False*, *hebrewmonths=False*, *hebrewyear=False*)

Bases: Calendar

Calendar base class.

This class extends the python library Calendar class for the Hebrew calendar. The weekdays are 1 for Sunday through 7 for Shabbos.

Parameters

- **firstweekday** (*int, optional*) – The weekday to start each week with. Default is 1 for Sunday.
- **hebrewnumerals** (*bool, optional*) – Default is True, which shows the days of the month with Hebrew numerals. False shows the days of the month as a decimal number.
- **hebrewweekdays** (*bool, optional*) – True to show the weekday in Hebrew. Default is False, which shows the weekday in English.
- **hebrewmonths** (*bool, optional*) – True to show the month name in Hebrew. Default is False, which shows the month name transliterated into English.
- **hebrewyear** (*bool, optional*) – True to show the year in Hebrew numerals. Default is False, which shows the year as a decimal number.

hebrewnumerals

Type
bool

hebrewweekdays

Type
bool

hebrewmonths

Type
bool

hebrewyear

Type
bool

Note: All of the parameters other than *firstweekday* are not used in the HebrewCalendar base class. They're there for use in child classes.

---

property firstweekday

Get and set the weekday the weeks should start with.

Return type
int

`iterweekdays()`

Return one week of weekday numbers.

The numbers start with the configured first one.

Yields
`int` – The next weekday with 1-7 for Sunday - Shabbos. The iterator starts with the `HebrewCalendar` object's configured first weekday ie. if configured to start with Monday it will first yield *2* and end with *1*.

`itermonthdates(year, month)`

Yield dates for one month.

The iterator will always iterate through complete weeks, so it will yield dates outside the specified month.

Parameters
- **year** (*int*) –
- **month** (*int*) – The Hebrew month starting with 1 for Nissan through 13 for Adar Sheni if necessary.

Yields
pyluach.dates.HebrewDate – The next Hebrew Date of the month starting with the first date of the week the first of the month falls in, and ending with the last date of the week that the last day of the month falls in.

`itermonthdays(year, month)`

Like `itermonthdates()` but will yield day numbers. For days outside the specified month the day number is 0.

Parameters
- **year** (*int*) –
- **month** (*int*) –

Yields
`int` – The day of the month or 0 if the date is before or after the month.

`itermonthdays2(year, month)`

Return iterator for the days and weekdays of the month.

Parameters
- **year** (*int*) –
- **month** (*int*) –

Yields
`tuple` of `int` – A tuple of ints in the form `(day of month, weekday)`.

`itermonthdays3(year, month)`

Return iterator for the year, month, and day of the month.

Parameters
- **year** (*int*) –
- **month** (*int*) –

Yields
`tuple` of `int` – A tuple of ints in the form `(year, month, day)`.

itermonthdays4(*year*, *month*)

Return iterator for the year, month, day, and weekday

Parameters

- **year** (*int*) –
- **month** (*int*) –

Yields

tuple of int – A tuple of ints in the form `(year, month, day, weekday)`.

yeardatescalendar(*year*, *width=3*)

Return data of specified year ready for formatting.

Parameters

- **year** (*int*) –
- **width** (*int, optional*) – The number of months per row. Default is 3.

Returns

Returns a list of month rows. Each month row contains a list of up to *width* months. Each month contains either 5 or 6 weeks, and each week contains 7 days. Days are `HebrewDate` objects.

Return type

list of list of list of list of *pyluach.dates.HebrewDate*

yeardays2calendar(*year*, *width=3*)

Return the data of the specified year ready for formatting.

This method is similar to the `yeardatescalendar` except the entries in the week lists are `(day number, weekday number)` tuples.

Parameters

- **year** (*int*) –
- **width** (*int, optional*) – The number of months per row. Default is 3.

Returns

Returns a list of month rows. Each month row contains a list of up to *width* months. Each month contains between 4 and 6 weeks, and each week contains 1-7 days. Days are tuples with the form `(day number, weekday number)`.

Return type

list of list of list of list of tuple

yeardayscalendar(*year*, *width=3*)

Return the data of the specified year ready for formatting.

This method is similar to the `yeardatescalendar` except the entries in the week lists are day numbers.

Parameters

- **year** (*int*) –
- **width** (*int, optional*) – The number of months per row. Default is 3.

Returns

Returns a list of month rows. Each month row contains a list of up to *width* months. Each month contains either 5 or 6 weeks, and each week contains 1-7 days. Each day is the day of the month as an int.

Return type
list of list of list of list of int

monthdatescalendar(*year*, *month*)

Return matrix (list of lists) of dates for month's calendar.

Each row represents a week; week entries are HebrewDate instances.

Parameters

- **year** (*int*) –
- **month** (*int*) –

Returns
List of weeks in the month containing 7 HebrewDate instances each.

Return type
list of list of *pyluach.dates.HebrewDate*

monthdays2calendar(*year*, *month*)

Return a matrix representing a month's calendar. Each row represents a week; week entries are (day number, weekday number) tuples. Day numbers outside this month are zero.

monthdayscalendar(*year*, *month*)

Return a matrix representing a month's calendar. Each row represents a week; days outside this month are zero.

class pyluach.hebrewcal.**HebrewHTMLCalendar**(*firstweekday=1*, *hebrewnumerals=True*, *hebrewweekdays=False*, *hebrewmonths=False*, *hebrewyear=False*, *rtl=False*)

Bases: *HebrewCalendar*, HTMLCalendar

Class to generate html calendars .

Adapts calendar.HTMLCalendar for the Hebrew calendar.

Parameters

- **firstweekday** (*int, optional*) – The weekday to start each week with. Default is 1 for Sunday.
- **hebrewnumerals** (*bool, optional*) – Default is True, which shows the days of the month with Hebrew numerals. False shows the days of the month as a decimal number.
- **hebrewweekdays** (*bool, optional*) – True to show the weekday in Hebrew. Default is False, which shows the weekday in English.
- **hebrewmonths** (*bool, optional*) – True to show the month name in Hebrew. Default is False, which shows the month name transliterated into English.
- **hebrewyear** (*bool, optional*) – True to show the year in Hebrew numerals. Default is False, which shows the year as a decimal number.
- **rtl** (*bool, optional*) – True to arrange the months and the days of the month from right to left. Default is False.

hebrewnumerals

Type
bool

hebrewweekdays

Type
bool

hebrewmonths

Type
bool

hebrewyear

Type
bool

rtl

Type
bool

formatday(*day*, *weekday*)

Return a day as an html table cell.

Parameters

- **day** (*int*) – The day of the month or zero for a day outside the month.
- **weekday** (*int*) – The weekday with 1 as Sunday through 7 as Shabbos.

Return type
str

formatweekday(*day*)

Return a weekday name as an html table header.

Parameters
day (*int*) – The day of the week 1-7 with Sunday as 1 and Shabbos as 7.

Return type
str

formatyearnumber(*theyear*)

Return a formatted year.

Parameters
theyear (*int*) –

Returns
If `self.hebrewyear` is `True` return the year as a Hebrew numeral, else return *theyear* as is.

Return type
int or str

formatmonthname(*theyear*, *themonth*, *withyear=True*)

Return month name as an html table row.

Parameters

- **theyear** (*int*) –
- **themonth** (*int*) – The month as an int 1-12 Nissan - Adar and 13 if leap year.
- **withyear** (*bool, optional*) – `True` to append the year to the month name. Default is `True`.

Return type
str

formatmonth(*theyear*, *themonth*, *withyear=True*)

Return a formatted month as an html table.

Parameters

- **theyear** (*int*) –
- **themonth** (*int*) –
- **withyear** (*bool, optional*) – `True` to have the year appended to the month name. Default is `True`.

Return type
str

formatyear(*theyear*, *width=3*)

Return a formatted year as an html table.

Parameters

- **theyear** (*int*) –
- **width** (*int, optional*) – The number of months to display per row. Default is 3.

Return type
str

property firstweekday

Get and set the weekday the weeks should start with.

Return type
int

formatweek(*theweek*)

Return a complete week as a table row.

formatweekheader()

Return a header for a week as a table row.

formatyearpage(*theyear*, *width=3*, *css='calendar.css'*, *encoding=None*)

Return a formatted year as a complete HTML page.

itermonthdates(*year*, *month*)

Yield dates for one month.

The iterator will always iterate through complete weeks, so it will yield dates outside the specified month.

Parameters

- **year** (*int*) –
- **month** (*int*) – The Hebrew month starting with 1 for Nissan through 13 for Adar Sheni if necessary.

Yields
pyluach.dates.HebrewDate – The next Hebrew Date of the month starting with the first date of the week the first of the month falls in, and ending with the last date of the week that the last day of the month falls in.

itermonthdays(*year*, *month*)

Like `itermonthdates()` but will yield day numbers. For days outside the specified month the day number is 0.

Parameters

- **year** (*int*) –
- **month** (*int*) –

Yields

`int` – The day of the month or 0 if the date is before or after the month.

itermonthdays2(*year*, *month*)

Return iterator for the days and weekdays of the month.

Parameters

- **year** (*int*) –
- **month** (*int*) –

Yields

`tuple` of `int` – A tuple of ints in the form `(day of month, weekday)`.

itermonthdays3(*year*, *month*)

Return iterator for the year, month, and day of the month.

Parameters

- **year** (*int*) –
- **month** (*int*) –

Yields

`tuple` of `int` – A tuple of ints in the form `(year, month, day)`.

itermonthdays4(*year*, *month*)

Return iterator for the year, month, day, and weekday

Parameters

- **year** (*int*) –
- **month** (*int*) –

Yields

`tuple` of `int` – A tuple of ints in the form `(year, month, day, weekday)`.

iterweekdays()

Return one week of weekday numbers.

The numbers start with the configured first one.

Yields

`int` – The next weekday with 1-7 for Sunday - Shabbos. The iterator starts with the `HebrewCalendar` object's configured first weekday ie. if configured to start with Monday it will first yield *2* and end with *1*.

monthdatescalendar(*year*, *month*)

Return matrix (list of lists) of dates for month's calendar.

Each row represents a week; week entries are HebrewDate instances.

Parameters

- **year** (*int*) –
- **month** (*int*) –

Returns
List of weeks in the month containing 7 `HebrewDate` instances each.

Return type
`list` of list of *pyluach.dates.HebrewDate*

monthdays2calendar(*year*, *month*)

Return a matrix representing a month's calendar. Each row represents a week; week entries are (day number, weekday number) tuples. Day numbers outside this month are zero.

monthdayscalendar(*year*, *month*)

Return a matrix representing a month's calendar. Each row represents a week; days outside this month are zero.

yeardatescalendar(*year*, *width=3*)

Return data of specified year ready for formatting.

Parameters

- **year** (*int*) –
- **width** (*int, optional*) – The number of months per row. Default is 3.

Returns
Returns a list of month rows. Each month row contains a list of up to *width* months. Each month contains either 5 or 6 weeks, and each week contains 7 days. Days are `HebrewDate` objects.

Return type
`list` of list of list of list of *pyluach.dates.HebrewDate*

yeardays2calendar(*year*, *width=3*)

Return the data of the specified year ready for formatting.

This method is similar to the `yeardatescalendar` except the entries in the week lists are `(day number, weekday number)` tuples.

Parameters

- **year** (*int*) –
- **width** (*int, optional*) – The number of months per row. Default is 3.

Returns
Returns a list of month rows. Each month row contains a list of up to *width* months. Each month contains between 4 and 6 weeks, and each week contains 1-7 days. Days are tuples with the form `(day number, weekday number)`.

Return type
`list` of list of list of list of `tuple`

yeardayscalendar(*year*, *width=3*)

Return the data of the specified year ready for formatting.

This method is similar to the `yeardatescalendar` except the entries in the week lists are day numbers.

Parameters

- **year** (*int*) –

- **width** (*int, optional*) – The number of months per row. Default is 3.

Returns
Returns a list of month rows. Each month row contains a list of up to *width* months. Each month contains either 5 or 6 weeks, and each week contains 1-7 days. Each day is the day of the month as an int.

Return type
list of list of list of list of int

class pyluach.hebrewcal.**HebrewTextCalendar**(*firstweekday=1*, *hebrewnumerals=True*, *hebrewweekdays=False*, *hebrewmonths=False*, *hebrewyear=False*)

Bases: *HebrewCalendar*, TextCalendar

Subclass of HebrewCalendar that outputs a plaintext calendar.

HebrewTextCalendar adapts calendar.TextCalendar for the Hebrew calendar.

Parameters

- **firstweekday** (*int, optional*) – The weekday to start each week with. Default is 1 for Sunday.
- **hebrewnumerals** (*bool, optional*) – Default is True, which shows the days of the month with Hebrew numerals. False shows the days of the month as a decimal number.
- **hebrewweekdays** (*bool, optional*) – True to show the weekday in Hebrew. Default is False, which shows the weekday in English.
- **hebrewmonths** (*bool, optional*) – True to show the month name in Hebrew. Default is False, which shows the month name transliterated into English.
- **hebrewyear** (*bool, optional*) – True to show the year in Hebrew numerals. Default is False, which shows the year as a decimal number.

hebrewnumerals

Type
bool

hebrewweekdays

Type
bool

hebrewmonths

Type
bool

hebrewyear

Type
bool

Note: This class generates plain text calendars. Any program that adds any formatting may misrender the calendars especially when using any Hebrew characters.

formatday(*day*, *weekday*, *width*)

Return a formatted day.

Extends calendar.TextCalendar formatday method.

Parameters

- **day** (*int*) – The day of the month.
- **weekday** (*int*) – The weekday 1-7 Sunday-Shabbos.
- **width** (*int*) – The width of the day column.

Return type

str

formatweekday(*day*, *width*)

Return formatted weekday.

Extends calendar.TextCalendar formatweekday method.

Parameters

- **day** (*int*) – The weekday 1-7 Sunday-Shabbos.
- **width** (*int*) – The width of the day column.

Return type

str

formatmonthname(*theyear*, *themonth*, *width=0*, *withyear=True*)

Return formatted month name.

Parameters

- **theyear** (*int*) –
- **themonth** (*int*) – 1-12 or 13 for Nissan-Adar Sheni
- **width** (*int, optional*) – The number of columns per day. Default is 0
- **withyear** (*bool, optional*) – Default is `True` to include the year with the month name.

Return type

str

formatyear(*theyear*, *w=2*, *l=1*, *c=6*, *m=3*)

Return a year's calendar as a multi-line string.

Parameters

- **theyear** (*int*) –
- **w** (*int, optional*) – The date column width. Default is 2
- **l** (*int, optional*) – The number of lines per week. Default is 1.
- **c** (*int, optional*) – The number of columns in between each month. Default is 6
- **m** (*int, optional*) – The number of months per row. Default is 3.

Return type

str

property firstweekday

Get and set the weekday the weeks should start with.

Return type
int

formatmonth(*theyear*, *themonth*, *w=0*, *l=0*)

Return a month's calendar string (multi-line).

formatweek(*theweek*, *width*)

Returns a single week in a string (no newline).

formatweekheader(*width*)

Return a header for a week.

itermonthdates(*year*, *month*)

Yield dates for one month.

The iterator will always iterate through complete weeks, so it will yield dates outside the specified month.

Parameters

- **year** (*int*) –
- **month** (*int*) – The Hebrew month starting with 1 for Nissan through 13 for Adar Sheni if necessary.

Yields
pyluach.dates.HebrewDate – The next Hebrew Date of the month starting with the first date of the week the first of the month falls in, and ending with the last date of the week that the last day of the month falls in.

itermonthdays(*year*, *month*)

Like `itermonthdates()` but will yield day numbers. For days outside the specified month the day number is 0.

Parameters

- **year** (*int*) –
- **month** (*int*) –

Yields
int – The day of the month or 0 if the date is before or after the month.

itermonthdays2(*year*, *month*)

Return iterator for the days and weekdays of the month.

Parameters

- **year** (*int*) –
- **month** (*int*) –

Yields
tuple of int – A tuple of ints in the form `(day of month, weekday)`.

itermonthdays3(*year*, *month*)

Return iterator for the year, month, and day of the month.

Parameters

- **year** (*int*) –

- **month** (*int*) –

Yields
`tuple` of `int` – A tuple of ints in the form `(year, month, day)`.

itermonthdays4(*year*, *month*)

Return iterator for the year, month, day, and weekday

Parameters

- **year** (*int*) –
- **month** (*int*) –

Yields
`tuple` of `int` – A tuple of ints in the form `(year, month, day, weekday)`.

iterweekdays()

Return one week of weekday numbers.

The numbers start with the configured first one.

Yields
`int` – The next weekday with 1-7 for Sunday - Shabbos. The iterator starts with the `HebrewCalendar` object's configured first weekday ie. if configured to start with Monday it will first yield *2* and end with *1*.

monthdatescalendar(*year*, *month*)

Return matrix (list of lists) of dates for month's calendar.

Each row represents a week; week entries are HebrewDate instances.

Parameters

- **year** (*int*) –
- **month** (*int*) –

Returns
List of weeks in the month containing 7 `HebrewDate` instances each.

Return type
`list` of list of *pyluach.dates.HebrewDate*

monthdays2calendar(*year*, *month*)

Return a matrix representing a month's calendar. Each row represents a week; week entries are (day number, weekday number) tuples. Day numbers outside this month are zero.

monthdayscalendar(*year*, *month*)

Return a matrix representing a month's calendar. Each row represents a week; days outside this month are zero.

prmonth(*theyear*, *themonth*, *w=0*, *l=0*)

Print a month's calendar.

prweek(*theweek*, *width*)

Print a single week (no newline).

pryear(*theyear*, *w=0*, *l=0*, *c=6*, *m=3*)

Print a year's calendar.

yeardatescalendar(*year*, *width=3*)

Return data of specified year ready for formatting.

Parameters

- **year** (*int*) –
- **width** (*int, optional*) – The number of months per row. Default is 3.

Returns

Returns a list of month rows. Each month row contains a list of up to *width* months. Each month contains either 5 or 6 weeks, and each week contains 7 days. Days are `HebrewDate` objects.

Return type

`list` of list of list of list of *pyluach.dates.HebrewDate*

yeardays2calendar(*year*, *width=3*)

Return the data of the specified year ready for formatting.

This method is similar to the `yeardatescalendar` except the entries in the week lists are `(day number, weekday number)` tuples.

Parameters

- **year** (*int*) –
- **width** (*int, optional*) – The number of months per row. Default is 3.

Returns

Returns a list of month rows. Each month row contains a list of up to *width* months. Each month contains between 4 and 6 weeks, and each week contains 1-7 days. Days are tuples with the form `(day number, weekday number)`.

Return type

`list` of list of list of list of `tuple`

yeardayscalendar(*year*, *width=3*)

Return the data of the specified year ready for formatting.

This method is similar to the `yeardatescalendar` except the entries in the week lists are day numbers.

Parameters

- **year** (*int*) –
- **width** (*int, optional*) – The number of months per row. Default is 3.

Returns

Returns a list of month rows. Each month row contains a list of up to *width* months. Each month contains either 5 or 6 weeks, and each week contains 1-7 days. Each day is the day of the month as an int.

Return type

`list` of list of list of list of `int`

pyluach.hebrewcal.**fast_day**(*date*, *hebrew=False*)

Return name of fast day or None.

Parameters

- **date** (`BaseDate`) – Any date instance from a subclass of `BaseDate` can be used.
- **hebrew** (*bool, optional*) – `True` if you want the fast_day name in Hebrew letters. Default is `False`, which returns the name transliterated into English.

Returns
The name of the fast day or `None` if the given date is not a fast day.

Return type
str or None

`pyluach.hebrewcal.`**`festival`**(*date*, *israel=False*, *hebrew=False*, *include_working_days=True*, *prefix_day=False*)

Return Jewish festival of given day.

This method will return all major and minor religous Jewish holidays not including fast days.

Parameters

- **`date`** (`BaseDate`) – Any subclass of `BaseDate` can be used.
- **`israel`** (*`bool, optional`*) – `True` if you want the festivals according to the Israel schedule. Defaults to `False`.
- **`hebrew`** (*`bool, optional`*) – `True` if you want the festival name in Hebrew letters. Default is `False`, which returns the name transliterated into English.
- **`include_working_days`** (*`bool, optional`*) – `True` to include festival days on which melacha (work) is allowed; ie. Pesach Sheni, Chol Hamoed, etc. Default is `True`.
- **`prefix_day`** (*`bool, optional`*) – `True` to prefix multi day festivals with the day of the festival. Default is `False`.

Returns
The name of the festival or `None` if the given date is not a Jewish festival.

Return type
str or None

Examples

```
>>> from pyluach.dates import HebrewDate
pesach = HebrewDate(2023, 1, 15)
>>> festival(pesach, prefix_day=True)
'1 Pesach'
>>> festival(pesach, hebrew=True, prefix_day=True)
' '
>>> shavuos = HebrewDate(5783, 3, 6)
>>> festival(shavuos, israel=True, prefix_day=True)
'Shavuos'
```

`pyluach.hebrewcal.`**`holiday`**(*date*, *israel=False*, *hebrew=False*, *prefix_day=False*)

Return Jewish holiday of given date.

The holidays include the major and minor religious Jewish holidays including fast days.

Parameters

- **`date`** (`pyluach.dates.BaseDate`) – Any subclass of `BaseDate` can be used.
- **`israel`** (*`bool, optional`*) – `True` if you want the holidays according to the israel schedule. Default is `False`.
- **`hebrew`** (*`bool, optional`*) – `True` if you want the holiday name in Hebrew letters. Default is `False`, which returns the name transliterated into English.

- **prefix_day** (*bool, optional*) – `True` to prefix multi day holidays with the day of the holiday. Default is `False`.

Returns
The name of the holiday or `None` if the given date is not a Jewish holiday.

Return type
str or None

Examples

```
>>> from pyluach.dates import HebrewDate
>>> pesach = HebrewDate(2023, 1, 15)
>>> holiday(pesach, prefix_day=True)
'1 Pesach'
>>> holiday(pesach, hebrew=True, prefix_day=True)
' '
>>> taanis_esther = HebrewDate(5783, 12, 13)
>>> holiday(taanis_esther, prefix_day=True)
'Taanis Esther'
```

1.6.3 parshios module

The parshios module has functions to find the weekly parasha.

Examples

```
>>> from pyluach import dates, parshios
>>> date = dates.HebrewDate(5781, 10, 5)
>>> parshios.getparsha(date)
'Vayigash'
>>> parshios.getparsha_string(date, hebrew=True)
''
>>> parshios.getparsha_string(dates.GregorianDate(2021, 3, 7), hebrew=True)
', '
```

Note: The algorithm is based on Dr. Irv Bromberg's, University of Toronto at http://individual.utoronto.ca/kalendis/hebrew/parshah.htm

All English parsha names are transliterated into the American Ashkenazik pronunciation.

`pyluach.parshios.`**PARSHIOS**

A list of the parshios transliterated into English.

Type
list of str

`pyluach.parshios.`**PARSHIOS_HEBREW**

A list of the parshios in Hebrew.

Type
list of str

pyluach.parshios.**getparsha**(*date*, *israel=False*)

Return the parsha for a given date.

Returns the parsha for the Shabbos on or following the given date.

Parameters

- **date** (BaseDate) – Any subclass of BaseDate. This date does not have to be a Shabbos.
- **israel** (*bool, optional*) – True if you want the parsha according to the Israel schedule (with only one day of Yom Tov). Defaults to False.

Returns

A list of the numbers of the parshios for the Shabbos of the given date, beginning with 0 for Beraishis, or None if the Shabbos doesn't have a parsha (i.e. it's on Yom Tov).

Return type

list of int or None

pyluach.parshios.**getparsha_string**(*date*, *israel=False*, *hebrew=False*)

Return the parsha as a string for the given date.

This function wraps getparsha returning the parsha name.

Parameters

- **date** (BaseDate) – Any subclass of BaseDate. The date does not have to be a Shabbos.
- **israel** (*bool, optional*) – True if you want the parsha according to the Israel schedule (with only one day of Yom Tov). Default is False.
- **hebrew** (*bool, optional*) – True if you want the name of the parsha in Hebrew. Default is False.

Returns

The name of the parsha separated by a comma and space if it is a double parsha or None if there is no parsha that Shabbos (ie. it's yom tov).

Return type

str or None

pyluach.parshios.**iterparshios**(*year*, *israel=False*)

Generate all the parshios in the year.

Parameters

- **year** (*int*) – The Hebrew year to get the parshios for.
- **israel** (*bool, optional*) – True if you want the parsha according to the Israel schedule (with only one day of Yom Tov). Defaults to False

Yields

list of int or None – A list of the numbers of the parshios for the next Shabbos in the given year. Yields None for a Shabbos that doesn't have its own parsha (i.e. it occurs on a yom tov).

pyluach.parshios.**parshatable**(*year*, *israel=False*)

Return a table of all the Shabbosos in the year

Parameters

- **year** (*int*) – The Hebrew year to get the parshios for.
- **israel** (*bool, optional*) – True if you want the parshios according to the Israel schedule (with only one day of Yom Tov). Defaults to False.

Returns
An ordered dictionary with the `HebrewDate` of each Shabbos as the key mapped to the parsha as a list of ints, or `None` for a Shabbos with no parsha.

Return type
OrderedDict

1.6.4 Changelog

This document records all notable changes to pyluach. This project adheres to Semantic Versioning.

2.2.0 (2023-02-28)

- Added *prefix_day* param to `festival` and `holiday` methods and functions.

2.1.0 (2023-02-12)

- Added `add` and `subtract` methods to `dates.HebrewDate`.
- Added `replace` method to `CalendarDateMixin`.
- Added missing documentation for *%y* and *%Y* in formatting `HebrewDate`.

2.0.2 (2022-10-24)

- Fix subtracting one date from another returning `float` instead of `int`.

2.0.1 (2022-08-24)

- Fix issue (#24) where Shavuos is returned in most months on day 7.

2.0.0 (2022-05-29)

- Changed equality comparers to compare object identity on unmatched types.
- Equal dates of different types will no longer be considered identical keys for dicts.
- Added `strftime` and `__format__` methods to `dates.GregorianDate`.
- Added `__format__` method to `dates.HebrewDate`.
- Added *withgershayim* parameter to `dates.HebrewDate.hebrew_day` and `dates.HebrewDate.hebrew_year` methods
- Added `monthcount` method to `hebrewcal.Year`.
- Removed deprecated `hebrewcal.Month.name` attribute.
- Implemented HebrewCalendar classes for generating calendars similar to Calendar classes in the standard library calendar module.

1.4.2 (2022-05-20)

- Fixed bug in `hebrewcal.Month` comparisons when one month is before Nissan and one is not.

1.4.1 (2022-03-25)

- Fixed mistakes in docstring and error message.

1.4.0 (2022-02-21)

- Added parameter *include_working_days* to `festival` method and function.
- Removed support for python < 3.6

1.3.0 (2021-06-09)

- Added option to get parsha in Hebrew.
- Added `dates.HebrewDate` methods to get hebrew day, month, year, and date string in Hebrew.
- Added method to get `hebrewcal.Month` names in Hebrew.
- Added methods to get year and month strings in Hebrew.
- Added classmethods to `hebrewcal.Year` and `hebrewcal.Month` to get objects from dates and pydates.
- Added methods to dates classes to get holidays, fast days and festivals.
- Implemented more consistent Hebrew to English transliterations for parshios.

1.2.1 (2020-11-08)

- Fixed molad having weekday of `0` when it should be `7`.

1.2.0 (2020-10-28)

- Created `isoweekday` method for all date types in the `dates` module.
- Created fast_day and festival functions (#11)
- Added Pesach Sheni to festival.

1.1.1 (2020-08-14)

- Fixed error getting parsha of Shabbos on Rosh Hashana.

1.1.0 (2020-06-03)

- Redesigned documentation.
- Added `molad` and `molad_announcement` methods to `hebrewcal.Month`.
- Stopped supporting python < 3.4 and modernized code.

1.0.1 (2019-03-02)

- Initial public release

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

p

INDEX

G

H

I

J

L

M

N

P

R

S

T

W

Y