02_calendar_module

November 14, 2019

[1]:
```
import calendar
```

[2]:
```
print(dir(calendar))
```

```
['Calendar', 'EPOCH', 'FRIDAY', 'February', 'HTMLCalendar', 'IllegalMonthError',
'IllegalWeekdayError', 'January', 'LocaleHTMLCalendar', 'LocaleTextCalendar',
'MONDAY', 'SATURDAY', 'SUNDAY', 'THURSDAY', 'TUESDAY', 'TextCalendar',
'WEDNESDAY', '_EPOCH_ORD', '__all__', '__builtins__', '__cached__', '__doc__',
'__file__', '__loader__', '__name__', '__package__', '__spec__', '_colwidth',
'_locale', '_localized_day', '_localized_month', '_spacing', 'c', 'calendar',
'datetime', 'day_abbr', 'day_name', 'different_locale', 'error', 'firstweekday',
'format', 'formatstring', 'isleap', 'leapdays', 'main', 'mdays', 'month',
'month_abbr', 'month_name', 'monthcalendar', 'monthlen', 'monthrange',
'nextmonth', 'prcal', 'prevmonth', 'prmonth', 'prweek', 'repeat',
'setfirstweekday', 'sys', 'timegm', 'week', 'weekday', 'weekheader']
```

Problem: For a given date, tell the week of the day

[3]:
```
def get_week_of_day():
    import calendar

    given_input = '08 23 2019'  # "MM DD YYYY" # input()
    month, day, year = map(int, given_input.split(' '))

    weeks = ('MONDAY', 'TUESDAY', 'WEDNESDAY', 'THURSDAY',
             'FRIDAY', 'SATURDAY', 'SUNDAY')
    print(calendar.weekday(year, month, day))

    print(weeks[calendar.weekday(year, month, day)])

get_week_of_day()
```

```
4
FRIDAY
```

[4]:
```
calendar.month(1947, 8)
```

[4]:
```
'    August 1947\nMo Tu We Th Fr Sa Su\n             1  2  3\n 4  5  6  7  8  9
10\n11 12 13 14 15 16 17\n18 19 20 21 22 23 24\n25 26 27 28 29 30 31\n'
```

[5]:
```
print(calendar.month(1947, 8))
```

```
    August 1947
Mo Tu We Th Fr Sa Su
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

[11]:
```
print(calendar.monthrange(1947, 8))
# Return weekday (0-6 ~ Mon-Sun) and number of days (28-31) for year, month
# 4 - Friday
```

```
(4, 31)
```

[6]:
```
print(calendar.monthcalendar(1947, 8))
```

```
[[0, 0, 0, 0, 1, 2, 3], [4, 5, 6, 7, 8, 9, 10], [11, 12, 13, 14, 15, 16, 17],
[18, 19, 20, 21, 22, 23, 24], [25, 26, 27, 28, 29, 30, 31]]
```

[7]:
```
for i in calendar.monthcalendar(1947, 8):
    print(i)
```

```
[0, 0, 0, 0, 1, 2, 3]
[4, 5, 6, 7, 8, 9, 10]
[11, 12, 13, 14, 15, 16, 17]
[18, 19, 20, 21, 22, 23, 24]
[25, 26, 27, 28, 29, 30, 31]
```

0.0.1 Displaying the calendar

[23]:
```
help(calendar.calendar)
```

```
Help on method formatyear in module calendar:

formatyear(theyear, w=2, l=1, c=6, m=3) method of calendar.TextCalendar instance
    Returns a year's calendar as a multi-line string.
```

[25]:
```
print(calendar.calendar(1947, 2, 1, 10))
```

```
                                  1947

      January                   February                  March
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
       1  2  3  4  5                  1  2                      1  2
 6  7  8  9 10 11 12       3  4  5  6  7  8  9       3  4  5  6  7  8  9
13 14 15 16 17 18 19      10 11 12 13 14 15 16      10 11 12 13 14 15 16
20 21 22 23 24 25 26      17 18 19 20 21 22 23      17 18 19 20 21 22 23
27 28 29 30 31            24 25 26 27 28            24 25 26 27 28 29 30
                                                    31

      April                     May                       June
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
    1  2  3  4  5  6               1  2  3  4                         1
 7  8  9 10 11 12 13       5  6  7  8  9 10 11       2  3  4  5  6  7  8
14 15 16 17 18 19 20      12 13 14 15 16 17 18       9 10 11 12 13 14 15
21 22 23 24 25 26 27      19 20 21 22 23 24 25      16 17 18 19 20 21 22
28 29 30                  26 27 28 29 30 31         23 24 25 26 27 28 29
                                                    30

      July                    August                  September
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
    1  2  3  4  5  6                  1  2  3                1  2  3  4
 7  8  9 10 11 12 13       4  5  6  7  8  9 10       8  9 10 11 12 13 14
14 15 16 17 18 19 20      11 12 13 14 15 16 17      15 16 17 18 19 20 21
21 22 23 24 25 26 27      18 19 20 21 22 23 24      22 23 24 25 26 27 28
28 29 30 31               25 26 27 28 29 30 31      29 30

      October                   November                  December
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
       1  2  3  4  5                  1  2                1  2  3  4  5
 6  7  8  9 10 11 12       3  4  5  6  7  8  9       8  9 10 11 12 13 14
13 14 15 16 17 18 19      10 11 12 13 14 15 16      15 16 17 18 19 20 21
20 21 22 23 24 25 26      17 18 19 20 21 22 23      22 23 24 25 26 27 28
27 28 29 30 31            24 25 26 27 28 29 30      29 30 31
```

[27]:
```
# European calendar starts with Monday
c = calendar.TextCalendar()
c.prmonth(1947, 8)
```

```
    August 1947
Mo Tu We Th Fr Sa Su
             1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

[28]:
```
# American calendar starts with Sunday
c = calendar.TextCalendar(calendar.SUNDAY)
c.prmonth(1947, 8)
```

```
    August 1947
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

[29]:
```
# Arabic calendar starts with Friday
c = calendar.TextCalendar(calendar.FRIDAY)
c.prmonth(1947, 8)
```

```
    August 1947
Fr Sa Su Mo Tu We Th
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

To display in Local language, https://docs.microsoft.com/en-us/cpp/c-runtime-library/language-strings?view=vs-2017

[30]:
```
import locale
import sys

for x in locale.windows_locale.values():
    if sys.platform == 'win32':
        print(x.replace('_', '-'))
    else:
        print(x)
```

```
af-ZA
sq-AL
gsw-FR
am-ET
ar-SA
ar-IQ
ar-EG
ar-LY
ar-DZ
ar-MA
```

ar-TN
ar-OM
ar-YE
ar-SY
ar-JO
ar-LB
ar-KW
ar-AE
ar-BH
ar-QA
hy-AM
as-IN
az-AZ
az-AZ
ba-RU
eu-ES
be-BY
bn-IN
bs-BA
bs-BA
br-FR
bg-BG
ca-ES
zh-CHS
zh-TW
zh-CN
zh-HK
zh-SG
zh-MO
zh-CHT
co-FR
hr-HR
hr-BA
cs-CZ
da-DK
gbz-AF
div-MV
nl-NL
nl-BE
en-US
en-GB
en-AU
en-CA
en-NZ
en-IE
en-ZA
en-JA
en-CB

en-BZ
en-TT
en-ZW
en-PH
en-IN
en-MY
en-IN
et-EE
fo-FO
fil-PH
fi-FI
fr-FR
fr-BE
fr-CA
fr-CH
fr-LU
fr-MC
fy-NL
gl-ES
ka-GE
de-DE
de-CH
de-AT
de-LU
de-LI
el-GR
kl-GL
gu-IN
ha-NG
he-IL
hi-IN
hu-HU
is-IS
id-ID
iu-CA
iu-CA
ga-IE
it-IT
it-CH
ja-JP
kn-IN
kk-KZ
kh-KH
qut-GT
rw-RW
kok-IN
ko-KR
ky-KG

lo-LA
lv-LV
lt-LT
dsb-DE
lb-LU
mk-MK
ms-MY
ms-BN
ml-IN
mt-MT
mi-NZ
arn-CL
mr-IN
moh-CA
mn-MN
mn-CN
ne-NP
nb-NO
nn-NO
oc-FR
or-IN
ps-AF
fa-IR
pl-PL
pt-BR
pt-PT
pa-IN
quz-BO
quz-EC
quz-PE
ro-RO
rm-CH
ru-RU
smn-FI
smj-NO
smj-SE
se-NO
se-SE
se-FI
sms-FI
sma-NO
sma-SE
sa-IN
sr-SP
sr-BA
sr-SP
sr-BA
si-LK

ns-ZA
tn-ZA
sk-SK
sl-SI
es-ES
es-MX
es-ES
es-GT
es-CR
es-PA
es-DO
es-VE
es-CO
es-PE
es-AR
es-EC
es-CL
es-UR
es-PY
es-BO
es-SV
es-HN
es-NI
es-PR
es-US
sw-KE
sv-SE
sv-FI
syr-SY
tg-TJ
tmz-DZ
ta-IN
tt-RU
te-IN
th-TH
bo-BT
bo-CN
tr-TR
tk-TM
ug-CN
uk-UA
wen-DE
ur-PK
ur-IN
uz-UZ
uz-UZ
vi-VN
cy-GB

```
wo-SN
xh-ZA
sah-RU
ii-CN
yo-NG
zu-ZA
```

[31]:
```
c = calendar.LocaleTextCalendar(locale='en-US')
c.prmonth(1947, 8)
```

```
    August 1947
Mo Tu We Th Fr Sa Su
                1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

[32]:
```
c = calendar.LocaleTextCalendar(locale='ru-RU')
c.prmonth(1947, 8)
```

```
    Àâãóñò 1947
Ïí Âò Ñð Œò Ïò Ñá Âñ
                1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

[33]:
```
c = calendar.LocaleTextCalendar(locale='vi-VN')
c.prmonth(1947, 8)
```

```
  Thaùng Taùm 1947
Ha Ba Tý Nã Sa Ba CN
                1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

[34]:
```
c = calendar.LocaleTextCalendar(locale='sah-RU')
c.prmonth(1947, 8)
```

```
 Àòûðäüàõ ûéà 1947
Áí Îï Ñð Œï Áò Ñá Áñ
                1  2  3
```

```
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

[12]:
```
calendar.weekday(1947, 8, 15)
# Return weekday (0-6 ~ Mon-Sun) for year, month (1-12), day (1-31).
```

[12]: 4

[13]:
```
calendar.isleap(2019)
```

[13]: False

[17]:
```
calendar.isleap(2020)
```

[17]: True

[14]:
```
calendar.leapdays(2000, 2030)
```

[14]: 8

[10]:
```
help(calendar)
```

```
Help on module calendar:

NAME
    calendar - Calendar printing functions

DESCRIPTION
    Note when comparing these calendars to the ones printed by cal(1): By
    default, these calendars have Monday as the first day of the week, and
    Sunday as the last (the European convention). Use setfirstweekday() to
    set the first day of the week (0=Monday, 6=Sunday).

CLASSES
    builtins.ValueError(builtins.Exception)
        IllegalMonthError
        IllegalWeekdayError
    builtins.object
        Calendar
            HTMLCalendar
                LocaleHTMLCalendar
            TextCalendar
                LocaleTextCalendar

    class Calendar(builtins.object)
     |  Calendar(firstweekday=0)
     |
     |  Base calendar class. This class doesn't do any formatting. It simply
     |  provides data to subclasses.
```

```
     |
     |  Methods defined here:
     |
     |  __init__(self, firstweekday=0)
     |      Initialize self.  See help(type(self)) for accurate signature.
     |
     |  getfirstweekday(self)
     |
     |  itermonthdates(self, year, month)
     |      Return an iterator for one month. The iterator will yield
datetime.date
     |      values and will always iterate through complete weeks, so it will
yield
     |      dates outside the specified month.
     |
     |  itermonthdays(self, year, month)
     |      Like itermonthdates(), but will yield day numbers. For days outside
     |      the specified month the day number is 0.
     |
     |  itermonthdays2(self, year, month)
     |      Like itermonthdates(), but will yield (day number, weekday number)
     |      tuples. For days outside the specified month the day number is 0.
     |
     |  itermonthdays3(self, year, month)
     |      Like itermonthdates(), but will yield (year, month, day) tuples.
Can be
     |      used for dates outside of datetime.date range.
     |
     |  itermonthdays4(self, year, month)
     |      Like itermonthdates(), but will yield (year, month, day,
day_of_week) tuples.
     |      Can be used for dates outside of datetime.date range.
     |
     |  iterweekdays(self)
     |      Return an iterator for one week of weekday numbers starting with the
     |      configured first one.
     |
     |  monthdatescalendar(self, year, month)
     |      Return a matrix (list of lists) representing a month's calendar.
     |      Each row represents a week; week entries are datetime.date values.
     |
     |  monthdays2calendar(self, 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(self, year, month)
```

```
     |      Return a matrix representing a month's calendar.
     |      Each row represents a week; days outside this month are zero.
     |
     |  setfirstweekday(self, firstweekday)
     |
     |  yeardatescalendar(self, year, width=3)
     |      Return the data for the specified year ready for formatting. The
return
     |      value is a list of month rows. Each month row contains up to width
months.
     |      Each month contains between 4 and 6 weeks and each week contains 1-7
     |      days. Days are datetime.date objects.
     |
     |  yeardays2calendar(self, year, width=3)
     |      Return the data for the specified year ready for formatting (similar
to
     |      yeardatescalendar()). Entries in the week lists are
     |      (day number, weekday number) tuples. Day numbers outside this month
are
     |      zero.
     |
     |  yeardayscalendar(self, year, width=3)
     |      Return the data for the specified year ready for formatting (similar
to
     |      yeardatescalendar()). Entries in the week lists are day numbers.
     |      Day numbers outside this month are zero.
     |
     |  ----------------------------------------------------------------------
     |  Data descriptors defined here:
     |
     |  __dict__
     |      dictionary for instance variables (if defined)
     |
     |  __weakref__
     |      list of weak references to the object (if defined)
     |
     |  firstweekday

    class HTMLCalendar(Calendar)
     |  HTMLCalendar(firstweekday=0)
     |
     |  This calendar returns complete HTML pages.
     |
     |  Method resolution order:
     |      HTMLCalendar
     |      Calendar
     |      builtins.object
     |
```

```
      |  Methods defined here:
      |
      |  formatday(self, day, weekday)
      |      Return a day as a table cell.
      |
      |  formatmonth(self, theyear, themonth, withyear=True)
      |      Return a formatted month as a table.
      |
      |  formatmonthname(self, theyear, themonth, withyear=True)
      |      Return a month name as a table row.
      |
      |  formatweek(self, theweek)
      |      Return a complete week as a table row.
      |
      |  formatweekday(self, day)
      |      Return a weekday name as a table header.
      |
      |  formatweekheader(self)
      |      Return a header for a week as a table row.
      |
      |  formatyear(self, theyear, width=3)
      |      Return a formatted year as a table of tables.
      |
      |  formatyearpage(self, theyear, width=3, css='calendar.css',
encoding=None)
      |      Return a formatted year as a complete HTML page.
      |
      |  ----------------------------------------------------------------------
      |  Data and other attributes defined here:
      |
      |  cssclass_month = 'month'
      |
      |  cssclass_month_head = 'month'
      |
      |  cssclass_noday = 'noday'
      |
      |  cssclass_year = 'year'
      |
      |  cssclass_year_head = 'year'
      |
      |  cssclasses = ['mon', 'tue', 'wed', 'thu', 'fri', 'sat', 'sun']
      |
      |  cssclasses_weekday_head = ['mon', 'tue', 'wed', 'thu', 'fri', 'sat',
'...
      |
      |  ----------------------------------------------------------------------
      |  Methods inherited from Calendar:
      |
```

```
 |  __init__(self, firstweekday=0)
 |      Initialize self.  See help(type(self)) for accurate signature.
 |
 |  getfirstweekday(self)
 |
 |  itermonthdates(self, year, month)
 |      Return an iterator for one month. The iterator will yield
datetime.date
 |      values and will always iterate through complete weeks, so it will
yield
 |      dates outside the specified month.
 |
 |  itermonthdays(self, year, month)
 |      Like itermonthdates(), but will yield day numbers. For days outside
 |      the specified month the day number is 0.
 |
 |  itermonthdays2(self, year, month)
 |      Like itermonthdates(), but will yield (day number, weekday number)
 |      tuples. For days outside the specified month the day number is 0.
 |
 |  itermonthdays3(self, year, month)
 |      Like itermonthdates(), but will yield (year, month, day) tuples.
Can be
 |      used for dates outside of datetime.date range.
 |
 |  itermonthdays4(self, year, month)
 |      Like itermonthdates(), but will yield (year, month, day,
day_of_week) tuples.
 |      Can be used for dates outside of datetime.date range.
 |
 |  iterweekdays(self)
 |      Return an iterator for one week of weekday numbers starting with the
 |      configured first one.
 |
 |  monthdatescalendar(self, year, month)
 |      Return a matrix (list of lists) representing a month's calendar.
 |      Each row represents a week; week entries are datetime.date values.
 |
 |  monthdays2calendar(self, 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(self, year, month)
 |      Return a matrix representing a month's calendar.
 |      Each row represents a week; days outside this month are zero.
 |
```

```
     |  setfirstweekday(self, firstweekday)
     |
     |  yeardatescalendar(self, year, width=3)
     |      Return the data for the specified year ready for formatting. The
return
     |      value is a list of month rows. Each month row contains up to width
months.
     |      Each month contains between 4 and 6 weeks and each week contains 1-7
     |      days. Days are datetime.date objects.
     |
     |  yeardays2calendar(self, year, width=3)
     |      Return the data for the specified year ready for formatting (similar
to
     |      yeardatescalendar()). Entries in the week lists are
     |      (day number, weekday number) tuples. Day numbers outside this month
are
     |      zero.
     |
     |  yeardayscalendar(self, year, width=3)
     |      Return the data for the specified year ready for formatting (similar
to
     |      yeardatescalendar()). Entries in the week lists are day numbers.
     |      Day numbers outside this month are zero.
     |
     |  ----------------------------------------------------------------------
     |  Data descriptors inherited from Calendar:
     |
     |  __dict__
     |      dictionary for instance variables (if defined)
     |
     |  __weakref__
     |      list of weak references to the object (if defined)
     |
     |  firstweekday

    class IllegalMonthError(builtins.ValueError)
     |  IllegalMonthError(month)
     |
     |  Inappropriate argument value (of correct type).
     |
     |  Method resolution order:
     |      IllegalMonthError
     |      builtins.ValueError
     |      builtins.Exception
     |      builtins.BaseException
     |      builtins.object
     |
     |  Methods defined here:
```

```
      |
      |  __init__(self, month)
      |      Initialize self.  See help(type(self)) for accurate signature.
      |
      |  __str__(self)
      |      Return str(self).
      |
      |  ----------------------------------------------------------------------
      |  Data descriptors defined here:
      |
      |  __weakref__
      |      list of weak references to the object (if defined)
      |
      |  ----------------------------------------------------------------------
      |  Static methods inherited from builtins.ValueError:
      |
      |  __new__(*args, **kwargs) from builtins.type
      |      Create and return a new object.  See help(type) for accurate
signature.
      |
      |  ----------------------------------------------------------------------
      |  Methods inherited from builtins.BaseException:
      |
      |  __delattr__(self, name, /)
      |      Implement delattr(self, name).
      |
      |  __getattribute__(self, name, /)
      |      Return getattr(self, name).
      |
      |  __reduce__(...)
      |      Helper for pickle.
      |
      |  __repr__(self, /)
      |      Return repr(self).
      |
      |  __setattr__(self, name, value, /)
      |      Implement setattr(self, name, value).
      |
      |  __setstate__(...)
      |
      |  with_traceback(...)
      |      Exception.with_traceback(tb) --
      |      set self.__traceback__ to tb and return self.
      |
      |  ----------------------------------------------------------------------
      |  Data descriptors inherited from builtins.BaseException:
      |
      |  __cause__
```

```
 |      exception cause
 |
 |  __context__
 |      exception context
 |
 |  __dict__
 |
 |  __suppress_context__
 |
 |  __traceback__
 |
 |  args

class IllegalWeekdayError(builtins.ValueError)
 |  IllegalWeekdayError(weekday)
 |
 |  Inappropriate argument value (of correct type).
 |
 |  Method resolution order:
 |      IllegalWeekdayError
 |      builtins.ValueError
 |      builtins.Exception
 |      builtins.BaseException
 |      builtins.object
 |
 |  Methods defined here:
 |
 |  __init__(self, weekday)
 |      Initialize self.  See help(type(self)) for accurate signature.
 |
 |  __str__(self)
 |      Return str(self).
 |
 |  ----------------------------------------------------------------------
 |  Data descriptors defined here:
 |
 |  __weakref__
 |      list of weak references to the object (if defined)
 |
 |  ----------------------------------------------------------------------
 |  Static methods inherited from builtins.ValueError:
 |
 |  __new__(*args, **kwargs) from builtins.type
 |      Create and return a new object.  See help(type) for accurate
signature.
 |
 |  ----------------------------------------------------------------------
 |  Methods inherited from builtins.BaseException:
```

```
     |
     |  __delattr__(self, name, /)
     |      Implement delattr(self, name).
     |
     |  __getattribute__(self, name, /)
     |      Return getattr(self, name).
     |
     |  __reduce__(...)
     |      Helper for pickle.
     |
     |  __repr__(self, /)
     |      Return repr(self).
     |
     |  __setattr__(self, name, value, /)
     |      Implement setattr(self, name, value).
     |
     |  __setstate__(...)
     |
     |  with_traceback(...)
     |      Exception.with_traceback(tb) --
     |      set self.__traceback__ to tb and return self.
     |
     |  ----------------------------------------------------------------------
     |  Data descriptors inherited from builtins.BaseException:
     |
     |  __cause__
     |      exception cause
     |
     |  __context__
     |      exception context
     |
     |  __dict__
     |
     |  __suppress_context__
     |
     |  __traceback__
     |
     |  args

    class LocaleHTMLCalendar(HTMLCalendar)
     |  LocaleHTMLCalendar(firstweekday=0, locale=None)
     |
     |  This class can be passed a locale name in the constructor and will
return
     |  month and weekday names in the specified locale. If this locale includes
     |  an encoding all strings containing month and weekday names will be
returned
     |  as unicode.
```

```
 |
 |  Method resolution order:
 |      LocaleHTMLCalendar
 |      HTMLCalendar
 |      Calendar
 |      builtins.object
 |
 |  Methods defined here:
 |
 |  __init__(self, firstweekday=0, locale=None)
 |      Initialize self.  See help(type(self)) for accurate signature.
 |
 |  formatmonthname(self, theyear, themonth, withyear=True)
 |      Return a month name as a table row.
 |
 |  formatweekday(self, day)
 |      Return a weekday name as a table header.
 |
 |  ----------------------------------------------------------------------
 |  Methods inherited from HTMLCalendar:
 |
 |  formatday(self, day, weekday)
 |      Return a day as a table cell.
 |
 |  formatmonth(self, theyear, themonth, withyear=True)
 |      Return a formatted month as a table.
 |
 |  formatweek(self, theweek)
 |      Return a complete week as a table row.
 |
 |  formatweekheader(self)
 |      Return a header for a week as a table row.
 |
 |  formatyear(self, theyear, width=3)
 |      Return a formatted year as a table of tables.
 |
 |  formatyearpage(self, theyear, width=3, css='calendar.css',
encoding=None)
 |      Return a formatted year as a complete HTML page.
 |
 |  ----------------------------------------------------------------------
 |  Data and other attributes inherited from HTMLCalendar:
 |
 |  cssclass_month = 'month'
 |
 |  cssclass_month_head = 'month'
 |
 |  cssclass_noday = 'noday'
```

```
     |
     |  cssclass_year = 'year'
     |
     |  cssclass_year_head = 'year'
     |
     |  cssclasses = ['mon', 'tue', 'wed', 'thu', 'fri', 'sat', 'sun']
     |
     |  cssclasses_weekday_head = ['mon', 'tue', 'wed', 'thu', 'fri', 'sat',
'...
     |
     |  ----------------------------------------------------------------------
     |  Methods inherited from Calendar:
     |
     |  getfirstweekday(self)
     |
     |  itermonthdates(self, year, month)
     |      Return an iterator for one month. The iterator will yield
datetime.date
     |      values and will always iterate through complete weeks, so it will
yield
     |      dates outside the specified month.
     |
     |  itermonthdays(self, year, month)
     |      Like itermonthdates(), but will yield day numbers. For days outside
     |      the specified month the day number is 0.
     |
     |  itermonthdays2(self, year, month)
     |      Like itermonthdates(), but will yield (day number, weekday number)
     |      tuples. For days outside the specified month the day number is 0.
     |
     |  itermonthdays3(self, year, month)
     |      Like itermonthdates(), but will yield (year, month, day) tuples.
Can be
     |      used for dates outside of datetime.date range.
     |
     |  itermonthdays4(self, year, month)
     |      Like itermonthdates(), but will yield (year, month, day,
day_of_week) tuples.
     |      Can be used for dates outside of datetime.date range.
     |
     |  iterweekdays(self)
     |      Return an iterator for one week of weekday numbers starting with the
     |      configured first one.
     |
     |  monthdatescalendar(self, year, month)
     |      Return a matrix (list of lists) representing a month's calendar.
     |      Each row represents a week; week entries are datetime.date values.
     |
```

```
     |  monthdays2calendar(self, 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(self, year, month)
     |      Return a matrix representing a month's calendar.
     |      Each row represents a week; days outside this month are zero.
     |
     |  setfirstweekday(self, firstweekday)
     |
     |  yeardatescalendar(self, year, width=3)
     |      Return the data for the specified year ready for formatting. The
return
     |      value is a list of month rows. Each month row contains up to width
months.
     |      Each month contains between 4 and 6 weeks and each week contains 1-7
     |      days. Days are datetime.date objects.
     |
     |  yeardays2calendar(self, year, width=3)
     |      Return the data for the specified year ready for formatting (similar
to
     |      yeardatescalendar()). Entries in the week lists are
     |      (day number, weekday number) tuples. Day numbers outside this month
are
     |      zero.
     |
     |  yeardayscalendar(self, year, width=3)
     |      Return the data for the specified year ready for formatting (similar
to
     |      yeardatescalendar()). Entries in the week lists are day numbers.
     |      Day numbers outside this month are zero.
     |
     |  ----------------------------------------------------------------------
     |  Data descriptors inherited from Calendar:
     |
     |  __dict__
     |      dictionary for instance variables (if defined)
     |
     |  __weakref__
     |      list of weak references to the object (if defined)
     |
     |  firstweekday

    class LocaleTextCalendar(TextCalendar)
     |  LocaleTextCalendar(firstweekday=0, locale=None)
     |
```

```
     |  This class can be passed a locale name in the constructor and will
return
     |  month and weekday names in the specified locale. If this locale includes
     |  an encoding all strings containing month and weekday names will be
returned
     |  as unicode.
     |
     |  Method resolution order:
     |      LocaleTextCalendar
     |      TextCalendar
     |      Calendar
     |      builtins.object
     |
     |  Methods defined here:
     |
     |  __init__(self, firstweekday=0, locale=None)
     |      Initialize self.  See help(type(self)) for accurate signature.
     |
     |  formatmonthname(self, theyear, themonth, width, withyear=True)
     |      Return a formatted month name.
     |
     |  formatweekday(self, day, width)
     |      Returns a formatted week day name.
     |
     |  ----------------------------------------------------------------------
     |  Methods inherited from TextCalendar:
     |
     |  formatday(self, day, weekday, width)
     |      Returns a formatted day.
     |
     |  formatmonth(self, theyear, themonth, w=0, l=0)
     |      Return a month's calendar string (multi-line).
     |
     |  formatweek(self, theweek, width)
     |      Returns a single week in a string (no newline).
     |
     |  formatweekheader(self, width)
     |      Return a header for a week.
     |
     |  formatyear(self, theyear, w=2, l=1, c=6, m=3)
     |      Returns a year's calendar as a multi-line string.
     |
     |  prmonth(self, theyear, themonth, w=0, l=0)
     |      Print a month's calendar.
     |
     |  prweek(self, theweek, width)
     |      Print a single week (no newline).
     |
```

```
     |  pryear(self, theyear, w=0, l=0, c=6, m=3)
     |      Print a year's calendar.
     |
     |  ----------------------------------------------------------------------
     |  Methods inherited from Calendar:
     |
     |  getfirstweekday(self)
     |
     |  itermonthdates(self, year, month)
     |      Return an iterator for one month. The iterator will yield
datetime.date
     |      values and will always iterate through complete weeks, so it will
yield
     |      dates outside the specified month.
     |
     |  itermonthdays(self, year, month)
     |      Like itermonthdates(), but will yield day numbers. For days outside
     |      the specified month the day number is 0.
     |
     |  itermonthdays2(self, year, month)
     |      Like itermonthdates(), but will yield (day number, weekday number)
     |      tuples. For days outside the specified month the day number is 0.
     |
     |  itermonthdays3(self, year, month)
     |      Like itermonthdates(), but will yield (year, month, day) tuples.
Can be
     |      used for dates outside of datetime.date range.
     |
     |  itermonthdays4(self, year, month)
     |      Like itermonthdates(), but will yield (year, month, day,
day_of_week) tuples.
     |      Can be used for dates outside of datetime.date range.
     |
     |  iterweekdays(self)
     |      Return an iterator for one week of weekday numbers starting with the
     |      configured first one.
     |
     |  monthdatescalendar(self, year, month)
     |      Return a matrix (list of lists) representing a month's calendar.
     |      Each row represents a week; week entries are datetime.date values.
     |
     |  monthdays2calendar(self, 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(self, year, month)
```

```
     |      Return a matrix representing a month's calendar.
     |      Each row represents a week; days outside this month are zero.
     |
     |  setfirstweekday(self, firstweekday)
     |
     |  yeardatescalendar(self, year, width=3)
     |      Return the data for the specified year ready for formatting. The
return
     |      value is a list of month rows. Each month row contains up to width
months.
     |      Each month contains between 4 and 6 weeks and each week contains 1-7
     |      days. Days are datetime.date objects.
     |
     |  yeardays2calendar(self, year, width=3)
     |      Return the data for the specified year ready for formatting (similar
to
     |      yeardatescalendar()). Entries in the week lists are
     |      (day number, weekday number) tuples. Day numbers outside this month
are
     |      zero.
     |
     |  yeardayscalendar(self, year, width=3)
     |      Return the data for the specified year ready for formatting (similar
to
     |      yeardatescalendar()). Entries in the week lists are day numbers.
     |      Day numbers outside this month are zero.
     |
     |  ----------------------------------------------------------------------
     |  Data descriptors inherited from Calendar:
     |
     |  __dict__
     |      dictionary for instance variables (if defined)
     |
     |  __weakref__
     |      list of weak references to the object (if defined)
     |
     |  firstweekday

    class TextCalendar(Calendar)
     |  TextCalendar(firstweekday=0)
     |
     |  Subclass of Calendar that outputs a calendar as a simple plain text
     |  similar to the UNIX program cal.
     |
     |  Method resolution order:
     |      TextCalendar
     |      Calendar
     |      builtins.object
```

```
 |
 |  Methods defined here:
 |
 |  formatday(self, day, weekday, width)
 |      Returns a formatted day.
 |
 |  formatmonth(self, theyear, themonth, w=0, l=0)
 |      Return a month's calendar string (multi-line).
 |
 |  formatmonthname(self, theyear, themonth, width, withyear=True)
 |      Return a formatted month name.
 |
 |  formatweek(self, theweek, width)
 |      Returns a single week in a string (no newline).
 |
 |  formatweekday(self, day, width)
 |      Returns a formatted week day name.
 |
 |  formatweekheader(self, width)
 |      Return a header for a week.
 |
 |  formatyear(self, theyear, w=2, l=1, c=6, m=3)
 |      Returns a year's calendar as a multi-line string.
 |
 |  prmonth(self, theyear, themonth, w=0, l=0)
 |      Print a month's calendar.
 |
 |  prweek(self, theweek, width)
 |      Print a single week (no newline).
 |
 |  pryear(self, theyear, w=0, l=0, c=6, m=3)
 |      Print a year's calendar.
 |
 |  ----------------------------------------------------------------------
 |  Methods inherited from Calendar:
 |
 |  __init__(self, firstweekday=0)
 |      Initialize self.  See help(type(self)) for accurate signature.
 |
 |  getfirstweekday(self)
 |
 |  itermonthdates(self, year, month)
 |      Return an iterator for one month. The iterator will yield
datetime.date
 |      values and will always iterate through complete weeks, so it will
yield
 |      dates outside the specified month.
 |
```

```
     |  itermonthdays(self, year, month)
     |      Like itermonthdates(), but will yield day numbers. For days outside
     |      the specified month the day number is 0.
     |
     |  itermonthdays2(self, year, month)
     |      Like itermonthdates(), but will yield (day number, weekday number)
     |      tuples. For days outside the specified month the day number is 0.
     |
     |  itermonthdays3(self, year, month)
     |      Like itermonthdates(), but will yield (year, month, day) tuples.
Can be
     |      used for dates outside of datetime.date range.
     |
     |  itermonthdays4(self, year, month)
     |      Like itermonthdates(), but will yield (year, month, day,
day_of_week) tuples.
     |      Can be used for dates outside of datetime.date range.
     |
     |  iterweekdays(self)
     |      Return an iterator for one week of weekday numbers starting with the
     |      configured first one.
     |
     |  monthdatescalendar(self, year, month)
     |      Return a matrix (list of lists) representing a month's calendar.
     |      Each row represents a week; week entries are datetime.date values.
     |
     |  monthdays2calendar(self, 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(self, year, month)
     |      Return a matrix representing a month's calendar.
     |      Each row represents a week; days outside this month are zero.
     |
     |  setfirstweekday(self, firstweekday)
     |
     |  yeardatescalendar(self, year, width=3)
     |      Return the data for the specified year ready for formatting. The
return
     |      value is a list of month rows. Each month row contains up to width
months.
     |      Each month contains between 4 and 6 weeks and each week contains 1-7
     |      days. Days are datetime.date objects.
     |
     |  yeardays2calendar(self, year, width=3)
     |      Return the data for the specified year ready for formatting (similar
```

```
to
     |      yeardatescalendar()). Entries in the week lists are
     |      (day number, weekday number) tuples. Day numbers outside this month
are
     |      zero.
     |
     |  yeardayscalendar(self, year, width=3)
     |      Return the data for the specified year ready for formatting (similar
to
     |      yeardatescalendar()). Entries in the week lists are day numbers.
     |      Day numbers outside this month are zero.
     |
     |  ----------------------------------------------------------------------
     |  Data descriptors inherited from Calendar:
     |
     |  __dict__
     |      dictionary for instance variables (if defined)
     |
     |  __weakref__
     |      list of weak references to the object (if defined)
     |
     |  firstweekday

FUNCTIONS
    calendar = formatyear(theyear, w=2, l=1, c=6, m=3) method of TextCalendar
instance
        Returns a year's calendar as a multi-line string.

    firstweekday = getfirstweekday() method of TextCalendar instance

    isleap(year)
        Return True for leap years, False for non-leap years.

    leapdays(y1, y2)
        Return number of leap years in range [y1, y2).
        Assume y1 <= y2.

    month = formatmonth(theyear, themonth, w=0, l=0) method of TextCalendar
instance
        Return a month's calendar string (multi-line).

    monthcalendar = monthdayscalendar(year, month) method of TextCalendar
instance
        Return a matrix representing a month's calendar.
        Each row represents a week; days outside this month are zero.

    monthrange(year, month)
        Return weekday (0-6 ~ Mon-Sun) and number of days (28-31) for
```

```
        year, month.

    prcal = pryear(theyear, w=0, l=0, c=6, m=3) method of TextCalendar instance
        Print a year's calendar.

    prmonth(theyear, themonth, w=0, l=0) method of TextCalendar instance
        Print a month's calendar.

    setfirstweekday(firstweekday)

    timegm(tuple)
        Unrelated but handy function to calculate Unix timestamp from GMT.

    weekday(year, month, day)
        Return weekday (0-6 ~ Mon-Sun) for year, month (1-12), day (1-31).

    weekheader = formatweekheader(width) method of TextCalendar instance
        Return a header for a week.

DATA
    __all__ = ['IllegalMonthError', 'IllegalWeekdayError', 'setfirstweekda...
    day_abbr = <calendar._localized_day object>
    day_name = <calendar._localized_day object>
    month_abbr = <calendar._localized_month object>
    month_name = <calendar._localized_month object>

FILE
    c:\users\udhayprakash\appdata\local\programs\python\python37\lib\calendar.py
```