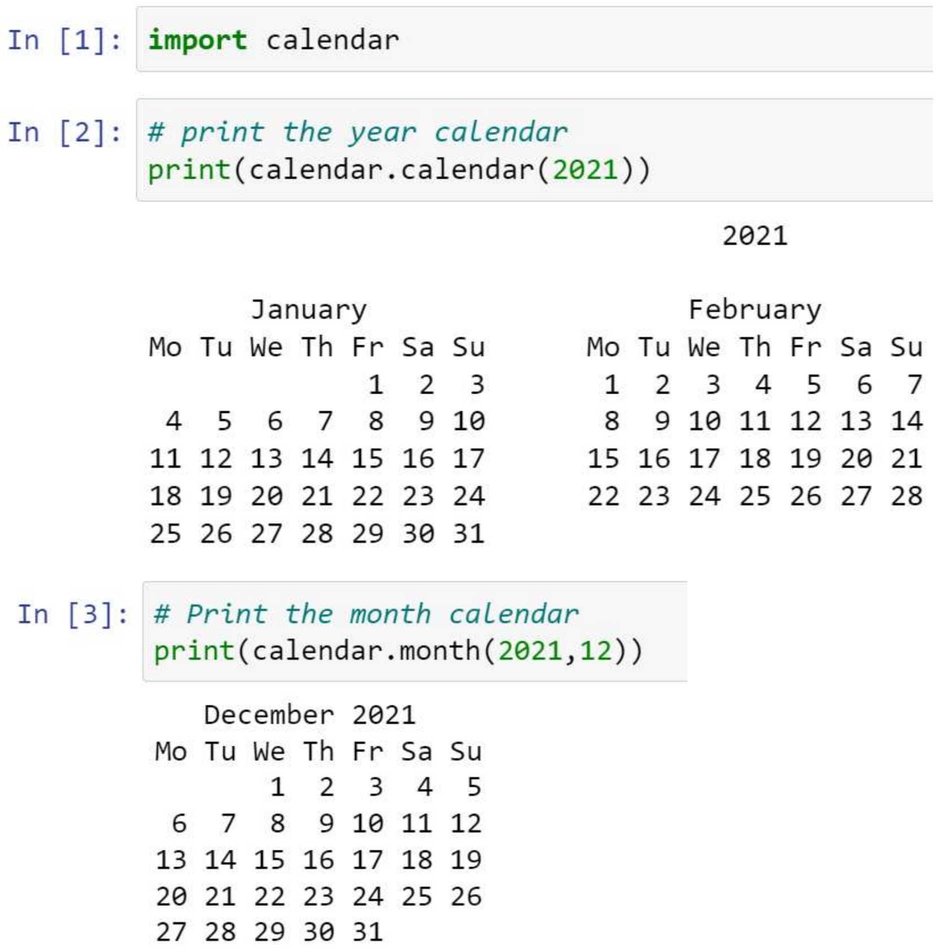
# Python library – calendar (*)

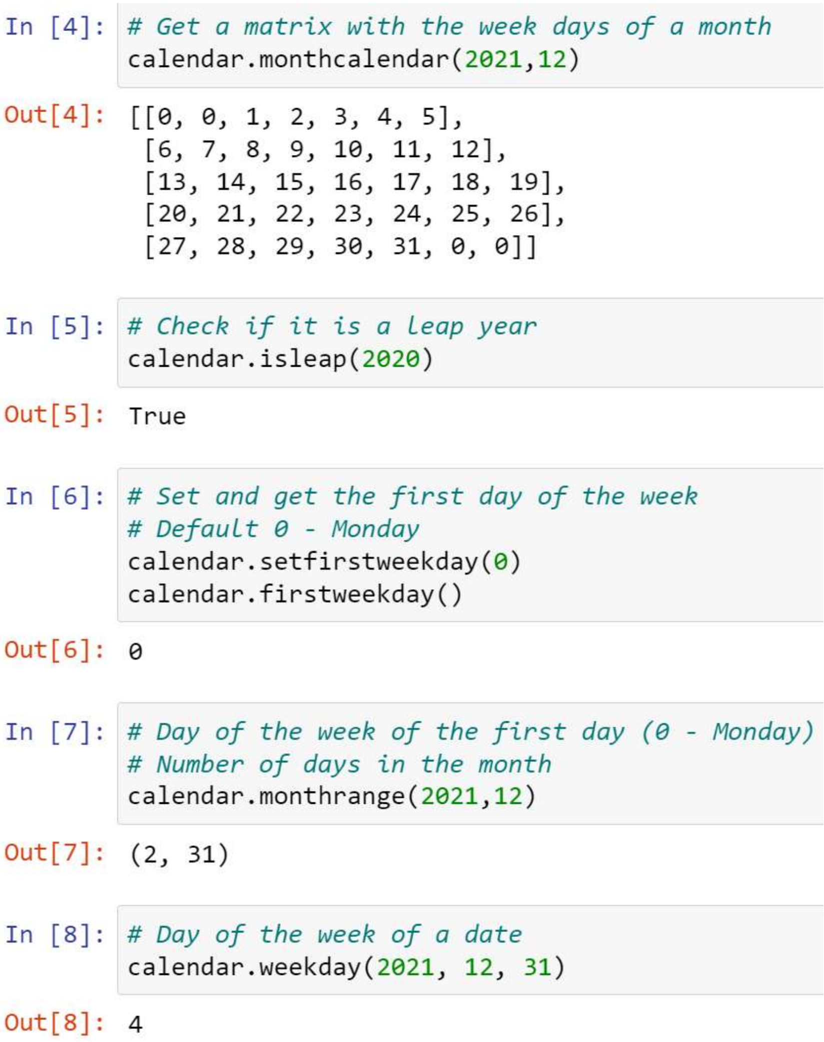
Output calendars and provides useful functions



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

```
In [2]: # print the year calendar
        print(calendar.calendar(2021))

                                  2021

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

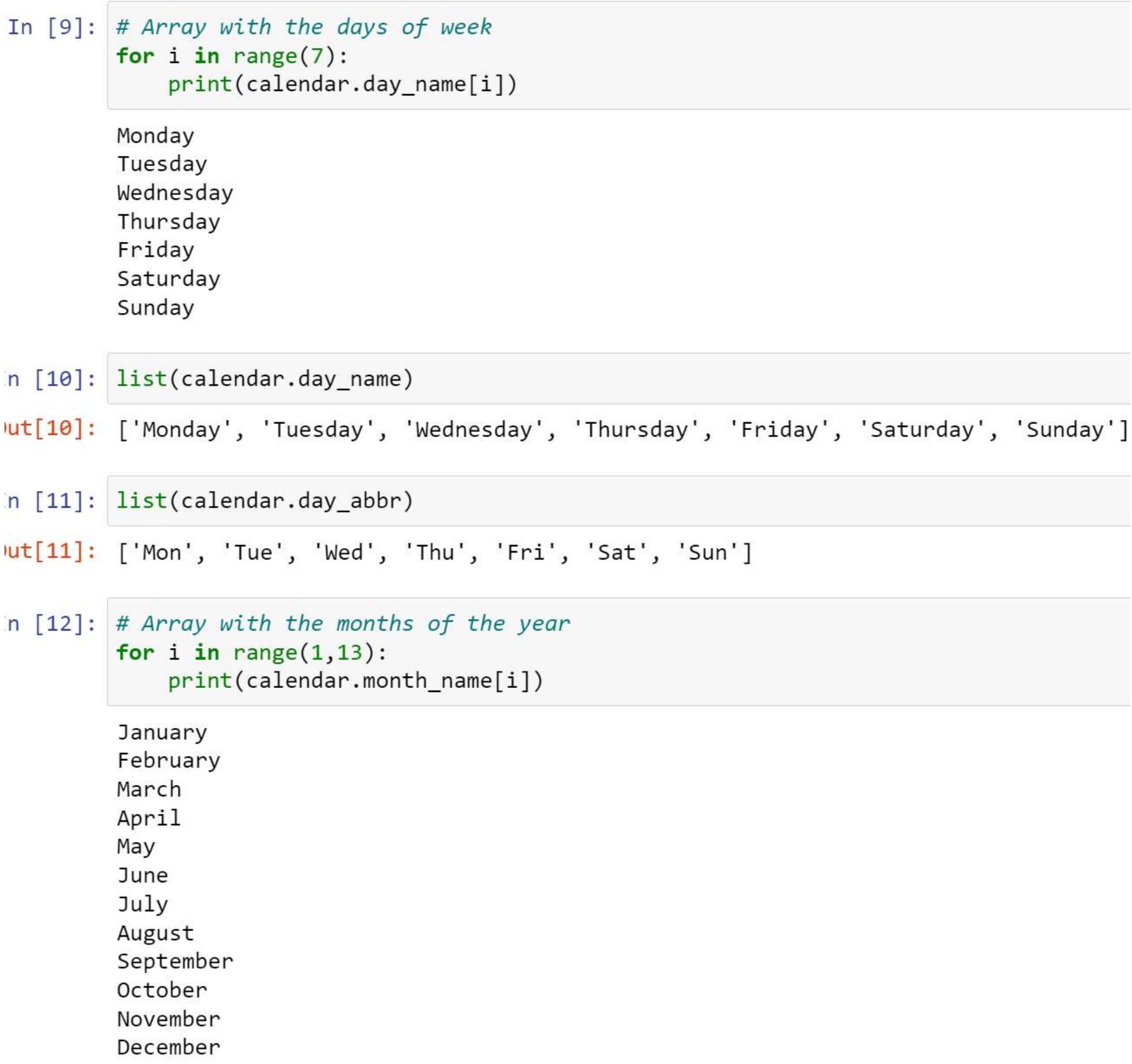
```
In [3]: # Print the month calendar
        print(calendar.month(2021,12))

   December 2021
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
```



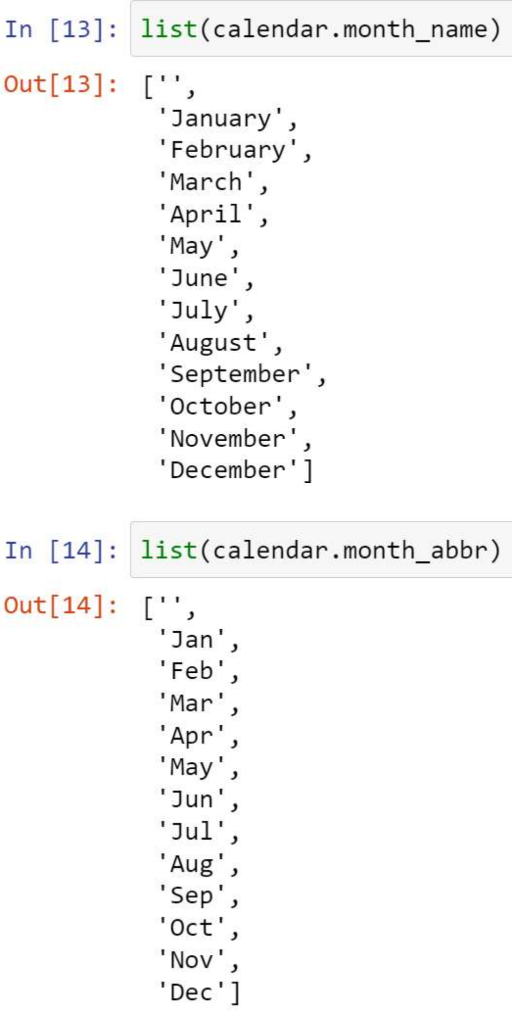
```
In [4]: # Get a matrix with the week days of a month
        calendar.monthcalendar(2021,12)

Out[4]: [[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]]

In [5]: # Check if it is a leap year
        calendar.isleap(2020)

Out[5]: True

In [6]: # Set and get the first day of the week
        # Default 0 - Monday
        calendar.setfirstweekday(0)
        calendar.firstweekday()

Out[6]: 0

In [7]: # Day of the week of the first day (0 - Monday)
        # Number of days in the month
        calendar.monthrange(2021,12)

Out[7]: (2, 31)

In [8]: # Day of the week of a date
        calendar.weekday(2021, 12, 31)

Out[8]: 4
```

(*) https://docs.python.org/3/library/calendar.html

# Python library – calendar

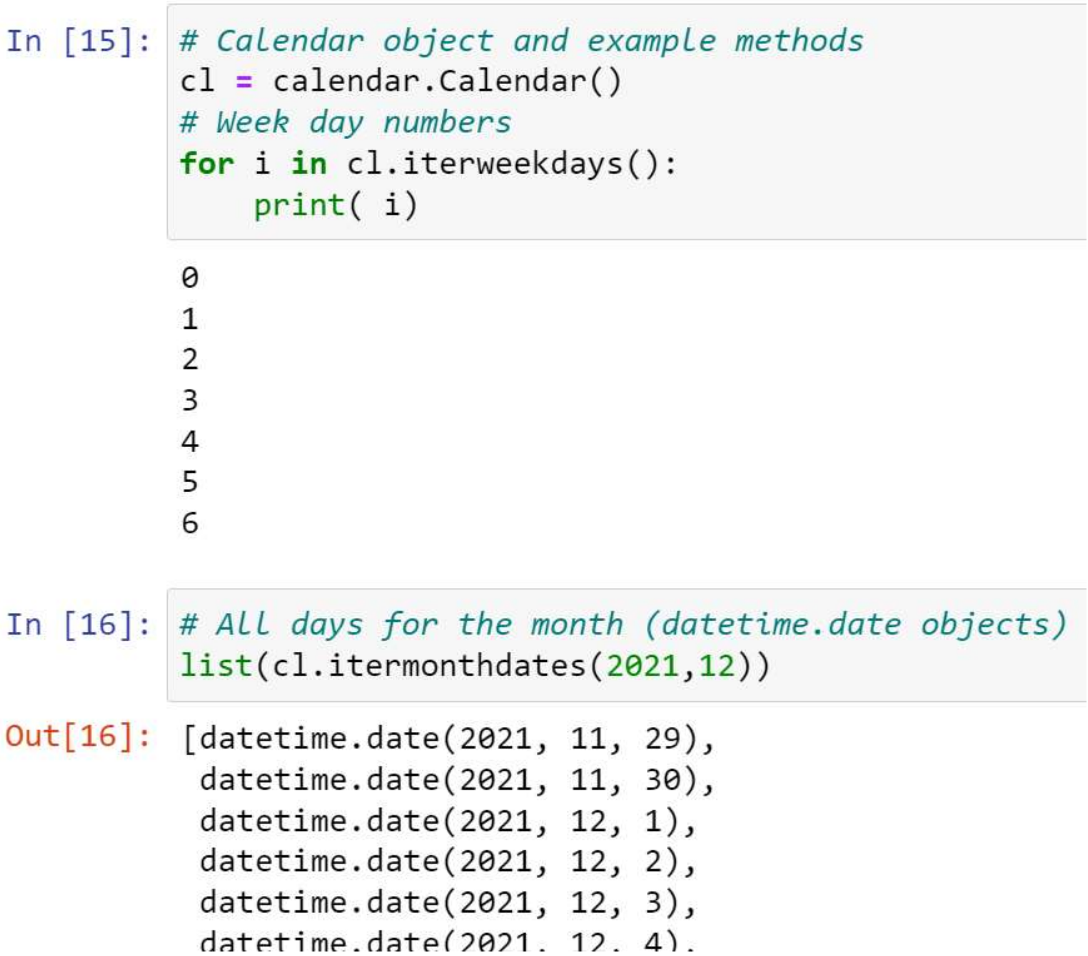


```
In [9]:  # Array with the days of week
         for i in range(7):
             print(calendar.day_name[i])

         Monday
         Tuesday
         Wednesday
         Thursday
         Friday
         Saturday
         Sunday

n [10]:  list(calendar.day_name)

ut[10]:  ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']

n [11]:  list(calendar.day_abbr)

ut[11]:  ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun']

n [12]:  # Array with the months of the year
         for i in range(1,13):
             print(calendar.month_name[i])

         January
         February
         March
         April
         May
         June
         July
         August
         September
         October
         November
         December
```

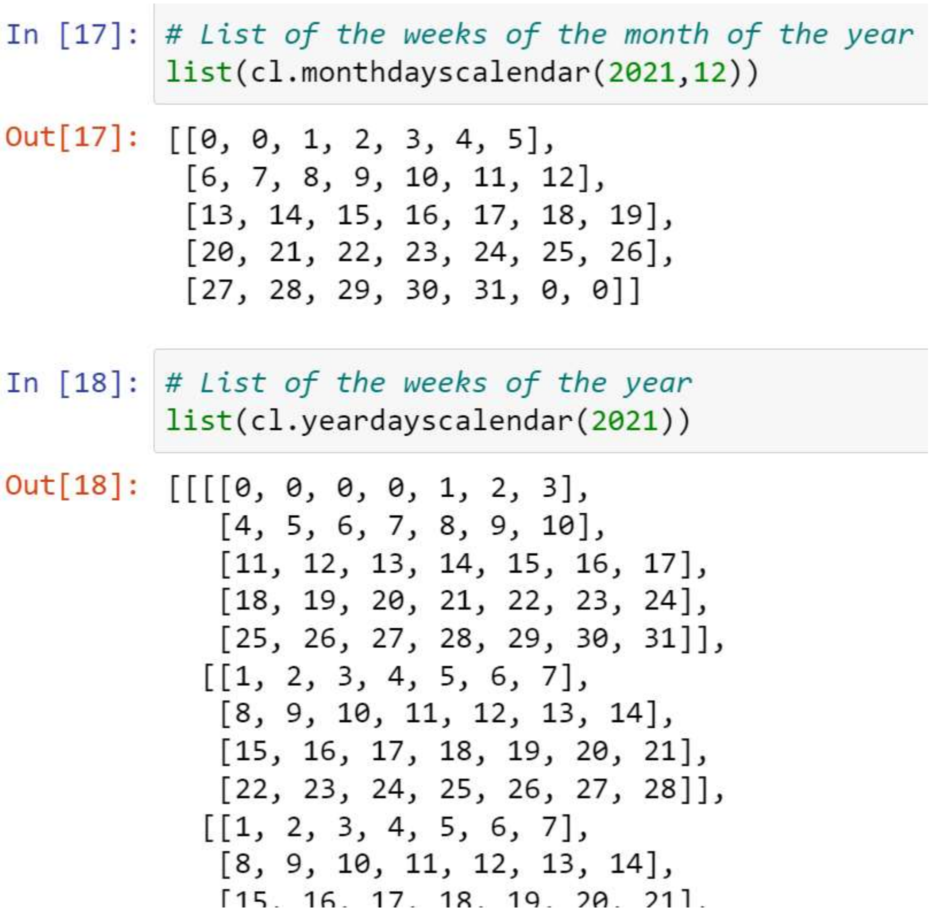


```
In [13]:  list(calendar.month_name)

Out[13]:  ['',
           'January',
           'February',
           'March',
           'April',
           'May',
           'June',
           'July',
           'August',
           'September',
           'October',
           'November',
           'December']

In [14]:  list(calendar.month_abbr)

Out[14]:  ['',
           'Jan',
           'Feb',
           'Mar',
           'Apr',
           'May',
           'Jun',
           'Jul',
           'Aug',
           'Sep',
           'Oct',
           'Nov',
           'Dec']
```

# Python library – Calendar object



```
In [15]: # Calendar object and example methods
         cl = calendar.Calendar()
         # Week day numbers
         for i in cl.iterweekdays():
             print( i)

         0
         1
         2
         3
         4
         5
         6

In [16]: # All days for the month (datetime.date objects)
         list(cl.itermonthdates(2021,12))

Out[16]: [datetime.date(2021, 11, 29),
          datetime.date(2021, 11, 30),
          datetime.date(2021, 12, 1),
          datetime.date(2021, 12, 2),
          datetime.date(2021, 12, 3),
          datetime.date(2021, 12, 4),
```



```
In [17]: # List of the weeks of the month of the year
         list(cl.monthdayscalendar(2021,12))

Out[17]: [[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]]

In [18]: # List of the weeks of the year
         list(cl.yeardayscalendar(2021))

Out[18]: [[[[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]],
            [[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]],
            [[1, 2, 3, 4, 5, 6, 7],
             [8, 9, 10, 11, 12, 13, 14],
             [15, 16, 17, 18, 19, 20, 21],
```

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

# Python library – Calendar object methods

- **iterweekdays**() - Return an iterator for the week day numbers that will be used for one week. The first value from the iterator will be the same as the value of the firstweekday property.
- **itermonthdates(year, month)** - Return an iterator for the month (1–12) in the year. This iterator will return all days (as datetime.date objects) for the month and all days before the start of the month or after the end of the month that are required to get a complete week.
- **itermonthdays(year, month)** - Return an iterator for the month in the year similar to itermonthdates(), but not restricted by the datetime.date range. Days returned will simply be day of the month numbers. For the days outside of the specified month, the day number is 0.
- **itermonthdays2(year, month)** - Return an iterator for the month in the year similar to itermonthdates(), but not restricted by the datetime.date range. Days returned will be tuples consisting of a day of the month number and a week day number.
- **itermonthdays3(year, month)** - Return an iterator for the month in the year similar to itermonthdates(), but not restricted by the datetime.date range. Days returned will be tuples consisting of a year, a month and a day of the month numbers.
- **itermonthdays4(year, month)** - Return an iterator for the month in the year similar to itermonthdates(), but not restricted by the datetime.date range. Days returned will be tuples consisting of a year, a month, a day of the month, and a day of the week numbers.
- **monthdatescalendar(year, month)** - Return a list of the weeks in the month of the year as full weeks. Weeks are lists of seven datetime.date objects.
- **monthdays2calendar(year, month)** - Return a list of the weeks in the month of the year as full weeks. Weeks are lists of seven tuples of day numbers and weekday numbers.
- **monthdayscalendar(year, month)** - Return a list of the weeks in the month of the year as full weeks. Weeks are lists of seven day numbers.
- **yeardatescalendar(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 (defaulting to 3). Each month contains between 4 and 6 weeks and each week contains 1–7 days. Days are datetime.date objects.
- **yeardays2calendar(year, width=3)** - Return the data for the specified year ready for formatting (similar to yeardatescalendar()). Entries in the week lists are tuples of day numbers and weekday numbers. Day numbers outside this month are zero.
- **yeardayscalendar(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.