Python library – dateutil (*)

dateutil module

```
In [50]: import dateutil
```

```
In [51]: dat1 = dt.datetime.now()
         dat1
Out[51]: datetime.datetime(2021, 12, 27, 11, 14, 20, 884046)
```

relativedelta

```
In [52]: # dateutil relativedelta example
         dat1 + dateutil.relativedelta.relativedelta(months=+1, weeks = +1, hour = 9)
Out[52]: datetime.datetime(2022, 2, 3, 9, 14, 20, 884046)
```

```
In [53]: # replace values (singular keywords)
         dat1 + dateutil.relativedelta.relativedelta(year = 2020, month = 1)
Out[53]: datetime.datetime(2020, 1, 27, 11, 14, 20, 884046)
```

dateutil offers a generic date/time string parser which is able to parse most known formats to represent a date and/or time.

```
In [54]: # parse example
         dateutil.parser.parse('Mon Dec 27 10:36:28')
Out[54]: datetime.datetime(2021, 12, 27, 10, 36, 28)
```

```
In [55]: dateutil.parser.parse("2021-12-27T10:49:41.5-03:00")
Out[55]: datetime.datetime(2021, 12, 27, 10, 49, 41, 500000,
         tzinfo=tzoffset(None, -10800))
```

(*) https://dateutil.readthedocs.io/en/stable/index.html

Python library – dateutil

Handling timezones with dateutil module

In [56]:
```
# dateutil tzutc example
# This is a tzinfo object that represents the UTC time zone
dat2 = dt.datetime.now(dateutil.tz.UTC)
dat2
```

Out[56]:
```
datetime.datetime(2021, 12, 27, 11, 14, 21, 6731, tzinfo=tzutc())
```

In [57]:
```
dat2.tzname()
```

Out[57]:
```
'UTC'
```

In [58]:
```
# dateutil tzoffset example
# A simple class for representing a fixed offset from UTC.
dt.datetime.now(dateutil.tz.tzoffset("BRST", -10800))
```

Out[58]:
```
datetime.datetime(2021, 12, 27, 8, 14, 21, 53276, tzinfo=tzoffset
('BRST', -10800))
```

In [59]:
```
# dateutil tzlocal example
dat3 = dt.datetime.now(tz=dateutil.tz.tzlocal())
dat3
```

Out[59]:
```
datetime.datetime(2021, 12, 27, 11, 14, 21, 79331, tzinfo=tzlocal
())
```

In [60]:
```
dat3.tzname()
```

Out[60]:
```
'GMT Standard Time'
```

In [61]:
```
# dateutil gettz example
lisbon_tz = dateutil.tz.gettz('Europe/Lisbon')
```

In [62]:
```
dat4 = dt.datetime.now(tz=lisbon_tz)
dat4
```

Out[62]:
```
datetime.datetime(2021, 12, 27, 11, 14, 21, 279023, tzinfo=tzfile
('Europe/Lisbon'))
```

In [63]:
```
dat4.tzname()
```

Out[63]:
```
'WET'
```

Python library – dateutil

Portugal timezone example

In [64]:
```
# 20/9/2021 at hour 11 in Lisbon timezone
dat5 = dt.datetime(2021, 9, 20, 11, 0, 0,
                   tzinfo=dateutil.tz.gettz('Europe/Lisbon'))
dat5
```

Out[64]:
```
datetime.datetime(2021, 9, 20, 11, 0, tzinfo=tzfile('Europe/Lisbon'))
```

In [65]:
```
# 20/9/2021 at hour 10 in UTC
dat6 = dt.datetime(2021, 9, 20, 10, 0, 0, tzinfo=dateutil.tz.UTC)
dat6
```

Out[65]:
```
datetime.datetime(2021, 9, 20, 10, 0, tzinfo=tzutc())
```

In [66]:
```
# Two dates are equal because Lisbon has a +1 hour dst
dat5 == dat6
```

Out[66]:
```
True
```

In [67]:
```
# In September Portugal timezone was WEST:
# Western European Summer Time
dat5.tzname()
```

Out[67]:
```
'WEST'
```

In [68]:
```
# In September Portugal had a +1 hour dst:
# Daylight Saving Time
dat5.dst()
```

Out[68]:
```
datetime.timedelta(seconds=3600)
```

In [69]:
```
dat5.tzinfo
```

Out[69]:
```
tzfile('Europe/Lisbon')
```