JUNAID GIRKAR
60004190057
TE COMPS A4

DateTime Module

In Python, date and time are not a data type of their own, but a module named datetime can be imported to work with the date as well as time. Python Datetime module comes built into Python, so there is no need to install it externally.

Python Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.
The DateTime module is categorized into 6 main classes –

date – An idealized naive date, assuming the current Gregorian calendar always was, and always will be, in effect. Its attributes are year, month and day.
time – An idealized time, independent of any particular day, assuming that every day has exactly 24*60*60 seconds. Its attributes are hour, minute, second, microsecond, and tzinfo.
datetime – Its a combination of date and time along with the attributes year, month, day, hour, minute, second, microsecond, and tzinfo.
timedelta – A duration expressing the difference between two date, time, or datetime instances to microsecond resolution.
tzinfo – It provides time zone information objects.
timezone – A class that implements the tzinfo abstract base class as a fixed offset from the UTC

1) datetime.date()

The date class is used to instantiate date objects in Python. When an object of this class is instantiated, it represents a date in the format YYYY-MM-DD. Constructor of this class needs three mandatory arguments year, month and date.

CODE:

```
import datetime
d=datetime.date(2020,11,29)
print(d)
```

OUTPUT:

```
2020-11-29
```

2) datetime.date.today()

To return the current local date today() function of date class is used. today() function comes with several attributes (year, month and day). These can be printed individually.

CODE:

```
import datetime
td=datetime.date.today()
print(td)
print(td.year)
print(td.month)
print(td.day)
```

OUTPUT:

```
2021-10-29
2021
10
29
```

3) weekday(), isoweekday() :

weekday() returns the day of the week as integer where Monday is 0 and Sunday is 6.
isoweekday() returns the day of the week as integer where Monday is 1 and Sunday is 7.

CODE:

```
import datetime
td=datetime.date.today()
print(td.weekday())
print(td.isoweekday())
```

OUTPUT:

```
4
5
```

4) datetime.time() :

The time class creates the time object which represents local time, independent of any day. It can take minutes, hours, seconds, milliseconds as parameters, it works without parameters as well.

CODE:

```
import datetime
t=datetime.time(10,35,55, 200)
print(t)
print(t.hour)
print(t.minute)
print(t.second)
```

OUTPUT:

```
10:35:55.000200
10
35
55
```

5) datetime.datetime() :

The datetime class contains information on both date and time. Like a date object, datetime assumes the current Gregorian calendar extended in both directions; like a time object, datetime assumes there are exactly 3600*24 seconds every day.

CODE:

```
import datetime
dt=datetime.datetime(2021,10,29,12,30,45,1000)
print(dt)
dt_today=datetime.datetime.today()
dt_now=datetime.datetime.now()
dt_utcnow=datetime.datetime.utcnow()
print(dt_today)
print(dt_now)
print(dt_utcnow)
```

OUTPUT:

```
2021-10-29 12:30:45.001000
2021-10-29 22:51:49.223397
2021-10-29 22:51:49.223397
2021-10-29 17:21:49.223397
```

6) pytz module :

pytz brings the Olson tz database into Python and thus supports almost all time zones. This module serves the date-time conversion functionalities and helps user serving international client's base.

CODE:

```
import pytz
for tz in pytz.all_timezones:
    print(tz)
```

OUTPUT:

```
Africa/Abidjan
Africa/Accra
Africa/Addis_Ababa
Africa/Algiers
.
```

```
.
.
US/Mountain
US/Pacific
US/Samoa
UTC
Universal
W-SU
WET
Zulu
```

7) datetime.datetime.strptime() :

Returns a DateTime object corresponding to the date string. You cannot create datetime object from every string. The string needs to be in a certain format

CODE:

```
import datetime
dt_str = 'September 24, 2001'
dt=datetime.datetime.strptime(dt_str,'%B %d, %Y')
print(dt)
```

OUTPUT:

```
2001-09-24 00:00:00
```

8) strftime()

The strftime() method is defined under classes date, datetime and time. The method creates a formatted string from a given date, datetime or time object.

CODE:

```
import datetime
x = datetime.datetime.today()
print(x.strftime("%B"))
print(x.strftime("%A"))
print(x.strftime("%Y"))
print(x.strftime("%X"))
print(x.strftime("%p"))
```

OUTPUT:

```
October
Friday
2021
22:56:51
PM
```

Conclusion : Datetime module in python is a very useful model for performing operations on date and time as an object and is widely used in programs that use timestamps or are time dependent.