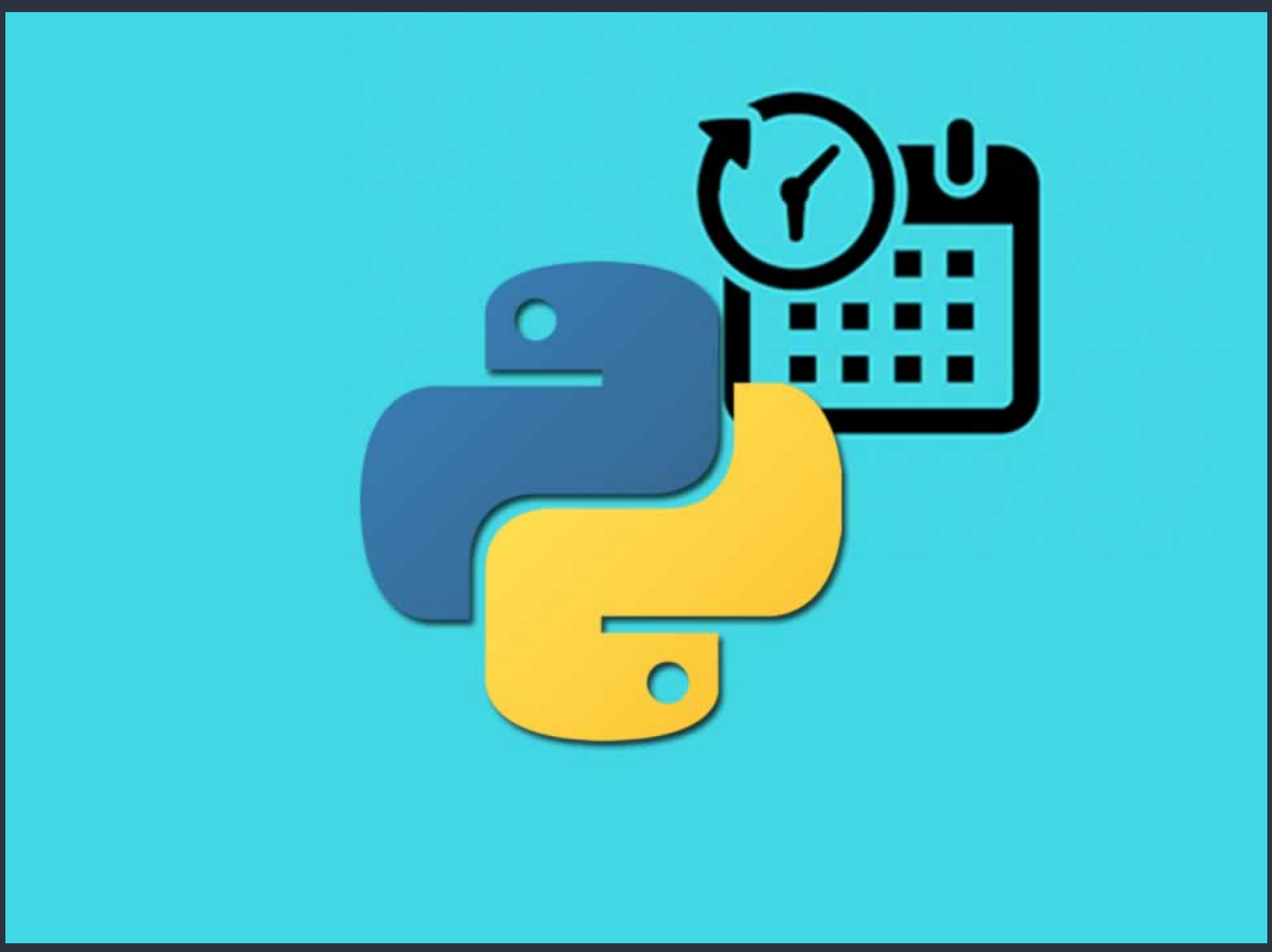
01

Python Datetime Module



DateTime Class

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 in every day.

```
class datetime.datetime(year, month, day, hour=0,
                        minute=0, second=0, microsecond=0,
                        tzinfo=None, *, fold=0)
```

The year, month and day arguments are mandatory. tzinfo can be None, rest all the attributes must be an integer

1.now()

Returns current local date and time with tz parameter

```
from datetime import datetime as dt

now_ = dt.now()
print(now_)

#################### Output ##################
2022-11-23 12:53:41.106961
```

2.astimezone()

Returns the DateTime object containing timezone information.

```
from datetime import datetime as dt

now_ = dt.now()
ast_time_zone = now_.astimezone()
print("ast_time_zone",ast_time_zone)

#################### Output ##################
ast_time_zone 2022-11-23 12:53:41.106961+00:00
```

3.date()

Return the Date class object

```
from datetime import datetime as dt

now_ = dt.now()
date_ = dt.date(now_)
print("date_",date_)

#################### Output ##################
date_ 2022-11-23
```

4.combine()

Combines the date and time objects and return a DateTime object

```
from datetime import datetime as dt
from datetime import time,date

d = date(2022,11,23)
t =  time(12,50,55)
combine_ = dt.combine(d,t)
print("combine_",combine_)

#################### Output ##################
combine_ 2022-11-23 12:50:55
```

5.ctime()

Returns a string representation of date and time

```
from datetime import datetime as dt

now_ = dt.now()
ctime_ = now_.ctime()
print('ctime_',ctime_)

#################### Output ##################
ctime_ Wed Nov 23 12:53:41 2022
```

6.fromisoformat()

Returns a datetime object from the string representation of the date and time.

```
from datetime import datetime as dt

from_iso_format = dt.fromisoformat("2022-11-23")
print("from_iso_format",type(from_iso_format))

#################### Output ##################
from_iso_format <class 'datetime.datetime'>
```

7.fromordinal()

Returns a date object from the proleptic Gregorian ordinal, where January 1 of year 1 has ordinal 1. The hour, minute, second, and microsecond are 0

```
from datetime import datetime as dt

from_ordinal= dt.fromordinal(738482)
print("from_ordinal",from_ordinal)
#################### Output ###################
from_ordinal 2022-11-23 00:00:00
```

8.fromtimestamp()

Return date and time from POSIX timestamp

```
from datetime import datetime as dt

from_time_stamp = dt.fromtimestamp(777777)
print("from_time_stamp",from_time_stamp)
#################### Output ###################
from_time_stamp 1970-01-10 00:02:57
```

9.isocalendar()

Returns a tuple year, week, and weekday

```python
from datetime import datetime as dt

now_ = dt.now()
iso_calender = dt.isocalendar(now_)
print("iso_calender",iso_calender)
#################### Output ##################
iso_calender (2022, 47, 3)
```

10.isoformat()

Return the string representation of date and time

```python
from datetime import datetime as dt

now_ = dt.now()
iso_format = dt.isoformat(now_)
print("iso_format",type(iso_format))
#################### Output ##################
iso_format <class 'str'>
```

11.isoweekday()

Returns the day of the week as integer where Monday is 1 and Sunday is 7

```
from datetime import datetime as dt

now_ = dt.now()
iso_weekday = dt.isoweekday(now_)
print("iso_weekday",iso_weekday)
#################### Output ###################
iso_weekday 3
```

12.replace()

Changes the specific attributes of the DateTime object

```
from datetime import datetime as dt

now_ = dt.now()
replace_ = now_.replace(year=2024,month=3)
print("replace_",replace_)
#################### Output ###################
replace_ 2024-03-23 12:53:41.106961
```

13.strftime()

Returns a string representation of the DateTime object with the given format

```
from datetime import datetime as dt

now_ = dt.now()
strf_time = now_.strftime("%d/%m/%Y")
print("strf_time",strf_time)
#################### Output ##################
strf_time 23/11/2022
```

14.strptime()

Returns a DateTime object corresponding to the date string

```
from datetime import datetime as dt

strp_time = dt.strptime("2022/03/12","%Y/%m/%d")
print("strp_time",strp_time)
#################### Output ##################
strp_time 2022-03-12 00:00:00
```

15.time()

Return the Time class object

```
from datetime import datetime as dt

now_=dt.now()
time_ = now_.time()
print("time_",time_)
#################### Output ###################
time_ 12:53:41.106961
```

16.timetuple()

Returns an object of type time.struct_time

```
from datetime import datetime as dt

now_=dt.now()
time_tuple = now_.timetuple()
print("time_tuple",time_tuple)
#################### Output ###################
time_tuple time.struct_time(tm_year=2022,
                            tm_mon=11, tm_mday=23,
                            tm_hour=12, tm_min=53,
                            tm_sec=41, tm_wday=2,
                            tm_yday=327, tm_isdst=-1)
```

17.timetz()

Return the Time class object

```
from datetime import datetime as dt

now_=dt.now()
time_tz = now_.timetz()
print("time_tz",time_tz)
#################### Output ##################
time_tz 12:53:41.106961
```

18.today()

Return local DateTime with tzinfo as None

```
from datetime import datetime as dt

today_ = dt.today()
print("today_",today_)
#################### Output ##################
today_ 2022-11-23 12:53:41.117303
```

19.toordinal()

Return the proleptic Gregorian ordinal of the date, where January 1 of year 1 has ordinal 1

```
from datetime import datetime as dt

now_ = dt.now()
to_ordinal = now_.toordinal()
print("to_ordinal",to_ordinal)
#################### Output ###################
to_ordinal 738482
```

20.tzname()

Returns the name of the timezone

```
from datetime import datetime as dt
from pytz import timezone

client_tz = timezone('Europe/Sofia')
schedule_date = dt.fromisoformat(
    "2022-10-26T15:00:00-00:00").astimezone(client_tz)
tz_name = schedule_date.tzname()
print("tz_name",tz_name)
#################### Output ###################
tz_name EEST
```

21.utcfromtimestamp()

Return UTC from POSIX timestamp

```
from datetime import datetime as dt

utc_form_time_stamp = dt.utcfromtimestamp(7777777)
print("utc_form_time_stamp",utc_form_time_stamp)
#################### Output ##################
utc_form_time_stamp 1970-04-01 00:29:37
```

22.utcoffset()

Returns the UTC offset

```
from datetime import datetime as dt
from pytz import timezone

client_tz = timezone('Europe/Sofia')
schedule_date = dt.fromisoformat(
  "2022-10-26T15:00:00-00:00").astimezone(client_tz)
utc_offset = dt.utcoffset(schedule_date)
print("utc_offset",utc_offset)
#################### Output ##################
utc_offset 3:00:00
```

23.utcnow()

Return current UTC date and time

```
from datetime import datetime as dt

now_ = dt.now()
utc_now = now_.utcnow()
print("utc_now",utc_now)
#################### Output ####################
utc_now 2022-11-23 12:53:41.184227
```

24.weekday()

Returns the day of the week as integer where Monday is 0 and Sunday is 6

```
from datetime import datetime as dt

now_ = dt.now()
week_days = now_.weekday()
print("week_days",week_days)
#################### Output ####################
week_days 2
```