Python Date and time

Python provides the **datetime** module work with real dates and times. In real-world applications, we need to work with the date and time. Python enables us to schedule our Python script to run at a particular timing.

The **datetime** classes are classified in the six main classes.

- **date** - It is a naive ideal date. It consists of the year, month, and day as attributes.
- **time** - It is a perfect time, assuming every day has precisely 24*60*60 seconds. It has hour, minute, second, microsecond, and **tzinfo** as attributes.
- **datetime** - It is a grouping of date and time, along with the attributes year, month, day, hour, minute, second, microsecond, and tzinfo.
- **timedelta -** It represents the difference between two dates, time or datetime instances to microsecond resolution.
- **tzinfo** - It provides time zone information objects.
- **timezone -** It is included in the new version of Python. It is the class that implements the **tzinfo** abstract base class.

How to get the current time?

The localtime() functions of the time module are used to get the current time tuple. Consider the following example.

Example

```
import time;
#returns a time tuple
print(time.localtime(time.time()))
```

Getting formatted time

The time can be formatted by using the **asctime()** function of the time module. It returns the formatted time for the time tuple being passed.

Example

```
import time
  #returns the formatted time
print(time.asctime(time.localtime(time.time())))
```

Python sleep time

The **sleep()** method of time module is used to stop the execution of the script for a given amount of time. The output will be delayed for the number of seconds provided as the float.

Consider the following example.

Example

```
import time
for i in range(0,5):
    print(i)
    #Each element will be printed after 1 second
    time.sleep(1)
```

The datetime Module

The **datetime** module enables us to create the custom date objects, perform various operations on dates like the comparison, etc.

To work with dates as date objects, we have to import **the datetime** module into the python source code.

Consider the following example to get the **datetime** object representation for the current time.

Example

```
import datetime
#returns the current datetime object
print(datetime.datetime.now())
```

Creating date objects

We can create the date objects bypassing the desired date in the datetime constructor for which the date objects are to be created.

Consider the following example.

Example

```
import datetime
#returns the datetime object for the specified date
print(datetime.datetime(2020,04,04))
```

The calendar module

Python provides a calendar object that contains various methods to work with the calendars.

Consider the following example to print the calendar for the last month of 2018.

Example

```
import calendar;
cal = calendar.month(2020,3)
#printing the calendar of December 2018
print(cal)
```

Printing the calendar of whole year

The prcal() method of calendar module is used to print the calendar of the entire year. The year of which the calendar is to be printed must be passed into this method.

Example

```
import calendar
#printing the calendar of the year 2019
s = calendar.prcal(2020)
```

Date Output

When we execute the code from the example above the result will be:

```
2023-08-21 17:44:07.276073
```

The date contains year, month, day, hour, minute, second, and microsecond.

The `datetime` module has many methods to return information about the date object.

Here are a few examples, you will learn more about them later in this chapter:

Example

Return the year and name of weekday:

```
import datetime

x = datetime.datetime.now()

print(x.year)
print(x.strftime("%A"))
```

The strftime() Method

The `datetime` object has a method for formatting date objects into readable strings.

The method is called `strftime()`, and takes one parameter, `format`, to specify the format of the returned string:

Example

Display the name of the month:

```
import datetime

x = datetime.datetime(2018, 6, 1)

print(x.strftime("%B"))

      import datetime
```

```
            x = datetime.datetime.now()

            print(x.strftime("%a"))
```

```
import datetime

x = datetime.datetime.now()

print(x.strftime("%A"))
```

```
            import datetime

            x = datetime.datetime.now()

            print(x.strftime("%w"))
```

```
            import datetime

            x = datetime.datetime.now()

            print(x.strftime("%d"))
```

```
import datetime

x = datetime.datetime.now()

print(x.strftime("%b"))
```

```
import datetime

x = datetime.datetime.now()

print(x.strftime("%B"))
```

```
import datetime

x = datetime.datetime.now()

print(x.strftime("%m"))
```

```
import datetime

x = datetime.datetime.now()

print(x.strftime("%y"))
```

```
import datetime

x = datetime.datetime.now()

print(x.strftime("%Y"))
```