DATE TIME AND CALENDAR MODULE

Aegis
SCHOOL OF DATA SCIENCE

Date and Time

➢What is Tick?

Time intervals are floating-point numbers in units of seconds. Particular instants in time are expressed in seconds since 12:00am, January 1, 1970(epoch).

Example:

```
import time; # This is required to
 include time module.
ticks = time.time()
print "Number of ticks since 12:00am,
 January 1, 1970:", ticks
```

This would produce a result something as follows:

```
Number of ticks since 12:00am, January 1,
1970: 7186862.73399
```

Date and Time

Date arithmetic is easy to do with ticks. However, dates before the epoch cannot be represented in this form.

Dates in the far future also cannot be represented this way - the cutoff point is sometime in 2038 for UNIX and Windows.

- **What is Epoch?**

The Epoch is the point in *time in Python* from which time is measured. It is labelled 12:00AM, Jan 1, 1970. It is the beginning of an era.

- **What is Daylight Savings Time(DST)**

In Agrarian culture, they set the clocks to be one hour faster. This way, they get up an hour before actual sunrise and get an hour extra after work. In northern and southern regions, days are considerably longer in the summer.

Aegis
SCHOOL OF DATA SCIENCE

Time Tuple

Index	Field	Values
0	4-digit year	2008
1	Month	1 to 12
2	Day	1 to 31
3	Hour	0 to 23
4	Minute	0 to 59
5	Second	0 to 61 (60 or 61 are leap-seconds)
6	Day of Week	0 to 6 (0 is Monday)
7	Day of year	1 to 366 (Julian day)
8	Daylight savings	-1, 0, 1, -1 means library determines DST

Time Tuple Structure

Index	Attributes	Values
0	tm_year	2008
1	tm_mon	1 to 12
2	tm_mday	1 to 31
3	tm_hour	0 to 23
4	tm_min	0 to 59
5	tm_sec	0 to 61 (60 or 61 are leap-seconds)
6	tm_wday	0 to 6 (0 is Monday)
7	tm_yday	1 to 366 (Julian day)
8	tm_isdst	-1, 0, 1, -1 means library determines DST

Aegis
SCHOOL OF DATA SCIENCE

Operations on Date and Time

➢ To get number of Ticks

```
import time
print(time.time())
```

➢ To get Local /current time

```
 import  time;
 localtime = time.localtime(time.time())
print ("Local current time :", localtime)
```

➢ To get the Formatted Time

```
import time;
localtime = time.asctime( time.localtime(time.time())
)
print ("Local current time :", localtime)
```

Aegis
SCHOOL OF DATA SCIENCE

Getting FormattedTime

```
import time;
localtime = time.asctime(
 time.localtime(time.time()) )
print "Local current time :", localtime
```

- This would produce following result:

```
Local current time : Sat Oct 08 16:22:08 2011
```

Aegis
SCHOOL OF DATA SCIENCE

Date

A date in Python is not a data type of its own, but we can import a module named datetime to work with dates as date objects.

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

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

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

Aegis
SCHOOL OF DATA SCIENCE

Date

- To Return the year and name of weekday

```
import datetime
x = datetime.datetime.now()
print(x.year)
print(x.strftime("%A"))
```

Creating Date Objects

To create a date, we can use the datetime() class (constructor) of the datetime module.

The datetime() class requires three parameters to create a date: year, month, day.

```
import datetime

x = datetime.datetime(2019, 11, 17)

print(x)
```

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

Parameters supported by strftime method

Directive	Description	Example
%a	Weekday, short version	Wed
%A	Weekday, full version	Wednesday
%w	Weekday as a number 0-6, 0 is Sunday	3
%d	Day of month 01-31	31
%b	Month name, short version	Dec
%B	Month name, full version	December
%m	Month as a number 01-12	12
%y	Year, short version, without century	18
%Y	Year, full version	2018
%H	Hour 00-23	17
%I	Hour 00-12	05
%p	AM/PM	PM
%M	Minute 00-59	41
%S	Second 00-59	08

Parameters supported by strftime method

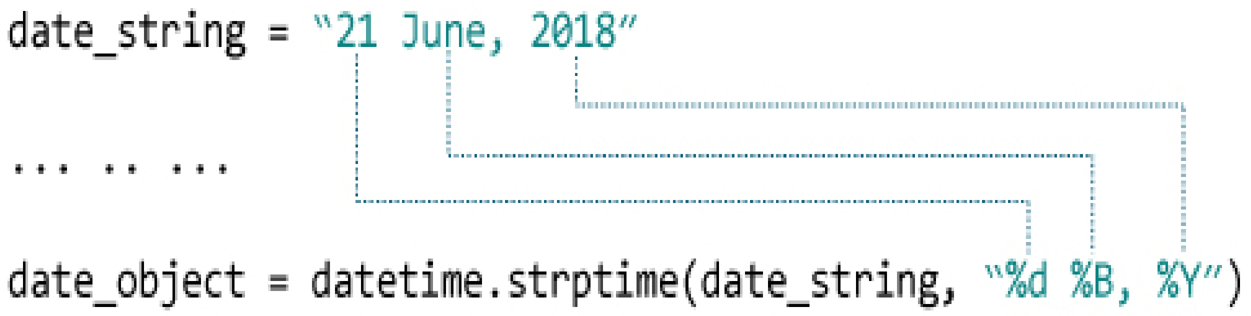
Directive	Description	Example
%f	Microsecond 000000-999999	548513
%z	UTC offset	+0100
%Z	Timezone	CST
%j	Day number of year 001-366	365
%U	Week number of year, Sunday as the first day of week, 00-53	52
%W	Week number of year, Monday as the first day of week, 00-53	52
%c	Local version of date and time	Mon Dec 31 17:41:00 2018
%x	Local version of date	12/31/18
%X	Local version of time	17:41:00
%%	A % character	%

strptime()

The strptime() class method takes two arguments:

- string (that be converted to datetime)
- format code

Based on the string and format code used, the method returns its equivalent datetime object.



```
date_string = "21 June, 2018"

... .. ...

date_object = datetime.strptime(date_string, "%d %B, %Y")
```

Aegis
SCHOOL OF DATA SCIENCE

timedelta()

timedelta() function is present under datetime library which is generally used for calculating differences in dates and also can be used for date manipulations in Python. It is one of the easiest ways to perform date manipulations.

Syntax : *datetime.timedelta(days=0, seconds=0, microseconds=0, milliseconds=0, minutes=0, hours=0, weeks=0)*

timedelta() Example

```
from datetime import datetime, timedelta
timenow = datetime.now()

past_date_before_2yrs = ini_time_for_now - \
                        timedelta(days = 730)
past_date_before_2hours = ini_time_for_now - \
                          timedelta(hours = 2)
```

The time Module

SN	Function with Description
1	time.altzone
	The offset of the local DST timezone, in seconds west of UTC, if one is defined. This is negative if the local DST timezone is east of UTC (as in Western Europe, including the UK). Only use this if daylight is nonzero.
2	time.asctime([tupletime])
	Accepts a time-tuple and returns a readable 24-character string such as 'Tue Dec 11 18:07:14 2008'.
3	time.clock( )
	Returns the current CPU time as a floating-point number of seconds. To measure computational costs of different approaches, the value of time.clock is more useful than that of time.time().
	https://docs.python.org/3/library/time.html Refer for time module

The time Module

SN	Function with Description
4	time.ctime([secs])
	Like asctime(localtime(secs)) and without arguments is like asctime( )
5	time.gmtime([secs])
	Accepts an instant expressed in seconds since the epoch and returns a time-tuple t with the UTC time. Note : t.tm_isdst is always 0
6	time.localtime([secs])
	Accepts an instant expressed in seconds since the epoch and returns a time-tuple t with the local time (t.tm_isdst is 0 or 1, depending on whether DST applies to instant secs by local rules).

Aegis
SCHOOL OF DATA SCIENCE

The time Module

SN	Function with Description
7	time.mktime(tupletime)
	Accepts an instant expressed as a time-tuple in local time and returns a floating-point value with the instant expressed in seconds since the epoch.
8	time.sleep(secs)
	Suspends the calling thread for secs seconds.
9	time.strftime(fmt[,tupletime])
	Accepts an instant expressed as a time-tuple in local time and returns a string representing the instant as specified by string fmt.

Aegis
SCHOOL OF DATA SCIENCE

The time Module

SN	Function with Description
10	time.strptime(str,fmt='%a %b %d %H:%M:%S %Y')
	Parses str according to format string fmt and returns the instant in time-tuple format.
11	time.time( )
	Returns the current time instant, a floating-point number of seconds since the epoch.
12	time.tzset()
	Resets the time conversion rules used by the library routines. The environment variable TZ specifies how this is done.

Aegis
SCHOOL OF DATA SCIENCE

The time Module

There are following two important attributes available with time module:

SN	Attribute with Description
1	time.timezone
	Attribute time.timezone is the offset in seconds of the local time zone (without DST) from UTC (>0 in the Americas; <=0 in most of Europe, Asia, Africa).
2	time.tzname
	Attribute time.tzname is a pair of locale-dependent strings, which are the names of the local time zone without and with DST, respectively.

Calendar

In order to get the calendar of the year calendar module is used

```
import calendar
```

- To get the calendar of a specific month

```
import  calendar
cal = calendar.month(2008, 1)
Print(cal)
```

The *calendar* Module

SN	Function with Description
1	calendar.calendar(year,w=2,l=1,c=6)
	Returns a multiline string with a calendar for year year formatted into three columns separated by c spaces. w is the width in characters of each date; each line has length 21*w+18+2*c. l is the number of lines for each week.
2	calendar.firstweekday( )
	Returns the current setting for the weekday that starts each week. By default, when calendar is first imported, this is 0, meaning Monday.
3	calendar.isleap(year)
	Returns True if year is a leap year; otherwise, False.

Aegis
SCHOOL OF DATA SCIENCE

The *calendar* Module

SN	Function with Description
4	**calendar.leapdays(y1,y2)**
	Returns the total number of leap days in the years within range(y1,y2).
5	**calendar.month(year,month,w=2,l=1)**
	Returns a multiline string with a calendar for month month of year year, one line per week plus two header lines. w is the width in characters of each date; each line has length 7*w+6. l is the number of lines for each week.
6	**calendar.monthcalendar(year,month)**
	Returns a list of lists of ints. Each sublist denotes a week. Days outside month month of year year are set to 0; days within the month are set to their day-of-month, 1 and up.

The *calendar* Module

SN	Function with Description
7	**calendar.monthrange(year,month)**
	Returns two integers. The first one is the code of the weekday for the first day of the month month in year year; the second one is the number of days in the month. Weekday codes are 0 (Monday) to 6 (Sunday); month numbers are 1 to 12.
8	**calendar.prcal(year,w=2,l=1,c=6)**
	Like print calendar.calendar(year,w,l,c).
9	**calendar.prmonth(year,month,w=2,l=1)**
	Like print calendar.month(year,month,w,l).

SCHOOL OF DATA SCIENCE

The *calendar* Module

SN	Function with Description
10	**calendar.setfirstweekday(weekday)**
	Sets the first day of each week to weekday code weekday. Weekday codes are 0 (Monday) to 6 (Sunday).
11	**calendar.timegm(tupletime)**
	The inverse of time.gmtime: accepts a time instant in time-tuple form and returns the same instant as a floating-point number of seconds since the epoch.
12	**calendar.weekday(year,month,day)**
	Returns the weekday code for the given date. Weekday codes are 0 (Monday) to 6 (Sunday); month numbers are 1 (January) to 12 (December).

SCHOOL OF DATA SCIENCE

TASK

- Download the Task sheet from the Portal

Aegis
SCHOOL OF DATA SCIENCE