```
#!/usr/bin/python
import time


print "time.time(): %f " %  time.time()
print time.localtime( time.time() )
print time.asctime( time.localtime(time.time()) )
```

When we run above program, it produces following result:

```
time.time(): 1234892919.655932
(2009, 2, 17, 10, 48, 39, 1, 48, 0)
Tue Feb 17 10:48:39 2009
```

82. time.tzset()

Description

The method **tzset()** resets the time conversion rules used by the library routines. The environment variable TZ specifies how this is done.

The standard format of the TZ environment variable is (whitespace added for clarity):

```
std offset [dst [offset [,start[/time], end[/time]]]]
```

- **std and dst:** Three or more alphanumerics giving the timezone abbreviations. These will be propagated into time.tzname.
- **offset:** The offset has the form: .hh[:mm[:ss]]. This indicates the value added the local time to arrive at UTC. If preceded by a '-', the timezone is east of the Prime Meridian; otherwise, it is west. If no offset follows *dst*, summer time is assumed to be one hour ahead of standard time.
- **start[/time], end[/time]:** Indicates when to change to and back from DST. The format of the start and end dates are one of the following:
    - **Jn:** The Julian day n (1 <= n <= 365). Leap days are not counted, so in all years February 28 is day 59 and March 1 is day 60.
    - **n:** The zero-based Julian day (0 <= n <= 365). Leap days are counted, and it is possible to refer to February 29.