**Date and Time**

Python is very useful in case of Date and Time. We can easily retrieve current date and time using Python.

**Retrieve Time**

To retrieve current time a predefined function localtime() is used. localtime() receives a parameter time.time() . Here,

time is a module,

time() is a function that returns the current system time in number of ticks since 12:00 am , January 1,1970. It is known as epoch.

Tick is simply a floating point number in seconds since epoch.

**eg:**

1. import time;
2. localtime = time.localtime(time.time())
3. print "Current Time is :", localtime

**Output:**

1. >>>
2. Current Time is :time.struct\_time(tm\_year=2014, tm\_mon=6, tm\_mday=18, tm\_hour=12,
3. tm\_min=35, tm\_sec=44, tm\_wday=2, tm\_yday=169, tm\_isdst=0)
4. >>>

**Explanation:**

The time returned is a time structure which includes 9 attributes. These are summoned in the table given below.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| tm\_year | Returns the current year |
| tm\_mon | Returns the current month |
| tm\_mday | Returns the current month day |
| tm\_hour | Returns the current hour. |
| tm\_min | Returns the current minute |
| tm\_sec | Returns current seconds |
| tm\_wday | Returns the week day |
| tm\_yday | Returns the year day. |
| tm\_isdst | It returns -1,0 or 1. |

**Formatted Time**

Python also support formatted time. Proceed as follows:

1. Pass the time structure in a predefined function asctime(). It is a function defined in time module.
2. It returns a formatted time which includes Day ,month, date, time and year.
3. Print the formatted time.

**eg:**

1. import time;
3. localtime = time.asctime( time.localtime(time.time()) )
4. print "Formatted time :", localtime

**Output:**

1. >>>
2. Formatted time : Sun Jun 22 18:54:20 2014
3. >>>

**time module:**

There are many built in functions defined in time module which are used to work with time.

|  |  |
| --- | --- |
| **Methods** | **Description** |
| time() | Returns floating point value in seconds since epoch i.e.,12:00am, January 1, 1970 |
| asctime(time) | It takes the tuple returned by localtime() as parameter. It returns a 24 character string. |
| sleep(time) | The execution will be stopped for the given interval of time. |
| strptime(String,format) | It returns an tuple with 9 time attributes. It receives an String of date and a format. |
| gtime()/gtime(sec) | It returns struct\_time which contains 9 time attributes. In case seconds are not specified it takes current second from epoch. |
| mktime() | Returns second in floating point since epoch. |
| strftime(format)/strftime(format,time) | Returns time in particular format. If time is not given, current time in seconds is fetched. |

**time()**

**eg:**

1. import time
2. printtime.time()

**Output:**

1. >>>
2. 1403700740.39
3. >>>

**asctime(time)**

1. import time
2. t = time.localtime()
3. printtime.asctime(t)

**Output:**

1. >>>
2. Wed Jun 25 18:30:25 2014
3. >>>

**sleep(time)**

**Eg:**

1. import time
3. localtime = time.asctime( time.localtime(time.time()) )
4. printlocaltime
5. time.sleep( 10 )
6. localtime = time.asctime( time.localtime(time.time()) )
7. printlocaltime

**Output:**

1. >>>
2. Wed Jun 25 18:15:30 2014
3. Wed Jun 25 18:15:40 2014
4. >>>

**strptime(String str,format f)**

**Eg:**

1. import time
3. timerequired = time.strptime("26 Jun 14", "%d %b %y")
4. printtimerequired

**Output:**

1. >>>
2. time.struct\_time(tm\_year=2014, tm\_mon=6, tm\_mday=26, tm\_hour=0, tm\_min=0,
3. tm\_sec=0, tm\_wday=3, tm\_yday=177, tm\_isdst=-1)
4. >>>

**Explanation:**

The strptime() takes a String and format as argument. The format refers to String passed as an argument. "%a %b %d %H:%M:%S %Y" are the default directives. There are many other directives which can be used. In the given example we have used three directives: %d%b%y which specifies day of the month, month in abbreviated form and year without century respectively. Some of them are given as:

|  |  |
| --- | --- |
| %a | weekday name. |
| %b | month name |
| %c | date and time |
| %e | day of a month |
| %m | month in digit. |
| %n | new line character. |
| %S | second |
| %t | tab character |

etc...

**gtime()**

**Eg:**

1. import time
2. printtime.gmtime()

**Output:**

1. >>>
2. time.struct\_time(tm\_year=2014, tm\_mon=6, tm\_mday=28, tm\_hour=9, tm\_min=38, tm\_sec=0,
3. tm\_wday=5, tm\_yday=179, tm\_isdst=0)
4. >>>

**mktime()**

**Eg:**

1. import time
2. t = (2014, 2, 17, 17, 3, 38, 1, 48, 0)
3. second = time.mktime( t )
4. print second

**Output:**

1. >>>
2. 1392636818.0
3. >>>

**strftime()**

**Eg:**

1. import time
2. t = (2014, 6, 26, 17, 3, 38, 1, 48, 0)
3. t = time.mktime(t)
4. printtime.strftime("%b %d %Y %H:%M:%S", time.gmtime(t))

**Output:**

1. >>>
2. Jun 26 2014 11:33:38
3. >>>

**Calendar**

Python provides calendar module to display Calendar.

**Eg:**

1. import calendar
2. print "Current month is:"
3. cal = calendar.month(2014, 6)
4. printcal

**Output:**

1. >>>
2. Current month is:
3. June 2014
4. Mo TuWe ThFr Sa Su
5. 1
6. 2  3  4  5  6  7  8
7. 9 10 11 12 1314 15
8. 16 1718 19 2021 22
9. 2324 25 26 27 28 29
10. 30
11. >>>

**Calendar module:**

Python provides calendar module which provides many functions and methods to work on calendar. A list of methods and function used is given below:

|  |  |
| --- | --- |
| **Methods** | **Description** |
| prcal(year) | Prints the whole calendar of the year. |
| firstweekday() | Returns the first week day. It is by default 0 which specifies Monday |
| isleap(year) | Returns a Boolean value i.e., true or false. True in case given year is leap else false. |
| monthcalendar(year,month) | Returns the given month with each week as one list. |
| leapdays(year1,year2) | Return number of leap days from year1 to year2 |
| prmonth(year,month) | Print the given month of the given year |

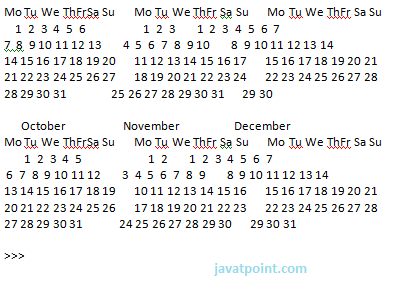
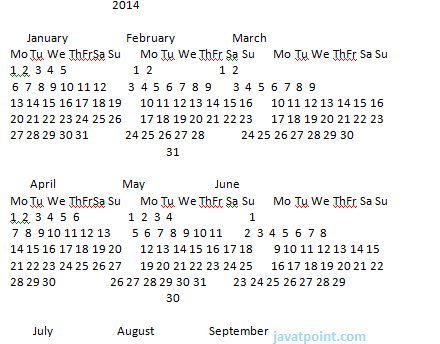
**prcal(year)**

**Eg:**

1. import calendar
2. calendar.prcal(2014)

**Output:**

1. >>> ================================ RESTART ================================
2. >>>



**firstweekday()**

**Eg:**

1. import calendar
2. printcalendar.firstweekday()

**Output:**

1. >>>
2. 0
3. >>>

**isleap(year)**

**Eg:**

1. import calendar
2. printcalendar.isleap(2000)

**Output:**

1. >>>
2. True
3. >>>

**monthcalendar(year,month)**

**Eg:**

1. import calendar
2. printcalendar.monthcalendar(2014,6)

**Output:**

1. >>>
2. [[0, 0, 0, 0, 0, 0, 1], [2, 3, 4, 5, 6, 7, 8], [9, 10, 11, 12, 13, 14, 15],
3. [16, 17, 18, 19, 20, 21, 22],
4. [23, 24, 25, 26, 27, 28, 29], [30, 0, 0, 0, 0, 0, 0]]
5. >>>

**prmonth(year,month)**

**Eg:**

1. import calendar
2. printcalendar.prmonth(2014,6)

**Output:**

1. >>>
2. June 2014
3. Mo Tu We ThFrSa Su
4. 1
5. 2  3  4  5  6  7  8
6. 9 10 11 12 13 14 15
7. 16 17 18 19 20 21 22
8. 23 24 25 26 27 28 29
9. 30
10. None
11. >>>