PYTHON TIME STRFTIME METHOD

Description

The method **strftime** converts a tuple or struct_time representing a time as returned by gmtime or localtime to a string as specified by the format argument.

If t is not provided, the current time as returned by localtime is used. format must be a string. An exception ValueError is raised if any field in t is outside of the allowed range.

Syntax

Following is the syntax for **strftime** method:

```
time.strftime(format[, t])
```

Parameters

- **t** -- This is the time in number of seconds to be formatted.

- **format** -- This is the directive which would be used to format given time. The following directives can be embedded in the format string:

Directive

- %a - abbreviated weekday name

- %A - full weekday name

- %b - abbreviated month name

- %B - full month name

- %c - preferred date and time representation

- %C - century number $the year divided by 100, range 00 to 99$

- %d - day of the month $01 to 31$

- %D - same as %m/%d/%y

- %e - day of the month $1 to 31$

- %g - like %G, but without the century

- %G - 4-digit year corresponding to the ISO week number see.

- %h - same as %b

- %H - hour, using a 24-hour clock $00 to 23$

- %I - hour, using a 12-hour clock $01 to 12$

- %j - day of the year $001 to 366$

- %m - month $01 to 12$

- %M - minute

- %n - newline character

- %p - either am or pm according to the given time value

- %r - time in a.m. and p.m. notation

- %R - time in 24 hour notation

- %S - second

- %t - tab character

- %T - current time, equal to %H:%M:%S

- %u - weekday as a number $1 to 7$, Monday=1. Warning: In Sun Solaris Sunday=1

- %U - week number of the current year, starting with the first Sunday as the first day of the first week

- %V - The ISO 8601 week number of the current year $01 to 53$, where week 1 is the first week that has at least 4 days in the current year, and with Monday as the first day of the week

- %W - week number of the current year, starting with the first Monday as the first day of the first week

- %w - day of the week as a decimal, Sunday=0

- %x - preferred date representation without the time

- %X - preferred time representation without the date

- %y - year without a century $range 00 to 99$

- %Y - year including the century

- %Z or %z - time zone or name or abbreviation

- %% - a literal % character

Return Value

This method does not return any value.

Example

The following example shows the usage of strftime method.

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

t = (2009, 2, 17, 17, 3, 38, 1, 48, 0)
t = time.mktime(t)
print time.strftime("%b %d %Y %H:%M:%S", time.gmtime(t))
```

When we run above program, it produces following result:

Feb 18 2009 00:03:38