:mod:`time` --- Time access and conversions

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 1); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 4)

Unknown directive type "module".

```
.. module:: time
   :synopsis: Time access and conversions.
```

This module provides various time-related functions. For related functionality, see also the :mod:`datetime` and :mod:`calendar` modules.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 9); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 9); *backlink*

Unknown interpreted text role "mod".

Although this module is always available, not all functions are available on all platforms. Most of the functions defined in this module call platform C library functions with the same name. It may sometimes be helpful to consult the platform documentation, because the semantics of these functions varies among platforms.

An explanation of some terminology and conventions is in order.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 22)

Unknown directive type "index".

```
.. index:: single: epoch
```

- The :dfn:`epoch` is the point where the time starts, the return value of `time.gmtime(0)`. It is January 1, 1970, 00:00:00 (UTC) on all platforms.

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 24); *backlink*

  Unknown interpreted text role "dfn".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 29)

Unknown directive type "index".

```
.. index:: seconds since the epoch
```

- The term :dfn:`seconds since the epoch` refers to the total number of elapsed seconds since the epoch, typically excluding leap seconds. Leap seconds are excluded from this total on all POSIX-compliant platforms.

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 31); *backlink*

  Unknown interpreted text role "dfn".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 36)

Unknown directive type "index".

```
.. index:: single: Year 2038
```

- The functions in this module may not handle dates and times before the epoch or far in the future. The cut-off point in the future is determined by the C library; for 32-bit systems, it is typically in 2038.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 42)

Unknown directive type "index".

```
.. index::
   single: 2-digit years
```

- Function :func:`strptime` can parse 2-digit years when given `%y` format code. When 2-digit years are parsed, they are converted according to the POSIX and ISO C standards: values 69--99 are mapped to 1969--1999, and values 0--68 are mapped to 2000--2068.

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]time.rst`, line 45); ***backlink***

  Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]time.rst`, line 50)

Unknown directive type "index".

```
.. index::
   single: UTC
   single: Coordinated Universal Time
   single: Greenwich Mean Time
```

- UTC is Coordinated Universal Time (formerly known as Greenwich Mean Time, or GMT). The acronym UTC is not a mistake but a compromise between English and French.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]time.rst`, line 59)

Unknown directive type "index".

```
.. index:: single: Daylight Saving Time
```

- DST is Daylight Saving Time, an adjustment of the timezone by (usually) one hour during part of the year. DST rules are magic (determined by local law) and can change from year to year. The C library has a table containing the local rules (often it is read from a system file for flexibility) and is the only source of True Wisdom in this respect.
- The precision of the various real-time functions may be less than suggested by the units in which their value or argument is expressed. E.g. on most Unix systems, the clock "ticks" only 50 or 100 times a second.
- On the other hand, the precision of :func:`.time` and :func:`sleep` is better than their Unix equivalents: times are expressed as floating point numbers, :func:`.time` returns the most accurate time available (using Unix :c:func:`gettimeofday` where available), and :func:`sleep` will accept a time with a nonzero fraction (Unix :c:func:`select` is used to implement this, where available).

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]time.rst`, line 71); ***backlink***

  Unknown interpreted text role "func".

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]time.rst`, line 71); ***backlink***

  Unknown interpreted text role "func".

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]time.rst`, line 71); ***backlink***

  Unknown interpreted text role "func".

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]time.rst`, line 71); ***backlink***

  Unknown interpreted text role "c:func".

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]time.rst`, line 71); ***backlink***

  Unknown interpreted text role "func".

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]time.rst`, line 71); ***backlink***

  Unknown interpreted text role "c:func".

- The time value as returned by :func:`gmtime`, :func:`localtime`, and :func:`strptime`, and accepted by :func:`asctime`, :func:`mktime` and :func:`strftime`, is a sequence of 9 integers. The return values of :func:`gmtime`, :func:`localtime`, and :func:`strptime` also offer attribute names for individual fields.

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]time.rst`, line 78); ***backlink***

  Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 78); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 78); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 78); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 78); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 78); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 78); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 78); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 78); *backlink*

Unknown interpreted text role "func".

See :class:`struct_time` for a description of these objects.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 84); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 86)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.3
   The :class:`struct_time` type was extended to provide the :attr:`tm_gmtoff`
   and :attr:`tm_zone` attributes when platform supports corresponding
   ``struct tm`` members.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 91)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.6
   The :class:`struct_time` attributes :attr:`tm_gmtoff` and :attr:`tm_zone`
   are now available on all platforms.
```

- Use the following functions to convert between time representations:

From	To	Use

From	To	Use
seconds since the epoch	:class:`struct_time` in UTC **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, **line 101**); ***backlink*** Unknown interpreted text role "class".	:func:`gmtime` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, **line 101**); ***backlink*** Unknown interpreted text role "func".
seconds since the epoch	:class:`struct_time` in local time **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, **line 104**); ***backlink*** Unknown interpreted text role "class".	:func:`localtime` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, **line 104**); ***backlink*** Unknown interpreted text role "func".
:class:`struct_time` in UTC **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, **line 107**); ***backlink*** Unknown interpreted text role "class".	seconds since the epoch	:func:`calendar.timegm` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, **line 107**); ***backlink*** Unknown interpreted text role "func".
:class:`struct_time` in local time **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, **line 110**); ***backlink*** Unknown interpreted text role "class".	seconds since the epoch	:func:`mktime` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, **line 110**); ***backlink*** Unknown interpreted text role "func".

Functions

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, **line 119**)

Unknown directive type "function".

```
.. function:: asctime([t])

   Convert a tuple or :class:`struct_time` representing a time as returned by
   :func:`gmtime` or :func:`localtime` to a string of the following
   form: ``'Sun Jun 20 23:21:05 1993'``. The day field is two characters long
   and is space padded if the day is a single digit,
   e.g.: ``'Wed Jun  9 04:26:40 1993'``.

   If *t* is not provided, the current time as returned by :func:`localtime`
   is used. Locale information is not used by :func:`asctime`.
   .. note::

      Unlike the C function of the same name, :func:`asctime` does not add a
      trailing newline.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 135)

Unknown directive type "function".

```
.. function:: pthread_getcpuclockid(thread_id)

   Return the *clk_id* of the thread-specific CPU-time clock for the specified *thread_id*.

   Use :func:`threading.get_ident` or the :attr:`~threading.Thread.ident`
   attribute of :class:`threading.Thread` objects to get a suitable value
   for *thread_id*.

   .. warning::
      Passing an invalid or expired *thread_id* may result in
      undefined behavior, such as segmentation fault.

   .. availability:: Unix (see the man page for :manpage:`pthread_getcpuclockid(3)` for
      further information).

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 152)

Unknown directive type "function".

```
.. function:: clock_getres(clk_id)

   Return the resolution (precision) of the specified clock *clk_id*.  Refer to
   :ref:`time-clock-id-constants` for a list of accepted values for *clk_id*.

   .. availability:: Unix.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 162)

Unknown directive type "function".

```
.. function:: clock_gettime(clk_id) -> float

   Return the time of the specified clock *clk_id*.  Refer to
   :ref:`time-clock-id-constants` for a list of accepted values for *clk_id*.

   Use :func:`clock_gettime_ns` to avoid the precision loss caused by the
   :class:`float` type.

   .. availability:: Unix.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 175)

Unknown directive type "function".

```
.. function:: clock_gettime_ns(clk_id) -> int

   Similar to :func:`clock_gettime` but return time as nanoseconds.

   .. availability:: Unix.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 184)

Unknown directive type "function".

```
.. function:: clock_settime(clk_id, time: float)

   Set the time of the specified clock *clk_id*.  Currently,
   :data:`CLOCK_REALTIME` is the only accepted value for *clk_id*.

   Use :func:`clock_settime_ns` to avoid the precision loss caused by the
   :class:`float` type.

   .. availability:: Unix.

   .. versionadded:: 3.3
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 197)

Unknown directive type "function".

```
.. function:: clock_settime_ns(clk_id, time: int)

   Similar to :func:`clock_settime` but set time with nanoseconds.

   .. availability:: Unix.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 206)

Unknown directive type "function".

```
.. function:: ctime([secs])

   Convert a time expressed in seconds since the epoch_ to a string of a form:
   ``'Sun Jun 20 23:21:05 1993'`` representing local time. The day field
   is two characters long and is space padded if the day is a single digit,
   e.g.: ``'Wed Jun  9 04:26:40 1993'``.

   If *secs* is not provided or :const:`None`, the current time as
   returned by :func:`.time` is used. ``ctime(secs)`` is equivalent to
   ``asctime(localtime(secs))``. Locale information is not used by
   :func:`ctime`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 219)

Unknown directive type "function".

```
.. function:: get_clock_info(name)

   Get information on the specified clock as a namespace object.
   Supported clock names and the corresponding functions to read their value
   are:

   * ``'monotonic'``: :func:`time.monotonic`
   * ``'perf_counter'``: :func:`time.perf_counter`
   * ``'process_time'``: :func:`time.process_time`
   * ``'thread_time'``: :func:`time.thread_time`
   * ``'time'``: :func:`time.time`

   The result has the following attributes:

   - *adjustable*: ``True`` if the clock can be changed automatically (e.g. by
     a NTP daemon) or manually by the system administrator, ``False`` otherwise
   - *implementation*: The name of the underlying C function used to get
     the clock value.  Refer to :ref:`time-clock-id-constants` for possible values.
   - *monotonic*: ``True`` if the clock cannot go backward,
     ``False`` otherwise
   - *resolution*: The resolution of the clock in seconds (:class:`float`)

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 244)

Unknown directive type "function".

```
.. function:: gmtime([secs])

   Convert a time expressed in seconds since the epoch_ to a :class:`struct_time` in
   UTC in which the dst flag is always zero.  If *secs* is not provided or
   :const:`None`, the current time as returned by :func:`.time` is used.  Fractions
   of a second are ignored.  See above for a description of the
   :class:`struct_time` object. See :func:`calendar.timegm` for the inverse of this
   function.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 254)

Unknown directive type "function".

```
.. function:: localtime([secs])

   Like :func:`gmtime` but converts to local time.  If *secs* is not provided or
   :const:`None`, the current time as returned by :func:`.time` is used.  The dst
   flag is set to ``1`` when DST applies to the given time.

   :func:`localtime` may raise :exc:`OverflowError`, if the timestamp is
   outside the range of values supported by the platform C :c:func:`localtime`
   or :c:func:`gmtime` functions, and :exc:`OSError` on :c:func:`localtime` or
   :c:func:`gmtime` failure. It's common for this to be restricted to years
   between 1970 and 2038.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 267)

Unknown directive type "function".

```
.. function:: mktime(t)

   This is the inverse function of :func:`localtime`.  Its argument is the
   :class:`struct_time` or full 9-tuple (since the dst flag is needed; use ``-1``
   as the dst flag if it is unknown) which expresses the time in *local* time, not
   UTC.  It returns a floating point number, for compatibility with :func:`.time`.
   If the input value cannot be represented as a valid time, either
   :exc:`OverflowError` or :exc:`ValueError` will be raised (which depends on
   whether the invalid value is caught by Python or the underlying C libraries).
   The earliest date for which it can generate a time is platform-dependent.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 279)

Unknown directive type "function".

```
.. function:: monotonic() -> float

   Return the value (in fractional seconds) of a monotonic clock, i.e. a clock
   that cannot go backwards.  The clock is not affected by system clock updates.
   The reference point of the returned value is undefined, so that only the
   difference between the results of two calls is valid.

   Use :func:`monotonic_ns` to avoid the precision loss caused by the
   :class:`float` type.

   .. versionadded:: 3.3

   .. versionchanged:: 3.5
      The function is now always available and always system-wide.

   .. versionchanged:: 3.10
      On macOS, the function is now system-wide.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 298)

Unknown directive type "function".

```
.. function:: monotonic_ns() -> int

   Similar to :func:`monotonic`, but return time as nanoseconds.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 304)

Unknown directive type "function".

```
.. function:: perf_counter() -> float

   .. index::
      single: benchmarking

   Return the value (in fractional seconds) of a performance counter, i.e. a
   clock with the highest available resolution to measure a short duration.  It
   does include time elapsed during sleep and is system-wide.  The reference
   point of the returned value is undefined, so that only the difference between
   the results of two calls is valid.

   Use :func:`perf_counter_ns` to avoid the precision loss caused by the
   :class:`float` type.

   .. versionadded:: 3.3

   .. versionchanged:: 3.10
      On Windows, the function is now system-wide.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 323)

Unknown directive type "function".

```
.. function:: perf_counter_ns() -> int

   Similar to :func:`perf_counter`, but return time as nanoseconds.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 330)

Unknown directive type "function".

```
.. function:: process_time() -> float

   .. index::
      single: CPU time
      single: processor time
      single: benchmarking
   Return the value (in fractional seconds) of the sum of the system and user
   CPU time of the current process.  It does not include time elapsed during
   sleep.  It is process-wide by definition.  The reference point of the
   returned value is undefined, so that only the difference between the results
   of two calls is valid.

   Use :func:`process_time_ns` to avoid the precision loss caused by the
   :class:`float` type.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 348)

Unknown directive type "function".

```
.. function:: process_time_ns() -> int

   Similar to :func:`process_time` but return time as nanoseconds.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 354)

Unknown directive type "function".

```
.. function:: sleep(secs)

   Suspend execution of the calling thread for the given number of seconds.
   The argument may be a floating point number to indicate a more precise sleep
   time.

   If the sleep is interrupted by a signal and no exception is raised by the
   signal handler, the sleep is restarted with a recomputed timeout.

   The suspension time may be longer than requested by an arbitrary amount,
   because of the scheduling of other activity in the system.

   On Windows, if *secs* is zero, the thread relinquishes the remainder of its
   time slice to any other thread that is ready to run. If there are no other
   threads ready to run, the function returns immediately, and the thread
   continues execution.  On Windows 8.1 and newer the implementation uses
   a `high-resolution timer
   <https://docs.microsoft.com/en-us/windows-hardware/drivers/kernel/high-resolution-timers>`_
   which provides resolution of 100 nanoseconds. If *secs* is zero, ``Sleep(0)`` is used.

   Unix implementation:

   * Use ``clock_nanosleep()`` if available (resolution: 1 nanosecond);
   * Or use ``nanosleep()`` if available (resolution: 1 nanosecond);
   * Or use ``select()`` (resolution: 1 microsecond).

   .. versionchanged:: 3.11
      On Unix, the ``clock_nanosleep()`` and ``nanosleep()`` functions are now
      used if available. On Windows, a waitable timer is now used.

   .. versionchanged:: 3.5
      The function now sleeps at least *secs* even if the sleep is interrupted
      by a signal, except if the signal handler raises an exception (see
      :pep:`475` for the rationale).
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 390)

Unknown directive type "index".

```
.. index::
   single: % (percent); datetime format
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 393)

Unknown directive type "function".

```
.. function:: strftime(format[, t])

   Convert a tuple or :class:`struct_time` representing a time as returned by
   :func:`gmtime` or :func:`localtime` to a string as specified by the *format*
   argument.  If *t* is not provided, the current time as returned by
   :func:`localtime` is used.  *format* must be a string.  :exc:`ValueError` is
   raised if any field in *t* is outside of the allowed range.

   0 is a legal argument for any position in the time tuple; if it is normally
   illegal the value is forced to a correct one.

   The following directives can be embedded in the *format* string. They are shown
   without the optional field width and precision specification, and are replaced
   by the indicated characters in the :func:`strftime` result:

   +-----------+------------------------------------------------+-------+
   | Directive | Meaning                                        | Notes |
   +===========+================================================+=======+
   | ``%a``    | Locale's abbreviated weekday name.             |       |
   |           |                                                |       |
   +-----------+------------------------------------------------+-------+
   | ``%A``    | Locale's full weekday name.                    |       |
   +-----------+------------------------------------------------+-------+
   | ``%b``    | Locale's abbreviated month name.               |       |
   |           |                                                |       |
   +-----------+------------------------------------------------+-------+
   | ``%B``    | Locale's full month name.                      |       |
   +-----------+------------------------------------------------+-------+
   | ``%c``    | Locale's appropriate date and time             |       |
   |           | representation.                                |       |
   +-----------+------------------------------------------------+-------+
   | ``%d``    | Day of the month as a decimal number [01,31].  |       |
   |           |                                                |       |
   +-----------+------------------------------------------------+-------+
   | ``%H``    | Hour (24-hour clock) as a decimal number       |       |
   |           | [00,23].                                       |       |
   +-----------+------------------------------------------------+-------+
   | ``%I``    | Hour (12-hour clock) as a decimal number       |       |
   |           | [01,12].                                       |       |
   +-----------+------------------------------------------------+-------+
   | ``%j``    | Day of the year as a decimal number [001,366]. |       |
   |           |                                                |       |
   +-----------+------------------------------------------------+-------+
   | ``%m``    | Month as a decimal number [01,12].             |       |
   |           |                                                |       |
   +-----------+------------------------------------------------+-------+
   | ``%M``    | Minute as a decimal number [00,59].            |       |
   |           |                                                |       |
   +-----------+------------------------------------------------+-------+
   | ``%p``    | Locale's equivalent of either AM or PM.        | \(1)  |
   |           |                                                |       |
   +-----------+------------------------------------------------+-------+
   | ``%S``    | Second as a decimal number [00,61].            | \(2)  |
   |           |                                                |       |
   +-----------+------------------------------------------------+-------+
   | ``%U``    | Week number of the year (Sunday as the first   | \(3)  |
   |           | day of the week) as a decimal number [00,53].  |       |
   |           | All days in a new year preceding the first     |       |
   |           | Sunday are considered to be in week 0.         |       |
   |           |                                                |       |
   |           |                                                |       |
   |           |                                                |       |
   +-----------+------------------------------------------------+-------+
   | ``%w``    | Weekday as a decimal number [0(Sunday),6].     |       |
   |           |                                                |       |
   +-----------+------------------------------------------------+-------+
   | ``%W``    | Week number of the year (Monday as the first   | \(3)  |
   |           | day of the week) as a decimal number [00,53].  |       |
   |           | All days in a new year preceding the first     |       |
   |           | Monday are considered to be in week 0.         |       |
   |           |                                                |       |
   |           |                                                |       |
   |           |                                                |       |
   +-----------+------------------------------------------------+-------+
   | ``%x``    | Locale's appropriate date representation.      |       |
   |           |                                                |       |
   +-----------+------------------------------------------------+-------+
   | ``%X``    | Locale's appropriate time representation.      |       |
   |           |                                                |       |
   +-----------+------------------------------------------------+-------+
   | ``%y``    | Year without century as a decimal number       |       |
   |           | [00,99].                                       |       |
   +-----------+------------------------------------------------+-------+
   | ``%Y``    | Year with century as a decimal number.         |       |
   |           |                                                |       |
   +-----------+------------------------------------------------+-------+
   | ``%z``    | Time zone offset indicating a positive or      |       |
   |           | negative time difference from UTC/GMT of the   |       |
   |           | form +HHMM or -HHMM, where H represents decimal|       |
   |           | hour digits and M represents decimal minute    |       |
   |           | digits [-23:59, +23:59]. [1]_                  |       |
   +-----------+------------------------------------------------+-------+
   | ``%Z``    | Time zone name (no characters if no time zone  |       |
   |           | exists). Deprecated. [1]_                      |       |
   +-----------+------------------------------------------------+-------+
   | ``%%``    | A literal ``'%'`` character.                   |       |
   +-----------+------------------------------------------------+-------+

   Notes:

   (1)
      When used with the :func:`strptime` function, the ``%p`` directive only affects
      the output hour field if the ``%I`` directive is used to parse the hour.

   (2)
      The range really is ``0`` to ``61``; value ``60`` is valid in
      timestamps representing `leap seconds`_ and value ``61`` is supported
      for historical reasons.

   (3)
      When used with the :func:`strptime` function, ``%U`` and ``%W`` are only used in
      calculations when the day of the week and the year are specified.

   Here is an example, a format for dates compatible with that specified  in the
   :rfc:`2822` Internet email standard.  [1]_ ::

      >>> from time import gmtime, strftime
      >>> strftime("%a, %d %b %Y %H:%M:%S +0000", gmtime())
      'Thu, 28 Jun 2001 14:17:15 +0000'

   Additional directives may be supported on certain platforms, but only the
   ones listed here have a meaning standardized by ANSI C.  To see the full set
   of format codes supported on your platform, consult the :manpage:`strftime(3)`
   documentation.

   On some platforms, an optional field width and precision specification can
   immediately follow the initial ``'%'`` of a directive in the following order;
```

```
   this is also not portable. The field width is normally 2 except for ``%j`` where
   it is 3.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 524)

Unknown directive type "index".

```
.. index::
   single: % (percent); datetime format
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 527)

Unknown directive type "function".

```
.. function:: strptime(string[, format])

   Parse a string representing a time according to a format.  The return value
   is a :class:`struct_time` as returned by :func:`gmtime` or
   :func:`localtime`.

   The *format* parameter uses the same directives as those used by
   :func:`strftime`; it defaults to ``"%a %b %d %H:%M:%S %Y"`` which matches the
   formatting returned by :func:`ctime`. If *string* cannot be parsed according
   to *format*, or if it has excess data after parsing, :exc:`ValueError` is
   raised. The default values used to fill in any missing data when more
   accurate values cannot be inferred are ``(1900, 1, 1, 0, 0, 0, 0, 1, -1)``.
   Both *string* and *format* must be strings.

   For example:

      >>> import time
      >>> time.strptime("30 Nov 00", "%d %b %y")   # doctest: +NORMALIZE_WHITESPACE
      time.struct_time(tm_year=2000, tm_mon=11, tm_mday=30, tm_hour=0, tm_min=0,
                       tm_sec=0, tm_wday=3, tm_yday=335, tm_isdst=-1)

   Support for the ``%Z`` directive is based on the values contained in ``tzname``
   and whether ``daylight`` is true.  Because of this, it is platform-specific
   except for recognizing UTC and GMT which are always known (and are considered to
   be non-daylight savings timezones).

   Only the directives specified in the documentation are supported.  Because
   ``strftime()`` is implemented per platform it can sometimes offer more
   directives than those listed.  But ``strptime()`` is independent of any platform
   and thus does not necessarily support all directives available that are not
   documented as supported.
```

The type of the time value sequence returned by :func:`gmtime`, :func:`localtime`, and :func:`strptime`. It is an object with a :term:`named tuple` interface: values can be accessed by index and by attribute name. The following values are present:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 562); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 562); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 562); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 562); *backlink*

Unknown interpreted text role "term".

Index	Attribute	Values

Index	Attribute	Values
0	:attr:`tm_year` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 571); *backlink* Unknown interpreted text role "attr".	(for example, 1993)
1	:attr:`tm_mon` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 573); *backlink* Unknown interpreted text role "attr".	range [1, 12]
2	:attr:`tm_mday` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 575); *backlink* Unknown interpreted text role "attr".	range [1, 31]
3	:attr:`tm_hour` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 577); *backlink* Unknown interpreted text role "attr".	range [0, 23]

Index	Attribute	Values
4	:attr:`tm_min` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 579); ***backlink*** Unknown interpreted text role "attr".	range [0, 59]
5	:attr:`tm_sec` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 581); ***backlink*** Unknown interpreted text role "attr".	range [0, 61]; see **(2)** in :func:`strftime` description **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 581); ***backlink*** Unknown interpreted text role "func".
6	:attr:`tm_wday` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 584); ***backlink*** Unknown interpreted text role "attr".	range [0, 6], Monday is 0
7	:attr:`tm_yday` **System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 586); ***backlink*** Unknown interpreted text role "attr".	range [1, 366]

Index	Attribute	Values
8	:attr:`tm_isdst` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, **line 588**); ***backlink*** Unknown interpreted text role "attr".	0, 1 or -1; see below
N/A	:attr:`tm_zone` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, **line 590**); ***backlink*** Unknown interpreted text role "attr".	abbreviation of timezone name
N/A	:attr:`tm_gmtoff` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, **line 592**); ***backlink*** Unknown interpreted text role "attr".	offset east of UTC in seconds

Note that unlike the C structure, the month value is a range of [1, 12], not [0, 11].

In calls to :func:`mktime`, :attr:`tm_isdst` may be set to 1 when daylight savings time is in effect, and 0 when it is not. A value of -1 indicates that this is not known, and will usually result in the correct state being filled in.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, **line 597**); ***backlink***

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, **line 597**); ***backlink***

Unknown interpreted text role "attr".

When a tuple with an incorrect length is passed to a function expecting a :class:`struct_time`, or having elements of the wrong type, a :exc:`TypeError` is raised.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, **line 601**); ***backlink***

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, **line 601**); ***backlink***

Unknown interpreted text role "exc".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, **line 605**)

Unknown directive type "function".

```
.. function:: time() -> float
```

```
   Return the time in seconds since the epoch_ as a floating point
   number. The handling of `leap seconds`_ is platform dependent.
   On Windows and most Unix systems, the leap seconds are not counted towards
   the time in seconds since the epoch_. This is commonly referred to as `Unix
   time <https://en.wikipedia.org/wiki/Unix_time>`_.

   Note that even though the time is always returned as a floating point
   number, not all systems provide time with a better precision than 1 second.
   While this function normally returns non-decreasing values, it can return a
   lower value than a previous call if the system clock has been set back
   between the two calls.

   The number returned by :func:`.time` may be converted into a more common
   time format (i.e. year, month, day, hour, etc...) in UTC by passing it to
   :func:`gmtime` function or in local time by passing it to the
   :func:`localtime` function. In both cases a
   :class:`struct_time` object is returned, from which the components
   of the calendar date may be accessed as attributes.

   Use :func:`time_ns` to avoid the precision loss caused by the :class:`float`
   type.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 630)

Unknown directive type "function".

```
.. function:: time_ns() -> int

   Similar to :func:`~time.time` but returns time as an integer number of
   nanoseconds since the epoch_.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 638)

Unknown directive type "function".

```
.. function:: thread_time() -> float

   .. index::
      single: CPU time
      single: processor time
      single: benchmarking

   Return the value (in fractional seconds) of the sum of the system and user
   CPU time of the current thread.  It does not include time elapsed during
   sleep.  It is thread-specific by definition.  The reference point of the
   returned value is undefined, so that only the difference between the results
   of two calls in the same thread is valid.

   Use :func:`thread_time_ns` to avoid the precision loss caused by the
   :class:`float` type.

   .. availability::  Windows, Linux, Unix systems supporting
      ``CLOCK_THREAD_CPUTIME_ID``.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 660)

Unknown directive type "function".

```
.. function:: thread_time_ns() -> int

   Similar to :func:`thread_time` but return time as nanoseconds.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 667)

Unknown directive type "function".

```
.. function:: tzset()

   Reset the time conversion rules used by the library routines. The environment
   variable :envvar:`TZ` specifies how this is done. It will also set the variables
   ``tzname`` (from the :envvar:`TZ` environment variable), ``timezone`` (non-DST
   seconds West of UTC), ``altzone`` (DST seconds west of UTC) and ``daylight``
   (to 0 if this timezone does not have any daylight saving time rules, or to
   nonzero if there is a time, past, present or future when daylight saving time
   applies).

   .. availability:: Unix.

   .. note::

      Although in many cases, changing the :envvar:`TZ` environment variable may
      affect the output of functions like :func:`localtime` without calling
      :func:`tzset`, this behavior should not be relied on.
      The :envvar:`TZ` environment variable should contain no whitespace.

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

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

   Where the components are:

   ``std`` and ``dst``
      Three or more alphanumerics giving the timezone abbreviations. These will be
      propagated into time.tzname

   ``offset``
      The offset has the form: ``Â± 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:

      :samp:`J{n}`
         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.

      :samp:`{n}`
         The zero-based Julian day (0 <= *n* <= 365). Leap days are counted, and
         it is possible to refer to February 29.

      :samp:`M{m}.{n}.{d}`
         The *d*'th day (0 <= *d* <= 6) of week *n* of month *m* of the year (1
         <= *n* <= 5, 1 <= *m* <= 12, where week 5 means "the last *d* day in
         month *m*" which may occur in either the fourth or the fifth
         week). Week 1 is the first week in which the *d*'th day occurs. Day
         zero is a Sunday.

      ``time`` has the same format as ``offset`` except that no leading sign
      ('-' or '+') is allowed. The default, if time is not given, is 02:00:00.

   ::

      >>> os.environ['TZ'] = 'EST+05EDT,M4.1.0,M10.5.0'
      >>> time.tzset()
      >>> time.strftime('%X %x %Z')
      '02:07:36 05/08/03 EDT'
      >>> os.environ['TZ'] = 'AEST-10AEDT-11,M10.5.0,M3.5.0'
      >>> time.tzset()
      >>> time.strftime('%X %x %Z')
      '16:08:12 05/08/03 AEST'

   On many Unix systems (including \*BSD, Linux, Solaris, and Darwin), it is more
   convenient to use the system's zoneinfo (:manpage:`tzfile(5)`)  database to
   specify the timezone rules. To do this, set the  :envvar:`TZ` environment
   variable to the path of the required timezone  datafile, relative to the root of
   the systems 'zoneinfo' timezone database, usually located at
   :file:`/usr/share/zoneinfo`. For example,  ``'US/Eastern'``,
   ``'Australia/Melbourne'``, ``'Egypt'`` or  ``'Europe/Amsterdam'``. ::

      >>> os.environ['TZ'] = 'US/Eastern'
      >>> time.tzset()
      >>> time.tzname
      ('EST', 'EDT')
      >>> os.environ['TZ'] = 'Egypt'
      >>> time.tzset()
      >>> time.tzname
      ('EET', 'EEST')
```

Clock ID Constants

These constants are used as parameters for :func:`clock_getres` and :func:`clock_gettime`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 760); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 760); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 763)

Unknown directive type "data".

```
.. data:: CLOCK_BOOTTIME

   Identical to :data:`CLOCK_MONOTONIC`, except it also includes any time that
   the system is suspended.

   This allows applications to get a suspend-aware monotonic  clock  without
   having to deal with the complications of :data:`CLOCK_REALTIME`, which may
   have  discontinuities if the time is changed using ``settimeofday()`` or
   similar.
   .. availability:: Linux 2.6.39 or later.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 778)

Unknown directive type "data".

```
.. data:: CLOCK_HIGHRES

   The Solaris OS has a ``CLOCK_HIGHRES`` timer that attempts to use an optimal
   hardware source, and may give close to nanosecond resolution.
   ``CLOCK_HIGHRES`` is the nonadjustable, high-resolution clock.

   .. availability:: Solaris.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 789)

Unknown directive type "data".

```
.. data:: CLOCK_MONOTONIC

   Clock that cannot be set and represents monotonic time since some unspecified
   starting point.

   .. availability:: Unix.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 799)

Unknown directive type "data".

```
.. data:: CLOCK_MONOTONIC_RAW

   Similar to :data:`CLOCK_MONOTONIC`, but provides access to a raw
   hardware-based time that is not subject to NTP adjustments.

   .. availability:: Linux 2.6.28 and newer, macOS 10.12 and newer.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 809)

Unknown directive type "data".

```
.. data:: CLOCK_PROCESS_CPUTIME_ID

   High-resolution per-process timer from the CPU.

   .. availability:: Unix.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 818)

Unknown directive type "data".

```
.. data:: CLOCK_PROF

   High-resolution per-process timer from the CPU.

   .. availability:: FreeBSD, NetBSD 7 or later, OpenBSD.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 826)

Unknown directive type "data".

```
.. data:: CLOCK_TAI

   `International Atomic Time <https://www.nist.gov/pml/time-and-frequency-division/nist-time-frequently-asked-que

   The system must have a current leap second table in order for this to give
   the correct answer.  PTP or NTP software can maintain a leap second table.

   .. availability:: Linux.

   .. versionadded:: 3.9
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 837)

Unknown directive type "data".

```
.. data:: CLOCK_THREAD_CPUTIME_ID

   Thread-specific CPU-time clock.

   .. availability::  Unix.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 846)

Unknown directive type "data".

```
.. data:: CLOCK_UPTIME

   Time whose absolute value is the time the system has been running and not
   suspended, providing accurate uptime measurement, both absolute and
   interval.

   .. availability:: FreeBSD, OpenBSD 5.5 or later.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 857)

Unknown directive type "data".

```
.. data:: CLOCK_UPTIME_RAW

   Clock that increments monotonically, tracking the time since an arbitrary
   point, unaffected by frequency or time adjustments and not incremented while
   the system is asleep.

   .. availability:: macOS 10.12 and newer.

   .. versionadded:: 3.8
```

The following constant is the only parameter that can be sent to :func:`clock_settime`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 867); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 871)

Unknown directive type "data".

```
.. data:: CLOCK_REALTIME

   System-wide real-time clock.  Setting this clock requires appropriate
   privileges.

   .. availability:: Unix.

   .. versionadded:: 3.3
```

Timezone Constants

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 886)

Unknown directive type "data".

```
.. data:: 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.  See note below.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 892)

Unknown directive type "data".

```
.. data:: daylight

   Nonzero if a DST timezone is defined.  See note below.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst`, line 896)

Unknown directive type "data".

```
.. data:: timezone

   The offset of the local (non-DST) timezone, in seconds west of UTC (negative in
   most of Western Europe, positive in the US, zero in the UK).  See note below.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 901)

Unknown directive type "data".

```
.. data:: tzname

   A tuple of two strings: the first is the name of the local non-DST timezone, the
   second is the name of the local DST timezone.  If no DST timezone is defined,
   the second string should not be used.  See note below.
```

Note

For the above Timezone constants (:data:`altzone`, :data:`daylight`, :data:`timezone`, and :data:`tzname`), the value is determined by the timezone rules in effect at module load time or the last time :func:`tzset` is called and may be incorrect for times in the past. It is recommended to use the :attr:`tm_gmtoff` and :attr:`tm_zone` results from :func:`localtime` to obtain timezone information.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 909); *backlink*

Unknown interpreted text role "data".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 909); *backlink*

Unknown interpreted text role "data".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 909); *backlink*

Unknown interpreted text role "data".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 909); *backlink*

Unknown interpreted text role "data".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 909); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 909); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 909); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 909); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]time.rst, line 916)

Unknown directive type "seealso".

```
.. seealso::

   Module :mod:`datetime`
      More object-oriented interface to dates and times.

   Module :mod:`locale`
      Internationalization services.  The locale setting affects the interpretation
      of many format specifiers in :func:`strftime` and :func:`strptime`.
```

```
Module :mod:`calendar`
   General calendar-related functions.   :func:`~calendar.timegm` is the
   inverse of :func:`gmtime` from this module.
```

Footnotes

[1] The use of `%Z` is now deprecated, but the `%z` escape that expands to the preferred hour/minute offset is not supported by all ANSI C libraries. Also, a strict reading of the original 1982 RFC 822 standard calls for a two-digit year (`%y` rather than `%Y`), but practice moved to 4-digit years long before the year 2000. After that, RFC 822 became obsolete and the 4-digit year has been first recommended by RFC 1123 and then mandated by RFC 2822.