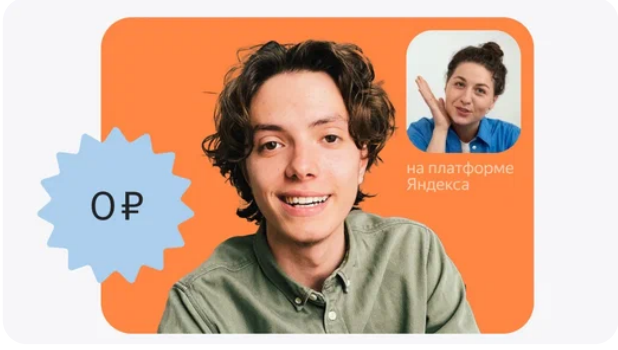


Модуль time в Python, реальное время ОС



practicum.yandex.ru

РЕКЛАМА · 18+ Я

Бесплатное занятие английским в Яндекс Практикуме

Тест на уровень языка • Разбор грамматики • Разговорная практика • За 30 минут
Узнать больше

[Стандартная библиотека Python3.](#) / Модуль time в Python, реальное время ОС

Работа с системным временем операционной системы

[Модуль time](#) предоставляет доступ к нескольким различным типам часов, каждый из которых используется для разных целей:

- Стандартные системные вызовы типа [time.time\(\)](#) сообщают системное время в секундах с начала "эпохи".
- Монотонные часы [time.monotonic\(\)](#) можно использовать для измерения прошедшего времени в каком то длительном процессе, т.к. функция не зависит от настроек системного времени и гарантирует, что никогда не передвинется назад, даже если системное время изменилось.
- Для тестирования производительности модуль предоставляет доступ к часам с максимально высоким разрешением [time.perf_counter\(\)](#), чтобы сделать измерения короткого времени более точными.
- Функция [time.process_time\(\)](#) возвращает объединенное время процессора и системное время.

Хотя этот модуль всегда доступен, на некоторых платформах доступны не все функции. Большинство функций, определенных в этом модуле, вызывают библиотечные функции языка C с тем же именем. Семантика этих функций варьируется в зависимости от платформы, по этому будет полезно ознакомиться с документацией вашей платформы.

Объяснение некоторых терминов и соглашений:

- Эпоха - это точка начала времени, которая зависит от платформы. Для Unix эпоха - 1 января 1970 года, 00:00:00 (UTC). Чтобы узнать, какова эпоха на данной платформе, посмотрите [time.gmtime\(0\)](#).
- Термин "секунды с начала эпохи" относится к общему количеству прошедших секунд с начала эпохи, обычно не включая високосные секунды. Дополнительные секунды исключаются из этой суммы на всех POSIX-совместимых платформах.
- Функции в этом модуле могут не обрабатывать даты и время до эпохи или в далеком будущем. Точка отсечения в будущем определяется библиотекой языка C. Для 32-битных систем это обычно 2038 год.
- Функция [time.strptime\(\)](#) может анализировать годы записанные в двухзначном формате, если задан код формата %. Когда анализируются двухзначные годы, они конвертируются в соответствии со стандартами POSIX и ISO: значения 69-99 будут отображаться как 1969-1999, а значения 0-68 отображаются как 2000-2068.
- UTC - Всемирное координированное время, ранее известное как среднее время по Гринвичу или GMT. Аббревиатура UTC - не ошибка, а компромисс между английским и французским языками.
- Летнее время - это настройка часового пояса, обычно на один час в течение части года. Правила летнего времени магические и определяются местным законодательством, могут меняться из года в год. Библиотека языка C имеет таблицу, содержащую локальные правила, часто для гибкости она читается из системного файла и является единственным источником в этом отношении.
- Точность различных функций реального времени может быть меньше, чем предлагается единицами, в которых выражается их значение или аргумент. Например на большинстве систем Unix такт часов составляет от 50 до 100 раз в секунду.
- С другой стороны, точность [time.time\(\)](#) и [time.sleep\(\)](#) лучше, чем их Unix-эквиваленты: время выражается в виде [чисел с плавающей запятой](#), [time.time\(\)](#) возвращает наиболее точное доступное время, а [time.sleep\(\)](#) будет принимать время с ненулевой дробью.
- Значение времени, возвращаемое [time.gmtime\(\)](#), [time.localtime\(\)](#) и [time.strptime\(\)](#) и принимаемое [time.asctime\(\)](#), [time.mktime\(\)](#) и [time.strftime\(\)](#) представляет собой последовательность из 9 целых чисел. Возвращаемые значения [time.gmtime\(\)](#), [time.localtime\(\)](#) и [time.strptime\(\)](#) также предлагают имена атрибутов для отдельных полей.

Смотрите [time.struct_time](#) для описания этих объектов.

Используйте следующие функции для преобразования между представлениями времени:

Из формата	В формат	Функция
Секунды эпохи	time.struct_time в UTC	time.gmtime()
Секунды эпохи	time.struct_time в местном времени	time.localtime()
time.struct_time в UTC	Секунды эпохи	calendar.timegm()
time.struct_time в местном времени	Секунды эпохи	time.mktime()

Примеры использования:

```
>>> import time
>>> sec_utc = time.time()
>>> sec_utc
# 1587728179.4579966

# переводим из секунд в 'struct_time'
>>> time_utc = time.gmtime(sec_utc)
>>> time_utc
# time.struct_time(tm_year=2020, tm_mon=4,
# tm_mday=24, tm_hour=11, tm_min=36, tm_sec=19,
# tm_wday=4, tm_yday=115, tm_isdst=0)

# Локальное время получаем сразу в 'struct_time'
>>> time_local = time.localtime()
# time.struct_time(tm_year=2020, tm_mon=4,
# tm_mday=24, tm_hour=14, tm_min=36, tm_sec=40,
# tm_wday=4, tm_yday=115, tm_isdst=0)

# получаем форматированную строку из 'struct_time'
>>> time.strftime('%d.%m.%Y г. %H часов %M мин. %S сек.', time_utc)
# '24.04.2020 г. 11 часов 36 мин. 19 сек.'
>>> time.strftime('%d.%m.%Y г. %H часов %M мин. %S сек.', time_local)
# '24.04.2020 г. 14 часов 36 мин. 40 сек.'

# теперь получим из строки со временем - структуру 'struct_time'
>>> struct_time = time.strptime('10/10/2020 10:15', '%d/%m/%Y %H:%M')
# time.struct_time(tm_year=2020, tm_mon=10,
# tm_mday=10, tm_hour=10, tm_min=15, tm_sec=0,
# tm_wday=5, tm_yday=284, tm_isdst=-1)

# переведем 'struct_time' в секунды "эпохи"
>>> time.mktime(struct_time)
# 1602314100.0
```

Для получения дополнительной информации смотрите также модули [datetime](#) и [calendar](#).

Содержание раздела:
<ul style="list-style-type: none">КРАТКИЙ ОБЗОР МАТЕРИАЛА.Функция asctime() модуля timeФункция ctime() модуля timeФункция get clock info() модуля timeФункция gmtime() модуля timeФункция localtime() модуля timeФункция mktime() модуля timeФункция monotonic() модуля timeФункция perf counter() модуля timeФункция process time() модуля timeФункция sleep() модуля timeФункция strftime() модуля time

- [Функция `strptime\(\)` модуля `time`](#)
- [Класс `struct time` модуля `time`](#)
- [Функция `time\(\)` модуля `time`](#)
- [Функция `tzset\(\)` модуля `time`](#)
- [Константы часового пояса модуля `time`](#)

ХОЧУ ПОМОЧЬ
ПРОЕКТУ



[DOCS-Python.ru](#)™, 2023 г.

(Внимание! При копировании материала ссылка на источник обязательна)

[@docs_python_ru](#)