|  |  |
| --- | --- |
| time.time() | 返回当前系统时间戳 |
| time.ctime([sec]) | 返回当前系统时间的字符串  -->传入时间戳时，返回该时间戳对应的时间 |
| time.gmtime([sec]) | 返回***0时区***时间元组  -->传入时间戳时，返回改时间戳对应的***0时区***下的时间元组 |
| time.localtime(sec) | 返回***当前时区***时间元组  -->传入时间戳时，返回改时间戳对应的***当前时***区下的时间元组 |
| time.mktime(time\_struct\_obj) | 根据传入的time\_struct\_obj，返回时间戳  P.s 根据同样的时间戳，使用time.gmtime以及time.localtime所得到的time\_struct不一样。再使用time.mktime再返回所得到的时间戳也不一样。这是因为time\_sturct没有时区，等于time.mktime接受的time\_struct根本不是同一个时间，所以返回的时间戳也不一样。 |
| time.sleep(secs) | 让程序挂起指定的秒数 |
| time.strftime(format, struct\_obj) | 传入struct\_obj并且按一定格式输出。 |
| time.strptime(str,format) | 传入时间字符串格式转为struct\_obj格式 |

Time 模块运用

Datetime模块

|  |  |
| --- | --- |
| time.time() | 返回当前系统时间戳 |
| time.ctime([sec]) | 返回当前系统时间的字符串  -->传入时间戳时，返回该时间戳对应的时间 |
| time.gmtime([sec]) | 返回***0时区***时间元组  -->传入时间戳时，返回改时间戳对应的***0时区***下的时间元组 |
| time.localtime(sec) | 返回***当前时区***时间元组  -->传入时间戳时，返回改时间戳对应的***当前时***区下的时间元组 |
|  | 根据传入的time\_struct\_obj，返回时间戳  P.s 根据同样的时间戳，使用time.gmtime以及time.localtime所得到的time\_struct不一样。再使用time.mktime再返回所得到的时间戳也不一样。这是因为time\_sturct没有时区，等于time.mktime接受的time\_struct根本不是同一个时间，所以返回的时间戳也不一样。 |
| time.sleep(secs) | 让程序挂起指定的秒数 |
| time.strftime(format, struct\_obj) | 传入struct\_obj并且按一定格式输出。 |
| time.strptime(str,format) | 传入时间字符串格式转为struct\_obj格式 |

|  |  |  |
| --- | --- | --- |
| datetime.date.today() | 返回今日日期的date对象 | |
| datetime.datetime.now() | 返回当前时区的此时此刻的datetime对象 | |
| datetime.datetime.strptime(时间字符串，格式) | 返回该时间字符串对应的datetime对象 | |
| datetime\_obj.timetuple() | 从datetime\_obj中返回一个时间元组类型 | |
| datetime\_obj - datetime.timedelta([days,hours,minutes…]) | | 通过datetime.timedelta构建timedelta类型，然后使用对datetime\_obj可以进行加减时间 |
| datetime\_obj 之间比较 [ > < =] | | |

Note:

1. UTC为UTC为世界标准时间，GMT为格林威治时间​（Greenwich Mean Time，简称G.M.T.）GMT时间与UTC时间相差不超过0.9秒，达到一致，如果将要超过0.9秒则通过闰秒抵消。
2. 在Python中通常有三种方式表达时间，时间戳，格式化的时间字符串，时间元组(struct\_time)
3. 时间戳(timestamp)代表从1970年1月1日0时，到现在为止的秒数。（1970年1月1日是Unix系统开始应用的时候）

时间戳均已UTC时间作为基准，在任何时区取得时间戳都一样。并且由于这个特性用于服务器之间的时间交流。

一样的时间戳，根据时区的不同，返回的时间也就不一样。

1. 时间元组struct\_time,里面含有9个元素。

|  |  |
| --- | --- |
| tm\_year | 年 |
| tm\_mon | 月(1-12) |
| tm\_mday | 日(1-31) |
| tm\_hour | 小时(0-23) |
| tm\_min | 分(0-59) |
| tm\_sec | 秒(0-59) |
| tm\_wday | 周几(0-6) |
| tm\_yday | 一年中的第几天 |
| tm\_isdsy | 是否为夏令时 |

P.s ***注意时间元组里面没有时区。***

1. datetime 在Python中分为两种类型。第一种是 offset-native就是没有设置时区的datetime, 另外一种是offset-aware设置时区的datetime. 同一种类型的datetime才可以进行比较。

使用datetime\_obj.replace函数可以将一个datetime\_obj 改成offset-native或者offset-aware

6. 常用的操作：

从字符串转为datetime对象：

datestring = “2019-07-08 10:28:00”

dd = datetime.datetime.strptime(datestring, "%Y-%m-%d %H:%M:%S")

为没有设置时区的datetime对象设置时区：

tz = pytz.timezone("Canada/Eastern")

dd = dd.replace(tzinfo=tz)