

# 常见时间名词

---

## GMT

格林威治标准时间 (Greenwich Mean Time, GMT) , 指的是太阳经过格林威治处的本初子午线时的时间。是最早的世界时 (UT, Universal Time) 标准。

## 时区

1884年制定了时区, 以本初子午线为起点, 经度每隔15度, 划分一个时区, 相邻时区相差一个小时。一些国家会根据行政管理方便, 调整时区。

## UTC

协调世界时 (Coordinated Universal Time) , 目前的世界时标准。

它由全球很多国家的铯原子钟时间协调得来, 与所处地点无关。为了适应地球自转误差, UTC会做人为调整, 比如闰秒。

可以简单的认为, GMT就是UTC + 0。

## CST

中国标准时间 (China Standard Time) , UTC + 8:00, 也称北京时间。当然CST也可以是美国中部时间等的缩写。

# 时间模块

---

## datetime模块

datetime类是时间高级类

- 类方法, 即使用类调用的方法, 由类方法获得一个时间对象
  - `now(tz=None)` 返回当前时间的datetime对象, 时间到微秒, 如果tz为None, 返回当前时区的不带时区信息的时间
  - `utcnow()` 不带时区的0时区时间
  - `fromtimestamp(timestamp, tz=None)` 从一个时间戳返回一个datetime对象
  - `tz=datetime.timezone.utc` 即0时区timezone对象
- 时间对象方法
  - `timestamp()` 返回一个到微秒的时间戳
    - Unix时间戳(Unix Epoch): 格林威治时间1970年1月1日0点到现在的秒数
  - 构造方法 `datetime.datetime(2016, 12, 6, 16, 29, 43, 79043)`
  - `year`、`month`、`day`、`hour`、`minute`、`second`、`microsecond`, 取datetime对象的年月日时分秒及微秒
  - `weekday()` 返回星期的天, 周一0, 周日6
  - `isoweekday()` 返回星期的天, 周一1, 周日7
  - `date()` 返回日期date对象
  - `time()` 返回时间time对象

```

1 import datetime
2
3 # 构造器
4 print(datetime.datetime(2021, 6, 17, 18, 20, 5))
5
6 # 类方法获得时间对象
7 print(datetime.datetime.now(datetime.timezone(datetime.timedelta(hours=8))))
8 # 时区时间
9 print(datetime.datetime.now()) # 无时区时间
10 print(datetime.datetime.utcnow()) # UTC时间, 可以认为是GMT或0时区时间
11
12 # 时间戳操作
13 stamp = datetime.datetime.now().timestamp() # 获得时间戳
14 print(stamp)
15 dt = datetime.datetime.fromtimestamp(stamp) # 从时间戳获得时间对象
16 print(dt)
17
18 print(type(dt.date()), dt.date())
19 print(type(dt.time()), dt.time())

```

Python中时间分为两种：

- naive, 没有时区信息的时间, 没法明确定位。这个时间表示那个地区的时间, 全看程序理解
- aware, 包含时区的时间
- 这两种时间不能混合计算

## 日期与格式化

- 类方法 `strptime(date_string, format)`, 返回datetime对象 (时间字符串+格式化字符串 => 时间对象)
- 对象方法 `strftime(format)`, 返回字符串 (时间对象通过格式字符串 => 时间字符串)
- 字符串format函数格式化 (时间对象通过格式字符串 => 时间字符串)

```

1 import datetime
2
3 datestr = '2018-01-10 17:16:08'
4 dt = datetime.datetime.strptime(datestr, '%Y-%m-%d %H:%M:%S') #由字符串到时间对象
5 print(type(dt), dt)
6 print(dt.strftime('%Y/%m/%d-%H:%M:%S')) # 输出为字符串
7 print("{:%Y/%m/%d %H:%M:%S}".format(dt)) # 输出为字符串

```

## timedelta类

- `datetime2 = datetime1 + timedelta`
- `datetime2 = datetime1 - timedelta`
- `timedelta = datetime1 - datetime2`
- 构造方法
  - `datetime.timedelta(days=0, seconds=0, microseconds=0, milliseconds=0, minutes=0, hours=0, weeks=0)`
  - `year = datetime.timedelta(days=365)`

- timedelta对象有方法total\_seconds(), 返回时间差的总秒数

## time模块

- time.sleep(secs) 将调用线程挂起指定的秒数