

# datetime 模块

datetime 模块是 date 和 time 模块的集合，datetime 有 2 个常量，**MAXYEAR=9999** 和 **MINYEAR=1**；datetime 还定义了 5 个 class，分别是：

- ① datetime.**date**                      表示日期的类
- ② datetime.**datetime**                表示日期时间的类
- ③ datetime.**time**                     表示时间的类
- ④ datetime.**timedelta**               表示时间间隔
- ⑤ datetime.**tzinfo**                  时区的相关信息

## 一、date 类

### 1. 构造方法

用法： **date(year,month,day)**

功能：构造一个 datetime.date 对象

```
from datetime import date
d = date(2019,5,13)
print(type(d))          # <class 'datetime.date'>
print(d)                 # 2019-05-13
```

### 2. Python 中日期时间格式化符号

符号	含义	示例
%y	两位数的年份表示	01-99
%Y	四位数的年份表示	0001-9999
%m	月份	01-12
%d	一个月的第几天	01-31
%H	24 小时制	00-23
%I	12 小时制	01-12
%M	分钟	00-59
%S	秒	00-59
%a	本地简化星期名称	Mon
%A	本地完整星期时间	Monday
%b	本地简化月份名称	Aug
%B	本地完整月份名称	August
%c	本地相应的日期表示和时间表示	Sat Aug 15 00:00:00 2020
%j	一年中的第几天	001-366
%p	本地 a.m.和 p.m.	AM/PM
%U	一年中的星期数，星期天为一星期开始	00-53
%x	本地相应的日期表示	08/15/20
%X	本地相应的时间表示	00:00:00
%Z	当前时区的名称	
%%	百分号	%

### 3. ctime()方法（调用背、本机 C 函数）

用法： `dateobject.ctime()`

功能：返回一种时间的表示，格式化字符串为 `"%s %s %2d 00:00:00 %04d"`

示例：

```
from datetime import date
d = date(2020,5,1)
print(d)                # 2020-05-01
print(d.ctime())        # Fri May  1 00:00:00 2020
```

### 4. today()方法

用法： `date.today()`

功能：返回当地日期

示例：

```
>>> from datetime import date
>>> date.today()
datetime.date(2019, 5, 13)
```

### 5. fromtimestamp()方法

用法： `date.fromtimestamp()`

功能：根据时间戳生产 date 对象示例，时间戳单位是秒，从 1970 年 1 月 1 日 0 点开始

示例：

```
>>> from datetime import date
>>> date.fromtimestamp(1596325456)
datetime.date(2020, 8, 2)
```

### 6. fromordinal()方法

用法： `date.fromordinal(ordinal)`

功能：生产一个 date 实例，参数 ordinal 是大于 0 的 int 整数，相对于 1 年 1 月 1 日时间

示例：

```
>>> from datetime import date
>>> date.fromordinal(1)
datetime.date(1, 1, 1)
>>> date.fromordinal(368)
datetime.date(2, 1, 3)
```

### 7. fromisoformat()方法

用法： `date.fromisoformat(date_string)`

功能：根据日记格式串生产 date 实例，格式形式为： `"yyyy-mm-dd"`

示例：

```
>>> from datetime import date
>>> date.fromisoformat('2018-08-08')
datetime.date(2018, 8, 8)
>>> date.fromisoformat('0001-08-08')
datetime.date(1, 8, 8)
```

## 8. min 和 max 属性

用法: **date.min** 和 **date.max**

功能: 返回 date 对象示例, 分别是 date 类所能表示日期的最早和最迟日期

示例:

```
>>> from datetime import date
>>> date.min
datetime.date(1, 1, 1)
>>> date.max
datetime.date(9999, 12, 31)
```

## 9. year, month 和 day 属性

用法: **dateobject.year**, **dateobject.month**, **dateobject.day**

功能: 返回 date 对象的日期对应的值

示例:

```
>>> from datetime import date
>>> d = date(2018,5,13)
>>> d.year
2018
>>> d.month
5
>>> d.day
13
```

## 10. replace()方法

用法: **dateobject.replace(year=self.year, month=self.month, day=self.day)**

功能: 修改日期, 产生新的 date 实例, 不改变原来对象的属性

示例:

```
>>> from datetime import date
>>> d = date(2019,5,13)
>>> d.replace(2018)
datetime.date(2018, 5, 13)
>>> d.replace(2017,3)
datetime.date(2017, 3, 13)
>>> d.replace(2016,1,1)
datetime.date(2016, 1, 1)
>>> d
datetime.date(2019, 5, 13)
```

## 11. toordinal()函数

用法: **dateobject.toordinal()**

功能: 返回相对于 1 年 1 月 1 日的距离 (单位: 天), 1 年 1 月 1 日返回 1

示例:

```
>>> from datetime import date
>>> d = date(1,2,5)
>>> d.toordinal()
36
```

## 12. weekday()方法

用法: [dateobject.weekday\(\)](#)

功能: 返回星期几, 其中 0 表示星期一, 6 表示星期日

示例:

```
>>> from datetime import date
>>> d = date(2019,5,13)
>>> d.weekday()
0
```

## 13. isoweekday()方法

用法: [dateobject.isoweekday\(\)](#)

功能: 也是返回星期, 不过, 星期一返回 1, 星期日返回 7

## 14. isocalendar()方法

用法: [dateobject.isocalendar\(\)](#)

功能: 返回一个三元组, 分别为: (ISO 年份, ISO 周编号, ISO 工作日)

简单地说, 第一个元素是年份, 第二个元素是一年的第多少周, 第三个元素是一周的第几天(周一算第一天)

示例:

```
>>> from datetime import date
>>> d = date(2019,5,13)
>>> d.isocalendar()
(2019, 20, 1)
```

## 15. isoformat()方法

用法: [dateobject.isoformat\(\)](#)

功能: 返回格式化的日期, 如果对日期用 [str\(d\)](#)方法, 则相当于调用 [isoformat\(\)](#)方法

示例:

```
>>> from datetime import date
>>> d = date(2019,5,13)
>>> d.isoformat()
'2019-05-13'
```

## 16. strftime()方法

用法: [dateobject.strftime\(format\)](#)

功能: 将 date 示例按照 format 格式转成字符串, 别用中文, 不知道为啥报错

示例:

```
>>> from datetime import date
>>> d = date(2019,5,13)
>>> d.strftime('%Y-%m-%d %A %p %H:%M:%S')
'2019-05-13 Monday AM 00:00:00'
>>> d.strftime('%y-%m-%d %A %p %I:%M:%S')
'19-05-13 Monday AM 12:00:00'
```

## 二、datetime 类

### 1. 构造函数

用法: **datetime(year,month,day,hour=0,min=0,sec=0,microsecond=0,tzinfo=None,\*,fold=0)**

功能: 创建 datetime 对象实例,microsecond 是微秒,  $0 \leq \text{microsecond} < 1000000$ , fold 一般不管, fold in {0,1}

示例:

```
>>> from datetime import datetime
>>> datetime(2019,5,13)
datetime.datetime(2019, 5, 13, 0, 0)
>>> datetime(2019,5,13,13,14,36)
datetime.datetime(2019, 5, 13, 13, 14, 36)
```

### 2. today()方法

用法: **datetime.today()**

功能: 创建现在的 datetime 对象, 是一个 tuple, 顺序为年月日时分秒微秒

示例:

```
>>> from datetime import datetime
>>> datetime.today()
datetime.datetime(2019, 5, 13, 13, 31, 3, 976922)
```

### 3. now()方法

用法: **datetime.now(tz=None)**

功能: 创建 datetime 对象, 默认和 today()一样, 若指定 tz 就是把元素值转换为指定时区

示例:

```
>>> from datetime import datetime
>>> datetime.now()
datetime.datetime(2019, 5, 13, 13, 36, 47, 246891)
```

### 4. fromtimestamp()方法

用法: **datetime.fromtimestamp(timestamp,tz=None)**

功能: 根据时间戳创建 datetime 对象, 时间戳单位是秒, 从 1970 年 1 月 1 日 0 点开始

示例:

```
>>> from datetime import datetime
>>> datetime.fromtimestamp(1565333665)
datetime.datetime(2019, 8, 9, 14, 54, 25)
```

### 5. fromordinal()方法

用法: **fromordinal(ordinal)**

功能: 生产一个 datetime 实例, ordinal 是大于 0 的 int 整数, 相对于 1 年 1 月 1 日时间

示例:

```
>>> from datetime import datetime
>>> datetime.fromordinal(60)
datetime.datetime(1, 3, 1, 0, 0)
```

## 6. combine()方法

用法: `datetime.combine(date,time,tzinfo=self.tzinfo)`

功能: 根据 date 实例和 time 实例创建 datetime 对象

示例:

```
>>> from datetime import date,datetime,time
>>> d = date(2019,5,20)
>>> t = time(13,14,0)
>>> datetime.combine(d,t)
datetime.datetime(2019, 5, 20, 13, 14)
```

## 7. fromisoformat()方法

用法: `datetime.fromisoformat(date_string)`

功能: 根据所给格式创建 datetime 对象, 其中, 格式为

`YYYY-MM-DD[*HH[:MM[:SS[.fff[fff]]]][+HH:MM[:SS[.ffffff]]]]` 一般用  
`YYYY-MM-DD` 或 `YYYY-MM-DD HH:MM:SS.ffffff`

示例:

```
>>> from datetime import datetime
>>> datetime.fromisoformat('2019-05-20 13:14:00.000001')
datetime.datetime(2019, 5, 20, 13, 14, 0, 1)
>>> datetime.fromisoformat('2018-09-22')
datetime.datetime(2018, 9, 22, 0, 0)
```

## 8. strptime()方法

用法: `datetime.strptime(date_string, format)`

功能: 按照 format 格式串解析 date\_string, 得到日期时间各元素值创建 datetime 对象

示例:

```
>>> from datetime import datetime
>>> datetime.strptime('2019/05/01 15:00:00','%Y/%m/%d %H:%M:%S')
datetime.datetime(2019, 5, 1, 15, 0)
```

## 9. min 和 max 属性

示例:

```
>>> datetime.min
datetime.datetime(1, 1, 1, 0, 0)
>>> datetime.max
datetime.datetime(9999, 12, 31, 23, 59, 59, 999999)
```

## 10. 实例属性

```
>>> d = datetime(2019,5,20,13,14,25,102386)
>>> d.year                # 2019
>>> d.month               # 5
>>> d.day                 # 20
>>> d.hour                # 13
>>> d.minute              # 14
>>> d.second              # 25
>>> d.microsecond         # 102386
```

## 11. date()方法和 time()方法

用法: **datetimeobject.date()** 和 **datetimeobject.time()**

功能: 把 datetime 对象转换为 date 对象或者 time 对象

示例:

```
>>> from datetime import datetime
>>> d=datetime(2013,1,4,5,20,13,14)
>>> d.date()
datetime.date(2013, 1, 4)
>>> d.time()
datetime.time(5, 20, 13, 14)
```

【注】

timetz()方法和 time()方法一样, 只是返回多了 tzinfo 的值

## 12. replace()方法

用法: **datetimeobject.replace(year, month, day, hour, minute, second, microsecond, tzinfo)**

功能: 修改某些属性值创建一个新的 datetime 对象, 参数都有默认值

示例:

```
>>> from datetime import datetime
>>> d=datetime(2013,1,4,5,20,13,14)
>>> d.replace(2048)
datetime.datetime(2048, 1, 4, 5, 20, 13, 14)
```

## 13. toordinal()方法

用法: **datetimeobject.toordinal()**

功能: 返回相对于 1 年 1 月 1 日过去的天数, 1 年 1 月 1 日返回 1,

作用:

```
>>> from datetime import datetime
>>> d=datetime(2013,1,4,5,20,13,14)
>>> d.toordinal()
734872
```

## 14. timestamp()方法

用法: **datetimeobject.timestamp()**

功能: 返回时间戳, 从 1970 年 1 月 1 日 0 点开始经过的秒数

## 15. weekday()方法

用法: **datetimeobject.weekday()**

功能: 返回星期几, 星期一为 0, 星期日为 6

## 16. isoweekday()方法

用法: **datetimeobject.isoweekday()**

功能: 返回星期几, 星期一为 1, 星期日为 7

## 17. isocalendar()方法

用法: `dateobject.isocalendar()`

功能: 返回一个三元组, 分别为: (ISO 年份, ISO 周编号, ISO 工作日)

简单地说, 第一个元素是年份, 第二个元素是一年的第多少周, 第三个元素是一周的第几天(周一算第一天)

## 18. isoformat()方法

用法: `datetimeobject.isoformat(seq='T', timespec='auto')`

功能: 返回 ISO 格式化的日期时间, 默认 yyyy-mm-ddTHH:MM:SS.ffffff

参数:

seq: 日期和时间的分隔字符, 默认是 T, 一般用空格字符

timespec: 精确度

auto——精确到秒

hours——精确到小时

minutes——精确到分

seconds——精确到秒

milliseconds——精确到毫秒

microseconds——精确到微秒

示例:

```
>>> from datetime import datetime
>>> d = datetime.now()
>>> d.isoformat(' ','minutes')
'2019-05-13 16:14'
>>> d.isoformat(' ','hours')
'2019-05-13 16'
```

## 19. strftime()方法

用法: `datetimeobject.strftime(format)`

功能: 将 datetime 实例按照 format 格式转成字符串

实例:

```
>>> from datetime import datetime
>>> d = datetime.now()
>>> d.strftime('%Y / %m / %d ** %H:%M:%S')
'2019 / 05 / 13 ** 16:14:43'
```



### 三、time 类

#### 1. 构造函数

用法: `time(hour=0,minute=0,second=0,microsecond=0,tzinfo=0)`

功能: 创建 time 对象

示例:

```
>>> from datetime import time
>>> time()
datetime.time(0, 0)
>>> time(13,14,5,20)
datetime.time(13, 14, 5, 20)
```

#### 2. max 和 min 属性

```
>>> from datetime import time
>>> time.min
datetime.time(0, 0)
>>> time.max
datetime.time(23, 59, 59, 999999)
```

#### 3. 实例属性

```
>>> t = time(13,14,5,20)
>>> t.hour           # 13
>>> t.minute         # 14
>>> t.second         # 5
>>> t.microsecond    # 20
```

#### 4. fromisoformat()方法

用法: `time.fromisoformat(time_string)`

功能: 根据时间字符串创建 time 对象 `HH[:MM[:SS[.fff[fff]]]]`

示例:

```
>>> from datetime import time
>>> time.fromisoformat('14:15:20.556')
datetime.time(14, 15, 20, 556000)
```

#### 5. replace()方法

用法: `timeobject.replace(hour,minute,second,microsecond,tzinfo)`

功能: 替换属性值, 创建新的 time 对象

#### 6. isoformat()方法

用法: `timeobject.isoformat(timespec='auto')`

功能: 返回 ISO 格式化的日期时间, 默认 `HH:MM:SS.ffffff`, 参数见前面同名函数

#### 7. strftime()方法

用法: `timeobject.strftime(format)`

功能: 按照格式化字符串输出时间字符串