

## 3.12 基本的日期与时间转换¶

### 问题¶

你需要执行简单的时间转换，比如天到秒，小时到分钟等的转换。

### 解决方案¶

为了执行不同时间单位的转换和计算，请使用 `datetime` 模块。比如，为了表示一个时间段，可以创建一个 `timedelta` 实例，就像下面这样：

```
>>> from datetime import timedelta
>>> a = timedelta(days=2, hours=6)
>>> b = timedelta(hours=4.5)
>>> c = a + b
>>> c.days
2
>>> c.seconds
37800
>>> c.seconds / 3600
10.5
>>> c.total_seconds() / 3600
58.5
>>>
```

如果你想表示指定的日期和时间，先创建一个 `datetime` 实例然后使用标准的数学运算来操作它们。比如：

```
>>> from datetime import datetime
>>> a = datetime(2012, 9, 23)
>>> print(a + timedelta(days=10))
2012-10-03 00:00:00
>>>
>>> b = datetime(2012, 12, 21)
>>> d = b - a
>>> d.days
89
>>> now = datetime.today()
>>> print(now)
2012-12-21 14:54:43.094063
>>> print(now + timedelta(minutes=10))
2012-12-21 15:04:43.094063
>>>
```

在计算的时候，需要注意的是 `datetime` 会自动处理闰年。比如：

```
>>> a = datetime(2012, 3, 1)
>>> b = datetime(2012, 2, 28)
>>> a - b
datetime.timedelta(2)
>>> (a - b).days
2
>>> c = datetime(2013, 3, 1)
>>> d = datetime(2013, 2, 28)
>>> (c - d).days
1
>>>
```

### 讨论¶

对大多数基本的日期和时间处理问题，`datetime` 模块已经足够了。如果你需要执行更加复杂的日期操作，比如处理时区，模糊时间范围，节假日计算等等，可以考虑使用 [dateutil模块](#)

许多类似的时间计算可以使用 `dateutil.relativedelta()` 函数代替。但是，有一点需要注意的就是，它会在处理月份(还有它们的天数差距)的时候填充间隙。看例子最清楚：

```
>>> a = datetime(2012, 9, 23)
>>> a + timedelta(months=1)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'months' is an invalid keyword argument for this function
>>>
>>> from dateutil.relativedelta import relativedelta
>>> a + relativedelta(months=+1)
datetime.datetime(2012, 10, 23, 0, 0)
>>> a + relativedelta(months=+4)
datetime.datetime(2013, 1, 23, 0, 0)
>>>
>>> # Time between two dates
>>> b = datetime(2012, 12, 21)
>>> d = b - a
>>> d
datetime.timedelta(89)
>>> d = relativedelta(b, a)
>>> d
relativedelta(months=+2, days=+28)
>>> d.months
2
>>> d.days
28
>>>
```