

## 3.13 计算上一个周五的日期¶

### 问题¶

你需要一个通用方法来计算一周中某一天上一次出现的日期，例如上一个周五的日期。

### 解决方案¶

Python的 `datetime` 模块中有工具函数和类可以帮助你执行这样的计算。下面是对类似这样的问题的一个通用解决方案：

```
#!/usr/bin/env python
# -*- encoding: utf-8 -*-
"""
Topic: 最后的周五
Desc :
"""
from datetime import datetime, timedelta

weekdays = ['Monday', 'Tuesday', 'Wednesday', 'Thursday',
              'Friday', 'Saturday', 'Sunday']

def get_previous_byday(dayname, start_date=None):
    if start_date is None:
        start_date = datetime.today()
    day_num = start_date.weekday()
    day_num_target = weekdays.index(dayname)
    days_ago = (7 + day_num - day_num_target) % 7
    if days_ago == 0:
        days_ago = 7
    target_date = start_date - timedelta(days=days_ago)
    return target_date
```

在交互式解释器中使用如下：

```
>>> datetime.today() # For reference
datetime.datetime(2012, 8, 28, 22, 4, 30, 263076)
>>> get_previous_byday('Monday')
datetime.datetime(2012, 8, 27, 22, 3, 57, 29045)
>>> get_previous_byday('Tuesday') # Previous week, not today
datetime.datetime(2012, 8, 21, 22, 4, 12, 629771)
>>> get_previous_byday('Friday')
datetime.datetime(2012, 8, 24, 22, 5, 9, 911393)
>>>
```

可选的 `start_date` 参数可以由另外一个 `datetime` 实例来提供。比如：

```
>>> get_previous_byday('Sunday', datetime(2012, 12, 21))
datetime.datetime(2012, 12, 16, 0, 0)
>>>
```

### 讨论¶

上面的算法原理是这样的：先将开始日期和目标日期映射到星期数组的位置上(星期一索引为0)，然后通过模运算计算出目标日期要经过多少天才能到达开始日期。然后用开始日期减去那个时间差即得到结果日期。

如果你要像这样执行大量的日期计算的话，你最好安装第三方包 `python-dateutil` 来代替。比如，下面是使用 `dateutil` 模块中的 `relativedelta()` 函数执行同样的计算：

```
>>> from datetime import datetime
>>> from dateutil.relativedelta import relativedelta
>>> from dateutil.rrule import *
>>> d = datetime.now()
```

```
>>> print(d)
2012-12-23 16:31:52.718111
```

```
>>> # Next Friday
>>> print(d + relativedelta(weekday=FR))
2012-12-28 16:31:52.718111
>>>
```

```
>>> # Last Friday
>>> print(d + relativedelta(weekday=FR(-1)))
2012-12-21 16:31:52.718111
>>>
```