

# Python小练习：解决strftime()中国时区乱码问题

作者：凯鲁嘎吉 - 博客园 <http://www.cnblogs.com/kailugaji/>

## 1. mytest.py

```
1 # -*- coding: utf-8 -*-
2 # Author: 凯鲁嘎吉 Coral Gajic
3 # https://www.cnblogs.com/kailugaji/
4 # Python小练习：解决strftime()中国时区乱码问题
5 import datetime
6 import dateutil.tz
7 now = datetime.datetime.now(dateutil.tz.tzlocal())
8 timestamp = now.strftime('%Y-%m-%d %A %H:%M:%S %Z').encode('latin-1').decode('GBK')
9 print('方法1: datetime模块 | ', timestamp)
10 # -----
11 import time
12 timestamp = time.strftime('%Y-%m-%d %A %H:%M:%S %Z').encode('latin-1').decode('GBK')
13 print('方法1: time模块 | ', timestamp)
14 # -----
15 import time
16 import locale
17 locale.setlocale(locale.LC_ALL, '')
18 timestamp = time.strftime('%Y-%m-%d %A %H:%M:%S %Z')
19 print('方法2: time模块 | ', timestamp)
20
21 # %y 两位数的年份表示(00-99)
22 # %Y 四位数的年份表示(000-9999)
23 # %m 月份(01-12)
24 # %d 月内中的一天(0-31)
25 # %H 24小时制小时数(0-23)
26 # %I 12小时制小时数(01-12)
27 # %M 分钟数(00=59)
28 # %S 秒(00=59)
29 # %a 本地简化星期名称
30 # %A 本地完整星期名称
31 # %b 本地简化的月份名称
32 # %B 本地完整的月份名称
33 # %c 本地相应的日期表示和时间表示
34 # %j 年内的一天(001-366)
35 # %p 本地A. M. 或P. M. 的等价符
36 # %U 一年中的星期数(00=53)星期天为星期的开始
```

```
37 # %w 星期(0-6)，星期天为星期的开始
38 # %W 一年中的星期数(00-53)星期一为星期的开始
39 # %x 本地相应的日期表示
40 # %X 本地相应的时间表示
41 # %Z 当前时区的名称
```

## 2. 结果

方法1: datetime模块		2023-04-18 Tuesday 18:10:43 中国标准时间
方法1: time模块		2023-04-18 Tuesday 18:10:43 中国标准时间
方法2: time模块		2023-04-18 星期二 18:10:43 中国标准时间

乱码问题是在Windows下出现的。例如："中国标准时间"六个汉字成了："ÖÐ¹ú±ê×¼Ê±¼ä"。

如果用Ubuntu，则直接使用如下代码：

```
1 import datetime
2 import dateutil.tz
3 now = datetime.datetime.now(dateutil.tz.tzlocal())
4 timestamp = now.strftime('%Y-%m-%d %A %H:%M:%S %Z')
5 print(timestamp)
```

结果：

2023-04-18 Tuesday 18:16:42 CST

其中CST表示中国标准时间(China Standard Time)。