Python时间处理方法

在从事Python相关工作中，Python时间模块与日常工作密切相关，因此，掌握时间处理模块及方法十分重要，以下是相关内容介绍：

**1. time模块**

time模块下有很多函数可以日期和时间的处理，如time.time()用于获取当前时间戳，localtime()将浮点数的时间戳向时间元组转换，asctime()和strftime()可以进行时间和日期的格式化，time.sleep()可以进行时间定时等。

实例：

import time #引入time模块

localtime = time.localtime(time.time()) #返回时间元组

newtime= time.asctime(localtime) #获取格式化时间

time.sleep(3600) #定时1个小时

**2.datetime模块**

datetime模块是Python时间处理的又一模块，主要包括date、time和datetime三个子模块，是一个更亲民、方便的模块。

实例：

import datetime #引入datetime模块

print datetime.datetime.now() #获取当前时间

**3. 时间加减处理**

datetime模块中的timedelta子模块在处理时间加减方面十分好用，可以让时间处理变得很灵活。

实例：

import datetime #引入datetime模块

d1=datetime.datetime(2018，4，19) #定义变量获取定义时间

d2= datetime.datetime(2018，2，19) #定义变量获取定义时间

print d1+datetime.timedelta(days=100) # 输出规定时间之后的第100天日期

print d1-datetime.timedelta(days=100) # 输出规定时间之前的第100天日期

print d1-d2 #d1和d2日期相差天数

**4. 日历**

日历在时间处理中是非常重要的一块，可以采用Python中的calendar模块实现。

实例：

import calendar #引入calendar模块

print calendar.month(2018，4) 输出2018年4月日历

**5. 字符串和日期的转换**

字符串和日期的转换需要用到strftime()模块和Python中时间日期格式化符号，常用时间日期格式化符号有：

%y 两位数的年份表示（00-99）

%Y 四位数的年份表示（000-9999）

%m 月份（01-12）

%d 月内中的一天（0-31）

%H 24小时制小时数（0-23）

%I 12小时制小时数（01-12）

%M 分钟数（00=59）

%S 秒（00-59）

把字符串转换成日期

import datetime #引入datetime模块

time= datetime.strptime('2018-4-19 11:19:59'，'%Y-%m-%d %H:%M:%S') #把字符串转换成时间

print time #输出时间信息

把日期转换成字符串

import datetime #引入datetime模块

str = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S') #获取当前时间并转化成字符串

print str #输出字符串信息

以上是Python时间处理常用模块和基础用法，除此之外，还有很多高深用法，感兴趣或者需要用到的可以深入学习！