python时间处理之datetime

标签： pythondateimportstringc

2012-09-12 23:21 51545人阅读 评论(0) 收藏 举报

分类： 玩转 Python（13）

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录(?)[+]

运行环境:Python3.4

[python] view plain copy

# -\*- coding: utf-8 -\*-

#datetime类

#datetime是date与time的结合体，包括date与time的所有信息。

#它的构造函数如下：

#datetime. datetime (year, month, day[ , hour[ , minute[ , second[ , microsecond[ , tzinfo] ] ] ] ] )

#各参数的含义与date、time的构造函数中的一样，要注意参数值的范围。

# 1. datetime类定义的类属性与方法：

#datetime.min、datetime.max：datetime所能表示的最小值与最大值；

#print: datetime.max: 9999-12-31 23:59:59.999999

#print: datetime.min: 0001-01-01 00:00:00

from datetime import \*

import time

print ('datetime.max:' +str(datetime.max ))

print ('datetime.min:' +str(datetime.min))

#datetime.resolution：datetime最小单位；

#print: datetime.resolution: 0:00:00.000001

print ('datetime.resolution:' + str(datetime.resolution ))

#datetime.today()：返回一个表示当前本地时间的datetime对象；

#print: today(): 2012-09-12 19:37:50.721000

print ('today():' +str(datetime.today() ))

#datetime.now([tz])：返回一个表示当前本地时间的datetime对象，如果提供了参数tz，则获取tz参数所指时区的本地时间；

#print: now(): 2012-09-12 19:37:50.738000

print ('now():'+str( datetime.now() ))

#datetime.utcnow()：返回一个当前utc时间的datetime对象；

#print: 2012-09-12 11:37:50.739000

print ('utcnow():' +str(datetime.utcnow() ))

#datetime.fromtimestamp(timestamp[, tz])：根据时间戮创建一个datetime对象，参数tz指定时区信息；

#print: fromtimestamp(tmstmp): 2012-09-12 19:37:50.741000

print ('fromtimestamp(tmstmp):' +str(datetime.fromtimestamp(time.time()) ))

#datetime.utcfromtimestamp(timestamp)：根据时间戮创建一个datetime对象；

#print: utcfromtimestamp(tmstmp): 2012-09-12 11:37:50.742000

print ('utcfromtimestamp(tmstmp):' +str(datetime.utcfromtimestamp(time.time())) )

#datetime.combine(date, time)：根据date和time，创建一个datetime对象；

#print: datetime.combine(date,time): 2012-09-12 19:46:05

d = date(2012,9,12)

from datetime import \*

t = time(19,46,5)

print ('datetime.combine(date,time): '+str(datetime.combine(d,t)))

#datetime.strptime(date\_string, format)：将格式字符串转换为datetime对象；

#print: 2007-03-04 21:08:12

print (datetime.strptime("2007-03-04 21:08:12", "%Y-%m-%d %H:%M:%S"))

#2. datetime类提供的实例方法与属性

dt = datetime.strptime("2012-09-12 21:08:12", "%Y-%m-%d %H:%M:%S")

#print: 2012 9 12 21 8 12 0 None

print (dt.year)

print(dt.month)

print(dt.day)

print(dt.hour)

print(dt.minute)

print(dt.second)

print(dt.microsecond)

print(dt.tzinfo)

print (dt.date())

print (dt.time())

print (dt.replace(year = 2013))

print (dt.timetuple())

print (dt.utctimetuple())

print (dt.toordinal())

print (dt.weekday())

print (dt.isocalendar())

#print dt.isoformat([sep])

#datetime. ctime ()：返回一个日期时间的C格式字符串，等效于time.ctime(time.mktime(dt.timetuple()))；

#3. 格式字符串

# datetime. strftime (format)

# %a 星期的简写。如 星期三为Web

# %A 星期的全写。如 星期三为Wednesday

# %b 月份的简写。如4月份为Apr

# %B月份的全写。如4月份为April

# %c: 日期时间的字符串表示。（如： 04/07/10 10:43:39）

# %d: 日在这个月中的天数（是这个月的第几天）

# %f: 微秒（范围[0,999999]）

# %H: 小时（24小时制，[0, 23]）

# %I: 小时（12小时制，[0, 11]）

# %j: 日在年中的天数 [001,366]（是当年的第几天）

# %m: 月份（[01,12]）

# %M: 分钟（[00,59]）

# %p: AM或者PM

# %S: 秒（范围为[00,61]，为什么不是[00, 59]，参考python手册~\_~）

# %U: 周在当年的周数当年的第几周），星期天作为周的第一天

# %w: 今天在这周的天数，范围为[0, 6]，6表示星期天

# %W: 周在当年的周数（是当年的第几周），星期一作为周的第一天

# %x: 日期字符串（如：04/07/10）

# %X: 时间字符串（如：10:43:39）

# %y: 2个数字表示的年份

# %Y: 4个数字表示的年份

# %z: 与utc时间的间隔 （如果是本地时间，返回空字符串）

# %Z: 时区名称（如果是本地时间，返回空字符串）

# %%: %% => %

dt = datetime.now()

#print: (%Y-%m-%d %H:%M:%S %f): 2012-09-12 23:04:27 145000

print ('(%Y-%m-%d %H:%M:%S %f): '+ str(dt.strftime('%Y-%m-%d %H:%M:%S %f')))

#print: (%Y-%m-%d %H:%M:%S %p): 12-09-12 11:04:27 PM

print ('(%Y-%m-%d %H:%M:%S %p): '+str(dt.strftime('%y-%m-%d %I:%M:%S %p')))

#print: %a: Wed

print ('%%a: %s ' % dt.strftime('%a'))

#print: %A: Wednesday

print ('%%A: %s ' % dt.strftime('%A'))

#print: %b: Sep

print ('%%b: %s ' % dt.strftime('%b'))

#print: %B: September

print ('%%B: %s ' % dt.strftime('%B'))

#print: 日期时间%c: 09/12/12 23:04:27

print ('日期时间%%c: %s ' % dt.strftime('%c'))

#print: 日期%x：09/12/12

print ('日期%%x：%s ' % dt.strftime('%x'))

#print: 时间%X：23:04:27

print ('时间%%X：%s ' % dt.strftime('%X'))

#print: 今天是这周的第3天

print ('今天是这周的第%s天 ' % dt.strftime('%w'))

#print: 今天是今年的第256天

print ('今天是今年的第%s天 ' % dt.strftime('%j'))

#print: 今周是今年的第37周

print ('今周是今年的第%s周 ' % dt.strftime('%U'))

运行结果

[python] view plain copy

atetime.max:9999-12-31 23:59:59.999999

datetime.min:0001-01-01 00:00:00

datetime.resolution:0:00:00.000001

today():2014-05-04 15:58:18.141186

now():2014-05-04 15:58:18.193146

utcnow():2014-05-04 07:58:18.243958

fromtimestamp(tmstmp):2014-05-04 15:58:18.291558

utcfromtimestamp(tmstmp):2014-05-04 07:58:18.342550

datetime.combine(date,time): 2012-09-12 19:46:05

2007-03-04 21:08:12

2012

9

12

21

8

12

0

None

2012-09-12

21:08:12

2013-09-12 21:08:12

time.struct\_time(tm\_year=2012, tm\_mon=9, tm\_mday=12, tm\_hour=21, tm\_min=8, tm\_sec=12, tm\_wday=2, tm\_yday=256, tm\_isdst=-1)

time.struct\_time(tm\_year=2012, tm\_mon=9, tm\_mday=12, tm\_hour=21, tm\_min=8, tm\_sec=12, tm\_wday=2, tm\_yday=256, tm\_isdst=0)

734758

2

(2012, 37, 3)

(%Y-%m-%d %H:%M:%S %f): 2014-05-04 15:58:19 326295

(%Y-%m-%d %H:%M:%S %p): 14-05-04 03:58:19 PM

%a: Sun

%A: Sunday

%b: May

%B: May

日期时间%c: Sun May 4 15:58:19 2014

日期%x：05/04/14

时间%X：15:58:19

今天是这周的第0天

今天是今年的第124天

今周是今年的第18周

运行环境:Python2.x

[python] view plain copy

# -\*- coding: utf-8 -\*-

#datetime类

#datetime是date与time的结合体，包括date与time的所有信息。

#它的构造函数如下：

#datetime. datetime (year, month, day[ , hour[ , minute[ , second[ , microsecond[ , tzinfo] ] ] ] ] )

#各参数的含义与date、time的构造函数中的一样，要注意参数值的范围。

# 1. datetime类定义的类属性与方法：

#datetime.min、datetime.max：datetime所能表示的最小值与最大值；

#print: datetime.max: 9999-12-31 23:59:59.999999

#print: datetime.min: 0001-01-01 00:00:00

from datetime import \*

import time

print 'datetime.max:' , datetime.max

print 'datetime.min:' , datetime.min

#datetime.resolution：datetime最小单位；

#print: datetime.resolution: 0:00:00.000001

print 'datetime.resolution:' , datetime.resolution

#datetime.today()：返回一个表示当前本地时间的datetime对象；

#print: today(): 2012-09-12 19:37:50.721000

print 'today():' , datetime.today()

#datetime.now([tz])：返回一个表示当前本地时间的datetime对象，如果提供了参数tz，则获取tz参数所指时区的本地时间；

#print: now(): 2012-09-12 19:37:50.738000

print 'now():' , datetime.now()

#datetime.utcnow()：返回一个当前utc时间的datetime对象；

#print: 2012-09-12 11:37:50.739000

print 'utcnow():' , datetime.utcnow()

#datetime.fromtimestamp(timestamp[, tz])：根据时间戮创建一个datetime对象，参数tz指定时区信息；

#print: fromtimestamp(tmstmp): 2012-09-12 19:37:50.741000

print 'fromtimestamp(tmstmp):' , datetime.fromtimestamp(time.time())

#datetime.utcfromtimestamp(timestamp)：根据时间戮创建一个datetime对象；

#print: utcfromtimestamp(tmstmp): 2012-09-12 11:37:50.742000

print 'utcfromtimestamp(tmstmp):' , datetime.utcfromtimestamp(time.time())

#datetime.combine(date, time)：根据date和time，创建一个datetime对象；

#print: datetime.combine(date,time): 2012-09-12 19:46:05

d = date(2012,9,12)

from datetime import \*

t = time(19,46,5)

print 'datetime.combine(date,time): ',datetime.combine(d,t)

#datetime.strptime(date\_string, format)：将格式字符串转换为datetime对象；

#print: 2007-03-04 21:08:12

print datetime.strptime("2007-03-04 21:08:12", "%Y-%m-%d %H:%M:%S")

#2. datetime类提供的实例方法与属性

dt = datetime.strptime("2012-09-12 21:08:12", "%Y-%m-%d %H:%M:%S")

#print: 2012 9 12 21 8 12 0 None

print dt.year,dt.month,dt.day,dt.hour,dt.minute,dt.second,dt.microsecond,dt.tzinfo

print dt.date()

print dt.time()

print dt.replace(year = 2013)

print dt.timetuple()

print dt.utctimetuple()

print dt.toordinal()

print dt.weekday()

print dt.isocalendar()

#print dt.isoformat([sep])

#datetime. ctime ()：返回一个日期时间的C格式字符串，等效于time.ctime(time.mktime(dt.timetuple()))；

#3. 格式字符串

# datetime. strftime (format)

# %a 星期的简写。如 星期三为Web

# %A 星期的全写。如 星期三为Wednesday

# %b 月份的简写。如4月份为Apr

# %B月份的全写。如4月份为April

# %c: 日期时间的字符串表示。（如： 04/07/10 10:43:39）

# %d: 日在这个月中的天数（是这个月的第几天）

# %f: 微秒（范围[0,999999]）

# %H: 小时（24小时制，[0, 23]）

# %I: 小时（12小时制，[0, 11]）

# %j: 日在年中的天数 [001,366]（是当年的第几天）

# %m: 月份（[01,12]）

# %M: 分钟（[00,59]）

# %p: AM或者PM

# %S: 秒（范围为[00,61]，为什么不是[00, 59]，参考python手册~\_~）

# %U: 周在当年的周数当年的第几周），星期天作为周的第一天

# %w: 今天在这周的天数，范围为[0, 6]，6表示星期天

# %W: 周在当年的周数（是当年的第几周），星期一作为周的第一天

# %x: 日期字符串（如：04/07/10）

# %X: 时间字符串（如：10:43:39）

# %y: 2个数字表示的年份

# %Y: 4个数字表示的年份

# %z: 与utc时间的间隔 （如果是本地时间，返回空字符串）

# %Z: 时区名称（如果是本地时间，返回空字符串）

# %%: %% => %

dt = datetime.now()

#print: (%Y-%m-%d %H:%M:%S %f): 2012-09-12 23:04:27 145000

print '(%Y-%m-%d %H:%M:%S %f): ', dt.strftime('%Y-%m-%d %H:%M:%S %f')

#print: (%Y-%m-%d %H:%M:%S %p): 12-09-12 11:04:27 PM

print '(%Y-%m-%d %H:%M:%S %p): ', dt.strftime('%y-%m-%d %I:%M:%S %p')

#print: %a: Wed

print '%%a: %s ' % dt.strftime('%a')

#print: %A: Wednesday

print '%%A: %s ' % dt.strftime('%A')

#print: %b: Sep

print '%%b: %s ' % dt.strftime('%b')

#print: %B: September

print '%%B: %s ' % dt.strftime('%B')

#print: 日期时间%c: 09/12/12 23:04:27

print '日期时间%%c: %s ' % dt.strftime('%c')

#print: 日期%x：09/12/12

print '日期%%x：%s ' % dt.strftime('%x')

#print: 时间%X：23:04:27

print '时间%%X：%s ' % dt.strftime('%X')

#print: 今天是这周的第3天

print '今天是这周的第%s天 ' % dt.strftime('%w')

#print: 今天是今年的第256天

print '今天是今年的第%s天 ' % dt.strftime('%j')

#print: 今周是今年的第37周

print '今周是今年的第%s周 ' % dt.strftime('%U')