小白必备时间操作练习

94 当前时间的浮点数表

当前时间浮点数

复制
```
import time
seconds = time.time()
seconds
# 1582341559.0950701
```

95 当前时间的时间数组

复制
```
import time
seconds = time.time()
local_time = time.localtime(seconds)

local_time
# time.struct_time(tm_year=2020, tm_mon=2, tm_mday=22, tm_hour=11, tm_min
=19, tm_sec=19, tm_wday=5, tm_yday=53, tm_isdst=0)
```

96 当前时间转为时间字符串

`time` 类 `asctime` 方法，转换 `struct_time` 为时间字符串

复制
```
import time
seconds = time.time()
local_time = time.localtime(seconds)
str_time = time.asctime(local_time)
str_time
# 'Sat Feb 22 11:19:19 2020'
```

97 格式化时间字符串

`time` 类 `strftime` 方法，按照时间格式要求，格式化 `struct_time` 为时间字符串

复制
```
import time
seconds = time.time()
local_time = time.localtime(seconds)
format_time = time.strftime('%Y-%m-%d %H:%M:%S',local_time)
format_time
# '2020-02-22 11:19:19'
```

98 字符时间转时间数组

`time` 类 `strptime` 方法，解析(`parse`) 输入的时间字符串为 `struct_time` 类型的时间。

复制
```
import time
seconds = time.time()
local_time = time.localtime(seconds)
format_time = time.strftime('%Y-%m-%d %H:%M:%S',local_time)
str_to_struct = time.strptime(format_time,'%Y-%m-%d %H:%M:%S')
str_to_struct
# time.struct_time(tm_year=2020, tm_mon=2, tm_mday=22, tm_hour=11, tm_min
=19, tm_sec=19, tm_wday=5, tm_yday=53, tm_isdst=-1)
```

注意：第二个参数的时间格式，要匹配上第一个参数的时间格式。

如果前后格式不匹配，执行下面这行代码：

复制
```
str_to_struct = time.strptime('2020-02-22 11:19:19','%Y/%m/%d %H:%M:%S')
```

就会抛出异常：

复制
```
ValueError: time data '2020-02-22 11:19:19' does not match format '%Y/%m/
%d %H:%M:%S'
```

记住常用的时间格式：

复制
```
    %Y  年
    %m  月 取值 [01,12]
    %d  天 取值 [01,31]
    %H  小时 取值 [00,23]
    %M  分钟 取值 [00,59]
    %S  秒 取值 [00,61]
```

99 打印当前日期

从 `datetime` 模块中，依次导入类 ： `date` , `datetime` , `time` , `timedelta`

复制
```
In [32]: from datetime import date, datetime, time, timedelta
In [35]: tod = date.today()
In [36]: tod
Out[36]: datetime.date(2020, 2, 22)
```

100 当前日期字符串

复制
```
In [32]: from datetime import date, datetime, time, timedelta
In [48]: str_date = date.strftime(tod,'%Y-%m-%d')
In [49]: str_date
Out[49]: '2020-02-22'
```

101 字符日期转日期

`date` 类里没有 `strptime` 方法，它的子类 `datetime` 才有解析字符串日期的方法 `strptime`

复制
```
In [32]: from datetime import date, datetime, time, timedelta
In [43]: str_to_date = datetime.strptime('2020-02-22','%Y-%m-%d')
```

```
In [44]: str_to_date
Out[44]: datetime.datetime(2020, 2, 22, 0, 0)
```

这样默认转化后的类为 `datetime`

102 打印当前时间

复制

```
In [32]: from datetime import date, datetime, time, timedelta
In [51]: right = datetime.now()

In [52]: right
Out[52]: datetime.datetime(2020, 2, 22, 15, 12, 33, 96095)
```

103 当前时间转字符串显示

复制

```
In [32]: from datetime import date, datetime, time, timedelta
In [51]: right = datetime.now()
In [57]: str_time = datetime.strftime(right,'%Y-%m-%d %H:%M:%S')

In [58]: str_time
Out[58]: '2020-02-22 15:12:33'
```

104 字符时间转时间类型

复制

```
In [32]: from datetime import date, datetime, time, timedelta
In [60]: str_to_time = datetime.strptime('2020-02-22 15:12:33','%Y-%m-%d
%H:%M:%S')

In [61]: str_to_time
Out[61]: datetime.datetime(2020, 2, 22, 15, 12, 33)
```

105 计算还有多久是女朋友生日

求两个 `datetime` 类型值的差，返回差几天：days，差几小时：hours 等。

相减的两个时间，不能一个为 `date` 类型，一个为 `datetime` 类型，尽管两个类型是父子关系。

案例：计算还有几天是女朋友生日

复制

```
from datetime import datetime,date

def get_days_girlfriend(birthday:str)->int:
    import re
    splits = re.split(r'[-.\s+/]',birthday)
    splits = [s for s in splits if s] # 去掉空格字符
    if len(splits) < 3:
        raise ValueError('输入格式不正确，至少包括年月日')
    splits = splits[:3] # 只截取年月日
    birthday = datetime.strptime('-'.join(splits),'%Y-%m-%d')
    tod = date.today()
    delta = birthday.date() - tod
    return delta.days
```

输入时间格式适配三种分隔符：

`-`

`/`

以及 1 个或多个连续空格

复制

```
In [71]: get_days_girlfriend('2020-05-20')
Out[71]: 88

In [72]: get_days_girlfriend('2020/5/20')
Out[72]: 88

In [93]: get_days_girlfriend('2021 1    9')
Out[93]: 322

In [99]: get_days_girlfriend('2020/5/20 10:00')
Out[99]: 88
```

输入时间字符串必须包括年月日，忽略时间值。

复制

```
In [100]: get_days_girlfriend('2020/5')

<ipython-input-98-04c0a68cbd9a> in get_days_girlfriend(birthday)
      4     splits = [s for s in splits if s] # 去掉空格字符
      5     if len(splits) < 3:
----> 6         raise ValueError('输入格式不正确，至少包括年月日')
      7     splits = splits[:3] # 只截取年月日
      8     birthday = datetime.strptime('-'.join(splits),'%Y-%m-%d')

ValueError: 输入格式不正确，至少包括年月日
```

106 绘制年的日历图

复制

```
import calendar
from datetime import date
mydate = date.today()
year_calendar_str = calendar.calendar(2019)
print(f"{mydate.year}年的日历图:{year_calendar_str}\n")
```

打印结果：

复制

```
2019

      January                   February                  March
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
    1  2  3  4  5  6                   1  2  3                   1  2  3
 7  8  9 10 11 12 13       4  5  6  7  8  9 10       4  5  6  7  8  9 10
14 15 16 17 18 19 20      11 12 13 14 15 16 17      11 12 13 14 15 16 17
21 22 23 24 25 26 27      18 19 20 21 22 23 24      18 19 20 21 22 23 24
28 29 30 31               25 26 27 28               25 26 27 28 29 30 31

       April                     May                       June
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
```

```
 1  2  3  4  5  6  7              1  2  3  4  5                     1  2
 8  9 10 11 12 13 14       6  7  8  9 10 11 12       3  4  5  6  7  8  9
15 16 17 18 19 20 21      13 14 15 16 17 18 19      10 11 12 13 14 15 16
22 23 24 25 26 27 28      20 21 22 23 24 25 26      17 18 19 20 21 22 23
29 30                     27 28 29 30 31            24 25 26 27 28 29 30

        July                    August                  September
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
 1  2  3  4  5  6  7                  1  2  3  4                        1
 8  9 10 11 12 13 14       5  6  7  8  9 10 11       2  3  4  5  6  7  8
15 16 17 18 19 20 21      12 13 14 15 16 17 18       9 10 11 12 13 14 15
22 23 24 25 26 27 28      19 20 21 22 23 24 25      16 17 18 19 20 21 22
29 30 31                  26 27 28 29 30 31         23 24 25 26 27 28 29
                                                    30

      October                   November                  December
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
    1  2  3  4  5  6                  1  2  3                        1
 7  8  9 10 11 12 13       4  5  6  7  8  9 10       2  3  4  5  6  7  8
14 15 16 17 18 19 20      11 12 13 14 15 16 17       9 10 11 12 13 14 15
21 22 23 24 25 26 27      18 19 20 21 22 23 24      16 17 18 19 20 21 22
28 29 30 31               25 26 27 28 29 30         23 24 25 26 27 28 29
                                                    30 31
```

107 月的日历图

复制

```python
import calendar
from datetime import date

mydate = date.today()
month_calendar_str = calendar.month(mydate.year, mydate.month)

print(f"{mydate.year}年-{mydate.month}月的日历图:{month_calendar_str}\n")
```

打印结果：

复制

```
December 2019
Mo Tu We Th Fr Sa Su
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
```

108 判断是否为闰年

复制

```python
import calendar
from datetime import date

mydate = date.today()
is_leap = calendar.isleap(mydate.year)
print_leap_str = "%s年是闰年" if is_leap else "%s年不是闰年\n"
print(print_leap_str % mydate.year)
```

打印结果：

复制

```
2019年不是闰年
```

109 判断月有几天

复制

```python
import calendar
from datetime import date

mydate = date.today()
weekday, days = calendar.monthrange(mydate.year, mydate.month)
print(f'{mydate.year}年-{mydate.month}月的第一天是那一周的第{weekday}天\n')
print(f'{mydate.year}年-{mydate.month}月共有{days}天\n')
```

打印结果：

复制

```
2019年-12月的第一天是那一周的第6天

2019年-12月共有31天
```

110 月的第一天

复制

```python
from datetime import date
mydate = date.today()
month_first_day = date(mydate.year, mydate.month, 1)
print(f"当月第一天:{month_first_day}\n")
```

打印结果：

复制

```
当月第一天:2019-12-01
```

111 月的最后一天

复制

```python
from datetime import date
import calendar
mydate = date.today()
_, days = calendar.monthrange(mydate.year, mydate.month)
month_last_day = date(mydate.year, mydate.month, days)
print(f"当月最后一天:{month_last_day}\n")
```

打印结果：

复制

```
当月最后一天:2019-12-31
```

112 输入日期， 判断这一天是这一年的第几天 ？

使用 `datetime` 模块，提取日期 `date` 对象，调用 `timetuple()` 方法，返回 一个 `struct_time` 对象，属性 `tm_yday` 便是这一年的第几天

复制

```python
from datetime import datetime
```

```
def get_day_of_year(y,m,d):
    return datetime(y,m,d).date().timetuple().tm_yday

In [45]: get_day_of_year(2020,2,1)
Out[45]: 32
In [46]: get_day_of_year(2019,12,31)
Out[46]: 365
```

下一章

互动评论

说点什么

评论

The Scrapper 2 个月前

练习

鼓掌

存