nsd_1908_py02_day01

time模块

时间表示方式

- 时间戳：表示的是从1970年1月1日00:00:00开始按秒计算的偏移量
- UTC时间：世界协调时。格林威治天文时间，世界标准时间。
- struct_time：9元组(年，月，日，时，分，秒，一周中的第几天，一年中的第几天，是否使用夏季节约时间)

```
>>> import time
>>> time.time()  # 运行命令时的时间戳
1578446773.7676122
>>> time.ctime()  # UTC时间
'Wed Jan  8 09:27:49 2020'
>>> time.localtime()  # struct_time
time.struct_time(tm_year=2020, tm_mon=1, tm_mday=8, tm_hour=9, tm_min=29, tm_sec=13,
tm_wday=2, tm_yday=8, tm_isdst=0)
>>> t = time.localtime()
>>> t.tm_year
2020

>>> time.sleep(3)  # 睡眠3秒
>>> time.strftime('%Y-%m-%d %H:%M:%S')
'2020-01-08 09:50:29'
>>> time.strftime('%a')
'Wed'
>>> time.strftime('%A')
'Wednesday'

# 将时间字符串转为9元组时间格式
>>> t1 = time.strptime('2020-01-08 10:30:00', '%Y-%m-%d %H:%M:%S')
>>> t1
time.struct_time(tm_year=2020, tm_mon=1, tm_mday=8, tm_hour=10, tm_min=30, tm_sec=0,
tm_wday=2, tm_yday=8, tm_isdst=-1)
>>> t = time.localtime()
>>> t
time.struct_time(tm_year=2020, tm_mon=1, tm_mday=8, tm_hour=10, tm_min=15, tm_sec=28,
tm_wday=2, tm_yday=8, tm_isdst=0)
>>> t > t1
False
```

datetime模块

```
>>> import datetime
>>> datetime.datetime.now()
datetime.datetime(2020, 1, 8, 10, 54, 52, 717452)
```

```
>>> from datetime import datetime
>>> datetime.now()  # 返回的是(年，月，日，时，分，秒，毫秒)
datetime.datetime(2020, 1, 8, 10, 56, 15, 757867)
>>> t.year
2020
>>> t.month
1
>>> t.day
8
>>> t.hour
10
>>> t.minute
57
>>> t.second
25
>>> t.microsecond
975479

>>> t.strftime('%Y/%m/%d %H:%M:%S')  # 转成时间字符串
'2020/01/08 10:57:25'

# 时间字符串转换成datetime对象
>>> datetime.strptime('2020-1-8 9:00:00', '%Y-%m-%d %H:%M:%S')
datetime.datetime(2020, 1, 8, 9, 0)

# 通过timedelta计算时间差额
>>> from datetime import datetime, timedelta
>>> days = timedelta(days=100, hours=1)  # 定义100天零1小时
>>> t = datetime.now()
>>> t - days  # 100天1小时之前的时间
datetime.datetime(2019, 9, 30, 10, 33, 23, 144003)
>>> t + days  # 100天1小时之后的时间
datetime.datetime(2020, 4, 17, 12, 33, 23, 144003)
```

触发异常

- 通过raise指定触发的异常
- 通过assert触发AssertionError异常。

os模块

```
>>> import os
>>> os.listdir()  # ls
>>> os.mkdir('/tmp/aaa')  # mkdir /tmp/aaa
>>> os.makedirs('/tmp/nsd1908/demo')  # mkdir -p /tmp/nsd1908/demo
>>> os.listdir('/tmp/nsd1908')  # ls /tmp/nsd1908
['demo']
>>> os.chdir('/tmp/nsd1908')  # cd /tmp/nsd1908
>>> os.getcwd()  # pwd
```

```
'/tmp/nsd1908'
>>> import shutil
>>> shutil.copy('/etc/hosts', 'hosts')  # cp /etc/hosts hosts
>>> os.mknod('test.txt')  # touch test.txt
>>> os.symlink('/etc/passwd', 'mima')  # ln -s /etc/passwd mima
>>> os.listdir()
['demo', 'hosts', 'test.txt', 'mima']

>>> os.chmod('hosts', 0o600)  # chmod 600 hosts
>>> os.chmod('hosts', 420)  # chmod 644 hosts

>>> os.listdir()
['demo', 'hosts', 'test.txt', 'mima']
>>> os.path.abspath('demo')  # 返回绝对路径
'/tmp/nsd1908/demo'
>>> os.path.isabs('aaa/bbb/ccc')  # 路径是绝对路径吗？
False
>>> os.path.isabs('/aaa/bbb/ccc')
True
>>> os.path.isfile('hosts')  # hosts存在，并且是文件吗？
>>> os.path.ismount('/etc')  # 是挂载点吗？
>>> os.path.isdir('demo')   # 存在，并且是目录吗？
>>> os.path.islink('mima')  # 存在，并且是软链接吗？
>>> os.path.exists('/etc/hostname')  # 存在吗？

>>> os.path.basename('/tmp/nsd1908/hosts')
'hosts'
>>> os.path.dirname('/tmp/nsd1908/hosts')
'/tmp/nsd1908'
>>> os.path.split('/tmp/nsd1908/hosts')
('/tmp/nsd1908', 'hosts')
>>> os.path.join('/tmp/nsd1908', 'hosts')
'/tmp/nsd1908/hosts'
```

os.walk

- 递归遍历目录下的内容

```
(nsd1908) [root@room8pc16 nsd1908]# touch aaa/{a,b,c}.txt
(nsd1908) [root@room8pc16 nsd1908]# touch bbb/{d,e,f}.txt
(nsd1908) [root@room8pc16 nsd1908]# touch demo/{x,y,z}.txt
(nsd1908) [root@room8pc16 nsd1908]# mkdir aaa bbb
(nsd1908) [root@room8pc16 nsd1908]# cd ..
(nsd1908) [root@room8pc16 tmp]# ls -R nsd1908/
nsd1908/:
aaa  bbb  demo  hosts  mima  test.txt

nsd1908/aaa:
a.txt  b.txt  c.txt

nsd1908/bbb:
d.txt  e.txt  f.txt
```

```
nsd1908/demo:
x.txt  y.txt  z.txt

>>> os.chdir('/tmp/')
>>> os.listdir('nsd1908')
['demo', 'hosts', 'test.txt', 'mima', 'aaa', 'bbb']

>>> list(os.walk('nsd1908'))
[('nsd1908', ['demo', 'aaa', 'bbb'], ['hosts', 'test.txt', 'mima']), ('nsd1908/demo', [],
['x.txt', 'y.txt', 'z.txt']), ('nsd1908/aaa', [], ['a.txt', 'b.txt', 'c.txt']),
('nsd1908/bbb', [], ['d.txt', 'e.txt', 'f.txt'])]

>>> alist = list(os.walk('nsd1908'))
>>> len(alist)
4
>>> alist[0]
('nsd1908', ['demo', 'aaa', 'bbb'], ['hosts', 'test.txt', 'mima'])
>>> alist[1]
('nsd1908/demo', [], ['x.txt', 'y.txt', 'z.txt'])
>>> alist[2]
('nsd1908/aaa', [], ['a.txt', 'b.txt', 'c.txt'])
>>> alist[3]
('nsd1908/bbb', [], ['d.txt', 'e.txt', 'f.txt'])
# 经过分析，列表由元组构成
# 每个元组都有相同的结构：(路径字符串，路径下目录列表，路径下文件列表)

>>> for data in alist:
...   print(data)
...
('nsd1908', ['demo', 'aaa', 'bbb'], ['hosts', 'test.txt', 'mima'])
('nsd1908/demo', [], ['x.txt', 'y.txt', 'z.txt'])
('nsd1908/aaa', [], ['a.txt', 'b.txt', 'c.txt'])
('nsd1908/bbb', [], ['d.txt', 'e.txt', 'f.txt'])

>>> for path, folders, files in os.walk('nsd1908'):
...   print(path, folders, files)
...
nsd1908 ['demo', 'aaa', 'bbb'] ['hosts', 'test.txt', 'mima']
nsd1908/demo [] ['x.txt', 'y.txt', 'z.txt']
nsd1908/aaa [] ['a.txt', 'b.txt', 'c.txt']
nsd1908/bbb [] ['d.txt', 'e.txt', 'f.txt']

>>> for path, folders, files in os.walk('nsd1908'):
...   print('%s:' % path)
...   for dir in folders:
...     print(dir, end='\t')
...   for file in files:
...     print(file, end='\t')
...   print('\n')
...
nsd1908:
demo    aaa     bbb     hosts   test.txt        mima
```

```
nsd1908/demo:
x.txt   y.txt   z.txt

nsd1908/aaa:
a.txt   b.txt   c.txt

nsd1908/bbb:
d.txt   e.txt   f.txt
```

pickle模块

- pickle模块可以将任意的数据类型写到文件中，还可以再无损的取出

```
>>> import pickle
>>> shopping_list = ['apple', 'banana', 'egg']
>>> with open('/tmp/a.txt', 'wb') as fobj:
...   pickle.dump(shopping_list, fobj)

>>> import pickle
>>> with open('/tmp/a.txt', 'rb') as fobj:
...   alist = pickle.load(fobj)
...
>>> type(alist)
<class 'list'>
>>> alist
['apple', 'banana', 'egg']
```

记账程序

日期　　收入　支出　余额　备注

2020-1-8 0 0 10000 init data

2020-1-8 10000 0 20000 salary

2020-1-8 0 100 19000 buy ticket

```
# 记账时，可以把整个记账本用一个列表来表示，每一笔记录又是一个小列表
>>> content = []
>>> line = ['2020-01-08', 0, 0, 10000, 'init data']
>>> content.append(line)
>>> content
[['2020-01-08', 0, 0, 10000, 'init data']]
>>> amount = 10000
>>> content[-1]  # 在大列表中取出最新的一笔记录
['2020-01-08', 0, 0, 10000, 'init data']
>>> content[-1][-2]   # 最新一笔记录还是列表，可以继续通过下标取元素
10000
>>> line = ['2020-01-08', 10000, 0, 20000, 'salary']
>>> content.append(line)
>>> content
[['2020-01-08', 0, 0, 10000, 'init data'], ['2020-01-08', 10000, 0, 20000, 'salary']]
```