

nsd1904_py02_day01

time模块

时间的表示方法

- 时间戳：1970-1-1 0:0:0 到某一时间点之间的秒数
- UTC：世界协调时
- 9元组时间：年/月/日/时/分/秒/一周中的第几天/一年中的第几天/是否使用夏时制

```
>>> import time
>>> time.time()    # 当前的时间戳
1567560766.9479718
>>> time.ctime()   # 当前时间的字符串形式
'Wed Sep  4 09:33:39 2019'
>>> time.localtime() # 当前时间的九元组形式
time.struct_time(tm_year=2019, tm_mon=9, tm_mday=4, tm_hour=9, tm_min=34, tm_sec=12,
tm_wday=2, tm_yday=247, tm_isdst=0)
>>> t = time.localtime()
>>> t.tm_year
2019
>>> t.tm_hour
9
>>> time.sleep(3)   # 睡眠3秒
>>> time.strftime("%Y-%m-%d %H:%M:%S") # 指定显示字符串格式
'2019-09-04 10:13:12'
>>> time.strftime('%a %A')
'Wed Wednesday'

# 将字符串时间格式转换为struct_time对象
>>> s1 = '2019-10-1 8:00:00'
>>> time.strptime(s1, '%Y-%m-%d %H:%M:%S')
time.struct_time(tm_year=2019, tm_mon=10, tm_mday=1, tm_hour=8, tm_min=0, tm_sec=0,
tm_wday=1, tm_yday=274, tm_isdst=-1)
```

datetime模块

```
>>> from datetime import datetime
>>> datetime.now() # 返回的是年、月、日、时、分、秒、毫秒
datetime.datetime(2019, 9, 4, 10, 52, 38, 619439)
>>> t = datetime.now()
>>> t.year, t.month, t.day, t.hour, t.minute, t.second, t.microsecond
(2019, 9, 4, 10, 53, 31, 787243)

# 转成时间字符串
>>> t.strftime('%Y-%m-%d %H:%M:%S')
'2019-09-04 10:53:31'
```

```
# 将时间字符串转成datetime对象
>>> datetime.strptime('2019-09-04 10:53:31', '%Y-%m-%d %H:%M:%S')
datetime.datetime(2019, 9, 4, 10, 53, 31)
# 手工创建某一时间点的datetime对象
>>> t9 = datetime(2019, 9, 4, 11)

>>> from datetime import datetime, timedelta
>>> now = datetime.now()
>>> days = timedelta(days=100, hours=3)
>>> now - days    # 100天3小时之前的时间
datetime.datetime(2019, 5, 27, 8, 26, 42, 31696)
>>> now + days    # 100天3小时之后的时间
datetime.datetime(2019, 12, 13, 14, 26, 42, 31696)
```

异常处理

- 如果程序没有异常处理，当程序遇到错误时，它就没有办法继续执行下去了，将会崩溃、终止执行
- 异常处理就是提前给出各种错误的解决方案，程序运行时仍然会遇到错误，但是因为给了解决方案，程序不再崩溃。

```
try:
    可能发生异常的语句
except 异常1:
    解决方案代码
except 异常2:
    解决方案代码
... ..
except 异常n:
    解决方案代码
else:
    异常不发生才执行的代码
finally:
    异常发不发生都要执行的代码
```

主动触发异常

- 使用raise指定触发什么样的异常
- 通过assert触发断言异常

os模块

- 对文件系统的访问大多通过python的os模块实现
- 该模块是python访问操作系统功能的主要接口

```
>>> import os
>>> os.getcwd()    # pwd
'/var/ftp/nsd2019/nsd1904/py02/day01'
>>> os.listdir()   # ls
```

```

>>> os.listdir('/tmp')    # ls /tmp
>>> os.mkdir('/tmp/demo')  # mkdir /tmp/demo
>>> os.makedirs('/tmp/nsd1904/demo')  # mkdir -p /tmp/nsd1904/demo
>>> os.chdir('/tmp/nsd1904/demo')  # cd /tmp/nsd1904/demo
>>> os.getcwd()
'/tmp/nsd1904/demo'
>>> os.listdir()
[]
>>> os.symlink('/etc/hosts', 'zhuji')  # ln -s /etc/hosts zhuji
>>> os.mknod('mytest.txt')  # touch mytest.txt
>>> os.listdir()
['zhuji', 'mytest.txt']
>>> os.chmod('mytest.txt', 0o644)  # Linux权限是8进制数
>>> os.chown('mytest.txt', 1015, 1015)  # 通过uid/gid改变文件的属主属组
>>> os.rmdir('/tmp/demo')  # rmdir /tmp/demo  # 只能删空目录
>>> os.remove('mytest.txt')  # 删除文件
>>> os.listdir()
['zhuji']

>>> os.listdir()
['zhuji']
>>> os.path.abspath('zhuji')  # 返回zhuji的绝对路径
'/tmp/nsd1904/demo/zhuji'
>>> os.path.basename('/tmp/nsd1904/demo/abc.txt')
'abc.txt'
>>> os.path.dirname('/tmp/nsd1904/demo/abc.txt')
'/tmp/nsd1904/demo'
>>> os.path.split('/tmp/nsd1904/demo/abc.txt')
('/tmp/nsd1904/demo', 'abc.txt')
>>> os.path.isfile('/etc/host')  # 存在并且是文件吗？
False
>>> os.path.isdir('/etc')  # 存在并且是目录吗？
True
>>> os.path.islink('/etc/grub2.cfg')  # 存在并且是链接吗？
True
>>> os.path.ismount('/home')  # 存在并且是挂载点吗？
False
>>> os.path.exists('/')  # 存在吗？
True
>>> os.path.split('/tmp/nsd1904/demo/abc.txt')
('/tmp/nsd1904/demo', 'abc.txt')
>>> os.path.join('/tmp/nsd1904/demo', 'abc.txt')
'/tmp/nsd1904/demo/abc.txt'

```

pickle模块

- 普通的文件操作，是对字符型数据进行读写
- pickle可以把任意的数据类型写入文件，还可以无损地取出

```
>>> import pickle
>>> user = {'name': 'bob', 'age': 20}
>>> f = open('/tmp/user.data', 'wb')
>>> pickle.dump(user, f)    # 将字典写入文件
>>> f.close()

>>> with open('/tmp/user.data', 'rb') as f:
...     data = pickle.load(f)
>>> data
{'name': 'bob', 'age': 20}
>>> type(data)
<class 'dict'>
```

记账程序练习

日期	收入	支出	余额	备注
2019-9-4	0	0	10000	init
2019-9-4	10000	0	20000	salary
2019-9-4	0	200	19800	eat

```
records = [
    ['2019-9-4', 0, 0, 10000, 'init'],
]
record = ['2019-9-4', 10000, 0, 20000, 'salary']
records.append(record)
print(records)
[
    ['2019-9-4', 0, 0, 10000, 'init'],
    ['2019-9-4', 10000, 0, 20000, 'salary']
]
```