

nsd1907_py02_day01

time模块

时间表示方法：

- 时间戳：自1970-1-1 0:0:0到某一时间之间的秒数
- UTC时间：世界协调时。
- struct_time9元组：(年，月，日，时，分，秒，一周中的第几天，一年中的第几天，是否是夏季节约时间)

```
>>> import time
# 获取当前时间的时间戳
>>> time.time()
1575854717.449896
# UTC的字符串形式时间
>>> time.ctime()
'Mon Dec  9 09:26:22 2019'
# 获取当前时间的9元组
>>> t = time.localtime()
>>> t.tm_year
2019
>>> t.tm_yday
343
# 睡眠3秒
>>> time.sleep(3)
# 返回指定的时间字符串格式
>>> time.strftime('%Y-%m-%d %H:%M:%S')
'2019-12-09 09:47:51'
>>> time.strftime('%a %A')
'Mon Monday'
# 把时间字符串转成9元组格式
>>> time.strptime('2019-12-09 09:47:51', '%Y-%m-%d %H:%M:%S')
>>> t = time.strptime('2019-12-09 09:47:51', '%Y-%m-%d %H:%M:%S')
>>> t.tm_mon
12
>>> t.tm_mday
9
>>> t1 = time.strptime('2019-12-09 09:47:51', '%Y-%m-%d %H:%M:%S')
>>> t2 = time.strptime('2019-11-08 10:50:25', '%Y-%m-%d %H:%M:%S')
>>> t1 > t2
True
```

datetime模块

```
>>> from datetime import datetime
# 取出当前时间的年、月、日、时、分、秒、毫秒
>>> datetime.now()
```

```

>>> t = datetime.now()
>>> t.year
2019
>>> t.month
12
>>> t.day
9
>>> t.hour
10
>>> t.minute
46
>>> t.second
33
>>> t.microsecond
600073

>>> t.strftime('%Y/%m/%d %H:%M:%S') # datetime对象转成指定字符串
'2019/12/09 10:46:33'
# 字符串转成datetime对象
>>> t9 = datetime.strptime('2019-12-9 11:00:00', '%Y-%m-%d %H:%M:%S')
>>> t9
datetime.datetime(2019, 12, 9, 11, 0)

# 计算时间差, 获取从现在开始前100天、后100天的时间
>>> from datetime import datetime, timedelta
>>> t = datetime.now()
>>> days = timedelta(days=100, hours=1)
>>> t
datetime.datetime(2019, 12, 9, 11, 20, 57, 745485)
>>> t - days
datetime.datetime(2019, 8, 31, 10, 20, 57, 745485)
>>> t + days
datetime.datetime(2020, 3, 18, 12, 20, 57, 745485)

```

异常处理

- 没有异常处理的代码, 当程序遇到错误时, 程序将会崩溃, 终止执行, 在屏幕上打印报错信息
- 异常处理是, 提前考虑到程序运行过程中将会出现的错误, 给出这些错误的解决方案, 使得程序不要崩溃, 还可以继续执行下去

```

try:
    有可能发生异常的语句
except 异常1:
    发生 异常1 执行的语句
except (异常2, 异常3):
    发生 异常2或异常3 执行的语句
else:
    不发生异常才执行的语句
finally:
    不管异常是否会发生, 都要执行的语句

```

主动触发异常的方法

- raise触发指定的异常
- 通过assert触发断言异常

os模块

```
>>> import os
>>> os.getcwd()    # pwd
'/var/ftp/nsd2019/nsd1907/py02/day01'
>>> os.listdir()   # ls
>>> os.mkdir('/tmp/mydir') # mkdir /tmp/mydir
>>> os.makedirs('/tmp/nsd1907/demo') # mkdir -p /tmp/nsd1907/demo
>>> os.listdir('/tmp/nsd1907/demo') # ls /tmp/nsd1907/demo
>>> os.chdir('/tmp/nsd1907/demo')   # cd /tmp/nsd1907/demo
>>> os.getcwd()
>>> import shutil
>>> shutil.copy('/etc/hosts', 'zhuji')
>>> shutil.copytree('/etc/security', 'anquan')
>>> os.listdir()

>>> os.symlink('/etc/passwd', 'mima') # ln -s /etc/passwd mima
# 注意：linux命令chmod后面的权限数字是8进制
>>> os.chmod('zhuji', 0o755) # chmod 755 zhuji
# 8进制数0o644转换成10进制数是420
>>> os.chmod('zhuji', 420) # chmod 644 zhuji
>>> os.stat('zhuji') # stat zhuji
>>> zj = os.stat('zhuji')
>>> zj.st_size # 文件大小，字节数
9768
>>> zj.st_atime # 最近的一次访问时间，access时间
1575877216.7025838
>>> time.ctime(1575877216.7025838)
'Mon Dec  9 15:40:16 2019'
>>> os.remove('mima') # rm -f mima

>>> os.path.isabs('/etc') # 是绝对路径吗？
True
>>> os.path.isabs('/abcd') # /abcd不存在也为True
True
>>> os.path.isfile('/etc/hosts') # 存在并且是文件吗？
>>> os.path.ismount('/etc') # 存在并且是挂载点吗？
>>> os.path.isdir('/') # 存在并且是目录吗？
>>> os.path.islink('/etc/grub2.cfg') # 存在并且是软链接吗？
>>> os.path.exists('/etc/hostname') # 存在吗？

>>> path = '/tmp/nsd1907/demo/hello.txt'
>>> os.path.split(path)
('/tmp/nsd1907/demo', 'hello.txt')
>>> os.path.basename(path)
'hello.txt'
```

```

>>> os.path.dirname(path)
'/tmp/nsd1907/demo'
>>> os.path.join('/tmp/nsd1907/demo', 'hello.txt')
'/tmp/nsd1907/demo/hello.txt'

# 递归列出目录中所有内容
>>> list(os.walk('anquan'))
[('anquan', ['console.apps', 'console.perms.d', 'limits.d', 'namespace.d'], ['access.conf', 'chroot.conf', 'console.handlers', 'console.perms', 'group.conf', 'limits.conf', 'namespace.conf', 'namespace.init', 'opasswd', 'pam_env.conf', 'sepermit.conf', 'time.conf', 'pwquality.conf']), ('anquan/console.apps', [], ['config-util', 'xserver', 'liveinst', 'setup']), ('anquan/console.perms.d', [], []), ('anquan/limits.d', [], ['20-nproc.conf']), ('anquan/namespace.d', [], [])]
>>> data = list(os.walk('anquan')) # 列表由5个元组构成
>>> len(data)
5
# 构成列表的元组有统一的结构：(字符串，列表，列表)
# 进一步观察元组结构：(路径字符串，该路径下目录列表，该路径下文件列表)
>>> data[0]
('anquan', ['console.apps', 'console.perms.d', 'limits.d', 'namespace.d'], ['access.conf', 'chroot.conf', 'console.handlers', 'console.perms', 'group.conf', 'limits.conf', 'namespace.conf', 'namespace.init', 'opasswd', 'pam_env.conf', 'sepermit.conf', 'time.conf', 'pwquality.conf'])
>>> data[1]
('anquan/console.apps', [], ['config-util', 'xserver', 'liveinst', 'setup'])
>>> data[2]
('anquan/console.perms.d', [], [])

# 遍历，打印5行元组
>>> for item in os.walk('anquan'):
...     print(item)

# 遍历，将元组中的三项分别赋值给3个变量
>>> for path, folders, files in os.walk('anquan'):
...     print(path, folders, files)

# 遍历，将目录列表和文件列表也逐一遍历出来
>>> for path, folders, files in os.walk('anquan'):
...     print('%s:' % path)
...     for mulu in folders:
...         print(mulu, end='\t')
...     for file in files:
...         print(file, end='\t')
...     print('\n')

```

pickle模块

- pickle模块，可以将任意的数据类型写到文件，还可以无损地取出。

```

>>> import pickle

```

```
>>> shopping_list = ['apple', 'banana', 'egg', 'orange']
# 将列表写入文件
>>> with open('/tmp/shopping.data', 'wb') as fobj:
...     pickle.dump(shopping_list, fobj)

# 在文件中将列表取出
>>> import pickle
>>> with open('/tmp/shopping.data', 'rb') as fobj:
...     alist = pickle.load(fobj)
...
>>> type(alist)
<class 'list'>
>>> alist
['apple', 'banana', 'egg', 'orange']
```