一、时间方法

**1、时间表示方式**

时间戳：从1970-1-1 00:00:00开始按秒计算的偏移量

UTC：世界协调时间(时区)

九元组：年、月、日、时、分、秒、一年中的第几周、一年中的第几天、是否是夏季节约时间

**2、time模块方法**

>>> import time

>>> time.time() #当前时间戳

1547712318.8084626

>>> time.ctime() #当前UTC时间

'Thu Jan 17 16:05:25 2019'

>>> t = time.localtime() #当前时间的九元组

time.struct\_time(tm\_year=2019, tm\_mon=1, tm\_mday=17, tm\_hour=16, tm\_min=5, tm\_sec=50, tm\_wday=3, tm\_yday=17, tm\_isdst=0)

>>> t.tm\_year

2019

>>> time.strftime('%Y-%m-%d %A %H:%M:%S')

'2019-01-17 Thursday 16:22:52'

1. **datetime模块方法**

>>> from datetime import datetime

#相当于datetime.datetime

>>> t = datetime.now()

datetime.datetime(2019, 1, 17, 16, 29, 27, 416845)

#返回对象有：年、月、日、时、分、秒、毫秒

>>> t.year

2019

1. **时间计算**

>>> from datetime import datetime, timedelta

>>> t = datetime.today()

>>> days = timedelta(days=60)

>>> t - days

datetime.datetime(2018, 11, 18, 16, 35, 43, 464375)

>>> t + days

datetime.datetime(2019, 3, 18, 16, 35, 43, 464375)

二、异常处理

程序发生异常时，程序崩溃，终止执行。因此要引入异常处理。

**1、完整语法**

try: #有可能发生异常的语句

num = int(input('number:'))

result = 100 / num

print(result)

except (ValueError, ZeroDivisionError): #异常的捕获与处理

print('Invalid input.')

except EOFError as e: #把错误保存到变量e中

print('错误:', e)

else: #不发生异常才执行

print(result)

finally: #不管是否发生异常，都会执行

print('Done')

#常用：try-excpt结构、try-finally结构

**2、触发异常**

def set\_age(name, age):

if not 0 < age < 120:

raise ValueError('年龄超出范围')

#触发异常，自定义的异常名称必须为系统异常

print('%s is %d years old' % (name, age))

def set\_age2(name, age):

assert 0 < age < 120, '年龄超出范围'

#断言异常，返回AssertionError:'年龄超出范围'

print('%s is %d years old' % (name, age))

if \_\_name\_\_ == '\_\_main\_\_':

# set\_age("bob", 233)

set\_age2("bob", 233)

1. OS相关模块

**1、pickle模块**

文件的常规write方法只支持字符串写入，不支持其他数据类型，因此向读取出原来的数据类型比较困难；

pickle允许把任意数据类型写入文件，将来还可以无损的取出。

>>> import pickle

>>> with open('/tmp/mydata', 'wb') as fobj:

... shopping\_list = ['egg', 'apple', 'banana']

... pickle.dump(shopping\_list, fobj)

>>> with open('/tmp/mydata', 'rb') as fobj:

... mylist = pickle.load(fobj)

print(type(mylist))

**2、os模块**

|  |  |
| --- | --- |
| **os模块方法** | **作用** |
| os.getcwd() | pwd |
| os.listdir() | ls |
| os.mkdir('/tmp/demo') | mkdir |
| os.listdir('/tmp/demo') | ls /tmp/demo |
| os.chdir('/tmp/demo') | cd |
| os.symlink('/etc/hosts', '/tmp/zhuji') | ln -s /etc/hosts /tmp/zhuji |
| os.mknod('hi.txt') | touch |
| os.chmod('hi.txt', 0o600) | chmod |
| os.walk('/tmp/demo') | 返回一个生成器对象  (path, [folders], [files]) |
| os.path.isfile('/etc') | 是文件吗？ |
| os.path.isdir('/etc') | 是目录吗？ |
| os.path.islink('/etc') | 是软连接吗？ |
| os.path.exists('/etc') | 存在吗？ |
| os.path.abspath('hi.txt') | 得到当前目录下文件的绝对路径 |
| os.path.basename('/tmp/hi.txt') | 'hi.txt' |
| os.path.dirname('/tmp/hi.txt') | '/tmp/demo' |
| os.path.split('/tmp/hi.txt') | ('/tmp/demo', 'hi.txt') |
| os.path.join('/tmp/hi.txt') | '/tmp/demo/hi.txt' |
| os.path.getsize('/tmp/hi.txt') | 文件大小 |
| os.link(app\_path, dest\_path)  os.unlink(dest\_path) | 创建硬连接  取消硬连接 |
| os.symlink(app\_path, dest\_path)  os.unlink(dest\_path) | 创建软连接  取消软连接 |

四、案例：记账程序

1）假设在记账时,有一万元钱

2）无论是开销还是收入都要进行记账

3）记账内容包括时间、金额和说明等

4）记账数据要求永久存储