时间戳: 自1970-1-1 0:0:0到某一时间点之间的秒数

struct\_time: 它是一个9元组

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)

(tm\_year=2019, tm\_mon=10, tm\_mday=12, tm\_hour=9, tm\_min=33, tm\_sec=10,tm\_wday=5, tm\_yday=285, tm\_isdst=0)

time模块

time.time() # 自1970-1-1 8:0:0到执行指令间的秒数

time.ctime() #

'Sat Oct 12 09:31:20 2019'

time.ctime(0) # 参数是一个秒数

'Thu Jan 1 08:00:00 1970'

time.localtime() # 当前时间的struct\_time

time.struct\_time(tm\_year=2019, tm\_mon=10, tm\_mday=12, tm\_hour=9, tm\_min=33, tm\_sec=10,tm\_wday=5, tm\_yday=285, tm\_isdst=0)

time.sleep(3) # 睡眠3秒

time.strftime('%Y-%m-%d %H:%M:%S') # 年-月-日 时:分:秒

将时间字符串转换为9元组时间

time.strptime('20191001 09:50:00', '%Y%m%d %H:%M:%S')

**datatime模块**

**datetime方法**

from datetime import datetime #导入的是两个datetime单词

>>> datetime.now() # 返回年月日时分秒 毫秒

*datetime.datetime(2019, 10, 12, 10, 39, 47, 593038)*

t1 = datetime.now()

t1.year t1.month t1.day t1.hour,

t1.minute t1.second t1.microsecond

(2019, 10, 12, 10, 40, 58, 342680)

t1.strftime('%H:%M:%S')

'10:40:58'

将时间字符串转成datetime对象

>>> t2 = datetime.strptime('20191001 09:50:00', '%Y%m%d %H:%M:%S')

>>> t2

datetime.datetime(2019, 10, 1, 9, 50)

**timedelta方法**

from datetime import datetime, timedelta

>>> t = datetime.now()

>>> days = timedelta(days=50, hours=2)

>>> t - days # 50天2小时之前

datetime.datetime(2019, 8, 23, 8, 54, 5, 724316)

>>> t + days # 50Gd 2小时之后

datetime.datetime(2019, 12, 1, 12, 54, 5, 724316)

异常

常见异常:

ValueError

ZeroDivisionError

KeyboardInterrupt

EOFError

try,

try:

(有可能发生异常的语句块)

except 报错关键字 as 变量 :

(发生异常执行的语句+ 变量)　　　　　　　　　　 # 发生异常执行放这

else:

(不发生异常执行的语句)　　　　　　　　　　　　 ＃不发生异常执行放这

finally:

(无论什么情况都要执行,上面有exit也执行)　　　　＃无论什么情况都执行放这

raise

os模块

os.getcwd()　　　　　　　 # pwd

os.listdir(**'/home/student'**) # ls

os.chdir(**'/tmp/demo'**) # cd

os.mkdir(**'/tmp/demo'**) # mkdir

os.makedirs(**'/tmp/xsx/xsx'**) # mkdir -p

os.mknod(**'sss.txt'**) # touch

os.rmdir(**'/tmp/demo'**) # 只能删除空目录

os.remove(**'sss.txt'**) # rm -f sss.txt

os.rename(**'sss.txt'**,**'sssll.txt'**) # mv sss.txt sssll.txt

link

os.path.abspath(**'times.txt'**) # 返回绝对路径

os.path.dirname(‘绝对路径’) # 返回去除文件名后的绝对路径

os.path.split(‘需输入绝对路径’) #拆分 绝对路径/ 文件名

os.path判断

**a = 'times.txt'**

os.path.exists(a) # 'times.txt'存在吗

os.path.isabs(a) # 是绝对路径吗

os.path.isfile(a) # 是文件吗

os.path.isdir(a) # 是目录吗

os.path.ismount(a) # 是挂载点吗

os.path.islink(a) # 是 吗

*pprint模块*

输出:

1.一个元组为一个路径,目录,文件

**pickle模块**

普通文件只能写入str或bytes类型的数据

pickle存储器可以把任意数据类型写入文件,还能无损地取出

import pickle

写入

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

>>> pickle.dump(shopping\_list, f)

读取

>>> mylist = pickle.load(f)

>>> type(mylist)

<class 'list'>

>>> mylist

['apple', 'egg', 'banana']