**IndexError: list assignment index out of range的解决方法**

Example：

filesize = [ ]

for i in range(len(filename)):

filenamestr = filename[i]

filesize[i]=os.path.getsize(join(Input,filenamestr))

报错，这是由于当filesize为一个空列表，直接按索引值向其中添加东西时，会报错，因为空的列表不能直接指定其位置

filesize.append(os.path.getsize(join(Input,filenamestr)))

将指定位置插入改为在在最后一个元素后插入即可。

# Python 获取文件的创建时间，修改时间和访问时间

# os.path.getatime(file) 输出文件访问时间  
# os.path.getctime(file) 输出文件的创建时间  
# os.path.getmtime(file) 输出文件最近修改时间

import time  
import os

def fileTime(file):  
　　return [  
　　　　time.ctime(os.path.getatime(file)),  
　　　　time.ctime(os.path.getctime(file)),  
　　　　time.ctime(os.path.getmtime(file))]

times = fileTime("d")

# 将时间戳转化为日期格式

## Strftime（）函数用法

time\_t t = time( 0 );

char tmp[64];

我们可以根据format指向字符串中格式命令把timeptr中保存的时间信息放在strDest指向的字符串中，最多向strDest中存放maxsize个字符。该函数返回向strDest指向的字符串中放置的字符数。

strftime( tmp, sizeof(tmp), "%Y%m%d%H%M%S", localtime(&t) ); //格式化输出.

函数strftime()的操作有些类似于sprintf()：识别以百分号(%)开始的格式命令集合，格式化输出结果放在一个字符串中。格式化命令说明串strDest中各种日期和时间信息的确切表示方法。格式串中的其他字符原样放进串中。格式命令列在下面，它们是区分大小写的。  
%a 星期几的简写  
%A 星期几的全称  
%b 月分的简写  
%B 月份的全称  
%c 标准的日期的时间串  
%C 年份的后两位数字  
%d 十进制表示的每月的第几天  
%D 月/天/年  
%e 在两字符域中，十进制表示的每月的第几天  
%F 年-月-日  
%g 年份的后两位数字，使用基于周的年  
%G 年分，使用基于周的年  
%h 简写的月份名  
%H 24小时制的小时  
%I 12小时制的小时  
%j 十进制表示的每年的第几天  
%m 十进制表示的月份  
%M 十时制表示的分钟数  
%n 新行符  
%p 本地的AM或PM的等价显示  
%r 12小时的时间  
%R 显示小时和分钟：hh:mm  
%S 十进制的秒数  
%t 水平制表符  
%T 显示时分秒：hh:mm:ss  
%u 每周的第几天，星期一为第一天 （值从0到6，星期一为0）  
%U 第年的第几周，把星期日做为第一天（值从0到53）  
%V 每年的第几周，使用基于周的年  
%w 十进制表示的星期几（值从0到6，星期天为0）  
%W 每年的第几周，把星期一做为第一天（值从0到53）  
%x 标准的日期串  
%X 标准的时间串  
%y 不带世纪的十进制年份（值从0到99）  
%Y 带世纪部分的十制年份  
%z，%Z 时区名称，如果不能得到时区名称则返回空字符。  
%% 百分号

## 例子

import time timeStamp = 1557502800

timeArray = time.localtime(timeStamp)

otherStyleTime = time.strftime("%Y-%m-%d %H:%M:%S", timeArray) print(otherStyleTime)

执行以上代码输出结果为：

2019-05-10 23:40:00

# python3 除法去掉小数，保留整数的做法

1、 / 操作符（真除法）

在python3中，/操作符在表达式中时运行的永远是真除法，无论接受任何类型，结果都会保持小数部分

2. // 操作符(Floor除法)

在python3中，// 操作符在表达式中时运行的永远是Floor除法，只有当接受类型中包含浮点型时，结果才会保留小数部分

小数的不同显示包含多种， floor， trunc，round需要导入

1） floor，把小数向下截断到它的下层，即小于小数的最大整数。并且对负数有效。

math.floor(2.5)) # 2

math.floor(-2.5)) # -3

2）trunc, 真截断，真正的截断了小数，所以当小数大于0时，效果与floor相同

math.trunc(2.5)) # 2

math.trunc(-2.5)) # -2

3）round，格式化一个小数，与小学数学中的约等类似。round是内置函数，不需要倒入。

round(2.567)) # 3

round(2.467)) # 2

round(2.567, 2)) # 2.57

原文链接：<https://blog.csdn.net/banxia1995/article/details/86219312>

# python3中获取文件当前绝对路径的两种方法

方法1:

import sys

print(sys.argv)

得到文件当前绝对路径字符串的一个列表

['D:/pycharm/PracticeProject/ClientServerNetworking.py']

方法2:

import os

print(os.getcwd())

print(os.listdir())

print(os.path.join(os.getcwd(),os.listdir()[1]))

D:\pycharm\PracticeProject

['.idea', 'ClientServerNetworking.py']

D:\pycharm\PracticeProject\ClientServerNetworking.py

os.getcwd()得到当前的工作路径,路径中不包括当前的文件名ClientServerNetworking.py,

os.listdir()得到当前路径下的文件和文件夹名字的列表

最后一步拼接路径就得到了当前文件的绝对路径的字符串

注意：方法1和方法2一个斜杠一个反斜杠

os.path.abspath(path)        #返回当前文件绝对路径



**os.abspath(path)**:获取文件的绝对路径。这里path指的是路径，例如我这里输入“data.csv”

[In] os.path.abspath('data.csv')

[Out] 'E:\\kaggle\\Titanic\\data.csv'

# 将数据写入到csv文件中

import csv

head = ["name","age"]

data = [

("小明",12),

("小小",12),

("小大",14),

]

with open("test.csv","w",encoding="utf-8") as f:

csvf = csv.writer(f)

csvf.writerow(head)

csvf.writerows(data)

python按照列写入csv文件

import pandas as pd

#a和b的长度必须保持一致，否则报错

a = [x for x in range(5)]

b = [x for x in range(5,10)]

#字典中的key值即为csv中列名

dataframe = pd.DataFrame({'a\_name':a,'b\_name':b})

#将DataFrame存储为csv,index表示是否显示行名，default=True

dataframe.to\_csv(r"G:\A1大论文内容\实验\test.csv",sep=',')

import csv

import pandas as pd

a=['2015-11-03', '2015-11-03 ', '2015-11-03', '2015-11-03 ']

# 将list转为dataframe 显然就变成一列了

d = pd.DataFrame(a)

# mode表示追加 在追加时会将列名也作为一行进行追加，故header隐藏表头（列名）

d.to\_csv('a.csv',index=False,mode='a',header=None)

# 容量换算关系

B     1B= 8b

KB  1KB=1024B

MB  1MB=1024KB

GB  1GB=1024MB

TB   1TB=1024GB

# python提取特定字符串前后的方法

提取该字符串中“atalog/”后面的字符。

a = "/Instruments-Meters-Catalog/Flowmeter.html"

b = "/Instruments-Meters-Catalog/Flowmeter.html"

filetype = b[b.rfind("."):]

filetype=filetype[2:]

print(filetype)

#输出 tml

# python中对list元素求和、求积的普通方法和便捷方法

求和、求积最先想到的是for循环，可以直接使用python中封装的函数，一行代码解决问题，一般来说，能不用for循环就不用，因为相比于封装函数，for循环太慢。

# -\*- encoding:utf-8 -\*-

from pandas.compat import reduce

a = [1, 2, 3, 4, 5, 6]

#遍历

s = 0

for x in a:

s += x

print('first sum value : ', s)

#直接sum

s = sum(a)

print('second sum value : ', s)

# 输出 ：

# first sum value : 21

# second sum value : 21

#求积，遍历

s = 1

for x in a:

s \*= x

print('first dot value : ', s)

#求积，算子

from operator import mul

reduce(mul, a)

print('second dot value : ', s)

# 输出：

# first dot value : 720

# second dot value : 720

# pandas to\_csv()写入函数参数详解

DataFrame.to\_csv(path\_or\_buf=None,

sep=',', na\_rep='',

float\_format=None,

columns=None,

header=True, index=True,

index\_label=None,mode='w',

encoding=None,

compression='infer',

quoting=None, quotechar='"',

line\_terminator=None, chunksize=None,

date\_format=None, doublequote=True,

escapechar=None, decimal='.', errors='strict')

# 数据类型

print(type(filename))#list<class 'list'>

print(type(filename[1]))#filename元素索引的类型 <class 'str'>

print(Input)

print(type(Input))#<class 'str'>

# Python中统计列表中各个元素出现的次数的几种方法

用函数set()

num = [1, 1, 2, 2, 3, 4, 4, 1]

set\_num = set(num) # set()会创建一个无序且不重复的集合

for i in set\_num:

输出结果：

1--数量：3

2--数量：2

3--数量：1

4--数量：2

print('%d--数量：%d' % (i,num.count(i)))

如何用Python统计含多个字符串的列表中每个单词出现的次数？用字典

List=['bmp', 'bmp', 'bmp', 'bmp', 'bmp', 'bmp', 'bmp', 'bmp', 'bmp', 'bmp', 'bmp', 'bmp', 'bmp', 'bmp', 'bmp', 'bmp', 'bmp', 'bmp', 'bmp', 'bmp', 'bmp', 'bmp', 'bmp', 'bmp', 'csv', 'docx', 'docx', 'doc']

dict={}

for list in List:

keys=list.split(" ")

for key in keys:

if key in dict.keys():

dict[key]=dict[key]+1

else:

dict[key]=1

print(dict)

# TypeError: 'dict' object is not callable

dict()是python的一个函数，但我又将dict自定义为一个python字典，在之后想调用dict()函数是会报出“TypeError: 'dict' object is not callable”的错误，只需将之前自定义的变量delete掉即可。

# 将两个列表合成一个字典

a=[1,2,3,4,5]

b=[6,7,8,9,10]

c = dict(zip(a,b))