os.walk()

该函数的原型是：

os.walk(top, topdown=Ture, onerror=None, followlinks=False)

该函数没有返回值。

参数

1. top -- 根目录下的每一个文件夹(包含它自己), 产生3-元组 (dirpath, dirnames, filenames)【文件夹路径, 文件夹名字, 文件名】。

2. topdown --可选，为True或者没有指定, 一个目录的的3-元组将比它的任何子文件夹的3-元组先产生 (目录自上而下)。如果topdown为 False, 一个目录的3-元组将比它的任何子文件夹的3-元组后产生 (目录自下而上)。

3. onerror -- 可选，是一个函数; 它调用时有一个参数, 一个OSError实例。报告这错误后，继续walk,或者抛出exception终止walk。

4. followlinks -- 设置为 true，则通过软链接访问目录。

1

2

3

4

5

函数执行之后得到一个三元tupple（dirpath， dirnames， filenams。

dirpath：string，是当前目录的路径；

dirnames：list， 是当前路径下所有的子文件夹名字；

filenames：list， 是当前路径下所有的非目录子文件的名字。

要获取完整的路径，dirnames和filenames是不包含路径信息的，可以使用

os.path.join(dirpath, dirnames)

1

获得文件的完整路径。

代码示例：

#!/usr/bin/python

# -\*- coding: UTF-8 -\*-

import os

for root, dirs, files in os.walk(".", topdown=False):

for name in files:

print(os.path.join(root, name))

for name in dirs:

print(os.path.join(root, name))

1

2

3

4

5

6

7

8

9

当需要特定类型的文件时，代码如下：

# -\*- coding: utf-8 -\*-

import os

def file\_name(file\_dir):

L=[]

for dirpath, dirnames, filenames in os.walk(file\_dir):

for file in filenames :

if os.path.splitext(file)[1] == '.jpg':

L.append(os.path.join(dirpath, file))

return L

1

2

3

4

5

6

7

8

9

10

11

其中os.path.splitext()函数将路径拆分为文件名+扩展名，例如os.path.splitext(“E:/lena.jpg”)将得到”E:/lena“+”.jpg”。

os.listdir()

os.listdir()函数返回指定路径下的文件和文件夹列表。

代码示例：

#!/usr/bin/python

# -\*- coding: UTF-8 -\*-

import os, sys

# 打开文件

path = "/var/www/html/"

dirs = os.listdir( path )

# 输出所有文件和文件夹

for file in dirs:

print file

————————————————

版权声明：本文为CSDN博主「JohnieLi」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/JohinieLi/article/details/76660733