



第 10 章 文件与文件夹操作



10.1 os 模块

方法	功能说明
chdir(path)	把 path 设为当前工作目录
curdir	当前文件夹
environ	包含系统环境变量和值的字典
extsep	当前操作系统所使用的文件扩展名分隔符
get_exec_path()	返回可执行文件的搜索路径
getcwd()	返回当前工作目录
listdir(path)	返回 path 目录下的文件和目录列表

10.1 os 模块

方法	功能说明
<code>remove(path)</code>	删除指定的文件，要求用户拥有删除文件的权限，并且文件没有只读或其他特殊属性
<code>rename(src, dst)</code>	重命名文件或目录，可以实现文件的移动，若目标文件已存在则抛出异常，不能跨越磁盘或分区
<code>replace(old, new)</code>	重命名文件或目录，若目标文件已存在则直接覆盖，不能跨越磁盘或分区
<code>scandir(path='.')</code>	返回包含指定文件夹中所有 <code>DirEntry</code> 对象的迭代对象，遍历文件夹时比 <code>listdir()</code> 更加高效
<code>sep</code>	当前操作系统所使用的路径分隔符
<code>startfile(filepath [, operation])</code>	使用关联的应用程序打开指定文件或启动指定应用程序
<code>system()</code>	启动外部程序

10.1 os 模块

```
>>> import os
>>> import os.path
>>> os.rename('C:\\dfg.txt', 'D:\\test2.txt') # 不同分区无法实现
>>> [fname for fname in os.listdir('.')
      if fname.endswith(('.pyc', '.py', '.pyw'))] # 结果略
>>> os.getcwd()                                # 返回当前工作目录
'C:\\Python35'
>>> os.mkdir(os.getcwd()+'\\temp')              # 创建目录
>>> os.chdir(os.getcwd()+'\\temp')              # 改变当前工作目录
>>> os.getcwd()
'C:\\Python35\\temp'
>>> os.mkdir(os.getcwd()+'\\test')
```

10.1 os 模块

```
>>> import time
>>> time.strftime('%Y-%m-%d %H:%M:%S',      # 查看文件创建时间
                  time.localtime(os.stat('yilaizhuru2.py').st_ctime))
'2016-10-18 15:58:57'
>>> os.startfile('notepad.exe')              # 启动记事本程序
```

10.1 os 模块

- **例 10-1** 使用递归法遍历指定目录下所有子目录和文件。

```
from os import listdir
from os.path import join, isfile, isdir

def listDirDepthFirst(directory):
    ''' 深度优先遍历文件夹 '''
    # 遍历文件夹，如果是文件就直接输出
    # 如果是文件夹，就输出显示，然后递归遍历该文件夹
    for subPath in listdir(directory):
        path = join(directory, subPath)
```

10.2 os.path 模块

方法	功能说明
<code>abspath(path)</code>	返回给定路径的绝对路径
<code>basename(path)</code>	返回指定路径的最后一个组成部分
<code>commonpath(paths)</code>	返回给定的多个路径的最长公共路径
<code>commonprefix(paths)</code>	返回给定的多个路径的最长公共前缀
<code>dirname(p)</code>	返回给定路径的文件夹部分
<code>exists(path)</code>	判断文件是否存在
<code>getatime(filename)</code>	返回文件的最后访问时间
<code>getctime(filename)</code>	返回文件的创建时间
<code>getmtime(filename)</code>	返回文件的最后修改时间
<code>getsize(filename)</code>	返回文件的大小

10.2 os.path 模块

方法	功能说明
isabs(path)	判断 path 是否为绝对路径
isdir(path)	判断 path 是否为文件夹
isfile(path)	判断 path 是否为文件
join(path, *paths)	连接两个或多个 path
realpath(path)	返回给定路径的绝对路径
relpath(path)	返回给定路径的相对路径，不能跨越磁盘驱动器或分区
samefile(f1, f2)	测试 f1 和 f2 这两个路径是否引用的同一个文件
split(path)	以路径中的最后一个斜线为分隔符把路径分隔成两部分，以元组形式返回
splitext(path)	从路径中分隔文件的扩展名
splitdrive(path)	从路径中分隔驱动器的名称

10.2 os.path 模块

```
>>> path='D:\\mypython_exp\\new_test.txt'
```

```
>>> os.path.dirname(path)
```

```
'D:\\mypython_exp'
```

```
>>> os.path.basename(path)
```

```
'new_test.txt'
```

```
>>> os.path.split(path)
```

```
('D:\\mypython_exp', 'new_test.txt')
```

```
>>> os.path.split('')
```

```
('', '')
```

```
>>> os.path.split('C:\\\\windows')
```

```
('C:\\\\', 'windows')
```

```
>>> os.path.split('C:\\\\windows\\\\')
```

返回路径的文件夹名

返回路径的最后一个组成部分

切分文件路径和文件名

切分结果为空字符串，两部分

以最后一个斜线为分隔符

10.3 shutil 模块

方法	功能说明
<code>copy(src, dst)</code>	复制文件，新文件具有同样的文件属性，如果目标文件已存在则抛出异常
<code>copy2(src, dst)</code>	复制文件，新文件具有原文件完全一样的属性，包括创建时间、修改时间和最后访问时间等等，如果目标文件已存在则抛出异常
<code>copyfile(src, dst)</code>	复制文件，不复制文件属性，如果目标文件已存在则直接覆盖
<code>copyfileobj(fsrc, fdst)</code>	在两个文件对象之间复制数据，例如 <code>copyfileobj(open('123.txt'), open('456.txt', 'a'))</code>
<code>copymode(src, dst)</code>	把 src 的模式位（mode bit）复制到 dst 上，之后二者具有相同的模式
<code>copystat(src, dst)</code>	把 src 的模式位、访问时间等所有状态都复制到 dst 上
<code>copytree(src, dst)</code>	递归复制文件夹
<code>disk_usage(path)</code>	查看磁盘使用情况
<code>move(src, dst)</code>	移动文件或递归移动文件夹，也可以给文件和文件夹重命名
<code>rmtree(path)</code>	递归删除文件夹
<code>make_archive(base_name, format, root_dir=None, base_dir=None)</code>	创建 tar 或 zip 格式的压缩文件
<code>unpack_archive(filename, extract_dir=None, format=None)</code>	解压缩压缩文件

10.3 shutil 模块

- 下面的代码演示了如何使用标准库 shutil 的 copyfile() 方法复制文件。

```
>>> import shutil                                # 导入 shutil 模块
>>> shutil.copyfile('C:\\dir.txt', 'C:\\dir1.txt') # 复制文件
```

- 下面的代码将 C:\Python35\Dlls 文件夹以及该文件夹中所有文件压缩至 D:\a.zip 文件：

```
>>> shutil.make_archive('D:\\a', 'zip', 'C:\\Python35', 'Dlls')
'D:\\a.zip'
```

- 下面的代码将刚压缩得到的文件 D:\a.zip 解压缩至 D:\a_unpack 文件夹：

```
>>> shutil.unpack_archive('D:\\a.zip', 'D:\\a_unpack')
```

10.3 shutil 模块

- 下面的代码使用 shutil 的 copytree() 函数递归复制文件夹，并忽略扩展名为 pyc 的文件和以“新”字开头的文件和子文件夹：

```
>>> from shutil import copytree, ignore_patterns
>>> copytree('C:\\python35\\test', 'D:\\des_test',
            ignore=ignore_patterns('*.pyc', '新*'))
```

10.4 综合案例解析

- **例 10-2** 把指定文件夹中的所有文件名批量随机化，保持文件类型不变。

```
from string import ascii_letters  
from os import listdir, rename  
from os.path import splitext, join  
from random import choice, randint
```

```
def randomFilename(directory):  
    for fn in listdir(directory):  
        # 切分，得到文件名和扩展名  
        name, ext = splitext(fn)  
        n = randint(5, 20)
```

10.4 综合案例解析

- **例 10-3** 编写程序，统计指定文件夹大小以及文件和子文件夹数量。

```
import os
```

```
totalSize = 0
```

```
fileNum = 0
```

```
dirNum = 0
```

10.4 综合案例解析

```
def visitDir(path):  
    global totalSize  
    global fileNum  
    global dirNum  
    for lists in os.listdir(path):  
        sub_path = os.path.join(path, lists)  
        if os.path.isfile(sub_path):  
            fileNum = fileNum+1                                # 统计文件数量  
            totalSize = totalSize+os.path.getsize(sub_path)    # 统计文件总大小  
        elif os.path.isdir(sub_path):  
            dirNum = dirNum+1                                    # 统计文件夹数量  
            visitDir(sub_path)                                   # 递归遍历子文件夹
```

10.4 综合案例解析

```
def main(path):  
    if not os.path.isdir(path):  
        print('Error:', path, ' is not a directory or does not exist.')  
        return  
    visitDir(path)
```

```
def sizeConvert(size):  
    K, M, G = 1024, 1024**2, 1024**3  
    if size >= G:  
        return str(size/G)+'G Bytes'  
    elif size >= M:
```

单位换算

```
        return str(size/M)+'M Bytes'
```

10.4 综合案例解析

```
def output(path):  
    print('The total size of '+path+' is:'+sizeConvert(totalSize)  
          +'('+str(totalSize)+' Bytes)')  
    print('The total number of files in '+path+' is:',fileNum)  
    print('The total number of directories in '+path+' is:',dirNum)  
  
if __name__=='__main__':  
    path = r'd:\idapro6.5plus'  
    main(path)  
    output(path)
```

10.4 综合案例解析

- **例 10-4** 编写程序，递归删除指定文件夹中指定类型的文件和大小为 0 的文件。

```
from os.path import isdir, join, splitext
from os import remove, listdir, chmod, stat

filetypes = ('.tmp', '.log', '.obj', '.txt')      # 指定要删除的文件类型
def delCertainFiles(directory):
    if not isdir(directory):
        return
    for filename in listdir(directory):
        temp = join(directory, filename)
        if isdir(temp):
```