**一、python中对文件、文件夹操作时经常用到的os模块和shutil模块常用方法。**  
1.得到当前工作目录，即当前Python脚本工作的目录路径: os.getcwd()  
2.返回指定目录下的所有文件和目录名:os.listdir()  
3.函数用来删除一个文件:os.remove()  
4.删除多个目录：os.removedirs（r“c：\python”）  
5.检验给出的路径是否是一个文件：os.path.isfile()  
6.检验给出的路径是否是一个目录：os.path.isdir()  
7.判断是否是绝对路径：os.path.isabs()  
8.检验给出的路径是否真地存:os.path.exists()  
9.返回一个路径的目录名和文件名:os.path.split() eg os.path.split('/home/swaroop/byte/code/poem.txt') 结果：('/home/swaroop/byte/code', 'poem.txt')   
10.分离扩展名：os.path.splitext()  
11.获取路径名：os.path.dirname()  
12.获取文件名：os.path.basename()  
13.运行shell命令: os.system()  
14.读取和设置环境变量:os.getenv() 与os.putenv()  
15.给出当前平台使用的行终止符:os.linesep Windows使用'\r\n'，Linux使用'\n'而Mac使用'\r'  
16.指示你正在使用的平台：os.name 对于Windows，它是'nt'，而对于Linux/Unix用户，它是'posix'  
17.重命名：os.rename（old， new）  
18.创建多级目录：os.makedirs（r“c：\python\test”）  
19.创建单个目录：os.mkdir（“test”）  
20.获取文件属性：os.stat（file）  
21.修改文件权限与时间戳：os.chmod（file）  
22.终止当前进程：os.exit（）  
23.获取文件大小：os.path.getsize（filename）  
**二、文件操作方法大全：**1.os.mknod("test.txt") #创建空文件  
2.fp = open("test.txt",w) #直接打开一个文件，如果文件不存在则创建文件  
3.关于open 模式：

复制代码 代码如下:

w：以写方式打开，  
a：以追加模式打开 (从 EOF 开始, 必要时创建新文件)  
r+：以读写模式打开  
w+：以读写模式打开 (参见 w )  
a+：以读写模式打开 (参见 a )  
rb：以二进制读模式打开  
wb：以二进制写模式打开 (参见 w )  
ab：以二进制追加模式打开 (参见 a )  
rb+：以二进制读写模式打开 (参见 r+ )  
wb+：以二进制读写模式打开 (参见 w+ )  
ab+：以二进制读写模式打开 (参见 a+ )

fp.read([size]) #size为读取的长度，以byte为单位  
fp.readline([size]) #读一行，如果定义了size，有可能返回的只是一行的一部分  
fp.readlines([size]) #把文件每一行作为一个list的一个成员，并返回这个list。其实它的内部是通过循环调用readline()来实现的。如果提供size参数，size是表示读取内容的总长，也就是说可能只读到文件的一部分。  
fp.write(str) #把str写到文件中，write()并不会在str后加上一个换行符  
fp.writelines(seq) #把seq的内容全部写到文件中(多行一次性写入)。这个函数也只是忠实地写入，不会在每行后面加上任何东西。  
fp.close() #关闭文件。python会在一个文件不用后自动关闭文件，不过这一功能没有保证，最好还是养成自己关闭的习惯。 如果一个文件在关闭后还对其进行操作会产生ValueError  
fp.flush() #把缓冲区的内容写入硬盘  
fp.fileno() #返回一个长整型的”文件标签“  
fp.isatty() #文件是否是一个终端设备文件（unix系统中的）  
fp.tell() #返回文件操作标记的当前位置，以文件的开头为原点  
fp.next() #返回下一行，并将文件操作标记位移到下一行。把一个file用于for … in file这样的语句时，就是调用next()函数来实现遍历的。  
fp.seek(offset[,whence]) #将文件打操作标记移到offset的位置。这个offset一般是相对于文件的开头来计算的，一般为正数。但如果提供了whence参数就不一定了，whence可以为0表示从头开始计算，1表示以当前位置为原点计算。2表示以文件末尾为原点进行计算。需要注意，如果文件以a或a+的模式打开，每次进行写操作时，文件操作标记会自动返回到文件末尾。  
fp.truncate([size]) #把文件裁成规定的大小，默认的是裁到当前文件操作标记的位置。如果size比文件的大小还要大，依据系统的不同可能是不改变文件，也可能是用0把文件补到相应的大小，也可能是以一些随机的内容加上去。  
**三、目录操作方法大全**  
1.创建目录  
os.mkdir("file")   
2.复制文件：  
shutil.copyfile("oldfile","newfile") #oldfile和newfile都只能是文件  
shutil.copy("oldfile","newfile") #oldfile只能是文件夹，newfile可以是文件，也可以是目标目录  
3.复制文件夹：  
4.shutil.copytree("olddir","newdir") #olddir和newdir都只能是目录，且newdir必须不存在  
5.重命名文件（目录）  
os.rename("oldname","newname") #文件或目录都是使用这条命令  
6.移动文件（目录）  
shutil.move("oldpos","newpos")   
7.删除文件  
os.remove("file")  
8.删除目录  
os.rmdir("dir") #只能删除空目录  
shutil.rmtree("dir") #空目录、有内容的目录都可以删  
9.转换目录  
os.chdir("path") #换路径