文件操作函数的使用大全

|  |  |
| --- | --- |
| **函数** | **使用说明** |
| os.mknod("test.txt") | 创建空文件 |
| 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 |
| 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，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） |
| fp.close() | 关闭文件 |
| 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） |
| 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 |
| 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 |
| shutil.copyfile("oldfile","newfile") | 复制文件，oldfile和newfile都只能是文件 |
| shutil.copy("oldfile","newfile") | 复制文件，oldfile只能是文件夹，newfile可以是文件，也可以是目标目录 |
| os.rename("oldname","newname") | 重命名文件 |
| shutil.move("oldpos","newpos") | 移动文件 |
| os.remove("file") | 删除文件 |