

文件操作函数的使用大全

<code>os.mknod("test.txt")</code>	创建空文件
<code>fp = open("test.txt",w)</code>	直接打开一个文件，如果文件不存在则创建文件
<code>fp.read([size])</code>	<code>size</code> 为读取的长度，以 <code>byte</code> 为单位
<code>fp.readline([size])</code>	读一行，如果定义了 <code>size</code> ，有可能返回的只是一行的一部分
<code>fp.readlines([size])</code>	把文件每一行作为一个 <code>list</code> 的一个成员，并返回这个 <code>list</code> ，如果提供 <code>size</code> 参数， <code>size</code> 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
<code>fp.write(str)</code>	把 <code>str</code> 写到文件中， <code>write()</code> 并不会在 <code>str</code> 后加上一个换行符
<code>fp.writelines(seq)</code>	把 <code>seq</code> 的内容全部写到文件中（多行一次性写入）
<code>fp.close()</code>	关闭文件
<code>fp.flush()</code>	把缓冲区的内容写入硬盘
<code>fp.fileno()</code>	返回一个长整型的“文件标签”
<code>fp.isatty()</code>	文件是否是一个终端设备文件（Unix 系统）
<code>fp.tell()</code>	返回文件操作标记的当前位置，以文件的开头为原点
<code>fp.next()</code>	返回下一行，并将文件操作标记位移到下一行
<code>fp.seek(offset[,whence])</code>	将文件打操作标记移到 <code>offset</code> 的位置。这个 <code>offset</code> 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 <code>whence</code> 参数就不一定了， <code>whence</code> 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 <code>a</code> 或 <code>a+</code> 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
<code>fp.truncate([size])</code>	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 <code>size</code> 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
<code>shutil.copyfile("oldfile","newfile")</code>	复制文件， <code>oldfile</code> 和 <code>newfile</code> 都只能是文件
<code>shutil.copy("oldfile","newfile")</code>	复制文件， <code>oldfile</code> 只能是文件夹， <code>newfile</code> 可以是文件，也可以是目标目录
<code>os.rename("oldname","newname")</code>	重命名文件

<code>shutil.move("oldpos","newpos")</code>	移动文件
<code>os.remove("file")</code>	删除文件

学习Python