## 文件

文件是计算机中非常重要的东西，在Python里把文件视作一种类型的对象，类似之前学习过的其它类型。文件包括普通的文本，图片，音频，视频等，在linux操作系统中还有“一切皆文件”的说法，也就是说所有的东西都被保存在文件中。所以文件对象不仅可以用来访问普通的磁盘文件, 而且也可以访问任何其它类型抽象层面上的“文件”。

在linux下文件共有七种，**块设备文件(b)，字符设备文件(r)，目录(d)，普通文件(-)，链接文件(l)，套接字文件(s)和管道文件(p)**。可以通过shell命令来查看。

## 打开关闭文件

作为打开文件之门的"钥匙”，内建函数 open() 提供了初始化I/O操作的通用接口。open()内建函数成功打开文件后时候会返回一个文件对象，否则引发一个错误。当操作失败，Python 会产生一个 IOError 异常。

f = open(‘filename’[,’r’,0])

功能：打开一个文件，返回一个文件流对象。

参数：file\_name————文件名；

access\_mode————打开文件的方式,如果不写默认为‘r’

r： 以**读**方式打开 文件必须**存在**

w: 以**写**方式打开,文件不存在则创建，存在则清空原有内容

a: 以**追加**模式打开

r+ : 以**读写**模式打开 文件必须存在

w+ : 以**读写模式**打开文件，不存在则创建，存在则清空原有内容

a+: 以**读写模式**打开 **追加模式**

rb: 以二进制读模式打开 同r

wb ; 以二进制写模式打开 同w

ab: 以二进制追加模式打开 同a

rb+: 以二进制读写模式打开 同r+

wb+: 以二进制读写模式打开 同w+

ab+: 以二进制读写模式打开 同a+

buffering 参数为0 表示无缓冲,1 表示有行缓冲,如果大于1,则直接指明缓冲区大小,不写或者为负数,使用系统默认的缓冲区机制,成功会返回文件流对象,失败则返回IOError

流（stream）：所有的I/O操作仅是简单的从程序移进或者移出，这种字节流，就称为流。

系统已经默认为我们打开了三个流，在sys模块中：

标注输入:sys.stdin

标准输出:sys.stdout

标准错误:sys.stderr

缓冲: 系统自动的在内存中为每一个正在使用的文件开辟一个缓冲区，从内存向磁盘输出数据必须先送到内存缓冲区，装满缓冲区在一起送到磁盘中去。从磁盘中读数据，则一次从磁盘文件将一批数据读入到内存缓冲区中，然后再从缓冲区逐个的将数据送到程序的数据区。

打开一个文件后我们就可以通过文件流对文件进行操作了，当操作结束后使用close（）关闭这个流是一个好习惯，这样可以防止一些误操作，也可以节省资源。此外我们还有一种方法可以不用close:

with open(‘file’,’r’) as f : print “....”

因为with生成的对象在语句块结束后会自动处理，所以也就不需要close了，但是这个流只能在with语句块内使用。

## 读取文件

read([size]):

直接读取字节到字符串中,最多读取给定数个字节,如果括号中没有传参,则使用默认值 -1,或者传入负值,文件将被读至文件末尾, 文件过大时,建议在non-blockiing 模式下使用

readline([size]):

读取打开文件的一行(读取下一个行开始符之前的所有字节),然后将包括该行结束符的整行内容,作为字符串返回,参数默认为-1,代表读至行尾,如果给size赋值,那么在超过size个字节后会返回前size个字节,

readlines():

该方法不返回字符串,该方法会读取所有(剩余)行,然后把他们作为一个字符串列表进行返回,可选参数为返回的最大字节大小,如果大于0,则返回的所有行应该大概有参数的字节(可能会稍微大于这个数字,因为要凑够缓冲区的大小)

## 写入文件

write(string):

将含有文本数据或二进制数据块的字符串写入到文件中

wrritelines(str\_list)

与 readlines()一样,是针对列表的操作,接受一个字符串列表作为参数,见他们写入文件中,行结束符不会自动加入,需要在列表中的每一项元素的末尾加上换行符,才能在写入的文档中换行

## 其他操作

### 缓冲区刷新

flush()函数用于手动刷新缓冲区。

### 文件位置偏移

当我们打开一个文件进行操作时系统会自动生成一个记录，记录中描述了我们对文件的一系列操作。其中包括每次操作到的文件位置。

seek()方法可以在文件中移动文件指针到不同的位置。

语法: seek(offset[,whence])

功能:移动文件位置   
 参数：

offset 代表相对于某个位置偏移量。如果为负数,表示向前移动n个字节。正值表示向后移动n个字节,

whence: 默认值为0,代表从文件开头算起,1代表从当前位置算起,2代表从文件末尾算起.

## 文件内容迭代

在Python2.2中，我们引进了迭代器和文件迭代，这使得一切变得完全不同，文件对象成为了它们自己的迭代器，这意味着用户不必调用 read\*()方法就可以在for循环中迭代文件的每一行。另外我们也可以使用迭代器的next方法，file.next()可以用来读取文件的下一行。和其它迭代器一样，Python也会在所有行迭代完成后引发StopIteration 异常。

语法:

f = open(‘file’)

for i in f:

print i

f.next()

## 查看文件信息

模块os我们之前已经接触过。其实在这个模块中有很多对于文件和文件夹的操作可以供同学们进行挖掘。下面介绍其中一个stat函数。

import os

os.stat(‘file’)