May23

1 文件 file

什么是文件？

文件是数据存储的单位

文件通常用来长期存储数据

文件中的数据是以字节为单位进行顺序存储的

文件的操作流程

1.打开文件

2.读写文件

3.关闭文件

任何操作系统，一个应用程序同时打开文件的数量有最大数限制

文件打开函数open

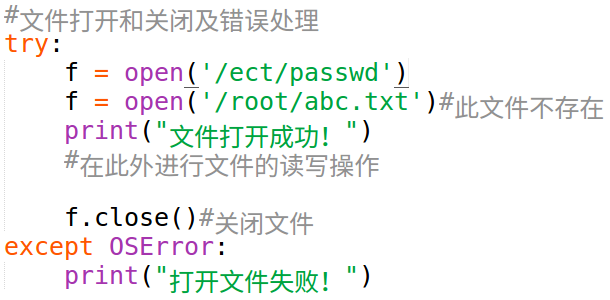
open(file ,mode = ‘rt’)

用于打开一个文件，返回此文件对应的文件流对象，如果打开失败，则会触发OSError错误！

文件关闭方法：

F.close() 关闭，释放系统资源

示例：



1.1 python 文件常用方法

| **方法** | **说明** |
| --- | --- |
| F.close() | 关闭文件(关闭后文件不能再读写会发生ValueError错误) |
| F.readline() | 读取一行数据, 如果到达文件尾则返回空行 |
| F.readlines(max\_chars=-1) | 返回每行字符串的列表,max\_chars为最大字符(或字节)数 |
| F.writelines(lines) | 每行字符串的列表 |
| F.flush() | 把写入文件对象的缓存内容写入到磁盘 |
| F.read(size = -1) | 从一个文件流中最多读取size个字符 |
| F.write(text) | 写一个字符串到文件流中，返回写入的字符数 |

1.2 python文件读写的两种类型

文本文件（text file）

二进制文件（binary file）

**1.2.1文本文件的操作**

默认文件中存储的都为字符数据，以行为单位进行分隔，在python内部统一用 ’\n’ 作为换行进行分隔

对文本文件读写需要用字符串（str）进行数据读取和写入

各种操作系统的换行符

|  |  |
| --- | --- |
| Linux换行符 | ‘\n’ |
| Window换行符 | ‘\r\n’ |
| 旧的Macintosh的换行符 | ‘\r’ |
| 新的MacOs | ‘\n’ |



练习：

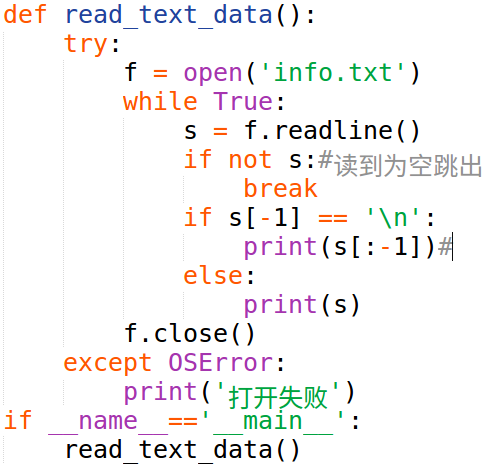
自己写一个文件‘info.txt’内部存一些文字信息

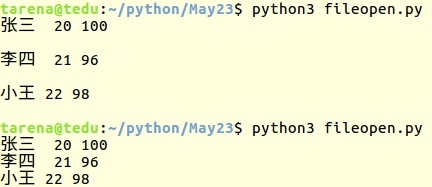
张三 20 100

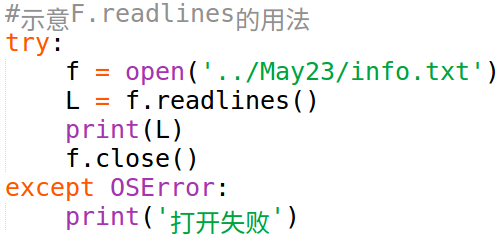
李四 21 96

小王 22 98

写程序读取出来打印在终端







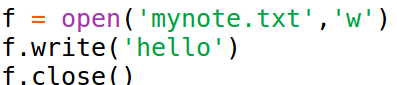


文本文件的写操作

* 缺省模式是 'rt'
* 'w+b' 可以实现二进制随机读写，当打开文件时，文件内容将被清零
* 'r+b' 以二进制读和更新模式打开文件,打开文件时不会清空文件内容
* 'r+' 以文本模式读和更新模式打开文件,打开文件时不会清空文件内容

mode 模式字符的含义

| **字符** | **含义** |
| --- | --- |
| 'r' | 以只读方式打开(默认) |
| 'w' | 以只写方式打开，删除原有文件内容(如果文件不存在，则创建该文件并以只写方式打开) |
| 'x' | 创建一个新文件, 并以写模式打开这个文件,如果文件存在则会产生"FileExistsError"错误 |
| 'a' | 以只写文件打开一个文件，如果有原文件则追加到文件末尾 |
| 'b' | 用二进制模式打开 |
| 't' | 文本文件模式打开 (默认) |
| '+' | 为更新内容打开一个磁盘文件 (可读可写) |



练习：

1.写一个程序，从键盘输入如下信息：

姓名 和 电话号码

如：请输入姓名：xiaozhang

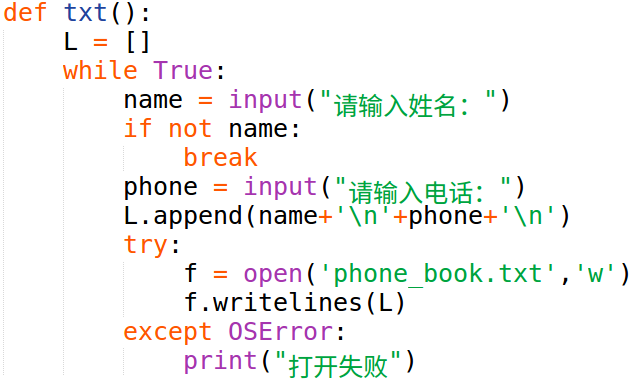
请输入电话：13888888888

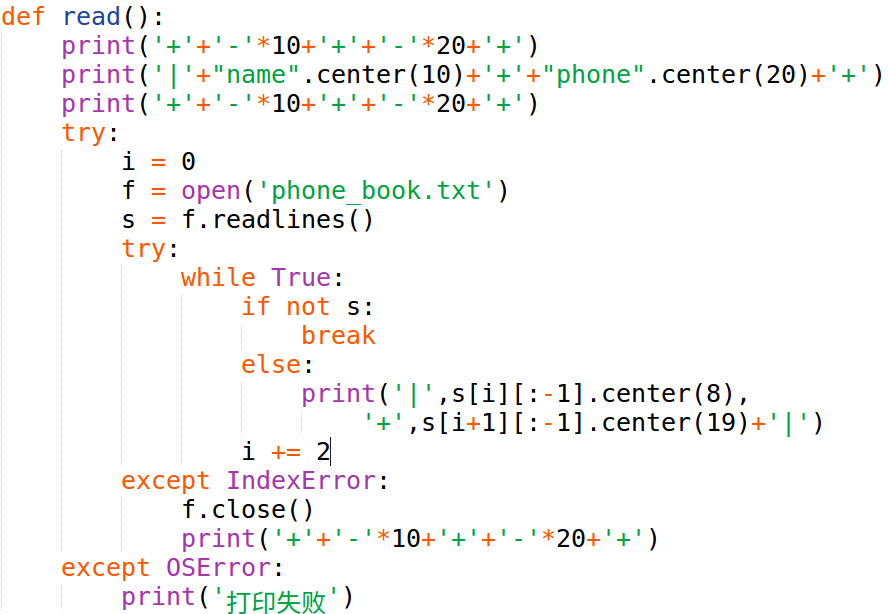
请输入姓名：xiaoli

请输入电话：

把从键盘读取的信息存入文件’phone\_book.txt’中，然后用sublime打开并查看写入的内容

2.写一个读取’phone\_book.txt’文件程序，把保存信息已表格的形式打印出来





文本文件的迭代读取

open（）函数返回来的文件流对象是可迭代对象

示例：

f = open(‘abc.txt’)

#每次取出一行，相当于line = f.readline()

for line in f:

print(line)

**1.2.2 二进制文件的操作**

‘b’二进制(binary)文件操作

对于二进制文件的读写通常需要用字节串(bytes)进行操作

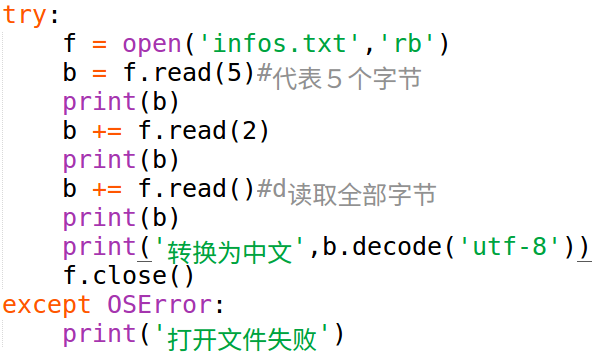
示例：

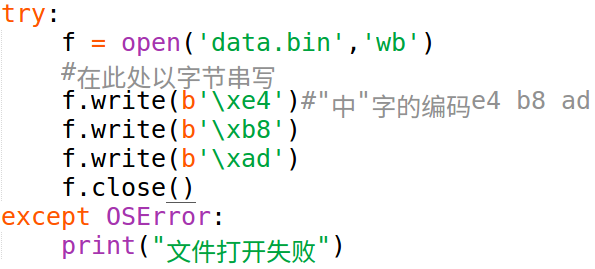
f = open(‘info.txt’,’rb’) #以二进制模式打开

#在此处对f绑定的对象以bytes为单位进行读写

f.close()

|  |  |
| --- | --- |
| 二进制文件操作方法 |  |
| F.tell() | 返回当前文件流的绝对位置（告诉当前读到哪里） |
| F.seek(offset, whence=0) | 改变数据流的位置，返回新的绝对位置 |
| F.readable() | 判断这个文件是否可读,可读返回True,否则返回False |
| F.writable() | 判断这个文件是否可写,可写返回True,否则返回False |
| F.seekable() | 返回这个文件对象是否支持随机定位 |
| F.truncate(pos = None) | 剪掉自pos位置之后的数据，返回新的文件长度(字节为单位) |





**F.seek（）的方法**

F.seek(偏移量，whence=相对位置)

相对位置：

0代表从文件头开始

1代表从当前读写位置开始偏移

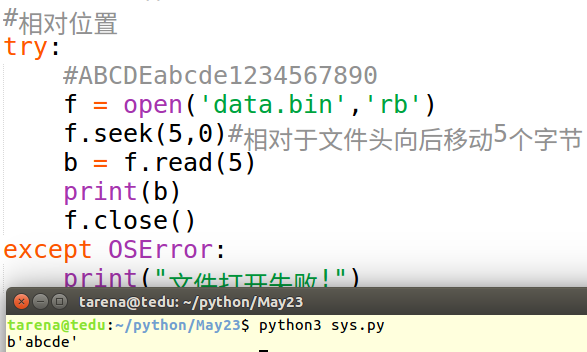
2代表从文件尾开始偏移

偏移量：

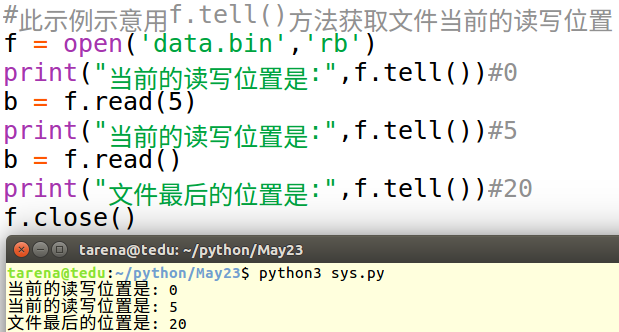
大于0代表向文件末尾方向移动

小于0代表向文件头方向移动

示例：



**tell ( ) 的用法示例：**



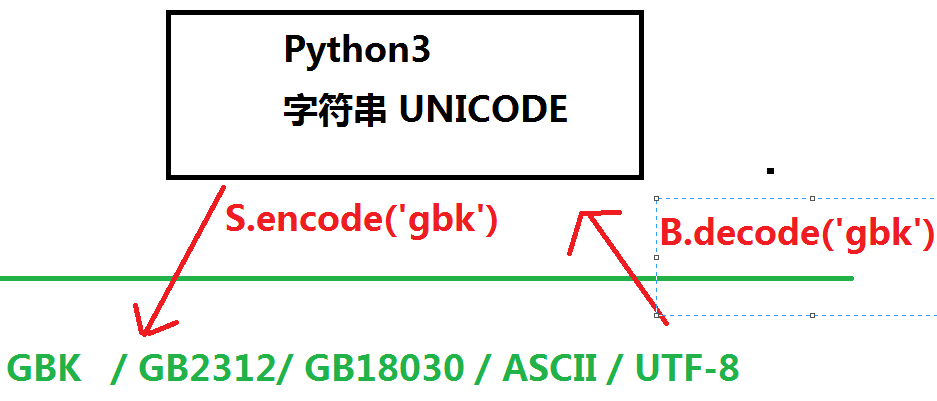
问题：

十个汉字占多少个字节

答案：GBK占20个字节，UTF-8占30个字节

**汉字编码：**

|  |  |
| --- | --- |
| 国标系列 | 国际标准 |
| GB18030（二字节编码或四字节编码） | UNICODE(2/4字节)<---->UTF-8(1~6字节) |
| GBK（二字节编码） | Linux/Mac OS X/IOS/Android常用 |
| GB2312（二字节编码） |  |



python 编码(encode)字符串

‘GB2312’

‘GB18030’

‘GBK’

‘UTF-8’

‘ascii’

编码注释：

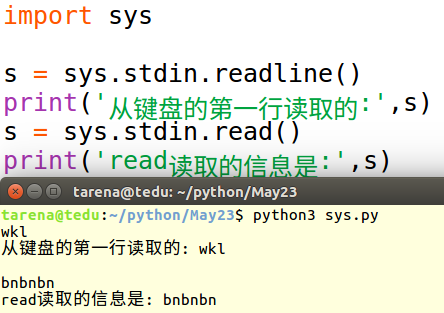
#-\*- coding:gbk -\*- 在python中设置编码格式是GBK编码（顶行写）

#-\*- coding:utf-8 -\*- 在python中设置编码格式是UTF-8

2 标准输入输出文件

模块名：sys

|  |  |
| --- | --- |
| sys.stdin | 默认为标准键盘输入设备（ctrl+D输入末尾标识符） |
| sys.stdout | 默认为屏幕终端 (help(print)) |
| sys.stderr | 默认为屏幕终端 |



字符的转换

练习：

1.写程序让用户输入一系列整数，当输入小于0的数时结束输入。

1）将输入的数字存于列表当中

2）将列表中的数字写入到文件numbers.txt中（需要将整数转换为字符串或字节串）

2.写程序，将文件中numbers.txt中的整数读入到内存中，重新形成数字组成的列表，计算这些数的最大值，最小值，和他们的和

3.为学生信息管理系统项目添加两个功能：

1）保存所有学生信息到文件（si.txt）

2）从文件中读取数据（si.txt）

（建议：以行为单位存数据，用空格或逗号作为姓名，年龄，成绩的分隔符）