



文件的基本处理

黄天羽

北京理工大学





■ 打开文件

- 建立磁盘上的文件与程序中的对象相关联
- 通过相关的文件对象获得

■ 文件操作

- 读取
- 写入
- 定位
- 其他：追加、计算等

■ 关闭文件

- 切断文件与程序的联系

- 写入磁盘，并释放文件缓冲区





打开文件

■ Open ()

`<variable> = open (<name>, <mode>)`

`<name>` 磁盘文件名

`<mode>` 打开模式





打开模式

r	只读。如果文件不存在，则输出错误
w	为只写（如果文件不存在，则自动创建文件
a	表示附加到文件末尾
rb	只读二进制文件。如果文件不存在，则输出错误
wb	只写二进制文件，如果文件不存在，则自动创建文件。
ab	附加到二进制文件末尾

r+ 读写





举例

- 打开一个名为 “numbers . dat” 的文本文件

```
>>> infile = open (“numbers.dat”, “r”)
```

- 打开一个名为 “music.mp3” 的音频文件

```
>>>infile = open (“music.mp3”, “rb”)
```



文件读取

- `read()` 返回值为包含整个文件内容的一个字符串
- `readline()` 返回值为文件下一行内容的字符串。
- `readlines()` 返回值为整个文件内容的列表，每项是以换行符为结尾的一行字符串。



示例：将文件内容输出到屏幕上

```
def main():  
    fname = eval(input("Enter filename: "))  
    infile = open(fname, "r")  
    data = infile.read()  
    print(data)  
  
main()
```



举例：输出文件前5行内容

```
infile = open(someFile, "r")
for i in range(5):
    line = infile.readline()
    print(line[:-1])
```




写入文件

- 从计算机内存向文件写入数据
- `write()`：把含有本文数据或二进制数据块的字符串写入文件中。
- `writelines()`：针对列表操作，接受一个字符串列表作为参数，将它们写入文件。



举例：写入操作

```
>>> outfile = open("outfile.txt ", "w")
>>> outfile.writelines(["Hello", " ", "world"])
>>> outfile.close()
>>> infile = open("outfile.txt", "r")
>>> infile.read()
'Hello world'
```



文件遍历

- 最常见的文件处理方法
- 举例
 - 拷贝文件
 - 根据数据文件定义行走路径
 - 将文件由一种编码转换为另外一种编码





遍历文件模板

■ 通用代码框架：

```
file = open (someFile, "r")  
For line in file.readlines():  
    #处理一行文件内容  
file.close()
```

■ 简化代码框架：

```
file = open (someFile, "r")  
For line in file:  
    #处理一行文件内容  
file.close()
```





举例：文件拷贝

```
def main():  
    # 用户输入文件名  
    f1 = input("Enter a source file:").strip()  
    f2 = input("Enter a source file:").strip()  
  
    # 打开文件  
    infile = open(f1, "r")  
    outfile = open(f2, "w")  
  
    # 拷贝数据  
    countLines = countChars = 0  
    for line in infile:  
        countLines += 1  
        countChars += len(line)  
        outfile.write(line)  
    print(countLines, "lines and", countChars, "chars copied")  
  
    infile.close()  
    outfile.close()  
  
main()
```