

文件

文件概述

## 程序的五大基本指令



## 文件概述

文件是一个存储在辅助存储器上的数据序列，可以包含任何数据内容。概念上，文件是数据的集合和抽象（类似地，函数是程序的集合和抽象）。用文件形式组织和表达数据更有效也更为灵活。文件包括两种类型：**文本文件**和**二进制文件**。

## 文件概述

- 文本文件：只包含基本文本字符，不包含字体、大小和颜色信息
- 二进制文件：是所有其他文件类型，诸如字处理文档、PDF、图像和可执行程序等。
- 二进制文件和文本文件最主要的区别：**是否有统一的字符编码。**  
二进制文件直接由比特0和比特1组成，没有统一字符编码，文件内部数据的组织格式与文件用途有关。
- 无论文件创建为文本文件或者二进制文件，都可以用“文本文件方式”和“二进制文件方式”打开，打开后的操作不同。

## 文件概述——字符编码

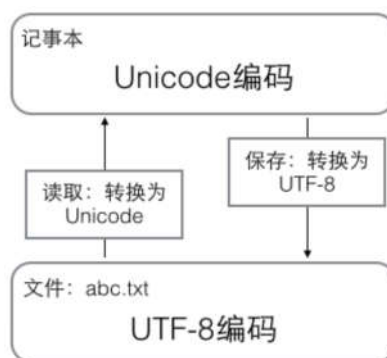
编码：表示符号的过程；

常见编码：ASCII, Unicode, GBK, Shift\_JIS等；

为什么要编码：计算机只能处理数字，如果要处理文本，就必须先把文本转换为数字才能处理；

## 文件概述——字符编码

字符	ASCII	Unicode	UTF-8
A	01000001	00000000 01000001	01000001
中	x	01001110 00101101	11100100 10111000 10101101



## 文件概述

- 采用文本方式读入文件，文件经过编码形成字符串，打印出有含义的字符；
- 采用二进制方式打开文件，文件被解析为字节（byte）流。由于存在编码，字符串中的一个字符由2个字节表示。

```

>>>
中国是个伟大的国家！
b'\xd6\xd0\xb9\xfa\xca\xc7\xb8\xf6\xce\xb0\xb4
\xfb\x5c\x4b9\xfa\xbc\xd2\xa3\xa1'
  
```

# 文件的打开与关闭

## 文件打开与关闭

Python对文本文件和二进制文件采用统一的操作步骤，即“**打开-操作-关闭**”



## 文件打开与关闭

在 Python 中，读写文件有 3 个步骤：

1. 调用 open()函数，返回一个 File 对象。
2. 调用 File 对象的 read()或 write()方法。
3. 调用 File 对象的 close()方法，关闭该文件

## 文件打开

Python通过解释器内置的open()函数打开一个文件，格式如下：

**<变量名> = open(<文件名>, <打开模式>)**

open()函数有两个参数：文件名和打开模式。文件名可以是文件的实际名字，也可以是包含完整路径的名字。注意在打开一个文件时，要向open()函数传递一个字符串路径，表明希望打开的文件。

## 文件打开——路径

文件有两个关键属性：“文件名”（通常写成一个单词）和“路径”。  
路径指明了文件在计算机上的位置。



## 文件打开——路径

制定一个文件的路径有两种方法：

- 绝对路径：总是从根文件夹开始
- 相对路径：它相对于程序的当前工作目录

*# 读取整个文件*

```
File1 = open('D:\\Hello.txt')
```

```
File2 = open('Hello.txt')
```

## 文件打开——打开模式

打开模式	含义
'r'	只读模式，如果文件不存在，返回异常 <code>FileNotFoundError</code> ，默认值
'w'	覆盖写模式，文件不存在则创建，存在则完全覆盖源文件
'x'	创建写模式，文件不存在则创建，存在则返回异常 <code>FileExistsError</code>
'a'	追加写模式，文件不存在则创建，存在则在原文件最后追加内容
'b'	二进制文件模式
't'	文本文件模式，默认值
'+'	与 <code>r/w/x/a</code> 一同使用，在原功能基础上增加同时读写功能

## 文件的读写



## 文件读入

要使用文本文件中的信息，首先需要将信息读取到内存中。为此，你可以一次性读取文件的全部内容，也可以以每次一行的方式逐步读取：

1. `read()`方法就返回保存在该文件中的字符串（将整个文件的内容读取为一个字符串）；
2. `readlines()`方法，从该文件取得一个字符串的列表，每个元素就是文本中的每一行；

## 文件读入

```
In [4]: File2 = open('Hello.txt')

In [5]: File2.read()
Out[5]: '锄禾日当午，\n汗滴禾下土。\n谁知盘中餐，\n粒粒皆辛苦。'

In [6]: File2.close()

In [7]: File2 = open('Hello.txt')

In [8]: File2.readlines()
Out[8]: ['锄禾日当午，\n', '汗滴禾下土。\n', '谁知盘中餐，\n', '粒粒皆辛苦。']

In [9]:
```

## 文件读入

如何进行文本逐行打印？

*# 进行文本逐行打印:*

```
fo = open('hello.txt', 'r')
for line in fo.readlines():
    print(line)
fo.close()
```

*#*

```
fo = open('hello.txt', 'r')
for line in fo:
    print(line)
fo.close()
```

```
In [12]: runfile('C:/Users/
锄禾日当午，
```

```
汗滴禾下土。
```

```
谁知盘中餐，
```

```
粒粒皆辛苦。
```

## 文件写入

Python允许你将内容写入文件，方式与 `print()` 函数将字符串“写”到屏幕上类似。但是，如果打开文件时用读模式，就不能写入文件。你需要以“写入纯文本模式”或“添加纯文本模式”打开该文件，或简称为“覆盖写模式”和“添加模式”。

## 文件写入——覆盖写模式

- 将'w'作为第二个参数传递给 open()，以写模式打开该文件。
- 将覆盖原有的文件，从头开始，就像你用一个新值覆盖一个变量的值。
- 如果传递给open()的文件名不存在，创建一个新的空文件。

## 文件写入——覆盖写模式

```
# 文件写入——覆盖写模式  
fo = open('bacon.txt', 'w')  
fo.write('ABCDEF\n')  
fo.close()
```

## 文件写入——追加写模式

- 在已有文件的末尾添加文本。
- 将'a'作为第二个参数传递给 open(), 以添加模式打开该文件。
- 如果传递给open()的文件名不存在, 创建一个新的空文件。

```
# 文件写入——追加写模式
fo = open('bacon.txt', 'a')
fo.write('ABCDEF\n')
fo.close()
```

## 文件写入——向文件写入列表

```
# 案例: 向文件写入列表
fo = open('new_bacon.txt', 'w+')
ls = ['唐诗', '宋词', '元曲']
fo.writelines(ls)
for line in fo:
    print(line)
fo.close()
```

## 文件写入——向文件写入列表

# 案例：向文件写入列表

```
fo = open('new_bacon.txt', 'w+')
ls = ['唐诗', '宋词', '元曲']
fo.writelines(ls)
for line in fo:
    print(line)
fo.close()
```

# 案例：向文件写入列表

```
fo = open('new_bacon.txt', 'w+')
ls = ['唐诗', '宋词', '元曲']
fo.writelines(ls)
fo.seek(0)
for line in fo:
    print(line)
fo.close()
```

## 文件读写——混合打开模式

常见打开模式：r, w, a, r+, w+, a+

如果打开文档不存在，r会报错，w和a则会创建新文档

详见教程：<https://www.runoob.com/python3/python3-file-methods.html>

## Recall：文件打开与关闭

在 Python 中，读写文件有 3 个步骤：

1. 调用 `open()`函数，返回一个 File 对象。
2. 调用 File 对象的 `read()`或 `write()`方法。
3. 调用 File 对象的 `close()`方法，关闭该文件

## 文件读写——with语句

Python的with语句可自动调用`close()`方法；

```
# with 语句打开文件  
with open('new_bacon.txt', 'r') as f:  
    print(f.read())
```

## 凯撒密码

利用凯撒密码对数据进行加密，并写入文件。

凯撒密码：是一种替换加密的技术，明文中的所有字母都在字母表上向后（或向前）按照一个固定数目进行偏移后被替换成密文。例如，当偏移量是 3 的时候，所有的字母 A 将被替换成 D，B 变成 E，以此类推。

例子：

原文：A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

密文：D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

要求：分别完成 4 个函数以及测试用例：加密 `enCaesar(s, n)`，解密 `deCaesar(s, n)`，读文件 `readFile(filename)`，和写文件 `writeFile(filename, s)`。

其中参数：s 为字符串，n 为偏移数，filename 为文件名。

- 1) 随机生成由 26 个英文字母组成的一串字符，字符个数自定。
- 2) 调用 `enCaesar` 函数对其进行加密，调用 `writeFile` 函数将密文写入文件中。
- 3) 调用 `readFile` 函数将密文从文件中读入，调用函数 `deCaesar` 进行解密并打印输出。