



Python 程序设计基础

Python Programming



打开文件

- **对文件进行操作前必须先打开文件。打开文件的含义是指将文件对象与外存上的文件建立联系。**

`file_object = open(filename, mode)`

`open` 函数第一个参数 `filename` 为文件名；第二个参数 `mode` 为文件模式，指定如何打开文件。返回文件对象 `file_object`。

- **成功打开文件后，对该文件的操作都将通过文件对象来完成。**



文件模式

➡ 在打开文件时，通常需要指定文件模式。文件模式指定如何打开文件。

模式	含义
'r'	只读模式（默认）。文件不存在，抛出 FileNotFoundError 异常
'w'	只写模式。文件不存在则创建；存在则清空文件原有内容
'x'	独占只写模式。文件不存在则创建；存在则抛出 FileExistsError 异常
'a'	只写模式。文件不存在则创建；存在则在文件原有内容末尾追加内容
'b'	二进制文件模式
't'	文本文件模式（默认）
'+'	读写模式。与其他模式组合使用

■ 文件模式可以组合。如 "r+" 以读写方法打开文件， "rb" 以只读方式打开二进制文件。



关闭文件

- 一个程序可以同时打开的文件数目通常是有限的，文件使用完毕后必须关闭文件。

`file_object.close()`

- 关闭文件是将文件对象与外存上的文件脱离联系，释放打开文件时占用的资源。
- 退出程序时会自动关闭所有打开的文件，但最好养成显式关闭文件的习惯。
- 把数据写入文件时，首先把数据存放在内存的缓冲区中，当缓冲区满了或关闭文件时，操作系统自动将当时缓冲区中所有数据真正写入文件，“刷新”缓冲区的操作是自动进行的。程序如果需要，可以强制“刷新”缓冲区。

`file_object.flush()`



文件对象属性

➡ 一旦创建了文件对象，就可以使用如下属性：

· 属性	· 描述
· closed	· 若文件已经关闭，返回 True，否则返回 False
· mode	· 返回文件模式
· name	· 返回文件名

文件对象的属性

```
fo = open("foo.txt", "w")      # 打开文件
print("文件名: ", fo.name)
print("是否关闭: ", fo.closed)
print("打开模式: ", fo.mode)
fo.close()                     # 关闭文件
```

```
文件名:  foo.txt
是否关闭:  False
打开模式:  w
```