

## 5.17 将字节写入文本文件¶

### 问题¶

你想在文本模式打开的文件中写入原始的字节数据。

### 解决方案¶

将字节数据直接写入文件的缓冲区即可，例如：

```
>>> import sys
>>> sys.stdout.write(b'Hello\n')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: must be str, not bytes
>>> sys.stdout.buffer.write(b'Hello\n')
Hello
5
>>>
```

类似的，能够通过读取文本文件的 `buffer` 属性来读取二进制数据。

### 讨论¶

I/O系统以层级结构的形式构建而成。文本文件是通过在一个拥有缓冲的二进制模式文件上增加一个Unicode编码/解码层来创建。`buffer` 属性指向对应的底层文件。如果你直接访问它的话就会绕过文本编码/解码层。

本小节例子展示的 `sys.stdout` 可能看起来有点特殊。默认情况下，`sys.stdout` 总是以文本模式打开的。但是如果你在写一个需要打印二进制数据到标准输出的脚本的话，你可以使用上面演示的技术来绕过文本编码层。