

# Python open 读写文件实现脚本

2010-08-06 10:44

Python 中文件操作可以通过 open 函数,这的确很像 C 语言中的 fopen。通过 open 函数获取一个 file object, 然后调用 read(), write() 等方法对文件进行读写操作。

## 1. open

使用 open 打开文件后一定要记得调用文件对象的 close() 方法。比如可以用 try/finally 语句来确保最后能关闭文件。

```
file_object = open('thefile.txt')
try:
    all_the_text = file_object.read()
finally:
    file_object.close()
```

注: 不能把 open 语句放在 try 块里, 因为当打开文件出现异常时, 文件对象 file\_object 无法执行 close() 方法。

## 2. 读文件

### 读文本文件

```
input = open('data', 'r')
#第二个参数默认为 r
input = open('data')
```

### 读二进制文件

```
input = open('data', 'rb')
```

### 读取所有内容

```
file_object = open('thefile.txt')
try:
    all_the_text = file_object.read()
finally:
    file_object.close()
```

## 读固定字节

```
file_object = open('abinfile', 'rb')
try:
    while True:
        chunk = file_object.read(100)
        if not chunk:
            break
        do_something_with(chunk)
finally:
    file_object.close()
```

## 读每行

```
list_of_all_the_lines = file_object.readlines()
```

如果文件是文本文件，还可以直接遍历文件对象获取每行：

```
for line in file_object:
    process line
```

## 3. 写文件

### 写文本文件

```
output = open('data', 'w')
```

### 写二进制文件

```
output = open('data', 'wb')
```

### 追加写文件

```
output = open('data', 'w+')
```

### 写数据

```
file_object = open('thefile.txt', 'w')
file_object.write(all_the_text)
```

```
file_object.close()
```

## 写入多行

```
file_object.writelines(list_of_text_strings)
```

注意，调用 `writelines` 写入多行在性能上会比使用 `write` 一次性写入要高。

`r` 以只读模式打开文件

`w` 以只写模式打开文件，且先把文件内容清空（truncate the file first）

`a` 以添加模式打开文件，写文件的时候总是写到文件末尾，用 `seek` 也无用。打开的文件也是不能读的

`r+` 以读写方式打开文件，文件可读可写，可写到文件的任何位置

`w+` 和 `r+` 不同的是，它会 truncate the file first

`a+` 和 `r+` 不同的是，它只能写到文件末尾