目 录

[第1章 Python3 文件操作 1](#_Toc481615662)

[1.1 对文件操作流程 1](#_Toc481615663)

[1.2 File(文件) 方法 1](#_Toc481615664)

[1.2.1 file.close() 1](#_Toc481615665)

[1.2.2 file.flush() 1](#_Toc481615666)

[1.2.3 file.fileno() 1](#_Toc481615667)

[1.2.4 file.isatty() 1](#_Toc481615668)

[1.2.5 file.next() 1](#_Toc481615669)

[1.2.6 file.read([size]) 2](#_Toc481615670)

[1.2.7 file.readline([size]) 2](#_Toc481615671)

[1.2.8 file.readlines([sizeint]) 2](#_Toc481615672)

[1.2.9 file.seek(offset[, whence]) 2](#_Toc481615673)

[1.2.10 file.tell() 2](#_Toc481615674)

[1.2.11 file.truncate([size]) 2](#_Toc481615675)

[1.2.12 file.write(str) 2](#_Toc481615676)

[1.2.13 file.writelines(sequence) 2](#_Toc481615677)

[1.2.14 三级示例 3](#_Toc481615678)

# Python3 文件操作

## 对文件操作流程

对文件操作流程：

* 打开文件，得到文件句柄并赋值给一个变量
* 通过句柄对文件进行操作
* 关闭文件

## File(文件) 方法

file 对象使用 open 函数来创建，下表列出了 file 对象常用的函数：

### file.close()

关闭文件。关闭后文件不能再进行读写操作。

### file.flush()

刷新文件内部缓冲，直接把内部缓冲区的数据立刻写入文件, 而不是被动的等待输出缓冲区写入。

### file.fileno()

返回一个整型的文件描述符(file descriptor FD 整型), 可以用在如os模块的read方法等一些底层操作上。

### file.isatty()

如果文件连接到一个终端设备返回 True，否则返回 False。

### file.next()

返回文件下一行。

### file.read([size])

从文件读取指定的字节数，如果未给定或为负则读取所有。

### file.readline([size])

读取整行，包括 "\n" 字符。

### file.readlines([sizeint])

读取所有行并返回列表，若给定sizeint>0，返回总和大约为sizeint字节的行, 实际读取值可能比 sizeint 较大, 因为需要填充缓冲区。

### file.seek(offset[, whence])

设置文件当前位置

### file.tell()

返回文件当前位置。

### file.truncate([size])

截取文件，截取的字节通过size指定，默认为当前文件位置。

### file.write(str)

将字符串写入文件，没有返回值。

### file.writelines(sequence)

向文件写入一个序列字符串列表，如果需要换行则要自己加入每行的换行符。

### 三级示例

/usr/sbin/groupadd mysql

/usr/sbin/useradd -g mysql mysql

tar zxf mysql-5.5.3-m3.tar.gz