

## 13.1 通过重定向/管道/文件接受输入 ¶

### 问题 ¶

你希望你的脚本接受任何用户认为最简单的输入方式。包括将命令行的输出通过管道传递给该脚本、重定向文件到该脚本，或在命令行中传递一个文件名或文件名列表给该脚本。

### 解决方案 ¶

Python内置的 `fileinput` 模块让这个变得简单。如果你有一个下面这样的脚本：

```
#!/usr/bin/env python3
import fileinput

with fileinput.input() as f_input:
    for line in f_input:
        print(line, end="")
```

那么你就能以前面提到的所有方式来为此脚本提供输入。假设你将此脚本保存为 `filein.py` 并将其变为可执行文件，那么你可以像下面这样调用它，得到期望的输出：

```
$ ls | ./filein.py      # Prints a directory listing to stdout.
$ ./filein.py /etc/passwd # Reads /etc/passwd to stdout.
$ ./filein.py < /etc/passwd # Reads /etc/passwd to stdout.
```

### 讨论 ¶

`fileinput.input()` 创建并返回一个 `FileInput` 类的实例。该实例除了拥有一些有用的帮助方法外，它还可被当做一个上下文管理器使用。因此，整合起来，如果我们要写一个打印多个文件输出的脚本，那么我们需要在输出中包含文件名和行号，如下所示：

```
>>> import fileinput
>>> with fileinput.input('/etc/passwd') as f:
>>>     for line in f:
...     print(f.filename(), f.lineno(), line, end="")
...
/etc/passwd 1 ##
/etc/passwd 2 # User Database
/etc/passwd 3 #

<other output omitted>
```

通过将它作为一个上下文管理器使用，可以确保它不再使用时文件能自动关闭，而且我们在之后还演示了 `FileInput` 的一些有用的帮助方法来获取输出中的一些其他信息。