

# Python 读取文本文件

**摘要：**在本教程中，您将学习在 Python 中读取文本文件的各种方法。

## TL;博士

下面显示了如何将 `readme.txt` 文件中的所有文本读入字符串：

```
with open('readme.txt') as f:  
    lines = f.readlines()
```

## 在 Python 中读取文本文件的步骤

要在 Python 中读取文本文件，请执行以下步骤：

- 首先，使用该 `open()` 函数打开一个文本文件进行读取。
- 其次，使用 file 、或 file 对象的方法从文本文件中 `read()` 读取 `readline()` 文本 `readlines()` 。
- 三、使用file `close()` 方法关闭文件。

### 1) open() 函数

该 `open()` 函数有许多参数，但您将关注前两个：

```
open(path_to_file, mode)
```

该 `path_to_file` 参数指定文本文件的路径。

如果程序和文件在同一个文件夹中，则只需指定文件的文件名。否则，您需要包含文件的路径以及文件名。

'/' 要指定文件的路径，即使在 Windows 上工作，也要使用正斜杠 ( )。

例如，如果文件作为程序 `readme.txt` 存储在 `sample` 文件夹中，则需要将文件的路径指定为

`c:/sample/readme.txt`

是 `mode` 可选参数。它是一个字符串，用于指定您要打开文件的模式。下表显示了打开文本文件的可用模式：

模式	描述
'r'	打开文本文件以读取文本
'w'	打开一个文本文件以写入文本
'a'	打开一个文本文件以附加文本

例如，要打开名称 `the-zen-of-python.txt` 与程序存储在同一个文件夹中的文件，请使用以下代码：

```
f = open('the-zen-of-python.txt','r')
```

该 `open()` 函数返回一个文件对象，您将使用它从文本文件中读取文本。

## 2) 阅读文本方法

文件对象为您提供了三种从文本文件中读取文本的方法：

- `read(size)` – 根据可选大小读取文件的某些内容并将内容作为字符串返回。如果省略大小，则该 `read()` 方法会从中断处读取，直到文件末尾。如果已到达文件末尾，则该 `read()` 方法返回一个空字符串。
- `readline()` – 从文本文件中读取一行并将该行作为字符串返回。如果已到达文件末尾，则 `readline()` 返回一个空字符串。
- `readlines()` – 将文本文件的所有行读入字符串列表。如果您有一个小文件并且想要操作该文件的整个文本，则此方法很有用。

## 3) close() 方法

您打开的文件将保持打开状态，直到您使用该 `close()` 方法关闭它。

由于以下原因，关闭不再使用的文件很重要：

- 首先，当您在脚本中打开文件时，文件系统通常会将其锁定，因此在您关闭它之前，其他程序或脚本无法使用它。
- 其次，您的文件系统具有有限数量的文件描述符，您可以在它们用完之前创建它们。尽管这个数字可能很高，但可能会打开大量文件并耗尽您的文件系统资源。
- 第三，打开多个文件可能会导致竞争条件，当多个[进程 \(https://www.pythontutorial.net/python-concurrency/differences-between-processes-and-threads/\)](https://www.pythontutorial.net/python-concurrency/differences-between-processes-and-threads/) 尝试同时修改一个文件时，可能会导致各种意外行为。

下面展示了如何调用 `close()` 关闭文件的方法：

```
f.close()
```

要在不调用该 `close()` 方法的情况下自动关闭文件，请使用如下 `with` 语句：

```
with open(path_to_file) as f:  
    contents = f.readlines()
```

在实践中，您将使用该 `with` 语句自动关闭文件。

## 读取文本文件示例

我们将使用 `-zen-of-python.txt` (<https://www.pythontutorial.net/wp-content/uploads/2020/10/the-zen-of-python.txt>) 文件进行演示。

以下示例说明如何使用该 `read()` 方法将文件的所有内容读入字符串：

```
with open('the-zen-of-python.txt') as f:  
    contents = f.read()  
    print(contents)
```

输出：

```
Beautiful is better than ugly.  
Explicit is better than implicit.  
Simple is better than complex.  
...
```

以下示例使用该 `readlines()` 方法读取文本文件并将文件内容作为字符串列表返回：

```
with open('the-zen-of-python.txt') as f:  
    [print(line) for line in f.readlines()]
```

输出：

```
Beautiful is better than ugly.  
  
Explicit is better than implicit.  
  
Simple is better than complex.  
  
Complex is better than complicated.  
  
...
```

您在文件的每一行之后看到一个空行的原因是文本文件中的每一行都有一个换行符 (`\n`)。要删除空行，可以使用 `strip()` 方法。例如：

```
with open('the-zen-of-python.txt') as f:  
    [print(line.strip()) for line in f.readlines()]
```

以下示例显示如何使用 `readline()` 逐行读取文本文件：

```
with open('the-zen-of-python.txt') as f:
    while True:
        line = f.readline()
        if not line:
            break
        print(line.strip())
```

输出：

```
Explicit is better than implicit.
Complex is better than complicated.
Flat is better than nested.
...
```

## 一种更简洁的逐行读取文本文件的方法

该 `open()` 函数返回一个文件对象，它是一个可迭代 (<https://www.pythontutorial.net/python-basics/python-iterables/>) 的对象。因此，您可以使用 `for` 循环来遍历文本文件的行，如下所示：

```
with open('the-zen-of-python.txt') as f:
    for line in f:
        print(line.strip())
```

这是一种更简洁的逐行读取文本文件的方法。

## 读取 UTF-8 文本文件

前面示例中的代码适用于 ASCII 文本文件。但是，如果您正在处理其他语言，例如日语、中文和韩语，则文本文件不是简单的 ASCII 文本文件。它很可能是一个 UTF-8 文件，它使用的不仅仅是标准的 ASCII 文本字符。

要打开 UTF-8 文本文件，您需要将传递 `encoding='utf-8'` 给 `open()` 函数以指示它从文件中接收 UTF-8 字符。

对于演示，您将使用以下 `quotes.txt` (<https://www.pythontutorial.net/wp-content/uploads/2020/10/quotes.txt>) 包含一些日语引号的文件。

下面展示了如何循环 `quotes.txt` 文件：

```
with open('quotes.txt', encoding='utf8') as f:
    for line in f:
        print(line.strip())
```

输出：

```
人生で何度も何度も失敗を繰り返してきました。だからこそ、私は成功を収めることができたのです。
どれだけ高く登れたかで人を評価しません。尻餅をついたあと、どれだけ変わったかで評価をするのです。
成功を収める人とは人が投げたきたレンガでしっかりした基盤を築くことができる人のことである。
```

## 概括

- 使用 `open()` 带有 `'r'` mode 的函数打开一个文本文件进行阅读。
- 使用 `read()`、`readline()` 或 `readlines()` 方法读取文本文件。
- `close()` 使用方法或 `with` 语句完成读取后始终关闭文件。
- 使用 `encoding='utf-8'` 读取 UTF-8 文本文件。