

## 5.19 创建临时文件和文件夹

### 问题

你需要在程序执行时创建一个临时文件或目录，并希望使用完之后可以自动销毁掉。

### 解决方案

`tempfile` 模块中有很多的函数可以完成这任务。为了创建一个匿名的临时文件，可以使用 `tempfile.TemporaryFile`：

```
from tempfile import TemporaryFile

with TemporaryFile('w+t') as f:
    # Read/write to the file
    f.write('Hello World\n')
    f.write('Testing\n')

    # Seek back to beginning and read the data
    f.seek(0)
    data = f.read()

# Temporary file is destroyed
```

或者，如果你喜欢，你还可以像这样使用临时文件：

```
f = TemporaryFile('w+t')
# Use the temporary file
...
f.close()
# File is destroyed
```

`TemporaryFile()` 的第一个参数是文件模式，通常来讲文本模式使用 `w+t`，二进制模式使用 `w+b`。这个模式同时支持读和写操作，在这里是很有用的，因为当你关闭文件去改变模式的时候，文件实际上已经不存在了。

`TemporaryFile()` 另外还支持跟内置的 `open()` 函数一样的参数。比如：

```
with TemporaryFile('w+t', encoding='utf-8', errors='ignore') as f:
    ...
```

在大多数Unix系统上，通过 `TemporaryFile()` 创建的文件都是匿名的，甚至连目录都没有。如果你想打破这个限制，可以使用 `NamedTemporaryFile()` 来代替。比如：

```
from tempfile import NamedTemporaryFile

with NamedTemporaryFile('w+t') as f:
    print('filename is:', f.name)
    ...

# File automatically destroyed
```

这里，被打开文件的 `f.name` 属性包含了该临时文件的文件名。当你需要将文件名传递给其他代码来打开这个文件的时候，这个就很有用了。和 `TemporaryFile()` 一样，结果文件关闭时会被自动删除掉。如果你不想这么做，可以传递一个关键字参数 `delete=False` 即可。比如：

```
with NamedTemporaryFile('w+t', delete=False) as f:
    print('filename is:', f.name)
    ...
```

为了创建一个临时目录，可以使用 `tempfile.TemporaryDirectory()`。比如：

```
from tempfile import TemporaryDirectory

with TemporaryDirectory() as dirname:
    print('dirname is:', dirname)
```

```
# Use the directory
...
# Directory and all contents destroyed
```

## 讨论

`TemporaryFile()`、`NamedTemporaryFile()` 和 `TemporaryDirectory()` 函数 应该是处理临时文件目录的最简单的方式了，因为它们会自动处理所有的创建和清理步骤。在一个更低的级别，你可以使用 `mkstemp()` 和 `mkdtemp()` 来创建临时文件和目录。比如：

```
>>> import tempfile
>>> tempfile.mkstemp()
(3, '/var/folders/7W/7WZl5sfZEF0pljrEB1UMWE+++TI/-Tmp-/tmp7fefhv')
>>> tempfile.mkdtemp()
'/var/folders/7W/7WZl5sfZEF0pljrEB1UMWE+++TI/-Tmp-/tmp5wvcv6'
>>>
```

但是，这些函数并不会做进一步的管理了。例如，函数 `mkstemp()` 仅仅就返回一个原始的OS文件描述符，你需要自己将它转换为一个真正的文件对象。同样你还需要自己清理这些文件。

通常来讲，临时文件在系统默认的位置被创建，比如 `/var/tmp` 或类似的地方。为了获取真实的位置，可以使用 `tempfile.gettempdir()` 函数。比如：

```
>>> tempfile.gettempdir()
'/var/folders/7W/7WZl5sfZEF0pljrEB1UMWE+++TI/-Tmp-'
>>>
```

所有和临时文件相关的函数都允许你通过使用关键字参数 `prefix`、`suffix` 和 `dir` 来自定义目录以及命名规则。比如：

```
>>> f = NamedTemporaryFile(prefix='mytemp', suffix='.txt', dir='/tmp')
>>> f.name
'/tmp/mytemp8ee899.txt'
>>>
```

最后还有一点，尽可能以最安全的方式使用 `tempfile` 模块来创建临时文件。包括仅给当前用户授权访问以及在文件创建过程中采取措施避免竞态条件。要注意的是不同的平台可能会不一样。因此你最好阅读 [官方文档](#) 来了解更多的细节。