

# Python

## Context Manager

by laike9m

laike9m@gmail.com

<https://github.com/laike9m>

# 什么是上下文管理器

- 传统的打开文件方式

```
file = open(filename)
```

```
file.readline()
```

```
file.close()
```

- 使用上下文管理器

```
with open(filename) as file:
```

```
    file.readline()
```

省下了file.close()

# 上下文管理器的实质

- 上下文管理器就是在某个对象上定义 `__enter__` 和 `__exit__` 方法
- 上面的上下文管理器基于 `file` 对象的 `__exit__()` 特殊方法。当我们使用上下文管理器的语法时，我们实际上要求 **Python** 在进入程序块之前调用对象的 `__enter__()` 方法，在结束程序块的时候调用 `__exit__()` 方法。对于文件对象 `f` 来说，它定义了 `__enter__()` 和 `__exit__()` 方法(可以通过 `dir(f)` 看到)。在 `f` 的 `__exit__()` 方法中，有 `self.close()` 语句。所以在使用上下文管理器时，我们就不用明文关闭 `f` 文件了。

# 看一个例子：切换工作目录

- 如果我们想频繁地进入其它目录执行某个命令，怎么做？
- 切换目录：`import os; os.chdir(some_other_directory)`

# 切换工作目录

```
import os

class cd:
    def __init__(self, newPath):
        self.newPath = newPath

    def __enter__(self):
        self.savedPath = os.getcwd()
        os.chdir(self.newPath)

    def __exit__(self, etype, value, traceback):
        os.chdir(self.savedPath)

with cd('wav'):
    os.system('python truncate_audio.py')

with cd('wav_processing'):
    os.system('python processing.py')

with cd('Texttiling'):
    os.system('python myTexttiling.py')

with cd('LexicalChain'):
    os.system('python get_chain_STT.py')

with cd('Pitch'):
    os.system('python get_pitch_forstt.py')

with cd('train'):
    os.system('python make_feature_dict.py')
    os.system('python write_train_file.py')
```

Q&A