Python编程技巧－使用状态机 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/181/2021_2022_Python_E7_BC_96_E7_c103_181699.htm 状态机从理论上说是几乎与计算机和编程相关的每件事的基础。从实用角度来看，状态机还有助于解决许多常见问题（特别适用于 Python 程序员）。本文中，David Mertz 讨论了何时以及如何使用 Python 编码状态机的实际例子。 什么是 Python？ Python 是由 Guido van Rossum 开发的免费高级解释型语言。其语法简单易懂，而其面向对象的语义功能强大（但又灵活）。Python 可以广泛使用并具有高度的可移植性。 什么是状态机？ 关于状态机的一个极度确切的描述是它是一个有向图形，由一组节点和一组相应的转移函数组成。状态机通过响应一系列事件而“ 运行 ”。每个事件都在属于“ 当前 ”节点的转移函数的控制范围内，其中函数的范围是节点的一个子集。函数返回“ 下一个 ”（也许是同一个）节点。这些节点中至少有一个必须是终态。当到达终态，状态机停止。 但一个抽象的数学描述（就像我刚给出的）并不能真正说明在什么情况下使用状态机可以解决实际编程问题。另一种策略就是将状态机定义成一种强制性编程语言，其中节点也是源码行。从实用角度看，这个定义尽管精确，但它和第一种描述一样，都是纸上谈兵、毫不实用。（对于说明型、函数型或基于约束的语言，例如，Haskell、Scheme 或 Prolog，不一定会发生这种情况。） 让我们尝试使用更适合身边实际任务的例子来进行讨论。逻辑上，每个规则表达式都等价于一个状态机，而每个规则表达式的语法分析器都实现这个状态机。实际上，大多数程序员

编写状态机时，并没有真正考虑到这一点。 在以下这个例子中，我们将研究状态机的真正探索性定义。通常，我们有一些不同的方法来响应一组有限数量的事件。某些情况下，响应只取决于事件本身。但在其它情况下，适当的操作取决于以前的事件。 本文中讨论的状态机是高级机器，其目的是演示一类问题的编程解决方案。如果有必要按响应事件行为的类别来讨论编程问题，那么您的解决方案很可能是显式状态机。文本处理状态机最可能调用显式状态机的一个编程问题涉及到处理文本文件。处理文本文件通常包括读取信息单元（通常叫做字符或行），然后对刚读取的单元执行适当操作。某些情况下，这个处理是“ 无状态的 ”（即每个这样的单元都包含了足够的信息，可以正确确定要执行什么操作）。在其它情况下，即使文本文件不是完全无状态，数据也只有有限的上下文（例如，操作取决于不比行号更多的信息）。但是，在其它常见文本处理问题中，输入文件是极具“ 状态 ”的。每一块数据的含义取决于它前面的字符串（也许是它后面的字符串）。报告、大型机数据输入、可读文本、编程源文件和其它种类的文本文件都是有状态的。一个简单例子是可能出现在 Python 源文件中的一行代码：myObject = SomeClass(this, that, other) 这行表示，如果恰好有以下几行围绕着这一行，则有部分内容不同："""How to use SomeClass:myObject = SomeClass(this, that, other)""" 我们应知道我们处于“ 块引用 ” 状态 以确定这行代码是一部分注释而不是 Python 操作。何时不使用状态机当开始为任何有状态的文本文件编写处理器的任务时，问一问自己，您希望在文件中找到什么类型的输入项。每种类型的输入项都是一种状态的

候选项。这些类型共有几种。如果数字很大或者不确定，则状态机也许不是正确的解决方法。（在这种情况下，某些数据库解决方案可能更适合。）还请考虑您是否需要使用状态机。许多情况下，最好从更简单的方法入手。也许会发现即使文本文件是有状态的，也有一种简单的方法可以分块读取它（其中每一块是一种类型的输入值）。实际上，在单一状态块中，仅当文本类型之间的转移需要基于内容的计算时，才有必要实现状态机。下面这个简单的例子说明了需要使用状态机的情况。请考虑用于将一列数字划分成几块的两个规则。在第一个规则中，列表中的零表示块之间的间断。第二个规则中，当一个块中的元素总和超过 100 时，会发生块之间的间断。由于它使用累加器变量来确定是否达到了阈值，您不能“ 马上 ”看到子列表的边界。因此，第二个规则也许更适合于类似于状态机的机制。稍微有些状态、但由 不 太适合用状态机处理的文本文件的例子是 Windows 风格的 .ini 文件。这种文件包括节头、注释和许多赋值。例如： . set the colorscheme and userlevel[colorscheme]background=redforeground=bluetitle=green [userlevel]login=2title=1 我们的例子没有实际含义，但它表明了 .ini 格式一些有趣的特性。 就某种意义而言，每一行的类型由它的第一个字符确定（可能是分号、左花括号或字母）。 从另一种角度看，这种格式是“ 有状态的 ”，因为关键字 "title" 大概表示如果它出现在每一节中，那么就有独立的内容。 您可以编写一个有 COLORSCHEME 状态和 USERLEVEL 状态的文本处理器程序，这个程序仍处理每个状态的赋值。但这好象不是处理此问题的 正确 方法。例如，可以使用

Python 代码在这个文本文件中只创建自然块，如： 处理 .INI 文件的分块 Python 代码 import stringtxt = open(hypothetical.ini).read()sects = string.split(txt, [) for sect in sects: # do something with sect, like get its name # (the stuff up to]) and read its assignments 或者，如果愿意，可以使用单个 current_section 变量来确定位置： 处理 .INI 文件的计算 Python 代码 for line in open(hypothetical.ini).readlines(): if line[0] == [: current_section = line(1:-2) elif line[0] == .: pass # ignore comments else : apply_value(current_section, line) 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com