

用Oracle和Python武装你的头脑(4) PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/143/2021\\_2022\\_\\_E7\\_94\\_A8Oracle\\_E5\\_c102\\_143099.htm](https://www.100test.com/kao_ti2020/143/2021_2022__E7_94_A8Oracle_E5_c102_143099.htm)

编写脚本 现在，您已经以交互方式试用了解释器，下面您将准备编写一个 python 脚本来处理整个文件。使用 ctrl-d（在 unix/linux 中）或 ctrl-z（在 windows 中）退出 python 解释器，然后创建一个文本文件 - 称之为 readinitora.py。

```
initfile = open( ' init_orcl.ora ' , ' r ' )
initparams = {}
rawtextline = initfile.readline()
while rawtextline:
    param, val = rawtextline.strip().split( ' = ' )
    initparams[param] = val
rawtextline = initfile.readline()
print initparams
```

在阅读此代码时，您很可能想到“命令结尾的分号哪去了？begin 和 end 语句以及用于标记块的大括号哪去了？”实际上，python 并不使用这些符号。按照惯例，几乎所有编程人员都使用换行符分隔命令，并使用缩进指示代码块，以使用户可以阅读代码。同时，python 以外的大多数语言都要求他们使用一套不同的符号与编译器或解释器进行通信。使用两套符号将使代码杂乱不堪，并将产生一个很大的风险 - 显示给人眼的符号可能与显示给计算机的符号不同步，从而产生另人难以理解的错误。而 python 可以像用户那样读取代码，从而消除了混乱并避免出现风险。下面我们来看看代码操作。在操作系统提示符（而非 python 解释器提示符）下，键入 c:\> python readinitora.py { ' undo\_management ' : ' auto ' , ' compatible ' : ' 9.2.0.0.0 ' , ' db\_block\_size ' : ' 8192 ' , ' fast\_start\_mttr\_target ' : ' 300 ' , ' sga\_max\_size ' : 157286400 , ' shared\_pool\_size ' : ' 50331648 ' }

如果您习惯

将 readinitora.py 中的所有行缩进几个空格，则将使 python 混淆并产生一个语法错误。缩进指示代码块，因此在不指示代码块的情况下请不要进行缩进。返回并确保不在块中的每一行都从第 1 列开始。您实际上需要在多个位置使用此代码，因此我们将其从简单的脚本改进为接受参数的函数定义。

```
def read(filename):  
    initfile = open(filename, 'r')  
    initparams = {}  
    rawtextline = initfile.readline()  
    while rawtextline:  
        param, val = rawtextline.strip().split(' = ')  
        initparams[param] = val  
    rawtextline = initfile.readline()  
    return initparams
```

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](http://www.100test.com)