

PCS103 缩进

概述

缩进可谓是 Python 的一大特色，估计再没有其他语言使用缩进作为它语法结构的一部分了。

应用

```
In [5]: f = open('words')

In [6]: result = dict()

In [7]:

In [8]: for line in f.readlines():
...:     line = line.strip()
...:     tmlist = line.split()
...:     for w in tmlist:
...:         if result.has_key(w):
...:             result[w] += 1
...:         else:
...:             result[w] = 1
...:

In [9]: for k, v in result.items():
...:     print k, '-->', v
```

Python 语言是利用缩进来区分不同的块的。不同的块具有不同的作用域，所以在 Python 中，缩进的功能相当于 C 或 Java 中的 {}。在上述例子中，三个 for 语句各有自己的块作

用域，它们很清晰地利用每行代码的缩进来区分自己属于哪个块。同一个块中具有一致的代码缩进格式，因此 Python 代码文件的结构看起来非常清晰。若本应属于同一块内的代码没有相同的缩进，那么，Python 解释器会认为她们属于不同的语义块，因而会引起异常。

```
In [10]: f = open(' words' )

In [11]: result = dict()

In [12]: for line in f.readlines():
...:     line = line.strip()
...:     tmlist = line.split()
...:     for w in tmlist:
...:         if result.has_key(w):
...:             result[w] += 1
...:         else:
...:             result[w] = 1
-----
IndentationError: unindent does not match any outer indentation level
(<i>python console>, line 8)
```

使用缩进有个要注意的地方，就是不要混用制表符和空格，因为虽然表面上看起来是同一层次，但 python 解释器还是会认为是不同的代码缩进。最好仅仅用空格，并且用 4 个空格作为一个缩进层次。

小结

在编写 Python 程序的时候，一定要注意缩进的使用，因为它是用于区分各个语句块的。缩进很容易被熟悉，在学习 Python 的过程中，应该逐渐养成熟练使用缩进的习惯，因为这不仅是语法需要，更能够让我们形成统一、良好的编程风格。

- 代码缩进：
http://www.woodpecker.org.cn/diveintopy_5_4/getting_to_know_python/indenting_code.html
精巧地址：<http://bit.ly/12Ys4p>