

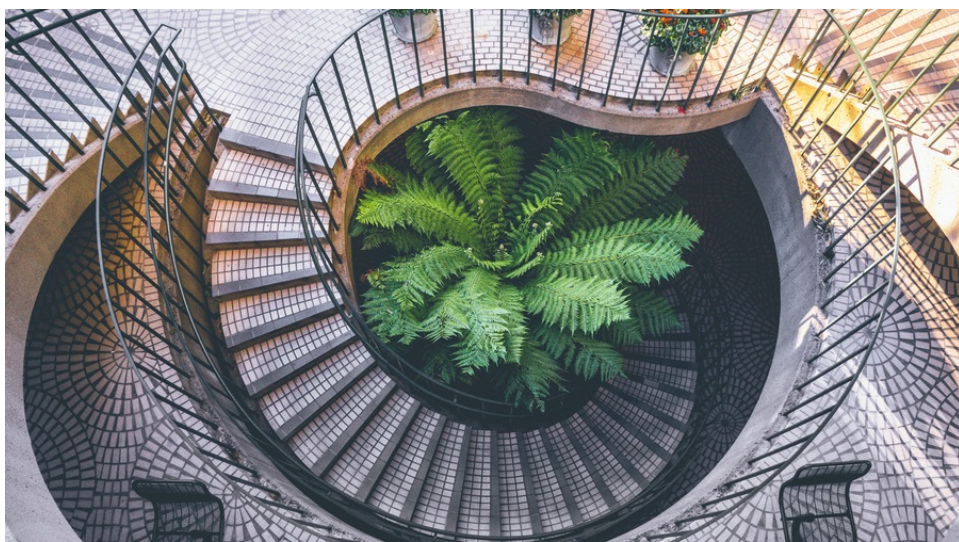


高手分享：如何立即跳出两层嵌套循环？



2 年前

6558



本文作者为 Coverage.py 的开发者 Ned Batchelder，是一位经验非常丰富的 Python 高手，而且也积极组织参加 Python 社区的活动。

本译文为 PythonTG 翻译组最新出品，译者为赵喧典，由编程派作者 EarlGrey 校对。

译者简介：赵喧典，浙江工业大学学生，专业是：计算机科学与技术 + 自动化。爱玩，应用控，技术控，致力于成为高玩/技术宅，终极目标是 hacker/geek。

以下是正文：

如何立即跳出嵌套的两重循环？这是大家经常会碰到的问题。例如，要检验字符串中是否存在相同的字符，如何在找到一对相同的字符之后就停止循环？经典的做法是，写一个两重嵌套循环，对字符串的索引进行迭代：

```
s = "a string to examine"
for i in range(len(s)):
    for j in range(i+1, len(s)):
        if s[i] == s[j]:
```

推荐阅读 热门文章 随机发

- 20天持续压测，云存储性能哪家更强？
- 国内公有云大幅降价后，首份一手云计算产品评测报告
- Python进阶、求职必看的前辈经验分享
- 硅谷码农用Python写了个机器人，租到了让女友满意的房子
- 使用 Python 进行科学计算：NumPy入门
- 十分钟入门Matplotlib
- 从零开发一个小游戏：PyGame 入门
- 好用！在 Notebook 中使用 Sublime Text 快捷键
- 十张GIFs让你弄懂递归等概念

热门标签

- IDE
- PyCon
- 编译
- Flask
- Codewars
- Postgresql
- Django
- Docker
- Git
- 程序员
- 开发库
- 漫画
- 编码风格

```
answer = (i, j)
break # 如何 break 两次呢???
```

此处，我们用两重循环来生成用于检验的两个索引。当条件满足时，我们希望能同时结束这两重循环。

对此，有一些常见的方法。但我个人很不喜欢它们：

- 将循环放在一个函数中进行，利用函数返回来跳出循环。但这样的做法不尽如人意，因为循环可能不适合重构为一个新的函数，也许你需要在循环的过程中访问其他局部变量。
- 抛出异常，并在两重循环之外捕获它。这是把异常当作 goto 语句来用了。但是这里并没有异常的条件，只是变向地利用了异常机制。
- 使用布尔变量来标记循环的结束，并在外循环中检查变量值以执行第二次 break 操作。该方法毫无技术成分，某些情况下可能是有效的，但大多数情况下只造成计算资源浪费，程序效率不高 (noise and bookkeeping)。

我更偏向于使用的方法，也是我在 PyCon 2013 上提到的一种方法，**更自然地循环**，就是将两重循环写成一重循环，然后简单地跳出循环。

这需要花更多的功夫在循环上，但对于抽象化迭代过程是一次很好的练习。这正是 Python 非常擅长的，但人们也很容易忽视 Python 的这一特点，而把它当作一般的语言来使用，不能发挥循环抽象的优势。

让我们重新考虑这个问题。真的需要两重循环吗？写代码之前，不妨再仔细看一遍问题描述：

要检验字符串中是否存在相同的字符，如何在找到一对相同的字符之后就停止循环？

在这描述中，我并没有看出两重循环的意味。事实上，只需要对索引对进行一重循环就可以了。写法如下：

```
def unique_pairs(n):
    """在 range(n) 范围内生成索引对"""
    for i in range(n):
        for j in range(i+1, n):
            yield i, j
```

```
s = "a string to examine"
for i, j in unique_pairs(len(s)):
    if s[i] == s[j]:
        answer = (i, j)
        break
```

此处，我们写了一个生成器用于生成需要的索引对。现在，我们的循环就成了对索引对的一重循环，而不是对索引的两重循环。两重循环依然存在，只是被抽象了出去，移到了 `unique_pairs` 生成器内部。

这使我们的代码更贴近自然语言的描述。还应注意到，我们不用再写两次 `len(s)` 了，这其实是原代码需要重构的另一个标志。而且，如果在其他地方也需要这样迭代的话，还能复用 `unique_pairs` 生成器。但是要记住，复用并不是写一个函数的必要条件。

我知道这个方法有点奇异。**但它真的是最佳的解决方案了。**如果你仍旧停留在两重循环阶段，多想想该如何组织程序。事实上，当你尝试着一次性跳出两重循环，从某种意义上来说，这两重循环就是一回事了，而不是两回事。**将二重性隐藏到一个生成器内部，就可以如你所想的那样组织代码了。**

Python 有着强大的抽象化工具，包括生成器和其他能抽象迭代过程的技术。如果你想了解更多，我在[更自然地循环](#)这场演讲中分享有更多的细节。

[点此查看原文链接。](#)

Python 翻译组是EarlGrey@编程派发起成立的一个专注于 Python 技术内容翻译的小组，目前已有近 30 名 Python 技术爱好者加入。

翻译组出品的内容（包括教程、文档、书籍、视频）将在编程派微信公众号首发，欢迎各位 Python 爱好者推荐相关线索。

推荐线索，可直接在编程派微信公众号推文下留言即可。

本站文章除注明转载外，均为本站原创或编译，如需转载，请联系微信公众号“编程派”获得授权。转载时，应注明来源、作者及原文链接。

[上一篇](#)



相关文章

<p>Bing搜索可以执行Python代码段了？</p>	<p>一个危险的Python函数，不推荐使用</p>	<p>Guido老爹谈Python的未来</p>

文章评论

登录

来说两句吧...

评论

2 人参与, 2 条评论

最新评论



飞龙在天 潜水 [江苏省南京市网友]

2017年5月17日 17:09

回复 2



amberkaka1363在搜狐 冒泡

2016年12月26日 15:18

[广东省深圳市网友]
这个方法很巧妙。python的生成器是个好工具。

回复

编程派正在使用畅言

关于编程派

站点地图
联系我们
粤ICP备16052947

关注编程派



导航

资讯
教程
练习
资源
其他

Powered by

