

神级 Coder 绝不犯的错误：为炫耀编出超短码

2016-06-25 实验楼



点击关注我们

本篇原文来自《Medium》，原文作者为 Allan Reyes 一名退伍军人兼工程师。本文以第一人称编译。

我常逛像是 Exercism 等网站，我在那里编写或重温一些习题来精进我的编码技巧。现在有个危险的流行正在蔓延，我注意到大家很推崇仅用短短几行的代码，认为这样很优雅、有创意，认为这超棒。

但这完全是鬼扯。Brian Kernighan 说得很对：「想帮短码除错比你重写程式还要难两倍。如果你真的觉得自己超会写短码，那么就等到你帮短码除错的时候再来看看是不是真的那么厉害吧。」

当你把编码的行数减少到令人费解时，这还会是个容易维护或可长久使用的代码吗？替编码抓错会变得更简单还是更有难度？更重要的是，如果原来的编码跟短码功能一样，那你是不是有点浪费时间呢？

较短的编码不见得代表是更好、更清楚的编码。当你不小心做的超过了，让编码变的难解，或用了模糊且不必要的模组，你可能会得对你的同事一边装可爱，一边说：「我浪费超多时间在完美化与复杂化这个简单模组，所以你现在才可以花超多时间来了解它。你不觉得这超棒的吗？♥揪咪，编码忍者敬上。」

这不但毫无意义而且很自私，而且完全只是种自负的表现。不必要而多馀的代码的确不好，但短码也不总是就是比较好。缩短编码与简化编码有很大的不同，因为缩短编码仅仅只是让编码变短。

我们用两种简单 Python 计算程式来找两个字串间的「汉明距离，又称信号距离（hamming distance）」。汉明距离在独立字元的计算中为不可或缺的角色。

- abcde 与 abcde 之间的汉明距离为 0
- abcde 与 edcba 之间的汉明距离为 4
- abc 与 abcde 之间的汉明距离为 2

以下是 Exercism 上被高度推崇的编码：

```
1 def hamming(a, b):  
2     return sum(x != y for x, y in map(None, a, b))
```

😄 头秃梭

这是个超棒的编码 ... 如果你的目的是写出混淆代码（code obfuscation）的话。

以下是网站上很不起眼的编码：

```

1 def hamming(s1,s2):
2     """ returns the hamming number (char differences) between two strings """
3     # initialize hamming distance to 0 and find min/max lengths
4     hamming_number = 0
5     minlength = min(len(s1),len(s2))
6     maxlength = max(len(s1),len(s2))
7
8     # iterate through strings up to minlength and compare by each char
9     for i in range(minlength):
10         if (s1[i] != s2[i]):
11             hamming_number += 1
12
13     # add remaining extra characters
14     hamming_number += maxlength - minlength
15
16     # return result
17     return hamming_number

```



它用了超过 15 行以上的编码，以及 517 个汉明距离，但让我来解释为什么这个比第一个范例来的好的原因：

- 伪代码与文档字符串的注解都很清楚。你可以很容易去辨读每个部分与字串的用意，所以任何语言的初级程式员都能理解。有几个「高级程式员（Pythonist）」能第一眼就理解那个比较短的编码？
- 每行都只执行 1 到 2 个方法（method）或操作（operation）。现在你在回头看看第一则裡有多少个 sum method、a != comparison、forloop 与神奇的 map method 在同一行裡。你可以比较一下，两则之间，哪个比较容易读呢？
- 有逻辑的帮变数命名，有「i」的代表单一字元的变数。你可以试著把第一个范例丢到一个更大型的程式码中，然后试著找找看「x、y、a 或 b」。

综合以上，我们可以清楚了解到每个作者的意图：

- 写短码的人是为他 /她自己而写的。
- 第二则范例的作者是为大家而写的。

我想说的是：拜托你，别当个笨蛋。

无论是前端的 HTML/CSS 或后端的 Python 与 Ruby on Rail，都请你写一个大家都可以读得懂的程式码。

（资料与图片来源：Medium）

转载自：程序师

文章地址：<http://www.techug.com/shorter-code>

英文原文：<http://buzzorange.com/techorange/2014/12/01/shorter-code/>



你觉得哪一种写法更好？你平时的写法趋向于哪一种呢？

推荐阅读：

- [关于“技术债务”的形象类比—母鸡的遭遇](#)
- [编程不需要天份，也不需要热情](#)
- [一个老码农的技术理想](#)
- [程序员为什么会是现在这个形象样子？](#)



实验楼
shiyanolou.com

专业的IT在线实训平台。
随时随地，动手实验！

帮您在动手实践中学会IT技术



关注微信

关注@实验楼官方微博，获得更多技术干货！

加入官方QQ群：**450412940**，畅聊IT技术！

