
程序猿训练师养成计划1.0

by Shell Xu

under cc-by-sa4.0



问题：在开发速度越来越快的今天，大量靠谱的程序猿是必不可少的。如何批量的，靠谱的复制出大量的Python程序猿？

这个问题的来由，是因为有很多朋友总是问我说，你有没有认识人在找工作的，Python。或者问我说，我要创业，用Python靠谱不靠谱。这时候很神奇的，居然找不到什么人用。

程序猿都去哪里了？



方法一：靠自然生长。



看来等不及了。自然长出来的程序员就够用，还需要今天我上台来说单口相声么？



方法二：我来培训。



累死我也培训不过来。



方法三：靠网络培训。



Python的几乎所有文档和教材，本来就基本都在网络上公开的。如果网络培训真的靠谱的话，为什么Python程序员还是远远不足呢？



结论：正好我没事干要教课过一段日子，亲自看看问题到底在哪。



亲自教课后的结论

Python的教学有不少问题，我们逐个来说。



问题1: 能自学的，早自学了

真正有自学能力的人，不仅可以自学Python，而且大概可以自学所有需要的技术。这种人，一般早就在各大公司里就业，根本不参与“Python程序员足不足”这种问题的争论。这种人有，而且很多。但实际情况是，比起实际需求根本远远不足。足的话，各大企业这Python荒是怎么回事？

一般来说，会在培训机构里参与培训的，多数都是自学能力不足，期望你把他就业需要的东西喂给他的人。偶尔也有觉得自学太耗费时间，因此期望你能让他学的更顺利一些的人。这种人一般来说在培训班里情况比较好，教什么东西也学的比较快。

培训机构和网络培训的最大不同，其实是沉没成本——钱都花下去了，你能不去上课么？人都到教室了，你还能不专心么？由此来抵抗各种干扰和诱惑。有一小部分是来自于老师能够现场解答问题带来的便利。



问题2: 门槛太低

在现代语言里，Python的入门门槛基本上可以说是几乎最低的了。Python拥有非常好的可读性，使得用户更多的将精力集中在逻辑上而非细节上。

我推荐大学生学习计算机的第一门语言学Python。对于非计算机专业的语言选修，也建议选Python而非C。

然而这一切，都使得Python从业人员的水平更低而不是更高。很多人，会使用Python语言解决简单问题，然后就宣称自己“会”使用Python。解决问题的难度多高呢？大概就是学过的例子改动几个简单参数的范围。

这应该叫做参数师，而不是工程师。

这种“会”Python的人越多，公司招聘Python工程师的成本就越高，也越难招聘到合适的人。



问题3: 好学生根本不来

很多企业，选用Python的最主要理由就是“人便宜”，“学两下就会了”。所以配的工资差不多也是香蕉的级别。结果就是在学校里，但凡靠谱一点的学生，优先会去选择Java和C开发班。因为预期工资高。

俗话说，只付得起香蕉的公司，就只招得到猴子。既然你希望找的人就是“便宜”，那就一定会“不靠谱”。于是开始在线上大出Bug。最后老板破口大骂，这是什么不靠谱的技术。。。



问题4: 学生和学校的浮躁

很多学生，希望出10顿饭的钱，学到能吃一辈子饭的本事。这也就是了。很多时候要学一身安身立命的本事，也不一定要花钱。问题是希望在三五天内神功大成。

你看我长的像有大还丹的样子吗？

学生希望快，学校就压缩课程。两天的课在一天里上掉，跳过所有课堂讨论和练习，这都是小意思。你见过要求入门课堂练习用爬虫/Web开发/数据分析/Saltstack的OpenStack搭建脚本做例子，然后又不对这些技术本身做详细解说，最后还能在大纲里宣称“课程涵盖基础知识/爬虫/Web开发/大数据/自动化运维/云计算”的么。。。

哦，对了，课时还是两天。。。



问题5: 没有标准化的评鉴体系

个人觉得，这个才是问题关键中的关键。虽然我们很多人都考过各种证，也嘲笑过各种证。但是无法否认考证在技术鉴定上的实效。最起码，在上一套系统的时候，RHEL可以说我们这里保证你能招到多少RHCE，足够支撑你的业务。Ubuntu可能说大陆这里还是有不少LPIC的。Python说今天天气不错哈哈哈。

没有证书的问题不仅仅是招聘的问题。作为老程序员，我们看一眼你的github仓库，大概都知道你是个什么样子的程序员了。但是这是我们知道，不是学生知道，也不是企业知道。学生只知道，市面上这么多Python培训，这家三天五个技术点，这家两天十个技术点。一样都没有任何证。那你觉得他们会如何选择？企业只知道，左边这个要价10k，右边这个要价5k。你说他会如何选择？

你说CTO？CTO在忙着写代码呢，没空管这种破事。。。



问题6: 死循环

越没公司用，越没好学生。越是一堆参数师，越容易出问题。名声越糟，越没公司用。搞到最后，用Python的要么是一堆脾气古怪的大咖，要么就是用+拼接字符串的小白。。。



怎么解决



出证书

根本上的方法是出Python的证书。不过很抱歉，这个我搞不定。一个证书的出现需要大量的企业支持，公司买帐才是证书含金量的保证。



那我能做什么？

既然教都教了，那我就分享一些我教课时的副产品吧。我把教课时的材料，全部做成了开放式的教材，基于cc-by-sa4.0开放。

为什么做这个事呢？主要是因为我教课的时候，每次都要准备自己的教案，比较不爽。很多时候都非常耗时。

如果我想用网络上其他人的材料，又有别的问题。网络上有很多入门书，但是不是每个都适合去做培训。很多入门书的形态真的就是书的形态，所以往往是照书讲，讲到什么地方照着翻，而不是slides。而且也没有课堂练习，讨论什么的。

有一些教案倒是slides没错，但是版权根本不确定。搞不好就是哪个学生拿了老师的slide放到网上，用了就不知道有没有坑。



所以我最后只能自己搞一份教案。最初版本是基于"a byte of Python", 随着教课次数多了, 里面发现了一些问题, 修改掉了。现在每次去讲课的时候基本都是稳定了。

把这个版本发出来, 是希望:

1. 如果有人希望做培训——商务培训或者内部培训——可以以比较低的成本来做。cc-by-sa并不阻止你用于商业用途。这样不需要准备材料, 只需要备课就行了。我希望这样可以促进更多的人来教授Python。
2. 如果你在教材里发现了什么问题, 请给我PR。这样我的教材可以不断进化。
3. 教学内容标准化。这样同样学了101的学生, 可以换人教学。对于培训之后能做到什么事情, 企业也会比较有底。

当然, 真正确保这件事情的, 需要有证。这点暂时没办法。



特点

1. 从基本元素到高级特性，从理论到工程。尽量在教授比较简单的内容的时候，不涉及尚未教授的特性。
2. 区分入门进阶，适应不同学生。有些人只需要简单的读写Python代码，有的人就需要正正经经拿来做工程。
3. 以Python为主干，覆盖一系列技术。目前已经讲过，材料正在整理中的有。Django，爬虫，运维，数据分析。
4. 我希望有能力的人帮我一起来完善这个体系。当然，前提是cc系列授权。



对学的人的建议

下多少苦功，得多少成就。做技术的第一原则就是，不要侥幸，不要捷径。你要写代码，你就得会写代码。你要解决底层问题，你就得了解原理。你当然可以通过自己的洞察和判断，避开没有前途的技术，选择更有未来的路线。但是显然不可能有什么大还丹，吃了之后就能脱胎换骨，让你掌握没学过的东西。



地址

<https://github.com/shell909090/learn-python>



微信

