

# 我为什么不学习编程语言

2022-02-16

## 2021.06.27 原文

这里的“编程语言”特指编程语言具体的实现形式，而不是编程语言的设计思想或者语言特性。

- 因为我早已知晓顶级编程语言专家的样子。
- 因为学习某种编程语言的某个特性，对我而言已经是手到擒来的事情。

至少在短时间内，我会保持和坚持目前的观点和做法。

## 2021.08.08 更新

昨天剧本杀，遇到一个 00 后的女学生，拼场来的，但同时也是那家店的兼职 DM。稍有感慨，自己 20 岁的时候在干嘛？

旧博客的链接一直挂在博客最下面，一般也不去看。稍微认真回忆了一下，感慨的事情先不管，我忽然明白，为什么我在毕业的时候会看不起同学的编程水平，为什么我刚参加工作的时候可以至少平等的水平面对同事，为什么最近在面试过程中即使代码能力受到质疑我也不为所动，为什么现在我已经不屑于学习编程语言在形式上的那些东西。

站在国家图书馆的书架前，我忍不住回想起自己大学时候在图书馆里度过的日子。

## 2021.08.29 更新

我又来补充这个问题了。/捂脸

在 QQ 空间上看到一个大学同学发的动态，是一个 Geek 风格的同学。最近的一条是「重学 Unix 高级编程」，上一条是「B 站上某技术会议的视频」。我曾经也尝试去找一些技术会议、meetup，后来发现除了演讲者本人受益，听众都是在陪跑而已，你不会真的指望从中学到什么。会议是一种圈子里面的事情。

对于 Unix 编程或者编程语言这样的东西，我心底是喜欢的，我能想象到，亲手写出一些命令按照自己的意愿达到预期效果，是一件多么原始的能让程序员开心的事情。如果我想学习这些内容，现在的我有足够可靠的信息源、足够有效和明确的方式方法，只要假以时日就可以足够全面系统地掌握这些内容以及其中的细节。

可是我总觉得，我应该关注更加「高级」、更加「上层」的东西，我甚至不知道这里的「高级」和「上层」是什么含义，但是我总觉得，我有更加重要的需要了解和掌握的内容，我不可以专门花费时间在这些东西上。

## 2022.02.16 更新

最近在工作上遇到一个可能搞不定的问题。大体是需要把 Golang 的智能合约用 Solidity 写一遍，在以太坊上运行。其中有一部分内容是基于 Bulletproofs（零知识证明的一种）来实现文件完整性的证明。且不说 Solidity 语言的表达能力好不好实现 Bulletproofs，零知识证明一直是我的黑洞，用什么语言都不会写，而且源文件中存在 magic number，有一个二进制文件和 seed 列表暂时不知道来源是什么。

当然工作问题还是交给工作来解决。我只是由此想到一些问题，感到困惑。例如，什么样的人可以写出 Bulletproofs 的实现？我为什么不可以？差距在哪儿，有多大？Bulletproofs 的学习成本有多大，门槛有多高？

于是几天前我在一个聊天室问了一下，这个是我的发言（<https://t.me/c/1711254099/593>）：

大家好，想借聊天室这个机会，请教一下大家的看法。

最近在了解零知识证明的时候，发现原理比较难理解。想要用代码实现像 bulletproofs 这样的系统 (<https://cathieyun.medium.com/building-on-bulletproofs-2faa58af0ba8>)，仅仅了解编程语言、会编程是不够的，需要一些其他方面（加密学）的前提知识。

由此想到，程序员的工作，或者设计一种软件、协议，很重要的是把编程的技能应用到各种领域中，对编程语言的掌握可能是最基础的技能。

《解密计算机科学》的文章中，最后提到文章已经包含了计算的全部内容。bulletproofs 的本质也是计算，但是要复杂更多吧。

所以我不确定这种看法：计算机科学其实位于比较“基础”的一层，研究编程语言是专业性很强的事情，并不适用于大多数人？即使编程技能很好，也还是需要花费很多时间去了解其他领域的。

这是一个专门讨论编程语言的聊天室，所以发言中强调了编程语言的概念。紧接着有人回复道 (<https://t.me/c/1711254099/595>)：

我觉得不同的编程语言类似于钢铁、木材这种基础材料，我如果想做个凳子，简单了解下材质的用法特性就可以用木材制作一个，如果要做把刀，就需要用钢铁来进行制作，也很简单。如果我要做一个汽车底盘，就需要了解更多的钢铁材质特性和力学知识来进行制作。如果建造摩天大楼就需要学习建筑相关知识，再结合材质特性进行制作。计算机 + 音频知识 = 数字音乐、计算机 + 图形学 + 设计知识 = ps 等...

他形象地比喻了编程语言和领域知识的关系。几天之后，另外一个人同意了我的说法 (<https://t.me/c/1711254099/757>)：

计算的定义，我想引用 yin 的博客：计算就是“机械化的信息处理”

编程语言就是一种表达“计算”的工具/手段。用 sicp 的话来讲是“capture how to knowledge”。

所以会用编程语言就可以表达 domain knowledge 了。而研究编程语言就是为了创造更好的工具（让表达力更强）。

即使编程技能很好，也还是需要花费很多时间去了解其他领域的。

很赞同

总而言之就是，光会写代码是不够的！只了解编程语言是不够的！

## 2022.03.05 更新

编程语言应该学习到什么程度呢？我的观点是，学习到可用的程度。对于不同的人，“可用”有不同的含义，需要自己掂量。

前几天看到一个关于 Personal development 的视频，YouTube 搜应该第一个就是，里面提到几句话，大概意思是，“当你寻找一份工作的时候，不要关心你会得到什么，而是关心你会成为什么（What you become）”。

如果你想成为编程语言专家，就竭尽全力研究编程语言。如果你想成为专精的资深开发者，就专研在自己岗位的技术方向上。如果你没有目标，或者志在别处，就降低对编程技能的要求，“不学习编程语言”。

所以最重要的问题还是，你想成为什么样的人？