



链滴

群分享：不要落入实施细节的陷阱

作者：[ibut](#)

原文链接：<https://ld246.com/article/1541751577230>

来源网站：[链滴](#)

许可协议：[署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p> </p>

<p>分享一下阮一峰的观点： </p>

<p>我引用了一个开发者对年轻程序员的告诫。

在软件开发中，技术变化如此之快，你花费了大量时间学习技术和工具，一旦这些技术被取代，你的识将变得毫无价值，因为它们大部分都是实施的细节。 </p>

<p>我最近总是在想这段话，软件开发算不算是真正的知识？

如果它是一种真正的知识，那么理论上，我们学到的东西大部分应该不会过时，就好像微积分不会过一样。可是实际上，我们都知道，软件开发技能有时效性，十年前学习的编程知识，十年后几乎肯定能用于生产。那样的话，软件开发就不能算真正的知识，只是一种实施的细节。

公司旁边有一家税务所，每天都有很多人排队交税。如果你是第一次来交税，肯定搞不清楚怎么交，税是一门学问，必须有人教你，要带哪些证件，要填哪些表，去哪些窗口排队等等。 </p>

<p>我现在认为，学习编程跟学习交税是一样的，都是学习实施的细节。一旦外部环境变了，原来的施细节就没用了。当代编程由于层层的抽象和封装，我们已经不必接触底层真正具有通用性的知识了大部分时候，所谓编程就是在写某个抽象层的配置。比如，网页样式就是在写 CSS 配置，你很难说到底是真正的知识，还是像《办税指南》那样的实施细节。

实施细节并不是知识，而是操作步骤。如果技术栈发生变更，实施细节就会毫无用处。但是，你又不学习它，不知道实施细节，就没法做出项目。我觉得，程序员应该要警惕，不要落入实施细节的陷阱，不要把全部精力花在实施细节上面，然后以为自己学到了真正的知识。对待各种语言和工具，正确态度应该是“进得去，出得来”，既要了解足够的细节，也要能够站在宏观的角度看待它，探寻底层底是怎么实现的。 </p>

<p>再说一下我的观点： </p>

<p>我感觉有些公众号，打着某某源码分析的口号，分析源码是应该有的，但是钻的太深，也不是一好事。 </p>

<p>原文出处： 金燕知识社群<a </p>