## 作为一个程序员 你的进步完全取决于你自己

我曾经在Hacker News网站上读过一篇帖子的评论，其中一个年轻的程序员说他们不想呆在那些只会让思想陈旧的老程序员离开而不是对他们再次培训的公司里，因为他明白，有一天他们也会这样。



　　也许这个人的本意是好的，但是他这种说法还是让我恼怒。首先仅仅因为一个程序员年龄较大并不能说明他思想陈旧（就像一个年轻的程序员你不能因为他年纪 轻就说他思想新潮）。我认为这个评论中最大的问题是：始终让自己接受教育并跟上时代的发展，这应该是你自己的责任而不是公司的问题。

　　我做程序员已经32年了，而且我一直努力让自己学习各种技术来保持进步。我知道很多像我这个年纪的人自从拿到计算机学士学位以后就再也没有学习过任何 新知识，因此他们最终会被淘汰。我花了6年的时间学习化学，但是一旦当我成为一名程序员，我发现我需要了解在我工作之外发生了些什么，而且我有不断学习的 欲望并一直在尝试各种各样的编程。当然在当年那个时候，所有事情都要慢得多，因为所有的学习都要靠阅读，这是一个很大的挑战（当时没有网络，你必须读纸质 版的东西！）

　　在我做程序员的第三年，我预见到微型计算机是未来发展的方向，因此我调整了我的工作重心，并加入了他们那个团队。有一次，在那里我读到一门新的语言即 C语言，并说服我的经理买下它。事实证明，会用C语言意味着我的第一个启动应用程序就可以用C语言来编写而不再用Pascal.

　　后来我又读过关于面向对象的编程，它点醒了我，于是我在C的基础上增加了一些可以取巧的对象以便利用。因此当后来C++出现的时候我已经游刃有余了。

　　当我在苹果工作了一段时间之后,我认为我需要在网上工作并去了咨询公司使用NeXT WebObjects写了一些被称之为称为 objective - c的模糊的语言。一旦Java J2EE出现，我相信公司会使用它来工作,结果就真的利用Java J2EE做了第一个项目。巧 合的是,当时做那个项目的公司就是现在我老板的母公司，这是他们的第一个完整的Java项目。

　　当你关注的是整个行业发生了什么，而不是只关注你自己正在做的事情的时候，一切就自然而然的出现了。你没办法知道哪件事会成为下一个大事件，但是你可以做到让自己对所有的事情都有了解，即使你没办法将它们一一尝试。

　　我记得2000年中期的时候有一个同事，是搞C++的头儿，当时他的IE浏览器出现了一些问题，我建议他试一下firefox。结果他像看一个疯子一 样的看着我，因为除了IE之外，他不知道还有别的浏览器。在那之后他也没有兴趣去尝试一下firefox，即使这并不会影响到他的工作。我没办法理解这种 态度。你没办法知道下一个大事件会从何而来，拒绝承认这个世界在改变并不会真的让世界停止改变。

　　我有一些获得本科学位的朋友，当大型机退役之后他们随之失业，当他们发现他们没办法再继续当程序员的时候，他们开始抱怨说早该花点时间学些别的东西。 但是当你意识到这些的时候，一切都太晚了。认为你的老板会来培训你无异于痴人说梦。有时候你的老板会连个通知都没有就开了你，你会丢了工作，项目，甚至是 整个公司，然后你不得不去找份新的工作。指望老板对你进行培训是不现实的，如果你是一个汽车生产线上的工人这还有可能，但作为一个程序员是完全不可能。

　　能不能进步取决于你自己，你要自己去尝试新的事物，即使它们并不重要。学习任何东西都是有用的，因为你尝试的编程技术和工具越多，再用其他的工具就会 越容易。你所学的东西可能会没什么用，但是学习本身永远不会做无用功。不管是你在写开源代码，或是自愿的编个程，或者仅仅是在家里写一个别人永远看不到的 东西，都没有关系。它训练了你的大脑开始接受新事物。即使是仅仅了解了在编程的世界里将要发生什么，也可能是有用的，因为不知道在什么时候，就会有老板问 到你关于这方面的问题。

　　当我第一次听说Ajax的时候（在这个术语出现了几个月之后）我正在写web应用程序，然后我开始使用它；再一次的，我先把这个新东西教给我的同事 们。遗憾的是这让公司的架构团队感到恐慌，他们认为我购买了一种未经批准使用的新技术，他们也怀疑这种新技术能否得到支持。他们中间没有人听说过这个东西 （其实是他们没有去关注过），当我告诉他们这只是一种Javascript的时候他们才勉强息怒。我在架构团队里，与时俱进的接受新事物是我的职责所在， 这也是个很痛苦的过程。

　　当iphone最初面世的时候，虽然还没有任何已开发的工具包，我仍然觉得它代表着未来。我的Objective-C方面的知识再次派上了用场，这实在很让人吃惊。

# 

现如今要与时俱进是很荒谬的一件事。用JavaScript做 web apps的人们每周都能搞出新的大事件；即使学的再快，你也没办法在新的框架 覆盖之前就学习完所有的东西。那些日子我的工作并不在这个领域（当时我只做ios），但是我仍然坚持学习，你永远不知道未来会朝什么方向发展，也没法知道 你会在哪里结束。

　　在老板给我的少数的几堂培训课里，我基本上没学到什么有意义的东西，因为我那些东西我都已经了解，而且有些新技术我都已经使用过了，即使它们对我的未来没那么重要。不过所有的事情都是这样，你没办法知道哪些东西会在以后帮到你。

　　在研究生期间学习化学的时候我学到了APL，除了玩，我从没用它干过别的事情，但是最后它成了我第一个产品Trapeze的基础。在1999年之后我 把Objective-C忘得干净，但是现在它成了我的主要语言。你没办法知道在未来能帮到你的东西跟现在有什么关联。

　　有的程序员会忽视掉他们工作方面以外的一切，这种行为深深刺激了我。我曾与一些利用4GL工具生成RPG代码的人一起工作。他们吹嘘说他们使用的工具 是世界上最好的，他们无需再了解别的什么东西。我甚至能够想象当他们中的大多数人下岗的时候，他们会开始疑惑，为什么他们没办法再找到一份新的工作。技术 很少会消失，但是你的工作会减少，除了可怕的维护工作外你可能什么都做不了，然后你就变得多余，直至被扫地出门。现如今任何人都没办法帮助你重新成为有用 的人，除了你自己。如果你还在等待，那么等到一切都已经太迟的时候，你的事业就会遭受到毁灭性的打击。

　　所以现在不要再拖延了，学点新东西吧，让自己更好的学习新东西吧。睁大你的双眼，你可能就会看到未来正在向你走来。当它展示在你面前的时候你会发现这 没什么大不了的。很幸运一直以来我能够保持专注，这让我在三十年之后仍然能够进行编程，能够在一些很现代化的东西上进行工作。

　　所以要睁大你自己的双眼，不要指望别人会帮助你。就像棒球投手Satchel Page所说的那样：“别回头，别人随时有可能赶上你”。在编程方面你需要一直向前看，因为你背后唯一的东西是讨厌的颠覆者。