

通过实践学习

今天我们来谈「Learn By Doing」。这件事情太重要了。学习很重要。而学习如何学习的重要性更是不言而喻。我发现要去创造东西，做中学学中做是最好的。比如我要学会 Android 开发，就真要去实践 Android 开发，做一个又一个有挑战性的项目。

Paul Graham 说计算机系的大学生生活怎么过。解决大量的困难问题。一来，从一开始到最后解决困难问题的同时，可以学到很多东西。比如做一个很有挑战性的项目，写一个微信客户端。如果事事亲为的话，也应该到最后，能用 PS 处理图片，懂得 Android 的方方面面。如果不用第三方的推送服务的话。自己写一个实时通信的服务器，就能学到更多了。

我们看完一本书，会觉得大体都懂了。但远远不够的。看完《疯狂 Android 讲义》的时候，我觉得 Android 编程我懂得不少了，上面的大体都会了。大约我用了 10 天时间看完了上面几百页的书。之后我用了 4 个月的时间熟练上面的知识。做了几个小项目，一个大点的项目才稍微熟悉了 Android 开发。

也就是看书掌握的知识远远不够。看书是浅薄的。当然既然我只花了 10 天时间看了书，也不是很花时间嘛。

我想到了，也许高中教育、初中教育、小学教育大多数时间都是被浪费的。这些都是基于书本的教育。

上次和大伯讨论石灰浆是怎么固定在墙上的。话说我现在也不清楚。CaO 和 O₂ 氧化成了 CaCO₃ 么，然后 CaCO₃ 是固体，比较坚硬。在生活中，我区分不清楚 Cao 和 CaCO₃，虽然我能配平他们，经过练习、做题，我能熟练知道他们能转换成什么，有什么属性。但很快，我又会忘了。已经快离开高考一年半了，能主动回想起的知识已经不多了。我高考考了 590，现在立刻再考一次，300？

还好，我高中三年，不少时间来学算法、来学编程了。没有所有时间都在玩一个无聊的书本、考试这些纸面上的游戏。但可惜的是，我当时没有学编程学得更多。

虽然，学习物化生语数英的过程，也磨练了我，为一次又一次的完美，锻炼了我的耐心、坚韧。但它的坏处也是显而易见的，它让你觉得学习是痛苦的，知识是无聊的，一切都是为了名次、高分。它进而影响你的天生的兴趣，影响你天生的好奇心，让你在编程的路上，也注意名次、也注意自己厉害还是 xx 厉害，没有享受知识本身的乐趣。

TJ (vision media) 说，我从来不去学校，也从来不看书。我学习编程就是去看别人的代码，然后很好奇，总是想弄清楚为什么。

Paul Graham 说，有野心的成年人，就是将好奇心集中在一小块领域。

知识越复杂，就越是不能够靠名利、靠自制力来驱动去习得，而只能够依赖自发的好奇心。

因为我从小到大习得知识更多是为了名利、为了名次、为了比别人厉害。我很需要外界的认同。当然自身也算是比较喜欢编程。

核心驱动我的是，跟一些朋友比，我还是太菜了。他们很厉害，跟我一样年龄，大二、20岁，有一个业余之外写写编译器，平时在外包公司里做得很不错，有一个朋友有两个公司、全职一份工作包了后端开发，还有认识的清华同学各种世界奖。

显而易见，自己能感觉到不少差距。不是说他们比我早学或者什么，而是他们效率又高还每天比我花的时间更多。

基于同龄人的压力，想比同龄人更厉害带来的动力是巨大的，不得不说。同事也有很厉害的。但潜意识觉得他们比我年长，也许我到他们那个年龄就跟他们差不多。这样一想，就没什么了。

所以，我们尽可能去认识和自己年龄相仿的，比自己更厉害的人。虽然那样对自己很残酷，不断要承认自己很菜，但是结果是好的。只要你坚强地面对一切，不自暴自弃，看到差距，默默努力，结果便是非常好的。

或许还要更多的好处。因为他们厉害，更优秀，更熟练擅长某样技能，那么他们更可能是真正喜欢那件事情的人。更可能是热爱编程的人。因此带给自己正面的影响，也去感受编程的乐趣。

编程这件事的好处是，一旦你投入，你会不由自主地喜欢它，从而忘记名次、忘记自己厉不厉害、忘记功名利禄。

我承认，我投入时间编程，一来是因为是我的工作我要向领导交代些什么，二来我觉得我很菜再不好好努力就晚了，三来我有个成为编程大牛然后全栈创业的梦想。但是因为我投入了，我能很快感受到动起手来创造东西的乐趣。解决了一个又一个问题的成就感。

动起手来，能更好地学到知识，也更有趣。