



下载APP



加餐三 | 10000小时定律：成为大牛的秘密是什么？

2021-01-06 李运华

大厂晋升指南

[进入课程 >](#)



讲述：安晓辉

时长 15:35 大小 14.29M



你好，我是华仔。

在第 16 讲中我向你介绍了我自己总结的一套系统的学习方法，而这套方法的指导原则就是 **10000 小时定律**。

那么这个理论是谁提出来的呢？它有用吗？要怎么用？用它的时候要注意些什么？今天这一讲，我将会带你探寻 10000 小时定律的来龙去脉，尝试破解成为大牛的秘密。

10000 小时定律的发展史



在 10000 小时定律的发展过程中，一共有 3 个不得不提的关键人物，他们分别是本杰明·布鲁姆（Benjamin Bloom）教授，安德斯·艾利克森（Anders Ericsson）教授以及作家马

尔科姆·格拉德威尔 (Malcolm Gladwell) 。

布鲁姆：长期大量的练习

布鲁姆是美国知名的教育心理学家，芝加哥大学的教授，在“教育目标分类”和“精通学习”这两个领域作出了很多贡献。

1985 年，他出版了一本书《如何培养天才》(Developing Talent in Young People)，专门介绍怎么从青少年群体中发现天才。

他研究了来自多个职业领域的 120 个成功人士的童年，包括音乐会上演奏的钢琴家、精湛的雕刻师、奥运会游泳运动员、世界级的网球运动员、杰出的数学家、杰出的神经学家等，涵盖了科学、艺术、体育、医学和工程等多个领域，想确认到底有没有“预测孩子未来成就的指标”（比如最广为人知的“智商”）。

但是研究结论却推翻了这个想法，不存在这样普遍适用的指标，智商和孩子将来的成就**没有直接关系**。

既然如此，那些成功人士又是靠什么获得成就的呢？难道是完全随机的吗？也不是。研究发现，对于大多数成功人士来说，最重要的因素是**家人的鼎力支持、长期大量的练习和专业老师的指导**。

但是对于“长期大量的练习”这个因素，布鲁姆没有明确研究出“长期”到底有多长，也没有提出“10000 小时定律”。

艾利克森：10000 小时练习时间

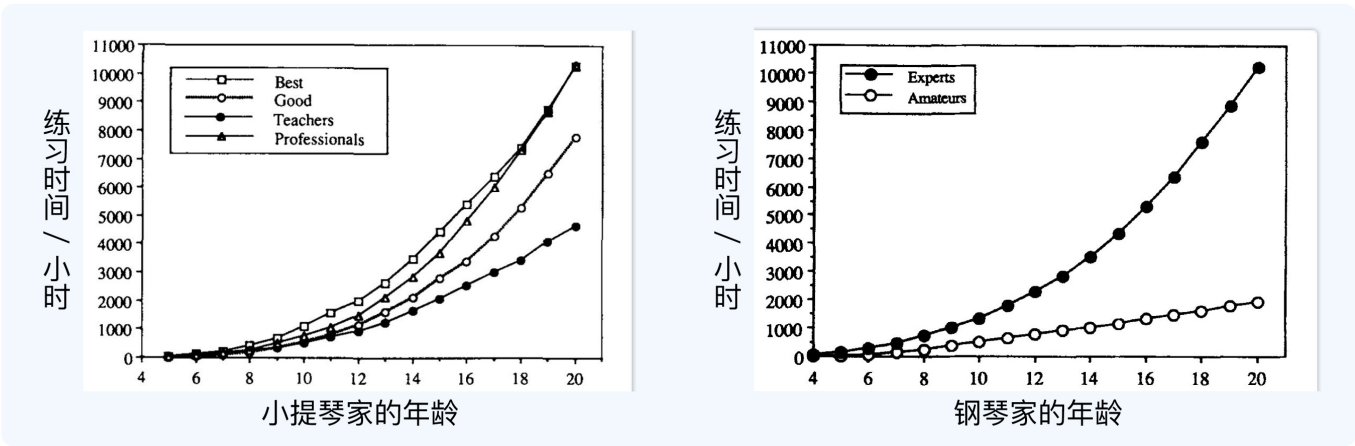
布鲁姆虽然没有研究出量化“长期”的方法，但是他的研究打开了一扇通往新领域的大门。于是很多学者开始跟进，其中美国佛罗里达州立大学的心理学教授艾利克森就发现了“**10000 小时练习时间**”这个现象。

艾利克森对柏林音乐学院的学生进行了研究。他让音乐教授根据潜力将小提琴学生分成三组，依次是顶尖小提琴家 (Best)、优秀小提琴家 (Good) 和音乐教师 (Teachers)，然后详细分析这三组学生之间水平差异的原因。结果他发现，只有**练习时间**这个因素是区分不同组的关键指标，顶尖小提琴家的练习时间比音乐教师多 3 倍。

后来，他又研究了中年专业小提琴演奏家（Professionals）年轻时的练习时间，同样发现了练习时间这个关键因素。

为了进一步证实结论，艾利克森又对钢琴演奏家进行了研究。这次他挑选了专业演奏家（Experts）和业余爱好者（Amateurs）进行对比，结果发现专业演奏家的练习时间是业余爱好者的 10 倍。

在这两组研究中，他都发现了 10000 小时这个数据，如下图所示。



引自艾利克森的论文

1993 年，艾利克森把研究成果整理成了论文 “[The Role of Deliberate Practice in the Acquisition of Expert Performance](#)” 发表。这篇论文不但描述了详细的研究细节，还介绍了各种跟成功有关的研究和它们的分析框架。

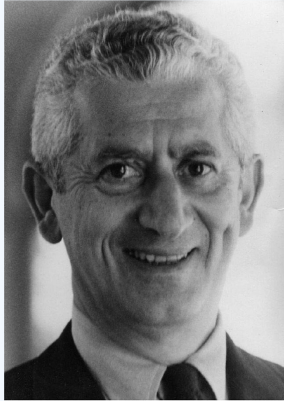
但是，他也没有提出 “10000 小时定律” 。

格拉德威尔：10000 小时定律

后来，加拿大作家格拉德威尔根据艾利克森的论文结论，提炼出了 “10000 小时定律” ，也就是说，要想成功就必须要有 10000 小时的投入。

他分析了很多成功案例来证明 “10000 小时定律” 的普适性，比如甲壳虫乐队走红前在德国汉堡的酒吧中演出超过 10000 个小时，Sun 公司创始人比尔·乔伊的编程时间超过 10000 小时，微软公司创始人比尔·盖茨的编程时间也超过 10000 小时，音乐神童莫扎特真正成才前的作曲时间超过 10000 小时等等。

2008 年，格拉德威尔把他的观点写进了新书《异类：不一样的成功启示录》（Outliers: The Story of Success，以下简称《异类》）。这本书上架以后，雄踞《纽约时报》排行榜榜首 20 个星期，半年时间北美销售量超过了 100 万册，从此“10000 小时定律”广为人知。



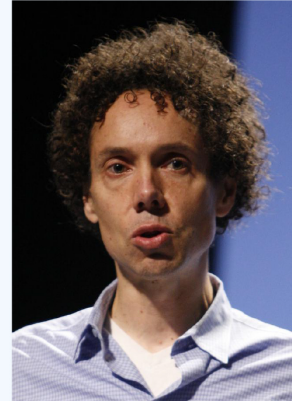
布鲁姆

奠基者，发现长期大量的练习是成功最重要的因素之一。



艾利克森

推动者，发现10000小时练习时间是成功的关键指标。



格拉德威尔

总结、传播者，提炼出了10000小时定律，要想成功就必须要有10000小时的投入。

10000 小时定律剖析

因为在《异类》这本书中，格拉德威尔只用了一章的篇幅来阐述 10000 小时定律，并不能涵盖艾利克森论文的完整内容，所以也引起了一些争议。

批评者的主要观点是，10000 小时定律**过于简化了**“如何才能成功”这个问题，会给人造成误导。

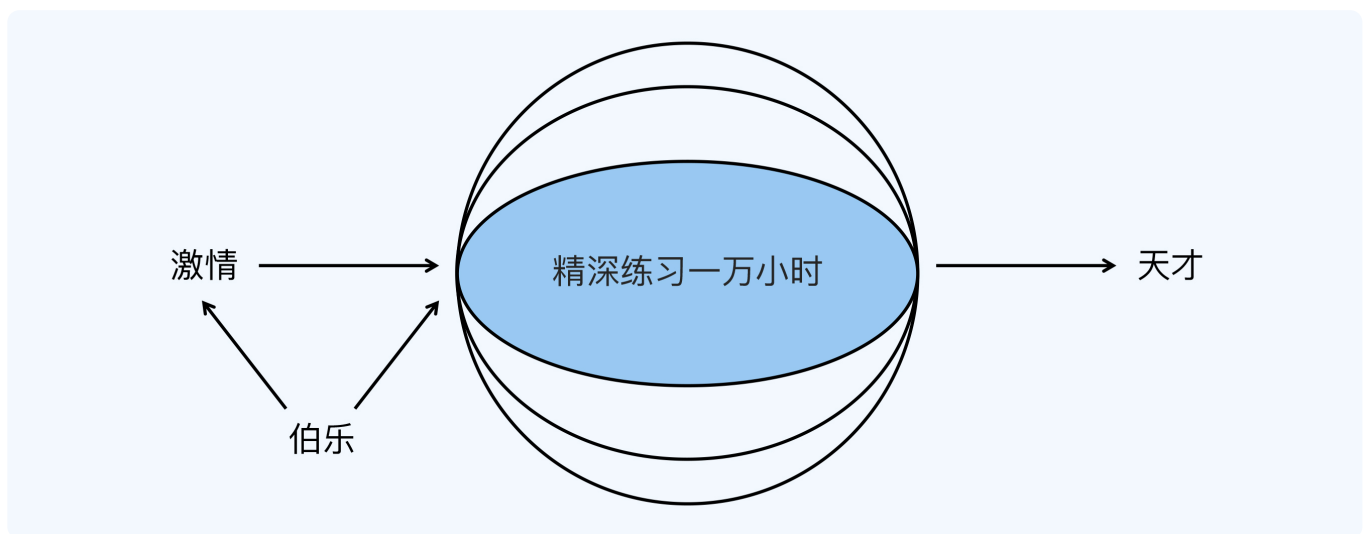
事实上，无论是格拉德威尔还是艾利克森，都没有说过“只要练习 10000 小时就一定可以成功”。格拉德威尔《异类》中也探讨了很多和成功相关的因素，包括环境、文化和时机等；艾利克森在论文中也分析了家庭和个人等因素对成功的影响。

所以，我们不必纠结 10000 小时定律到底是否全面。合理的做法是，把 10000 小时定律理解为成功**必要条件**，而不是**充分条件**。换句话说，没有 10000 小时的投入，很难成为专家，但有 10000 小时投入，也不一定能成为专家！

要想通过 10000 小时的练习成为专家，还有几个关键的因素也不能忽略。美国作家丹尼尔·科伊尔的《一万小时天才理论》这本书就做了很好的总结：**精深练习、激情、伯乐三个因素是 10000 小时定律的关键。**

1. **精深练习**：你需要设定努力的目标，然后挑战自己的能力极限，不断地重复练习更高要求的技能才能提升自己。写 10000 小时 “Hello world” 不会让你成为编程高手，唱 10000 小时《两只老虎》也不会让你成为周杰伦。
2. **激情**：10 年 10000 小时的持续投入并不是小菜一碟，而是一项非常大的挑战，靠外力的强制或者自我意志力来强迫达成是不可能的，必须要有个人的激情作为持续投入的动力。所以你自己要喜欢这件事情，能够从中感受到快乐和满足。
3. **伯乐**：每个领域都有大量的经验教训积累，单纯靠学员去试错来找到所有这些经验和教训是不可能的，需要有伯乐对学员进行观察，然后指出需要改正的地方和练习的方法，这样学员才能够快速提升。

《一万小时天才理论》整理了一个完整的 **理论图示**，非常具有参考意义。



引自《一万小时天才理论》（重新绘制）

从这张图里我们很容易看出来，单纯练习 10000 小时是不够的，还需要个人的激情作为动力，以及优秀的伯乐进行指导。这也就回应了人们对于“10000 小时定律”的批评。

互联网技术领域如何落地？

不过，刚才我们提到的研究主要集中在音乐（小提琴、钢琴）和体育（足球、网球）等传统领域。在互联网时代，尤其是现在的移动互联网时代，10000 小时定律的应用可能还会面临一些新的问题。

1. 没有伯乐怎么办？

传统领域的发展都有 100 年以上的历史，训练体系非常成熟，有很多优秀的专业教练能够对你进行指导。相比之下，互联网行业发展的历史就很短了，也不存在成熟的训练体系，更没有专业的教练能够像小提琴、网球那样进行一对一的指导和训练（要是能够请教练了，谁还 996 上班啊）。

所以为了保证 10000 小时投入的效果，我们需要一些变通的方法。

第一种方法是在团队内部找导师，不一定是主管，同事中的高手也可以。在代码 Review、设计评审和方案讨论的时候，拉上导师一起参与，给你提建议。

第二种方法是看书和学习线上课程。书籍和课程都是作者对知识和技能的一次梳理与整合，对经验和教训的一次总结和传承，所以购买一本书或者一门课就相当于请了一个教练，虽然它不能提供实时和具体的指导，但是我们可以通过它来详细地了解一个领域。

第三种方法是参加行业会议。行业会议会邀请行业内的专家来进行分享，每个分享主题也都是很有价值的经验总结，对你的提升具有指导意义。

第四种方法是参加线下的训练营。现在有一些机构开始尝试线下的训练营模式，邀请行业内的优秀人才作为导师，针对某个主题，集中进行一段时间的强化训练来提升学员的能力。训练营的模式和运动员教练很类似，能够实时地对学员进行指导，效果是最好的，但时间成本和资金成本也是最高的。

2. 技术变化太快怎么办？

跟传统领域相比，互联网行业的技术更新换代要快得多，比如最近 10 年影响比较大的新技术就有大数据、App 开发、微服务、容器化和人工智能等；而且各个细分领域的技术变化也很多，典型的就前端开发，包括 jQuery、HTML5、Node 和 Vue/Reactor/Angular 等。

技术的快速变化确实会导致之前的一部分技术积累在新的环境下失去了原有的作用，比如现在我们很少用 Flash 来做开发了，但这并不意味着我们之前在领域的积累完全归零。

首先，**很多基础的技术是不会频繁变化的**，比如操作系统、数据库、浏览器、网络等。比如虽然 iOS 和 Android 开发是最近十几年才兴起的，但它们的基础仍然是操作系统、计算机网络和编程语言这些“老”技术。

其次，**新技术往往是在老的技术基础上进化出来的**，它们的目的是更好地解决老技术的问题。比如 jQuery 是为了解决 JavaScript DOM 编程太复杂的问题而设计出来的，Vue/Reactor/Angular 等前端框架又是为了解决大型项目中使用 jQuery 所导致的难以维护和协同的问题而设计出来的。

所以技术的变化不但不会让我们之前的积累失去价值，反而还会让我们之前的积累更有价值。绝大部分新技术的出现，都是业界顶尖的公司或者专家，结合他们以往的经验，再发挥他们天才的灵感才创造出来的。如果没有足够的经验积累，也就无法推陈出新。

20 小时学习法

10000 小时定律关注的是**怎么成为顶尖的领域专家**，比如小提琴家和钢琴家等。但是无论在日常生活还是工作中，我们都不可能在每个领域都成为专家，更多的时候只是想**熟练掌握一门技能**而已。

比如我们大部分人学开车，只是为了上下班通勤、节假日旅游或者当司机赚钱，而不是成为 F1 赛车手；大部分技术人员学习 Redis，也只是为了学习原理，方便在项目中使用，而不是想成为 Redis 的开发者。

这种情况下，如果还只靠 10000 小时定律来规划学习安排，显然是不够的。

美国学者乔希·考夫曼（Josh Kaufman）在《关键 20 小时，快速学会任何技能！》（The First 20 Hours: How to Learn Anything... Fast!）这本书中指出：如果学习目的不是“学精”（成为专家甚至大师），而只是“学会”（知道怎么用），那么只要花 20 小时，就可以快速掌握一项新技能。

考夫曼并没有否定 10000 小时定律，而是指出针对不同的目标应该采取不同的方式，不要一概而论，如果全都套用 10000 小时定律，时间和精力肯定都不够用。

所以他总结出了一套提升学习效率的“20 小时学习法”，分为四部分：

1. 分解步骤：把技能最大程度地细分，分成若干小步骤。
2. 充分学习：基于分解步骤得到的小步骤，逐一练习。
3. 克服困难：克服练习过程中的各种困难，包括生理、心理、情绪、工具、环境等。
4. 集中练习：至少用 20 小时集中学习最重要的小步骤。

虽然我暂时还没有看到针对 20 小时学习法的严谨的科学研究和证明，但它看起来确实很符合人的直观感觉。

比如我们学车的过程，就非常符合 20 小时学习法：考试分为四个科目，每个科目有固定的考试项目，我们在教练的指导下针对考试项目逐一练习，最后通过考试拿到驾照，真正练习的时间也就差不多 20 小时。

在互联网技术行业，如果你想初步入门某项技术，可以按照 20 小时定律来进行实践，不要看到某个技术就觉得要花费太多时间，还没开始就把自己吓到了，结果一直都不去学习。20 小时定律同时也提醒我们，不要一上来深入研究源码这些，可以先从掌握基本的使用开始来学习技术，这样能够快速掌握基本的使用，然后有时间和有需要后再逐步深入。

小结

这一讲我跟你分享了 10000 小时定律和 20 小时学习法这两个跟时间相关的学习理论。其中 10000 小时定律是专业领域提升的总的指导原则，而 20 小时定律适合指导你快速入门学习单项技术。

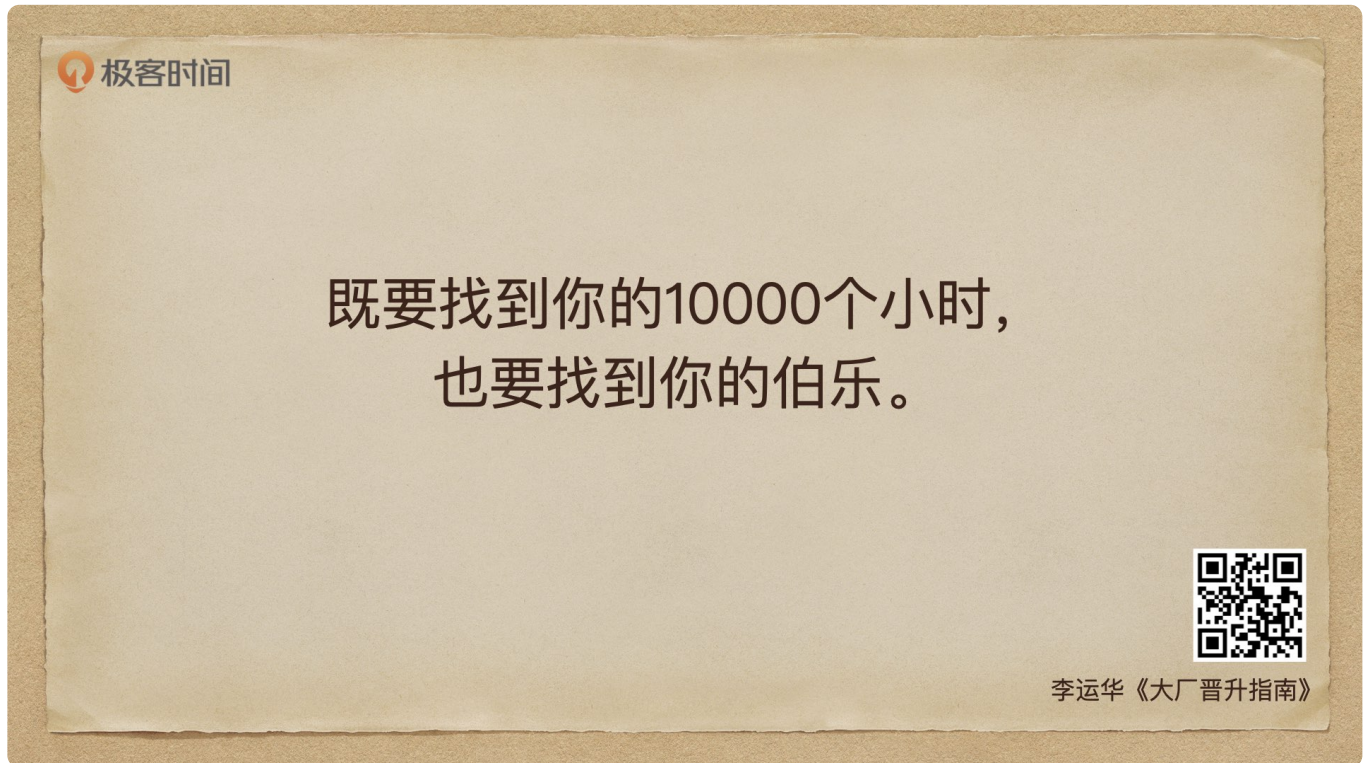
现在，我们回顾一下这一讲的重点：

1. 布鲁姆发现了“长期大量的练习”是成功最重要的因素之一，阿利克森发现了“10000 小时练习时间”是成功的关键指标，格拉德威尔提炼出了“10000 小时定律”并加以传播。
2. 单纯练习 10000 小时是不够的，还需要个人的激情作为动力，以及优秀的伯乐进行指导。
3. 10000 小时定律适用于在某个领域成为专家，而如果想熟练掌握一项技能，采用 20 小时学习法会更合适。

思考题

这就是今天的全部内容，留一道课后思考题给你吧。评估一下你在目前的专业领域大概投入了多少时间？如果你觉得自己投入了足够的时间，但是能力却没多大的提升，你觉得可能的原因在哪里？

欢迎你把答案写到留言区，和我一起讨论。相信经过深度思考的回答，也会让你对知识的理解更加深刻。



提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 加餐二 | 提名词：怎么夸自己才最加分？

下一篇 加餐四 | 学习基础技术：你对“基础”的理解准确吗？

精选留言 (5)

写留言



sugar

2021-01-06

回答思考题：作为职业码农干了5年，一直在BAT，目前级别是p7。如果把本科计算机系4年时间也算上，投入在纯技术方面学习和实践（不能按工作日8小时算，因为有一半时间在开会、paperwork、和pm或其他岗位扯皮，自我感觉这些对技术能力的提升是无效的）总共算下来大概7-8千小时。目前级别是BAT大厂p7，不敢说专家，不过确实自己在技术领域里几乎没有盲区了，也有自己长期深耕的一些技术领域。目前比较大的感受是，对...
展开

作者回复: 5年到P7算正常水平，不过你的时间折算应该没有8000小时那么多，如果你一直保持学习和提升，平均1年算1000小时差不多。

P7这个级别并不是不能通过“看书、线上课、conf、训练营”来提升自己，而是要看更有深度的书，跟偏方法论的线上课，同时光自我学习还不够，还要有机会应用这些，所以你会觉得光自己学好像是不够的。

当然，进一步成功在职业发展上就是“晋升”了，确实不能只靠自我提升就够了，还要有机会，这就是前面P8级别讲的内容。

要获取机会就要有影响力，影响力来源于你的专业度、拿到的结果、综合能力；所以除了继续提升自己的技术能力外，确实要多提升自己管理和业务的能力，多跟产品运营等上下游打交道。

1

9



test

2021-01-07

想问下各位，对写代码的激情，到了什么程度才算是热爱？

作者回复: 你可以看看我2014年的知乎回答：<https://www.zhihu.com/question/26834363/answer/34302717>

简单来说：就是遇到问题会一直想，想得睡不着：)

3

3



Jxin

2021-01-06

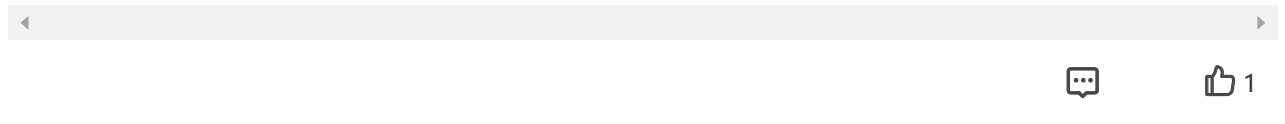
1.软件开发，差不多6-7千小时吧。感觉自己还是只有入门的水准。技术上大部分的技术原

理都懂，但自己也没全去写过，仅仅看来的总归是浅的。架构上，目前也只是熟练面相对象相关的思想和方法论，函数式编程会用但用不精。为什么成为不了专家？野路子讨论少，但靠自学缺少环境还是会差一些。另外也不好定义啥是专家，毕竟软件开发能细分的领域比较多，任何一个细分领域都不容易。...

展开 ∨

作者回复: 你应该是广泛的去看过各种技术，但是没有比较深入的研究透几个典型，也没有对某个技术领域形成方法论。

并不一定要自己亲自做过才能理解很深，具体怎么学后面会有讲到。



右耳朵猫咪

2021-01-06

如何理解文中 “Vue/Reactor/Angular 等前端框架又是为了解决大型项目中使用 jQuery 所导致的难以维护和协同的问题而设计出来的”？难以维护和协同具体指的是什么？

作者回复: 详细你可以查查jQuery与vue等的对比，或者你要是写了jQuery和vue（小程序也可以）就有感触了。

jQuery本质上是一个方便的DOM选择器，在写页面的时候数据和视图是混合在一起的，如果是大型项目，视图复用不方便，单个页面逻辑太复杂的时候jQuery也容易出错。

vue是一个MVVM的框架，将视图和数据分离，然后用双向绑定来将视图和数据关联起来，在大型项目开发的时候，视图组件复用方便，复杂页面的实现时也要清晰很多。



蛙

2021-01-08

华仔，工作5+年，职级偏低怎么破？上一次跳槽准备的不好

作者回复: 对照职级提升，有信心了可以申请晋升可以跳槽

