

答疑加餐 | 学了这么多前端的“小众”知识，到底对我有什么帮助？

winter 2019-04-23



你好，我是 winter，这一期我想跟你谈谈前端知识的学习姿势。

课程进展至今，我已经把大部分困难的知识点都讲完了。我在后台收到了一些留言，有的同学针对前端专栏的学习方法和学习方向提出了一些问题，在本文中，为了让同学们更好地理解我们专栏的重点，最大程度地吸收知识，在今天的文章中，我会逐一回答同学们在学习方法上的困惑。

1. 老师讲的内容是不是工作中用不到的，对掌握前端的实际工作有什么帮助呢，我们有必要掌握这些比较偏的内容吗？

我自己一直奉行着一个观点，不要执着于知识的“临时”实用性。因为我总是感觉，掌握知识越多的人，更喜欢花费时间学习一些暂时可能无法变现的知识，原因有两点：一是其实学知识花费的时间比想象中的要少，这边在纠结它有没有用，那边可能都学完了；二是知识的实用性其实不怎么好判定，比如当年黎曼搞出非欧几何的时候，全世界都觉得纯粹是数学的思维游戏，直到黎曼死了很多年后，相对论出世，黎曼几何有了实际用途。

不过，这里我还是希望讲清楚，我想通过我们的课程传达一些什么样知识内容。

我们的每一节课标题中，都会带一个有点“小众”的问题，但是，实际上，这个标题主要是引起你学习兴趣的一个引子，解决问题只是我们学习课程的一个自然结果。

我希望的是，通过这个有点偏的问题，引起你对这部分知识领域的关注，知道这部分知识的边界在哪里，从而形成一个完备的知识网络。让你在遇见不会的问题时候，可以快速定位到知识点，达成有效学习。并且，你也可以通过自己之前没有关注过的不同视角，来重新学习一遍这部分的知识。

比如在 JavaScript 课程中，我提供了几个不同的视角来讲解 JavaScript 语言，每一个视角下，都是完整的 JavaScript 知识。

比如说，当我们写下 $1+1$ 的时候，我们从词法的角度看，这是两个数字直接量和一个加法符号，我们从类型的角度看，两个 Number 类型相加得到的也是 Number 类型，我们从语法的角度看，这是一个加法表达式。

我们从不同的维度去认识了 JavaScript 语言，这些视角，远比你记住我们课上讲的细节更重要。如果你记住了“数字直接量可以用科学计数法，E 还可以小写”，却没有学会从词法的角度去分析 JavaScript 语言，那可真是“入宝山空手而归了”。

2. 我从业时间不长，文章看得迷迷糊糊，是我的基础不够吗？该怎么办？

有一种说法，世界上的知识分成“我知道的”“我知道自己不知道的”和“我不知道自己不知道的”。

重学前端定位是有一定经验的前端工程师，所以它最重要的作用之一，就是帮助前端工程师发现问题，找出知识盲点。

课程设计上，我的主要思路也是“帮助”大家形成自己的体系，而不是“替”大家形成自己的体系。我在前言部分就讲到过，若论丰富全面，有 MDN 文档；若论准确权威，有标准文本，但是，我们课程的作用是传达思路，如果有知识上的缺失，你其实可以通过阅读 MDN 来补全。

3. 如果阅读文章时候有的内容看不懂，该如何学习，如何定位这块的知识呢？

这个问题比较抽象，我认为这个课程设计其实也是一种定位了。

比如，对 JavaScript 问题，先搞清楚看不懂的是词法问题、语法问题、还是运行时问题？定位清楚了问题，你已经距离解决问题前进了一大步。

在这里，我想跟大家说一下：如果你看不懂文章里的某一块知识，你可以给我留言，把具体的位置和知识点告诉我，这样我们可以进行更好的沟通和反馈，从而解决这个问题。如果你只是说不懂，我可能会一头雾水，也无从下手去帮助你解决问题。多做实时、有效的反馈，会让知识吸收的效果更好。

4. 为什么文章里有那么多的术语和英文呢，为什么不换成更通俗的名字呢？

其实，在我们的课程中，有很多次讨论到术语问题，比如“排版”，我们讲了中国古代的活字印刷，比如“渲染”，我们讲到了国画的技法。恰当地使用术语，对于传达知识是非常关键的。

翻译是一项很专业的工作，文学类的翻译讲究信、达、雅，对于技术类的名词，或许“雅”这方面可以稍微打点折扣，但是表意清晰，字句通达仍然是必须的。

很多术语有约定俗成的翻译，当我们阅读不同的文档时，可以快速地通过术语建立联系。并且，有的时候翻译本身也会造成一部分信息的丢失，所以，我有时会直接把英文也写出来，这样有助于你通过原文去理解和对照。

而为什么我不把术语换成更通俗的名字呢？

我们所讲的多数技术，跟现实生活联系不大，这种情况下，“通俗”的名字往往意味着误导。有时候，我们确定术语时，反而会尽量使它远离已有的概念。当然，确定术语并非是我的工作与专长，我们课程中的绝大多数术语，都不是我的发明创造。

5. 标准里有些东西还是看不太懂，如果可以的话，希望老师可以稍微讲解一下如何看懂标准？

我并不推荐每一个前端工程师都去阅读标准，标准一半是写给实现者，一半是写给使用者，这里本来就有很多知识上的落差，多数时候，MDN 是更好的选择。

如果一定要阅读标准，建议从自己做一个极简实现开始，我在浏览器部分，有讲解浏览器相关的知识，在 JavaScript 部分，我还设计了编译原理实验。我想，把它们落到代码上会是一个很好的开始。

6. 接手了一个新项目，怎么对前端合理规划，老师能不能提供一些这方面的指导和建议？

这个问题其实跟前端学习关系不大，但是我可以讲讲。

任何规划其实都差不多，得有背景、目标、方案、计划、预期结果。其实在我看来，“项目”是规划的最小单位，在项目中拆出前端来做规划，是不太合适的。

背景和目标通常来自公司的业务，方案跟具体的技术相关，计划是项目管理的领域，最后根据这些来给出预期结果。

再往下细说，一个完整的方案可能包括产品、运营、市场、技术，不同的项目，各个职能的难度不一样，有些项目可能干脆不需要某些方案——比如多数淘宝的产品，首页开个入口就有访问量了，不需要独立去做市场。

具体到技术方案，前端、后端和公司的基础设施都有一定影响，有些公司会找一个架构师来做整体方案，有些公司则是得几个商量一下边做边出，其实因地制宜最重要，能达到目标的方案都是好方案。

有了整体方案，到前端的一亩三分地上，技术选型、工程规范是绕不开的，有些公司有统一的前端团队，框架和工具都定好了，那么项目里面，就剩下分工和代码设计问题了，也有些公司有些项目具有特殊性，需要特别定制。

实际上，我很难给出具体的“框架选 Vue”，工具用“webpack”这样的建议，因为工程领域本来就是需要很多妥协和权衡的。

不过，在我们课程的最后一部分，我选择了几个典型的基础设施和体系来讲，会给你分享这几个领域中我的认知。

在本篇文章中，我主要针对一些同学在学习上的疑问，给出了我的答案。你对前端的学习方法有什么样的困惑，欢迎给我留言，我们一起讨论。