加餐三 | 轻松一刻：我想和你聊聊前端的未来

2020-09-07 月影

跟月影学可视化 进入课程 >

讲述：月影
时长 08:53 大小 8.14M

你好，我是月影。今天咱们来聊一个轻松点的话题。

我做前端工程师，也有 15 年了。常常听到有前端开发“抱怨”，“别更新了，学不动了”，也会有人经常问我，Deno、TypeScript 等新轮子层出不穷，未来前端重点方向在哪？还有，在大前端浪潮下，前端开发该如何持续学习、成长？所以今天，我想和你围绕这些话题来聊一聊。

别更新了，学不动了？

我曾经听一位前端技术专家说过，“前端十八个月难度翻一番”，这句话真的说出了前端领域更新换代之快背后的前端开发血泪史。也因此，“别更新了，学不动了”这句话成为了不少前端开发玩梗的口头禅。

但是对我来说，技术发展得越多、越快我就越兴奋。我非常喜欢研究技术，尝试新东西，不怕学习，也更没有学不动这种感觉。我一直觉得，如果一个行业的新东西层出不穷，说明这个行业一直在高速发展，这本身对于从业者来说是一个非常好的事情，因为这说明这个行业中有更多的机会和成长空间。

不过，一些前端开发对技术更新的担忧，我也能够理解。我的建议是，如果你**不盲目**地去追求所谓的“时髦”技术，不去刻意担心自己是否落伍，而是去多观察这个行业，找到技术发展内在的规律和脉络，那么你就知道该怎么前进，不会有任何恐慌了。

在任何一个领域或方向，知识体系都可以分为基础知识和领域知识，而领域知识又可以分为通用领域知识和专用领域知识。它们之间的变化是不一样的，基础知识的变化最慢，其次是通用领域知识，然后是专用领域知识。

用可视化这个方向来举例，基础知识是数学和图形学知识，比如向量、矩阵运算、三角剖分这些知识属于基础知识，它们基本上不会随着时间发生很大变化。而 JavaScript、WebGL 这种属于通用领域知识，它们会改变，也会慢慢发展，比如从 WebGL1.0 发展到 WebGL2.0，从 ES2019 发展到 ES2020，但不会变化、发展得那么快。而类似 ThreeJS、BabylonJS、SpriteJS 和 D3.js 这些属于专用领域知识，很有可能一个大版本升级就会有很大的变化。

学习这些知识，也有不同的方法。一般来说，如果是基础知识，随便什么时候我们都可以学，而且越早学习越好。基础知识就像是你的内功，学好它们，融会贯通之后，学习其他的知识都是事半功倍的。如果是通用的领域知识，一旦你下决心从事这个领域，也是能够尽早学习它比较好，不过由于这些知识是领域相关的，如果能一边学习，一边通过实践来打磨就会掌握得更快。专用领域知识，不一定要很早去学，有一个技巧是，当你用到的时候再去学习它们。如果你没有用到，你可以知道有这门技术，能做什么就行了，不用花大量时间和精力去钻研它们。

如果你觉得技术更新太快，学不过来，很可能就是被这些专用领域知识给“迷惑”了。比如，我听人说前端工程化里的代码打包很重要，于是今天学习了 webpack，明天又去学习 rollup。可实际上这种专用领域知识，我们只需要知道它们能做什么，在用到的时候再去详细学习就好了。

如何看待 Deno、TS 和未来的前端重点方向?

好，解决了第一个问题，我们再来说说 Deno、TS 和未来前端的重点方向。

最近几年流行的编程语言很多都号称是 JavaScript 的替代语言，比如 TypeScript。前端三大框架现在也基本都增加了对 TypeScript 的支持，这背后的本质原因是什么呢？

我认为，近几年 JavaScript 的语言标准发展很快，这背后依托的依然是 Web 应用领域的高速发展，JavaScript 是 Web 领域事实上的“原生语言”和技术标准，很多编程语言都是 JavaScript 的衍生语言。TypeScript 就是其中之一，它是一个很优秀的编程语言，其静态类型对一些规模较大的项目提高代码的可维护性很有帮助，因此现在写 TypeScript 的开发者越来越多，三大框架增加对其支持是顺其自然的事。

Deno 最近也发布了正式的 1.0 版本。我认为它是一个很好的 Runtime，在 Node.js 之后走了另外一条道路，规避了 Node.js 设计上的不足之处。不过，未来 Deno 不见得会取代 Node，它们两个很有可能会一直共同发展下去。但是 Deno 的设计本身就是建立在对 Node 的思考和改进之上的，所以我们学习它，对理解 Node.js 的精髓也非常有帮助。不过，我也只对 Deno 有简单的了解，也希望之后有机会可以去深入地学习和使用它。

因为我自己这两年的主要精力放在可视化领域，主要是可视化渲染方面，所以我觉得可视化是非常值得前端工程师重视的一个领域。随着 Web 技术的发展，视觉特别是 WebGL/GPU 相关的应用场景会越来越丰富，对技术要求也会越来越高。与前端其他大部分技术不同，WebGL 的上手门槛比较高，需要对数学、图形学有比较扎实的基础，而图形学和视觉呈现技术本身的天花板非常高，未来这块一定会有非常大的发展空间。

另外，AI 以及 VR/AR 也是未来前端的发展方向。对于 VR/AR，主流浏览器也开始支持 webXR 技术，而且无论 AI 还是 XR 这些领域，其实也和 GPU 息息相关，所以它们和可视化技术也是有关联的。

除此之外，还有一些跨端技术，从 RN 到 Flutter，经过了很多的发展，但还不是很成熟，而跨端本来就有很多应用场景，未来依然有很大的成长空间。PC 端的 Electron 也不容忽视，作为跨平台应用开发，它是一个非常好用的工具。

最后是一些非常新的技术，比如 Web Assembly、JS Binding，它们是一些跨界交叉领域发展来的前沿的技术，同样也值得我们持续关注。

给前端开发的一些真诚建议

最后，我想从前端工程师以及技术管理的角度，总结一些我自己的经验分享给你。

首先，你要确定自己是不是真正喜欢和热爱前端开发这个职业。当然我相信，大多数同学成为前端工程师，是因为内心真正喜欢这个职业。但是，之前我也听到有些同学说，因为觉得在程序员中前端比较“简单”，或者觉得自己数学或算法基础不好，做前端对这些要求不高，再或者就是觉得前端工程师算是份体面的职业，所以才选择它，其实内心并没不是真的热爱这个职业。

如果你仅仅把它当作一份谋生工作的话，那么你可能在这个职业道路上也走不了多远，肯定也无法达到很高的高度。所以我建议你反思一下，自己是否真的适合前端开发这个职业。

如果你确实热爱这个职业，正在考虑长远发展，我建议你最好选择一个好的平台，一个技术氛围好的团队，一份节奏合适的工作。我说的节奏合适指的是忙闲交替，既不会长时间特别忙，也不会持续特别闲。在这样的节奏下，项目积累再加上自己的学习沉淀，你就可以快速地成长了，而且技术氛围好的团队，也可以加快你学习沉淀的速度。

想在专业上达到一定的高度，因为每个人的情况不一样，所以我们要根据自己的情况来规划。不过总还是能找到一些共通点的，我觉得有一点很合适：找到并突破前端领域的“边界”。这个边界可以是某些有深度领域的技术前沿，也可以是某个交叉领域，如与服务端的交界，与移动客户端的交界等等。如果我们能在这些边界上做出突破，就肯定可以步入前端专家的行列了。

其实前端专家除了需要技术能力以外，还需要有意识地打造自己的个人影响力，锻炼自己的领导力，要让自己心态开放、眼界开阔，不排斥新技术，拥抱开源，多参与社区。

总之一句话，想要在职业之路上达一个比较高的高度，软实力和硬实力我们要两手抓。

今天说了这么多，其实希望如果你真的下定决心在前端这条路上钻研下去，一定别忘了，方向和努力缺一不可。最后，希望正在看这节课的你，可以成为未来优秀的前端专家，我们一起让前端行业变得更好。

今天的分享有没有解决你的疑惑呢？快把这节课分享给你的朋友吧！今天的内容就到这里了，我们下节课见！

提建议

上一篇 用户故事 | 非前端开发，我为什么要学可视化?

精选留言 (2)

写留言

李冬杰
2020-09-07

很庆幸在奇舞团实习，工作期间学习到了月影所讲知识分层，也从有忙有闲的工作中不断迭代自己的能力，工作两年左右找到自己的方向并能为之努力实为幸事。

展开

1

hao-kuai

2020-09-07

知识的分层：基础知识、通用领域知识、专业领域知识。这个点豁然开朗，一直在查找大佬们是如何快速掌握新技术。结合自身经验来说，iOS 开发中 View Controller 有自己的生命周期，在学习 React 的时候很快理解并掌握；学习 Node.js 的 http 模块用来做前后端通信的时候，相对费力，因为之前对于路由-》参数解析-》生成数据-》返回数据这些流程不熟悉，相信其他语言后台开发的同学来学习，上手速度要快很多。那么这里的生命...

展开

1