

结束语 | WebAssembly, 未来已来

2020-10-21 于航 来自北京

《WebAssembly入门课》



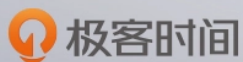
于航

PayPal 高级软件工程师

你好, 我是于航。

面对一项创新性的技术, 我们应该择其重点, 做到知己知彼, 方能在实际工作中灵活运用。除此之外, “实践出真知” 也是对 “浅尝辄止” 的最好补充。

愿 WebAssembly 的未来一路有你。



你好, 我是于航。

首先, 感谢你这一路的陪伴。我本以为就现阶段来说, Wasm 作为一个相对小众的技术领域而言, 大家可能更多地是抱着一种 “看热闹” 的心态, 在我的预想里反馈可能不多。但最后结果大大超出我的预期。同学们都在持续不断地学习, 并且给出了你们对 Wasm 这门技术的反馈。这让我看到了大多数开发者对 Wasm 所抱有的疑问和期待, 相信这也是整个互联网技术圈子对 Wasm 认知的一个缩影。

从七月初开始准备, 到十月中旬的这最后一节课, 历时三个多月的时间, 《WebAssembly 入门课》专栏终于迎来了它的 “尾声”。但专栏的结束却并不意味着学习之路的结束, Wasm 仍然在以惊人的速度不断发展, 每两周一次的 CG 会议也在如期举行。相信时过半年或一年后, 你再回顾来时的路, 会发现这一切都已经有了新的变化。

其实早在 2017 年末我决定开始写书并进行选题策划时，就遇到过这样一个问题。“对于 Wasm 这类正处在技术成长期的开始阶段，并且还在不断快速发展中的技术，我应该怎样选题才能够保障书籍内容的最长时效性？”这是传统纸质出版书籍的一个弊端。对于如今日新月异的互联网技术来说，用传统的纸质出版去呈现会显得有些沉重和单调。当然这里我并不是一概而论，但这种现象是着实存在的。

因此，在《深入浅出 WebAssembly》出版后的第二年，我觉得有必要来跟大家再“交代”一下。Wasm 自 2018 年末到 2020 年这两年内发生的变化太多了，因此，我选择了线上专栏的形式来承载这些内容。这样，当日后 Wasm 再有哪些技术更新的时候，我也能够快速、方便地更新专栏的内容，甚至再根据大家的学习反馈来补充一些“加餐”。

技术领域融合

不知道你在第一次听说 “WebAssembly” 这个词的时候有没有这样的感觉，其中，Web 代表着这是一门 Web 技术，而 Assembly 又表明这门技术跟“汇编”有着密切的联系。因此，“Wasm 这门技术适合哪类同学来学习？”便也成为了一个被开发者们广泛聊起的话题。

事实上，不知道你有没有发现，自上世纪 90 年代以来，伴随着 Web 浏览器大战以及 JavaScript 语言的诞生，在经过这二十几年的发展后，Open Web 开始逐渐“侵蚀”互联网技术的各个角落。因此，一条被称为 Atwood 定律的名言便出现了，那就是：**“凡是可以用 JavaScript 来写的应用，最终都会用 JavaScript 来写”**。

这条定律是由 Jeff Atwood 于 2007 年提出的，Jeff Atwood 是 StackOverflow.com 的创始人之一，也是《程序员的修炼——从优秀到卓越》和《高效能程序员的修炼》这两本书的作者。这里我们不谈及这条名言本身的意义和价值，但可以从侧面看出的是，Atwood 对 JavaScript 语言，甚至是对未来 Web 技术的信心。

而时至今日，相信你也能够明确地感受到，基于 Web 技术的“技术领域融合”正在不断发生。我们不再受限于 C#、Java、Swift 和 Kotlin 等语言，可以直接使用 JavaScript 来编写横跨三端（Web、Mobile 和 Native）的应用。不仅如此，我们也可以使用 JavaScript 来基于 Node.js 开发后端应用。甚至某种程度上，使用 JavaScript 开发 IOT 应用程序也成为了可能。

随着 Web 前端技术逐渐被渗透到其他各个技术领域，你会发现部分前端开发者们也开始逐渐接触那些原来与 Web 前端根本毫不沾边的其他技术体系，比如“编译技术”“自动化脚本”甚至是“沙盒技术”。而这一切都源于 Web 前端技术被应用的场景越来越广泛、其工程越来越复杂，特殊的 case 也越来越多。

我们再回到之前那个问题，到底谁应该来学习 Wasm 这门技术呢？

我的答案是“只要你感到好奇，就可以来学习”。Wasm 诞生于 Web，但却不止于 Web。相信跟随课程一路过来的你肯定深有体会，Wasm 这项技术本身所涉及到的大部分知识领域，都完全超出了 Web 前端体系的知识范畴。但是于你而言，对于理解下面三个问题来说，实际上并不困难。

1. Wasm 是什么？
2. Wasm 现阶段和未来能做什么？
3. Wasm 怎么简单应用？

因此，我的建议是，对于暂时没有实际工作需求的你，只需要去了解关于 Wasm 的这三个问题即可。通过这三个问题，你能够做到心中有数，在日常工作中便会多一份选择。而在真正找到适合的应用场景后，再根据方向逐一进行深入。

当然，不仅是对 Wasm 这类新兴技术，对待其他的技术领域也应该如此。T 字型人才的一个最重要的特点就是“一专多能”。其中的“一专”是指你能够在在一个技术方向深入下去，拥有自己的产出。而“多能”则是在如今这个互联网技术快速融合的时代，能够紧跟变化的一项重要的重要的个人能力。**每个人所拥有的时间都是固定的，在知识海洋中怎样选择必要的部分便显得尤为重要。**

横向与纵向

在专栏上线后的一段时间里，曾有同学私信过我一个问题，我觉得这个问题很典型，因此我想在这里和你谈谈我的看法。这个问题是这样的：“老师你好，我是一名前端开发，现在感觉很迷茫，前端技术没有做到专精，杂七杂八的东西倒是接触了不少，有种什么都想学，却又能力不够的感觉。我是不是应该在前端领域让自己沉下去或者在某个细分领域钻研下去？”

这个问题的答案也正如我刚刚讲的那样，技术体系下的 T 字型人才是一种最理想的状态。但现实的情况是，我们总是会处在一个交叉路口，往左走是回归原始和基础，往右走是各类新奇的事物。如果选择往左走就会被时代发展落下，而选择往右走却又仿佛会丢失原本最重要的东西。

对我而言，我可能会选择往左走，当然这只是我个人的答案，仅供你的参考。

其实对于大部分的新技术来说，你会发现它们早在几年或者十几年前就已经有了类似方案，无非是应用的方式和场景不太相同而已，而技术的本质并没有任何变化。就拿 ASM.js 和 Wasm 为例，这两种技术无非都是想通过减少程序运行时的类型判断，来提升代码的执行性能。只是前者通过 Annotation 的方式来实现，后者则是创造了一种新的 V-ISA 来实现，但本质上这两种方式并不新奇，新奇的点只是它们被应用在了 Web 浏览器中。

包括我们之前介绍的 WASI 的 Capability-based Security 安全模型，也是早在十几年前就已经出现了，只是没有人以 WASI 这种方式来应用而已。说了这么多，我想表达的意思就是说：“在互联网技术体系中，创新性技术大多都是在基础性理论之上构建而来”。而这就意味着，当你把基础打牢后，对于新技术的理解会变得更加容易，甚至你自己会有着更加创新的想法。

5 年，Wasm 从脑海中的一个想法，变成了如今被众人关注的开创性技术。这一路走来我们看到了它的成长，但这还远远不够。未来的 5 年，希望 Wasm 不再只是一个小众技术。“纸上得来终觉浅，绝知此事要躬行”，也希望此刻的你能够在真正的实践中体会到 Wasm 的力量，而 Wasm 的未来发展，将需要你们的支持。

最后，感谢你的一路同行。我还为你准备了一个 [📄 毕业问卷](#)，希望你用 2 分钟填写一下，和我分享一下你对这门课的建议以及你对 Wasm 的想法。希望未来的 Wasm 之路上，还会看到你的身影！

© 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。

精选留言 (4)



Yarco

2020-10-21

私以为这是桌面软件向web应用方向演化的最重要的过程，以前的话受限于cpu,带宽等等诸如此类的硬件条件，web/浏览器中的虚拟机不可能有这种效果。当硬件层次的提上去，当webassembly慢慢成熟，5G普及. 那么桌面软件web化趋势将不可避免.....
与其说是前端开发，不如说是桌面开发..... 是桌面软件应用的技术在web领域复现一次.



4



大土豆

2020-10-21

5年前，他是小众。5年后，他依然小众。。。.



4



言言周

2020-10-21

感谢老师



1



国旗

2021-04-20

老师课程代码有线上仓库地址么？

作者回复: 可以参考这里哈: <https://github.com/Becavalier/geektime-wasm-tutorial>

共 2 条评论 >



1