

## 第185讲 | 狼叔：2019年前端和Node的未来一大前端篇（下）

2019-03-12 阿里巴巴前端技术专家狼叔

技术领导力300讲

[进入课程 >](#)



讲述：黄洲君

时长 11:42 大小 10.72M



你好，我是阿里巴巴前端技术专家狼叔，上一篇文章中，我与你分享了 React/Vue/Angular 三大框架标准化、应用层封装进入爆发期、PWA 进入稳定期、小程序火爆等前端趋势，今天，我将继续跟你聊聊移动端局面、多端拉齐的必然性等内容，以及 2019 年不可忽视的 TypeScript 和 WebAssembly 这两大技术。

### 多端拉齐，并重用户体验

在 AI 时代，没有“端”的支持可以么？明显是不可以的。首先感谢苹果，将用户体验提升到了前无古人的位置。移动互联网兴起后，PC Web 日渐没落。我个人非常欣赏玉伯，在当年无线 ALL IN 战略中，他还是选择留下来继续做 PC Web 的前端。不过，虽然很多公

司的重点转向无线，但 PC 业务也一直没停，这是很多公司的现状，也是客观事实。那么，PC 端这样的“老古董”的出路到底在哪里呢？

1. 我们可以利用 PC/H5 快速发版本的优势，快速验证 AI 算法，继而为移动端提供更好的模型和数据上的支撑。
2. 多端对齐，打好组合拳。既然不能在移动端有更大的突破，大家只能在细节上血拼。

大家的战场已经不是点了，已经升级到打组合策略的阶段了。未来一定是多端拉齐，并重用户体验的。

今天的大前端，除了 Web 外，还包括各种端，比如移动端、OTT，甚至是一些新的物联网设备。我们有理由相信 Chrome OS 当年的远见：“给我一个浏览器，我就能给你一个世界。”如果说的苟且一点：“给我一个 Webview，我就能给你一个世界。”

## TypeScript

我之前就非常关注 TypeScript，但迟迟未下定决心在团队内落地。今年 1 月份北京 Node Party 上组了个局，和几位嘉宾一起聊了一下，确认提效非常明显，落地难度也不大，大家一致认为 2019 年 TypeScript 将有非常大的增长。本身前端团队变大，规模化编程也必然依赖类型系统和面向对象的，从这点上看，TypeScript 也是完胜的。

这里再简单介绍一下 TypeScript，它是有类型定义的 JavaScript 的超集，包括 ES5、ES5+ 和其他一些诸如反射、泛型、类型定义、命名空间等特征的集合，为了大规模 JavaScript 应用开发而生。复杂软件需要用复杂的设计，面向对象就是一种很好的设计方式，使用 TypeScript 的一大好处就是 TypeScript 提供了业界认可的类（ES5+ 也支持）、泛型、封装、接口面向对象设计能力，以提升 JavaScript 的面向对象设计能力。市面上的框架也对 TypeScript 提供了非常好的支持。

- 1.React 对.tsx 支持非常好，比如我在 Midway controller 里支持 tsx 写法，这是非常大胆的，对于后面 react SSR 来说是一个极大便利；
- 2.Vue 从 v2.5.0 之后对 ts 支持就非常好；
- 3.Node.js Web 框架，尤其是 Egg.js 对 ts 支持非常好，当然还有更高级更专注的 Midway 框架，Midway 基于 Egg 生态，同时提供 IoC 等高级玩法；

在使用 Webpack 编译前端应用时，通过 TypeScript-loader 可以很轻松地将 TypeScript 引入到 Webpack 中。有了 TypeScript-loader，就可以一边使用 TypeScript 编写新代码，一边零碎地更新老代码。毕竟 ts 是 js 超集，你有空就改，非强制，特别包容。

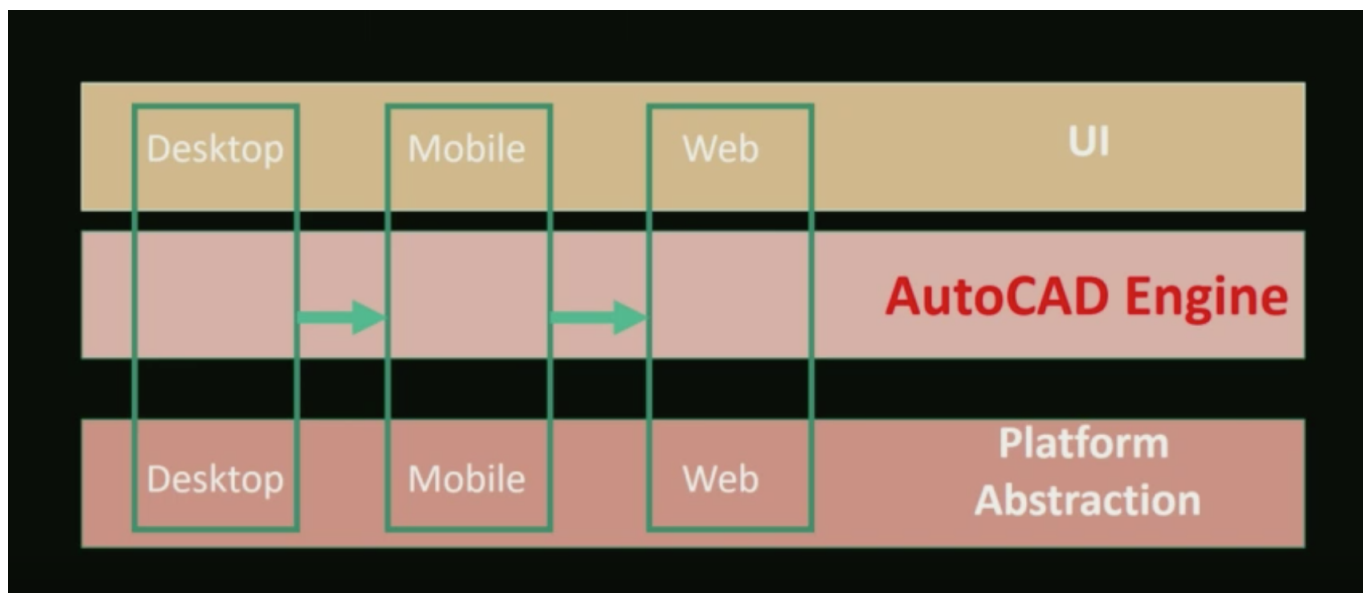
## WebAssembly

WebAssembly 是一种新的字节码格式，目前主流浏览器都已经支持 WebAssembly。和 JS 需要解释执行不同的是，WebAssembly 字节码和底层机器码很相似，可以快速装载运行，因此性能相对于 JS 解释执行而言有了极大的提升。也就是说 WebAssembly 并不是一门编程语言，而是一份字节码标准，需要用高级编程语言编译出字节码放到 WebAssembly 虚拟机中才能运行，浏览器厂商需要做的就是根据 WebAssembly 规范实现虚拟机。这很像 Java 早年的 Applet，能够让其他语言运行在浏览器里。Applet 是一种 Java 程序，它可以运行在支持 Java 的 Web 浏览器内。因为它有完整的 Java API 支持，所以 Applet 是一个全功能的 Java 应用程序。

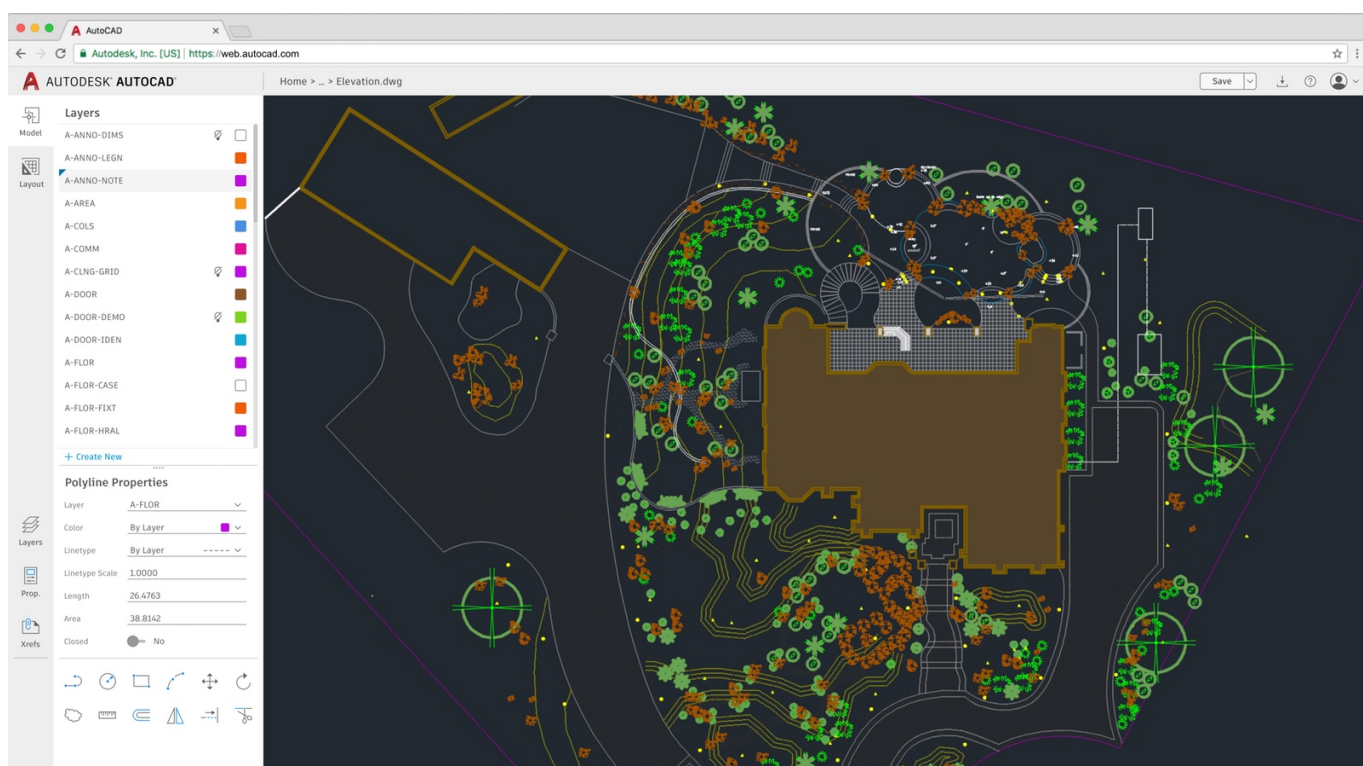
有了 WebAssembly，在浏览器上可以跑任何语言。从 Coffee 到 TypeScript，到 Babel，这些都是需要转译为 js 才能被执行的，而 WebAssembly 是在浏览器里嵌入 vm，直接执行，不需要转译，执行效率自然高得多。

举个例子，AutoCAD 软件是由美国欧特克有限公司 (Autodesk) 出品的一款自动计算机辅助设计软件，可以用于绘制二维制图和基本三维设计。使用它时，无需懂得编程，即可自动制图，因此它在全球被广泛应用于土木建筑、装饰装潢、工业制图、工程制图、电子工业、服装加工等诸多领域。

AutoCAD 是由大量 C++ 代码编写的软件，经历了非常多的技术变革，从桌面到移动端再到 web。之前，InfoQ 上有一个演讲，题目是《AutoCAD & WebAssembly: Moving a 30 Year Code Base to the Web》，即通过 WebAssembly，让很多年代久远的 C++ 代码在 Web 上可以运行，并且保证了执行效率。



本来，我以为 WebAssembly 离我们很远，但在 2018 年 Google I/O 大会亲眼见到 AutoCad Web 应用后，非常震撼，效果如下图所示。



能够让如此庞大的项目跑在 Web 端，真的是非常了不起。通过 WebAssembly 技术，既能复用之前的 C++ 代码，又能完成 Web 化，这也许就是所谓的两全其美吧。

之前，全民直播的前端研发经理赵洋曾分享了 WebAssembly 在全民直播里对直播编解码方面的应用，效果也非常不错。

另外，许式伟在 ECUG Con 2018 上也分享了一个 Topic，主题是《再谈 Go 语言在前端的应用前景》，Go 的发展也遇到了瓶颈，专注后端开发是没办法让 Go 排到第一的，目前的一个方向是借助 GopherJS，将 Go 代码编译为 JS。这种实践是没问题的，和 Kotlin 类似，对于绝大部分 Go 用户也是非常好的。但问题在于，真正的前端不太可能换语言，目前连 Babel、ts 这种都折腾的心累，更何况切换到 Go。“求别更新了，老子学不动了”，这是大部分前端工程师的心声。

从 WebAssembly 的现状来看，对于复杂计算耗时的部分采用其他语言实现，确实是比较好的一种方式。从趋势上看，WebAssembly 让所有语言都能跑在浏览器上，浏览器上有了 vm，浏览器不就是操作系统了吗？

Chrome 的核心 JavaScript 引擎 V8 目前已包含了 Liftoff 这一新款 WebAssembly baseline 编译器。Liftoff 简单快速的代码生成器极大地提升了 WebAssembly 应用的启动速度。不过在桌面系统上，V8 依然会通过让 TurboFan 在后台重新编译代码的方式最终让代码运行性能达到峰值。目前，V8 v6.9 (Chrome 69) 中的 Liftoff 已经设置为默认工作状态，也可以显式地通过 `--liftoff/--no-liftoff` 或者 `chrome://flags/#enable-webassembly-baseline` 开关来控制。另外，Node.js v11 采用的 v8 引擎的 v7 版本，对 WebAssembly 支持更好，虽然这没啥意义，但练手还是蛮好的。

## 移动端

Flutter 是 Google 推出的帮助开发者在 Android 和 iOS 两个平台，同时开发高质量原生应用的全新移动 UI 框架，和 React-native/Weex 一样支持热更新。Flutter 使用 Google 自己家的 Dart 语言编写，刚好今年 Dart 2 也正式发布，不知道二者之间是否有关联。目前 Dart 主攻 Flutter 和 Web 两块，同时提供了 pub 包管理器，俨然是一门全新的语言，学习成本有些高。反观 TypeScript 就非常容易被接受，基于 npm 生态，兼容 ES 语法，因此，2019 年对 Dart 我还是会持观望态度。

除了不喜欢 Dart 外，Flutter 的其他方面都很好，在移动端现在强运营的背景下，支持热更新是必备能力。

关于 Weex，一边骂一边用，很无奈的一种状态。Weex 本身是好东西，捐给了 Apache，目前在孵化中，会有一个不错的未来。但社区维护的非常差，问题 issue 不及时，文档不更新。如果公司没有架构组，还是比较难搞定的。

不过也有很多不错的案例，比如 2018 年优酷双十一活动就是使用 Weex 开发的，效果非常不错。通过自建的可视化活动搭建平台，能够非常高效的完成开发，结合 App 内的缓存，整体效果比 H5 好的多。



我对 Weex 的看法是，以前 Weex 只是解决 H5 渲染效率的问题，但如今强运营的背景，使得 Weex 承载了非常多的内容，比如动画、游戏甚至是图形图像处理等。可以看到，未来 Weex 还会战略性的增加。

总结一下，2018 年大前端的现象：

1. 前端三大框架已趋于平稳，标准化，向 Web Components 看齐。
2. 应用层面开始进入过渡封装周边的阶段，很多细节都会埋在框架里。
3. PWA 平稳发展，兼容 4/5 浏览器，workbox 3 进一步简化开发，另外 PWA 桌面版已经开始兴起，未来会更多。
4. 多端受到重视，不再只是 all in mobile。
5. WebAssembly 让更多语言可以运行在浏览器上，AutoCAD 的 web 版是非常好的例子。
6. 强运营背景下，移动端以前端开发为主，已成定局。Flutter 局势暂不好说，还在观望中（主要是不喜欢 Dart）。
7. TypeScript 落地很好，包容性更好：React 对 .tsx 支持非常好，Vue 从 v2.5.0 之后对 ts 支持就非常好，Node.js（尤其是 Egg.js、midway）对 ts 支持也非常好。



8.5G 时代快来了，互联网的长期在线情况有可能会被打破。本地设备即客户端，可以大胆想想。对前端来说，本地 web 服务，辅助日常开发，类似于 je 这样的模块会越来越多。

综上所述，未来浏览器会越来越重要，Web Os 的概念正在慢慢落地。另外三大框架趋于稳定，写法上也越来越像，学习成本是降低的。但周边应用层面的封装还会是爆发式增长，更多复杂的细节会被包装到应用框架里，可能还有很多不一样的开发方式需要大家熟悉。

对于开发者而言，唯一不变的就是学习能力。掌握了学习能力就能够应对这些趋势变化，无论是在三大框架混战时代，还是后面周边封装时代都能很开心的“折腾”。哪怕有一天 AI 真的能够替人写代码，能应变的人自然也是不怕的。

关于大前端的现状和未来我就分享到这里，希望能对你有所帮助，而提到大前端，就不能忽视 Node.js，它在大前端布局里的意义非常重大，接下来的两篇文章，我将会着重分享一些跟 Node.js 结合比较密切的点，如 API 演进、GraphQL、SSR 等，欢迎继续关注，也欢迎留言与我多多交流。

## 作者简介

狼叔（网名 i5ting），现为阿里巴巴前端技术专家，Node.js 技术布道者，Node 全栈公众号运营者。曾就职于去哪儿、新浪、网秦，做过前端、后端、数据分析，是一名全栈技术的实践者，目前主要关注技术架构和团队梯队建设方向。即将出版《狼书》3 卷。

---

# 技术领导力 300讲

每个技术人都应该知道的管理心经

梁宁 / 著名产品人

张雪峰 / 饿了么CTO

陈皓 左耳朵耗子 / 知名创业者

许式伟 / 七牛云创始人兼CEO

李大学 / 前京东CTO

汤峥嵘 / turtorABC COO

右军 / 蚂蚁金服

程浩 / 迅雷创始人

郭炜 / 易观CTO

新版升级：点击「👤请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 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 第184讲 | 狼叔：2019年前端和Node的未来——大前端篇（上）

下一篇 第186讲 | 赵晓光：如何培养团队竞争力（上）

## 精选留言

写留言

由作者筛选后的优质留言将会公开显示，欢迎踊跃留言。