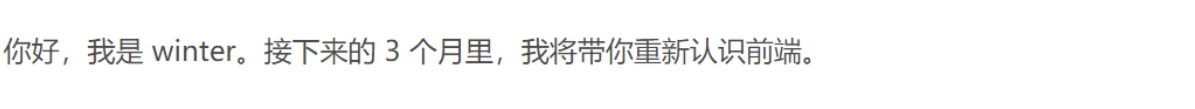
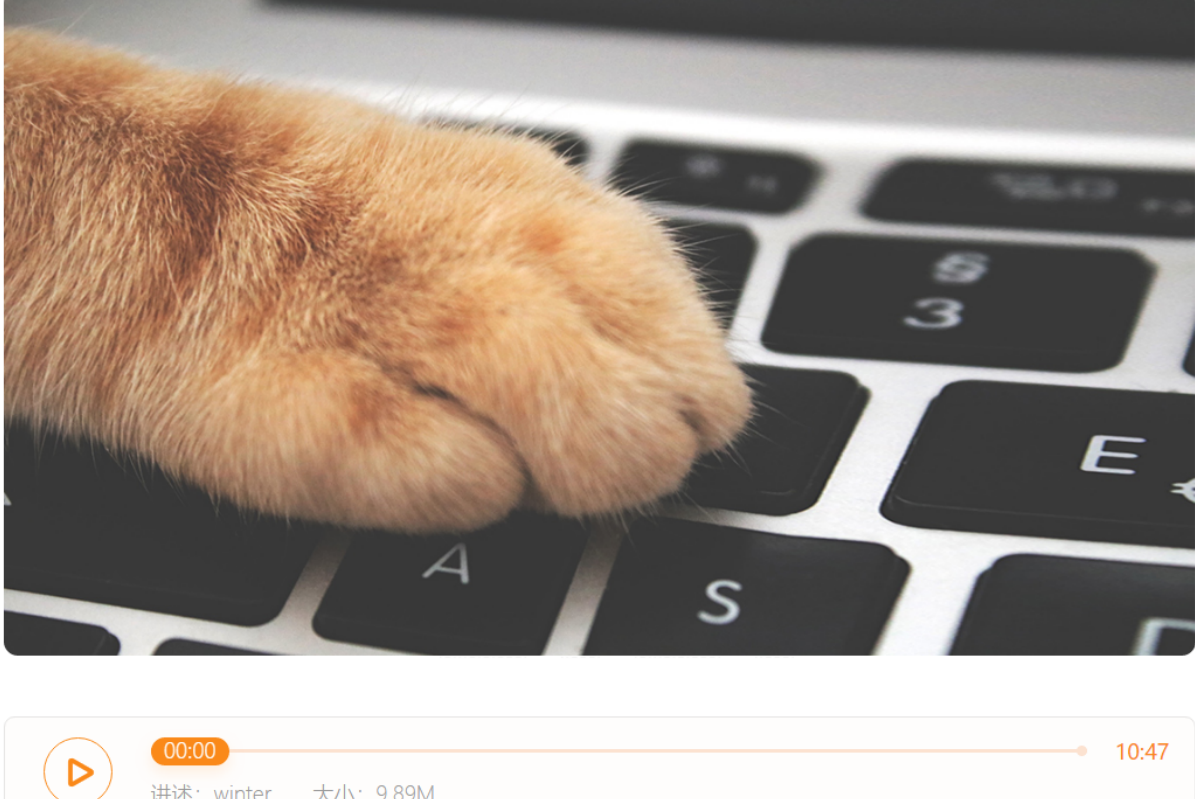


# 开篇词 | 从今天起，重新理解前端

winter 2019-01-14



你好，我是 winter。接下来的 3 个月里，我将带你重新认识前端。

## 前端发展史：从青铜到黄金时代

我自己是在 2006 年开始接触前端的。现在回想起来，那会儿前端还处于史前的“青铜时代”，甚至网页的主要交互都还是依靠切换超链接来完成的。

### 1. 前端的史前记忆：“青铜时代”

那时候，谷歌刚刚基于 Ajax 发布的 Gmail 也没多久，虽然这项伟大的技术标志着 Web 1.0（静态网页）到 Web 2.0（动态网页）的迈进，但在国内依然少有人懂，如果当时谁可以对这项技术侃侃而谈，那简直就是大神的级别了。

当时我还是个学生，喜欢前端纯粹是兴趣使然。那时我混黑白棋社区，想着给黑白棋界面写插件，但自己又不懂界面相关的知识，于是开始通过各种方式学习前端。

真想学的时候才发现网络上的前端资料很是稀缺，所以我基本上都是先从图书馆借书，然后再在电脑上跑案例验证这样的方式来学习的，现在想起来，还真是一段艰难的岁月。

当然，这段经历也为我日后的前端生涯悄悄埋下了一颗种子，我逐渐开始把自己的职业规划路线放在了前端上。

这在当时是个不可思议的想法，因为那时的前端岗位不论从收入上还是在职责上，都远落后于其他岗位。但是，我基于对技术发展趋势的判断，认为前端在未来会越来越重要。

### 2. 进入发展期的前端：“白银时代”

2008 年，我毕业了，也很幸运地得到了一个既能发挥我的 C++ 长处，又能兼顾前端发展规划的职位：微软北京的软件开发工程师，恰好负责的是 Windows CE 上的 IE 浏览器开发，在这里，我接触到了当时最先进的软件工程体系，并且积累了很多 UI 架构经验。

两年后，我加入了盛大做电子书，负责电子书的文本排版工作，这个工作是一个既写底层又写 JavaScript 的岗位，同时排版引擎也是浏览器的重要组成部分，也让我对浏览器的工作原理有了更深入的理解。

在盛大后期，我加入了 WebOS 项目，负责前端框架，我开始基于移动的角度思考前端交互和框架，这份工作让我离前端又近了一步。

但是很遗憾，因为种种原因，我在微软和盛大的几个项目都不算成功，除了电子书实际上市但销量不高，Windows CE 7.0 和盛大的 WebOS 都在公司内部夭折。

自己亲手构建的产品，却因为非技术原因没有服务到最终用户，对我来说，是件非常遗憾的事情。不过，这段时间，也让我更加确信前端技术的价值。

回过头来看，那几年，前端技术开始了它的大踏步发展，那一段时间，可以说是前端的“白银时代”。最直观的表现之一就是前端逐步从后端分离了出来，它的代码也变得复杂了起来，还需要保存数据、处理数据、生成视图等等。

悄然之间，我发现前端已经从零散的“工序”逐步发展成为有体系和发展目标的职能，同时，在越来越大的前端团队中，工程化的思想也逐渐萌芽。我深有感触，前端已经不再是别人眼中的“小菜一碟”了。

### 3. 从前端到“全端”：“黄金时代”

在这样的行业背景里，从盛大离职后，我加入了阿里巴巴做手机淘宝开发，这也是我首次从事真正的前端工作。

在手机淘宝，前端团队的各种基础设施也逐渐建立了起来，从最开始的多屏适配方案、基础库、工具链到页面搭建平台和性能体系，最后到客户端融合方案 Weex，我随着团队一起经历了业务发展、团队自身成长和行业变革。

与此同时，在我加入阿里巴巴后的这段时间里，随着移动时代的到来，前端也开启了自己的“黄金时代”，它的职责变得更加重要，有了独立的发布权限，技术也变得更加复杂。

一些传统软件开发和互联网服务端的方法论逐步移植到前端开发中，并形成了前端自己的工程体系，诸如持续集成、前后端分离、线上监控……

架构方面，前端架构的任务也从简单的解决兼容和风格问题，逐步过渡到提倡组件化和 UI 架构模式，最后形成了新一代的前端框架 React、Vue 和 Angular，他们也在竞争和互相学习中成长。

## 前端开发之痛：散点自学 + 基础不牢

正当处于“黄金时代”的前端技术在全力以赴极速前行之时，我却发现，前端开发者们的步伐似乎渐渐有些跟不上了。

因为在我职业发展的后半段，面试和培养前端工程师已经成为我的长期工作职责。在这期间，我意识到，目前的前端教育几乎是完全缺失的。

在面试应届生过程中，我会习惯性地问表现比较好的同学“你是如何学习前端的”，而我得到的答案多是“自学”“在社团学习”，却从未听到过“在学校学习过”这样的答案。

而对于工作之后的前端开发者来说，没有系统学习的问题仍然存在，常常有一些具有多年从业经验的工程师，仍然会在看到一些用法时惊呼：“还可以这样！”

在我看来，这些用法都是一些基础的不能再基础的知识点，但是他们却浑然不知。

**如果深入进去了解，你会发现，表面上看他们可能是一时忘记了，或者之前没注意，但实际上是他们对前端的知识体系和底层原理没有真正系统地理解。**

在阿里工作的时候，我戏称很多同学学前端的方式是“土法学前端”，他们对于知识的理解基本都停留在点上，从来没有大范围把这些点串成线，形成自己的知识体系，因此才会出现上面说的遗漏和盲点。

这个问题在一些一直在小公司工作的前端工程师身上非常突出。

经常能看到一些案例，一些有技术追求、有热情的工程师，因为技术敏感度和主观能动性都不错，所以工作了五六年之后，逐步开始在自己的公司做一些技术管理相关的事情了。

但是，由于他们所在公司的业务并不复杂，也没有技术积累，所以他们自身的技术水平其实并不高，可以说还处于非常初级的阶段（可能面试连阿里 P6 都过不了）。

**做了管理，技术没跟上，并且还错过了最佳的学习时间，这个境遇可想而知，他们在工作中的大概率只能是被动地接受需求解决问题，然后也同时焦虑着自己的未来，焦虑着自己的竞争力。**

关于前端工程师成长，我认为需要两个视角。一是立足标准，系统性总结和整理前端知识，建立自己的认知和方法论；二是放眼团队，从业务和工程角度思考前端团队的价值和发展需要。只有这样做，才能够持续发展，在高速发展的技术和工程浪潮中稳稳立足。

这也正是“重学前端”这个专栏的初衷，我希望提供一些视角，带你以完备、体系化的方式理解和思考前端的基础知识和工程实践。

除此之外，前端工程师也是开发工程师的一员，除了前端自身的领域知识和工程特点外，你还需要了解程序员通用的编程能力和架构能力。

所以，想要成为优秀的前端工程师，我觉得你需要通过系统地学习和总结获取知识，通过练习获取编程能力，通过工作经验来获取架构和工程能力。

当然，一个为期 3 个月的专栏无法穷尽前端庞杂的知识，讲知识点也不是我们的目标。知识点的再好再全，也不一定能够记得住。

我们专栏的目标是帮助你建立自己的知识体系，根据你自己的理解把前端的领域知识链接起来，形成结构，这样做，不但能帮助你记忆知识，还能在其中发现自己知识的缺失，甚至可以凭借知识体系来判断知识的重要性，来决定是否要深入学习。

在这个专栏里，我将知识分成四个模块来讲解：

- JavaScript
- CSS 和 HTML
- 浏览器实践
- 前端综合应用。

前三个模块是前端的基础知识，是个人的前端能力提升，而模块四则是前端团队发展相关的内容，有助于你和团队的整体提高。

在 **JavaScript 部分** 中，我主要会从文法和运行时的角度去讨论 JavaScript 语言。它们是互相关联的，而语义就是文法到运行时之间的桥梁；它们分别又是完备的，任何语言特性都离不开两者，所以从语法和运行时的角度，我们都可以了解完整的 JavaScript。

**CSS 和 HTML 部分**，会侧重从语言和设计思想的角度来讲解，我们同样可以对两者的全貌建立一些认知。

**浏览器部分**，包含了浏览器工作的原理和一些重要的 API，包括 BOM、DOM、CSSOM 和其他一些内容。了解了这些知识，你才能把 JavaScript 和 HTML、CSS 连接起来，用 JavaScript 来实现功能。

**前端综合应用部分**，主要是我的一些工作经验，我会选择我在手淘和淘宝工作中的一些案例来辅助讲解。

前面，我说到前端是一个非常年轻的职业，但我仍然认为前端具有很多空间和机会，一些基础设施仍然简陋，前端的能力可以带来更多的业务场景，这些有待于我们去发掘。

前端社区非常活跃，新技术也在不断出现。在这样的环境下，机会和竞争并存，学习也犹如逆水行舟，不进则退，建立自己的知识体系和方法论，你才能够保持领先优势。

我希望从我的经验出发，给你一些启发和帮助，并借由这个专栏帮你建立自己的前端知识体系。同时，我也相信，在你们中间一定会产生更多能够带领前端领域取得突破的、优秀的前端工程师。

最后，也希望你能和我分享你的前端故事和经历，你是怎么走上前端道路的？你希望将来成为怎样的前端工程师？欢迎在留言区与我分享。