

01 | 明确你的前端学习路线与方法

2019-01-17 winter



你好，我是**winter**。今天我们一起聊聊前端的学习路线与方法。

在“开篇词”中，我和你简单回顾了前端行业的发展，到现在为止，前端工程师已经成为研发体系中的重要岗位之一。可是，与此相对的是，我发现极少或者几乎没有大学的计算机专业愿意开设前端课程，更没有系统性的教学方案出现。大部分前端工程师的知识，其实都是来自于实践和工作中零散的学习。

这样的现状就引发了一系列的问题。

首先是**前端的基础知识**，常常有一些工作多年的工程师，在看到一些我认为很基础的**JavaScript**语法的时候，还会惊呼“居然可以这样”。是的，基础知识的欠缺会让你束手束脚，更限制你解决问题的思路。

其次，技术上存在短板，就会导致前端开发者的上升通道不甚顺畅。特别是一些小公司的程序员，只能靠自己摸索，这样就很容易陷入重复性劳动的陷阱，最终耽误自己的职业发展。

除此之外，前端工程师也会面临技术发展问题带来的挑战。前端社区高度活跃，前端标准也在快速更新，这样蓬勃发展对技术来说无疑是好事，但是副作用也显而易见，它使得前端工程师的学习压力变得很大。

我们就拿**JavaScript**标准来说，**ES6**中引入的新特性超过了过去十年的总和，新特性带来的实践就更多了，仅仅是一个**Proxy**特性的引入，就支持了**VueJS**从**2.0**到**3.0**的内核原理完全升级。

缺少系统教育+技术快速革新，在这样的大环境下，前端工程师保持自学能力就显得尤其重要了。

那么，前端究竟应该怎么学呢？我想，我可以简单分享一下自己的经验。

学习路径与学习方法

首先是**0基础入门**的同学，你可以读几本经典的前端教材，比如《JavaScript高级程序设计》《精通CSS》等书籍，去阅读一些参考性质的网站也是不错的选项，比如[MDN](#)。

如果你至少已经有了**1年以上的工作经验**，希望在技术上有一定突破，那么，这个专栏就可以是你技术进阶的一个选项了。

在这个专栏中，我希望传达的不仅仅是具体的知识点，还有体系架构和学习方法。我希望达到三个目标：

- 带你摸索出适合自己的前端学习方法；
- 帮助你建立起前端技术的知识架构；
- 让你理解前端技术背后的核心思想。

在开始具体的知识讲解之前，这篇文章中，我想先来谈两个前端学习方法。

第一个方法：建立知识架构

第一个方法是建立自己的知识架构，并且在这个架构上，不断地进行优化。

我们先来讲讲什么叫做知识架构？我们可以把它理解为知识的“目录”或者索引，它能够帮助我们**把零散的知识组织起来，也能够帮助我们发现一些知识上的盲区**。

当然，知识的架构是有优劣之分的，最重要的就是逻辑性和完备性。

我们来思考一个问题，如果我们要给JavaScript知识做一个顶层目录，该怎么做呢？

如果我们把一些特别流行的术语和问题，拼凑起来，可能会变成这样：

- 类型转换；
- this指针；
- 闭包；
- 作用域链；
- 原型链；
-

这其实不是我们想要的结果，因为这些知识点之间，没有任何逻辑关系。它们既不是并列关系，

又不是递进关系，合在一起，也就没有任何意义。这样的知识架构，无法帮助我们去发现问题和理解问题。

如果让我来做，我会这样划分：

- 文法
- 语义
- 运行时

为什么这样分呢，因为对于任何计算机语言来说，必定是“用规定的文法，去表达特定语义，最终操作运行时的”一个过程。

这样，JavaScript的任何知识都不会出现在这个范围之外，这是知识架构的完备性。我们再往下细分一个层级，就变成了这个样子：

- 文法
 - 词法
 - 语法
- 语义
- 运行时
 - 类型
 - 执行过程

我来解释一下这个划分。

文法可以分成词法和语法，这来自编译原理的划分，同样是完备的。语义则跟语法具有一一对应关系，这里暂时不区分。

对于运行时部分，这个划分保持了完备性，**我们都知道：程序 = 算法 + 数据结构，那么，对运行时来说，类型就是数据结构，执行过程就是算法。**

当我们再往下细分的时候，就会看到熟悉的概念了，词法中有各种直接量、关键字、运算符，语法和语义则是表达式、语句、函数、对象、模块，类型则包含了对象、数字、字符串等.....

这样逐层向下细分，知识框架就初见端倪了。在顶层和大结构上，我们通过逻辑来保持完备性。如果继续往下，就需要一些技巧了，我们可以寻找一些线索。

比如在JavaScript标准中，有完整的文法定义，它是具有完备性的，所以我们可以根据它来完成，我们还可以根据语法去建立语义的知识架构。实际上，因为JavaScript有一份统一的标准，所以相对来说不太困难。

如果是浏览器中的API，那就困难了，它们分布在w3c的各种标准当中，非常难找。但是我们要

想找到一些具有完备性的线索，也不是没有办法。我喜欢的一个办法，就是用实际的代码去找：**for in** 遍历**window**的属性，再去找它的内容。

我想，学习的过程，实际上就是知识架构不断进化的过程，通过知识架构的自然延伸，我们可以更轻松地记忆一些原本难以记住的点，还可以发现被忽视的知识盲点。

建立知识架构，同样有利于面试，没人能够记住所有的知识，当不可避免地谈到一个记不住的知识，如果你能快速定位到它在知识架构中的位置，把一些相关的点讲出来，我想，这也能捞回不少分。（关于前端具体的知识架构，我会在02篇文章中详细讲解。）

第二个方法：追本溯源

第二个方法，我把它称作追本溯源。

有一些知识，背后有一个很大的体系，例如，我们对比一下**CSS**里面的两个属性：

- **opacity**;
- **display**。

虽然都是“属性”，但是它们背后的知识量完全不同，**opacity**是个非常单纯的数值，表达的意思也很清楚，而**display**的每一个取值背后都是一个不同的布局体系。我们要讲清楚**display**，就必须关注正常流（**Normal Flow**）、关注弹性布局系统以及**grid**这些内容。

还有一些知识，涉及的概念本身经历了各种变迁，变得非常复杂和有争议性，比如**MVC**，从1979年至今，概念变化非常大，**MVC**的定义几乎已经成了一段公案，我曾经截取了**MVC**原始论文、**MVP**原始论文、微软**MSDN**、**Apple**开发者文档，这些内容里面，**MVC**画的图、箭头和解释都完全不同。

这种时候，就是我们做一些考古工作的时候了。追本溯源，其实就是关注技术提出的背景，关注原始的论文或者文章，关注作者说的话。

操作起来也非常简单：翻翻资料（一般**wiki**上就有）找找历史上的文章和人物，再顺藤摸瓜翻出来历史资料就可以了，如果翻出来的是历史人物（幸亏互联网的历史不算悠久），你也可以试着发封邮件问问。

这个过程，可以帮助我们理解一些看上去不合理的東西，有时候还可以收获一些趣闻，比如**JavaScript**之父 **Brendan Eich** 曾经在**Wikipedia**的讨论页上解释**JavaScript**最初想设计一个带有**prototype**的**scheme**，结果受到管理层命令把它弄成像**Java**的样子（如果你再挖的深一点，甚至能找到他对某位“尖头老板”的吐槽）。

根据这么一句话，我们再去看看**scheme**，看看**Java**，再看看一些别的基于原型的语言，我们就可以理解为什么**JavaScript**是现在这个样子了：函数是一等公民，却提供了**new this instanceof**等特性，甚至抄来了**Java**的**getYear**这样的Bug。

结语

今天我带你探索了前端的学习路径，并提出了两个学习方法：你要试着建立自己的知识架构，除此之外，还要学会追本溯源，找到知识的源头。

这个专栏中，我并不奢望通过短短的40篇专栏，事无巨细地把前端的所有知识都罗列清楚，这本身是MDN这样的参考手册的工作。但是，我希望通过这个专栏，把前端技术背后的设计原理和知识体系讲清楚，让你能对前端技术产生整体认知，这样才能够在未来汹涌而来的新技术中保持领先的状态。

在你的认识中，前端知识的结构是怎样的？欢迎留言告诉我，我们一起讨论。



精选留言



莲

15

问题一提出来不知道怎么回答，看评论区才反应过来，我的认识中前端就是html负责骨架,css负责外表和js负责行为

2019-01-17



CC

8

在阅读这篇文章之前，我对于前端知识结构的划分是基于「语言」：

前端 = HTML + CSS + JavaScript

每种语言下，又混杂了基础知识以及相关的工具。

比如我之前认为的 JavaScript 包括：

- * JavaScript 基础
- * Package Manager (NPM, Yarn)
- * Build tools (NPM Scripts, ESLint, Webpack...)
- * Frameworks (React, Vue.js, Angular)
- * Testing tools (Jest ...)

现在知道自己的认识有两个误区：

1. 这个知识架构没有完备性，这样划分，总有在这个结构之外的知识；
2. 学习的时候会疲于奔命，总想“完整”的学完所有的知识，却走了弯路。

感谢 winter 老师。

2019-01-17



把那个产品经理祭天

👍 6

对于科班出身的学技术都是这一套思路吧！

感觉 winter 叔这篇主要是站在非科班出身或半路出家的前端从业者的角度来看的，确实我刚开始接触前端这一陌生领域来说就是这样，没有知识体系，都是知识点拼接的一张记忆网络，很多点觉得不可思议，难以记忆。实际是上不了解本质，不了解来龙去脉，所以很多知识在脑海中就不够形象。

期待接下来 winter 的思路！

2019-01-17



吴灿城

👍 4

如果让自己描述前端的知识结构，大致会按照下面这个roadmap来。

<https://github.com/kamranahmedse/developer-roadmap/blob/master/readme.md>

winter老师第一种建立知识架构的方式，更像是做了进一步抽象。

2019-01-17



棚头傀儡

👍 3

我是黑马出身，emmm我的知识体系是初级前端要会哪些，中级要会哪些，又比如原型链和this之类的是中级必会的

2019-01-17



stanny

👍 2

接触前端一年左右的时间，正式做前端大概也就四五个月的时间，我对前端的知识结构理解大概如下划分：

1. ECMAScript 标准，DOM、BOM、ES6、Node.js

2. html/css

3. 各种前端框架以及构建工具

4. http

2019-01-17



芳玥

👍 2

我所认为的前端的话，基础系列**html+css+js**，进阶一点就**css3+js**高级应用+前端框架+构建工具。工作快近2年，最近觉得焦虑。是自己太水，看看面试要求，反省自己，一看只会基础的但又不精。现在跟着课程体系，想建立好自己的知识体系，把握19年，不至于荒废时光。

2019-01-17



reece

👍 2

1.1，代码打包，如何打包代码发送到服务器。

1.1.1，资源加载

1.1.2，webpack打包

1.1.3，摇树优化

1.2，网络传输，因为传统上是浏览器发起页面请求，然后服务器端发送html页面。

1.2.1，资源加载

1.2.2，首屏页面加载优化

1.3，浏览器端的页面渲染。

1.3.1，浏览器如何渲染一张html页面

1.3.2，渲染性能优化

1.4代码编写

1.4.1，html页面架构

1.4.1.1

...

1.4.2，css页面样式

1.4.2.1，

.....

1.4.3，js交互

1.4.3.1，

.....

2019-01-17



to be a better man

👍 2

在我学习初的知识划分就是**html**页面内容，**css**页面布局样式，**js**逻辑实现。**js**的知识划分基本上就和**js**高级程序设计这本书的目录一样，因为好多书的目录结构关于**js**基础部分的都是这么划分的。期待后面的内容

2019-01-17



37°C^boy

👍 1



现在还没有意识到追本溯源的重要性，但是对于知识点深入是需要打破沙锅问到底的精神，后面可以尝试下，看下效果怎么样

2019-01-17



Destroy、

👍 1

说实在，我现在知识体系都是乱的。都是东看一点西看一点拼凑起来的。

2019-01-17



岑中归月

👍 1

工作一年多，我对前端的知识架构的认识就是：顶层目录html+css+javascript，二级目录是把这三个里面的知识点生硬并且毫无关联的列出来，三级目录是继续划分，，，，，看了老师的讲解，觉得这样划分的知识架构毫无逻辑可言，期待老师后面的课程。

2019-01-17



新哥

👍 1

前端不是切页面的吗？

2019-01-17



张优秀

👍 1

很有感触，一直在追寻新技术的同时也需要建立自己的知识架构，受人以鱼，不如受人以渔。
winter老师做到了后者，

2019-01-17



YUANWOW

👍 1

大家都觉得HTML容易，非常简单

就算不会，坐下来花一天时间看看MDN几乎都能写出来了

我们都知道写HTML要语义化，但是HTML5的语义我感觉非常复杂

有多少人能保证写出来的网页完完全全遵循了正确的语义？

能达到屏幕阅读器理解的程度？

我们想写语义正确的网页，但是也不想被语义复杂性所拖累（本来读的也不是文科，div一分钟能写出来的东西在语义上要纠结好久）

程老师，我的问题是，作为一名前端工程师，平常工作的时候写HTML究竟要语义化到哪种程度呢？

2019-01-17



Russell

👍 1

感谢winter老师,话说我的名字里也有一个冬，因为我是冬天早晨出生的，所以叫晓冬，有缘吧。

刚工作那会儿，觉得大学里学的好多东西都没有用。

工作时间越久，越觉得大学里，学的好多理论并不是没有用的。

2019-01-17



北去

👍 1



北南



在我之前的理解中我把前端按语言分三个大类型，**html**负责网页的结构，**css**负责网页的样式，**js**负责网页的逻辑和交互，通过深入学习后，当然主要是深入**JavaScript**的学习，我也是按高程那样，以**API**来划分，基本的**ecma**，数组的，对象的，字符串的，.....

2019-01-17



七七



半路出家自学上岗，没有经过系统的学习，基础不扎实，有一定的全局意识，做了管理岗，总希望做的更好一点，奈何技术不够过硬，焦虑又慌张，希望听完课有所收获和提升

2019-01-17



翼扬



和我的很多想法很吻合，早一天看到这个，或许当初能少走些弯路。

2019-01-17



Hanson



学习方法:1.知识体系结构; 2.对知识追本溯源;
我心目中的前段:首先是有最基础的**JavaScript**、**Html**、**css**组成;

2019-01-17



Sunshine



任何计算机语言都是用来解决实际问题，解决人工痛点，用程序化来提高效率。而前段在这一个过程中是离用户最近的，为用户提供了友好的交互界面，承接后端数据逻辑和用户沟通的功能

从技术角度web前端=**js**（动态交互）+**css**（样式负责装饰）+**html**（页面骨架结构）。这一体系每一部分技术都在不断演进，出新的标准知识等。随着前端的不断复杂引入了工程化的配套知识体系帮助简化工作，提升效率，解决各种实际问题等。工程化演进过程有**grunt**、**gulp**、**roll up**、**webpack**等等。前端框架也在不断的演进发展。就web前端而言**js**、**css**、**html**又都是运行在浏览器上的深入学习不可避免需要了解浏览器原理，不同内核又不太一样，在了解差异过程也是那些优秀工程师解决问题的思路也有助于个人提高。目前前端知识体系比如**es**新标准，可编程**css**类似**sass**、**less**、**postcss**等这些浏览器还不识别需要工程化工具去做转译

2019-01-17



无垠



我认为前端大体上来说就分三块:

1.HTML

2.CSS

3.JavaScript

其后的每一项都是他们的分支。

2019-01-17



种豆苗



的确，大学里面并不会老师专门教我们前端。因为前端的知识太杂，知识体系过于庞大，更新速度太快，可能老师今天说的东西，明天就已经淘汰了，或者是老师自始至终说的都是淘汰

的类容。有的老师，可能自己都对前端没有一个明确的概念，更不用说知识体系了。所以在这样的教育环境下，我对前端的划分，也就简单到：结构→HTML 样式→CSS 逻辑（交互）操作→JS。根本从未想过再往下划分会是怎样，因为太过于零碎，或者说是我自己压根就没法细分下去。毕业参加工作后，都是用到什么，就去学什么，知识就更加碎片化了，希望可以跟着winter老师的思路，可以慢慢的建立起自己的知识体系。

2019-01-17



chen

0

前端知识体系现在是越发重要了，之前太满足于对于新知识和框架的尝试，但是又没有深究，满足于自己知道那些基础知识就忽略了比用法更加基础的东西，现在慢慢知道前端整个知识体系的重要性了，新的东西未必好，但是东西肯定都有共同点，知识体系完备后再去看那些框架实现才能有记忆点也更加知道这个框架好还是不好，有什么不足的地方

2019-01-17



CrazyRaya

0

后端前端，刚工作3个多月，对于前端的一些框架还不是很熟练，js也就是原来学的基础，尤其是css，头痛致死.....希望能系统学完之后对前端有一个新的认识.....

2019-01-17



cheer

0

我认为：

基础部分：（html/css/js）

延伸部分：（.../html4/html4.1/html5）+（less/sass/postcss/...）+（jquery/react/vue/angular/...）

。

支持部分：（开发工具（webstorm/HBuilder/...）+打包工具（webpack/rollup/gulp/...）+依赖（nodejs））。

流程部分：搭建项目的步骤。

解决的问题目标：快速/高效/友好 动态展示一个让用户浏览起来舒服的页面。

2019-01-17



Cassie

0

加油，第二天

2019-01-17



sophie

0

前端知识结构：html搭框架，就类似于做一个人的模型；css就是给这个框架做上样式，类似于给人穿上好看的衣服；js就是操作，类似于给人一些行为动作，让她活起来。

2019-01-17



Ace Chen

0

css怎么布局，js怎么模块化，我个人感觉比较难

2019-01-17



琪琪fia

0



我之前对前端的理解就是写页面（HTML+CSS+JS）,等到前后端分离了，我对前端的理解就变成了框架（vue, React）+业务逻辑，平时对于前端知识的学习很零散，公司用什么就学什么，遇到问题解决问题，从来没有梳理过，使得自己的学习很被动，很凌乱，看了老师的文章有种醍醐灌顶的感觉，希望这40篇文章可以帮助自己对前端有一个真正的认识新的学习思路及方法，谢谢老师

2019-01-17



_Nomo

期待干货！

0

2019-01-17



yy不会笑

是的，大学都没系统学习过前端。

0

2019-01-17



夜空中最亮的星（华仔）

运维同学也已经混入前端阵营

0

2019-01-17



醉月

前端重要的还是在展示和交互吧
这也是我以后打算涉猎点ui的原因
还是那句话前端更人性化
html是基本的骨架
css是皮肤
js是神经和血脉
三者缺一不可

0

2019-01-17



^O^

为了偷懒，前端一直在发展

0

2019-01-17



Artyhacker

我认为html css js是过去的前端三大件，现在的知识结构应该是

1. 基础知识(包括html css js、网络协议);
2. 开发模式(模块组件的开发、框架的设计思想和实现原理);
3. 架构知识(前端系统架构、代码转译、打包优化);
4. 浏览器(编译、渲染的原理及优化方法)。

0

2019-01-17



Jack Q

自学进入前端两年半了，一直在思考前端是干什么的？

0

我理解中的前端就是为用户提供可以用的视图界面。

这里就细分为视图和可以用。

视图：

在哪里展现的？ -> 浏览器（粗略上的运行环境，细扣就是内核了）

通过什么形式展现的？ -> **html + css**

可以用：

交互：**html**、**js**（事件提供了交互动作+脚本则是注入了业务）

提供更好的用户体验：

1、视觉上**css**、**js**（特效）

2、功能与体验感 **js**（性能优化提升流畅度、工程化让功能更快的上线）

3、**html**（语义化提升**SEO**机制，更易让用户找到、看到）

和大神的理解还是差别挺大的。[]

2019-01-17



亮亮

👍 0

一直觉得基础是**html**，**css**，**js**，进阶是框架**vue**，**react**，小程序等等，在新的框架中去巩固基础。

2019-01-17



jviLee

👍 0

粗鄙认为前端知识架构可以分为渲染层和网络层两方面，不论前端如何发展变化，比如**react**，**vue**，**ng**，还是**flutter**，小程序等等，本质上脱离不了这两方面，另外前端工程师在接触一个新语言或新框架，最难不是学习语言本身（官方文档可以解决），反而棘手的是如何去工程化（更好组织代码，打包工程）和去优化，这是做前端两年的感受

2019-01-17



脉不语

👍 0

设计师转前段一枚！在前段知识点碎片化之后看了一些书来系统学习！可是发现书里面讲的都不是最新的知识了！这时候就去关注一些技术的作者的各种社区和文章

2019-01-17



多啦斯基周

👍 0

总是感觉前端的东西太琐碎太杂，要学的东西太多，没有一个清晰的思路和体系

2019-01-17



谌文琦[]

👍 0

前端的知识架构：**html** **css** **JavaScript** 然后**react** **vue** 等框架

2019-01-17



谭鹏

👍 0

前端包涵甚广,html css javascript iOS android 以及 react native weex这样的跨平台技术

2019-01-17



王智

👍 0

学习后端的我也来蹭点经验.无论是前端还是后端,我感觉学习方法都是相通的,甚至有时候我觉得前端的更新以及新技术的出现会更甚于后端,所以学习方法很有必要.我觉得老师说的两个方法都很好,追本溯源了解技术出现的背景和解决的问题,可以更快地了解这个技术以及进行学习,根据自己的问题来学习技术. 同样,构建知识架构也是很有必要的,我一直想总结一下后端的知识体系,一直没有思路,等下篇文章看看前端的知识体系来总结一下吧. (◡‿◡)

加油加油!!!

2019-01-17



波拉拉

👍 0

0基础来学习思路的, 前端架构大概就是视图+逻辑处理吧。winter老师的追本溯源法第一次听说, 但真的让人豁然开朗!

2019-01-17



越努力越幸运

👍 0

感谢winter老师, 希望后面更多干货

2019-01-17



allen-黄河水

👍 0

知识结构的思维导图需要重构了

2019-01-17



FU½

👍 0

即便是现在的大学依然没听说把前端独立出来作为一门课程, 从这个角度上说, 大家都是半路出家

2019-01-17



Lyy

👍 0

听书很火, 我就来了, 刚学前端的小白, vue开发

2019-01-17



李锐浩的微信

👍 0

更新太慢, 希望日更

2019-01-17



Harry

👍 0

对我来说, 前端就是 JavaScript+HTML+CSS

2019-01-17



尹先生

👍 0



winter老师您好，模块里面的课程是以什么方式授课呢，是通过视频进行代码讲解还是通过类似于文章的形式去讲解呢。

祝工作顺利！

2019-01-17



magic

0

js很渣，又不知道从哪学起，导致自己技术上升不了，希望老师可以指导一下

2019-01-17



刘滨涛(陆逾)

0

在我的理解中呢，抛去浏览器内核等，直接就是js,html,css了。这三个技术构成了一个完整的前端生态，无论前端怎么发展，比如各种框架，各种打包工具，都是基于其中之一或者混合产生的技术。对您的追本溯源我还是倍感认可，并且茅塞顿开，原来还有这条路可以帮忙理解一们语言。之前会去学习一们语言的原理和设计，但却忽略了知其然而不知其所以然，工作3年多了，希望还来得及去弥补下缺失的知识。

2019-01-17



AlexZENG·Q

0

对于前端认知还是对html,css和js三个构建网页的阶段，包括基本语法，加上用法，接着是实践设计网页。

2019-01-17



kisnotundercover

0

我是第一个，先占个坑，感谢winter老师

2019-01-17



kisnotundercover

0

我是第一个，先占个坑，感谢winter老师

2019-01-17