

02 | 列一份前端知识架构图

winter 2019-01-19



你好，我是 winter。

在上一篇文章中，我们简要地总结了前端的学习路径与方法，我们提到的第一个学习方法就是：建立知识框架。那么，今天我们就一起来列一份前端的知识框架图。

在开始列框架之前，我想先来谈谈我们的目标。实际上，我们在网上可以找到很多参考资料，比如 MDN 这样的参考手册，又比如一份语言标准，但是我们的课程既不是一本参考手册，也不是一份语言标准。参考手册希望做到便于查阅、便于理解和全面，语言标准的目标是严谨、无遗漏、无歧义。

而我们的课程有什么不同呢？我认为，作为一个课程，有两个目标：一个是把无法通过查阅解决的原理和背景讲清楚，另一个是把不方便查阅和记忆的内容整理好。

我会尽量避免像前面提到的两种文档一样逐条目罗列知识点和细节，当然，这不是在说两种文档没有价值，而是我们各有分工，参考手册和语言标准做的事情，我们没必要重复去做，即使做了也不一定能做得更好。

在这个课程里，我希望能和你一起打造一个前端知识的框架，再把知识点做个遍历，这其中，有原理和背景的部分，我去讲解知识的原理和背景。如果没有的话，我们就去讲整理和记忆这部分知识的方法，这样，即使你遇见无法一下子记住的知识，也可以很容易地查阅参考手册和标准来解决。

如果让我做一个划分，前端的知识在总体上分成基础部分和实践部分，基础部分包含了 JavaScript 语言（模块一）、CSS 和 HTML（模块二）以及浏览器的实现原理和 API（模块三），这三个模块涵盖了一个前端工程师所需要掌握的全部知识。

学完这三个部分，你再结合基本的编程能力，就可以应对基本的前端开发工作了。实践部分（模块四）重点会介绍我在工作过程中遇到的问题和解决方案，希望这块内容能够帮助你和你的前端团队找到可能的发展方向 and 着力点。

JavaScript



上面是我整理的 JavaScript 知识架构图，下面我们来具体解释一下。

在 JavaScript 的模块中，首先我们可以把语言按照文法、语义和运行时来拆分，这符合编程语言的一般规律：**用一定的词法和语法，表达一定语义，从而操作运行时。**

接下来，我们又按照程序的一般规律，把运行时分为数据结构和算法部分：数据结构包含类型和实例（JavaScript 的类型系统就是它的 7 种基本类型和 7 种语言类型，实例就是它的内置对象部分）。所谓的算法，就是 JavaScript 的执行过程。

类型部分中，对象比其它所有类型加起来都要更为复杂，所以我们会用较长的篇幅来讲解对象，包括它的一些历史和设计思路。

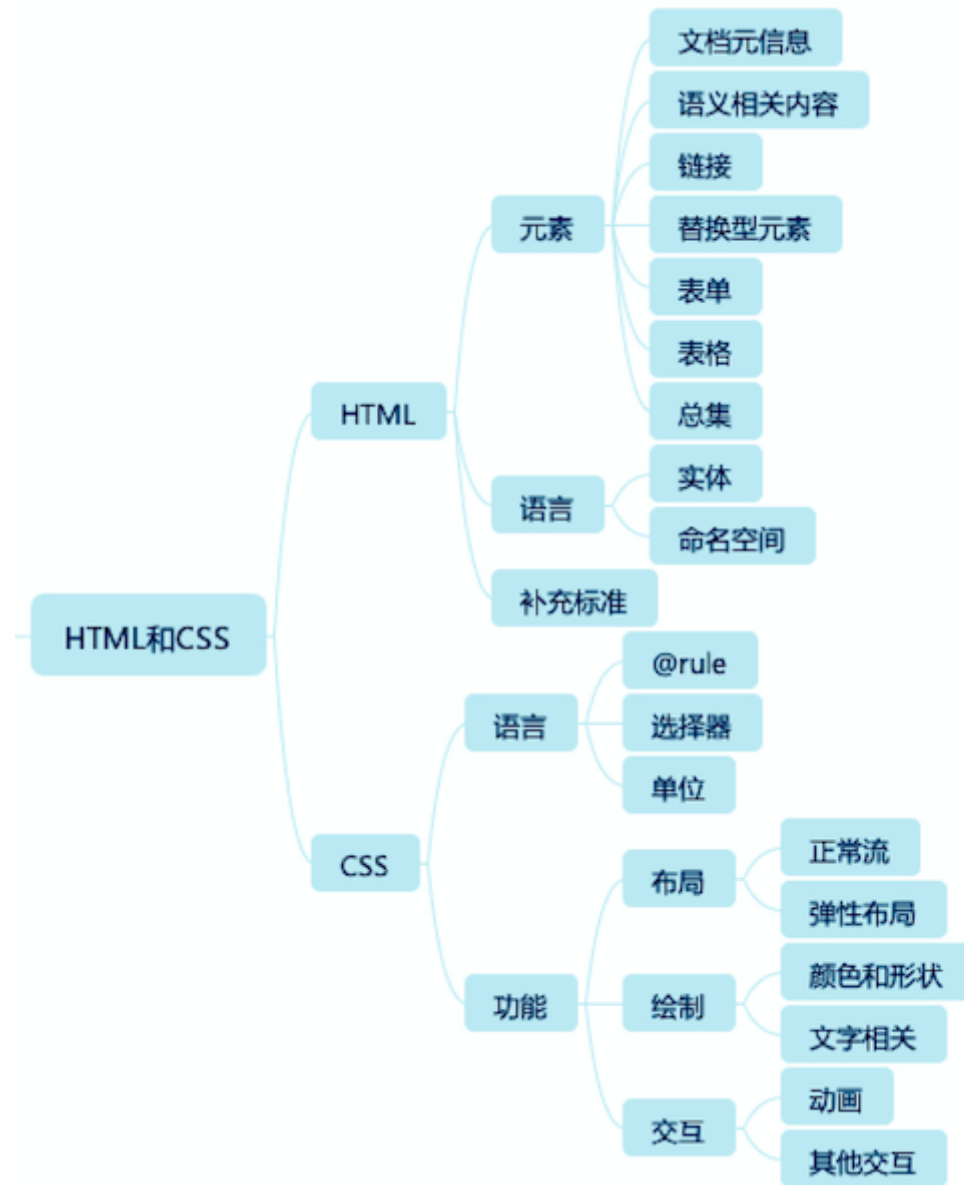
执行过程我们则需要按照从大结构到小结构的角度讲解，从最顶层的程序与模块、事件循环和微任务，到函数、再到语句级的执行。我们从粗到细地了解执行过程。

实例部分，对 JavaScript 来说类似基础库，JavaScript 的内置对象多达 150 以上，考虑到我们即使逐次讲解也必定不如 MDN 更加细致全面，所以我们会从应用和机制的角度，挑选其中几个体系来讲解。

文法中的语法和语义基本是一一对应关系，在 JavaScript 标准中有一份语法定义表，它同样不适合一一讲解，我们会从 JavaScript 语法中特别的地方，以及与日常开发比较相关的地方来重点讲解，剩下的内容和词法部分，我们会带领大家做一些数据挖掘工作，从这份表格中找到一些和我们日常开发息息相关的内容。

语义的大部分内容我们会在运行时的讲解中透出，同时它又跟语法有对应的关系，所以我们不再单独拿出来讲解。

HTML 和 CSS



上面是我整理的 HTML 和 CSS 的知识架构图，我们来具体解释一下。

在 **HTML 的部分**，我们会按照功能和语言来划分它的知识，HTML 的功能主要由标签来承担，所以我们首先会把标签做一些分类，并对它们分别进行讲解。

我们都知道 HTML 的标签可以分为很多种，head 里面的我们称为元信息类标签，诸如 title、meta、style、link、base 这些，它们用来描述文档的一些基本信息。还有一类是一些诸如 section、nav 的标签，它们在视觉表现上跟 div 并没有区别，但是各有各的适用场景，我们把它们称作语义类标签。另外一类是 img、video、audio 之类的替换型媒体类标签，用来引入外部内容，平常开发中你也会经常用到。再有就是表单类的，比如 input、button。

所以，基于这样的分类，我把标签分成下面几种。

1. 文档元信息：通常是出现在 head 标签中的元素，包含了描述文档自身的一些信息；
2. 语义相关：扩展了纯文本，表达文章结构、不同语言要素的标签；
3. 链接：提供到文档内和文档外的链接；
4. 替换型标签：引入声音、图片、视频等外部元素替换自身的一类标签；
5. 表单：用于填写和提交信息的一类标签；
6. 表格：表头、表尾、单元格等表格的结构。

我们的重点会放在前四种标签上，表单和表格较少用到，而且基本以查阅型知识为主，这里就不拿出来讲解了。

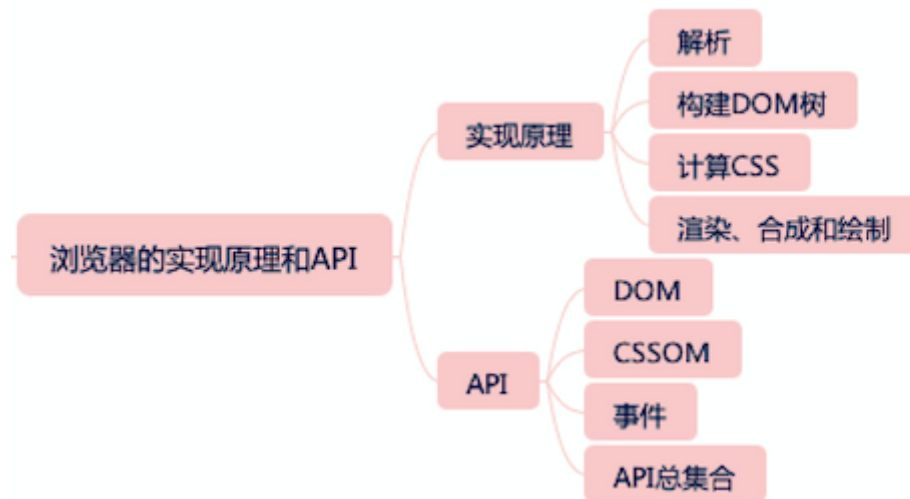
除了标签之外，我们还应该把 HTML 当作一门语言来了解下，当然，标记语言跟编程语言不太一样，没有编程语言那么严谨，所以，我们会简要介绍 HTML 的语法和几个重要的语言机制：实体、命名空间。

最后我们会介绍下 HTML 的补充标准：ARIA，它是 HTML 的扩展，在可访问性领域，它有至关重要的作用。

CSS 部分，按照惯例，我们也会从语言和功能两个角度去介绍。在语言部分，我们会从大到小介绍 CSS 的各种语法结构，比如 @rule、选择器、单位等等。功能部分，我们大致可以分为布局、绘制和交互类。

在布局类我们介绍两个最常用的布局：正常流和弹性布局。绘制类我们则会分成图形相关的和文字相关的绘制。最后我们会介绍动画和其它交互。

浏览器的实现原理和 API



上面是我整理的浏览器知识架构图，我们来具体看一下。

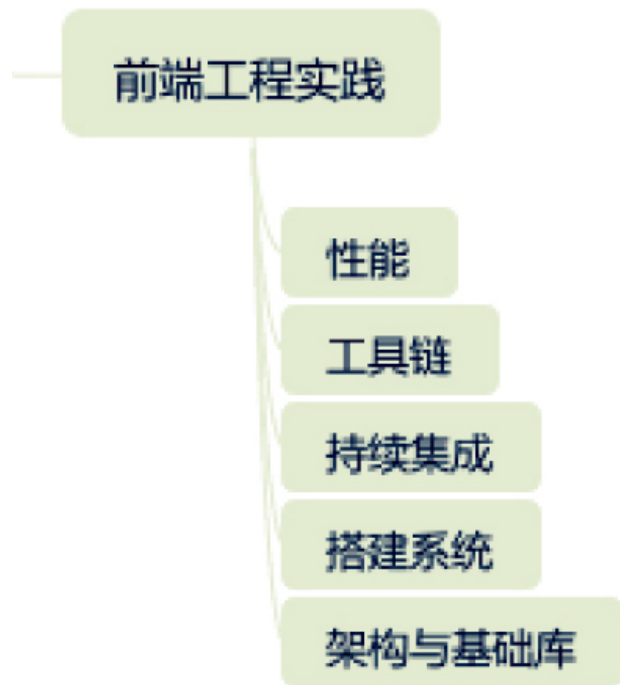
浏览器部分我们会先介绍下浏览器的实现原理，这是我们深入理解 API 的基础。

我们会从一般的浏览器设计出发，按照解析、构建 DOM 树、计算 CSS、渲染、合成和绘制的流程来讲解浏览器的工作原理。

在 API 部分，我们会从 W3C 零散的标准中挑选几个大块的 API 来详细讲解，主要有：事件、DOM、CSSOM 几个部分，它们分别覆盖了交互、语义和可见效果，这是我们工作中用到的主要内容。

其他的 API 怎么办呢，别着急，在最后，我会给出一份 Chrome 已经实现的 API 跟 W3C 标准的对应关系和它的生成过程，来覆盖其它部分。

前端工程实践



最后一个模块是前端工程实践。我们在掌握了前面的基础知识之后，也就基本掌握了做一个前端工程师的底层能力。在这个模块中，我选择了性能、工具链、持续集成、搭建系统、架构与基础库这几个方向的前端工程实践案例，来与你一起分享我的经验。

性能

首先我们会谈谈性能。对任何一个前端团队而言，性能是它价值的核心指标，从早年“重构”的实践开始，前端有通过性能证明自己价值的传统。

但是性能并非细节的堆砌，也不是默默做优化，所以，我会从团队的角度来跟你一起探讨性能的方法论和技术体系。

工具链

下一个案例是工具链。这一部分，我将会探讨企业中工具链的建设思路。对一个高效又合作良好的前端团队来说，一致性的工具链是不可或缺的保障，作为开发阶段的入口，工具链又可以和性能、发布、持续集成等系统链接到一起，成为团队技术管理的基础。

持续集成

接下来还会给大家介绍前端的持续集成，持续集成并非一个新概念，但是过去持续集成概念和理论都主要针对软件开发，而对前端来说，持续集成是一个新的课题（当然对持续集成来说，前端也是一个新课题），比如 daily build 就完全不适用前端，前端代码必须是线上实时可用的。这一部分内容将会针对前端的持续集成提出一些建设的思路。

搭建系统

接下来的案例是搭建系统，前端工作往往多而繁杂，针对高重复性、可模块化的业务需求，传统的人工开发不再适用，搭建系统是大部分大型前端团队的选择。这一部分内容我将会介绍什么是搭建系统，以及一些常见的搭建系统类型。

架构与基础库

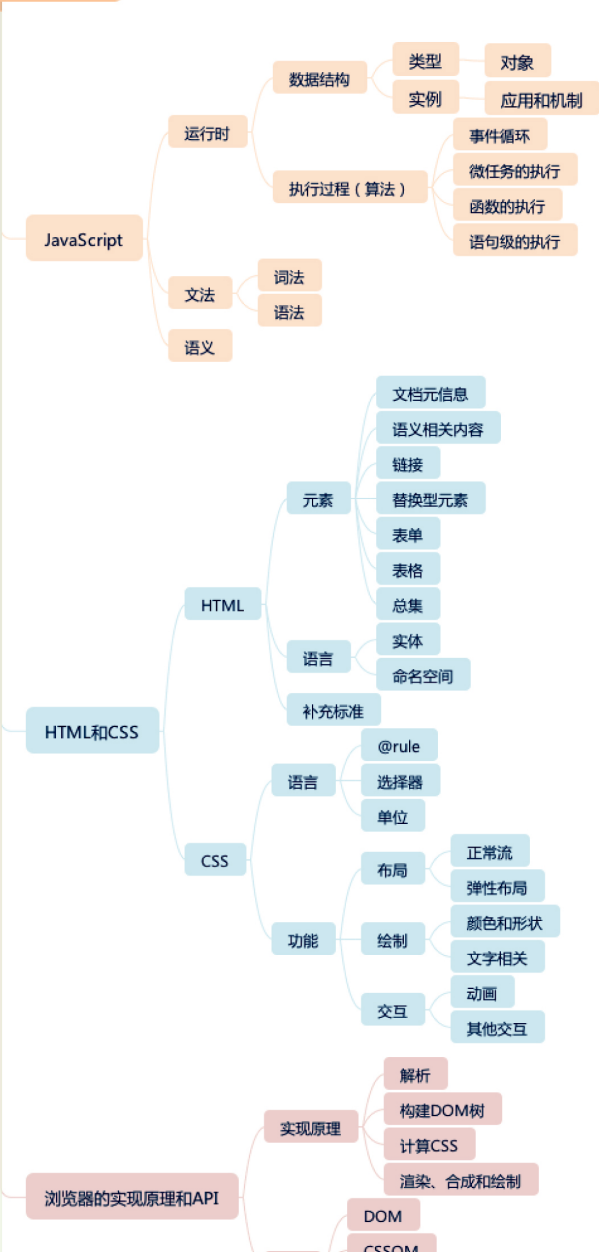
最后一个部分，会给大家介绍前端架构和基础库的知识。软件架构师主要解决功能复杂性的问题，服务端架构师主要解决高流量问题，而前端是页面间天然解耦，分散在用户端运行的系统，但是前端架构也有自己要解决的问题。

前端需求量大、专业人才稀缺，更因为前端本身运行在浏览器中，有大量兼容工作要做。所以前端架构的主要职责是兼容性、复用和能力扩展。这一部分文章我将会介绍前端架构工作的一些思路和切入点。

上面的这些案例来自我在领导手淘前端团队时的经验，和我在阿里巴巴工作参与晋升面试时听到的案例，这些内容几乎是每一个年轻的前端团队成长过程中都会需要的基础设施。

好了，前端的知识体系我们大致列出来了。你可能发现了，知识体系图中的每一个知识点，专栏里都有与之对应的文章，这也是我的初衷：希望借由讲解这 40 余个知识点，帮你建立起前端的知识框架。

前端知识框架图



《重学前端》专栏目录

- ① 开篇词
- ② 明确你的前端学习路线
- ③ 一起列一份前端知识架构图

模块一 · JavaScript

- ① 类型：关于类型，有哪些你忽视的细节？
- ② 对象：JavaScript是面向对象还是基于对象的？
- ③ 对象：JavaScript中，我们真的需要模拟类吗？
- ④ 函数：this有什么用？应该怎么用？
- ⑤ 函数：JavaScript里的闭包到底要怎么用？适合用在哪里？
- ⑥ 事件循环与微任务：Promise里的代码为什么比setTimeout先执行？
- ⑦ 语句：try里面放return，finally还会执行吗？
- ⑧ 词法：为什么 1.toString 会报错？
- ⑨ 运算符与表达式：新加入的**运算符，好像有哪里不一样？
- ⑩ 语句、程序与模块：在script标签里写 export 为什么会报错？
- ⑪ 分号自动补全：到底要不要写分号呢？

模块二 · HTML和CSS

HTML：

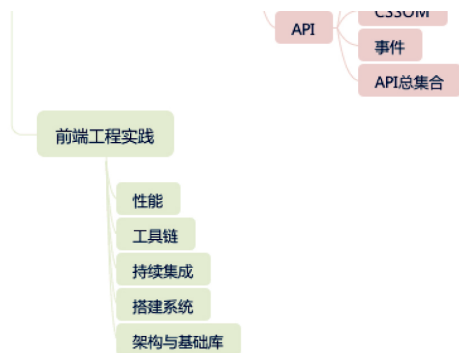
- ① HTML语义：只用div和span是不是就够了？
- ② HTML语义：一篇wiki文章究竟会用到哪些语义元素？
- ③ 文档元信息：你知道head里一共能写哪几种标签吗？
- ④ 链接：除了a标签，还有标签叫链接？
- ⑤ 嵌入型元素：怎么link一个css要用href，而引入js要用src呢？
- ⑥ HTML标签：标签总结
- ⑦ HTML语言：dtd到底是什么？
- ⑧ ARIA：可访问性不止是给盲人用户的特性

CSS：

- ① 媒体、分页、视口：除了属性和选择器，你还要知道这些带@的规则
- ② 选择器：为什么只有子元素选择器，没有父元素选择器
- ③ 布局正常流：其实从毕升开始，我们就在用正常流了
- ④ 布局弹性盒：垂直居中为什么这么难？
- ⑤ 文本：baseline到底是啥东西？
- ⑥ 颜色与形状：CSS是如何绘制颜色与形状的？
- ⑦ 动画与交互：为什么动画要用贝塞尔曲线这么奇怪的东西？
- ⑧ CSS属性：CSS总结

模块三 · 浏览器实现原理与API

- ① 浏览器的工作原理：你最熟悉的浏览器，是怎么写出来的？（上）
- ② 浏览器的工作原理：你最熟悉的浏览器，是怎么写出来的？（中）
- ③ 浏览器的工作原理：你最熟悉的浏览器，是怎么写出来的？（下）
- ④ DOM：如何用JS操作文档？



- ⑤ CSSOM：如何获取一个元素的准确位置？
- ⑥ 事件：捕获模型和冒泡模型到底是怎么来的？
- ⑦ 其它一些API：API总集合

模块四·前端综合应用

- ① 性能：前端性能到底对业务数据有多大的影响？
- ② 工具链：什么样的工具链才能提升团队效率？
- ③ 持续集成：几十个前端一起工作，怎么样保证基本质量？
- ④ 搭建系统：大量的低价值需求如何应对？
- ⑤ 架构与基础库： 如何设计基础库？
- ⑥ 答疑文章

(长按保存高清图)

讲述形式

基于这份知识框架图，我们的课程主要采用两种讲述形式：一种是重点讲解的课程，一种是知识图谱型的课程。

重点讲解的课程我们会从技术的背景、原理和设计出发，把知识的内容呈现出来。这种形式适用于有体系和源流的知识，比较适合系统学习和理解，比如 JavaScript 中的对象、CSS 的排版。

知识图谱型的课程则提供一些方法，用表格或者脑图的形式来整理知识的结构。这种形式适用于零散的知识，比较适合记住大概，用到时去查阅，比如 JavaScript 的词法、HTML 中的所有标签、以及浏览器中的 API 就十分适合这样的讲解方式。

结语

今天我带你一起划分了前端的知识内容，前端的基础知识分成 JavaScript、HTML、CSS 以及浏览器四大重点模块，每个模块也分别有自己的技术重点。你可以在框架中，挑选你最需要的前端知识，按需学习。

当然，这篇文章最重要的是，我希望能帮你建立一个理解前端的全景图。这样，任何时候，你都能够体系地思考问题，分析问题。

你觉得你的划分跟我一样吗，你还有其他想法，你觉得是否有想了解的知识不在其中，欢迎给我留言。

附录：知识点发布顺序

知识点发布顺序

HTML · 语义

JavaScript · 类型

JavaScript · 对象

CSS · 媒体、分页、视口

浏览器 · 工作原理

HTML · 文档元信息

JavaScript · 函数

JavaScript · 事件循环与微任务

CSS · 选择器

浏览器 · DOM

HTML · 链接

JavaScript · 语句

JavaScript · 语法

JavaScript · 语法

CSS · 布局正常流

浏览器 · CSSOM

HTML · 嵌入型元素

JavaScript · 运算符表达式

JavaScript · 语句、程序与模块

CSS · 布局弹性盒

浏览器 · 事件

HTML · 标签

JavaScript · 分号自动补全

CSS · 文本

浏览器 · API

HTML · 语言

HTML · ARIA

CSS · 颜色与形状

CSS · 动画与交互





重学前端

每天 10 分钟，重构你的前端知识体系

winter 程劭非
前手机淘宝前端负责人



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，未经许可不得转载



bd2star