

## 87 | 程序员练级攻略：前端基础和底层原理

2018-07-31 陈皓

左耳听风

[进入课程 >](#)



讲述：柴巍

时长 14:45 大小 6.76M



对于前端的学习和提高，我的基本思路是这样的。首先，前端的三个最基本的东西 HTML 5、CSS 3 和 JavaScript ( ES6 ) 是必须要学好的。这其中有很多很多的技术，比如，CSS 3 引申出来的 Canvas ( 位图 )、SVG ( 矢量图 ) 和 WebGL ( 3D 图 )，以及 CSS 的各种图形变换可以让你做出非常丰富的渲染效果和动画效果。

ES6 简直就是把 JavaScript 带到了一个新的台阶，JavaScript 语言的强大，大大释放了前端开发人员的生产力，让前端得以开发更为复杂的代码和程序，于是像 React 和 Vue 这样的框架开始成为前端编程的不二之选。

我一直认为学习任何知识都要从基础出发，所以这篇文章我会着重介绍基础知识和基本原理，尤其是如下的这些知识，都是前端程序员需要花力气啃下来的硬骨头。

**JavaScript 的核心原理。**这里我会给出好些网上很不错的讲 JavaScript 的原理的文章或图书，你一定要学好语言的特性，并且详细了解其中的各种坑。

**浏览器的工作原理。**这也是一块硬骨头，我觉得这是前端程序员需要了解和明白的关键知识点，不然，你将无法深入下去。

**网络协议 HTTP。**也是要着重了解的，尤其是 HTTP/2，还有 HTTP 的几种请求方式：短连接、长连接、Stream 连接、WebSocket 连接。

**前端性能调优。**有了以上的这些基础后，你就可以进入前端性能调优的主题了，我相信你可以很容易上手各种性能调优技术的。

**框架学习。**我只给了 React 和 Vue 两个框架。就这两个框架来说，Virtual DOM 技术是其底层技术，组件化是其思想，管理组件的状态是其重点。而对于 React 来说，函数式编程又是其编程思想，所以，这些基础技术都是你需要好好研究和学习的。

**UI 设计。**设计也是前端需要做的一个事，比如像 Google 的 Material UI，或是比较流行的 Atomic Design 等应该是前端工程师需要学习的。

而对于工具类的东西，这里我基本没怎么涉及，因为本文主要还是从原理和基础入手。那些工具我觉得都很简单，就像学习 Java 我没有让你去学习 Maven 一样，因为只要你去动手了，这种知识你自然就会获得，我们还是把精力重点放在更重要的地方。

下面我们从前端基础和底层原理开始讲起。先来讲讲 HTML5 相关的内容。

## HTML 5

HTML 5 主要有以下几本书推荐。

[HTML 5 权威指南](#)，本书面向初学者和中等水平 Web 开发人员，是牢固掌握 HTML 5、CSS 3 和 JavaScript 的必读之作。书看起来比较厚，是因为里面的代码很多。

[HTML 5 Canvas 核心技术](#)，如果你要做 HTML 5 游戏的话，这本书必读。

对于 SVG、Canvas 和 WebGL 这三个对应于矢量图、位图和 3D 图的渲染来说，给前端开发带来了重武器，很多 HTML5 小游戏也因此蓬勃发展。所以，你可以学习一下。

学习这三个技术，我个人觉得最好的地方是 MDN。

[SVG: Scalable Vector Graphics](#)

[Canvas API](#)

[The WebGL API: 2D and 3D graphics for the web](#)

最后几个资源列表。

[Awesome HTML5](#)。GitHub 上的 Awesome HTML5，其中有大量的资源和技术文章。

[Awesome SVG](#)

[Awesome Canvas](#)

[Awesome WebGL](#)

## CSS

在《程序员练级攻略》系列文章最开始，我们就推荐过 CSS 的在线学习文档，这里再推荐一下 [MDN Web Doc - CSS](#)。我个人觉得只要你仔细读一下文档，CSS 并不难学。绝大多数觉得难的，一方面是文档没读透，另一方面是浏览器支持的标准不一致。所以，学好 CSS 最关键的还是要仔细地读文档。

之后，在写 CSS 的时候，你会发现，你的 CSS 中有很多看起来相似的东西。你的 DRY - Don't Repeat Yourself 洁癖告诉你，这是不对的。所以，你需要学会使用 [LESS](#) 和 [SaSS](#) 这两个 CSS 预处理工具，其可以帮你提高很多效率。

然后，你需要学习一下 CSS 的书写规范，前面的《程序员修养》一文中提到过一些，这里再补充几个。

[Principles of writing consistent, idiomatic CSS](#)

[Opinionated CSS styleguide for scalable applications](#)

[Google HTML/CSS Style Guide](#)

如果你需要更有效率，那么你还需要使用一些 CSS Framework，其中最著名的就是 Twitter 公司的 [Bootstrap](#)，其有很多不错的 UI 组件，页面布局方案，可以让你非常方便也非常快速地开发页面。除此之外，还有，主打清新 UI 的 [Semantic UI](#)、主打响应式界面的 [Foundation](#) 和基于 Flexbox 的 [Bulma](#)。

当然，在使用 CSS 之前，你需要把你浏览器中的一些 HTML 标签给标准化掉。所以，推荐几个 Reset 或标准化的 CSS 库：[Normalize](#)、[MiniRest.css](#)、[sanitize.css](#) 和 [unstyle.css](#)。

关于更多的 CSS 框架，你可以参看[Awesome CSS Frameworks](#) 上的列表。

接下来，是几个公司的 CSS 相关实践，供你参考。

[CodePen' s CSS](#)

[Github 的 CSS](#)

[Medium' s CSS is actually pretty f\\*\\*\\*ing good](#)

[CSS at BBC Sport](#)

[Refining The Way We Structure Our CSS At Trello](#)

最后是一个可以写出可扩展的 CSS 的阅读列表 [A Scalable CSS Reading List](#)。

## JavaScript

下面是学习 JavaScript 的一些图书和文章。

[JavaScript: The Good Parts](#)，中文翻译版为《JavaScript 语言精粹》。这是一本介绍 JavaScript 语言本质的权威图书，值得任何正在或准备从事 JavaScript 开发的人阅读，并且需要反复阅读。学习、理解、实践大师的思想，我们才可能站在巨人的肩上，才有机会超越大师，这本书就是开始。

[Secrets of the JavaScript Ninja](#)，中文翻译版为《JavaScript 忍者秘籍》，本书是 jQuery 库创始人编写的一本深入剖析 JavaScript 语言的书。适合具备一定 JavaScript 基础知识的读者阅读，也适合从事程序设计工作并想要深入探索 JavaScript 语言的读者阅读。这本书有很多晦涩难懂的地方，需要仔细阅读，反复琢磨。

[Effective JavaScript](#)，Ecma 的 JavaScript 标准化委员会著名专家撰写，作者凭借多年标准化委员会工作和实践经验，深刻辨析 JavaScript 的内部运作机制、特性、陷阱和编程最佳实践，将它们高度浓缩为极具实践指导意义的 68 条精华建议。

接下来是 ES6 的学习，这里给三个学习手册源。

[ES6 in Depth](#) , InfoQ 上有相关的中文版 - [ES6 深入浅出](#)。还可以看看 [A simple interactive ES6 Feature list](#) , 或是看一下 [阮一峰翻译的 ES6 的教程](#) 。

[ECMAScript 6 Tools](#) , 这是一堆 ES6 工具的列表, 可以帮助你提高开发效率。

[Modern JS Cheatsheet](#) , 这个 Cheatsheet 在 GitHub 上有 1 万 6 千颗星, 你就可见其影响力了。

然后, 还有一组很不错的《[You Don' t Know JS 系列](#)》的书。

[You Don' t Know JS: "Up & Going"](#)

[You Don' t Know JS: "Scope & Closures"](#)

[You Don' t Know JS: "this & Object Prototypes"](#)

[You Don' t Know JS: "Types & Grammar"](#)

[You Don' t Know JS: "Async & Performance"](#)

[You Don' t Know JS: "ES6 & Beyond"](#)

接下来是一些和编程范式相关的文章。

[Glossary of Modern JavaScript Concepts: Part 1](#) , 首先推荐这篇文章, 其中收集了一些编程范式方面的内容, 比如纯函数、状态、可变性和不可变性、指令型语言和声明式语言、函数式编程、响应式编程、函数式响应编程。

[Glossary of Modern JavaScript Concepts: Part 2](#) , 在第二部分中主要讨论了作用域和闭包, 数据流, 变更检测, 组件化.....

下面三篇文章是德米特里·索什尼科夫 ( Dmitry Soshnikov ) 个人网站上三篇讲 JavaScript 内在的文章。

[JavaScript. The Core: 2nd Edition](#)

[JavaScript. The Core \(older ES3 version\)](#)

[JS scope: static, dynamic, and runtime-augmented](#)

**"How JavaScript Works"** 是一组非常不错的文章 ( 可能还没有写完 ) , 强烈推荐。这一系列的文章是 SessionStake 的 CEO 写的, 现在有 13 篇, 我感觉可能还没有写完。这个叫 [亚历山大·兹拉特科夫 \( Alexander Zlatkov \)](#) 的 CEO 太猛了。

[An overview of the engine, the runtime, and the call stack](#)

[Inside the V8 engine + 5 tips on how to write optimized code](#) , 了解 V8 引擎。  
这里, 也推荐 [Understanding V8' s Bytecode](#) 这篇文章可以让你了解 V8 引擎的底层字节码。

[Memory management + how to handle 4 common memory leaks](#) , 内存管理和 4 种常见的内存泄露问题。

[Event loop and the rise of Async programming + 5 ways to better coding with async/await](#) , Event Loop 和异步编程。

[Deep dive into WebSockets and HTTP/2 with SSE + how to pick the right path](#) , WebSocket 和 HTTP/2。

[A comparison with WebAssembly + why in certain cases it' s better to use it over JavaScript](#) , JavaScript 内在原理。

[The building blocks of Web Workers + 5 cases when you should use them](#) , Web Workers 技术。

[Service Workers, their lifecycle and use cases](#) , Service Worker 技术。

[The mechanics of Web Push Notifications](#) , Web 端 Push 通知技术。

[Tracking changes in the DOM using MutationObserver](#) , Mutation Observer 技术。

[The rendering engine and tips to optimize its performance](#) , 渲染引擎和性能优化。

[Inside the Networking Layer + How to Optimize Its Performance and Security](#) , 网络性能和安全相关。

[Under the hood of CSS and JS animations + how to optimize their performance](#) , CSS 和 JavaScript 动画性能优化。

接下来是 Google Chrome 工程经理 [阿迪·奥斯马尼 \( Addy Osmani \)](#) 的几篇 JavaScript 性能相关的文章, 也是非常好的。

[The Cost Of JavaScript](#)

[JavaScript Start-up Performance](#)

其它与 JavaScript 相关的资源。



[JavaScript has Unicode Problem](#)，这是一篇很有价值的 JavaScript 处理 Unicode 的文章。

[JavaScript Algorithms](#)，用 JavaScript 实现的各种基础算法库。

[JavaScript 30 秒代码](#)，一堆你可以在 30 秒内看懂各种有用的 JavaScript 的代码，在 GitHub 上有 2 万颗星了。

[What the f\\*ck JavaScript](#)，一堆 JavaScript 搞笑和比较 tricky 的样例。

[Airbnb JavaScript Style Guide](#)，Airbnb 的 JavaScript 的代码规范，GitHub 上有 7 万多颗星。

[JavaScript Patterns for 2017](#)，YouTube 上的一个 JavaScript 模式分享，值得一看。

## 浏览器原理

你需要了解一下浏览器是怎么工作的，所以，你必需要看《[How browsers work](#)》。这篇文章受众之大，后来被人重新整理并发布为《[How Browsers Work: Behind the scenes of modern web browsers](#)》，其中还包括中文版。这篇文章非常非常长，所以，你要有耐心看完。如果你想看个精简版的，可以看我在 Coolshell 上发的《[浏览器的渲染原理简介](#)》或是看一下[这个幻灯片](#)。

然后，是对 Virtual DOM 的学习。Virtual DOM 是 React 的一个非常核心的技术细节，它也是前端渲染和性能的关键技术。所以，你有必要要好好学习一下这个技术的实现原理和算法。当然，前提条件是你需要学习过前面我所推荐过的浏览器的工作原理。下面是一些不错的文章可以帮你学习这一技术。

[How to write your own Virtual DOM](#)

[Write your Virtual DOM 2: Props & Events](#)

[How Virtual-DOM and diffing works in React](#)

[The Inner Workings Of Virtual DOM](#)

[深度剖析：如何实现一个 Virtual DOM 算法](#)

以及两个 Virtual-DOM 实现供你参考：

[Matt-Esch/Virtual-DOM](#)

[Maquette](#)

# 网络协议

[High Performance Browser Networking](#)，本书是谷歌公司高性能团队核心成员的权威之作，堪称实战经验与规范解读完美结合的产物。本书目标是涵盖 Web 开发者技术体系中应该掌握的所有网络及性能优化知识。

全书以性能优化为主线，从 TCP、UDP 和 TLS 协议讲起，解释了如何针对这几种协议和基础设施来优化应用。然后深入探讨了无线和移动网络的工作机制。最后，揭示了 HTTP 协议的底层细节，同时详细介绍了 HTTP 2.0、XHR、SSE、WebSocket、WebRTC 和 DataChannel 等现代浏览器新增的能力。

另外，[HTTP/2](#)也是 HTTP 的一个新的协议，于 2015 年被批准通过，现在基本上所有的主流浏览器都默认启用这个协议。所以，你有必要学习一下这个协议。下面相关的学习资源。

[Gitbook - HTTP/2 详解](#)

[http2 explained](#) ( [中译版](#) )

[HTTP/2 for a Faster Web](#)

[Nginx HTTP/2 白皮书](#)

HTTP/2 的两个 RFC：

[RFC 7540 - Hypertext Transfer Protocol Version 2 \(HTTP/2\)](#)，HTTP/2 的协议本身。

[RFC 7541 - HPACK: Header Compression for HTTP/2](#)，HTTP/2 的压缩算法。

新的 HTML5 支持 [WebSocket](#)，所以，这也是你要学的一个重要协议。

[HTML5 WebSocket: A Quantum Leap in Scalability for the Web](#)，这篇文章比较了 HTTP 的几种链接方式，Polling、Long Polling 和 Streaming，并引入了终极解决方案 WebSocket。你知道的，了解一个技术的缘由是非常重要的。

[StackOverflow: My Understanding of HTTP Polling, Long Polling, HTTP Streaming and WebSockets](#)，这是 StackOverflow 上的一个 HTTP 各种链接方式的比较，也可以让你有所认识。

[An introduction to Websockets](#)，一个 WebSocket 的简单教程。

[Awesome Websockets](#)，GitHub 的 Awesome 资源列表。

一些和 WebSocket 相关的想法，可以开阔你的思路：

[Introducing WebSockets: Bringing Sockets to the Web](#)



[Websockets 101](#)

[Real-Time Web by Paul Banks](#)

[Are WebSockets the future?](#)

## 小结

总结一下今天的内容。我一直认为学习任何知识都要从基础出发，所以今天我主要讲述了 HTML 5、CSS 3 和 JavaScript ( ES6 ) 这三大基础核心，给出了大量的图书、文章以及其他一些相关的学习资源。之后，我建议你学习浏览器的工作原理和网络协议相关的内容。我认为，掌握这些原理也是学好前端知识的前提和基础。值得花时间，好好学习消化。

下篇文章中，我们将讲讲如何做前端性能优化，并推荐一些好用的前端框架。敬请期待。

下面是《程序员练级攻略》系列文章的目录。

[开篇词](#)

入门篇

[零基础启蒙](#)

[正式入门](#)

修养篇

[程序员修养](#)

专业基础篇

[编程语言](#)

[理论学科](#)

[系统知识](#)

软件设计篇

[软件设计](#)

高手成长篇

[Linux 系统、内存和网络（系统底层知识）](#)

[异步 I/O 模型和 Lock-Free 编程（系统底层知识）](#)

[Java 底层知识](#)

[数据库](#)

[分布式架构入门（分布式架构）](#)

[分布式架构经典图书和论文（分布式架构）](#)

[分布式架构工程设计（分布式架构）](#)

[微服务](#)

[容器化和自动化运维](#)

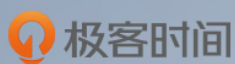
[机器学习和人工智能](#)

[前端基础和底层原理（前端方向）](#)

[前端性能优化和框架（前端方向）](#)

[UI/UX 设计（前端方向）](#)

[技术资源集散地](#)



# 左耳朵耗子

全年独家专栏《左耳听风》

20000 名程序员的练级攻略

陈皓 资深技术专家  
骨灰级程序员



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。

## 精选留言 (28)

写留言



小薛薛

2018-07-31

15

0.0 从零基础入门看到现在，这知识体量，好害怕。

展开

作者回复: 我用了20年，我能做到，你一定也能



少年姜太公

2018-07-31

6

21世纪信息获取越来越便利了，网络上有各种书籍，博客，论文，有中文也有外文的。古代，资源匮乏，为了读一本书可能要跑很远借阅并手抄，而现代我们是生活在资源的海洋中，获取资源效率提高了。然而不幸的是在资源的海洋中，我们迷茫了，难以取舍。如何获取优良的资源并迅速掌握吸收转化为生产力已是竞争的核心。望陈老师加以开示。

展开

作者回复: 我没什么好的经验，我也是只有在看过所有的资料后才知道什么是好的。比较幸运的是，我的基础知识很扎实，所以学什么都很快。



图·美克尔

2018-07-31

4

这系列真的看不动了...

展开

作者回复: 有时候，学习就像拉弓蓄力一样，学习基础知识感觉很枯燥很不实用，工作上用不到，然而学习这些知识是为了未来可以学得更快。基础打牢，学什么都快，别看我的专栏后面的那些文章很多，打好基础，那些就跟切菜一样。



寻路之人

2018-07-31

👍 4

耗哥，大概什么时候讲解一下，如何阅读来源项目源码的经验分享吗？期待中



Geek\_1af8d...

2018-08-02

👍 2

没想到看到这个领域，耗子叔还是能带来惊喜

展开 ▾



小柒

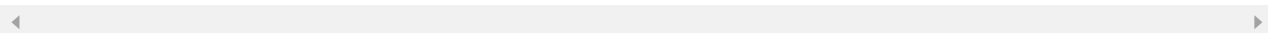
2018-07-31

👍 2

终于等到前端了，react和vue学习资源好少啊

展开 ▾

作者回复: 还没到呢



邱桂新

2018-10-04

👍 1

您用了二十年，谁能像您一样花二十年学这些？(纯疑问)



秋天

2018-08-22

👍 1

老师这是把收藏夹里的东西都倾倒出来啦？？？？

展开 ▾



Geek\_1af8d...

2018-08-02

👍 1

没想到看到这个领域，耗子叔还是能带来惊喜

展开 ▾



朵朵李

2018-08-01

👍 1

以我自己的经验，CSS需要学习一些基础，例如布局，动画，选择器，适配等，剩下的主要是从优秀的框架例如bootstrap学习如何利用LESS,SASS等预处理器的变量，函数，mixin等，完成组件，页面到整站的模块化，体系化。当然跟JS类似，后处理器逐渐流行，所以POSTCSS也需要了解使用。JavaScript感觉主要是权威指南夯实基础，ES6 ES7等学习掌握，其他的书籍辅助，至少后处理器babel要会用，喜欢严格的就学习使用TS。其实...  
展开 ▾

---



\_俊腾

2018-07-31

👍 1

耗子叔，es6的学习推荐exploring es6，它有一个免费的在线版本  
<http://exploringjs.com/es6/index.html>

---



方式方法

2018-07-31

👍 1

20年的功力果然不一般啊!

展开 ▾

---



godtrue

2019-01-12

👍

学会九阳神功后学什么都快，少林龙抓手也能现学现用，这个系列的文章就是九阳神功。

---



王善鹏

2018-12-28

👍

耗子叔，我想知道我学完这些技术是不是也过时了，我6年前写过一段时间前端，那会还没有ES6，框架用的都是jquery，现在都是ES6，vue，react，bootstrap。感觉技术更新太快了。

---



是小小白啊

2018-09-19

👍

《你不知道的JavaScript》也很不错，对JavaScript的剖析也很深入，语言也比较风趣，很适合有一定基础的人读。

展开 ▾



**丑石**

2018-08-18



去看了一下Alexander Zlatkov在medium上的文章，果然篇篇精品。大牛！



**屈超**

2018-08-14



安卓学习路线可以参考这个:<https://github.com/JsonChao/Awesome-Android-Notebook>



**木易**

2018-08-13



大数据方面的会有吗？hadoop spark

展开 ▾



**付剑津**

2018-08-08



耗子叔，在学JavaScript，但有朋友说TypeScript更值得学，按照我的观念，先学好JavaScript肯定是没错的，但还是想来请教下耗子叔，嘿嘿

展开 ▾



**Grace微雪**

2018-08-04



要学的东西太多了...□

展开 ▾