





程序员练级攻略（2018）：前端基础和底层原理

陈皓

- 00:01 / 14:45

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HTML5、CSS3和JavaScript（ES6）是必须要学好的。这其中有很多很多的技术，比如，CSS3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，所以我会有很大的篇幅在讲各种技术的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一块一块啃掉的硬骨头。

- 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和其中的各种坑。
- 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东西，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。

HTML5
HTML5主要有以下几本书推荐。

- [HTML5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HTML5、CSS3和JavaScript的必读之作。书看起来比较厚，是因为里面的代码很多。
- [HTML5 Canvas核心技术](#)，如果你要做HTML5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（2018）》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阅读一下文档，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](#) 、主要响应式界面的 [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**king 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膀上，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, 你必要要看《[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HTML5、CSS3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（2018）》系列文章的目录（持续更新中）。

- [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](#)
 - [分布式架构工程设计](#)
 - [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
 - UI/UX设计
 -

左耳听风

洞悉技术的本质
享受科技的乐趣



极客时间
最好的知识，最好的成长

陈 皓

资深技术专家
骨灰级程序员


扫码订阅

陈敬秀	2018-07-31
耗子叔，如何去体味各种语言中的坑呢？ 作者回复	
参看《编程范式游记》	2018-07-31
小柒	2018-07-31
终于等到前端了，react和vue学习资源好少啊 作者回复	
还没到呢	2018-07-31
寻路之人	2018-07-31
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	
小薛薛	2018-07-31
0.0 从零基础入门看到现在，这知识体量，好害怕。	

作者回复

我用了20年，我能做到，你一定也能

2018-07-31

沫沫（美丽人生）

陈老师，早上好，我们团队现在正在做一个自动建站的项目，主要是广告的landing page和blog形式的，想请教一下，您有没有这方面的开源框架可以推荐，谢谢啦！

2018-07-31

