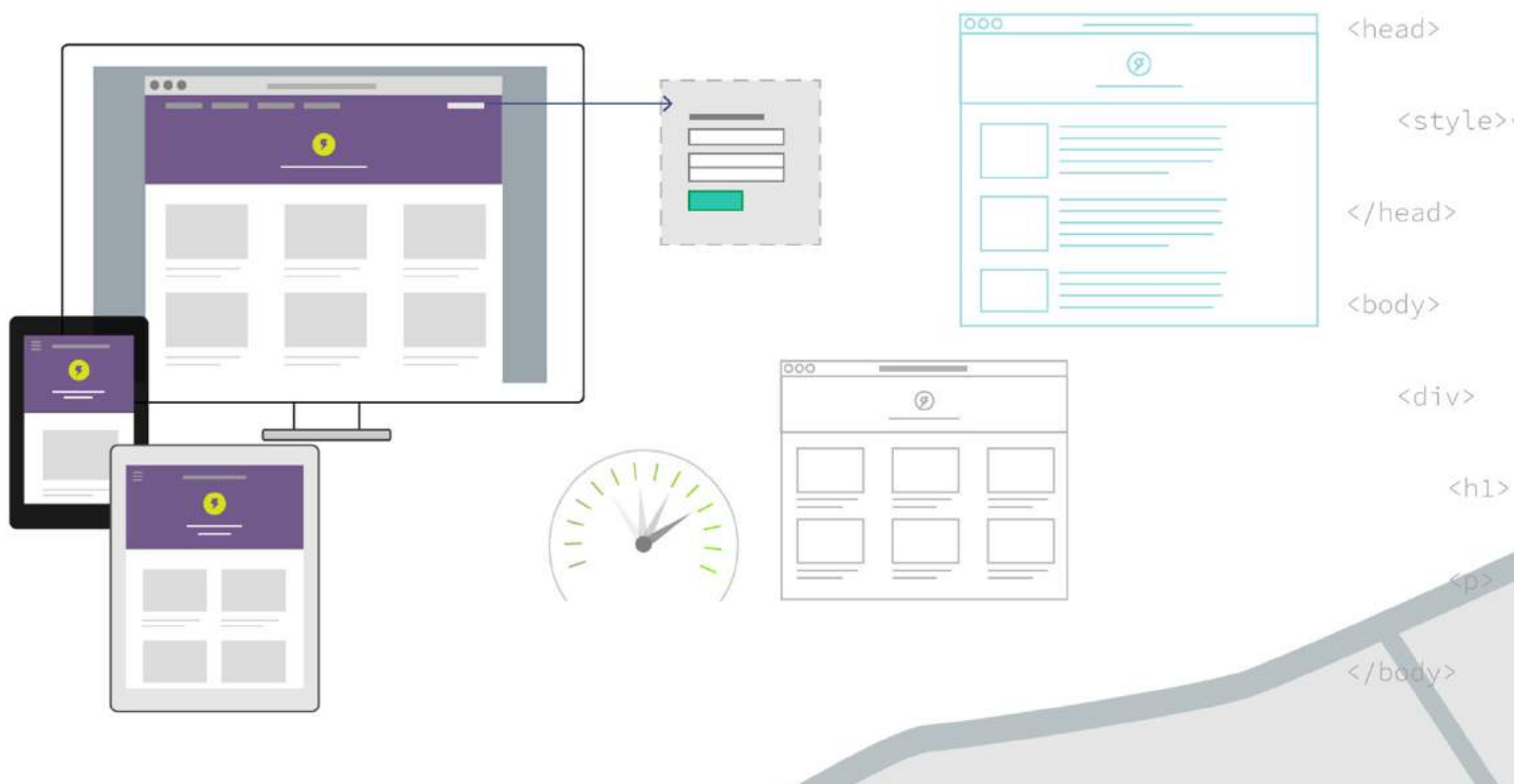


前端工程师技能清单

获得你的第一份前端开发工作



欢迎

感谢阅读前端开发工程师技能清单！

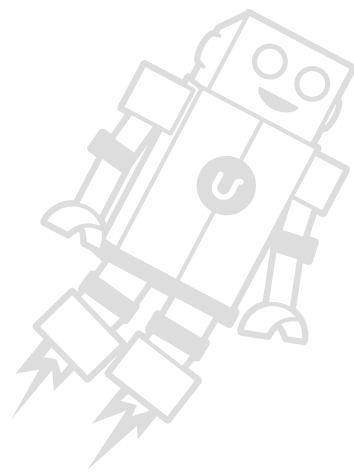
你即将踏上一条充满惊喜的职业道路。你会有很多机会让他人的生活变得更好，并拓展自己的创新能力。

拥有很多选择总是一件好事。但有时候，你还需要一个指导，所以在此，我们将帮助你清除干扰。

我们最近推出了《[前端工程师](#)》认证项目，该课程会以实战项目的形式指导学员学习相关技能，帮助你在 Web 开发领域获得自己的第一份工作。我们通过与行业领先企业沟通交流，收集了大量实用信息，确保我们在课程中教授最前沿的技能，我们迫不及待地想要向你展示这个技能列表。

在该指南中，你将看到成为前端开发工程师所需的终极技能检查清单，以及可以帮助你起步的一些资源。

恭喜！你踏出了你在 Web 开发者职业道路上的第一步！请阅读下文的前端开发终极技能清单和推荐的资源。



前端开发工程师技能清单

以下是成为前端工程师需要掌握的技能明细。请花些时间阅读该列表，你已经具备哪些技能？

要详细了解这些技能并获取学习资料，请点击前往相应页面。



<input type="checkbox"/> HTML	04
<input type="checkbox"/> CSS	05
<input type="checkbox"/> JavaScript	06
<input type="checkbox"/> 响应式网页设计	07
<input type="checkbox"/> CSS 框架	08
<input type="checkbox"/> JavaScript 框架	09
<input type="checkbox"/> 版本控制	10
<input type="checkbox"/> 网页性能	11
<input type="checkbox"/> 浏览器开发工具	12
<input type="checkbox"/> 构建和自动化工具	13
<input type="checkbox"/> 测试	14
<input type="checkbox"/> 软技能	15

学习资源

16

- 前端工程师认证项目
- 单项课程
- 网络开发资源和社区

HTML

编写网站需要用到以下三种语言：HTML、CSS 和 JavaScript，而 HTML 是首先需要学习的语言。HTML 本身并不是编程语言，它用来描述元素应该如何网站上布局，并向浏览器提供网站所需的其他所有文件列表（例如 CSS 和 JavaScript）。你可以将 HTML 看做盖房的图纸。它可以告诉你房间有多大，里面应该有什么东西，但是它不会告诉你外观如何。



- ☐ **语义元素**：语义元素明确地对浏览器和开发者描述其含义；元素包括文章和部分，而不是到处使用 `div`
- ☐ **块级元素**：块级元素占据了父元素的整个空间
- ☐ **内嵌元素**：内嵌元素仅占据由内嵌元素定义标记界定的空间
- ☐ **表格**：表格表示文档部分，其中包含向网络服务器提交信息的交互式控件
- ☐ **输入类型**：输入元素用于为网络表格创建交互式控件，以便接受用户输入的数据

CSS

如果说 HTML 描述的是房子的布局，那么 CSS 描述的就是房子的外观。层叠样式表（简称 CSS）负责控制网站的外观。颜色、字体，甚至一些动画都由 CSS 控制。和 HTML 一样，CSS 不是编程语言。它是一种文本文档，就像室内设计师的说明，使网站看起来很美观。



- **显示值类型**：显示属性使你能够控制图表或容器元素的渲染效果
- **盒模型**：盒模型负责定义矩形框（表示文档中的元素）的尺寸
- **基本定位**：定位属性会为定位元素选择替代规则
 - **静态定位**：静态定位使元素能够使用常规行为
 - **绝对定位**：不会为元素留空间，而是位于相对于祖先或容器块的特定位置
 - **固定**：不会为元素留空间，而是位于相对于屏幕视口的特定位置
 - **弹性盒**：一种布局模式，可以组织网页上的元素，使元素行为能够符合预期，这样网页布局就能够满足不同的屏幕大小和设备显示器的要求
 - **悬浮型**：指定元素应该遵守常规版型，并放置在容器的左侧或右侧
- **字体样式和网络字体**：字体样式使我们能够更改文本的外观；网络字体使我们能够加载只有部分客户端能使用的网络字体文件
- **背景**：背景使我们能够定义用作容器背景的颜色或图片
- **伪选择器**：伪选择器使我们能够选择出现在 HTML 中定义的元素周围的假定元素
- **动画和过渡**：动画和过渡使我们能够对元素设定动画效果或在元素的两个状态之间定义过渡

JavaScript

在三大网络语言中，**JavaScript** 是唯一的编程语言。**JavaScript** 负责控制网站的交互操作。就像一位勤杂工人，可以拆掉墙壁、建造新的房间和重新装饰房屋。对于简单的静态网站，你不需要使用太多的 **JavaScript**。但是对于动态网络应用来说，你将需要深入学习该语言。



- **语法**：定义如何组织语言的一般规则
- **数据类型**：该语言支持的不同变量类型（例如字符串和整型）
- **函数**：用来执行特定任务的代码块
- **对象字面值**：**JavaScript** 中的所有内容都是对象，但是自己编写对象字面值可以简化代码
- **面向对象的编程**：在 **JavaScript** 中，你可以采用多种方式来实现面向对象的编程，包括函数方式、原型方式和伪类方式
- **设计模式**：设计模式是可以重复利用的常见问题解决方案
- **AJAX**：**AJAX** 使我们能够异步地从网络服务器上请求数据，不需要重新加载网页
- **jQuery**：**jQuery** 是一种非常热门的库，使我们能够更轻松地进行跨浏览器 DOM 遍历和操作、处理事件和执行 **AJAX** 操作

响应式网页设计

打开网站并缩小浏览器窗口大小。网页内容是否更改了布局，以适应新的屏幕？这就是响应式设计在起作用。人们希望现代网站能够在手机、平板电脑和笔记本上都具有美观的界面。通过学习响应式设计原则，你将了解如何使网站能够缩放，并进行自我调整，从而在所有设备上都能提供超棒的体验。



- **@媒体查询**：媒体查询使内容能够根据具体的输出设备的范围调整呈现方式，不用更改内容本身
- **相对大小单位**：CSS 还提供了除像素 (px) 之外的很多其他衡量单位，例如 em、rem、vw、vh 和 vmin

CSS 框架

Bootstrap 是一个典型的 CSS 框架示例。框架使我们能够轻松地设计网站结构和构建网站。它们会提供自定义 CSS 类，简化了内容布局操作，确保无论是何种设备，你的内容都能看起来很美。框架可以帮助你遵循行业最佳做法和现代设计原则。

- ❑ **Bootstrap:** Bootstrap 是一种 CSS 框架，一开始由 Twitter 开发而成，使创建响应式设计变得更加轻松
- ❑ **Foundation:** Foundation 是另一种 CSS 框架，开发者为 Zurb，同样使创建响应式设计变得更加轻松



JavaScript 库和框架

JavaScript 库和框架会强制要求各种最佳做法，并且通常会强制要求我们在处理各种文件时遵守组织性格式，使我们能够轻松地编写网络应用。JavaScript 库和框架还会处理你可能会遇到的大部分跨浏览器兼容问题，包括各种性能优化。JavaScript 库和框架示例包括 AngularJS、EmberJS 和 KnockoutJS。

- ❑ **AngularJS:** 支持双向数据绑定，使你能够扩展 HTML 词汇以创建前端网络应用
- ❑ **EmberJS:** 通过使用严格的文件和对象命名规范，不用再使用样板代码
- ❑ **KnockoutJS:** 通过声明性绑定系统，使我们能更轻松地创建由数据驱动的应用



版本控制

对于复杂的项目，你该如何维护代码？**Git** 等版本控制软件可以帮助全球的软件开发者保存和维护代码，即使项目发展到需要数百个开发者参与协作，并包含十几个子项目，也没问题。

❑ **Git**: Git 是一种分布式版本控制系统

- ❑ clone
- ❑ add
- ❑ commit
- ❑ push
- ❑ pull
- ❑ branch
- ❑ log

❑ **GitHub**: GitHub 是一种 Git 资源库网络托管服务，提供各种其他功能，使开发者能够相互协作。

- ❑ forking
- ❑ pull requests

```
<!DOCTYPE html>
<head>
  <meta charset="utf-8">
  <title>Awesome Site</title>
  <link rel="stylesheet" href="css/style.css">
</head>
<body>
  <div class="top">
    <div class="container">
      <div class="three columns info-top">
        <p class="font-icon">q</p>
        <p>(123)-456 -789-00</p>
      </div>
      <div class="four columns info-top">
        <p class="font-icon">w</p>
        <p>yourmail@company.com</p>
      </div>
    </div>
  </div>
</body>
</html>
```

网站性能

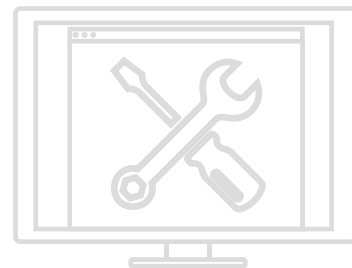
构建好网站后，如何确保网站运行速度很快？通过理解几种简单的浏览器渲染原则，你将能够确保向用户呈现快速、高效的网站。



- ❑ **关键呈现路径：**关键呈现路径是浏览器用来将 **HTML**、**CSS** 和 **JavaScript** 转换成实际像素并发送到用户屏幕的流程
- ❑ **图片优化：**图片优化是针对图片内容使用正确的图片类型和从图片文件中移除多余的元数据的流程
- ❑ **JavaScript 压缩：**JavaScript 压缩是指从 **JavaScript** 文件中删除不必要的字符以减小文件大小的流程

浏览器开发者工具

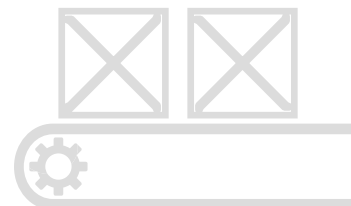
在构建网络应用时，你肯定会在代码中遇到 **bug**、性能问题，或者网页在浏览器中呈现时会出现异常。解决这些问题的唯一方式是了解浏览器如何解析你的代码。浏览器开发者工具提供了所有的详细信息，可以帮助你揭开代码的神秘面纱，是测试、衡量和迭代改进代码的重要平台。



- ❑ **元素检测：**元素面板使你能够查看一个 **DOM** 树中的所有内容，并检查和在运行中修改 **DOM** 元素
- ❑ **网络：**网络面板可以录制关于应用中每项网络操作的信息
- ❑ **时间轴：**时间轴面板使你能够录制和分析应用在运行期间的所有活动
- ❑ **应用调试：**调试面板使你能够观察应用在运行期间消耗的内存
- ❑ **资源：**资源面板使你能够检测应用加载的资源

构建和自动化工具

构建应用远远不止编写代码这么简单！你需要运行测试套件、优化图片、遵守你所在单位的代码格式指南，甚至准备将代码部署到成品服务器上。还需要完成大量的重复性甚至枯燥的工作。**Grunt** 和 **Gulp** 等构建和自动化工具可以在后台帮你处理所有这些任务，使你能够专注于构建强大的网络应用。



- ❑ **npm:** npm 是 Node.js 的默认程序包管理器，后者是大多数构建和自动化工具编写代码时用到的框架
- ❑ **Grunt:** Grunt 是基于任务的命令行构建工具，可以与硬盘上的文件交互
- ❑ **Gulp:** Gulp 是基于程序的命令行构建工具，负责阅读硬盘上的文件，然后以流的形式与这些文件交互
- ❑ **Bower:** Bower 是 HTML、CSS 和 JavaScript 库的程序包管理器，使你能够定义和检索依赖项并确定其版本
- ❑ **Yeoman:** Yeoman 是一种支架应用，能够根据你所定义的框架和库自动为各种应用生成样板代码

测试

当你的应用变得越来越复杂时，就容易出现 **bug**，或者完全破坏现有的功能。单元、集成和行为测试是很棒的测试方法，可以确保你在添加新功能时不会破坏代码。出色的测试框架示例包括 **Mocha** 和 **Jasmine**。



- ❑ **Mocha:** Mocha 是一个 JavaScript 测试框架，在 Node.js 和浏览器中运行，使异步测试变得简单起来
- ❑ **Jasmine:** Jasmine 是一个开源、行为驱动的 JavaScript 测试框架

软技能

掌握工作所需的技术技能非常重要，但是软技能也同样重要。软技能使你能够在团队环境下或自己一个人出色地完成工作。培养自己的沟通交流能力、指导和激励他人的能力，并能够从宏观层面看待工作，将使你成为潜在雇主青睐的候选人士。



- **善于沟通：**能够向庞大的团队（包括管理层、同事和客户）传达目标、进度和问题
- **灵活地处理问题：**了解问题背后的问题，并将问题分解成更小、更易解决的小问题
- **积极上进：**及时掌握这一领域的发展动态，并了解最新的技术发展状况
- **自我激励：**愿意实验和探索；勇于冒险，但是在面临挫折时能够坚持不懈

下一步做什么？

学习资源

你已经读完整个技能清单了，恭喜你！

无论你是否已经掌握了多项技能，还是从头开始学习这些技能，都好好鼓励下自己吧，你终于踏出了阅读这一指南的第一步。

正如我们在本指南的开头提到的，我们将帮助你排除干扰，做出正确的学习决策。



欢迎查看我们的《前端工程师》认证项目。这门课程将系统地帮助你学习所有这些技能。学习期间，你将获得项目导师和其他学员的帮助。

在《前端工程师》认证项目中，你将通过完成多个实战项目学习 Web 开发的基础原理和应用知识，你将构建项目作品集，向雇主展示你掌握的技能。你可以将这个技能清单看做你的学习蓝图，将该课程看做行动计划。

你将获得 Udacity 基石纳米学位或纳米学位认证，向雇主证明你的专业知识和技术水平！你每周需要投入 6-8 小时学习，可以在 3 个月内学完该课程。

如果你想制定一个不用花费太多时间的学习计划，或者想提高自己的某项特定技能，不妨看看以下各门课程：

[HTML 和 CSS 入门](#) – 了解如何将数字设计模块变成静态网页，以及如何构建自适应作品集网站来展示你的作品。

[JavaScript 基础知识](#) – 通过为你的作品集创建在线简历，学习 JavaScript 语法和网络开发工程师创建交互式动态网站采用的编程规范。

[jQuery 入门](#) – 了解如何使用 jQuery 访问和修改 DOM！这门课程将讲授如何使用 jQuery 的核心功能：DOM 元素选择、遍历和操纵。

[面向对象的 JavaScript](#) – 构建各种 JavaScript 对象，并了解它们的继承模型对应用的内存模型有何影响。使你的代码更简化，并更具模块性。

[HTML5 Canvas](#) – 了解如何使用 HTML5 Canvas 创建和修改图片，甚至包括互动动画。

[网站性能优化](#) – 了解浏览器如何将 HTML、CSS 和 JavaScript 变成网站，并学习使用 Chrome 开发者工具衡量和优化网站速度！

[AJAX 入门](#) – 在这门课程中，你将利用 Google 街景、《纽约时报》和维基百科中的数据构建一款网络应用，学习如何使用 jQuery 的 AJAX 功能通过 JavaScript 发出异步请求，以及如何执行异步请求！

请参阅以下资源 and 社区，及时获悉这一蓬勃发展的领域的最新资讯：

我们的合作伙伴

- ☐ [Ilya Grigorik](#)
- ☐ [Paul Lewis](#)
- ☐ [Paul Kinlan](#)
- ☐ [Pete LePage](#)

图书

- ☐ [Dive Into HTML5](#)
- ☐ [Eloquent JavaScript](#)
- ☐ [JavaScript Design Patterns](#)

简报

- ☐ [HTML5 Weekly](#)
- ☐ [Tales from the Front End](#)
- ☐ [JavaScript Weekly](#)
- ☐ [Hacker Newsletter](#)

博客/网站

- ☐ [HTML5 Doctor](#)
- ☐ [HTML5 Hub](#)
- ☐ [HTML5 Rocks](#)
- ☐ [Smashing Magazine](#)
- ☐ [A List Apart](#)
- ☐ [CSS Tricks](#)

