**前端开发指南**

# 什么是前端开发？

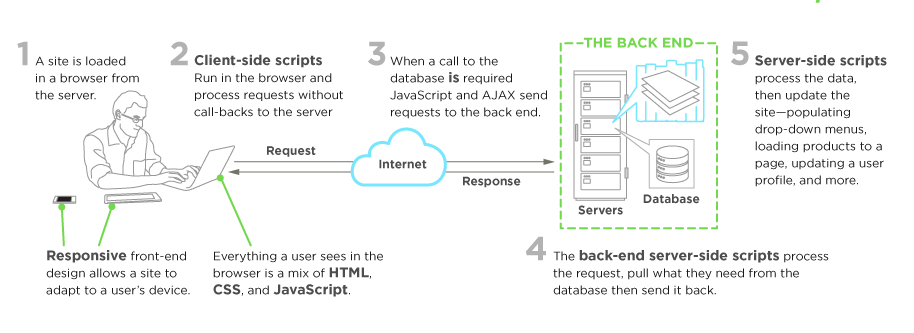
前端Web开发也称为客户端开发，是为网站或Web应用程序生成HTML，CSS和JavaScript以便用户可以直接查看并与之交互的实践。 与前端开发相关的挑战是用于创建网站前端的工具和技术不断变化，因此开发人员需要不断了解该领域的发展方向。

设计网站的目的是确保当用户打开网站时，他们以易于阅读和相关的格式查看信息。 用户现在使用具有不同屏幕尺寸和分辨率的各种设备，从而迫使设计者在设计网站时考虑到这些方面，这一事实进一步复杂化。 他们需要确保他们的网站在不同的浏览器（跨浏览器），不同的操作系统（跨平台）和不同的设备（跨设备）上正确显示，这需要开发人员的周密计划。

— [*维基百科*](https://en.wikipedia.org/wiki/Front-end_web_development)

##### HTML，CSS和JavaScript：

前端开发人员架构师使用Web技术（即HTML，CSS，DOM和JavaScript）开发网站和应用程序，[网络平台](https://en.wikipedia.org/wiki/Open_Web_Platform)或者充当非web平台环境的编译输入（即，[NativeScript](https://www.nativescript.org/)).



通常，一个人通过学习开发HTML，CSS和JS代码，进入前端开发领域[网页浏览器](https://en.wikipedia.org/wiki/Web_browser), [无头浏览器](https://en.wikipedia.org/wiki/Headless_browser), [的WebView](http://developer.telerik.com/featured/what-is-a-webview/)，或作为本机运行时环境的编译输入。 下面将解释这四个运行时间场景。

##### 网页浏览器

网络浏览器是用来检索，呈现和遍历信息的软件[万维网](https://en.wikipedia.org/wiki/World_Wide_Web)。 通常情况下，浏览器可以在台式机或笔记本电脑，平板电脑或手机上运行，​​但最近浏览器几乎可以找到任何东西（例如，在冰箱，汽车等）。

最常见的网页浏览器（以最常用的顺序显示）：

* [铬](http://www.google.com/chrome/)
* [IE浏览器](https://en.wikipedia.org/wiki/Internet_Explorer)（注意：不是[边缘](http://dev.modern.ie/)，参照IE 9至IE 11）
* [火狐](https://www.mozilla.org/firefox/)
* [苹果浏览器](http://www.apple.com/safari/)

##### 无头浏览器

无头浏览器是一种没有图形用户界面的Web浏览器，可以通过编程方式从命令行界面进行控制，以实现网页自动化（例如，功能测试，抓取，单元测试等）。 将无头浏览器想象成可以从可以检索和遍历网页的命令行运行的浏览器。

最常见的无头浏览器是：

* [PhantomJS](http://phantomjs.org/)
* [slimerjs](http://slimerjs.org/)
* [trifleJS](http://triflejs.org/)

##### 网页视图

[网页视图](http://developer.telerik.com/featured/what-is-a-webview/)由本地应用程序中的本机操作系统使用，以运行网页。 想想一个[网页流量](http://developer.telerik.com/featured/what-is-a-webview/)如嵌入在设备上运行的本机应用程序中的web浏览器的iframe或单个选项卡（例如，[iOS版](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIWebView_Class/), [安卓](http://developer.android.com/reference/android/webkit/WebView.html), [视窗](https://msdn.microsoft.com/library/windows/apps/windows.ui.xaml.controls.webview.aspx)).

最常见的解决方案[网页流量](http://developer.telerik.com/featured/what-is-a-webview/)发展是：

* [科尔多瓦](https://cordova.apache.org/)（通常用于本地手机/平板电脑应用）
* [NW.js](https://github.com/nwjs/nw.js)（通常用于桌面应用程序）
* [电子](http://electron.atom.io/)（通常用于桌面应用程序）

##### 原生来自Web Tech

最终，从Web浏览器开发中学到的东西可以被前端开发人员用来为不受浏览器引擎推动的环境编写代码。 到最近，开发环境正在梦想使用Web技术（例如，CSS和JavaScript），而无需使用Web引擎来创建本机应用程序。

这些环境的一些例子是：

* [NativeScript](https://www.nativescript.org/)
* [反应原生](https://facebook.github.io/react-native/)

###### 笔记：

确保你清楚“网络平台”究竟意味着什么。 读，[“Web平台：它是什么”](http://tess.oconnor.cx/2009/05/what-the-web-platform-is)并阅读，[“开放式网络平台”](https://en.wikipedia.org/wiki/Open_Web_Platform)维基百科页面。

**前端工作职位**

以下是各种前端职位名单的列表和说明。 前端开发人员常用的或最常用的（即通用的）标题是“前端开发人员”或“前端工程师”。 请注意，包含单词“前端”，“客户端”，“网络用户界面”，“HTML”，“CSS”或“JavaScript”的任何作业通常会推断出某人具有某种程度的HTML，CSS， DOM和JavaScript专业知道如何。

**前端开发人员**

这是一个通用职位名称，描述了一些在HTML，CSS，DOM和JavaScript方面具有一定技能并且在Web平台上实现这些技术的开发人员。

**前端工程师（也称为JavaScript开发人员或全栈JavaScript开发人员）**

这个职位给来自计算机科学，工程和背景的开发人员，并且正在使用这些技能来处理前端技术。 这个角色通常需要计算机科学学位和多年的软件开发经验。 如果职位名称中包含“JavaScript应用程序”一词，则表示开发人员应该是具有高级编程，软件开发和应用程序开发技能（即具有多年构建前端应用程序的经验）的高级JavaScript开发人员。

**CSS / HTML开发人员**

前端工作职位描述了熟练掌握HTML和CSS的开发人员，不包括JavaScript和应用程序。

**前端Web设计器**

当职位名称中包含“设计师”这个词时，这表示设计师将拥有前端技能（即HTML和CSS），但也包含专业设计（​​视觉设计和交互设计）技能。

**Web /前端用户界面（又名UI）开发人员/工程师**

当职位名称中包含“接口”或“用户界面”这个词时，这表示除了前端开发人员技能或前端工程技能外，开发人员还应具备交互设计技能。

**移动/平板电脑前端开发人员**

当职位名称中包含“移动”或“平板电脑”这个词时，这表示开发人员具有开发在移动设备或平板电脑设备上运行的前端的经验（无论是本地还是在Web平台上，即在浏览器中）。

**前端搜索引擎优化专家**

当职位名称中包含“SEO”这个词时，这表示开发人员具有丰富的制定SEO策略的前端技术的丰富经验。

**前端可访问性专家**

当职位名称中包含“辅助功能”一词时，这表示开发人员在制定支持辅助功能要求和标准的前端技术方面拥有丰富的经验。

**前端开发 行动**

当职位名称中包含“DevOps”这个词时，这表示开发人员在与协作，集成，部署，自动化和测量相关的软件开发实践方面拥有丰富的经验。

**前端测试/质量保证**

当职位名称中包含“测试”或“QA”这个词时，这表示开发人员具有丰富的测试和管理软件的经验，这些软件涉及单元测试，功能测试，用户测试和A / B测试。

请注意，如果您遇到职位名称中的“完整堆栈”或通用“Web开发人员”字样，这些字可能会被雇主用来描述负责Web /应用程序开发所有方面的角色，即前（可能包括设计）和后端。

# 前端开发人员使用的网络技术



前端开发人员采用以下核心Web技术（请考虑按此顺序学习）：

1. 统一资源定位器（又名URL）
2. 超文本传输​​协议（又名HTTP）
3. 超文本标记语言（又名HTML）
4. 级联样式表（又名CSS）
5. JavaScript编程语言（又名ECMAScript 262）
6. JavaScript对象表示法（又名JSON）
7. 文档对象模型（又名DOM）
8. Web API（又名HTML5和朋友或浏览器API）
9. 网页内容可访问性指南（又名WCAG）和可访问的富互联网应用程序（又名ARIA）

这些技术在相关文档和规范中定义如下。 有关所有网络相关规范的综合列表，请参阅[platform.html5.org](https://platform.html5.org/).

##### 超文本标记语言（又名HTML）

超文本标记语言（通常称为HTML）是用于创建网页的标准标记语言。 Web浏览器可以读取HTML文件并将其呈现为可见或可听的网页。 HTML描述了网站的语义结构和演示提示，使其成为标记语言，而不是编程语言。

— [*维基百科*](https://en.wikipedia.org/wiki/HTML)

最相关的规格/文件：

* [所有W3C HTML规范](http://www.w3.org/standards/techs/html#w3c_all)
* [生活标准中的HTML元素](https://html.spec.whatwg.org/multipage)
* [全局属性](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes)
* [来自W3C的HTML 5.2](http://w3c.github.io/html/)
* [HTML属性参考](https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes)
* [HTML元素参考](https://developer.mozilla.org/en-US/docs/Web/HTML/Element)
* [HTML语法](https://html.spec.whatwg.org/multipage/syntax.html#syntax)来自生活水平

##### 级联样式表（又名CSS）

层叠样式表（Cascading Style Sheets，CSS）是一种样式表语言，用于描述用标记语言编写的文档的外观和格式。 尽管通常用于改变用HTML和XHTML编写的网页和用户界面的风格，但该语言可以应用于任何类型的XML文档，包括纯XML，SVG和XUL。 与HTML和JavaScript一起，CSS是大多数网站使用的基础技术，可用于创建视觉吸引力的网页，Web应用程序的用户界面以及许多移动应用程序的用户界面。

— [*维基百科*](https://en.wikipedia.org/wiki/Cascading_Style_Sheets)

最相关的规格/文件：

* [所有W3C CSS规范](http://www.w3.org/Style/CSS/current-work#roadmap)
* [级联样式表级别2修订2（CSS 2.2）规范](https://drafts.csswg.org/css2/)
* [CSS参考](https://developer.mozilla.org/en-US/docs/Web/CSS/Reference)
* [选择器3级](http://www.w3.org/TR/css3-selectors/)

##### 文档对象模型（又名DOM）

文档对象模型（DOM）是一种跨平台和独立于语言的约定，用于表示HTML，XHTML和XML文档中的对象并与其交互。 每个文档的节点都组织在一个树形结构中，称为DOM树。 DOM树中的对象可以通过使用对象上的方法来寻址和操作。 DOM的公共接口在其应用程序编程接口（API）中指定。

— [*维基百科*](https://en.wikipedia.org/wiki/Document_Object_Model)

最相关的规格/文件：

* [文档对象模型（DOM）级别3事件规范](https://www.w3.org/TR/DOM-Level-3-Events/)
* [DOM生活标准](https://dom.spec.whatwg.org/)
* [W3C DOM4](https://www.w3.org/TR/2015/REC-dom-20151119/)

##### JavaScript编程语言（又名ECMAScript 262）

JavaScript是一种高级的，动态的，无类型的和解释性的编程语言。 它已经在ECMAScript语言规范中标准化。 除HTML和CSS外，它还是万维网内容制作的三大关键技术之一; 大多数网站都使用它，它受到所有现代网页浏览器的支持，无需插件。 JavaScript是基于原型的一流功能，使其成为一种多范式语言，支持面向对象，强制性和功能性编程风格。 它有一个用于处理文本，数组，日期和正则表达式的API，但不包括任何I / O，例如网络，存储或图形设备，它们依赖于嵌入它的主机环境。

— [*维基百科*](https://en.wikipedia.org/wiki/JavaScript)

最相关的规格/文件：

* [ECMAScript®2017语言规范](https://tc39.github.io/ecma262/)

##### Web API（又名HTML5和朋友）

当使用JavaScript为Web编写代码时，有很多API可用。 以下是您在开发Web应用程序或网站时可以使用的所有接口（即对象类型）的列表。

— [*Mozilla的*](https://developer.mozilla.org/en-US/docs/Web/API)

最相关的文件：

* [Web API接口](https://developer.mozilla.org/en-US/docs/Web/API)

##### 超文本传输​​协议（又名HTTP）

超文本传输​​协议（HTTP）是分布式协作超媒体信息系统的应用协议。 HTTP是万维网数据通信的基础。

— [*维基百科*](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol)

最相关的规格：

* [超文本传输​​协议 - HTTP / 1.1](https://tools.ietf.org/html/rfc2616)
* [HTTP/2](https://http2.github.io/)

##### 统一资源定位器（又名URL）

统一资源定位符（URL）（也称为Web地址）是对资源的引用，用于指定计算机网络上资源的位置以及用于检索它的机制。 URL是特定类型的统一资源标识符（URI），尽管很多人可以互换使用这两个术语。 URL意味着访问指示资源的手段，这不是每个URI的真实情况。 URL通常用于引用网页（http），但也用于文件传输（ftp），电子邮件（mailto），数据库访问（JDBC）以及许多其他应用程序。

— [*维基百科*](https://en.wikipedia.org/wiki/Uniform_Resource_Locator)

最相关的规格：

* [统一资源定位符（URL）](http://www.w3.org/Addressing/URL/url-spec.txt)
* [URL生活标准](https://url.spec.whatwg.org/)

##### JavaScript对象表示法（又名JSON）

c它是用于异步浏览器/服务器通信（AJAJ）的主要数据格式，主要取代XML（由AJAX使用）。 虽然最初源自JavaScript脚本语言，但JSON是独立于语言的数据格式。 解析和生成JSON数据的代码很容易在许多编程语言中使用。 JSON格式最初由Douglas Crockford指定。 它目前由两个相互竞争的标准描述，即RFC 7159和ECMA-404。 ECMA标准最小，只描述允许的语法语法，而RFC也提供了一些语义和安全考虑。 JSON的官方Internet媒体类型是application / json。 JSON文件扩展名是.json。

— [*维基百科*](https://en.wikipedia.org/wiki/JSON)

最相关的规格：

* [介绍JSON](http://json.org/)
* [JSON API](http://jsonapi.org/)
* [JSON数据交换格式](http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf)

##### 网页内容可访问性指南（又名WCAG）和可访问的富互联网应用程序（又名ARIA）

无障碍是指为残障人士设计的产品，设备，服务或环境。 可访问设计的概念确保了“直接访问”（即无辅助）和“间接访问”两者的含义，即与个人的辅助技术（例如电脑屏幕阅读器）兼容。

— [*维基百科*](https://en.wikipedia.org/wiki/Accessibility)

* [可访问的富互联网应用程序（WAI-ARIA）当前状态](http://www.w3.org/standards/techs/aria#w3c_all)
* [无障碍网络倡议（WAI）](http://www.w3.org/WAI/)
* [网页内容无障碍指引（WCAG）目前状况](http://www.w3.org/standards/techs/wcag#w3c_all)

# 前端开发技能



基本到高级HTML，CSS，DOM，JavaScript，HTTP / URL和浏览器技能都假定适用于任何类型的前端开发人员。

除了HTML，CSS，DOM，JavaScript，HTTP / URL和浏览器开发知识之外，前端开发人员还可以熟练掌握以下一项或多项内容：

* 内容管理系统（又名CMS）
* Node.js的
* CrossBrowser测试
* 跨平台测试
* 单元测试
* 跨设备测试
* 无障碍/ WAI-ARIA
* 搜索引擎优化（又名SEO）
* 交互或用户界面设计
* 用户体验
* 可用性
* 电子商务系统
* 门户系统
* 线框
* CSS布局/网格
* DOM操作（例如，jQuery）
* 移动Web性能
* 负载测试
* 性能测试
* 渐进式增强/优雅的降解
* 版本控制（例如，GIT）
* MVC / MVVM / MV\*
* 功能编程
* 数据格式（例如JSON，XML）
* 数据API（例如Restful API）
* Web字体嵌入
* 可缩放矢量图形（又名SVG）
* 常用表达
* 内容策略
* 微数据/微格式
* 任务运行程序，构建工具，过程自动化工具
* 响应式网页设计
* 面向对象编程
* 应用架构
* 模块
* 依赖经理
* 包管理员
* JavaScript动画
* CSS动画
* 图表/图表
* 用户界面小部件
* 代码质量测试
* 代码覆盖测试
* 代码复杂性分析
* 集成测试
* 命令行/ CLI
* 模板策略
* 模板引擎
* 单页应用程序
* XHR请求（aka AJAX）
* 网络/浏览器安全
* HTML语义
* 浏览器开发工具

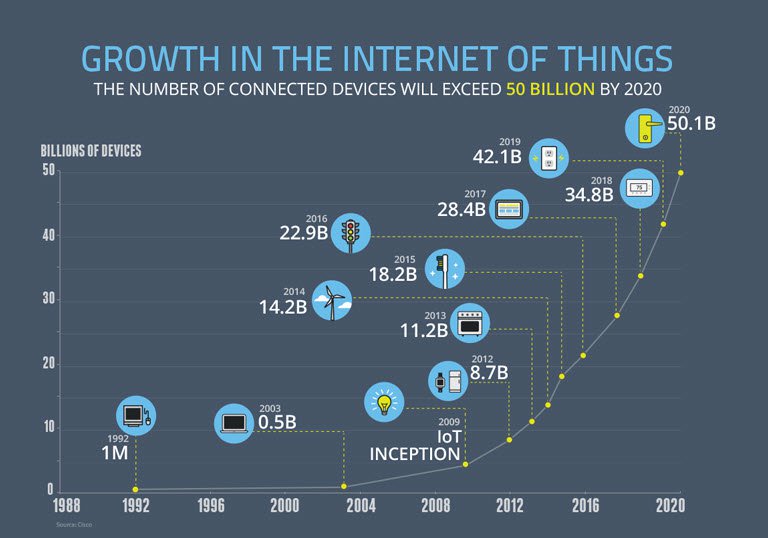
# 前端开发人员为...开发

前端开发人员通常在HTML上运行HTML，CSS和JS[网络平台](http://tess.oconnor.cx/2009/05/what-the-web-platform-is)（例如网络浏览器）从以下操作系统之一（又名操作系统）提供：

* Android的
* 铬
* iOS版
* OS X
* Ubuntu（或者某些Linux的风格）
* Windows Phone
* 视窗

这些操作系统通常运行在以下一个或多个设备上：

* 台式电脑
* 笔记本电脑/上网本电脑
* 移动电话
* 片剂
* 电视
* 看
* 事物（即任何你能想象的事物，汽车，冰箱，灯光，恒温器等）



图片来源：[*https://www.enterpriseirregulars.com/104084/roundup-internet-things-forecasts-market-estimates-2015/*](https://www.enterpriseirregulars.com/104084/roundup-internet-things-forecasts-market-estimates-2015/)

一般而言，前端技术可以使用以下运行时Web平台方案在上述操作系统和设备上运行：

* 网页浏览器（例如：[Chrome，IE，Safari，Firefox](http://outdatedbrowser.com/en)).
* A [无头浏览器](https://en.wikipedia.org/wiki/Headless_browser)（例子：[phantomJS](http://phantomjs.org/)).
* A [的WebView](http://developer.telerik.com/featured/what-is-a-webview/)/浏览器选项卡（认为iframe）嵌入在本机应用程序中作为与本地API桥接的运行时。 WebView应用程序通常包含从Web技术构建的用户界面。 （即HTML，CSS和JS）。 （例子：[阿帕奇科尔多瓦](https://cordova.apache.org/), [NW.js](http://nwjs.io/), [电子](http://electron.atom.io/))
* 从网络技术构建的本地应用程序，在运行时通过桥接本地API进行解释。 UI将利用本地UI部分（例如，iOS本地控件）而不是Web技术。 （例子：[NativeScript](https://www.nativescript.org/), [反应原生](https://facebook.github.io/react-native/))

# 团队的前端

前端开发人员通常只是设计和开发网站，Web应用程序或从Web技术运行的本机应用程序的团队中的一员。

一个用于为Web平台构建专业网站或软件应用程序的裸骨架开发团队通常最少会包含以下角色。

* 视觉设计师（即字体，颜色，间距，情感，视觉概念和主题）
* UI /交互设计器/信息架构师（即线框，指定所有用户交互和UI功能，结构化信息）
* FrontEnd Developer（即，编写在客户端/设备上运行的代码）
* 后端开发者（即，编写在服务器上运行的代码）

角色根据重叠的技能进行排序。 前端开发人员通常会很好地处理UI /交互设计以及后端开发。 团队成员通过承担重叠角色的责任来填补多个角色并不罕见。

假设上述团队由项目负责人或某种产品所有者（即利益相关者，项目经理，项目负责人等）领导，

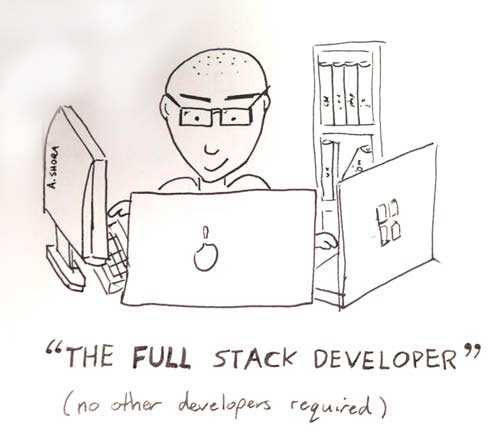
一个更大的网络团队可能包括以下未显示的角色：

* SEO策略师
* DevOps工程师
* API开发人员
* 数据库管理员
* QA工程师/测试员

###### 笔记：

一个“全栈开发者”负责前端和后端开发人员的职责似乎正在发生一个小的趋势。

# 通用/全堆神话



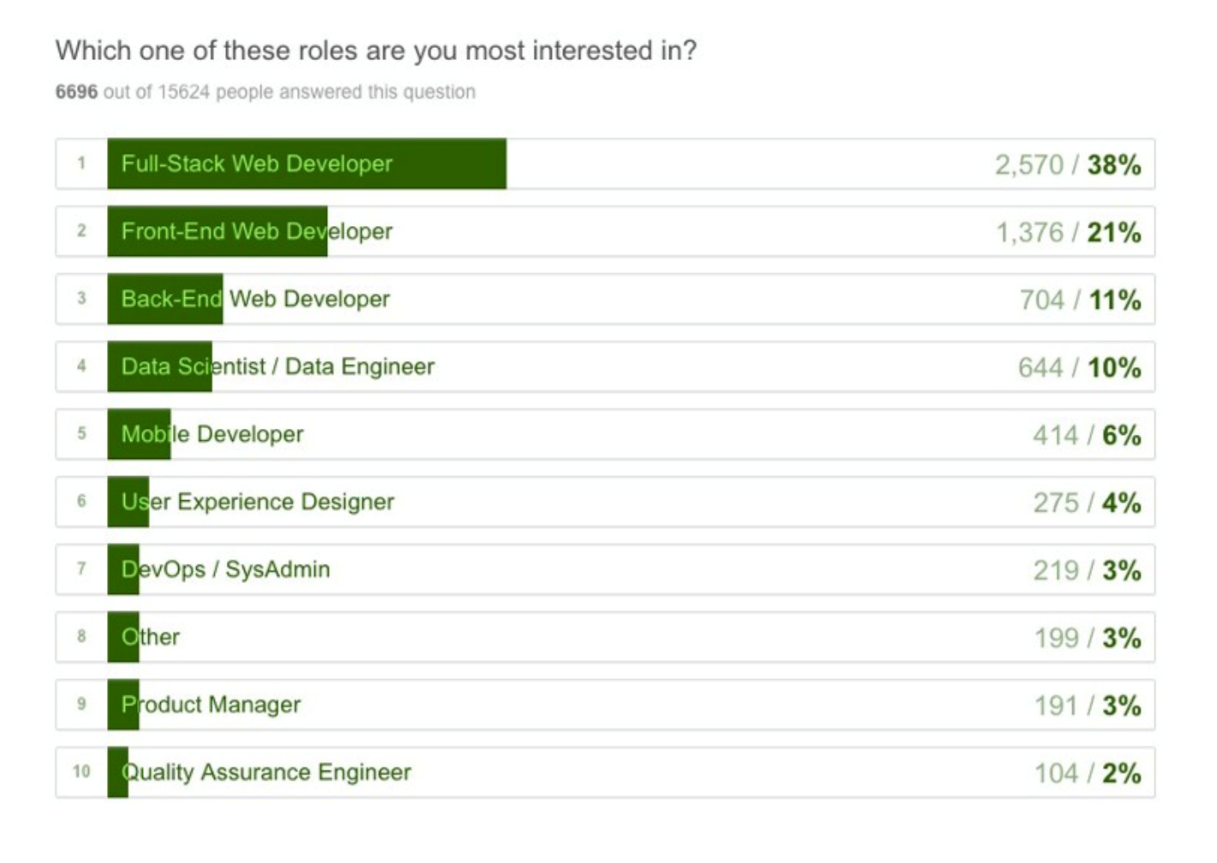
设计和开发Web解决方案所需的角色需要在视觉设计，UI /交互设计，前端开发和后端开发领域拥有深厚的技能和丰富的经验。 任何能够以专业水平填补这四个角色中的一个或多个的人是非常罕见的商品。

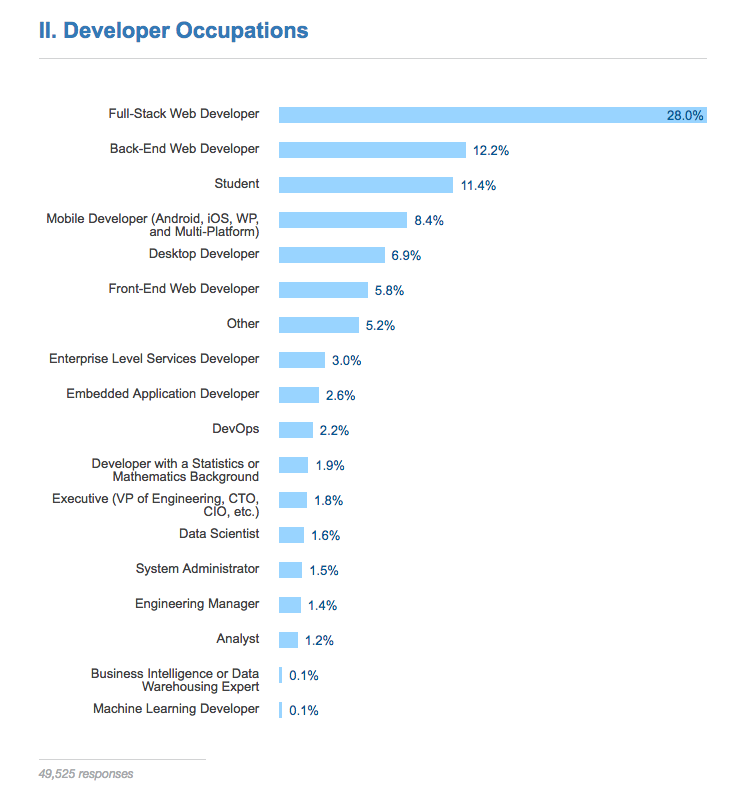
从实际角度来看，您应该寻求成为或寻求雇用这些角色中的专家（即视觉设计，交互设计/ IA，前端开发人员，后端开发人员）。 那些声称在一个或多个角色中以专家级别进行操作的人非常罕见，而且可能是神话般的。

但是，鉴于JavaScript渗透了技术堆栈的所有层（例如React，node.js，express，couchDB，gulp.js等），找到能够编码前端和后端的完整堆栈JS开发人员是变得不那么神话。 通常，这些完整的堆栈开发人员只处理JavaScript。 可以对前端，后端，API和数据库进行编码的开发人员不像以前那样荒谬（不包括视觉设计，交互设计和CSS）。 在我看来仍然是神话，但并不像以前那样不寻常。 因此，我不会推荐一个开发者成为一个“全栈”开发者。 在极少数情况下，它可以工作。 但是，作为构建前端开发人员职业生涯的一般概念，我主要关注前端技术。

###### 笔记：

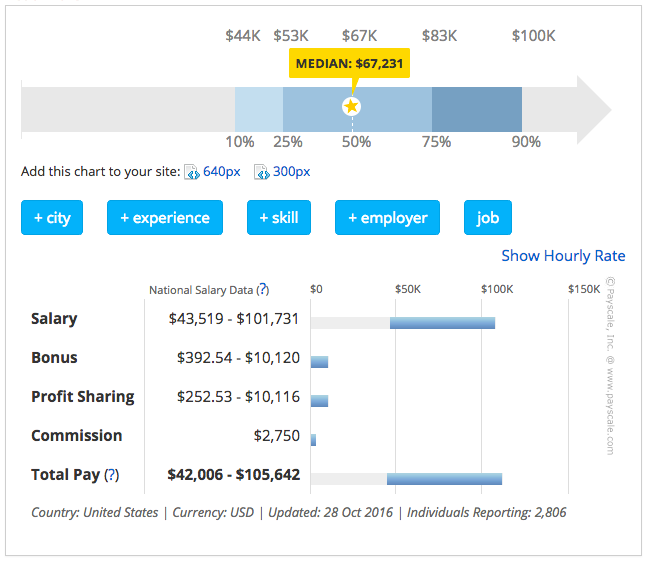
“Full-Stack”开发人员已经具备了多种含义。 这么多，这个词在使用时没有一个意思是明确的。 请考虑下面显示的两项调查的结果。 这些结果会让人相信大多数开发人员都是全栈开发人员。 但是，在我近20年的经验中，这不过是这样。





# 前端工资

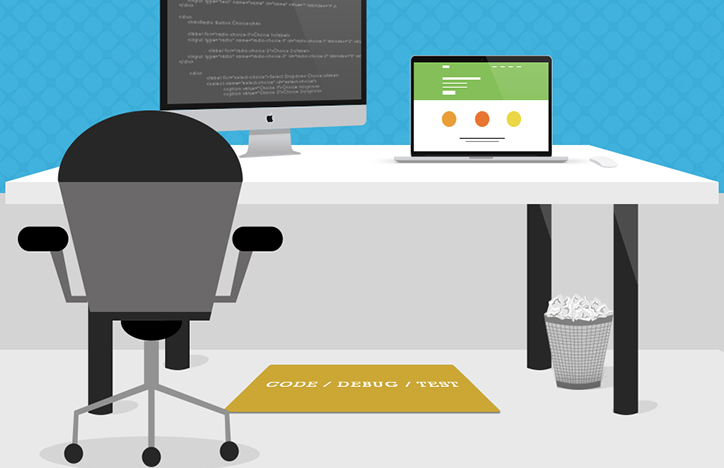
在美国，中等水平的前端开发人员的全国平均水平在附近[$75k](http://www.glassdoor.com/Salaries/front-end-web-developer-salary-SRCH_KO0,23.htm)。 当然，当你第一次开始期望以35k左右进入该领域时，取决于位置和投资组合。



###### 笔记：

一位领先/高级前端开发人员/工程师可以随时随地居住（即远程工作），年销售额超过150万美元（访问[angel.co](https://angel.co/jobs)，注册，审查超过150,000美元的前端工作或检查工资范围[堆栈溢出作业](https://stackoverflow.com/jobs?q=front-end&sort=y)).

# 前端开发人员是如何制造的



如何成为前端开发人员？ 好吧，这很复杂。 今天仍然不能上大学，并且期望毕业时获得前端工程学位。 而且，我很少听到或遇到前端开发人员，这些开发人员因可能被弃用的计算机科学学位或平面设计学位而苦于无法专业编写HTML，CSS和JavaScript。 从我的角度来看，今天大多数从事前端工作的人员似乎都是自学成才，或者来自未经认可的计划，课程或训练营。

如果你今天要成为一名前端开发人员，我会松散地努力遵循下面列出的流程（第二部分，[“学习前端开发”](https://frontendmasters.gitbooks.io/front-end-handbook-2017/content/learning/self-direct-learning.html)，深入学习资源的更多细节）。

1. 大致了解网络的工作原理。 确保您知道域名，DNS，URL，HTTP，网络，浏览器，服务器/托管，JSON，数据API，HTML，CSS，DOM和JavaScript的“什么”和“哪里”。 不要深入研究任何东西，只要理解它们之间的关系并松散地融合在一起。 关注前端架构的高级概要。 从简单开始[网页](https://github.com/h5bp/html5-boilerplate/blob/master/dist/index.html)并简要研究[前端应用程序（又名SPA）](http://developer.telerik.com/featured/front-end-driven-applications-new-approach-applications/)
2. 学习HTML
3. 学习CSS
4. [学习JavaScript](https://youtu.be/QjKH1J77gjI?list=PL055Epbe6d5bQubu5EWf_kUNA3ef_qbmL)
5. 了解DOM
6. 了解JSON和数据API
7. 了解用户界面设计的基础知识（即UI模式，交互设计，用户体验设计和可用性）。
8. 了解CLI /命​​令行
9. 学习软件工程的实践（即应用程序设计/架构，模板，Git，测试，监控，自动化，代码质量，开发方法）。
10. 获得舆论和定制你的工具箱与任何有意义的大脑（例如Webpack，React和Redux）。
11. 学习Node.js

关于学习的简短建议。[在学习抽象之前学习实际的基础技术。](https://youtu.be/QjKH1J77gjI?list=PL055Epbe6d5bQubu5EWf_kUNA3ef_qbmL)不要学习jQuery，学习DOM。 不要学习SASS，学习CSS。 不要学习HAML，学习HTML。 不要学习CoffeeScript，学习JavaScript。 不要学习Handlebars，学习JavaScript ES6模板。 不要只使用Bootstrap，学习UI模式。

当你开始时，你应该害怕大多数隐藏复杂性的事情。 在错误的手中抽象可以使高级技能的外观，而一直隐藏的事实，即开发人员对基本知识或基本概念的理解较差。

本书的其余部分将向读者指出可用于学习前端开发的潜在资源以及在实施前端开发时使用的工具。 假设在这个旅程中，你不仅学习，而且在你学习和调查工​​具时也会这样做。 有些人建议只做学习。 而其他人则建议只学习做。 我建议你找到与你的大脑如何工作和完成相匹配的两种组合。 但是，当然，这是一个混合！ 所以，不要只是读它，做它。 学习，做。 学习，做。 无限重复，因为事情变化很快。 这就是为什么学习基础知识而不是抽象如此重要。

最近出现了许多未经认证的昂贵的前端代码学校/ bootcamps。 这些成为前端开发人员的途径通常是教师指导的课程，它遵循来自官方教师（即教学大纲，考试，测验，项目，团队项目，成绩等）的更传统的学习方式。 请记住，如果你正在考虑一个昂贵的培训计划，这是网络！ 你所需要学习的一切都在网络上进行，费用从小到无。 但是，如果您需要某人告诉您如何学习和学习实际上免费的东西，并且要求您对其进行学习负责，那么您可以考虑采用有组织的课程。 否则，我不知道有任何其他职业可以通过互联网进行免费接入，每月有一百美元用于屏幕录制会员资格以及对知识的渴望。