**基础知识：**

1、HTML + CSS 这部分建议在[W3school 在线教程](https://link.jianshu.com?t=http://www.w3school.com.cn/" \t "_blank)上学习，边学边练，每章后还有小测试。 之后可以模仿一些网站做些页面。在实践中积累了一些经验后，可以系统的读一两本书，推荐《Head First HTML 与 CSS 中文版》，这本书讲的太细了，我没能拿出耐心细读。你可以根据情况斟酌。

2、Javascript 要学的内容实在很多，如果没有其他编程语言的基础的话，学起来可能要费些力，还是建议先在 W3school上学习。之后建议马上看《Javascript语言精粹》，JS是一门很混乱的语言，这本书能够帮助你区分哪些是语言的精华，哪些是糟粕，对于语言精华，应该深入学习。糟粕部分能看懂别人写的代码就行，自己就不用尝试了。



**进阶：**

有了以上基础，就可以进行一般的静态网页设计，不过对于复杂的页面还需要进一步学习。

1、CSS。必看《精通CSS》，看完这本书你应该对：盒子模型，流动，Block，inline，层叠，样式优先级，等概念非常了解了。作为练习可以看下《CSS艺门之匠》这本书，它对标题，背景，圆角，导航条，table，表单等主题都有详细的介绍。

2、Javascript。上面提到内容还不足以让你胜任JS编程。在有了基础之后，进一步学习内容包括：

a） 框架。

推荐jQuery，简单易用，在W3school简单学习js后，直接上手jQuery即可完成一些简单的项目。学习方法也很简单，照着产品文档做几个页面就行了，不用面面俱到，以后遇到问题查文档就行了。框架可以帮你屏蔽浏览器的差异性，让你能更专注与Web开发学习的精髓部分。补充： 可以使用[Codecademy](https://link.jianshu.com?t=http://www.codecademy.com/" \t "_blank)学习 Javascript，jQuery，用户体验真的很好（感谢 TonyOuyang ）。

b） Javascript 语言范式 。这个名字可能并不恰当，只是我找不到可以描述“面向对象”，“函数式”这个两个概念的概念。Javascript不完全是一个面向对象的语言，它的很多设计理念都有函数编程语言的影子，甚至说如果你不用面向对象，完全可以把它理解成一门函数式编程语言。

Javascript的很多语言特性，都是因为他具有函数式语言的特点才存在的。这部分推荐先学习面向对象的基本理论，对封装，继承，多态等概念要理解，维基百科，百度百科会是你的帮手，另外推荐《Object Oriented Javascript》，应该有中文版。对与函数式编程我了解的也不系统，不好多说，可以自己百度一下。

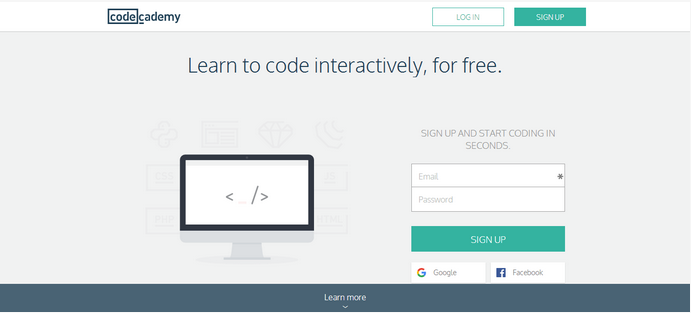
c） Javascript 语言内部机制。必须弄清如下概念：JS 中变量的作用域，变量传递方式，函数的定义环境与执行环境，闭包，函数的四种调用方式（一般函数，对象的方法，apply，call），以及四种调用方式下，‘this’指向的是谁。这部分内容你会在《Javascript语言精粹》中详细了解。另外，你必须理解 json。

d） dom编程，这个Web前端工程师的核心技能之一。必读《Dom编程艺术》，另外《高性能 Javascript》这本书中关于dom编程的部分讲的也很好。

e） Ajax编程，这是另一核心技术。Ajax建议在网上查些资料，了解这个概念的来龙去脉，百度百科，维基百科上的内容就足够了。真正编程是很容易的，如今几乎所有框架都对Ajax有良好的封装，编程并不复杂。

f） 了解浏览器差异性。这部分包括CSS和js两部分，浏览器差异内容很多，建议在实践中多多积累。另外对于浏览器的渲染模式，DOCTYPE等内容应该系统学习。

3、HTML5和CSS3 。HTML5规范已经于2014年10月28日发布了，移动端HTML5和CSS3已经得到了非常广泛的使用，必知必会呀。



**再进一阶 · 代码层面：**

有了以上知识，对于大多数小型网站，你应该已经可以写出能够工作的代码了。但要想成为更专业的前端，你还需继续努力。更高的要求大概还有四方面：1）易维护，2）可测试，3）高性能，4）低流量（移动端）。

1）易维护。对于页面你该理解‘样式’，‘数据’，‘行为’三者分离，对应的当然就是CSS,HTML,js。对于js代码，你最好了解设计模式，重构，MVC等内容。

2）可测性。

3）高性能。必读《高性能Javascript》

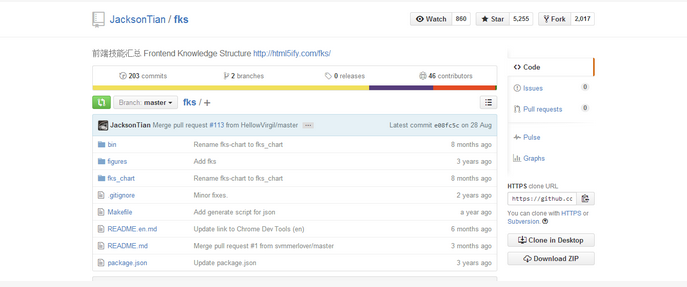
4）低流量。移动端关注比较多。

**再进一阶 · 工程层面：**

前端项目同样面临软件生命周期的各个环节，首先是代码管理，你必须学会使用Svn和Git。其次是代码的构建，如今前端代码构建已经不是简单的压缩一下了，需要进行依赖管理、模块合并、各种编译，比需要学会使用Grunt、Gulp等前端构建工具。

**然后呢？**

以上内容只是简单说了前端学习的顺序。前端工程师应该有的知识结构请参考这里：[JacksonTian/fks · GitHub](https://link.jianshu.com?t=https://github.com/JacksonTian/fks" \t "_blank)



对于前段开发，基本内容就这些了，可以根据自己的兴趣爱好选择性学习以下内容。

1、交互设计。大公司依然有专业人士搞这些，不过不懂交互的前端一定不是好前端。推荐《简约至上》。

附上超多的交互自学资料：《[交互设计师修炼指南！教你从零开始成为优秀交互设计师](https://link.jianshu.com?t=http://www.uisdc.com/interaction-designers-guide" \t "_blank)》

2、后端。

应该说前段工程师必须至少了解一门后端语言，不过如果爱好也可深入学习，入手难度比较低的应该是PHP了。这部分由可分为基于页面，基于框架两种。大型项目都是基于框架开发的，建议至少了解一个MVC框架，比如PHP的Ci、Yii、Yaf 等，好还框架的设计思想都大同小异。如今NodeJs在大公司已经得到普遍的使用，推荐大家使用在Node上使用Express框架做一些后端服务的开发。

作者：毛毛家的大熊  
链接：https://www.jianshu.com/p/0b909aca6bed  
來源：简书  
简书著作权归作者所有，任何形式的转载都请联系作者获得授权并注明出处。