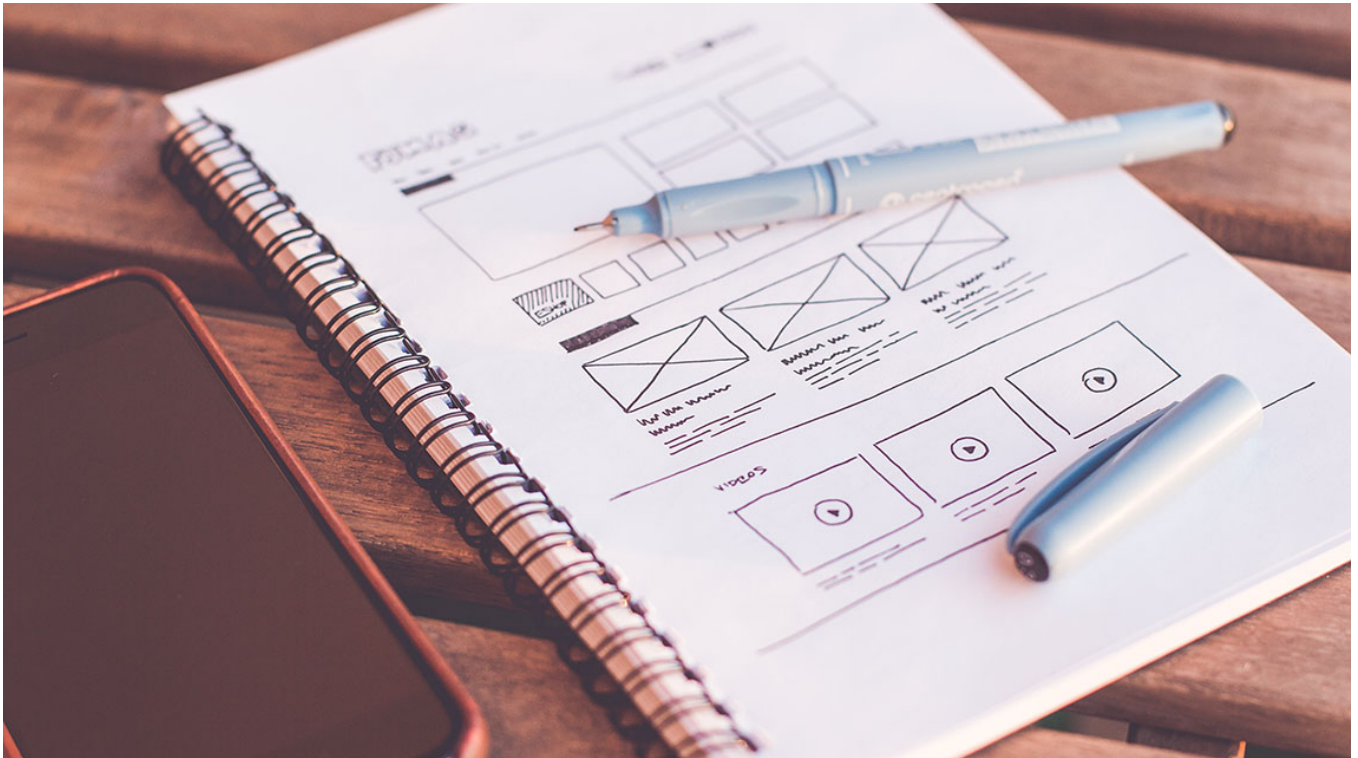


# 程序员练级攻略 ( 2018 ) : UI/UX设计

2018-08-07 陈皓



## 程序员练级攻略 ( 2018 ) : UI/UX设计

朗读者：柴巍 14'58" | 6.86M

上面的技术都讲完了，前端还有一个很重要的事就是设计。作为前端人员，我们有必要了解现在的一些知名且流行的设计语言或是一些设计规范或是设计方法，学习它们的设计思想和方法，有助于我们拓宽眼界、与时俱进。我并不觉得这些内容是设计师要学习的，如果你要成为一个前端程序员，那么学习这些设计上的东西可以让你有更好的成长空间。

对于学习设计的新手来说，推荐看看 [7 steps to become a UI/UX designer](#)，这是一篇很不错的让新手入门的文章，非常具有指导性。首先，你得开始学习设计的一些原则和套路，如配色、平衡、排版、一致性等。还有用户体验的 4D 步骤——Discover、Define、Develop 和 Delivery。然后，开始到一些网站上找灵感。接下来，是到不同的网站上读各种文章和资源，开始学习使用设计工具，最后是找人拜师。此外，其中还链接了其它一些不错的文章、网站、博客和工具。我认为，这篇文章是一篇很不错的设计师从入门到精通的练级攻略。

虽然有这么一个速成的教程，但我觉得还是应该系统地学习一下，所以有了下面这些推荐。

## 图书和文章推荐

先推荐几本书。

- [Don't Make Me Think](#)，这是我看的的第一本和设计相关的书。这本书对我的影响也比较深远。这本书践行了自己的理论，整本书短小精悍，语言轻松诙谐，书中穿插大量色彩丰富的屏幕截图、趣味丛生的卡通插图以及包含大量信息的图表，使枯燥的设计原理变得平易近人。
- [Simple and Usable Web, Mobile, and Interaction Design](#)，中文版译名为《简约至上》。本书作者贾尔斯（Giles）有 20 多年交互式设计的探索与实践。提出了合理删除、分层组织、适时隐藏和巧妙转移这四个达成简约至上的终极策略，讲述了为什么应该站在主流用户一边，以及如何从他们的真实需求和期望出发，简化设计，提升易用性。
- [Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules](#)，中文版译名为《认知与设计：理解 UI 设计准则》。这本书语言清晰明了，将设计准则与其核心的认知学和感知科学高度统一起来，使得设计准则更容易在具体环境中得到应用。涵盖了交互计算机系统设计方方面面，为交互系统设计提供了支持工程方法。不仅如此，这也是一本人类行为原理的入门书。
- [Designing Interfaces: Patterns for Effective Interaction Design](#)，中文版译名为《界面设计模式》。这本书开篇即总结了“与人有关”的各类问题，为读者提供了界面设计总体思路上的指引，帮助读者举一反三。然后，收集并分析了很多常用的界面设计模式，帮助读者理解在实现级别的各种常用解决方案，将它们灵活地运用到自己的设计中。

除了上面的这几本书，还有下面的这几篇文章也是很不错的，推荐一读。

- [The Psychology Principles Every UI/UX Designer Needs to Know](#)，这篇文章讲述了 6 大用户界面用户体验设计的心理学原则。
- [18 designers predict UI/UX trends for 2018](#)，我倒不觉得这篇文章中所说的 UI/UX 是在 2018 年的趋势，我反而觉得，这 18 条原则是指导性的思想。
- [The Evolution of UI/UX Designers Into Product Designers](#)，这篇文章是 Adobe 公司的一篇博客，其在回顾整个产品设计的演化过程中有一些不错的思考和想法，并提供了一些方法论。

## 原子设计 ( Atomic Design )

在 2013 年网页设计师布拉德·弗罗斯特 ( Brad Frost ) 从化学中受到启发：原子 ( Atoms ) 结合在一起，形成分子 ( Molecules )，进一步结合形成的生物体 ( Organisms )。布拉德将这个概念应用在界面设计中，我们的界面就是由一些基本的元素组成的。

乔希·杜克 ( Josh Duck ) 的 "HTML 元素周期表" 完美阐述了我们所有的网站、App、企业内部网、hoobadyboops 等是如何由相同的 HTML 元素组成的。通过在大层面 ( 页 ) 和小层面 ( 原

子)同时思考界面,布拉德认为,可以利用原子设计建立一个适应组件的动态系统。

为什么要玩原子设计,我认为,这对程序员来说是非常好理解的,因为这就是代码模块化重用化的体现。于是,你就是要像搭积木一样开发和设计网页,当你把其模块化组件化了,也更容易规范整体的风格,而且容易维护.....这些都意味着你可以更容易地维护你的代码。所以,这个方法论导致了 Web 组件化的玩法。这是设计中非常重要的方法论。

关于这个设计方法论,你可以阅读一下下面这几篇文章。

- [Atomic Design 原子设计 | 构建科学规范的设计系统](#)
- [网页设计: Atomic Design 简介及工作实例](#)

但是,真正权威的地方还是布拉德·弗罗斯特的电子书、博客和实验室,可以从中获取更多的信息。

- [电子书: Atomic Design by Brad Frost](#) 是布拉德·弗罗斯特写的一本书。
- [博客: Atomic Design](#) 是布拉德·弗罗斯特的博客。
- [实验室: Pattern lab](#) 是布拉德·弗罗斯特依照这个设计系统所建立的一套工具,可以前往 Pattern Lab 的 [GitHub](#) 来试试 Atomic design。

接下来是关于这个设计方法和 React.js 框架的几篇文章。

- [Atomic Design with React](#)
- [Atomic Components: Managing Dynamic React Components using Atomic Design](#)

## 设计语言和设计系统

下面来介绍一下设计语言和设计系统。

### Fluent Design System

[Fluent Design System](#) 中文翻译为流畅设计体系,是微软于 2017 年开发的设计语言。流畅设计是 Microsoft Design Language 2 的改版,其中包含为所有面向 Windows 10 设备和平台设计的软件中的设计和交互的指导原则。

该体系基于五个关键元素:光感、深度、动效、材质和缩放。新的设计语言包括更多对动效、深度及半透明效果的使用。过渡到流畅设计体系是一个长期项目,没有具体的完成目标,但是从创作者更新以来,新设计语言的元素已被融入到个别应用程序中。它将在未来的 Windows 10 秋季创作者更新中更广泛地使用,但微软也表示,该设计体系不会在秋季创作者更新内完成。

微软于 2017 年 5 月 11 日的 Microsoft Build 2017 开发者大会上公开了该设计体系。

- [What's new and coming for Windows UI: XAML and composition](#) , 从概念上讲了一下 Fluent Design System 的各个部分。
- [Introducing Fluent Design](#) , 介绍了 Fluent Design System 的各个部分。

还有 Build 2018 上的一些微软的 YouTube 分享。

- [Fluent Design: Evolving our Design System : Build 2018](#)
- [Microsoft Build 2018 - Fluent Design System Demo](#)
- [Microsoft Build 2018 - Fluent Design System Evolution](#)
- [Fluent Design System inside of Microsoft: Office : Build 2018](#)

## Material Design

[Material Design](#) 中文翻译为质感设计，或是材质设计、材料设计。这是由 Google 开发的设计语言。扩展于 [Google Now](#) 的 " 卡片 " 设计，Material Design 基于网格的布局、响应动画与过渡、填充、深度效果（如光线和阴影）。设计师马蒂亚斯·杜阿尔特（Matías Duarte）解释说：" 与真正的纸张不同，我们的数字材质可以智能地扩大和变形。材质具有实体的表面和边缘。接缝和阴影表明组件的含义。"Google 指出他们的新设计语言基于纸张和油墨。

Material Design 于 2014 年的 Google I/O 大会上发布（参看 [Google I/O 2014 - Material witness: How Android material applications work](#)）。其可借助 v7 appcompat 库用于 Android 2.1 及以上版本，几乎支持所有 2009 年以后制造的 Android 设备。随后，Material Design 扩展到 Google 的网络和移动产品阵列，提供一致的跨平台和应用程序体验。Google 还为第三方开发人员发布了 API，开发人员可将质感设计应用到他们的应用程序中。

除了到 [官网](#) 学习 Material Design，你还可以访问 [Material Design 中文版](#) 来学习。

另外，Wikipedia 上有一张 [Material Design 实现的比较表](#)，供你参考。

下面是几个可供你使用的 Material UI 的工程实现。

- [Material Design Lite](#)，这是 Google 官方的框架，简单易用。
- [Materialize](#)，一组类似于 Bootstrap 的前端 UI 框架。
- [Material-UI](#) 是基于 Google Material Design 的 React 组件实现。
- [MUI](#) 是一个轻量级的 CSS 框架，遵循 Google 的 Material Design 设计方针。

## 其它公司

接下来再来推荐其它几家公司的设计语言。



- [苹果公司的设计指南](#)，在这个网站有苹果的各种设备的设计规范和指导，一方面可以让你的 App 能和苹果的 UI 融合在一起，另一方面，你也可以从中看到苹果的审美和思维方式。
- [IBM 公司的设计语言](#)，我们总觉得 IBM 公司是一家比较传统的没有新意的公司，但是并不是这样的。IBM 公司的这个设计语言的确比较出众。所以，在这里推荐一下。
- [Salesforce 公司的 Lightning Design System](#)，是在 Salesforce 生态系统中用于创建统一 UI 的设计模式、组件和指南的集合，是一个企业级的产品。
- [Facebook Design - What's on our mind?](#)，Facebook 的设计师们收集的一系列的文、视频和资源。很不错哦。

## 动画效果设计

我认为，要了解 Web 动画效果设计的第一步，最好的地方是 [CodePen](#)。这个网站不只是让人分享 HTML、CSS 和 JavaScript 代码的网站。其中也有很多分享样例都和动画效果有关。这个网站可以让你对动画效果有一些感性认识，当然还有代码供你参考。

接下来，我们要了解动画效果设计的一些方法。基本上来说，动画设计都会受 "[动画的 12 项基本法则](#)" 的影响，这个方法论源自于迪士尼动画师奥利·约翰斯顿 (Ollie Johnston) 和弗兰克·托马斯 (Frank Thomas) 在 1981 年所出的《The Illusion of Life: Disney Animation》一书。这些法则已被普遍采用，至今仍与制作 3D 动画法则有关联。这里还有一篇文章 "[Understand the 12 principles of animation](#)" 是对这个法则的解读和理解。

除此之外，还有几个动画设计指南和相关文章供你参考和学习。

- [6 Animation Guidelines for UX Design](#)。这是 Prototypr 公司的一个指南，其中主要指出，动画效果不是为了炫配，而是能让你的 UI/UX 能活起来，自然，不消耗时间，并且是生动故事型的动画效果。其中还推荐了如下几篇很不错的文章。
  - [Transitional Interfaces](#)
  - [UI Animation and UX: A Not-So-Secret Friendship](#)
  - [Invisible animation](#)
  - [Creating Usability with Motion: The UX in Motion Manifesto](#)
- [Designing Interface Animation](#)，这篇文章同样说明，任何一个小动画都是要讲一个微故事的，而且这些微故事会和你的品牌和产品理念相融合。动画会给人更深的印象，让人们更容易记住你。这篇文章主要是讲品牌动画。
- [Animation principles in motion design](#)，这篇文章有点像设计模式，给了一些动画效果的套路和演示。

- [Creating Usability with Motion: The UX in Motion Manifesto](#)
- [Integrating Animation into a Design System](#)
- Great UI/UX Animations 是设计师丹尼尔 ( Daniel ) 收集的一些很不错的动画，可以给你一些灵感。
  - [Great UI/UX Animations 第一组](#)
  - [Great UI/UX Animations 第二组](#)

## 相关资源

下面分享一下 UI/UX 设计的相关资源。文章资源主要有以下这些。

## 文章资源

- [Web Designer News](#)，一个文章聚合的网站。除此之外，还有两个文章聚合网站，你也可以订阅。一个是[Designer News](#)，另一个是[Reddit Web Design](#)。
- [Marvel Blog](#)，Marvel 团队的博客。
- [The Next Web](#)，内容主要涵盖国际技术新闻、商业和文化等多个方面。
- [Medium - Design](#)，Medium 现在已经成为一个好文章的集散地了，这个地方必去。
- [Smashing Magazine](#)，这个地方是给专业的 Web 设计师和程序员的。不但有设计还有 HTML、CSS 和 JavaScript 等各种资源。
- [Sitepoint](#)，这个网站上也有很多不错的给 Web 前端程序员和设计师看的文章（当然，给程序员看的有点简单了，我觉得更像是让设计师来学写程序的网站）。

## 设计收集

接下来推荐一些优秀设计的聚集地。

- [Awwwards](#)，这个网站给一些设计得不错网站的评分，在这里你可以看到很多设计不错的网站。
- [One Page Love](#)，就是一个单页的网页设计的收集。
- [Inspired UI](#)，移动 App 的设计模式。
- [Behance](#)，这个地言有很不错的很有创意的作品。
- [Dribbble](#)，这应该是设计师都知道也都爱去的网站。除了你可以看到一些很不错的作品外，你还可以在这里看到很多不错的设计师。

- [UI Movement](#)，也是个设计的收集网站，上面有很多很不错的 UI 设计，大量的动画。虽说会像抖音一样，让你不知不觉就看了好几小时，但是它比抖音让你的收获大多了。

## 小结

总结一下今天的内容。我并不认为 UI/UX 设计这些内容只是设计师要学习的，如果你要成为一个前端程序员，那么学习这些设计上的东西可以让你有更好的成长空间。首先，我推荐了一些图书和文章，让你更好地了解经典的设计原则和指导思想。

然后介绍了原子设计，以及深入学习和理解这一设计方法论的图书、文章和其他相关资源。最后分享了当下主流和知名公司中在用的设计语言和设计系统，并给出了大量的学习资源，推荐了一些优秀设计的聚集地。相信通过学习这些内容，你在 UI/UX 设计方面不仅能收获方法，还能获得非常多的灵感。

下篇文章是程序员练级攻略 2018 版高手成长篇的最后一篇，我将推荐大量有价值的技术资源，这些内容将会为你后续的学习和成长提供很大的助力。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 \(前端方向\)](#)
- [UI/UX 设计 \(前端方向\)](#)
- 技术资源集散地

## 左耳听风，分享有赏

8月1日 – 8日，老用户邀请好友订阅  
可享受 **¥50** 现金返现

多邀多的，上不封顶，立即提现



版权归极客邦科技所有，未经许可不得转载

..... 通过留言可与作者互动 .....