

专业技术引导实践 详尽案例精彩纷呈



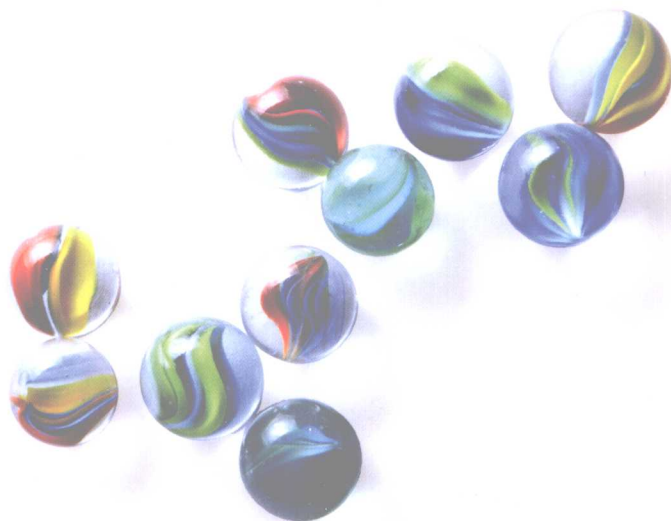
JavaScript

完全解析

Simply JavaScript

[加] Kevin Yank 著
Cameron Adams

黄少年 施游 田支斌 译



JavaScript

完全解析

[加] Kevin Yank 著
Cameron Adams

黄少年 施游 田支斌 译

人 民 邮 电 出 版 社

北 京

图书在版编目 (CIP) 数据

JavaScript完全解析 / (加) 扬克 (Yank, K.), (加) 亚当斯 (Adams, C.) 著; 黄少年, 施游, 田支斌译. —北京: 人民邮电出版社, 2009.6
ISBN 978-7-115-20728-9

I. J… II. ①扬…②亚…③黄…④施…⑤田… III. JAVA语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字 (2009) 第057748号

版 权 声 明

Simply JavaScript

Kevin Yank Cameron Adams

Copyright ©2007 SitePoint Pty. Ltd.

All Rights Reserved.

本书由澳大利亚 SitePoint Pty. 公司授权人民邮电出版社出版。未经出版者书面许可, 对本书的任何部分不得以任何方式或任何手段复制和传播。

版权所有, 侵权必究。

JavaScript 完全解析

-
- ◆ 著 [加] Kevin Yank Cameron Adams
译 黄少年 施游 田支斌
责任编辑 刘映欣
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京顺义振华印刷厂印刷
 - ◆ 开本: 800×1000 1/16
印张: 17.75
字数: 385 千字 2009 年 6 月第 1 版
印数: 1—3 500 册 2009 年 6 月北京第 1 次印刷

著作权合同登记号 图字: 01-2008-1173 号

ISBN 978-7-115-20728-9/TP

定价: 39.00 元

读者服务热线: (010)67132705 印装质量热线: (010)67129223

反盗版热线: (010)67171154

内 容 提 要

本书从 JavaScript 基本语法开始，逐一讲解了 JavaScript 中的各种常见特性，同时详细介绍了调试方法及 Ajax 等新技术。本书将指出使用 JavaScript 的正确方法，并通过讲解在专业网站中得以充分应用的实例，使读者不仅能够编写 JavaScript 代码，而且有能力理解别人编写的代码，甚至能够从以前的程序中发现有害的、弊大于利的代码段。

不管是 JavaScript 初学者还是已经有经验的技术人员，通过学习本书都能够掌握阅读和编写 JavaScript 的方法和技能，使自己具备实战的能力。对于那些已经了解了 HTML 和 CSS，想通过 JavaScript 增强网页效果的 Web 开发人员而言，本书是很好的指南性书籍。

谨以本书献给 Jessica，我生活中的伴侣，你是我酒中的柠檬。

——Kevin

Lisa，如果没有你的支持，我无法完成本书的写作。我只希望回报给你的爱和支持与你给予我的一样多。

——Cameron

前言

从表面上看，JavaScript 是一门简单的程序设计语言，它可以在浏览器显示网页的同时对网页进行实时修改。掌握它到底有多难呢？听起来好像是一个下午就可以解决的事情一样。

但是 JavaScript 的实质要比其表象复杂得多。如果你是 Dr. Who 的粉丝，那么就可能将 JavaScript 称为 TARDIS 式的程序设计语言¹。如果不是，那么就像年轻的男女极客一样，目不转睛地跟着我一起来学习吧！

所有人都跟过来了吗？Jimmy，放下你手中的机器人 Daleks 吧。

正如前面所讲，JavaScript 听起来应该是一门简单的程序设计语言。然而，在它 10 年的发展历程中，使用 JavaScript 的最佳方式似乎也随着时间的变化而变化；并且到处都可以看到关于如何编写优质 JavaScript 代码的建议，如“按照这种方式吧，它将运行得更快”、“用这段代码吧，它可以在更多的浏览器中正常运行”、“远离这个特性吧，它将引起内存泄漏”等。

有很多 JavaScript 书籍（其中一部分书籍的作者在业界非常受人尊敬）只教了一些解决简单问题的简单方法。于是到目前为止，当这些读者需要独自解决一个实际问题的时候，仍然会感觉到无从下手。当他们花费九牛二虎之力从网页上搜索到一个所需的实例时，却无法理解这些 JavaScript 代码的含义。这是因为读者所购买的书没有包含这门语言中多数真正有用的特征，如对象字面量（object literals）、事件监听器（event listeners）以及闭包（closures）。

本书的目标和其他书籍不一样。从第一页开始，本书就将指出使用 JavaScript 的正确方法。通过完成在专业网站中得以充分应用的实例，大家不仅能够编写 JavaScript 代码，而且有能力理解别人编写的代码，甚至能够发现有害的、弊大于利的老式代码。

通观整本书，除了基本知识的介绍之外，还增加了很多额外的内容，特别是介绍了一些新的、功能强大的 JavaScript 开发技术（如 Ajax），它们正在改变着网站的面貌。有一些章节中还专门探讨了 JavaScript 程序库的新成果，如 jQuery、Prototype、Yahoo!UI 和 Dojo。这都使得本书成为能涵盖这些强大、省时技术的 JavaScript 入门书籍。

所有这些因素使得本书的编写更加困难，但这也是出版社支付更多稿酬的原因。

¹ 译者注：Dr. Who 是 BBC 电视台所播放的一部电视剧中的科学家，Tardis 是他发明的时光机，现在泛指内部比外部复杂得多的事物。

读者对象

无论你是从来没有阅读过 JavaScript 代码的初学者，还是已经阅读过大量非预期的 JavaScript 代码，本书都将让你理解 JavaScript 的工作机制。

本书假设读者已经可以能够熟练地使用超文本标记语言(HTML)和层叠样式表(CSS)来完成网页设计。读者不必成为这些语言的专家，但我们相信 JavaScript 将成为又一个谜团。对基本的网页设计技术掌握得越好，就越能发挥出 JavaScript 在增强网页效果方面的优势。

如果需要复习 HTML 和 CSS 知识，这里极力推荐大家阅读 *Build Your Own Web Site The Right Way Using HTML&CSS* (中译版《HTML 与 CSS 网站设计实践之旅》)。

本书内容

- 第 1 章：Web 的三层结构

学习 JavaScript 时最重要的是搞清楚何时最适合使用它，以及何时只需使用 HTML 和 CSS 技术就能提供一个有效的解决方案。在深入学习 JavaScript 之前，本书将花少量篇幅来回顾一下如何使用 HTML 和 CSS 来构建 Web 站点，并了解怎样将 JavaScript 技术融合进来。

- 第 2 章：JavaScript 编程

JavaScript 是一种程序设计语言。要使用这门语言就必须了解计算机程序的工作方式，从某种程度上说就是要像计算机那样进行思考。本章将介绍如语句、变量、表达式、循环、函数和对象等基本概念，这些都是在编写任何 JavaScript 程序时需要具备的基础知识。

- 第 3 章：文档访问

虽然有些人会出于兴趣编写独有的 JavaScript 代码，让你不想对其进行彻夜研究。作为一个专业的网页开发者，经常会希望借助文档对象模型 (Document Object Model, DOM) 来对网页内容做些改变。幸运的是，本书将专门用一章的篇幅来说明如何完成这些改变。

- 第 4 章：事件

这是本书到目前为止最重要的部分，在此将介绍如何编写与用户对网页进行交互相关的 JavaScript 代码。正如读者会看到的那样，这可以通过多种方式完成，现在的浏览器对这些方式提供了不同程度的支持。

- 第 5 章：动画

的确，花费长篇内容介绍 JavaScript 各种微小的增强功能，肯定不会让读者满足，除非是介绍一些能让页面动起来的東西。本章将讲述一些可能真正应用的动画知识。

- 第 6 章：表单增强

众所周知，表单很烦人。没有人会在早晨刚从床上起来时，就捏得指关节咯咯响，并兴奋地嚷嚷：“今天，我要填一些表单！”。但是，一旦借助本章中介绍的增强技术对表单效果做些改进后，这种情况也是可能发生的。为了多增加一些趣味性，本章还将介绍如何在自己的网页上设计一个可拖动的元素。

- 第 7 章：错误和调试

当使用其他程序设计语言时，如果程序出现错误，计算机通常会抛出一个有效的错误信息，直到修复这个问题为止。然而使用 JavaScript，计算机则会把双臂放在胸前，抱歉地说：“可能你在期望什么发生吧？”。本章将介绍如何修复那些没有实现预期功能的脚本。

- 第 8 章：Ajax

读者或许已经听说过 Ajax，它能使网页看起来像一个桌面应用程序，并且能让不可靠的商业投机看起来像有效的投资。正因为如此，本书也将对它做一些介绍。

- 第 9 章：展望

JavaScript 并不只是拥有一个美好未来，JavaScript 就是未来！当然，读者可能认为这有点言过其实，但是当阅读完本章并了解 JavaScript 能实现的很多令人不可思议的功能之后，或许读者会改变自己的看法。

- 附录 A：JavaScript 核心程序库

随着阅读的深入，书中将给出解决许多常见问题的代码。为了避免大家在每次需要的时候重写代码，本书已经将这些代码收录到一个 JavaScript 程序库中。这样大家就能在自己的项目中复用这些代码，避免大量的编码工作。

附录中对这个 JavaScript 程序库中的所有代码进行了分析和概述，并说明了如何使用它。

本书网站

网站 <http://www.sitepoint.com/books/javascript1/> 为本书提供了支持，它将提供以下资源。

- 源代码包

在阅读本书时，大家将会注意到许多代码清单上给出了文件名。这些文件都可以在源代码包中找到，这个可供下载的 ZIP 文件里完整地收录了本书中所出现的所有示例代码。只要在本书的网站上点击名为 Code Archive（源代码包）的链接就可以下载该文件了。

- 更新和勘误表

没有一本书是没有错误的，细心的读者肯定会在本书中发现至少一个或两个错误。本书网站（<http://www.sitepoint.com/books/javascript1/errata.php>）中名为 Corrections & Typos 的链接中将提供最新的印刷和编码错误信息，并为新的浏览器版本和相关标准提供必要的更新。

致谢

在此我对 SitePoint 网站的共同创始人 Mark Harbottle 和总经理 Luke Cuthbertson 表示衷心的感谢。他们使我在 2006 年后半段能安心坐下来开始我的第二职业的生涯，并使我相信远离 SitePoint 的日常管理事务来写一本书不是最坏的经历。我还欠 Simon Mackie 一杯饮料，因为写这本书的想法是他首先提出的。让我们期待有人来买这本书吧！

非常感谢 Jessica，许多个晚上我都因留在办公室写作而延误了回家的时间。她给了我无限的支持，并忍受了我无数次的晚归。我心中有着无限的歉意。

非常感谢订阅 SitePoint Tech Times 时事通讯 (<http://www.sitepoint.com/newsletter/>) 的读者，感谢他们和我一起分享了最终写进本书的许多想法，同时他们也提供了许多有价值 and 具有挑战性的反馈信息。

——Kevin Yank

我所积累的 JavaScript 知识有很多来源，在此无法一一列举。我能和他人分享这些知识是成百个（即使不是成千个）慷慨的程序员花费了许多宝贵的时间为他人奉献自己所擅长的知识的结果。如果你的确也能出一份力，那么就尽最大努力这样做吧。尽管如此，我还想大声地对 Webmonkey 团队（特别是 Thau 和 Taylor）说句感谢，他们从一开始就给我很多鼓励。同时还想感谢和我一起编码的同事们，当我的工作停滞不前时，他们总是能快速地解决问题或开展讨论，他们分别是 Derek Featherstone、Dustin Diaz、Jonathan Snook、Jeremy Keith、Peter-Paul Koch 以及 Dan Webb。

——Cameron Adams

阅读须知

本书中采用了一些特定的字体和版式，以传递不同的信息。

(1) 代码示例

本书中所有的代码都使用下列字体显示，例如：

```
<h1>A perfect summer's day</h1>
<p>It was a lovely day for a walk in the park. The birds
were singing and the kids were all back at school.</p>
```

如果在本书的源代码包中可以找到它，则其相应的文件名称将放在程序清单的顶部，如下所示：

```
.footer {  
  background-color: #CCC;  
  border-top: 1px solid #333;  
}
```

如果仅仅是源代码包中某文件内容的一部分，则还会加上“节选”标识，如下所示：

example.css (节选)

```
border-top: 1px solid #333;
```

(2) 技巧、提示和警告



嗨，你好！

为读者提供有用的技巧。



啊哈，请注意……

这是一些与当前主题相关的信息，但并不是关键的。读者可以将其视为额外添加的信息简报。



确保始终……

……注意这些要点。



小心！

警告将强调读者在学习过程中可能遇到的任何障碍。

作者简介

Kevin Yank

作为 SitePoint 网站的技术总监，Kevin Yank 一直处于 Web 新技术发展的前沿。这一点，从他编写的 *Build Your Own Database Driven Website Using PHP & MySQL* (<http://www.sitepoint.com/books/phpmysql1/>) 一书中就能够看出，该书现在已经推出了第 3 版。他同时还是 SitePoint Tech Times (<http://www.sitepoint.com/newsletter/>) 的主编，这是一个免费的、双周出版的时事通讯，其全球订阅量已经超过了 15 万份。

除了参加一些会议、拜访朋友以及回到位于加拿大的家之外，Kevin Yank 长期居住在澳大利亚的墨尔本市。他喜欢和 Impro Melbourne 剧团 (<http://www.impromelbourne.com.au/>) 一起参加即兴剧表演，还喜欢飞行。他还有一个名为 “Yes, I’m Canadian” (是的，我是加拿大人) 的个人博客 (<http://yesimcanadian.com/>)。

Cameron Adams

Cameron Adams 同时具备了计算机科学和 10 年以上图形设计的双重背景，从而创造了界面设计的独特方法。他凭借这样的技能游转在设计 and 代码之间，一直致力于创建有趣、创新的网站或应用程序。

他曾供职于大型企业、政府部门、非营利组织及初创公司，深得 Internet 相关工作的要领。除了分配精力在项目上之外，他还经常把自己对 Web 设计的思考发表到博客上，也就是知名的 “The Man in Blue” 博客 (<http://themaninblue.com/>)；同时他还编写了一些书籍，内容涉及 JavaScript、CSS 及 Web 设计。

Cameron Adams 有时居住在墨尔本，其他时间则周游世界，和他的那些极客朋友们探讨设计和编程方面的话题。如果你在酒吧里遇到他，不妨买上一杯 Baileys 并和他打声招呼。

目 录

第 1 章 Web 的三层结构.....1	独立..... 36
1.1 页面分离.....2	2.7 对象..... 37
1.2 三层结构.....3	2.8 实际应用中的低调脚本..... 39
1.2.1 用HTML设置内容.....4	2.9 小结..... 40
1.2.2 用CSS描述页面展现效果.....6	第 3 章 文档访问..... 41
1.2.3 用JavaScript实现行为.....7	3.1 文档对象模型：对 HTML 的映射..... 41
1.3 正确的方法.....7	3.1.1 文本节点..... 43
1.4 JavaScript 程序库.....8	3.1.2 属性节点..... 44
1.5 让我们开始吧.....9	3.2 访问所需的节点..... 45
第 2 章 JavaScript 编程.....10	3.2.1 通过ID属性查找元素..... 45
2.1 运行 JavaScript 程序.....10	3.2.2 通过标签名称查找元素..... 47
2.2 语句：给浏览器看的最小程序块.....13	3.2.3 通过类名查找元素..... 50
2.3 注释：给程序员看的最小程序块.....13	3.2.4 遍历DOM树..... 53
2.4 变量：为程序存储数据.....14	3.2.5 操作元素属性..... 56
2.5 条件和循环：控制程序流.....23	3.3 更改样式..... 57
2.5.1 条件：做出选择.....24	3.4 实例：建立带底纹的表格..... 62
2.5.2 循环：使重复代码最少化.....29	3.4.1 查找所有类名为dataTable的表格..... 63
2.6 函数：编写后续将使用的代码.....33	3.4.2 获取每个表格的行..... 63
2.6.1 参数：向函数传递数据.....33	3.4.3 为偶数行添加类alt..... 65
2.6.2 返回语句：函数的数据输出.....35	3.4.4 合并全部代码..... 65
2.6.3 作用域：保持变量相互	3.5 程序库简介..... 67
	3.5.1 Prototype程序库..... 67
	3.5.2 jQuery程序库..... 68
	3.5.3 Dojo程序库..... 69
	3.6 小结..... 70

第4章 事件	71	5.3 再谈富提示条	119
4.1 重要的历史	71	5.4 用新风格实现老式动画	120
4.2 事件句柄	72	5.5 基于路径的运动	124
4.2.1 默认操作	75	5.5.1 二维空间内的动画效果	130
4.2.2 this关键字	76	5.5.2 创建更真实的运动效果	132
4.2.3 事件句柄中存在的问题	78	5.5.3 进一步学习	136
4.3 事件监听器	79	5.6 再谈折叠菜单	136
4.3.1 默认行为	81	5.6.1 如何为折叠菜单添加动画效果	136
4.3.2 事件传送	83	5.6.2 代码修改	137
4.3.3 this关键字	86	5.7 程序库简介	143
4.3.4 IE的内存泄漏问题	88	5.8 小结	145
4.3.5 合并全部代码	88	第6章 表单增强	146
4.4 实例：富提示条	90	6.1 HTML DOM 扩展	147
4.4.1 静态页面	91	6.1.1 实例：创建关联的表单域	148
4.4.2 生成提示条	91	6.1.2 实例：级联菜单	156
4.4.3 核心方法	93	6.2 表单验证	165
4.4.4 动态样式	96	6.2.1 截获表单提交	166
4.4.5 合并全部代码	97	6.2.2 正则表达式	167
4.5 实例：折叠菜单	98	6.2.3 实例：可复用的验证脚本	171
4.5.1 静态页面	99	6.3 自定义表单控件	175
4.5.2 核心方法	100	6.4 程序库简介	186
4.5.3 动态样式	102	6.4.1 表单验证	187
4.5.4 合并全部代码	103	6.4.2 自定义控件	188
4.6 程序库简介	109	6.5 小结	189
4.7 小结	110	第7章 错误和调试	190
第5章 动画	111	7.1 什么也没有发生	191
5.1 动画原理	111	7.2 常见错误	195
5.2 用JavaScript 控制时间	112	7.2.1 语法错误	195
5.2.1 在setTimeout函数中使用变量	115		
5.2.2 停止计时器	117		
5.2.3 建立重复计时器	118		
5.2.4 停止重复计时器	119		

7.2.2 运行错误	199	8.6 小结	238
7.2.3 逻辑错误	201	第9章 展望	239
7.3 用 Firebug 调试	204	9.1 使 Web 变得更加丰富多彩	240
7.4 小结	209	9.1.1 更简单的浏览方式	240
第8章 Ajax	210	9.1.2 更好的视觉效果	241
8.1 XMLHttpRequest: 处理小 内容块	210	9.1.3 独特的交互机制	242
8.1.1 创建XMLHttpRequest 对象	211	9.2 富互联网应用	244
8.1.2 访问服务器	213	9.3 摆脱网页的 JavaScript	247
8.1.3 数据处理	216	9.4 程序库简介	248
8.2 关于屏幕阅读器	217	9.4.1 Dojo	249
8.3 Ajax 实战	218	9.4.2 Google Web Toolkit	251
8.4 基于 Ajax 的无缝表单提交	227	9.5 小结	251
8.5 程序库简介	234	附录 A JavaScript 核心程序库	253
8.5.1 Prototype	235	A.1 对象	253
8.5.2 Dojo	236	A.2 事件监听器方法	254
8.5.3 jQuery	236	A.3 脚本启动	262
8.5.4 YUI	237	A.4 CSS 类管理方法	264
8.5.5 MooTools	237	A.5 获取派生的样式	265
		A.6 完整的程序库	265

第 1 章

Web 的三层结构

从前有……小读者们马上会说，“一个国王！”不对，小朋友们，你们错了，从前有一段木头……

——《木偶奇遇记》

没有 JavaScript 也能做很多事情。使用超文本标记语言 (HTML¹)，就能做出一个详细描述页面内容的复杂文档。而使用层叠样式表 (CSS)，就能用各种不同的方式来呈现页面内容，例如在单个颜色的基础上添加各种细微的变化效果，用图像代替文本以实现更具吸引力的效果。

然而，不论如何装扮页面，HTML 和 CSS 也仅能实现漂亮的静态网页，这种网页和百货商店里的服装模特类似，最多只能实现一些小的动画效果，当鼠标指针靠近它时，它会不停地抖动！而使用 JavaScript 就可以让这种像木偶一样笨拙的页面变得生动起来，而读者也将

从 Web 设计的菜鸟变成高手。

然而，读者创建的网页是像时装模特一样步态优雅，还是像怪物 Dr. Frankenstein 那样步履蹒跚，更多地取决于网页中 HTML 和 CSS 源代码以及赋予网页生命的 JavaScript 代码本身的质量。

¹ 原文注：在本书中，我们将 HTML 和 XHTML 都统称为 HTML。使用 HTML 还是 XHTML 完全取决于读者自己，而这跟 JavaScript 没有多大关系。在此必须提醒读者，本书中的 HTML 代码是严格遵守 XHTML1.0 标准的。

因此，在学习创造奇迹之前，让我们花点时间来回顾一下如何创建一个具有良好外观和内部结构的网站，再来了解 JavaScript 为什么适用。

1.1 页面分离

就在不久之前，专业网页设计人员都还会愉快地将 HTML 代码、CSS 代码及 JavaScript 代码堆砌在一个称为 `index.html`¹ 的文件里，并将它称之为网页（如图 1.1 所示）。现在，仍旧可以用这种方法来创建网页，但那样就要做好忍受同事们嘲笑此网页的心理准备了。

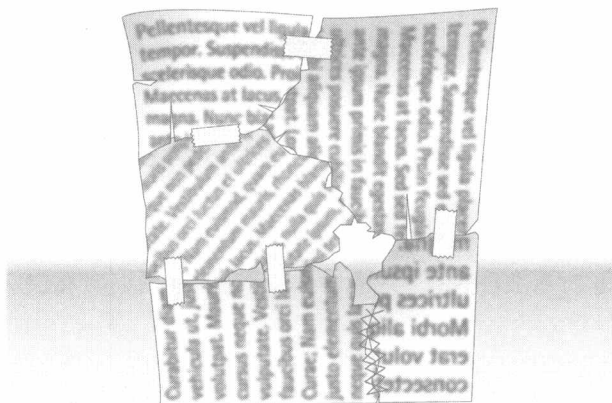


图 1.1 堆砌的网页

在这个过程中，网页设计人员认识到，这些被放在一个 Web 页面中的代码完成了以下三件基本事情：

- 描述页面的内容；
- 指定内容的展现形式；
- 控制内容的行为。

他们也认识到将网页中的上述 3 种代码分离开会使网页制作变得更加容易（如图 1.2 所示），同时这种方法也可以让网页在某些不利条件下更好地运行，例如当用户的浏览器禁用 JavaScript 脚本时。

计算机极客²很早就认识到了这一点，甚至已经为这种原则赋予了一个怪异的名称：关注

¹ 原文注：那些被微软洗过脑的网页设计人员，肯定会将其称为 `default.htm`。

² 译者注：Geek，也译为奇客，通常指具有极高的技术水平，对计算机与网络的痴迷有时会达到不正常状态的计算机高手。

点分离。

然而，意识到是一回事，行动起来却是另一回事，特别是对计算机极客之外的人而言。我虽然是一名计算机极客，但却经常忍不住以错误的方式行事。

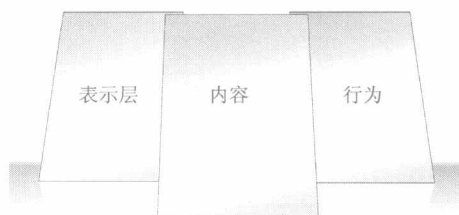


图 1.2 关注点分离

当我突然发现一个很不错的想法时，例如为网页文本加一点灰色阴影，或者将文本向左移动一点，又或者将网页的背景设置为最近在 SitePoint 办公室里拍摄的大头像等，我总会快乐地直接编辑那些描述网页内容的 HTML 代码，而且还是马上动手。

然而哪种方法更容易呢？是打开一个独立的 CSS 文件来修改页面的样式表，还是直接把这些样式属性添加到已经编写好的 HTML 代码中？

与编写功能函数时一样，确保不同类型的代码相互分离也是一项铁的原则。一旦大家理解了这样做的好处，那么不管是顺利还是遭遇挫折，都能够继续坚持下去了。

1.2 三层结构

在任何编程活动中，尽可能使不同类型的代码相互分离都是个好主意。这样做使得在将来的项目中重用部分代码变得更加容易，也减少了复制代码的数量，也使得以后发现和修复问题更加容易。

对于 Web 设计而言，决定对代码进行分离还有一个更重要的理由：它能使网页在用户以不同的方式访问时都能够正常应对。

这主要取决于网站的访问者情况，大多数的访问者都会使用对 CSS 和 JavaScript 支持良好的桌面浏览器，但有一些访问者因受公司 IT 政策的制约而不得不使用老版本的浏览器，或者禁用了某些特性（如 JavaScript）。

视力不好的用户经常通过屏幕阅读器或屏幕放大软件来浏览网页，对他们来说，那些华而不实的视觉设计可能就是一种障碍而非有效的帮助。

还有一些用户甚至根本不会直接访问网站，而是选择用 RSS 订阅或其他类似方式来浏览网页。而创建订阅服务功能时，通常希望去掉所有 JavaScript 及 CSS 代码，而将纯粹的 HTML