

HTML5

用户指南

INTRODUCING
HTML5

(美) Bruce Lawson Remy Sharp 著
刘红伟 等译



机械工业出版社
China Machine Press

HTML5

用户指南

INTRODUCING

HTML5

Bruce Lawson Remy Sharp 著
刘红伟 等译



机械工业出版社
China Machine Press

本书共分为10章,系统全面地介绍了HTML 5规范的核心内容,以及这些内容在当前浏览器中的支持情况,并告知开发者如何在当前的环境下应用这些功能,开发出漂亮的Web应用。本书短小精悍,但是信息量巨大;本书注重实践,其中的示例代码都具有很强的启发性和可操作性。对于初学者,本书是了解HTML 5的很好的入门材料;而对于想要了解HTML 5知识和应用的有经验的Web开发者,阅读本书更是轻车熟路,能进一步加深对Web开发和HTML 5的理解。

Authorized translation from the English language edition entitled *Introducing HTML 5* by Bruce Lawson and Remy Sharp, published by Pearson Education, Inc, publishing as New Riders, Copyright © 2010.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanic, including photocopying, recording, or by any information storage retrieval system, without permission of Pearson Education, Inc.

Chinese simplified language edition published by China Machine Press.

Copyright © 2010 by China Machine Press.

本书中文简体字版由美国Pearson Education培生教育集团授权机械工业出版社独家出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

封底无防伪标均为盗版

版权所有,侵权必究

本书法律顾问 北京市展达律师事务所

本书版权登记号:图字:01-2010-6145

图书在版编目(CIP)数据

HTML 5用户指南 / (美)罗森(Lawson, B.), (美)夏普(Sharp, R.)著;刘红伟等译. —北京:机械工业出版社, 2011.1

(UI设计丛书)

书名原文: *Introducing HTML 5*

ISBN 978-7-111-32278-8

I. H… II. ①罗… ②夏… ③刘… III. 超文本标记语言, HTML—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2010)第202160号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:陈佳媛

北京京师印务有限公司印刷

2011年1月第1版第1次印刷

186mm × 240mm • 13印张

标准书号:ISBN 978-7-111-32278-8

定价:39.00元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

客服热线:(010) 88378991; 88361066

购书热线:(010) 68326294; 88379649; 68995259

投稿热线:(010) 88379604

读者信箱:hzsj@hzbook.com

HTML 5是Web的未来，也是Web开发者的新希望。

HTML 5的工作开始于2004年，在 W3C HTML工作组和 WHATWG 的共同努力下，现在正在全面贯彻落实。很多关键的著名厂商都参与了W3C的努力，给予HTML 5极大的支持，其中最具代表性的是4大浏览器厂商：Apple、Mozilla、Opera和Microsoft。还有一系列的有着不同利益和专业技术的其他机构以及个人，也都对HTML 5的工作倾注了极大的热情，包括JavaScript的发明者兼Mozilla的CTO Brendan Eich、Safari和WebKit架构师David Hyatt、CSS的发明者Håkon Wium Lie等知名人士。可以说，HTML 5承载了互联网行业和Web开发者的厚望。

同时，HTML 5也给人们带来了众多惊喜，这包括如下一些优点和新特性：

- 新特性应该基于 HTML、CSS、DOM 和 JavaScript。
- 减少了对外部插件的需求（比如 Flash）。
- 更优秀的错误处理。
- 更多取代脚本的标记。
- HTML 5 应该独立于设备。
- 用于绘画的 canvas 元素。
- 用于媒介回放的 video 和 audio 元素。
- 对本地离线存储的更好的支持。
- 新元素和表单控件。

而这些新特性，正在如今的浏览器最新版本中得到越来越普遍的实现，越来越多的开发者开始学习和使用这些新特性。我们注意到，Web应用程序甚至在发挥和表现着与桌面应用程序类似的功能和性能。我们相信，随着HTML 5的支持和普及，这种趋势会越来越明显。

本书由刘红伟、关志兴、王建勇、毛立涛、闫柳青、姜巧生、沈海峰、李振胜、李秋强等翻译。翻译的过程中，我们深感作者技术功底深厚，语言诙谐幽默，这也给我们的理解和翻译带来了很大的难度，如有不到之处，还请读者批评指正，可通过liuhongwei198004@sina.com与译者交流。

译者

2010年9月

前言

欢迎打开本书。我们是两个开发者，从2008年圣诞节开始研究HTML 5，我们在邮件组列表中体验、参与，并且在学习这门语言的同时尽全力帮助其成形。

由于我们是开发者，所以我们对于构建东西感兴趣。这就是为什么本书关注于HTML 5所能解决的问题，而不是对这门语言进行学术研究。还有一点要注意，尽管Bruce供职于Opera Software，而该公司最早开始的可行性研究工作最终导致了HTML 5的形成，但Bruce在这里不属于任何特定的团队，他的身份是使用该语言的一名开发者。

本书的目标读者

不需要有HTML 5的知识，但我们期望你是一位有经验的（X）HTML读者，熟悉语意标记的概念。你是否很熟悉HTML或XHTML文档类型无关紧要，但是，你应该能够熟练地编写任意一种严格的标记。

尽管你不必是一位JavaScript高手，但是你应该理解它在现代Web开发中扮演着日益重要的角色，并且像DOM和API这样的术语不至于吓得你扔下本书仓皇而逃。

你还在这里？很好。

不适合阅读本书的人

本书不是一本参考书。我们不想以顺序的方式依次介绍每个元素或API，详细讨论每一个元素，之后再继续进行下一个。这些应该是规范的工作，但绝对只是给出最基本的细节。

规范做不到的事情是，教授如何使用每个元素或API，或者讲述它们如何在上下文中一起工作。随着新主题的讨论，我们将构建一些示例，并且当有需要注意的新内容出现时再返回来学习。

从本书的书名和内容，你还会看到，要很好地掌握本书并不需要费太大的力气，本书不是面面俱到。详细地解释规范需要900页的篇幅（相比而言，第一份HTML规范只有3页纸那么长），相当于一本中等厚度的书，而且，还需要类似时空穿梭机Tardis的技术（这可能会很酷）或者需要使用显微镜才能看见的字体，你愿意选哪一种呢？

HTML 5表示什么

这听起来似乎是一个很傻的问题，但是，在标准专家中有这样一种日趋强烈的倾向，就是把所有令人兴奋的、新的Web技术都塞入到一个带有HTML 5标签的盒子中。例如，我看到过将SVG（Scalable Vector Graphics，可缩放矢量图形）描述为“HTML 5技术组合之一”，即便它是一个独立的W3C图形规范，而且已经存在6年之久了。

官方的W3C规范有点像是阿米巴虫，这一事实引起了进一步的混淆。这些规范总是一点点地剥离出来，成为自己独立的规范，例如Web Sockets或Web Storage（虽然都源于同一工作组，由相同的项目编辑负责）。

那么，我们在本书中介绍“源自于WHATWG的HTML5和相关规范”（稍后将更详细地介绍WHATWG这一激动人心的缩略语）到底意味着什么呢？我们还给这个领域增加了一点“额外知识”，即Geolocation，它与HTML 5的定义无关，但是，我们包含它只是因为它真的很酷，我们对其感到很兴奋，并且它是编写Web应用程序的、令人激动的技术中的新贵。

HTML 5简史

计算机图书中的历史部分通常会令我们烦恼。要理解如何编写一种新的语言，不需要了解ARPANET的相关知识和HTTP的历史。

尽管如此，了解HTML 5的由来还是很有用的，因为它将帮助你理解为什么HTML 5的某些方面会像现在这样，并且有望预防（或者至少平复）骂出“什么玩意儿！他们为什么这么设计？”的时刻。

HTML 5差点难产

1998年，W3C决定不再继续推进HTML。他们相信，XML是未来。因此，HTML规范在4.01版就冻结了，并且发布了一个新的名为XHTML的规范，它是HTML的一个XML版本，需要一些XML语法规则，如用引号括起属性、结束某些在其他情况下是自结束的标记等，增加了两种内容（实际上是三种，如果你关注HTML Frame，但是我们希望你不要关注它，因为它在HTML 5中已经取消了）。有一种XHTML过渡形式，用来帮助人们转向XHTML严格形式的黄金标准。

这一切进展顺利，它鼓励一代开发者（至少是专业标准的开发者）考虑有效的、结构良好的代码。然而，针对XHTML 2.0规范的工作随后开始了，

这对该语言是一次革命性的改变，因为它为了变得更有逻辑性和更好设计而打破了向后兼容。

然而，Opera的一小部分群众，不相信XML是所有Web开发者的未来。这些人开展了额外的工作，以证明扩展HTML形式且不必违反向后兼容的规范是可行的。该规范最终变成了Web Forms 2.0，随后加入到HTML 5规范中。来自于Mozilla的个人很快加入了这一工作，这一组织由Ian “Hixie” Hickson领导，私自地进行规范研究工作，而Apple则“站在一旁加油助威”，这个小组织自称WHATWG（Web Hypertext Application Technology Working Group，www.whatwg.org）。在规范的WHATWG版本的版权声明中，仍然可以看到这一起源，“©Copyright 2004-C2009 Apple Computer, Inc., Mozilla Foundation, and Opera Software ASA”（注意，你可以使用、复制并创建衍生产品）。

Hickson从Opera跳槽去了Google，在那里，他继续全职担任HTML 5（随后改名为Web Applications 1.0）的编辑。2006年，W3C承认他们在期待该领域迁移到XML（并且经过扩展成为XHTML 2.0）方面过于乐观。Tim Berners-Lee说：“有必要持续地发展HTML。让整个世界转向XML的尝试，包括引用属性值以及在空标记和命名空间中使用斜线，都立即无法工作。”（<http://dig.csail.mit.edu/breadcrumbs/node/166>）。

重新开始工作的HTML工作组通过投票，使用WHATWG的Web Applications规范作为HTML的新版本的基础，并由此开始了奇怪的过程，因为同一规范也由W3C（由IBM的Sam Ruby、Microsoft 的Chris Wilson，以及后来的Microsoft的Ruby、Paul Cotton和Apple的Maciej Stachowiak共同担任主席）和WHATWG同步开发，而Hickson继续担任编辑。

这一过程在几个方面显得很不同寻常。首先是不寻常的开放性，任何人都可以加入WHATWG邮件列表并为此规范做出贡献。每一封邮件都由Hickson或WHATWG核心团队阅读（包括JavaScript的发明者兼Mozilla的CTO Brendan Eich、Safari和WebKit架构师David Hyatt、CSS的发明者和Opera CTO Håkon Wium Lie等知名人士）。

不管这些想法来自何处，以及它们代表谁，或者这些想法最初是从哪里提出的，总之，好的想法就会实现，糟糕的想法就会拒绝。好的想法可以通过Twitter、博客和在线聊天系统获得。

查找规范

由于HTML 5规范由W3C和WHATWG共同开发，因此，该规范存在不同版本。

www.w3.org/TR/html5/是W3C的官方快照，而<http://dev.w3.org/html5/spec/>是最新编辑的草案和可能的修改。

要查看WHATWG的版本，可访问<http://whatwg.org/html5>，但要留意其标题是“HTML5 (including next generation additions still in development)”，这很大程度上包含了实验性的思路，例如<device>这样的元素。不要假设这个文档中的内容已经到处都实现了，或者甚至完全完成了。但这个文档注明了规范在不同浏览器中的实现状态。

还有完整的WHATWG规范的一个单页版本，叫做“Web Applications 1.0”，它整合了WHATWG中的所有内容，位于<http://www.whatwg.org/specs/web-apps/current-work/complete.html>，但是，由于它拥有很多脚本，可能会使你的浏览器崩溃。

有点晕了？http://wiki.whatwg.org/wiki/FAQ#What_are_the_various_versions_of_the_spec.3F列出并介绍了这些不同的版本。

Geolocation并不是一个WHATWG规范，它位于<http://www.w3.org/TR/geolocation-API/>。

2009年，W3C停止了关于XHTML 2.0的工作，并且将资源转向HTML，由此，HTML 5显然赢得了这场理论之战（追求纯设计，即便违反了向后兼容，对阵实用主义且“不要破坏Web”）。HTML 5工作组是由来自各个浏览器厂商的代表组成的，这一情况也很重要。如果厂商不愿意实现规范的某部分（例如，Microsoft不愿意实现<dialog>，或Mozilla反对<bb>），就放弃该部分。Hickson曾经说过：“实际上，浏览器厂商对于规范中的一切拥有最终投票权，因为，如果他们不实现它，那么规范只不过是一部科幻作品。”（<http://www.webstandards.org/2009/05/13/interview-with-ian-hickson-editor-of-the-html-5-specification/>）。很多参与者发现浏览器厂商劫持了“自己的Web”，他们纷纷以此为借口来抱怨。

W3C和WHATWG之间的工作关系并没有像预期的那样顺利。W3C以一种基于民意的方式运作，而Hickson继续扮演他在WHATWG的角色，即一位乐善好施的独裁者（很多人可能会对我们在这里使用“独裁者”一词嗤之以鼻）。可以肯定的是，Hickson对于这门语言应该如何发展有着非常强硬的想法。

HTML 5背后的哲学

HTML 5的背后是一系列确定的设计原则(<http://www.w3.org/TR/html->

design-principles)。HTML 5有3个主要目标：

- 确定当前浏览器能够互操作的行为。
- 首次定义错误处理。
- 将该语言发展为Web应用程序编写者易于使用的形式。

不要破坏已有的Web页面

我们当前开发站点和应用的很多方法，依赖于没有正式文档记录（或至少没有规范化）的功能，它们随着时间的流逝而加入到浏览器中。例如，无数的AJAX驱动的站点的XMLHttpRequest（XHR）功能。该功能由Microsoft发明，随后经过逆向工程并加入到所有其他的浏览器中，但是还没有作为标准规范化（Opera的Anne van Kesteren最终将其作为WHATWG的一部分规范化）。众多站点的如此重要的一个部分，竟然完全留给了逆向工程！因此，HTML 5的首要任务之一，就是将未文档化的功能文档化，以便给Web开发者和浏览器实现者留下较小的猜测工作，从而增加互操作性。

明确地定义浏览器和其他的用户代理应该如何处理无效的标记，这也是必要的。在XML的世界中，这不是一个问题，因为如果浏览器发现一个错误，会要求它停止显示。Web得以快速普及和成功的主要原因之一是（以我们的观点），即便糟糕的代码也能够争取机会让某些或所有的浏览器显示。在Web上发布信息的门槛相当低，但是每款浏览器自由地决定如何显示糟糕的代码。如下的简单内容：

```
<b><i>Hello mum!</b></i>
```

（注意，结束标记匹配错误）在不同的浏览器中产生不同的DOM。不同的DOM可能导致相同的CSS有完全不同的显示，并且，它们可能没必要地增加了编写可跨浏览器运行的JavaScript的难度。一致的DOM对于HTML 5的设计如此重要，因为该语言自身就是根据DOM来定义的。

为了更好的互操作性，不同的浏览器之间保持相同的错误处理是很重要的，因此，即便面对有错误的HTML，也要产生完全相同的DOM。为了做到这点，有必要规范它。正如我们所说的，HTML 5规范印刷出来有900页的厚度，但是，只有300页的内容与Web开发者有关，剩下的内容都是针对浏览器的实现者的，明确告诉他们如何解析标记，即便是糟糕的标记。

Web应用程序

Web上逐渐增多的站点，就是我们所谓的Web应用程序，也就是说，它们模仿桌面应用程序，而不是模仿组成Web的大多数内容的、传统的静态文本—图像—链接文档。例子包括在线字处理程序、照片编辑工具、地图站点

➤ 注意

有一个HTML 5规范只处理和Web开发者相关的方面，它由位于<http://dev.w3.org/html5/markup/>的主资源自动产生。

等。通过JavaScript, 这些应用将HTML 4的功能发挥到了极致。HTML 5为拖曳、服务器发送事件、绘图、视频等指定了新的DOM API。HTML页面通过DOM中的对象将这些接口暴露给JavaScript, 使其很容易使用严格规范的标准而不是几乎没有文档化的hack技术来编写这样的应用。

更为重要的是, 我们需要一个开放的标准 (自由地使用和自由地实现), 能够与Adobe Flash 或Microsoft Silverlight这样的专有性标准竞争。不管你对于这些技术或公司有什么样的想法, 我们相信Web对于社会、商业和通信来说是非常重要的平台, 而不应该掌握在一家厂商的手中。如果Caxton在印刷出版生产技术方面持有专利和垄断的话, 文艺复兴的进展将何其不同?

不要破坏Web

Web上已经有了巨量的页面了, 并且它们势必要继续显示下去。因此, HTML 5 (主要) 是HTML 4的一个超集, 继续定义浏览器应该如何处理、<center>这样的遗留标记, 以及其他这样的表现型标签, 因为数以百万计的Web页面在使用它们。但是开发者不应该再使用它们, 因为它们已经废弃了。对于Web开发者来说, 语义性标记在今天仍然流行, 尽管对于HTML 5是否包含了足够多的语义或过多的元素, 每个读者都会形成自己的结论。

作为额外的好处, HTML 5的明确的解析规则应该可以确保陈旧的页面能够满足互操作性的要求, 因为HTML 5解析器将用于所有的HTML文档 (还没有浏览器默认地配备一个HTML 5解析器, 尽管在编写本书时Firefox已经有了一个实验性的HTML 5解析器, 可以通过about:config命令把偏好设置html5.enable改为true, 从而打开该解析器)。

XML怎么样

HTML 5 不是一种XML语言 (它甚至不是一种SGML语言)。它必须充当text/HTML。如果你需要使用XML的话, 也有一个XML系列版本XHTML 5。它支持所有相同的功能, 但是 (毫不奇怪的是) 需要一种更为严格的语法 (如果你习惯了编写XHTML, 那么这和你以前所编写的完全相同)。它必须是结构良好的XML, 并且必须充当一种XML MIME类型, 即便IE 8及其更早版本都不能处理它 (会为其提供下载而不是显示)。因此, 我们在本书中使用HTML而不是XHTML语法。

HTML 5支持

HTML 5正在快速发展, 该规范已经在2009年10月进入了第一个最终草案, 在此日期之前, 浏览器已经开始实现对HTML 5的支持了 (特别是围绕

API)。随着浏览器开始推出新的功能，HTML 5的支持将同样持续地增加。

在编写本书时，我们已经几次修改章节，已考虑到规范中的修改，而规范现在看上去已经相当稳定了。我们提到“这只在浏览器××中支持”的那些地方，很快将会过时，这真是好事情。

让我们开始动手吧

好了，这就是我们的历史课，还加入了一点点哲学内容。这就是为什么HTML 5有时候固执地与其他规范不一致，对于向后兼容，它常常定义浏览器实际做什么，而不是像一个RFC规范应该做的那样。这就是为什么有时候HTML 5像是一个杂牌组合或折中意见，它就是这样的。如果这就是我们为一个可以互操作的开放Web所要付出的代价，那么Web开发者会说：“实用主义万岁！”

系好安全带了吗？

让我们开始HTML 5的探索之旅吧！

目 录

译者序 前 言

第1章 主要结构

<head>.....	1
使用新的HTML 5结构化元素	5
使用CSS样式化HTML 5	9
何时使用新的HTML 5结构化元素	12
小结	20

第2章 文本

构造主要内容区域	21
添加博客帖子 and 评论	26
使用HTML 5大纲	27
理解WAI-ARIA	43
更多新结构	47
重新定义的元素	51
全局属性	55
本书中没有介绍的功能	58
小结	59

第3章 表单

我们爱HTML，现在它反过来爱我们了	61
新的输入类型	62
新的属性	66
综合应用	71
对遗留浏览器的向后兼容	74
样式化新表单字段和错误消息	75
覆盖浏览器默认	75
使用JavaScript DIY验证	76
避免验证	78
小结	80

第4章 视频和音频

本地多媒体：为什么、是什么、如何做	81
编码解码器	86
滚动定制控件	90
多媒体可访问性	98
小结	100

第5章 画布

画布基础	105
绘制路径	109
使用变换：伪装的像素	110
捕获图像	113
推送像素	116
对画布绘制实现动画	120
小结	125

第6章 数据存储

存储选项	128
Web存储	128
Web SQL数据库	136
小结	146

第7章 离线

拔掉插件：离线进行	147
缓存清单	147
如何提供清单	151
浏览器-服务器过程	151
applicationCache	153
使用清单来检测连接性	155
清除缓存	156
小结	156

第8章 拖放

深入拖动	157
拖动数据的互操作性	161

如何拖动任意元素	163
添加定制的拖动图标	164
可访问性	164
小结	166
第9章 Geolocation	
为访问者标记一个别针	167
API方法	169
抓住你：成功处理程序	170
它到底如何工作：这是魔术	173
小结	174
第10章 Messages、Worker和Socket	
用Messaging API实现聊天	175
使用Web Workers来线程化	177
Web Sockets：使用流数据	188
小结	191
结语	191

尽管HTML 5主要是为了开发交互式应用程序，但还是对标记熟练者和JavaScript爱好者照顾有加。带有新语义的28种新元素可以用在传统的“静态”页面中，还有大量的新的表单控件可以完全取代JavaScript的表单验证。

让我们直接进入代码，并且开始用HTML 5标记一个典型的页面。我们使用<nav>、<header>、<footer>、<aside>和<article>等新的HTML 5结构化元素，将当前的<div>的标记结构转换为一种更具语义的系统。我们将看看这是如何工作的，HTML 5文档是如何有一个明确的大纲，以及为什么（确实）更加“语义化”。

首先，我们需要打开一个代码编辑器，并且输入一个DOCTYPE和初步的信息。

```
<head>
```

首先是DOCTYPE:

```
<!doctype html>
```

这就行了。甚至没有需要大多数Web开发者剪贴的URL。注意，这里没有版本号。就是这样。一条指令从未如此像一句咒语：浏览器需要一个doctype来触发标准模式，并且，这是能够可靠地做到这一点的最短的字符串。

然后，我们需要定义文档的字符编码。不这么做的话，可能会导致不清晰，而且会有真正的安全性风险（参见<http://code.google.com/p/doctype/wiki/ArticleUtf7>）。这应该在文档的前512个字节中。我们推荐UTF-8作为字符编码，除非你有一个极好的理由不使用它：

```
<!doctype html>
```

```
<meta charset=utf-8>
```

仔细地看一下<meta>标签。习惯于编写XHTML的人应该注意3点。首先，这个<meta>比我们目前为止所用过的任何一个都短，我们已经习惯于使用<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">。这种也是有可能用到的，但较短的方式往往更容易

录入，并且也已经随处可用。

你还会注意到，我没有像`charset="utf-8"`一样用引号把属性括起来。也没有像`<meta charset=utf-8 />`一样自结束标签。

HTML 5不是一种XML语言，因此，不需要做这些事情。但是，如果你愿意，也可以这样做。所有下面这些在HTML 5中都同样有效：

```
<META CHARSET=UTF-8>
<META CHARSET=UTF-8 />
<META CHARSET="UTF-8">
<META CHARSET="UTF-8" />
<meta charset=utf-8>
<meta charset=utf-8 />
<meta charset="utf-8">
<meta charset="utf-8" />
<MeTa CHARset=utf-8>
```

为什么语法如此极为松散？答案很简单：如果XHTML作为text/html发送的话，浏览器不在乎其语法，只有XHTML验证器在乎。因此，在HTML 5中喜欢一种形式而胜过另一种形式完全是任意的，并且会导致不遵从该格式的页面成为无效的，尽管它们在任何浏览器中都能很好地工作。因此，HTML 5对于你使用哪一种语法是不置可否的。

选择一种样式并坚持使用它

你可以使用上述格式中的任何一种，但并不意味着你应该将它们都混合在一起使用。这么做将带来一场维护梦魇，在较大的团队中尤其如此。

我们的建议是挑选一种适合你的样式并坚持使用它。选择哪一种无关紧要。Remy喜欢XHTML语法，而Bruce喜欢小写字母、属性简写（因此，使用`controls`而不是`controls="controls"`），并且只有在必要的时候才用引号把属性括起来，例如，给一个元素添加两个类，使用`<div class=important>`而不是`<div class="important logged-in">`。你在本书中将会看到这两种样式，因为我们每个人使用自己最喜欢的样式，所以你需要能够阅读两种样式。

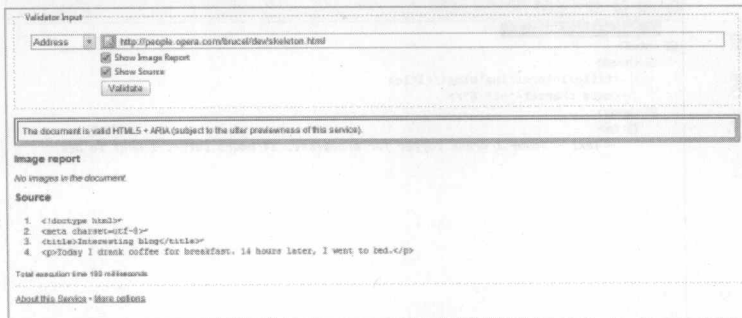
作为一位勇敢的新的HTML 5开发者，你可以自由选择，但是，必须选择，并坚持它。

由于我们在讨论极为松散的语法规则（从XHTML的角度），让我们糊弄一回吧，并且，在添加了文档标题之后，直接来看内容：

```
<!doctype html>
<meta charset=utf-8>
<title>Interesting blog</title>
<p>Today I drank coffee for breakfast. 14 hours later,
I went to bed.</p>
```

如果我们验证这一令人喜欢的博客，会发现它能很好地验证，尽管它没有<html>标签，没有<head>，没有<body>（如图1.1所示）。

图1.1 令人惊讶，没有head、body或html标签，文档能通过验证



这可能就是我在前言中提到的WTF（什么玩意儿）的时刻之一。这三个元素（XHTML开发者，你坐稳了吗？）完全是可选的，因为浏览器会假设它们。在Opera Dragonfly中看一眼浏览器幕后的操作，可以证实这一点（如图1.2所示）。

图1.3使用IE8开发工具展示了这一点。由于浏览器会这么做，所以HTML 5不要求这些标签是必需的。尽管如此，在你的标记中忽略这些元素可能会把你的同事搞晕。此外，<html>也会伤害屏幕阅读器用户，因为这是你设置文档主语言的地方：

```
<html lang=en>
```

图1.2 Opera Dragonfly调试器显示浏览器添加了漏掉的元素

