A BOOK APART

■【英】Jeremy Keith 著

HTML5
网页设计入门必读

HTML5 FOR WEB DESIGNERS

人民邮电出版社
POSTS & TELECOM PRESS

A BOOK APART

■【英】Jeremy Keith 【美】Dan Cederholm 著 ■ 邢薇薇 郭俊飞 王雪 译

HTML5+CSS3 网页设计入门必读

HTML5 FOR WEB DESIGNERS
CSS3 FOR WEB DESIGNERS

人 民 邮 电 出 版 社
北 京

图书在版编目（CIP）数据

HTML5+CSS3网页设计入门必读 / （英）基思（Keith, J.），（美）塞德霍尔姆（Cederholm, D.）著；邢薇薇，郭俊飞，王雪译. -- 北京 : 人民邮电出版社，2014.11

ISBN 978-7-115-34933-0

Ⅰ. ①H… Ⅱ. ①基… ②塞… ③邢… ④郭… ⑤王… Ⅲ. ①超文本标记语言－程序设计②网页制作工具 Ⅳ. ①TP312②TP393.092

中国版本图书馆CIP数据核字(2014)第216662号

版权声明

◆ 著 [英] Jeremy Keith [美] Dan Cederholm

译 邢薇薇 郭俊飞 王 雪

责任编辑 赵 轩

责任印制 彭志环 杨林杰

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 http://www.ptpress.com.cn

北京瑞禾彩色印刷有限公司印刷

◆ 开本：720×960 1/16

印张：12.25

字数：188 千字 2014 年 11 月第 1 版

印数：1-2 500 册 2014 年 11 月北京第 1 次印刷

著作权合同登记号 图字：01-2013-7661 号

定价：39.00 元（全二册）

读者服务热线：(010)81055410 印装质量热线：(010)81055316

反盗版热线：(010)81055315

前言

当我和 Mandy Brown、Jason Santa Maria 成立 A Book Apart 时，我们十分希望对一个主题内容进行整理并成书出版，而且我们觉得，只有一个作者适合这项工作。

网页设计是基于标准的。虽然“real fonts”或 CSS3 也对网页设计产生了深远的影响，但 HTML 的影响是其他方面所无法比拟的。HTML 是开发人员对 W3C 步调和政策不满的结果，它是为网页应用程序（而不仅仅是文件）而设计的，因此，虽然这个新版本的网络语言令网页设计界为之兴奋，但也导致了同等程度的愤怒和困惑。

在编写了关于 DOM 和 JavaScript 的书籍之后，Jeremy Keith 以一种独特的方式阐述了 HTML5，并介绍了对于网页设计师来说最重要的内容。另外，他还在这一部分中使用了尽可能多的文字和图片以达到更好的效果。

除了这本书之外，还有其他一些关于 HTML5 的图书，而且以后还会越来越多。开发者的需求推动了 HTML5 的发展。未来将会出现为应用开发者编写的、长达 500 页的技术书籍，甚至还会有为浏览器厂商编写的、字数更多的精品图书。这些图书可以解决一些超出你我能力范围的技术难题。

但是，这是一本为创建网页内容、标记网页和语义、设计辅助接口和体验的开发人员所写的书。它可以被称为 HTML5 用户指南。本书清晰快速地阐明了每个棘手的主题，从而使读者可以尽快应用到工作中。

——Jeffrey Zeldman
A Book Apart 创始人

目录

第1章
标记简史

HTML 是万维网（World Wide Web）的统一语言。通过它所提供的标签，人类已经创建了各种各样令人惊奇的超链接文档网络。从 Amazon、eBay 和 Wikipedia，到个人博客和猫咪主题网站，这些无一不是 HTML 的杰作。

HTML5 是这门通用语言的最新版。自诞生之日起，这门语言一直在不停地发展。虽然这次升级的变化之大史无前例，但 HTML 已经不是第一次进行更新换代了。

在发明 Web 的同时，蒂莫西•约翰 - 伯纳斯—李爵士创建了 HTML（Hyper Text Markup Language，超文本标记语言）。1991 年，他撰写了一篇名为“HTML Tags”的文档，在该文档中，他推荐了将近 20 个用来编写网页的元素。

使用尖括号包围文本这种形式的标签并不是蒂姆先生的首创。在此之前，SGML（Standard Generalized Markup Language，标准通用标记语言）就已经开始使用这种标签了。蒂姆先生并没有发明新的语言，而是利用已经存在的技术——在 HTML5 的发展过程也体现了这种倾向。

1.1 从 IETF 到 W3C：HTML 4 的诞生过程

实际上，根本不存在HTML 1。最早的HTML官方规范是由IETF(Internet Engineering Task Force，因特网工程任务组）发布的HTML 2.0。这一规范中的许多特性都是在已有实现的基础上归纳总结出来的。例如，1994 年居于市场领导地位的 Mosaic 浏览器提供了 `<img>` 标签，开发人员可以通过该标签在自己的文档中嵌入图像。后来，`img` 元素就出现在了HTML 2.0 中。

继 IETF 之后，W3C（World Wide Web Consortium，万维网联盟）成为了 HTML 后续标准的制定者，其官方网站为 http://www.w3.org。20 世纪 90 年代中期以后，W3C 对 HTML 进行了几次升级，直至 1999 年发布的 HTML 4.01。

此时，HTML 的发展走到了一个十字路口。

1.2 XHTML 1：符合 XML 标准的 HTML

HTML 4.01 之后的修订版为 XHTML 1.0。其中，X 表示“eXtreme（极端）”。当时的网页开发人员在提到这个字母的时候，必须双臂交叉，作出一个 X 的形状来。

这只是个玩笑。实际上，X 表示的是“eXtensible（可扩展）”。另外，也没有必要在提到它时交叉双臂。

XHTML 1.0 规范的内容与 HTML 4.01 完全相同。没有添加任何新元素或新属性。这两个规范唯一的差别就是对 HTML 语法作出了不同的规定。HTML 为开发人员提供了很大的自由度，他们可以按照自己的意愿去编写元素和属性，但 XHTML 却要求开发人员遵从 XML 规则。XML 是 W3C 大多数技术

规范的基础，也是一种更为严格的标记语言。

更加严格的语法规则并没有什么坏处，反而可以促使开发人员按照统一的样式来编写标签。此前的标签和属性可以是大写、小写，或者任意大小写字母的组合，而 XHTML 1.0 文档则要求所有标签和属性都必须为小写。

XHTML 1.0 发布的时候恰逢浏览器普遍开始支持 CSS。开发人员意识到了网页标准的出现，特别是在 Web 标准项目（The Web Standards Project）的倡导下，XHTML 规定的这种更为严格的语法被看成是编写标记的“最佳实践”。

在此之后，W3C 发布了 XHTML 1.1。

如果说 XHTML 1.0 只不过是用 XML 重新表示的 HTML，那么 XHTML 1.1 才是真正且纯粹的 XML。也就是说，不能将 text/html 的 MIME 类型提供给 XHTML 1.1 文档。但是，如果开发人员以 XML 的 MIMI 类型来发布文档，那么当时世界上最流行的 Web 浏览器——Internet Explorer——就无法呈现该文档。

W3C 似乎已经开始与日常的网页发布脱节了。

1.3 XHTML 2：不被接受

如果 Dustin Hoffman 在电影《The Graduate》（毕业生）中的角色是一名网页设计师，那么 W3C 只会对他说一个词：XML。

W3C 在接管 HTML 的时候，HTML 已经发展到了第 4 版（version4）。然后他们又着手开发 XHTML 2，其目的是将 Web 建立在 XML 之上。

虽然 XHTML 2 的名字听起来与 XHTML 1 非常类似，但它们的差别却非常之大。与 XHTML 1 不同，XHTML 2 与已有的网页内容都不兼容，甚至与以前版本的 HTML 也不兼容。XHTML 2 的目的就是成为一门纯粹的语言，也就是不与以前的规范建立任何关系。

但这却是一场灾难。

1.4 分裂：WHATWG TF

一股反抗势力在W3C内部逐步壮大。W3C热衷于从理论角度构建单纯的标准，却无视网页设计人员的需求。来自 Opera、Apple 和 Mozilla 的代表对这种倾向非常反感，他们希望那些支持创建 Web 应用的特性能够得到更多的关注。

2004 年的一次工作组会议成为了矛盾激化的导火索。伊恩·希克森（当时仍效力于 Opera Software）建议，应以支持创建 Web 应用为目标来扩展 HTML，但这个提议被驳回了。

于是，心怀不满的反抗者们建立了自己的组织：Web Hypertext Application Technology Working Group（Web 超文本应用技术工作组），简称 WHATWG。

1.5 从 Web Apps 1.0 到 HTML5

从一开始，WHATWG 的工作方式就与 W3C 截然不同。W3C 采取基于表决的工作方式：提出议题、讨论议题、投票表决。WHATWG 同样会提出和讨论议题，但哪些特性可以被写入规范最终由编辑决定。而这个编辑就是伊恩•希克森。

表面上看，W3C 的流程更民主，也更公平。但实际上，政治博弈和内部争论经常会导致流程停滞不前。而在 WHATWG 中，任何人都可以自由地发表意见，但负责最后决议的则只有编辑一个人，因此其工作效率明显高很多。其实编辑也并非拥有绝对的权力：一个仅由受邀人员组成的指导委员会可以质疑编辑的偏执做法。

最初，WHATWG 的大部分工作被分为两个规范：Web Form 2.0 和 Web Apps 1.0。这两个规范都是在 HTML 的基础上扩展而来的。后来，这两个规范又被合并到一起，同时被简单地称为 HTML5。

1.6 再次联手

在 WHATWG 开发 HTML5 期间，W3C 继续制定了 XHTML 2 规范。如果说 XHTML2 规范的制定速度很快，那是不准确的。实际上，这个过程是十分缓慢的。

2006 年 10 月，蒂姆先生发表了一篇博文，承认将 Web 从 HTML 迁移到 XML 是行不通的。几个月后，W3C 签发了新委任状，成立了一个 HTML 工作组。这个工作组并没有采取一切从头开始的方式，而是明智地决定：应该在 WHATWG 工作成果的基础上开发未来版本的 HTML。

这样，时断时续的做法反而使情况变得令人困惑。W3C 同时有两个工作组，分别负责制定不同的、互不兼容的标记语言：XHTML2 和 HTML 5（注意数字 5 前面有一个空格）。与此同时，还有一个独立的组织，即 WHATWG，正在开发 HTML5（没有空格）规范，而该规范将作为上述 W3C 中一个规范的基础。

网页设计师们会发现，搞清楚上述状况比理解电影《记忆碎片》、《雷管》、甚至导演大卫·林奇的所有作品都要困难。

1.7 XHTML 已被废弃：XHTML 的语法永存

种种迷团终于在 2009 年烟消云散。W3C 宣布不再续领 XHTML 2 工作组的委任状。实际上，这种格式已经被废弃好几年了。这次的宣布差不多可以看成是为它补发了一张死亡证明。

奇怪的是，XHTML 2 并没有平静地逝去，不少兴灾乐祸的人跳出来大放厥词。XML 的反对者趁机奚落使用 XHTML 1 的开发人员——甚至忽略了 XHTML 1 和 XHTML 2 几乎没有任何共同点这一事实。

这时候，那些遵照 XHTML 1 严格规则的开发人员又担心起来，生怕 HTML5 又重新开始支持过去的标记。

其实，这样担心是多余的。虽然 HTML5 允许相对随意的标记，但它也支持严格的标记，到底选择哪种风格行事完全取决于使用人员。

1.8 总结

切记，HTML5 并不是一门凭空造出来的新语言。其标记的变化都是革新性的而非革命性的。无论开发人员正在使用哪个版本的 HTML 创建网站，他都可以说自己已经在使用 HTML5 了。

第2章
HTML5的设计

法国大革命是极端的政治和社会变革时期。这种革命热情也被倾注于对计时系统的改革中。在一段时期内，法兰西共和国引入了十进制计时制，即1天分为10小时，且1小时分为100分钟。该计时制的逻辑性和清晰性明显优于六十进制的计时制。

但十进制的计时制失败了。没有人使用这种计时制度。而XHTML 2的命运与之相似。W3C再次证明了法国大革命的教训：改变现有的行为习惯是非常非常困难的。

2.1 设计原则

为了避免过去所犯的错误，WHATWG起草了一系列设计原则以指导HTML5的开发。其中一项主要原则是“支持已有内容”。这意味着对于HTML5来说，并不存在创立的起始时间。

XHTML 2试图废弃之前的一切。而与之不同的是，HTML5建立在现有规范和实现的基础之上。HTML 4.01的大部分内容在HTML5中都得到了保留。

一些其他的设计原则，例如“不要做重复的工作”和“沿着足迹铺路”的意思是，对于网页设计师来说，如果存在一种普遍的方法来完成某项任务，那么即使它不是最好的方法，也应该被编入 HTML5 中，也就是说“别去修理没坏的东西”。

涉足过微格式（http://microformats.org）的网页设计师应该十分熟悉这些设计原则。HTML5 社区具有同样的务实方针以实现标准格式的统一，所以无需担心理论问题。

这种态度体现在“最终用户优先”的设计原则中，该原则规定：在发生冲突时，最终用户优先，其次是作者、实现者、标准制定者，最后才是理论上的完满。

伊恩·希克森已经多次表示，浏览器厂商才是 HTML5 真正的仲裁者。如果浏览器供应商拒绝支持某项协议，那么在规范中添加该协议就变得没有任何意义，因为这会使规范不够切合实际。根据最终用户优先的原则，网页设计师的意见更具有意义。如果网页设计师拒绝使用规范的某些内容，那么规范同样不够切合实际。

2.2 切合现实

持续的内部张力推动了 HTML5 的创立。一方面，规范需要足够强大，从而有能力支持创建网页应用程序，另一方面，虽然大多数现有内容都处于完全混乱的状态，但是 HTML5 仍需要支持已有的内容。如果 HTML5 的规范在某一个方向上偏离得太远，那么它将重蹈 XHTML 2 的覆辙。但是，如果它在另一个相反的方向上偏离得太远，那么它就会认为 `<font>` 标签和表单是万能的，因为这两者是大量网页建立的基础。

这是一种微妙的平衡，保持这种平衡需要务实且冷静的方法。

2.3 错误处理

HTML5 不仅声明了浏览器应该如何处理规范格式的标记，还首次规范了浏

览器该如何处理格式不规范的文件。

浏览器厂商曾不得不独自研究如何处理错误。无论最流行的浏览器做出怎样的尝试，该过程通常都会涉及逆向工程，这会耗费浏览器厂商的时间。与其浪费时间模仿竞争对手处理有缺陷的标记，倒不如尝试实现新功能。

在 HTML5 中定义错误处理恐怕难以实现。虽然 HTML5 具有与 HTML 4.01 完全相同的元素和属性，并且完全没有添加新特性，但在 2012 年年底之前完成错误处理的定义仍然是徒劳的。

网页设计人员可能对错误处理不大感兴趣，特别是在他们一开始就会编写有效并且格式规范的文件的情况下，但错误处理对于浏览器厂商来说却非常重要。以往的标记规范都是为创作者编写的，而 HTML5 却是为创作者和实施者编写的。网页设计人员在细读规范时应牢记这一点。这就解释了为什么 HTML5 规范的内容如此之多，同时也解释了为什么该规范含有一些通常为专家所保留的细节。

2.4 DOCTYPE：形式更简洁

文档类型声明（Document Type Declaration）简称为 doctype，一直用于指定文档所编写的标记类型。

HTML 4.01 的 doctype 如下所示（» 为自动换行标记）：

```
<!DOCTYPE HTML PUBLIC »
"-//W3C//DTD HTML 4.01//EN" »
"http://www.w3.org/TR/html4/strict.dtd">
```

XHTML 1.0 的 doctype 如下所示：

```
<!DOCTYPE html PUBLIC »
"-//W3C//DTD XHTML 1.0 Strict //EN" »
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

这些 doctype 看起来并不易读，但它们以其独特的方式简单地说明了：“该

文档用 HTML 4.01 编写”或“该文档用 XHTML 1.0 编写”。

如果 doctype 声称“该文档用 HTML5 编写”，那么按道理其中应该会出现数字 5。但事实并非如此。HTML5 的 doctype 如下所示：

```
<!DOCTYPE html>
```

该 doctype 是如此之短，甚至可以让人将其背诵下来。

这实在是太不可思议了！如果 doctype 中不含有版本号，那么该如何指定其他版本的 HTML 呢？

第一次看到 HTML5 的 doctype 的时候，我认为这是高度傲慢的结果。心想：“难道他们真相信这就是标记规范的最终版了吗？”

然而事实上，HTML5 的 doctype 是非常务实的。由于 HTML5 需要支持现有内容，所以其 doctype 可以应用于现有的 HTML 4.01 文档和 XHTML 1.0 文档。任何未来版本的 HTML 也需要支持 HTML5 中的现有内容，因此应用版本号来标记文档的观念是有缺陷的。

事实上，doctype 并不那么重要。假设需要为一个文档提供 HTML 4.01 的 doctype。如果该文档中包含来自另一个规范的元素，如 HTML 3.2 或 HTML5，那么浏览器将仍然呈现该文档的这一部分。这是因为浏览器支持的是特性，而非 doctype。

起初，文档类型声明（Document Type Declaration）是为验证器而非为浏览器而设计的。浏览器仅在“doctype 转换”的情况下才会关注 doctype——“doctype 转换”（doctypy switching）是一个聪明的小黑客，它根据是否存在合适的 doctype 来转换显示模式，即怪异模式（quirks mode）或标准模式（standard mode）。

为了确保浏览器以标准模式显示，至少需要 HTML5 的 doctype。事实上，这是包含 doctype 的唯一原因。不用 HTML5 的 doctype 编写的 HTML 文档仍然是有效的 HTML5。

2.5 保持简洁

HTML5 中简化的不仅仅是 doctype。

如果想要指定一个标记文档的字符编码，那么最好的方法就是确保服务器发送了正确的 Content-Type 文件头（header）。如果想要双倍保险，那么也可以使用 <meta> 标签来指定字符集。<meta> 是一个用 HTML 4.01 编写的文件声明：

```
<meta http-equiv="Content-Type" content="text/html; »
charset=UTF-8">
```

下面的示例实现了与前一示例在 HTML5 中所实现的相同功能，但其实现方式更加令人印象深刻：

```
<meta charset="UTF-8">
```

这种简化了字符编码的 doctype 包含了需要浏览器编译的最少字符数。

另一处可以缩减字符数量的地方是 标签。常见的做法是为 元素添加 属性，属性值为“text / javascript”：

```
<script type="text/javascript" src="jquery.min.js">
</script>
```

浏览器并不需要该属性。它们假定该脚本已用 JavaScript 编写。JavaScript 是最流行的网页脚本语言（实际上，JavaScript 也是唯一的一种网络脚本语言）：

```
<script src="jquery.min.js">
</script>
```

同样，不需要在每次链接到 CSS 文件时都指定一个值为“text/ CSS”的 属性：

```
<link type="text/css" rel="stylesheet" href="mystyle.css">
```

只需简单地写为：