第一图书网, tushu007.com

<<HTML5秘籍>>

图书基本信息

书名：<<HTML5秘籍>>

13位ISBN编号：9787115290182

10位ISBN编号：7115290180

出版时间：2012-8

出版时间：人民邮电出版社

作者：Matthew MacDonald

页数：360

字数：560000

译者：李松峰,朱巍

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：http://www.tushu007.com

前言

乍一看，你可能觉得HTML5是网页编写语言HTML的第5个版本。
但实际上，这背后的故事可乱得多。
HTML5是一个叛逆。
它是由一群自由思想者组成的团队设计出来的，这个团队的成员并不负责制定官方HTML标准。
它允许使用10年前就被禁止的网页编写方式。
它费尽心机、苦口婆心地告诉浏览器开发商怎么处理而不是彻底拒绝标记中的错误。
它最终实现了不依赖Flash等浏览器插件播放视频。
而且它引入了一大批JavaScript驱动的功能，让网页可以像桌面软件那样丰富多彩、富有交互能力。
理解HTML5可没有那么简单。
最主要的困难在于人们用HTML5这个词指代十几甚至更多种独立的标准。
（后面我们会介绍到，这是HTML5发展演进的结果。
一开始时它只有一个标准，但后来就拆分成了很多容易管理的分支。
）事实上，HTML5现在代表的是“HTML5及所有相关标准”，甚至还可以更宽泛，代表“下一代网页编写技术”。
这就是本书要带领大家探索的HTML5：既包括HTML5核心语言，也包括与HTML5纠缠在一块但在其标准中永远找不到的那些新功能。
于是，第二个困难又摆在了你的面前：浏览器支持。
不同的浏览器支持HTML5的不同部分，而且还有一些让人难受的新功能，任何平台的浏览器都不支持。
抛开这些困难，有一个事实接受起来毫无挑战性：HTML5代表未来。
苹果、谷歌等大软件公司都在鼎力支持它；W3C（World Wide Web Consortium，万维网联盟）已经放弃了XHTML，从而使HTML5成为正式标准并得到认可；而且所有浏览器开发商现在都对它的大部分功能给予了支持。
如果你在看这本书，那就有可能在它还让人觉得好玩和刺激的时候加入HTML5阵营，并创造出如图0-1所示的那种酷炫的网页。
阅读本书的条件本书介绍的HTML5是HTML标准最新最好的版本。
虽然不一定非得是标记大师才能看懂这本书，但阅读本书的的确确还是需要一些Web设计经验的。
以下就是几个必要条件。
写过网页。
本书假设你以前至少写过一些网页（或者至少知道怎么使用HTML元素把内容分成标题、段落和列表，等等）。
如果你才刚刚接触Web设计，那最好是先找一本合适的入门书看一看，比如我的Creating a Website：The Missing Manual。
（不过别担心，你不会被限制在过去的技术中，Creating a Website这本书里的示例都是有效的HTML5文档。
）图0-1：在Web世界黑暗的过去（也就是去年），要编写网页游戏，必须依赖Flash这样的浏览器插件。
但有了HTML5的新功能——包括canvas（图中显示的就是）之后，你可以使用可靠而又免插件的JavaScript。
这个图展示的是用HTML5技术开发的迷宫游戏（第7章将详细讨论）懂样式表。
没有CSS（Cascading Style Sheet，层叠样式表）就没如今的网站。
CSS为页面提供布局和格式。
要想顺利阅读本书，你应该知道样式表的基本知识，包括怎么创建样式表，里面都有什么，以及怎么把它应用到网页上。
如果你不太清楚CSS是干什么的，可以先看一看附录A（“CSS简明教程”）。

如果你需要更多帮助，或者想提高自己的CSS技能，以便真正做出漂亮的布局和样式，建议你看看David Sawyer McFarland的CSS： The Missing Manual（O’Reilly）。
懂JavaScript。
当然，编写HTML5页面用不着JavaScript。
可是，如果你想使用HTML5不计其数的那些超酷功能——比如在画布上画图或者与Web服务器通信，那就需要JavaScript了。
如果你有一些浅显的编程经验，但对JavaScript还一知半解，附录B（“JavaScript简明教程”）可以帮你掌握一些新情况。
不过，要是一听到写代码这几个字，马上就像被窝里爬进一条蟒蛇那样魂飞魄散，那要么你根本不必看本书中的很多章节了，要么你得通过David Sawyer McFarland的JavaScript & jQuery： The Missing Manual （O’Reilly）补补课。
如果这些必要条件让你头晕目眩——好吧，这就是活在Web设计最前沿必须付出的代价。
编写HTML5编写HTML5页面可以使用编写HTML页面时使用的软件。
可以是个再简单不过的文本编辑器，像Windows中的记事本，或者Mac中的TextEdit。
目前也有很多设计工具（比如Adobe Dreamweaver和Microsoft Expression Web）提供了快速创建新HTML5文档的模板。
不过，HTML5页面的基本结构确实非常简单，任何网页编辑软件（即使不是为HTML5设计的）都没有问题。
注意 当然啦，不管你上网和编写网页时用的计算机是Windows PC，还是最新的MacBook，同样也无所谓，因为HTML5与操作系统无关。
查看HTML5每个人都想问一个问题：“哪些浏览器支持HTML5·”可悲的是，这个问题没有明确的答案。
本书后面会介绍，HTML5实际上是一组独立标准的集合。
有些标准已经得到了支持，而另一些标准几年内（甚至永远）不会得到支持。
其他所有标准则介于这两种情况之间；换句话说，HTML5在某些浏览器的某些版本中能够运行。
下面列出的浏览器无需什么变通手段，就可以支持HTML5的绝大部分。
Internet Explorer 9及更高版本Firefox 3.5及更高版本谷歌Chrome 8及更高版本Safari 4及更高版本Opera 10.5及更高版本更高版本的支持程度更高。
也就是说，Firefox 5能比Firefox 3.5更好地支持HTML5。
在鼓励大家使用新的HTML5功能之前，本书会清楚地说明当前浏览器对这些功能的支持情况。
当然，浏览器版本的变化相对比较快，因此在尝试某些可能有问题的功能之前，你自己应该先搜索一下最新的支持情况。
推荐一个网站：http：//caniuse.com，你可以在上面搜索某个具体的功能，然后它会告诉你到底哪个浏览器的哪个版本支持该功能。
（1.6节还将更详细地介绍这个工具的用法。
）注意 本书会讨论那些已知在某些浏览器中不能使用的功能。
别慌，如果你只想对HTML5有所了解，而专注于那些今天可以使用的功能，这样不是挺好嘛。
你可以通过这些功能窥见Web的未来。
什么时候可以使用HTML5简短的答案是“现在”。
就连遭人唾弃的Internet Explorer 6，这个问世长达10年之久、补丁摞补丁的家伙都可以显示HTML5文档。
这是因为创建HTML5标准时，就想让它能涵盖并扩展原来的HTML。
更详尽的答案是“视情况而定”。
前面刚刚提到过，HTML5是一组不同标准的集合，浏览器对这些标准有着不同程度的支持。
因此，尽管现在任何Web开发人员都可以转而编写HTML5文档（Google、YouTube和Wikipedia等一些大型网站已经这样做了），但要放心地使用大部分HTML5 的新奇功能——至少不必针对那些不够开化的浏览器采取变通手段，恐怕还要再等几年。

注意 某项功能到底属于哪个规范并不重要，重要的是现在有没有浏览器支持它（以及尚未支持它的浏览器将来有没有可能支持它）。
本书每介绍一项新功能，都会告诉读者它来自哪个规范，以及都有哪些浏览器支持它。
作为有标准意识的开发人员，恐怕你也对这些标准什么时候正式颁布感兴趣。
但这个问题有点复杂，因为设计HTML5的人遵循的理念有点不合常规。
他们经常说，正式的标准中怎么说并不重要，关键是有没有浏览器支持。
（换句话说，只要你觉得可行，现在就可以采用任何你想使用的功能。
）但不少开发人员、大公司、政府机关以及其他组织，通常会根据一种语言的标准是否正式颁布来判断是否可以采用它。
从技术上说，HTML5语言现在还是W3C手中的工作草案（working draft）。
对标准的这种称谓表明它已经相当成熟了，但在成为候选推荐标准（candidate recommendation）的时候（可能是2012年的某一天），仍然可能会有改动。
而到真正的推荐标准（recommendation）阶段，因为必须有足够的测试，可能就是很多年之后的事了。
但那确实不怎么重要了，因为到了该阶段，即便有改动也会很少，而想要使用HTML5的人也早已经有了自己的选择。
本书内容本书把完整的HTML5教程分为12章，具体内容如下。
第一部分：认识新语言第1章（“HTML5简介”）介绍HTML发展到HTML5的历程。
我们会看一看HTML5文档的样子，看看它跟以前的HTML有何不同，另外也看一下浏览器的支持情况。
第2章（“构造网页的新方式”）讨论HTML5的语义元素（semantic element），也就是一组可以为标记赋予含义的元素。
恰当地使用这些元素，可以让浏览器、屏幕阅读器、Web设计工具以及搜索引擎基于它们提供的额外信息更智能地工作。
第3章（“有意义的标记”）进一步讨论语义的概念，涉及微数据（microdata）等标准。
尽管这一章的内容有点偏理论，但透彻理解这个概念可以给Web开发人员带来巨大的回报：在Google等搜索引擎的结果列表中显示更全面、更详尽的内容。
第二部分：制作新网页第4章（“Web表单”）探索HTML5 Web表单元素的变化，包括文本框、选择列表、复选框和其他用来从访客那里收集信息的微件（widget）。
HTML5为捕获数据输入错误提供了一些辅助和基本工具。
第5章（“音频与视频”）讲一讲HTML最激动人心的新功能，即支持音频和视频播放。
这一章将介绍如何避免遭遇“Web视频编解码器大战”，创建出在所有浏览器中都能工作的播放页面，同时还要学习创建自己定制的播放器。
第6章（“基本Canvas绘图”）介绍名为画布（canvas）的二维绘图表面。
你将会学习怎样在画布上绘制图形、图像、文本，甚至还将构建一个简单的绘图程序（使用一系列强大的JavaScript代码）。
第7章（“高级Canvas技术”）进一步提升你的“绘画”技术。
这一章将会学习投影、花哨的模式，以及可点击的交互图形和动画等更加令人神往的Canvas技术。
第8章（“使用CSS3”）将介绍最新版本的CSS3标准，它与HTML5可谓绝配。
我们将学习如何应用新奇的字体让文本变得活泼可爱，如何让页面适应不同的移动设备，以及利用变换添加吸引人的效果。
第三部分：构建桌面式Web应用第9章（“数据存储”）讨论在访客计算机中保存小段数据的新Web存储功能。
（非常像cookie的超级简捷版。
）这一章还将介绍如何在网页而不是在Web服务器中，使用JavaScript代码处理用户选择的文件。
第10章（“离线应用”）探索新的HTML5缓存功能，这个功能可以实现在断网的情况下仍然能够通过浏览器查看网页。

第11章（“与Web服务器通信”）将把目光投向与Web服务器通信这个主题上。
为此，将介绍久负盛名的XMLHttpRequest对象，JavaScript通过它可以联系Web服务器并请求信息。
然后再讨论两个比较新的功能：服务器端事件和（影响更加深远但还需要完善的）Web套接字。
第12章（“更酷的JavaScript技术”）介绍了解决现代Web应用开发难题的三个新功能。
第一是可以确定访客位置的地理定位；第二是在后台执行复杂任务的Web Worker；第三是能够同步网页URL到当前状态的新的浏览器历史功能。
最后有两个附录，可以为你掌握HTML5补习一些基础知识。
附录A是对CSS的一个简要介绍，附录B则会简单地介绍JavaScript。
在线资源作为Missing Manual丛书的读者，你所得到的不仅仅是一本书。
在网上，你还可以找到示例文件以及技巧、文章，甚至是一两段视频。
你可以跟Missing Manual团队交流，告诉我们你喜欢（或讨厌）本书的哪一方面。
请访问网站，或直接阅读后面的某一小节。
Missing CD本书没有附带光盘，但这对学习本书一点影响都没有。
读者可以访问本书的Missing CD页面，下载本书讨论和展示的网页示例，这样你就不必自己动手敲那些长长的网页地址了。
这个页面中列出了全书每一章提到的网站的链接。
提示 假如你想找某个特定的例子，我教给你一个好办法——看插图。
在插图中，文件名一般都会出现在浏览器地址栏的末尾。
比如，看到文件路径C：\HTML5\Chapter01\SuperSimple HTML5.html（图1-1），就知道对应的示例文件名叫SuperSimpleHTML5.html。
试验站点还有另一种使用本书示例的方法，就是访问在线示例网站。
在这个网站上可以看到本书的每一个示例，并直接在浏览器中运行它们。
因为HTML5的某些功能需要一个真正的Web服务器，所以直接使用这个网站其实可以省点心。
（如果你直接从计算机硬盘上运行网页，这些功能可能会导致一些怪异的现象，或者完全不能用。）而使用这个网站，就可以先看到某个例子的运行结果，然后再下载该页面并动手尝试。
注意 别担心自己不知道哪些HTML5功能需要Web服务器，到时候本书会给出提示的。
注册如果你在oreilly.com注册了这本书，可能会享受到一些优惠，比如购买Creating a Website： The Missing Manual的新版时可以打个折。
注册其实只需点几次鼠标。
在浏览器地址栏里输入http：//tinyurl.com/registerbook，直接就可以跳到注册（Registration）页面。
反馈有问题要问？
需要更多信息？
想给我们写个书评？
在反馈（Feedback）页面上，你可以向专家请教自己看书时碰到的问题，也可以分享自己对Missing Manual丛书的看法，甚至找到一些志同道合的朋友，听听他们谈论在做网站过程中的一些体会。
勘误为了尽可能保证本书切合实际、准确无误，每次重印我们都会纠正一些确认的勘误。
这些勘误信息也会在本书网站上发布出来，以便读者更正自己手里这本书的错误。
要提交或查看勘误，请访问http：//tinyurl.com/3q56k7v。
简报订阅我们免费的电子邮件简报可以随时了解Missing Manual丛书的新动向。
这样可以方便你与作者、编辑联系，并能看到免费视频和图书样章等。
Safari· Books OnlineSafari· Books Online是一个按需阅读的数字图书馆，有7500种技术图书和视频可供搜索。
通过订阅，可以在此阅读所有图书，观看任何视频。
甚至可以在新书印刷之前阅读到它们。
可以复制粘贴示例代码，收藏喜欢的内容，下载整章内容，为关键部分创建书签，添加评注，打印页面，以及享受其他众多省时省力的阅读体验。
O’Reilly Media已经将本书上传到Safari Books Online。

访问网站并免费注册，可以看到本书及O’Reilly和其他出版社图书的完整电子版。

内容概要

《HTML5秘籍》共包括四个部分，共12章。
第一部分介绍了HTML5的发展历程，利用HTML5重新构造网页，以及HTML5的语义元素。
第二部分介绍了HTML5对传统Web表单的翻新、HTML5中的音频与视频、Canvas绘图技术、CSS3等内容。
第三部分介绍了数据存储、离线应用、与Web服务器通信，以及HTML5与JavaScript技术的强大结合等内容。
第四部分为附录，简单介绍了CSS和JavaScript。

《HTML5秘籍》既适合新手学习，也能助有经验的Web开发人员解决日常工作中遇到的难题。

作者简介

Matthew MacDonald
开发人员、技术作家、Visual Basic和.NET讲师。
从Visual Basic和ASP一发布就一直在使用它们，并编写了十几本这方面的图书，包括《ASP.NET 4高级程序设计》、The Book of VB .NET（No Starch出版社）和Visual Basic 2005: A Developer's Notebook（O’Reilly出版社）。
他还编写了Missing Manual系列图书之Excel 2007和Access 2007（O’Reilly出版社）。
他的个人网站http://www.prosetech.com/。

“The Missing Manual系列绝对是入门指南最为明智的选择……”
——凯文?凯利，《连线》联合创始人

书籍目录

目　录

第一部分　认识新语言
第1章　HTML5简介　2
1.1　HTML5的故事　2
1.1.1　XHTML 1.0：更严格的标准　2
1.1.2　XHTML 2：意想不到的失败　3
1.1.3　HTML5：起死回生　3
1.1.4　HTML：活着的语言　5
1.2　HTML5的三个主要原理　6
1.2.1　不破坏Web　6
1.2.2　修补牛蹄子路　7
1.2.3　实用至上　8
1.3　HTML5标记初体验　8
1.3.1　HTML5文档类型　10
1.3.2　字符编码　11
1.3.3　页面语言　11
1.3.4　添加样式表　12
1.3.5　添加JavaScript　12
1.3.6　最终结果　13
1.4　HTML5语法　13
1.4.1　放松的规则　14
1.4.2　HTML5验证　15
1.4.3　XHTML的回归　17
1.5　HTML5元素家族　18
1.5.1　新增的元素　18
1.5.2　删除的元素　18
1.5.3　改变的元素　19
1.5.4　调整的元素　20
1.5.5　标准化的元素　21
1.6　今天开始用HTML5　22
1.6.1　了解浏览器支持情况　23
1.6.2　浏览器装机情况统计　24
1.6.3　通过Modernizr检测功能　26
1.6.4　使用“腻子脚本”填补功能缺陷　28
第2章　构造网页的新方式　30
2.1　语义元素　30
2.2　改造传统的HTML页面　32
2.2.1　构造页面的老办法　32
2.2.2　使用HTML5构造页面　36
2.2.3　用标注副标题　39
2.2.4　用添加插图　40
2.2.5　用添加附注　42
2.3　浏览器对语义元素的支持情况　43
2.4　使用语义元素设计站点　45

2.4.1 理解 46
2.4.2 用标注导航链接 48
2.4.3 理解 52
2.4.4 理解区块 54
2.5 HTML5纲要 55
2.5.1 如何查看纲要 55
2.5.2 基本纲要 56
2.5.3 分块元素 58
2.5.4 解决一个纲要问题 60
第3章 有意义的标记 64
3.1 回顾语义元素 64
3.1.1 使用标注日期和时间 65
3.1.2 使用标注JavaScript返回值 66
3.1.3 使用标注突显文本 68
3.2 其他语义标准 69
3.2.1 ARIA 70
3.2.2 RDFa 71
3.2.3 Microformats 71
3.2.4 Microdata 76
3.3 Google Rich Snippets 79
3.3.1 增强搜索结果 79
3.3.2 菜谱搜索引擎 82
第二部分 制作新网页
第4章 Web表单 86
4.1 理解表单 86
4.2 传统表单翻新 88
4.2.1 通过占位符文本添加提示 91
4.2.2 焦点：挑选正确的起点 93
4.3 验证：阻止错误 93
4.3.1 HTML5验证的原理 94
4.3.2 关闭验证 95
4.3.3 验证样式挂钩 96
4.3.4 使用正则表达式 97
4.3.5 自定义验证 98
4.3.6 浏览器对验证的支持 99
4.4 新的输入控件 102
4.4.1 电子邮件地址 104
4.4.2 网址 105
4.4.3 搜索框 105
4.4.4 电话号码 105
4.4.5 数值 105
4.4.6 滑动条 106
4.4.7 日期和时间 107
4.4.8 颜色 108
4.5 新元素 108
4.5.1 使用显示输入建议 109
4.5.2 进度条和计量条 111

4.5.3　使用和创建工具条和菜单　113
4.6　网页中的HTML编辑器　113
4.6.1　使用contentEditable编辑元素　114
4.6.2　使用designMode编辑页面　116
第5章　音频与视频　118
5.1　理解今天的视频　118
5.2　HTML5音频与视频　119
5.2.1　使用播放点噪音　120
5.2.2　了解　122
5.3　格式之争与后备措施　123
5.3.1　谈谈格式　124
5.3.2　浏览器对媒体格式的支持情况　125
5.3.3　多种格式：如何讨好每一款浏览器　127
5.3.4　使用元素　128
5.3.5　以Flash作后备　129
5.4　使用JavaScript控制播放器　132
5.4.1　添加音效　133
5.4.2　创建自定义视频播放器　136
5.4.3　JavaScript媒体播放器　138
5.4.4　字幕与无障碍性　140
第6章　基本Canvas绘图　142
6.1　Canvas起步　142
6.1.1　画直线　145
6.1.2　路径与形状　148
6.1.3　绘制曲线　149
6.1.4　变换　152
6.1.5　透明度　155
6.2　构建基本的画图程序　157
6.2.1　准备工作　158
6.2.2　在画布上绘图　160
6.2.3　将画布保存为图像　161
6.3　浏览器对Canvas的支持情况　164
6.3.1　填平补齐Canvas　164
6.3.2　Canvas后备及功能检测　166
第7章　高级Canvas技术　168
7.1　高级Canvas绘图　168
7.1.1　绘制图像　168
7.1.2　裁剪、切割和伸缩图片　170
7.1.3　绘制文本　172
7.2　阴影与填充　173
7.2.1　添加阴影　173
7.2.2　填充图案　175
7.2.3　填充渐变　176
7.2.4　综合示例：绘制图解　180
7.3　赋予图形交互能力　184
7.3.1　记录绘制的内容　184
7.3.2　基于坐标的碰撞检测　187

7.4　给Canvas添加动画　189
7.4.1　基本的动画　190
7.4.2　多物体动画　191
7.5　实例：迷宫游戏　195
7.5.1　布置迷宫　196
7.5.2　让笑脸动起来　198
7.5.3　基于像素颜色的碰撞检测　199
第8章　使用CSS3　203
8.1　使用CSS3　203
8.1.1　选择一：用能用的　204
8.1.2　选择二：将CSS功能作为增强　204
8.1.3　选择三：Modernizr　205
8.1.4　特定于浏览器的样式　208
8.2　Web排版　209
8.2.1　Web字体格式　211
8.2.2　使用字体包　212
8.2.3　使用谷歌的Web字体　214
8.2.4　使用自己的字体　216
8.2.5　多栏文本　218
8.3　适用不同的设备　220
8.3.1　媒体查询　220
8.3.2　高级媒体查询　224
8.3.3　替换整个样式表　225
8.3.4　识别移动设备　226
8.4　多变的盒子　227
8.4.1　透明盒子　227
8.4.2　圆角盒子　229
8.4.3　背景盒子　230
8.4.4　阴影盒子　231
8.4.5　渐变盒子　232
8.5　创建过渡效果　234
8.5.1　基本的颜色过渡　235
8.5.2　更多的过渡思路　237
8.5.3　变换　237
第三部分　构建桌面式Web应用
第9章　数据存储　242
9.1　Web存储简介　242
9.1.1　存储数据　244
9.1.2　实战：保存游戏中的最后位置　246
9.1.3　浏览器对Web存储的支持情况　247
9.2　深入Web存储　248
9.2.1　删除数据项　248
9.2.2　查找所有数据项　248
9.2.3　保存数值和日期　249
9.2.4　保存对象　250
9.2.5　响应存储变化　251
9.3　读取文件　253

9.3.1 取得文件 254
9.3.2 浏览器对File API的支持情况 254
9.3.3 读取文本文件 255
9.3.4 替换标准上传控件 257
9.3.5 一次读取多个文件 257
9.3.6 读取图片文件 258
第10章 离线应用 261
10.1 通过描述文件缓存资源 262
10.1.1 创建描述文件 262
10.1.2 使用描述文件 264
10.1.3 把描述文件放到Web服务器 265
10.1.4 更新描述文件 267
10.1.5 浏览器对离线应用的支持情况 269
10.2 实用缓存技术 270
10.2.1 访问未缓存的文件 270
10.2.2 添加后备内容 271
10.2.3 检测连接 273
10.2.4 通过JavaScript指定更新 274
第11章 与Web服务器通信 277
11.1 向Web服务器发送消息 277
11.1.1 XMLHttpRequest对象 278
11.1.2 向Web服务器提问 279
11.1.3 取得新内容 283
11.2 服务器发送事件 286
11.2.1 消息格式 287
11.2.2 通过服务器脚本发送消息 288
11.2.3 在网页中处理消息 289
11.2.4 轮询服务器端事件 291
11.3 Web Socket 292
11.3.1 访问Web Socket 293
11.3.2 简单的Web Socket客户端 294
11.3.3 使用现成的Web Socket服务器 295
第12章 更酷的JavaScript技术 298
12.1 地理定位 298
12.1.1 地理定位的基本原理 299
12.1.2 查找访客的坐标 301
12.1.3 处理错误 303
12.1.4 设置地理定位选项 305
12.1.5 显示地图 306
12.1.6 跟踪访客移动 309
12.2 Web Workers 309
12.2.1 费时的任务 311
12.2.2 把任务放在后台 313
12.2.3 处理Worker错误 316
12.2.4 取消后台任务 316
12.2.5 传递复杂消息 317
12.3 历史管理 320

第一图书网, tushu007.com

12.3.1　URL问题　320
12.3.2　以往的解决方案：Hashbang URL　321
12.3.3　HTML5的方案：会话历史　322
12.3.4　浏览器对会话历史的支持情况　325
第四部分　附录
附录A　CSS简明教程　328
附录B　JavaScript简明教程　342

章节摘录

版权页： 插图： 4.2.2焦点：挑选正确的起点 加载完表单之后，用户要做的第一件事儿就是填写表单。
然而，除非用户按下Tab切换到第一个控件，或者在其中单击一下鼠标，从而让第一个控件获得焦点，否则就不能输入。
在相应的元素上通过JavaScript调用focus（）方法，可以帮用户完成焦点切换。
但这样就得编写代码，而且有时还会出现问题。
比如，在调用focus（）方法之前，用户已经单击了其他控件并开始输入的情况也可能发生。
这时硬性地把焦点切换到第一个控件显得很粗鲁。
但如果浏览器自己能控制焦点，它就可以在用户操作之前，先把焦点给予正确的控件。
这就是HTML5添加autofocus属性的初衷，但只能给一个或元素添加这个属性： Name* 与placeholder属性一样，除了IE，其他浏览器都支持autofocus属性。
同样，对于IE还是有办法的。
可以使用Modernizr（参考1.6.3节）检测浏览器是否支持autofocus.然后自己编写脚本实现。
也可以使用别人已经开发好的JavaScript程序（http:／／tinyurl.com／polyfills）。
不过，很少有人为了这点功能如此兴师动众，除非你也想让IE支持其他表单功能，比如下面要讨论的数据验证。
4.3验证：阻止错误 表单中的字段是为了从网页访客那里收集信息的。
但是，无论你多么彬彬有礼地询问，都可能得不到想要的结果。
性子急躁的或者糊里糊涂的访客，有可能跳过重要的部分，只填写少量信息，或者只是不小心按错了键、最后怎么样？
他单击了“ 提交 ”按钮，你的网站收集到了一堆乱七八糟的数据。
很多网页都需要验证，就是在发生错误时捕捉到它（而更好的方案是防止出错）。
很多年来，开发人员都要自己写JavaScript验证脚本，有时候也会用到专业的avaScript库。
应该说，这些验证方法效果都非常好。
然而，鉴于验证是那么常用（可能要对每个人都做错误检查），验证解决的问题是那么集中（例如，发现无效的电子邮件或日期），以及编写验证脚本那么讨厌（没有人真心喜欢为每个表单都编写一遍同样的代码，更不要说测试这些脚本了），因此这个问题一定还有改进的空间。
HTML5规范的制定者决定让浏览器帮忙解决这些小问题，以后就不用让开发人员操心了。
为此，他们设计了一套客户端验证方法（参见后面的“ 在两个地方验证 ”），让我们可以在字段里嵌入常用的错误检查规则。
而且，嵌入验证规则的方法很简单，只要指定正确的属性就行。

编辑推荐

第一图书网, tushu007.com

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:http://www.tushu007.com