

SAMS  
**Teach  
Yourself**

- 全球销量逾百万册的系列图书
- 连续十余年打造的经典品牌
- 直观、循序渐进的学习教程
- 掌握关键知识的最佳起点
- “Read Less, Do More”（精读多练）的教学理念
- 以示例引导读者完成最常见的任务

每章内容针对初学者精心设计，**1**小时轻松阅读学习，  
**24**小时彻底掌握关键知识

每章**案例与练习题**助你轻松完成常见任务，  
通过**实践**提高应用技能，巩固所学知识

# HTML、CSS和 JavaScript

## 入门经典（第2版）

[美] Julie Meloni 著  
陈宗斌 译



中国工信出版集团



人民邮电出版社  
POSTS & TELECOM PRESS

# HTML、CSS和 JavaScript 入门经典 (第2版)

[美] Julie Meloni 著  
陈宗斌 译

人民邮电出版社  
北京

## 图书在版编目(CIP)数据

HTML、CSS和JavaScript入门经典：第2版 / (美)  
梅洛尼 (Meloni, J.) 著；陈宗斌译. — 北京：人民邮  
电出版社，2015.12  
ISBN 978-7-115-40594-4

I. ①H… II. ①梅… ②陈… III. ①超文本标记语言  
—程序设计②网页制作工具③JAVA语言—程序设计 IV.  
①TP312②TP393.092

中国版本图书馆CIP数据核字(2015)第246589号

## 版 权 声 明

Julie C. Meloni: Sams Teach Yourself HTML, CSS and JavaScript All in One(2nd Edition)  
ISBN: 0672337142

Copyright © 2015 by Pearson Education, Inc.

Authorized translation from the English languages edition published by Pearson Education, Inc.

All rights reserved.

本书中文简体字版由美国 Pearson 公司授权人民邮电出版社出版。未经出版者书面许可，对本书任何部分不得以任何方式复制或抄袭。

版权所有，侵权必究。

- 
- ◆ 著 [美] Julie Meloni
  - 译 陈宗斌
  - 责任编辑 傅道坤
  - 责任印制 张佳莹 焦志炜
  - ◆ 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>  
北京昌平百善印刷厂印刷
  - ◆ 开本：787×1092 1/16  
印张：34  
字数：851千字 2015年12月第1版  
印数：1-2500册 2015年12月北京第1次印刷  
著作权合同登记号 图字：01-2014-8394号

---

定价：79.00元

读者服务热线：(010)81055410 印装质量热线：(010)81055316  
反盗版热线：(010)81055315

# 内容提要

东非干夫

本书采用直观的、逐步深入的方法介绍如何使用 HTML、CSS (Cascading Style Sheet, 层叠样式表) 和 JavaScript 设计、创建和维护世界级的网站, 并且提供了清晰的指导。

本书信息丰富, 涵盖了 HTML、CSS 和 JavaScript 的基础知识和高级应用, 还介绍了如何组织和管理网站以及帮助人们找到你的 Web 页面的知识。在每一章末尾还配有相应的问答题和作业, 可以帮助你检验所学的知识。

本书示例丰富, 操作步骤简洁、明了, 非常适合于从事 Web 设计和网站开发的初学者阅读, 对于有一定经验的读者, 也有很大的参考价值。

## 关于作者

## 要 目 录

Julie C. Meloni 是一位居住在美国华盛顿特区的软件开发经理和技术顾问。她编写过多本图书和若干篇文章，都是关于基于 Web 的程序设计语言和数据库等主题的，包括畅销书 *Sams Teach Yourself PHP, MySQL and Apache All in One*。

# 目 录

第 1 章 理解 Web 的工作方式	1	第 2 章 构造 HTML 文档	18
1.1 HTML 和 WWW 简史	2	2.1 准备工作	19
1.2 创建 Web 内容	2	2.2 从一个简单的 Web 页面开始	19
1.3 理解 Web 内容递送	3	2.3 每个 Web 页面都必须具有的 HTML 标签	23
1.4 选择 Web 托管提供商	5	2.4 利用段落和换行符组织页面	24
1.5 利用多种 Web 浏览器执行测试	6	2.5 利用标题组织内容	26
1.6 创建一个示例文件	7	2.6 理解语义元素	29
1.7 使用 FTP 传输文件	8	2.7 以多种方式使用 <header> 元素	33
1.7.1 选择 FTP 客户端	8	2.8 理解 <section> 元素	34
1.7.2 使用 FTP 客户端	9	2.9 使用 <article> 元素	36
1.8 了解在 Web 服务器上放置文件的位置	11	2.10 实现 <nav> 元素	36
1.8.1 基本的文件管理	11	2.11 何时使用 <aside> 元素	37
1.8.2 使用索引页面	13	2.12 有效地使用 <footer> 元素	38
1.9 在不涉及 Web 服务器的情况下分发内容	13	2.13 小结	39
1.9.1 在本地发布内容	14	2.14 问与答	40
1.9.2 在博客上发布内容	14	2.15 测验	41
1.10 关于测试 Web 内容的提示	14	2.15.1 问题	41
1.11 小结	15	2.15.2 答案	41
1.12 问与答	16	2.15.3 练习	42
1.13 测验	16	第 3 章 理解层叠样式表	43
1.13.1 问题	16	3.1 CSS 的工作方式	44
1.13.2 答案	17		
1.13.3 练习	17		

3.2	基本的样式表	45	第 5 章	验证和调试代码	76
3.3	CSS 样式的基础知识	49	5.1	验证 Web 内容	76
3.3.1	基本的布局属性	50	5.2	使用 Developer Tools 调试 HTML 和 CSS	78
3.3.2	基本的格式化属性	51	5.2.1	使用检查器调试 HTML	80
3.4	使用样式类	53	5.2.2	使用检查器调试 CSS	84
3.5	使用样式 ID	55	5.3	使用 Developer Tools 调试 JavaScript 代码	88
3.6	内部样式表和内联样式	56	5.4	小结	93
3.7	小结	58	5.5	问与答	93
3.8	问与答	59	5.6	作业	93
3.9	测验	59	5.6.1	测验	93
3.9.1	问题	59	5.6.2	答案	94
3.9.2	答案	60	5.6.3	练习	94
3.9.3	练习	60	第 6 章	处理字体、文本块、列表和表格	95
第 4 章	理解 JavaScript	61	6.1	处理特殊字符	96
4.1	学习 Web 脚本编程的基础知识	62	6.2	粗体、斜体和特殊的文本格式化	99
4.1.1	脚本和程序	62	6.3	调整字体	102
4.1.2	JavaScript 简介	62	6.4	使用 Web 字体	106
4.2	JavaScript 如何适应 Web 页面	63	6.5	在页面上对齐文本	108
4.2.1	使用单独的 JavaScript 文件	64	6.5.1	使用属性	108
4.2.2	理解 JavaScript 事件	65	6.5.2	对齐块级元素	108
4.3	探索 JavaScript 的能力	65	6.6	三类 HTML 列表	110
4.3.1	改进导航	65	6.7	在列表内放置列表	112
4.3.2	验证表单	66	6.8	创建一个简单的表格	117
4.3.3	特殊效果	66	6.9	控制表格大小	121
4.3.4	远程脚本调用 (AJAX)	66	6.10	表格内的对齐和跨越	123
4.4	利用 JavaScript 显示时间	66	6.11	利用表格进行页面布局	126
4.4.1	在变量中存储数据	67	6.12	使用 CSS 栏	127
4.4.2	计算结果	68	6.13	小结	130
4.4.3	创建输出	68	6.14	问与答	132
4.4.4	向 Web 页面中添加脚本	68	6.15	测验	132
4.5	测试脚本	70	6.15.1	问题	132
4.5.1	修改脚本	70	6.15.2	答案	133
4.5.2	处理 JavaScript 错误	72	6.15.3	练习	133
4.6	小结	73	第 7 章	使用外部链接和内部链接	135
4.7	问与答	73	7.1	使用 Web 地址	135
4.8	作业	74	7.2	使用锚在页面内链接	138
4.8.1	测验	74			
4.8.2	答案	75			
4.8.3	练习	75			

7.2.1 利用锚标识页面中的位置	138	8.17 使用背景图像	181
7.2.2 链接到锚位置	138	8.18 使用图像映射	182
7.3 在自己的 Web 内容之间 创建链接	141	8.18.1 为什么图像映射并非总是 必要的	183
7.4 链接到外部 Web 内容	144	8.18.2 映射图像内的区域	184
7.5 链接到电子邮件地址	144	8.18.3 为图像映射创建 HTML 代码	184
7.6 在新的浏览器窗口中 打开链接	146	8.19 链接到多媒体文件	186
7.7 使用 CSS 编排超链接 的样式	146	8.20 嵌入多媒体文件	190
7.8 小结	150	8.21 为音频和视频播放使用 纯 HTML5 代码	192
7.9 问与答	150	8.22 关于使用多媒体的额外提示	195
7.10 测验	151	8.23 小结	196
7.10.1 问题	151	8.24 问与答	199
7.10.2 答案	151	8.25 测验	199
7.10.3 练习	152	8.25.1 问题	200
<b>第 8 章 处理颜色、图像和多媒体</b>	<b>153</b>	8.25.2 答案	200
8.1 选择颜色的最佳实践	154	8.25.3 练习	200
8.2 理解 Web 颜色	155	<b>第 9 章 处理边距、填充、对齐和浮动</b>	<b>202</b>
8.3 使用十六进制的颜色值	157	9.1 使用边距	202
8.4 使用 CSS 设置背景、文本和 边框颜色	158	9.2 填充元素	209
8.5 选择图形软件	160	9.3 使所有的一切保持对齐	212
8.6 关于图形需要知道的 最少知识	161	9.4 理解 float 属性	212
8.7 准备照片图像	162	9.5 小结	215
8.7.1 裁剪图像	163	9.6 问与答	215
8.7.2 调整图像大小	164	9.7 测验	215
8.7.3 调整图像颜色	165	9.7.1 问题	215
8.7.4 控制 JPEG 压缩	166	9.7.2 答案	216
8.8 创建横幅和按钮	167	9.7.3 练习	216
8.9 减少或删除图像中的颜色	168	<b>第 10 章 理解 CSS 方框模型和定位</b>	<b>217</b>
8.10 创建拼贴式背景图像	169	10.1 CSS 方框模型	217
8.11 创建动画式 Web 图形	170	10.2 关于定位的独家新闻	220
8.12 在 Web 页面上放置图像	171	10.3 控制元素的堆叠方式	224
8.13 利用文本描述图像	174	10.4 管理文本流	226
8.14 指定图像的高度和宽度	174	10.5 小结	226
8.15 对齐图像	175	10.6 问与答	227
8.15.1 水平图像对齐	175	10.7 测验	227
8.15.2 垂直图像对齐	177	10.7.1 问题	227
8.16 把图像转变成链接	179	10.7.2 答案	227
		10.7.3 练习	228



第 11 章 使用 CSS 进一步处理列表、 文本和导航 .....	229	第 13 章 理解动态 Web 站点和 HTML5 应用程序 .....	268
11.1 HTML 列表复习 .....	230	13.1 理解不同类型的脚本 .....	268
11.2 CSS 方框模型如何影响 列表 .....	230	13.2 在 HTML 中包括 JavaScript 代码 .....	269
11.3 放置列表项指示符 .....	233	13.3 显示随机内容 .....	271
11.4 利用列表项和 CSS 创建 图像映射 .....	234	13.4 理解文档对象模型 .....	275
11.5 导航列表与常规的列表有 何区别 .....	237	13.5 基于用户交互更改图像 .....	275
11.6 利用 CSS 创建垂直导航 系统 .....	238	13.6 提前考虑开发 HTML5 应用 程序 .....	277
11.6.1 编排单级垂直导航 系统的样式 .....	240	13.7 小结 .....	278
11.6.2 编排多级垂直导航 系统的样式 .....	242	13.8 问与答 .....	278
11.7 利用 CSS 创建水平 导航系统 .....	246	13.9 测验 .....	278
11.8 小结 .....	249	13.9.1 问题 .....	279
11.9 问与答 .....	250	13.9.2 答案 .....	279
11.10 测验 .....	250	13.9.3 练习 .....	279
11.10.1 问题 .....	251	第 14 章 初识 JavaScript 程序设计 .....	280
11.10.2 答案 .....	251	14.1 基本概念 .....	280
11.10.3 练习 .....	251	14.1.1 语句 .....	280
第 12 章 创建固定布局或流动布局 .....	252	14.1.2 把任务与函数相结合 .....	281
12.1 理解固定布局 .....	253	14.1.3 变量 .....	282
12.2 理解流动布局 .....	254	14.1.4 了解对象 .....	282
12.3 创建固定/流动混合布局 .....	256	14.1.5 条件语句 .....	283
12.3.1 从基本的布局结构开始 .....	256	14.1.6 循环语句 .....	283
12.3.2 在固定/流动混合布局中 定义两栏 .....	258	14.1.7 事件处理程序 .....	283
12.3.3 设置布局的最小宽度 .....	260	14.1.8 首先运行哪个脚本 .....	285
12.3.4 在固定/流动混合布局中 处理栏高度 .....	260	14.2 JavaScript 语法规则 .....	286
12.4 考虑响应性 Web 设计 .....	265	14.2.1 大小写敏感性 .....	286
12.5 小结 .....	266	14.2.2 变量、对象和函数名称 .....	286
12.6 问与答 .....	266	14.2.3 保留字 .....	286
12.7 测验 .....	267	14.2.4 空白 .....	286
12.7.1 问题 .....	267	14.3 使用注释 .....	287
12.7.2 答案 .....	267	14.4 关于 JavaScript 的最佳 实践 .....	287
12.7.3 练习 .....	267	14.5 理解 JSON .....	288
		14.6 小结 .....	289
		14.7 问与答 .....	289
		14.8 作业 .....	289
		14.8.1 测验 .....	289
		14.8.2 答案 .....	290
		14.8.3 练习 .....	290

第 15 章 使用文档对象模型 (DOM) · 291	16.5 使用 String 对象····· 320
15.1 了解文档对象模型····· 291	16.5.1 创建 String 对象····· 321
15.2 使用 window 对象····· 292	16.5.2 赋值····· 321
15.3 处理 document 对象····· 292	16.5.3 计算字符串的长度····· 322
15.3.1 获取关于文档的信息····· 293	16.5.4 转换字符串的大小写····· 323
15.3.2 在文档中编写文本····· 294	16.6 处理子串····· 323
15.3.3 使用链接和锚····· 295	16.6.1 使用字符串的一部分····· 324
15.4 访问浏览器的历史记录····· 295	16.6.2 获取单个字符····· 324
15.5 使用 location 对象····· 297	16.6.3 查找子串····· 325
15.6 关于 DOM 结构的更多 知识····· 298	16.7 使用数值型数组····· 325
15.6.1 节点····· 299	16.7.1 创建数值型数组····· 325
15.6.2 父对象与子对象····· 299	16.7.2 了解数组长度····· 326
15.6.3 兄弟对象····· 300	16.7.3 访问数组元素····· 327
15.7 处理 DOM 节点····· 300	16.8 使用字符串数组····· 327
15.7.1 基本的节点属性····· 300	16.8.1 创建字符串数组····· 327
15.7.2 节点的关系属性····· 301	16.8.2 拆分字符串····· 328
15.7.3 文档方法····· 301	16.8.3 对字符串数组进行排序····· 328
15.7.4 节点方法····· 301	16.9 对数值型数组进行排序····· 328
15.8 创建可定位的元素 (图层)····· 302	16.10 小结····· 331
15.9 隐藏和显示对象····· 306	16.11 问与答····· 331
15.10 修改页面内的文本····· 308	16.12 作业····· 332
15.11 向页面中添加文本····· 309	16.12.1 测验····· 332
15.12 小结····· 311	16.12.2 答案····· 333
15.13 问与答····· 311	16.12.3 练习····· 333
15.14 作业····· 311	第 17 章 使用 JavaScript 函数和 对象····· 334
15.14.1 测验····· 311	17.1 使用函数····· 334
15.14.2 答案····· 312	17.1.1 定义函数····· 335
15.14.3 练习····· 312	17.1.2 调用函数····· 336
第 16 章 使用 JavaScript 变量、字符串 和数组····· 314	17.1.3 返回值····· 338
16.1 使用变量····· 314	17.2 介绍对象····· 339
16.1.1 选择变量名····· 315	17.2.1 创建对象····· 340
16.1.2 使用局部变量和全局变量····· 315	17.2.2 了解方法····· 340
16.1.3 给变量赋值····· 316	17.3 使用对象简化脚本编程····· 341
16.2 了解表达式和运算符····· 317	17.3.1 定义对象····· 341
16.2.1 使用 JavaScript 运算符····· 317	17.3.2 定义对象的方法····· 342
16.2.2 运算符优先级····· 318	17.3.3 创建对象的实例····· 343
16.3 JavaScript 中的数据类型····· 319	17.4 扩展内置对象····· 345
16.4 在数据类型之间转换····· 320	17.5 使用 Math 对象····· 347
	17.5.1 四舍五入和截尾····· 347
	17.5.2 生成随机数····· 348
	17.5.3 其他 Math 方法····· 348
	17.6 使用 Math 方法····· 348

17.7 处理日期.....	350	19.1.1 对象和事件.....	375
17.7.1 创建 Date 对象.....	351	19.1.2 创建事件处理程序.....	375
17.7.2 设置 Date 值.....	351	19.1.3 利用 JavaScript 定义事件 处理程序.....	376
17.7.3 读取 Date 值.....	351	19.1.4 支持多个事件处理程序.....	376
17.7.4 处理时区.....	352	19.1.5 使用 event 对象.....	377
17.7.5 在日期格式之间转换.....	352	19.2 使用鼠标事件.....	378
17.8 小结.....	353	19.2.1 移入和移出.....	379
17.9 问与答.....	353	19.2.2 按下和释放(以及单击).....	379
17.10 作业.....	353	19.3 使用键盘事件.....	382
17.10.1 测验.....	353	19.4 使用 onload 和 onunload 事件.....	384
17.10.2 答案.....	354	19.5 使用 onclick 更改<div>的 外观.....	385
17.10.3 练习.....	354	19.6 小结.....	391
<b>第 18 章 利用条件和循环控制流程</b> .....	<b>355</b>	19.7 问与答.....	391
18.1 if 语句.....	356	19.8 作业.....	391
18.1.1 条件运算符.....	356	19.8.1 测验.....	391
18.1.2 利用逻辑运算符把条件 结合起来.....	357	19.8.2 答案.....	392
18.1.3 else 关键字.....	358	19.8.3 练习.....	392
18.2 使用简写的条件表达式.....	359	<b>第 20 章 使用窗口</b> .....	<b>393</b>
18.3 利用 if 和 else 测试多个 条件.....	359	20.1 利用对象控制窗口.....	393
18.3.1 HTML 文件.....	360	20.1.1 window 对象的属性.....	393
18.3.2 JavaScript 文件.....	361	20.1.2 创建新窗口.....	394
18.4 利用 switch 使用多个 条件.....	362	20.1.3 打开和关闭窗口.....	395
18.5 使用 for 循环.....	363	20.2 移动和调整窗口.....	397
18.6 使用 while 循环.....	365	20.3 使用定时器.....	398
18.7 使用 do...while 循环.....	366	20.4 显示对话框.....	400
18.8 使用循环.....	366	20.5 小结.....	402
18.8.1 创建无限循环.....	367	20.6 问与答.....	402
18.8.2 退出循环.....	367	20.7 作业.....	402
18.8.3 继续执行循环.....	368	20.7.1 测验.....	403
18.9 遍历对象属性.....	368	20.7.2 答案.....	403
18.10 小结.....	371	20.7.3 练习.....	403
18.11 问与答.....	371	<b>第 21 章 JavaScript 最佳实践</b> .....	<b>404</b>
18.12 作业.....	372	21.1 脚本编程的最佳实践.....	404
18.12.1 测验.....	372	21.1.1 不要过度使用 JavaScript.....	405
18.12.2 答案.....	373	21.1.2 内容、表示和行为.....	405
18.12.3 练习.....	373	21.1.3 优雅降级.....	406
<b>第 19 章 响应事件</b> .....	<b>374</b>	21.1.4 渐进增强.....	407
19.1 了解事件处理程序.....	374	21.1.5 添加事件处理程序.....	407
		21.1.6 Web 标准: 避免特定于浏览器.....	409

21.1.7 很好地处理错误	409	23.3 选择 DOM 和 CSS 内容	439
21.1.8 注解代码	410	23.4 操作 HTML 内容	439
21.1.9 有用性	411	23.4.1 显示和隐藏元素	441
21.1.10 可访问性	412	23.4.2 制作元素的动画	442
21.1.11 设计模式	412	23.5 结合使用各种方法来创建 jQuery 动画	443
21.1.12 尽可能地重用代码	412	23.6 利用 jQuery 处理事件	446
21.2 读取浏览器信息	413	23.7 小结	447
21.2.1 显示浏览器信息	414	23.8 问与答	448
21.2.2 应对不诚实的浏览器	415	23.9 作业	448
21.3 跨浏览器的脚本编程	416	23.9.1 测验	448
21.3.1 特性检测	416	23.9.2 答案	449
21.3.2 应对浏览器的怪癖	417	23.9.3 练习	449
21.4 支持未启用 JavaScript 的 浏览器	418	<b>第 24 章 利用 jQuery UI 创建丰富交互 的先期步骤</b>	450
21.4.1 使用<noscript>标签	418	24.1 准备使用 jQuery UI	450
21.4.2 保持 JavaScript 是可选的	419	24.2 使用 jQuery UI 中的 选择器	451
21.4.3 避免错误	419	24.3 利用 jQuery UI 定位 UI 元素	452
21.5 小结	423	24.4 应用 jQuery UI 特效	456
21.6 问与答	423	24.5 把 jQuery UI 构件用于高级 交互	463
21.7 作业	423	24.5.1 理解鼠标交互构件	464
21.7.1 测验	423	24.5.2 实现可拖动的项目	464
21.7.2 答案	424	24.6 接下来何去何从	472
21.7.3 练习	424	24.7 小结	473
<b>第 22 章 使用第三方库和框架</b>	425	24.8 问与答	473
22.1 使用第三方 JavaScript 库	425	24.9 作业	474
22.1.1 jQuery	426	24.9.1 测验	474
22.1.2 Prototype	428	24.9.2 答案	474
22.1.3 Script.aculo.us	428	24.9.3 练习	474
22.1.4 其他流行的 JavaScript 库	429	<b>第 25 章 AJAX: 远程脚本编程</b>	475
22.2 使用第三方库添加 JavaScript 特效	429	25.1 AJAX 简介	475
22.3 使用 JavaScript 框架	432	25.1.1 JavaScript 客户 (前端)	476
22.4 小结	433	25.1.2 服务器端脚本 (后端)	476
22.5 问与答	434	25.1.3 XML	477
22.6 作业	434	25.1.4 流行的 AJAX 示例	477
22.6.1 测验	434	25.2 使用 XMLHttpRequest	478
22.6.2 答案	435	25.2.1 创建请求	478
22.6.3 练习	435	25.2.2 打开 URL	478
<b>第 23 章 详细探讨 jQuery</b>	436		
23.1 准备使用 jQuery	436		
23.2 熟悉\$(document).ready 处理 程序	437		

25.2.3	发送请求	478	26.7.2	单选按钮	505
25.2.4	等待响应	479	26.7.3	选择列表	506
25.2.5	解释响应数据	479	26.7.4	文本框、文本区及其他 输入类型	508
25.3	创建简单的 AJAX 库	479	26.8	使用 HTML5 表单验证	509
25.3.1	ajaxRequest 函数	480	26.9	提交表单数据	512
25.3.2	ajaxResponse 函数	480	26.10	利用 JavaScript 访问表单 元素	513
25.3.3	使用库	481	26.11	显示来自表单的数据	513
25.4	使用库创建 AJAX 测验	481	26.12	小结	515
25.4.1	HTML 文件	481	26.13	问与答	517
25.4.2	XML 文件	482	26.14	作业	517
25.4.3	JavaScript 文件	483	26.14.1	测验	517
25.4.4	测试示例	484	26.14.2	答案	517
25.5	调试基于 AJAX 的应用 程序	485	26.14.3	练习	518
25.6	将 jQuery 的内置函数 用于 AJAX	490	第 27 章	组织和管理 Web 站点	519
25.7	小结	491	27.1	当一个页面就足够时	520
25.8	问与答	491	27.2	组织简单的站点	521
25.9	作业	492	27.3	组织较大的站点	524
25.9.1	测验	492	27.4	编写易于维护的代码	526
25.9.2	答案	492	27.4.1	利用注释给代码加注解	527
25.9.3	练习	492	27.4.2	缩进代码以使之清晰	528
第 26 章	处理基于 Web 的表单	493	27.5	考虑版本控制	529
26.1	HTML 表单的工作方式	494	27.6	使用 HTML 和 CSS 框架	530
26.2	创建表单	494	27.7	小结	531
26.3	接受文本输入	499	27.8	问与答	531
26.4	命名各种表单数据	500	27.9	作业	532
26.5	标记各种表单数据	501	27.9.1	测验	532
26.6	组合表单元素	502	27.9.2	答案	532
26.7	探索表单输入控件	503	27.9.3	练习	532
26.7.1	复选框	503			

# 第 1 章

## 理解 Web 的工作方式

---

在本章中你将学到：

- WWW（World Wide Web，万维网）简史；
- 术语 Web 页面的含义是什么，为什么这个术语并不总会反映所涉及的全部内容；
- 怎样把内容从你的个人计算机发送到别人的 Web 浏览器上；
- 怎样选择 Web 托管提供商；
- 不同的 Web 浏览器和设备类型可能怎样影响内容；
- 如何使用 FTP 把文件传输到 Web 服务器上；
- 应该把文件存放在 Web 服务器上的什么位置；
- 在不涉及 Web 服务器的情况下怎样分发 Web 内容；
- 如何使用其他发布方法，比如博客；
- 关于测试 Web 内容的外观和功能的提示。

在学习 HTML（Hypertext Markup Language，超文本标记语言）、CSS（Cascading Style Sheets，层叠样式表）和 JavaScript 的复杂知识之前，获得对技术的稳固理解很重要，它们有助于把这些纯文本文件转换成你在计算机或手持设备上浏览 WWW 时看到的丰富的多媒体显示。

例如，如果不使用 Web 浏览器查看，那么包含标记和客户端代码的 HTML 和 CSS 文件将是无用的，并且除非使用 Web 服务器，否则除你之外的其他人将无法查看你的内容。Web 服务器使你的内容可供其他人使用，他们反过来又使用其 Web 浏览器导航到一个地址并等待服务器向自己发送信息。你将密切地参与这个发布过程，因为你必须创建文件，然后把它们

放到服务器上，使它们在第一时间可用，并且必须确保你的内容像预期的那样出现在最终用户面前。

## 1.1 HTML 和 WWW 简史

在月球上还没有任何人类足迹的时候，一些有远见的人决定去看看他们是否能够连接几个主要的计算机网络。我将和你分享它们的名称和故事（可以说非常丰富），但是最终的结果是“所有网络之母”，我们称之为 Internet。

到 1990 年，通过 Internet 访问信息还相当具有技术性。事实上，它是如此困难，以至于那些拥有博士学位的物理学家们在尝试交换数据时也通常会遭受挫折。有这样一位物理学家，即现在非常著名的 Tim Berners-Lee 爵士，他设计了一种方式，可以通过超文本链接轻松地交叉引用 Internet 上的文本。

这不是一种新思想，但是他开发的简单 HTML 却成功地兴盛起来，虽然更多雄心勃勃的超文本项目步履维艰。超文本最初意指以电子形式存储的文本，并且在页面之间具有交叉引用的链接。它现在是一个更宽泛的术语，指可以链接到其他对象的任何几何对象（文本、图像、文件等）。超文本标记语言（Hypertext Markup Language）是一种用于描述如何组织和链接文本、图形以及包含其他信息的文件的语言。

到 1993 年，全世界只有 100 台左右的计算机被配备成提供 HTML 页面。这些互连的页面被称为 WWW（World Wide Web，万维网），并且编写了几个 Web 浏览器程序，允许人们查看 Web 页面。由于 Web 日益普及，几个程序员不久之后编写了可以查看文本以及图形图像的 Web 浏览器。从此，Web 浏览器软件的持续开发以及 HTML（和 XHTML）语言的标准化管理带领我们进入了今天的世界，在这个世界里，有超过 5 亿个 Web 站点提供数十亿个文本和多媒体文件。

上面几段文字实际上描述了 Web 发展简史中的一个不同寻常的阶段。今天的大学生从来也不了解 WWW 不存在的那段时间，并且永远在线的信息和无处不在的计算的思想对我们以后生活的各个方面产生深远的影响。不要把 Web 内容创建和管理视作少数技术人员（如果你愿意的话，也可以称他们为怪才）所拥有的一组技能，到学完本书时，你将把 Web 看作是任何人都可以掌握的技能，而不仅限于那些天生的怪才。

**By the Way**

**注意：**

有关 WWW 发展历史的更多信息，参见关于这个主题的维基百科的文章：[http://en.wikipedia.org/wiki/History\\_of\\_the\\_Web](http://en.wikipedia.org/wiki/History_of_the_Web)。

## 1.2 创建 Web 内容

你可能注意到本书中使用的术语是“Web 内容”（Web content），而不是“Web 页面”（Web page），我们是有意这样做的。尽管我们谈论的是“访问 Web 页面”，但是实际的意思是“查看我们计算机上的一个地址上的所有文本和图像”。我们阅读的文本和查看的图像都是通过 Web 浏览器呈现的。浏览器将按照文件中的指示来呈现它们。

这些文件可以包含被 HTML 代码标记或包围的文本，告诉浏览器如何显示文本——作为标题、作为段落或者以红色字体显示等。一些 HTML 标记告诉浏览器显示图像或视频文件，而不是纯文本，由此得出以下结论：发送给 Web 浏览器的是不同类型的内容，因此仅仅称作 Web 页面将无法完全涵盖它们。我们在这里代之以使用 Web 内容这个术语，以涵盖文本、图像、音频、视频以及在线发现的其他媒体的完整范围。

在后面几章中，你将学习链接到或创建在 Web 站点中发现的各类多媒体 Web 内容的基础知识。此时，你只需记住你正在控制用户在访问你的 Web 站点时所查看的内容。从一个文件开始，其中包含要显示的文本或者告诉服务器给用户的 Web 浏览器发送一幅图形，你将不得不计划、设计和实现最终构成 Web 状态（Web presence）的所有部分。如你将在整本书中所学到的，它不是一个困难的过程，只要在学习过程中理解每一个小步骤即可。

就其最基本的形式而言，Web 内容开始于一个包含 HTML 标记的简单文本文件。在本书中，你将学习和创建符合标准的 HTML5 标记。编写符合标准的代码的一个好处是：在将来，将无需担心必须要回顾代码，以从根本上改变它。这样，你的代码将（很可能）总是适用的，只要 Web 浏览器遵守标准即可（在一段较长的时间内有望如此）。

## 1.3 理解 Web 内容递送

在许多不同的位置会发生多个过程，最终将产生你可以查看的 Web 内容。这些过程将发生得非常快（以毫秒级的速度），并且是在幕后发生的。换句话说，尽管我们可能认为我们所做的全部事情是打开 Web 浏览器，输入一个 Web 地址，并立即查看所请求的内容，后台的技术就会代表我们努力工作。图 1.1 显示了浏览器与服务器之间的基本交互。



图 1.1

浏览器请求和服务器响应

不过，在看到所请求的站点的全部内容之前，这个进程将会涉及多个步骤，并且可能在浏览器与服务器之间会往返多次。

假设你想执行 Google 搜索，必然会在地址栏中输入“www.google.com”，或者从书签列表中选择 Google 书签。浏览器几乎立即就会显示如图 1.2 所示的内容。

图 1.2 显示了一个包含文本以及一幅图像（Google 标志）的 Web 站点。用于从 Web 服务器获取文本和图像并将其显示在屏幕上的过程的简单版本如下。

1. Web 浏览器对位于 www.google.com 地址上的一个 index.html 文件发送请求。index.html 文件不必是在地址栏中输入的地址的一部分，在学习本章后面的内容时，将会了解关于 index.html 文件的更多知识。

2. 在收到对特定文件的请求后，Web 服务器进程将在其目录内容中寻找特定的文件，打开它，并把该文件的内容发送回 Web 浏览器。



图 1.2

访问 [www.google.com](http://www.google.com)



3. Web 浏览器接收 `index.html` 文件的内容，该文件是利用 HTML 代码标记的文本，并基于这些 HTML 代码呈现内容。在呈现内容时，浏览器将遇到用于 Google 标志的 HTML 代码，在图 1.2 中可以看到该标志。HTML 代码看起来如下所示：

```

```

这个 HTML 标签提供了一些属性，告诉浏览器显示这个标志所必需的信息：文件源地址 (`src`)、宽度 (`width`) 和高度 (`height`)。在后面各章中将会学习关于属性的更多知识。

4. 浏览器查看 `<img/>` 标签中的 `src` 属性，查找源位置。在这种情况下，可以在与浏览器获取 HTML 文件相同的 Web 地址 (`www.google.com`) 上的 `images` 目录中找到 `logo3w.png` 图像。

5. 浏览器请求位于 Web 地址 `http://www.google.com/images/srpr/logo4w.png` 上的文件。
6. Web 服务器解释该请求，找到文件，并把该文件的内容发送给请求它的 Web 浏览器。
7. Web 浏览器在显示器上显示图像。

在 Web 内容递送过程的描述中可以看到，Web 浏览器所做的不仅仅是充当用于查看内容的图片框架。浏览器将会依据文件中的 HTML 命令组合 Web 内容成分，并排列那些部分。

也可以在本地或者在你自己的硬盘驱动器上查看 Web 内容，从而无需 Web 服务器。获取和显示内容的过程与上述步骤中列出的过程相同，这是由于浏览器将寻找并解释 HTML 文件的代码和内容，但是往返行程更短：浏览器将在你自己的计算机的硬盘驱动器上（而不是在远程机器上）寻找文件。如果文件中嵌入了任何基于服务器的程序设计语言，将需要 Web 服务器解释它们，但是这超出了本书的范围。事实上，无需拥有自己的 Web 服务器，就可以顺利学完本书中的所有课程，但是这样的话，除你自己之外其他任何人都将无法查看你的杰作。需要全本请在线购买：[www.epitongbook.com](http://www.epitongbook.com)