

HTML 5

网页设计与制作全程揭秘

贾勇 编著

1000个素材源文件

130个实战案例

635分钟视频教程



Book features



内容全面，循序渐进

遵循从基础到高级，从入门到精通的思路，帮助读者轻松建立起正确的知识体系。



技术专业，易于操作

在技术讲解的同时，配合丰富的操作练习，理论与操作相结合，使读者轻松理解和掌握知识点。



专业作者，如临亲授

作者来自于一线专业公司和高校教师，实践与教学的融合，更清楚学习者的知识需求。



扫码微课，碎片学习

读者可以扫描书中的二维码，利用零碎的时间随时随地通过视频课程进行学习。



源于实践，回归实战

穿插大量的实战操作案例，均来源于实践，通过一步步操作的方式再现给读者。



海量资源，辅助学习

提供立体化的学习资源：所有案例的素材、源文件；多媒体语音视频教程；与教学配套的PPT课件。

KEYWORD

专业学习手册+完全配套资源+PPT课件+6部网页设计电子书+海量辅助设计资源+作者技术交流群

SUBSCRIBE

清华大学出版社

内 容 简 介

经过了 Web 2.0 时代, 基于互联网的应用已经越来越丰富, 同时也对互联网应用提出了更高的要求。如今, HTML5 俨然已经成为互联网领域热门的词语之一。本书按照循序渐进的思路, 系统全面地讲解了 HTML5 语言中的所有功能和特性。

全书共分 21 章, 包括从 HTML 到 HTML5、HTML5 页面基本设置、设置文本与段落、插入并设置图像、创建和设置列表、创建和设置超链接、插入多媒体、插入和设置表单元素、表格与 Div、HTML5 文档结构、使用 HTML5 画布绘图、HTML5 的音频和视频、使用 HTML5 的表单元素、文件与拖放处理、HTML5 本地存储、HTML5 离线应用缓存、使用 Web Workers 处理线程、跨源通信和 WebSocket 双向通信、使用 HTML5 获取地理位置、HTML5 网页综合实战、HTML5 手机网页实战。书中所有知识点都结合具体实战练习进行讲解, 涉及的程序代码给出了详细的注释说明, 可以使读者轻松理解 HTML5 语言的精髓, 快速掌握 HTML5 的应用。

本书适合 Web 设计与开发的初学者和爱好者自学, 也适合有一定 Web 前端开发基础的网页开发人员阅读, 同时也可作为计算机培训班和各院校相关专业的教材。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目 (CIP) 数据

HTML5 网页设计与制作全程揭秘 / 贾勇 编著. — 北京: 清华大学出版社, 2019
(网页设计与开发殿堂之路)
ISBN 978-7-302-52699-5

I . ①H… II . ①贾… III . ①超文本标记语言—程序设计 IV . ①TP312

中国版本图书馆 CIP 数据核字 (2019) 第 057419 号

责任编辑: 李 磊 焦昭君

封面设计: 王 晨

版式设计: 思创景点

责任校对: 牛艳敏

责任印制: 李红英

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦A座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 北京富博印刷有限公司

装 订 者: 北京市密云县京文制本装订厂

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 23.75 字 数: 686千字

版 次: 2019年8月第1版 印 次: 2019年8月第1次印刷

定 价: 69.80元

产品编号: 077885-01

前言



随着互联网信息技术的发展，特别是移动互联网的迅速崛起，原来的网页描述语言 HTML 已经不能满足日益丰富的网页设计制作需要。HTML5 是下一代 HTML 的标准，与 HTML4 相比，HTML5 的发展有着革命性的进步。在 HTML5 中，不但新增了许多全新的实用功能，而且对涉及的每一个细节都做出了明确的规定。HTML5 以其简洁、高效的特点，在网页中的应用越来越广泛。

本书全面系统地向读者介绍 HTML5 中的各个知识点，每个重要的知识点都配合实战练习进行讲解，将知识点与实战练习紧密结合，避免枯燥无味的基础知识讲解，使读者能够更加轻松地掌握和应用最新的 HTML5 网页开发技术，提高学习效率，并能够学以致用。全书共分 21 章，各章内容介绍如下。

第 1 章 从 HTML 到 HTML5，介绍 HTML、XHTML 和 HTML5 的相关基础知识，使读者能够清晰地了解 HTML 的发展，以及 HTML、XHTML 和 HTML5 之间的联系和区别，认识到 HTML5 的优势。

第 2 章 HTML5 页面基本设置，介绍如何在 HTML 中对网页头部 <head> 标签和网页主体 <body> 标签进行设置，从而达到控制网页整体属性的目的。

第 3 章 设置文本与段落，文本是网页中重要的基本元素之一。本章详细介绍 HTML5 中对文本和段落进行设置的相关标签和属性设置方法，使读者能够轻松地对网页中的文本和段落进行处理。

第 4 章 插入并设置图像，介绍在 HTML5 页面中插入图像的方法，以及在图像标签中添加各种属性对图像进行设置的方法。

第 5 章 创建和设置列表，介绍在 HTML5 中创建项目列表、编号列表和定义列表的方法，以及对各种列表进行设置的相关属性。

第 6 章 创建和设置超链接，超链接是互联网的基础。本章详细介绍网页超链接的创建和设置方法，以及各种特殊链接的创建方法，并且还介绍了链接路径的相关知识。

第 7 章 插入多媒体，介绍如何在 HTML5 页面中应用各种不同类型的多媒体元素，包括 Flash 动画、背景音乐和普通视频文件等。

第 8 章 插入和设置表单元素，表单是网页交互的重要途径。本章主要介绍在 HTML5 页面中插入各种不同类型的表单元素，以及对表单元素属性进行设置的方法和技巧。

第 9 章 表格与 Div，主要介绍在 HTML5 页面中创建表格以及对表格和单元格属性进行设置的方法和技巧，还介绍了在 HTML5 页面中插入 IFrame 框架和应用 Div 的方法。

第 10 章 HTML5 文档结构，主要介绍 HTML5 中新增的文档结构标签的作用和使用方法，以及如何通过使用 HTML5 文档结构标签创建规范的 HTML5 文档。

第 11 章 使用 HTML5 画布绘图，canvas 元素是 HTML5 的亮点之一，通过使用 canvas 元素可以在网页中绘制出各种几何图形。本章详细介绍使用 HTML5 中的 canvas 元素在网页中绘制图形、文字、渐变的方法。

第 12 章 HTML5 的音频和视频，多媒体的应用也是 HTML5 的一大亮点。本章详细介绍



HTML5 中 Video 与 Audio 元素的使用方法和属性设置技巧。

第 13 章 使用 HTML5 的表单元素, 在 HTML5 中新增了许多表单属性和表单元素, 通过这些新增的表单属性可以方便地对表单元素的有效性进行验证, 新的表单元素则能够方便用户创建出更加友好的表单应用。

第 14 章 文件与拖放处理, 本章详细介绍 HTML5 中新增的文件 API 和拖放 API 功能, 通过文件 API 功能可以同时上传多个文件, 通过拖放 API 功能可以实现网页元素拖放处理。

第 15 章 HTML5 本地存储, 介绍 HTML5 中的两种本地存储方式, 分别是 Web Storage 和 Web SQL 数据库, 通过使用 HTML5 的本地存储功能, 可以更轻松地开发 Web 应用程序。

第 16 章 HTML5 离线应用缓存, 介绍 HTML5 中新增的离线应用缓存功能, 包括缓存清单文件的编写方式、离线应用缓存的应用等内容。

第 17 章 使用 Web Workers 处理线程, 介绍 Web Workers 的相关知识, 重点讲解专属线程和共享线程的作用、创建和使用方法等内容。

第 18 章 跨源通信和 WebSocket 双向通信, 介绍 HTML5 中新增的跨源通信和 WebSocket 双向通信的相关知识, 实现简单的跨文档信息传输, 对跨源通信和 WebSocket 有全新的认识 and 了解。

第 19 章 使用 HTML5 获取地理位置, 介绍获取地理位置信息的原理, 以及使用 HTML5 中新增的 Geolocation API 实现获取用户地理位置信息的方法。

第 20 章 HTML5 网页综合实战, 通过两个网站页面案例的制作讲解, 使读者能够更加轻松地掌握使用 HTML5 中的各种标签制作符合 HTML5 标准的网站页面的方法和技巧。

第 21 章 HTML5 手机网页实战, 介绍响应式网页的制作方法和注意事项, 并通过响应式网页实例的制作讲解, 使读者能够轻松地掌握制作 HTML5 手机网页的方法。

本书由贾勇编著, 另外张晓景、李晓斌、高鹏、胡敏敏、张国勇、林秋、胡卫东、姜玉声、周晓丽、郭慧等人也参与了部分编写工作。本书在写作过程中力求严谨, 由于作者水平所限, 书中难免有疏漏和不足之处, 希望广大读者批评、指正, 欢迎与我们沟通和交流。QQ 群名称: 网页设计与开发交流群; QQ 群号: 705894157。

为了方便读者学习, 本书为每个实例提供了教学视频, 只要扫描一下书中实例名称旁边的二维码, 即可直接打开视频进行观看, 或者推送到自己的邮箱中下载后进行观看。本书配套的立体化学习资源中提供了书中所有实例的素材源文件、最终文件、教学视频和 PPT 课件, 并附赠海量实用资源。读者在学习时可扫描下面的二维码, 然后将内容推送到自己的邮箱中, 即可下载获取相应的资源 (注意: 请将这两个二维码下的压缩文件全部下载完毕后, 再进行解压, 即可得到完整的文件内容)。



编 者

Search

目录



第1章 从 HTML 到 HTML5

1.1 HTML 基础	1
1.1.1 HTML 概述	1
1.1.2 HTML 特性	1
1.1.3 HTML 文档结构	2
1.1.4 HTML 的基本语法	2
1.1.5 HTML 编写注意事项	3
1.2 XHTML 基础	3
1.2.1 XHTML 概述	3
1.2.2 XHTML 文档结构	4
1.2.3 XHTML 的文档类型	5
1.2.4 名字空间	5
1.3 HTML5 基础	5
1.3.1 HTML5 概述	6
1.3.2 HTML5 的文档结构	6
1.3.3 HTML5 的优势	6
1.3.4 HTML5 精简的头部	7
1.4 HTML 文件的编写方式	8
1.4.1 使用记事本编写	8
实战 使用记事本制作 HTML 页面	8
1.4.2 使用 Dreamweaver 编写	9
实战 使用 Dreamweaver 制作 HTML 页面	10
1.5 HTML5 中的标签	11
1.6 HTML5 的标准属性	13
1.7 HTML5 的事件属性	14

第2章 HTML5 页面基本设置

2.1 网页头部——<head> 标签	16
2.1.1 网页标题——<title> 标签	16
实战 使用 <title> 标签设置网页标题	17
2.1.2 基底网址——<base> 标签	17
实战 设置网页基底网址	18

2.2 元信息——<meta> 标签	18
2.2.1 设置网页关键字	19
2.2.2 设置网页说明	19
2.2.3 设置网页作者信息	20
2.2.4 设置网页编辑软件	20
实战 设置网页基础元信息	20
2.2.5 设置网页定时跳转	21
实战 设置网页定时跳转	22
2.2.6 限制搜索方式	22
2.2.7 设置网页文字及语言	23
2.2.8 设置网页有效期限	23
2.2.9 禁止缓存调用	24
2.2.10 删除过期的 cookie	24
2.2.11 强制打开新窗口	24
2.3 网页主体——<body> 标签	25
2.3.1 网页边距——margin 属性	25
实战 设置网页整体边距	25
2.3.2 网页背景颜色——bgcolor 属性	26
实战 设置网页背景颜色	27
2.3.3 背景图像——background 属性	27
实战 设置网页背景图像	28
2.3.4 文字颜色——text 属性	28
实战 设置网页默认文字颜色	29
2.3.5 链接文字颜色——link 属性	29
实战 设置网页默认链接文字颜色	29
2.4 在 HTML 代码中添加注释	30

第3章 设置文本与段落

3.1 设置文本效果	32
3.1.1 文字样式—— 标签	32
实战 设置网页文字基本样式	33
3.1.2 倾斜文字——<i> 和 标签	34
实战 设置文字倾斜效果	34



3.1.3 加粗文字—— 和 	35
标签	35
实战 设置网页文字加粗效果	36
3.1.4 下画线——<u> 标签	36
实战 为文字添加下画线	37
3.1.5 删除线——<s> 和 <strike> 标签	37
实战 为文字添加删除线	37
3.1.6 上标与下标——<sup> 和 <sub>	38
标签	38
实战 设置上标和下标文字效果	38
3.1.7 等宽文本——<code> 和 <samp>	39
标签	39
3.1.8 标题文本——<h1> 至 <h6> 标签	39
实战 设置标题文字	40
3.2 特殊文字标签	40
3.2.1 空格—— 	40
实战 在网页文本中添加空格	41
3.2.2 其他特殊符号	41
实战 在网页中添加特殊文本符号	42
3.2.3 水平线——<hr> 标签	42
实战 在网页中插入并设置水平线	42
3.3 文本的分行与分段	43
3.3.1 文本换行—— 标签	43
实战 为网页文本进行换行处理	43
3.3.2 强制不换行——<no> 标签	44
实战 强制网页文本不换行	44
3.3.3 文本分段——<p> 标签	45
实战 为网页文本进行分段处理	45
3.3.4 段落文字对齐——align 属性	46
实战 设置文字水平对齐效果	46
3.3.5 保留原始排版——<pre> 标签	47
实战 保留网页文本原始排版效果	47
3.4 其他文字标签	47
3.4.1 文字标注——<ruby> 标签	47
实战 在网页中实现文字标注说明效果	47
3.4.2 声明变量——<var> 标签	48
实战 使用声明变量标签	48
3.4.3 忽视 HTML——<plaintext> 和 <xmp>	49
标签	49
实战 使用忽视 HTML 标签	49
3.5 滚动文本——<marquee> 标签	50
实战 制作网页文本滚动效果	50

第4章 插入并设置图像

4.1 了解网页中的图像格式	52
----------------	----

4.1.1 网页常用的图像格式	52
4.1.2 选择合适的图像格式	53
4.2 插入图像—— 标签	53
实战 制作图像页面	54
4.3 设置图像属性	54
4.3.1 图像宽度和高度——width 和 height	55
属性	55
实战 插入图像并设置图像宽度和高度	55
4.3.2 图像边框——border 属性	56
实战 为图像添加边框	57
4.3.3 图像替代文本——alt 属性	57
实战 为图像添加替代文本和提示文字信息	58
4.3.4 图像相对于文字的对齐方式——align	59
属性	59
实战 设置图像相对于文字的对齐效果	59
4.3.5 图文混排效果——align 属性	60
实战 制作图文混排页面	60
4.4 滚动图像——<marquee> 标签	62
实战 制作滚动图像效果	62

第5章 创建和设置列表

5.1 认识列表标签	64
5.2 项目列表	64
5.2.1 创建项目列表—— 标签	64
实战 创建新闻列表	65
5.2.2 项目列表符号——type 属性	66
实战 设置列表符号效果	67
5.3 编号列表	68
5.3.1 创建编号列表—— 标签	68
实战 创建排行列表	69
5.3.2 编号列表符号——type 属性	69
实战 设置编号列表符号	70
5.3.3 编号列表起始值——start 属性	71
实战 设置编号列表起始数值	72
5.4 定义列表——<dl> 标签	73
实战 制作复杂新闻列表	73

第6章 创建和设置超链接

6.1 超链接基础	75
6.1.1 什么是超链接	75
6.1.2 超链接路径	76
6.2 创建超链接	77
6.2.1 超链接——<a> 标签	77
6.2.2 超链接提示——alt 属性	77

6.2.3 超链接打开方式——target 属性	78	8.3.3 图像域——image 类型	100
实战 为文字和图像设置超链接	78	8.3.4 隐藏域——hidden 类型	104
6.3 创建锚点链接	79	8.3.5 复选框——checkbox 类型	104
6.3.1 插入锚点	79	8.3.6 单选按钮——radio 类型	107
6.3.2 创建锚点链接	80	8.3.7 文件域——file 类型	109
实战 制作锚点链接页面	81	8.3.8 普通按钮——button 类型	110
6.4 创建特殊超链接	83	8.3.9 提交按钮——submit 类型	110
6.4.1 空链接	83	8.3.10 重置按钮——reset 类型	110
6.4.2 文件下载链接	83	8.4 其他类型表单元素	110
实战 创建空链接和文件下载链接	83	8.4.1 文本区域——<textarea> 标签	111
6.4.3 脚本链接	84	8.4.2 列表/菜单——<select> 标签	111
实战 创建关闭窗口脚本链接	85	实战 制作网站搜索	112
6.4.4 E-mail 链接	85	8.5 制作用户注册页面	114
实战 创建电子邮件链接	86	实战 用户注册页面	114
第 7 章 插入多媒体			
7.1 插入 Flash 动画——<embed> 标签	87	第 9 章 表格与 Div	
实战 制作 Flash 欢迎页	87	9.1 了解表格	119
7.2 添加背景音乐	88	9.2 创建表格	119
7.2.1 网页中支持的音频格式	88	9.2.1 表格——<table> 标签	119
7.2.2 背景音乐——<bgsound> 标签	89	9.2.2 表格标题——<caption> 标签	120
7.2.3 音乐循环播放次数——loop 属性	89	9.2.3 表头——<th> 标签	121
实战 为网页设置背景音乐	90	实战 创建学习安排表	122
7.2.4 嵌入音频——<embed> 标签	90	9.3 表格结构标签	123
实战 在网页中嵌入音频播放控制条	91	9.3.1 表格头部——<thead> 标签	123
7.3 插入普通视频	92	9.3.2 表格主体——<tbody> 标签	124
7.3.1 网页中支持的视频格式	92	9.3.3 表格尾部——<tfoot> 标签	125
7.3.2 插入视频——<embed> 标签	92	实战 制作学生成绩数据表	126
7.3.3 设置自动播放——autostart 属性	92	9.4 设置表格属性	127
实战 在网页中插入视频	93	9.4.1 表格宽度与高度——width 和 height 属性	127
7.3.4 隐藏播放控件——hidden 属性	94	9.4.2 表格对齐方式——align 属性	128
第 8 章 插入和设置表单元素			
8.1 关于表单	95	9.4.3 表格边框粗细——border 属性	129
8.1.1 插入表单域——<form> 标签	95	9.4.4 表格边框颜色——bordercolor 属性	130
8.1.2 表单动作——action 属性	96	9.4.5 表格背景颜色——bgcolor 属性	131
8.1.3 表单名称——name 属性	96	9.4.6 表格背景图像——background 属性	132
8.1.4 表单传送方式——method 属性	96	实战 为表格设置背景颜色和背景图像	133
8.1.5 表单编码方式——enctype 属性	97	9.5 设置单元行和单元格属性	135
8.1.6 目标显示方式——target	98	9.5.1 设置单元行内容水平和垂直对齐	135
8.2 添加表单元素	98	9.5.2 单元格间距	136
8.3 输入类型表单元素	99	9.5.3 单元格边框	137
8.3.1 文本域——text 类型	99	9.5.4 合并单元格	138
8.3.2 密码域——password 类型	100		



实战 使用表格制作新闻列表	139	11.2 绘制基本图形	169
9.6 IFrame 框架	141	11.2.1 绘制直线	169
9.6.1 IFrame 框架源文件	142	实战 在网页中绘制直线	169
9.6.2 IFrame 框架的宽度和高度	142	11.2.2 绘制矩形	170
9.6.3 IFrame 框架的对齐方式	142	实战 在网页中绘制矩形	171
9.6.4 IFrame 框架是否显示滚动条	142	11.2.3 绘制圆形	172
9.6.5 IFrame 框架的边框	142	实战 在网页中绘制圆形	173
实战 设置 IFrame 框架属性	142	11.2.4 绘制三角形	174
9.7 插入 Div	143	实战 在网页中绘制三角形	174
9.7.1 <div> 标签	144	11.2.5 清除图形	175
9.7.2 设置 Div 属性	144	实战 清除使用 canvas 元素所绘制的部分	175
9.7.3 与 <div> 标签	144	图形	175
实战 制作欢迎页面	145	11.3 绘制文本	176
第 10 章 HTML 5 文档结构		11.3.1 使用文本	176
10.1 认识 HTML5 文档结构	147	实战 在网页中绘制文字	177
10.2 HTML5 元素分类	148	11.3.2 获取文字宽度	177
10.2.1 结构片段元素	149	实战 获取所绘制文字宽度	178
10.2.2 进度信息元素	149	11.4 绘制弧线和曲线	178
10.2.3 交互性元素	149	11.4.1 使用 arc() 方法	179
10.2.4 内嵌应用元素和辅助元素	149	实战 通过中心点和半径绘制弧线	179
10.2.5 文档和应用中使用的元素	150	11.4.2 使用 arcTo() 方法	180
10.2.6 注释元素	150	实战 通过辅助线绘制弧线	180
10.2.7 文本和文本标记元素	150	11.4.3 使用 quadraticCurveTo() 方法	181
10.2.8 其他元素	150	实战 绘制二次样条曲线	182
10.3 创建 HTML5 主体内容	151	11.4.4 使用 bezierCurveTo() 方法	183
10.3.1 文章——<article> 标签	151	实战 绘制贝塞尔曲线	183
10.3.2 章节——<section> 标签	152	11.5 图形的组合与裁切	184
10.3.3 导航——<nav> 标签	154	11.5.1 图形组合	184
10.3.4 辅助信息——<aside> 标签	156	实战 绘制组合图形效果	185
10.3.5 发布日期——<time> 标签与微		11.5.2 使用图像	186
格式	157	实战 使用 canvas 元素绘制图像	187
10.4 添加语义模块	158	11.5.3 使用图像模式	188
10.4.1 页眉——<header> 标签	158	实战 设置图像平铺效果	188
10.4.2 标题组——<hgroup> 标签	159	11.5.4 裁切路径	189
10.4.3 页脚——<footer> 标签	159	实战 在网页中实现圆形裁切图像效果	189
10.4.4 联系信息——<address> 标签	160	11.6 图形变换处理	191
10.5 使用文档结构元素制作页面	160	11.6.1 移动变换操作	191
实战 制作企业网站首页面	161	实战 移动所绘制图形位置	191
第 11 章 使用 HTML5 画布绘图		11.6.2 缩放变换操作	193
11.1 使用 HTML5 画布绘图	166	实战 使用缩放操作绘制椭圆形	193
11.1.1 了解 canvas 元素	166	11.6.3 旋转变换操作	194
11.1.2 在网页中插入 canvas 元素	166	实战 绘制矩形并进行旋转	195
11.1.3 如何使用 canvas 元素实现		11.6.4 矩阵变形操作	196
绘图	167	11.7 图形颜色与样式设置	196
		11.7.1 绘制线性渐变	196
		实战 在网页中绘制矩形并填充线性渐变	197

11.7.2 绘制径向渐变	198	13.1.3 浏览器对 HTML5 表单的支持情况	220
实战 在网页中绘制圆形并填充径向渐变	198	13.2 HTML5 新增表单输入类型	220
11.7.3 不同的线型	199	13.2.1 url 类型	220
11.7.4 创建对象阴影	201	13.2.2 email 类型	220
实战 为图形添加阴影效果	201	13.2.3 range 类型	220
第 12 章 HTML5 的音频和视频		13.2.4 number 类型	221
12.1 HTML5 多媒体基础	203	13.2.5 tel 类型	221
12.1.1 视频文件格式	203	13.2.6 search 类型	221
12.1.2 在线多媒体的发展	203	13.2.7 color 类型	221
12.1.3 HTML5 音频和视频的优势	204	13.2.8 date 类型	221
12.1.4 音频和视频编解码器	204	13.2.9 month、week、time、datetime、datetime-local 类型	222
12.1.5 HTML5 音频和视频的不足	205	13.2.10 使用 HTML5 表单元素	222
12.1.6 检查浏览器是否支持 HTML5 音频和视频	205	实战 制作留言表单页面	222
12.2 使用 HTML5 音频	206	13.3 HTML5 新增其他表单元素	226
12.2.1 <audio> 标签所支持的音频格式	206	13.3.1 datalist 元素和 list 属性	226
12.2.2 使用 <audio> 标签	206	13.3.2 keygen 元素	226
实战 在网页中嵌入音频播放	206	13.3.3 output 元素	227
12.3 使用 HTML5 视频	207	13.4 HTML5 新增表单属性	227
12.3.1 <video> 标签所支持的视频格式	207	13.4.1 form 属性	227
12.3.2 使用 <video> 标签	207	13.4.2 formaction 属性	227
实战 在网页中嵌入视频播放	208	13.4.3 formmethod、formenctype、formnovalidate、formtarget 属性	228
12.3.3 使用 <source> 标签	209	13.4.4 placeholder 属性	228
12.4 <audio> 与 <video> 标签的属性	209	实战 为表单元素设置默认提示内容	228
12.4.1 元素的标签属性	209	13.4.5 autofocus 属性	229
12.4.2 元素的接口属性	210	13.4.6 autocomplete 属性	229
实战 为网页文本进行分段处理	211	13.5 HTML5 表单验证标签属性	229
12.5 <audio> 与 <video> 标签的接口方法和事件	212	13.5.1 required 属性	230
12.5.1 <audio> 与 <video> 标签的接口方法	212	13.5.2 pattern 属性	230
实战 控制视频的播放和暂停	212	13.5.3 min、max 和 step 属性	230
12.5.2 <audio> 与 <video> 标签的接口事件	214	13.5.4 novalidate 属性	230
12.5.3 <audio> 与 <video> 标签接口事件的使用方法	214	实战 对网页表单进行验证	230
12.5.4 自定义视频播放控制组件	214	13.6 HTML5 表单验证 API	231
实战 自定义视频播放控制组件	215	13.6.1 表单验证的属性	231
第 13 章 使用 HTML5 的表单元素		13.6.2 ValidityState 对象	232
13.1 了解 HTML5 表单	219	13.6.3 checkValidity() 方法验证表单	233
13.1.1 HTML 表单的发展	219	实战 使用 checkValidity() 方法验证表单	233
13.1.2 HTML5 新增表单元素的作用	219	13.6.4 setCustomValidity() 方法验证表单	234
		实战 使用 setCustomValidity() 方法验证表单	234
		13.6.5 表单验证事件	236
		实战 使用 invalid 事件验证表单	236

**第 14 章 文件与拖放处理**

14.1 文件 API	238
14.1.1 新增的上传表单元素属性	238
14.1.2 File 对象与 FileList 对象	239
实战 同时上传多个文件并显示文件名称	239
14.1.3 Blob 对象	240
实战 获取所上传文件数据	240
14.2 FileReader 接口	242
14.2.1 检查浏览器是否支持 FileReader	
接口	242
14.2.2 FileReader 接口的属性	242
14.2.3 FileReader 接口的方法	243
实战 使用 FileReader 接口方法将上传的	
文件读取为不同数据	243
14.2.4 FileReader 接口的事件	245
实战 FileReader 接口的事件响应顺序	245
14.3 拖放 API	247
14.3.1 新增的 draggable 属性	247
14.3.2 新增的鼠标拖放事件	247
14.3.3 DataTransfer 对象	247
14.3.4 把图像拖放到网页中	248
实战 把图像拖放到网页中	248

第 15 章 HTML5 本地存储

15.1 Web Storage	251
15.1.1 什么是 Web Storage	251
15.1.2 Cookie 存储数据的不足	251
15.1.3 使用 Web Storage 存储的优势	252
15.1.4 会话存储与本地存储的区别	252
15.2 使用 Web Storage	252
15.2.1 检查浏览器是否支持	
Web Storage	252
实战 检查浏览器是否支持 Web Storage	253
15.2.2 设置和获取 Storage 数据	254
实战 使用 sessionStorage 和 localStorage	254
15.2.3 Storage API 的属性和方法	255
实战 使用 storage 对象保存页面内容	256
15.2.4 格式化数据	257
实战 使用 Storage 对象存储 JSON 数据	257
15.2.5 Storage API 事件	259
实战 使用 Web Storage 制作简单留言板	260
15.3 本地数据库 Web SQL	261
15.3.1 了解 Web SQL 数据库	262
15.3.2 Web SQL 数据库的基本操作	262
15.3.3 使用 Web SQL 数据库	263

实战 实现选择网页背景颜色 263

第 16 章 HTML5 离线应用缓存

16.1 Web 离线应用缓存	268
16.1.1 新增的离线应用缓存	268
16.1.2 离线应用缓存与传统页面缓存的	
区别	269
16.1.3 离线应用缓存与本地数据存储的	
区别	269
16.2 manifest 缓存清单文件	269
16.2.1 manifest 文件的结构	269
16.2.2 如何使用 manifest 文件	270
16.2.3 服务器端的配置	270
16.2.4 检查浏览器网络状态	271
实战 检查浏览器的网络状态	271
16.3 应用缓存接口 applicationCache	273
16.3.1 检查浏览器是否支持应用缓存	
接口	273
实战 检查浏览器是否支持 Web 离线	
应用	273
16.3.2 applicationCache 接口	274
16.3.3 接口的 status 属性	274
16.3.4 接口的方法	275
16.3.5 接口的事件	275
实战 离线应用缓存的事件发生顺序	275
16.4 离线缓存网页内容	277
实战 离线缓存网页内容	277

第 17 章 使用 Web Workers 处理线程

17.1 了解 Web Workers	281
17.1.1 什么是 Web Workers	281
17.1.2 Web Workers 线程的特点	281
17.1.3 Web Workers 体系结构	282
17.2 专属线程 Dedicated Worker	282
17.2.1 检查浏览器是否支持 Worker	
对象	282
17.2.2 创建专属线程	282
17.2.3 为线程添加监听消息事件	282
17.2.4 向线程中发送消息	283
17.2.5 编写线程处理的脚本文件	283
17.2.6 在线程中加载多个文件	283
17.2.7 监听线程错误	283
实战 简单的专属线程应用	284
17.2.8 多线程嵌套	285
17.2.9 单层线程嵌套	286
实战 线程嵌套输出最大质数	286

17.2.10 多层线程嵌套	287
17.3 共享线程 Shared Worker	288
17.3.1 共享线程的基本用法	288
17.3.2 使用共享线程	289
实战 使用共享线程显示连接数	289
17.4 Web Workers 接口框架	291
17.4.1 线程外部接口	291
17.4.2 线程内部接口	293

第 18 章 跨源通信和 WebSocket 双向通信

18.1 了解跨文档信息传输	296
18.1.1 跨文档信息传输的实现	296
18.1.2 网页源安全	297
18.2 使用 postMessage 接口	297
18.2.1 检查浏览器是否支持 postMessage 接口	298
18.2.2 使用 postMessage() 方法发送信息	298
18.2.3 使用 message 事件监听收到的信息	298
实战 实现简单的跨文档信息传输	299
18.3 信息事件接口 MessageEvent	301
18.3.1 MessageEvent 接口清单	301
18.3.2 MessageEvent 接口属性	301
18.3.3 initMessageEvent() 接口方法	302
18.3.4 MessageEvent() 接口说明	302
18.4 了解 XMLHttpRequestLevel2 规范	302
18.4.1 XMLHttpRequestLevel2 规范的优势	302
18.4.2 XMLHttpRequestLevel2 规范的接口	303
18.4.3 XMLHttpRequestLevel2 规范中新的事件	304
18.4.4 检查浏览器是否支持全新的 XMLHttpRequest 对象	304
18.4.5 构建跨源请求	304
18.4.6 添加监听事件	304
18.4.7 部署服务器	305
18.5 了解 WebSocket	305
18.5.1 WebSocket 概述	305
18.5.2 WebSocket 的优势	306
18.5.3 WebSocket 编程接口	306
18.6 了解 WebSocket 编程基础	308
18.6.1 检查浏览器是否支持 WebSocket	308

18.6.2 创建连接	308
18.6.3 添加状态和消息监听事件	309
18.6.4 发送信息	309
18.6.5 关闭连接	309

第 19 章 使用 HTML5 获取地理位置

19.1 Geolocation API	310
19.1.1 地理位置坐标信息	310
19.1.2 地理位置信息的来源	310
19.1.3 地理位置信息的应用	311
19.1.4 Geolocation API 中的隐私保护	311
19.2 使用 Geolocation 前的准备	312
19.2.1 Geolocation 的浏览器支持情况	312
19.2.2 检查浏览器是否支持 Geolocation API	312
实战 检查浏览器是否支持 Geolocation API	312
19.3 使用 Geolocation API 获取地理位置	313
19.3.1 getCurrentPosition() 方法	313
19.3.2 回调函数 successCallback()	314
19.3.3 回调函数 errorCallback()	314
19.3.4 可选参数 options	315
19.3.5 单次获取地理位置信息	315
实战 单次获取地理位置信息	316
19.3.6 重复更新地理位置信息	319
19.4 Geolocation 接口	319
19.4.1 NavigatorGeolocation 接口清单	319
19.4.2 Geolocation 接口清单	319
19.4.3 PositionOptions 接口清单	320
19.4.4 Position 接口清单	320
19.4.5 Coordinates 接口清单	321
19.4.6 PositionError 接口清单	321
19.5 在地图上显示位置	321
实战 在地图上显示位置	322

第 20 章 HTML5 网页综合实战

20.1 制作电子商务网站页面	327
20.1.1 设计分析	327
20.1.2 布局结构分析	327
20.1.3 制作 HTML5 页面	328
实战 制作页眉 header 和导航 nav 区域	328
实战 制作网页广告区域	330
实战 制作网页侧边栏区域	332
实战 制作网页主体内容区域	334
实战 制作网页版底 footer 区域	338



20.2	制作企业网站页面	340
20.2.1	设计分析	340
20.2.2	布局结构分析	340
20.2.3	制作 HTML5 页面	341
实战	制作页面头部 header 区域	341
实战	制作网页焦点轮换广告图	344
实战	制作页面主体内容区域	346
实战	制作页脚部分内容	348

第 21 章 HTML5 手机网页实战

21.1	如何制作响应式网站页面	351
------	-------------	-----

21.1.1	什么是响应式设计	351
21.1.2	响应式设计的相关术语	352
21.1.3	<meta> 标签设置	353
21.2	制作响应式摄影图片网页	354
21.2.1	设计分析	354
21.2.2	布局结构分析	354
21.2.3	制作 HTML5 响应式网页	355
实战	制作页面导航区域	355
实战	制作页面主体内容区域	360
实战	制作页面版底信息区域	365

第1章

从 HTML 到 HTML5



HTML 是 Internet 上制作网页的主要语言。网页中所包含的图像、动画、表单和多媒体等复杂的元素，其基础本质都是 HTML。随着互联网的飞速发展，网页设计语言也在不断地变化和发展，从 HTML 到 XHTML 再到 HTML5，每一次的发展变革都是为了适应互联网发展的需求。本章将向读者介绍有关 HTML、XHTML 和 HTML5 的基础知识，使读者对 HTML 的发展有所了解，并且理解 HTML5 与 HTML 之间有哪些共同点以及有哪些改进。

本章知识点：

- 了解 HTML 的基础知识
- 理解 HTML 的基本语法和编写注意事项
- 了解 XHTML 的相关基础知识
- 掌握 HTML5 的文档结构并理解 HTML5 的优势
- 掌握两种 HTML 文件的编写方式
- 认识 HTML5 中的标签
- 了解 HTML5 中的标准属性和事件属性

1.1

HTML 基础



HTML 主要运用标签使页面显示出预期的效果，也就是在文本文件的基础上，加上一系列的网页元素展示效果，最后形成扩展名为 .htm 或 .html 的文件。当用户通过浏览器阅读 HTML 文件时，浏览器负责解释插入 HTML 文本中的各种标签，并以此为依据显示文本内容，把 HTML 语言编写的文件称为 HTML 文本，HTML 语言即网页的描述语言。

1.1.1 HTML 概述



在介绍 HTML 语言之前，不得不介绍 World Wide Web(万维网)。万维网是一种建立在 Internet 上的全球性的、交互的、动态多平台分布式的图形信息系统。它采用 HTML 语法描述超文本(Hypertext)文件。Hypertext 一词有两个含义：一个是链接相关联的文件；另一个是内含多媒体对象的文件。

HTML 的英文全称是 Hyper Text Markup Language，中文通常称为超文本标记语言，HTML 是 Internet 中用于编写网页的主要语言，HTML 提供了精简而有力的文件定义，可以设计出多姿多彩的超媒体文件。通过 HTTP 通信协议，HTML 文件得以在万维网上进行跨平台的文件交换。

1.1.2 HTML 特性



HTML 文件制作简单，且功能强大，支持不同数据格式的文件导入，其主要有以下几个特点。

- HTML 文件容易创建，只需要一个文本编辑器就可以完成。
- HTML 文件存储容量小，能够尽可能快速地在网络中进行传输和显示。
- HTML 文件与操作平台无关，HTML 独立于操作系统平台，能够与多种平台兼容，只需要一个浏览器就可以在操作系统中浏览网页文件。
- 简单易学，不需要很深的专业编程知识。



HTML 具有扩展性，HTML 的广泛应用带来了加强功能、增加标识符等要素，HTML 采取了类元素的方式，为系统扩展提供了保证。

提示

HTML 文件可以直接由浏览器解释执行，而无须编译。当用浏览器打开网页时，浏览器读取网页中的 HTML 代码，分析其语法结构，然后根据解释的结果显示网页内容，正因如此，网页显示的速度同网页代码的质量有很大的关系，保持精简和高效的 HTML 源代码是十分重要的。

1.1.3 HTML 文档结构

HTML 的所有标签都是由“<”和“>”括起来，如 <html>。在起始标签的标签名前加上符号“/”便是其终止标签，如 </html>。HTML 文档内容要包含在 <html> 与 </html> 标签之间，完整的 HTML 网页文档应该包括头部和主体两大部分。

HTML 文件基本结构如下。

<html>	<!--HTML 文件开始 -->
<head>	<!--HTML 文件的头部开始 -->
网页头部内容	
</head>	<!--HTML 文件的头部结束 -->
<body>	<!--HTML 文件的主体开始 -->
网页主体内容部分	
</body>	<!--HTML 文件的主体结束 -->
</html>	<!--HTML 文件结束 -->

1. <html>...</html>

告诉浏览器 HTML 文件开始和结束，<html> 标签出现在 HTML 文档的第一行，用来表示 HTML 文档的开始。</html> 标签出现在 HTML 文档的最后一行，用来表示 HTML 文档的结束。两个标签一定要一起使用，网页中的所有内容都要放在 <html> 与 </html> 之间。

2 <head>...</head>

网页的头标签，用来定义 HTML 文档的头部信息，该标签也是成对使用的。

3. <body>...</body>

在 <head> 标签之后就是 <body> 与 </body> 标签，该标签也是成对出现的。<body> 与 </body> 标签之间为网页主体内容和其他用于控制内容显示的标签。

1.1.4 HTML 的基本语法

绝大多数元素都有起始标签和结束标签，在起始标签和结束标签之间的部分是元素体。例如，<body>...</body>。第一个元素都有名称和可选择的属性，元素的名称和属性都在起始标签内标明。HTML 中的标签主要分为普通标签和空标签两种类型。

1. 普通标签

普通标签是由一个起始标签和一个结束标签所组成的，其语法格式如下。

<x> 控制文字 </x>

其中，x 代表标签名称。<x> 和 </x> 就如同一组开关：起始标签 <x> 为开启某种功能，而结束标签 </x>（通常为起始标签加上一个斜杠 /）为关闭功能，受控制的内容便放在两个标签之间。例如下面的代码。

 加粗文字

标签之中还可以附加一些属性，用来实现或完成某些特殊效果或功能。例如下面的代码。

```
<x a1="v1" a2="v2"...an="vn"> 控制文字 </x>
```

其中，a1, a2, ..., an 为属性名称，而 v1, v2, ..., vn 则是其所对应的属性值。属性值加不加引号，目前所使用的浏览器都可接受，但根据 W3C 的新标准，属性值是要加引号的，所以最好养成加引号的习惯。

2. 空标签

虽然大部分的标签是成对出现的，但也有一些是单独存在的，这些单独存在的标签称为空标签，其语法格式如下。

```
<x>
```

同样，空标签也可以附加一些属性，用来完成某些特殊效果或功能。例如下面的代码。

```
<x a1="v1" a2="v2"...an="vn">
```

1.1.5 HTML 编写注意事项

HTML 由标签和属性构成，在编写 HTML 文档时，需要注意以下几点。

- ❶ “<” 和 “>” 是任何标签的开始和结束。元素的标签需要使用这对尖括号括起来，并且在结束标签的前面加上符号 “/”，如 <p> 和 </p>。

- ❷ 在 HTML 代码中不区分大小写。

- ❸ 任何空格和回车在 HTML 代码中均不起作用。为了 HTML 代码的清晰，建议不同的标签之间按回车键进行换行。

- ❹ 在 HTML 标签中可以添加各种属性设置。例如下面的 HTML 代码。

```
<p align="center">这里是段落文本 </p>
```

- ❺ 需要正确地输入 HTML 标签，输入 HTML 标签时，不要输入多余的空格，否则浏览器可能无法识别这个标签，导致无法正确显示。

- ❻ 在 HTML 代码中合理地使用注释。<!-- 需要注释的内容 --> 注释语句只会出现在 HTML 代码中，不会在浏览器中显示。

1.2 XHTML 基础

XHTML 是当前 HTML 版本的发展和延伸。HTML 语法要求比较松散，这样对网页编写者来说比较方便，但对于机器来说，语言的语法越松散，处理起来就越困难。对于传统的计算机来说，还有能力兼容松散语法，但对于许多移动设备，比如手机，难度就比较大。因此产生了由 DTD 定义规则，语法要求更加严格的 XHTML。

1.2.1 XHTML 概述

XHTML 是 HTML 的一种扩展，即 Extensible Hyper Text Markup Language 的缩写，表示 XHTML 是可扩展的超文本标记语言，与 HTML 相比，具有更加规范的书写标准、更好的跨平台能力。

HTML 是一种基本的网页设计语言，XHTML 是一种基于 XML 的标识语言，看起来与 HTML 非常相似，只有一些细节的重要区别，XHTML 就是一个扮演着类似 HTML 角色的 XML，所以本质上说，XHTML 是一个过渡技术，融合了部分 XML 的强大功能及大多数 HTML 的简单特性。

XHTML1.0 是在 HTML4 的基础上进行优化和改进的新语言，它与 HTML 最主要的不同之处在于：XHTML 元素一定要正确地嵌套，XHTML 元素必须要关闭，标签名称必须使用小写字母，XHTML 文档必须拥有根元素。



1.2.2 XHTML 文档结构



首先看一个最简单的 XHTML 页面实例，其代码如下。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title> 无标题文档 </title>
</head>
<body>
  文档内容部分
</body>
</html>
```

在这段代码中，包含一个 XHTML 页面必须具有的页面结构，其具体结构包含以下几个部分。

1. 文档类型声明部分

文档类型声明部分由 <!DOCTYPE> 元素定义，其对应的页面代码如下。

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

2. <html> 元素和名字空间

<html> 元素是 XHTML 文档中必须使用的元素，所有的文档内容（包括文档头部内容和文档主体内容）都要包含在 <html> 元素之中，<html> 元素的语法结构如下。

```
<html> 文档内容部分 </html>
```

起始标签 <html> 和结束标签 </html> 一起构成一个完整的 <html> 元素，其包含的内容要写在起始标签和结束标签之间。

名字空间是 <html> 元素的一个属性，写在 <html> 元素的起始标签里面，其在页面中的相应代码如下。

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

名字空间属性用 xmlns 来表示，用来定义识别页面标签的网址。

3. 文档头部内容

网页头部元素 <head> 也是 XHTML 文档中必须使用的元素，其作用是定义页面头部的信息，其中可以包含标题元素、<meta> 元素等，<head> 元素的语法结构如下。

```
<head> 头部内容部分 </head>
```

<head> 元素所包含的内容不会显示在浏览器的窗口中，但是部分内容会显示在浏览器的特定位置，如标题栏等。

4. 标题元素

页面标题元素 <title> 用来定义页面的标题，其语法结构如下。

```
<title> 页面标题 </title>
```

在预览和发布页面时，页面标题中包含的文本会显示在浏览器的标题栏中。

5. 文档主体元素

主体元素 <body> 用来定义页面所要显示的内容，页面的信息主要通过页面主体来传递，在 <body> 元素中，可以包含所有页面元素，<body> 元素的语法结构如下。

```
<body> 页面主体 </body>
```

定义了以上几个元素后，便构成一个完整的 XHTML 页面。而且以上所有元素都是 XHTML 页面

所必须具有的基本元素。

1.2.3 XHTML 的文档类型

文档类型又可以写为 DOCTYPE，是 Document Type 的简写，在页面中用来说明页面所使用的 XHTML 是什么版本。制作 XHTML 页面，一个必不可少的关键组成部分就是 DOCTYPE 声明，只有确定了一个正确的 DOCTYPE，XHTML 里的标识和级联样式才能正常生效。

在 XHTML1.0 中有 3 种 DTD(文档类型定义)声明可以选择：过渡的 (transitional)、严格的 (strict)、框架的 (frameset)，分别介绍如下。

1. 过渡的 DTD

这是一种要求不很严格的 DTD，允许用户使用一部分旧的 HTML 标签来编写 XHTML 文档，帮助用户慢慢适应 XHTML 的编写，过渡的 DTD 的写法如下。

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

2. 严格的 DTD

这是一种要求严格的 DTD，不允许使用任何表现层的标识和属性，例如
 等，严格的 DTD 的写法如下。

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

3. 框架的 DTD

这是一种专门针对框架页面所使用的 DTD，当页面中包含框架元素时，就要采用这种 DTD，框架的 DTD 的写法如下。

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

使用严格的 DTD 来制作页面当然是最理想的方式，但对于没有深入了解 Web 标准的网页设计者，比较适合使用过渡的 DTD。因为这种 DTD 还允许使用表现层的标识、元素和属性。DOCTYPE 的声明一定要放置在 XHTML 文档的头部。

1.2.4 名字空间

名字空间的英文是 Namespace，含义就是通过一个网址指向来识别页面上的标签。在 XHTML 中使用的是 xmlns，也就是 XHTML Namespace 的缩写。用来识别 XHTML 页面上的标签的网址指向是 <http://www.w3.org/1999/xhtml>。关于名字空间定义的完整写法如下。

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

当使用可视化的网页开发工具新建文档时，选择适当格式的文档类型，DOCTYPE 的声明和名字空间的声明都会自动生成。到目前为止，XHTML 的 4 种文档类型的名字空间都是 <http://www.w3.org/1999/xhtml>。

1.3

HTML5 基础

HTML5 是近十年来 Web 标准巨大的飞跃。和以前的版本不同，HTML5 并非仅仅用来表示 Web 内容，它的使命是将 Web 带入一个成熟的应用平台，在这个平台上，视频、音频、图像、动画，以及与计算机的交互都被标准化。尽管 HTML5 的实现还有很长的路要走，但 HTML5 正在改变 Web。



1.3.1 HTML5 概述

W3C 在 2010 年 1 月 22 日发布了最新的 HTML5 工作草案。HTML5 的工作组包括 AOL、Apple、Google、IBM、Microsoft、Mozilla、Nokia、Opera 以及数百个其他的开发商。制定 HTML5 的目的是取代 1999 年 W3C 所制定的 HTML4.01 和 XHTML1.0 标准, 希望在网络应用迅速发展的同时, 网页语言能够符合网络发展的需求。

HTML5 实际上是指包括 HTML、CSS 样式和 JavaScript 脚本在内的一整套技术的组合, 希望通过 HTML5 能够轻松地实现许多丰富的网络应用需求, 而减少浏览器对插件的依赖, 并且提供更多能有效增强网络应用的标准集。

在 HTML5 中添加了许多新的应用标签, 其中包括 <video>、<audio> 和 <canvas> 等标签, 添加这些标签是为了设计者能够更轻松地网页中添加或处理图像和多媒体内容。其他新的标签还有 <section>、<article>、<header> 和 <nav>, 这些新添加的标签是为了能够更加丰富网页中的数据内容。除了添加许多功能强大的新标签和属性外, 同样还对一些标签进行了修改, 以方便适应快速发展的网络应用。同时, 也有一些标签和属性在 HTML5 标准中已经被去除。

1.3.2 HTML5 的文档结构

HTML5 的文档结构与前面所介绍的 HTML 和 XHTML 的文档结构非常类似, 基础的文档结构如下。

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title> 无标题文档 </title>
  </head>
  <body>
    页面主体内容部分
  </body>
</html>
```

与 XHTML 相比, HTML5 的文档结构非常简洁, 第一行代码 <!doctype html> 声明文档是一个 HTML 文档, 然后使用 <html> 标签包含头部内容 <head> 标签和主体内容 <body> 标签, 从而构成 HTML5 文档的基本结构。

1.3.3 HTML5 的优势

对于用户和网站开发者而言, HTML5 的出现意义非常重大。因为 HTML5 解决了 Web 页面存在的诸多问题, HTML5 的优势主要表现在以下几个方面。

1. 化繁为简

HTML5 为了做到尽可能简化, 避免了一些不必要的复杂设计。例如, DOCTYPE 声明的简化处理, 在过去的 HTML 版本中, 第一行的 DOCTYPE 过于冗长, 在实际的 Web 开发中也没有什么意义, 而在 HTML5 中 DOCTYPE 声明就非常简洁。

为了让一切变得简单, 同时避免造成误解, HTML5 对每一个细节都有着非常明确的规范说明, 不允许有任何的歧义和模糊出现。

2. 向下兼容

HTML5 有着很强的兼容能力。在这方面, HTML5 没有颠覆性的革新, 允许存在不严谨的写法。例如, 一些标签的属性值没有使用英文引号括起来; 标签属性中包含大写字母; 有的标签没有闭合等。

然而这些不严谨的错误处理方案，在 HTML5 的规范中都有明确的规定，也希望未来在浏览器中有一致的支持。当然对于 Web 开发者来说，还是遵循严谨的代码编写规范比较好。

对于 HTML5 的一些新特性，如果旧的浏览器不支持，也不会影响页面的显示。在 HTML 规范中，也考虑了这方面的内容。例如，在 HTML5 中 `<input>` 标签的 `type` 属性增加了很多新的类型，当浏览器不支持这些类型时，默认会将其视为 `text`。

3. 支持合理

HTML5 的设计者花费了大量的精力来研究通用的行为。例如，Google 分析了上百万份的网页，从中提取了 `<div>` 标签的 `id` 名称，很多网页开发人员都按如下标记导航区域。

```
<div id="nav">
  // 导航区域内容
</div>
```

既然该行为已经大量存在，HTML5 就会想办法去改进，所以就直接增加了一个 `<nav>` 标签，用于网页导航区域。

4. 实用性

对于 HTML 无法实现的一些功能，用户会寻求其他方法来实现。例如，对于绘图、多媒体、地理位置和实时获取信息等的应用，通常会开发一些相应的插件间接地去实现。HTML5 的设计者们研究了这些需求，开发了一系列用于 Web 应用的接口。

HTML5 规范的制定是非常开放的，所有人都可以获取草案的内容，也可以参与进来提出宝贵的意见。因为开放，所以可以得到更加全面的发展。一切以用户需求为最终目的。所以，当用户在使用 HTML5 的新功能时，会发现正是期待已久的功能。

5. 用户优先

在遇到无法解决的冲突时，HTML5 规范把最终用户的诉求放在第一位。因此，HTML5 的绝大部分功能都是非常实用的。用户与开发者的重要性远远高于规范和理论。例如，有很多用户都需要实现一个新的功能，HTML5 规范的设计者会研究这种需求，并纳入规范。HTML5 规范了一套错误处理机制，以便当 Web 开发者写了不够严谨的代码时，接纳这种不严谨的写法。所以 HTML5 比以前版本的 HTML 更加友好。

1.3.4 HTML5 精简的头部

HTML5 避免了不必要的复杂性，DOCTYPE 和字符集都极大地简化了。

DOCTYPE 声明是 HTML 文件中必不可少的内容，它位于 HTML 文档的第一行，声明了 HTML 文件遵循的规范。HTML4.01 的 DOCTYPE 声明代码如下。

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

这么长的一串代码恐怕极少有人能够默写出来，通常都是通过复制、粘贴的方法添加这段代码。而在 HTML5 中的 DOCTYPE 代码则非常简单，如下所示。

```
<!DOCTYPE html>
```

这样就简洁了许多，不需要再复制、粘贴代码。同时，这种声明也标志性地让人感觉到这是符合 HTML5 规范的页面。如果使用了 HTML5 的 DOCTYPE 声明，则会触发浏览器以标准兼容的模式来显示页面。

字符集的声明也是非常重要的，它决定了页面文件的编码方式。在过去都是使用如下的方式来指定字符集的。

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



HTML5 对字符集的声明也进行了简化处理, 简化后的声明代码如下。

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

在 HTML5 中, 以上两种字符集的声明方式都可以使用, 这是由 HTML5 向下兼容的原则决定的。

1.4

HTML 文件的编写方式



网页文件即扩展名为 .htm 或 .html 的文件, 本质上是文本类型的文件, 网页中的图片、动画等资源是通过网页文件的 HTML 代码链接的, 与网页文件分开存储。

由于 HTML 语言编写的文件是标准的 ASCII 文本文件, 因此可以使用任意一种文本编辑器来打开或编辑 HTML 文件。例如, Windows 操作系统中自带的记事本或者专业的网页制作软件 Dreamweaver。

1.4.1 使用记事本编写



HTML 是一个以文字为基础的语言, 并不需要什么特殊的开发环境, 可以直接在 Windows 操作系统自带的记事本中进行编辑, 其优点是方便快捷; 缺点是无任何语法提示、无行号提示和格式混乱等, 初学者使用困难。

实战 使用记事本制作 HTML 页面



最终文件: 最终文件\第 1 章\1-4-1.html

视频: 视频\第 1 章\1-4-1.mp4



01 在 Windows 操作系统中执行“开始”>“所有程序”>“附件”>“记事本”命令, 打开记事本窗口, 如图 1-1 所示。在记事本中按正确的文档结构编写 HTML 页面代码, 如图 1-2 所示。



图 1-1



图 1-2

02 所编写的完整页面 HTML 代码如下。

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>在记事本中编写 HTML 页面</title>
</head>
<body>

</body>
</html>
```

提示

此处所编写的 HTML 页面代码非常简单, 主要在头部的 <title> 与 </title> 标签之间输入网页的标题, 在页面主要内容 <body> 与 </body> 标签之间使用 标签插入一张图片, 并且添加了图片宽度和高度属性的设置。

03 执行“文件”>“另存为”命令, 弹出“另存为”对话框, 将文件保存为“源文件\第 1 章\1-4-1.html”, 如图 1-3 所示。单击“保存”按钮, 即可将记事本编写的 HTML 代码保存为网页文件, 在浏览器中预览该网页文件, 可以看到网页的效果, 如图 1-4 所示。



图 1-3



图 1-4

技巧

在将记事本中编写的 HTML 代码保存为网页 .html 格式时, 需要注意“编码”选项的设置。默认情况下, 需要将“编码”设置为 UTF-8, 否则网页中的中文在浏览器中可能会显示为乱码。

1.4.2 使用 Dreamweaver 编写

Dreamweaver 是网页制作的主流软件, 其优点是有所见即所得的设计视图, 能够通过鼠标拖放直接创建并编辑网页文件, 自动生成相应的 HTML 代码。Dreamweaver 的代码视图有非常完善的语法自动提示、自动完成和关键词高亮等功能。可以说, Dreamweaver 是一个非常全面的网页制作工具, 如图 1-5 所示为最新版本的 Dreamweaver CC 2018 软件的工作界面。



图 1-5

🔍 菜单栏: 菜单栏中包含所有 Dreamweaver 操作所需要的命令。这些命令按照操作类别分为“文件”“编辑”“查看”“插入”“工具”“查找”“站点”“窗口”和“帮助”9 个菜单。

🔍 视图选项: 在 Dreamweaver 中提供了 3 种视图窗口形式, 分别是“代码视图”“拆分视图”和“实时视图”, 其中“实时视图”中还包含旧版本的“设计视图”选项。默认情况下使用“拆分视图”形式, 这样可以在编写 HTML 代码的同时实时地查看页面的效果。

🔍 实时视图: 在该窗口中显示当前所制作页面的效果, 也是可视化操作的窗口, 可以使用各种工具, 在该窗口中输入文字、插入图像等, 是所见即所得的视图。

🔍 代码视图: 在该窗口中将显示当前所编辑页面的 HTML 代码, 可以直接在该窗口中进行 HTML 页面代码的编写操作。在代码视图中, Dreamweaver 会通过不同的颜色对不同的 HTML 代码进行区别, 并且在编写过程中会实时地为用户提供相应的提示, 非常方便。

🔍 工具栏: 在 Dreamweaver 工作界面的左侧为用户提供了文档编辑操作的相关工具, 可以对文档 HTML 代码的快速格式化处理、添加注释等操作, 用户也可以自定义工具栏中所显示的工具选项。



❶ 标签选择器：显示环绕当前选定内容标签的层次结构。单击该层次结构中的任何标签可以选择该标签及其全部内容。

❷ 面板组：用于帮助用户监控和修改工作，例如，“插入”面板、“CSS 设计器”面板。单击相应的面板名称，可以折叠或展开相应的工具面板。

实战 使用 Dreamweaver 制作 HTML 页面

最终文件：最终文件\第 1 章\1-4-2.html

视频：视频\第 1 章\1-4-2.mp4



❶ 打开 Dreamweaver CC，执行“文件”>“新建”命令，弹出“新建文档”对话框，选择 HTML 选项，如图 1-6 所示。单击“创建”按钮，新建 HTML5 文档，在代码视图中可以看到文档的 HTML 代码，如图 1-7 所示。



图 1-6

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>无标题文档</title>
6 </head>
7
8 <body>
9 </body>
10 </html>
11
```

图 1-7

❷ 执行“文件”>“保存”命令，弹出“另存为”对话框，将该网页保存为“源文件\第 1 章\1-4-2.html”，如图 1-8 所示。在页面的 <title> 与 </title> 标签之间输入网页的标题，如图 1-9 所示。



图 1-8

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>使用Dreamweaver制作HTML页面</title>
6 </head>
7
8 <body>
9 </body>
10 </html>
11
```

图 1-9

❸ 在 <body> 标签中添加 style 属性设置代码，如图 1-10 所示。在 <body> 与 </body> 标签之间编写相应的网页正文内容代码，如图 1-11 所示。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>使用Dreamweaver制作HTML页面</title>
6 </head>
7
8 <body style="background-color:#7847B2; text-align:center; color:#FFF;">
9 </body>
10 </html>
11
```

图 1-10

提示

在 <body> 标签中添加 style 属性设置，实际上是 CSS 样式的一种使用方式，称为内联 CSS 样式。此处通过内联 CSS 样式设置页面整体的背景颜色、水平对齐方式和文字颜色。

❹ 完成该网页 HTML 代码的编写，执行“文件”>“保存”命令，保存网页，

```
8 <body style="background-color:#7847B2; text-align:center; color:#FFF;">
9 <br>
10 <br>
11 <br>
12 <br>
13 <br>
14 <br>
15 
16 <br>
17 <b>进入网站</b></body>
18 </html>
19
```

图 1-11

在浏览器中预览该网页，可以看到网页的效果，如图 1-12 所示。



图 1-12

提示

Dreamweaver 是一款专业的网页制作软件，在 Dreamweaver 中新建 HTML 页面，会自动给出 HTML 文档结构的基础代码，编写 HTML 代码还具有代码提示等功能，非常适合初学者使用。

在 Dreamweaver CC 中新建的 HTML 文档默认为符合 HTML5 规范的文档，本书中所有的代码编写都将在 Dreamweaver 的代码视图中进行。

1.5

HTML5 中的标签

通过制定如何处理所有 HTML 元素及如何从错误中恢复的精确规则，HTML5 改进了互操作性，并减少了开发成本。HTML5 中的标签介绍如表 1-1 所示。

表 1-1 HTML5 中的标签

标签	描述	HTML4	HTML5
<!--...-->	定义注释	√	√
<!DOCTYPE>	定义文件类型	√	√
<a>	定义超链接	√	√
<abbr>	定义缩写	√	√
<acronym>	HTML5 中已不支持，定义首字母缩写	√	×
<address>	定义地址元素	√	√
<applet>	HTML5 中已不支持，定义 applet	√	×
<area>	定义图像映射中的区域	√	√
<article>	HTML5 新增，定义 article	×	√
<aside>	HTML5 新增，定义页面内容之外的内容	×	√
<audio>	HTML5 新增，定义声音内容	×	√
	定义粗体文本	√	√
<base>	定义页面中所有链接的基准 URL	√	√
<basefont>	HTML5 中已不支持，可使用 CSS 代替	√	×
<bdo>	定义文本显示的方向	√	√
<big>	HTML5 中已不支持，定义大号文本	√	×
<blockquote>	定义摘自另一个源的块引用	√	√
<body>	定义 body 元素	√	√
 	插入换行符	√	√
<button>	定义按钮	√	√
<canvas>	HTML5 新增，定义图形	×	√
<caption>	定义表格标题	√	√
<center>	HTML5 中已不支持，定义居中的文本	√	×
<cite>	定义引用	√	√
<code>	定义计算机代码文本	√	√
<col>	定义表格列的属性	√	√
<colgroup>	定义表格列的分组	√	√
<command>	HTML5 新增，定义命令按钮	×	√
<datagrid>	HTML5 新增，定义树列表中的数据	×	√



(续表)

标签	描述	HTML4	HTML5
<datalist>	HTML5 新增, 定义下拉列表	×	√
<dataelement>	HTML5 新增, 定义数据模板	×	√
<dd>	定义自定义的描述	√	√
	定义删除文本	√	√
<details>	HTML5 新增, 定义元素的细节	×	√
<dialog>	HTML5 新增, 定义对话	×	√
<dir>	HTML5 中已不支持, 定义目录列表	√	×
<div>	定义文档中的一个部分	√	√
<dfn>	定义一个自定义项目	√	√
<dl>	定义自定义列表	√	√
<dt>	定义自定义列表中的项目	√	√
	定义强调文本	√	√
<embed>	HTML5 新增, 定义外部交互内容或插件	×	√
<event-source>	HTML5 新增, 为服务器发送的事件定义目标	×	√
<fieldset>	定义 fieldset	√	√
<figure>	HTML5 新增, 定义媒介内容的分组, 以及它们的标题	×	√
	HTML5 中已不支持, 定义文本的字体、尺寸和颜色	√	×
<footer>	HTML5 新增, 定义 section 或 page 的页脚	×	√
<form>	定义表单	√	√
<frame>	HTML5 中已不支持, 定义子窗口 (框架)	√	×
<frameset>	HTML5 中已不支持, 定义框架的集	√	×
<h1> to <h6>	定义标题 1 到标题 6	√	√
<head>	定义关于文档的信息	√	√
<header>	HTML5 新增, 定义 section 或 page 的页眉	×	√
<hr>	定义水平线	√	√
<html>	定义 HTML 文件	√	√
<i>	定义斜体文本	√	√
<iframe>	定义行内的子窗口 (框架)	√	√
	定义图像	√	√
<input>	定义输入域	√	√
<ins>	定义插入文本	√	√
<isindex>	HTML5 中已不支持, 定义单行的输入域	√	×
<kbd>	定义键盘文本	√	√
<label>	定义表单控件的标注	√	√
<legend>	用于为 fieldset 元素定义标题	√	√
	定义列表的项目	√	√
<link>	定义资源引用	√	√
<mark>	HTML5 新增, 定义有记号的文本	×	√
<map>	定义图像映射	√	√
<menu>	定义菜单列表	√	√
<meta>	定义元信息	√	√
<meter>	HTML5 新增, 定义预定义范围内的度量	×	√
<nav>	HTML5 新增, 定义导航链接	×	√
<nest>	HTML5 新增, 定义数据模板中的嵌套点	×	√
<noframes>	HTML5 中已不支持, 可为那些不支持框架的浏览器显示文本	√	×
<noscript>	HTML5 中已不支持, 定义 noscript 部分	√	×
<object>	定义嵌入对象	√	√

(续表)

标签	描述	HTML4	HTML5
	定义有序列表	√	√
<optgroup>	定义选项组	√	√
<option>	定义下拉列表中的选项	√	√
<output>	HTML5 新增, 定义输出的一些类型	×	√
<p>	定义段落	√	√
<param>	为对象定义参数	√	√
<pre>	定义预格式化文本	√	√
<progress>	HTML5 新增, 定义任何类型的任务进度	×	√
<q>	定义短的引用	√	√
<rule>	HTML5 新增, 为升级模板定义规则	×	√
<s>	HTML5 中已不支持, 定义加删除线的文本	√	×
<samp>	定义样本计算机代码	√	√
<script>	定义脚本	√	√
<section>	HTML5 新增, 定义 section	×	√
<select>	定义可选列表	√	√
<small>	HTML5 中已不支持, 定义小号文本	√	×
<source>	HTML5 新增, 定义媒介源	×	√
	定义文档中的 section	√	√
<strike>	HTML5 中已不支持, 定义加删除线的文本	√	×
	定义强调文本	√	√
<style>	定义样式定义	√	√
<sub>	定义上标文本	√	√
<sup>	定义下标文本	√	√
<table>	定义表格	√	√
<tbody>	定义表格的主体	√	√
<td>	定义表格单元	√	√
<textarea>	定义文本区域	√	√
<tfoot>	定义表格的脚注	√	√
<th>	定义表格内的表头单元格	√	√
<thead>	定义表格表头	√	√
<time>	HTML5 新增, 定义日期/时间	×	√
<title>	定义文档的标题	√	√
<tr>	定义表格行	√	√
<tt>	HTML5 中已不支持, 定义打字机文本	√	×
<u>	HTML5 中已不支持, 定义下划线文本	√	×
	定义无序列表	√	√
<var>	定义变量	√	√
<video>	HTML5 新增, 定义视频	×	√
<xmp>	HTML5 中已不支持, 定义预格式文本	√	×

1.6

HTML5 的标准属性

在 HTML 中标签拥有属性, 在 HTML5 中新增的属性有 contenteditable、contextmenu、draggable、irrelevant、ref、registrationmark、template。

HTML5 的标准属性如表 1-2 所示。



表 1-2 HTML5 标准属性

属性	值	描述	HTML4	HTML5
accesskey	character	设置访问一个元素的键盘快捷键	×	√
class	class_rule or style_rule	元素的类名	√	√
contenteditable	true false	设置是否允许用户编辑元素	×	√
contextmenu	id of a menu element	给元素设置一个上下文菜单	×	√
dir	ltr rtl	设置文本方向	√	√
draggable	true false auto	设置是否允许用户拖动元素	×	√
id	id_name	元素的唯一 id	√	√
irrelevant	true false	设置元素是否相关，不显示非相关的元素	×	√
lang	language_code	设置语言代码	√	√
ref	url of elementID	引用另一个文档或文档上另一个位置，仅在 template 属性设置时使用	×	√
registrationmark	registration mark	为元素设置拍照，可以规定于任何 <rule> 元素的后代元素，除了 <nest> 元素	×	√
style	style_definition	行内的样式定义	√	√
tabindex	number	设置元素的 Tab 键控制顺序	√	√
template	url or elementID	引用应该应用到该元素的另一个文档或本文档上另一个位置	×	√
title	tooltip_text	显示在工具提示中的文本	√	√

1.7

HTML5 的事件属性



HTML 元素可以拥有事件属性，这些属性在浏览器中触发行为，比如，当用户单击一个 HTML 元素时启动一段 JavaScript 脚本。下面列出了事件属性，可以将它们插入 HTML 中定义事件行为。HTML5 中的新事件有 onabort、onbeforeunload、oncontextmenu、ondrag、ondragend、ondragenter、ondragleave、ondragover、ondragstart、ondrop、onerror、onmessage、onmousewheel、onresize、onscroll、onunload。不再支持的 HTML 4.0.1 属性有 onreset。

HTML5 所支持的事件属性如表 1-3 所示。

表 1-3 HTML5 的事件属性

属性	值	描述	HTML4	HTML5
onabort	script	发生 abort 事件时运行脚本	×	√
onbeforeunload	script	在元素加载前运行脚本	×	√
onblur	script	当元素失去焦点时运行脚本	√	√
onchange	script	当元素改变时运行脚本	√	√
onclick	script	在鼠标单击时运行脚本	√	√
oncontextmenu	script	当菜单被触发时运行脚本	×	√
ondblclick	script	当鼠标双击时运行脚本	√	√
ondrag	script	只要脚本在被拖动就运行脚本	×	√
ondragend	script	在拖动操作结束时运行脚本	×	√
ondragenter	script	当元素被拖动到一个合法的放置目标时运行脚本	×	√

(续表)

属性	值	描述	HTML4	HTML5
ondragleave	script	当元素离开合法的放置目标时运行脚本	×	√
ondragover	script	只要元素正在合法的放置目标上拖动时就运行脚本	×	√
ondragstart	script	当拖动操作开始时运行脚本	×	√
ondrop	script	当元素正在被拖动时运行脚本	×	√
onerror	script	当元素加载的过程中出现错误时运行脚本	×	√
onfocus	script	当元素获得焦点时运行脚本	√	√
onkeydown	script	当按下按钮时运行脚本	√	√
onkeypress	script	当按下按键时运行脚本	√	√
onkeyup	script	当松开按钮时运行脚本	√	√
onload	script	当文档加载时运行脚本	√	√
onmessage	script	当 message 事件触发时运行脚本	×	√
onmousedown	script	当按下鼠标按键时运行脚本	√	√
onmousemove	script	当鼠标指针移动时运行脚本	√	√
onmouseover	script	当鼠标指针移动到一个元素上时运行脚本	√	√
onmouseout	script	当鼠标指针移出元素时运行脚本	√	√
onmouseup	script	当松开鼠标按键时运行脚本	√	√
onmousewheel	script	当滚动鼠标滚轮时运行脚本	×	√
onreset	script	不支持, 当表单重置时运行脚本	√	×
onresize	script	当元素调整大小时运行脚本	×	√
onscroll	script	当元素滚动条被滚动时运行脚本	×	√
onselect	script	当元素被选中时运行脚本	√	√
onsubmit	script	当提交表单时运行脚本	√	√
onunload	script	当文档卸载时运行脚本	×	√

第2章

HTML5 页面基本设置



本章从 HTML 控制网页整体属性入手，全面开始对 HTML 网页技术的学习，主要介绍 HTML 文档的基本标签，这些都是一个完整的网页必不可少的，通过这些标签可以了解网页的基本结构和工作原理。通过本章的学习，读者将掌握 HTML 网页文件的头部信息设置，及网页主体的基本设置，对 HTML 有基本的了解和掌握，并能够将两者灵活地结合运用。

本章知识点：

- 理解头部 `<head>` 标签
- 掌握标题 `<title>` 标签设置
- 掌握基底网址 `<base>` 标签的作用及设置方法
- 掌握使用 `<meta>` 标签设置网页头信息的方法
- 掌握在 `<body>` 标签中添加属性对网页整体进行设置的方法
- 了解 HTML 代码中添加注释的方法

2.1

网页头部——`<head>` 标签



通过前面章节对 HTML 网页的基础知识的学习，可知道 HTML 网页分为 `<head>`、`</head>` 部分和 `<body>`、`</body>` 部分。`head` 中文的意思即头部，因此一般把 `<head>`、`</head>` 部分称为网页的头部信息。头部信息部分的内容虽然不会在网页中显示，但它能影响网页的全局设置。

网页头部信息的设置属于页面总体设定的范围，包括页面的标题、页面的说明、关键字、作者信息等内容，虽然它们中大多数不能直接在网页上看到效果，但从功能上，很多都是必不可少的。

2.1.1 网页标题——`<title>` 标签



HTML 文档的标题显示在浏览器的标题栏中，用于说明文件的用途，每个 HTML 文档都应该有标题。网页标题与文章标题的性质是一样的，它们都表示重要的信息，允许用户快速浏览网页，找到自己需要的信息。在互联网上，这是非常重要的，因为网站访问者并不总是阅读网页上的所有文字。在网页中设置网页的标题，只需在 HTML 文件的头部 `<title>` 与 `</title>` 标签之间输入标题信息，就可以在浏览器上显示。

`<title>` 标签的基本语法如下。

```
<head>
<title>...</title>
</head>
```

网页的标题只有一个，位于 HTML 文档的头部 `<head>` 与 `</head>` 标签之间。

提示

网页标题向浏览者提供了网页的内容信息，方便浏览者对页面的选择。设计者也可以在网页标题中加入一些特殊符号，如★、◆等，以增加网页的个性化。