

计算机应用案例教程系列

# HTML5+CSS3 网页设计案例教程

辛明远 石 云 编著

清华大学出版社

北 京

## 内 容 简 介

本书以通俗易懂的语言、翔实生动的案例全面介绍应用 HTML5+CSS3 设计网页的方法与技巧。全书共分 13 章, 内容包括 HTML5 概述、设计网页文本、设计网页图像、设计超链接、CSS3 概述、CSS3 文本样式、CSS3 图像样式、CSS3 盒子模型、CSS3 移动布局、CSS3 变形和动画、设计表格、设计表单和设计多媒体。

本书提供配套的素材文件、与内容相关的扩展教学视频以及云视频教学平台等资源的 PC 端下载地址, 以方便读者扩展学习。本书具有很强的实用性和可操作性, 是一本适合于高等院校及各类社会培训学校的优秀教材, 也是广大中初级计算机用户的首选参考书。

本书对应的电子课件及其他配套资源可以到 <http://www.tupwk.com.cn/teaching> 网站下载, 也可以扫描前言中的二维码推送配套资源到邮箱。

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+CSS3 网页设计案例教程 / 辛明远, 石云 编著. —北京: 清华大学出版社, 2020.6

计算机应用案例教程系列

ISBN 978-7-302-55433-2

I. ①H… II. ①辛… ②石… III. ①超文本标记语言—程序设计—教材 ②网页制作工具—教材  
IV. ①TP312.8 ②TP393.092.2

中国版本图书馆 CIP 数据核字(2020)第 082010 号

责任编辑: 胡辰浩

装帧设计: 孔祥峰

责任校对: 成凤进

责任印制: 宋 林

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](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈: 010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 订 者: 北京鑫海金澳胶印有限公司

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 18.75 彩 插: 2 字 数: 480 千字

版 次: 2020 年 6 月第 1 版 印 次: 2020 年 6 月第 1 次印刷

定 价: 69.00 元

---

产品编号: 076398-01

# 前言

熟练使用计算机已经成为当今社会不同年龄层次的人群必须掌握的一门技能。为了使读者在短时间内轻松掌握计算机各方面应用的基本知识,并快速解决生活和工作中遇到的各种问题,清华大学出版社组织了一批教学精英和业内专家特别为计算机学习用户量身定制了这套“计算机应用案例教程系列”丛书。

## 丛书和配套资源

### ▶ 选题新颖,结构合理,内容精炼实用,为计算机教学量身打造

本套丛书注重理论知识与实践操作的紧密结合,同时贯彻“理论+实例+实战”3阶段教学模式,在内容选择、结构安排方面更加符合读者的认知习惯,从而达到老师易教、学生易学的目的。丛书采用双栏紧排的格式,合理安排图与文字的占用空间,在有限的篇幅内为读者提供更多的计算机知识和实战案例。丛书完全以高等院校及各类社会培训学校的教学需要为出发点,紧密结合学科的教学特点,由浅入深地安排章节内容,循序渐进地完成各种复杂知识的讲解,使学生能够一学就会、即学即用。

### ▶ 配套资源丰富,全方位扩展知识能力

本套丛书配套的素材文件、与本书内容相关的扩展教学视频以及云视频教学平台等资源,可通过在PC端的浏览器中下载后使用。用户也可以扫描下方二维码推送配套资源到邮箱。

(1) 本书配套素材和扩展教学视频文件的下载地址如下。

<http://www.tupwk.com.cn/teaching>

(2) 本书配套资源的二维码如下。



扫码推送配套资源到邮箱

### ▶ 在线服务,疑难解答,贴心周到,方便老师定制教学课件

便捷的教材专用通道(QQ: 22800898)为老师量身定制实用的教学课件。老师也可以登录本丛书的信息支持网站(<http://www.tupwk.com.cn/teaching>)下载图书对应的电子课件。

## 本书内容介绍

《HTML5+CSS3 网页设计案例教程》是这套丛书中的一本,该书从读者的学习兴趣和实际需求出发,合理安排知识结构,由浅入深、循序渐进,通过图文并茂的方式讲解应用



HTML5+CSS3 设计网页的基本知识和常用技巧。全书共分 13 章，主要内容如下。

第 1 章介绍 HTML5 的发展历程、文档结构、编写方法等基础知识。

第 2 章介绍在网页中设计标题、文字、段落、列表、水平线等文本元素的方法。

第 3 章介绍如何使用 HTML5 为网页插入图像。

第 4 章介绍在网页中设计页间链接、锚记链接、下载链接以及图像热点的方法。

第 5 章介绍 CSS3 的基础知识与基本用法。

第 6 章介绍应用 CSS3 设计网页文本样式的方法。

第 7 章介绍使用 CSS3 控制图像大小、边框样式以及阴影等特殊效果的方法。

第 8 章对 CSS3 新增的盒模型属性和功能进行详细介绍。

第 9 章通过具体的实例，重点介绍多列流动布局和弹性盒布局这两种布局排版模式。

第 10 章详细介绍 transform、transition 和 animation 等功能，并通过实例详细讲解在网页中应用变形和动画的方法。

第 11 章通过实例介绍设计 HTML5 表格，并使用 CSS3 定义表格样式(如制作斑马线表格、圆角表格、单线表格等)的方法。

第 12 章结合 HTML5 与 CSS3 介绍在网页中设计表单及表单元素的方法。

第 13 章介绍在网页中设计多媒体元素的方法。

### 读者定位和售后服务

本套丛书为所有从事计算机教学的老师和自学人员而编写，是一套适合于高等院校及各类社会培训学校的优秀教材，也可作为广大初中级计算机用户的首选参考书。

如果您在阅读图书或使用计算机的过程中有疑惑或需要帮助，可以登录本丛书的信息支持网站(<http://www.tupwk.com.cn/teaching>)，本丛书的作者或技术人员会提供相应的技术支持。

本书分为 13 章，黑河学院的辛明远编写了第 1、2、5、7、10~13 章，石云编写了第 3、4、6、8、9 章。

由于作者水平所限，书中难免有不足之处，欢迎广大读者批评指正。我们的邮箱是 [huchenhao@263.net](mailto:huchenhao@263.net)，电话是 010-62796045。

“计算机应用案例教程系列”丛书编委会  
2020 年 6 月

# 目录

<b>第 1 章 HTML5 概述</b> .....1	
1.1 什么是 HTML5.....2	
1.2 HTML5 发展历程.....2	
1.3 HTML5 文档结构.....5	
1.3.1 文档类型声明.....6	
1.3.2 主标签.....6	
1.3.3 头部信息.....6	
1.3.4 主体内容.....10	
1.4 HTML5 文件的编写方法.....11	
1.4.1 手动编写 HTML5 文件.....11	
1.4.2 使用 HTML 编辑器.....12	
1.5 案例演练.....12	
<b>第 2 章 设计网页文本</b> .....17	
2.1 定义标题.....18	
2.2 定义段落.....19	
2.2.1 使用段落标签.....19	
2.2.2 使用换行标签.....19	
2.3 定义文字格式.....20	
2.3.1 字体.....20	
2.3.2 字号.....21	
2.3.3 颜色.....22	
2.3.4 强调.....23	
2.3.5 注解.....23	
2.3.6 备选.....24	
2.3.7 上下标.....24	
2.3.8 术语.....25	
2.3.9 代码.....26	
2.3.10 预定义格式.....26	
2.3.11 缩写词.....27	
2.3.12 编辑提示.....27	
2.3.13 引用.....28	
2.3.14 引述.....29	
2.3.15 换行显示.....29	
2.4 定义文字效果.....30	
2.4.1 高亮.....30	
2.4.2 进度.....31	
2.4.3 刻度.....31	
2.4.4 时间.....32	
2.4.5 联系信息.....33	
2.4.6 显示方向.....34	
2.4.7 换行断点.....34	
2.4.8 旁注.....35	
2.5 案例演练.....35	
<b>第 3 章 设计网页图像</b> .....37	
3.1 网页图像概述.....38	
3.1.1 网页支持的图片格式.....38	
3.1.2 网页图像的路径.....39	
3.2 定义图像.....40	
3.3 定义流.....41	
3.4 定义图标.....42	
3.5 定义响应式图像.....43	
3.6 案例演练.....47	
<b>第 4 章 设计超链接</b> .....49	
4.1 超链接概述.....50	
4.1.1 超链接的类型.....50	
4.1.2 超链接的路径.....50	
4.2 页间链接.....51	
4.3 块链接.....52	
4.4 锚记链接.....53	
4.5 目标链接.....53	
4.6 邮件链接.....54	
4.7 下载链接.....54	
4.8 图像热点链接.....55	
4.9 框架链接.....55	
4.10 案例演练.....56	
<b>第 5 章 CSS3 概述</b> .....61	
5.1 什么是 CSS3.....62	
5.1.1 CSS 历史.....62	
5.1.2 CSS3 模块.....62	
5.1.3 CSS3 特性.....64	
5.2 CSS3 基本用法.....65	
5.2.1 CSS3 样式概述.....65	
5.2.2 应用 CSS3 样式.....66	



5.2.3	CSS3 样式表	67	6.5	特殊值	102
5.2.4	CSS3 代码注释	68	6.6	文本效果	105
5.2.5	CSS3 代码格式化	68	6.6.1	文本阴影	105
5.2.6	CSS3 继承性	69	6.6.2	文本特效	107
5.2.7	CSS3 层叠性	70	6.7	颜色模式	110
5.3	CSS3 选择器	70	6.8	动态内容	115
5.3.1	标签选择器	71	6.9	自定义字体	118
5.3.2	类选择器	71	6.10	案例演练	120
5.3.3	ID 选择器	71	<b>第 7 章</b>	<b>CSS3 图像样式</b>	<b>123</b>
5.3.4	包含选择器	72	7.1	设计图像	124
5.3.5	子选择器	73	7.1.1	图像大小	124
5.3.6	相邻选择器	73	7.1.2	图像边框	125
5.3.7	兄弟选择器	74	7.1.3	半透明图像	126
5.3.8	属性选择器	74	7.1.4	圆形图像	126
5.3.9	结构伪类选择器	77	7.1.5	阴影图像	127
5.3.10	否定伪类选择器	79	7.2	图像背景	128
5.3.11	状态伪类选择器	79	7.2.1	定义背景图像	128
5.3.12	目标伪类选择器	81	7.2.2	背景原点/位置/裁剪	129
5.3.13	动态伪类选择器	81	7.2.3	控制大小	131
5.3.14	伪对象选择器	82	7.2.4	固定显示	132
5.4	案例演练	83	7.3	渐变背景	132
<b>第 6 章</b>	<b>CSS3 文本样式</b>	<b>87</b>	7.3.1	线性渐变与重复线性渐变	133
6.1	CSS3 文本模块概述	88	7.3.2	径向渐变与重复径向渐变	135
6.2	字体样式	92	7.4	案例演练	137
6.2.1	字体	92	<b>第 8 章</b>	<b>CSS3 盒子模型</b>	<b>149</b>
6.2.2	大小	92	8.1	显示方式	150
6.2.3	颜色	93	8.2	可控大小	151
6.2.4	粗细	93	8.3	内容溢出	152
6.2.5	斜体	94	8.4	轮廓线	153
6.2.6	修饰线	94	8.5	圆角边框	155
6.2.7	变体	95	8.6	图像边框	156
6.2.8	大小写	95	8.7	盒子阴影	158
6.3	文本格式	95	8.8	布局方式	166
6.3.1	对齐	95	8.8.1	流动布局	166
6.3.2	间距	97	8.8.2	浮动布局	167
6.3.3	行高	97	8.8.3	定位布局	168
6.3.4	缩进	98	8.9	案例演练	170
6.3.5	换行	98			
6.4	书写模式	100			

<b>第 9 章 CSS3 移动布局</b> .....	173		
9.1 多列布局 .....	174		
9.1.1 定义列宽 .....	174		
9.1.2 定义列数 .....	175		
9.1.3 定义列间距 .....	175		
9.1.4 定义列边框 .....	176		
9.1.5 定义跨列显示 .....	177		
9.1.6 定义列的高度 .....	178		
9.2 盒布局模型 .....	178		
9.2.1 定义宽度 .....	179		
9.2.2 定义顺序 .....	180		
9.2.3 定义方向 .....	181		
9.2.4 自适应大小 .....	182		
9.2.5 消除空白 .....	183		
9.2.6 定义对齐方式 .....	185		
9.3 弹性盒布局 .....	186		
9.3.1 定义弹性盒 .....	186		
9.3.2 定义伸缩方向 .....	187		
9.3.3 定义行数 .....	188		
9.3.4 定义对齐方式 .....	189		
9.3.5 定义伸缩项目 .....	190		
9.4 媒体查询 .....	191		
9.5 案例演练 .....	193		
<b>第 10 章 CSS3 变形和动画</b> .....	197		
10.1 CSS3 变形 .....	198		
10.1.1 2D 旋转 .....	198		
10.1.2 2D 缩放 .....	199		
10.1.3 2D 移动 .....	200		
10.1.4 2D 倾斜 .....	201		
10.1.5 2D 矩阵 .....	202		
10.1.6 变形原点 .....	204		
10.1.7 3D 变形 .....	205		
10.1.8 3D 位移 .....	207		
10.1.9 3D 缩放 .....	208		
10.1.10 3D 旋转 .....	209		
10.2 过渡样式 .....	211		
10.2.1 定义过渡 .....	211		
10.2.2 定义过渡时间 .....	212		
10.2.3 定义延迟 .....	212		
10.2.4 定义动画效果 .....	213		
10.2.5 定义触发时机 .....	213		
10.3 关键帧动画 .....	215		
10.3.1 定义关键帧 .....	216		
10.3.2 定义关键帧动画 .....	216		
10.4 案例演练 .....	218		
<b>第 11 章 设计表格</b> .....	221		
11.1 定义表格 .....	222		
11.1.1 简单表格 .....	222		
11.1.2 列标题 .....	222		
11.1.3 表格的标题 .....	222		
11.1.4 行分组 .....	223		
11.1.5 列分组 .....	223		
11.2 设置表格 .....	224		
11.2.1 内/外框线 .....	224		
11.2.2 单元格间距 .....	225		
11.2.3 细线边框 .....	225		
11.2.4 内容摘要 .....	226		
11.3 设置单元格 .....	226		
11.3.1 跨单元格显示 .....	226		
11.3.2 表头单元格 .....	226		
11.3.3 绑定表头 .....	227		
11.3.4 信息缩写 .....	227		
11.3.5 单元格分类 .....	228		
11.4 设置表格样式 .....	228		
11.5 案例演练 .....	232		
<b>第 12 章 设计表单</b> .....	239		
12.1 定义表单 .....	240		
12.1.1 设计表单结构 .....	240		
12.1.2 组织表单结构 .....	240		
12.1.3 添加提示文本 .....	241		
12.2 定义表单控件 .....	242		
12.2.1 文本框 .....	242		
12.2.2 密码框 .....	245		
12.2.3 文本区域 .....	246		
12.2.4 单选按钮和复选框 .....	247		
12.2.5 选择框 .....	249		
12.2.6 文件域和隐藏域 .....	250		
12.2.7 按钮 .....	250		



12.2.8	数据列表	251	12.3.13	替换文本	261
12.2.9	密钥生成器	252	12.4	设计表单样式	262
12.2.10	输出结果	252	12.5	定制表单	267
12.3	设置表单属性	253	12.6	案例演练	270
12.3.1	名称和值	253	<b>第 13 章</b>	<b>设计多媒体</b>	<b>275</b>
12.3.2	布尔型属性	253	13.1	使用 audio 元素	276
12.3.3	必填属性	255	13.2	使用 video 元素	277
12.3.4	禁止验证	256	13.3	设置媒体属性	278
12.3.5	多选属性	257	13.4	使用媒体方法	282
12.3.6	自动完成	257	13.5	使用媒体事件	283
12.3.7	自动获取焦点	258	13.6	使用<embed>标签	286
12.3.8	所属表单	259	13.7	使用<object>标签	286
12.3.9	表单重写	259	13.8	案例演练	287
12.3.10	高度和宽度	260			
12.3.11	最小值/最大值/步长	260			
12.3.12	匹配模式	261			

# 第 1 章

## HTML5 概述

HTML5 是超文本标记语言 HTML 的第 5 次修订版，是近年来 Web 标准的巨大飞跃。HTML5 相比之前版本的不同之处在于，HTML5 并非仅仅用于表示 Web 内容，它还为 Web 使用者带来了一个无缝的网络，人们无论是通过各种计算机、平板电脑，还是智能手机，都能够方便地浏览基于 HTML5 的各类网站。

本章作为全书的开端，将通过介绍 HTML5 的基础知识，帮助刚刚接触网页设计的新手用户快速入门。



## 1.1 什么是 HTML5

HTML 是用来描述网页的一种语言，是一种标记语言(包含一套标签，HTML 使用标签来描述网页)而不是编程语言。HTML 是制作网页的基础语言，主要用于描述超文本中内容的显示方式。

HTML5 是用于取代于 1999 年制定的 HTML 4.01 和 XHTML 1.0 标准的 HTML 标准版本。HTML5 当前对多媒体的支持更强，新增了以下功能：

- ▶ 语义化标签，使文档结构明确。
- ▶ 文档对象模型(DOM)。
- ▶ 实现了 2D 绘图的 Canvas 对象。
- ▶ 可控媒体播放。
- ▶ 离线存储。
- ▶ 文档编辑。
- ▶ 拖放。
- ▶ 跨文档消息。
- ▶ 浏览器历史管理。
- ▶ MIME 类型和协议注册。

HTML5 最大的优势是语法结构非常简单，它具有以下几个特点。

### 编写简单

HTML5 编写简单，即便是没有任何编程经验的用户，也可以轻易地使用 HTML5 来设计网页，只需要为文本加上一些标签即可。

### 标签数目有限

在 W3C 建议使用的 HTML5 规范中，所

有控制标签都是固定且数目有限的。固定指的是控制标签的名称固定不变，且每个控制标签都已被定义过，提供的功能与相关属性的设置都是固定的。由于 HTML 中只能引用 Strict DTD、Transitional DTD 或 Frameset DTD 中的控制标签，且 HTML 并不允许网页设计者自行创建控制标签，因此控制标签的数目是有限的，设计者在充分了解每个控制标签的功能后，就可以开始设计网页了。

### 语法较弱

在 W3C 指定的 HTML5 规范中，对于 HTML5 在语法结构上的规格限制是比较松散的，如<HTML>、<Html>或<html>在浏览器中具有同样的功能，是不区分大小写的。另外，也没有严格要求每个控制标签都要有对应的结束标记，如<tr>就不一定需要结束标记</tr>。



### 知识点滴

HTML5 最基本的语法是<标记符></标记符>。标记符通常成对使用，有一个起始标记和一个结束标记。结束标记只是在起始标记的前面加上一个斜杠/。当浏览器收到 HTML 文件后，就会解释里面的标记符，然后把标记符对应的功能表达出来。

## 1.2 HTML5 发展历程

从 2010 年开始，HTML5 和 CSS3 就一直是互联网技术中最受关注的两个话题。在 2010 年的 MIX 10 大会上，微软工程师在介绍 IE 9 浏览器时，从 Web 技术的角度把互联网的发展分为以下三个阶段。

- ▶ 阶段一：以内容为主的 Web 1.0 网络时代，Web 主流技术是 HTML 和 CSS。
- ▶ 阶段二：以 Web 2.0 为主的 Ajax 应用，热门技术是 JavaScript、DOM 和异步数

据请求。

- ▶ 阶段三：以 HTML5+CSS3 为主的网络时代，两者相辅相成，使互联网进入一个崭新的发展阶段。

## 1. HTML 历史

HTML 最早是从 2.0 版开始的,没有 1.0 版官方规范。HTML Tag 文档可以算作 HTML 的第一个版本,但不是正式版本。第一个正式版本 HTML 2.0 也不是出自 W3C,而是由 IETF(Internet Engineering Task Force, 互联网工程任务组)制定的。从第三个版本开始, W3C 开始接手并负责后续版本的制定工作。

在 20 世纪 90 年代, HTML 有过几次快速发展。从 1997 年到 1999 年, HTML 的版本从 3.2 更新到 4.0,再发展到 4.01。在 HTML 4.01 之后, W3C 提出了 XHTML 1.0 概念。虽然听起来完全不同,但 XHTML 1.0 与 HTML 4.01 其实是一样的。唯一不同的就是 XHTML 1.0 要求使用 XML 语法。例如,所有属性都必须使用小写字母,所有元素也必须使用小写字母,所有属性值都必须加引号,所有标签都必须有结束标记(对于 `img` 和 `br` 等单独使用的标签,需要使用自结束标记)。

到了 2000 年, Web 标准项目(Web Standards Project)的开展如火如荼,开发人员对浏览器里包含的各种专有特性已经忍无可忍。当时 CSS 有了长足的发展,而且与 XHTML 1.0 的结合也很紧密, CSS+XHTML 1.0 基本上算是最佳实践了。



虽然 HTML 4.01 与 XHTML 1.0 没有本质上的不同,但是大部分开发人员接受了 CSS+XHTML 1.0 这个组合。专业的开发人员能做到元素全部小写,属性全部小写,属性值也全部加引号。此时,由于专业人员起

到带头作用,越来越多的人也都开始支持并使用这种语法。

XHTML 1.0 之后出现了 XHTML 1.1, XHTML 1.1 与 XHTML 1.0 相比,本身并没有什么新东西,元素也都基本相同(属性也相同),唯一的变化就是必须把文档标记为 XML 文档。但是,这样做带来了一些问题,例如:

- ▶ 将文档标记为 XML 文档后,当时低版本 IE 浏览器不能处理(当时,IE9 以下版本无法处理接收到的 XML 文档)。

- ▶ XHTML 1.1 规范要求以 XML 类型来发送文档,这对于大部分用户而言非常不方便。

XHTML 1.1 之后的版本是 XHTML2,但 XHTML2 并没有完成,从理论角度讲, XHTML2 实际上是一个非常好的规范。如果所有人都同意使用,也一定会是一种非常好的格式。只不过 XHTML2 有些不切实际,不可能实现,例如:

- ▶ XHTML2 仍然使用 XML 错误处理模型,用户必须保证以 XML 类型发送文档。

- ▶ XHTML2 有意不再向后兼容已有的 HTML 各个版本。XHTML2 甚至曾经讨论过废除 `img` 元素,这对于每天都在做 Web 开发的人来说难以接受(虽然从理论上讲,使用 `object` 元素可能更好)。

因此,无论 XHTML2 在理论上是多么完美的一种格式,但却从未有机会付诸实践。之所以难以付诸实践,主要的原因是开发人员不支持使用。同时,由于 HTML2 不向后兼容,浏览器厂商也不会支持。



### 知识点滴

为什么 XHTML 1.1 没有像 XML 语法那样得到真正广泛的应用?为什么 XHTML2 从未落到实处?因为它们违反了一条设计原则,那就是著名的伯斯塔法则:“发送时要保守,接收时要开放。” XHTML 1.1 和 XHTML2 都使用 XML 错误处理模型, XML 错误处理模型过于苛刻,不符合接收时要开放的原则,遇到错误就停止解析。



## 2. HTML5 的诞生

20 世纪末, W3C 考虑改良 HTML 语言。2004 年, 在 W3C 成员内部的一次研讨会上, 当时 Opera 公司的代表伊恩·希克森(Ian Hickson)提出了扩展和改进 HTML 的建议, 他建议新的任务组可以跟 XHTML2 并行, 但是在已有 HTML 的基础上开展工作, 目标是对 HTML 进行扩展。

但是 W3C 投票表示反对, 理由是他们认为 HTML 已经落伍, XHTML2 才是未来的发展方向。随后, Opera、Apple 等浏览器厂商以及一些其他成员陆续脱离了 W3C, 并联合成立了 WHATWG(Web Hypertext Applications Technology Working Group, Web 超文本应用技术工作组), 这就为 HTML5 未来的命运埋下了伏笔。

WHATWG 决定完全脱离 W3C, 在 HTML 的基础上开展工作, 向其中添加一些新内容。这个工作组的成员有浏览器厂商, 因此可以保证实现各种新奇、实用的想法。结果工作组中的成员不断提出一些好点子, 并逐一做到了浏览器中。

WHATWG 的工作效率很高, 不久就初见成效。而此时反观 W3C, 在此期间 XHTML2 却没有实质性的进展。这深深触动了 W3C。2006 年, 蒂姆·伯纳斯·李(Tim Berners-Lee)写了一篇博客来反思 HTML 的发展历史, 承认“企图让 Web 一夜之间跨入 XML 时代”的想法不切实际, 应该重新组建 HTML5 工作组。



万维网之父蒂姆·伯纳斯·李

随后, 越来越多的 W3C 成员开始反思, 并最终达成一致。2007 年, W3C 组建了 HTML5 工作组。这个工作组面临两个问题。第一个问题就是,“我们是从头开始做起呢? 还是在 2004 年成立的那个名为 WHATWG 的工作组的既有成果基础上开始工作呢?” 答案是显而易见的, W3C 当然希望从已经取得的成果着手, 以之为基础展开工作。于是, W3C 又组织了一次投票, 同意在 WHATWG 工作成果的基础上继续开展工作。

第二个问题就是如何理顺两个工作组之间的关系。HTML5 规范的编辑应该由谁担任? 是不是还让 WHATWG 的编辑伊恩·希克森(Ian Hickson)兼任? 于是, W3C 又一次组织投票, 投票的结果是赞成让伊恩·希克森担任 HTML5 规范的编辑, 同时兼任 WHATWG 的编辑。

多年来, 两个小组在同一编辑的领导下共同工作。2011 年, 两个小组得出新结论, 他们有不同的目标: W3C 希望为 HTML5 推荐的功能划清界限, 而 WHATWG 希望继续致力于 HTML 标准, 不断维护规范和添加新功能。2012 年, W3C 组建了一支新的编辑团队, 负责创建 HTML5 推荐标准, 并开始为下一个 HTML 版本准备工作草案。

## 3. HTML5 的发展

随着计算机技术和移动网络的不断发展, 可以看到 HTML5 在未来几年的发展将会呈现井喷式增长。具体表现形式如下。

### 向移动端方向发展

HTML5 技术未来的主要发展市场还是移动端互联网领域, 现阶段移动浏览器有应用体验不佳、网页标准不统一的劣势, 这两方面是移动端网页发展的障碍, 而 HTML5 技术能够解决这两个问题, 并且将劣势转变为优势, 整体推动整个移动端网页方面的发展。

### Web 内核标准提升

目前，移动端网页内核大多采用 Web 内核，相信在未来几年内随着智能端逐渐普及，HTML5 在 Web 内核方面的应用将会凸显。

### 提升 Web 操作体验

随着硬件性能的提升、WebGL 标准化的普及以及手机游戏的逐渐成熟，手机游戏向 3D 化发展是大势所趋。

### 网络营销游戏化发展

通过一些游戏化、场景化以及跨屏互动等环节，不仅能够提升用户游戏体验，还能够满足广告主大部分的营销需求，在推销产品的过程中，让用户体验游戏的乐趣。

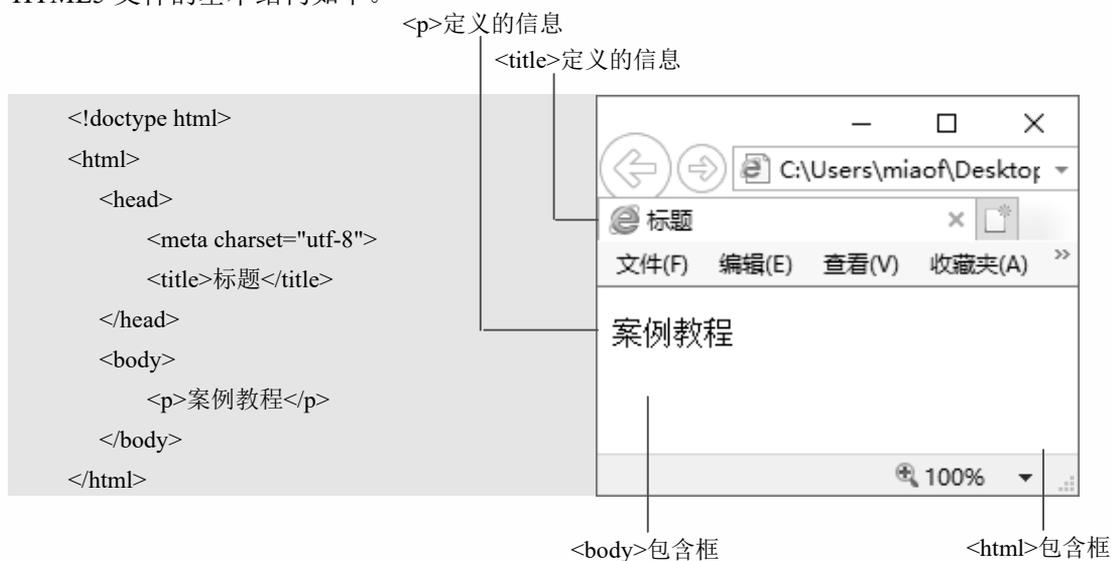
### 移动视频、在线直播

HTML5 将会改变视频数据的传输方式，让视频播放更加流畅，与此同时，视频还能够与网页相结合，让用户看视频就像看图片一样轻松。

## 1.3 HTML5 文档结构

一个完整的 HTML5 文件包括标题、段落、列表、表格、绘制的图形以及各种嵌入对象，这些对象统称为 HTML 元素。

HTML5 文件的基本结构如下。



以上代码中所用标签的说明如下表所示。

HTML5 文件的基本结构中各标签的说明

标 签	说 明	标 签	说 明
<!doctype html>	文档类型声明	<title>	网页标题标签
<html>	主标签	<body>	主体内容标签
<head>	头部信息	<p>	段落标签
<meta>	元信息标签		



从上面的代码可以看出，在 HTML 文件中，几乎所有的标签都是成对使用的，起始标记为< >，结束标记为</>，在这两个标记中可以添加内容。

### 1.3.1 文档类型声明

<!doctype>类型声明必须位于 HTML5 文档的第一行，也就是位于<html>标签之前。该标记用于告知浏览器文档使用的 HTML 规范。<!doctype>类型声明不属于 HTML 标记；它是一条指令，用于告诉浏览器编写页面时所用标记的版本。

HTML5 对文档类型声明进行了简化，简单到 15 个字符就可以了，具体代码如下：

```
<!doctype html>
```

### 1.3.2 主标签

主标签<html>说明当前页面使用 HTML 语言，从而使浏览器软件能够准确无误地解释、显示页面。<html>代表文档的开始。由于 HTML5 语法的松散特性，主标签可以省略，但是为了使之符合 Web 标准和文档的完整性，养成良好的编写习惯，建议不要省略主标签。

主标签以<html>开头、以</html>结尾，文档的所有内容书写在它们之间，语法格式如下：

```
<html>
...
</html>
```

### 1.3.3 头部信息

头部标签<head>用于说明文档头部的相关信息，一般包括标题信息、元信息、CSS 样式和脚本代码等。HTML 的头部信息以<head>开始、以</head>结束，语法格式如下：

```
<head>
...
```

```
</head>
```

<head>标签的作用范围是整篇文档，定义在 HTML 语言头部的内容往往不会在网页上直接显示。

#### 1. 网页标题

HTML 页面的标题一般用来说明页面的用途，显示在浏览器的标题栏中。在 HTML 文档中，标题信息设置在<head>与</head>之间。标题标签以<title>开始、以</title>结束，语法格式如下：

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

标题的内容写在<title>和</title>之间，可以帮助用户更好地识别页面。预览网页时，设置的标题显示在浏览器的左上方标题栏中。此外，在 Windows 任务栏中显示的也是网页标题。

#### 【例 1-1】使用<title>标签定义网页标题。

```
<html>
<head>
▶ <title>定义标题</title>
</head>
<body>
HTML5 标签列表
</body>
</html>
```

浏览器会将头部信息放在窗口的标题栏或状态栏中显示，如下图所示。



当用户为文档加入用户的链接列表、收藏夹或书签列表时，标题将作为文档链接的默认名称。



### 知识点滴

<title>标签是必需的，其不能包含任何格式、HTML、图像或指向其他页面的链接。一般网页编辑器会预先为页面标题填写默认文字(例如，“网页标题”)。

## 2. 网页元信息

元信息标签<meta>可以提供有关页面的元信息(meta-information)，比如针对搜索引擎和更新频度的描述及关键词。

<meta>标签位于文档的头部，不包含任何内容。<meta>标签的属性定义了与文档相关联的名称/值，<meta>标签提供的属性及取值说明如下表所示。

<meta>标签提供的属性及取值说明

属性	值	描述
charset	character encoding	定义文档的字符编码方式
content	some_text	定义与 http-equiv 或 name 属性相关的元信息
scheme	some_text	定义用于翻译 content 属性值的格式
http-equiv	content-type expires refresh set-cookie	把 content 属性关联到 HTTP 头部
name	author description keywords generator revised others	把 content 属性关联到名称

下面将介绍常用元信息的设置代码。

### 定义网页编码信息

通过使 http-equiv 等于 content-type，可以设置网页的编码。

▶ 下面的代码告诉浏览器，网页使用 utf-8 编码显示：

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



### 知识点滴

HTML5 简化了字符编码设置方式，简化后的 <meta charset="utf-8">与以上代码的作用是相同的。

▶ 以下代码告诉浏览器，网页使用 gb2312 编码显示：

```
<meta http-equiv="content-type"
content="text/html; charset=gb2312" />
```



### 知识点滴

每个 HTML 文档都需要设置字符编码类型，否则可能会出现乱码，其中 utf-8 是国家通用编码，独立于任何语言。

### 定义搜索引擎的关键词

早期，meta keywords 关键字标签不仅对搜索引擎的排名算法起到一定的作用，而且也是许多人进行网页优化的基础。关键字在浏览时是看不到的，其使用格式如下：

```
<meta name="keywords" content="关键字,keyword" />
```



此处应注意的是：

➤ 不同的关键字之间，应使用半角逗号隔开(英文输入状态下)，不要使用空格或进行间隔。

➤ 是 keywords 而不是 keyword。

➤ 关键字标签中的内容应该是一个个的短语而不是一段话。



#### 知识点滴

关键字标签曾经是搜索引擎排名中很重要的因素，但现在已经被很多搜索引擎完全忽略。虽然加上关键字标签对网页的综合表现没有坏处，但是，如果使用不恰当，对网页非但没有好处，还有欺诈骗的嫌疑。

### 定义网页描述信息

meta description 元标签(描述元标签)是一种 HTML 元标签，用来简略描述网页的主要内容，通常被搜索引擎用在搜索结果页上，给最终用户展示一段文字。网页描述信息在网页中是显示不出来的，其使用格式如下：

```
<meta name="description" content="网页介绍文字" />
```

### 定义网页语言代码

使用 content-language 属性值可以定义网页语言代码。例如，设置中文版本语言的代码如下：

```
<meta http-equiv="content-language" content="zh-CN" />
```

### 定义页面定时跳转

使用<meta>标签可以使网页在经过一定时间后自动刷新，这可通过将 http-equiv 属性值设置为 refresh 来实现。content 属性值可以设置为更新时间。

你在浏览网页时经常会看到一些显示了欢迎信息的页面，经过一段时间后，这些页面会自动转到其他页面，这就是网页的跳转。定义页面定时跳转的语法格式如下：

```
<meta http-equiv="Refresh" content="秒;[url=网址]" />
```

上面的“[url=网址]”部分是可选项，如果有这部分，页面定时刷新并跳转；如果省略这部分，页面只定时刷新，不进行跳转。例如，要实现每 5 秒刷新一次页面，将下面的代码放入 head 标记部分即可：

```
<meta http-equiv="Refresh" content="5" />
```

例如，10 秒后跳转到百度首页：

```
<meta http-equiv="refresh" content="10; url=https://www.baidu.com" />
```

### 定义网页缓存时间

使用 expires 属性值可以设置网页缓存时间，例如：

```
<meta http-equiv="expires" content="Sunday 20 October 2022 03:00 GMT" />
```

也可以使用以下方式设置页面不缓存：

```
<meta http-equiv="pragma" content="no-cache" />
```



#### 知识点滴

元信息的设置还包括设置网页作者、设置网页创建时间、设置禁止搜索引擎检索、设置网页版权信息等，用户可以参考 HTML 手册。

### 3. 文档视口

在移动 Web 开发中，经常需要定义 viewport(视口)——浏览器显示页面内容的屏幕区域。一般浏览器都默认设置了<meta name="viewport">标签，从而定义虚拟的布局视口，用于解决早期页面在手机上显示的问题。

由于 iOS 和 Android 等系统基本都将视口分辨率设置为 980px，因此桌面网页基本能够在手机上呈现，只不过显示得很小，用户可以通过手动缩放网页进行阅读。但这种方式的用户体验很差，建议使用<meta name="viewport">标签设置视口大小。

设置<meta name="viewport">标签的具体代码如下：

```
<meta id="viewport" name="viewport" content="width=device-width; initial-scale=1.0; maximum-scale=1; user-scalable=no; ">
```

其中，各属性说明如下表所示。

<meta name="viewport">标签的属性设置说明

属 性	值	描 述
width	正整数或 device-width	定义视口的宽度，单位为像素
height	正整数或 device-height	定义视口的高度，单位为像素(一般不用)
initial-scale	[0.0~10.0]	定义初始缩放值
minimum-scale	[0.0~10.0]	定义缩小的最小比例，必须小于或等于 maximum-scale 设置
maximum-scale	[0.0~10.0]	定义放大的最大比例，必须大于或等于 minimum-scale 设置
user-scalable	yes/no	定义是否允许用户手动缩放页面(默认值为 yes)

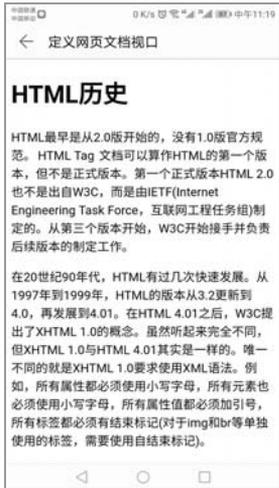
例如，以下代码可在网页中输入标题和文本。

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>定义网页文档视口</title>
▶ <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<body>
<h1>HTML 历史</h1>
<p>HTML 最早是从 2.0 版开始的，没有 1.0 版官方规范。HTML Tag 文档可以算作 HTML 的第一个版本，但不是正式版本。第一个正式版本 HTML 2.0 也不是出自 W3C，而是由 IETF(Internet Engineering Task Force，互联网工程任务组)制定的。从第三个版本开始，W3C 开始接手并负责后续版本的制定工作。</p>
<p>在 20 世纪 90 年代，HTML 有过几次快速发展。从 1997 年到 1999 年，HTML 的版本从 3.2 更新到 4.0，再发展到 4.01。在 HTML 4.01 之后，W3C 提出了 XHTML 1.0 的概念。虽然听起来完全不同，但 XHTML 1.0 与 HTML 4.01 其实是一样的。唯一不同的就是 XHTML 1.0 要求使用 XML 语法。例如，所有属性都必须使用小写字母，所有元素也必须使用小写字母，所有属性值都必须加引号，所有标签都必须有结束标记(对于 img 和 br 等单独使用的标签，需要使用自结束标记)。</p>
</body>
</html>
```

如果没有设置文档视口，那么在移动设置中呈现的效果如下页左图所示；而设置了文档视口之后，呈现的效果将如下页右图所示。



默认情况下缩小的页面视图



正常显示的页面视图

### 1.3.4 主体内容

网页上要显示的内容都放在网页的主体标签内，它是 HTML 文件的重点所在。主体标签以<body>开始、以</body>结束，语法格式如下：

```
<body>
...
</body>
```

HTML5 包含一百多个标签，大部分继承自 HTML4(HTML5 中增加了 30 个标签)，这些标签基本上都被放置在主体标签中(本书将在后面的章节中详细介绍)。

正确选用 HTML5 标签可以避免代码冗余。在设计网页时不仅需要使用<div>标签来构建网页通用结构，还需要使用下面几类标签完善网页结构。

▶ <h1><h2><h3><h4><h5><h6>：定义文档标题，其中 h1 表示一级标题，h6 表示 6 级标题，网页中常用的标题包括一级、二级和三级标题。

▶ <p>：定义段落文本。

▶ <ul><ol><il>等：定义信息列表、导航列表、榜单结构等。

▶ <table><tr><td>等：定义表格结构。

▶ <form><input><textarea>等：定义表

单结构。

▶ <span>：定义行内包含框。

**【例 1-2】**使用 HTML 标签创建一个简单的网页，演示主体内容是如何在浏览器中显示的。 

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>主体内容在浏览器中的显示</title>
</head>
<body>
<h1>HTML5 文档结构</h1>
<p>一个完整的 HTML5 文件包括标题、段落、列表、表格、绘制的图形及各种嵌入对象，这些对象统称为 HTML 元素。</p>
<ul>
<li>文档类型声明</li>
<li>主标签</li>
<li>头部信息</li>
<li>主体内容</li>
</ul>
</body>
</html>
```

将以上 HTML5 文件使用浏览器打开后，可以看到如下图所示的效果。



网页显示效果

为了更好地使用标签，用户可以参考 w3cschool 网站：

<http://www.w3school.com.cn/tags/index.asp>

## 1.4 HTML5 文件的编写方法

HTML5 文件的编写方法有以下两种。

### 1.4.1 手动编写 HTML5 文件

由于 HTML5 是一种标记语言，主要以文本形式存在，因此所有记事本工具都可以作为开发环境。HTML 文件的扩展名为 .html 或 .htm，将 HTML 源代码输入记事本、Sublime Text、WebStorm 等编辑器并保存之后，就可以在浏览器中打开文档以查看其效果。

**【例 1-3】**使用记事本工具编写一个简单的 HTML5 文件。  **素材**

**step 1** 使用 Windows 系统自带的记事本工具新建一个文本文件，保存为 index.html(注意，扩展名为.html 而不是.txt)。

**step 2** 输入以下多行字符：

```
<html>
<title>一个简单的网页</title>
```

```
<body>
  <h1>HTML5 简介</h1>
  <h3>HTML5 的新增功能</h3>
  <h3>HTML5 的语法特点</h3>
  <h2>HTML5 文件的基本结构</h2>
  <h2>HTML5 文件的编写方法</h2>
</body>
</html>
```

**step 3** 双击保存的网页文件，即可在浏览器中预览效果。





## 1.4.2 使用 HTML 编辑器

除了使用记事本工具手动编写 HTML5 文件以外，用户还可以使用 HTML 编辑器编写网页。目前，可用于网页开发的 HTML 编辑器有很多，例如 Adobe Dreamweaver、EditPlus、Sublime Text 以及 WebStorm。用户可以根据自己的使用习惯进行选择。

### Adobe Dreamweaver

Adobe Dreamweaver 是 Adobe 公司推出的网站开发工具，是一款集网页制作和网站管理于一身的所见即所得的网页编辑工具。利用 Dreamweaver，用户可以轻而易举地设计出跨平台且不受浏览器限制的网页效果。



### EditPlus

EditPlus 是 Windows 系统中的文本、HTML、PHP 及 Java 编辑器。EditPlus 不但是“记事本”工具的很好的替代工具，同时也为网页制作者和程序设计者提供了许多强大的功能。

EditPlus 为 HTML、PHP、Java、C/C++、CSS、ASP、Perl、JavaScript 和 VBScript 提

供了语法突出显示功能。同时，根据自定义语法文件，EditPlus 能够扩展支持其他程序语言。此外，支持使用无缝网络浏览器预览 HTML 页面，以及使用 FTP 命令上载本地文件到 FTP 服务器。

### Sublime Text

Sublime Text 支持对多种编程语言的语法进行高亮显示，并且拥有优秀的代码自动完成和代码片段功能。



用户可以使用 Sublime Text 将常用的代码片段保存起来，在需要使用时随时调用。Sublime Text 支持 VIM 模式、支持宏。使用该工具编写网页代码，可以大大提高编码的速度和效率。

### WebStorm

WebStorm 是 JetBrains 公司旗下的一款 JavaScript 开发工具。该工具被广大中国 JavaScript 开发者誉为 Web 开发神器、最强大的 HTML5 编辑器、最智能的 JavaScript IDE。该工具与 IntelliJ IDEA 同源，集成了 IntelliJ IDEA 强大的 JavaScript 部分的功能。

## 1.5 案例演练

本章简单介绍了 HTML5 的发展历史、文档结构、编写方法等知识。实际上，HTML5 的基础知识远不止这些。下面的案例演练部分将通过实例，帮助用户进一步掌握 HTML5 页面的特征。

**【例 1-4】** 在网页中使用 div 元素。  **素材**

在网页中 <div> 标签可以把页面划分为

独立的、不同的部分。div 元素可以用作严格的组织工具、通用容器，且不使用任何格

式与之关联。例如:

```
<!doctype html>
<html><head>
<meta charset="utf-8">
</head><body>
<div>
  <article>
    <h1>文章标题</h1>
    <p>文章内容</p>
    <footer>
      <p>注释信息</p>
      <address><a href="#">HTML5_CSS3</a></address>
    </footer>
  </article>
</div>
</body></html>
```



以上代码在浏览器中的预览效果如右上图所示。在页面中,我们用一个 div 元素包裹着所有的内容,语义没有发生改变,但有了一个可以用 CSS 添加样式的通用容器(div 元素是一种完全没有任何语义含义的容器,网页设计者可以为之添加样式或 JavaScript 效果)。

**【例 1-5】**在网页代码中使用 span 元素对段落文本中的部分信息进行分隔显示,以便应用不同的样式。📁 素材

在 HTML 中,【例 1-4】中介绍的 div 并不是唯一没有语义价值的元素。span 是与 div 对应的另一个元素(div 是块级别内容的无语义容器,而 span 则是短语内容的无语义容器)。span 元素可以放在段落元素 p 内,例如:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<h1>文章标题</h1>
<p>文章内容</p>
```

```
<p>发布于<span class="date">2022 年 12 月
▶</span>, 由<span class="author">王小燕
</span>最后编辑</p>
</body>
</html>
```

以上代码在浏览器中的预览效果如下图所示。



**【例 1-6】**构建一个简单的列表结构,为其分配一个 id,并自定义导航模块。📁 素材

HTML 是一种简单的文本标识语言而不是界面语言。HTML 文档结构大部分使用 <div> 标签来完成,为了能够识别不同的结构,一般通过定义 id 或 class 来为它们赋予额外的语义,给 CSS 提供有效的“钩子”。



例如:

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
▶<ul id="nav">
  <li><a href="#">首页</a></li>
  <li><a href="#">视频</a></li>
  <li><a href="#">发现</a></li>
  <li><a href="#">游戏</a></li>
  <li><a href="#">设置</a></li>
</ul>
</body>
</html>

```

在使用 id 标识页面上的元素时, id 名必须是唯一的。id 可以用来标识持久的结构性元素(例如主导航或内容区域), 或者用来标识一次性元素(如某个链接或表单元素)。此外, 在整个网站中, id 名应该应用于语义相似的元素以避免混淆(例如, 如果商品表单和商品详细信息显示在不同的页面上, 那么可以给它们分配同样的 id 名 contact; 但是, 如果在外部样式表中为它们定义样式, 就会遇到问题, 因此建议使用不同的 id 名, 如 contact\_form 和 contact\_details)。

**【例 1-7】**在网页中为单个元素应用 class 标识。



与【例 1-6】中介绍的 id 不同, class 标识可以应用于页面上任意数量的元素, 因此 class 非常适合标识样式相同的对象。例如:

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>

```

```

<h1 class="newsHead">新闻标题</h1>
<p class="newsText">第一条新闻</p>
▶<p class="newsText"><a href="news.php"
class="newsLink">相关新闻</a></p>
</div>
</body>
</html>

```

以上代码在浏览器中的预览效果如下图所示。我们设计了一个简单的新闻页面, 页面中的每个元素都使用一个与新闻相关的类名进行标识。这使得新闻标题和新闻内容可以采用与页面其他部分不同的样式。



在实际设计中, 如果新闻条目较多, 往往不需要使用这么多类来区分每个元素。设计人员也可以将新闻条目放在一个包含框中, 并加上类名 news, 从而标识整个新闻条目, 然后使用包含框选择器识别新闻标题或文本:

```

<div class="news">
  <h1>新闻标题</h1>
  <p>第一条新闻</p>
  <p><a href="news.php">相关新闻</a>
  </p>
</div>

```

由此可以看出, 删除不必要的类有助于简化代码, 使页面更简洁。

**【例 1-8】**为网页中的任意元素(例如链接)添加 title 属性。 **素材**

在设计网页时,可以使用 title 属性为文档中的任何部分加上提示标签。例如:

```
<ul title="列表提示">
  <li><a href="#" title="链接提示">列表项
  目</a></li>
</ul>
```

以上代码在网页中的预览效果如下图所示。当光标指向添加了说明的标签元素时,就会显示 title 信息。



**【例 1-9】**演示在网页的文档结构中应用 role 属性。 **素材**

下面的代码使用 role 属性告诉屏幕阅读器,此处有一个复选框,并且已经被选中:

```
<div role="checkbox" aria-checked="checked">
<input type="checkbox" checked></div>
```

role 是 HTML5 新增属性,作用是告诉屏幕阅读器当前元素所扮演的角色。使用 role 属性可以增强网页文本的可读性和语义化。

常用的 role 角色值如下表所示。

常用的 role 角色值

角色值	说 明
role="banner"(横幅)	面向全站内容,通常包含网站标志、网站赞助者标志、全站搜索工具等。横幅通常显示在页面的顶端,而且通常横跨整个页面 <b>使用方法:</b> 应将其添加到页面级的 header 元素,每个页面只用一次
role="navigation"(导航)	文档内不同部分或相关文档的导航性元素(通常为链接)的集合 <b>使用方法:</b> 与 nav 元素是对应关系,应将其添加到每个 nav 元素,或添加到其他包含导航性链接的容器。这个角色可在每个页面上使用多次,但是同 nav 一样,不应过度使用
role="main"(主体)	文档的主要内容 <b>使用方法:</b> 与 main 元素是对应关系,最好将其添加到<main>标签中,也可以添加到其他表示主题内容的元素(可能是 div 元素)中。每个页面只用一次
role="complementary"(补充性内容)	文档中作为主体内容补充的支撑部分,对区分主题内容是有意义的 <b>使用方法:</b> 与 aside 元素是对应关系,应将其添加到 aside 或 div 元素中(前提是它们仅包含补充性内容)。可以在一个页面上包含多个 complementary 角色,但不要过度使用
role="contentinfo"(内容信息)	包含关于文档信息的大块可感知区域,这类信息的例子包括版权说明、指向隐私权声明的链接等 <b>使用方法:</b> 应将其添加至整个页面的页脚(通常为 footer 元素),每个页面只用一次



下面的代码演示了在网页文档结构中如何应用 role 属性。

```
<!--开始页面容器-->
<div class="container">
  <header role="banner">
    <nav role="navigation">[包含多个链接的列表]</nav>
  </header>
  <!--应用 CSS 后的第一栏-->
    <main role="main">
      <article></article>
      <article></article>
      [其他区块]
    </main>
  <!--结束第一栏-->
  <!--应用 CSS 后的第二栏-->
    <div class="sidebar">
      <aside role="complementary"></aside>
      <aside role="complementary"></aside>
      [其他区块]
    </div>
  <!--结束第二栏-->
  <footer role="contentinfo"></footer>
</div>
<!--结束页面容器-->
```

```
<footer role="contentinfo"></footer>
</div>
<!--结束页面容器-->
```

在主要区块的开头和结尾添加注释是一种常见的做法，代码中的注释信息在页面中无法看到。这样当多人合作开发网页时，每个参与编辑网页的人员都能理解其他人所写代码的意义，便于对代码进行修改。

在将网页发布到网站之前，使用浏览器预览一下添加了注释的页面，能够避免由于弄错注释格式而导致注释内容直接暴露给网页浏览者的情况发生。

**【例 1-10】**在 HTML 文档中添加注释。  **素材**

通过在 HTML 文档中添加注释，可以标明区块的开始和结束位置，提示某段代码的意图，或者阻止内容的显示。例如：

```
<!--开始页面容器-->
<div class="container">
  <header role="banner"></header>
  <!--应用 CSS 后的第一栏-->
    <main role="main"></main>
  <!--结束第一栏-->
  <!--应用 CSS 后的第二栏-->
    <div class="sidebar"></div>
  <!--结束第二栏-->
```