



普通高等教育“十一五”国家级规划教材



“十二五”职业教育国家规划教材

获全国优秀畅销书奖

网页设计与制作教程

Web 前端开发

第6版

刘瑞新 主编

曹利 张红娟 副主编



机械工业出版社

本书依据《Web 前端开发职业技能等级标准（初级）》和部分示范院校的《Web 前端技术课程教学标准》编写，内容包括 Web 页面制作基础、HTML5 和 CSS3 开发基础与应用、JavaScript 程序设计和现代标准的社区新闻网站制作实例。本书以模块化的结构来组织章节，选取静态网站设计与制作的典型应用作为教学案例。

本书适合作为高等院校 Web 前端技术课程、网页设计与制作课程的教材，也可以作为 1+X 证书中 Web 前端开发职业技能等级（初级）的教学及参考用书。

本书配有微课视频、电子课件、授课计划、课程标准、模拟题及答案，以及书中所有例题、习题、实训的素材和源代码等资源，其中微课视频扫描书中二维码即可观看，需要其他配套资源的教师可登录 www.cmpedu.com 免费注册，审核通过后下载，或联系编辑索取（微信：15910938545，电话：010-88379739）。

图书在版编目（CIP）数据

网页设计与制作教程：Web 前端开发 / 刘瑞新主编. —6 版. —北京：机械工业出版社，2021.1（2021.2 重印）

“十二五”职业教育国家规划教材

ISBN 978-7-111-66646-2

I. ①网… II. ①刘… III. ①网页制作工具—高等职业教育—教材
IV. ①TP393.092

中国版本图书馆 CIP 数据核字（2020）第 184217 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）

策划编辑：王海霞 责任编辑：王海霞 汤 枫

责任校对：张艳霞 责任印制：郜 敏

北京富博印刷有限公司印刷

2021 年 2 月·第 6 版·第 2 次印刷

184mm×260mm·20.25 印张·502 千字

2501—4400 册

标准书号：ISBN 978-7-111-66646-2

定价：69.00 元

电话服务

客服电话：010-88361066

010-88379833

010-68326294

封底无防伪标均为盗版

网络服务

机 工 官 网：www.cmpbook.com

机 工 官 博：weibo.com/cmp1952

金 书 网：www.golden-book.com

机工教育服务网：www.cmpedu.com

前 言

“Web 前端开发”课程（以前称“网页制作”）是计算机类专业的专业支撑课程。

随着移动互联网技术的高速发展，Web 前端开发技术被应用到众多领域，已经成为网站开发、App 开发及智能终端设备界面开发的主要技术，因而前端开发岗位出现了极为明显的人才短缺。《国家职业教育改革实施方案》《国务院办公厅关于深化产教融合的若干意见》《国家信息化发展战略纲要》对培养人才提出了新的要求，工业和信息化部教育与考试中心依据教育部《职业技能等级标准开发指南》中的相关要求，制定了《Web 前端开发职业技能等级标准》。

本书依据《Web 前端开发职业技能等级标准（初级）》和部分示范院校的《Web 前端技术课程教学标准》编写，内容包括 Web 页面制作基础、HTML5 和 CSS3 开发基础与应用、JavaScript 程序设计和现代标准的社区新闻网站制作实例。本书以模块化的结构来组织章节，选取静态网站设计与制作的典型应用作为教学案例。以网站建设和网页设计为中心，以实例为引导，把介绍知识与实例设计、制作、分析融于一体，贯穿于教材之中。考虑到网页制作较强的实践性，本书配备大量的例题并提供源码和网页效果图，能够有效地帮助读者理解所学习的理论知识，系统全面地掌握 Web 前端开发技术。每章附有练习题，并提供习题答案，供读者在课外巩固所学的内容。

本书采用考试中心推荐的 HBuilder X 编辑网页代码，在 Google Chrome 或 Microsoft Edge 浏览器中调试运行，同时详细讲解了浏览器开发者工具的使用。

为配合教学，方便教师讲课，编者精心制作了微课视频、电子课件、授课计划、课程标准、模拟题及答案素材和源代码等教学资源。其中，微课视频可以搭建网络课程，扫描书中二维码即可观看；课件浓缩了本书的教学要点，可作为教师的板书来演示；老师们可从机械工业出版社教育服务网（<http://www.cmpedu.com>）下载。

本书适合作为高等院校 Web 前端技术课程、网页设计与制作课程的教材，也可以作为 1+X 证书中 Web 前端开发职业技能等级（初级）的教学及参考用书。

本书由刘瑞新担任主编，曹利、张红娟担任副主编，参与编写的作者有刘瑞新（第 1、2、5、10 章）、曹利（第 3、9、11 章）、张红娟（第 4、8 章）、侯燕萍（第 6 章）、孙立友（第 7 章），第 12 章及资料的收集整理由徐维维完成。全书由刘瑞新教授统稿。

由于作者水平有限，书中疏漏和不足之处难免，敬请广大师生指正。

编 者

目 录

前言

第1章 HTML5 概述 1

1.1 Web 的基本概念 1

1.1.1 WWW 1

1.1.2 Web 服务器 1

1.1.3 网页浏览器 1

1.1.4 网站 2

1.1.5 网页 2

1.1.6 URL 2

1.1.7 标记语言 2

1.1.8 网页标准 2

1.2 HTML5 的基本结构和语法规则 3

1.2.1 HTML5 文档的基本结构 3

1.2.2 HTML5 的基本语法 5

1.2.3 HTML 的全局属性 7

1.2.4 元素的分类 8

1.2.5 HTML 的字符实体和颜色表示 10

1.2.6 HTML5 开发人员编码规范 11

1.3 编辑 HTML 文件 12

1.3.1 常用 HTML 编辑软件 12

1.3.2 网页文件的创建 13

1.4 实训——制作社区网版权信息 13

习题 1 14

第2章 块级元素 15

2.1 基本块级元素 15

2.1.1 标题元素 h1~h6 15

2.1.2 段落元素 p 和换行元素 br 15

2.1.3 水平线元素 hr 16

2.1.4 注释元素 17

2.2 列表元素 17

2.2.1 无序列表元素 ul 17

2.2.2 有序列表元素 ol 18

2.2.3 定义列表元素 dl 19

2.2.4 嵌套列表 20

2.3 表格元素 table 20

2.3.1 基本表格 20

2.3.2 跨行跨列表格 21

2.3.3 表格数据的分组 22

2.3.4 调整列的格式 23

2.4 表单 24

2.4.1 表单元素 form 24

2.4.2 输入元素 input 25

2.4.3 标签元素 label 27

2.4.4 选择栏元素 select 27

2.4.5 按钮元素 button 29

2.4.6 多行文本元素 textarea 29

2.5 分区元素 div 30

2.6 缩排元素 blockquote 31

2.7 实训——制作精选信息版块 32

习题 2 33

第3章 行内元素 34

3.1 格式化元素 34

3.1.1 字体样式元素 34

3.1.2 短语元素 35

3.2 图像元素 img 36

3.3 超链接元素 a 37

3.4 图像热区超链接元素 map、area 41

3.5 范围元素 span 42

3.6 多媒体元素 42

3.6.1 音频元素 audio 43

3.6.2 视频元素 video 44

3.7 用 HBuilder X 编辑 HTML 文档 45

3.8 实训——制作广告版块 47

习题 3 49

第4章 CSS3 基础 50

4.1 CSS 简介 50

4.1.1 CSS 的发展历史 50

4.1.2 CSS 设计与编写原则 51

4.2 在 HTML 中使用 CSS 的方法 52

4.2.1 行内样式 52

4.2.2 内部样式 53

4.2.3 链入外部样式表文件 54

4.2.4 导入外部样式表文件 56

4.3 CSS 的两个主要特性	57	5.2.4 小写字体属性 font-variant	101
4.4 CSS 的基本语法	59	5.2.5 字体粗细属性 font-weight	102
4.5 CSS 的选择器	60	5.2.6 字体简写属性 font	102
4.5.1 元素选择器	60	5.2.7 CSS3 新增使用服务器字体	103
4.5.2 通配符选择器	61	5.3 CSS 文本属性	103
4.5.3 属性选择器	61	5.3.1 文本颜色属性 color	104
4.5.4 派生选择器	63	5.3.2 文本方向属性 direction	104
4.5.5 兄弟选择器	65	5.3.3 字符间隔属性 letter-spacing	104
4.5.6 id 选择器	66	5.3.4 行高属性 line-height	104
4.5.7 class 选择器	67	5.3.5 文本水平对齐方式属性 text-align	105
4.5.8 伪类选择器	67	5.3.6 为文本添加装饰属性 text-decoration	105
4.5.9 UI 元素状态伪类选择器	70	5.3.7 段落首行缩进属性 text-indent	105
4.5.10 结构伪类选择器	76	5.3.8 文本的阴影属性 text-shadow	105
4.5.11 其他伪类选择器	79	5.3.9 文本的大小写属性 text-transform	106
4.5.12 伪元素选择器	81	5.3.10 元素内部的空白属性 white-space	106
4.6 属性值的写法和单位	84	5.3.11 单词之间的间隔属性 word-spacing	106
4.6.1 长度、百分比单位	84	5.3.12 文本的截断效果属性 text-overflow	106
4.6.2 色彩单位	85	5.3.13 文本的换行方式属性 word-break	107
4.7 HTML 文档结构与元素类型	86	5.3.14 单词断字属性 word-wrap	107
4.7.1 文档结构的基本概念	86	5.4 CSS 尺寸属性	108
4.7.2 元素类型	87	5.4.1 宽度属性 width	108
4.8 实训——制作内容详情页	87	5.4.2 高度属性 height	109
习题 4	89	5.4.3 最小宽度属性 min-width	109
第 5 章 CSS3 的属性	90	5.4.4 最大宽度属性 max-width	109
5.1 CSS 背景属性	90	5.4.5 最小高度属性 min-height	110
5.1.1 背景颜色属性 background-color	90	5.4.6 最大高度属性 max-height	110
5.1.2 背景图像属性 background-image	91	5.5 CSS 列表属性	111
5.1.3 重复背景图像属性 background-repeat	92	5.5.1 图像作为列表项标记属性 list-style-image	111
5.1.4 固定背景图像属性 background- attachment	93	5.5.2 列表项标记位置属性 list-style-position	111
5.1.5 背景图像位置属性 background- position	93	5.5.3 列表项标记类型属性 list-style-type	111
5.1.6 背景图像大小属性 background-size	95	5.5.4 列表简写属性 list-style	112
5.1.7 背景属性 backgroud	95	5.6 CSS 表格属性	114
5.1.8 背景覆盖区域属性 background-clip	96	5.6.1 合并边框属性 border-collapse	114
5.1.9 背景图像起点属性 background-origin	96	5.6.2 边框间隔属性 border-spacing	114
5.1.10 背景渐变属性 background-image	97	5.6.3 标题位置属性 caption-side	115
5.2 CSS 字体属性	100	5.6.4 单元格无内容显示方式属性 empty-cells	115
5.2.1 字体类型属性 font-family	101		
5.2.2 字体尺寸属性 font-size	101		
5.2.3 字体倾斜属性 font-style	101		

5.6.5 表格设置方式属性 table-layout	116	6.5.2 列宽属性 column-width	162
5.7 CSS 内容属性	117	6.5.3 列宽属性 column	162
5.8 CSS 属性的应用	118	6.5.4 列之间的间隔属性 column-gap	162
5.8.1 设置图像样式	118	6.5.5 是否横跨所有列属性 column-span	163
5.8.2 设置链接	121	6.5.6 列间隔样式属性 column-rule-style	163
5.8.3 创建导航菜单	122	6.5.7 列之间间隔颜色属性 column-rule-color	163
5.9 实训——制作社区网页面	125	6.5.8 列之间宽度属性 column-rule-width	163
5.9.1 制作通知公告版块	125	6.5.9 列之间间隔所有属性 column-rule	163
5.9.2 制作导航栏	127	6.6 CSS 基本布局样式	165
习题 5	128	6.6.1 CSS 布局类型	165
第 6 章 CSS 的盒模型	130	6.6.2 CSS 布局样式	165
6.1 CSS 盒模型的组成和大小	130	6.7 实训——制作社区网网页	169
6.1.1 盒子的组成	130	6.7.1 制作新闻图片板块	169
6.1.2 盒子的大小	131	6.7.2 制作热点关注板块	171
6.1.3 块级元素与行内元素的宽度和 高度	132	习题 6	172
6.2 CSS 盒模型的属性	133	第 7 章 JavaScript 程序设计基础	174
6.2.1 CSS 内边距属性 padding	133	7.1 JavaScript 概述	174
6.2.2 CSS 外边距属性 margin	135	7.1.1 JavaScript 的诞生	174
6.2.3 CSS 边框属性 border	137	7.1.2 ECMAScript 的版本	175
6.2.4 圆角边框属性 border-radius	140	7.1.3 JavaScript 的特点	175
6.2.5 盒模型的阴影属性 box-shadow	141	7.2 在 HTML 文档中使用 JavaScript	176
6.2.6 图片边框属性 border-image	142	7.2.1 在 HTML 文档中嵌入脚本程序	176
6.2.7 CSS 轮廓属性 outline	145	7.2.2 链接脚本文件	177
6.2.8 调整大小属性 resize	148	7.2.3 在 HTML 标签内添加脚本	178
6.3 CSS 布局属性	148	7.3 数据类型	179
6.3.1 元素的布局方式概述	148	7.3.1 数据类型的分类	179
6.3.2 CSS 浮动属性 float	150	7.3.2 基本数据类型	179
6.3.3 清除浮动属性 clear	151	7.3.3 数据类型的判断	180
6.3.4 裁剪属性 clip	152	7.3.4 数据类型的转换	181
6.3.5 内容溢出时的显示方式属性 overflow	153	7.4 标识符、变量、运算符和表达式	183
6.3.6 元素显示方式属性 display	154	7.4.1 标识符	183
6.3.7 元素可见性属性 visibility	155	7.4.2 常量	184
6.4 CSS 盒子定位属性	156	7.4.3 变量	184
6.4.1 定位位置属性 top、right、 bottom、left	156	7.4.4 运算符和表达式	185
6.4.2 定位方式属性 position	156	7.4.5 语句的书写规则	187
6.4.3 层叠顺序属性 z-index	161	7.5 流程控制语句	188
6.5 CSS3 多列属性	162	7.5.1 顺序结构语句	188
6.5.1 列数属性 column-count	162	7.5.2 条件选择结构语句	191
		7.5.3 循环结构语句	193
		7.6 函数	195

7.6.1 函数的声明	195	8.2.4 HTML DOM 对象	243
7.6.2 函数的调用	198	8.2.5 HTML Document 对象	244
7.6.3 变量的作用域	200	8.2.6 HTML Element 对象	245
7.6.4 内嵌函数	201	8.2.7 Node 操作实例	246
7.6.5 闭包函数	201	习题 8	251
7.6.6 系统函数	203	第 9 章 JavaScript 事件处理	253
7.7 对象	203	9.1 事件概述	253
7.7.1 对象的概念	203	9.1.1 事件的概念	253
7.7.2 创建类	204	9.1.2 事件的类型	255
7.7.3 对象的实例化	204	9.1.3 事件处理程序的绑定方式	256
7.7.4 对象的属性	205	9.2 window 事件	259
7.7.5 对象的方法	207	9.2.1 load 事件	260
7.7.6 对象的遍历	208	9.2.2 resize 事件	261
7.7.7 对象的事件	209	9.2.3 scroll 事件	262
7.8 内置对象	209	9.2.4 focus 和 blur 事件	262
7.8.1 数学对象	209	9.3 mouse 事件	263
7.8.2 字符串对象	210	9.3.1 click 事件	264
7.8.3 日期对象	212	9.3.2 dblclick 事件	264
7.8.4 数组对象	214	9.3.3 mouseover 和 mouseout 事件	265
7.9 正则表达式	220	9.3.4 mousedown、mousemove 和 mouseup 事件	266
7.9.1 创建正则表达式	220	9.4 keyboard 事件	268
7.9.2 正则表达式的组成	221	9.4.1 keydown 事件	268
7.9.3 正则表达式使用的方法	223	9.4.2 keypress 事件	269
7.10 使用开发者工具调试 JavaScript 程序	223	9.4.3 keyup 事件	269
7.10.1 开发者工具调试使用方法	224	9.5 form 事件	270
7.10.2 调试示例	230	9.5.1 onsubmit 和 onreset 事件	271
习题 7	232	9.5.2 子元素事件	271
第 8 章 JavaScript 对象模型	234	9.6 事件捕捉与事件冒泡	273
8.1 BOM 的对象	234	习题 9	274
8.1.1 BOM 概述	234	第 10 章 CSS3 变形、过渡和动画属性	276
8.1.2 window 对象	235	10.1 变形	276
8.1.3 document 对象	236	10.1.1 CSS 的坐标系统	276
8.1.4 location 对象	238	10.1.2 transform 属性	277
8.1.5 navigator 对象	238	10.1.3 transform-origin 属性	281
8.1.6 screen 对象	239	10.1.4 transform-style 属性	283
8.1.7 history 对象	240	10.1.5 perspective 和 perspective-origin 属性	284
8.2 DOM 的对象	240	10.1.6 backface-visibility 属性	285
8.2.1 节点和节点树	240	10.2 过渡	285
8.2.2 DOM 的操作	241	10.2.1 过渡属性	285
8.2.3 Node 对象	242		

10.2.2 过渡事件	287	11.2.3 绘制图形的步骤	299
10.3 动画	288	11.2.4 绘制图形	300
10.3.1 动画属性	288	习题 11	307
10.3.2 动画事件	291	第 12 章 综合案例——社区新闻网的	
习题 10	293	设计与实现	309
第 11 章 HTML5 的拖放和画布	295	12.1 网站的开发流程和组织结构	309
11.1 拖放	295	12.1.1 创建站点目录	309
11.1.1 draggable 属性	295	12.1.2 网站页面的组成	309
11.1.2 拖放事件	295	12.2 制作社区新闻网首页	309
11.1.3 数据传递对象 dataTransfer	296	12.3 制作社区新闻网的列表页	312
11.2 画布	298	12.4 制作社区新闻网的内容页	313
11.2.1 创建 canvas 元素	298	习题 12	315
11.2.2 构建绘图环境	298	参考文献	316

第1章 HTML5 概述

HTML（Hyper Text Markup Language，超文本标记语言）是制作网页的基础语言，是初学者必学的内容。在学习 HTML 之前，需要了解一些与 Web 相关的基础知识，有助于初学者学习后面章节的相关内容。本章将对网页的基础知识、Web 标准、编写语言、HTML5 的运行环境和创建方法进行详细讲解。

通过对本章的学习，应掌握 Web 相关的概念、HTML5 的基本结构和语法规则，熟练使用记事本编辑网页。

1.1 Web 的基本概念

对于网页设计开发者，在动手制作网页之前，应该先了解 Web 的基础知识。

1.1.1 WWW

WWW 是 World Wide Web 的缩写，又称 3W 或 Web，中文译名为“万维网”。WWW 是 Internet 的最核心部分，它是 Internet 上支持 WWW 服务和 HTTP 的服务器集合。WWW 在使用上分为 Web 客户端和 Web 服务器。用户可以使用 Web 客户端（浏览器）访问 Web 服务器上的页面。

1.1.2 Web 服务器

Web 服务器也称为 WWW（World Wide Web）服务器，一般指网站服务器。WWW 是 Internet 的多媒体信息查询工具，是 Internet 上发展最快和目前用得最广泛的服务。正是因为 WWW 工具，近年来 Internet 迅速发展，且用户数量飞速增长。

Web 服务器的主要功能是提供网上信息浏览服务。Web 服务器可以解析 HTTP，当 Web 服务器接收到一个 HTTP 请求时，会返回一个 HTTP 响应，这样浏览器等 Web 客户端就可以从服务器上获取网页（HTML），包括 CSS、JS、音频、视频等资源。

1.1.3 网页浏览器

网页浏览器（Web Browser）是在客户端浏览 Web 服务端的应用程序，其主要作用是显示网页和解释脚本。通过浏览器可以访问互联网上世界各地的文档、图片、视频等信息，并让用户与这些文件互动。浏览器的种类很多，目前常用的有 Google 的 Chrome、Microsoft 的 Edge、Mozilla 的 Firefox、Opera、Apple 的 Safari 浏览器等。

浏览器最重要的核心部分是 Rendering Engine（渲染引擎），一般称为“浏览器内核”，负责对网页语法（如 HTML、JavaScript）进行解释并渲染（显示）网页。不同的浏览器内核对网页编写语法的解释会有所不同，因此同一网页在不同内核的浏览器里的渲染效果也可能不同，这正是网页编写者需要在不同内核的浏览器中测试网页显示效果的原因。现在主流浏览器采用的内核见表 1-1。

表 1-1 主流浏览器采用的内核

浏览器名称	内核	其他采用相同内核的浏览器
IE	Trident（IE 内核）	
Google Chrome	之前是 WebKit，2013 年后换成 Blink（Chromium、谷歌内核）	Edge、Opera、360、UC、百度、搜狗高速、傲游 3、猎豹、微信、世界之窗等

(续)

浏览器名称	内核	其他采用相同内核的浏览器
Safari	WebKit	
Firefox	Gecko	
Microsoft Edge	之前为 EdgeHTML，2018 年 12 月后换成 Blink (Chromium)	

1.1.4 网站

网站 (Website) 是指在因特网上根据一定的规则，使用 HTML 等工具制作的用于展示特定内容相关网页的集合。简单地说，网站是一种沟通工具，人们可以通过网站来发布自己想要公开的信息，或者利用网站来提供相关的网络服务。人们可以通过网页浏览器来访问网站，获取自己需要的信息或者享受网络服务。

网站是在互联网上拥有域名或地址并提供一定网络服务的主机，是存储文件的空间，以服务器为载体。人们可通过浏览器等进行访问、查找文件，也可通过远程文件传输 (FTP) 方式上传、下载网站文件。

1.1.5 网页

网页 (Web Page) 是一个包含 HTML 标签的纯文本文件 (文件扩展名为.html 或.htm)。网页是网站中的一“页”，网页是构成网站的基本元素。换句话说，网站就是由网页组成的。网页要通过网页浏览器来阅读。本书就是介绍网页制作的教材。

1.1.6 URL

URL (Uniform Resource Locator，统一资源定位器) 就是 Web 地址，俗称“网址”。Internet 上的每一个网页都具有一个唯一的名称标识，通常称之为 URL 地址。它是对从互联网上得到资源的位置和访问方法的一种简洁的表示，是互联网上标准资源的地址。在 WWW 上浏览，必须在网页浏览器中输入目标的地址。这种地址可以是本地磁盘，也可以是局域网上的某一台计算机，更多的是 Internet 上的站点。URL 的一般格式如下：

协议：//主机地址 (IP 地址) /文件夹/.../文件名/参数

协议是指 URL 所链接的网络服务性质，常见的协议有 HTTP (Hyper Text Transfer Protocol，超文本传输协议)、FTP (File Transfer Protocol，文件传输协议)、TELNET (Telecom Munication Network Protocol，远程登录协议)、File (File Protocol，本地文件传输协议)。

URL 的参数通常放在 URL 后面，用“?”开头，用“&”将多个参数连接起来。例如，https://www.baidu.com/s?wd=%E5%A5%B4&rsv_spt=1&rsv_iqid=0 中“?”后面的字符是参数。

URL 只能用 ASCII 字符编码集中的可显示字符表示。如果包含非 ASCII 字符集的字符，则需要转换。例如，“OK 好的”会转成“OK%E5%A5%BD%E7%9A%84”。URL 中不能包含空格，如要用空格，URL 编码通常用“+”代替空格，例如“OK+好的”。

1.1.7 标记语言

标记语言是一种将文本 (Text) 以及与文本相关的其他信息结合起来，展现出关于文档结构和数据处理细节的计算机文字编码。标记语言的种类有很多，常见的有 XML、HTML、XHTML 等。

1.1.8 网页标准

Web 应用开发需要遵循的标准就是网页标准 (Web Standard)。网页标准不是某一种标准，而是一系列标准的集合。网页标准主要分为 3 类：结构 (Structure) 标准、表现 (Presentation) 标准和行

为（Behavior）标准。其中，结构标准语言主要包括 XML、HTML 和 XHTML，表现标准语言主要为 CSS，行为标准主要包括对象模型 DOM、ECMAScript 等。这些标准大部分由 W3C 起草和发布，也有一些是其他标准组织制定的，如 ECMA（European Computer Manufacturers Association，欧洲计算机制造联合会）的 ECMAScript 标准。

1.2 HTML5 的基本结构和语法规则

每个网页都有其基本的结构，包括 HTML 的语法结构、文档结构、标签的格式以及代码的编写规范等。

1.2.1 HTML5 文档的基本结构

HTML 5 文档的基本结构如下：

```
<!DOCTYPE html>
<html lang="zh-CN">
  <head>
    <meta charset="UTF-8">
    <title>文档标题</title>
  </head>
  <body>
    文档正文部分
  </body>
</html>
```

HTML 文档可分为文档头（Head）和文档体（Body）两部分。文档头的内容包括网页语言、关键字和字符集的定义等；文档体中的内容就是页面要显示的信息。

HTML 文档的基本结构由 3 对标签负责组织，即<html>、<head>和<body>。其中<html>标签标识 HTML 文档，<head>标签标识头部区域，<body>标签标识主体区域。

图 1-1 所示是一个可视化的 HTML 页面结构，只有<body>与</body>之间的白色区域才会在浏览器中显示。

1. <!DOCTYPE html>标签

<!DOCTYPE>标签位于文档的最前面，用于向浏览器说明当前文档使用哪种 HTML 标准规范。只有开头处使用<!DOCTYPE>声明，浏览器才能将该页面作为有效的 HTML 文档，并按指定的文档类型进行解析。文档类型声明的语法格式如下：

```
<!DOCTYPE html>
```

这行代码称为 DOCTYPE（DOCument Type，文档类型）声明。要建立符合标准的网页，DOCTYPE 声明是必不可少的重要组成部分。<!DOCTYPE html>声明必须放在每一个 HTML 文档的最顶部，在所有代码和标签之前。

2. <html>...</html>标签

<html>标签位于<!DOCTYPE html>标签之后，称为 HTML 文档标签，也被称为根标签，用于告诉浏览器其自身是一个 HTML 文档。HTML 文档标签的语法格式为：

```
<html lang="zh-CN">
  HTML 文档的内容
</html>
```

<html>处于文档的最前面，表示 HTML 文档的开始，即浏览器从<html>开始解释，直到遇到



1 HTML5 的基本结构和语法规则

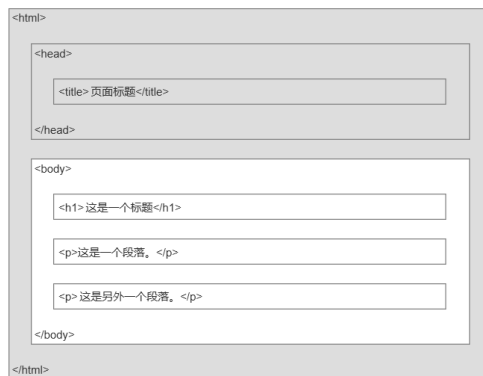


图 1-1 可视化的 HTML 页面结构

</html>为止。每个 HTML 文档均以<html>开始，以</html>结束。lang 属性为文档设置语言，对于简体中文，设置为“zh-CN”。

3. <head>...</head>标签

HTML 文档包括文档头和文档体。<head>标签用于定义 HTML 文档的头部信息，也称为头部标签，紧跟在<html>标签之后，主要用来封装其他位于文档头部的标签。HTML 文档头标签的语法格式为：

```
<head>
    头部内容
</head>
```

文档头部内容在开始标签<html>和结束标签</html>之间定义，一个 HTML 文档只能含有一对<head>...</head>标签。网页中经常设置页面的基本信息，如页面的标题、作者、和其他文档的关系等。为此 HTML 提供了一系列的标签，这些标签通常都写在<head>标签内，因此被称为头部相关标签。绝大多数文档头部包含的数据都不会真正作为内容显示在页面中。

4. <meta charset>标签

<head>...</head>标签中的<meta charset>指定网页文档中的字符集，称为 HTML 文档编码，HTML5 文档直接使用 meta 元素的 charset 属性指定文档编码，语法格式如下：

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

为了被浏览器正确解释和通过 W3C 代码校验，所有的 HTML 文档都必须声明所使用的编码语言。文档声明的编码应该与实际的编码一致，否则就会呈现为乱码。对于中文网页的设计者来说，指定代码的字符集为“UTF-8”。

5. <title>...</title>标签

HTML 文件的标题显示在浏览器的标题栏中，用以说明文件的用途。标题文字位于<title>和</title>标签之间，其语法格式如下：

```
<title>网页标题</title>
```

每个 HTML 文档都应该有标题，在 HTML 文档中，<title>和</title>标签位于 HTML 文档的头部，即<head>和</head>标签之间。例如，<title>哔哩哔哩(゜-゜)つロ干杯~bilibili</title>，如图 1-2 所示（在 Google Chrome 浏览器中，单击地址栏右端的“更多”按钮，在弹出的下拉列表中选择“更多工具”→“开发者工具”，单击“Elements”标签，再单击▶<head>...</head>元素前的箭头，展开该元素）。



图 1-2 <title>...</title>标签在浏览器中的显示

6. <body>...</body>标签

<body>标签定义 HTML 文档要显示的内容，也称为主体标签。浏览器中显示的所有文本、图像、表单与多媒体元素等信息都必须位于<body>...</body>标签内，<body>标签内的信息才是最终展示给用户看的。HTML 文档主体标签的语法格式为：

```
<body>
    网页内容
</body>
```

文档体位于文档头之后，以<body>为开始标签，</body>为结束标签。一个 HTML 文档只能含有一对<body>...</body>标签，且<body>...</body>标签必须在<html>...</html>标签内，位于<head>头部标签之后，与<head>标签是并列关系。<body>标签定义网页上显示的主要内容与显示格式，是整个网页的核心，网页中要真正显示的内容都包含在主体中。

浏览器在解释 HTML 文档时是按照层次顺序进行解释的，其顺序为：document→html→body→div 父元素→input 子元素。document 是最上层祖先元素，input 是最下层后代元素。

1.2.2 HTML5 的基本语法

HTML 文档由元素构成，元素由开始标签、结束标签、属性及元素的内容 4 部分组成。

1. 标签 (Tag)

HTML 用标签来规定网页元素在文档中的功能。标签是用一对尖括号“<>”括起来的单词或单词缩写。标签有两种形式：双标签和单标签。

(1) 双标签

双标签包括开始标签和结束标签，其格式为：

<标签>受标签影响的内容</标签>

开始标签标志一段内容的开始，结束标签是指与开始标签相对的标签。结束标签比开始标签多一个斜杠“/”。双标签也称闭合标签。

例如，HTML 文档从<html>开始，到</html>结束。<head>和</head>标签描述 HTML 文档的相关信息，之间的内容不会在浏览器窗口上显示出来。<body>和</body>标签包含所有要在浏览器窗口上显示的内容，也就是 HTML 文档的主体部分。

(2) 单标签

单标签没有相应的结束标签的标签，也称空标签。其格式为：

<标签> 或 <标签 />

例如，换行标签
或
。

其他没有相应结束标签的标签有<area>、<base>、<basefont>、
、<col>、<hr>、、<input>、<param>、<link>、<meta>等。

(3) 标签的嵌套

标签可以放在另外一个标签所能影响的片段中，以实现某一段文档的多重标签效果，但是要注意必须要正确嵌套。例如，下面的嵌套是错误的：

<p>Hello Word!</p>

改正如下：

<p>Hello World!</p>

需要注意以下两点。

1) 每个标签都要用一对尖括号“< >”括起来，如<p>、<table>，以表示这是 HTML 代码而非普通文本。注意，“<”“>”与标签名之间不能留有空格或其他字符。

2) 在标签名前加上符号“/”便是其结束标签，表示该标签内容的结束，如</h1>。标签也有不用</标签>结尾的，称之为单标签。例如，换行标签
。

2. 元素 (Element)

HTML 文档中的元素是指从开始标签到结束标签的所有代码。HTML 元素分为有内容的元素和空元素两种。

(1) 有内容的元素

有内容的元素是由开始标签、结束标签以及两者之间的元素内容组成的，其中元素内容既可以

是需要显示在网页中的文字内容，也可以是其他元素。例如，<title>和</title>是标签，下面的代码是一个 title 元素：

```
<title>淘宝网 - 淘！我喜欢</title>
```

(2) 空元素

空元素只有开始标签而没有结束标签，也没有元素内容。例如，br、hr（横线）元素就是空元素。

(3) 元素的嵌套

除了 HTML 文档元素<html>外，其他的 HTML 元素都是被嵌套在另一个元素之内的。在 HTML 文档中，html 是最外层元素，也称为根元素。head、body 元素是嵌套在 html 元素内的。body 元素内又嵌套许多元素。HTML 中的元素可以多级嵌套，但是不能互相交叉。例如，下面的代码对于<head>和</head>标签来说，是一个 head 元素：

```
<head><title>淘宝网 - 淘！我喜欢</title></head>
```

同时，这个 title 元素又是嵌套在 head 元素中的另一个元素。

例如，下面是不正确的嵌套写法，p 元素的开始标签在 b 元素的外层，但它的结束标签却放在了 b 元素结束标签内。

```
<p>这是<b>第一段</p>文字</b>
```

正确的 HTML 写法如下：

```
<p>这是<b>第一段</b>文字</p>
```

为了防止出现错误的 HTML 元素嵌套，在编写 HTML 文档时，建议先写外层的一对标签，然后逐渐往里写，这样既不容易忘记写 HTML 元素的结束标签，也可以减少 HTML 元素的嵌套错误。例如，在编写 HTML 文档时，可以像下面这样写。

第 1 步：

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

第 2 步：

```
<html>
  <head>
  </head>
  <body>
  </body>
</html>
```

第 3 步：

```
<html>
  <head>
    <title></title>
  </head>
  <body>
  </body>
</html>
```

3. 属性

属性用来说明元素的特征，借助于元素属性，HTML 网页才会展现丰富多彩且格式美观的内容。

元素的属性放置在元素的开始标签内，每个属性对应一个属性值，通常以“属性名=“值””的形式来表示，出现在元素开始标签的“>”之前，用空格隔开，可以指定多个属性，并且在指定多个属性时不用区分顺序。属性的使用格式如下：

<标签 属性 1="属性值 1" 属性 2="属性值 2" ...>受标签影响的内容</标签>

例如，下面代码中的“style="color:#ff0000;font-size:30px"”就是 p 元素的属性：

```
<p style="color:#ff0000;font-size:30px">第一段内容</p>
```

定义属性值时注意以下几点。

1) 不定义属性值。HTML 规定属性也可以没有值，例如：

```
<dl compact>
```

浏览器会使用 compact 属性的默认值。但对于没有默认值的属性，不能省略属性值。

2) 属性中的属性值可以包含空格，但是这种情况下必须使用引号。例如，下面的代码正确定义了方法：

```

```

下面的代码则是错误的：

```
<img src=C:/Documents and Settings/test.jpg width=1024 height=36/>
```

也就是说，属性值一定是连续字符序列，如果不是连续序列，则要加引号标注。

3) 单引号和双引号都可以作为属性值。当属性值中含有单引号时，不能再单引号来包括属性值，要用双引号来包括属性值。但是，当属性值中有双引号时，属性值中的双引号就要用数字字符引用（'）或者字符实体引用（"）来代替双引号。例如，下面的代码是错误的：

```
<p title="欢迎游览"迪士尼"">乐园</p>
```

4) HTML 要求属性和属性值使用小写，虽然属性和属性值对大小写不敏感。

1.2.3 HTML 的全局属性

1. HTML 的全局标准属性

全局属性是指可用于大多数 HTML 元素的属性。在 HTML 和 HTML5 规范中规定的全局标准属性，见表 1-2。后续章节将介绍这些属性。

表 1-2 HTML 和 HTML5 的全局标准属性

属性	描述
accesskey	指定激活某个元素的快捷键。支持本属性的元素有 a、area、button、input、label、legend、textarea
class	指定元素的类名。通常用于引用 CSS 样式表中的类，本属性通常写在<body>...</body>元素中。本属性不能用于的元素有 base、head、html、meta、param、script、style、title
id	指定元素的唯一 ID。本属性值在整个 HTML 文档中要唯一，本属性的主要作用是通过 JavaScript 和 CSS 为指定的 ID 改变或添加样式、动作等
style	指定元素的行内 CSS 样式。使用本属性后将覆盖任何全局的样式设定
title	指定元素的额外信息。通常在鼠标指针移到元素上时显示定义的提示文本
dir	指定元素中内容的文本方向。本属性只有 ltr 和 rtl 两种，含义分别是 left to right 和 right to left。本属性对大部分有文本内容的元素有效，无效的元素有 base、br、frame、frameset、hr、iframe、param、script
lang	指定元素内容的语言。与 dir 属性一样，本属性对大部分有文本内容的元素有效，无效的元素有 base、br、frame、frameset、hr、iframe、param、script
tabindex	指定元素的〈Tab〉键次序。支持本属性的元素有 a、area、button、input、object、select、textarea
contenteditable	规定元素内容是否可编辑
designMode	相当于一个全局的 contenteditable 属性。如果将 designMode 属性设为 on，则页面上所有支持 contenteditable 属性的元素都变成可编辑状态。designMode 属性默认为 off。designMode 属性只有用 JavaScript 修改
draggable	规定元素是否可拖动
dropzone	规定在拖动被拖动数据时是否进行复制、移动或链接
hidden	属性值为 true 时显示，为 false 时隐藏
spellcheck	属性值为 true 时对元素进行拼写和语法检查，默认为 false
translate	规定是否翻译元素内容

2. HTML 的全局事件属性

事件是针对某个元素的可识别的动作。例如，针对“确定”按钮的单击事件，文本框内容变化事件、复选框的选中或取消选中事件等。

HTML 有使事件在浏览器中触发动作的能力，例如，当用户单击某个元素时执行 JavaScript 程序。在 HTML 中，事件既可以通过 JavaScript 直接触发，也可以通过全局事件属性触发。所谓全局事件属性是指可用于大多数 HTML 元素的事件属性。有关事件编程的知识，将在第 9 章 JavaScript 事件处理中介绍。

1.2.4 元素的分类

依据元素的作用不同，可将元素分为元信息元素、语义元素和无语义元素。

1. 元信息元素

元信息（Meta-information）或称元数据（Metadata）类元素是指用于描述文档自身信息的一类元素。`meta` 元素定义元信息，包含页面的描述、关键字、最后的修改日期、作者及其他元信息。`<meta>` 标签写在`<head>...</head>` 标签中。元信息元素提供给浏览器、搜索引擎（关键字）以及其他 Web 服务调用，一般不会显示给用户。对于样式和脚本的元数据，可以直接在网页里定义，也可以链接到包含相关信息的外部文件。

meta 元素是元信息元素，在 HTML 中是一个单标签的空元素。该元素可重复出现在头部元素中，用来指明本页的作者、制作工具、所包含的关键字，以及其他一些描述网页的信息。

meta 元素的常用属性如下。

- 1) **charset**: 定义文档的字符编码, 常用的是 UTF-8。
- 2) **content**: 定义与 **name** 和 **http-equiv** 相关的元信息。
- 3) **name**: 关联 **content** 的名称 (常用的有 **keywords** 关键字、**author** 作者名、**description** 页面描述)。

不同的属性又有不同的参数值，这些不同的参数值就实现了不同的网页功能。本节主要介绍 `name` 属性，用于设置搜索关键字和描述。`meta` 元素的 `name` 属性的语法格式为：

<meta name="参数" content="参数值">

name 属性主要用于描述网页摘要信息，与之对应的属性值为 content。content 中的内容主要用于搜索引擎查找信息和分类信息。

name 属性主要有以下两个参数：keywords 和 description。其中，keywords 用来告诉搜索引擎网页使用的关键字；description 用来告诉搜索引擎网站主要的内容。

例如，哔哩哔哩网站主页的关键字信息如图 1-3 所示。打开方法是单击 Google Chrome 浏览器地址栏右端的“更多”按钮，选择“更多工具”→“开发者工具”选项。选择“Elements”选项卡，再单击▶<head>...</head>选项卡前的箭头展开该元素。当浏览者通过百度搜索引擎搜索“哔哩哔哩”时，就可以看到搜索结果中显示出网站主页的标题、关键字和内容描述，如图 1-4 所示。

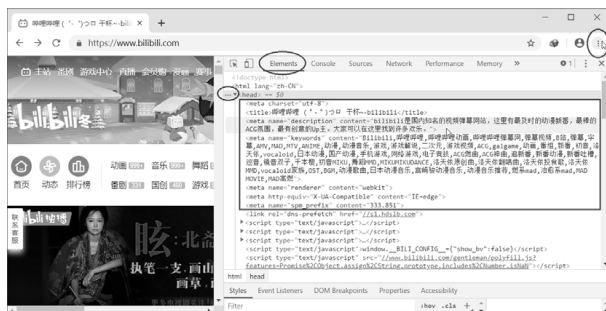


图 1-3 关键字信息



图 1-4 摘要信息

2. 语义元素

语义元素是指清楚地向浏览器和开发者描述其意义的元素，如标题元素、段落元素、列表元素等。有些语义元素在网页中可以呈现显示效果，有些没有显示效果。

元素的语义化能够呈现出很好的内容结构，语义化使得代码更具有可读性，让其他开发人员更加理解 HTML 结构，减少差异化；方便其他设备解析，如屏幕阅读器、盲人阅读器、移动设备等，以有意义的方式来渲染网页。爬虫依赖标签来确定关键字的权重，帮助爬虫抓取更多的有效信息。

大约有 100 多个 HTML 语义元素可供选择。语义元素分为块级元素、行内（内联）元素、行内块元素等。

（1）块级元素（block）

块级元素是指本身属性为 `display:block` 的元素。因为它自身的特点，通常使用块级元素进行大布局（大结构）的搭建。块级元素的特性如下。

- 1) 每个块级元素总是独占一行，表现为另起一行开始，而且其后的元素也必须另起一行显示。
- 2) 块级元素可以直接控制宽度（width）、高度（height）以及盒子模型的相关 CSS 属性，内边距（padding）和外边距（margin）等都可控制。
- 3) 在不设置宽度的情况下，块级元素的宽度是它父级元素内容的宽度。
- 4) 在不设置高度的情况下，块级元素的高度是它本身内容的高度。

常用的块级元素主要有 `p`、`div`、`ul`、`ol`、`li`、`dl`、`dt`、`dd`、`h1~h6`、`hr`、`form`、`address`、`pre`、`table`、`blockquote`、`center`、`dir`、`fieldset`、`isindex`、`menu`、`noframes`、`noscript` 等。

（2）行内元素（inline）

行内元素也称内联元素，是指本身属性为 `display:inline` 的元素，行内元素可以和相邻的行内元素在同一行，对宽、高属性值不生效，完全靠内容撑开宽、高。因为它自身的特点，通常使用块级元素来进行文字、小图标（小结构）的搭建。行内元素的特性如下。

- 1) 行内元素会与其他行内元素从左到右在一行显示。
- 2) 行内元素不能直接控制宽度、高度以及盒子模型的相关 CSS 属性，例如内边距的 `top`、`bottom`（`padding-top`、`padding-bottom`）和外边距的 `top`、`bottom`（`margin-top`、`margin-bottom`）都不可改变。但是可以设置内外边距水平方向的值。也就是说，对于行内元素的 `margin` 和 `padding`，只有 `margin-left`/`margin-right` 和 `padding-left`/`padding-right` 是有效的，但是竖直方向的 `margin` 和 `padding` 无效。
- 3) 行内元素的宽高是由本身内容的大小决定（文字、图片等）。
- 4) 行内元素只能容纳文本或者其他行内元素（不能在行内元素中嵌套块级元素）。

常用的行内元素主要有 `a`、`span`、`em`、`strong`、`b`、`i`、`u`、`label`、`br`、`abbr`、`acronym`、`bdo`、`big`、`br`、`cite`、`code`、`dfn`、`em`、`font`、`img`、`input`、`kbd`、`label`、`q`、`s`、`samp`、`select`、`small`、`span`、`strike`、`strong`、`sub`、`sup`、`textarea`、`tt`、`var` 等。

利用 CSS 可以摆脱上面 HTML 标签归类的限制，自由地在不同标签或元素上应用需要的属性。常用的 CSS 样式有以下三个。

`display: block`: 显示为块级元素。

`display: inline`: 显示为行内元素。

`display: inline-block`: 显示为行内块元素。表现为同行显示并可修改宽、高、内外边距等属性。例如，将 `<ul>` 元素加上 `display: inline-block` 样式，原本垂直的列表就可以水平显示了。

（3）行内块元素

还有一种元素结合行内元素和块级元素，不仅可以对宽和高的属性值生效，还可以多个元素在一行显示，这种元素被称为行内块元素。行内块元素能和其他元素待在一行，能设置宽和高。常用的行内块元素有 `img`、`input`、`textarea` 等。行内块元素的特点是结合行内元素和块级元素的优点，不仅可以对宽和高的属性值生效，还可以多个标签在一行显示。

块级元素可以嵌套行内元素，行内元素不可以嵌套块级元素。

(4) 可变元素

可变元素根据上下文关系确定该元素是块级元素还是行内元素。常用的可变元素有 `applet`、`button`、`del`、`iframe`、`ins`、`map`、`object`、`script` 等。

(5) HTML5 中新增的结构语义元素

在 HTML5 之前，页面只能用 `div` 元素作为结构元素来分隔不同的区域，由于 `div` 元素无任何语义，给代码设计者和阅读者带来困扰，所以在 HTML5 中增加了结构语义元素。HTML5 增加的结构语义元素明确了一个 Web 页面的不同部分，如图 1-5 所示。

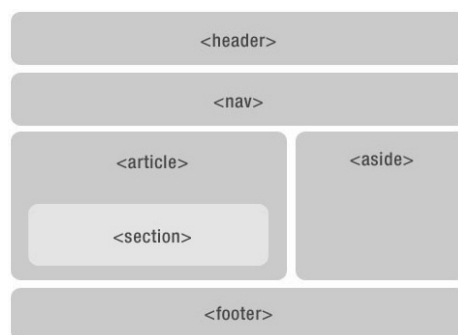


图 1-5 结构语义元素

HTML5 常用的结构语义元素如下。

- 1) `header` 元素用于定义文档的头部区域，为文档或节规定页眉，常被用作介绍性内容的容器，可以包含标题元素、Logo、搜索框等。一个文档中可以有多个 `header` 元素。
 - 2) `nav` 元素用于定义页面的导航链接部分区域，导航有顶部导航、底部导航和侧边导航等。
 - 3) `article` 元素用于定义文档内独立的文章，可以是新闻、条件、用户评论等。
 - 4) `section` 元素用于定义文档中的一个区域或节。节是指有主题的内容组，通常有标题。可以将网站首页划分为简介、内容、联系信息等节。
 - 5) `aside` 元素用于定义页面主区域内容之外的内容（比如侧边栏）。`<aside>` 标签的内容是独立的，与主区域的内容无关。
 - 6) `footer` 元素用于定义文档的底部区域。一个页脚通常包含文档的作者、著作权信息、链接的使用条款链接、联系信息等，文档中可以使用多个 `footer` 元素。
 - 7) `figure` 元素用于定义一段独立的引用内容，经常与 `figcaption` 元素配合使用，通常用在主文本的图片、代码、表格等中。如果这部分内容被转移或删除，不会影响到主体。
 - 8) `figcaption` 元素用于表示与其相关联的引用的说明或标题，描述其父节点 `figure` 元素中的其他内容。`figcaption` 元素应该被置于 `figure` 元素的第一个或最后一个子元素的位置。
 - 9) `main` 元素用于规定文档的主内容。
 - 10) `mark` 元素用于定义重要的或强调的文本。
 - 11) `details` 元素用于定义用户能够查看或隐藏的额外细节。
 - 12) `summary` 元素用于定义 `details` 元素的可见标题。
 - 13) `time` 元素用于定义日期或时间。
- 以上的元素除了 `figcaption` 外，都是块级元素。

3. 无语义元素

无语义元素无须考虑其内容，有 `div` 和 `span` 两个无语义元素。`div` 是块级元素，`span` 是行内元素。

常用 `div` 元素划分区域或者节，`div` 元素可以用作组织工具，而不使用任何格式。所谓 Div+CSS 的网页布局，就是用 `div` 元素组织要显示的数据（文字、图、表等）结构，用 CSS 显示数据的样式，从而做到结构与样式的分离，这种布局代码简单，易于维护。

1.2.5 HTML 的字符实体和颜色表示

1. 字符实体

一些字符在 HTML 中拥有特殊的含义，例如，尖括号 “>” “<” 已作为 HTML 的语法符号。因此，如果希望在浏览器显示这些特殊字符，就需要在 HTML 源代码中插入相应的 HTML 代码，这些特殊符号对应的 HTML 代码被称为字符实体。

字符实体由三部分组成：以一个符号（&）开头，中间是一个实体名称，以一个分号（;）结束。例如，要在 HTML 文档中显示小于号，输入“<”。需要强调的是，实体书写对大小写是敏感的。常用的特殊符号及对应的字符实体见表 1-3。

表 1-3 常用的特殊符号及对应的字符实体

特殊符号的描述	字符实体	显示结果	示 例
空格	 		公司 咨询热线：400-810-6666
大于号（>）	>	>	3>2
小于号（<）	<	<	2<3
双引号（"）	"	"	HTML 属性值必须使用成对的"括起来
单引号（'）	'	'	She said 'hello';
和号（&）	&	&	a & b
版权号（©）	©	©	Copyright ©
注册商标号（®）	®	®	鑫牌®
节号（§）	§	§	§1.1
乘号（×）	×	×	10 × 20
除号（÷）	÷	÷	10 ÷ 2

空格是 HTML 中最常用的字符实体。通常情况下，在 HTML 源代码中，如果通过按〈Space〉键输入了多个连续空格，浏览器会只保留一个空格，删除其他空格。在需要添加多个空格的位置，使用多个“ ”就可以在文档中增加空格。

2. HTML 的颜色表示

在 HTML 中，颜色有两种表示方式。一种是用颜色的英文名称表示，比如 blue 表示蓝色，red 表示红色；另外一种是用十六进制的数值表示 RGB 的颜色值。

RGB 颜色的表示方式为#rrggbb。其中，rr、gg、bb 三色对应的取值范围都是 00~FF，如白色的 RGB 值是（255，255，255），用#ffffff 表示；黑色的 RGB 值是（0，0，0），用#000000 表示。

1.2.6 HTML5 开发人员编码规范

HTML5 作为前端网页结构的超文本标记语言，网页的 HTML 代码必须符合 HTML5 书写规范。规范目的是提高团队协作效率，使 HTML5 代码风格保持一致，容易被理解、维护和升级。

1. HTML 书写规范

- 1) 文档第一行添加 HTML5 的声明类型<!DOCTYPE html>。
- 2) 建议为<html>根标签指定 lang 属性，从而为文档设置正确的语言 lang="zh-CN"。
- 3) 编码统一为<meta charset="utf-8"/>。
- 4) <title>标签必须设置为<head>的直接子元素，并紧随<meta charset>声明之后。
- 5) 文档中除了开头的 DOCTYPE、utf-8（或 UTF-8）和 zh-CN 或者<head>几种特殊情况可以大写外，其他 HTML 标签名必须使用小写字母。
- 6) 标签的闭合要符合 HTML5 的规定。
- 7) 必须遵守标签的嵌套规则，例如，div 不得置于 p 中，tbody 必须置于 table 中。
- 8) 属性名必须使用小写字母，其属性值必须用双引号包围。布尔类型的属性建议不添加属性值。自定义属性推荐使用 data-。

2. 标签的规范

- 1) 标签分单标签和双标签，双标签往往是成对出现，所有标签（包括空标签）都必须关闭，如
、、<p>...</p>等。
- 2) 标签名和属性建议都用小写字母。

- 3) 多数 HTML 标签可以嵌套, 但不允许交叉。
- 4) HTML 文档中一行可以写多个标签, 但标签中的一个单词不能分两行写。

3. 属性的规范

1) 根据需要可以使用该标签的所有属性, 也可以只用其中的几个属性。在使用时, 属性之间没有顺序。

- 2) 属性值都要用双引号括起来。
- 3) 并不是所有的标签都有属性, 如换行标签就没有。

4. 元素的嵌套

1) 块级元素可以包含行内元素或其他块级元素, 但行内元素却不能包含块级元素, 它只能包含其他的行内元素。

2) 有几个特殊的块级元素只能包含内行元素, 不能再包含块级元素, 这几个特殊的块级元素是 h1、h2、h3、h4、h5、h6、p、dt。

5. 代码的缩进

HTML5 代码并不要求在书写时缩进, 但为了文档的结构性和层次性, 建议代码缩进设置为 4 个空格, 即使用 4 个空格作为一个缩进层级, 标签首尾对齐, 每层的内容向右缩进 4 个空格。

1.3 编辑 HTML 文件

1.3.1 常用 HTML 编辑软件

网站制作及前端开发软件是指用于制作 HTML 网页的工具软件。

1. Dreamweaver

Dreamweaver 是美国 Adobe 公司推出的一套拥有可视化编辑界面, 用于制作并编辑网站和移动应用程序的网页设计软件。由于 Dreamweaver 支持代码、拆分、设计、实时视图等多种方式来创作、编写和修改网页, 对于初级人员, 无须编写任何代码就能快速创建 Web 页面。其成熟的代码编辑工具更适用于 Web 开发高级人员的创作, 新版本使用了自适应网格版面创建页面, 在发布前使用多屏幕预览审阅设计, 可大大提高工作效率。所以, Dreamweaver 是一款比较好的 HTML 代码编辑器。

2. Visual Studio Code

Visual Studio Code (简称 VS Code) 是 Microsoft 公司开发的运行于 Windows、Mac OS X 和 Linux 系统之上的, 开源、免费、跨平台、高性能、轻量级的代码编辑器。它在性能、语言支持、开源社区方面都做得很不错。该编辑器支持多种语言和文件格式的编写, 也集成了所有现代编辑器所应该具备的特性, 包括语法高亮、可定制的热键绑定、括号匹配以及代码片段收集。

3. Hbuilder X

HBuilder X (简称 HX) 编辑器是 DCloud (数字天堂) 推出的一款支持 HTML5 的 Web 开发软件。该软件体积小, 启动快。通过完整的语法提示和代码输入法、代码块等, 大幅提升 HTML、JS、CSS 的开发效率。Hbuilder X 在使用上比较符合中国人的开发习惯, 用 HBuilder X 编写 HTML 代码还是很方便的。

4. Sublime Text3 汉化版

Sublime Text 是一款具有代码高亮、语法提示、自动完成且反应快速的编辑器软件, 它不仅具有华丽的界面, 还支持插件扩展, 而且很容易上手。

5. Notepad++

Notepad++ 旨在替代 Windows 默认的 notepad, 它比 notepad 的功能更为强大。Notepad++ 支持插件, 可通过添加不同的插件支持不同的功能。Notepad++ 属于轻量级的文本编辑类软件, 比其他一些



2 编辑 HTML 文件

专业的文本编辑类工具启动更快，占用资源更少，其功能不亚于专业工具。

6. 记事本

任意文本编辑器都可以用于编写网页源代码，最常见的文本编辑器就是 Windows 自带的记事本。

1.3.2 网页文件的创建

本书前几章的网页源代码采用在记事本中手工输入，有助于初学者对网页结构和样式更深入地了解。后几章在 HBuilderX 中编辑，以提高编辑效率。

一个网页可以简单得只有几个文字，也可以复杂得像一张或几张海报。下面创建一个只有文本组成的简单页面，来学习网页的编辑和保存过程。下面用最简单的“记事本”来编辑网页文件。

1) 打开记事本。单击 Windows 的“开始”按钮，在“Windows 附件”中单击“记事本”。

2) 在记事本窗口中输入 HTML 代码，如图 1-6 所示。

3) 在记事本的“文件”菜单中，选择“保存”选项。显示“另存为”对话框，在“保存在”下拉列表框中选择文件要存放的路径，在“文件名”文本框输入以.html 为后缀的文件名，如 index.html，在“保存类型”下拉列表框中选择“文本文档 (*.txt)”选项，在“编码”下拉列表框中选择“UTF-8”或“Unicode”选项，如图 1-7 所示。最后单击“保存”按钮，将记事本中的内容保存在磁盘中。

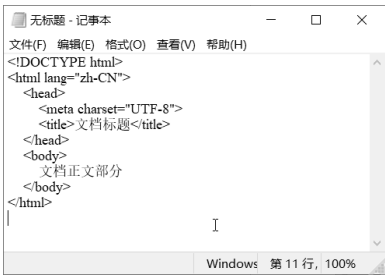


图 1-6 输入 HTML 代码

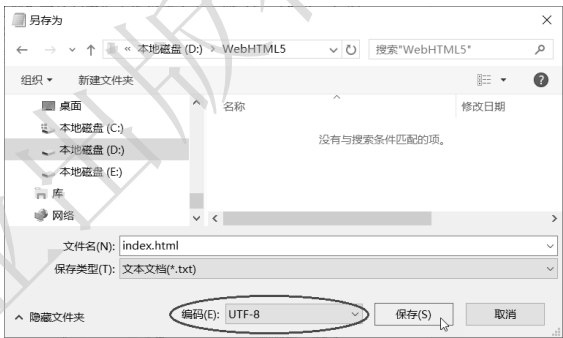


图 1-7 “记事本”的“另存为”对话框

4) 在如图 1-8 所示的文件资源管理器中，双击 index.html 文件启动浏览器，即可看到网页的显示结果，如图 1-9 所示。如果保存时选择的编码是“ANSI”，中文将显示为乱码。



图 1-8 双击 index.html 文件

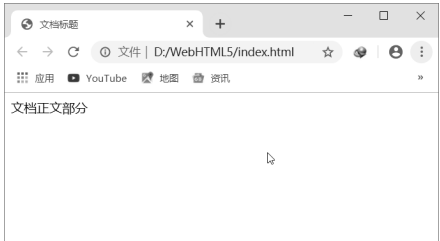


图 1-9 在浏览器中显示网页

本书所有的应用实例均是在 Windows 10 操作系统下的 Google Chrome 浏览器中运行。

1.4 实训——制作社区网版权信息

【实训 1-1】制作社区网的页脚版权信息，页面中包括版权符号、空格。本实例文件 pt1-1.html 在浏览器中显示的效果如图 1-10 所示。

```
<!DOCTYPE html>
<html>
<head>
```



图 1-10 社区网的版权信息

落元素的格式为:

在 HTML5 中，推荐使用 CSS 设置段落元素 p 的属性。

【例 2-2】 列出包含<p>标签的多种属性。本例文件 2-2.html 在浏览器中的显示效果如图 2-2 所示。

图 2-2 段落、换行示例

[illegible]

2.1.3 水平线元素 hr

使用水平线元素 `hr`，可以在浏览器中创建一条水平标尺线（Horizontal Rule），可以在视觉上将文档分割成多个部分。线段的样式由标签的参数决定。水平线元素的格式为：

在 HTML5 中，推荐使用 CSS 设置水平线元素的其他属性（线条粗细、长度、颜色等）。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>hr 标签示例</title>
  </head>
  <body>
    <p>hr 标签定义水平线: </p>
    <hr />
    <p>这是段落。</p>
    <hr />
    <p>这是段落。</p>
    <hr />
    <p>这是段落。</p>
```

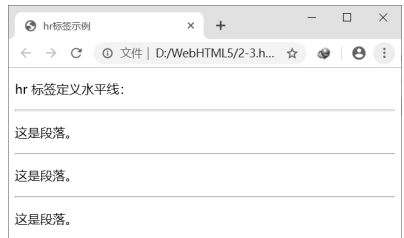


图 2-3 水平线元素示例

```
</body>
</html>
```

`<hr />` 标签强制执行一个简单的换行，将导致段落的对齐方式重新回到默认值设置（左对齐）。在 HTML5 中，所有 `<hr />` 标签的呈现属性可以使用，但不推荐使用，要想更灵活地控制并美化页面布局，推荐使用 CSS 设置。

2.1.4 注释元素

注释元素用于在源代码中插入注释。注释的作用是方便自己和项目组的其他人员阅读和调试代码，便于以后对自己代码的理解和修改。当浏览器遇到注释时会自动忽略注释内容，浏览者在浏览器中看不见这些注释，注释只有在用文本编辑器打开文档源代码时才可见。注释元素的格式为：

```
<!--注释内容-->
```

注释并不局限于一行，长度不受限制。结束标签与开始标签可以不在一行上。

【例 2-4】 注释示例。本例文件 2-4.html 在浏览器中的显示效果如图 2-4 所示。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>注释示例</title>
  </head>
  <body>
    <!--这是一段注释，不会在浏览器中显示。-->
    <p>HTML 是制作网页的基础语言，是初学者必学的内容。</p>
    <script type="text/javascript">
      <!--
        function displayMsg() {
          alert("Hello World!")
        }
      -->
    </script>
  </body>
</html>
```



图 2-4 注释显示效果

使用注释标签来隐藏浏览器不支持的脚本也是一个好习惯，这样就不会把脚本显示为纯文本。

2.2 列表元素

把相关内容用列表的形式展示，可以使内容显得更加有条理性。HTML5 提供了 3 种列表模式，即无序列表、有序列表和定义列表。本节主要介绍对应的三种列表元素。

2.2.1 无序列表元素 ul

无序列表中每项的前缀都显示一个项目符号（如●、○等符号）。用 `<ul>` 标签定义无序列表，用 `<li>` 定义列表项。`<ul>` 标签支持全局标准属性和全局事件属性。定义无序列表元素的格式为：

```
<ul>
  <li>第一个列表项</li>
  <li>第二个列表项</li>
  ...
</ul>
```



7 无序列表元素 ul

从浏览器上看，无序列表的特点是，列表项目作为一个整体，与上下段文本间各有一行空白；列表项向右缩进并左对齐，每个列表项前面都有项目符号。

HTML5 推荐用 CSS 样式来定义列表的类型。

【例 2-5】 无序列表示例。本例文件 2-5.html 在浏览器中的显示效果如图 2-5 所示。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>无序列表示例</title>
  </head>
  <body>
    <h4>征文排行</h4>
    <ul>
      <li>有一种毕业，叫先走一步</li>
      <li>用今生的修行 为你们的来生祈福</li>
      <li>我的初恋</li>
      <li>当你老了</li>
      <li>我就在这里旅行，给你讲沿途的风景</li>
      <li>有多少爱可以重来</li>
      <li>我的“天空之城”</li>
    </ul>
  </body>
</html>
```

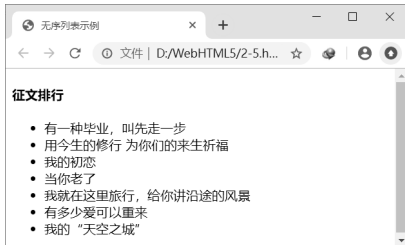


图 2-5 无序列表显示效果

2.2.2 有序列表元素 ol

有序列表的前缀通常为序号标志（如数字、字母等），通过带序号的列表可以更清楚地表达信息的顺序。使用标签可以建立有序列表，表项的标签仍为。定义有序列表元素的格式为：

```
<ol>
  <li>第一个列表项</li>
  <li>第二个列表项</li>
  ...
</ol>
```

在浏览器中显示时，有序列表整个表项与上下段文本之间各有一行空白；列表项目向右缩进并左对齐；各个列表项前都带顺序号。有序列表的每个列表项的序号默认为数字。

HTML5 推荐使用样式表 CSS 改变有序列表中的序号类型。

【例 2-6】 有序列表示例。本例文件 2-6.html 在浏览器中的显示效果如图 2-6 所示。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>有序列表示例</title>
  </head>
  <body>
    <h4>征文排行</h4>
    <ol><!--列表样式为默认的数字-->
      <li>有一种毕业，叫先走一步</li>
      <li>用今生的修行 为你们的来生祈福</li>
      <li>我的初恋</li>
```

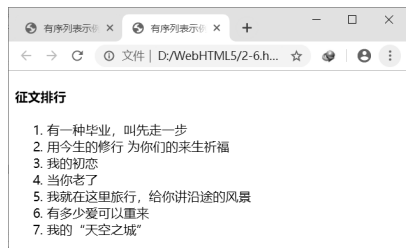


图 2-6 有序列表显示效果

```

        <li>当你老了</li>
        <li>我就在这里旅行，给你讲沿途的风景</li>
        <li>有多少爱可以重来</li>
        <li>我的“天空之城”</li>
    </ol>
</body>
</html>

```



9 定义列表元素 dl

2.2.3 定义列表元素 dl

定义列表又称为释义列表或字典列表，用<dl>标签定义。它的内容不仅仅是一列项目，而是项目及其注释的组合。定义列表的内部可以有多个列表项标题，每个列表项标题用<dt>标签定义，列表项标题内部又可以有多个列表项描述，用<dd>标签定义。定义列表元素的格式为：

```

<dl>
  <dt>第一个标题项</dt>
  <dd>对第一个标题项的解释文字 1</dd>
  <dd>对第一个标题项的解释文字 2</dd>
  ...
  <dt>第二个标题项</dt>
  <dd>对第二个标题项的解释文字 1</dd>
  ...
</dl>

```

在<dl>、<dt>和<dd>3个标签组合中，<dt>是标题，<dd>是内容，<dl>可以看作是承载它们的容器。当出现多组这样的标签组合时，应尽量使用一个<dt>标签配合一个<dd>标签的方法。如果<dd>标签中内容很多，可以嵌套<p>标签使用。

【例 2-7】 使用列表显示高分电影排行榜。本例文件 2-7.html 在浏览器中的显示效果如图 2-7 所示。

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>定义列表示例</title>
  </head>
  <body>
    <h4>高分电影排行榜</h4>
    <dl>
      <dt>按类型排行</dt>
      <dd>爱情</dd>
      <dd>喜剧</dd>
      <dd>其他类型</dd>
      <dt>按年代排行</dt>
      <dd>2020</dd>
      <dd>2019</dd>
      <dd>其他年代</dd>
    </dl>
  </body>
</html>

```



图 2-7 页面显示效果

在上面的示例中，<dl>列表中每一项的名称用<dt>标签，后面跟由<dd>标签标记的条目定义或解释。默认情况下，浏览器在左边界显示条目的名称，并在下一行缩进显示其定义或解释。

2.2.4 嵌套列表

所谓嵌套列表就是无序列表与有序列表嵌套混合使用。嵌套列表可以把页面分为多个层次，给人以很强的层次感。有序列表和无序列表不仅可以自身嵌套，而且可彼此互相嵌套。嵌套方式可分为：无序列表中嵌套无序列表、有序列表中嵌套有序列表、无序列表中嵌套有序列表、有序列表中嵌套无序列表等方式，读者需要灵活掌握。

【例 2-8】 在无序列表中嵌套无序列表、有序列表和定义列表。
本例文件 2-8.html 在浏览器中的显示效果如图 2-8 所示。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>青青博客</title>
  </head>
  <body>
    <ul><!--无序列表-->
      <li>点击排行</li>
      <ol><!--无序列表-->
        <li>十条设计原则教你学会如何设计网页布局</li>
        <li>6 条网页设计配色原则,让你秒变配色高手</li>
        <li>三步实现滚动条触动 css 动画效果</li>
      </ol>
      <hr /><!--水平分隔线-->
      <li>猜你喜欢</li>
      <ul><!--嵌套有序列表-->
        <li>安静地做一名爱设计的女子</li>
        <li>个人博客, 属于我的小世界</li>
      </ul>
      <hr /><!--水平分隔线-->
      <li>欢迎联系</li>
      <dl>
        <dt>电话: </dt>
        <dd>010-22363123</dd>
        <dt>地址: </dt>
        <dd>北京市东城区长安街 3 号</dd>
      </dl>
    </ul>
  </body>
</html>
```

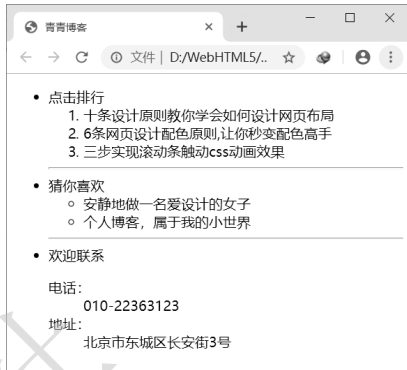


图 2-8 页面显示效果

2.3 表格元素 table

表格是由指定数目的行和列组成，每行的列数通常一致，同行单元格高度一致且水平对齐，同列单元格宽度一致且垂直对齐，这种严格的约束形成了一个不易变形的长方形盒子结构，堆叠排列起来结构很稳定，表格中的内容按照相应的行或列进行分类和显示。表格将文本和图像按行、列排列，它与列表一样，有利于表达信息。

2.3.1 基本表格

表格用<table>标签定义，标签标题用<caption>标签定义；每个表格有若干行，



用<tr>标签定义；每行被分隔为若干个单元格，用<td>标签定义；当单元格是表头时，用<th>标签定义。定义表格元素的格式为：

```
<table border="n" width="x|x%" height="y|y%" cellspacing="i" cellpadding="j">
  <caption>标题</caption>
  <tr> <th>表头 1</th> <th>表头 2</th> <th>...</th> <th>表头 n</th></tr>
  <tr> <td>表项 1</td> <td>表项 2</td> <td>...</td> <td>表项 n</td></tr>
  ...
  <tr> <td>表项 1</td> <td>表项 2</td> <td>...</td> <td>表项 n</td></tr>
</table>
```

表格是一行一行建立的，在每一行中填入该行每一列的表项数据。可以把表头看作一行，只不过用的是<th>标签。在浏览器中显示时，<th>标签的文字按粗体显示，<td>标签的文字按正常字体显示。表格的整体外观由 table 元素的属性决定。

- 1) border 属性：定义表格边框的粗细，n 为整数，单位为像素。如果省略，则不带边框。
- 2) width 属性：定义表格的宽度，x 为像素数或百分数（占窗口的）。
- 3) height 属性：定义表格的高度，y 为像素数或百分数（占窗口的）。
- 4) cellspacing 属性：定义表项间隙，i 为像素数。
- 5) cellpadding 属性：定义表项内部空白，j 为像素数。

【例 2-9】在页面中添加一个 4 行 3 列的表格。本例文件 2-9.html 在浏览器中的显示效果如图 2-9 所示。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>表格示例</title>
  </head>
  <body>
    <table>
      <caption>班级名单</caption>
      <tr><th>姓名</th><th>性别</th><th>专业</th></tr>
      <tr><td>张三丰</td><td>男</td><td>大数据与信息处理技术</td></tr>
      <tr><td>李四萍</td><td>女</td><td>软件工程</td></tr>
      <tr><td>王五一</td><td>女</td><td>计算机科学与技术</td></tr>
    </table>
    <table border="1" cellspacing="10" cellpadding="20">
      <caption>班级名单</caption>
      <tr><th>姓名</th><th>性别</th><th>专业</th></tr>
      <tr><td>张三丰</td><td>男</td><td>大数据与信息处理技术</td></tr>
      <tr><td>李四萍</td><td>女</td><td>软件工程</td></tr>
      <tr><td>王五一</td><td>女</td><td>计算机科学与技术</td></tr>
    </table>
  </body>
</html>
```



图 2-9 表格的显示效果

表格所使用的边框粗细等样式一般应放在专门的 CSS 样式文件中（后续章节讲解），此处讲解这些属性仅仅是为了演示表格案例中的页面效果，在真正设计表格外观的时候是通过 CSS 样式完成的。

2.3.2 跨行跨列表格

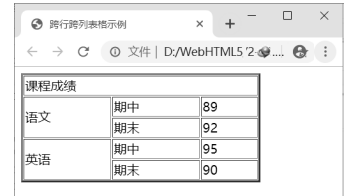
在表格中合并单元格，跨行是指单元格在垂直方向上合并，跨列是指单元格在水平方向上合并。<th>标签可以使用 rowspan 和 colspan 两个属性，分别表示该单元格纵跨多少行和横跨多少列。定义

跨行跨列表格的格式为：

```
<table>
  <tr><td rowspan="所跨的行数" colspan="所跨的列数">单元格内容</td></tr>
</table>
```

【例 2-10】 跨行跨列表格示例。本例文件 2-10.html 在浏览器中的显示效果如图 2-10 所示。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>跨行跨列表格示例</title>
  </head>
  <body>
    <table width="300" border="2">
      <tr>
        <td colspan="3">课程成绩</td><!--设置单元格水平跨 3 列-->
      </tr>
      <tr>
        <td rowspan="2">语文</td><!--设置单元格垂直跨 2 行-->
        <td>期中</td>
        <td>89</td>
      </tr>
      <tr>
        <td>期末</td>
        <td>92</td>
      </tr>
      <tr>
        <td rowspan="2">英语</td><!--设置单元格垂直跨 2 行-->
        <td>期中</td>
        <td>95</td>
      </tr>
      <tr>
        <td>期末</td>
        <td>90</td>
      </tr>
    </table>
  </body>
</html>
```



课程成绩		
语文	期中	89
	期末	92
英语	期中	95
	期末	90

图 2-10 跨行跨列的显示效果

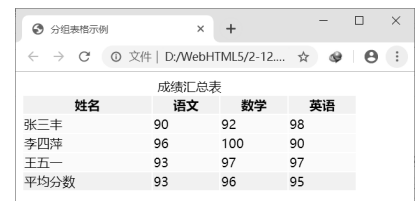
表格跨行跨列以后，并不改变表格的特点。表格中同行的内容总高度一致，同列的内容总宽度一致，各单元格的宽度或高度互相影响，结构相对稳定，不足之处是不能灵活地进行布局控制。

2.3.3 表格数据的分组

表格数据的分组标签包括<thead>、<tbody>和<tfooter>，主要用于对表格数据进行逻辑分组。其中，<thead>标签对应表格的表头；<tbody>标签对应表格的主体；<tfooter>对应表格的页脚，即对各分组数据汇总的部分。各分组标签内由多行<tr>组成，子元素仅有 td 和 th。

标签<tbody>、<thead>、<tfoot>通常用于对表格内容进行分组，当创建某个表格时，希望拥有一个标题行、一些带有数据的行，以及位于底部的一个总计行。这种划分使浏览器有能力支持独立于表格标题和页脚的表格正文滚动。当长的表格被打印时，表格的表头和页脚可被打印在包含表格数据的每张页面上。

【例 2-11】 表格分组示例。本例文件 2-11.html 在浏览器中的显示效果如图 2-11 所示。



成绩汇总表			
姓名	语文	数学	英语
张三丰	90	92	98
李四萍	96	100	90
王五一	93	97	97
平均分	93	96	95

图 2-11 表格分组的显示效果

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>分组表格示例</title>
  </head>
  <body>
    <table border="0" width="420"><!--设置表格宽度为 420px，无边框-->
      <caption>成绩汇总表</caption>
      <thead style="background:#FAF0E6"><!--设置表格的页眉-->
        <tr>
          <th>姓名</th>
          <th>语文</th>
          <th>数学</th>
          <th>英语</th>
        </tr>
      </thead><!--表格页眉结束-->
      <tbody style="background:#FFFACD"><!--设置表格主体-->
        <tr>
          <td>张三丰</td>
          <td>90</td>
          <td>92</td>
          <td>98</td>
        </tr>
        <tr>
          <td>李四萍</td>
          <td>96</td>
          <td>100</td>
          <td>90</td>
        </tr>
        <tr>
          <td>王五一</td>
          <td>93</td>
          <td>97</td>
          <td>97</td>
        </tr>
      </tbody><!--表格主体结束-->
      <tfoot style="background:#FAF0E6"><!--设置表格的数据页脚-->
        <tr>
          <td>平均分数</td>
          <td>93</td>
          <td>96</td>
          <td>95</td>
        </tr>
      </tfoot><!--表格页脚结束-->
    </table>
  </body>
</html>

```

为了区分表格各部分的颜色，这里使用了 style 样式属性分别为 thead、tbody 和 tfooter 元素设置背景色，此处只是为了演示页面效果。

2.3.4 调整列的格式

为了调整列的格式，对表格中的列组合后，可以用以下标签对表格中的列定义属性值。

1) <colgroup>: 对表格中的列进行组合, 以便对其进行格式化。

2) <col>: 为表格中一个或者多个列定义属性值, 通常位于 colgroup 元素内。

【例 2-12】列格式示例。本例文件 2-12.html 在浏览器中的显示效果如图 2-12 所示。



姓名	姓名	专业
张三丰	男	大数据与信息处理技术
李四萍	女	软件工程
王五一	女	计算机科学与技术

图 2-12 列格式的显示效果

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>分组表格示例</title>
  </head>
  <body>
    <table border="1">
      <colgroup>
        <col width="150" style="background:#FFFAF0">
        <col width="100" style="background:#8d8d8d">
        <col width="200" style="background:#FFFAF0">
      </colgroup>
      <tr>
        <th>姓名</th>
        <th>姓名</th>
        <th>专业</th>
      </tr>
      <tr>
        <td>张三丰</td>
        <td>男</td>
        <td>大数据与信息处理技术</td>
      </tr>
      <tr>
        <td>李四萍</td>
        <td>女</td>
        <td>软件工程</td>
      </tr>
      <tr>
        <td>王五一</td>
        <td>女</td>
        <td>计算机科学与技术</td>
      </tr>
    </table>
  </body>
</html>
```

2.4 表单

用户在注册、登录、搜索等时, 网页中用于输入内容的文本框、单选框、复选框、下拉列表框、按钮等, 可以用表单来实现。当访问者在表单中输入信息并单击提交按钮后, 这些信息将被发送到服务器, 服务器端脚本或应用程序将对这些信息进行处理。

2.4.1 表单元素 form

网页上具有可输入表项及项目选择等控制所组成的栏目称为表单。<form>标签用于创建供用户

输入的 HTML 表单。form 元素是块级元素，其前后会产生折行。form 元素的基本格式为：

```
<form name="表单名" action="URL" method="get|post" ...>
...
</form>
```

<form>标签主要用于表单结果的处理和传送，常用属性如下。

- 1) action 属性：规定当提交表单时向何处发送表单数据，是网址或 E-mail 地址。这个属性必须有。
- 2) method 属性：规定用于发送表单数据时的发送类型，其属性值可以是 get 或 post，具体是哪一个，取决于后台程序。这个属性必须有。
- 3) enctype 属性：规定在发送表单数据之前如何对其进行编码。enctype 属性有以下 3 个值。
 - application/x-www-form-urlencoded：默认的编码方式，在发送前编码所有字符。
 - multipart/form-data：被编码为一条二进制消息，网页上的每个控件对应消息中的一个部分，包括文件域指定的文件。在使用包含文件上传控件的表单时，必须使用这个值。
 - text/plain：空格转换为加号 (+)，但不特殊字符编码。
- 4) name 属性：表单的名字，在一个网页中用于识别表单的唯一标识，与 id 属性值相同。
- 5) target 属性：规定使用哪种方式打开目标 URL，它的属性值可以是_blank、_self、_parent 或_top 中的一个，使用方法与 a 元素的 target 属性相同。

2.4.2 输入元素 input

input 元素用来定义用户输入数据的输入字段。根据不同的 type 属性值，输入字段可以是文本字段、密码字段、复选框、单选按钮、按钮、隐藏域、图像、文件等。input 元素的基本格式为：

```
<input type="表项类型" name="元素名" size="x" maxlength="y" />
```

input 元素的常用属性如下。

- 1) type 属性：指定要加入表单项目的类型，type 属性值见表 2-1。

表 2-1 input 元素的 type 属性值

type 的属性值	描 述
text	单行文本输入框，可以输入一行文本，可通过 size 和 maxlength 定义显示的宽度和最大字符数
password	密码输入框，同单行文本框，不同的是，该区域字符会被掩码
radio	单选按钮，相同 name 属性的单选按钮只能选中一个，默认选中用 checked="checked"
checkbox	复选框，可以同时选中多个，默认选中用 checked="checked"
submit	提交按钮，单击本按钮后将表单数据发送到服务器
reset	重置按钮，单击本按钮后会清除表单中输入的所有数据
button	按钮，大部分情况下执行的是 JavaScript 脚本
image	图片形式的提交按钮，效果同提交按钮，必须使用 src 属性定义图片的 URL，并且使用 alt 定义当图片无法显示时的替代文字。height 和 width 属性定义图片的高和宽
file	选择文件控件，用于上传文件
hidden	隐藏的输入区域，一般用于定义隐藏的参数
color	让用户从拾色器中选择一个颜色
date	让用户从一个日期选择器选择一个日期
datetime	让用户从一个 UTC 日期和时间选择器中选择一个日期。有的浏览器不支持
datetime-local	让用户从日期时间选择器中选择一个本地的日期和时间
time	让用户从时间选择器中选择小时和分
month	让用户从月份选择器中选择月份，包括年和月



type 的属性值	描 述
week	让用户从周、年选择器中选择周和年
email	生成一个 E-Mail 地址的输入框
number	生成一个只能输入数值的输入框
range	生成一个拖动条, 通过拖动输入一定范围内的数字值
search	生成一个用于输入搜索关键字的文本框
tel	生成一个只能输入电话号码的文本框
url	生成一个 URL 的输入框

2) name 属性: 定义 input 元素的名称。

3) size 属性: 定义该控件的宽度。

4) maxlength 属性: 规定输入字段中字符的最大长度。

5) checked 属性: 当页面加载时是否预先选择该 input 元素 (适用于 type="checkbox" 或 type="radio")。

6) readonly 属性: 规定输入字段为只读, 字段的值无法修改。

7) autofocus 属性: 规定输入字段在页面加载时是否获得焦点 (不适用于 type="hidden")。

8) disabled 属性: 当页面加载时是否禁用该 input 元素 (不适用于 type="hidden")。

9) value 属性: 规定 input 元素的默认值。

【例 2-13】制作不同类型的表单按钮, 本例文件 2-13.html 在浏览器中的显示效果如图 2-13 所示。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>表单的 input 示例</title>
  </head>
  <body>
```

```
    <form action="" method="">
```

```
      账号: <input type="text" name="user" size=30 /><br />
```

```
      密码: <input type="password" name="passwd" size=30 /><br />
```

```
      性别: <input type="radio" name="sex" value="male" /> 男
```

```
      <input type="radio" name="sex" value="female" checked="checked" /> 女<br />
```

```
      技术: <input type="checkbox" name="tech" value="java" />Java
```

```
      <input type="checkbox" name="tech" value="html" />html
```

```
      <input type="checkbox" name="tech" value="css" />CSS<br />
```

```
      选择上传文件: <input type="file" name="file" /><br />
```

```
      图片按钮: <input type="image" src="images/ClickEnter.jpg" width="80" height="25"><br />
```

```
      隐藏组件: <input type="hidden" name="mykey" value="myvalue" /><br />
```

```
      选择你喜欢的颜色: <input type="color" name="favcolor"><br />
```

```
      工作日期: <input type="date" name="bday"><br />
```

```
      生日(日期和时间): <input type="datetime-local" name="bdaytime"><br />
```

```
      选择时间: <input type="time" name="usr_time"><br />
```

```
      生日(月和年): <input type="month" name="bdaymonth"><br />
```

```
      数量(1 到 5 之间): <input type="number" name="quantity" min="1" max="5"><br />
```

```
      强度: <input type="range" name="points" min="1" max="10"><br />
```

```
      <input type="reset" />&nbsp;&nbsp;&nbsp;<input type="submit" />&nbsp;&nbsp;&nbsp;<input type=
```

```
      "reset" value="自定义按钮" />
    </form>
```

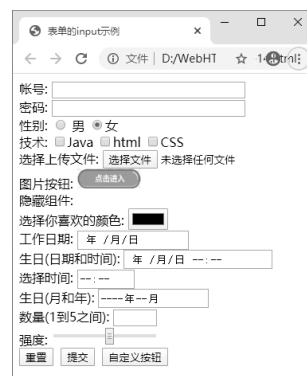


图 2-13 不同类型的按钮

```
</body>
</html>
```

2.4.3 标签元素 label

<label>标签用于为表单中的其他控件元素添加说明文字。当用户在浏览器中单击 label 元素生成的标签时，会自动将焦点转到与该标签相关的表单控件上。label 元素的格式为：

<label for="id">说明文字</label>

<label>标签最重要的属性是 for 属性。for 属性把 label 元素绑定到另外一个元素，把 for 属性的值设置为相关元素的 id 属性的值。使<label>标签与表单控件关联的方法有以下两种。

- 将<label>标签的 for 属性值，指定为关联表单控件的 id。
- 把说明与表单控件一起放入<label>...</label>标签内部。

【例 2-14】 label 元素的示例。本例文件 2-14.html 在浏览器中的显示效果如图 2-14 所示，单击“密码”标签，焦点将定位到其关联的文本框中。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>label 元素示例</title>
  </head>
  <body>
    <form action="" method="post">
      <label for="username">用户名: </label><input id="username" type="text" name="user" />
<br />
      <label>密码: <input type="password" name="passwd" /></label><br />
    </form>
  </body>
</html>
```



图 2-14 label 元素的显示效果

2.4.4 选择栏元素 select

select 元素可用来创建菜单或者下拉列表框，实现单选或多选菜单。<select>标签必须配合<option>标签和<optgroup>标签使用，<option>标签定义列表中的可用选项；<optgroup>标签表示一个选项组，该元素中只能有 option 子元素。

1. select 元素

select 元素的格式为：

```
<select size="x" name="控件名" multiple="multiple">
  <optgroup>
    <option ...> ... </option>
    <option ...> ... </option>
    ...
  </optgroup>
  ...
</select>
```

select 元素的属性如下。

- 1) size 属性：指定下拉列表中同时显示选项的数目，默认值为 1。
- 2) name 属性：下拉列表的名称。
- 3) multiple 属性：指定可选择多个选项，属性值只能是 multiple。无此属性时则为单选。

2. option 元素

option 元素定义下拉列表中的一个选项。浏览器将<option>标签中的内容作为<select>标签的菜单或是滚动列表中的一个元素显示。option 元素必须位于 select 元素内部。option 元素的格式为：

```
<option value="选项值" selected="selected">...</option>
```

option 元素的属性如下。

- 1) value 属性：定义该列表项对应的送往服务器的参数。若省略，则初值为 option 中的内容。
- 2) selected 属性：指定该选项的初始状态为选中，其属性值只能是 selected。

3. optgroup 元素

如果列表选项很多，可以使用<optgroup>标签对相关选项分组。optgroup 元素的格式为：

```
<optgroup>
  <option ...> ... </option>
  <option ...> ... </option>
  ...
</optgroup>
```

optgroup 元素的属性如下。

- 1) label 属性：为选项组指定说明文字，本属性必须设置。
- 2) disabled 属性：设置该选项组是否可用，属性值是 disabled。

【例 2-15】 制作问卷调查的下拉列表。本例文件 2-15.html 在浏览器中的显示效果如图 2-15 所示。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>表单的 input 示例</title>
  </head>
  <body>
    <form action="" method="post">
      你希望从事的专业？（单选）
      <select>
        <option value="front">前端开发</option>
        <option value="back">后端开发</option>
        <option value="ai">人工智能</option>
      </select><br /><br />
      你熟悉的技术有哪些？（多选）
      <select size="3" multiple="multiple">
        <option value="html">HTML</option>
        <option value="jq" selected="selected">jQuery</option>
        <option value="mysql">MySQL</option>
        <option value="asp">ASP.NET</option>
      </select><br /><br />
      你希望到哪个城市工作？（多选）
      <select size="8" multiple="multiple">
        <optgroup label="华北地区">
          <option value="beijing">北京市</option>
          <option value="tianjin">天津市</option>
          <option value="hebei">河北省</option>
        </optgroup>
        <optgroup label="华东地区">
          <option value="shanghai">上海市</option>
          <option value="jiangsu">江苏省</option>
        </optgroup>
      </select>
    </form>
  </body>
</html>
```

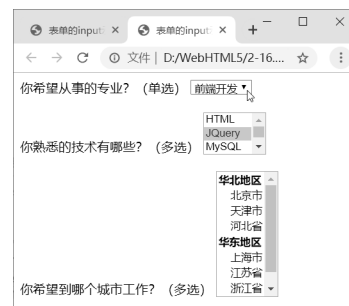


图 2-15 页面的显示效果

</html>

2.4.5 按钮元素 button

`button` 元素定义一个按钮。`<button>`与`</button>`标签之间的所有内容都是按钮的内容，其中包括任何可接收的内容，包括文本、图像或多媒体内容。这是该元素与 `input` 元素创建的按钮之间的不同之处。`button` 元素与`<input type="button">`相比，前者提供了更强大的功能和更丰富的内容。`button` 元素的格式为：

`<button type="按钮的类型">文本、图像元素</button>`

button 元素的属性如下。

- 1) **type** 属性：指定按钮的类型，只能是 **button**、**reset** 或 **submit**，与 **input** 元素的 3 种类型的按钮相对应。
- 2) **autofocus** 属性：指定当页面加载时按钮自动地获得焦点。
- 3) **disabled** 属性：指定禁用该按钮。
- 4) **name** 属性：指定按钮的名称。
- 5) **value** 属性：指定按钮的初始值，可由脚本进行修改。

【例 2-16】 按钮元素示例。本例文件 2-16.html 在浏览器中的显示效果如图 2-16 所示。

[illegible]

图 2-16 按钮元素

2.4.6 多行文本元素 textarea

`textarea` 元素定义多行文本输入控件。该控件可以用来输入多个段落的文字，文本区中可容纳无限数量的文本。`textarea` 元素的格式为：

<textarea name="名称" rows="行数" cols="列数">

初始文本内容

</textarea>

textarea 元素的属性如下。

- 1) cols 属性：指定 textarea 文本区内的宽度。此属性必须设置。
- 2) rows 属性：指定 textarea 文本区内的可见行数，即高度。此属性必须设置。
- 3) maxlength 属性：指定文本区域的最大字符数。行数和列数是指不拖动滚动条就可看到的部分。
- 4) name 元素：指定本标签的 ID 名称。
- 5) placeholder 元素：指定描述文本区的简短提示。
- 6) readonly 元素：指定文本区为只读。这个属性的属性值只能是 readonly。
- 7) required 元素：指定文本区是必填的。这个属性的属性值只能是 required。

通过 cols 和 rows 属性来规定 textarea 的尺寸，不过更好的办法是使用 CSS 的 height 和 width 属性。

注释：在文本区内的文本行间，用 “%OD%OA”（回车/换行）进行分隔。

【例 2-17】 多行文本元素示例。本例文件 2-17.html 在浏览器中的显示效果如图 2-17 所示。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>textarea 元素示例</title>
  </head>
  <body>
    <form action="" method="post">
      <p>学习经历</p>
      <textarea rows="5" cols="60" placeholder="从初中开始，必填" required="required">
    </textarea><br />
      <p>备注</p>
      <textarea rows="4" cols="60"></textarea><br />
      <input type="submit" name="" id="" value="确定" />&nbsp;&nbsp;&nbsp;<input type="reset"
name="" id="" value="重置输入" />
    </form>
  </body>
</html>
```



图 2-17 多行文本元素

2.5 分区元素 div



前面讲解的几类块级元素一般用于组织小区块的内容，为了方便管理，许多小区块还需要放到一个大区块中进行布局。分区元素 div 常用于页面布局时区块的划分，它相当于一个大“容器”，可定义文档中的分区。div 是 division 的简写，意为分割、区域、分组。div 元素可以把文档分割为独立的、不同的部分。它可以用作严格的组织工具，并且不使用任何格式与其关联。div 元素可以容纳无序列表、有序列表、表格、表单等块级标签，同时也可以容纳普通的标题、段落、文字、图片等内容。

div 元素是一个块级元素，也就是说，浏览器通常会在 div 元素前后放置一个换行符。实际上，换行是 div 元素固有的唯一格式表现。通常使用 div 元素来组合块级元素，这样就可以使用样式对它们进行格式化。由于 div 元素没有明显的外观效果，因此需要为其添加 CSS 样式属性，才能看到区块的外观效果。div 元素的格式为：

<div id="控件 id" class="类名">文本、图像或表格</div>

div 元素的属性如下。

【例 2-19】 blockquote 元素的基本用法。本例文件 2-19.html 在浏览器中的显示效果如图 2-19 所示。

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>blockquote 元素示例</title>
  </head>
  <body>
    <h4>院系概况</h4>
    <blockquote>
      <p>计算机科学系成立于 1988 年 3 月，由计算机科学微机培训中心、计算机教研室组建而成。现下设系党政办公室、团学办公室、计算机实验中心（含计算机公共课部）、网络管理中心等四个科级管理部门。</p>
    </blockquote>
    <p>计算机科学系有一支以中青年业务骨干为核心，实力雄厚、治学严谨、年龄结构合理、学科梯队健全、专业结构优势互补的专业师资队伍。</p>
  </body>
</html>

```

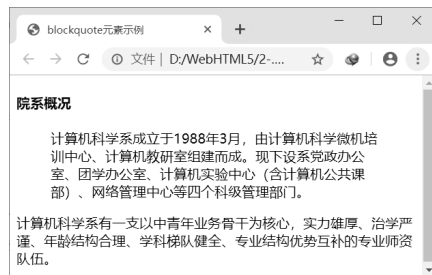


图 2-19 <blockquote>标签示例

2.7 实训——制作精选信息版块

【实训 2-1】 用<div>标签组织网页内容，制作精选信息版块。本实训文件 pt2-1.html 在浏览器中的显示效果如图 2-20 所示。

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>社区网主页</title>
  </head>
  <body>
    <div style="width: 585px;border:1px solid rgb(220,220,220); margin-top: 16px;margin-left: 25px;">
      <div style="height: 34px;padding-top: 5px;border:1px solid rgb(220,220,220);">
        <label style="font-size: 18px;">精选信息</label>
        <label style="float: right;">more>></label>
      </div>
      <div style="margin: 8px;font-size: 14px;line-height: 33px; color:#555454;">
        <ul>
          <li>1 在活动现场看到的最有趣的场面<label style="float: right;">2020-08-23</label></li>
          <li>2 在活动现场看到的最有趣的场面<label style="float: right;">2020-08-23</label></li>
          <li>3 在活动现场看到的最有趣的场面<label style="float: right;">2020-08-23</label></li>
          <li>4 在活动现场看到的最有趣的场面<label style="float: right;">2020-08-23</label></li>
          <li>5 在活动现场看到的最有趣的场面<label style="float: right;">2020-08-23</label></li>
          <li>6 在活动现场看到的最有趣的场面<label style="float: right;">2020-08-23</label></li>
        </ul>
      </div>
    </div>
  </body>
</html>

```

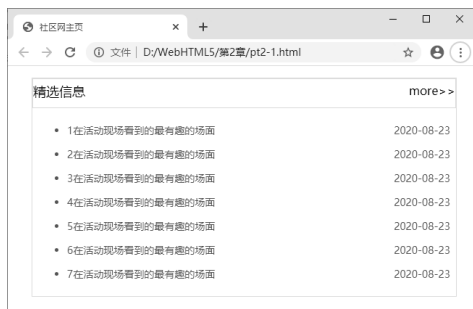


图 2-20 精选信息版块

</label>

7 在活动现场看到的最有趣的场面<label style="float: right;">2020-08-23

</label>

</div>

</div>

</body>

</html>

【说明】由于页面中的内容并未设置 CSS 样式，因此整个页面看起来并不美观，在后续章节的练习中将利用 CSS 样式对该页面进行美化。

习题 2

1. 制作如图 2-21 所示的课程表。
2. 制作如图 2-22 所示的注册表单。



时间\日期	一	二	三	四	五
上午	9:30-10:15	语文	语文	语文	语文
	10:25-11:10	语文	语文	语文	语文
下午	14:30-15:15	语文	语文	语文	语文
	15:25-16:10	语文	语文	语文	语文

图 2-21 课程表



用户名:

密码:

性别: ☐ 男 ☐ 女

爱好: ☐ 游泳 ☐ 读书 ☐ 看电影 ☐ 玩游戏 ☐ 篮球 ☐ 围棋

学历:

生日:

自我介绍:

图 2-22 注册表单

3. 使用<div>标签组织段落等网页内容，如图 2-23 所示。

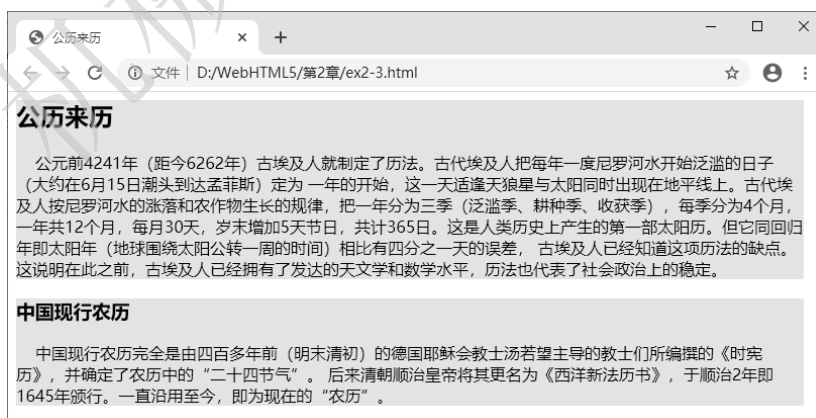


图 2-23 使用<div>标签组织网页内容

第3章 行内元素

行内元素也称为行级元素、内联元素。当设计者使用块级元素完成网页元素的组织与布局后，要为其中的每个小区块添加内容，就需要用到行内元素。

通过对本章的学习，应掌握 HTML 的行内元素及其属性。

3.1 格式化元素

HTML 的格式化元素包括字体样式元素、短语元素。

3.1.1 字体样式元素

字体样式元素可以使文本内容在浏览器中呈现特定的文字效果。但是，这些文本格式化元素仅能实现简单的、基本的文本格式化。在 HTML5 中，建议使用 CSS 样式来实现更加丰富的文本格式化效果。对于简单地更改字体样式，文本格式化元素也会经常用到。字体样式元素中全是成对出现的标签，而且不使用属性。常用的字体样式元素见表 3-1。

表 3-1 字体样式元素

元 素	描 述
b	本元素定义粗体文本，是 bold 的缩写。呈现粗体文本效果。根据 HTML5 规范，在没有其他标签更合适时，才使用标签。HTML5 规范声明，使用<h1>~<h6>表示标题，使用标签表示强调的文本，使用标签表示重要文本，使用<mark>标签表示标注的或突出显示的文本
big	本元素呈现大一号字体效果，使用<big>标签可放大字体，浏览器显示包含在<big>标签和其结束标签</big>标签之间的文字时，其字体比周围的文字要大一号。但是，如果文字已经是最大号字体，这个<big>标签将不起任何作用。甚至可以嵌套<big>标签来放大文本。每一个<big>标签都可以使字体大一号，直到上限 7 号文本，正如字体模型所定义的那样。但是使用<big>标签的时候还是要小心，因为浏览器总是“试图”去理解各种标签，对于那些不支持<big>标签的浏览器来说，它经常将其认为是粗体字标签
i	本元素将包含其中的文本以斜体字（ <i>italic</i> ）或者倾斜（ <i>oblique</i> ）字体显示。如果这种斜体字对该浏览器不可用的话，可以使用高亮、反白或加下画线等样式
small	本元素呈现小号字体效果。<small>标签和它所对应的<big>标签类似，但它是缩小字体。如果被包围的字体已经是字体模型所支持的最小字号，那么<small>标签将不起任何作用。<small>标签也可以嵌套，从而连续地把文字缩小。每个<small>标签都把文本的字体变小一号，直到达到下限字号
tt	本元素呈现类似打字机或者等宽的文本效果。对于那些已经使用了等宽字体的浏览器来说，这个标签在文本的显示上就没有什么特殊效果了
sup	本元素定义上标文本。包含在^{标签和其结束标签}中的内容将会以当前文本流中字符高度的一半来显示上标，但是与当前文本流中文字的字体和字号都是一样的。这个元素在向文档添加脚注以及表示方程式中的指数值时非常有用。如果和<a>标签结合起来使用，就可以创建出很好的超链接脚注
sub	本元素定义下标文本。包含在_{标签和其结束标签}中的内容将会以当前文本流中字符高度的一半来显示下标，但是与当前文本流中文字的字体和字号都是一样的。无论是<sub>标签还是和它对应的<sup>标签，在数学等式、科学符号和化学公式中都非常有用

【例 3-1】字体样式元素示例。本例文件 3-1.html 在浏览器中的显示效果如图 3-1 所示。

```
<!DOCTYPE html>
<html>
<head>
```

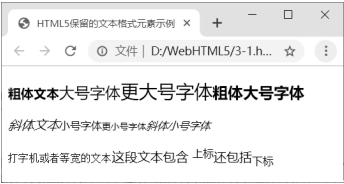


图 3-1 字体样式元素示例

```

<meta charset="utf-8">
<title>HTML5 保留的文本格式元素示例</title>
</head>
<body>
<p><b>粗体文本</b><big>大号字体</big><big><big>更大号字体</big></big><b><big>粗
体大号字体</big></b></p>
<p><i>斜体文本</i><small>小号字体</small><small><small>更小号字体</small> </small>
<i><small>斜体小号字体</small></i></p>
<p><tt>打字机或者等宽的文本</tt>这段文本包含<sup>上标</sup>还包括<sub>下标
</sub> </p>
</body>
</html>

```

3.1.2 短语元素

HTML 还提供了一些拥有明确语义的元素，用以标注特殊用途的文本，这类特殊的文本格式化元素都会呈现特殊的样式。在文本中加入强调也需要有技巧。如果强调太多，有些重要的短语就会被漏掉；如果强调太少，就无法真正突出重要的部分。这与调味品一样，最好是适量使用。就像其他与计算机编程和文档相关的标签一样，语义标签不只是为了让用户更容易理解和浏览文档，将来某些自动系统还可以利用这些恰当的标签，从文档中提取信息以及有用参数。提供给浏览器的语义信息越多，浏览器就可以越好地把这些信息展示给用户。如果只是为了达到某种视觉效果而使用这些标签，则不建议使用，而应该使用 CSS 样式。常用的特殊语义短语元素见表 3-2。

表 3-2 常用的特殊语义短语元素

元素	描述
em	本元素内的文本为强调的内容。浏览器会把这段文字使用斜体来显示
strong	与 em 元素一样，本元素用于强调文本，但它强调的程度更强一些。在浏览器中使用粗的字体来显示
code	本元素表示计算机源代码或者其他机器可以阅读的文本内容。在浏览器中显示等宽、类似电传打字机样式的字体（Courier）
kbd	本元素表示文本从键盘输入。浏览器通常用等宽字体来显示该标签中包含的文本
var	本元素表示变量的名称，或者由用户提供的值。用<var>标签标记的文本通常显示为斜体
dfn	本元素标记特殊术语或短语。浏览器通常用斜体来显示<dfn>标签中的文本
cite	本元素通常表示它所包含的文本对某个参考文献的引用，比如书籍或者杂志的标题。按照惯例，引用的文本将以斜体显示
address	本元素定义文档或文章的作者或拥有者的联系信息，显示为斜体字
q	本元素定义短的引用，浏览器把引用的内容添加双引号
pre	本元素定义预格式化的文本。pre 元素中的文本会保留空格和换行符，文本呈现为等宽字体。<pre>标签的一个常见应用就是表示计算机的源代码。pre 元素中允许的文本可以包括物理样式和基于内容的样式变化，还有链接、图像和水平分隔线
del	本元素定义文档中已经被删除（delete）的文本，文字上显示一条删除线
ins	本元素定义已经被插入（insert）文档中的文本。<ins>与标签配合使用，来描述文档中的更新和修正
samp	本元素定义计算机程序代码的样本文本。该元素并不经常使用，只有在要从正常的上下文中将某些短字符序列提取出来，对它们加以强调的极少情况下，才使用该元素
abbr	本元素用来表示一个缩写词或者首字母缩略词，如“WWW”。通过对缩写词语进行标记，能够为浏览器、拼写检查程序、翻译系统以及搜索引擎分度器提供有用的信息
bdo	bdo（Bi-Directional Override）元素定义文字方向，使用 dir 属性，属性值是 ltr（left to right，从左到右）或者 rtl（right to left，从右到左）

【例 3-2】短语元素示例。本例文件 3-2.html 在浏览器中的显示效果如图 3-2 所示。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>短语元素示例</title>
  </head>
  <body>
    <p>em 标签告诉浏览器把文本表示为强调的内容，<em>用斜体来显示。</em></p>
    <p>strong 强调的程度更强一些，<strong>用粗的字体来显示。</strong></p>
    <p><code>
      <pre>
PI = 3.1415926
r = int(input('r=')) #请输入 <code>100</code>，其中变量 <code>r</code> 表示圆的半径
s = PI*r**2
print('s=', s)
      </pre>
    </code>
    </p>
    <p>She said <code>q</code>I didn't know.</code></p>
    <p>一打有 <code>del>20</del> <code>ins>12</ins> 件。</p>
  </body>
</html>
```

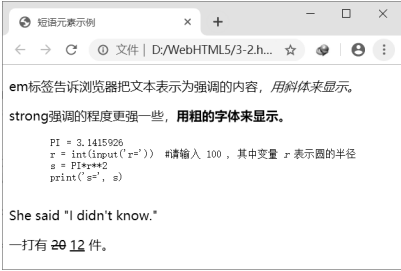


图 3-2 短语元素示例

3.2 图像元素



图像也称图片，是网页中不可缺少的元素，它可以美化网页，使网页看起来更加美观大方。虽然有很多种计算机图像格式，但由于受网络带宽和浏览器的限制，在 Web 上常用的图像格式有 3 种：GIF、JPEG 和 PNG。

img 元素向网页中嵌入一幅图像。从技术上讲，标签并不会在网页中插入图像，而是从网页上链接图像。标签创建的是被引用图像的占位空间。img 元素的格式为：

img 元素中的属性说明如下。

- 1) src: 指出要加入图片的位置，即“图像文件的 URL/图像文件名”，其中 URL 可以是相对路径，也可以是绝对路径。本属性是必需的属性。
- 2) alt: 在浏览器尚未完全读入图像或显示的图像不存在时，在图像位置显示的文字。本属性是必需的属性。
- 3) width: 设置图像的宽度（像素数或百分数）。如果不设置图像的大小，图像将按照其本身的大小显示。属性值可取像素数，也可取百分数。百分数是指相对于当前浏览器窗口的百分比。
- 4) height: 设置图像的高度（像素数或百分数）。
- 5) title: 为浏览者提供额外的提示或帮助信息。

【例 3-3】图像元素示例。本例文件 3-3.html 在浏览器中正常显示的效果如图 3-3 所示。当显示的图像路径错误时，显示效果如图 3-4 所示。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
```

```

<title>图像元素示例</title>
</head>
<body>
  <h3>荷兰郁金香公园</h3>
  <p>库肯霍夫公园位于阿姆斯特丹近
郊的小镇利瑟(Liess), 公园内郁金香的品种、数量、质量以及布置手法堪称世界之最。</p>
  <p>
    
    
  </p>
</body>
</html>

```



图 3-3 正常显示的图像效果



图 3-4 图像路径错误时的显示效果

当显示的图像不存在时,页面中图像的位置将显示网页图片丢失的信息,但由于设置了 alt 属性,因此在或的右边显示替代文字;同时,由于设置了 title 属性,因此在替代文字附近还显示提示信息。因此,在使用标签时,最好同时使用 alt 属性和 title 属性,避免图片路径错误带来的错误信息,同时,增加了提示信息也方便浏览者阅读。

3.3 超链接元素 a

超链接(Hyperlink)按照标准叫法称为锚(Anchor),是使用<a>标签定义的。超链接可以是一个字、词组、句子或图像。当网页中包含超链接时,在所有浏览器中,链接的默认外观是:未被访问的链接带有下划线而且是蓝色的;已被访问的链接带有下划线而且是紫色的;活动链接带有下划线而且是红色的。把鼠标指针移动到网页中的某个超链接上时,鼠标指针变为一只小手,单击可以从当前网页跳转到其他位置,包括当前页的某个位置、Internet、本地硬盘或局域网上的其他网页或文件,包括跳转到声音、图像等多媒体文件。

1. a 元素

锚由标签<a>定义,在网页上建立超文本链接。通过单击一个词、句或图像,可从此处转到另一个链接资源(目标资源),目标资源有唯一的地址(URL)。具有以上特点的词、句或图像就称为热点。a 元素的格式为:

```
<a href="URL" target="打开窗口方式">热点</a>
```

a 元素中的属性说明如下。

1) href: 规定超链接指向页面的 URL。如果要创建一个不链接到其他位置的空超链接,可用“#”代替 URL。超链接目标可以是站内目标,也可以是站外目标;站内目标可以用相对路径,也可以用绝对路径,站外目标则必须用绝对路径。



2) target: 超链接被单击后会产生网页跳转动作, 该属性指定打开目标页面的方式。其属性值如下。

2. 用图像作为超链接热点

图 3-5 文本超链接热点和图片超链接热点

3. 指向其他页面的超链接

（1）链接到同一目录内的网页文件

热点文本

（2）链接到下一级目录中的网页文件

热点文本

热点文本

(4) 链接到同级目录中的网页文件

`<a href=" ../子目录名/目标文件名.html">热点文本</a>`

表示先退到上一级目录中，然后进入目标文件所在的目录。

【例 3-5】 指向其他页面的超链接示例。当前页 3-5.html 中包含两个超链接，分别指向“友情链接”页 3-3.html 和“图像元素示例”页 3-4.html，如图 3-6 所示，单击超链接热点将分别打开图 3-3 和图 3-5 所示。

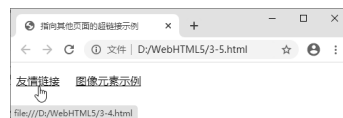
[illegible]

图 3-6 页面之间的超链接

4. 指向书签的超链接

书签就是用标签对网页元素做一个记号，其功能类似于用于固定船的锚，所以书签也称锚记或锚点。如果页面中有多个书签超链接，对不同目标元素要设置不同的书签名。书签名在标签的name 属性中定义，格式为：

目标文本附近的字符串

(1) 指向页面内书签的超链接

要在当前页面内实现书签超链接，需要定义两个标签：一个为超链接标签，另一个为书签标签。超链接标签的格式为：

`<a href="#记号名"> 热点文本 </a>`

即单击“热点文本”，将跳转到“记号名”开始的网页元素。

【例 3-6】 制作指向页面内书签的超链接。在当前页 3-6.html 的上部单击 “[什么是超文本?]” 链接时, 将跳转到页面下方的 “什么是超文本?” 位置处, 如图 3-7 所示。



图 3-7 指向页面内书签的超链接

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>指向页面内书签的链接示例</title>
    </head>
    <body>
        <p><a href="3-4.html">友情链接</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&
```


虽然这也是空超链接，但它其实有锚点#top 的意思，会产生回到顶部的效果。

(2) `<a href="javascript:void(0);">热点文本</a>`

`href="javascript:void(0);"`的含义是让超链接去执行一个 JavaScript 函数，而不是跳转到一个地址。`void(0)`表示一个空的方法，它不做任何操作，这样会防止超链接跳转到其他页面。这么做往往是为了保留超链接的样式，但不让超链接执行任何实际操作。

3.4 图像热区超链接元素 map、area

除了对整幅图像设置超链接外，还可以将图像划分为若干区域，叫作热区，每个热区可设置不同的超链接。此时，包含热区的图像称为映射图像，即带有可单击区域的图像。图像热区使用的不再是 a 元素，而是 area 元素。图像热区超链接的使用步骤如下。

1. 通过<map>...</map>标签定义图像地图

`<map>`标签用于图像映射。`<map>...</map>`标签中可以包含一个以上的热区`<area>`标签，每个热区`<area>`标签都有独立的超链接。area 元素始终嵌套在`<map>...</map>`标签之中。语法格式为：

```
<map name="映射图像名" id="映射图像名">
  <area shape="热区形状 1" coords="热区坐标 1" href="超链接地址 1" />
  <area shape="热区形状 2" coords="热区坐标 2" href="超链接地址 2" />
  ...
  <area shape="热区形状 n" coords="热区坐标 n" href="超链接地址 n" />
</map>
```

map 元素中的 name 和 id 属性，在 XHTML 中，name 属性已经废弃，使用 id 属性代替它。在 HTML5 中必须同时指定 name 和 id 属性相同的“映射图像名”。

area 元素有两个重要属性：

1) shape 属性：定义热区形状，它有以下 3 个值。

- circle：圆形区域。
- rect：矩形区域。
- poly：多边形区域。

2) coords 属性：定义矩形、圆形或多边形区域的坐标。图像的左上角坐标是 (0,0)，x 轴向右为正，y 轴向下为正。coords 属性的格式如下。

- 如果 shape="circle"，则 coords 包含 3 个参数，分别为 x、y 和 r。这 3 个参数是圆心坐标 (x, y) 和圆的半径 r。
- 如果 shape="rect"，则 coords 包含 4 个参数，分别为 x1、y1、x2、y2。这 4 个参数分别是矩形的左上角的坐标 (x1, y1) 和右下角的坐标 (x2, y2)。
- 如果 shape="poly"，则 coords 需要按顺序取多边形各个顶点的坐标 (x, y)，因此形式为 "x1, y1, x2, y2, ..., xn, yn"，其数量必须是偶数。可以是逆时针，也可以是顺时针。HTML 会按照定义顶点的顺序将它们连接起来，形成多边形热区。

2. 将 img 元素的 usemap 属性与 map 元素的 name、id 属性相关联

在图像文件中设置映射图像名，格式为：

```

```

img 元素中的 usemap 属性要引用 map 元素的 id 或 name 属性，所以应同时向 map 元素添加 id 和 name 属性。也就是说，img 元素的 usemap 属性的“映射图像名”必须与 map 元素的 name 和 id 属性的“映射图像名”相同，使得 img 元素的 usemap 属性与 map 元素的 name、id 属性相关联，以创建图像与映射之间的关系。

【例 3-7】 设计带有可点击区域的图像映射。本例文件 3-7.html 在浏览器中的显示效果如图 3-8 所示。



图 3-8 带有可点击区域的图像映射

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>图像热点链接</title>
  </head>
  <body>
    <map name="image_link">
      <area shape="circle" coords="50,50,50" href="3-4.html" alt="" />
      <area shape="rect" coords="100,50,200,200" href="3-5.html" alt="" />
      <area shape="poly" coords="250,35,300,20,250,80" href="3-6.html" alt="" />
    </map>
    
  </body>
</html>
```

3.5 范围元素 span

span 元素用来组合文档中的行内元素。span 元素没有固定的格式表现，当对它应用样式时，它才会产生视觉上的变化。span 元素用于标识行内的某个范围，以实现行内某个部分的特殊设置以区别于其他内容。其格式为：

`<span>内容</span>`

例如，`<p><span>文本内容</span>其他内容</p>`。

如果不对 span 元素应用样式，那么 span 元素中的文本与其他文本不会有任何视觉上的差异。尽管如此，上例中的 span 元素仍然为 p 元素增加了额外的结构。

可以为 span 元素应用 id 或 class 属性，这样既可以增加适当的语义，又便于对 span 元素应用样式。

span 元素与 div 元素的区别在于，span 元素仅仅是个行内元素，不会换行，而 div 元素是一个块级元素，它包围的元素会自动换行。块级元素相当于在行内元素前后各加了一个 `<br />` 标签。用容器这一词更容易理解它们的区别，块级元素 div 相当于一个大容器，而行内元素 span 相当一个小容器，大容器当然可以盛放小容器。

另外，span 元素本身没有任何属性，没有结构上的意义，当其他元素都不合适的时候可以使用，同时 div 元素可以包含 span 元素，反之则不行。

3.6 多媒体元素

多媒体元素包括音频元素和视频元素。

1. 音频格式

(1) OGG Vorbis

OGG Vorbis 是一种新的音频压缩格式, 类似于 MP3 等现有的音乐格式。它是完全免费、开放和没有专利限制的。OGG Vorbis 有一个很出众的特点, 就是支持多声道。OGG Vorbis 文件的扩展名是 .ogg, 这种文件的设计格式非常先进, 目前创建的 OGG 文件可以在未来的任何播放器上播放。因此, 这种文件格式可以不断地进行大小和音质的改良, 而不影响旧有的编码器或播放器。

(2) MP3

MP3 格式诞生于 20 世纪 80 年代的德国。所谓的 MP3 是指 MPEG 标准中的音频部分, 也就是 MPEG 音频层。MPEG 音频文件的压缩是一种有损压缩, 通过牺牲声音文件中 12~16kHz 之间的高音频部分的质量来压缩文件的大小。相同时间长度的音乐文件, 用 MP3 格式存储时一般只有 WAV 文件的 1/10, 音质也次于 CD 格式和 WAV 格式的声音文件。

(3) WAV

WAV 格式是 Microsoft 公司开发的一种声音文件格式, 用于保存 Windows 平台的音频信息资源, 被 Windows 平台及其应用程序所支持, 支持多种音频位数、采样频率和声道, 是目前 PC 上广为流行的声音文件格式。几乎所有的音频编辑软件都能识别 WAV 格式的文件。

2. 视频格式

(1) OGG

OGG 也是 HTML5 所使用的视频格式之一。OGG 采用多通道编码技术, 可以在保持编码器的灵活性的同时而不损害原本的立体声空间影像, 而且实现的复杂程度比传统的联合立体声方式要低。

(2) H.264 (MP4)

MP4 的全称是 MPEG-4 Part 14, 是一种储存数字音频和数字视频的多媒体文件格式, 文件扩展名为 .mp4。MP4 封装格式是基于 QuickTime 容器格式定义的, 媒体描述与媒体数据分开, 目前被广泛应用于封装 H.264 视频和 AAC 音频, 是高清视频的代表。

(3) WebM

WebM 由 Google 公司提出, 是一个开放、免费的媒体文件格式。WebM 影片格式其实是以 Matroska (即 MKV) 容器格式为基础开发的新容器格式, 包括 VP8 影片轨和 Ogg Vorbis 音轨。WebM 标准的网络视频更加偏向于开源并且是基于 HTML5 标准的, WebM 项目旨在为对每个人都开放的网络开发高质量、开放的视频格式, 其重点是解决视频服务这一核心的网络用户体验。

3.6.1 音频元素 audio

HTML5 提供了播放音频的标准。audio 元素能够播放声音文件或者音频流, 当前, audio 元素支持三种音频格式: OGG、MP3 和 WAV。audio 元素的格式为:

`<audio src="音频文件的 URL" controls="controls" ...>文本</audio>`

audio 元素的属性见表 3-3。<audio>与</audio>之间插入的文本是供不支持 audio 元素的浏览器显示的提示文字。

表 3-3 audio 元素的属性

属 性	值	描 述
src	url	要播放的音频的 URL
controls	controls	如果出现该属性, 则显示控件, 如播放按钮、暂停按钮、音量按钮
autoplay	autoplay	如果出现该属性, 则音频在就绪后马上播放
loop	loop	如果出现该属性, 则每当音频结束时重新开始播放
preload	preload	如果出现该属性, 则音频在页面加载时进行加载, 并预备播放。如果使用 autoplay="autoplay", 则忽略本属性

【例 3-8】 在网页中添加音频播放控件。本例文件 3-8.html 在浏览器中的显示效果如图 3-9 所示。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>audio</title>
  </head>
  <body>
    <audio src="邓丽君 - 甜蜜蜜.mp3" controls="controls">
      当前浏览器不支持 audio
    </audio>
  </body>
</html>
```

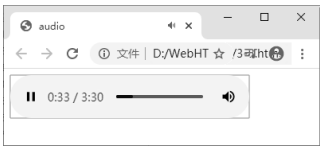


图 3-9 网页中的音频播放控件

3.6.2 视频元素 video

video 元素定义视频，比如电影片段或其他视频流。目前 video 元素支持三种视频格式：MP4、WebM、OGG。video 元素的格式为：

```
<video src="音频文件的 URL" controls="controls" ...>文本</video>
```

video 元素的属性见表 3-4。可以在<video>和</video>标签之间放置文本内容，这样不支持 video 元素的浏览器就可以显示出该标签的信息。

表 3-4 video 元素的属性

属 性	值	描 述
src	URL	要播放视频的 URL
controls	controls	如果出现该属性，则显示控件，如播放按钮等
width	像素值 pixels	设置视频播放器的宽度
height	像素值 pixels	设置视频播放器的高度
autoplay	autoplay	如果出现该属性，则视频在就绪后马上播放
loop	loop	如果出现该属性，则当媒介文件完成播放后再次开始播放
muted	muted	如果出现该属性，视频的音频输出为静音
poster	URL	规定视频正在下载时显示的图像，直到用户单击播放按钮
preload	auto、metadata 或 none	如果出现该属性，则视频在页面加载时进行加载，并预备播放。如果使用"autoplay"，则忽略该属性

【例 3-9】 网页中添加视频播放控件。本例文件 3-9.html 在浏览器中的显示效果如图 3-10 所示。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>video</title>
  </head>
  <body>
    <video src="邓丽君-我只在乎你.mp4" width="800" height="" controls="controls">
      当前浏览器不支持 video 直接播放，点击这里下载视频：<a href="myvideo.webm">
        下载视频</a>
    </video>
  </body>
</html>
```

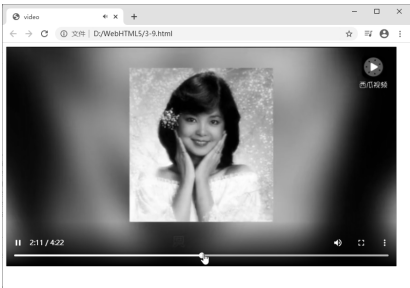


图 3-10 网页中的视频播放控件

3.7 用 HBuilder X 编辑 HTML 文档

上一章为了帮助读者理解 HTML 文档，采用记事本编辑 HTML 网页。为了提高效率，本章以后采用 HBuilder X 编辑 HTML 文档。用 HBuilder X 编辑 HTML 文档非常简单，只需简单几个步骤。下面以 HBuilder X 标准版为例介绍其操作步骤。



17 用 HBuilder X 编辑 HTML 文档

1) 在桌面上双击 HBuilder X 的快捷方式图标。

2) 打开 HBuilder X，如果是初次使用 HBuilder X，将显示“历次更新说明”，如图 3-11 所示。如果以前编辑过网页，将显示上次编辑的 HTML 文档，如图 3-12 所示。不需要则关闭该标签卡。



图 3-11 初次打开时的显示效果



图 3-12 显示上次编辑的 HTML 代码

3) 新建一个 HTML 文档，选择“文件”→“新建”→“7.html 文件”菜单命令，如图 3-13 所示。

4) 显示“新建 html 文件”对话框，如图 3-14 所示。在文件名框中输入 HTML 文档名，例如“3-7”，文件类型保持.html 不变。



图 3-13 新建 HTML 文档



图 3-14 “新建 html 文件”对话框

5) 单击“浏览”按钮，显示“选择文件夹”对话框，浏览到保存 HTML 文档的文件夹，例如“D:\WebHTML5”，单击“选择文件夹”按钮，如图 3-15 所示。返回“新建 html 文件”对话框，单击“创建”按钮，如图 3-16 所示。



图 3-15 “选择文件夹”对话框



图 3-16 返回“新建 html 文件”对话框

6) 显示代码编辑区，其中已经有 HTML5 的网页结构代码，如图 3-17 所示。在<title></title>标签之间单击，输入网页标题，例如“个人博客网站”。

7) 在标签<body>后按〈Enter〉键，插入空行，将自动缩进。输入标签后按〈Enter〉键，例如输入<h3></h3>标签，输入“h”，显示 h 开头的标签，接着输入第 2 个字符“3”，或者按〈↓〉键到 h3 或用鼠标选择“h3”选项，如图 3-18 所示，按〈Enter〉键，则该标签插入到当前位置。在<h3></h3>标签之间输入文字。

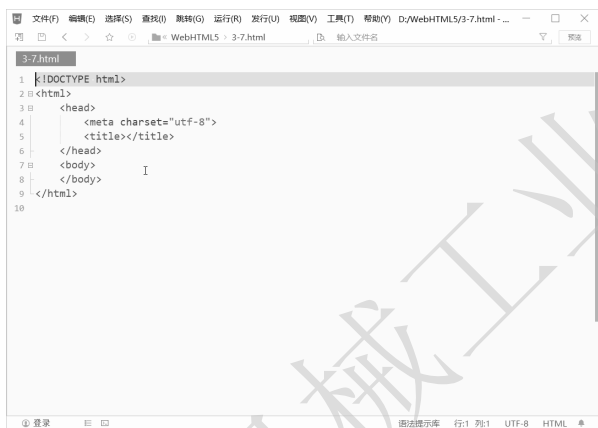


图 3-17 HTML5 网页结构代码

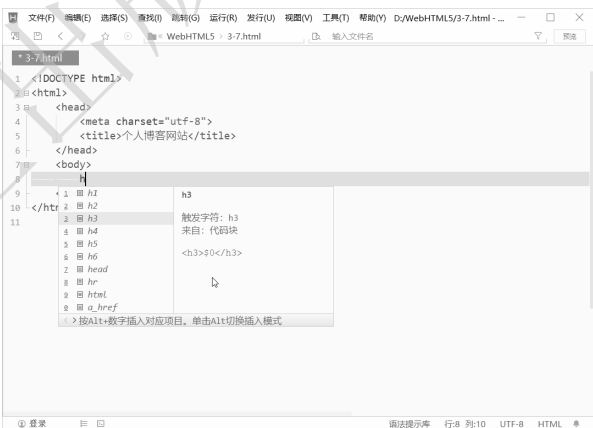


图 3-18 输入标签的第一个字母

8) 在<h3>...</h3>后面按〈Enter〉键插入一个空行。按〈p〉键，再按〈Enter〉键，插入<p></p>标签，如图 3-19 所示。在<p></p>标签之间输入“img”，如图 3-20 所示，按〈Enter〉键，则插入标签，鼠标指针在 src 后的双引号中，输入“D:\WebHTML5\images\blog.jpg”。

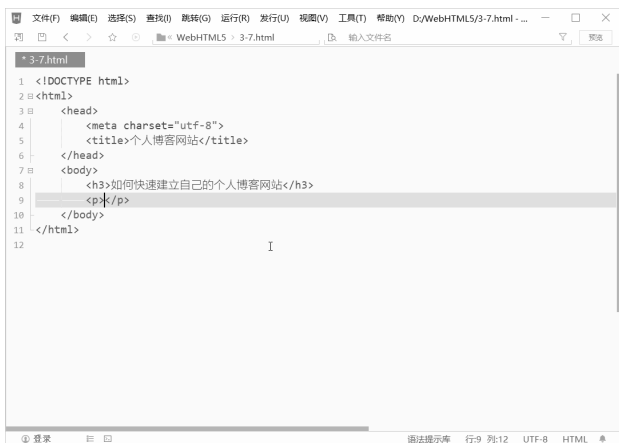


图 3-19 插入<p></p>标签



图 3-20 输入“img”

9) 选择“编辑”→“插入”→“向下插入空行”菜单命令，或者直接按〈Ctrl+Enter〉组合键插入空行。输入“p”后按〈Enter〉键，在<p></p>之间输入相关文字。

10) 如果文档中的缩进排列不整齐，在文档中右击，从快捷菜单中选择“重排代码格式”，如图 3-21 所示，或者直接按〈Ctrl+K〉组合键重排文档。

11) 单击窗口左上角的“保存”按钮，保存文件。

12) 选择“运行”→“运行到浏览器”→“Chrome”菜单命令，或者选择自己安装的浏览器，如图 3-22 所示。



图 3-21 快捷菜单



图 3-22 “运行”菜单

13) 运行结果显示在 Chrome 浏览器中，如图 3-23 所示。

HBuilder X 还有许多提高编辑效率的方法，读者可以在使用过程中逐步熟悉。

3.8 实训——制作广告版块

【实训 3-1】 制作社区网首页的广告版块。本实训文件 pt3-1.html 在浏览器中的显示效果如图 3-24 所示。



图 3-23 运行结果



图 3-24 广告版块

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
```

```

<title>社区网首页</title>
</head>
<body>
<div style="width: 1200px;margin: 0 auto;text-align: center;">
    
    <span style="font-size: 44px;color: #e95419;">和谐社区</span>
    
    <span style="font-size: 44px;color: #e95419;">诚信守法</span>
    
    <span style="font-size: 44px;color: #e95419;">交通出行</span>
    
    <span style="font-size: 44px;color: #e95419;">便民服务</span>
</div>
<div style="width: 1200px;margin: 0 auto;text-align: center;">
    <div>
        <h3>商家广告</h3>
    </div>
    <div>
        <a href="#"></a>
        <a href="#"></a>
        <a href="#"></a>
        <a href="#"></a>
        <a href="#"></a>
    </div>
    <div>
        <a href="#"></a>
        <a href="#"></a>
        <a href="#"></a>
        <a href="#"></a>
        <a href="#"></a>
    </div>
</div>
<div style="width: 1200px;margin: 0 auto;text-align: center;">
    <span><a href="">01 友情链接</a></span>|
    <span><a href="">02 友情链接</a></span>|
    <span><a href="">03 友情链接</a></span>|
    <span><a href="">04 友情链接</a></span>|
    <span><a href="">05 友情链接</a></span>|
    <span><a href="">06 友情链接</a></span>|
    <span><a href="">07 友情链接</a></span>|
    <span><a href="">08 友情链接</a></span>|
    <span><a href="">09 友情链接</a></span>|
    <span><a href="">10 友情链接</a></span>|
    <span><a href="">11 友情链接</a></span>|
    <span><a href="">12 友情链接</a></span><br />
    <span><a href="">13 友情链接</a></span>|
    <span><a href="">14 友情链接</a></span>|
    <span><a href="">15 友情链接</a></span>|
    <span><a href="">16 友情链接</a></span>|
    <span><a href="">17 友情链接</a></span>|
    <span><a href="">18 友情链接</a></span>|
    <span><a href="">19 友情链接</a></span>|
    <span><a href="">20 友情链接</a></span>|
    <span><a href="">21 友情链接</a></span>|

```

```

<span><a href="">22 友情链接</a></span>|
<span><a href="">23 友情链接</a></span>|
<span><a href="">24 友情链接</a></span>
</div>
</body>
</html>

```

【说明】对于复杂的页面，使用表格布局必须采用多层嵌套才能实现布局效果，但过多的表格嵌套将影响页面的打开速度。

习题 3

1. 使用列表元素和超链接元素制作如图 3-25 所示的网页。
2. 使用表格和列表制作如图 3-26 所示的清单。



图 3-25 列表和超链接

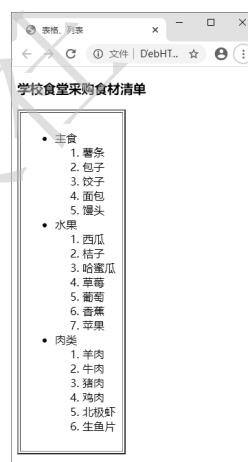


图 3-26 表格、列表

3. 使用图片元素和超链接元素制作如图 3-27 所示的网页。
4. 制作图文混排网页，如图 3-28 所示。

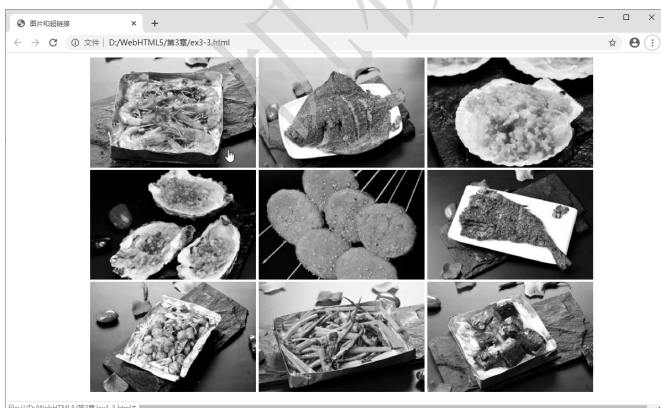


图 3-27 图片和超链接



图 3-28 图文混排网页