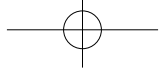


Web 前端开发实战： HTML5+CSS3+JavaScript+ Vue+Bootstrap (微视频版)

王莉莉 杨海陆 尹 芳 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,全面介绍了使用 HTML5、CSS3、JavaScript、Bootstrap、Vue 等技术搭建 Web 前端的方法和技巧,全书共分 20 章,内容涵盖了 HTML5 基础、文本和图像、音频和视频、列表和超链接、CSS3 基础、文本样式、图像和背景样式、表格、表单、CSS3 盒子模型、CSS3 移动布局、CSS3 变形和动画、JavaScript 基础、事件处理、BOM 和 DOM、Bootstrap 基础、CSS 通用样式、CSS 组件、JavaScript 插件和使用 Vue 等,力求为读者带来良好的学习体验。

与书中内容同步的案例操作教学资源可供读者随时扫码学习,读者可扫描案例旁边的二维码在线观看。本书具有很强的实用性和可操作性,可以作为初学者的自学用书,也可作为 Web 前端开发技术人员的首选参考书,还可作为高等院校 Web 前端开发、网站设计等相关专业的教材。

本书对应的电子课件、完整代码文档和实例源文件可以到 <http://www.tupwk.com.cn/downpage> 网站下载,也可以通过扫描前言中的二维码获取。

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。举报:010-62782989, beiqinquan@tup.tsinghua.edu.cn。

图书在版编目(CIP)数据

Web 前端开发实战: HTML5+CSS3+JavaScript+Vue+Bootstrap: 微视频版/王莉莉, 杨海陆, 尹芳编著.
—北京: 清华大学出版社, 2023.4

ISBN 978-7-302-63066-1

I. ①W… II. ①王… ②杨… ③尹… III. ①网页制作工具 IV. ①TP393.092.2

中国国家版本馆 CIP 数据核字 (2023) 第 045074 号

责任编辑: 胡辰浩

封面设计: 高娟妮

版式设计: 孔祥峰

责任校对: 成凤进

责任印制: 朱雨萌

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: 010-83470000 邮 购: 010-62786544

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

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

印 装 者: 大厂回族自治县彩虹印刷有限公司

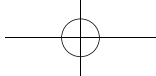
经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 22.25 字 数: 514 千字

版 次: 2023 年 6 月第 1 版 印 次: 2023 年 6 月第 1 次印刷

定 价: 98.00 元

产品编号: 097927-01



前言

—preface—

近些年来，随着“互联网+”、大数据、云计算、人工智能、机器学习等新名词、新概念的不断涌现，相关产业发展得如火如荼。移动互联网已经深入人们日常生活的方方面面，为了让用户有更好的互联网体验效果，Web前端开发和移动终端开发相关技术飞速发展。HTML5、CSS3、JavaScript、Bootstrap、Vue等技术相互配合使用大大提高了Web前端开发人员的工作效率，减轻了工作量，降低了前端开发成本。本书设计了大量的示例，旨在帮助读者尤其是初学者快速掌握Web前端开发的技术精髓，用最短的时间使前端设计从外观上变得更加精美实用。

一、本书内容特点

○ 内容合理，适合自学

本书总结了编者团队多年的设计经验及教学心得，在编写时充分考虑到初学者的特点，内容设置由浅入深、循序渐进、案例丰富，力求全面细致地展现Web前端设计应用领域的各项功能，使读者能够通过书中的示例掌握前端设计工作中需要的各项技术。

○ 案例专业，资源丰富

本书中的大部分案例源于实际工作设计项目。为了提高读者的学习效率，书中为大部分的示例代码配备了相应的完整源代码和扩展知识(可扫描案例旁边的二维码在线观看，也可下载观看)，详细讲解了HTML5、CSS3、JavaScript、Bootstrap和Vue的技术要点，并在知识点的关键处给出了解释、提示和注释，可以帮助读者大大提高阅读本书的效率，快速掌握相关内容。

○ 知行合一，通俗易懂

本书结合Web前端设计中的实际案例，详细讲解各类技术的实战技巧。读者可以通过案例，快速掌握Web前端设计的基础知识，同时提升自身的设计实践能力。每个案例都是专业知识和经验的提炼，让读者在学习能够更快、更容易地理解所学的内容，并且能够独立高效地完成各种设计任务。

二、本书内容简介

本书全面细致地介绍了HTML5、CSS3、JavaScript、Bootstrap和Vue的相关技术知识，全书共分20章，各章内容简介如下。

章 节	内容说明
第1章 HTML5基础	主要介绍HTML5的基础知识
第2章 文本和图像	主要介绍在网页中添加文本和图像的方法
第3章 音频和视频	主要介绍在网页中添加音频和视频的方法
第4章 列表和超链接	主要介绍在网页中设计列表和超链接的方法
第5章 CSS3基础	主要介绍CSS3的基础知识
第6章 文本样式	主要介绍定义网页文本样式
第7章 图像和背景样式	主要介绍定义网页图像和背景样式
第8章 表格	主要介绍在网页中设计表格和表格样式的方法
第9章 表单	主要介绍在网页中设计表单、表单控件和表单属性的方法
第10章 CSS3盒子模型	主要介绍CSS3盒子模型
第11章 CSS3移动布局	主要介绍CSS3移动布局
第12章 CSS3变形和动画	主要介绍CSS3变形和动画效果
第13章 JavaScript基础	主要介绍JavaScript的基础知识
第14章 事件处理	主要介绍JavaScript事件与事件处理
第15章 BOM和DOM	主要介绍BOM和DOM知识
第16章 Bootstrap基础	主要介绍Bootstrap的基础知识
第17章 CSS通用样式	主要介绍CSS通用样式
第18章 CSS组件	主要介绍常用CSS组件
第19章 JavaScript插件	主要介绍JavaScript插件
第20章 使用Vue	主要介绍Vue.js的基础知识和使用方法

三、本书配套资源及服务

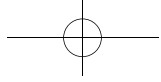
与书中内容同步的案例操作教学资源可供读者随时扫码学习，读者可扫描书中案例旁边的二维码在线观看。此外，本书免费提供电子课件、完整代码文档和实例源文件，读者可以扫描下方二维码获取，也可以进入本书信息支持网站(<http://www.tupwk.com.cn/downpage>)下载。



扫码推送配套资源到邮箱

本书由哈尔滨理工大学的王莉莉、杨海陆和尹芳合作编写完成，其中王莉莉编写了第1、2、3、4、6、13、14、15、19、20章，杨海陆编写了第5、7、16、17、18章，尹芳编写了第8~12章。由于作者水平有限，本书难免有不足之处，欢迎广大读者批评指正。我们的邮箱是992116@qq.com，电话是010-62796045。

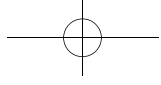
编 者
2022年12月



目 录

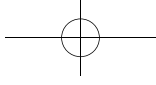
—contents—

第 1 章	HTML5基础	1		
1.1	HTML5概述	2		
1.1.1	HTML5发展历程	2		
1.1.2	HTML5设计理念	2		
1.2	新建HTML5文档	4		
1.2.1	创建空白文档	4		
1.2.2	添加网页内容	5		
1.3	头部信息	7		
1.3.1	网页标题	7		
1.3.2	网页元信息	7		
1.3.3	文档视口	8		
1.4	基本结构	10		
1.4.1	定义文档结构	10		
1.4.2	使用div元素	11		
1.4.3	使用id和class	12		
1.4.4	使用title	12		
1.4.5	使用role	13		
1.4.6	HTML5注释	13		
1.5	主体结构	14		
1.5.1	页眉	14		
1.5.2	导航	14		
1.5.3	主要区域	15		
1.5.4	文章块	15		
1.5.5	区块	16		
1.5.6	附栏	16		
1.5.7	页脚	17		
第 2 章	文本和图像	18		
2.1	标题和段落	19		
2.1.1	定义标题	19		
2.1.2	定义段落	20		
2.2	文本格式	20		
2.2.1	字体	20		
2.2.2	字号	21		
2.2.3	颜色	22		
2.2.4	强调	23		
2.2.5	注解	23		
2.2.6	备选	24		
2.2.7	上下标	24		
2.2.8	术语	25		
2.2.9	代码	26		
2.2.10	预定义格式	26		
2.2.11	缩写词	27		
2.2.12	编辑提示	27		
2.2.13	引用	28		
2.2.14	引述	28		
2.2.15	修饰	29		
2.2.16	换行显示	30		
2.3	特殊效果	30		
2.3.1	高亮	30		
2.3.2	进度	31		
2.3.3	刻度	31		
2.3.4	时间	32		
2.3.5	联系信息	33		
2.3.6	旁注	34		
2.4	网页图像	34		
2.4.1	定义图像	34		
2.4.2	定义流	35		
2.4.3	定义图标	36		
2.4.4	定义响应式图像	36		
第 3 章	音频和视频	37		
3.1	使用audio元素	38		
3.2	使用video元素	39		
3.3	使用embed元素	40		
3.4	使用object元素	41		
第 4 章	列表和超链接	43		
4.1	列表	44		
4.1.1	无序列表	44		



4.1.2 有序列表	44	6.2 文本格式	73
4.1.3 项目编号	45	6.2.1 对齐	73
4.1.4 嵌套列表	46	6.2.2 间距	74
4.1.5 描述列表	47	6.2.3 行高	75
4.1.6 菜单列表	47	6.2.4 缩进	75
4.2 超链接	48	6.2.5 换行	75
4.2.1 页间链接	48	6.3 书写模式	77
4.2.2 块级链接	49	6.4 特殊值	78
4.2.3 锚记链接	50	6.5 文本效果	80
4.2.4 目标链接	50	6.5.1 文本阴影	81
4.2.5 邮件链接	51	6.5.2 文本特效	82
4.2.6 下载链接	51	6.6 颜色模式	84
4.2.7 图像热点	51	6.6.1 RGBA	84
第5章 CSS3基础	53	6.6.2 HSL	85
5.1 CSS3概述	54	6.6.3 HSLA	85
5.1.1 CSS3样式	54	6.6.4 opacity	86
5.1.2 应用CSS3样式	55	6.7 动态内容	86
5.1.3 CSS3样式表	55	第7章 图像和背景样式	90
5.1.4 CSS3代码注释	57	7.1 美化图像	91
5.1.5 CSS3代码格式化	57	7.1.1 图像大小	91
5.1.6 CSS3继承性	57	7.1.2 图像边框	92
5.1.7 CSS3层叠性	58	7.1.3 半透明图片	92
5.2 CSS3选择器	59	7.1.4 图像圆角	93
5.2.1 元素选择器	59	7.1.5 图像阴影	94
5.2.2 关系选择器	61	7.2 背景图像	95
5.2.3 属性选择器	63	7.2.1 定义背景图像	95
5.2.4 伪类选择器	64	7.2.2 定义背景原点/位置/裁剪	96
5.2.5 伪对象选择器	67	7.2.3 控制背景图像显示大小	97
第6章 文本样式	68	7.2.4 固定背景图像	99
6.1 字体样式	69	7.3 渐变背景	100
6.1.1 字体	69	7.3.1 线性渐变	100
6.1.2 大小	70	7.3.2 径向渐变	102
6.1.3 颜色	71	第8章 表格	105
6.1.4 粗体	71	8.1 定义表格	106
6.1.5 斜体	71	8.1.1 普通表格	106
6.1.6 修饰线	71	8.1.2 列标题	106
6.1.7 变体	72	8.1.3 表格标题	107
6.1.8 大小写	73		

8.1.4 表格行/列分组	107	9.3.10 高和宽	131
8.2 表格属性	108	9.3.11 最小值/最大值/步长	132
8.2.1 内/外框线	108	9.3.12 匹配模式	132
8.2.2 单元格间距	109	9.3.13 替换文本	132
8.2.3 细线边框	110	9.4 表单样式与定制表单	133
8.2.4 内容摘要	110		
8.3 单元格属性	110	第 10 章 CSS3 盒子模型	134
8.3.1 跨单元格显示	110	10.1 显示方式	135
8.3.2 单元格表头	111	10.2 可控大小	136
8.3.3 绑定表头	111	10.3 内容溢出	137
8.3.4 信息缩写	112	10.4 轮廓线	137
8.3.5 单元格分类	112	10.5 圆角边框	138
8.4 表格样式	113	10.6 图像边框	139
第 9 章 表单	116	10.7 盒子阴影	140
9.1 定义表单	117	10.8 布局方式	142
9.1.1 设计表单结构	117	10.8.1 流式布局	142
9.1.2 表单对象分组	117	10.8.2 浮动布局	143
9.1.3 添加提示文本	119	10.8.3 定位布局	144
9.2 表单控件	119	第 11 章 CSS3 移动布局	146
9.2.1 文本框	120	11.1 多列布局	147
9.2.2 密码框	121	11.1.1 定义列宽	147
9.2.3 文本区域	121	11.1.2 定义列数	148
9.2.4 单选按钮	122	11.1.3 定义列间距	149
9.2.5 复选框	122	11.1.4 定义列边框	149
9.2.6 选择框	123	11.1.5 定义跨列显示	150
9.2.7 文件域和隐藏域	124	11.1.6 定义列高度	150
9.2.8 按钮	124	11.2 盒布局模型	151
9.2.9 数据列表	125	11.2.1 定义宽度	151
9.3 表单属性	126	11.2.2 定义顺序	152
9.3.1 名称和值	126	11.2.3 定义方向	153
9.3.2 布尔型属性	126	11.2.4 定义自适应	153
9.3.3 必填属性	127	11.2.5 定义对齐方式	153
9.3.4 禁止验证	128	11.3 弹性盒布局	155
9.3.5 多选属性	129	11.3.1 定义弹性盒	155
9.3.6 自动完成	129	11.3.2 定义伸缩方向	156
9.3.7 自动获取焦点	130	11.3.3 定义行数	157
9.3.8 所属表单	131	11.3.4 定义对齐方式	158
9.3.9 表单重写	131	11.3.5 定义伸缩项目	159



11.4 媒体查询	161
-----------------	-----

第12章 CSS3变形和动画 164

12.1 变形	165
12.1.1 2D旋转	165
12.1.2 2D缩放	166
12.1.3 2D移动	167
12.1.4 2D倾斜	167
12.1.5 2D矩阵	168
12.1.6 变形原点	168
12.1.7 3D变形	169
12.1.8 3D位移	170
12.1.9 3D缩放	171
12.1.10 3D旋转	172
12.2 过渡样式	174
12.2.1 定义过渡	175
12.2.2 定义过渡时间	175
12.2.3 定义延时	176
12.2.4 定义动画效果	176
12.2.5 定义触发时机	177
12.3 关键帧动画	178
12.3.1 定义关键帧	178
12.3.2 定义帧动画	179

第13章 JavaScript基础 181

13.1 JavaScript概述	182
13.1.1 JavaScript的主要特点	182
13.1.2 JavaScript的基本语法	182
13.1.3 JavaScript在HTML中的使用	183
13.2 JavaScript程序	185
13.2.1 语句和语句块	185
13.2.2 代码	186
13.2.3 消息框	186
13.2.4 JavaScript注释	188
13.3 标识符和变量	188
13.3.1 标识符	188
13.3.2 变量	189
13.4 数据类型	191
13.5 运算符	191

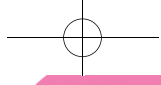
13.5.1 算术运算符	191
13.5.2 比较运算符	192
13.5.3 赋值运算符	192
13.5.4 逻辑运算符	193
13.5.5 条件运算符	193
13.5.6 其他运算符	194
13.5.7 运算符优先级	195
13.6 表达式和赋值语句	196
13.6.1 表达式	196
13.6.2 赋值语句	196
13.7 流程控制语句	196
13.7.1 条件判断语句	197
13.7.2 循环语句	201
13.7.3 跳转语句	203
13.8 JavaScript函数	204
13.8.1 常用系统函数	204
13.8.2 自定义函数	205
13.8.3 带参数返回的return语句	206
13.8.4 函数变量的作用域	206

第14章 事件处理 207

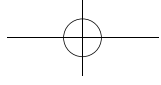
14.1 JavaScript事件概述	208
14.1.1 事件类型	208
14.1.2 事件句柄	208
14.1.3 事件处理	209
14.1.4 程序返回值	212
14.2 表单事件	212
14.2.1 获取与失去焦点	212
14.2.2 提交与重置	213
14.2.3 改变与选择	215
14.3 鼠标事件	216
14.4 键盘事件	216
14.5 窗口事件	217

第15章 BOM和DOM 218

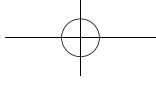
15.1 JavaScript常用对象	219
15.1.1 Array对象	219
15.1.2 Date对象	221
15.1.3 Math对象	223



15.1.4	Number对象	224	16.7.3	图像对齐	258
15.1.5	String对象	225	16.8	表格样式	258
15.1.6	Boolean对象	226	第 17 章	CSS通用样式	259
15.2	BOM	227	17.1	文本处理	260
15.2.1	window对象	228	17.1.1	文本对齐	260
15.2.2	navigator对象	229	17.1.2	文本换行	261
15.2.3	screen对象	229	17.1.3	大小写转换	261
15.2.4	history对象	230	17.1.4	粗体和斜体	262
15.2.5	location对象	230	17.1.5	其他文本样式类	263
15.3	DOM	231	17.2	颜色样式	263
15.3.1	DOM节点树	232	17.2.1	文本颜色	263
15.3.2	DOM节点	232	17.2.2	链接颜色	264
15.3.3	DOM节点访问	234	17.2.3	背景颜色	264
15.3.4	DOM节点操作	237	17.3	边框样式	264
第 16 章	Bootstrap基础	239	17.3.1	添加边框	264
16.1	Bootstrap概述	240	17.3.2	边框颜色	265
16.1.1	什么是Bootstrap	240	17.3.3	圆角边框	265
16.1.2	Bootstrap的优势	240	17.4	宽度和高度	266
16.1.3	Bootstrap的构成模块	240	17.4.1	相对于父元素	266
16.2	下载与安装	241	17.4.2	相对于视口	267
16.2.1	下载Bootstrap	241	17.5	边距	268
16.2.2	安装Bootstrap	242	17.5.1	定义边距	268
16.3	开发工具	243	17.5.2	响应式边距	269
16.4	基本架构	244	17.6	浮动样式	269
16.4.1	文件结构	244	17.6.1	实现浮动样式	270
16.4.2	布局基础	245	17.6.2	响应式浮动样式	270
16.4.3	网格系统	248	17.7	display属性	271
16.4.4	布局工具类	253	17.7.1	隐藏/显示元素	271
16.5	排版优化	254	17.7.2	响应式隐藏/显示元素	271
16.5.1	标题	254	17.8	嵌入网页元素	272
16.5.2	段落	255	17.9	内容溢出	273
16.5.3	强调/对齐/列表	256	17.10	定位元素	273
16.6	显示代码	256	17.11	阴影效果	274
16.6.1	行内代码	256	第 18 章	CSS组件	275
16.6.2	多行代码块	257	18.1	按钮和按钮组	276
16.7	响应式图片	257	18.1.1	按钮	276
16.7.1	同步缩放	257	18.1.2	按钮组	278
16.7.2	缩略图	257			



18.2	下拉菜单	281	19.3	下拉菜单	299
18.2.1	设计分裂式按钮样式	282	19.3.1	调用下拉菜单	300
18.2.2	设计菜单展开方向	282	19.3.2	设置下拉菜单	301
18.2.3	设计菜单分割线	282	19.3.3	为下拉菜单添加用户行为	301
18.2.4	激活/禁用菜单项	283	19.4	弹窗	301
18.2.5	设计菜单项对齐方式	283	19.4.1	设计弹窗方向	303
18.2.6	设计菜单偏移量	283	19.4.2	调用弹窗	303
18.2.7	设计更多菜单内容	283	19.4.3	为弹窗添加用户行为	304
18.3	导航组件	284	19.5	工具提示	304
18.3.1	设计水平对齐布局	285	19.5.1	工具提示方向	305
18.3.2	设计垂直对齐布局	285	19.5.2	调用工具提示	306
18.3.3	设计标签页导航	285	19.5.3	为工具提示添加用户行为	307
18.3.4	设计胶囊式导航	286	19.6	标签页	307
18.3.5	设计导航内容填充和对齐	286	19.6.1	调用标签页	308
18.3.6	设计导航选项卡	286	19.6.2	为标签页添加用户行为	309
18.4	超大屏幕	287	19.7	其他JavaScript插件	310
18.5	其他CSS组件	288	19.7.1	按钮	310
18.5.1	徽章	288	19.7.2	警告框	310
18.5.2	警告框	288	19.7.3	折叠	311
18.5.3	媒体对象	289			
18.5.4	进度条	290			
18.5.5	导航栏	291			
第 19 章 JavaScript插件			第 20 章 使用Vue	313	
19.1	JavaScript插件概述	293	20.1	Vue概述	314
19.1.1	安装插件	293	20.1.1	Vue.js产生的背景	314
19.1.2	调用插件	294	20.1.2	Vue.js的安装和导入	314
19.1.3	事件	295	20.2	Vue实例和组件	315
19.2	模态框	295	20.2.1	Vue实例	315
19.2.1	设计模态框垂直居中	297	20.2.2	Vue组件	316
19.2.2	设计模态框大小	298	20.3	Vue模板语法	324
19.2.3	添加模态框和工具提示	298	20.3.1	插值表达式	324
19.2.4	调用模态框	298	20.3.2	指令	327
19.2.5	为模态框添加用户行为	299	20.4	Vue的data属性	337
			20.5	Vue方法	338
			20.6	Vue计算属性	340
			20.7	Vue监听器	341



第 1 章

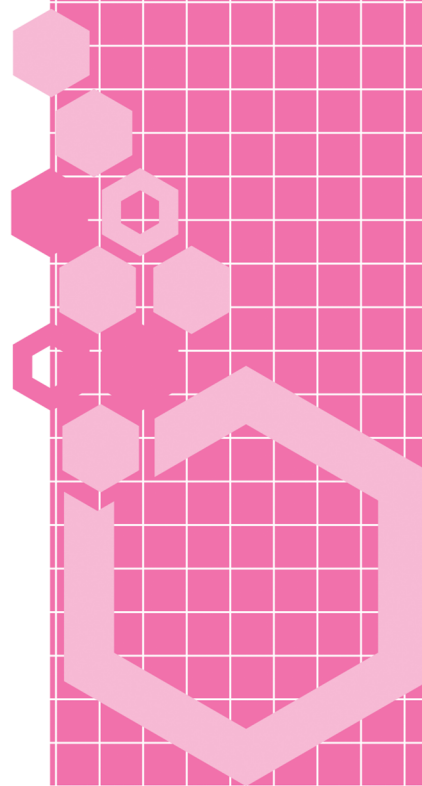
HTML5基础

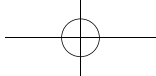
内容简介

HTML5是超文本语言HTML的第5次修订，是近些年来Web标准的巨大飞跃。HTML5和以前版本HTML不同的是，HTML5并非仅仅用于表示Web内容，而且为Web的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。

学习重点

- HTML5的发展历程
- 创建HTML5文档的方法
- HTML5的基本结构
- HTML5的主体结构





1.1 HTML5概述

HTML是用来描述网页的一种语言。该语言是一种标记语言(即一套标记标签，HTML使用标记标签来描述网页)，而不是编程语言。它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。

1.1.1 HTML5发展历程

HTML的发展历史可以追溯到很久以前。1993年，HTML首次以因特网草案的形式发布。20世纪90年代，人们见证了HTML的快速发展，从HTML1.0版、HTML2.0版，到HTML3.2版和HTML4.0版，再到1999年的HTML4.01版，一直发展到HTML5，如图1-1所示。

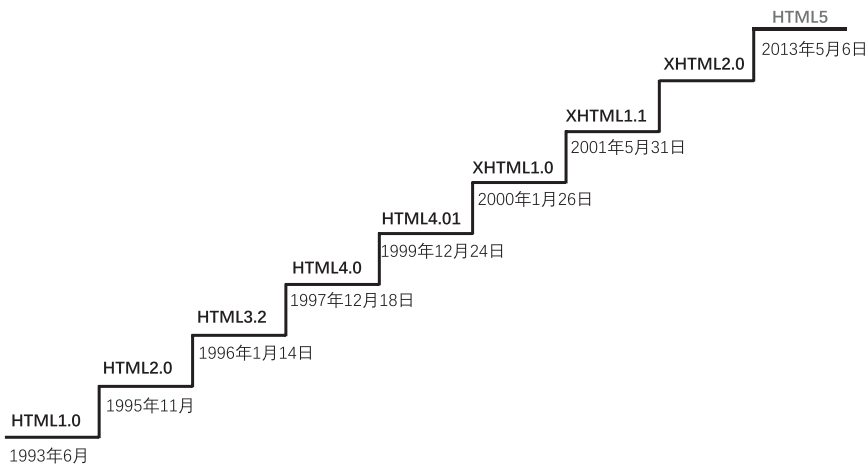


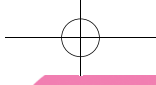
图 1-1 HTML5 的发展历程

HTML5相比之前的版本提供了一些新的元素和一些有趣的新特性，同时也建立了一些新的规则。这些元素、特性和规则提供了许多新的网页功能，例如使用网页实现动态渲染图形、图表、图像和动画，以及在不需要安装任何插件的情况下，直接使用网页播放视频等。

1.1.2 HTML5设计理念

HTML5是一个里程碑式的规范，它为下一代Web发明指明了方向，其设计理念囊括了诸如兼容性、实用性、互用性之类的概念，即便W3C(万维网联盟)与WHATWG(网页超文本应用技术工作组)之间有多大的分歧，它们也都遵循HTML5的设计理念。

(1) 化繁为简。HTML5为了尽可能简化，避免了一些不必要的复杂设计。如doctype的改进，在以往的HTML版本中，第一行doctype过于复杂，几乎没有几个人能够记住，在实



际的Web开发中也没有意义，在HTML5中就非常简单：

```
<!DOCTYPE html>
```

为了让一切变得简单，HTML5对每个细节都有非常明确的规范说明，不允许有任何歧义和模糊出现。

(2) 向下兼容。HTML5有很强的兼容能力，其允许存在不严谨的写法，例如一些属性的属性值没有用引号括起来，标签属性中包含大写字母，有的标签没有闭合等，对于这些不严谨的错误处理方案，在HTML5的规范中都有着明确的规定。

对于HTML5的一些新特性，如果浏览器不支持，也不会影响网页的显示。例如，HTML5中的input标签的type属性增加了很多类型，如果浏览器不支持这些类型，则默认会将其视为text，实现了优雅降级。

(3) 支持合理存在的内容。HTML5的设计者们花费了大量的精力来研究通用行为，例如，谷歌公司分析了上百万的页面，从中提取了div标签的id名称，如很多人采用以下方法标记导航区域：

```
<div class="nav">  
  <!--导航区域-->  
</div>
```

既然这样的行为已经大量存在，HTML5就会进行改进，增加nav标签，用于导航区域。

(4) 解决现实问题。对于HTML5无法实现的一些功能，将会寻求其他方法来实现。例如，绘图、多媒体、地理位置、实时获取信息等功能，通常会开发一些相应的插件间接实现这些功能。HTML5的设计者专门研究了这些功能需求，开发了一系列用于Web应用的接口。

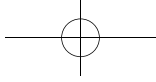
HTML5规范的制定是开放的，所有人都可以获取草案的内容，也可以参与规范的制定，提出自己的意见。因为其开放性，所以HTML5可以得到更加全面的发展，一切以用户需求为最终目的，用户需要什么，就规范什么内容。

(5) 最终用户优先。这在本质上是一种解决冲突的机制。在遇到无法解决的冲突时，HTML5规范会把最终用户的诉求放在第一位。因此，HTML5的绝大部分功能都非常实用。HTML5规范的制定遵循以下优先顺序：用户>编写HTML的开发者>浏览器厂商>规范制定者>纯粹的理念。

(6) 通用访问。HTML5的通用访问原则包括以下3方面。

- 可访问性：HTML5考虑了残障用户的需求，以屏幕阅读器为基础的元素已经被添加进规范中。
- 媒体中立：HTML5规范不仅仅是为了某些浏览器而设计的。HTML5的新功能可能未来的某一天在不同设备和平台上都能够运行。
- 支持所有的语种：HTML5规范支持所有的语种。

下面我们将从新建HTML5文档开始，介绍HTML5的基础知识。



1.2 新建HTML5文档

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它的开发环境。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，可以在浏览器中打开文档以查看其效果。

1.2.1 创建空白文档

一个完整的HTML文档包括头部信息(<head>)和主体内容(<body>)。为了使网页内容清晰、明确，并容易被他人阅读，或者被浏览器及各种设备所理解，新建HTML5文档后需要构建基本的网页结构。

【例1-1】使用Windows系统自带的记事本工具编写一个名为index.html的空白网页，其标题为“网页标题”，如图1-2所示。

- (1) 使用记事本新建一个文本文件，并将其保存为index.html。
- (2) 在文本文件中输入以下多行字符，如图1-2左图所示。
- (3) 保存文件，双击index.html使用浏览器打开网页后，效果如图1-2右图所示。



```
index.html - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>网页标题</title>
</head>
<body>
</body>
</html>
```

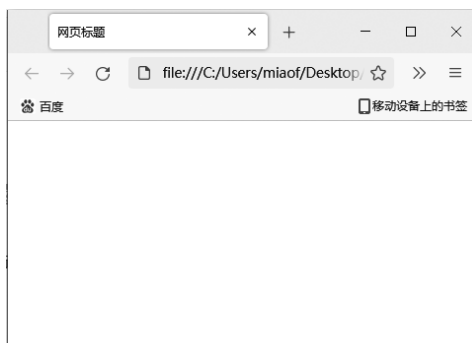


图 1-2 空白网页文档 index.html

【示例解析】

通过上面的示例代码，可以看到空白网页是由固定的结构构建的。图1-2右图所示的页面是一个空白页面，是因为访问者看到的内容主体部分(即<body>和</body>之间的部分)没有内容，是空的。

每个网页都包含doctype、html、head和body元素，它们是网页的基础。在上面的示例页面中，用户可以定制的内容包括两项：一项是设置lang属性的语言代码；另一项是<title>和</title>之间的文字。HTML使用“<”和“>”字符包含HTML标签，开始标签(如<head>)用于标记元素的开始，结束标签(如</head>)用于标记元素的结束。有的元素没有结束标签，如meat。

网页内容位于主体部分，<body>开始标签以上的内容都是为浏览器和搜索引擎准备的。<!doctype html>部分(简称为doctype)告诉浏览器这是一个HTML5页面，其应该始终位

于代码的第一行，写在HTML页面的顶部。

html元素包含着页面的其余部分，即<html>开始标签和</html>结束标签(表示页面的结尾)之间的内容。

<head>和</head>标签之间的区域表示网页文档的头部。头部代码中，有一部分是浏览器可见的，即<title>和</title>之间的文本。这些文本会出现在浏览器标签页中。某些浏览器会在窗口的顶部显示这些文本，作为网页的标题显示。此外，头部文本通常还是浏览器书签的默认名称，它们对搜索引擎来说是非常重要的信息。

1.2.2 添加网页内容

通常，网页都包括以下3部分内容。

- 文本：网页中的纯文字，如新闻资讯、产品介绍、注册声明等。
- 外部引用：包括图像、视频、音频、动画、样式表(控制页面显示效果)和JavaScript文件(为页面添加行为)等引用。这些引用还可以指向其他HTML页面和资源。
- 标记：对文本内容进行描述，确保浏览器能够正确显示。

【示例代码】

【例1-2】继续例1-1的操作，在<body>和</body>之间编写代码，在空白网页文档中添加如图1-3所示的文本和图片内容。

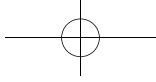
```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>网站登录界面</title>
</head>
<body>
  <article>
    <h1>使用手机扫码登录</h1>
    
    <p>请扫描<em>二维码</em>登录网站查看<a href="https://wx.qq.com" rel="external" title=
      "登录网站">网站信息</a></p>
  </article>
</body>
</html>
```



图 1-3 在网页中添加内容

【示例解析】

以上代码包含了3部分：功能介绍、外部文件引用(图像的src值和链接的href值)和标记。HTML提供了很多元素，本例演示了6种常见的元素，分别是a、article、em、h1、img、p，每个元素都有各自的含义，如h1是标题，img是图像，a是链接。



【知识点滴】

一般情况下，在编写代码时，在代码行与行之间使用回车符分开，不会影响页面的显示效果。对HTML进行代码缩进，与内容在浏览器中的显示效果没有任何关系。但要注意的是：`pre`元素是一个例外，在习惯上，需要对嵌套结构的代码进行缩进排版，以显示元素之间的层级关系。

此外，在编写HTML5文档时，也可以通过省略一些标签，简化HTML5文档。

【示例代码】

【例1-3】编写一个遵循HTML5语法规则的网页文档(效果如图1-4所示)，省略`<html>``<head>``<body>`等标签，使用HTML5的doctype声明文档类型，简化`<meta>`的charset属性设置，省略`<p>`标签的结束标记，使用`<元素>`的方式来结束`<meta>`和`
`标签。



```
<!doctype html>
<meta charset="utf-8">
<title>科普HTML5</title>
<h1>HTML5简介</h1>
<p>HTML5 是 HyperText Markup Language 5 的缩写。
<br/>HTML5 由不同的技术构成，其在互联网中得到了非常广泛
了非常广泛的应用，提供更多增强网络应用的标准集。与传统的
技术相比，HTML5 的语法特征更加明显，并且结合了 SVG 的内容。
这些内容在网页中使用可以更加便捷地处理多媒体内容，而且HTML5
中还结合了其他元素，对原有的功能进行调整和修改，进行标准化工作。
```



图 1-4 简化 HTML5 文档

【示例解析】

本例通过短短几行代码就完成了网页的设计，这说明了HTML5语法的简洁。HTML5不是一种XML(Extensible Markup Language，可扩展标记语言)，其语法可以十分随意，例如第1行代码：

```
<!doctype html>
```

不需要包括版本号，仅告诉浏览器一个doctype来触发标准模式即可。

接下来说明文档的字符编码(如果没有这一行，浏览器不能正确解析网页)：

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

HTML5不区分大小写，不需要标记结束符，不介意属性值是否加引号，以上代码和以下代码同等有效。

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

```
<META charset=UTF-8/>
```

同时，在网页主体中，可以省略主体标记，直接编写需要显示的内容。在例1-3编写的代码中虽然省略了`<html>``<head>``<body>`标记，但在浏览器进行解析时，将会自动进行添

加(考虑代码的可维护性,在编写代码时应尽量增加这些基本结构标签)。

1.3 头部信息

HTML文档的头部区域用于说明文档头部的相关信息(包括网页标题、网页元信息、文档视口参数等),这些信息主要被浏览器和搜索引擎所采用,不会显示在网页中。

1.3.1 网页标题

HTML页面的标题一般用来说明页面的用途,它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。

【示例代码】

【例1-4】使用<title>标签定义一个名为“HTML5新特性”的网页标题,其在浏览器中的显示效果如图1-5所示。

```
<html>
<head>
<title>HTML5新特性</title>
</head>
<body>
    HTML5智能表单
</body>
</html>
```

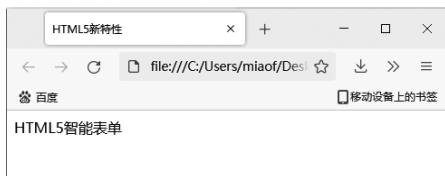


图 1-5 网页标题

【示例解析】

以上代码在浏览器中显示的效果如图1-5所示。浏览器将会把网页标题放在其标题栏或状态栏中显示。当用户将文档加入链接列表、收藏夹或书签列表时,网页标题将作为文档链接的默认名称。

title元素必须位于head部分。页面标题会被百度、Google等搜索引擎采用,从而使搜索引擎能够大致了解页面内容,并将页面标题作为搜索结果中的链接显示。

1.3.2 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和关键词。

○ 定义网页描述信息

```
<meta name="Description" content="HTML5网页设计技术" />
```

○ 定义网页关键词:

```
<meta name="Keywords" content="HTML,CSS,JavaScript,Bootstrap,Vue.js" />
```

<meta>标签位于文档头部,标签内不包含任何内容。使用<meta>标签的属性,可以定义与文档相关联的名称/值对。<meta>标签提供的属性及取值说明如表1-1所示。

表1-1 <meta>标签提供的属性及取值说明

属性	取值	说明
charset	character encoding	定义文档的字符编码
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属性关联到一个名称

下面列举常用元信息的设置代码。

(1) 设置UTF-8编码。

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

HTML5简化了字符编码设置方式：<meta charset="UTF-8">，其作用与上面代码是相同的。

(2) 设置GB2312编码。

```
<meta http-equiv="Content-Type" content="text/html"; charset="GB2312" />
```

每个HTML文档都需要设置字符编码类型，否则网页在最终显示时可能会出现乱码现象。上面介绍的两种编码形式中，UTF-8是国家通用编码，独立于任何语言，因此都可使用。

使用refresh属性值可以设置页面刷新时间或跳转页面，如8秒之后刷新页面。

```
<meta http-equiv="refresh" content="8; url=https://www.baidu.com" />
```

使用expires属性值可以设置网页缓存时间。

```
<meta http-equiv="expires" content="Sunday 25 September 2022 12:00 GMT" />
```

类似设置还有以下内容。

- 设置网页作者名称：<meta name="author" content="https://www.baidu.com"/>。
- 设置网页版权：<meta name="copyright" content="https://www.baidu.com"/>。
- 设置网页创建时间：<meta name="date" content="2022-12-19T18:17:31+00:00"/>。

1.3.3 文档视口

在移动Web开发中，经常需要定义viewport(视口)，即浏览器显示页面内容的屏幕区域。一般移动端浏览器都默认设置一个<meta name="viewport">标签，定义一个虚拟的布

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等系统基本都将其视口分辨率设置为980px。因此，PC端网页基本能够在手机端上呈现，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; ">
```

【示例代码】

【例1-5】编写代码，在页面中输入一个标题和两段文本，如果没有设置文档视口，则在移动设备中所呈现的效果如图1-6左图所示；如果设置了文档视口，则呈现图1-6右图所示的效果(扫码可查阅完整代码)。



```
<!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……</p>
<p>20世纪90年代HTML有过几次快速发展……</p>
</body>
</html>
```

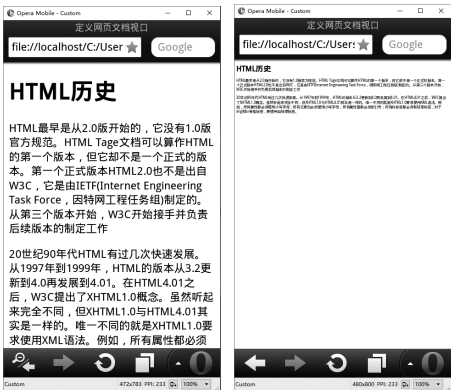


图 1-6 定义文档视口

【属性说明】

<meta name="viewport">标签中各属性说明如表1-2所示。

表1-2 <meta name="viewport">标签的属性说明

属性	取值	说明
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]	定义缩小最小比例，其须小于或等于maximum-scale设置
maximum-scale	[0.0-10.0]	定义放大最大比例，其须大于或等于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)

1.4 基本结构

HTML文档的主体部分包括了要在浏览器中显示的所有信息。这些信息需要在特定的结构中呈现。

1.4.1 定义文档结构

HTML5包含一百多个标签(大部分继承自HTML4)，这些标签基本上都被放置在主体区域内(<body>)，下面是常用的一些标签。

- <h1>、<h2>、<h3>、<h4>、<h5>、<h6>：用于定义文档标题。
- <p>：用于定义段落文本。
- 、、等：用于定义信息列表、导航列表、榜单结构等。
- <table>、<tr>、<td>等：用于定义表格结构。
- <form>、<input>、<textarea>等：用于定义表单结构。
- ：用于定义行内包含框。

在设计网页时，正确使用HTML5标签，可以避免代码冗余。

【示例代码】

【例1-6】编写如图1-7所示的HTML页面，使用HTML5标签演示网页文档应包含的内容，以及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>
```



图 1-7 网页效果

【知识点滴】

为了更好地使用标签，可参考W3School网站(<http://www.w3school.com.cn/tags/index.>) ASP页面中的信息。

1.4.2 使用div元素

在设计网页时经常需要在内容外围添加一个容器，从而可以为内容应用CSS样式或JavaScript效果，此时就需要使用div元素。

【示例代码】

【例1-7】编写一段网页代码，在代码中为页面中的文本内容加上div，添加更多样式的通用容器(扫码可查阅完整代码)，呈现的页面效果如图1-8所示。



```
<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>
```



图 1-8 使用 div 元素

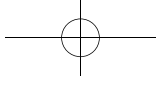
【示例解析】

以上代码在浏览器中的运行效果如图1-8所示。代码中有一个div元素包含所有内容，页面的语义并没有发生变化，但提供了一个可以用CSS添加样式的通用容器。与footer、nav、h1~h6、p、header、article、section、aside等元素一样，在默认情况下div元素自身没有任何样式，只是其包含的内容从新的一行开始。可以对div添加样式，以实现网页效果的设计。

div没有任何语义。在一般情况下，如果语义不合适，使用header、footer、main(仅使用一次)、article、section、aside或nav代替div会更合适。但若语义不合适，也不必为了避免使用div而使用上述元素。div适合所有页面容器，可以作为HTML5的备用容器使用。

【知识点滴】

div元素对使用JavaScript实现一些特定的交互行为或效果也有帮助。例如，在页面中展示一张图片或一个对话窗口，同时让背景页面覆盖一个半透明的层(这个层通常是一个div)。



1.4.3 使用id和class

在使用<div>标签定义页面文档中大部分结构时，为了能够识别不同的结构，一般通过定义id或class属性给它们赋予额外的语义，给CSS样式提供有效的“钩子”。

1. id属性

id属性用于规定元素的唯一id，常与CSS3样式相结合，实现对元素的控制。例如，以下代码是在页面中设计一个列表结构，并为其分配一个id，自定义页面导航模块。

```
<ul id="nav">
  <li><a href="#">首页</a></li>
  <li><a href="#">视频</a></li>
  <li><a href="#">发现</a></li>
  .....
</ul>
```

2. class属性

class属性定义了元素的类名，用于设置<div>标签中的元素的样式，其值为CSS3样式中的class选择符。与id不同，同一个class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，以下代码在<div>标签中使用class属性，使页面中的标题和段落居中显示，如图1-9所示。

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<style>
.center{
  text-align: center;
}
</style>
</head>
<body>
  <h1 class="center">标题居中</h1>
  <p class="center">段落居中。</p>
</body>
</html>
```



图 1-9 网页效果

【知识点滴】

在代码中使用id标识页面中的元素时，id名称必须是唯一的。id可以用于标识持久的结构性元素，如主导航或内容区域；id还可以用来标识一次性元素，如某个链接或表单元元素。

1.4.4 使用title

title 属性规定关于元素的额外信息。在页面中使用title属性可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>
```

当访问者将鼠标光标指向添加了title属性的标签时，就会显示title，如图1-10所示。



图 1-10 显示元素信息

1.4.5 使用role

role属性的作用是告诉Accessibility类应用(如平面阅读器等)当前元素所扮演的角色，主要供残疾人使用。使用role可以增强文本的可读性和语义化。例如，以下代码告诉屏幕阅读器，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常用的属性值如下。

(1) 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的宽度。

使用方法：将其添加到页面级的header元素，每个页面只用一次。

(2) 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。

使用方法：与nav元素是对应关系。应将其添加到每个nav元素，或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。

(3) role="main"(主体)：文档的主要内容。

使用方法：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中(可能是div)。在每个页面只用一次。

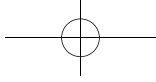
1.4.6 HTML5注释

HTML5注释的写法如下：

```
<!-- 注释内容 -->
```

在HTML5文档中添加注释，可以标明区块开始和结束的位置，提示代码开发者某段代码的意图，或者阻止内容显示等(这些注释只会在源代码中可见，在浏览器中则不可见)，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-->
<footer role="contentinfo"></footer>
</div>
<!--结束页面容器-->
```



1.5 主体结构

使用HTML5结构化元素，用户可以方便地创建网页的主体框架。

1.5.1 页眉

`<header>`标签是一个语义标签，用于定义文档或者文档的一部分区域的页眉。

【示例代码】

【例1-8】编写一段网页代码，在代码中使用`<header>`标签放置一个内容区块的标题(扫码可查阅完整代码)，如图1-11所示。



```
<body>
  <header>
    <h1>晨光洒落，秋日明朗</h1>
    <h3>藏在市井里的美好，朴素而平常</h3>
  </header>
</body>
```

【知识点滴】

以上代码在浏览器中的显示结果如图1-11所示。通常header用于定义整个页面的标题栏，或者一个内容块的标题区域。一个页面可以设计多个header结构，具体含义会根据其上下文而有所不同。例如，位于页面顶端的header代表整个页面的页头，位于栏目区域内的header代表栏目的标题。



图 1-11 网页效果

1.5.2 导航

`<nav>`标签用于表示HTML页面中的导航，可以是页与页之间的导航，也可以是页内的段与段之间的导航。

【示例代码】

【例1-9】编写一段网页代码，在代码中使用`<nav>`标签创建一个可拖动的导航区域(扫码可查阅完整代码)，如图1-12所示。

```
<nav>
  <a href="index.html">官网</a>
```



```
<a href="e-shop.html">商城</a>
<a href="register.html">注册</a>
</nav>
```

【知识点滴】

在以上代码中，nav元素包含3个用于导航的超链接，即“官网”“商城”“注册”。该导航链接可用于全局导航，也可以放置在页面中的某个板块，作为区域导航。

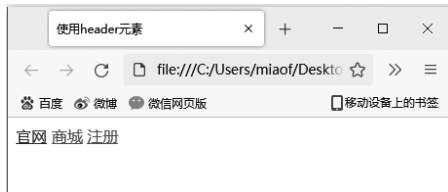


图 1-12 导航效果

1.5.3 主要区域

<main>标签用于指定文档的主体内容，并且标签中的内容在文档中是唯一的，不包含在文档中重复出现的内容，如侧边栏、导航栏、版权信息、站点标志或搜索表单等。

【示例代码】

【例1-10】编写一段网页代码，在代码中使用<main>标签包围表示页面主题的内容(扫码可查阅完整代码和示例效果)。



```
<main role="main">
  <article>
    <h4>01</h4>
    <p>岁月不晚不早，秋日恰似春朝。</p>
  </article>
  <article>
    <h4>02</h4>
    <p>夏天倾尽所有，酝酿了一个浪漫的秋。</p>
  </article>
  <article>
    <h4>03</h4>
    <p>秋天的声音很动听，日子也变得柔情。</p>
  </article>
</main>
```

【知识点滴】

main元素在一个页面中只能使用一次。在main开始标签中加上role="main"，可以帮助平面阅读器定位页面中的主要区域。

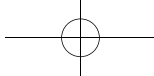
1.5.4 文章块

<article>标签定义外部的内容，用于标识页面中一块完整的、独立的、可被转发的内容，如论坛帖子、新闻文章、用户评论、微博信息条等。

【示例代码】

【例1-11】编写一段网页代码，在代码中演示<article>标签的使用方法(扫码可查阅完整代码和示例效果)。





```
<main role="main">
  <article>
    <h1>今日看点</h1>
    <p>我国自主研发大丝束碳纤维问世，被称为“新材料之王”。</p>
  </article>
</main>
```

【知识点滴】

可以将article嵌套在另一个article中，只要里面的article与外面的article是部分与整体的关系。一个页面可以有多个article元素。例如，新闻页面中一般包括多篇新闻文章，其中每一篇都是其自身的article。一个article可以包含一个或多个section元素。

1.5.5 区块

<section>标签用于定义文档中的节，多用于对内容进行区分(如章节、页眉、页脚或文档中的其他部分)。

【示例代码】

【例1-12】编写一段网页代码，在代码中使用<section>标签定义页面中特定的区块(扫码可查阅完整代码和示例效果)。



```
<section>
  <article>
    <header><h4>网友A的回复</h4></header>
    <p>评论内容……</p>
    <footer>发布时间：2023/12/3</footer>
  </article>
  <article>
    <header><h4>网友B的回复</h4></header>
    <p>评论内容……</p>
    <footer>发布时间：2023/12/2</footer>
  </article>
</section>
```

【知识点滴】

section用于定义通用区块，但不要将其与div元素混淆。从语义上讲，section标记的是页面中的特定区域，而div则不传达任何语义。div元素关注结构的独立性，而section元素包含的内容可以单独存储到数据库中，或输出到Word文档中。当一个容器需要被直接定义样式或通过脚本定义行为时，推荐使用div元素。

1.5.6 附栏

<aside>标签用于定义当前页面或者文章的附属信息部分，可以包含与当前页面或主要内容相关的引用、侧边栏、广告、导航条等其他类似的有别于主要内容的部分。

【示例代码】

【例1-13】编写一段网页代码，在代码中使用<aside>标签为页面添加一个友情链接导航(扫码可查阅完整代码和示例效果)。



```
<aside>
  <nav>
    <h2>友情链接导航</h2>
    <ul>
      <li><a href="https://www.zhihu.com/">知乎</a></li>
      <li><a href="https://www.ctrip.com/">携程</a></li>
      <li><a href="https://www.douyin.com/">抖音</a></li>
    </ul>
  </nav>
</aside>
```

【知识点滴】

aside元素主要有两种用法：一种是包含在article中，aside内容可以是与当前内容有关的参考资料或解释等；另一种是作为页面或站点辅助功能部分，在article之外使用，如设计侧边栏，其中的内容可以是文章列表、友情链接导航、评论列表等。

1.5.7 页脚

<footer>标签通常用于定义网页的底部布局，表示页脚，可以用作article、aside、blockquote、body、details、fieldset、figure、nav、section或td结构的页脚。

【示例代码】

【例1-14】编写一段网页代码，在代码中使用<footer>标签(扫码可查阅完整代码和示例效果)。



```
<footer>
  <p>© 2028 Gandalf</p>
</footer>
```

【知识点滴】

不能在footer内嵌套header或另一个footer。同时，也不能将footer嵌套在header或address元素里。