

“十三五”普通高等教育规划教材

Web 前端开发技术

卢 冶 白素琴 张其亮 张小立 刘永良 编著



机械工业出版社

本书紧贴互联网行业发展对 Web 前端开发工程师岗位的技术与能力的要求,详细地介绍了前端开发的三个基本要素:HTML、CSS 和 JavaScript,也涵盖了当前较为流行的 HTML5、CSS3、jQuery,以及 Ajax 和 JSON 等技术。教材包含大量的实例、思考题,并包含配套的实验,便于读者学习、自主练习与提高,以期达到熟练掌握各类前端开发技术的目的。

本书可以作为应用型本科院校计算机科学与技术、软件工程、信息管理与信息系统、通信工程相关专业“Web 前端开发技术”“Web 应用程序设计”或者计算机公共基础的“网页开发与设计”“网页制作”等课程的教材,也可以作为高职高专院校相关专业的教材,或者作为 Web 应用程序开发人员的参考用书。

图书在版编目(CIP)数据

Web 前端开发技术 / 卢冶等编著. —北京:机械工业出版社, 2019.1

“十三五”普通高等教育规划教材

ISBN 978-7-111-61820-1

I. ①W… II. ①卢… III. ①网页制作工具—高等学校—教材
IV. ①TP393.092.2

中国版本图书馆 CIP 数据核字(2019)第 036305 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:李馨馨 责任编辑:李馨馨

责任校对:张艳霞 责任印制:孙 炜

天津嘉恒印务有限公司印刷

2019 年 3 月第 1 版·第 1 次印刷

184mm×260mm·19 印张·465 千字

0001-3000 册

标准书号:ISBN 978-7-111-61820-1

定价:59.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

服务咨询热线:(010) 88379833

机工官网:www.cmpbook.com

读者购书热线:(010) 68326294

机工官博:weibo.com/cmp1952

教育服务网:www.cmpedu.com

封面无防伪标均为盗版

金书网:www.golden-book.com

前 言

Web 前端技术指 HTML、CSS、JavaScript 以及这些技术衍生的各种技术、框架、解决方案，主要用于实现互联网产品的用户界面交互。

前端技术的发展是互联网自身发展变化的一个缩影。在 Web 1.0 时代，由于网速和终端能力的限制，大部分网站只能呈现简单的图文信息，并不能满足用户对界面的需求。随着硬件的完善、高性能浏览器的出现和宽带的普及，前端技术领域迸发出非常旺盛的生命力。尤其是最近几年，移动互联网带来了大量高性能的移动终端设备以及快速的无线网络，HTML5、Node.js 得到广泛应用，各类框架类库层出不穷。前端开发技术的要素也演变成为现今的 HTML5、CSS3、jQuery。

Web 前端开发工程师是一个新职业，既要与上游的交互设计师、视觉设计师和产品经理沟通，又要与下游的服务器端工程师沟通，需要掌握的技能非常多。这就从知识的广度上对 Web 前端开发工程师提出了要求。本书正是为满足应用型软件人才培养过程中对前端开发工程师的知识和技术需求而编写的。

本书分为两大部分。第一部分（第 1~6 章）是 Web 基础：第 1、2 章介绍 HTML 基础和 HTML 高级应用；第 3 章介绍 CSS 基础语法，并结合实例讲解框模型与背景、文本格式化、表格、显示与定位等样式；第 4、5 章介绍 JavaScript 基础语法、DOM、常用内置对象、事件处理等知识，结合大量实例讲解运用 JavaScript 实现页面特效；还介绍了正则表达式；第 6 章讲解轻量级 JavaScript 库 jQuery，包括 jQuery 选择器、jQuery HTML 操作、jQuery 事件、jQuery 特效、jQuery 遍历。第二部分（第 7~10 章）是 Web 进阶：第 7 章围绕 HTML5 新技术，讲解 HTML5 新特性、File API、拖放、Canvas API、SVG、音频和视频、Geolocation API、Communication API、WebSockets API、Web Workers API、Web Storage API、离线 Web 应用；第 8 章讲解最新的层叠样式表 CSS3，结合实例讲解字体、动画、过渡、2D/3D 转换、多列布局等；第 9 章介绍 Ajax 原理、实现与 jQuery Ajax；第 10 章介绍轻量级的文本数据交换格式 JSON，以及从前台到后台的完整实例。

本书在内容选择、深度把握上充分考虑初学者的特点，内容安排上力求做到循序渐进。每章都配备了大量的实例，方便读者阅读、调试和运行，并辅助读者更好地理解所学内容。同时，每章都配有相应的习题，重点部分包含配套的实验，使读者加深印象、学以致用。本书不仅适合应用型本科院校相关专业 Web 应用开发的课程教学，也可以作为高职高专院校相关专业的教材，或者作为 Web 应用程序开发人员的参考用书。

由于作者水平有限，书中难免存在不足之处，敬请各位专家、老师和读者批评指正。

作 者

2018 年 9 月

目 录

前言

第一部分 Web 基础

第 1 章 HTML 基础	1	2.1.1 HTML 输入类型	12
1.1 HTML 简介	1	2.1.2 下拉列表框	14
1.1.1 什么是 HTML	1	2.1.3 文本域	15
1.1.2 HTML 标签	1	2.1.4 Button 元素	15
1.1.3 HTML 文档	1	2.1.5 用<fieldset>组合表单数据	15
1.1.4 HTML 元素	2	2.1.6 表单实例	15
1.1.5 HTML 属性	2	2.2 HTML 块	17
1.1.6 HTML 编辑	3	2.2.1 HTML 块级元素和内联元素	17
1.2 基本的 HTML 标签	3	2.2.2 HTML<div>元素	17
1.2.1 HTML 标题	3	2.2.3 HTML元素	18
1.2.2 HTML 段落	3	2.3 网站布局	18
1.2.3 HTML 折行	3	2.4 HTML 文档类型	19
1.2.4 HTML 水平线	4	2.4.1 HTML 版本	19
1.2.5 HTML 列表	4	2.4.2 常用的声明	20
1.2.6 HTML 特殊符号	5	2.5 HTML 头部元素	20
1.2.7 HTML 文本格式化标签	5	2.5.1 HTML<title>元素	20
1.2.8 HTML 注释	6	2.5.2 HTML<base>元素	21
1.3 HTML 链接	6	2.5.3 HTML<link>元素	21
1.3.1 HTML 链接语法	6	2.5.4 HTML<style>元素	21
1.3.2 HTML 链接——target 属性	6	2.5.5 HTML<meta>元素	21
1.3.3 HTML 链接——name 属性	7	2.5.6 HTML<script>元素	22
1.4 HTML 图像	7	2.6 HTML 统一资源定位器	22
1.4.1 源属性 (src)	7	2.7 HTML 多媒体	22
1.4.2 替换文本属性 (alt)	7	2.7.1 HTML object 元素	23
1.5 HTML 表格	8	2.7.2 HTML 音频	23
1.5.1 表格和边框属性	8	2.7.3 HTML 视频	24
1.5.2 表格的表头	8	思考题	24
1.5.3 跨行或跨列的表格单元格	9	第 3 章 CSS 基础	26
思考题	10	3.1 CSS 简介	26
第 2 章 HTML 高级应用	12	3.1.1 什么是 CSS	26
2.1 HTML 表单	12	3.1.2 CSS 的作用	26

3.2	CSS 基础语法	27	4.3.6	JavaScript 函数	56
3.3	如何创建 CSS	28	4.3.7	JavaScript 变量的生存期	58
3.3.1	外部样式表	28	4.3.8	JavaScript 运算符	58
3.3.2	内部样式表	28	4.3.9	JavaScript 语句	60
3.3.3	内联样式	29	4.3.10	JavaScript 错误	64
3.3.4	多重样式	29	4.4	HTML DOM	66
3.4	CSS 选择器	30	4.4.1	HTML DOM 树	66
3.4.1	元素选择器	30	4.4.2	查找 HTML 元素	67
3.4.2	id 选择器	30	4.4.3	改变 HTML	68
3.4.3	类选择器	30	4.4.4	改变 CSS	69
3.4.4	属性选择器	31	4.4.5	HTML DOM 事件	70
3.4.5	派生选择器	32	4.4.6	操作 HTML 元素	77
3.4.6	伪类	34	4.5	JavaScript 对象	78
3.4.7	伪元素	34	4.5.1	创建并访问对象	78
3.4.8	选择器组合	34	4.5.2	JavaScript Number 对象	80
3.5	CSS 定位与盒模型	35	4.5.3	JavaScript String 对象	82
3.5.1	元素可见性	35	4.5.4	JavaScript Date 对象	83
3.5.2	CSS 定位	36	4.5.5	JavaScript Array 对象	85
3.5.3	CSS 盒模型	38	4.5.6	JavaScript Boolean 对象	88
3.6	CSS 实例	41	4.5.7	JavaScript Math 对象	89
3.6.1	设置背景颜色和图片	41	4.6	Window 对象	91
3.6.2	修饰文本	42	4.6.1	Window 尺寸	91
3.6.3	修饰列表	44	4.6.2	其他 window 方法	92
3.6.4	定位	45	4.6.3	Window Screen	93
3.6.5	超链接	45	4.6.4	Window Location	93
3.6.6	修饰表格	46	4.6.5	Window History	94
思考题		47	4.6.6	Window Navigator	95
第 4 章	JavaScript	49	4.6.7	JavaScript 消息框	95
4.1	JavaScript 简介	49	4.6.8	JavaScript 计时	97
4.2	JavaScript 使用	49	4.6.9	JavaScript Cookies	98
4.2.1	<script>标签	49	4.7	JavaScript 应用实例	99
4.2.2	JavaScript 函数和事件	50	4.7.1	制作浮动的带关闭按钮的广告	99
4.2.3	外部的 JavaScript	51	4.7.2	制作输入提示的特效	100
4.3	JavaScript 基本语法	51	4.7.3	级联功能	102
4.3.1	JavaScript 输出	51	4.7.4	树形菜单	105
4.3.2	JavaScript 语句	52	4.7.5	带按钮的广告图片轮播	106
4.3.3	JavaScript 注释	53	思考题		108
4.3.4	JavaScript 变量	53			
4.3.5	JavaScript 数据类型	54			

第 5 章 正则表达式	110	6.3.6 jQuery CSS 选择器	126
5.1 正则表达式简介	110	6.3.7 更多选择器示例	126
5.1.1 RegExp 对象	110	6.4 jQuery 事件	127
5.1.2 RegExp 对象属性	111	6.4.1 jQuery 事件函数	127
5.1.3 RegExp 对象方法	113	6.4.2 单独文件中的函数	128
5.1.4 支持正则表达式的 String 对象的方法	114	6.4.3 jQuery 名称冲突	128
5.2 正则表达式语法	115	6.4.4 jQuery 编程原则	129
5.2.1 限定符	115	6.5 jQuery 中的 DOM 操作	129
5.2.2 选择匹配符	115	6.5.1 获取与设置内容	129
5.2.3 分组组合与反向引用符	116	6.5.2 获取与设置属性	131
5.2.4 特殊字符	116	6.5.3 jQuery 添加元素	132
5.2.5 字符匹配符	116	6.5.4 jQuery 删除元素	134
5.2.6 定位符	117	6.5.5 jQuery 获取并设置 CSS 类	136
5.2.7 原义字符	117	6.6 jQuery 遍历节点	140
5.3 正则表达式实例	118	6.6.1 jQuery 遍历祖先	140
5.3.1 模式范例	118	6.6.2 jQuery 遍历后代	143
5.3.2 常用表单验证	118	6.6.3 jQuery 遍历同胞	145
思考题	119	6.6.4 jQuery 过滤	146
第 6 章 jQuery	121	6.7 jQuery 效果	151
6.1 jQuery 简介	121	6.7.1 hide()、show()和 toggle()	151
6.1.1 jQuery 库	121	6.7.2 jQuery 淡入淡出	152
6.1.2 jQuery 安装	121	6.7.3 jQuery 滑动	156
6.1.3 jQuery 语法	123	6.7.4 jQuery 动画	157
6.1.4 文档就绪函数	123	6.7.5 jQuery 停止动画	158
6.2 jQuery 对象和 DOM 对象	124	6.7.6 jQuery Callback 方法	159
6.2.1 DOM 对象	124	6.7.7 jQuery 链式编程	160
6.2.2 jQuery 对象	124	6.8 jQuery 应用实例	160
6.2.3 jQuery 对象和 DOM 对象的相互转换	124	6.8.1 jQuery 遍历函数	160
6.3 jQuery 选择器	125	6.8.2 评分控件	162
6.3.1 jQuery 元素选择器	125	6.8.3 表格选取	163
6.3.2 jQuery #id 选择器	125	6.8.4 倒计时读秒阅读协议	164
6.3.3 jQuery .class 选择器	125	6.8.5 搜索框效果	164
6.3.4 jQuery 属性选择器	125	6.8.6 全选/全不选/反选	165
6.3.5 jQuery 层次选择器	126	6.8.7 提示文字	166
		思考题	167

第二部分 Web 进阶

第 7 章 HTML5	169	7.1 HTML5 简介	169
-------------	-----	--------------	-----

7.2 HTML5 新特性	170	7.11.1 localStorage 方法	205
7.2.1 简化的文档类型和字符集	170	7.11.2 sessionStorage 方法	206
7.2.2 使用新的 HTML5 解析器	170	7.11.3 IndexedDB	207
7.2.3 HTML5 文档结构	171	7.12 HTML5 应用程序缓存	210
7.2.4 HTML5 增强的 iframe 元素	172	7.12.1 Cache Manifest 基础	210
7.2.5 HTML5 新增的内联元素	174	7.12.2 Manifest 文件	210
7.2.6 HTML5 表单的新特性	174	7.12.3 更新缓存	211
7.3 HTML5 文件处理	180	7.13 HTML5 Web Worker	213
7.3.1 选择文件的表单控件	180	7.13.1 Web Worker 工作过程	213
7.3.2 HTML5 File API	180	7.13.2 Web Worker 工作实例	214
7.3.3 FileList 接口	181	7.13.3 其他类型的 Worker	215
7.3.4 FileReader 接口	182	7.14 HTML5 服务器发送事件	215
7.4 HTML5 视频	184	7.14.1 接收 Server-Sent 事件通知	215
7.4.1 视频格式	184	7.14.2 检测 Server-Sent 事件支持	216
7.4.2 在 HTML5 中显示视频	185	7.14.3 服务器端代码示例	216
7.4.3 <video>标签的属性	185	7.14.4 EventSource 对象	216
7.5 HTML5 音频	186	7.15 Web 通信	217
7.5.1 音频格式	186	7.15.1 跨文档消息机制	217
7.5.2 在 HTML5 中播放音频	186	7.15.2 XMLHttpRequest Level 2	219
7.5.3 <audio>标签的属性	187	7.15.3 Web Socket	222
7.6 HTML5 拖放	187	思考题	226
7.6.1 HTML5 拖放事件	187	第 8 章 最新的层叠样式表 CSS3	228
7.6.2 HTML5 拖放实例	188	8.1 CSS3 简介	228
7.6.3 实例分析	188	8.2 CSS3 新技术	228
7.7 HTML5 Canvas	189	8.2.1 CSS3 边框	228
7.7.1 使用 Canvas 元素	190	8.2.2 CSS3 背景	231
7.7.2 绘制图形实例	190	8.2.3 CSS3 文本效果	232
7.7.3 图形的操作	194	8.2.4 CSS3 字体	234
7.8 HTML5 内联 SVG	196	8.2.5 CSS3 2D 转换	236
7.8.1 SVG 介绍	196	8.2.6 CSS3 3D 转换	238
7.8.2 嵌入.svg 文件	197	8.2.7 CSS3 过渡	240
7.8.3 HTML 页面直接定义 SVG 代码	197	8.2.8 CSS3 动画	242
7.8.4 HTML 5 Canvas 与 SVG	200	8.2.9 CSS3 多列	244
7.9 HTML5 MathML	201	8.3 CSS3 应用实例	245
7.10 HTML5 地理定位	202	8.3.1 设计页面布局	245
7.10.1 地理位置	202	8.3.2 设计登录页面	247
7.10.2 使用地理位置实例	202	8.3.3 设计 3D 导航菜单	250
7.11 HTML5 Web 存储	205	8.3.4 设计自动轮播效果	252

思考题	255	10.3.1 将 JSON 文本转换为 JavaScript 对象	274
第 9 章 Ajax 技术	256	10.3.2 将 JSON 对象转换为 JSON 字符串	275
9.1 Ajax 基础	256	10.4 JSON 特点及后台使用	275
9.1.1 XMLHttpRequest 对象	256	10.5 综合应用	276
9.1.2 XHR 请求	257	10.5.1 JSP 页面	276
9.1.3 XHR 响应	260	10.5.2 Servlet 编写	277
9.1.4 XHR readyState	263	思考题	278
9.1.5 Ajax 应用的 5 个步骤	264	附录 实验	279
9.2 jQuery Ajax	264	实验一 使用 JavaScript 实现网页 特效	279
9.2.1 jQuery 加载	265	实验二 使用 jQuery 实现网页特效	280
9.2.2 jQuery get()和 post()	266	实验三 HTML5 表单及文件处理	281
9.2.3 jQuery \$.ajax()	268	实验四 使用 Canvas API 画图	283
思考题	269	实验五 获取浏览器的地理位置 信息	286
第 10 章 JSON 简介	270	实验六 Web 通信	288
10.1 JSON 与 XML	270	实验七 使用 CSS3 表现页面	289
10.2 JSON 语法	271	实验八 Ajax 技术应用	293
10.2.1 JSON 语法规则	271	参考文献	294
10.2.2 JSON 名称/值对	272		
10.2.3 JSON 使用 JavaScript 语法	273		
10.2.4 JSON 文件	274		
10.3 JSON 使用	274		

第一部分 Web 基础

第 1 章 HTML 基础

1.1 HTML 简介

HTML 指超文本标记语言，是通向 Web 技术世界的钥匙。

1.1.1 什么是 HTML

HTML 是用来描述网页的一种语言。

- HTML 指的是超文本标记语言（Hyper Text Markup Language）。
- HTML 不是一种编程语言，而是一种标记语言（markup language）。
- 标记语言是一套标记标签（markup tag）。
- HTML 使用标记标签来描述网页。

所谓超文本，是由信息结点和表示信息结点间相关性的链接构成的一个具有一定逻辑结构和语义的网络。

传统的文本是顺序的，线性表示的，而超文本不是顺序的，它是一个非线性的网状结构，把文本按其内部固有的独立性和相关性划分成不同的基本信息块。超文本是一种用于文本、图形或者计算机的信息组织形式，它使得单一的信息块之间相互交叉“引用”。这种“引用”并不是通过复制来实现的，而是通过指向对方的地址字符串来指引用户获取相应的信息。这种信息组织形式是非线性的，它使得 Internet 成为真正为大多数人所接受的交互式的网络。

1.1.2 HTML 标签

HTML 标记标签通常被称为 HTML 标签（HTML tag）。

- HTML 标签是由尖括号包围的关键词，例如<html>。
- HTML 标签通常是成对出现的，例如<body>和</body>。
- 标签对中的第一个标签是开始标签，第二个标签是结束标签。
- 开始和结束标签也被称为开放标签和闭合标签。

1.1.3 HTML 文档

HTML 文档也被称为网页。HTML 文档包含 HTML 标签和纯文本。Web 浏览器的作用

就是读取 HTML 文档，并以网页的形式呈现它们。浏览器不会显示 HTML 标签，而是使用标签来解释页面的内容。

例 1-1:

```
<html>
  <body>
    <h1>My First Heading</h1>
    <p>My first paragraph.</p>
  </body>
</html>
```

其中:

- <html>与</html>之间的文本描述网页。
- <body>与</body>之间的文本是可见的页面内容。
- <h1>与</h1>之间的文本被显示为标题。
- <p>与</p>之间的文本被显示为段落。

1.1.4 HTML 元素

HTML 元素指的是从开始标签 (start tag) 到结束标签 (end tag) 的所有代码。

1. HTML 元素语法

- HTML 元素以开始标签起始。
- HTML 元素以结束标签终止。
- 元素的内容是开始标签与结束标签之间的内容。
- 某些 HTML 元素具有空内容 (empty content)。
- 空元素在开始标签中关闭。
- 大多数 HTML 元素可以拥有属性。
- 大多数 HTML 元素可以嵌套，即可以包含其他 HTML 元素。
- HTML 文档由嵌套的 HTML 元素构成。

2. 空的 HTML 元素

没有内容的 HTML 元素称为空元素。空元素是在开始标签中关闭的。例如:
就是没有关闭标签的空元素,
标签定义换行。在开始标签中添加斜杠——
, 是关闭空元素的正确方法。

3. HTML 使用小写标签

HTML 标签对大小写不敏感: <P>等同于<p>。许多网站都使用大写的 HTML 标签。万维网联盟 (W3C) 在 HTML 4 中推荐使用小写, 在未来的(X)HTML 版本中强制使用小写。

1.1.5 HTML 属性

HTML 属性为 HTML 元素提供附加信息。属性总是以名称/值对的形式出现, 例如: name="value"。属性总是在 HTML 元素的开始标签中定义。

例 1-2:

HTML 链接由<a>标签定义。链接的地址在 href 属性中指定:

`江科大张家港校区`

1. 使用小写属性

属性和属性值对大小写不敏感。万维网联盟（W3C）在 HTML 4 中推荐使用小写，在未来的(X)HTML 版本中强制使用小写。

2. 始终为属性值加引号

属性值应该始终被包括在引号内。双引号是最常用的，不过使用单引号也没有问题。在某些个别的情况下，例如属性值本身就含有双引号，那么就必须使用单引号，例如：`name="Tom "HelloWorld" Smith"`。

1.1.6 HTML 编辑

可以使用 Notepad 或者 TextEdit 来编写 HTML 文档，也可以使用专业的 HTML 编辑器，例如 Adobe Dreamweaver。当保存 HTML 文件时，既可以使用 .htm 也可以使用 .html 扩展名。两者没有区别，完全根据个人喜好。HTML 文件可以在浏览器中运行。

1.2 基本的 HTML 标签

1.2.1 HTML 标题

HTML 标题（Heading）通过 `<h1>` - `<h6>` 标签进行定义。应该将 `h1` 用做主标题（最重要的），其次是 `h2`（次重要的），再次是 `h3`，依此类推。

例 1-3:

```
<h1>This is a heading</h1>
<h2>This is a heading</h2>
<h3>This is a heading</h3>
```

1.2.2 HTML 段落

HTML 段落是通过 `<p>` 标签进行定义的，`<p>` 是块级元素。块级元素（block element）在浏览器中显示时，通常会以新行来开始和结束，和其对应的是内联元素（inline element）。内联元素的显示可以形象地称为“文本模式”，即一个挨着一个，都在同一行按从左至右的顺序显示，不单独占一行。

例 1-4:

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

1.2.3 HTML 折行

如果希望在不产生一个新段落的情况下换行，应使用 `
` 标签。`
` 元素是一个空的 HTML 元素。由于关闭标签没有任何意义，因此它没有结束标签。

例 1-5:

```
<p>This is<br/>a para<br/>graph with line breaks</p>
```

1.2.4 HTML 水平线

`<hr/>` 标签在 HTML 页面中创建水平线，可用于分隔内容。`<hr>` 标签同样也没有结束标签。使用水平线（`<hr/>` 标签）来分隔文章中的小节是一个办法，但并不是唯一的办法。

例 1-6:

```
<p>This is a paragraph.</p>
<hr/>
<p>This is a paragraph.</p>
<hr/>
<p>This is a paragraph.</p>
```

1.2.5 HTML 列表

1. 无序列表

无序列表是一个项目的列表，此列项目使用粗体圆点进行标记。无序列表始于 `` 标签，每个列表项始于 ``。

例 1-7:

```
<ul>
  <li>Apple</li>
  <li>Banana</li>
  <li>Orange</li>
</ul>
```

2. 有序列表

有序列表也是一列项目，列表项目使用数字进行标记。有序列表始于 `` 标签，每个列表项始于 `` 标签。`` 标签的 `type` 属性表示列表前缀的格式，通常有 1、a、A、i、I 五个值，而 `start` 属性表示从 `type` 类型的第几个数字开始。

例 1-8:

```
<ol type="i" start="20">
  <li>Apple</li>
  <li>Banana</li>
  <li>Orange</li>
</ol>
```

3. 自定义列表（dl-dt-dd）

自定义列表不只是一列项目，而是项目及其注释的组合。自定义列表以 `<dl>` 标签开始，每个自定义列表项以 `<dt>` 开始，每个自定义列表项的定义以 `<dd>` 开始。在自定义列表中，`dt` 和 `dd` 中有了缩进。使用有序和无序列表实现此结构就要用到列表的嵌套。

例 1-9:

```
<dl>
  <dt>一级</dt>
  <dd>二级</dd>
  <dd>二级</dd>
  <dt>一级</dt>
  <dd>二级</dd>
  <dd>二级</dd>
</dl>
```

1.2.6 HTML 特殊符号

有时候需要向网页中加入特殊符号使网页更好看，更确切地表达意思。向 HTML 页面中输入特殊字符，需要在 HTML 代码中加入以&开头的字母组合或者以&#x开头的数字。常用的 HTML 特殊符号见表 1-1。

表 1-1 HTML 特殊符号

特殊符号	命名实体	十进制编码
空格	 	
©	©	©
®	®	®
<	<	<
>	>	>
&	&	&
"	"	"

1.2.7 HTML 文本格式化标签

常用的 HTML 文本格式化标签见表 1-2。

表 1-2 HTML 文本格式化标签

标 签	描 述
	定义粗体文本
	标记重点强调的文本，以斜体形式呈现
	文本以加粗形式呈现
<pre>	定义预格式文本

标签和标签虽然在网页中显示效果一样，但实际目的却不同。标签对应 bold，即文本加粗，其目的仅仅是为了加粗显示文本，是一种样式风格需求。标签的意思是加强，表示该文本比较重要，提醒读者或者终端注意。为了达到这个目的，浏览器等终端将其加粗显示。

1.2.8 HTML 注释

可以将注释插入 HTML 代码中，这样能够提高程序的可读性，使代码更容易被人理解。浏览器会忽略注释，也不会显示它们。合理地使用注释可以对未来的代码编辑、维护工作产生帮助。注释示例如下：

例 1-10：

```
<!-- This is a comment -->
```

1.3 HTML 链接

HTML 使用超链接与网络上的另一个文档相连。几乎在所有的网页中都可以找到链接。超链接可以是一个字、一个词或者一组词，也可以是一幅图像，点击这些内容可以跳转到新的文档或者当前文档中的某个部分。当用户把鼠标指针移动到网页中的某个链接上时，箭头会变为一只小手。

通过使用标签在 HTML 中创建链接。有两种使用标签的方式：

- 通过使用 href 属性创建指向另一个文档的链接。
- 通过使用 name 属性创建文档内的书签。

1.3.1 HTML 链接语法

链接的 HTML 代码通过标签进行定义，href 属性规定链接的目标。开始标签和结束标签之间的文字被作为超级链接来显示。“链接文本”未必一定是文本，图片或者其他 HTML 元素都可以成为链接。

例 1-11：

```
<a href="http://zjg.just.edu.cn/">访问江苏科技大学（张家港）</a>
```

上面这行代码显示为：访问江苏科技大学（张家港）。点击这个文本超链接会把用户带到江苏科技大学（张家港）的首页。

1.3.2 HTML 链接——target 属性

使用 target 属性，可以定义被链接的文档在何处显示。target 属性的取值见表 1-3。

表 1-3 target 属性取值

属性值	描述
_blank	在新窗口中打开被链接文档
_self	默认，在相同的框架中打开被链接文档
_parent	在父框架集中打开被链接文档
_top	在整个窗口中打开被链接文档

下面这行代码会在新窗口中打开文档：

```
<a href="http://zjg.just.edu.cn/" target="_blank">访问江苏科技大学（张家港）</a>
```

1.3.3 HTML 链接——name 属性

name 属性规定锚点 (anchor) 的名称, 使用它可以创建 HTML 页面中的书签。命名锚点链接 (也叫书签链接) 常常用于那些内容庞杂、烦琐的网页。通过点击命名锚点, 不仅能够指向文档, 还能够指向页面里的特定段落, 更能当作“精准链接”的便利工具, 让链接对象接近焦点。这样浏览者就无需不停地滚动页面来寻找他们需要的信息了, 便于查看网页内容, 类似于书籍中的目录页码或者章回提示。在需要指定到页面的特定部分时, 标记锚点是最佳的方法。锚点的名称可以是用户喜欢的任何名字, 也可以使用 id 属性来替代 name 属性, 命名锚点同样有效。命名锚点的语法:

```
<a name="label">锚点 (显示在页面上的文本) </a>
```

例 1-12: 锚点链接

首先, 创建一个书签, 即在 HTML 文档中对锚点进行命名:

```
<a name="tips">基本的注意事项 - 有用的提示</a>
```

然后, 在同一个文档中创建指向该锚点的链接:

```
<a href="#tips">有用的提示</a>
```

也可以在其他页面中创建指向该锚点的链接:

```
<a href="http://zjg.just.edu.cn/html/guide.htm#tips">有用的提示</a>
```

在上面的代码中, 将#符号和锚点名称添加到 URL 的末端, 就可以直接链接到 tips 这个命名锚点了。

1.4 HTML 图像

在 HTML 中, 图像由标签定义。是空标签, 即只包含属性, 并且没有闭合标签。

1.4.1 源属性 (src)

要在页面上显示图像, 需要使用源属性 (src), src 指 source。源属性的值是图像的 URL 地址。定义图像的语法是:

```

```

URL 指存储图像的位置。如果名为 boat.gif 的图像位于 zjg.just.edu.cn 的 images 目录中, 那么其 URL 为 http://zjg.just.edu.cn/images/boat.gif。

浏览器将图像显示在文档中图像标签出现的地方。例如, 将图像标签置于两个段落之间, 那么浏览器会首先显示第一个段落, 然后显示图片, 最后显示第二段。

1.4.2 替换文本属性 (alt)

alt 属性用来为图像定义一串预备的可替换的文本。替换文本属性的值是用户自定义

的。在浏览器无法载入图像时，替换文本属性将告诉读者他们看不到的信息。此时，浏览器会显示这个替代性的文本而不是图像。为页面上的图像都加上替换文本属性是个好习惯，这样有助于更好地显示信息，并且对于那些使用纯文本浏览器的用户是非常有用的。

```

```

1.5 HTML 表格

表格由<table>标签来定义。每个表格均有若干行，由<tr>标签定义，每行被分割为若干单元格。字母 td 指表格数据 (table data)，即数据单元格的内容。数据单元格可以包含文本、图片、列表、段落、表单、水平线、表格等。

例 1-13:

```
<table border="1">
  <tr>
    <td>row 1, cell 1</td>
    <td>row 1, cell 2</td>
  </tr>
  <tr>
    <td>row 2, cell 1</td>
    <td>row 2, cell 2</td>
  </tr>
</table>
```

在浏览器中显示如下:

row 1, cell 1	row 1, cell 2
row 2, cell 1	row 2, cell 2

1.5.1 表格和边框属性

如果不定义边框属性，表格将不显示边框，但是大多数时候还是希望显示边框的。以下代码使用边框属性来显示一个带有边框的表格:

```
<table border="1"> ...</table>
```

1.5.2 表格的表头

使用<th>标签定义表头。大多数浏览器会把表头显示为粗体居中的文本。

例 1-14:

```
<table border="1">
  <tr>
    <th>Heading</th>
    <th>Another Heading</th>
```