

网页设计与制作教程

主 编 欧静敏 阚钿玉

副主编 梁英坚 付 琳 黄 翹

西北工业大学出版社

西 安

目 录

| | |
|---------------------|-----|
| 第 1 章 网页制作的基础知识 | 001 |
| 1.1 认识网页和网站 | 001 |
| 1.2 网页的基本构成元素 | 003 |
| 1.3 页面布局结构 | 004 |
| 1.4 Web 前端技术简介 | 006 |
| 1.5 HTML 语法基础 | 010 |
| 1.6 创建 HTML 文档 | 013 |
| 1.7 网页头部标签 | 017 |
| 1.8 HTML5 文档注释和特殊符号 | 020 |
| 1.9 综合案例——临江仙·送钱穆父 | 022 |
| 本章习题 | 023 |
| 第 2 章 网页制作的排版方法 | 025 |
| 2.1 文字与段落排版 | 025 |
| 2.2 超链接 | 034 |
| 2.3 图像 | 040 |
| 2.4 列表 | 047 |
| 2.5 综合案例——无线吸尘器说明书 | 053 |
| 本章习题 | 057 |
| 第 3 章 表格布局与表单交互 | 058 |
| 3.1 表格概述 | 058 |
| 3.2 表格属性设置 | 061 |
| 3.3 表格嵌套 | 073 |
| 3.4 表单 | 075 |
| 3.5 综合案例——表格与表单 | 090 |
| 本章习题 | 096 |
| 第 4 章 CSS 样式基础 | 098 |
| 4.1 CSS 概述 | 098 |
| 4.2 CSS 语法基础 | 101 |
| 4.3 CSS 引入方式 | 107 |
| 4.4 CSS 的属性单位 | 113 |
| 4.5 CSS 继承与层叠 | 115 |



| | | |
|-------------|------------------------------|------------|
| 4.6 | 元素类型 | 117 |
| | 本章习题 | 123 |
| 第5章 | CSS 盒模型 | 125 |
| 5.1 | 盒模型的定义 | 125 |
| 5.2 | CSS 元素的高度和宽度 | 126 |
| 5.3 | 边距设置和边框设置 | 129 |
| 5.4 | CSS 元素的定位 | 144 |
| 5.5 | CSS 元素的浮动 | 151 |
| 5.6 | 综合案例——昵心美食空间 | 156 |
| | 本章习题 | 166 |
| 第6章 | 元素应用 CSS | 168 |
| 6.1 | 使用 CSS 设置字体样式 | 168 |
| 6.2 | 使用 CSS 设置文本样式 | 174 |
| 6.3 | 使用 CSS 设置图像样式 | 185 |
| 6.4 | 使用 CSS 设置表单样式 | 195 |
| 6.5 | 综合案例——商城的注册页面 | 200 |
| | 本章习题 | 204 |
| 第7章 | 利用 CSS 和多媒体美化页面 | 206 |
| 7.1 | CSS 链接的美化 | 206 |
| 7.2 | CSS 列表的美化 | 213 |
| 7.3 | CSS 表格的美化 | 222 |
| 7.4 | 多媒体的添加与美化 | 228 |
| 7.5 | 综合案例——海洋旅游胜地 | 235 |
| | 本章习题 | 242 |
| 第8章 | 利用 CSS 制作导航菜单 | 244 |
| 8.1 | 水平顶部导航栏 | 244 |
| 8.2 | 纵向侧边导航栏 | 257 |
| 8.3 | 底部固定导航栏 | 271 |
| 8.4 | 综合案例——优名养生馆 | 276 |
| | 本章习题 | 292 |
| 第9章 | DIV+CSS 布局 | 294 |
| 9.1 | DIV+CSS 概述 | 294 |
| 9.2 | DIV+CSS 的应用 | 299 |
| 9.3 | DIV+CSS 的典型布局 | 306 |
| 9.4 | 综合案例——众成远程教育 | 311 |
| | 本章习题 | 322 |
| 第10章 | JavaScript 的应用 | 325 |
| 10.1 | JavaScript 概述 | 325 |

| | | |
|---------------|-------------------------------|------------|
| 10.2 | JavaScript 语法 | 331 |
| 10.3 | JavaScript 对象 | 339 |
| 10.4 | JavaScript 事件 | 351 |
| 10.5 | 综合案例——轮播广告 | 359 |
| | 本章习题 | 364 |
| 第 11 章 | 综合案例——“精品课程网站”开发 | 366 |
| 11.1 | 网站的开发流程 | 366 |
| 11.2 | 设计首页布局 | 368 |
| 11.3 | 首页的制作 | 370 |
| 11.4 | 制作“教学课件”子页 | 383 |
| 11.5 | 制作“作品展示”子页 | 392 |
| | 参考文献 | 396 |

第 1 章 网页制作的基础知识



学习目标

1. 理解网页、网站、网址等基本概念；
2. 了解常用的网页制作工具和网页素材制作工具；
3. 对网页的基本结构做初步了解。



知识要点

1. 网站、网页的概念；
2. 网页的基本构成元素和布局结构；
3. Web (World Wide Web, WWW, 万维网, 简称 Web) 前端技术；
4. HTML (Hyper Text Markup Language, 超文本标识语言) 语法基础和文档创建；
5. 网页头部标签；
6. HTML5 文档注释和特殊符号。

1.1 认识网页和网站

1.1.1 网页、网站

1. 网页

网页是由 HTML 编写, 通过 WWW 网传输, 且被浏览器编译后供用户获取信息的页面文件, 又称为 Web 页, 集合了文本、图片、声音、动画、数字电影和超级链接等各种网页元素。如果说 WWW 是 Internet (因特网) 上的一个大型图书馆, 那么图书馆的每一本书就是一个 Web 站点, 而网页就是书中的某一页, 页码就是网址, 网址与页面一一对应, 多个网页关联组合在一起就成为一个 Web 站点。Web 站点中有一个特殊的网页叫作主页 (Homepage, 也叫首页), 相当于书的封面。

HTML 不是一种程序语言, 而是一种描述文档结构的标记语言, 是用户与电子计算机 (俗称“电脑”) 之间进行交流的一种文本技术。各种网页均是用 HTML 来描述的, 用 HTML 编写的网页文件的扩展名一般为“*.htm”或“*.html”。

由于网页中包含超级链接, 网页也被称为超文本 (Hypertext), 传输超文本的协议被称



为超文本传输协议 (Hyper Text Transfer Protocol, HTTP)。

2. 网站

Web 站点也称为网站, WWW 信息是由无数的 Web 站点组成的, 网页则是 Web 站点的基本信息单位, 超链接将站点中的一个或多个网页连接成为一个便于浏览的有机整体。

Web 站点其实是 Internet 上能够提供 Internet 服务的一个位置, 这个位置由独一无二的 IP 地址或者域名来描述, 一个网站需要有一台或者是多台服务器来实现其 WWW 服务。不同网站的规模与大小各不相同: 大的网站如新浪、搜狐等, 需要多台服务器; 小的网站如个人主页等, 仅占据某台服务器上一个很小的空间。

网站是多个网页的集合, 按网站内容可将网站分为 5 种类型: 门户网站、企业网站、个人网站、专业网站和职能网站。按网站的设计技术可将网站分为静态网站和动态网站 (ASP, JSP, PHP 和 Web 后台服务器软件)。

网站是有独立域名、独立存放空间的内容的集合, 这些内容可能是网页, 也可能是程序或者其他文件。不一定需要很多的网页, 只要有独立域名和空间, 哪怕只有一个页面也可以称为网站。

3. 常用术语

网页设计有其专业的常用术语, 如 Internet、WWW、浏览器、URL、IP、域名、HTTP、FTP、站点、发布、超链接、导航条、客户机和服务器等, 作为一名网页设计师, 必须熟练掌握这些常用术语。

Internet: 由各种不同类型的计算机网络连接起来的全球性网络。

WWW: 其功能是让 Web 客户端 (常用浏览器) 访问 Web 服务器中的网页。

浏览器: 将 Internet 中的文本文档和其他文件翻译成网页的软件, 通过浏览器可以快捷地获取 Internet 中的内容。常用的浏览器有 Internet Explorer (IE) 浏览器, Firefox 浏览器和 Chrome 浏览器等。

URL: 统一资源定位符, 指定通信协议和地址, 如 “http://www.baidu.com” 是一个 URL, “http://” 表示通信协议为超文本传输协议, “www.baidu.com” 表示网站名称。

IP: 网际协议。Internet 中的每台计算机都有唯一的 IP 地址, 表示该计算机在 Internet 中的位置。IP 通常分为 A、B 和 C 三类。

域名: 指网站名称, 在全世界是唯一的。通常把域名看成网站的网址。域名由固定的网络域名管理组织进行全球统一管理。域名需向各地的网络管理机构进行申请才能获取。域名的书写格式: 机构名. 主机名. 类别名. 地区名。

HTTP: 超文本传输协议, 是互联网上应用最为广泛的一种网络协议。所有的 WWW 文件都必须遵守这个标准。

FTP: 文件传输协议, 通过该协议, 可把文件从一个地方传到另外一个地方, 从而真正地实现资源共享。

发布: 指将制作好的网页传到网络上的过程, 也称为上传网站。

站点: 一个站点就是一个网站所有内容所存放的文件夹。Dreamweaver 的使用是以站点为基础的, 必须为每一个要处理的网站建立一个本地站点。站点可分父子站点。站点管理是对一个 Internet 的站点进行组织、维护和管理的功能集合。

超链接：从一个网页指向一个目标的连接关系，这个目标可以是另一个网页，可以是相同网页的不同位置，也可以是一个图片、一个电子邮件地址、一个文件，甚至是一个程序。在浏览网页时单击超链接就能跳转到与之相应的页面。

客户机和服务器：浏览网页是由个人计算机向 Internet 中的计算机发出请求，Internet 中的计算机在接受到请求后响应请求，将需要的内容发到个人计算机上，这种发送请求的个人计算机称为客户机或客户端，而 Internet 中的计算机称为服务器或服务端。

1.1.2 静态网页和动态网页

1. 静态网页

静态网页是指客户端的浏览器发送 URL 请求给 WWW 服务器，服务器查找需要的超文本文件并不加处理地直接下载到客户端，运行在客户端的页面是已经事先做好并存放在服务器中的网页。静态网页通常由纯粹的 HTML/CSS 语言编写。

2. 动态网页

动态网页能够根据不同浏览者的请求来显示不同的内容，无论网页本身是否具有视觉意义上的动态效果，只要采用动态网站技术生成的网页都称为动态网页，其本质主要体现在交互性方面。根据程序运行区域的不同，动态网页可分为客户端动态网页与服务端动态网页。

客户端动态网页不需要与服务器进行交互，实现动态功能的代码往往采用脚本语言形式直接嵌入网页中，常见的客户端动态网页技术包括 JavaScript, ActiveX 和 Flash 等。

服务端动态网页则需要与客户端共同参与，客户端通过浏览器发出页面请求后，服务端根据 URL 携带的参数运行服务端程序，产生的结果页面再返回客户端。动态网页比较注重交互性，即网页会根据客户端的要求和选择而动态改变和响应。一般涉及数据库操作的网页（如注册、登录和查询等）都需要服务端动态网页程序。

1.2 网页的基本构成元素

在初次设计网页之前，首先应该认识一下构成网页的基本元素，这样才能在设计中得心应手，根据需要合理地组织和安排网页内容。

一般网站的首页包含常见的网页元素，如导航栏、广告动画、图片、交互表单、文本和超链接等。

1. 文本

文本具有体积小、网络传输速度快等特点，可使用户更方便地浏览和下载文本信息，是网页最主要的基本元素，也是页面中最主要的信息载体。与图片相比，文本虽然不如图片那样能够很快引起浏览者的注意，但却能准确地表达信息的内容和含义。为了克服文本固有的缺点，人们赋予了网页中文本更多的属性，如字体、字号、颜色、底纹和边框等，通过不同格式的区别，突出显示重要的内容。此外，用户还可以在网页中设计各种各样的文字列表，清晰地表达一系列项目。



2. 图片和动画

图片比文本更加生动和直观，可以传递一些文本不能表达的信息，具有强烈的视觉冲击力。网页中的网站标识、背景、链接等都可以是图片。用户在网页中使用的图片格式主要包括 GIF、JPEG 和 PNG 等，其中使用最广泛的是 GIF 和 JPEG 两种格式。在网页中，为了更有效地吸引浏览者的注意，许多网站的广告都做成了动画形式。

3. 超链接

超链接技术是 WWW 流行起来的最主要的原因。超链接是从一个网页指向另一个目的端的链接，指定从一个位置跳转到另一个位置，可以是文本链接、图像链接和锚链接等。超链接可以在当前页面中进行跳转，也可以在页面外进行跳转。

4. 音频和视频

音频文件可使网页效果多样化，网页中常用的音频格式有 mid 和 mp3。其中 mp3 为压缩文件，其压缩率非常高，音质也不错，是背景音乐的首选。

网页中的视频文件一般为 flv 格式，它是一种基于 Flash MX 的视频流格式，具有文件小、加载速度快等特点，是网络视频格式的首选。

5. 交互表单

网页中的表单通常用来接收用户在浏览器端的输入，然后将这些信息发送到用户设置的目标端。这个目标可以是文本文件、网页和电子邮件，也可以是服务器端的应用程序。表单一般用来收集联系信息，接收用户要求，获得反馈意见，让浏览者注册为会员并以会员的身份登录站点等。

6. 其他常见元素

网页中除了以上几种最基本的元素之外，还有一些其他的常用元素，包括悬停按钮、JavaScript 与 ActiveX 等各种特效，它们不仅能点缀网页，使网页更活泼有趣，还在网上娱乐、电子商务等方面也有着不可忽视的作用。

1.3 页面布局结构

对一个网站来说，除了内容之外，还要对网站进行整体规划设计。格局凌乱的网站内容再精彩，也不是一个好网站。在网页插入各种对象和装饰效果前，要先确定网页的总体风格和布局。

1. 网页页面布局

根据不同的网页制作风格，可以将网页分为个人网页和商业网页。商业网页内容丰富、信息量大，一般都有统一的布局；个人网页风格比较多样、内容比较专一、形式比较灵活，更容易创造出美感。

确定好网页风格，要对网页的布局进行规划，也就是确定网页上网站标识、导航栏以及菜单元素等的位置。在网页设计中，结构布局是根据设计元素在网页中的位置分布进行分类的。常见的结构布局包括“国”字型、“厂”字型、海报型和 Flash 型等，如图 1-1 所示。



图 1-1 常见网页页面布局

2. 网页色彩搭配

(1) 网页安全色。网页安全色是指在不同硬件环境、不同操作系统、不同浏览器中都能够正常显示的颜色集合。在使用网页安全色进行配色后，这些颜色在任何终端用户的显示设备上都将显示为相同的效果。

网页安全色是红色、绿色和蓝色。当颜色数字信号值为 0, 51, 102, 153, 204 和 255 时，构成的颜色组合一共有 216 种颜色。

(2) 网页中色彩的表达。在网页设计中，颜色值最常见的表达方式是十六进制。十六进制是计算机中数据的一种表示方法，由数字 0~9 和字母 A~F 组成，字母不区分大小写。颜色值可以采用 6 位的十六进制来进行表示，并且需要在前面加上特殊符号“#”，如 #0E533D。还可通过 RGB, HSB, Lab 和 CMYK 来进行表示；RGB 色彩模式是通过红 (R)、绿 (G)、蓝 (B) 3 个颜色通道的变化以及相互之间的叠加来得到各种颜色，是目前运用最广的颜色系统之一。HSB 色彩模式是普及型设计软件中常见的色彩模式。Lab 颜色模型由亮度 (L) 和 a、b 两个颜色通道组成，这种颜色混合后将产生具有明亮效果的色彩。CMYK 也称作印刷色彩模式，由青、洋红 (品红)、黄和黑 4 种色彩组合成各种颜色。

(3) 常见的色彩搭配：

采用相近色配色：相近色是指相同色系的颜色，使用相近色进行网页色彩的搭配，可以使网页的效果更加统一和谐，如暖色调和冷色调就是相近色的两种运用。

采用近似色配色：在色相环中每一个颜色对面的颜色，称为互补色，也是对比最强的色组。也可以指两种明显区分的色彩，包括色相对比、明度对比、饱和度对比、冷暖对比等，如黄和蓝，紫和绿，红和青。任何色彩和黑、白、灰，深色和浅色，冷色和暖色，亮色和暗色都是对比色关系。



1.4 Web 前端技术简介

1.4.1 初识 Web 前端

Web 前端即指平常上网浏览的网页，如上网浏览新闻、查询快递信息、淘宝购物等都是在浏览网页。但网页制作还需要了解与网页相关的基本概念，下面对 Web 前端的相关概念进行详细讲解。

1991 年 8 月 6 日，来自欧洲核子研究中心的科学家 Tim Berners-Lee，启动了世界上第一个可以正式访问的网站（<http://ino.cern.ch>），标志着万维网时代的到来。随着互联网的飞速发展，网站开发人员也变得炙手可热。

Web 前端开发是从网页演变而来的，名称上有明显的时代特征。随着用户体验要求越来越高，前端开发的技术难度越来越大，Web 前端开发这个职业也从设计和制作不分的局面中独立出来。

早期的前端其实就是 Tabel 布局，后来发展到 Div+CSS 网站重构，再到 JavaScript 逐渐成为 Web 前端开发的语言以及 Web 2.0 的出现，每个阶段都涌现出相应的产品，如 SNS、博客、微博等。随着人们对网页需求的不断增大，Web 前端技术也正加速地发展。

1.4.2 Web 前端开发的三大核心技术

W3C 组织，即万维网联盟，创建于 1994 年，是 Web 技术领域最具权威和影响力的国际中立性技术标准机构。W3C 专门负责制定网页相关的标准，Web 前端的相关技术都是基于 W3C 标准实现的。

下面来介绍 Web 前端开发所包括的三大核心技术：HTML、CSS 语言和 JavaScript 语言。

1. HTML

HTML 是制作网页的标准语言。“超文本”就是指页面内可以包含图片、超链接、音乐、程序等非文字元素。超文本标记语言的结构包括“头部”（Head）和“主体”（Body）两部分，其中“头部”提供关于网页的信息，“主体”提供网页的具体内容。

2. CSS 语言

CSS 是一种用来表现 HTML 或 XML 等文件样式的计算机语言。CSS 不仅可以静态地修饰网页，还可以配合各种脚本语言动态地对网页各元素进行格式化。CSS 能够对网页中元素位置的排版进行像素级精确控制，支持几乎所有的字体字号样式，并拥有对网页对象和模型样式进行编辑的能力。

3. JavaScript 语言

JavaScript 是一种属于网络的脚本语言，已经被广泛地用于 Web 应用开发，常用来为网

页添加各式各样的动态功能，为用户提供更流畅美观的浏览效果。它的解释器被称为 JavaScript 引擎，属于浏览器的一部分，因此 JavaScript 代码由浏览器边解释边执行。通常 JavaScript 脚本通过嵌入在 HTML 中的方式来实现自身的功能。

万维网联盟（W3C）规定，Web 标准需要将网页的结构、样式和行为三者进行分离。HTML+CSS+ JavaScript 本质上构成一个 MVC 框架，即 HTML 用于描述网页的结构（Model），CSS 用于描述网页的样式（View），JavaScript 用于描述网页的行为即调度数据和实现某种呈现逻辑（Controller）。

用盖房子的例子来描述三者之间的关系。首先需要把房子的地基和结构搭建好，有一个良好的结构（HTML）。然后给房子刷上油漆和添加窗户，对房子样式进行美化（CSS）。最后给房子添加电梯和地暖，与住户进行一些行为上的交互（JavaScript），这样房子才算搭建完毕。

1.4.3 Web 前端开发工具

“工欲善其事，必先利其器。”在 HTML+CSS 开发过程中，需要先来选择适合的相关开发工具。HTML+CSS 开发过程中主要涉及三大类工具：浏览器、网页编辑器和切图软件。

1. 浏览器

浏览器是网页的运行平台，是可以把 HTML 文件展示在其中，供用户进行浏览的一种软件。目前主流的浏览器有 IE 浏览器、Chrome 浏览器、Firefox 浏览器、Safari 浏览器和 Opera 浏览器等，如图 1-2 所示。由于某些因素，这些浏览器没有完全采用统一的 Web 标准，或者说不同的浏览器对同一个 CSS 样式有不同的解析，这就导致了同样的页面在不同的浏览器下显示效果可能不同。用户使用的浏览器可能不同，因此制作网页时，需要保证该网页可以兼容所有的主流浏览器。



图 1-2 主流浏览器

下面介绍几种主流浏览器。

(1) IE 浏览器。IE 是 Internet Explorer 的简称，是由微软公司推出的一款网页浏览器，采用 Trident 内核实现，有 6.0，7.0，8.0，9.0，10.0 和 11.0 等版本。在 IE 7.0 以前，中文直译为“网络探路者”，但在 IE 7.0 以后官方便直接称为“IE 浏览器”。一些用户仍然在使用低版本的浏览器，因此在制作网页时，一般也需要兼容低版本的浏览器。一些其他的浏览器也是基于 IE 内核的，如 360 安全浏览器和搜狗浏览器等，只要兼容 IE 浏览器，这些基于 IE 内核的浏览器也都兼容。



(2) Chrome 浏览器。Chrome 浏览器一般指 Google Chrome。Google Chrome 是由 Google 公司开发的一款设计简单、高效的 Web 浏览器，采用 JavaScript 引擎，可快速运行复杂的大型网站，从而降低浏览者访问的等待时长。该浏览器基于其他开源软件撰写，采用 Webkit 和 Blink 内核实现，目标是提升稳定性、速度和安全性，并创造出简单且有效率的使用者界面。本书运行环境采用 Chrome 浏览器，版本为 540.2840.5，如图 1-3 所示。



图 1-3 Chrome 浏览器首页

(3) Firefox 浏览器。Firefox 浏览器一般指 Mozilla firefox，中文俗称“火狐”，是由 Mozilla 公司出品的一款自由的及开放源代码的 Web 浏览器，采用 Gecko 内核实现，支持多种操作系统，如 Windows 和 Mac OS X 及 GNU/Linux 等。

(4) Safari 浏览器。Safari 浏览器是由苹果公司出品的用于苹果计算机操作系统 Mac OS X 中的浏览器，采用 Webkit 内核实现，使用了 KDE 的 KHTML 作为浏览器的运算核心。无论是在 Mac 还是在 PC 上运行时，Safari 都可提供极致愉悦的网络体验方式。

(5) Opera 浏览器。Opera 浏览器是由挪威 Opera Software ASA 公司制作的一款支持多页面标签式浏览的浏览器，采用 Presto 内核实现。它是跨平台浏览器，可以在 Windows，Mac 和 Linux 3 个操作系统平台上运行。

注：浏览器内核负责对网页语法进行解释并渲染（显示）网页。通常所谓的浏览器内核也就是浏览器所采用的渲染引擎，渲染引擎决定了浏览器如何显示网页的内容以及页面的格式信息。

2. 网页编辑器

HTML，CSS 和 JavaScript 源代码文件均为纯文本内容，用计算机操作系统中自带的写

字板或记事本工具就可以打开和编辑源代码内容。因此本书不对开发工具作特定要求，使用任意一款纯文本编辑器均可以进行网页内容的编写。这里介绍几款常用的网页开发工具软件：Adobe Dreamweaver，Sublime Text，NotePad++，EditPlus 和 HBuilder。

(1) Adobe Dreamweaver。Adobe Dreamweaver 是一款所见即所得的网页编辑器，中文名称为“梦想编织者”或“织梦”。该软件最初的 1.0 版是 1997 年由美国 Macromedia 公司发布的，该公司于 2005 年被 Adobe 公司收购。Dreamweaver 也是当时第一套针对专业 Web 前端工程师所设计的可视化网页开发工具，整合了网页开发与网站管理的功能。

Dreamweaver 支持 HTML5/CSS3 源代码的编辑和预览功能，最大的优点是可视化性能带来的直观效果，开发界面可以分屏为代码部分与预览视图，当开发者修改代码部分时，预览视图会随着修改内容实时变化，如图 1-4 所示。

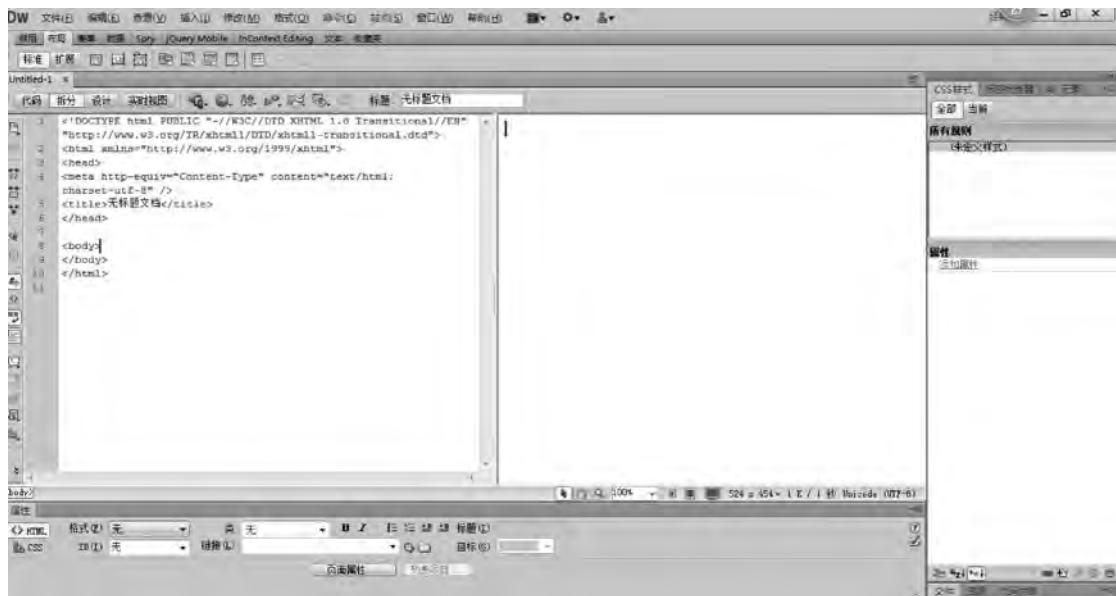


图 1-4 Dreamweaver 可视化开发界面

Dreamweaver 也有它的弱点，由于不同浏览器存在兼容性问题，Dreamweaver 的预览视图难以达到与所有浏览器完全一致的效果。如需考虑跨浏览器兼容问题，预览画面仅能作为辅助参考。

(2) Sublime Text。Sublime Text 的界面布局非常有特色，它支持文件夹导航图和代码缩略图效果。该软件支持多种编程语言的语法高亮显示，也具有代码自动完成提示功能。该软件还具有自动恢复功能，如果在编程过程中意外退出，在下次启动该软件时文件会自动恢复为关闭之前的编辑状态。

(3) NotePad ++。NotePad ++ 的名称来源于 Windows 系列操作系统自带的记事本 NotePad，在此基础上多了两个加号，立刻带来了质的飞跃。这是一款免费开源的纯文本编辑器，具有完整中文化接口并支持 UTF-8 技术，如图 1-5 所示。它具有语法高亮显示、代码折叠等功能，因此也非常适合作为计算机程序的编辑器。

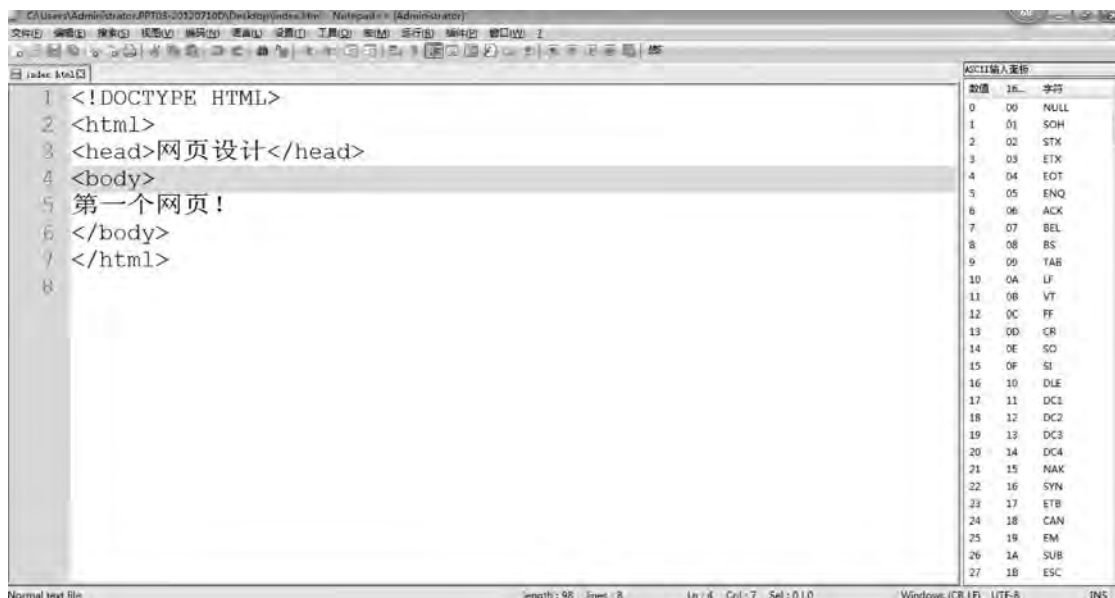


图 1-5 Notepad++开发界面

(4) EditPlus。EditPlus 是由韩国 Sangil Kim (ES-Computing) 公司发布的一款文字编辑器，支持 HTML，CSS，JavaScript，PHP 和 Java 等多种计算机程序的语法高亮显示与代码折叠功能。其中最具特色的是 EditPlus 具有自动完成功能。例如，在 CSS 源文件中输入字母 b，加上空格，就会自动生成“border: 1px solid red”语句，开发者可以自行编辑快捷键所代表的代码块，然后在开发过程中使用快捷方式让 EditPlus 自动完成指定的代码内容。

(5) HBuilder。HBuilder 是由 DCloud (数字天堂) 推出的一款支持 HTML5 的 Web 开发 IDE。HBuilder 主要用于开发 HTML，JS (JavaScript) 和 CSS，同时配合 HTML 的后端脚本语言如 PHP 和 JSP 也可以适用，还有前端的预编译语言如 less 以及人们钟爱的 markdown 都可以良好地进行编辑。从 2013 年夏天发布至今，HBuilder 已经成为业内主流的开发工具。目前，HBuilderX 发布了 1.0 正式版。

3. 切图软件

切图软件是对 UI 设计师设计出的效果图进行切图操作，也可以对网页中的图片进行修改等处理。常用的切图软件有 Photoshop 和 Fireworks 两种。

1.5 HTML 语法基础

HTML 是制作网页的基础语言，是初学者必学的内容。虽然现在有许多所见即所得的网页制作工具，但是这些工具生成的代码仍然是以 HTML 为基础的，因此学习 HTML 代码端设计网页非常重要。

1.5.1 HTML 概述

HTML 是 Hyper Text Markup Language 的缩写，即超文本标记语言，是一种用来制作超文本文档的简单标记语言。

1. 语言

HTML 作为一种超文本标记语言，有指定的语法规则，超文本传输协议规定了浏览器在运行 HTML 文档时所遵循的规则和进行的操作。协议的制定使浏览器在运行超文本时有了统一的规则 and 标准。用 HTML 编写的超文本文档称为 HTML 文档，它能独立于各种操作系统平台。自 1990 年以来 HTML 就一直被用作 WWW 的信息表示语言，使用 HTML 描述的文件，需要通过 Web 浏览器 HTTP 显示出效果。

2. 超文本

超文本是可以加入图片、声音、动画、影视等内容的文本。事实上每一个 HTML 文档都是一种静态的网页文件，这个文件里面包含了 HTML 指令代码，这些指令代码并不是一种程序语言，它只是一种排版网页中资料显示位置的标记结构语言，简单且易学易懂。

3. 标记

HTML5 实际上不算是一种编程语言，而是一种标记语言。HTML5 文件是由一系列成对出现的元素标签嵌套组合而成的。这些标签用“<”和“>”括起来。它们被称为标记，也称标签，是用来划分网页的元素，以形成文本的布局、文字的格式及五彩缤纷的画面。标签通过指定某块信息为段落或标题等来标识文档的某个部分。在 HTML 中每个用作标签的符号都是一条命令，它会告诉浏览器如何显示文本。这些标签均由“<”和“>”符号以及一个字符串组成。如<head>，<body>等。而浏览器的功能是对这些标记进行解释，显示出文字、图片、动画和播放声音。这些标签符号用“<标签名字属性>”来表示。标签分为单标签和双标签两大类：

单标签指的是只存在一个标签的写法，如<meta><input>等。

双标签指的是存在一对标签的写法，如<head> </head>，<body> </body>等。

注意，在双标签中第一个标签称为起始标签，第二个标签称为结束标签，结束标签需要在左尖括号后添加一个关闭符“/”。

HTML 只是一个纯文本文件。创建一个 HTML 文档，需要 HTML 编辑器和 Web 浏览器两个工具。HTML 编辑器是用于生成和保存 HTML 文档的应用程序。Web 浏览器用来打开 Web 网页文件，提供查看 Web 资源的客户端程序。

1.5.2 HTML 基本结构

HTML 文档是由一系列的元素和标签组成的。元素名不区分大小写，HTML 用标签来规定元素的属性和它在文件中的位置，HTML 文档分为头部和主体两部分，在文档头部对文档进行一些必要的定义，主体部分是文档要显示的信息。其基本结构见例 1.1。

【例 1.1】HTML 文档的基本结构。

```
<!--示例 1.1 程序-->
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>HTML 文档结构</title>
  </head>
  <body>
    <p>HTML 文档结构由 head,body 标记组成</p>
    <h3>标题 3</h3>
  </body>
</html>
```

本例浏览效果如图 1-6 所示。



图 1-6 文档的基本结构

1. HTML 文档标签<html>…</html>

HTML 文档标签的格式为

```
<html>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</html>为止。每个 HTML 文档均以<html>开始，以</html>结束。

2. HTML 文档头标签<head>…</head>

HTML 文档包括头部（head）和主体（body）。HTML 文档头标签的格式为

```
<head>头部的内容</head>
```

文档头部内容开始标签<html>和结束标签</html>之间定义，其内容可以是标题名或者文本文件地址、创作信息等网页信息说明。

3. 文档编码

文档编码格式如下：

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

为了被浏览器正确解释和通过 W3C 代码标准，所有的 HTML 文档都必须声明它们所使

用的编码语言。文档声明的编码应该与实际编码一致，否则会成乱码。对于中文网页的设计者来说，用户一般使用 GB2313（简体中文）。

4. HTML 文档主体标签<body>…</body>

HTML 文档主体标签的格式为

```
<body>网页的内容</body>
```

主体位于头部以后，以<body>为开始标签，</body>为结束标签。它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。

1.6 创建 HTML 文档

一个网页可以简单得只有文字，也可以复杂得既有图片、文字，又有超链接和视频、音频等。下面使用 HBuilder 快速编辑一个 HTML 文件，通过它来学习网页的编辑、保存过程。

(1) 使用 HBuilder 新建项目。依次点击选择“文件”→“新建”→“Web 项目”，如图 1-7 所示。

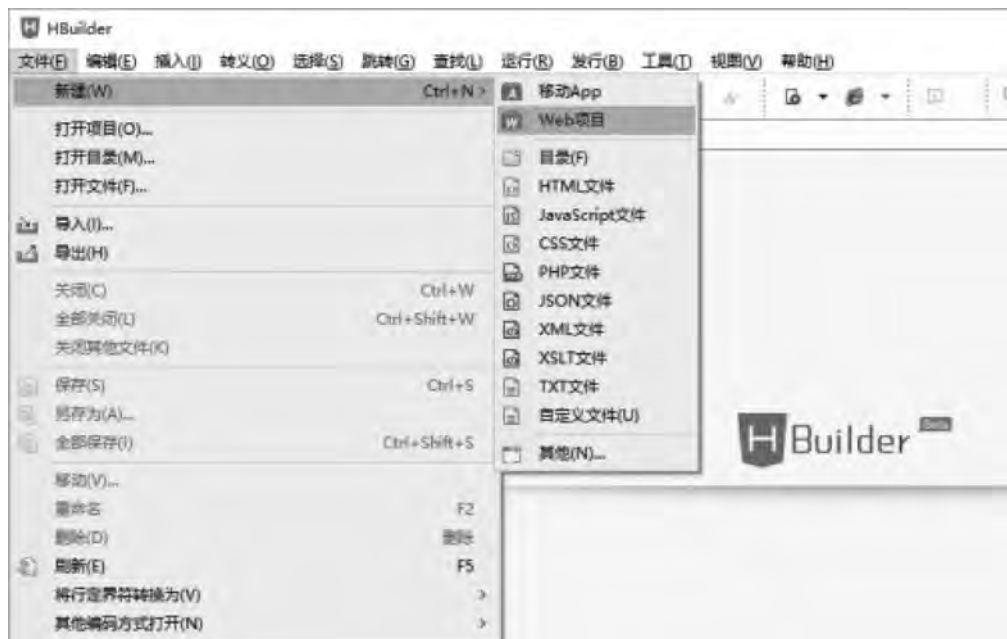


图 1-7 使用 HBuilder 新建项目

(2) 创建项目名称，设置项目所在的位置确定即可，完成之后会在左边的项目栏管理器中出现，如图 1-8 所示。在本示例中，项目名称为“课堂练习”。



图 1-8 创建 Web 项目

(3) 打开项目“课堂练习”的文件夹，会看到里面有首页 index.html，如图 1-9 所示，有 js 文件夹，有 css 文件夹，还有图片的文件夹，基本齐全。

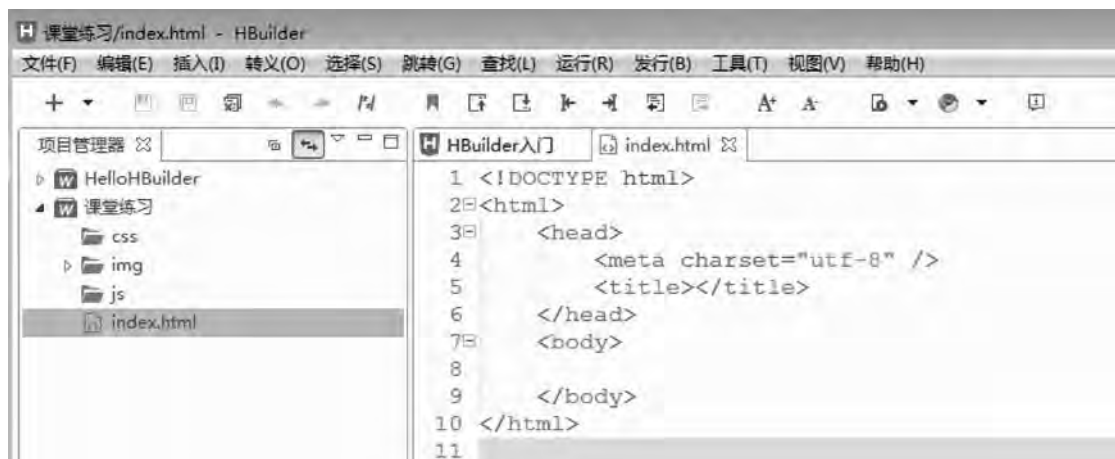


图 1-9 index 首页

(4) 到了这一步之后，便可以编写网页代码了，可以直接在 index.html 中编写代码。也

可以单击鼠标右键，新建一个 HTML 文件，如图 1-10 所示。

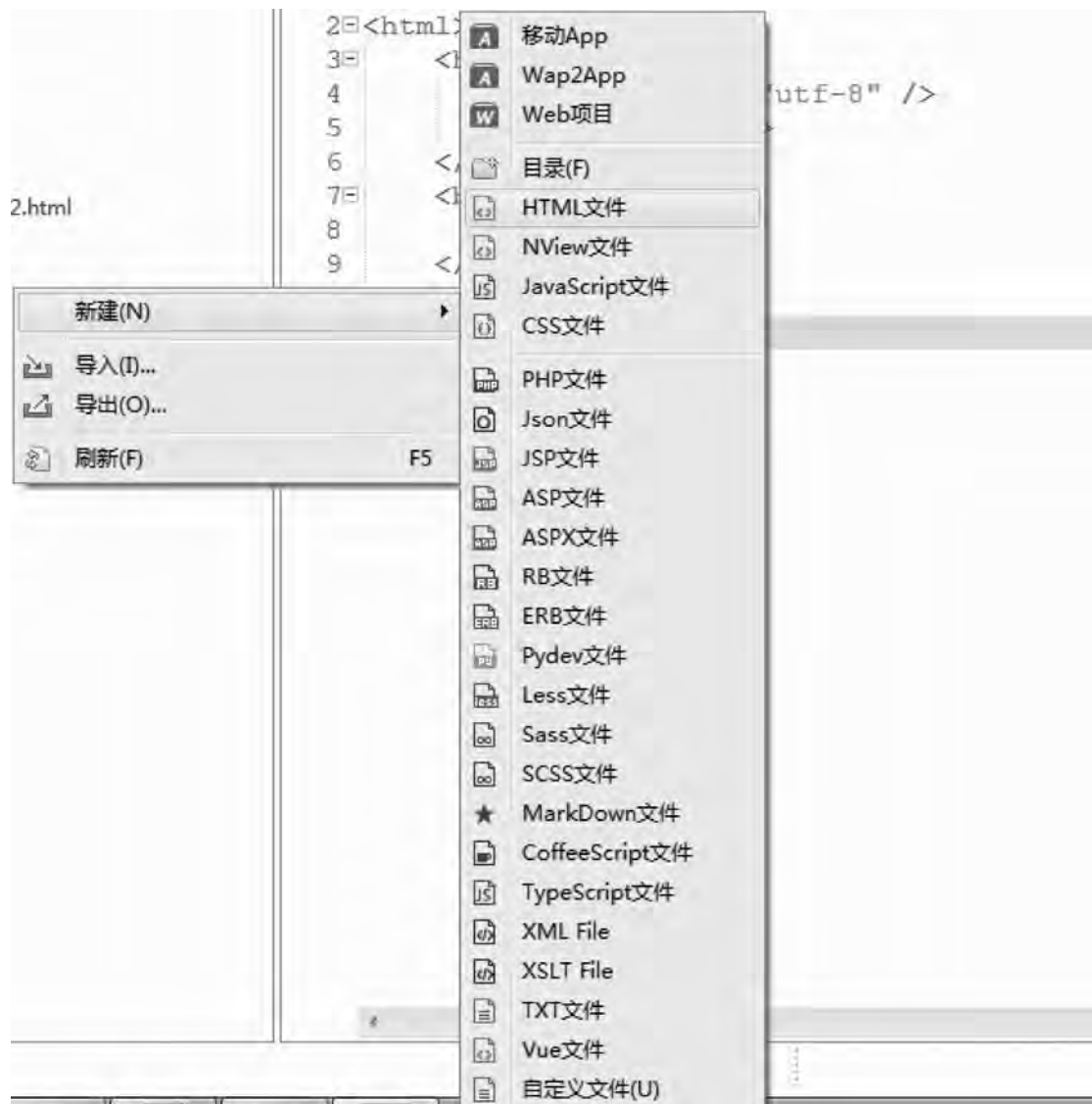


图 1-10 新建网页 (1)

(5) 在“创建文件向导”对话框中，选择好文件所在目录，并为文件命名，模板为html5，如图 1-11 所示。



图 1-11 新建网页 (2)

(6) 本实例将文件命名为“示例程序 1.2. html”，开始编写网页代码，如图 1-12 所示。

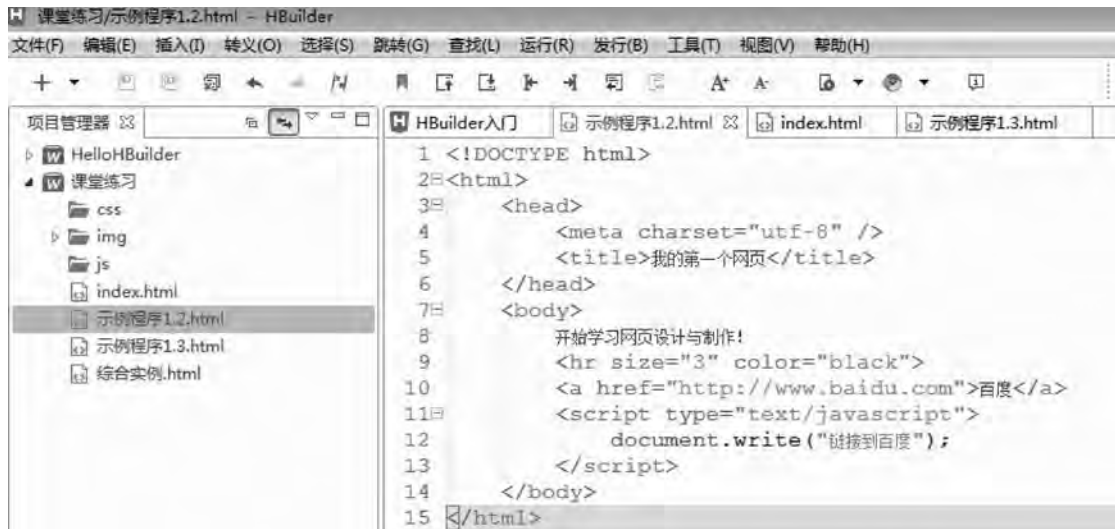


图 1-12 编辑网页

(7) 网页完成后，可以执行菜单“运行”→“浏览器运行”，选择电脑中已有的浏览

器进行预览，本书使用 Chrome 浏览器，如图 1-13 所示。



图 1-13 预览网页

创建 HTML 文档的完整代码见例 1.2。

【例 1.2】 HTML 文档的创建。

```
<! --示例 1.2 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>我的第一个网页</title>
  </head>
  <body>
    开始学习网页设计与制作!
    <hr size="3" color="black">
    <a href="http://www.baidu.com">百度</a>
    <script type="text/javascript">
      document.write("链接到百度");
    </script>
  </body>
</html>
```

1.7 网页头部标签

在网页的头部中，通常存放一些介绍页面内容的信息。例如，页面标题、描述和关键词链接的 CSS 样式文件和客户端的 JavaScript 脚本文件等。其中，页面标题及页面描述称为页



面的摘要信息，摘要信息的生成在不同的搜索引擎中会存在比较大的差别，即使是同一个搜索引擎也会由于页面的实际情况而有所不同。一般情况下，搜索引擎会提取页面标题标签中的内容作为摘要信息的标题，而描述则常来自页面描述标签的内容或直接从页面正文中截取。如果希望自己发布的网页能被百度等搜索引擎搜索，那么在制作网页时就需要注意编写网页的摘要信息。

1.7.1 <title>标签

<title>标签是页面标题标签。它将 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。这个标签只能应用于<head>与</head>之间。<title>标签是对文件内容的概括。一个好的标题能使读者从中判断出该文件的大概内容。

网页的标题不会显示在文本窗口中，而以窗口的名称显示出来，每个文档只允许有一个标题。网页的标题能给浏览者带来方便，如果浏览者喜欢该网页，将它加入书签中或保存到磁盘上，标题就作为该页面的标志或文件名。另外，使用搜索引擎时显示的结果也是页面的标题。

<title>标签位于<head>与</head>中，用于标示文档标题，格式如下：

```
<title>标题</title>
```

例如，京东商城的文档标题：

```
<title>京东(JD.COM)-正品低价、品质保障、配送及时、轻松购物! </title>
```

打开网页后，在网页文档头部定义的标题内容不在浏览器窗口中显示，而是在浏览器的标题栏中显示，尽管文档头部定义的信息很多，但能在浏览器标题栏中显示的信息只有标题内容。

1.7.2 <meta>标签

<meta>标签是元信息标签，在 HTML 中是一个单标签。该标签可重复出现在头部标签中，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>标签分两大属性：HTTP 标题属性（http-equiv）和页面描述属性（name）。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。本节主要讲解的 name 属性，用于设置搜索关键字和描述。<meta>标签的 name 属性的语法：

```
<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 中的内容主要是便于搜索引擎查找信息和分类信息用。

name 属性主要有以下两个参数：keywords（关键字）和 description（网站内容描述）。

1. keywords

keywords 用来告诉搜索引擎网页使用的关键字。例如，著名的京东商城网，其主页的关

键字设置如下：

```
<meta name="keywords" content="网上购物,网上商城,家电,手机,电脑,服装,居家,母婴,美妆,个护,食品,生鲜,京东" />
```

2. description

description 用来告诉搜索引擎网站主要的内容。例如，京东商城网站主页的内容描述设置如下：

```
<meta name="description" content="京东 JD.COM-专业的综合网上购物商城,为您提供正品低价的购物选择、优质便捷的服务体验。商品来自全球数十万品牌商家,囊括家电、手机、电脑、服装、居家、母婴、美妆、个护、食品、生鲜等丰富品类,满足各种购物需求。" />
```

当浏览者通过百度搜索引擎（百度）搜索“京东”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 1-14 所示。



图 1-14 在百度搜索“京东”

1.7.3 <link>标签

<link>标签用于连接外部资源和当前 HTML 文档，它只在首部标签<head>和</head>中，



通常用于连接外部样式表。<link>标签并不将其他文档实际链接到当前文档中，只是提供链接该文档的一个路径，link 标签最常用的是用来链接 CSS 样式文件，格式如下：

```
<link rel="stylesheet" href="外部样式表文件名.css" type="text/css" />
```

如果需要同时引用多个外部样式表文件，则需要为每一个 CSS 样式文件单独使用一次 <link> 标签。

1.7.4 <script>标签

<script>标签是脚本标签，用于为 HTML 文档定义客户端脚本信息。此标签可以在文档中包含一段客户端脚本程序，可以位于文档中任何位置，但常位于<head>标签内，以便于维护。格式如下：

```
<script type="text/css" src="脚本文件名.n.js"></script>
```

1.8 HTML5 文档注释和特殊符号

1.8.1 注释

为增加 HTML5 文档的可读性，可为其添加注释部分。注释是文档中的说明文字，不会被浏览器执行。HTML5 使用<! --...-->标签为文档进行注释，注释标签以“<! --”开头，以“-->”结束，中间的“...”替换为注释文字内容即可。<! --...-->标签支持单行和多行注释。

【例 1.3】 注释标签的应用。

```
<! --示例 1.3 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>带有注释语句的页面</title>
  </head>
  <body>
    你好,HTML5!
    <! --接下来出现一条水平线-->
    <hr size="3" color="black">
    <! --
      这是一个多行注释语句
      注释语句不会被浏览器执行
```


1.9 综合案例——临江仙·送钱穆父

以一首宋词为例子，参照给定的 HTML 代码和图片资源，利用 HBuilder 设计 Web 网页。在编写时要遵守 HTML 代码的编写规范，并养成良好的编码习惯。

以下为完整代码，以供参考。

```
<! --综合案例-->
<! DOCTYPE html>
<html>
  <head>
    <meta name="keywords" content="宋词,苏轼" charset="utf-8" />
    <meta name="description" content="本网站收录精选宋词" />
    <title>宋词精选</title>
    <style type="text/css">
      p{
        text-align:center;
        font-size:larger;
      }
    </style>
  </head>
  <body bgcolor="antiquewhite" text="#333333" >
    <h2 align="center">临江仙·送钱穆父</h2>
    <p>宋 苏轼</p>
    <! --使用<br/>的效果-->
    <p>一别都门三改火,天涯踏尽红尘。<br />
      依然一笑作春温。<br />
      无波真古井,有节是秋筠。<br />
      惆怅孤帆连夜发,送行淡月微云。<br />
      尊前不用翠眉颦。<br />
      <font color="brown" face="微软雅黑">人生如逆旅,我亦是行人。</font></p>
    
    <! --水平线-->
    <hr size="2" color="black" width="100%" />
    <p align="center">网页制作教程 Copyright&copy; 广东南方职业学院</p>
  </body>
</html>
```

本例浏览效果如图 1-16 所示。



图 1-16 《临江仙·送钱穆父》网页效果

本章习题

1. 填空题。

(1) HTML 是一种_____语言。

(2) HTML 文档是由_____构成的_____文件。

(3) HTML 全程是_____。

(4) 目前常用的主流网络浏览器有_____、_____、_____、_____和_____。

(5) HTML 文档通常以_____或_____作为后缀名，网站的首页文件通常命名为_____。

(6) HTML 文档是用来描述网页的，一般由_____和_____两部分组成。

2. 使用 HBuilder 编写符合以下要求的文档：网页标题为“网页学习”，在浏览器窗口中显示“欢迎大家一起开始学习网页制作”。完成效果如图 1-17 所示。其中网页所有文字的颜色为 blue，背景颜色为 #99ffff；水平分割线粗细为 5，颜色为 #ff3333。



图 1-17 网页学习网页效果

3. 使用 HBuilder 编写符合以下要求的文档：网页标题为“百度搜索”，在浏览器窗口中显示“欢迎使用百度搜索”和百度图片，完成效果如图 1-18 所示。其中网页背景颜色为 #fff333；水平分割线粗细为 5，颜色为 #0033ff。



图 1-18 百度搜索网页效果

第 2 章 网页制作的排版方法



学习目标

1. 理解文字与段落排版的方法；
2. 掌握超链接的设置方法；
3. 掌握图像的设置方法；
4. 掌握列表的使用方法。



知识要点

1. 文字和段落排版；
2. 超链接的概念和应用；
3. 图文混排；
4. 无序列表、有序列表的设置。

2.1 文字与段落排版

在网页制作过程中，通过文字与段落的基本排版即可制作出简单的网页。以下讲解常用的文字与段落排版所使用的标签。

2.1.1 段落标签

由于浏览器忽略用户在 HTML 编辑器中输入的回车符，为了使文字段落排列得整齐、清晰，常用段落标签 `<p>…</p>` 实现这一功能。段落标签 `<p>` 是 HTML 格式中特有的段落元素，在 HTML 格式里不需要在意文章每行的宽度，不必担心文字是不是太长了而被截掉，它会根据窗口的宽度自动转换到下一行。段落标签的语法为：

```
<p align="left|center|right">文字</p>
```

属性 `align`：设置段落文字在网页上的对齐方式，包括 `left`（左对齐）、`center`（居中）和 `right`（右对齐），默认为 `left`。

格式中的“|”表示“或者”，即多项选其一。

【例 2.1】 列出包含 `<p>` 标签的多种属性。

```
<! --示例 2.1 程序-->
```

```
<! DOCTYPE html>
```

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>段落 P 标签示例</title>
  </head>
  <body>
    <p align="center">广东南方职业学院</p>
    <p align="right">作者:信息与计算机学院</p>
    <p align="left">信息与计算机学院开设了计算机应用技术、软件技术、数字媒体应用技术、艺术设计、服装与服饰设计、服装设计与工艺和大数据技术与应用等七个专业。其中软件技术专业是我院首批重点建设专业。</p>
    <p align="center">Copyright &copy;2019 信息与计算机学院</p>
  </body>
</html>
```

本例浏览效果如图 2-1 所示。



图 2-1 <p>标签示例效果

以上示例中，段落标签会在段落前后加上额外的空行，不同段落间的间距等于连续加了两个换行标签
，用以区别文字的不同段落。

2.1.2 标题标签

在页面中，标题是一段文字内容的核心，所以总是用加强的效果来表示。标题使用<h1>至<h6>标签进行定义。<h1>定义最大的标题，<h6>定义最小的标题，HTML 会自动在标题前后添加一个额外的换行。标题文字标签的格式为：

```
<h #align="left|center|right">标题文字</h#>
```

属性 align：设置标题在页面中的对齐方式，包括 left（左对齐）、center（居中）或 right

(右对齐), 默认为 left。

【例 2.2】列出 HTML 中的各级标题, 本例浏览效果如图 2-2 所示。

```
<! --示例 2.2 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>标题示例</title>
  </head>
  <body>
    <h1>一级标题</h1>
    <h2>二级标题</h2>
    <h3>三级标题</h3>
    <h4>四级标题</h4>
    <h5>五级标题</h5>
    <h6>六级标题</h6>
  </body>
</html>
```



图 2-2 各级标题

2.1.3 换行标签

网页内容并不都是像段落那样, 有时没有必要用多个<p>标签去分割内容。如果编辑网

页内容只是为了换行，而不是从新段落开始，可以使用
标签。

标签将打断 HTML 文档中正常段落的行间距和换行。
放在任意一行中都会使该行换行，如果
放在一行的末尾，可以使后面的文字、图像、表格等显示于下一行，而又不会在行与行之间留下空行，即强制文本换行。换行标签的语法：

文字

浏览器解释时，从该处换行。换行标签单独使用，可使页面清晰、整齐。

【例 2.3】制作信息与计算机学院联系方式的页面。

```
<! --示例 2.3 程序-->
```

```
<! DOCTYPE html>
```

```
<html>
```

```
  <head>
```

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

```
    <title>换行标签</title>
```

```
  </head>
```

```
  <body>
```

```
    <h3>联系我们</h3>
```

```
    联系地址:广东省江门市五邑路683号(市内13和25路公共汽车到校门口下车)<br />
```

```
    联系电话:+86-750-3963333(20条线)<br />
```

```
    传真电话:+86-750-3073898 3073890<br />
```

```
    网址:WWW.GDNFU.COM<br />
```

```
  </body>
```

```
</html>
```

本例浏览效果如图 2-3 所示。



图 2-3 联系方式的页面效果

2.1.4 水平线标签

水平线可以作为段落与段落之间的分隔线，使文档结构清晰，层次分明。当浏览器解释到 HTML 文档中的 `<hr/>` 标签时，会在此处换行，并加入一条水平线段。

水平线标签的格式为：

```
<hr align=" left | center | right" size = " 横线粗细" width = " 横线长度" color = " 横线色彩" noshade = " noshade" />
```

属性 `size`：设定线条粗细，以像素为单位，默认值为 2。

属性 `width`：设定线段长度，可以是绝对值（以像素为单位）或相对值（相对于当前窗口的百分比）。所谓绝对值，是指线段的长度是固定的，不随窗口尺寸的改变而改变。所谓相对值，是指线段的长度相对于窗口的宽度而定，窗口的宽度改变时，线段的长度也随之增减，默认值为 100%，即始终当前窗口。

属性 `color`：设定线条色彩，默认为黑色。色彩可以用相应的英文名称或以“#”引导的一个十六进制代码来表示，见表 2-1。

表 2-1 色彩代码

| 色彩 | 色彩英文名称 | 十六进制代码 | 色彩 | 色彩英文名称 | 十六进制代码 |
|-----|--------|---------|-----|-------------|---------|
| 黑色 | black | #000000 | 粉红色 | pink | #ffc0cb |
| 蓝色 | blue | #0000ff | 红色 | red | #ff0000 |
| 棕色 | brown | #a52a2a | 白色 | white | #ffffff |
| 青色 | cyan | #00ffff | 黄色 | yellow | #ffff00 |
| 灰色 | gray | #808080 | 深红色 | crimson | #cd061f |
| 绿色 | green | #008000 | 黄绿色 | greenyellow | #0b6eff |
| 乳白色 | ivory | #ffff00 | 紫色 | purple | #800080 |
| 橘黄色 | orange | #ffa500 | 淡紫色 | lavender | #d8bfd8 |

【例 2.4】 `<hr/>` 标签的基本用法。

代码如下：

```
<! --示例 2.4 程序-->
```

```
<! DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

```
<title>hr 标签示例</title>
```

```
</head>
```

```
<body>
```

```
<p>关于 2016 届信息与计算机学院优秀毕业生的名单</p>
```

```
<hr color=" blue" />
```

```
13 软件技术 1 班(9 名):杜其亮,梁冠辉,梁泽民,赖辉旭,陈祖涛,林伟涛,刘付靖,陈俊明,林
```

```
坚辉<br />
```

13 计算机应用 1 班(9 名):黄燕飞,麦指眉,林海林,李斯伟,樊启星,李碧云,徐长卿,吴学鹏,

```
尹雪飞<br />
```

```
</body>
```

```
</html>
```

本例浏览效果如图 2-4 所示。



图 2-4 <hr/>标签的基本用法实例效果

以上示例中, <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)。

2.1.5 预格式化标签

<pre>标签可定义预格式化的文本。被包围在<pre>标签中的文本通常会保留空格和换行符, 而文本也会呈现为等宽字体。<pre>标签的一个常见应用就是用来表示计算机的源代码。预格式化标签的语法为:

```
<pre>文本块</pre>
```

【例 2.5】 <pre>标签的基本用法。

代码如下:

```
<! --示例程序 2.5-->
```

```
<! DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

```
<title>pre 标签示例</title>
</head>
<body>
  <pre>
    这是
    预格式文件。
    它保留了空格和换行。
  </pre>
  <p>pre 标签很适合显示计算机代码:</p>
  <pre>
    for i=1 to 10
      print i
    next i
  </pre>
</body>
</html>
```

本例浏览效果如图 2-5 所示。

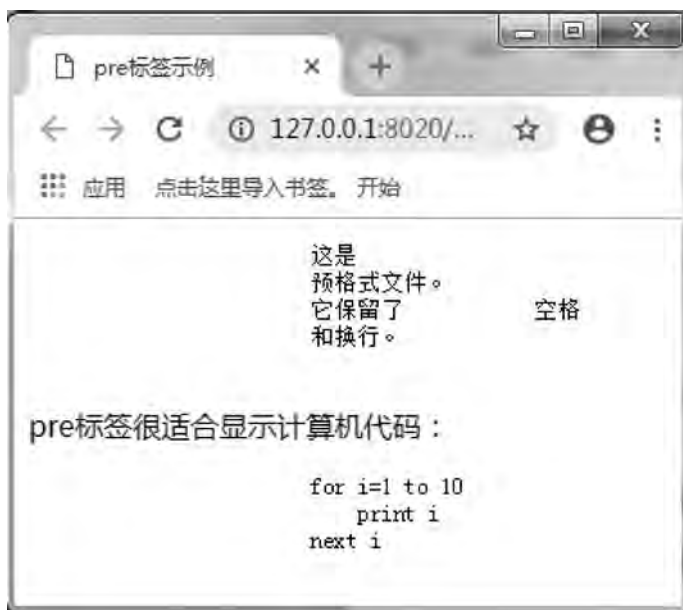


图 2-5 <pre>标签的基本用法实例效果

2.1.6 缩排标签

<blockquote>标签可定义一个块引用。<blockquote>与</blockquote>之间的所有文本都会从常规文本中分离出来，经常会在左、右边进行缩进，而且有时会使用斜体。也就是说，块引用拥有它们自己的空间。缩排标签的语法为：

```
<blockquote>文本</blockquote>
```

【例 2.6】 <blockquote>标签的基本用法。

```
<! --示例 2.6 程序-->
```

```
<! DOCTYPE html>
```

```
<html>
```

```
  <head>
```

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

```
    <title>blockquote 标签示例</title>
```

```
  </head>
```

```
  <body>
```

```
    这里有一段长文本引用：
```

```
    <blockquote>广东南方职业学院是经广东省人民政府批准、国家教育部备案的全日制普通高等院校。办学以来,学校坚持以就业为导向,把人才培养质量放在学校工作的首位,针对广东省特别是珠三角区域经济发展需求和新兴人工智能、智能制造的行业发展趋势,不断调整和优化专业结构,形成了工、经、管、文等学科全面发展的专业布局。</blockquote>
```

```
    * 请注意,浏览器在blockquote 标签前后添加了换行,并增加了外边框。
```

```
  </body>
```

```
</html>
```

本例浏览效果如图 2-6 所示。



图 2-6 <blockquote>标签的基本用法实例效果

2.1.7 案例

【例 2.7】 使用文字与段落的基本排版知识制作“全球汇商城简介页面”。本例浏览效果如图 2-7 所示。

假货必赔:第三方保障,中国人保正品保险,假货必赔。

</blockquote>

</body>

</html>

2.2 超 链 接

一个网站是由多个页面组成的,创建超链接有利于页面与页面之间的跳转,从而将整个网站中的页面有机地连接起来,它是网页中至关重要的元素。超链接在本质上属于网页的一部分,通过超链接将各个网页链接在一起后,才能真正构成一个网站。

2.2.1 超链接简介

1. 超链接的定义

所谓超链接(Hyperlink),是指从一个网页指向一个目标的连接关系,这个目标可以是另一个网页,也可以是相同网页上的不同位置,还可以是一个图片,一个电子邮件地址,一个文件,甚至是一个应用程序。

超链接除了可链接文本外,也可链接各种媒体,如声音、图像和动画等,通过它们可以将网站建设成一个丰富多彩的多媒体世界。当网页中包含超链接时,其外观形式为彩色(一般为蓝色)且带下划线的文字或图像。单击这些文本或图像,可跳转到相应位置。鼠标指针指向超链接时,将变成手形。

2. 超链接的分类

根据超链接目标文件的不同,超链接可分为页面超链接、锚点超链接和电子邮件超链接等;根据超链接单击对象的不同,超链接可分为文字超链接、图像超链接和图像映射等。

3. 路径

URL——统一资源定位器,指的就是每一个网站都具有的独立的地址。同一个网站下的每一个网页都属于同一个地址下,但是,当创建网页时,不可能也不需要为每一个链接都输入完全的地址。用户只需要确定当前文档同站点根目录之间的相对路径关系。

创建超链接时必须了解链接与被链接文本的路径。在一个网站中,路径通常有3种表示方式:绝对路径、根目录相对路径和文档目录相对路径。

(1) 绝对路径。绝对路径就是主页上的文件或目录在硬盘上真正的路径(URL和物理路径)。例如,D:\web\index.html代表了index.html文件的物理绝对路径;http://www.hao123.com/index.html代表了一个URL绝对路径。

(2) 根目录相对路径。根目录相对路径是指从站点根文件夹到被链接文档经过的路径。站点上所有公开的文件都存放在站点的根目录下。站点根目录相对路径以一个正斜杠(/)开始。例如,/support/tips.htm是文件(tips.htm)的站点根目录相对路径,该文件位于站点根文件夹的support子文件夹中。

(3) 文档目录相对路径。它是相对于某个基准目录的路径，以当前文件所在的路径为起点，进行相对文件的查找。相对路径适合于创建网站内部链接。

2.2.2 超链接的应用

超链接的另外一个叫法称为锚，它是使用<a>标签标记的，可以用两种方式表示，一是通过使用<a>标签的 href 属性来创建超文本链接，以链接到同一文档的其他位置或其他文档中，在这种情况下，当前文档就是链接的源，href 属性的值就是 URL 的目标；二是通过使用<a>标签的 name 属性或 id 属性在文档中创建一个文档内部的书签。

1. 锚点标签<a>...

HTML 使用<a>标签来建立一个链接，通常<a>标签又称为锚。建立超链接的标签以<a>开始，以结束。锚可以指向网络上的任何资源：一张 HTML 页面、一幅图像、一个声音或视频文件等。<a>标签的语法：

```
<a href="url" title="指向链接显示的文字" target="窗口名称">文本文字</a>
```

用户可以单击<a>和标签之间的文本文字来实现网页的浏览访问，通常<a>和标签之间的文本文字用颜色和underline加以强调。

建立超链接时，href 属性定义了这个超链接所指的目标地址，也就是路径。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。

属性 target：设定链接被单击后所要开始窗口的方式有以下 4 种：

- (1) blank。在新窗口中打开被链接的文档。
- (2) self。默认值。在相同的框架中打开被链接的文档。
- (3) parent。在父框架集中打开被链接的文档。
- (4) top。在整个窗口中打开被链接的文档。

2. 指向其他页面的超链接

创建指向其他页面的超链接，就是在当前页面与其他相关页面之间建立超链接。根据目标文件与当前文件的目录关系，有 4 种写法。注意，应该尽量采用相对路径。

(1) 链接到同一目录内的网页文件，语法：

```
<a href="目标文件名.html">热点文本</a>
```

其中，“目标文件名”是链接所指向的文件。

(2) 链接到下一级目录中的网页文件，格式：

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

(3) 链接到上一级目录中的网页文件，语法：

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

其中，“../”表示退到上一级目录中

(4) 链接到同级目录中的网页文件，格式：

```
<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，然后再进入目标文件所在的目录。

【例 2.8】制作全球汇商城页面之间的链接，分别链接指向注册页和登录页，本例浏览效果如图 2-8 所示。



图 2-8 页面之间的链接效果图

代码如下：

```
<! --示例 2.8 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>页面之间的链接</title>
  </head>
  <body>
    <a href="register.html">免费注册</a><! --链接到同一目录内的网页文件-->
    <a href="login.html">账户登录</a><! --链接到同一目录内的网页文件-->
  </body>
</html>
```

3. 指向书签的超链接

在浏览页面时，如果页面篇幅很长，要不断地拖动滚动条，就给用户浏览带来不便。浏览者需要既可以从头阅读到尾，又可以很快找到自己感兴趣的特定内容进行部分阅读时，就可以通过书签链接来实现。当浏览者单击页面上的某一“标签”，就能自动跳到网页相应的位置进行阅读，给浏览者带来方便。

书签就是用<a>标签对网页元素作一个记号，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链接，对不同目标元素要设置不同的书签名。书签

名在<a>标签的 name 属性中定义，语法：

```
<a name="记号名">目标文本附近的内容</a>
```

(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：

```
<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 2.9】制作“杜甫诗词全集”网页，完成指向页面内书签的超链接，本例浏览效果如图 2-9 所示。



图 2-9 “杜甫诗词全集”网页

代码如下：

```
<! --示例 2.9 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>指向页面内书签的超链接</title>
  </head>
  <body>
    <h1>杜甫诗词全集</h1>
    <a href="#one">1. 《绝句》</a>
    <a href="#two">2. 《登高》</a>
```



```

<a href="#three">3.《茅屋为秋风所破歌》</a>
<a href="#four">4.《望岳》</a>
<a href="#five">5.《春望》</a>
<hr color="chocolate"/>
<br />
<br />
<a name="one">《绝句》</a>
<p>两个黄鹂鸣翠柳,<br/>一行白鹭上青天。<br/>窗含西岭千秋雪,<br/>门泊东吴万里船。
<br/></p>
<a name="two">《登高》</a>
<p>风急天高猿啸哀,<br/>渚清沙白鸟飞回。<br/>无边落木萧萧下,<br/>不尽长江滚滚来。
<br/>万里悲秋常作客,<br/>百年多病独登台。<br/>
艰艰苦恨繁霜鬓,<br/>潦倒新停浊酒杯。<br/></p>
<a name="three">《茅屋为秋风所破歌》</a>
<p>八月秋高风怒号,<br/>卷我屋上三重茅。<br/>
茅飞渡江洒江郊,<br/>高者挂胃长林梢,<br/>下者飘转沉塘坳。<br/>
南村群童欺我老无力,<br/>忍能对面为盗贼,<br/>公然抱茅入竹去。<br/>
唇焦口燥呼不得,<br/>归来倚杖自叹息。<br/>
俄顷风定云墨色,<br/>秋天漠漠向昏黑。<br/>
布衾多年冷似铁,<br/>娇儿恶卧踏里裂。<br/>
床头屋漏无干处,<br/>雨脚如麻未断绝。<br/>
自经丧乱少睡眠,<br/>长夜沾湿何由彻? <br/>
安得广厦千万间,<br/>大庇天下寒士俱欢颜,<br/>风雨不动安如山! <br/>
呜呼! <br/>何时眼前突兀见此屋,<br/>吾庐独破受冻死亦足! <br/></p>
<a name="four">《望岳》</a>
<p>南岳配朱鸟,秩礼自百王。歎吸领地灵,鸿洞半炎方。<br/>
邦家用祀典,在德非馨香。巡守何寂寥,有虞今则亡。<br/>
洎吾隘世网,行迈越潇湘。渴日绝壁出,漾舟清光旁。<br/>
祝融五峰尊,峰峰次低昂。紫盖独不朝,争长彙相望。<br/>
恭闻魏夫人,群仙夹翱翔。有时五峰气,散风如飞霜。<br/>
牵迫限修途,未暇杖崇冈。归来覬命驾,沐浴休玉堂。<br/>
三叹问府主,曷以赞我皇。牲璧忍衰俗,神其思降祥。</p>
<a name="five">《春望》</a>
<p>国破山河在,<br/>城春草木深。<br/>感时花溅泪,<br/>恨别鸟惊心。<br/>烽火连三月,
<br/>家书抵万金。<br/>白头搔更短,<br/>浑欲不胜簪。<br/></p>
</body>
</html>

```

(2) 指向其他页面书签的超链接。书签超链接还可以在不同页面间进行链接。当单击书签超链接标题,页面会根据超链接中的 href 属性所定的地址,将网页跳转到目标地址中书签名称所表示的内容。要在其他页面内实现书签链接,需要定义两个标签:一个为当前页面的超链接标签,另一个为跳转页面的书签标签。当前页面的超链接标签的语法:

```
<a href="目标文件名.html #记号名">热点文本</a>
```

单击“热点文本”,将跳转到目标页面“记号名”开始的网页元素。

【例 2.10】制作指向其他页面书签的链接,在 intro.html 页面“企业简介”文本前定义

一个书签 intro，当单击当前页面示例程序 2.10.html 中的“企业简介”超链接时，将跳转到页面 intro.html 中企业简介位置处，本例浏览效果如图 2-10 所示。

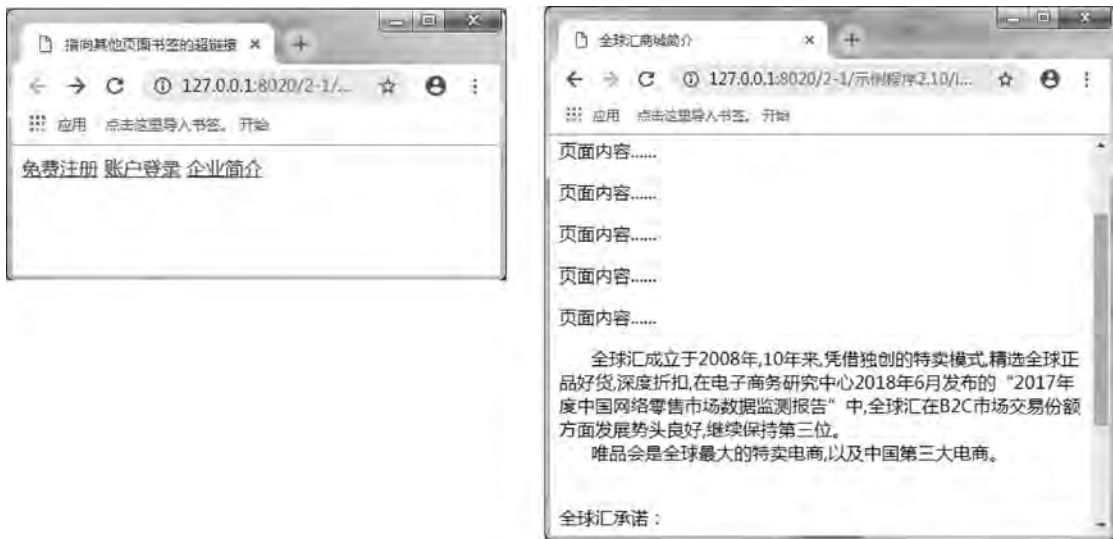


图 2-10 指向其他页面书签的超链接网页效果

代码如下：

```
<! --示例 2.10 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>指向其他页面书签的超链接</title>
  </head>
  <body>
    <a href="register.html">免费注册</a><! --链接到同一目录内的网页文件-->
    <a href="login.html">账户登录</a><! --链接到同一目录内的网页文件-->
    <a href="intro.html#intro">企业简介</a><! --链接到页面 intro.html 内的书签 intro-->
  </body>
</html>
```

跳转页面 intro.htm 的代码如下：

```
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>全球汇商城简介</title>
  </head>
  <body>
    <h1 align="center">100% 正品 十重保障</h1>
    <hr color="deeppink" />
    <p>页面内容……</p>
```


PNG。下面进行简单的介绍，以便初学者在网页制作过程中做出适当的选择。

(1) GIF。GIF 是 Internet 上应用最广泛的图像文件格式之一，是一种索引颜色的图像格式，此格式在网页中使用较多。它的特点是体积小，支持小型翻页型动画，GIF 图像最多可以使用 256 种颜色，最适合制作徽标、图标、按钮和其他颜色、风格比较单一的图片。

(2) JPEG。JPEG 是 Internet 上应用最广泛的图像文件格式之一，适用于摄影或连续色调图像。JPEG 文件可以包含多达数百万种颜色，因此 JPG 格式的文件体积较大，图片质量较佳，通常可以通过压缩 JPG 文件在图像品质和文件大小之间取得良好的平衡。当网页中对图像的质量有要求时，建议使用此格式。

(3) PNG。PNG 是一种新型的无专利权限的图像格式，兼有 GIF 和 JPG 的优点。它的显示速度很快，只需下载 1/64 的图像信息就可以显示出低分辨率的预览图像。它可以用来代替 GIF 格式，同样支持透明层，在质量和体积方面都具有优势，适合在网络中传输。

2. 使用网页图像的要点

(1) 高质量的图像因其图像体积过大，不太适合网络传输。一般在网页设计中选图像不要超过 8 KB，如必须选用较大图像时，可先将其分成若干小图像，显示时再通过将这些小图像拼合起来。

(2) 网页中的动画并非越多越好，页面中应合理使用动画，太炫酷的动画会分散网站访客的注意力。

(3) 如果在同一文件中多次使用相同的图像时，最好使用相对路径查找该图像。相对路径是相对于文件而言的，从相对文件所在目录依次往下直到文件所在的位置。例如，文件 X.Y 与 A 文件夹在同一目录下，那么文件 B.A 在目录 A 下的 B 文件夹中，它对于文件 X.Y 的相对路径则为 A/B/B.A，如图 2-11 所示。

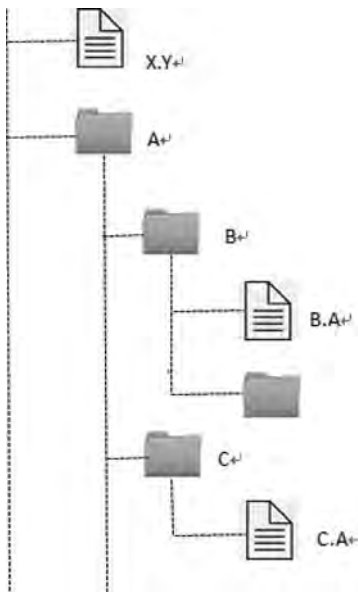


图 2-11 相对路径

2.3.2 图像标签

在 HTML 中，用标签在网页中添加图像，图像是以嵌入的方式添加到网页中的。图像标签的格式：

```

```

标签中的属性说明见表 2-2，其中 src 是必需的属性。

需要注意的是，在 width 和 height 属性中，如果只设置了其中的一个属性，则另一个属性会根据已设置的属性按原图等比例显示。如果对两个属性都进行了设置，且其比例和原图大小的比例不一致，那么显示的图像会相对于原图变形或失真。

表 2-2 图像标签的常用属性

| 属 性 | 说 明 |
|--------|---|
| src | 指定图像源，即图像的 URL 路径 |
| alt | 如果图像无法显示，代替图像的说明文字 |
| title | 为浏览者提供额外的提示或帮助信息，方便用户使用 |
| width | 指定图像的显示宽度（像素数或百分数），通常只设为图像的真实大小以免失真。若需要改变图像大小，最好事先使用图像编辑工具进行修改。百分数是指相对于当前浏览器窗口的百分比 |
| height | 指定图像的显示高度（像素数或百分数） |
| border | 指定图像的边框大小，用数字表示，默认单位为像素，默认情况下图片没有边框，即 border=0 |
| align | 指定图像的对齐方式，设定图像在水平（环绕方式）或垂直方向（对齐方式）上的位置，包括 left（图像居左，文本在图像的右边）、right（图像居右，文本在图像的左边）、top（文本与图像在顶部对齐）、middle（文本与图像在中央对齐）或 bottom（文本与图像在底部对齐） |

1. 图像的替换文本说明

有时，由于网速过慢或者用户在图片还没有下载完全就单击了浏览器的停止键，在浏览器中就看不到要看到图片，这时替换文本说明就十分有必要了。替换文本说明应该简洁而清晰，能为用户提供足够的图片说明信息，在无法看到图片的情况下也可以了解图片的内容信息。

【例 2.11】图像的替换文本说明，正常显示的效果如图 2-12 所示；当显示的图像路径错误时，效果如图 2-13 所示。



图 2-12 正常的显示图像效果



图 2-13 图像路径错误时的显示效果


```
<a href="waitao.html"> <! --单击图像则打开 waitao.html-->

</a>
```

去除图像超链接边框的方法是为图像标签添加样式 `style="border:none"`，代码如下：

```
<a href="waitao.html"> <! --单击图像则打开 waitao.html-->

</a>
```

2.3.4 设置网页背景图像

在网页中可以利用图像作为背景，但是要注意不要让背景图像影响网页内容的显示，因为背景图像只是起到渲染网页的作用。此外，背景图片最好不要设置边框，这样有利于生成无缝背景。

背景属性将背景设置为图像。属性值为图片的 URL。如果图像尺寸小于浏览器窗口，那么图像将在整个浏览器窗口进行复制。格式：

```
<body background=" 背景图像路径">
```

设置网页背景图像应注意以下要点：

- (1) 背景图像是否增加了页面的加载时间，背景图像文件大小不应超过 10 KB。
- (2) 背景图像是否与页面中的其他图像搭配良好。
- (3) 背景图像是否与页面中的文字颜色搭配良好。

例如，设置品牌 LOGO 图像作为网页的背景图像，其浏览效果如图 2-14 所示。

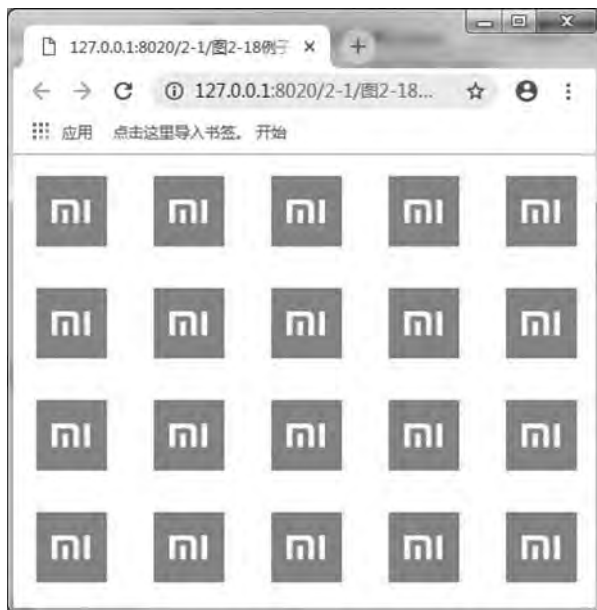


图 2-14 设置网页背景图像

代码如下：

```
<body background=" images/logo. JPG">
```

2.3.5 图文混排

图文混排技术是指设置图像与同一行中的文本、图像、插件或其他元素的对齐方式。在制作网页时往往要在网页中的某个位置插入一个图像，使文本环绕在图像的周围。img 标签的 align 属性用来指定图像与周围元素的对齐方式，实现图文混排效果。其取值见表 2-3。

表 2-3 align 的取值

| align 的取值 | 说明 | align 的取值 | 说明 |
|-----------|--------------|-----------|-----------------|
| left | 在水平方向上向上左对齐 | top | 图片顶部与同行其他元素顶部对齐 |
| center | 在水平方向上向上居中对齐 | middle | 图片中部与同行其他元素中部对齐 |
| right | 在水平方向上向上右对齐 | bottom | 图片底部与同行其他元素底部对齐 |

与其他元素不同的是，图像的 align 属性既包括水平对齐方式，又包括垂直对齐方式。align 属性的默认值为 bottom。

【例 2.12】使用图文混排技术制作全球汇商品描述的图片简介。本例浏览效果如图 2-15 所示。



图 2-15 商品描述的图片简介效果图

代码如下：

```
<! --示例 2.12 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>图文混排</title>
  </head>
  <body>
    <h1 align="center">商品描述</h1>
    <hr />
    
    产地:新西兰奶粉配方:常规配方<br/>
    包装:罐装适用人群:婴幼儿<br/>
    供货渠道:海外原装品牌名称:Aptamil<br/>
    商品名称:澳洲 Aptamil 金装婴幼儿配方奶粉 3 段 900 克保质期:18 个月<br/>
    成分:矿物质(1.2%),维生素(0.10%),奶粉(全脂奶粉、益生元、鱼油)(98.7%)<br/>
    生产日期:2018 年 4 月(具体以收到实物为准)<br/>
    规格:900 g/罐配件/备注:见详情<br/>
    商品编号:9418783002844<br/>
    储存方法:<br/>
    1. 请将奶粉存放在阴凉干燥处,开罐后密封保存,并请于开罐后四周内用完。<br/>
    2. 请在奶粉罐底部注明的日期前食用。<br/>
  </body>
</html>
```

在示例 2.12 中,如果不设置文本对图像的环绕,图像在页面中将占用一整片空白区域。利用

2.4 列表

在制作网页时,列表经常被用到写提纲和品种说明书。通过列表标记的使用能使这些内容在网页中条理清晰、层次分明、格式美观地表现出来。本节将重点介绍列表标签的使用。

列表的存在形式主要分为无序列表、有序列表、定义列表以及嵌套列表等。

2.4.1 无序列表

无序列表,就是列表中列表项的前导符号没有一定的次序,而是用黑点、圆圈和方框等一些特殊符号标识。无序列表并不是使列表项杂乱无章,而是使列表项的结构更清晰、更合理。

当创建一个无序列表时，主要使用 HTML 的标签和标签来标记。其中标签标识一个无序列表的开始；标签标识一个无序列表项。格式：

```
<ul type="符号类型">
  <li type="符号类型 1">第一个列表项 </li>
  <li type="符号类型 2">第二个列表项 </li>
  .....
</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；表项向右缩进并左对齐，每行前面有项目符号。

标签的 type 属性用来定义一个无序列表的前导字符，如果省略了 type 属性，浏览器会默认显示为 disc 前导字符。type 取值可以为 disc（实心圆）、circle（空心圆）、square（方框）。设置 type 属性的方法有两种。

1. 在后指定符号的样式

在后指定符号的样式，可设定直到的加重符号。例如

```
<ul type="disc">                                符号为实心圆点●
<ul type="circle">                              符号为空心圆点○
<ul type="square">                              符号为方块■
<ul img src="graph.gif">                       符号为指定的图片文件
```

2. 在后指定符号的样式

在后指定符号的样式，可以设置从该起直到的项目符号。格式：

```
<li type="disc">                                符号为实心圆点●
<li type="circle">                              符号为空心圆点○
<li type="square">                              符号为方块■
<li img src="graph.gif">                       符号为指定的图片文件
```

【例 2.13】学校二级学院机构的无序列表浏览效果如图 2-16 所示。



图 2-16 学校二级学院机构的无序列表的效果图

代码如下：

```
<! --示例 2.13 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>无序列表</title>
  </head>
  <body>
    <h2 align="center">学校二级学院机构</h2>
    <ul type="disc"></ul>      <! --列表样式为实心圆点-->
    <li>信息与计算机学院</li>
    <li>管理学院</li>
    <li>经济贸易学院</li>
    <li>建设工程学院</li>
    <li>智能学院</li>
    <li>马克思主义学院</li>
  </body>
</html>
```

以上示例中，在后指定符号的样式为 type="disc"，因此每个列表显示为实心圆点。

2.4.2 有序列表

有序列表是一个有特定顺序的列表项的集合。在有序列表中，各个列表项有先后顺序之分，它们之间以编号来标记。使用标签可以建立有序列表，表项的标签仍为。格式：

```
<ol type="符号类型">
  <li type="符号类型 1">表项 1
  <li type="符号类型 2">表项 2
  .....
</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；各表项前带顺序号。

有序的符号标识包括阿拉伯数字、小写英文字母、大写英文字母、小写罗马字母和大写罗马字母。标签的 type 属性用来定义一个有序列表的符号样式，在后指定符号的样式，可设定直到的表项加重记号。格式：

| | |
|--------------|-----------|
| <oltype="1"> | 序号为数字 |
| <oltype="A"> | 序号为大写英文字母 |
| <oltype="a"> | 序号为小写英文字母 |
| <oltype="I"> | 序号为大写罗马字母 |
| <oltype="i"> | 序号为小写罗马字母 |

在后指定符号的样式，可设定该表项前的加重记号。在格式上只需把上面的 ol 改

为 li。

【例 2.14】制作退货流程的有序列表，本例浏览效果如图 2-17 所示。

代码如下：

```
<! --示例 2.14 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>有序列表</title>
  </head>
  <body>
    <h2 align="center">退货流程</h2>
    <ol type="i">      <! --列表样式为小写罗马字母-->
      <li>进入订单管理,点击"退货"按钮;</li>
      <li>填写退货原因,提交退货申请;</li>
      <li>审核通过,全球汇,短信发送退货地址;</li>
      <li>自行选择快递,邮寄商品到全球汇;</li>
      <li>商品返仓后,全球汇退款并补贴 10 元。</li>
    </ol>
  </body>
</html>
```

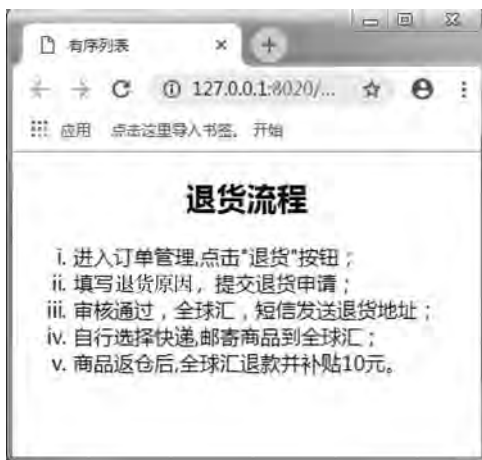


图 2-17 退货流程的有序列表效果图

2.4.3 定义列表

定义列表又称为释义列表或字典列表，定义列表不是带有前导字符的列项目，而是一列实物以及与其相关的解释。当创建一个定义列表时，主要用到 3 个 HTML 标签：<dl>标签、<dt>和<dd>标签。可以使用 dl 创建自定义列表，使用 dt 和 dd 定义列表中具体的数据项。一般情况下使用 dt 定义列表的二级列表项，也可以认为是 dd 的一个概要信息，使用 dd 来

定义最底层的列表项。格式：

```
<dl>
  <dt>…… 第一个标题项……</dt>
  <dd>…… 对第一个标题项的解释文字……</dd>
  <dt>…… 第二个标题项……</dt>
  ……
  <dd>…… 对第二个标题项的解释文字……</dd>
</dl>
```

默认情况下，浏览器一般会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。如果<dd>标签中内容很多，可以嵌套<p>标签使用。

【例 2.15】使用定义列表显示联系方式浏览效果如图 2-18 所示。



图 2-18 使用定义列表显示联系方式浏览效果图

代码如下：

```
<! --示例 2.15 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>定义列表</title>
  </head>
  <body>
    <h2 align="center">联系我们</h2>
    <dl>
      <dt>E-mail :</dt>
      <dd>worldall@ 163. com</dd>
```

```

        <dt>传真:</dt>
        <dd>0750-3118888</dd>
        <dt>地址:</dt>
        <dd>广东省江门市江海区五邑路</dd>
    </dl>
</body>
</html>

```

在示例 2.15 中, <dl>列表中每一项的名称不再是标签, 而是用<dt>标签进行标记, 后面跟由<dd>标签标记的条目定义或解释。

2.4.4 嵌套列表

所谓嵌套列表, 就是无序列表与有序列表出嵌套混合使用。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, 而且彼此可互相嵌套。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、在有序列表中嵌套无序列表等方式, 读者需要灵活掌握。

【例 2.16】制作全球汇商城支付向导页面, 在无序列表中嵌套无序列表、有序列表和定义列表, 本例浏览效果如图 2-19 所示。



图 2-19 嵌套列表网页效果图

代码如下:

```

<! --示例 2.16 程序-->
<! DOCTYPE html>
<html>

```



```
<head>
  <meta charset="UTF-8" >
  <title>嵌套列表</title>
</head>
<body>
  <h2>全球汇支付向导</h2>
  <ul type="circle">
    <li>全球汇支付方式</li>
    <ul type="disc">
      <li>货到付款</li>
      <li>微信支付</li>
      <li>支付宝</li>
      <li>网银在线</li>
    </ul>
    <hr />
    <li>网银在线支付步骤:</li>
    <ol>
      <li>选择您要使用的网上银行;</li>
      <li>显示您的应付总价,单击"确认无误,付款";</li>
      <li>确定您在银行的预留信息,单击"确定"按钮;</li>
      <li>输入您的网银账号,登录密码,验证码;</li>
      <li>支付成功,提示"已完成付款"。</li>
    </ol>
    <hr />
    <li>联系我们</li>
    <dl>
      <dt>E-mail:</dt>
      <dd>worldall@163.com</dd>
      <dt>传真:</dt>
      <dd>0750-3118888</dd>
      <dt>地址:</dt>
      <dd>广东省江门市江海区五邑路</dd>
    </dl>
  </ul>
</body>
</html>
```

2.5 综合案例——无线吸尘器说明书

本章综合前面学习的网页制作排版知识,可以利用文字与段落排版、超链接、图像和列表设置。网页设计要求如图 2-20 所示,效果图如图 2-21、图 2-22 所示。

Dyson V8 absolute 无线吸尘器

商品参数

吸尘器类别：手持式吸尘器
 功率：500 W
 储尘类型：旋风尘盒/尘桶
 吸尘类型：干/湿吸均可
 适用面积：91 m²-150 m²
 产品特色：除螨,HEPA滤网
 线长：充电式（无线）
 吸嘴类型：圆毛刷
 适用环境：厨房,地板,地毯,床铺,汽车,沙发,瓷砖,酒店
 售后服务：海外进口商品不支持国内联保，无法享受与产地同等售后服务
 品牌名称：Dyson
 商品名称：我家那闺女Dyson V8 absolute 无线吸尘器
 产地：马来西亚
 商品编号：V8absoluteUS

商品展示



图 2-20 网页设计要求

以下为完整代码，以供参考。

```

<! --综合案例.html-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>无线吸尘器说明书</title>
  </head>
  <body>
    <h1>Dyson V8 absolute 无线吸尘器</h1>
    <hr color="deeppink" />
    <ul style="font-size:25px">
      <li type="square" style="color:deeppink">商品参数</li>
      <p style="font-size:25px">吸尘器类别:手持式吸尘器<br />功率:500 W<br />
      储尘类型:旋风尘盒/尘桶<br />吸尘类型:干/湿吸均可<br />
    
```

```

适用面积:91 m2-150 m2<br />产品特点:除螨,HEPA 滤网<br />
线长:充电式(无线)<br />吸嘴类型:圆毛刷<br />
适用环境:厨房,地板,地毯,床铺,汽车,沙发,瓷砖,酒店<br />售后服务:海外进口商品不支持
国内联保,无法享受与产地同等售后服务<br />
品牌名称:Dyson<br />商品名称:我家那闺女 Dyson V8 absolute 无线吸尘器<br />
产地:马来西亚<br />商品编号:V8absoluteUS<br /></p>
<li type="square" style="color:deeppink">商品展示</li></ul>

<h2>产品信息</h2>
<p style="font-size:25px;">品牌:Dyson 戴森<br />
品名:无线手持吸尘器美版<br />
型号:V8 Absolute<br />
吸力:高达 115 空气瓦特<br />
使用时间:普通模式约 40 分钟、MAX 模式约 7 分钟<br />
(电动驱动吸头在非 MAX 模式下运行约 25 分钟)<br />
充电时间:约 5 小时<br />
产品尺寸:长 124.4 厘米,宽 18 厘米,高 22.4 厘米<br />
重量:2.61 公斤<br />
容量:0.54 升<br />
吸头配件:软绒滚筒吸头、升级版直驱吸头、缝隙吸头、<br />
电动床褥吸头、二合一吸头、软除尘毛吊<br />
产品特点:电力升级、新形式集尘桶、双层放射式气旋、HEPA<br />
过滤系统、噪声分贝更小<br />
</p>
<h2>产品特点</h2>

<h2>戴森科技,英国品质</h2>
<p style="font-size: 25px">戴森 V8 美版无绳吸尘器高配版,新的动力设计。<br />
有助吸除家中四处的过敏原,吸附 99.97% 小至 0.3 微米的微尘,<br />
运行时间高达 40 分钟,2 款地板吸头及 4 款配件吸头,清洁空间高达 291 平米。</p>
<br /><br /><h1>
HEPA 过滤系统<br />
吸附微尘 99.97% 小至 0.3 微米<br />
米的微尘,减少过敏原回流空至<br />
气中。<br /></h1>
<br />
<br />

<blockquote style="font-size: 25px">官网地址:<a href="https://shop.dyson.cn/" >
https://shop.dyson.cn/</a></blockquote>
</h1>
</body>
</html>

```



产品信息

品牌: Dyson戴森
 品名:无线手持吸尘器美版
 型号:V8 Absolute
 吸力:高达115空气瓦特
 使用时间:普通模式约40分钟、MAX模式约7分钟
 (电动启动吸头在非MAX模式下运行约25分钟)
 充电时间:约5小时
 产品尺寸:长124.4厘米,宽18厘米,高22.4厘米
 重量:2.61公斤
 容量:0.54升
 吸头配件:软绒滚筒吸头、升级版直驱吸头、缝隙吸头、
 电动床褥吸头、二合一吸头、软除尘毛吊
 产品特点:电力升级、新形式集尘桶、双层放射式气旋、HEPA
 过滤系统、噪声分贝更小

产品特点



戴森科技, 英国品质

戴森V8美版无线吸尘器高配版,新的动力设计。
 有助吸除家中四处的过敏原,吸附99.97%小至0.3微米的微尘。

图 2-21 产品说明书效果图 (1)

运行时间高达40分钟,2款地板吸头及4款配件吸头,清洁空间高达291平米。



HEPA过滤系统
 吸附微尘99.97%小至0.3微米的微尘,减少过敏原回流至空气中。



声学设计, 降低噪音

戴森工程师将重要的声学改进应用在新一代的戴森无线技术中。但不会对吸力作出任何降低的妥协妥协。

官网地址: <https://shop.dyson.cn/>

图 2-22 产品说明书效果图 (2)

本章习题

1. 用嵌套的列表制作“心语花店”导航栏，图 2-23 供参考。



图 2-23 题 1 网页效果图

2. 使用图文并排方式制作“公司名片”，图 2-24 供参考。

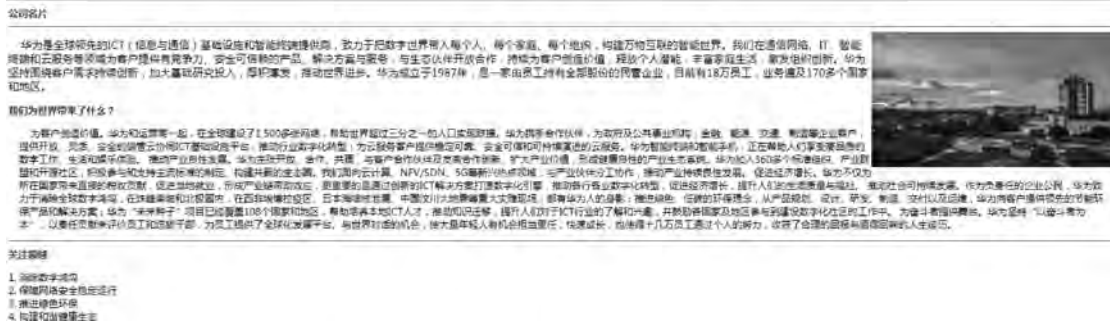


图 2-24 题 2 网页效果图

第3章 表格布局与表单交互

学习目标

1. 掌握表格的标记和标记属性；
2. 掌握表格单元格的各属性设置；
3. 掌握使用表格实现页面布局、灵活运用嵌套表格；
4. 掌握表单各元素的灵活运用。

知识要点

1. 表格的结构和使用；
2. 表单的工作原理；
3. 表单标签和表单元素。

网页的布局是指对网页元素的位置进行合理的安排。一个具有好的布局的网页，往往给浏览者带来赏心悦目的感受，而表格是网页布局中一种最简单的技术；表单是网站管理者与访问者之间进行信息交流的桥梁，利用表单可以收集用户意见，做出科学决策。

3.1 表格概述

表格是网页中的一个重要容器元素，可包含文字和图像。表格使网页结构紧凑整齐，使网页内容的显示一目了然。表格除了用来显示数据外，还用于搭建网页的结构。几乎所有HTML页面都或多或少地采用了表格。表格以简洁明了和高效快捷的方式将图片、文本、数据和表单的元素有序地显示在页面上，从而可以呈现出漂亮的页面。表格在网页设计中能将网页分成多个任意的矩形区域。

3.1.1 表格的结构

表格是由行和列组成的二维表，而每行又由一个或多个单元格组成，用于放置数据或其他内容。表格中的单元格是行与列的交叉部分，是组成表格的最基本单元。单元格的内容是数据，也称数据单元格。数据单元格可以包含文本、图片、列表、段落、表单、水平线或表格等元素。表格中的内容按照相应的行或列进行分类和显示，如图3-1所示。

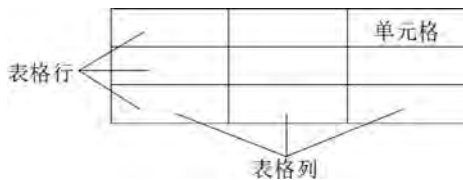


图 3-1 表格的基本结构

3.1.2 表格的基本语法

在 HTML 中，常用表格主要通过 5 个标记来构成：`<table>`，`<caption>`，`<th>`，`<tr>`和`<td>`，各标记说明如表 3-1 所示。

表 3-1 常用表格标记形式及说明表

| 标记形式 | 说 明 |
|--|--------|
| <code><table></table></code> | 表格标记 |
| <code><caption></caption></code> | 表格标题标记 |
| <code><th></th></code> | 表格表头标记 |
| <code><tr></tr></code> | 表格的行标记 |
| <code><td></td></code> | 表格的列标记 |

语法：

```

<table>
  <caption>表格标题</caption>
  <tr>
    <th></th>
    <th></th>
    <th></th>
  </tr>
  <tr>
    <td></td>
    <td></td>
    <td></td>
  </tr>
  .....
</table>

```

- (1) table 标记是成对标记，`< table>`表示表格开始，`< /table>`表示表格结束。
- (2) caption 标记是成对标记，`<caption>`表示标题开始，`</caption>`表示标题结束。使用 caption 标记可以给表格添加标题，该标题应位于 table 标记与 tr 标记之间的位置。
- (3) tr (Table Row) 标记是成对标记，`<tr>`表示行开始，`</tr>`表示行结束。
- (4) th (Table Heading 表头) 标记是成对标记，`<th>`表示表头开始，`</th>`表示表头结



束，设计表格时，表头常常作为表格的第 1 行或者第 1 列，用来对表格单元格的内行说明。表头文字内容一般居中、加粗显示。

(5) td (Table Data) 标记是成对标记，定义单元格或列。以<td>开始，以</td>结束。表头可以用 th 标记定义，也可以用 td 标记定义，但<td></td>两标记之间的内容不自动居中、不加粗。

在一个表格中，可以插入多个 tr 标记，表示多行，一组<tr></tr>标记表示插入一行，一行中可以有多个列，列中的内容可以是文字、数据、图像、超链接、表单元素等。

【例 3.1】设计班级课程表。

```
<! --示例 3.1 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>定义表格</title>
  </head>
  <body>
    <table width="700" height="150" border="1">
      <caption><h2>18 软件技术 7 班课程表</h2></caption>
      <tr>
        <th>节次</th>
        <th>星期一</th>
        <th>星期二</th>
        <th>星期三</th>
        <th>星期四</th>
        <th>星期五</th>
      </tr>
      <tr>
        <td>第 1-2 节</td>
        <td>数据库</td>
        <td>C 语言</td>
        <td>网页设计</td>
        <td>java</td>
        <td>php</td>
      </tr>
      <tr>
        <td>第 3-4 节</td>
        <td>php</td>
        <td>计算机数学</td>
        <td></td>
        <td></td>
        <td>C 语言</td>
      </tr>
    </table>
  </body>
</html>
```

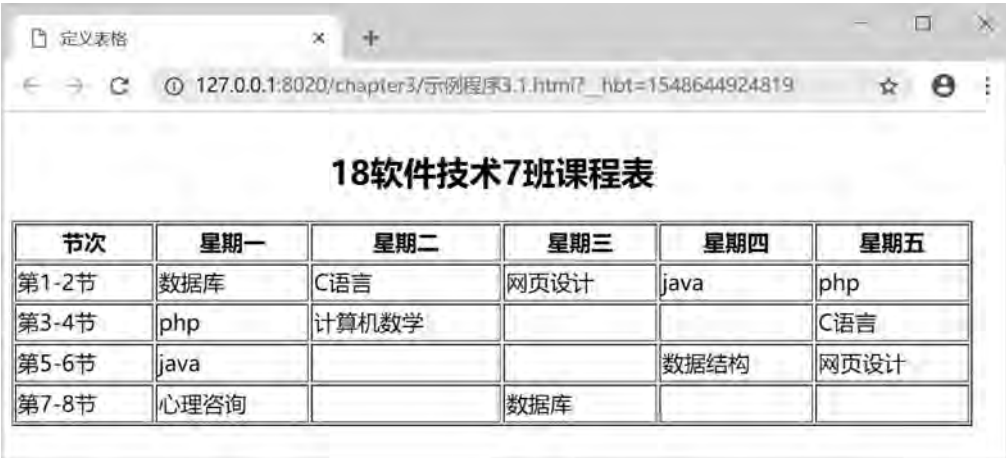


```

<tr>
  <td>第 5-6 节</td>
  <td>java</td>
  <td></td>
  <td></td>
  <td>数据结构</td>
  <td>网页设计</td>
</tr>
<tr>
  <td>第 7-8 节</td>
  <td>心理咨询</td>
  <td></td>
  <td>数据库</td>
  <td></td>
  <td></td>
</tr>
</table>
</body>
</html>

```

例 3.1 浏览效果如图 3-2 所示。



| 节次 | 星期一 | 星期二 | 星期三 | 星期四 | 星期五 |
|-------|------|-------|------|------|------|
| 第1-2节 | 数据库 | C语言 | 网页设计 | java | php |
| 第3-4节 | php | 计算机数学 | | | C语言 |
| 第5-6节 | java | | | 数据结构 | 网页设计 |
| 第7-8节 | 心理咨询 | | 数据库 | | |

图 3-2 班级课程表

3.2 表格属性设置

表格是网页文件中布局的重要元素，制作网页的过程中常常需要对网页中的表格做一些设置，对表格的设置实质是对表格标记属性的一些设置。表格标记的属性如表 3-2 所示。

表 3-2 表格标记的属性、取值及说明表

| 属 性 | 取 值 | 说 明 |
|-------------|---|-----------------|
| align | left center right | 规定表格相对周围元素的对齐方式 |
| bgcolor | #rrggbb colname b (r%, g%, b%) rgb (rr, gg, bb) | 规定表格的背景颜色 |
| border | pixels | 规定表格边框的宽度 |
| cellpadding | pixels % | 规定单元边沿与其内容之间的空白 |
| cellspacing | pixels % | 规定单元格之间的空白 |
| frame | above below hside vside lhs rhs border void | 规定外侧边框的哪个部分是可见的 |
| rules | none all rows cols groups | 规定内侧边框的哪个部分是可见的 |
| height | % pixels | 规定表格的高度 |
| width | % pixels | 规定表格的宽度 |

3.2.1 表格边框属性

设置表格标记中的边框属性可以改变表格的外观。表格边框属性如表 3-3 所示。表格中的属性同样适用于单元格。

表 3-3 表格边框属性说明表

| 边框属性 | 说 明 | 边框属性 | 说 明 |
|-------------|----------|------------------|-----------|
| border | 表示表格边框粗细 | bordercolorlight | 表示表格亮边框颜色 |
| bordercolor | 表示表格边框颜色 | bordercolordark | 表示表格暗边框颜色 |

语法：

```
<table border="" bordercolor="" bordercolorlight="" bordercolordark="">...</table>
```

- (1) border 属性。用于设置边框的粗细，单位是像素。
- (2) bordercolor 属性。用于设置表格边框的颜色，可以使用 rgb 函数、十六进制数和颜色英文名称。
- (3) bordercolorlight 属性。用于设置表格亮边框，对表格左上边框生效。
- (4) bordercolordark 属性。用于设置表格暗边框，对表格右下边框生效。

3.2.2 表格的宽度和高度属性

通过设置 width 属性和 height，属性可以设置表格的宽度和高度。

语法：

```
<table width="" height="">...</table>
```

- (1) width。其单位可以是长度单位或百分比，用于定义表格的宽度。
- (2) height。其单位可以是长度单位或百分比，用于定义表格的高度。

设置表格标记中的高度与宽度为百分比时，表格会跟随浏览器窗口的改变而自动调整。

3.2.3 表格背景颜色与背景图像属性

设置表格的 `bgcolor` 属性可以改变表格的背景颜色，设置表格的 `background` 属性可以为表格增添背景图像效果，使表格更加美观。

语法：

```
<table bgcolor="" background="">...</table>
```

- (1) `bgcolor`。可以用 `rgb` 函数、十六进制、英文颜色名称来设置背景颜色。
- (2) `background`。设置背景图像，图像的路径可以是绝对路径或相对路径。
- (3) 同时设置背景颜色和背景图像属性时，背景图像会部分或完全覆盖背景颜色。

【例 3.2】 设置表格边框属性。

```
<! -- 示例 3.2 程序 -->
```

```
<! DOCTYPE html>
```

```
<html>
```

```
  <head>
```

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

```
    <title>设置表格边框、背景</title>
```

```
  </head>
```

```
  <body>
```

```
    <h4 align="center">设置表格边框、背景</h4>
```

```
    <table align="center" width="700px" height="150px"
```

```
      border="15" bordercolor="black" bordercolorlight="#ff0000" bordercolordark="#6600ff" bgcolor="#99cccc" >
```

```
      <tr>
```

```
        <th>学号</th>
```

```
        <th>姓名</th>
```

```
        <th>所在院系</th>
```

```
      </tr>
```

```
      <tr>
```

```
        <td>1616010218</td>
```

```
        <td>李明</td>
```

```
        <td>信息学院</td>
```

```
      </tr>
```

```
      <tr>
```

```
        <td>1416020109</td>
```

```
        <td>王丹</td>
```

```
        <td>管理学院</td>
```

```
      </tr>
```

```
    </table>
```

```
    <hr>
```

```
    <table align="center" border="15px" bordercolor="grey" width="700px" height="150px" background="img/bj01.jpg" bgcolor="#99cccc" >
```

```

<tr>
  <th>学号</th>
  <th>姓名</th>
  <th>所在院系</th>
</tr>
<tr>
  <td>1616010218</td>
  <td>李明</td>
  <td>信息学院</td>
</tr>
<tr>
  <td>1416020109</td>
  <td>王丹</td>
  <td>管理学院</td>
</tr>
</table>
</body>
</html>

```

例 3.2 浏览效果如图 3-3 所示。



图 3-3 设置表格的边框和背景属性

例 3.2 中第一个表格中定义了边框粗细、边框颜色、亮边框颜色、暗边框颜色和背景颜色；但是亮边框颜色和暗边框颜色对浏览器要求严格，请自行测试，第二个表格同时设置了表格的背景颜色与背景图像，但背景图像覆盖了背景颜色。

3.2.4 表格边框样式属性

设置表格标记中的 `frame` 属性可以改变表格边框的样式；设置表格标记中的 `rules` 属性可

以改变表格内部边框的样式。

语法:

```
<table frame="" rules="">……内容……</table>
```

说明: frame 和 rules 属性值及说明如表 3-4 所示。

表 3-4 frame 和 rules 常见属性值及说明

| frame 属性值 | 说 明 | rules 属性值 | 说 明 |
|-----------|--------|-----------|-----------|
| above | 显示上边框 | all | 显示所有内部边框 |
| below | 显示下边框 | none | 不显示内部边框 |
| hsides | 显示上下边框 | rows | 仅显示行边框 |
| vsides | 显示左右边框 | cols | 仅显示列边框 |
| lhs | 显示左边框 | groups | 显示介于行列间边框 |
| rhs | 显示右边框 | border | 显示上下、左右边框 |
| void | 不显示边框 | | |

【例 3.3】 设置表格的边框样式属性。

```
<! --示例 3.3 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>>设置边框样式</title>
  </head>
  <body>
    <table align="center" border="2" bordercolor="blue" width="400px" height="120px" frame="
hsides" rules="all">
      <caption><b>表格边框样式定义</b></caption>
      <tr>
        <th>姓名 </th>
        <th>院系名称 </th>
        <th>学号</th>
      </tr>
      <tr>
        <td>李明</td>
        <td>信息学院</td>
        <td>1616010218</td>
      </tr>
      <tr>
        <td>王丹</td>
        <td>管理学院</td>
        <td>1416020109</td>
```

```

</tr>
<tr>
  <td>张小林 </td>
  <td>工程学院 </td>
  <td>1716030115</td>
</tr>
</table>
</body>
</html>

```

例 3.3 浏览效果如图 3-4 所示。



图 3-4 设置表格边框样式

例 3.3 中设置 `fame` 属性值为 `hsides`，只显示表格上下边框，不显示左右边框。设置 `rules` 属性值为 `all`，显示表格内部的所有边框。

3.2.5 表格单元格间距、单元格边距属性

设置表格标记中的 `cellspacing` 属性可以改变表格单元格之间的间隔，使网页中的表格内容稍微松散一些。设置表格标记中的 `cellpadding` 属性可以增加表格单元格的内容与内部边框之间的距离。

语法：

```

<table cellspacing="" cellpadding=""
</table>

```

(1) `cellspacing`。值的单位为像素或百分比，默认值为 `2px`。

(2) `cellpadding`。值的单位为像素或百分比。

【例 3.4】 设置表格的单元格间距和边距。

```

<! --示例 3.4 程序-->
<! DOCTYPE html>
<html>
  <head>

```

```
<meta charset="UTF-8" >
<title>设置单元格间距和边距</title>
<style type="text/css">
strong {
    background:#ceffcc;
}
td {
    background:#99ccff;
}
</style>
</head>
<body>
<b>设置单元格间距和边距</b>
<table width="380" border="4" cellspacing="50px" cellpadding="20px" bgcolor="#9966ff" >
<tr>
    <td><strong>网页设计</strong></td>
    <td><strong>大学体育</strong></td>
</tr>
</table>
</body>
</html>
```

例 3.4 浏览效果如图 3-5 所示。



图 3-5 设置单元格间距和边距

例 3.4 中定义了 strong 标记样式，作用是设置单元内容的背景颜色；td 样式定义单元格的背景颜色；设置了单元格的间距为 50px、单元格边距为 20px。

3.2.6 表格水平对齐属性

通过设置表格标记的 align 属性，可以设定表格在水平方向上的对齐方式。对齐方式有居左、居中、居右 3 种。

语法：

```
<table align="left | center | right">...</table>
```



说明: align 属性的取值可以为 left (默认居左)、center (居中) 和 right (居右)。

【例 3.5】 设置表格的水平对齐属性。

```
<! --示例 3.5 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>设置表格水平对齐方式</title>
    <style type="text/css">
      div {
        width:100%;
        height:100px;
      }
    </style>
  </head>
  <body>
    <div>
      <table width="400" align="left" border="2">
        <caption>学生信息表(左对齐)</caption>
        <tr>
          <td>李明</td>
          <td>信息学院</td>
          <td>1616010218</td>
        </tr>
        <tr>
          <td>王丹</td>
          <td>管理学院</td>
          <td>1416020109</td>
        </tr>
      </table>
    </div>
    <div>
      <table width="400" align="center" border="2">
        <caption>学生信息表(居中对齐)</caption>
        <tr>
          <td>李明</td>
          <td>信息学院</td>
          <td>1616010218</td>
        </tr>
        <tr>
          <td>王丹</td>
          <td>管理学院</td>
          <td>1416020109</td>
        </tr>
      </table>
    </div>
  </body>
</html>
```



```

        </tr>
    </table>
</div>
</body>
</html>

```

例 3.5 浏览效果如图 3-6 所示。

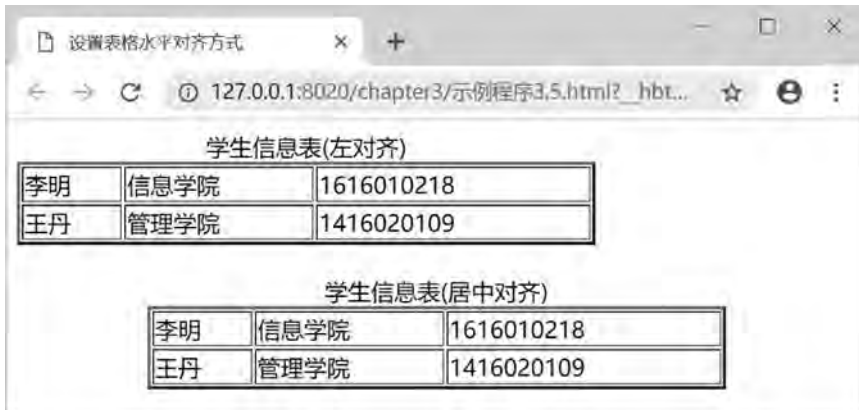


图 3-6 设置表格水平对齐方式

例 3.5 中通过两个图层 div 设置两种表格水平对齐方式。第一个表格设置了左对齐，第二个表格设置了居中对齐。

3.2.7 设置表格的 (tr) 标记行的属性

表格行 tr 标记的属性用于设置表格某一行的样式，其属性设置如表 3-5 所示。

表 3-5 tr 标记的属性表

| 属性值 | 说 明 | 属性 | 说 明 |
|---------|---------|------------------|---------|
| align | 行内容水平对齐 | bordercolor | 行的边框颜色 |
| valign | 行内容垂直对齐 | bordercolorlight | 行的亮边框颜色 |
| bgcolor | 行的背景颜色 | bordercolordark | 行的暗边框颜色 |

通过 tr 标记的 align 属性可以设置行内容的水平对齐方式。水平对齐方式有居左对齐、居中对齐和居右对齐。通过 tr 标记的 valign 属性可以设置行内容的垂直对齐方式。垂直对齐方式有顶部对齐、居中对齐和底部对齐。

语法：

```

<table align="center">
    <tr align="left | center | right" valign="top | middle | bottom">
        <td>...</td>
    </tr>
    .....

```



```
</table>
```

说明：left 表示设置行内容居左对齐；center 表示设置行内容居中对齐；right 表示设置行内容居右对齐。top 表示设置行内容顶部对齐；middle 表示设置行内容居中对齐；bottom 表示设置行内容底部对齐。其中行垂直居中对齐属性值与行水平居中对齐属性值不同。

【例 3.6】 设置表格行内容对齐属性。

```
<! -- 示例 3.6 程序 -->
```

```
<! DOCTYPE html>
```

```
<html>
```

```
  <head>
```

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

```
    <title>设置行内容对齐方式</title>
```

```
    <style type="text/css" >
```

```
      td{
```

```
        background:#ceffcc;
```

```
      }
```

```
    </style>
```

```
  </head>
```

```
  <body>
```

```
    <table border="1" width="450px" height="240px" align="center" bordercolor="#6600ff" >
```

```
      <caption><b>学生信息表(设置表行内容对齐方式)</b></caption>
```

```
      <tr>
```

```
        <th>姓名 </th>
```

```
        <th>院系名称 </th>
```

```
        <th>班级</th>
```

```
      </tr>
```

```
      <tr align="left" valign="top" >
```

```
        <td>李明</td>
```

```
        <td>信息学院</td>
```

```
        <td>1616010218</td>
```

```
      </tr>
```

```
      <tr align="center" valign="middle" >
```

```
        <td>王丹</td>
```

```
        <td>管理学院</td>
```

```
        <td>1416020109</td>
```

```
      <tr align="right" valign="bottom" >
```

```
        <td>张小林 </td>
```

```
        <td>工程学院 </td>
```

```
        <td>1716030115</td>
```

```
      </tr>
```

```
    </table>
```

```
  </body>
```

```
</html>
```

例 3.6 浏览效果如图 3-7 所示。

| | | |
|-----|------|------------|
| 姓名 | 院系名称 | 班级 |
| 李明 | 信息学院 | 1616010218 |
| 王丹 | 管理学院 | 1416020109 |
| 张小林 | 工程学院 | 1716030115 |

图 3-7 设置行内容对齐方式

例 3.6 中设置了表格中的所有单元格 td 标记样式的背景颜色；设置表格的第 2 行内容对齐方式为水平居左、垂直居顶；第 3 行内容对齐方式为水平、垂直均居中；第 4 行内容对齐方式为水平居右、垂直居底。

3.2.8 设置单元格的属性

表格的列 (td) 标记的属性可以设置表格单元格的显示风格。td 标记常用的属性如表 3-6 所示。

表 3-6 td 标记的属性表

| 属性值 | 说 明 | 属性值 | 说 明 |
|-------------|-----------|------------------|-----------|
| align | 单元格内容水平对齐 | bordercolorlight | 单元格的亮边框颜色 |
| valign | 单元格内容垂直对齐 | bordercolordark | 单元格的暗边框颜色 |
| bgcolor | 单元格的背景颜色 | rowspan | 单元格跨行 |
| background | 单元格背景图像 | colspan | 单元格跨列 |
| bordercolor | 单元格的边框颜色 | width | 单元格宽度 |
| height | 单元格高度 | | |

3.2.9 表格单元格跨行、跨列属性

1. 单元格跨行

使用单元格 td 标记的 rowspan 属性可以设置单元格跨行合并。

语法:

```
<td rowspan="行数">...</td>
```

说明: rowspan 属性可以设置单元格跨行。通过 rowspan="n" (n 是正整数), 可以设置某一单元格跨 n 行, 当前行下的 n-1 行内的单元格数量都需要减少一个, 即少定义一个 td 标记。

2. 单元格跨列

使用单元格 td 标记的 colspan 属性可以设置单元格跨列合并。

语法:

```
<td colspan="列数">...</td>
```

说明: colspan 属性可以设置单元格跨列。通过 colspan="n" (n 是正整数), 可以设置某一单元格跨 n 列, 当前行内的单元格数量需要减少 n-1 个, 即删除 n-1 个 td 标记。

【例 3.7】 设置表格单元格合并。

```
<! --示例 3.7 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>设置单元格跨列、跨行属性</title>
  </head>
  <body>
    <h3 align="center">设置单元格跨列、跨行属性</h3>
    <table border="1" width="500px" align="center" bordercolor="#3366ff">
      <caption>专业研讨会日程安排</caption>
      <tr align="center">
        <td colspan="2">上午</td>
        <td colspan="2">下午</td>
      </tr>
      <tr >
        <td>8:00-10:00</td>
        <td>10:10-12:00 </td>
        <td>14:00-16:00</td>
        <td>16:10-18:00</td>
      </tr>
      <tr align="center">
        <td rowspan="2">学校领导讲话 </td>
        <td>大会主题报告</td>
        <td>行业企业专题报告</td>
        <td rowspan="2">总结报告</td>
      </tr>
      <tr align="center">
        <td>专家报告</td>
        <td>分组讨论</td>
      </tr>
    </table>
  </body>
</html>
```

```

</tr>
  <tr align="center">
    <td colspan="4">全天参观人工智能实训中心</td>
  </tr>
</table>
</body>
</html>

```

例 3.7 浏览效果如图 3-8 所示。

The screenshot shows a web browser window with the title '设置单元格跨列、跨行属性'. The address bar shows '127.0.0.1:8020/chapter3/示例程序3.7...'. The main content area displays a table titled '专业研讨会日程安排'. The table has a header row with '上午' and '下午'. The first row of data shows time slots: '8:00-10:00', '10:10-12:00', '14:00-16:00', and '16:10-18:00'. The second row shows activities: '学校领导讲话', '大会主题报告', '行业企业专题报告', and '总结报告'. The third row shows '专家报告' and '分组讨论'. The final row is a single cell with '全天参观人工智能实训中心'.

| 上午 | | 下午 | |
|--------------|-------------|-------------|-------------|
| 8:00-10:00 | 10:10-12:00 | 14:00-16:00 | 16:10-18:00 |
| 学校领导讲话 | 大会主题报告 | 行业企业专题报告 | 总结报告 |
| | 专家报告 | 分组讨论 | |
| 全天参观人工智能实训中心 | | | |

图 3-8 设置单元格跨行、跨列属性

3.3 表格嵌套

表格嵌套是一种常用的页面布局方式。利用表格嵌套可以设计比较复杂且美观的页面效果。通常情况下，使用表格嵌套时，表格不宜过多使用，否则会降低网站访问速度。表格嵌套一般采用在单元格内嵌套表格。

语法：

```

<table>
  <tr>
    .....
  </tr>
  <tr>
    <td>      <!--单元格内嵌表格-->
      <table>
        <tr>
          .....
        </tr>
      </table>
    </td>
  </tr>
</table>

```

```
</td>
</tr>
<tr>
.....
</tr>
</table>
```

【例 3.8】 利用嵌套表格布局页面。

```
<! --示例 3.8 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" >
    <title>嵌套表格布局页面</title>
    <style>
    body { font-size:36px; }
    </style>
  </head>
  <body>
    <h4 align="center">嵌套表格布局页面</h4>
    <table width="660px" border="1" align="center" bordercolor="#3333ff">
  <tr>
    <td height="100">
      <table width="100%" border="1" bordercolor="red">
        <tr height="50" align="center">
          <td rowspan="2" width="100">logo</td>
          <td>广告条</td>
        </tr>
        <tr height="50" align="center">
          <td>导航</td>
        </tr>
      </table>
    </td>
  </tr>
  <tr>
    <td height="300">
      <table width="100%" border="1" bordercolor="#33ff99">
        <tr align="center">
          <td height="300" width="30%">左栏目</td>
          <td height="300" width="70%">正文内容</td>
        </tr>
```

```
</table>
</td>
</tr>
  <tr align="center">
    <td height="100">版权信息</td>
  </tr>
</table>
</body>
</html>
```

例 3.8 浏览效果如图 3-9 所示。



图 3-9 嵌套表格布局页面

3.4 表 单

HTML 中的表单是网页中最常用的元素，是网站服务器端与客户端之间沟通的桥梁。一个完整的交互表单由两部分组成：一是客户端包含的表单页面，用于填写浏览者进行交互的信息；另一个是服务端的应用程序，用于处理浏览者提交的信息，浏览者在表单中输入信息，然后将这些信息提交给服务器；服务器中的应用程序会对这些信息进行处理响应，这样就完成了浏览者和服务器之间的交互。

3.4.1 表单标记

表单 form 标记为成对标记，以<form>开始和</form>结束。表单定义了采集数据的范围，其所包含的数据内容将被完整地提交给服务器。

语法：

```
<form method="post" action="" >
  <input type="text" name="" />
  <textarea name="" rows="" cols="" >.....</textarea>
  <select name="" >
    <option value="" selected></option>
    <option value="" ></option>
  </select>
</form>
```

说明：<form>和</form>之间可包含各种表单信息输入标记。表单标记的属性主要有 name, action, method 和 enctype 等，其属性、取值及说明如表 3-7 所示。

表 3-7 表单标记属性、取值及说明

| 属 性 | 取 值 | 说 明 |
|---------|------------|--|
| name | name | 规定表单的名称 |
| action | url | 规定当提交表单时，向何处发送表单数据 |
| method | get post | 规定如何发送表单数据。post 方法主要包含名称/值对，并且无须包含于 action 属性的 URL 中。get 方法把名称/值对加在 action 的 URL 后面并且把新的 URL 送至服务器，不推荐使用 |
| enctype | MIME_ type | 规定表单数据在发送到服务器之前应该如何编码 |

【例 3.9】 表单的应用。

```
<! --示例 3.9 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>表单的使用示例</title>
  </head>
  <body>
    <form name="form1" method="post" action="form_action.jsp" enctype="text/plain">
      <h3>输入课程成绩</h3>
      姓名:<input type="text" /><br/>
      网页设计:<input type="text" size="15" />
      数据结构:<input type="text" size="15" /><br/><br/>
      <input type="submit" value="成绩提交" />
      <input type="reset" value="成绩重置" />
    </form>
  </body>
</html>
```

例 3.9 浏览效果如图 3-10 所示。



图 3-10 表单使用示例

例 3.9 中定义了一个表单，指定该表单的名称为 form1，提交方式为 post，处理程序为 form_action.jsp，编码方式为 text/plain，另外还定义了 3 个单行文本输入框，用于输入学生的姓名和课程成绩，定义一个提交按钮和一个重置按钮。

3.4.2 定义域和域标题

利用 fieldset 标记可以在网页上定义域，在表单中使用域可以将表单的相关元素进行分组，fieldset 标记将表单内容的一部分打包，生成一组相关表单的字段。当一组表单元素放到 fieldset 标记内时，浏览器会以特殊方式来显示它们，它们可能有特殊的边界、3D 效果，或者可创建一个子表单来处理这些元素。legend 标记为 fieldset 标记定义域标题。

语法：

```
<form>
  <fieldset>
    <legend align="left | center | right">域标题内容</legend>
  </fieldset>
</form>
```

说明：fieldset 标记没有属性，是成对标记。legend 标记必须位于 fieldset 标记内，也是成对标记；有一个对齐 align 属性，属性值分别为 left，center 和 right。

【例 3.10】 域和域标题标记的应用。

```
<!-- 示例 3.10 程序 -->
<! DOCTYPE html>
<html>
  <meta charset="UTF-8">
  <title>定义域和域标题示例</title>
</head>
<body>
```

```
<form>
  <fieldset>
    <legend align="center">基本信息</legend>
    姓名:<input name="name" type="text"/>
    性别:<input name="sex" type="text"/>
  </fieldset>
  <fieldset>
    <legend align="center">其他信息</legend>
    身高:<input name="height" type="text"/>
    体重:<input name="weight" type="text"/>
  </fieldset>
</form>
</body>
</html>
```

例 3.10 浏览效果如图 3-11 所示。

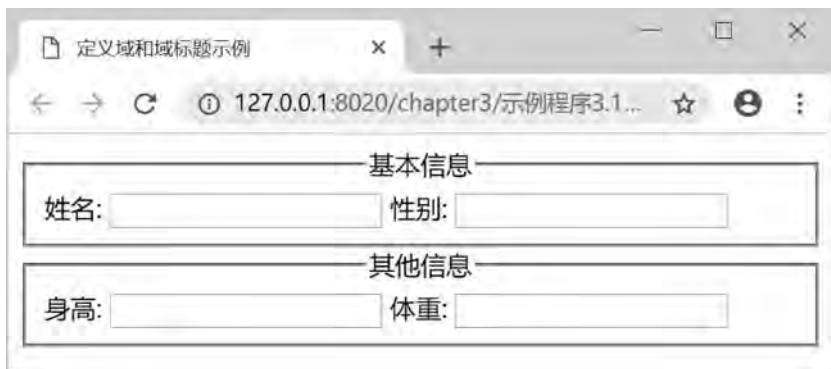


图 3-11 定义域和域标题示例

3.4.3 表单信息输入

表单的主要功能是为用户提供输入信息的接口，将输入信息发送到服务器并等待服务器响应。表单中输入信息的标记是 input 标记，可以输入一行信息。input 标记是单个标记。

语法：

```
<input name="" type="" />
```

说明：input 标记的属性主要有 name 和 type 等，输入类型是由类型 type 属性定义的，type 属性有很多不同的值，设置属性值不同，就会产生不同界面效果。input 标记的属性、取值及说明如表 3-8 所示。

表 3-8 表单信息输入标记属性、取值及说明

| 属 性 | 取 值 | 说 明 |
|------|---|--|
| name | name | 定义 input 元素的名称 |
| type | text password checkbox radio image submit reset button file hidden | 规定 input 元素的类型。text 为单行文本输入框，password 为密码输入框，checkbox 为复选框，radio 为单选按钮，image 为图像按钮，submit 为提交按钮，reset 为重置按钮，button 为普通按钮，file 为文件选择框，hidden 为隐藏框 |

1. 单行文本输入框

设置 input 标记的 type 属性值为 text，可以实现向表单中插入一个单行文本框。在单行文本框中可以输入任意类型的数据，但是输入的数据只能单行显示，不能换行。

语法：

```
<input name="" type="text" maxlength="" size="" value="" readonly/>
```

说明：单行文本输入框的主要属性有 name，maxlength，size，value 和 readonly，其属性及取值说明如表 3-9 所示

表 3-9 文本输入框标记属性、取值及说明

| 属 性 | 取 值 | 说 明 |
|-----------|----------------|-----------------------|
| name | name | 定义 input 元素的名称 |
| maxlength | number | 规定输入字段中的字符的最大长度 |
| size | number_of_char | 定义输入字段的宽度。其值小于或等于最大长度 |
| value | value | 规定 input 元素的默认值 |
| readonly | readonly | 规定文本框中内容只读，不能修改和编辑 |

2. 密码输入框

设置 input 标记的 type 属性值为 password，可以实现向表单中插入一个密码输入框。密码输入框中可以输入任意类型的数据，与单行文本输入框有所不同，这些数据不是实时地显示在页面上，而是被显示字符“·”所取代，这样设计可以保障用户输入的密码不被泄露。

语法：

```
<input name="" type="password" maxlength="" size="" />
```

说明：密码输入框的主要属性有 name，maxlenth 和 size，取值及说明与单行文本输入框相同。

【例 3.11】用户信息输入。

```
<! --示例 3.11 程序-->
```

```
<! DOCTYPE html>
```

```
<html>
```

```
  <head>
```


表 3-10 复选框属性、取值及说明

| 属 性 | 取 值 | 说 明 |
|---------|---------|----------------|
| name | name | 定义 input 标记的名称 |
| value | value | 规定 input 标记的值 |
| checked | checked | 预先选定复选框 |

由于复选择框可以支持多选，每一个复选框都是不同的，一组复选框的所有 name 属性应该不同，value 属性值也应该不同。

4. 单选按钮

设置 input 标记的 type 属性值为 radio，可以实现向表单中插入一个单选按钮，用户可利用单选按钮在网页上为某一选择设置多个单选项。

语法：

```
<input name="" type="radio" value="" checked/>
```

说明：单选按钮的属性有 name、value 和 checked 等，其属性、取值及说明与复选框相同。由于单选按钮必须是唯一的，在一组单选按钮中，只能选择一个单选按钮，所以一组单选按钮的所有 name 属性值必须相同，value 属性取值应该不同。

【例 3.12】 复选框与单选按钮的应用。

```
<head>
  <meta charset="UTF-8">
  <title>复选框与单选按钮的应用</title>
  <style type="text/css">
    fieldset {
      width:300px;
      height:120px;
      border:2px double #003399;
      padding-left:30px;
    }
  </style>
</head>
<body>
  <form>
    <fieldset>
      <legend>请填写个人信息</legend><br>
      姓名:<input type="text" name="name" maxlength="10" size="10"/><br>
      爱好:<input type="checkbox" name="n1" value="跳舞"/>跳舞
      <input type="checkbox" name="n2" value="踢足球" checked="checked"/>踢足球
      <input type="checkbox" name="n3" value="打游戏" checked="checked"/>打游戏<br>
      性别:<input type="radio" name="sex" value="male" checked="checked"/>男
      <input type="radio" name="sex" value="female"/>女
    </fieldset>
  </form>
</body>
```

```

        </fieldset>
    </form>
</body>
</html>

```

例 3.12 浏览效果如图 3-13 所示。

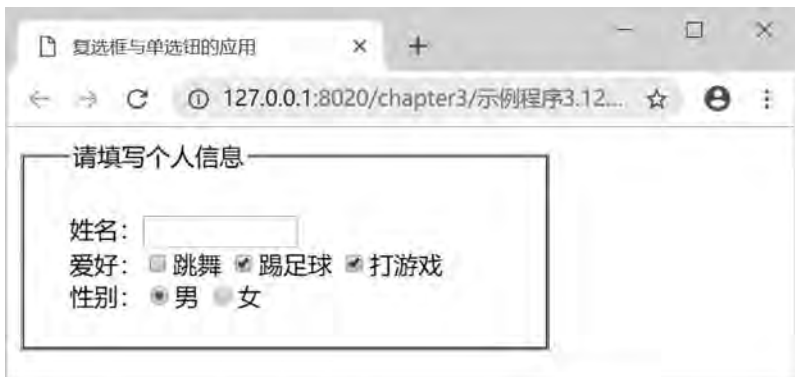


图 3-13 复选框与单选框的应用

在例 3.12 中插入了 3 个复选框，name 属性值分别为 n1，n2 和 n3，value 属性取值分别为“跳舞”“踢足球”和“打游戏”，并给 input 标记设置 checked 属性，将名称为 n2 和 n3 复选框设置为预选；另外还插入了两个单选按钮，name 属性值均为 sex，value 属性值分别 male 和 female，并给 input 标记设置 checked 属性，将“男性”单选按钮设置成预选项。

5. 图像按钮

设置 input 标记的 type 属性值为 image，可以实现向表单中插入一个图像按钮，用户可利用图像按钮在网页中插入一张图像，通过 src 属性加载图像。

语法：

```
<input name="" type="image" src="" width="" height="" />
```

说明：图像按钮主要属性有 name，src，width 和 height。其属性、取值及说明如表 3-11 所示。

表 3-11 图像按钮属性、取值及说明

| 属 性 | 取 值 | 说 明 |
|--------|--------|---------------------|
| name | name | 定义 input 标记的名称 |
| src | URL | 定义以提交按钮形式显示的图像的 URL |
| width | width | 规定图像的宽度，单位为像素 |
| height | height | 规定图像的高度，单位为像素 |

【例 3.13】 在网页中使用图像按钮。

```

<! --示例 3.13 程序-->
<! DOCTYPE html>
<html>

```

```
<head>
  <meta charset="UTF-8">
  <title>图像按钮应用</title>
  <style type="text/css">
  body {
    text-align:center;
  }
  input {
    width:150px;
    height:150px;
  }
</style>
</head>
<body>
  <form>
    <h3>Play Game</h3>
    <input type="image" name="start" src="img/start_game.png" align="center" />
    <input type="submit" value="提交" />
  </form>
</body>
</html>
```

例 3.13 浏览效果如图 3-14 所示。



图 3-14 图像按钮应用

6. 提交按钮

设置 input 标记的 type 属性值为 submit，可以实现向表单中插入一个提交按钮，提交按钮用于将表单的信息提交至服务器进行处理。

语法：

```
<input name="" type="submit" value="提交"/>
```

说明：提交按钮的属性主要有 name 和 value，其属性、取值及说明如表 3-12 所示。

表 3-12 提交按钮属性、取值及说明

| 属 性 | 取 值 | 说 明 |
|-------|-------|----------------|
| name | name | 定义 input 标记的名称 |
| value | value | 规定 input 标记的值 |

在表单中插入提交按钮时，如果不设置属性 value 的值，它的初始值是“提交查询按钮”。所以一定要给 value 属性赋值。

7. 重置按钮

设置 input 标记的 type 属性值为 reset，可以实现向表单中插入一个重置按钮，重置按钮用于将表单中所有的输入信息清空，然后让用户可以重新填写。

语法：

```
<input name=" " type="reset" value="" />
```

说明：重置按钮的属性主要有 name 和 value，其属性、取值及说明与提交按钮的相同。

8. 普通按钮

设置 input 标记的 type 属性值为 button，可以实现向表单中插入一个普通按钮。普通按钮在网页设计中非常有用，如果不通过表单提交按钮来处理事件，则可以给普通按钮绑定事件代码来实现所需的功能。

语法：

```
<input name="" type="button" value="" onclick="" />
```

说明：普通按钮的属性有 name，value 和 onclick，其属性、取值及说明如表 3-13 所示。

表 3-13 普通按钮属性、取值及说明

| 属 性 | 取 值 | 说 明 |
|---------|-------|-----------------------|
| name | name | 定义 input 标记的名称 |
| value | value | 规定 input 标记的值 |
| onclick | 事件代码 | 绑定事件代码、自定义函数或直接使用脚本代码 |

【例 3.14】三种按钮的应用。

```
<! --示例 3.14 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" >
    <title>三种按钮的应用</title>
    <style type="text/css">
      input{
        width:100px;
        height:25px;
      }
    </style>
  </head>
</html>
```



```
body { text-align:center; }
fieldset {
    width:400px;
    height:180px;
}
</style>
</head>
<body>
    <form>
        <fieldset>
            <legend>三种按钮的应用</legend>
            <h3>请输入用户信息:</h3>
            用户名:<input type="text" name="username" size="10"/>
            密码:<input type="password" name="password" size="10"/><br/><br/>
            &nbsp;&nbsp;&nbsp;<input type="submit" name="submit" value="提交"/>
            <input type="reset" name="reset" value="重置"/>
            <input type="button" name="button" value="注册新用户" onclick="javascript:alert('
注册新用户');"/>
        </fieldset>
    </form>
</body>
</html>
```

例 3.14 浏览效果如图 3-15 所示。



图 3-15 三种按钮的应用

例 3.14 在表单中插入一个提交按钮，名称为 submit，值为“提交”；插入一个重置按钮，名称为 reset，值为“重置”；还插入一个普通按钮，名称为 button，值为“注册新用户”；点击“注册新用户”按钮后，触发 onclick 事件，执行 JavaScript 代码，弹出“注册新用户”告警框，如图 3-15 所示。

9. 文件选择框

设置 input 标记的 type 属性值为 file，可以实现向表单中插入一个文件选择框。

语法：

```
<input name="" type="file" value="" />
```

说明：

name，定义 input 标记的名称。页面上会自动添加一个文本输入框和一个“浏览……”按钮。单击“浏览…”按钮可以从“选择要加载的文件”对话框选择某一个文件，然后自动将文件名称回填到文本输入框中。

10. 隐藏框

设置 input 标记的 type 属性值为 hidden，可以实现向表单中插入一个隐藏框，用户提交表单时，隐藏框的信息也会一起提交到服务器，但隐藏框在网页中是不可见的。

语法：

```
<input name="" type="hidden" value="" />
```

说明：隐藏框的属性有 name 和 value，其属性、取值及说明如表 3-14 所示。

表 3-14 隐藏框属性、取值及说明

| 属 性 | 取 值 | 说 明 |
|-------|-------|----------------|
| name | name | 定义 input 标记的名称 |
| value | value | 规定 input 标记的值 |

【例 3.15】 文件选择框与隐藏框的应用。

```
<! --示例 3.15 程序-->
```

```
<! DOCTYPE html>
```

```
<html>
```

```
  <head>
```

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

```
    <title>文件选择框与隐藏框的应用例</title>
```

```
    <style type="text/css" >
```

```
      fieldset {
```

```
        width:500px;
```

```
        height:200px;
```

```
        margin:20p;
```

```
      }
```

```
    </style>
```

```
  </head>
```

```
<body>
```

```
<form>
  <fieldset>
    <legend align="middle">文件选择框与隐藏框的应用例</legend>
    <h4>请输入个人信息:</h4>
    姓名:<input type="text" name="name" size="10" />
    性别:<input type="radio" name="sex" value="male" />男
    <input type="radio" name="sex" value="female" />女 &nbsp;   
    年龄:<input type="text" name="age" size="8" /><br/>
    <h4>请选择文件:</h4>
    <input type="file" name="file" size="40" /><br>
    <input type="hidden" name="admin" value="ABCD" />
  </fieldset>
</form>
</body>
</html>
```

例 3.15 浏览效果如图 3-16 所示。



图 3-16 文件选择框与隐藏框的应用

在例 3.15 中插入一个文件选择框，名称为 file，用户可选择相关文件。选择“浏览……”按钮后，弹出“选择要添加的文件”对话框，单击“打开”按钮，所选文件的名称自动回填到文本输入框内。最后还插入了一个名称为 admin 的隐藏框，但是浏览时并没有显示。

3.4.4 多行文本输入框

textarea 标记可以向表单中插入多行文本输入框。多行文本输入框可以用来输入较多的文字信息，而且可以换行，并将这些信息提交到服务器。

语法：

```
<textarea name="" rows="" cols="" wrap="" />初始信息内容</textarea>
```

说明：多行文本输入框 `textarea` 标记是成对标记，其主要属性有 `name`，`rows`，`cols` 和 `wrap` 等，其属性、取值及说明如表 3-15 所示。默认情况下，当用户在文本区域中输入文本后，浏览器会将它们按照输入时的状态发送给服务器。只有在用户按下 `Enter` 键的地方生成换行。

表 3-15 多行文本输入框属性、取值及说明

| 属 性 | 取 值 | 说 明 |
|-------------------|---|---|
| <code>name</code> | <code>name</code> | 定义 <code>textarea</code> 标记的名称 |
| <code>rows</code> | <code>number</code> | 规定文本区内的可见行数 |
| <code>cols</code> | <code>number</code> | 规定文本区内的可见宽度 |
| <code>wrap</code> | <code>wrap</code> <code>virtual</code> <code>physical</code> <code>off</code> <code>soft</code> <code>hard</code> | <code>wrap</code> 属性规定当在表单中提交时，文本区域中的文本如何换行 <code>wrap</code> ——文本区会包含一行文本，用户必须将光标移动到右边才能看到全部文本，这时将把一行文本传送给服务器； <code>virtual</code> ——将实现文本区内的自动换行，但在传输给服务器时，文本只在用户按下 <code>Enter</code> 键的地方进行换行，其他地方没有换行的效果； <code>physical</code> ——将实现文本区内的自动换行，并以这种形式传送给服务器； <code>off</code> ——不会自动换行，输入内容超出文本域右界时，文本将向左滚动，必须按 <code>Enter</code> 键才能将插入点移到下行。 HTML5 中， <code>soft</code> 表示提交时不换行， <code>hard</code> 表示提交时换行 |

【例 3.16】 征求意见表。

```
<! --示例 3.16 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>多行文本输入框的应用</title>
  </head>
  <body>
    <form>
      <h3>请您填写宝贵意见:</h3>
      <textarea name="info" rows="4" cols="50" wrap="virtual">
      </textarea>
    </form>
  </body>
</html>
```

例 3.16 浏览效果如图 3-17 所示。



图 3-17 多行文本输入框的应用

3.4.5 下拉列表框

下拉列表可以在表单中接收用户的输入。下拉列表通常需要同时使用 `select` 和 `option` 标记来在表单中插入下拉菜单和列表项。

语法：

```
<select name="" size="" multiple>
  <option value="" selected>文字信息 1</option>
  <option value="">文字信息 2</option>
  .....
</select>
```

说明：`select` 标记是成对标记，`option` 标记是单个标记，但应该把它补成成对标记，结构更为清晰。`select` 标记有 `name`、`size` 和 `multiple` 等属性。`option` 标记有 `value` 和 `selected` 等属性。`select` 标记与 `option` 标记必须配合使用。每一选项必须指定一个显示的文本和一个 `value` 值，显示文本通常附在 `option` 标记后面。它们的属性、取值及说明如表 3-16 所示。

表 3-16 `select` 和 `option` 标记属性、取值及说明

| 标记名称 | 属 性 | 取 值 | 说 明 |
|--------|----------|----------|-----------------|
| select | name | name | 定义 select 标记的名称 |
| | size | number | 规定下拉列表框中可见选项的数目 |
| | multiple | multiple | 规定可选择多个选项 |
| option | value | value | 规定列表项的值 |
| | selected | selected | 设置预选列表项 |

【例 3.17】 下拉列表框的应用。

```
<!-- 示例 3.17 程序 -->
<! DOCTYPE html>
```

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>下拉列表框的应用</title>
  </head>
  <body>
    <form>
      <h3>请选择您的课程:</h3>
      <select name="course" size="4" multiple>
        <option value="c1" selected>C 语言程序设计</option>
        <option value="c2">计算数学</option>
        <option value="c3">数据结构</option>
        <option value="c4">Java 程序设计</option>
        <option value="c5">网页设计与制作</option>
      </select>
    </form>
  </body>
</html>
```

例 3.17 浏览效果如图 3-18 所示。



图 3-18 下拉列表框的应用

在例 3.17 中插入了一个下拉列表框，名称为 `course`，选项数目为 5，设置 `multiple` 属性支持多选；列表项“C 语言程序设计”为默认选择项。

3.5 综合案例——表格与表单

本节将通过两个综合案例来说明表格布局和表单交互的使用。

案例 1——达维工作室“联系我们”子页

采用嵌套表格制作达维工作室的“联系我们”子页，其页面效果如图 3-19 所示。



图 3-19 “联系我们”子页

以下为完整代码，以供参考。

```
<! --综合案例 1. html-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>达维工作室——联系我们</title>
    <style type="text/css">
      .chu {
        font-weight:bold;
      }
      .zi1 {
        font-family:"微软雅黑";
        font-size:20px;
        font-weight:bold;
        color:#ED630A;
      }
      .zi2 {
        font-family:"微软雅黑";
        font-weight:bold;
      }
    </style>
  </head>
  <body>
    <table border="1">
      <tr>
        <td>网站首页</td>
        <td>关于我们</td>
        <td>团队合作</td>
        <td>相关作品</td>
        <td>设计理念</td>
        <td>人物介绍</td>
        <td>联系我们</td>
      </tr>
      <tr>
        <td colspan="7">
          <table border="1">
            <tr>
              <td style="text-align:center">联系我们</td>
              <td style="text-align:center">关于我们</td>
              <td style="text-align:center">团队合作</td>
            </tr>
            <tr>
              <td>地址: 广东省江门市XXXXXXXXXX  
电话: 0750-XXXXXX  
传真: 0750-XXXXXX  
QQ: 12345678  
电子邮箱:  
123@163.com  
工作室网站:  
www.XXXX.com</td>
              <td>达维工作室是专业从事互联网相  
关开发的公司。  
专门提供全方位的优质服务和最  
专业的网站建设方案为企业打造  
全新电子商务平台。  
达维工作室成立于2014年, 已经  
成为国内著名的网站建设提供  
商, 多年的风雨历程...</td>
              <td>成员都具有多年的实际设计工作  
经验, 满足客户的国际化需求。  
设计新创意的思维模式, 提供最  
适合的设计方案。  
  
本工作室设计与制作的网站均属  
原创, 不套用网上的任何模板,  
根据每个公司特点, 设计出属于  
客户...</td>
            </tr>
            <tr>
              <td style="text-align:right"><a href="#">更多>></a></td>
              <td style="text-align:right"><a href="#">更多>></a></td>
              <td style="text-align:right"><a href="#">更多>></a></td>
            </tr>
          </table>
        </td>
      </tr>
    </table>
  </body>
</html>
```



```

        <td width="280" valign="top">
        <table width="100%" border="0" cellspacing="0" cellpadding="20">
        <tr>
        <td height="30" align="center" bgcolor="#FFFFFF" class="zi1">联系我们</td>
        </tr>
        <tr>
        <td height="196" bgcolor="#FFAF03">
        <p class="zibai">地址:广东省江门市 XXXXXXXXXX<br />
        电话:0750-XXXXXX<br />
        传真:0750-XXXXXX<br />
        QQ:12345678<br />
        电子邮箱:<br />
        123@163.com<br />
        工作室网站:<br />
        www.XXXX.com</p>
        </td>
        </tr>
        </table>
    </td>
    <td width="280" valign="top">
    <table width="100%" border="0" cellspacing="0" cellpadding="20">
    <tr>
    <td height="30" align="center" bgcolor="#FFFFFF" class="zi1">关于我们</td>
    </tr>
    <tr>
    <td height="278" valign="top" bgcolor="#FC880D"><p class="zibai">达维工作室是专业
    从事互联网相关开发的公司。<br />
    专门提供全方们的优质服务和最专业的网站建设方案为企业打造全新电子商务平台。<br
    />
    达维工作室成立于2014年,已经成为国内著名的网站建设提供商。多年的风雨历程……
    </p>
    <p class="zibai">&nbsp;&nbsp;&nbsp;</p>
    <p class="zibai chu"><a href="#">更多 &gt;&gt;</a></p>
    </td>
    </tr>
    </table>
    </td>
    <td width="280">
    <table width="100%" border="0" cellspacing="0" cellpadding="20">
    <tr>
    <td height="30" align="center" bgcolor="#FFFFFF" class="zi1">团队合作</td>
    </tr>
    <tr>

```



```
<html>
  <head>
    <meta charset="UTF-8">
    <title>用户注册信息</title>
    <style type="text/css">
      fieldset {
        width:700px;
      }
    </style>
  </head>
  <body>
    <form action="" method="post" enctype="multipart/form-data" name="form1" id="form1">
      <fieldset>
        <legend>用户注册信息</legend>
        <table width="600" border="0" align="center" cellpadding="0" cellspacing="0">
          <tr>
            <td width="200" align="right">用户名:</td>
            <td><input type="text" name="textfield" id="textfield" /></td>
          </tr>
          <tr>
            <td width="200" align="right">密码:</td>
            <td><input type="password" name="textfield2" id="textfield2" /></td>
          </tr>
          <tr>
            <td width="200" align="right">确认密码:</td>
            <td><input type="password" name="textfield3" id="textfield3" /></td>
          </tr>
          <tr>
            <td width="200" align="right">性别:</td>
            <td><input name="radio" type="radio" id="radio" value="radio" checked="checked" />
            男
            <input type="radio" name="radio" id="radio2" value="radio2" />
            女</td>
          </tr>
          <tr>
            <td width="200" align="right">出生年月:</td>
            <td><input name="textfield4" type="text" id="textfield4" size="12" />年
            <select name="select" id="select">
              <option>1</option>
              <option>2</option>
              <option>3</option>
              <option>4</option>
            </select>
          </td>
        </table>
      </fieldset>
    </form>
  </body>
</html>
```

```
<option>5</option>
<option>6</option>
<option>7</option>
<option>8</option>
<option>9</option>
<option>10</option>
<option>11</option>
<option>12</option>
</select>
    月</td>
</tr>
<tr>
    <td width="200" align="right">业余爱好:</td>
    <td><input type="checkbox" name="checkbox" id="checkbox" />看书
    <input type="checkbox" name="checkbox2" id="checkbox2" />上网
    <input type="checkbox" name="checkbox3" id="checkbox3" />打球
</td>
</tr>
<tr>
    <td width="200" align="right">相片:</td>
    <td height="25"><input type="file" name="fileField" id="fileField" /></td>
</tr>
<tr>
    <td width="200" align="right">意见或建议:</td>
    <td>
        <textarea name="textarea" id="textarea" cols="45" rows="5"></textarea>
    </td>
</tr>
<tr>
    <td colspan="2" align="center">
        <input type="submit" name="button" id="button" value="提交" />
        <input type="reset" name="button2" id="button2" value="重置" />
    </td>
</tr>
</table>
</fieldset>
</form>
</body>
</html>
```

本章习题

1. 采用表格布局完成“CASIO 计算器”外观设计，其中表格的每一个单元格均需要设

计带边框，效果如图 3-21 所示。



图 3-21 计算器页面布局

2. 编写程序实现如图 3-22 所示的“登录页面”。



图 3-22 登录页面

第 4 章 CSS 样式基础



学习目标

1. 了解 CSS 的概念、特点；
2. 掌握 CSS 基本语法、选择器分类与声明的结构；
3. 掌握 CSS 的定义及引用的方式；
4. 理解 CSS 继承与层叠的含义。



知识要点

1. CSS 的概念和语法基础；
2. CSS 的引入方式；
3. CSS 的属性理解；
4. CSS 的继承和层叠对比；
5. 元素类型介绍。

4.1 CSS 概述

随着 HTML 的发展，为了满足页面设计者的要求，HTML 添加了很多显示功能，但是随着这些功能的增加，使得 HTML 变得越来越杂乱，HTML 页面也越来越臃肿，CSS 便诞生了。CSS 是用于简化 HTML 标签，把关于样式部分的内容提取出来，进行单独地控制，使结构与样式分离式开发。对页面布局等的控制也更加精确，同时能够实现内容和表现的分离，使得网站的设计风格趋向统一、维护更加容易，并且能够被多种浏览器支持。

4.1.1 CSS 的基本概念

CSS 全称为“Cascading Style Sheet”，中文解释为“层叠样式表”，它是以 HTML 为基础，设置网页的外观显示样式，如字体、颜色、背景的控制及整体的布局等，还可以针对不同的浏览器设置不同的样式。在网页设计时采用 CSS 技术，可以有效地对页面的布局、字体、颜色、背景和其他效果实现更加精确的控制。只要对相应的代码做一些简单的修改，既可以改变同一页面的不同部分效果，也可以改变同一个网站中不同网页的外观和格式。

4.1.2 传统 HTML 的缺点

HTML 标记是用来定义文档内容，比如通过 `h1`、`p` 和 `table` 等标记表达“这是标题”、“这是段落”、“这是表格”等信息，而文档布局由浏览器完成。随着新的 HTML 标记（比如字体标记和颜色属性）添加到 HTML 规范中，要实现页面美工、文档内容清晰、独立于文档表现层的站点变得越来越困难。传统 HTML 的缺点主要体现在如下几方面。

1. 维护困难

为了修改某个特殊标记的格式，需要花费很多的时间，尤其对于整个网站而言，后期修改和维护的成本很高。

2. 标记不足

HTML 自身的标记并不丰富，很多标记都是为网页内容服务的，而关于美工的标记，例如文字间距、段落缩进等，这些 HTML 中都很难找到。

3. 网页过“胖”

由于对各种风格样式没有统一进行控制，用 HTML 编写的页面往往是体积过大，占用了很多宝贵的带宽。

4. 定位困难

在整体页面布局时，HTML 对于各个模块的位置调整很困难。

4.1.3 CSS 的特点和优势

CSS 通过定义标记如何表现，对页面结构风格进行控制，分离文档的内容和表现，克服了传统 HTML 的缺点。将 CSS 嵌入在页面中，通过浏览器解释执行，而且 CSS 文件是文本文件，只要理解 HTML 都可以掌握它。其特点主要如下。

(1) 页面的字体变得更漂亮，更容易编排。

(2) 可以轻松控制页面的布局。

(3) 可以在大多数浏览器上使用。

(4) 以前一些必须通过图像转换才能实现的功能，现在只要用 CSS 就可以轻松实现，页面下载更快。

(5) 可以使用一个 CSS 文件控制整个网站的显示风格。只要修改该 CSS 文件中相应的行，就可以改变整个网站上页面的显示样式。

CSS 可以称得上 Web 设计领域的一个突破，具有以下优势。

1. 表现和内容分离

CSS 通过定义 HTML 标记如何显示控制网页的格式，使得页面内容和表现分离，简化了网页格式设计，也使得对网页格式的修改更方便。

2. 增强了网页的表现力

CSS 样式属性提供了比 HTML 更多的格式设计功能。例如，可以通过 CSS 样式去掉网页中超链接的下划线，甚至可以为文字添加阴影效果等。

3. 使整个网站显示风格趋于统一

将 CSS 样式定义到样式表文件中，然后在多个网页中同时应用样式表文件中的样式，就可以确保多个网页具有一致的格式，并且可以随时更新样式表文件，实现自动更新多个网页的格式功能，从而大大降低网站的开发与维护的成本。

4.1.4 CSS 的编写规则

利用 CSS 样式设计虽然很强大，但是如果设计人员管理不当将导致样式混乱。本节介绍学习 CSS 编写中的一些技巧和规则。

1. 目录结构命名规则

存放 CSS 样式文件的目录一般命名为 style 或 css。

2. 样式文件的命名规则

在项目初期，会把不同类别的样式放于不同的 CSS 文件，是为了 CSS 编写和调试的方便；在项目后期，为了网站性能上的考虑，会将不同的 CSS 文件整合到一个 CSS 文件中，这个文件一般命名为 style.css 或 css.css。

3. 选择器的命名规则

所有选择器必须由小写英文字母或下划线组成，必须以字母开头，不能为纯数字。设计者要用有意义的单词或缩写组合来命名选择器，做到“见名知义”，这样就节省了查找样式的时间。样式名必须能够表示样式的大概含义（禁止出现如 Div1, Div2 和 Style 等自名），读者可以参考表 4-1 中的样式命名。

表 4-1 常用样式命名参考

| 页面功能 | 说 明 | 页面功能 | 说 明 |
|----------------------|----------|------------------|------|
| #container 或#content | 容器，用于最外层 | #main | 页面主体 |
| #layout | 布 局 | #tag | 标 签 |
| #hea, #header | 头 部 | #msg | 提示信息 |
| #foot, #footer | 页脚部分 | #tips | 小技巧 |
| #nav | 主导航 | #vote | 投 票 |
| #subnav | 二级导航 | #friendlink | 友情连接 |
| #menu | 菜 单 | #title | 标 题 |
| #copyright | 版权信息 | #siteinfo | 网站信息 |
| #branding | 商 标 | #siteinfoLegal | 法律声明 |
| #logo | 网站 LOGO | #siteinfoCredits | 信 誉 |
| #joinus | 加入我们 | | |

4. CSS 代码注释

像其他语言一样，CSS 允许用户在源代码中嵌入注释。CSS 注释被浏览器忽略，不影响

网页效果。注释有助于记住复杂的样式规则的作用、应用的范围等，便于样式规则的后期维护和应用。CSS 注释以字符“/*”开始，以字符“*/”结束。注释可以是单行，也可以是多行，并可以出现在 CSS 代码的任何地方。

5. CSS 代码注释

代码缩进可以保证 CSS 代码的清晰可读。在实际应用中，可以按一次 Tab 键来缩进选择器，按两次 Tab 键来缩进声明和结束大括号。这样的排版规则便于查询，避免代码混乱。

4.2 CSS 语法基础

CSS 控制页面是通过 CSS 规则实现的，CSS 规则由选择器和声明组成，声明由属性和属性值对组成。CSS 提供了丰富的选择器类型，包括标记选择器、类选择器、id 选择器及伪类选择器等，能够灵活地对整个页面、页面中的某个标记或一类标记进行样式设置。

4.2.1 CSS 基本语法

CSS 就是一个包含一个或多个规则的文本文件，CSS 规则由两个主要的部分构成：选择器（Selector 选择器）和声明（Declaration）。

选择器通常是是需要改变样式的 HTML 元素。

声明由一个或多个属性与属性值对组成。属性是 CSS 的关键字，如 font-family（字体）color（颜色）和 border（边框）等，属性用于指定选择器某一方面的特性，而属性值则用于指定选择器的特性的具体特征。

1. 基本语法

```
selector { property1: value1; property2: value2; property3: value3; ... }
```

选择器 { 属性 1: 属性值 1; 属性 2: 属性值 2; 属性 3: 属性值 3; ... }

2. 语法说明

(1) 选择器。选择器可以是 HTML 标记名称或者属性的值，也可以是自定义的标识符。

(2) 属性/属性值对。“属性：属性值”必须一一对应，属性与属性值之间必须用“:”连接，每个属性/属性值对之间用分号（;）隔开。

(3) 属性。在 CSS 中对属性命名与脚本语言中有一点不同，即属性名称的写法，在 CSS 中，凡是属性名为两个或两个以上的单词构成时，单词之间以连词符号（-）分隔，例如背景颜色属性 background-color；而在脚本语言中，对象属性则连写成 backgroundColor。在脚本语言中，属性由两个以上单词构成，则从第二个单词开始向后，所有单词首字母必须大写。

下面是一个简单样式表的示例。



```
p { background-color:red;font-size:20px;color:green;}
```

上例的 CSS 规则中 p 为选择器，background-color，font-size 和 color 为属性，red，20px 和 green 为属性值。该 CSS 规则将 HTML 中的所有段落统一设置成“背景色为红色、字体大小为 20 像素以及字体颜色为绿色”。通常为了增强样式定义的可读性，建议每行只描述一个属性，代码如下所示：

```
p {  
    background-color:red;  
    font-size:20px;  
    color:green;  
}
```

(4) 复合属性。在 CSS 中有些属性可以表示多个属性的值。如关于文字的设置，有 font-family（字体）、font-size（字体大小）、font-style（字体风格），这些可以用一个属性 font 来表示，例如：

```
p {  
    font-style:italic;  
    font-size:20px;  
    font-family:黑体;  
}
```

可以如下表示：

```
p {  
    font:italic 20px 黑体;  
}
```

值得注意的是，当使用 font 复合属性在一个声明中设置所有字体属性时，应按照 font-style，font-variant，font-weight，font-size/line-height，font-family 的顺序，可以不设置其中的某个值，比如“font: 100% verdana;”仅设置了 font-size 和 font-family 属性，其他未设置的属性会使用其默认值。类似的复合属性还有 border，margin 和 padding 等。

(5) 多个属性值。在 CSS 中有些属性可以设置多个属性值，用逗号（,）分隔。例如

```
selector {  
    font-family:"楷体_gb2312","黑体","Times New Roman";  
}
```

该样式表说明了可以使用楷体_gb2312、黑体、Times New Roman 3 种字体来设置 selector 的字体效果。若系统中找不到楷体_gb2312，则使用黑体，若也没有黑体，则使用 Times New Roman，即按出现的先后顺序优先选择。

4.2.2 CSS 选择器类型

CSS 选择器主要有 4 种类型：标记选择器、class 选择器、ID 选择器及伪类选择器等。

1. 标记选择器

标记选择器（也可称为“元素选择器”）即直接使用 HTML 标记名称作为选择器。它定义的样式作用于页面中所有与选择器同名的标记。

2. class 选择器

class 选择器也叫“类”选择器，可以给指定的标签设置一个 class 属性和 class 值然后通过 class 选择器找到对应的标签，为其设置 class 值，即添加 CSS 样式。使用 class 类选择器时，需要用英文（.）进行表示，代码如下所示。

【例 4.1】 class 选择器和标记选择器。

```
<! --示例 4.1 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>class 选择器</title>
    <style type="text/css">
      p{ /* 标记选择器 */
        color:blue;
        font-size:28px;
      }
      . red{ /* class 选择器 */
        color:red; /* 红色 */
        font-size:18px; /* 文字大小 */
      }
      . green{
        color:green; /* 绿色 */
        font-size:20px; /* 文字大小 */
      }
    </style>
  </head>
  <body>
    <p>class 选择器与标记选择器</p>
    <p class="red">class 选择器 1</p>
    <p class="green">class 选择器 2</p>
    <h3 class="green">h3 同样适用</h3>
  </body>
</html>
```

例 4.1 浏览效果如图 4-1 所示。



图 4-1 class 选择器和标记选择器

3. ID 选择器

ID 选择器用来对某个单一元素定义单独的样式。ID 选择器只能在 HTML 页面中使用一次，针对性更强。定义 ID 选择器时，需要在 id 名称前加一个“#”。接下来通过实例来演示 ID 选择器。

【例 4.2】 ID 选择器。

```
<! --示例 4.2 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" >
    <title></title>
    <style type="text/css">
      #green {
        font-size:30px;           /* 字体大小 */
        color:#009900;          /* 颜色 */
      }
    </style>
  </head>
  <body>
    <p id="green">ID 选择器 1</p>
  </body>
</html>
```

例 4.2 浏览效果如图 4-2 所示。



图 4-2 ID 选择器

对于 CSS 来说，ID 选择器与 class 选择器的功能很相似但不完全相同。一般来说，class 选择器更加灵活，能完成 ID 选择器的所有作用，也能完成更加复杂的功能应用。如果对样式可重用性要求较高，则应该使用 class 选择器将新元素添加到类中来完成，对于需要唯一标识的页面元素，则可以使用 ID 选择器。

4. 伪类选择器

前面介绍的选择器都是能够与 HTML 中具体标记对应的，但是像段落的第 1 行，超链接访问前与访问后等，就没有 HTML 标记与之对应，从而也没有简单的 CSS 选择器应用，为此 CSS 引进了伪类选择器。常用伪类见表 4-2。

表 4-2 常用伪类

| 伪类名 | 说 明 |
|-------------|----------------------|
| link | 用于可以设置未被访问的链接的样式 |
| visited | 用于设置已经被访问的链接的样式 |
| hover | 用于设置将鼠标悬浮在链接上的样式 |
| active | 用于设置鼠标点击链接时到鼠标松开时的样式 |
| focus | 用于设置用键盘将焦点放在链接上时的样式 |
| first-child | 设置第一个标记的样式 |
| lang | 设置具有 lang 属性的标记的样式 |

接下来通过实例来演示伪类选择器。

【例 4.3】 伪类选择器。

```
<! --示例 4.3 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>伪类选择器</title>
    <style type="text/css">
      a:link { color:gray;text-decoration:none; }
      a:visited { color:blue;text-decoration:none; }
      a:hover { color:red;text-decoration:none; }
      a:active { color:yellow;text-decoration:none; }
      p:first-letter { font-weight:bold;font-family:"微软雅黑";font-size:larger; }
    </style>
  </head>
  <body>
    <p>伪类选择器</p>
    <a href="http://www.sina.com">新浪首页</a>
  </body>
</html>
```

例 4.3 浏览效果如图 4-3 所示。



图 4-3 伪类选择器

特别注意：`a: hover` 必须位于 `a: link` 和 `a: visited` 之后，才有效；`a: active` 必须位于 `a: hover` 之后才有效。

4.2.3 CSS 选择器声明

在声明各种 CSS 选择器时，如果某些选择器的风格是完全相同的，或者部分相同，都可以使用集体声明的方法，用“，”分隔多个选择器，对风格相同的 CSS 选择器同时声明。

1. 集体声明

集体声明代码如下：

```
h3,h4,h5,p{ color:red;font-size:18px;}
```

2. 全局声明

对于实际网站中的一些小型页面，例如弹出的小对话框和上传附件的小窗口等，希望这些页面中所有的标记都使用同一种 CSS 样式，但又不希望通过逐个加入集体声明列表的方式。这时可以利用全局声明符号“*”，代码如下：

```
* {  
    color:red;  
    font-size:18px;  
}
```

3. 派生选择器（上下文选择器）

另外，根据标记所在位置的上下文关系来定义样式，可以使标记更加简洁，派生选择器允许根据文档的上下文关系来确定某个标记的样式。通过合理地使用派生选择器，可以使 CSS 代码变得更加整洁。例如，要让列表项 `` 中的 `` 标记变为斜体字，而不是通常的粗体字，可以这样定义一个派生选择器：

```
li strong { font-style:italic;font-weight:normal;}  
strong { font-weight:bold;}
```

代码如下：

```
<p><strong>我是粗体字,不是斜体字,因为我不在列表当中,所以这个规则对我不起作用</strong></p>  
<ol>  
    <li><strong>我是斜体字,这是因为 strong 元素位于 li 标记内</strong></li>  
    <li>我是正常的字体</li>  
</ol>
```

在上面的例子中，有两个 `strong` 标记，但只有 `li` 元素中的 `strong` 元素的样式为斜体字，而且无须为 `strong` 标记定义特别的 `class` 或 `id`，应用派生选择器，代码更加简洁。

4.3 CSS 引入方式

CSS 按其位置可以分为 4 种：内联样式表（Inline Style Sheet）、内部样式表（Internal

Style Sheet)、链接外部样式表 (Link External Style Sheet) 以及导入外部样式表 (Import External Style Sheet)。

4.3.1 内联样式表 (行内样式表)

内联样式表的 CSS 规则写在首标记内, 只对所在的标记有效, 几乎任何一个 HTML 标记上都可以设置 style 属性, 属性值可以包含 CSS 规则的声明, 不包含选择器。

1. 基本语法

```
<标记 style="属性 1:属性值 1;属性 2:属性值 2;...">修饰的内容</标记>
```

2. 语法说明

语法说明:

- (1) 标记是指 HTML 标记, 如 p, h1 和 body 等标记;
- (2) 标记的 style 定义的声明只对自身起作用;
- (3) style 属性的值可以包含多个声明, 每一声明之间用 “;” 分隔;
- (4) 标记自身定义的 style 样式优先于其他所有样式定义。

内联样式表的应用如以下实例所示。

【例 4.4】 内联样式表。

```
<! --示例 4.4 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>内联样式表</title>
  </head>
  <body>
    <p style="color:#FF0000;font-size:20px;text-decoration:underline;">正文内容 1</p>
    <p style="color:#000000;font-style:italic;">正文内容 2</p>
    <p style="color:#FF00FF;font-size:25px;font-weight:bold;">正文内容 3</p>
    <p>这段话没有内联样式</p>
  </body>
</html>
```

例 4.4 浏览效果如图 4-4 所示。



图 4-4 内联样式表

4.3.2 内部样式表

内部样式表写在 HTML 的 `<head></head>` 里面，只对所在的网页有效。使用 `< style ></ style >` 标记对来放置 CSS 规则。

1. 基本语法

```
! <style type="text/css">
选择器 1 { 属性 1:属性值 1;属性 2:属性值 2;... }
选择器 2 { 属性 1:属性值 1;属性 2:属性值 2;... }
选择器 n { 属性 1:属性值 1;属性 2:属性值 2;... }
</style>
```

2. 语法说明

`style` 标记是成对标记，有 1 个 `type` 属性是指 `style` 元素以 CSS 的语法定义。

选择器 1, ..., 选择器 n, 可以定义 n 个选择器，再定义声明部分。

属性和属性值之间用冒号连接，“属性/属性值”对之间用分号分隔。

本书中，示例程序 4.1~4.3 全部使用内部样式表，此处不再举例。

4.3.3 外部样式表

外部样式表是将 CSS 规则写在以 `.css` 为后缀的 CSS 文件里，在需要用到此样式的网页



里引用该 CSS 文件。根据引用的方式不同可以分为链接外部样式表和导入外部样式表，它们形式上的区别在于链接外部样式表通过链接 link 标记，导入外部样式表必须在内部样式表内首行通过“@import url (外部样式文件);”来定义。

1. 链接外部样式表

链接式是将所有的样式放在一个或多个以“.css”为扩展名的外部样式表中，通过网页的<head></head>标签中使用<link>标记将外部样式表文件链接到 HTML 文档中。

(1) 基本语法:

```
<link type="text/css" rel="stylesheet" href="外部样式表名.css">
```

(2) 语法说明:

语法中必须指定<link>标记的 3 个属性，其中 href 是定义链接外部样式表文件的 url 可以是相对路径和绝对路径；rel 是定义当前文档与被链接文档之间的关系，这里指定 stylesheet，表示被链接的文档是样式表文件；type 是定义链接文档的类型，这里类型指定为 text/css，表示链接的外部文件为 CSS 样式表。

这里面涉及 3 个属性，rel="stylesheet" 规定当前文档与被链接文档之间的关系。type="text/css" 规定被链接文档的 MIME 类型。href="mystyle.css" 规定被链接文档的位置，只需要记住这个固定写法即可。

以下示例演示链接外部 CSS 样式表:

【例 4.5】链接外部样式表。

```
<! --示例 4.5 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>链接外部样式表</title>
    <link href="4-5.css" type="text/css" rel="stylesheet">
  </head>
  <body>
    <h2>CSS 标题</h2>
    <p>这是正文内容……</p>
    <h2>CSS 标题</h2>
    <p>通过 link 标签将外部样式表 4-5.css 链接到 HTML 文档中</p>
  </body>
</html>
```

例 4.5 浏览效果如图 4-5 所示。



图 4-5 链接外部样式表

在例 4.5 中，链接引入外部样式表的同一个 CSS 文件可以供多个网页引用，一个 HTML 文档也可以通过多个 `<link>` 标签链接多个 CSS 样式表。

2. 导入外部样式表

导入式是将一个独立的 CSS 文件导入 HTML 文档中，其是在 HTML 文档 `< head>` 中应用 `< style>` 标签，并在 `< style>` 标签中的开头处用 `@ import` 语句，即可导入外部样式表文件。

(1) 基本语法：

```
<style type="text/css" >
@ import url("CSS 文件路径");
/* 此处还可以存放其他 CSS 样式 */
</style>
```

(2) 语法说明。导入样式表必须在 `style` 标记内开头的位置定义，可以同时导入多个外部样式表，每条语句必须以“;”结束。一般导入外部样式写在最前面，内部样式写在后面。

“`@ import`”必须连续书写，即“`@`”和“`import`”之间不能留有任何空格。外部样式表的名称在书写时，必须是全称再加上后缀名“`.css`”，如“`4-5.css`”。

导入式会在整个网页加载完后再加载 CSS 文件，因此如果网页比较大则会出现先显示无样式的页面，再出现网页的样式的情况。这是导入式固有的一个缺陷。

虽然导入式和链接式功能基本相同，但大多数网站都是采用链接式引入外部样式表，这是因为两者的加载时间和顺序不同。在加载页面时，`<link>` 标签引用的 CSS 样式表将同时被加载，而“`@ import`”引用的 CSS 样式表会等整个网页下载结束后再被加载。可能会显示

无样式的页面，造成不好的用户体验。因此，大多数的网站采用链接外部样式表是使用频率最高、最实用的 CSS 样式表。它可以将 HTML 代码与 CSS 代码分离为两个或多个文件，实现类结构和表现的完全分离，使网页的前期制作和后期维护都变得十分方便。

以下示例演示导入外部 CSS 样式表：

【例 4.6】 导入外部样式表。

```
<! --示例 4.6 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>导入外部样式表</title>
    <style type="text/css">
      @import url("4-5.css");
    </style>
  </head>
  <body>
    <h2>CSS 标题</h2>
    <p>这是正文内容...</p>
    <h2>CSS 标题</h2>
    <p>通过 style 标签将外部样式表 4-5.css 导入到 HTML 文档中</p>
  </body>
</html>
```

例 4.6 浏览效果如图 4-6 所示。



图 4-6 导入外部样式表

4.4 CSS 的属性单位

在 CSS 文字、排版、边界等的设置上，常常会在属性值后加上长度或者百分比单位，通过本节的学习将掌握两种单位的使用。

4.4.1 长度、百分比单位

长度单位通常由两个英文字母的缩写表示，且设置大部分属性时仅能使用正数，只有少数属性可以使用正、负数的设置。而且必须注意，若属性值设置为负数，且超过浏览器所能接受的范围，以至于无法支持时，浏览器将会选择比较靠近且能支持的数值，长度单位在设置时分别为以下两种不同的类型。

1. 相对类型

其指以该属性的前一个属性的单位值为基础来完成目前的设置，在浏览器内常用且支持的两种长度单位是以相对类型出现的。如用 `em`（当前字母中，字母的宽度）作为属性单位时，将会依据父元素的 `font-size` 属性，如果没有父元素，则参考浏览器的默认值字号。

2. 绝对类型

绝对类型所使用的单位不会随着显示设备的不同而改变，也就是当属性值使用绝对单位时，不论在哪种设备上显示都是一样的，例如屏幕上的 1cm 与打印机上的 1cm 是一样长的。表 4-3 为浏览器支持的绝对类型的长度单位。

表 4-3 常用长度单位

| 类型 | 长度单位 | 说明 |
|------|------|--|
| 绝对类型 | in | 英寸 Inches (1 英寸 = 2.54 厘米) |
| | cm | 厘米 Centimeters |
| | mm | 毫米 Millimeters |
| | pt | 点 Points (1 点 = 1/72 英寸) |
| | pc | 皮卡 Picas (1 皮卡 = 12 点) |
| 相对类型 | em | 元素的字体高度 The height of the element's font |
| | ex | 字母 x 的高度 The height of the letter " x" |
| | px | 像素 Pixels |
| | % | 百分比 Percentage |

百分比单位是一种常用的相对类型，通常的参考依据为 `font-size` 属性。百分比值总是相对于另一个值。下面的语句将 `margin` 属性的值设置为 `font-size` 的 200%：

```
p{margin:200%;}
```

需要注意的是，不管使用哪种单位，在设置的时候数值与单位之间不需要加空格。



4.4.2 色彩单位

HTML 网页和 CSS 样式都是按照 RGB 来设置颜色的。在 HTML 标记中只提供了两种设置色彩的方法：十六进制数和色彩英文名称。CSS 则提供了三种定义色彩的方法：十六进制数、色彩英文名称、rgb 函数和 rgba 函数。

1. 用十六进制数方式表示色彩值

在计算机中，定义每种色彩的强度范围为 0~255。当所有色彩的强度都为 0 时，将产生黑色；当所有色彩的强度都为 255 时，将产生白色。

在 HTML 中，使用 RGB 概念指定色彩时，前面是一个“#”号，再加上 6 个十六进制数字表示，表示方法为：#RRGGBB，其中，前两个数字代表红光强度（Red），中间两个数字代表绿光强度（Green），后两个数字代表蓝光强度（Blue），以上 3 个参数的取值范围为 00~ff。参数必须是两位数，对于只有 1 位的参数，应在前面补 0。这种方法共可表示 $256 \times 256 \times 256$ 种色彩，即 16M 种色彩。而红色、绿色、黑色、白色的十六进制设置值分别为 #FF0000、#00FF00、#0000FF、#000000、#FFFFFF。例如下面的代码：

```
div{
    color:#FF0000;
}
```

如果每个参数各自在两位上的数字都相同，也可缩写为 #RGB 的方式。例如 #cc9900 可以缩写为 #c90。

2. 用色彩名称方式表示色彩值

在 CSS 中也提供了与 HTML 一样的用色彩英文名称表示色彩的方式。CSS 只提供 16 种色彩名称，见表 4-4。

表 4-4 CSS 规范推荐的颜色名称

| 名称 | 颜色 | 名称 | 颜色 | 名称 | 颜色 |
|-------|----|--------|----|---------|----|
| black | 纯黑 | silver | 浅灰 | navy | 深蓝 |
| blue | 浅蓝 | green | 深绿 | lime | 浅绿 |
| teal | 靛青 | aqua | 天蓝 | maroon | 深红 |
| red | 大红 | purple | 深紫 | fuchsia | 品红 |
| olive | 褐黄 | yellow | 明黄 | gray | 深灰 |
| white | 壳白 | | | | |

3. 使用 RGB (x, y, z) 函数表示

x、y、z、分别是红色、绿色、蓝色的值， $x, y, z \in [0, 225]$ ，亦可用百分比表示；对比 rgba (x, y, z, a) 中 a 是 alpha 通道设置透明度取值 0~1。

4.5 CSS 继承与层叠

CSS 继承即子标记会继承父标记的所有样式风格，并且可以在父标记样式风格的基础上再加以修改，产生新的样式，而子标记的样式风格完全不影响父标记，值得注意的是，并不是所有的属性都会自动传给子元素，有的属性不会继承父标记的属性值，例如边框属性就是非继承的。

【例 4.7】 CSS 继承。

```
<! --示例 4.7 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" >
    <title>继承关系演示</title>
    <style>
      h1 {
        color:blue;                /* 颜色 */
        text-decoration:underline; /* 下划线 */
      }
      em {
        color:red;                 /* 颜色 */
      }
    </style>
  </head>
  <body>
    <h1>学习<em>Web 开发</em>教程</h1>
    <p>如果您有任何问题,欢迎联系我们</p>
  </body>
</html>
```

本例浏览效果如图 4-7 所示。

以上示例中，<h1>标签设置为蓝色，加下划线。但标签设置为斜体，红色。它继承了父标签的下划线。

CSS 的全称是“层叠样式表”，层叠特性和“继承”不一样，可以把层叠特性理解成“冲突”的解决方案，即对同一内容设置了多个不同类型样式产生冲突时的处理，CSS 规定如下优先级为：行内样式 > id 样式 > class 样式 > 标记样式。



图 4-7 继承关系演示

【例 4.8】 CSS 层叠。

```
<! --示例 4.8 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>CSS 层叠</title>
    <style type="text/css">
      p {
        color: green;
      }
      . red {
        color: red;
      }
      . purple {
        color: purple;
      }
      #line3 {
        color: blue;
      }
    </style>
  </head>
  <body>
    <p >这是第 1 行文本</p>
```



```
<p class="red">这是第 2 行文本</p>  
<p id="line3" class="red">这是第 3 行文本</p>  
<p style="color:orange;" id="line3">这是第 4 行文本</p>  
<p class="purple red">这是第 5 行文本</p>  
</body>  
</html>
```

本例浏览效果如图 4-8 所示。



图 4-8 CSS 层叠

样式表的层叠性就是继承性，样式表的继承规则是：外部的元素样式会保留下来，由各元素所包含的其他元素继承；所有在元素中嵌套的元素都会继承外层元素指定的属性，有时会把多层底套的样式叠加在一起，除非进行更改；遇到冲突的地方，以最后定义为准。

4.6 元素类型

在前面已经以文档结构树形图的形式讲解了文档中元素的层次关系，这种层次关系同时也要依赖于这些元素类型间的关系。CSS 使用 `display` 属性规定元素应该生成的框的类型，任何元素都可以通过 `display` 属性改变默认的显示类型。

1. 块级元素 (`display: block`)

`display` 属性设置为 `block` 将显示块级元素，块级元素的宽度为 100%，而且后面隐藏附带有换行符，使块级元素始终占据一行，如 `<div>` 常常被称为块级元素，这意味着这些元素显示为一块内容，标题、段落、列表、表格、分区 `div` 和 `body` 等元素都是块级元素。



2. 行级元素 (display: inline)

行级元素也称内联元素, display 属性设置为 inline 将显示行级元素, 元素前后没有换行符, 行级元素没有高度和宽度, 因此也就没有固定的形状, 显示时只占据其内容的大小超链接、图像、范围 (span)、表单元素等都是行级元素。

3. 列表项元素 (display: list-item)

list-item 属性值表示列表项目, 其实质上也是块状显示, 是一种特殊的块状类型, 它加了缩进和项目符号。

4. 隐藏元素 (display: none)

none 属性值表示隐藏并取消盒模型, 所包含的内容不会被浏览器解析和显示。通过把 display 设置为 none, 该元素及其所有内容就不再显示, 也不占用文档中的空间。

综合案例: 开平碉楼申遗大事记。

以开平碉楼的申遗大事记为例子, 参照给定的 HTML 代码、CSS 设置和图片资源, 利用 HBuilder 设计 Web 网页。

要求:

- (1) 标题设置为 h2、“微软雅黑”、居中显示;
- (2) 小标题设置为“黑体”, 字体颜色设置为“#8B4513”, 段落背景颜色为“#D3B175”, 行高为 200%;
- (3) 正文其他所有的字体设置为“微软雅黑”、18px、首行缩进 2 字符;
- (4) 使用外部样式表链接到 HTML 文档中。

```
<!--综合实例 HTML 页面参考代码-->
```

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

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

```
    <title>开平碉楼</title>
```

```
    <link type="text/css" rel="stylesheet" href="css/z4.css"/>
```

```
  </head>
```

```
  <body>
```

```
    <div id="content">
```

```
      <h2 class="biaoti">开平碉楼申遗大事记</h2>
```

```
      <hr color="#D3B175" size="5"/>
```

```
      
```

```
      <p class="tx1">2000 年, 开平成立申遗小组</p>
```

```
      <p>2000 年 2 月, 从国务院信访局到开平挂市委常委职务的李玫同志在香港侨团团拜会上向胡耀坤、吴荣治等侨领首次提出将开平碉楼申报为世界文化遗产。
```

```
      2000 年 10 月 30 日, 开平市成立"开平碉楼"申报世界文化遗产领导小组。领导小组下设办公室, 分整治组、资料组、宣传组。
```

```
    </p>
```

<p class="tx1">2001年,省申遗领导小组成立</p>

<p>2001年4月30日,开平市市长办公会议研究将开平碉楼申遗项目名称定为"开平碉楼与民居"。

2001年6月19日,申遗办谭伟强、张健文、邝积康3位负责人赴北京,在李玫同志的陪同下拜访了中国国际古迹遗址理事会秘书长郭旂、清华大学教授陈志华,就申遗的有关事宜进行会谈。

2001年3月至6月,开平市开展"开平碉楼"的普查工作,并请上海同济大学、华南理工大学专家对33座碉楼测绘了立面图、剖面图、平面图和细部特征图。

2001年6月,江门市成立了以时任市长雷于蓝为组长的江门市申报世界文化遗产领导小组。

2001年7月11日至17日,清华大学陈志华、楼庆西、李秋香教授,五邑大学张国雄、梅伟强教授,深入到三埠、长沙、沙冈、塘口、蚬冈、百合、月山、赤坎、大沙等镇(办事处)对碉楼与民居进行了认真细致的考察。开平市政府特聘几位专家为"开平碉楼与民居"申报世界文化遗产工作高级顾问。

2001年9月6日至7日,时任副省长的李兰芳到江门市视察碉楼,对开平碉楼申遗工作给予了充分的肯定,并表示省政府对此将大力支持。

2001年9月12日,联合国世界遗产评估机构——国际古迹遗址理事会总协调员亨利博士考察开平碉楼,对开平碉楼的价值和开平乃至广东的文物保护工作表示了充分的肯定。

2001年11月,广东省成立申报世界文化遗产领导小组及其办公室。

2001年,开平籍旅港同胞、恒生银行董事长利国伟爵士就开平碉楼申报事宜专门写信给时任国务院总理的朱镕基。

</p>

<p class="tx1">2004年,世遗中心受理开平申遗</p>

<p>2004年4月10日至11日,联合国世界遗产专家乔拉·索拉先生及夫人、以色列文物专家阿里·拉哈米莫夫先生及夫人,中国国际古迹遗址理事会秘书长郭旂先生,澳门文化局文化财产厅陈泽成厅长的陪同下到开平考察碉楼。

2004年5月,任仲夷同志致函省委书记张德江,请德江同志向中央汇报开平碉楼申报世界文化遗产工作。

2004年5月17日,张德江同志在开平市委、市政府《关于请张德江书记向中央汇报开平碉楼申报世界遗产工作的请示》中批示:我赞成和支持开平碉楼申报世界文化遗产。

2004年6月,开平市被中国建筑学会授予"中国碉楼之乡"的称号。

2004年7月2日,中国100多个申遗"预备清单"中已有5个申报项目被世界遗产中心受理,分别是澳门历史建筑群、开平碉楼、殷墟、福建土楼和红河哈尼梯田。

2004年10月28日,蚬冈镇锦江里昇峰楼业主旅加华侨胡黄惠英(乳名:黄滚盛)女士及其兄长黄雄畅先生在梁金山世外桃源酒店正式将他们的碉楼无偿委托给开平人民政府管理。

</p>

<p class="tx1">2005年,申报名称变更</p>

<p>2005年4月21日,由清华大学建筑学院与开平市申遗办合作开展的第二次全市性碉楼普查工作于圆满完成。本次碉楼普查从2005年3月16日开始,历时13个月。第二次碉楼普查深入到全市每条村庄逐幢碉楼进行登记、拍照、调查核实,拍摄2万多张图片,采集到一些珍贵的、鲜为人知的历史资料。

2005年9月18日,联合国世界遗产协调员乔拉先生、建筑规划师阿里先生来到开平,指导开平碉楼申报世界文化遗产工作。



2005年9月24日,由省人民政府主办,省文化厅、江门市人民政府、开平市人民政府承办的广东省开平碉楼申报世界文化遗产动员大会在省博物馆门前隆重举行。

2005年11月,开平碉楼申报世界文化遗产的项目名称由"开平碉楼与民居"变为"开平碉楼与村落",极大地延伸了开平碉楼的文化内涵和历史价值,更有利于申报世界文化遗产。

2005年12月5日,北京大学世界遗产中心沈文权博士,中国古文物保护专家张志平女士(郭旂夫人)抵达开平,着手开展修改北大制订的开平碉楼文化遗产保护规划,为适应申报的需要,该规划将改名为《开平碉楼与村落保护管理规划》,并对相关内容进行增补。

</p>

<p class="tx1">2006年,世遗专家评估开平申遗</p>

<p>2006年1月11日,国家文物局局长单霁翔正式签署英文申报文本。

2006年1月,国务院正式批准"开平碉楼与村落"作为2007年代表中国向联合国申报世界文化遗产的项目,英文申报文本已被联合国教科文组织确认合格接收。

2006年3月2日至3日,联合国教科文组织专家乔拉·索拉先生和阿里先生按原计划来到开平考察"申遗"准备工作。

2006年4月3日下午,国家文物局、省政府、江门和开平市委、市政府在开平市行政大楼召开了开平碉楼与村落申报世界文化遗产工作会议,研究申遗工作进入第二个重要环节——环境整治阶段存在的问题,提出下一步工作意见。

2006年5月9日,黄华华省长在江门市委书记陈继兴、市长王南健的陪同下到江门市检查指导开平碉楼与村落申遗工作。

2006年8月21日至22日,经业主方其祥、方其赏、方其锦授权,塘口镇自力村三座尘封了大半世纪的居庐——叶生居庐、官生居庐、澜生居庐被依次打开,并发现大量珍贵文物。

2006年8月27日至28日,著名歌唱家阎维文与中央电视台摄制组一行10多人来到开平,拍摄开平碉楼与村落申报世界文化遗产主题曲——《碉楼颂》音乐电视。

2006年9月,中国华侨出版社出版了两本关于开平碉楼与村落的学术著作,总结开平碉楼与村落研究所取得的丰硕成果,支持开平碉楼与村落申报世界文化遗产。

2006年9月15日,开平市举行"共同的心愿——全力支持开平碉楼与村落申报世界文化遗产"为主题的申遗文艺晚会,世遗评估专家卢光裕先生、副省长雷于蓝、国家文物局有关领导观看了晚会。晚会结束时,共收到来自社会各界的捐款1480多万元,其中,著名侨领吴荣治先生及夫人捐款100万元。

2006年9月15至18日,受联合国教科文组织世界遗产中心的委派,"世遗"评估专家卢光裕先生到江门市对开平碉楼与村落进行评估。

</p>

<p class="tx1">2007年,申遗成功</p>

<p>2007年1月,开平碉楼与村落申遗项目顺利通过国际古迹遗址(ICOMOS)委员会执委会在巴黎召开的专家会议的评审。

2007年6月20日,开平市赴新西兰、澳大利亚申遗代表团一行30多人经香港转机后抵达新西兰。

2007年6月22日,中国政府代表团飞赴新西兰,出席世界遗产委员会第31届大会,为开平碉楼与村落申遗助威。

2007年6月底,开平碉楼与村落申遗项目顺利通过在新西兰召开的第31届世界遗产大会的表决,正式列入《世界遗产名录》。

```
</p>
    <p id="t1">来源:开平碉楼</p>
</div>
</body>
</html>
```

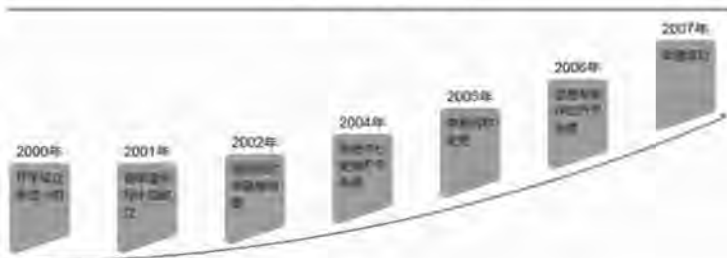
本案例采用链接外部样式表来格式化网页, CSS 文件参考代码如下:

```
/* -----CSS 参考代码----- */
/* -----全局声明----- */
* {
    width:1024px;
    margin-left:auto;
    margin-right:auto;
}
/* -----正文标题格式----- */
.biaoti {
    font-family:"微软雅黑";
    text-align:center;
}
/* -----正文所有内容样式----- */
#content {
    text-indent:2em;
    font-family:"微软雅黑";
    font-size:18px;
}
/* -----小标题样式----- */
.tx1 {
    font-weight:bold;
    font-family:"黑体";
    font-size:24px;
    color:#8B4513;
    background-color:#D3B175;
    line-height:200%; /* 设置行高 */
}
/* -----结尾样式----- */
#t1 {
    color:#333;
    text-align:right;
}
```

本例浏览效果如图 4-9 所示。



开平碉楼申遗大事记



2000年，开平成立申遗小组

2000年2月，从国务院信访局到开平挂市委常委职务的李牧同志在香港侨团拜会上向胡耀坤、吴荣治等侨领首次提出将开平碉楼申报为世界文化遗产。2000年10月30日，开平市成立“开平碉楼”申报世界文化遗产领导小组。领导小组下设办公室，分规划组、资料组、宣传组。

2001年，国家立项并申报立项

2001年4月30日，开平市委书记会议研究将开平碉楼申报项目名称定为“开平碉楼与民居”。2001年6月19日，申遗办张耀文、卢耀文、卢耀文3位负责人赴北京，在李牧同志的陪同下拜访了中国国际古迹遗址理事会秘书长郭彤。清华大学教授陈志华、张中迪的有关事宜进行会谈。2001年3月至6月，开平市开展“开平碉楼”的普查工作，并请上海同济大学、华南理工大学专家对33座碉楼进行了立面图、剖面图、平面图和细部的特征图。2001年6月，江门市成立了以时任市长潘永强为组长的江门市申报世界文化遗产领导小组。2001年7月11日至17日，清华大学陈志华、杨庆堃、李秋香教授，五邑大学张国维、梅伟强教授，深入到大塘、长沙、沙冈、墟口、蛟冈、百合、月山、赤坎、大沙等镇（办事处）对碉楼与民居进行了认真细致的考察。开平市政府特聘几位专家为“开平碉楼与民居”申报世界文化遗产工作高级顾问。2001年9月6日至7日，时任副省长的李兰芳到江门市视察碉楼，对开平碉楼申遗工作给予了充分的肯定，并表示省政府对此将大力支持。2001年9月12日，联合国世界遗产评估机构——国际古迹遗址理事会总协调员亨利博士考察开平碉楼，对开平碉楼的价值和开平乃至广东的文物保护工作表示了充分的肯定。2001年11月，广东省成立申报世界文化遗产领导小组及其办公室。2001年，开平面临港澳问题，恒生银行董事长利国伟爵士就开平碉楼申报事宜专门写信给时任国务院总理的朱鎔鼎。

2004年，世遗中心受理开平申请

2004年4月10日至11日，联合国世界遗产专家萨拉·索拉先生及夫人、以色列文物专家阿耶·拉哈米先生及夫人，中国国际古迹遗址理事会秘书长郭彤、澳门文化局文化财产厅迹成厅长的陪同下到开平考察碉楼。2004年5月，任仲夷同志致函省委书记张德江，请德江同志向中央汇报开平碉楼申报世界文化遗产工作。2004年5月17日，张德江同志在开平市委、市政府《关于请张德江书记向中央汇报开平碉楼申报世界文化遗产请示》中批示：我赞成和支持开平碉楼申报世界文化遗产。2004年6月，开平市政府向中国世界遗产中心授予“中国碉楼之乡”的称号。2004年7月2日，中国100多个“申遗”预备清单中已有5个申报项目被世界遗产中心受理，分别是澳门历史建筑群、开平碉楼、福建、福建土楼和红河哈尼梯田。2004年10月28日，鹤洞镇鹤江里黄峰峰业主施加华侨胡惠英（乳名：黄琼英）女士及其兄长黄峰翰先生在梁金山世外豪园酒店正式将他们的碉楼无偿委托给开平人民政府管理。

2005年，开展大规模普查

2005年4月21日，由清华大学建筑学院与开平市申遗办合作开展的第二次全市性碉楼普查工作于圆满完成。本次碉楼普查从2005年3月16日开始，历时13个月。第二次碉楼普查深入到全市每条村庄逐幢碉楼进行登记、拍照、调查核实，拍摄2万多张照片，采集到一批珍贵的、鲜为人知的历史资料。2005年9月18日，联合国世界遗产协调员克拉先生、建筑规划师阿德里先生来到开平，指导开平碉楼申报世界文化遗产工作。2005年9月24日，由省人民政府主办，省文化厅、江门市人民政府、开平市人民政府承办的广东省开平碉楼申报世界文化遗产动员大会在开平博物馆门前隆重举行。2005年11月，开平碉楼申报世界文化遗产的项目名称由“开平碉楼与民居”变为“开平碉楼与村落”，极大地延伸了开平碉楼的文化内涵和历史价值，更有利于申报世界文化遗产。2005年12月5日，北京大学世界遗产中心沈文敏博士，中国古文物保护专家张立志女士（郭彤夫人）抵达开平，着手开展修改开平碉楼文化遗产保护规划，为适应申报的需要，该规划将改名为《开平碉楼与村落保护管理规划》，并对相关内容进行修订。

2006年，制定专项申请开平碉楼

2006年1月11日，国家文物局局长单霁翔正式签署英文申报文本。2006年1月，国务院正式批准“开平碉楼与村落”作为2007年代表中国向联合国申报世界文化遗产的项目，英文申报文本已被联合国教科文组织确认合格接收。2006年3月2日至3日，联合国教科文组织专家萨拉·索拉先生和阿里先生按照计划来到开平考察“申遗”筹备工作。2006年4月3日下午，国家文物局、省政府、江门市和开平市委、市政府在开平市行政大楼召开了开平碉楼与村落申报世界文化遗产工作会议，研究申遗工作进入第二个重要环节——环境整治前存在的问题，提出下一步工作意见。2006年5月9日，黄华华省长在江门市委书记陈炳兴、市长王南雄的陪同下到江门市检查指导开平碉楼与村落申遗工作。2006年8月21日至22日，经业主方推荐、方其赞、方其耀授权，墟口镇自力村三座尘封了大半个世纪的碉楼——叶生居、官生居、潘生居被依次打开，并发现大量珍贵文物。2006年8月27日至28日，著名歌唱家阎维文与中央电视台摄制组一行10多人来到开平，拍摄开平碉楼与村落申报世界文化遗产主题曲——《碉楼颂》音乐电视。2006年9月，中国华侨出版社出版了两本关于开平碉楼与村落的学术著作，总结开平碉楼与村落研究所取得的丰硕成果，支持开平碉楼与村落申报世界文化遗产。2006年9月15日，开平市举行“共同的心愿——全力支持开平碉楼与村落申报世界文化遗产”为主题的申遗文艺晚会，世遗评估专家卢光裕先生、副馆长潘于、国家文物局有关领导观看了晚会。晚会结束时，共收到来自社会各界的捐款1480多万元，其中，著名侨领吴荣治先生及夫人捐款100万元。2006年9月15日至18日，受联合国教科文组织世界遗产中心的委派，“世遗”评估专家卢光裕先生到江门市对开平碉楼与村落进行评估。

2007年，申遗成功

2007年1月，开平碉楼与村落申报项目顺利通过国际古迹遗址（ICOMOS）委员会执委会在巴黎召开的专家会议的评审。2007年6月20日，开平市委书记李新、澳大利亚申遗代表团一行30多人经香港转机后抵达新西兰。2007年6月22日，中国政府代表团赴新西兰，出席世界遗产委员会第31届大会，为开平碉楼与村落申遗助威。2007年6月底，开平碉楼与村落申报项目顺利通过在新西兰召开的第31届世界遗产大会的表决，正式列入《世界遗产名录》。

来源：开平碉楼

图 4-9 《开平碉楼申遗大事记》网页效果

本章习题

1. 选择题。

(1) id 选择器的特点是 ()。

- A. 唯一性 B. 普遍性 C. 可以重复应用 D. id 选择器

(2) 在 HTML 文档中, 导入外部样式表的方法是 ()。

- A. <link> B. <url> C. src D. @important

(3) 以下代码正确的有 ()。

- A. <p style=font-size: bold; ></p> B. <p id=" id1 id2" ><p>
C. <p class=" class1 class2" ></p> D. <p id=" id", class=" class1" >

(4) CSS 的继承并不是完全的“克隆”, 有些继承是不起作用的, 在下面几个属性的继承是不起作用的 ()。

- A. border B. margin C. font D. padding

(5) 在网页中最常用的单位是 ()。

- A. in B. cm C. pc D. px

(6) 以下哪个伪类选择器用来表示鼠标按下的样式 ()?

- A. link B. active C. hover D. visited

2. 使用 HBuilder 编写符合以下要求的文档: 在浏览器窗口中显示超链接文字“WELCOME”。完成效果如图 4-10 所示。其中上边 div 的背景颜色为#FF99CC, 文字居中显示; 水平分割线粗细为 5, 颜色为#FF99CC; 超链接文字的字体为“arial black”, 字号为 60px, 颜色为#FFF, 无下划线; 当鼠标滑过文字变为#FF00FF; 当鼠标按下文字变为#FF00CC; 访问过后连接文字变为#FFFFCC。脚部文字颜色为#666。CSS 样式采用内部样式表。



图 4-10 网页效果

3. 使用 HBuilder 编写符合以下要求的文档: 使用所给素材文件夹中的 hwbanner.png 和

文字，完成效果如图 4-11 所示。网页采用链接外部样式表的方式。内容部分的 div 宽度设置为居中（margin-right: auto; margin-left: auto;）。根据网页效果，使用 h1, h2, 设置标题。“为客户创造价值”等语句，使用内联样式，加粗，加大。



图 4-11 网页效果

第 5 章 CSS 盒模型



学习目标

1. 理解 CSS 盒模型的定义；
2. 掌握 CSS 元素高度、宽度、定位和浮动的设置方法；
3. 利用内边距、外边距和边框实现内容的适应性。



知识要点

1. 盒模型的概述；
2. 元素高度和宽度的设置；
3. 元素的边距和边框设置；
4. 元素的定位和浮动。

盒模型是 CSS 定位布局的核心内容，页面中所有的元素都是放进一个容器内的，这个容器可以看成是一个盒子。可以通过 CSS 来控制这些盒子的各种显示属性，把这些盒子进行定位，完成整个页面的布局。在掌握了盒子模型以及其中每个元素的用法后，才能拥有较完善的布局观。

5.1 盒模型的定义

Web 页面上大部分的元素（特别是块状元素）都可以看作是一个盒子，W3C 组织建议把所有网页上的对象都放在一个盒子（box）中，设计者可以通过创建定义来控制这个盒子的各种属性，这些对象包括段落、列表、标题、图片及层。

盒子的结构可以看作一个矩形框，包括边框（border）、外边距（margin）、内边距（padding），以及元素内容（content），元素会在指定的高度和宽度范围内在页面上显示出来。它们之间的关系，如图 5-1 所示。

一个页面由许多这样的盒子组成，这些盒子层层嵌套，互相影响，设计者在布局网页和定位 Web 元素时要充分考虑到这些要素，才能更自如地摆弄这些盒子。

而网页元素的定位实际上就是这些大大小小的盒子在页面中的定位，这些盒子在页面中是“浮动”的，当某个块状元素被 CSS 设置了浮动属性，这个盒子就会移动到上一块级元素所处的行中。要关注的是这些盒子在页面中如何摆放、如何嵌套的问题，而这么多盒子摆

在一起，其中盒子尺寸计算、是否浮动等要素是最需要关注的。

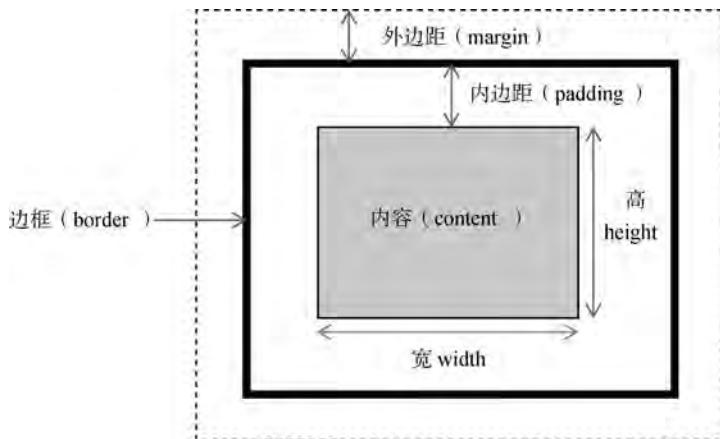


图 5-1 盒模型示意图

默认情况下盒子的边框是“无”，背景色是透明的，所以在默认情况下看不到盒子。内边距、边框和外边距这些属性都是可选的，默认值都是 0。边框的作用就是在内边距和外边距之间创建一个隔离带，以避免视觉上的混淆。

5.2 CSS 元素的高度和宽度

当设计者布局一个网页时，网页最终的宽度和高度会超出预计的数值，这是因为盒模型的宽度或高度计算误差造成的。指定一个 CSS 元素的宽度和高度属性时，只是设置内容区域的宽度和高度。而浏览器实际大小的元素，还必须添加内边距、外边距和边框。而增加或减少内边距、外边距和边框，不会影响内容区域的尺寸，只会增加盒模型的总尺寸。

【例 5.1】 元素 CSS 的宽度是 100 像素。

```
<!-- 示例 5.1 程序 -->
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 5.1</title>
    <style type="text/css">
      * {
        margin:0;
        padding:0;
      }
      div {
        width:100px;
        border:10px solid red;
```

```

padding:30px;
margin:10px;
float:left;
}
</style>
</head>
<body>
<h1>盒模型示例</h1>
<p>CSS 盒模型本质上是一个盒子,外边距 10px;内边距 30px;边框 10px 的红色实线;盒子左浮动在同一行处。</p>
<div><img src = "img/float1. jpg" /></div>
<div><img src = "img/float2. jpg" /></div>
</body>
</html>

```

本例浏览效果如图 5-2 所示。



图 5-2 元素尺寸设置

1. 盒模型的宽度

盒模型的宽度 = 左外边距 (margin-left) + 左边框 (border-left) + 左内边距 (padding-left) + 内容宽度 (width) + 右内边距 (padding-right) + 右边框 (border-right) + 右外边距 (margin-right)。

2. 盒模型的高度

盒模型的高度 = 上外边距 (margin-top) + 上边框 (border-top) + 上内边距 (padding-top) + 内容高度 (height) + 下内边距 (padding-bottom) + 下边框 (border-bottom) + 下外边距 (margin-bottom)。

所以，图 5-2 中的图片内容是 100 像素，而浏览器显示的页面实际大小是 200 像素。默认情况下，块级元素可以设置宽度和高度，但行级元素是不能设置的。

【例 5.2】 块级元素和行级元素的宽度与高度。

```
<! --示例 5.2 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 5.2</title>
    <style type="text/css">
      .ap {
        width:200px;
        height:100px;
        border:2px solid red;
        margin:10px;
        background-color:lightblue;
      }
    </style>
  </head>
  <body>
    <div class="ap">这是 div 块级元素,可设置宽度和高度</div>
    <span class="ap">这是 span 行级元素,不能设置宽度和高度</span>
  </body>
</html>
```

本例浏览效果如图 5-3 所示。

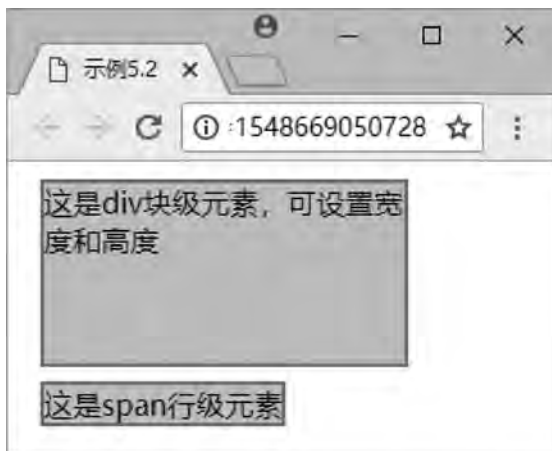


图 5-3 行级元素不可设置宽度和高度

如要为行级元素设置宽度和高度，需要为元素设置 display 属性，如在示例程序例 5.2 中加入属性 “display: block;”，浏览效果如图 5-4 所示。代码如下：

```
<style type="text/css">
```

```
.ap{  
    width:200px;  
    height:100px;  
    border:2px solid red;  
    margin:10px;  
    background-color:#00FF00;  
    display:block;                /* 块级显示 */  
}  
</style>
```



图 5-4 行级元素可以设置宽度和高度

5.3 边距设置和边框设置

margin-border-padding 模型是最常用的盒子布局形式。对于任何一个盒子，都可以分别通过设置四条边各自的外边距（margin）、边框（border）和内边距（padding），实现各种各样的排版效果，而且它们各自的四条边在多参数同时设置时，均按照上→右→下→左的顺序（顺时针）。

5.3.1 外边距设置

外边距是指元素与元素之间的距离，外边距设置属性，可分别设置 margin-top、margin-right、margin-bottom、margin-left，也可以用 margin 属性一次性设置所有外边距。

1. 上外边距

语法：margin-top: length | percent | auto


```
#small2 {
    margin-right:20px;
    margin-top:10px;
    float:right;
}
#small3 {
    margin-bottom:5px;
}
#small4 {
    margin-left:10px;
    margin-top:15px;
}
</style>
</head>
<body>
    <div id="big">
    <div id="small1">A:左外边距 20 像素,下外边距 30 像素</div>
    <div id="small2">B:右外边距 20 像素,上外边距 10 像素,右浮动</div>
    <div id="small3">C:下外边距 5 像素</div>
    <div id="small4">D:左外边距 10 像素,上外边距 15 像素</div>
    </div>
</body>
</html>
```

本例浏览效果如图 5-5 所示。

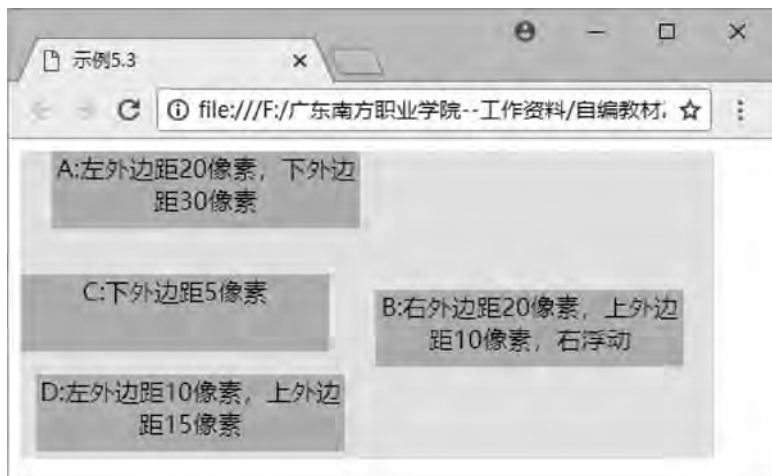


图 5-5 各种外边距的设置

以上示例中共设置了 1+4 个盒子，由于 B 盒子设置为向右浮动，因此紧随其后的 C 盒子自然移动上来，和 B 盒子同处一行，A 盒子设置为“margin-bottom: 30px;”，而 C 盒子没有设置 margin-top，故 A 盒子与 C 盒子之间的距离为 30 像素，B 盒子设置为“margin-top: 10px;”，所以 A 盒子与 B 盒子之间的距离为 30+10=40 像素。



5. 外边距

语法: margin: length | percent | auto

参数: length 包括长度值和长度单位, 包括绝对单位和相对单位。percent 是基于父对象的高度, 左右外边距允许使用负数。auto 值为自动提取边距值, 是默认值。

说明: 设置对象四边的外边距, 包括 margin-top (上外边距)、margin-right (右外边距)、margin-bottom (下外边距)、margin-left (左外边距), 外边距始终是透明的。

如果只提供 1 个参数, 将应用于全部的外边距。

如果提供 2 个参数, 第 1 个参数应用于上、下外边距, 第 2 个参数应用于左、右外边距。

如果提供 3 个参数, 第 1 个参数应用于上外边距, 第 2 个参数应用于左、右外边距, 第 3 个参数应用于下外边距。

【例 5.4】 外边距总设置。

```
<! -- 示例 5.4 程序 -->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 5.4</title>
    <style type="text/css">
      div {
        border:solid #0000FF 1px;
        width:auto;
        margin:5px;
        float:left;
      }
      .margin1 {
        background-color:#9F6;
        border:none; /* 无边框 */
        width:200px;
        height:80px;
        margin:0px 15px 5px 30px;
      }
      .margin2 {
        background-color:yellow;
        border:none;
        width:200px;
        height:80px;
        margin:5px 30px 15px;
      }
      .margin3 {
        background-color:lightgreen;
        border:none;
```



```
width:200px;
height:80px;
margin:10px 30px;
}
.margin4{
background-color:#FC0;
border:none;
width:200px;
height:80px;
margin:15px;
}
</style>
</head>
<body>
<div>
<div class="margin1">上、右、下、左外边距分别为:0px、15px、5px、30px</div>
</div>
<div>
<div class="margin2">上外边距为:5px,下外边距为:15px,左右外边距为:30px</div>
</div>
<div>
<div class="margin3">上、下外边距为10px,左、右外边距为:30px</div>
</div>
<div>
<div class="margin4">上、下、左、右外边距均为:15px</div>
</div>
</body>
</html>
```

本例浏览效果如图 5-6 所示。

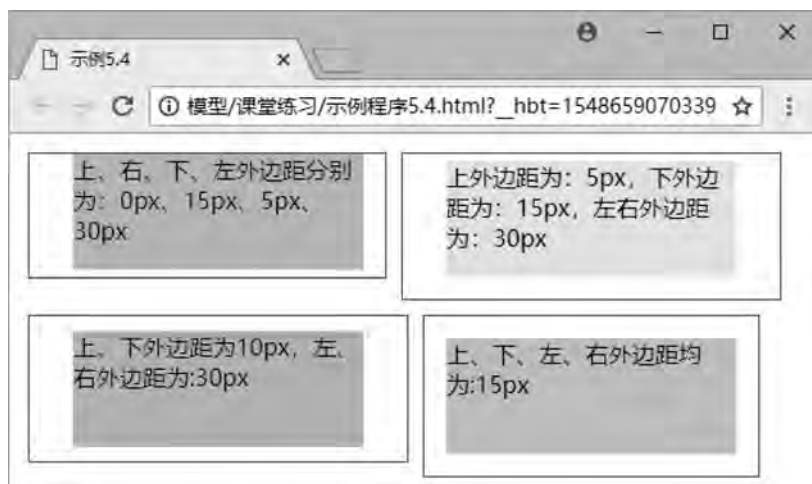


图 5-6 外边距总设置



5.3.2 外边距的合并

通常，盒子与盒子之间的外边距相遇会互相影响，必须对 margin 属性深入了解，才能精确地控制盒子的位置。

1. 行级元素外边距合并

行级元素的盒子相遇，盒子与盒子之间的距离等于两个盒子外边距的总和。

【例 5.5】 行级元素之间的外边距设置。

```
<! --示例 5.5 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 5.5</title>
    <style type="text/css">
      * {
        margin:50px;
      }
      .hb1 {
        background-color:yellow;
        margin-right:20px;
        padding:30px;
      }
      .hb2 {
        background-color:lightpink;
        margin-left:30px;
        padding:30px;
      }
    </style>
  </head>
  <body>
    <span class="hb1">黄色 span</span><span class="hb2">粉红色 span</span>
  </body>
</html>
```

本例浏览效果如图 5-7 所示。



图 5-7 行级元素盒子相遇

2. 块级元素外边距合并

块级元素的盒子相遇，盒子与盒子之间的距离等于两盒子中外边距的较大者。

【例 5.6】垂直相遇块级元素之间的外边距设置。

```
<! --示例 5.6 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 5.6</title>
    <style type="text/css">
      * {
        margin:50px;
      }
      .div1 {
        background-color:yellow;
        margin-bottom:30px;
        padding:30px;
      }
      .div2 {
        background-color:lightpink;
        margin-top:30px;
        padding:30px;
      }
    </style>
  </head>
  <body>
```

```
<div class="div1">黄色 div</div>
<div class="div2">粉红色 div</div>
</body>
</html>
```

本例浏览效果如图 5-8 所示。



图 5-8 垂直块级元素盒子相遇

5.3.3 内边距设置

元素的内边距用来控制边框和内容区之间的空白距离，并非实体，用 `padding` 属性表示，类似于 HTML 中表格单元格的填充属性。内边距（`padding`）和外边距（`margin`）很相似，都是透明不可见的，只是内边距和外边距之间还有边框。从语法和用法上来说，内边距的属性与外边距的属性也是类似的，既可以使用复合属性，也可以使用单边属性，`padding` 属性接受 `length` 值或 `percent` 值，区别于外边距，内边距不可以使用负值。

【例 5.7】 内边距的设置。

```
<!-- 示例 5.7 程序 -->
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 5.7</title>
    <style type="text/css">
      span {
        background-color:#FFFF99;
      }
      div {
        border:solid #000000 1px;
```

```
width:auto;
height:auto;
margin:15px;
float:left;
}
.padding1{
padding-top:30px;
padding-left:15px;
}
.padding2{
padding-bottom:30px;
padding-right:30px;
}
.padding3{
padding:5px 30px;
}
.padding4{
padding:20px;
}
</style>
</head>
<body>
<div class="padding1">
<span>文字元素的上内边距为 30px,左内边距为 15px</span>
</div>
<div class="padding2">
<span>文字元素的下内边距为 30px,右内边距为 30px</span>
</div>
<div class="padding3">
<span>文字元素的上、下内边距为 5px,左、右内边距为:30px</span>
</div>
<div class="padding4">
<span>文字元素的上、右、下、左内边距均为:20px</span>
</div>
</body>
</html>
```

本例浏览效果如图 5-9 所示。



图 5-9 各种内边距的设置

5.3.4 边框设置

元素外边距内就是元素的边框 (border)，它是围绕内边距和元素内容的一条或多条线，在内边距和外边距之间。边框的四条边分别用 border-top、border-right、border-bottom、border-left 表示，它们的属性与内外边距的属性也是类似的，既可以使用复合属性，也可以使用单边属性。

边框作为盒模型的某个组成部分，边框的 CSS 样式设置将直接影响到盒子的尺寸和外观。而通过使用 border 属性，可以创建出最佳的边框效果，还可以应用于任何元素。border 属性设置通常有 3 种：样式 (border-style)、宽度 (border-width) 和颜色 (border-color)。

1. 上边框

语法：border-top: border-style | border-width | border-color

参数：该属性是复合属性。需要通过参数设置来实现。

2. 右边框

语法：border-right: border-style | border-width | border-color

参数：该属性是复合属性。需要通过参数设置来实现。

3. 下边框

语法：border-bottom: border-style | border-width | border-color

参数：该属性是复合属性。需要通过参数设置来实现。

4. 左边框

语法：border-left: border-style | border-width | border-color

参数：该属性是复合属性。需要通过参数设置来实现。

5. 边框样式

在 CSS 中，样式是边框最重要的一个设置，因为如果没有样式，在 Web 页面中边框就不会显示。

`border-style` 是一个复合属性，可以同时取 1~4 个值，取值方法与外边距相似。边框属性有 12 个值可选，包括默认（`initial`）和无边框（`none`）。属性值说明见表 5-1。

表 5-1 `border-style` 属性值

| 属性值 | 说明 |
|----------------------|---|
| <code>initial</code> | 默认值，默认没有样式 |
| <code>none</code> | 定义无边框 |
| <code>hidden</code> | 与 <code>none</code> 相同，但应用于表时， <code>hidden</code> 用于解决边框冲突 |
| <code>solid</code> | 定义实线 |
| <code>dashed</code> | 定义虚线 |
| <code>dotted</code> | 定义点线 |
| <code>double</code> | 定义双线 |
| <code>inherit</code> | 定义继承父元素边框样式 |
| <code>groove</code> | 定义 3D 凹型线边框。其效果取决于 <code>border-color</code> 的值 |
| <code>ridge</code> | 定义 3D 凸型线边框。其效果取决于 <code>border-color</code> 的值 |
| <code>inset</code> | 定义 3D 嵌入式边框。其效果取决于 <code>border-color</code> 的值 |
| <code>outset</code> | 定义 3D 嵌出式边框。其效果取决于 <code>border-color</code> 的值 |

6. 边框宽度

在 CSS 中，宽度是通过 `border-width` 属性来设置边框粗细的。

与 `border-style` 属性相同，`border-width` 也是一个复合属性。设置边框宽度时，可以直接输入 `length` 确定长度值，如 `5px` 或 `2cm`，但不可以为负值；或者选择系统预设属性值。属性值说明见表 5-2。

表 5-2 `border-width` 属性值

| 属性值 | 说明 |
|----------------------|-------------|
| <code>initial</code> | 默认值，默认为中等边框 |
| <code>inherit</code> | 定义继承父元素边框宽度 |
| <code>thin</code> | 定义细边框 |
| <code>medium</code> | 定义中等边框 |
| <code>thick</code> | 定义粗边框 |



7. 边框颜色

在 CSS 中, 边框颜色是通过 `border-color` 属性来设置的, 该属性可以使用任何类型的颜色值, 包括用颜色命名的值、十六进制参数或 RGB 值。但是如果对象的 `border-style` 设置为 `none` 或者 `border-width` 设置为 0, 本属性将失去作用。

【例 5.8】 边框的设置。

```
<!-- 示例 5.8 程序 -->
```

```
<! DOCTYPE html>
```

```
<html>
```

```
  <head>
```

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

```
    <title>示例 5.8</title>
```

```
    <style type="text/css">
```

```
      h2{
```

```
        text-align:center;
```

```
        border-bottom:dashed 5px red;
```

```
      }
```

```
      p{
```

```
        font-family:"楷体";
```

```
        border-top:solid thin purple;
```

```
        border-right:dashed 5px #99FF66;    /* 水绿色 */
```

```
        border-bottom:double thick purple;
```

```
        border-left:dashed 5px #FF9999;    /* 粉红色 */
```

```
      }
```

```
      div{
```

```
        border-style:solid dashed double;
```

```
        border-width:10px;
```

```
        border-color:deepskyblue;
```

```
      }
```

```
    </style>
```

```
  </head>
```

```
  <body>
```

```
    <h2>边框的实现</h2>
```

```
    <p>这个页面主要显示边框的不同样式、宽度和颜色,以实现对象的美观化。</p>
```

```
    <div>1. h2 处所用边框 CSS 是 5 像素红色虚线的下边框;<br>
```

```
    2. p 处所用边框 CSS 是细的紫色实线上边框、5 像素水绿色虚线右边框、粗的紫色双实线下边框、5 像素粉红色虚线左边框;<br>
```

```
    3. div 处所用边框 CSS 是 10 像素蓝色实线、虚线、双实线。
```

```
    </div>
```

```
  </body>
```

```
</html>
```

本例浏览效果如图 5-10 所示。



图 5-10 各种边框的设置

5.3.5 新增边框属性

CSS3 中对边框新增了几个属性，设计者可以通过这些属性创建圆角边框、添加边框阴影、使用图片来绘制边框等。

1. 圆角边框

`border-radius`：设置边框四个角有弧度成为圆角，需要设置相关参数。

【例 5.9】圆角边框的设置。

```
<! --示例 5.9 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 5.9</title>
    <style type="text/css">
      div {
        border:2px solid red;
        border-radius:25px;
      }
    </style>
  </head>
  <body>
    <center>
```

```
<div>利用 border-radius 属性设置四个圆角边框。</div>
</center>
</body>
</html>
```

本例浏览效果如图 5-11 所示。



图 5-11 圆角边框

2. 阴影边框

box-shadow: 向四个边框添加一到多个阴影, 需要设置相关参数。

【例 5.10】 阴影边框的设置。

```
<!--示例 5.10 程序-->
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 5.10</title>
    <style type="text/css">
      div {
        margin: 100px;
        border: 2px solid blue;
        box-shadow: -15px -10px 50px red;
      }
    </style>
  </head>
  <body>
    <div>利用 box-shadow 属性设置边框的红色阴影。其中阴影向左偏移 15 像素, 向上偏移 20 像素, 模糊距离为 50 像素。
  </div>
</body>
</html>
```

本例浏览效果如图 5-12 所示。



图 5-12 阴影边框

3. 图片绘制边框

`border-image`: 设置所有边框用图片显示, 需要嵌入相关图片, 但是部分浏览器不支持此属性。

【例 5.11】 图片边框的设置。

```
<!-- 示例 5.11 程序 -->
```

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

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

```
    <title>示例 5.11</title>
```

```
    <style type="text/css">
```

```
      div {
```

```
        margin: 100px;
```

```
        border: 50px solid blue;
```

```
        border-image: url(img/border.jpg) 5 10 round;
```

```
      }
```

```
    </style>
```

```
  </head>
```

```
  <body>
```

```
    <div>利用 border-image 属性设置图片边框铺满效果。上下向内偏移 5 像素, 左右向内偏移 10 像素。</div>
```

```
  </body>
```

```
</html>
```

本例浏览效果如图 5-13 所示。



图 5-13 图片边框

5.4 CSS 元素的定位

前面所讲的盒模型都是标准流情况下可用的，但是盒模型仅有的几种排版对布局有很大的限制，导致元素无法在页面中随意地摆放，因此，我们需要使用盒子的定位（position）来增加排版的灵活性和适应性。

定位（position）的思想是，它允许你定义元素框相对于其正常位置应该出现的位置，或者相对于父元素、另一个元素甚至浏览器窗口本身的位置。

position 属性值有 4 个。

语法：position: static | relative | absolute | fixed

参数：static 是默认值，默认没有定位，或者用于取消特殊定位的继承，恢复默认，又称静态定位。relative 是相对定位，生成相对定位的元素，相对于其正常位置进行定位。absolute 是绝对定位，相对于父元素或者浏览器窗口进行定位，需要 top、right、bottom 和 left 属性辅助完成。fixed 是固定定位，相对于浏览器窗口进行定位，需要 top、right、bottom 和 left 属性辅助完成。

5.4.1 static 定位

static 是 HTML 元素的默认值，不受 top、right、bottom 和 left 属性影响，元素出现在正常的文档流中。

【例 5.12】 静态定位

```
<! --示例 5.12 程序-->
```

```
<! DOCTYPE html>
```

```
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 5.12</title>
    <style type="text/css">
      .father {
        border:2px solid red;
        width:300px;
        height:250px;
      }
      .son1 {
        border:2px double red;
        background-color:yellow;
        width:200px;
        height:80px;
      }
      .son2 {
        border:2px double red;
        width:200px;
        height:25px;
        margin-top:50px;
      }
    </style>
  </head>
  <body>
    <div class="father">父盒子;无定位
    <div class="son1">子盒子 1;无定位的盒子
    <h2>静态定位的盒子</h2>
    </div>
    <div class="son2">子盒子 2;无定位的盒子
    </div>
    </div>
  </body>
</html>
```

本例浏览效果如图 5-14 所示。

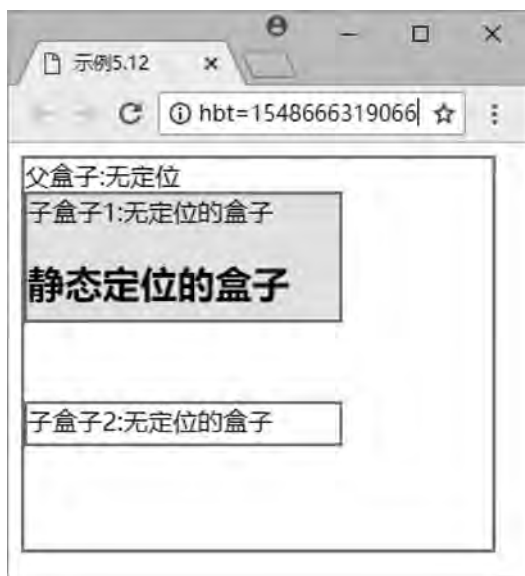


图 5-14 盒子的静态定位

以上示例中父盒子和两个子盒子都没有设置任何的 `position` 属性，所以盒子均处于静态定位，页面布局不受影响。

5.4.2 relative 定位

`relative` 不脱离文档流的布局，需要参照父元素的四条边（不是浏览器），设置自身的 `top`、`right`、`bottom` 和 `left` 属性的参数，从盒子中独立出来浮在上面。相对定位只改变自身的位置，在文档流原先的位置留出空白区域。定位的起始位置为此元素原先在文档流的位置。

在例 5.12 示例程序中把 `son1` 的 CSS 添加 `position`、`top` 和 `left` 属性，即可把静态定位改为相对定位。

【例 5.13】 相对定位。

```
<style type="text/css" >
.son1 {
    border:2px double red;
    background-color:yellow;
    width:200px;
    height:80px;
    position:relative;
    top:10px;left:30px;
}
```

本例浏览效果如图 5-15 所示。

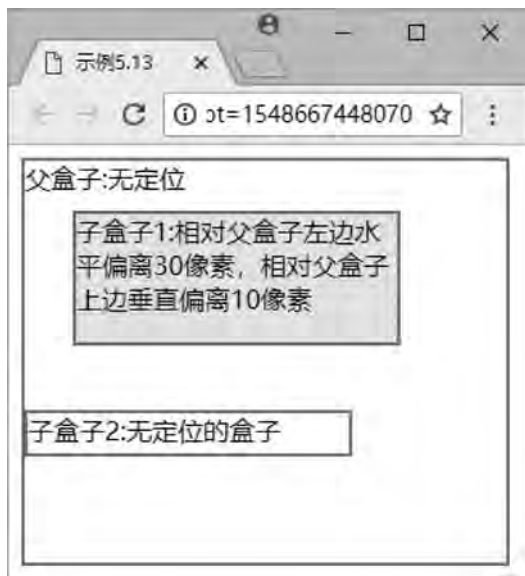


图 5-15 盒子的相对定位

以上示例中父盒子和子盒子 2 都没有设置任何的 `position` 属性, 子盒子 1 利用 `position`、`top` 和 `left` 的参数偏移了原来位置, 其原来位置就留出空白区域, 子盒子 2 的位置不受影响。

5.4.3 absolute 定位

`absolute` 脱离原来文档流的布局, 浮在其他盒子上面, 独立出来。子盒子原来位置的空间由后面的盒子填充。绝对定位的起始位置为最近已定位的父盒子, 如果父盒子没有定位, 那么子盒子的起始位置为浏览器, 并随着滚动条的移动而改变位置。

1. 相对浏览器绝对定位

【例 5.14】只需要在例 5.12 示例程序中把 `son1` 的 CSS 修改 `position`、`top` 和 `right` 属性即可。

```
<style type="text/css">
.father{
    border:2px solid red;
    width:300px;height:250px;
}
.son1{
    border:2px double red;
    background-color:yellow;
    width:200px;height:80px;
    position:absolute;
```

```

        top:10px;right:30px;
    }
    .son2{
        border:2px double red;
        width:200px;height:25px;
        margin-top:50px;
    }
</style>

```

本例浏览效果如图 5-16 所示。



图 5-16 盒子相对浏览器绝对定位

以上示例中父盒子和子盒子 2 都没有设置任何的 position 属性，所以子盒子 1 利用 position、top 和 left 的参数相对浏览器绝对定位，独立开了原来位置，其原来位置被子盒子 2 上挤填充，子盒子 2 的位置受到影响。

2. 相对父盒子绝对定位

【例 5.15】只需要在例 5.14 示例程序中把 father 和 son1 的 CSS 修改 position、top 和 right 属性即可。

```

<style type="text/css">
    .father{
        border:2px solid red;
        width:300px;
        height:250px;
    }

```



```
        position: relative;
    }
    .son1 {
        border: 2px double red;
        background-color: yellow;
        width: 200px;
        height: 80px;
        position: absolute;
        top: 30px; right: 10px;
    }
    .son2 {
        border: 2px double red;
        width: 200px;
        height: 25px;
        margin-top: 50px;
    }
}
</style>
```

本例浏览效果如图 5-17 所示。

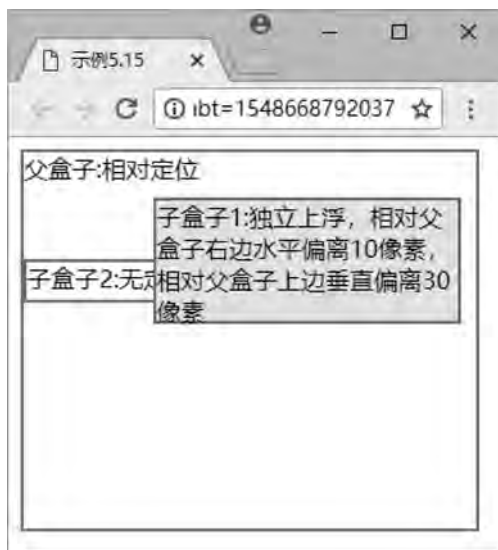


图 5-17 盒子相对父盒子绝对定位

以上示例中子盒子 2 没有设置任何的 position 属性，父盒子设置了相对定位，所以子盒子 1 以最近的已定位的父盒子为基准，进行偏移，独立开了原来位置，其原来位置被子盒子 2 上挤填充，子盒子 2 的位置受到影响。

本例浏览效果如图 5-18 所示。

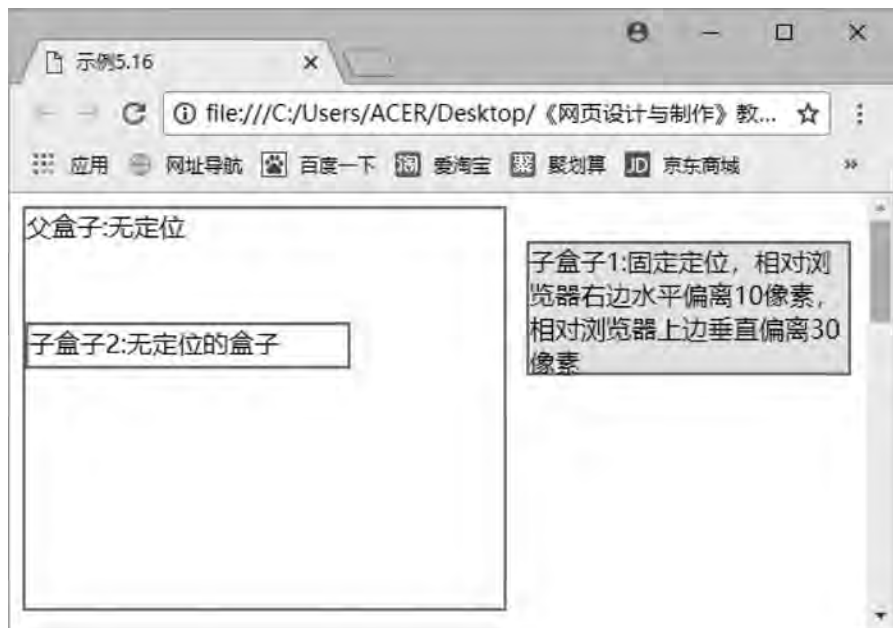


图 5-18 盒子的固定定位

以上示例中父盒子和子盒子 2 都没有设置任何的 position 属性，所以子盒子 1 利用 position、top 和 left 的参数相对浏览器固定定位，独立开了原来位置，其原来位置被子盒子 2 上挤填充，子盒子 2 的位置受到影响。子盒子 1 随着滚动条的移动位置不变。

5.5 CSS 元素的浮动

在 CSS 的盒模型布局上，除了使用定位避免按照标准流的方式进行排版的限制性问题，还可以使用浮动来避免。而使用浮动 (float) 和清除 (clear) 属性设置，可以解决各种页面错位的现象。

5.5.1 盒子的浮动添加

一般情况下，页面中的块状元素在水平方向上宽度会自动延伸，直到父元素的边界；而在垂直方向上会按照元素在页面中出现的先后次序依次排列，即所说的标准流排列，如图 5-19 所示。

任何元素都可以浮动，浮动元素会变成一个块状元素，元素的水平浮动就是通过其 float 属性的设置，使元素向其父元素的左侧或右侧靠拢，间接设置一个元素的文本环绕方式，从而改变原来的竖直排列方式。此时元素的宽度不再延伸，大小将由其内容的宽度而定。

语法: float: left | right | none

参数: left 元素浮动到左边, 即向左侧靠拢, 则右边可以有文字环绕; right 元素浮动到右边, 即向右侧靠拢, 左边可以有文字环绕; 默认值 none 就是标准流通常的显示状态。

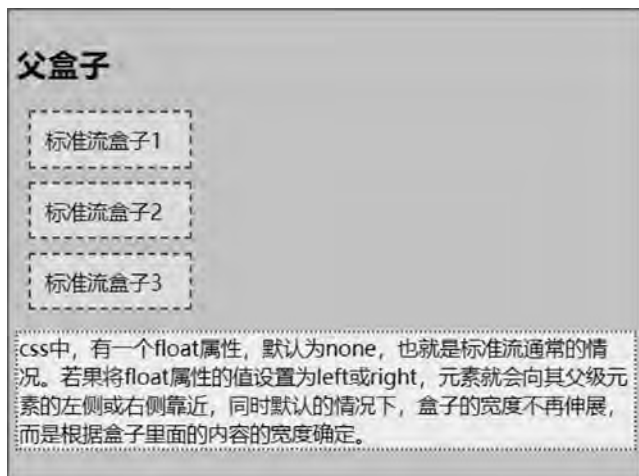


图 5-19 标准流元素状态

【例 5.17】 right 浮动

<! -- 示例 5.17 程序 -->

<! DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>示例 5.17</title>

<style type="text/css">

. father {

background-color:#FFCCFF;

border:2px solid red;

padding:5px;

}

. father div {

width:100px;

height:20px;

padding:10px;

margin:10px;

border:2px dashed blue;

background-color:#CCFFFF;

}

```

    .father p {
        border:2px dotted green;
        background-color:#FFFF99;
    }
</style>
</head>
<body>
    <div class="father">
        <h2>父盒子</h2>
        <div style="float:right;">右浮动盒子 1</div>
        <div >标准流盒子 2</div>
        <div >标准流盒子 3</div>
        <p>css 中,有一个 float 属性,默认为 none,也就是标准流通常的情况。若果将 float 属性的值设置为 left 或 right,元素就会向其父级元素的左侧或右侧靠近,同时默认的情况下,盒子的宽度不再伸展,而是根据盒子里面的内容的宽度确定。</p>
    </div>
</body>
</html>

```

本例浏览效果如图 5-20 所示。



图 5-20 单边右浮动

以上示例中只有子盒子 1 应用了 right 右浮动,所以子盒子 1 相对父盒子离开原来位置,浮动到右边,即向右侧靠拢。其原来位置被下面元素能够填充。

如果把子盒子 1 的 float 属性修改,可实现子盒子 1 的左浮动,下面元素依次上移填充。效果如图 5-21 所示。代码如下:

```
<div style="float:left;">左浮动盒子 1</div>
```



图 5-21 单边左浮动

如果把三个子盒子的 float 属性同时修改为右浮动，可实现多盒子处于同一行内，并有文字环绕效果。效果如图 5-22 所示。代码如下：

```
<div style="float:right;">右浮动盒子 1</div>
<div style="float:right;">右浮动盒子 2</div>
<div style="float:right;">右浮动盒子 3</div>
```



图 5-22 多盒子右浮动

以上示例中三个子盒子均应用了 right 右浮动，所以子盒子 1 相对父盒子先离开原来位置，浮动到右边，即向右侧靠拢，子盒子 2 相对子盒子 1 浮动到右边，子盒子 3 相对子盒子 2 浮动到右边，文字盒子没有应用浮动，直接上移填充。

5.5.2 盒子的浮动清除

元素浮动后，下面的元素内容会自动上移，结果就会受到上面浮动元素的影响，如果想要清除这种影响，需要使用 `clear` 属性完成。

由于浮动元素可以清除，是相对定位属性的优势，因而浮动属性成为控制分栏布局的最好工具。

语法：`clear: left | right | both | none`

参数：`left` 清除左边浮动元素，即不允许左边有浮动对象；`right` 清除右边浮动元素，即不允许右边有浮动对象；`Both` 同时清除左右两边的浮动元素，即不允许左右两边有浮动对象；默认值 `none`。

根据图 5-22 可以看到，三个子盒子均设置右浮动后，文字元素因为没有设置浮动而上移至原来子盒子 1 的位置进行填充，形成了元素叠加情况。因此，想要解决这种情况，可以利用 `clear` 属性设置。

【例 5.18】清除浮动。

<! --示例 5.18 程序-->

<! DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>示例 5.18</title>

<style type="text/css">

.father{

width:500px;

background-color:#FFCCFF;

border:2px solid red;

padding:5px;

}

.father div{

width:100px;height:20px;

padding:10px;margin:10px;

border:2px dashed blue;

background-color:#CCFFFF;

}

.father p{

border:2px dotted green;

background-color:#FFFF99;

clear:both;

}

</style>

</head>

<body>

```

<div class="father">
<h2>父盒子</h2>
<div style="float:right;">右浮动盒子 1</div>
<div style="float:right;">右浮动盒子 2</div>
<div style="float:right;">右浮动盒子 3</div>

```

<p>css 中,有一个 float 属性,默认为 none,也就是标准流通常的情况。若果将 float 属性的值设置为 left 或 right,元素就会向其父级元素的左侧或右侧靠近,同时默认的情况下,盒子的宽度不再伸展,而是根据盒子里面的内容的宽度确定。</p>

```

</div>
</body>
</html>

```

本例浏览效果如图 5-23 所示。



图 5-23 清除左右浮动

以上示例中,在段落文字设置了 `clear: both` 属性后,文字元素左右两边均不允许有浮动对象,因此,文字元素下移一行,与三个子盒子独立开来。

段落文字设置了 `clear: left` 属性后,浏览效果也如图 5-22 所示。

段落文字设置了 `clear: right` 属性后,浏览效果也如图 5-23 所示。

5.6 综合案例——昵心美食空间

本章综合前面学习的盒模型布局排版知识,可以利用各种定位和浮动制作典型的网页布局。本章项目页面布局要求如下:

页面要求用 div 布局,总的 div-all 有上下 4 行区域,分别为 div-banner、div-menu、div-main 和 div-bottom。而 div-main 区域又分为 div-left、div-middle、div-right 三列区域,div-middle 区域上中下分为 div-one、div-two、div-three 三行小区域,如图 5-24 所示。

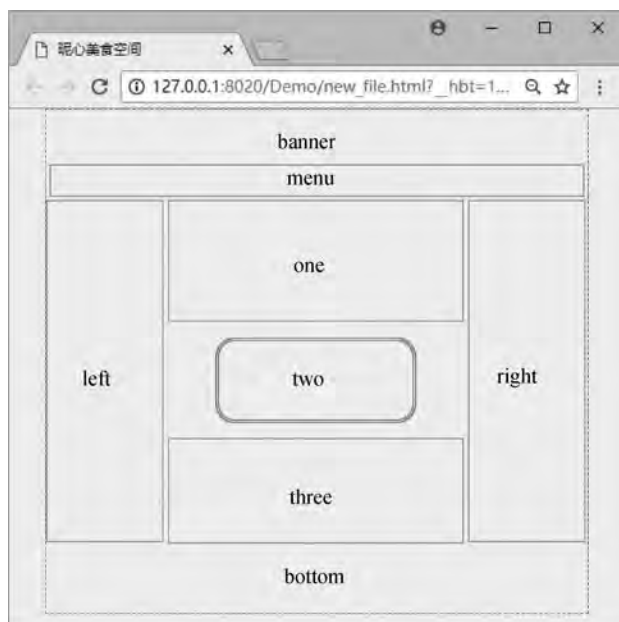


图 5-24 网页布局结构图

从图 5-24 可以看出整个页面的结构是相对浏览器水平居中的，盒模型布局代码如下：

```
<!-- 综合案例-布局.html -->
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>昵心美食空间</title>
    <style type="text/css">
      * {
        margin: 0px auto;
        background-color: #FFFF99;
      }
      .all {
        width: 700px;
        height: 650px;
        border: 2px dashed red;
      }
      .banner {
        width: 700px;
        height: 70px;
      }
      .menu {
        width: 690px;
```

```
        height:40px;
        border:1px solid #999;
    }
    .main{
        width:700px;
        height:450px;
        margin:5px 0px;
        position:relative;
    }
    .left{
        width:150px;
        height:440px;
        border:1px solid #999;
        float:left;
    }
    .middle{
        width:384px;
        height:450px;
        margin:0px 5px;
        float:left;
    }
    .one{
        width:380px;
        height:155px;
        border:1px solid #999;
    }
    .two{
        width:250px;
        height:100px;
        border:5px double red;
        margin-top:20px;
        margin-bottom:20px;
        border-radius:25px;
    }
    .three{
        width:380px;
        height:135px;
        border:1px solid #999;
    }
    .right{
        width:150px;
```

```
        height:440px;
        border:1px solid #999;
        float:left;
    }
    .bottom{
        width:700px;
        height:70px;
    }
</style>
</head>
<body>
    <div class="all">
        <div class="banner"></div>
        <div class="menu"></div>
        <div class="main">
            <div class="left"></div>
            <div class="middle">
                <div class="one"></div>
                <div class="two"></div>
                <div class="three"></div>
            </div>
            <div class="right"></div>
        </div>
        <div class="bottom"><hr></div>
    </div>
</body>
</html>
```

(1) 为 div-banner 的 div-menu 添加图片信息, 并设置相应 CSS 样式。代码如下:

```
<style>
    .menu{
        padding:5px;
    }
</style>
<body>
<div class="banner">
    
</div>
<div class="menu">
    
</div>
</body>
```



(2) 为 div-main 的 div-left 和 div-right 添加图片信息。代码如下:

```
<body>
<div class="left">
  <marquee direction="up">
    
    
    
  </marquee>
</div>
<div class="right">
  <marquee direction="up">
    
    
    
  </marquee>
</div>
</body>
```

(3) 为 div-middle 的 div-one 添加图文信息, 并设置相应 CSS 样式。代码如下:

```
<style>
.middle{
  font-size:20px;
  font-family:"楷体";
  font-weight:700;
  color:#0000FF;
}
</style>
<body>
<div class="one">
  为您推荐
  <br><br>
  
  
  
  
  
  
</div>
</body>
```

(4) 为 div-middle 的 div-two 添加文字信息。代码如下:

```
<body>
<center>
```

```
<div class="two">
  <h1>昵心美食空间</h1>
</div>
</center>
</body>
```

(5) 为 div-middle 的 div-three 添加图文信息, 并设置相应 CSS 样式。代码如下:

```
<style>
a {
  color:red;
}
</style>
<body>
<div class="three">
  团购信息
  <br>
  <a href="#">1. 火锅团购</a><br>
  <a href="#">2. 烧烤团购</a><br>
  <a href="#">3. 自助餐团购</a><br>
  <a href="#">4. 新春特惠</a>
</div>
</body>
```

(6) 为 div-bottom 添加文字信息。代码如下:

```
<body>
<div class="bottom">
  <hr color="#0000FF">
  <center style="font-family:'楷体';">版权所有 &copy;昵心美食空间<br />
  地址:江门市大学路 XXX 号 邮编:500000 电话:0750-9999999
  </center>
</div>
</body>
```

(7) 最后, 为 div-all 设置相应 CSS 样式。代码如下:

```
<style>
.all {
  width:700px;
  height:650px;
  margin:10px auto;
  padding:5px;
  background-image:url(img/bg1.jpg);
}
</style>
```

完成以上步骤, 综合案例最终网页浏览效果如图 5-25 所示。



图 5-25 网页浏览效果图

以下为完整代码，以供参考。

```

<!--综合案例.html-->
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>昵心美食空间</title>
    <style type="text/css">
      * {
        background-color:#FFFF99;
      }
    </style>
  </head>
  <body>
    <div id="header">
      <h1>昵心美食空间</h1>
      <ul>
        <li><a href="#">首页</a></li>
        <li><a href="#">菜谱</a></li>
        <li><a href="#">食材</a></li>
        <li><a href="#">日志</a></li>
        <li><a href="#">话题</a></li>
        <li><a href="#">健康</a></li>
        <li><a href="#">活动</a></li>
      </ul>
    </div>
    <div id="content">
      <div id="recommend">
        <h3>为您推荐</h3>
        <ul>
          <li>红烧肉</li>
          <li>鱼香肉丝</li>
          <li>可乐鸡翅</li>
          <li>酸菜鱼</li>
          <li>意大利面</li>
          <li>糖醋排骨</li>
        </ul>
      </div>
      <div id="group-buy">
        <h3>团购信息</h3>
        <ol>
          <li>1. 火锅团购</li>
          <li>2. 烧烤团购</li>
          <li>3. 自助餐团购</li>
          <li>4. 新春特惠</li>
        </ol>
      </div>
    </div>
    <div id="footer">
      <p>版权所有©昵心美食空间  
地址: 江门市大学路XXX号 邮编: 500000 电话: 0750-9999999</p>
    </div>
  </body>
</html>

```

```
a {
    color: red;
}

.all {
    width: 700px;
    height: 650px;
    margin: 10px auto;
    padding: 5px;
    background-image: url(img/bg1.jpg);
}

.banner {
    width: 700px;
    height: 70px;
}

.menu {
    width: 690px;
    height: 40px;
    padding: 5px;
}

.main {
    width: 700px;
    height: 450px;
    margin: 5px 0px;
    position: relative;
}

.left, .right {
    width: 150px;
    height: 440px;
    border: 1px solid #999;
    float: left;
}

.middle {
    width: 384px;
    height: 450px;
    margin: 0px 5px;
    float: left;
    font-size: 20px;
    font-family: "楷体";
```

```

        font-weight:700;
        color:#0000FF;
    }
    .one {
        width:380px;
        height:155px;
        border:1px solid #999;
    }
    .two {
        width:255px;
        height:100px;
        border:5px double red;
        margin-top:20px;
        margin-bottom:20px;
        border-radius:25px;
    }
    .three {
        width:380px;
        height:135px;
        border:1px solid #999;
    }
    .bottom {
        width:700px;
        height:70px;
    }
</style>
</head>
<body>
    <div class="all">
    <div class="banner">
    
    </div>
    <div class="menu">
    
    </div>
    <div class="main">
    <div class="left">
    <marquee direction="up">

```



```



</marquee>
</div>
<div class="middle">
<div class="one">
为您推荐
<br><br>






</div>
<center>
<div class="two">
<h1>点心美食空间</h1>
</div>
</center>
<div class="three">
团购信息
<br>
<a href="#">1. 火锅团购</a><br>
<a href="#">2. 烧烤团购</a><br>
<a href="#">3. 自助餐团购</a><br>
<a href="#">4. 新春特惠</a>
</div>
</div>
<div class="right">
<marquee direction="up">



</marquee>
</div>
</div>
```

```

<div class="bottom">
<hr color="#0000FF">
<center style="font-family:'楷体';>版权所有 &copy; 呢心美食空间<br />
地址:江门市大学路 XXX 号 邮编:500000 电话:0750-9999999</center>
</div>
</div>
</body>
</html>

```

本章习题

1. 用盒模型技术，制作一个“走进内心的文学”页面。图 5-26 供参考。



图 5-26 题 1 网页效果图

2. 用盒模型技术，制作一个“人生哲理”页面。图5-27供参考。



图5-27 题2网页效果图

第 6 章 元素应用 CSS



学习目标

1. 掌握使用 CSS 设置字体样式、设置文本样式；
2. 掌握使用 CSS 设置图像样式；
3. 综合运用 CSS 设置表单样式。



知识要点

1. CSS 的字体样式的常用属性；
2. CSS 的文本样式的常用属性；
3. CSS 的图像控制的常用属性；
4. CSS 的表单设置的常用属性。

CSS 为了能对网页布局、字体、颜色、背景和其他图文效果实现更加精确的控制，定义了相当数量的属性。从本章开始逐一介绍网页设计的各种元素的 CSS 属性，进而实现格式化排版。

6.1 使用 CSS 设置字体样式

在学习 HTML 时，通常也会使用 HTML 对文本字体进行一些非常简单的样式设置，而使用 CSS 对字体样式进行设置远比使用 HTML 灵活、精确得多。

CSS 样式中有关字体样式的常用属性见表 6-1。

表 6-1 字体样式的常用属性

| 属性 | 说明 | 属性 | 说明 |
|-------------|---------|-------------|---------|
| font-family | 设置字体的类型 | font-weight | 设置字体的粗细 |
| font-size | 设置字体的大小 | font-style | 设置字体的倾斜 |

6.1.1 字体类型

字体具有两方面的作用：一是传递语义功能，二是有美学效应。由于不同的字体给人带

来不同的风格感受，所以对于网页设计人员来说，首先需要考虑的问题就是准确地选择字体。

通常，访问者的计算机中不会安装诸如“方正综艺简体”和“方正水柱简体”等特殊字体，如果网页设计者使用这些字体，极有可能造成访问者看到的页面效果与设计者的本意存在很大差异。为了避免这种情况的发生，一般使用系统默认的“宋体”“仿宋体”“黑体”“楷体”、Aial、Verdana 和 Times New Roman 等常规字体。

CSS 提供 font-family 属性来控制文本的字体类型。

格式如下：

font-family: 字体名称；

参数：字体名称按优先顺序排列，以逗号隔开。如果字体名称包含空格，则应用引号括起。

说明：用 font-family 属性可控制显示字体。不同的操作系统，其字体名是不同的。对于 Windows 系统，其字体名就如 Word 中的“字体”列表中所列出的字体名称。

【例 6.1】字体类型设置，本例浏览效果如图 6-1 所示。



图 6-1 字体类型的页面显示效果

代码如下：

```
<!-- 示例 6.1 程序 -->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>字体类型设置</title>
    <style type="text/css">
      h1 {
        font-family:黑体;
      }
      p {
        font-family:arial,"times new roman";
      }
    </style>
  </head>
  <body>
    <h1>经营理念</h1>
    <p>用户是上帝,也是我们的衣食父母,坚持用户利益至上,不断倾听和深刻
    理解用户需求,不断给用户惊喜,不断提供超预期的体验和服务,不断创
    造新的用户价值。</p>
  </body>
</html>
```

```

    }
    </style>
</head>
<body>
    <h1>经营理念</h1>
    <p>用户是上帝,也是我们的衣食父母,坚持用户利益至上,不断倾听和深刻理解用户需求,不断给用户惊喜,不断提供超预期的体验和服务,不断创造新的用户价值。</p>
</body>
</html>

```

以上示例中,中文页面尽量首先使用“宋体”,英文页面可以使用 Arial、Verdana 和 Times New Roman 等字体。

6.1.2 字体大小

在设计页面时,通常使用不同大小的字体来突出要表现的主题,在 CSS 样式中使用 font-size 属性设置字体的大小,其值可以是绝对值也可以是相对值。常见的有“px”(绝对位)、“pt”(绝对单位)、“em”(相对单位)和“%”(相对单位)等。

语法:

font-size:绝对尺寸|相对尺寸;

参数:绝对字体尺寸是根据对象字体进行调节的,包括 xx-small、x-small、small、medium、large、x-large 和 xx-large 的 7 种字体尺寸,这些尺寸都没有精确定义,只是相对而言的,在不同的设备下,这些关键字可能会显示不同的字号。

相对尺寸是利用百分比或者 em 以相对父元素大小的方式来设置字体尺寸。

【例 6.2】字体大小设置,本例浏览效果如图 6-2 所示。



图 6-2 字体大小的页面显示效果

在例 6.1 的基础上,只修改段落的 CSS 定义就可以,代码如下:

```

p{
    font-family:arial,"times new roman";

```

```
font-size:16pt;  
}
```

以上示例中，不同尺寸的字在网页中有些是美观的，有些却不合适。本例为了演示正文字体放大的效果，将段落的字体尺寸定义为 16pt。但在实际的应用中，宋体 9pt 是公认的美观字号，绝大多数网页的正文都用它。

6.1.3 字体粗细

CSS 样式中使用 `font-weight` 属性设置字体的粗细，它包含 `normal`、`bold`、`bolder`、`lighter`、100、200、300、400、500、600、700、800 和 900 多个属性值。语法：

```
font-weight:bold|number|normal|lighter|100-900;
```

参数：`normal` 表示默认字体，`bold` 表示粗体，`bolder` 表示粗体再加粗，`lighter` 表示比默认字体还细，100~900 共分为 9 个层次（100、200、…、900，数字越小字体越细、数字越大字体越粗，数字值 400 相当于关键字 `normal`，700 等价于 `bold`）。

说明：设置文本字体的粗细。

【例 6.3】下面进行字体粗细的设置，本例浏览效果如图 6-3 所示。



图 6-3 字体粗细页面效果图

代码如下：

```
<!-- 示例 6.3 程序 -->  
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="UTF-8">  
    <title>字体粗细设置</title>  
    <style type="text/css">  
      h1 {  
        font-family:黑体;  
      }  
    </style>  
  </head>  
</html>
```

```
p {
    font-family:arial," times new roman" ;
}
.one {
    font-weight:normal;
    font-size:30px;
}
.two {
    font-weight:bold;
    font-size:30px;
}
.three {
    font-weight:bolder;
    font-size:30px;
}
.four {
    font-weight:lighter;
    font-size:30px;
}
.five {
    font-weight:900;
    font-size:30px;
}
</style>
</head>
<body>
    <h1>经营理念</h1>
    <p><span class="one">用户</span>是上帝,也是我们的衣食父母,坚持<span class="two">用户
</span>利益至上,不断倾听和深刻理解<span class="three">用户</span>需求,不断给<span class="four">用户
</span>惊喜,不断提供超预期的体验和服务,不断创造新的<span class="five">用户</span>价值。</p>
</body>
</html>
```

实际上大多数操作系统和浏览器还不能很好地实现非常精细的文本加粗设置,通常只能设置“正常”(normal)和“加粗”(bold)两种粗细效果。

6.1.4 字体倾斜

CSS 中的 font-style 属性用来设置字体的倾斜。语法:

font-style:normal|italic|oblique;

参数: normal 为“正常”(默认值), italic 为“斜体”, oblique 为“倾斜体”。

说明: 设置文本字体的倾斜。

【例 6.4】下面进行字体倾斜的设置,本例浏览效果如图 6-4 所示。



图 6-4 字体倾斜的设置效果图

代码如下:

```
<! --示例 6.4 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>字体倾斜设置</title>
    <style type="text/css">
      h1 {
        font-family:黑体;
      }
      p {
        font-family:arial,"times new roman";
        font-style:italic;/* 设置斜体 */
      }
    </style>
  </head>
  <body>
    <h1>经营理念</h1>
    <p>用户是上帝,也是我们的衣食父母,坚持用户利益至上,不断倾听和深刻理解用户需求,不断给用户惊喜,不断提供超预期的体验和服务,不断创造新的用户价值。</p>
  </body>
</html>
```

6.2 使用 CSS 设置文本样式

网页的排版离不开对文本的设置，本节主要讲述常用的文本样式，包括文本对齐方式、行高、文本修饰、段落首行缩进、首字下沉、字符间距、文本截断、文本颜色及背景色等。

CSS 样式中有关文本样式的常用属性见表 6-2。

表 6-2 文本样式的常用属性

| 属性 | 说明 | 属性 | 说明 |
|-----------------|-------------|------------------|-----------|
| text-align | 设置文本的水平对齐方式 | letter-spacing | 设置字符间距 |
| line-height | 设置行高 | text-overflow | 设置文本的截断 |
| text-decoration | 设置文本修饰效果 | color | 设置文本的颜色 |
| text-indent | 设置段落的首行缩进 | background-color | 设置文本的背景颜色 |
| first-letter | 设置首字下沉 | | |

6.2.1 文本水平对齐方式

使用 text-align 属性可以设置元素中文本的水平对齐方式。语法：

text-align:left| right | center | justify;

参数：left 为左对齐，right 为右对齐，center 为居中，justify 为两端对齐。

说明：设置对象中文本的对齐方式。

【例 6.5】下面设置文本的水平对齐方式，本例浏览效果如图 6-5 所示。



图 6-5 文本水平对齐方式的设置网页效果图

代码如下：

```
<! --示例 6.5 程序-->
```

```
<! DOCTYPE html>
```

```
<html>
```

```
  <head>
```

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

```
    <title>文本的水平对齐方式设置</title>
```

```
    <style type="text/css">
```

```
      h1 {
```

```
        font-family: 黑体;
```

```
        text-align: center;
```

```
      }
```

```
      p {
```

```
        font-family: arial, "times new roman";
```

```
        text-align: left;
```

```
      }
```

```
      p. center {
```

```
        text-align: center;
```

```
      }
```

```
      p. right {
```

```
        text-align: right;
```

```
      }
```

```
    </style>
```

```
  </head>
```

```
  <body>
```

```
    <h1>经营理念</h1>
```

```
    <p class="center">作者:全球汇</p>
```

```
    <p class="right">发布时间:2019年1月31日</p>
```

```
    <p>用户是上帝,也是我们的衣食父母,坚持用户利益至上,不断倾听和深刻理解用户需求,不断给用户惊喜,不断提供超预期的体验和服务,不断创造新的用户价值。
```

```
  </p>
```

```
  </body>
```

```
</html>
```

6.2.2 行高

段落中两行文本之间垂直的距离称为行高。在 HTML 中是无法控制行高的,在 CSS 样式中,使用 `line-height` 属性控制行与行之间的垂直间距。语法:

```
line-height: length | normal;
```

参数: `length` 为由百分比数字或由数值、单位标识符组成的长度值,允许为负值。其百分比取值是基于字体的高度尺寸。`normal` 为默认行高。

说明: 设置对象的行高。

【例 6.6】 设置行高,本例浏览效果如图 6-6 所示。



图 6-6 行高设置的效果图

代码如下:

```
<!--示例 6.6 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>行高设置</title>
    <style type="text/css">
      h1 {
        font-family: 黑体;
      }
      p {
        font-family: arial, "times new roman";
        line-height: 200%; /* 设置行高为字体高度的 2 倍 */
      }
    </style>
  </head>
  <body>
    <h1>经营理念</h1>
    <p>用户是上帝,也是我们的衣食父母,坚持用户利益至上,不断倾听和深刻理解用户需求,不断给用户惊喜,不断提供超预期的体验和服务,不断创造新的用户价值。</p>
  </body>
</html>
```

以上示例中,需要注意的是,使用像素值对行高进行设置固然可以,但如果将当前文本

字号放大或缩小，原本适合的行间距也会变得过紧或过松。解决的方法是，在 `line-height` 属性中使用百分比或数值设置行高。

6.2.3 文本的修饰

使用 CSS 样式可以对文本进行简单的修饰，`text-decoration` 属性所提供的 `text-decoration` 属性，主要实现文本加下划线、顶线、删除线及文本闪烁等效果。语法：

```
text-decoration:underline| blink| overline| line-through| none;
```

参数：`underline` 为下划线，`blink` 为闪烁，`overline` 为上划线，`line-through` 为贯穿线，`none` 为无装饰。

说明：设置对象中文本的修饰。对象 `a`、`u`、`ins` 的文本修饰默认值为 `underline`。对象 `strike`、`s`、`del` 的默认值是 `line-through`。如果应用的对象不是文本，则此属性不起作用。

【例 6.7】 设置文本的修饰效果，本例浏览效果如图 6-7 所示。

代码如下：

```
<! --示例 6.7 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>文本修饰效果的设置</title>
    <style type="text/css">
      h1 {
        font-family:黑体;
      }
      p {
        font-family:arial,"times new roman";
        line-height:200%;/* 设置行高为字体高度的 2 倍 */
      }
      .one {
        font-size:30px;
        text-decoration:underline;/* 设置下划线 */
      }
      .two {
        font-size:30px;
        text-decoration:blink;/* 设置闪烁 */
      }
      .three {
        font-size:30px;
        text-decoration:overline;/* 设置上划线 */
      }
      .four {
```

```

        font-size:30px;
        text-decoration:line-through; /* 设置贯穿线 */
    }
    .five {
        font-size:30px;
        text-decoration:none;
    }
</style>
</head>
<body>
    <h1>经营理念</h1>
    <p>用户是<span class="one">上帝</span>,也是我们的衣食父母,坚持用户<span class="
two">利益至上</span>,不断<span class="three">倾听和深刻理解</span>用户需求,不断给用户惊喜,不断
<span class="four">提供</span>,不断<span class="five">创造</span>新的用户价值。
    </p>
</body>
</html>

```

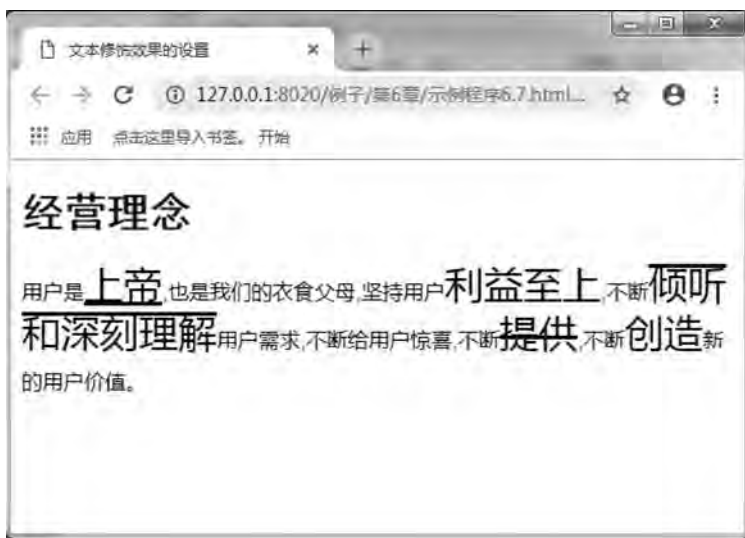


图 6-7 文本修饰效果图

6.2.4 段落首行缩进

首行缩进是指段落的第一行从左向右缩进一定的距离,而首行以外的其他行保持不变,其目的是便于阅读和区分文章整体结构。

在 Web 页面中,将段落的第一行进行缩进,同样是一种最常用的文本格式化效果。在 CSS 样式中 text-indent 属性可以方便地实现文本缩进。可以为所有块级元素应用 text-

indent, 但不能应用于行级元素。如果想把一个行级元素的第一行缩进, 可以用左内边距或外边距创造这种效果。语法:

```
text-indent:length;
```

参数: length 为百分比数字或由浮点数字、单位标识符组成的长度值, 允许为负值。

说明: 设置对象中的文本段落的缩进。本属应用于整块的内容。

【例 6.8】 设置段落的首行缩进, 本例浏览效果如图 6-8 所示。

在例 6.1 的基础上, 本例只修改段落定义, 代码如下:

```
p{  
    font-family:arial,"times new roman";  
    text-indent:2em;  
}
```



图 6-8 段落的首行缩进效果图

以上示例中, text-indent 属性是以各种长度为属性值, 为了缩进两个汉字的距离, 最常用的是 2em 这个距离。1em 等于一个中文字符, 两个英文字符相当于一个中文字符。因此, 如果用户需要英文段落的首行缩进两个英文字符, 只需设置 “text-indent: 1em;” 即可。

6.2.5 首字下沉

在许多文档的排版中经常出现首字下沉的效果, 所谓首字下沉是指设置段落的第一行第一个字的字体变大, 并且向下一一定的距离, 而段落的其他部分保持不变。

在 CSS 样式中伪对象 “: first-letter” 可以实现对象内第一个字符的样式控制。

【例 6.9】 设置首字下沉, 本例浏览效果如图 6-9 所示。



图 6-9 首字下沉的效果图

在例 6.1 的基础上,本例在 CSS 样式里添加,代码如下:

```
p:first-letter{
    float:left;                /* 设置浮动,其目的是占据多行空间 */
    font-size:2em;            /* 设置下沉字体大小为其他字体的 2 倍 */
    font-weight:bold;        /* 设置首字体加粗显示 */
}
```

以上示例中,如果不使用伪对象“: first-letter”来实现首字下沉的效果,就要对段落中第一个文字添加标签,然后定义标签的样式。但是这样做的后果是,每个段落都要对第一个文字添加标签,非常烦琐。因此,使用伪对象“: first-letter,来实现首字下沉就提高了网页排版的效率。

6.2.6 字符间距

letter-spacing 为字符间距属性,可以设置字符与字符间的距离。语法:

```
letter-spacing:length | normal;
```

参数: normal 为默认值,定义字符间的标准间距。length 表示由浮点数字和单位标识符组成的长度值,允许为负值。

说明: 该属性定义元素中字符之间插入多少空白符。如果指定为长度值,会调整字符之间的标准间距,允许指定负长度值,这会让字符之间变得更拥挤。

【例 6.10】设置字符间距,本例浏览效果如图 6-10 所示。



图 6-10 设置字符间距的效果图

代码如下：

```
<!-- 示例 6.10 程序 -->
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>字符间距的设置</title>
    <style type="text/css">
      h1 {
        font-family: 黑体;
      }
      p {
        font-family: arial, "times new roman";
      }
      p.loose {
        letter-spacing: 30px;
      }
      p.tight {
        letter-spacing: -0.25em;
      }
    </style>
  </head>
  <body>
    <h1>经营理念</h1>
    <p>坚持用户利益至上</p>
    <p class="loose">坚持用户利益至上</p>
    <p class="tight">坚持用户利益至上</p>
  </body>
</html>
```

以上示例中，从页面的显示效果可以看出，没有应用样式的段落中的字符保持默认的标准间距。凡是应用了 loose 类样式的段落中的字符松散，应用了 tight 类样式的段落中的字符变得拥挤。

6.2.7 文本的截断

在 CSS 样式中 text-overflow 属性可以实现文本的截断效果，该属性包含 clip 和 ellipsis 两个属性值。前者表示简单的裁切，不显示省略标记 (…); 后者表示当文本溢出时显示省略标记 (…)。语法:

```
text-overflow: clip | ellipsis;
```

参数: clip 定义简单的裁切，不显示省略标记。ellipsis 定义当文本溢出时显示省略标记。

说明: 设置文本的截断。要实现溢出文本显示省略号的效果，除了使用 text-overflow 属性以外，还必须配合 white-space: nowrap (强制文本在一行内显示) 和 overflow: hidden (溢出内容为隐藏) 同时使用才能实现。

【例 6.11】设置文本的截断，本例浏览效果如图 6-11 所示。



图 6-11 文本的截断设置网页设置

代码如下:

```
<! --示例 6.11 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>文本截断的设置</title>
    <style type="text/css">
      h1 {
        font-family: 黑体;
      }
      p {
        font-family: arial, "times new roman";
      }
    </style>
  </head>
  <body>
    <h1>经营理念</h1>
    <p>用户是上帝,也是我们的衣食父母,...</p>
  </body>
</html>
```

```

p.ellipsis {
    width:260px;                /* 设置裁切的宽度 */
    height:20px;               /* 设置裁切的高度 */
    overflow:hidden;          /* 溢出隐藏 */
    white-space:nowrap;       /* 强制文本在一行内显示 */
    text-overflow:ellipsis;   /* 当文本溢出时显示省略标记(……) */
}
</style>
</head>
<body>
    <h1>经营理念</h1>
    <p class="ellipsis">用户是上帝,也是我们的衣食父母,坚持用户利益至上,不断倾听和深刻理
解用户需求,不断给用户惊喜,不断提供超预期的体验和服务,不断创造新的用户价值。</p>
</body>
</html>

```

6.2.8 文本的颜色

在 CSS 样式中,对文本增加颜色修饰十分简单,只需添加 color 属性即可。color 属性的语法:

color:颜色值;

这里颜色值可以使用多种书写方式:

```

color:red;                    /* 规定颜色值为颜色名称的颜色 */
color:#000000;               /* 规定颜色值为十六进制值的颜色 */
color:rgb(0,0,255);         /* 规定颜色值为 rgb 代码的颜色 */
color:rgb(0%,0%,80%);      /* 规定颜色值为 rgb 百分数的颜色 */

```

【例 6.12】 设置文本的颜色,本例浏览效果如图 6-12 所示。

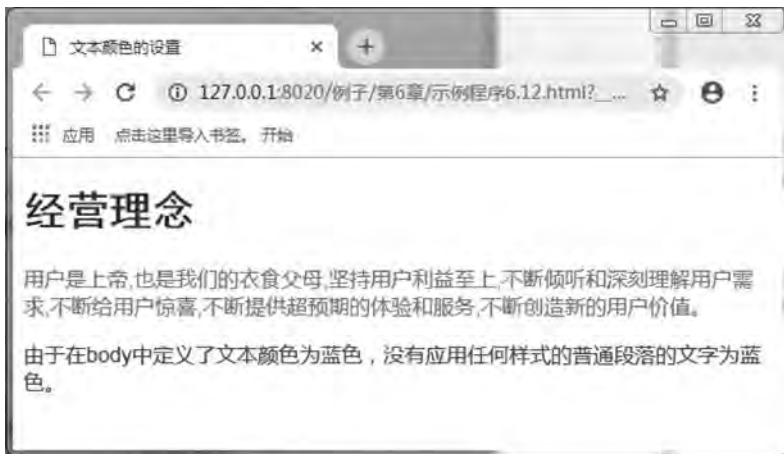


图 6-12 文本颜色设置的效果图

代码如下:

```
<! --示例 6.12 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>文本颜色的设置</title>
    <style type="text/css">
      body{
        color:blue;                                /* body 中的文本显示蓝色 */
      }
      h1{
        font-family:黑体;
        color:#333;                                /* h1 标签的文本显示深灰色 */
      }
      p{
        font-family:arial,"times new roman";
      }
      p. red{
        color:rgb(255,0,0);                        /* 该段落文本是红色的 */
      }
    </style>
  </head>
  <body>
    <h1>经营理念</h1>
    <p class="red">用户是上帝,也是我们的衣食父母,坚持用户利益至上,不断倾听和深刻理解用户需求,不断给用户惊喜,不断提供超预期的体验和服务,不断创造新的用户价值。
    </p>
    <p>由于在 body 中定义了文本颜色为蓝色,没有应用任何样式的普通段落的文字为蓝色。</
  p>
  </body>
</html>
```

6.2.9 文本的背景颜色

在 HTML 中,可以使用标签的 bgcolor 属性设置网页的背景颜色。而在 CSS 里,不仅可以用 background-color 属性来设置网页背景颜色,还可以设置文本的背景颜色。语法:

```
background-color:color | transparent
```

参数: color 用于指定颜色。transparent 表示透明的意思,也是浏览器的默认值。

说明: background-color 不能继承,默认值是 transparent。如果一个元素没有指定背景颜色,那么背景就是透明的,这样其父元素的背景才能看见。

【例 6.13】设置文本的背景颜色，本例浏览效果如图 6-13 所示。



图 6-13 文本的背景颜色设置效果图

在例 6.1 的基础上，本例增加了 body 背景色的定义，并为 h1 和 p 增加了背景色的定义。

代码如下：

```
body{
    background-color:#eee;
}
h1{
    font-family:黑体;
    background-color:orange;
}
p{
    font-family:arial,"times new roman";
    background:rgb(0,255,255);
}
```

以上示例中，需要说明的是，background-color 属性默认值为透明，如果一个元素没有指定背景色，则背景色就是透明的，这样其父元素的背景才能看见。

6.3 使用 CSS 设置图像样式

图像是网页中不可缺少的内容，它能使页面更加丰富多彩，能让人更直观地感受网页所要传达给浏览者的信息。本节详细介绍 CSS 设置图像风格样式的方法，包括图像的边框、图像的缩放和背景图像等。

图像即 img 元素，在页面中的风格样式仍然用盒模型来设计。CSS 样式中有关图像控制的常用属性见表 6-3。

表 6-3 图像样式的常用属性

| 属性 | 说明 | 属性 | 说明 |
|------------------|-----------|---------------------|-------------|
| border | 设置图像的边框样式 | background-repeat | 设置背景图像的重复方式 |
| width、height | 设置图像的缩放 | background-position | 设置背景图像的定位 |
| background-image | 设置背景图像 | | |

作为单独的图像本身，它的很多属性可以直接在 HTML 中进行调整，但是通过 CSS 统一管理，不但可以更加精确地调整图像的各种属性，还可以实现很多特殊的效果。下面首先讲解用 CSS 设置图像基本属性的方法，为进一步深入探讨打下基础。

6.3.1 设置图像边框

图像的边框就是利用 border 属性作用于图像元素而呈现的效果。在 HTML 中可以直接通过 标记的 border 属性值为图像添加边框，属性值为边框的粗细，以像素为单位，从而控制边框的粗细。当设置 border 属性值为 0 时，则显示为没有边框。例如以下代码：

```
 <!--显示为没有边框-->
 <!--设置边框的粗细为 1px-->
 <!--设置边框的粗细为 2px-->
 <!--设置边框的粗细为 3px-->
```

通过浏览器的解析，图像的边框粗细从左至右依次递增，效果如图 6-14 所示。

使用这种方法存在很大的限制，即所有的边框都只能是黑色，而且风格十分单一，都是实线，只是在边框粗细上能够进行调整。

如果希望更换边框的颜色，或者换成虚线边框，仅仅依靠 HTML 都是无法实现的。下面的实例讲解了如何用 CSS 样式美化图像的边框。

【例 6.14】 设置图像的边框，本例浏览效果如图 6-14 所示。

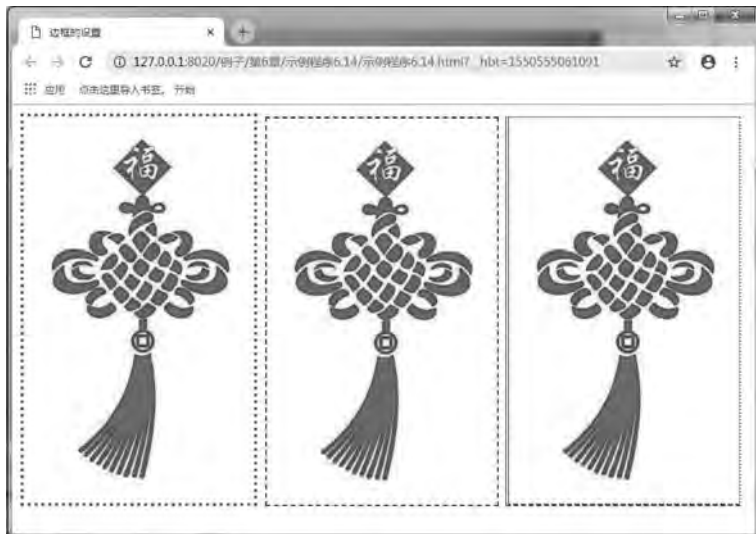


图 6-14 图像边框设置效果图

代码如下：

```
<! --示例 6.14 程序-->
! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>边框的设置</title>
    <style type="text/css">
      .test1 {
        border-style:dotted;           /* 点画线边框 */
        border-color:#996600;        /* 边框颜色为金黄色 */
        border-width:4px;            /* 边框粗细为 4px */
        margin:2px;
      }
      .test2 {
        border-style:dashed;         /* 虚线边框 */
        border-color:blue;           /* 边框颜色为蓝色 */
        border-width:2px;            /* 边框粗细为 2px */
        margin:2px;
      }
      .test3 {
        border-style:solid dotted dashed double; /* 四边的线型依次为实线、点画线、虎线和双线边
框 */
        border-color:red green blue purple; /* 四边的颜色依次为红色、绿色、蓝色、紫色 */
        border-width:1px 2px 3px 4px; /* 四边的边框粗细依次为为 1px、2px、3px、4px */
        margin:2px;
      }
    </style>
  </head>
  <body>
    
    
    
  </body>
</html>
```

以上示例中，如果希望分别设置 4 条边框的不同样式，在 CSS 中也是可以实现的，只需要分别设定 `border-left`、`border-right`、`border-top` 和 `border-bottom` 的样式，依次对应于左、右、上、下 4 条边框。

6.3.2 图像缩放

使用 CSS 样式控制图像的大小，可以通过 `width` 和 `height` 两个属性来实现。需要注意的

是，当 width 和 height 两个属性的取值使用百分比数值时，它是相对于父元素而言的，如果将这两个属性设置为相对于 body 的宽度或高度，就可以实现当浏览器窗口改变时，图像大小也发生相应变化的效果。

【例 6.15】 设置图像的缩放，本例浏览效果如图 6-15 所示。

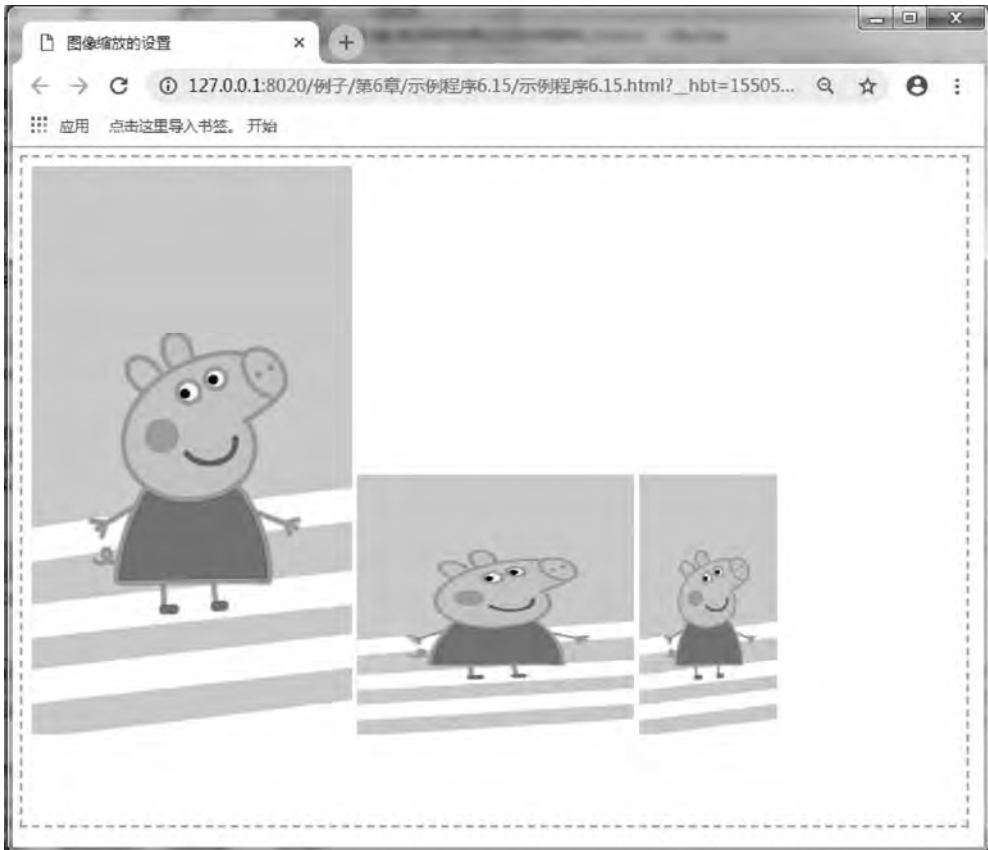


图 6-15 图像缩放设置的效果图

代码如下：

```
<! --示例 6.15 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>图像缩放设置</title>
  </head>
  <style type="text/css">
    #box {
      padding:10px;
      width:1000px;
      height:700px;
```



```
border:2px dashed #9c3;
}
img.test1 {
width:30%;           /* 相对宽度为 30% */
height:40%;         /* 相对高度为 40% */
}
img.test2 {
width:150px;        /* 绝对宽度为 150px */
height:280px;      /* 绝对高度为 280px */
}
</style>
<body>
<div id="box">
 <!-- 图像原始大小-->
 <!-- 相对于父元素缩放的大小-->
 <!-- 绝对像素缩放的大小-->
</div>
</body>
</html>
```

以上示例中，图像的父亲元素为 id = " box" 的 Div 容器，在 img.test1 中定义 width 和 height 两个属性的取值为百分比数值，该数值是相对于 id = " box" 的 Div 容器而言的，而不是相对于图像本身。

img.test2 中定义 width 和 height 两个属性的取值为绝对像素值，图像将按照定义的像素值显示大小。

6.3.3 设置背景图像

在网页设计中，无论是单一的纯色背景，还是加载的背景图片，都能够给整个页面带来丰富的视觉效果。CSS 除了可以设置背景颜色，还可以用 background-image 来设置背景图像。语法：

```
background-image:url(url)| none;
```

参数：url 表示要插入背景图像的路径。none 表示不加载图像。

说明：设置对象的背景图像。若把图像添加到整个浏览器窗口，可以将其添加到 <body> 标签中。

【例 6.16】 设置背景图像，本例浏览效果如图 6-16 所示。

```
body {
background-color:#7bc144;
background-image:url(images/fu.JPG);
background-repeat:no-repeat;
}
```



图 6-16 背景图像设置效果图

以上示例中，需要说明的是，如果网页中某元素同时具有 `background-image` 属性和 `background-color` 属性，那么 `background-image` 属性优先于 `background-color` 属性，说背景图像永远覆盖于背景色之上。

6.3.4 设置背景重复

背景重复 (`background-repeat`) 属性的主要作用是设置背景图片以何种方式在网页中显示。通过背景重复，设计人员使用很小的图片就可以填充整个页面，有效地减少图片字节的大小。

在默认情况下，图像会自动向水平和垂直两个方向平铺。如果不希望平铺，或者只希望沿着一个方向平铺，可以使用 `background-repeat` 属性来控制。语法：

```
background-repeat:repeat | no-repeat | repeat-x | repeat-y;
```

参数：`repeat` 表示背景图像在水平和垂直方向平铺，是默认值；`repeat-x` 表示背景图像在水平方向平铺；`repeat-y` 表示背景图像在垂直方向平铺；`no-repeat` 表示背景图像不平铺。

说明：设置对象的背景图像是否平铺及如何平铺，必须先指定对象的背景图像。

【例 6.17】 设置背景重复，本例浏览效果如图 6-17 所示。

背景不重复的 CSS 定义代码如下：

```
body{  
    background-color:#7bc144;  
    background-image:url(images/fu.JPG);  
    background-repeat:no-repeat;  
}
```

背景水平重复的 CSS 定义代码如下：

```
body{  
    background-color:#7bc144;  
    background-image:url(images/fu.JPG);  
    background-repeat:repeat-x;  
}
```

背景垂直重复的 CSS 定义代码如下：

```
body{  
    background-color:#7bc144;  
    background-image:url(images/fu.JPG);  
    background-repeat:repeat-y;  
}
```

背景重复的 CSS 定义代码如下：

```
body{  
    background-color:#7bc144;  
    background-image:url(images/fu.JPG);  
    background-repeat:repeat;  
}
```

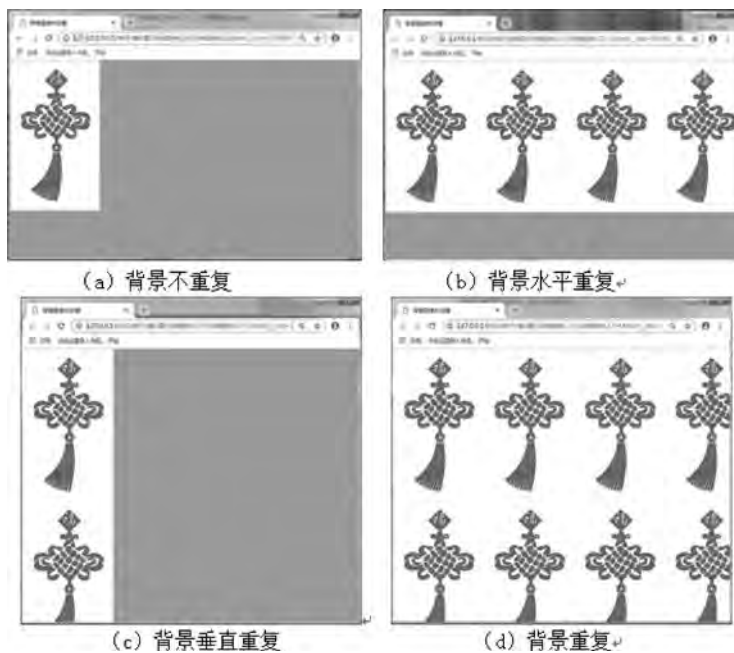


图 6-17 设置背景重复效果图

6.3.5 背景图像定位

当在网页中插入背景图像时，每一次插入的位置，都是位于网页的左上角，可以通过 `background-position` 属性来改变图像的插入位置。语法：

```
background-position:length | length;
```

```
background-position:positional | position;
```

参数: length 为百分比或者由数字和单位标识符组成的长度值, position 可取 top、center、bottom、left、right 之一。

说明: 利用百分比和长度来设置图像位置时, 都要指定两个值, 并且这两个值都要用空格隔开, 一个代表水平位置, 一个代表垂直位置。水平位置的参考点是网页页面的左边, 垂直位置的参考点是网页页面的上边。关键字在水平方向的主要有 left、center、right, 关键字在垂直方向的主要有 top、center、bottom。水平方向和垂直方向相互搭配使用。

设置背景定位有以下 3 种方法。

1. 使用关键字进行背景定位

关键字参数的取值及含义如下:

top: 将背景图像同元素的顶部对齐。

bottom: 将背景图像同元素的底部对齐。

left: 将背景图像同元素的左边对齐。

right: 将背景图像同元素的右边对齐。

center: 将背景图像相对于元素水平居中或垂直居中。

【例 6.18】使用关键字进行背景定位, 本例浏览效果如图 6-18 所示。

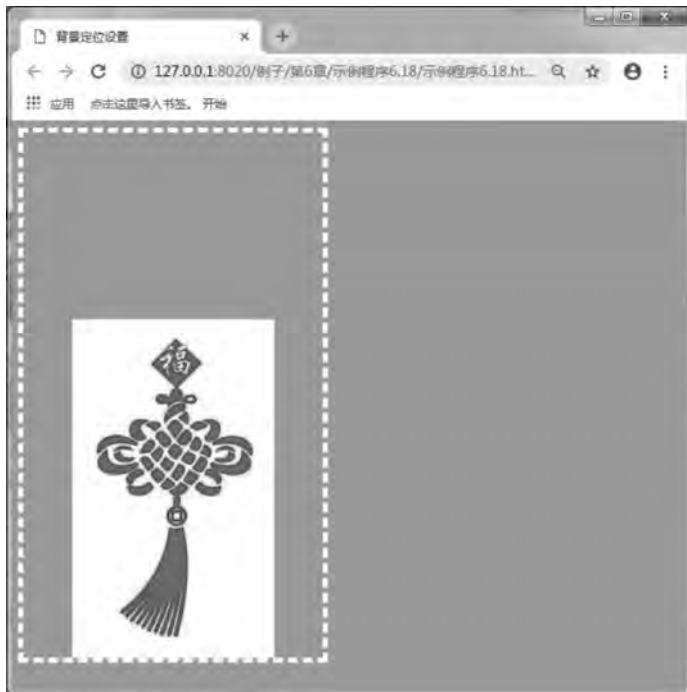


图 6-18 使用关键字进行背景定位效果图

```
<!-- 示例 6.18 程序 -->
```

```
<! DOCTYPE html>
```

```

<html>
  <head>
    <meta charset="UTF-8">
    <title>背景定位设置</title>
    <style type="text/css">
      body {
        background-color:#7bc144;
      }
      #box {
        width:400px;           /* 设置元素的宽度 */
        height:700px;         /* 设置元素的高度 */
        border:6px dashed #fff; /* 6px 白色虚线边框 */
        background-image:url(images/fu.JPG); /* 背景图像 */
        background-repeat:no-repeat; /* 背景图像不重复 */
        background-position:center bottom; /* 定位背景向 box 的底部中央对齐 */
      }
    </style>
  </head>
  <body>
    <div id="box"></div>
  </body>
</html>

```

以上示例中，根据规范，关键字可以按任何顺序出现，只要保证不超过两个关键字，一个对应水平方向，另一个对应垂直方向。如果只出现一个关键字，则认为另一个关键字是 center。

2. 使用长度进行背景定位

长度参数可以对背景图像的位置进行更精确的控制，实际上定位的是图像左上角相对于元素左上角的位置

【例 6.19】 使用长度进行背景定位，本例浏览效果如图 6-19 所示。

在例 6.18 的基础上，修改 box 的 CSS 定义，代码如下：

```

#box {
  width:900px;           /* 设置元素的宽度 */
  height:700px;         /* 设置元素的高度 */
  border:6px dashed #fff; /* 6px 白色虚线边框 */
  background-image:url(images/fu.JPG); /* 背景图像 */
  background-repeat:no-repeat; /* 背景图像不重复 */
  background-position:150px 70px; /* 定位背景向在距容器左侧 150px,距顶部 70px 的
位置 */
}

```



图 6-19 使用长度进行背景定位的效果图

3. 使用百分比进行背景定位

使用百分比进行背景定位，其实是将背景图像的百分比指定的位置和元素的百分比值置对齐。也就是说，百分比定位改变了背景图像和元素的对齐基点，不再像使用关键字或长度单位定位时，使用背景图像和元素的左上角为对齐基点。

【例 6.20】 使用百分比进行背景定位，本例浏览效果如图 6-20 所示。



图 6-20 使用百分比进行背景定位效果图

在例 6.18 的基础上，修改 box 的 CSS 定义，代码如下：

```
#box {
    width:900px;                /* 设置元素的宽度 */
    height:700px;              /* 设置元素的高度 */
    border:6px dashed #f33;     /* 6px 红色虚线边框 */
    background-image:url(images/fu.JPG); /* 背景图像 */
    background-repeat:no-repeat; /* 背景图像不重复 */
    background-position:100% 50%; /* 定位背景向在容器 100% (水平方向),50% (垂直方向)的位置 */
}
```

以上示例中，使用百分比进行背景定位时，其实就是将背景图像的“100% (right), 5% (center)”这个点和 box 容器的“100% (right), 50% (center)”这个点对齐。

6.4 使用 CSS 设置表单样式

在前面章节中讲解的表单设计大多采用表格布局，这种布局方法对表单元素的样式控制很少，仅局限于功能上的实现。本节主要讲解如何使用 CSS 控制和美化表单。

6.4.1 使用 CSS 修饰常用的表单元素

表单中的元素很多，包括常用的文本域、单选钮、复选框、下拉菜单和按钮等。下面通过一个实例讲解怎样使用 CSS 修饰常用的表单元素。

1. 修饰文本域

文本域主要用于采集用户在其中编辑的文字信息，通过 CSS 样式可以对文本域内的字体、颜色以及背景图像加以控制。下面以示例的形式介绍如何使用 CSS 修饰文本域。

【例 6.21】使用 CSS 修饰文本域，本例浏览效果如图 6-21 所示。



图 6-21 文本域修饰的效果图

代码如下:

```
<! --示例 6.21 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>文本域修饰</title>
    <style type="text/css">
      .text1 {
        border:1px solid #f60;           /* 1px 实线红色边框 */
        color:#03C;                     /* 文字颜色为蓝色 */
      }
      .text2 {
        border:1px solid #C3C;          /* 1px 实线紫红色边框 */
        height:20px;
        background:#fff url(images/password_bg.JPG) left center no-repeat; /* 背景图像无重复 */
        padding-left:20px;
      }
      .area {
        border:1px solid #00f;          /* 1px 实线蓝色边框 */
        overflow:auto;
        width:99%;
        height:100px;
      }
    </style>
  </head>
  <body>
    <p>
      <input type="text" name="normal" />
      默认样式的文本域
    </p>
    <p>
      <input name="chbd" type="text" value="输入的文字显示为蓝色" class="text1" />
      改变边框颜色和文字颜色的文本域,看起来更加醒目
    </p>
    <p>
      <input name="pass" type="password" class="text2" />
      增加了背景图片的文本域,看起来更加形象直观
    </p>
    <p>
      <textarea name="cha" cols="45" rows="5" class="area">改变边框颜色的多行文本域</
textarea>
    </p>
  </body>
</html>
```


2. 修饰按钮

按钮主要用于控制网页中的表单。通过 CSS 样式可以对按钮的字体、颜色、边框以及背景图像加以控制。下面以示例的形式介绍如何使用 CSS 修饰按钮。

【例 6.22】 使用 CSS 修饰按钮，本例浏览效果如图 6-22 所示。

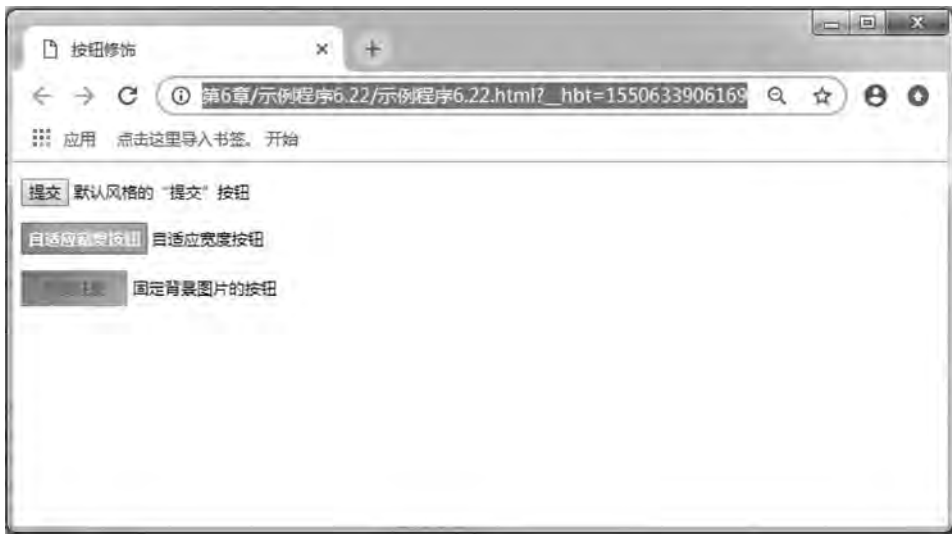


图 6-22 使用 CSS 修饰按钮的效果图

代码如下：

```
<! --示例 6.22 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>按钮修饰</title>
  </head>
  <style type="text/css">
    .btn01 {
      background:url(images/btn_bg02.jpg)repeat-x; /* 背景图像水平重复 */
      border:1px solid #F00; /* 1px 实线红色边框 */
      height:32px;
      font-weight:bold; /* 字体加粗 */
      padding-top:2px;
      cursor:pointer; /* 鼠标样式为手形 */
      font-size:14px;
      color:#FFF; /* 文字颜色为白色 */
    }
    .btn02 {
      background:url(images/btn_bg03.jpg)0 0 no-repeat; /* 背景图像无重复 */
      width:107px;
```

```

height:37px;
border:none; /* 无边框,背景图像本身就是边框风格的图像 */
font-size:14px;
font-weight:bold; /* 字体加粗 */
color:#d84700;
cursor:pointer; /* 鼠标样式为手形 */
}
</style>
<body>

```

3. 制作登录表单

在许多网站中都有登录表单的应用,而登录表单所包含的元素通常有用户名文本域、密码域、登录按钮和注册按钮等,这些元素是根据网站的实际需求而确定的。

【例 6.23】 使用 CSS 制作登录表单,本例浏览效果如图 6-23 所示。



图 6-23 使用 CSS 制作登录表单的效果图

代码如下:

```

<!-- 示例 6.23 程序 -->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>登录表单的制作</title>
  </head>
  <style type="text/css">
    .login { /* 登录表单容器的样式 */
      margin:0 auto; /* 容器水平居中对齐 */
      width:280px;
      padding:14px;

```

```

border:dashed 2px #b7ddf2;          /* 2px 虚线淡蓝色边框 */
background:#ebf4fb;
}
.login * {                          /* 容器中所有元素的样式 */
margin:0;
padding:0;
font-family: 宋体;
font-size:12px;
line-height:1.5em;
}
.login h2 {                          /* 容器中 2 级标题的样式 */
text-align:center;                 /* 文字水平居中对齐 */
font-size:18px;
font-weight:bold;                  /* 字体加粗 */
margin-bottom:10px;
padding-bottom:5px;
border-bottom:solid 1px #b7ddf2; /* 下边框为 1px 实线淡蓝色边框 */
}
.login .content {                   /* 容器内容区域的样式 */
padding:5px;
}
.login .frm_cont {                  /* 内容区域的样式 */
margin-bottom:8px;
}
.login .username input,. login .password input { /* 用户名文本域和密码域的样式 */
width:180px;
height:18px;
padding:2px 0px 2px 18px;          /* 文本框左内边距 18px, 以便为背景图像预留显示
空间 */
border:solid 1px #aacfe4;          /* 1px 实线淡蓝色边框 */
}
.login .username input {            /* 用户名文本域背景图像 */
background:#fff url(images/username.JPG)no-repeat left center;
}
.login .password input {           /* 密码背景图像 */
background:#fff url(images/lock.JPG)no-repeat left center;
}
.login .btns {                      /* 按钮的样式 */
text-align:center;                 /* 文字水平居中对齐 */
}
</style>
<body>
<div class="login">
<h2>用户登录</h2>
<div class="content">

```


1. 前期准备

(1) 栏目目录结构。在栏目文件夹下创建文件夹 images 和 css，分别用来存放图像素材和外部样式表文件。

(2) 页面素材。将本页面需要使用的图像素材存放在文件夹 images 下。

(3) 外部样式表。在文件夹 css 下新建一个名为 style.css 的样式表文件。

2. 制作页面

CSS 文件代码如下：

```
/* 页面全局样式——父元素 */
* {
    margin:0;                /* 所有元素外边距为 0 */
    padding:0;              /* 所有元素内边距为 0 */
}
/* 设置页面整体样式 */
body {
    font-size:12px;        /* 文字大小为 12px */
    color:#333;           /* 深灰色文字 */
}
ol,ul {
    list-style:none;      /* 列表无修饰 */
}
img,a {
    border:0;             /* 列表无修饰 */
    text-decoration:none; /* 图像无边框 */
}
a {
    color:#333;          /* 链接无修饰 */
}
a:hover {
    color:#f00;          /* 设置悬停链接样式 */
}
/* 会员注册表单的样式(与登录表单的样式共享) */
.loginLogo {
    width:100%;          /* 注册页面中的网站标志样式 */
    border-bottom:#efefef 1px solid; /* 底部边框 1px,深灰色实线 */
}
.logoMid {
    width:1040px;        /* 顶部容器的样式 */
    margin:0 auto;       /* 容器水平居中对齐 */
}
.loginReg {
    height:30px;         /* 顶部注册提示的样式 */
}
```

```

        line-height: 30px;                                /* 行高等于设定的高度,内容垂直方向居中对
    齐 */
        text-align: right;                                /* 文本水平居中对齐 */
    }
    .loginReg a {                                        /* 顶部注册提示链接的样式 */
        color: #7bc144;                                  /* 绿色文字 */
    }
    .loginReg a: hover {                                 /* 顶部注册提示悬停链接的样式 */
        color: #f00;                                     /* 红色文字 */
    }
    .loginBox {                                          /* 注册内容区域的样式 */
        width: 1050px;
        margin: 30px auto;                               /* 上下边距为 30px,水平居中对齐 */
        position: relative; /* 相对定位 */
    }
    .regList {                                           /* 注册内容项的样式 */
        height: 35px;
        line-height: 35px;                               /* 行高等于设定的高度,内容垂直方向居中对
    齐 */
        margin-bottom: 30px;                             /* 上下边距 30px */
        position: relative;                               /* 相对定位 */
    }
    .regList label {                                     /* 注册内容项提示标签的样式 */
        float: left;                                     /* 向左浮动 */
        width: 105px;
        margin-right: 10px;                              /* 右外边距 10px */
        text-align: right;                               /* 文本水平右对齐 */
        color: #999;
    }
    .regList input {                                     /* 注册表单中 input 元素的样式 */
        margin: 0;                                       /* 外边距 0px */
        padding: 0;                                       /* 内边距 0px */
        width: 283px;
        height: 33px;
        border: 3px solid #373840;                       /* 1px 深红色实线边框 */
        background: #feffdf;                             /* 浅黄色背景 */
        padding-left: 3px;                                /* 左内边距 3px */
    }
    .regList .yanzheng {                                 /* 注册表单中验证码区域的样式 */
        width: 135px;
    }
    .regList img {                                       /* 注册表单中验证码图片的样式 */
        left: 260px;                                     /* 距离容器左侧为 260px */
    }

```

```

        position: absolute;                /* 绝对定位 */
    }
    .xieyi {                               /* 注册表单中注册协议的样式 */
        height: 30px;
        line-height: 30px;                /* 行高等于设定的高度,内容在垂直方向上居中对
齐 */
        font-size: 12px;
        padding-left: 115px;              /* 左内边距 115px */
    }
    .xieyi input {                         /* 注册表单中接受协议复选框的样式 */
        position: relative;               /* 相对定位 */
        top: 2px;                          /* 距离容器顶部为 2px */
    }
    .xieyi a {                             /* 注册表单中协议链接的样式 */
        color: #7BC144;                   /* 绿色文字 */
    }
    .reg {                                  /* 注册表单中注册按钮的样式 */
        padding-left: 115px;               /* 左内边距 115px */
        margin-top: 10px;                  /* 上外边距 10px */
    }
    .chengnuo {                            /* 注册表单中右侧的样式 */
        position: absolute;                /* 绝对定位 */
        right: 0;
        top: 0;
    }

```

网页结构文件的 HTML 代码如下:

```

<! DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>会员注册</title>
    <link type="text/css" href="CSS/style.css" rel="stylesheet"/>
  </head>
  <body style="background:#fff;">
    <div class="loginLogo">
    <div class="logoMid">
    <h1 class="logo" style="height:71px;padding-top:10px;">
    <a href="index.html">
    
    </a>

```

```
</h1>
<div class="loginBox">

<form action="#.html" method="get" class="reg">
<div class="regList">
<label><span class="red"> * </span>用户名</label>
<input type="text" /><span style="color:#999;">请输入邮箱/用户名/手机号</span>
</div>
<div class="regList">
<label><span class="red"> * </span>请设置密码</label>
<input type="text" />
</div>
<div class="regList">
<label><span class="password"> * </span>请确认密码</label>
<input type="text" />
</div>
<div class="regList">
<label><span class="red"> * </span>验证码</label>
<input type="text" class="yanzheng" />

</div>
<div class="xieyi">
<input type="checkbox" />
我已经阅读并同意<a href="#">商城用户注册协议</a>
</div>
<div class="reg">
<input type="image" src="images/reg.jpg" />
</div>
</form>
<div class="clears"></div>
</div>
</div>
</div>
</body>
</html>
```

本章习题

1. 使用 CSS 对页面网页元素加以修饰，制作“旅游攻略”网站。图 6-25 供参考。

第 7 章 利用 CSS 和多媒体美化页面

学习目标

1. 了解 CSS 对网页美化的重要性；
2. 掌握常用链接、列表、表格对网页的修饰方法；
3. 学会多种多媒体标签的添加和使用。

知识要点

1. 文字链接、按钮链接和背景链接的应用；
2. 列表类型、列表图像、列表位置和复合列表样式的应用；
3. 美化表格的属性设置；
4. 多媒体标签的使用。

通过对网页中元素、布局和色彩的合理设计，可以使网页达到较好的视觉效果。网页中常用超链接、列表和表格等网页元素，所以，对这些元素进行美化，可以大大提高网站的质量。尤其是多媒体元素的加入，使网页更加丰富多彩。

7.1 CSS 链接的美化

在前面的章节中，已经学习了超链接的简单知识。知道超链接是一种允许其他网页或站点之间进行连接的元素，各个网页链接在一起后，才能真正构成一个网站。但是默认的连接样式往往达不到设计者的要求，所以可以通过对超链接的 CSS 样式设置，来达到理想的视觉效果。

7.1.1 文字链接的美化

在 HTML5 中，`<a>` 标签始终定义超链接，用于从一张页面链接到另一张页面。`<a>` 元素最重要的属性是 `href` 属性，它指示链接的目标，如果未设置 `href` 属性，后续多个属性均无法使用，则只是超链接的占位符。

在谷歌浏览器中，鼠标悬停链接时无明显效果，其余链接的默认外观如图 7-1 所示。

设计者为了改变这种最基本的超链接样式，利用前面章节所讲的伪类选择器，按照“a: link→a: visited→a: hover→a: active→”的顺序，设置不同的链接访问状态效果，为文本超链接添加复杂而多样的样式。



图 7-1 链接的默认样式

【例 7.1】 设置文字链接的外观。

```
<! --示例 7.1 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 7.1</title>
    <style type="text/css">
      #menu {
        text-align:center;                /* div 内的元素水平居中 */
      }
      a {
        margin:10px;                      /* 链接元素外边距为 10 像素 */
      }
      a:link {                             /* 链接未被访问 */
        font-size:20px;
      }
      a:hover {                            /* 鼠标悬停链接 */
        font-size:18px;
        color:red;
        text-decoration:overline underline;
      }
      a:active {                           /* 链接活动 */
        font-size:20px;
        color:green;
        text-decoration:none;
      }
    </style>
```

```
</head>
<body>
  <div id="menu">
    <h2>服务中心菜单栏</h2>
    <a href="index.html1" target="_blank">加入我们</a>
    <a href="index.html2" target="_blank">媒体报道</a>
    <a href="index.html3" target="_blank">官方微博</a>
    <a href="index.html4" target="_blank">帮助中心</a>
  </div>
</body>
</html>
```

例 7.1 程序中, 链接未被访问时, 设置文本大小为 20 像素, 文本颜色为默认的蓝色, 浏览效果如图 7-2 (a) 所示。

为了实现文本的跳跃效果, 一般会把不同状态下的文本大小设置不同, 所以, 鼠标悬停链接时, 设置文本大小为 18 像素, 文本颜色为红色, 文本描述同时有上划线与下划线, 浏览效果如图 7-2 (b) 所示。

链接活动时, 文本大小还原为 20 像素, 文本颜色为绿色, 文本无描述, 浏览效果如图 7-2 (c) 所示。



(a) 链接未被访问



(b) 鼠标悬停链接



(c) 链接活动

图 7-2 例 7.1 浏览效果图

7.1.2 按钮链接的美化

为了让链接外观看起来像按钮特效，设计者会为链接加上不同的边框阴影和相同的背景颜色加以实现。

【例 7.2】 设置按钮链接的外观。

```
<! -- 示例 7.2 程序 -->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 7.2</title>
    <style type="text/css">
      #menu {
        text-align:center;                /* div 内的元素水平居中 */
      }
      a {
        font-size:20px;
        font-weight:700;
        text-decoration:none;
        background-color:lightcyan;
        color:red;
        margin:5px;
        padding:10px 15px;
        border-radius:30px;
      }
      a:link,a:visited {                  /* 链接未被访问以及访问过后 */
        box-shadow:-5px -5px 10px black;
      }
      a:hover {                          /* 鼠标悬停链接 */
        font-size:26px;
      }
      a:active {                          /* 链接活动 */
        font-size:20px;
        box-shadow:6px 6px 10px black;
      }
    </style>
  </head>
  <body>
    <div id="menu">
      <h2>服务中心菜单栏</h2>
      <a href="#" target="_blank">加入我们</a>
      <a href="#" target="_blank">媒体报道</a>
```

```

<a href="#" target="_blank">官方微博</a>
<a href="#" target="_blank">帮助中心</a>
</div>
</body>
</html>

```

本例程序中，先把链接的文本样式统一设置为大小 20 像素、加粗、无描述、背景颜色为淡蓝色、颜色为红色、外边距 5 像素、上下内边距 10 像素、左右内边距 15 像素，和圆角边框等。

当链接未被访问时以及访问过后，边框设置为左上角的黑色阴影，浏览效果如图 7-3 (a) 所示。

鼠标悬停时，因文本大小变大为 26 像素，所以按钮整体变大，浏览效果如图 7-3 (b) 所示。

活动链接时，边框设置为右下角的黑色阴影，浏览效果如图 7-3 (c) 所示。



(a) 链接未被访问以及访问过后



(b) 鼠标悬停链接



(c) 链接活动

图 7-3 例 7.2 浏览效果图

7.1.3 背景链接的美化

除了文字链接和按钮链接，设计者还会为链接设置不同的背景颜色或背景图片，实现背景链接。

【例 7.3】 背景颜色链接。

```
<! --示例 7.3 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 7.3</title>
    <style type="text/css">
      #menu {
        text-align:center;           /* div 内的元素水平居中 */
      }
      a {
        font-size:20px;
        text-decoration:none;
        padding:10px 15px;
        margin:-3px;
      }
      a:link,a:visited {           /* 链接未被访问以及访问过后 */
        background-color:lightgrey;
        color:red;
      }
      a:hover {                   /* 鼠标悬停链接 */
        background-color:brown;
        color:white;
      }
      a:active {                  /* 链接活动 */
        font-weight:700;
        color:lawngreen;
      }
    </style>
  </head>
  <body>
    <div id="menu">
      <h2>服务中心菜单栏</h2>
      <a href="#" target="_blank">加入我们</a>
      <a href="#" target="_blank">媒体报道</a>
      <a href="#" target="_blank">官方微博</a>
```

```

<a href="#" target="_blank">帮助中心</a>
</div>
</body>
</html>

```

本例程序中，先把链接的文本样式统一设置为大小 20 像素、无描述、上下内边距 10 像素、左右内边距 15 像素，为了链接与链接之间没有空隙，设置链接的外边距为 -3 像素等。

当链接未被访问时以及访问过后，链接的背景颜色是浅灰色，文本颜色是红色，浏览效果如图 7-4 (a) 所示。

鼠标悬停时，链接的背景颜色是棕色，文本颜色是白色，浏览效果如图 7-4 (b) 所示。

活动链接时，链接的背景颜色不变，文本颜色变为绿色，并文本加粗，浏览效果如图 7-4 (c) 所示。



(a) 链接未被访问以及访问过后



(b) 鼠标悬停链接



(c) 链接活动

图 7-4 例 7.3 浏览效果图

在示例程序例 7.3 的基础上，把链接未被访问时和鼠标悬停时的背景颜色分别改为不同的背景图片，可得到不一样的链接风格，如图 7-5 (a) (b) (c) 所示。代码如下：

```

<style type="text/css">
a:link,a:visited{
background-image:url(img/menu_1.jpg);
/* 链接未被访问以及访问过后 */

```



```
color:red;
}
a:hover{ /* 鼠标悬停链接 */
background-image:url(img/menu_2.jpg);
color:white;
}
</style>
```



(a) 链接未被访问以及访问过后



(b) 鼠标悬停链接



(c) 链接活动

图 7-5 例 7.4 浏览效果图

7.2 CSS 列表的美化

在前面的章节中，已经学习了列表的简单知识。列表元素是网页设计中常见的元素，它是 HTML 中组织多个段落文本的一种方式，可以使信息显示结构清晰、直观，便于理解，多



用于新闻、产品，或者是其他内容的罗列等。由于列表如此重要，所以，当引入 CSS 后，设计者就迫切希望，能利用列表许多新的属性，开发出更加丰富的列表样式来。

在 CSS 样式中，主用是通过 list-style-type、list-style-image 和 list-style-position 这 3 个属性改变列表修饰符的类型，常用的 CSS 列表属性见表 7-1。

表 7-1 常用的 CSS 列表属性

| 属性 | 说明 |
|---------------------|----------------------|
| list-style | 复合属性，在一个声明中设置所有的列表属性 |
| list-style-type | 设置列表项标记的类型 |
| list-style-image | 使用图像来替换列表项的标记 |
| list-style-position | 设置在何处放置列表项标记 |

7.2.1 列表项类型 (list-style-type)

此属性用来设置或检索对象的列表项所使用的标记类型。若 list-style-image 属性值为 none 或指定 url 地址的图片不能被显示时，此属性将发生作用。常用的 list-style-type 属性值见表 7-2。

表 7-2 常用的 list-style-type 属性值

| 属性值 | 说明 | 属性值 | 说明 |
|-------------|------------|-------------|-----------|
| disc | 默认值，标记是实心圆 | circle | 标记是空心圆 |
| square | 标记是实心方块 | decimal | 标记是阿拉伯数字 |
| lower-roman | 标记是小写罗马数字 | upper-roman | 标记是大写罗马数字 |
| lower-alpha | 标记是小写英文字母 | upper-alpha | 标记是大写英文字母 |
| none | 无列表项标记 | | |

【例 7.4】各种列表标记。

```
<! -- 示例 7.4 程序 -->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 7.4</title>
    <style type="text/css">
      div{
        width:110px;
        height:130px;
        border:1px solid red;
```

```
float:left;
}
ol. one {
    list-style-type:decimal;    /* 标记阿拉伯数字 */
}
ol. two {
    list-style-type:lower-roman; /* 标记小写罗马数字 */
}
ol. three {
    list-style-type:upper-roman; /* 标记大写罗马数字 */
}
ol. four {
    list-style-type:lower-alpha; /* 标记小写英文字母 */
}
ol. five {
    list-style-type:upper-alpha; /* 标记大写英文字母 */
}
ol. six {
    list-style-type:square;     /* 标记方块 */
}
li. circle {
    list-style-type:circle;     /* 标记空心圆 */
}
</style>
</head>
<body>
<center><h2>食品分类</h2></center>
<div>
<h3>饮料</h3>
<ol class="one">
    <li>咖啡</li>
    <li>茶</li>
    <li>可口可乐</li>
</ol>
</div>
<div>
<h3>甜品</h3>
<ol class="two">
    <li>蛋糕</li>
```

```
        <li>糖水</li>
        <li>奶酪</li>
    </ol>
</div>
<div>
    <h3>水果</h3>
    <ol class="three">
        <li>梨子</li>
        <li>苹果</li>
        <li>猕猴桃</li>
    </ol>
</div>
<div>
    <h3>蔬菜</h3>
    <ol class="four">
        <li>茄子</li>
        <li>南瓜</li>
        <li>大白菜</li>
    </ol>
</div>
<div>
    <h3>主食</h3>
    <ol class="five">
        <li>米饭</li>
        <li>饺子</li>
        <li>面条</li>
    </ol>
</div>
<div>
    <h3>调料</h3>
    <ol class="six">
        <li>八角</li>
        <li class="circle">麻油</li>
        <li>胡椒粉</li>
    </ol>
</div>
</body>
</html>
```

本例浏览效果如图 7-6 所示。



图 7-6 各种类型标记的页面显示

7.2.2 列表项图像 (list-style-image)

此属性用来设置或检索对象的列表项标记的图像。常用的 list-style-image 属性值见表 7-3。

表 7-3 常用的 list-style-image 属性值

| 属性值 | 说明 |
|---------|-------------|
| none | 默认值, 无图形被显示 |
| url | 图像的路径 |
| inherit | 规定从父元素继承值 |

【例 7.5】图像作列表项标记。

```
<! --示例 7.5 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 7.5</title>
    <style type="text/css">
      ol{
        list-style-image:url(img/list3.jpg);
        list-style-type:upper-alpha;
      }
      li.none{
        list-style-image:none;          /* 设置列表项图像为不显示 */
      }
    </style>
```

```

</head>
<body>
  <h3>护肤品系列</h3>
  <ol>
    <li>洗面奶</li>
    <li>柔肤水</li>
    <li class="none">保湿霜</li>
    <li>眼部精华</li>
    <li class="none">防晒乳</li>
  </ol>
</body>
</html>

```

本例浏览效果如图 7-7 所示。



图 7-7 图像列表项标记的页面显示

以上示例中列表项标记用图片来显示。设置列表项图片时，通常会多设置一个“list-style-type”属性以防图像不可用。

7.2.3 列表项位置 (list-style-position)

此属性用来设置对象的列表项标记相对于列表项内容的位置。常用的 list-style-position 属性值见表 7-4。

表 7-4 常用的 list-style-position 属性值

| 属性值 | 说明 |
|---------|---|
| outside | 默认值，保持标记位于文本的左侧。列表项标记放置在文本以外，且环绕文本不根据标记对齐 |
| inside | 列表项标记放置在文本以内，且环绕文本根据标记对齐 |
| inherit | 规定从父元素继承值 |

【例 7.6】 列表项标记的位置设置。

```
<! --示例 7.6 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 7.6</title>
    <style type="text/css">
      ol{
        list-style-position:outside;
      }
      ul{
        list-style-position:inside;
      }
      li{
        list-style-image:url(img/list2.jpg);
        border:1px solid black;
        width:100px;
      }
    </style>
  </head>
  <body>
    <h3>鲜花</h3>
    <ol>
      <li>玫瑰</li>
      <li>蔷薇</li>
      <li>牡丹</li>
    </ol>
    <h3>鲜花</h3>
    <ul>
      <li>玫瑰</li>
      <li>蔷薇</li>
      <li>牡丹</li>
    </ul>
  </body>
</html>
```

本例浏览效果如图 7-8 (a) 所示。



图 7-8 例 7.6 浏览效果图

以上示例中，上面的列表项标记位于文本内容左侧外边，下面的列表项标记位于文本内容左侧内边。

如果在示例程序例 7.6 的基础上为标签添加代码“padding-left: 30px;”，浏览效果如图 7-8 (b) 所示，发现下面的列表项环绕文本根据标记对齐，上面的列表项则否。

7.2.4 复合列表样式 (list-style)

这个属性是设定列表项样式的一个快捷的综合写法。用这个属性可以同时设置列表项类型属性 (list-style-type)，列表项图片属性 (list-style-image) 和列表项位置属性 (list-style-position)。

语法：list-style: list-style-type | list-style-position | list-style-image

【例 7.7】复合的列表样式。

```
<! --示例 7.7 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 7.7</title>
    <style type="text/css">
      ul {
        list-style:square inside url(img/list1.jpg);
      }
    </style>
  </head>
  <body>
    <h3>饮料</h3>
    <ul>
```



```

        <li>咖啡</li>
        <li>茶</li>
        <li>可口可乐</li>
    </ul>
</body>
</html>

```

本例浏览效果如图 7-9 (a) 所示。



图 7-9 例 7.7 浏览效果图

以上示例中用了复合列表样式，方块和图片同时存在时，只显示图片，当指定的图片不可用时才会显示方块。

`list-style` 属性可以不设置其中的某个值，比如“`list-style: square url (img/list1.jpg);`”也是允许的。未设置的属性会使用其默认值，如图 7-9 (b) 所示。

7.2.5 利用背景图像实现列表项标记

虽然使用 `list-style-image` 属性很容易实现设置列表项图像的目的，但这个属性不能够精确控制图像替换的项目符号距离文字的位置。因此，可以利用 `background-image` 加以实现。

【例 7.8】 列表样式。

```

<! --示例 7.8 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 7.8</title>
    <style type="text/css">
      li {
        list-style-type:none;           /* 设置列表项类型为不显示 */
        background:url(img/list4.jpg) no-repeat; /* 设置列表项背景图片,并不重复 */
        padding-left:30px;             /* 设置文本左内边距 30 像素 */
        padding-bottom:10px;          /* 设置文本下内边距 10 像素 */
      }
    </style>

```

```

</head>
<body>
  <h3>快递公司</h3>
  <ol>
    <li>顺丰快递</li>
    <li>圆通速递</li>
    <li>百世汇通</li>
    <li>韵达快递</li>
  </ol>
</body>
</html>

```

本例浏览效果如图 7-10 所示。



图 7-10 背景图像代替列表项标记

7.3 CSS 表格的美化

常用网页的布局形式是 DIV+CSS，但并不是所有的布局都是如此，必要时，表格的使用对零散内容的汇总分类更为便利。所以，尽管在前面的章节中已经学习了表格的基本用法，但是设计者需要利用 CSS 设置表格样式的多样化，进而美化表格的外观。常用的 CSS 表格属性见表 7-5。

表 7-5 常用的 CSS 表格属性

| 属性 | 说明 |
|-----------------|-------------|
| border-collapse | 设置单元格边框是否合并 |
| border-spacing | 设置单元格的间距 |
| caption-side | 设置表格标题的位置 |
| empty-cells | 对空单元格的处理方式 |

7.3.1 border-collapse

border-collapse 属性设置表格的边框是否被合并为一个单一的边框，还是按照标准的HTML 中那样分开显示。

语法：border-collapse: separate | collapse | inherit

参数：separate 是默认值，边框分开不合并，不会忽略 border-spacing 和 empty-cells 属性。collapse 边框合并，如果相邻，则共用一个边框，会忽略 border-spacing 和 empty-cells 属性。Inherit 是从父元素继承值。

【例 7.9】 表格边框设置。

```
<! --示例 7.9 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 7.9</title>
    <style type="text/css">
      table {
        border-collapse: collapse;
      }
      table, td, th {
        border: 1px solid black;
      }
    </style>
  </head>
  <body>
    <table>
      <tr>
        <th>课程</th><th>学分</th><th>考试形式</th>
      </tr>
      <tr>
        <td>SQL 数据库技术</td><td>4.0</td><td>笔试</td>
      </tr>
      <tr>
        <td>PHP 程序设计</td><td>6.0</td><td>机考</td>
      </tr>
    </table>
  </body>
</html>
```

本例浏览效果如图 7-11 (a) 所示。



图 7-11 例 7.9 浏览效果图

以上示例中属性 border-collapse 取值 collapse, 所以表格的边框被合并为一个单一的边框。

如果属性 border-collapse 取值 separate 或者不设置, 表格将分别有各自的边框, 如图 7-11 (b) 所示。

7.3.2 border-spacing

border-spacing 属性作用等同于标签属性 cellspacing, 即当表格边框独立时 (border-collapse 属性等于 separate), 相邻单元格边框之间的横向和纵向的间距。

语法: border-spacing: length

参数: length 由数字和单位组成的长度值, 不允许负值。如果提供全部两个 length 值时, 第一个作用于横向间距, 第二个作用于纵向间距。如果只提供一个 length 值时, 这个值将同时作用于横向和纵向上的间距。

【例 7.10】表格的间距设置。

```
<! -- 示例 7.10 程序 -->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 7.10</title>
    <style type="text/css">
      .one {
        border-collapse: separate;
        border-spacing: 10px;
      }
      .two {
        border-collapse: separate;
        border-spacing: 10px 30px;
      }
    </style>
  </head>
  <table border="1">
    <tr>
      <td class="one">课程</td>
      <td class="one">学分</td>
      <td class="one">考试形式</td>
    </tr>
    <tr>
      <td class="one">SQL数据库技术</td>
      <td class="one">4.0</td>
      <td class="one">笔试</td>
    </tr>
    <tr>
      <td class="one">PHP程序设计</td>
      <td class="one">6.0</td>
      <td class="one">机考</td>
    </tr>
  </table>
</html>
```

```
border:1px solid black;
}
</style>
</head>
<body>
  <table class="one">
    <tr>
      <td>大数据</td><td>物联网</td>
    </tr>
    <tr>
      <td>云计算</td><td>人工智能</td>
    </tr>
  </table>
  <br>
  <table class="two">
    <tr>
      <td>大数据</td><td>物联网</td>
    </tr>
    <tr>
      <td>云计算</td><td>人工智能</td>
    </tr>
  </table>
</body>
</html>
```

本例浏览效果如图 7-12 所示。



图 7-12 不同表格间距显示

以上示例中，表格的相邻单元格之间，间距会合并，并取两者之间的最大值。



7.3.3 caption-side

caption-side 属性设置或检索表格标题在表格的哪一边，此属性必须和表格的 caption 标签同时使用。

语法：caption-side: top | right | bottom | left

参数：top 是默认值，指定标题在表格上边。right 指定标题在表格右边。bottom 指定标题在表格下边。left 指定标题在表格左边。

【例 7.11】表格的标题设置。

```
<!-- 示例 7.11 程序 -->
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 7.11</title>
    <style type="text/css">
      .cap {
        caption-side: bottom;
      }
      table, th, td {
        border: 1px solid black;
      }
    </style>
  </head>
  <body>
    <table class="cap">
      <caption><h2>值班表</h2></caption>
      <tr>
        <th>时间</th><th>值日生</th>
      </tr>
      <tr>
        <td>08:00-12:00</td><td>张红敏</td>
      </tr>
      <tr>
        <td>14:00-17:00</td><td>李凯全</td>
      </tr>
      <tr>
        <td>19:00-22:00</td><td>曾天艺</td>
      </tr>
    </table>
  </body>
</html>
```

本例浏览效果如图 7-13 所示。



图 7-13 各种外边距的设置

以上示例中，表格的标题设置在表格的下边。

7.3.4 empty-cells

empty-cells 属性设置或检索当表格的单元格无内容时，是否显示该单元格的边框。只有当表格边框独立时（即 border-collapse 属性等于 separate）此属性才起作用。

语法：empty-cells: show | hide

参数：show 是默认值，指定当表格的单元格无内容时，显示该单元格的边框。hide 指定当表格的单元格无内容时，隐藏该单元格的边框。

【例 7.12】空单元格的边框设置。

```
<! --示例 7.12 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 7.12</title>
    <style type="text/css">
      .emp{
        border-collapse:separate;
        empty-cells:hide;
      }
      table,th,td{
        border:1px solid black;
      }
    </style>
  </head>
  <body>
    <table class="emp">
      <tr>
        <td>有内容的单元格</td><td>有内容的单元格</td>
```

```

</tr>
<tr>
  <td>有内容的单元格</td><td></td>
</tr>
<tr>
  <td>有内容的单元格</td><td>有内容的单元格</td>
</tr>
</table>
</body>
</html>

```

本例浏览效果如图 7-14 所示。



图 7-14 空单元格边框隐藏

以上示例中第二行第二列的单元格没有内容，所以当 `empty-cells` 取值 `hide` 时，此单元格的边框将被隐藏不可见。

7.4 多媒体的添加与美化

对于很多网站而言，经常会涉及到一些动画或者是视频、音频之类的内容，设计得合理，这些元素就能够很好地突出主题，如此，网页浏览者在视觉和听觉上都会有空间联系和层次丰富的感觉，从而产生愉悦的感受。

早期的网页大多是由文字或者图像构成的，由于多媒体技术的发展，音乐、动画、视频等媒体的应用越来越广泛。音乐网站、电影网站、播客等融合多媒体技术的网站越来越多。

本节将简单介绍 `<embed>`、`<bgsound>`、`<audio>`、`<video>` 标签在网页中的应用。

7.4.1 `<embed>` 标签的使用

利用 `<embed>` 标签可以在网页中插入各种类型的多媒体文件，如 WMV、MP3、AVI、MP4、SWF、MOV、RMVB 等。但是在使用 `<embed>` 标签之前，需要安装相应的插件，否则多媒体文件不能正常播放。常用的 `embed` 属性见表 7-6。

表 7-6 常用的 embed 属性

| 属性 | 说明 |
|--------|-----------------|
| src | 必选项，定义嵌入内容的 URL |
| type | 定义嵌入内容的类型 |
| width | 设置嵌入内容的宽度 |
| height | 设置嵌入内容的高度 |

【例 7.13】 利用<embed>标签插入 Flash 动画文件。

```
<! --示例 7.13 程序-->
```

```
<! DOCTYPE html>
```

```
<html>
```

```
  <head>
```

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

```
    <title>示例 7.13</title>
```

```
  </head>
```

```
  <body>
```

```
    <center>
```

```
      <h2>三月祝福</h2>
```

```
      <embed src="media/阳春三月.swf" width="390" height="350"></embed>
```

```
    </center>
```

```
  </body>
```

```
</html>
```

本例浏览效果如图 7-15 所示。



图 7-15 Flash 文件的显示效果

以上示例在网页中显示一个 Flash 动画。

【例 7.14】 利用<embed>标签插入 MP4 视频文件。

```
<!-- 示例 7.14 程序 -->
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 7.14</title>
  </head>
  <body>
    <center>
      <h2>第五空间</h2>
      <embed src="media/第五空间.mp4" width="600px" height="400px"></embed>
    </center>
  </body>
</html>
```

本例浏览效果如图 7-16 所示。



图 7-16 MP3 文件的显示效果

【例 7.15】 利用<embed>标签插入 MP3 音乐文件。

```
<!-- 示例 7.15 程序 -->
<!DOCTYPE html>
<html>
```

```

<head>
  <meta charset="utf-8" />
  <title>示例 7.15</title>
</head>
<body>
  <center>
    <h2>听! 是谁在唱歌</h2>
    <embed src="media/听! 是谁在唱歌 . mp3" height="200"></embed>
  </center>
</body>
</html>

```

本例浏览效果如图 7-17 所示。



图 7-17 MP3 文件的显示效果

7.4.2 <bgsound>标签的使用

<bgsound>是 IE 浏览器中设置网页背景音乐的元素，利用它可以将音乐文件嵌入到网页中又不占页面空间，可以播放的声音文件格式包括 WAV、AU、MIDI、MP3 等。

<bgsound>标签参数设置不多，常用的 bgsound 属性见表 7-7。

表 7-7 常用的 bgsound 属性

| 属性 | 说明 |
|---------|--------------------------------------|
| src | 必选项，定义嵌入内容的 URL |
| loop | 该属性表明音频被播放的次数，取值“-1”为无限循环 |
| balance | 该属性取值在-10000 到+10000，它决定扬声器之间的音量如何分配 |
| volume | 该属性取值在-10000 到 0，它决定音频的音量大小 |
| delay | 该属性设置音频的播放延时 |



【例 7.16】 背景音乐的设置。

```
<!--示例 7.16 程序-->
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 7.16</title>
  </head>
  <body>
    <center>
      <h2>童话镇--背景音乐</h2>
      <bgsound src="media/童话镇.mp3" loop="-1"></bgsound>
    </center>
  </body>
</html>
```

由于<bgsound>标签的非标准特性，所以设置网页背景音乐时，一些主流的浏览器常用的方法还是使用<embed>标签。

7.4.3 HTML5 新增的多媒体标签

以前大多数音频、视频是通过插件（例如 Flash）来播放的，然而并非所有的浏览器都拥有同样的插件。

HTML5 新增了 audio 和 video 元素，分别用来插入声音和视频。至于格式，只要使用支持 HTML5 的浏览器就可以了，不需要使用其他的任何插件。<audio>和<video>标签的属性大致相同，见表 7-8。

表 7-8 常用的<audio>和<video>标签属性

| 属性 | 说明 |
|----------|--|
| autoplay | 定义音频或视频在就绪后马上播放 |
| controls | 定义控件的显示（比如播放/暂停按钮） |
| loop | 定义音频或视频结束时是否重新开始播放 |
| preload | 规定当网页加载时，音频或视频是否默认被加载以及如何被加载。如果使用“autoplay”，则忽略该属性 |
| src | 必选项，定义音频或视频的 URL |
| height | 设置视频播放器的高度 |
| width | 设置视频播放器的宽度 |

1. <audio>标签

<audio>标签定义声音，比如音乐或其他音频流，此标签默认是隐藏的。

【例 7.17】 单个音频的嵌入。

```
<! --示例 7.17 程序-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>示例 7.17</title>
  </head>
  <body>
    <center>
      <h2>audio 声音文件</h2>
      <audio src="media/铃铛.wav" controls="controls">您的浏览器不支持 audio 元素。</
audio>
    </center>
  </body>
</html>
```

本例浏览效果如图 7-18 所示。



图 7-18 音频的应用

以上示例中，使用 `controls` 属性添加播放控件，包含播放、暂停和音量调节按钮等。由于 IE8 或更早版本的 IE 浏览器不支持 `<audio>` 标签，所以可以在 `<audio>` 和 `</audio>` 之间放置文本内容，这些文本信息将会被显示在那些不支持 `<audio>` 标签的浏览器中。

`<audio>` 标签允许包含多个 `<source>` 标签，`<source>` 标签可以链接不同的音频文件，浏览器将使用第一个可识别的格式。代码如下：

```
<audio controls="controls">
  <source src="media/Easy.ogg" type="audio/ogg"></source>
  <source src="media/铃铛.mp3" type="audio/mp3"></source>
  您的浏览器不支持 audio 元素。
</audio>
```

2. `<video>` 标签

`<video>` 标签定义视频，例如电影片段或其他视频流。`<video>` 标签在页面中嵌入视频的代码格式和 `<audio>` 标签的使用相似。

【例 7.18】 视频嵌入网页。

```
<! --示例 7.18 程序-->
```

```
<! DOCTYPE html>
```

```
<html>
```

```
  <head>
```

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

```
    <title>示例 7.18</title>
```

```
  </head>
```

```
  <body>
```

```
    <center>
```

```
      <h2>video 视频文件</h2>
```

```
      <video src="media/movie. ogg" controls="controls">您的浏览器不支持 video 元素。
```

```
    </video>
```

```
    </center>
```

```
  </body>
```

```
</html>
```

或者

```
<video width="500" height="500" controls="controls" >
```

```
  <source src="media/movie. ogg" type="video/ogg"></source>
```

```
  <source src="media/第五空间. mp4" type="video/mp4"></source>
```

```
  您的浏览器不支持 video 元素。
```

```
</video>
```

本例浏览效果如图 7-19 所示。

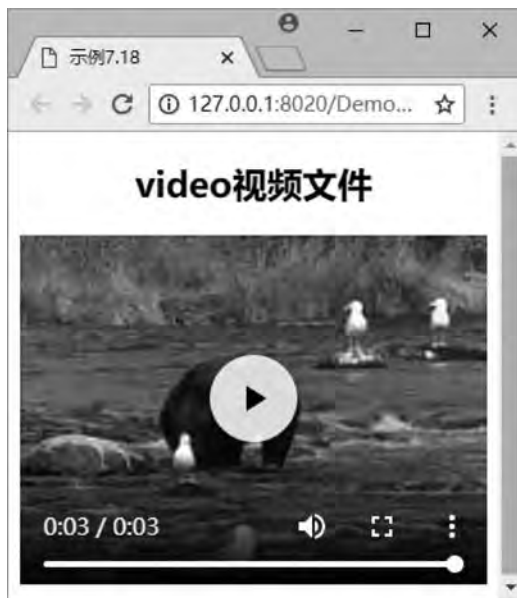


图 7-19 视频的应用

7.5 综合案例——海洋旅游胜地

本章综合前面学习的 CSS 和多媒体知识，可以利用链接、列表、表格和多媒体制作并美化网页。本章项目页面布局要求如下：

页面要求有外层的 div-all，内层按照顺序嵌套 4 块区域，分别为 div-top、div-menu、div-left 和 div-right，如图 7-20 所示。

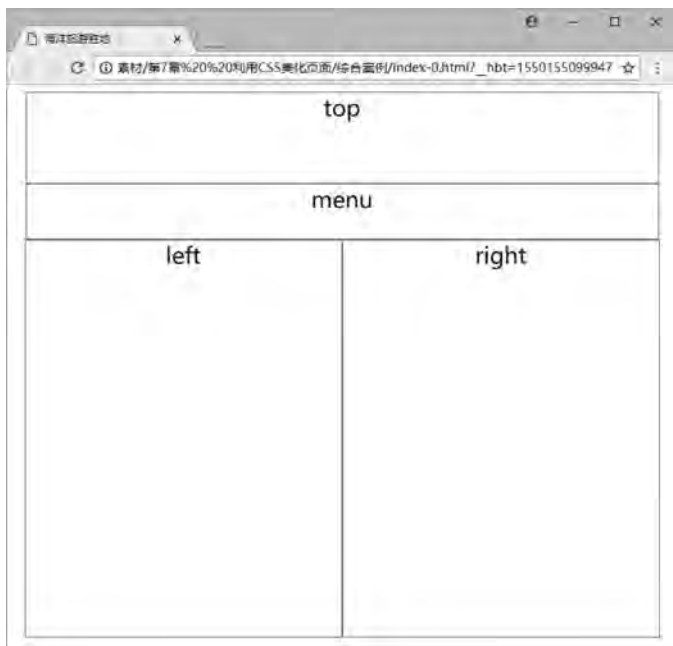


图 7-20 网页布局结构图

从图 7-20 可以看出 div-left 和 div-right 是经过浮动后 (float: left;)，处于同一行中。处理 div-all 的外边距 (margin: 0px auto;)，使整个页面的结构相对浏览器水平居中。最后对各个区域的高度和宽度做相应布置，布局代码如下：

```
<!--综合案例-布局.html-->
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>海洋旅游胜地</title>
    <style type="text/css">
      .all {
        margin:0px auto;
        width:704px;
        height:600px;
```

```
    }  
    .top {  
        width:700px;  
        height:100px;  
        border:1px solid red  
    }  
    .menu {  
        width:700px;  
        height:60px;  
        border:1px solid red  
    }  
    .left,.right {  
        width:350px;  
        height:440px;  
        float:left;  
        border:1px solid red;  
    }  
}   
</style>  
</head>  
<body>  
    <div class="all">  
        <div class="top">top</div>  
        <div class="menu">menu</div>  
        <div class="left">left</div>  
        <div class="right">right</div>  
    </div>  
</body>  
</html>
```

布局完成后，通过各种图文信息的输入，并设置相关 CSS 样式。

(1) 为 div-top 添加 logo 图片。代码如下：

```
<style>  
.top {  
    background-image:url(img/top.jpg)  
}  
</style>  
<body>  
<div class="top"></div>  
</body>
```

(2) 为 div-menu 添加按钮链接以制作导航菜单。代码如下：

```
<style>  
.menu {
```



```

        text-align:center;
    }
    a{
        font-size:13px;
        font-weight:700;
        text-decoration:none;
        background-color:lightcyan;
        color:red;
        margin:20px;
        padding:10px 15px;
        border-radius:10px;
    }
    a:link,a:visited{ /* 链接未被访问以及访问过后 */
        box-shadow:6px 6px 10px black;
    }
    a:hover{ /* 鼠标悬停链接 */
        font-size:14px;
    }
    a:active{ /* 链接活动 */
        font-size:13px;
        box-shadow:-5px -5px 10px black;
    }
</style>
<body>
<div class="menu">
    <br>
    <a href="#" target="_blank">交通路况</a>
    <a href="#" target="_blank">娱乐设施</a>
    <a href="#" target="_blank">美食特产</a>
    <a href="#" target="_blank">历史文化</a>
    <a href="#" target="_blank">注意事项</a>
</div>
</body>

```

(3) 为 div-left 上方添加图片列表以制作新闻动态。代码如下:

```

<style>
h3{
    color:brown;
}
ol{
    list-style-image:url(img/list2.jpg);
    list-style-type:upper-alpha;
}

```



```

</style>
<body>
<div class="left">
  <h3>新闻动态</h3>
  <ol>
    <li>英比奥山顶景区</li>
    <li>新加坡空中缆车</li>
    <li>天际线斜坡滑车</li>
    <li>圣淘沙名胜世界</li>
    <li>海洋馆和水上探险乐园</li>
  </ol>
</div>
</body>

```

(4) 为 div-left 下方添加宣传视频文件。代码如下:

```

<body>
<div class="left">
  <video src="media/shengtaosha.ogg" width="320px" height="250px" controls="controls">
  </video>
</div>
</body>

```

(5) 为 div-right 上方添加无框表格实现图片墙效果。代码如下:

```

<style>
table{
  border-collapse: separate;
  border-spacing: 20px;
}
</style>
<body>
<div class="right">
  <table>
    <tr>
      <td></td>
      <td></td>
    </tr>
    <tr>
      <td></td>
      <td></td>
    </tr>
  </table>
</div>
</body>

```

(6) 为 div-right 下方添加文字信息和音频文件。代码如下:

```

<style>

```

```

p {
    text-indent: 2em;
    line-height: 22px;
    font-weight: 700;
    color: brown;
}
</style>
<body>
<div class="right">
    <p>这里不过是个平凡的小岛,岛上居民过着简单质朴的生活。当新加坡政府正式收回这个小岛时,决定将它改造成一个休闲度假的胜地。</p>
    <br><br><br>
    <audio src="media/铃铛.mp3" controls="controls" loop="loop"></audio>
</div>
</body>

```

(7) 为 div-all 添加背景图片 (background-image: url (img/bg.jpg);)。

(8) 为 body 添加背景图片 (background-image: url (img/bg-0.jpg);)。

完成以上步骤,去除各个区域的边框,对高度和宽度稍作调整,综合案例最终网页浏览效果如图 7-21 所示。



图 7-21 网页浏览效果图



以下为完整代码，以供参考。

```
<! --综合案例.html-->
<! DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>海洋旅游胜地</title>
    <style type="text/css">
      body{
        background-image:url(img/bg-0.jpg);
      }
      .all{
        margin:0px auto;
        width:700px;
        height:600px;
        background-image:url(img/bg.jpg);
      }
      .top{
        width:700px;
        height:100px;
        background-image:url(img/top.jpg)
      }
      .menu{
        width:700px;
        height:60px;
        text-align:center;
      }
      .left,.right{
        width:350px;
        height:440px;
        float:left;
      }
      a{
        font-size:13px;
        font-weight:700;
        text-decoration:none;
        background-color:lightcyan;
        color:red;
        margin:20px;
```

```
padding:10px 15px;
border-radius:10px;
}
a:link,a:visited{ /* 链接未被访问以及访问过后 */
box-shadow:6px 6px 10px black;
}
a:hover{ /* 鼠标悬停链接 */
font-size:14px;
}
a:active{ /* 链接活动 */
font-size:13px;
box-shadow:-5px -5px 10px black;
}
h3{
color:brown;
}
ol{
list-style-image:url(img/list2.jpg);
list-style-type:upper-alpha;
}
table{
border-collapse:separate;
border-spacing:20px;
}
p{
text-indent:2em;
line-height:22px;
font-weight:700;
color:brown;
}
</style>
</head>
<body>
<div class="all">
<div class="top"></div>
<div class="menu">
<br>
<a href="#" target="_blank">交通路况</a>
<a href="#" target="_blank">娱乐设施</a>
```



```
<a href="#" target="_blank">美食特产</a>
<a href="#" target="_blank">历史文化</a>
<a href="#" target="_blank">注意事项</a>
</div>
<div class="left">
  <h3>新闻动态</h3>
  <ol>
    <li>英比奥山顶景区</li>
    <li>新加坡空中缆车</li>
    <li>天际线斜坡滑车</li>
    <li>圣淘沙名胜世界</li>
    <li>海洋馆和水上探险乐园</li>
  </ol>
  <video src="media/shengtaosha.ogg" width="320px" height="250px" controls="controls">
</video>
</div>
<div class="right">
  <table>
    <tr>
      <td></td>
      <td></td>
    </tr>
    <tr>
      <td></td>
      <td></td>
    </tr>
  </table>
  <p>这里不过是个平凡的小岛,岛上居民过着简单质朴的生活。当新加坡政府正式收回这个
  小島时,决定将它改造成一个休闲度假的胜地。</p>
  <br><br><br>
  <audio src="media/铃铛.mp3" controls="controls" loop="loop"></audio>
</div>
</div>
</body>
</html>
```

本章习题

1. 利用 CSS 技术, 结合表格和列表, 制作并美化“翡翠阁”页面。图 7-22 供参考。

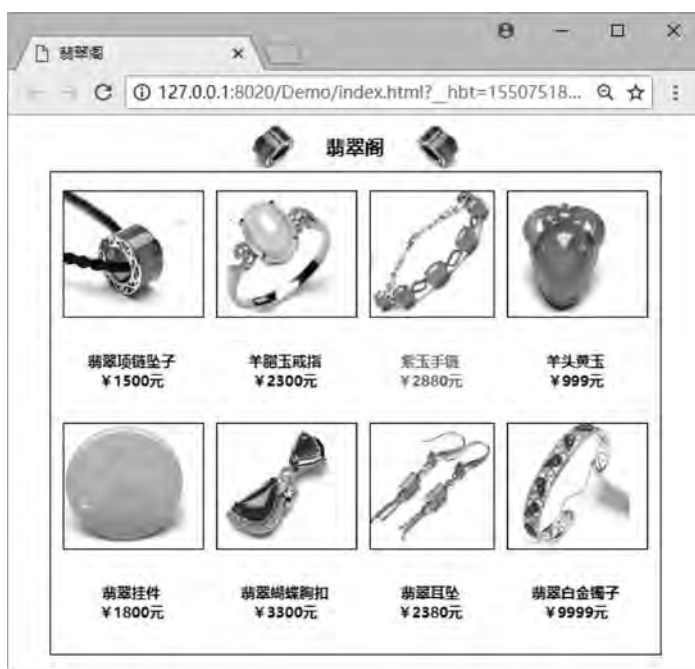


图 7-22 题 1 网页效果图

2. 用表格布局页面，利用 CSS 技术，及添加多媒体，制作并美化“心灵之音”页面。
图 7-23 供参考。



图 7-23 题 2 网页效果图

第 8 章 利用 CSS 制作导航菜单

学习目标

1. 了解导航菜单对于网站建设的重要性；
2. 了解导航菜单的种类和作用；
3. 掌握各种导航菜单的设计与制作方法。

知识要点

1. 水平导航菜单和纵向导航菜单的设计与实现；
2. 下拉子菜单和出式子菜单的设计与实现；
3. 底部导航菜单的固定、设计与实现。

导航对于一个网站来说，起到重要的引导作用。一个优秀的导航设计可以使网站结构清晰、有条理，让用户快速地找到所需，让用户清晰明了地了解到网站的结构框架，能起到重要的指引作用。

导航栏是网站设计中必不可少的一个模块。常见的网站导航有顶部水平导航、侧边纵向导航和底部导航等，区别于传统的表格布局，直接利用 CSS，结合链接元素和列表元素来制作导航菜单，将大大简化设计的流程。

8.1 水平顶部导航栏

水平菜单导航栏是网站设计中应用范围最广的导航设计，一般放置在页面的顶部。水平导航适用性强，几乎所有类型的网站都可以使用，设计难度较低。

如果导航过于普通，无法容纳复杂的信息结构，就需要在内容模块较多的情况下，结合下拉子导航使用。

8.1.1 简单水平导航栏的设计与实现

1. 导航栏的创建

<nav>标签是 HTML5 新增的文档结构标签，用于标记导航栏，以便后续与网站的其他内

容整合，所以常用<nav>标签在页面上创建导航栏菜单区域。

例如，在<nav>的首尾标签之间使用无序列表标签配合列表选项创建菜单选项。其中选项内容使用超链接的形式，链接地址当前设置为空链接，用户可根据实际情况改成具体的 URL 地址。代码如下：

```
<body>
<center><h3>简单水平菜单导航栏</h3></center>
<hr>
<nav>
  <ul>
    <li><a href="#">前端</a></li>
    <li><a href="#">后台</a></li>
    <li><a href="#">移动应用</a></li>
    <li><a href="#">数据库</a></li>
    <li><a href="#">操作系统</a></li>
    <li><a href="#">服务器</a></li>
  </ul>
</nav>
</body>
```

本段代码运行效果如图 8-1 所示。

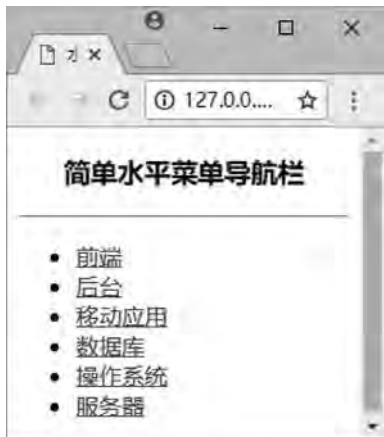


图 8-1 导航栏菜单的初步创建

由图 8-1 可见，导航菜单中的文字内容和超链接已经完成，用户单击不同的选项可以跳转至指定的页面。

2. 列表样式的设计

此时会带有元素的默认样式，即每个列表选项前面有实心点标记，可以利用 CSS 在<head>标签中进行声明，对列表样式进行改变。

例如，使用 list-style-type 属性去掉默认标记。为了预防不同的浏览器会出现预设值，设置 margin 和 padding 属性为 0，避免最终布局效果可能产生的误差；为了使列表元素能够

在同一行显示，为列表选项定义浮动效果。代码如下：

```
<style type="text/css" >
ul {
    margin:0;
    padding:0;
    list-style-type:none;
}
li {
    float:left;
}
</style>
```

添加本段代码后，运行效果如图 8-2 所示。



图 8-2 导航栏菜单的行内显示

由图 8-2 可见，列表已经去掉了实心点标记，并且每个列表项均在同一行显示。

3. 超链接样式的设计

利用 CSS 为超链接重新设置样式，例如，对超链接的 a: link 和 a: visited 进行设置，表示超链接未被访问和已访问状态。然后将其 display 属性设置为 block，使得超链接成为块级元素，这样才可以为超链接设置 80 像素的宽度。代码如下：

```
<style type="text/css" >
a:link,a:visited {
    display:block;
    width:80px;
    font-weight:700;
    background-color:greenyellow;
    color:red;
    padding:10px;
    text-align:center;
    text-decoration:none;
}
</style>
```

添加本段代码后，运行效果如图 8-3 所示。



图 8-3 导航栏菜单的超链接设计

由图 8-3 可见，背景颜色为黄绿色，字体颜色为红色，宽度为 80 像素，各边的内边距为 10 像素，文本居中显示，并且去掉了文本的下划线样式。

4. 鼠标事件

利用 CSS 为 `a: hover` 和 `a: active` 进行样式设置，表示鼠标悬停和活动链接状态时的样式变化。代码如下：

```
<style type="text/css">
a: hover,a: active {
    background-color:darkblue;
    color:white;
}
</style>
```

添加本段代码后，运行效果如图 8-4 所示。



图 8-4 导航栏菜单的鼠标事件设计

由图 8-4 可见，鼠标悬浮的效果已实现。将鼠标在超链接上悬停和活动链接的样式更新成背景颜色为深蓝色，字体颜色为白色。

至此，水平导航栏菜单的设计与实现已全部完成，用户可根据实际需要用到网站开发中。示例程序 8.1 为本次完整的水平导航栏代码。

【例 8.1】 简单水平菜单导航栏的设计与实现。

```
<!--示例 8.1 程序-->
<!DOCTYPE html>
<html>
```

```
<head>
<meta charset="utf-8" />
<title>水平导航栏的制作</title>
<style type="text/css">
ul {
    margin:0;
    padding:0;
    list-style-type:none;
}
li {
    float:left;
}
a:link,a:visited {
    display:block;
    width:80px;
    font-weight:700;
    background-color:greenyellow;
    color:red;
    padding:10px;
    text-align:center;
    text-decoration:none;
}
a:hover,a:active {
    background-color:darkblue;
    color:white;
}
</style>
</head>
<body>
<center><h3>简单水平菜单导航栏</h3></center>
<hr>
<nav>
    <ul>
        <li><a href="#">前端</a></li>
        <li><a href="#">后台</a></li>
        <li><a href="#">移动应用</a></li>
        <li><a href="#">数据库</a></li>
        <li><a href="#">操作系统</a></li>
        <li><a href="#">服务器</a></li>
    </ul>
</nav>
</body>
</html>
```

8.1.2 下拉子菜单导航栏的设计与实现

下拉子菜单导航是网站设计中常见的子导航设计，一般适用于多级别、内容庞杂的网站。这种导航模式可以让整个网站设计更加整洁有序，现下大热的汉堡包导航也是采用这种隐藏式的导航设计。

1. 导航栏的创建

仍然是在<nav>的首尾标签之间，使用<div>标签创建菜单范围，结合无序列表和有序列表标签，配合列表选项创建菜单选项。其中选项内容使用超链接的形式，链接地址当前设置为空链接。代码如下：

```
<body>
<center><h3>下拉子菜单导航栏</h3></center>
<hr>
<nav>
  <div id="nav">
    <ul>
      <li><a href="#">前端</a>
        <ol>
          <li><a href="#">HTML5</a></li>
          <li><a href="#">CSS</a></li>
          <li><a href="#">Javascript</a></li>
        </ol>
      </li>
      <li><a href="#">后台</a>
        <ol>
          <li><a href="#">Java</a></li>
          <li><a href="#">PHP</a></li>
          <li><a href="#">Python</a></li>
        </ol>
      </li>
      <li><a href="#">移动应用</a>
        <ol>
          <li><a href="#">Android</a></li>
          <li><a href="#">IOS</a></li>
        </ol>
      </li>
      <li><a href="#">数据库</a>
        <ol>
          <li><a href="#">MySQL</a></li>
          <li><a href="#">SqlServer</a></li>
        </ol>
      </li>
    </ul>
  </div>
</nav>
```

```
<li><a href="#">操作系统</a>
  <ol>
    <li><a href="#">Linux</a></li>
    <li><a href="#">Unix</a></li>
    <li><a href="#">Windows</a></li>
  </ol>
</li>
<li><a href="#">服务器</a>
  <ol>
    <li><a href="#">Tomcat</a></li>
    <li><a href="#">WebSphere</a></li>
    <li><a href="#">WebLogic</a></li>
  </ol>
</li>
</ul>
</div>
</nav>
</body>
```

本段代码运行效果如图 8-5 所示。



图 8-5 下拉子菜单的初步创建

由图 8-5 可见，导航菜单中的文字内容和超链接已经完成，用户单击不同的选项可以跳转至指定的页面。无序列表是一级菜单，有序列表是二级菜单。

2. 列表样式的设计

此时会带有元素的实心点标记，及元素的阿拉伯序号，可以利用 CSS 在<head>标签中进行声明，对列表样式进行改变。

同样，使用 list-style-type 属性去掉默认标记，设置 margin 和 padding 属性为 0，为列表选项定义浮动效果，设置 text-decoration 属性 none。这里使用通配符“*”整体设置。代码如下：

```
<style type="text/css">
* {
    margin:0;
    padding:0;
    list-style-type:none;
    text-decoration:none;
}
ul li {
    float:left;
}
ul li ol li {
    float:none;
}
</style>
```

添加本段代码后，运行效果如图 8-6 所示。



图 8-6 下拉子菜单的行内显示

由图 8-6 可见，超链接去掉了文本的下划线样式，列表已经去掉了实心点标记和数字，无序列表每个列表选项均在同一行显示，有序列表由于设置为无浮动，所以每个列表选项依然垂直分行显示。

3. 二级菜单的隐藏和显示设计

为了使有序列表的二级菜单隐藏起来，需要设置 display 属性为 none。

为了鼠标悬停一级菜单时可以显示二级菜单，需要设置有序列表的 `display` 属性为 `block`。

为了二级菜单能在从属的一级菜单正下方显示，需要设置二级菜单的绝对定位及一级菜单的相对定位。代码如下：

```
<style type="text/css">
ul li {
    float:left;
    position:relative;
}
ul li ol {
    background-color:lightcyan;
    position:absolute;
    top:40px;
    left:0;
    display:none; /* 将二级菜单隐藏; */
}
ul li:hover ol {
    display:block; /* 鼠标悬停二级菜单从属的一级菜单时,二级菜单显示 */
}
</style>
```

添加本段代码后，运行效果如图 8-7 (a) (b) 所示。



图 8-7 运行效果图

由图 8-7 (a) 可见，二级菜单已经隐藏不见，在鼠标悬停在“数据库”一级菜单时，包含的二级菜单即以背景为淡青色显示出来，并在从属的一级菜单正下方显示，如图 8-7 (b) 所示。

4. DIV 样式的设计

完成以上关键步骤后，对内容所处的 `div` 样式进行设置，包括宽度、高度、边框和对齐方式。代码如下：


```
<style type="text/css">
#nav {
    width:600px;
    height:40px;
    background-color:lightcyan;
    border:5px solid blue;
    margin:0 auto;
}
</style>
```

添加本段代码后，运行效果如图 8-8 所示。



图 8-8 导航栏的外观显示效果

由图 8-8 可见，整个 div 边框为 5 像素的红色实线，背景颜色为淡青色，并相对页面居中对齐。

5. 超链接样式的设计

利用 CSS 为列表超链接重新设置样式，例如，对超链接的 `a: link` 和 `a: visited` 进行设置，表示超链接未被访问和已访问状态。然后将其 `display` 属性设置为 `block`，使得超链接成为块级元素，这样才可以为超链接设置 80 像素的宽度。代码如下：

```
<style type="text/css">
ul li a:link,a:visited {
    display:block;
    width:80px;
    font-weight:700;
    color:black;
    padding:10px;
    text-align:center;
}
</style>
```

添加本段代码后，运行效果如图 8-9 所示。

由图 8-9 可见, 字体颜色为黑色加粗, 宽度为 80 像素, 各边的内边距为 10 像素, 文本居中显示。



图 8-9 导航栏菜单的超链接设计

6. 鼠标事件

利用 CSS 为无序列表选项 `li: hover` 和有序列表的 `a: hover` 进行样式设置, 表示鼠标悬停状态时的样式变化。代码如下:

```
<style type="text/css" >
ul li: hover {
    background-color: pink;
}
ul li ol li a: hover {
    background-color: red;
    color: white;
}
</style>
```

添加本段代码后, 运行效果如图 8-10 所示。



图 8-10 导航栏菜单的鼠标事件设计

由图 8-10 可见, 鼠标悬浮的效果已实现。将鼠标在一级菜单上悬停的样式更新成背景颜色为粉红色, 二级菜单上的样式更新成背景颜色为红色, 字体颜色为白色。

至此, 下拉子导航栏菜单的设计与实现已全部完成, 用户可根据实际需要用到网站开发

中。示例 8.2 程序为本次完整的下拉子导航栏代码。

【例 8.2】复杂下拉子菜单导航栏的设计与实现。

```
<!-- 示例 8.2 程序 -->
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>下拉子导航栏的制作</title>
<style type="text/css">
* {
margin:0;
padding:0;
list-style-type:none;
text-decoration:none;
}
ul li {
float:left;
position:relative;
}
ul li ol li {
float:none;
}
ul li ol {
background-color:lightcyan;
position:absolute;
top:40px;
left:0;
display:none;
}
ul li:hover ol {
display:block;
}
#nav {
width:600px;
height:40px;
background-color:lightcyan;
border:5px solid red;
margin:0 auto;
}
ul li a:link,a:visited {
display:block;
width:80px;
font-weight:700;
```

```
        color:black;
        padding:10px;
        text-align:center;
    }
    ul li:hover{
        background-color:pink;
        color:white;
    }
    ul li ol li a:hover{
        background-color:red;
        color:white;
    }
</style>
</head>
<body>
<center><h3>下拉子菜单导航栏</h3></center>
<hr>
<nav>
    <div id="nav">
        <ul>
            <li><a href="#">前端</a>
                <ol>
                    <li><a href="#">HTML5</a></li>
                    <li><a href="#">CSS</a></li>
                    <li><a href="#">Javascript</a></li>
                </ol>
            </li>
            <li><a href="#">后台</a>
                <ol>
                    <li><a href="#">Java</a></li>
                    <li><a href="#">PHP</a></li>
                    <li><a href="#">Python</a></li>
                </ol>
            </li>
            <li><a href="#">移动应用</a>
                <ol>
                    <li><a href="#">Android</a></li>
                    <li><a href="#">IOS</a></li>
                </ol>
            </li>
            <li><a href="#">数据库</a>
                <ol>
                    <li><a href="#">MySQL</a></li>
```

```
        <li><a href="#" >SqlServer</a></li>
    </ol>
</li>
<li><a href="#" >操作系统</a>
    <ol>
        <li><a href="#" >Linux</a></li>
        <li><a href="#" >Unix</a></li>
        <li><a href="#" >Windows</a></li>
    </ol>
</li>
<li><a href="#" >服务器</a>
    <ol>
        <li><a href="#" >Tomcat</a></li>
        <li><a href="#" >WebSphere</a></li>
        <li><a href="#" >WebLogic</a></li>
    </ol>
</li>
</ul>
</div>
</nav>
</body>
</html>
```

8.2 纵向侧边导航栏

侧边菜单导航栏是网站设计中应用频率仅次于水平导航栏的导航设计，一般放置在首页的左上角，普遍使用文字链接作为导航项，可以容纳很多链接。

如果导航菜单太长会增加用户的浏览难度，因此，侧边导航栏通常会跟出式子导航结合使用，商城网站建设大多应用这种复合的导航设计。

8.2.1 简单纵向导航栏的设计与实现

1. 导航栏的创建

普通的纵向导航同样可以使用超链接和列表元素来实现，只需要将文字链接设置成块级元素。区别于水平导航栏，纵向导航栏不需要设置列表选项的浮动属性。

在<nav>的首尾标签之间使用<div>标签创建菜单范围，结合无序列表标签和列表选项标签创建菜单选项。这里的超链接地址暂时设置为空链接，用户可根据实际情况改成具体的 URL 地址。代码如下：

```
<body>
<nav>
    <div id="nav">
```

```
<h3 class="tit">所有商品分类</h3>
<ul>
  <li><a href="#">家居用品</a></li>
  <li><a href="#">手机数码</a></li>
  <li><a href="#">家用电器</a></li>
  <li><a href="#">厨卫工具</a></li>
  <li><a href="#">汽车用品</a></li>
  <li><a href="#">运动器材</a></li>
</ul>
</div>
</nav>
</body>
```

本段代码运行效果如图 8-11 所示。

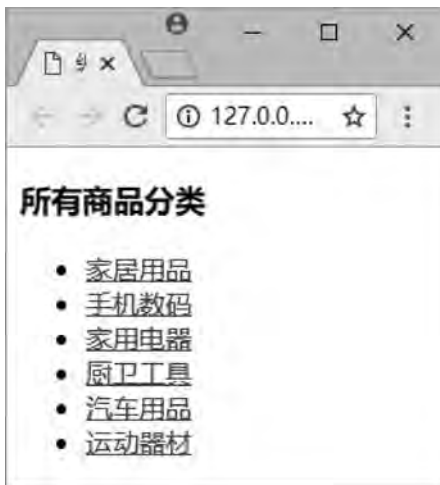


图 8-11 导航栏菜单的初步创建

由图 8-11 可见，导航菜单中的文字内容和超链接已经完成，用户单击不同的选项可以跳转至指定的页面。

2. DIV 样式的设计

对导航栏所处的 div 样式进行设置，包括宽度、边框和背景，同时对大标题文字的 CSS 进行设置，包括高度、行高、颜色和对齐方式。代码如下：

```
<style type="text/css">
#nav {
  width:150px;
  border:1px solid red;
  background-color:lightgreen;
}
.tit {
  height:16px;
  line-height:16px;
```

```

        color:#FFFFFF;
        text-align:center;
    }
</style>

```

添加本段代码后，运行效果如图 8-12 所示。



图 8-12 导航栏的外观显示效果

由图 8-12 可见，整个 div 宽度为 150 像素，边框为 1 像素的红色实线，背景颜色为淡绿色，“所有商品分类”大标题为 16 像素的白色字体，并相对 div 居中对齐。

3. 列表样式的设计

此时列表选项还带有元素的默认实心圆点样式，需要使用 list-style-type 属性去掉默认标记，设置 margin 和 padding 的属性值。并同时给列表选项添加背景颜色，使大小标题容易区分。代码如下：

```

<style type="text/css">
#nav ul {
    margin:0px;
    padding:0px;
    list-style-type:none; /* 隐藏默认列表符号 */
    background-color:white;
}
#nav ul li {
    height:30px;
    line-height:30px;
    text-align:center;
    border-bottom:1px solid gray;
}
</style>

```

添加本段代码后，运行效果如图 8-13 所示。



图 8-13 导航栏菜单的列表显示效果

由图 8-13 可见，列表选项已经去掉了实心点标记，背景更新为白色，文字居中，并且有 1 像素的灰色下边框。

4. 超链接样式的设计

利用 CSS 为超链接重新设置样式，对超链接的 `a: link` 和 `a: visited` 进行设置，表示超链接未被访问和已访问状态。同时，利用 CSS 为 `a: hover` 和 `a: active` 进行样式设置，表示鼠标悬停和活动链接状态时的样式变化。并且，将 `display` 属性设置为 `block`，使得超链接成为块级元素，以及字体 CSS 样式的设置。代码如下：

```
<style type="text/css" >
a:link,a:visited{
    display:block;
    font-size:18px;
    color:brown;
    text-decoration:none;
}
a:hover,a:active{
    font-size:15px;
    color:red;
    border:3px solid red;
    font-weight:bold;
    background-image:url(img/bg.JPG);
}
</style>
```

添加本段代码后，运行效果如图 8-14 所示。



图 8-14 菜单的超链接设计



图 8-15 菜单的鼠标事件设计

由图 8-14 可见，字体为 18 像素棕色，去掉了文本的下划线样式。

由图 8-15 可见，鼠标悬浮的效果已实现。将鼠标在超链接上悬停和活动链接的样式更新成背景图片和 3 像素的红色实线边框，字体缩小为 15 像素红色，并加粗显示。

至此，纵向导航栏菜单的设计与实现已全部完成，用户可根据实际需要用到网站开发中。示例 8.3 程序为本次完整的纵向导航栏代码。

【例 8.3】 简单纵向菜单导航栏的设计与实现。

```
<! --示例 8.3 程序-->
```

```
<! DOCTYPE html>
```

```
<html>
```

```
<head>
```

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

```
<title>纵向导航栏的制作</title>
```

```
<style type="text/css">
```

```
#nav{
```

```
width:150px;
```

```
border:1px solid red;
```

```
background-color:lightgreen;
```

```
}
```

```
.tit{
```

```
height:16px;
```

```
line-height:16px;
```

```
color:#FFFFFF;
```

```
/* 白色 */
```

```
text-align:center;
```

```
}
```

```
#nav ul{
```

```
margin:0px;
padding:0px;
list-style-type:none;/* 隐藏默认列表符号 */
background-color:white;
}
#nav ul li {
height:30px;
line-height:30px; /* 行距 */
text-align:center;
border-bottom:1px solid gray;
}
a:link,a:visited {
display:block;
font-size:18px;
color:brown;
text-decoration:none;/* 隐藏超链接默认下划线 */
}
a:hover,a:active {
font-size:15px;
color:red;
border:3px solid red;
font-weight:bold;
background-image:url(img/bg.JPG);
}
</style>
</head>
<body>
<nav>
<div id="nav">
<h3 class="tit">所有商品分类</h3>
<ul>
<li><a href="#">家居用品</a></li>
<li><a href="#">手机数码</a></li>
<li><a href="#">家用电器</a></li>
<li><a href="#">厨卫工具</a></li>
<li><a href="#">汽车用品</a></li>
<li><a href="#">运动器材</a></li>
</ul>
</div>
</nav>
```

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

8.2.2 出式子菜单导航栏的设计与实现

出式子菜单导航跟下拉子菜单导航一样，是网站设计中常见的子导航设计，只是出式子菜单多用于侧边纵向导航栏。这种子菜单适用于多级别、内容庞杂的网站，可以让整个网站设计更加整洁有序。

1. 导航栏的创建

在<nav>的首尾标签之间，使用<div>标签创建菜单范围，结合无序列表和有序列表标签，配合列表选项创建菜单选项。其中选项内容使用超链接的形式，链接地址当前设置为空链接。代码如下：

```
<body>  
<div id="nav">  
  <h3 class="tit">所有商品分类</h3>  
  <ul>  
    <li><a href="#">家居用品</a>  
      <ol>  
        <li><a href="#">收纳箱</a></li>  
        <li><a href="#">雨伞雨具</a></li>  
        <li><a href="#">家务清洁</a></li>  
        <li><a href="#">茶具</a></li>  
        <li><a href="#">环保袋</a></li>  
      </ol>  
    </li>  
    <li><a href="#">手机数码</a>  
      <ol>  
        <li><a href="#">电脑主机</a></li>  
        <li><a href="#">鼠标键盘</a></li>  
        <li><a href="#">无人机</a></li>  
        <li><a href="#">数码相机</a></li>  
      </ol>  
    </li>  
    <li><a href="#">家用电器</a>  
      <ol>  
        <li><a href="#">吸尘器</a></li>  
        <li><a href="#">电饭煲</a></li>  
        <li><a href="#">冰箱</a></li>  
      </ol>  
    </li>  
</div>
```

```
<li><a href="#">厨卫工具</a>
<ol>
  <li><a href="#">消毒柜</a></li>
  <li><a href="#">抽油烟机</a></li>
  <li><a href="#">碗柜</a></li>
  <li><a href="#">花洒笼头</a></li>
  <li><a href="#">洗手台</a></li>
</ol>
</li>
<li><a href="#">汽车用品</a>
<ol>
  <li><a href="#">打蜡服务</a></li>
  <li><a href="#">超级试驾</a></li>
  <li><a href="#">导航仪</a></li>
  <li><a href="#">机油</a></li>
</ol>
</li>
<li><a href="#">运动器材</a>
<ol>
  <li><a href="#">跑鞋</a></li>
  <li><a href="#">帐篷</a></li>
  <li><a href="#">山地车</a></li>
</ol>
</li>
</ul>
</div>
</body>
```

本段代码运行效果如图 8-16 所示。

由图 8-16 可见，导航菜单中的文字内容和超链接已经完成，用户单击不同的选项可以跳转至指定的页面。无序列表是一级菜单，有序列表是二级菜单。

2. DIV 样式的设计

同 8.2.1 节的第二步骤（2. DIV 样式的设计）的方法一样。对导航栏所处的 div 和文字的 CSS 样式进行设置后，运行效果如图 8-17 所示。

由图 8-17 可见，列表已经去掉了实心点标记和数字，整个 div 宽度为 150 像素，边框为 1 像素的红色实线，背景颜色为淡绿色，“所有商品分类”大标题为 16 像素的白色字体，并相对 div 居中对齐。



图 8-16 出式子菜单的初步创建图



图 8-17 出式子菜单的外观显示效果

3. 二级菜单的隐藏和显示设计

为了使有序列表的二级菜单隐藏起来，需要设置 display 属性为 none。

为了鼠标悬停一级菜单时可以显示二级菜单，需要设置有序列表的 display 属性为 block。

为了二级菜单能在从属的一级菜单正右方显示，需要设置二级菜单的绝对定位及一级菜单的相对定位。代码如下：

```
<style type="text/css">
#nav ul li{
    height:30px;
    line-height:30px;
    text-align:center;
```

```

border-bottom:1px solid gray;
position:relative;/* 先对一级菜单的列表选项设置相对定位 */
}
#nav ul li ol{
display:none;/* 默认二级菜单隐藏 */
width:120px;
border:1px solid red;
position:absolute;/* 二级菜单相对一级菜单正右边显示 */
top:-1px;
left:150px;
}
#nav ul li:hover ol{
display:block;/* 当鼠标悬停在一级菜单时,二级菜单成块级元素显示 */
}
</style>

```

添加本段代码后,运行效果如图 8-18 (a) (b) 所示。



图 8-18 运行效果图

由图 8-18 可见,一级菜单和二级菜单都有 1 像素的灰色下边框,并且文字居中。由图 8-18 (a) 显示,二级菜单已经隐藏不见。由图 8-18 (b) 显示,鼠标悬停在一级菜单“手机数码”时,二级菜单在从属的一级菜单正右方显示。

4. 列表样式的设计

此时列表选项还带有元素的实心圆点样式,及元素的阿拉伯序号,需要使用 list-style-type 属性去掉默认标记,设置 margin 和 padding 的属性值。并同时给列表选项添加背景颜色,使大小标题容易区分。代码如下:

```

<style type="text/css">
#nav ul,#nav ol{
margin:0px;

```

```
padding:0px;
list-style-type:none;
background-color:white;
}
</style>
```

添加本段代码后，运行效果如图 8-19 所示。



图 8-19 导航栏菜单的列表显示效果

由图 8-19 可见，列表选项已经去掉了相关默认标记，背景更新为白色。

5. 超链接样式的设计

利用 CSS 为超链接重新设置样式，对超链接的 `a: link` 和 `a: visited` 进行设置，表示超链接未被访问和已访问状态。同时，利用 CSS 为 `a: hover` 和 `a: active` 进行样式设置，表示鼠标悬停和活动链接状态时的样式变化。并且，将 `display` 属性设置为 `block`，使得超链接成为块级元素，以及字体 CSS 样式的设置。代码如下：

```
<style type="text/css">
#nav ul li a:link,a:visited{
    display:block;
    font-size:18px;
    color:brown;
    text-decoration:none;
}
#nav ul li:hover{
    background-image:url(img/bg2.JPG);
    font-weight:bold;
}
#nav ul li ol li a:link{
    font-size:15px;
}
#nav ul li ol li a:hover{
```

```
background-color:red;
color:white;
}
</style>
```

添加本段代码后，运行效果如图 8-20 所示。



图 8-20 运行效果图

由图 8-20 (a) 可见，文本去除了下划线，一级菜单未被访问时，字体为 18 像素棕色。

由图 8-20 (b) 可见，鼠标悬浮的效果已实现。将鼠标在一级菜单上悬停的样式更新成背景图片，字体加粗显示；二级菜单上字体缩小为 15 像素，鼠标悬停其上时，样式更新成背景颜色为红色，字体颜色为白色。

至此，出式子导航栏菜单的设计与实现已全部完成，用户可根据实际需要用到网站开发中。示例程序 8.4 为本次完整的出式子导航栏代码。

【例 8.4】 复杂出式子菜单导航栏的设计与实现。

```
<!-- 示例 8.4 程序 -->
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>出式子导航栏的制作</title>
<style type="text/css">
#nav {
width:150px;
border:1px solid red;
background-color:lightgreen;
}
.tit {
height:16px;
```



```
        line-height:16px;
        color:#FFFFFF;
        text-align:center;
    }
#nav ul li {
    height:30px;
    line-height:30px;
    text-align:center;
    border-bottom:1px solid gray;
    position:relative; /* 先对一级菜单的列表选项设置相对定位 */
}
#nav ul li ol {
    display:none; /* 默认二级菜单隐藏 */
    width:120px;
    border:1px solid red;
    position:absolute; /* 二级菜单相对一级菜单正右边显示 */
    top:-1px;
    left:150px;
}
#nav ul li: hover ol {
    display:block; /* 当鼠标悬停在一级菜单时,二级菜单成块级元素显示 */
}
#nav ul,#nav ol {
    margin:0px;
    padding:0px;
    list-style-type:none;
    background-color:white;
}
#nav ul li a:link,a:visited {
    display:block;
    font-size:18px;
    color:brown;
    text-decoration:none; /* 隐藏超链接默认下划线 */
}
#nav ul li: hover {
    background-image:url(img/bg2.JPG);
    font-weight:bold;
}
#nav ul li ol li a:link {
    font-size:15px;
}
#nav ul li ol li a: hover {
    background-color:red;
```



```
        color:white;
    }
</style>
</head>
<body>
<nav>
    <div id="nav">
        <h3 class="tit">所有商品分类</h3>
        <ul>
            <li><a href="#">家居用品</a>
                <ol>
                    <li><a href="#">收纳箱</a></li>
                    <li><a href="#">雨伞雨具</a></li>
                    <li><a href="#">家务清洁</a></li>
                    <li><a href="#">茶具</a></li>
                    <li><a href="#">环保袋</a></li>
                </ol>
            </li>
            <li><a href="#">手机数码</a>
                <ol>
                    <li><a href="#">电脑主机</a></li>
                    <li><a href="#">鼠标键盘</a></li>
                    <li><a href="#">无人机</a></li>
                    <li><a href="#">数码相机</a></li>
                </ol>
            </li>
            <li><a href="#">家用电器</a>
                <ol>
                    <li><a href="#">吸尘器</a></li>
                    <li><a href="#">电饭煲</a></li>
                    <li><a href="#">冰箱</a></li>
                </ol>
            </li>
            <li><a href="#">厨卫工具</a>
                <ol>
                    <li><a href="#">消毒柜</a></li>
                    <li><a href="#">抽油烟机</a></li>
                    <li><a href="#">碗柜</a></li>
                    <li><a href="#">花洒笼头</a></li>
                    <li><a href="#">洗手台</a></li>
                </ol>
            </li>
            <li><a href="#">汽车用品</a>
```

```
<ol>
  <li><a href="#">打蜡服务</a></li>
  <li><a href="#">超级试驾</a></li>
  <li><a href="#">导航仪</a></li>
  <li><a href="#">机油</a></li>
</ol>
</li>
<li><a href="#">运动器材</a>
  <ol>
    <li><a href="#">跑鞋</a></li>
    <li><a href="#">帐篷</a></li>
    <li><a href="#">山地车</a></li>
  </ol>
</li>
</ul>
</div>
</nav>
</body>
</html>
```

8.3 底部固定导航栏

底部导航，顾名思义是放在页脚部位，是网站设计中一种辅助导航。底部导航应用性不是很广，比较常出现在一些活动或个性化的网站当中。并且，底部导航被广泛使用的并不是在 PC 端中，而是在移动端。如果在 PC 端中采用底部导航的形式，一般都是采用固定的方式。

底部导航可以减少用户的使用成本，但对于结构复杂有二级或三级导航的网站，就不是很适合。而且，用户浏览的习惯都是从上到下、从左往右的，将导航放置底部，这样的设计比较挑战用户的使用习惯。所以，很多网站在设计的时候首页导航会放在底部，其后的导航就会固定到顶部。

1. 导航栏的创建

在<nav>的首尾标签之间，使用<div>标签创建菜单范围，结合无序列表配合列表选项创建菜单选项。其中选项内容使用超链接的形式，链接地址当前设置为空链接。代码如下：

```
<body>
<nav>
  <div class="fix-footer">
    <ul>
      <li><a href="#">首页</a></li>
      <li><a href="#">前端</a></li>
```

```
<li><a href="#">后台</a></li>
<li><a href="#">移动应用</a></li>
</ul>
</div>
</nav>
</body>
```

本段代码运行效果如图 8-21 所示。



图 8-21 底部菜单的初步创建图

由图 8-21 可见，导航菜单中的文字内容和超链接已经完成，用户单击不同的选项可以跳转至指定的页面。此时，文字内容处于页面的左上角。

2. 列表样式的设计

此时会带有元素的默认实心点标记，可以利用 list-style-type 属性去掉，同时设置 margin 和 padding 属性。列表选项设置 float 属性，让列表元素能够在同一行显示。代码如下：

```
<style type="text/css">
ul{
    margin:0;
    padding:0;
    list-style-type:none;
}
li{
    width:25%;
    float:left;
    line-height:50px;
}
</style>
```

添加本段代码后，运行效果如图 8-22 所示。

由图 8-22 可见，列表已经去掉了实心点标记，并且每个列表项均在同一行显示，并且，每个列表选项相对整个 div 均是 25% 的宽度。此时，文字内容处于页面的正顶部。



图 8-22 导航栏菜单的行内显示

3. 菜单固定底部的设计

为了美化文字内容的外观，可以设置 div 的高度、宽度、背景、边框和文本对齐方式等。为了使导航菜单始终处于页面的最上层，需要设置 `z-index` 属性的优先值。为了使导航菜单固定于页面底部，需要设置位置属性 `position` 和各个方向的位置值。代码如下：

```
<style type="text/css">
.fix-footer{
    height:50px;
    width:100%;
    background-color:lightgreen;
    border-top:2px solid blue;
    text-align:center;
    z-index:9999;
    position:fixed;
    bottom:0;
    left:0;
}
</style>
```

添加本段代码后，运行效果如图 8-23 所示。



图 8-23 菜单的底部固定效果

由图 8-23 可见，`<div>` 元素高度为 50 像素，宽度为整个页面的 100%，背景颜色更新为淡绿色，有 2 像素的蓝色上边框，每个列表选项的文本居中对齐。此时，整个 `<div>` 块已经

固定于页面的正底部，无论页面内容有多少，底部导航的位置都不会随着滚动条的移动而改变。

4. 超链接样式的设计

利用 CSS 为超链接重新设置样式，对超链接的 `a: link` 和 `a: visited` 进行设置，表示超链接未被访问和已访问状态。然后将其 `display` 属性设置为 `block`，使得超链接成为块级元素。代码如下：

```
<style type="text/css" >
a:link,a:visited{
    display:block;
    font-size:20px;
    color:black;
    text-decoration:none;
    font-weight:bold;
}
</style>
```

添加本段代码后，运行效果如图 8-24 所示。



图 8-24 菜单的底部固定效果

由图 8-24 可见，字体为 20 像素黑色，加粗显示，并且去掉了文本的下划线样式。

5. 鼠标事件

利用 CSS 为 `a: hover` 和 `a: active` 进行样式设置，表示鼠标悬停和活动链接状态时的样式变化。代码如下：

```
<style type="text/css" >
a:hover,a:active{
    background-color:darkgray;
    border-left:2px solid white;
    border-right:2px solid white;
    font-size:16px;
    color:red;
    font-weight:bold;
}
</style>
```

添加本段代码后，运行效果如图 8-25 所示。



图 8-25 菜单的底部固定效果

由图 8-25 可见，鼠标悬浮的效果已实现。将鼠标在超链接上悬停和活动链接的样式更新成背景颜色为深灰色，字体缩小为 16 像素红色，加粗显示，每个块级元素的左、右边框均有 2 像素白色实线。

至此，底部导航栏菜单的设计与实现已全部完成，用户可根据实际需要用到网站开发中。示例 8.5 程序为本次完整的底部导航栏代码。

【例 8.5】底部菜单导航栏的设计与实现。

```
<! --示例 8.5 程序-->
<! DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>底部导航栏的制作</title>
<style type="text/css">
ul {
    margin:0;
    padding:0;
    list-style-type:none;
}
li {
    width:25%;
    float:left;
    line-height:50px;
}
.fix-footer {
    height:50px;
    width:100%;
    background-color:lightgreen;
    border-top:2px solid blue;
    text-align:center;
    z-index:9999;
    position:fixed;
```

```
        bottom:0;
        left:0;
    }
    a:link,a:visited{
        display:block;
        font-size:20px;
        color:black;
        text-decoration:none;
        font-weight:bold;
    }
    a:hover,a:active{
        background-color:darkgray;
        border-left:2px solid white;
        border-right:2px solid white;
        font-size:16px;
        color:red;
        font-weight:bold;
    }
</style>
</head>
<body>
<nav>
    <div class="fix-footer">
        <ul>
            <li><a href="#">首页</a></li>
            <li><a href="#">前端</a></li>
            <li><a href="#">后台</a></li>
            <li><a href="#">移动应用</a></li>
        </ul>
    </div>
</nav>
</body>
</html>
```

网站建设选用哪种类型的导航，完全取决于网站的类型和设计者想要达到的效果。以上介绍的三种导航方式，各有利弊，但无论哪一种，导航既然是导航，就应该起到为用户快速便捷找到所需的作用，提高网站的易用性和易操作性。并不是花哨的导航就是好的，用户往往喜欢简单醒目的，这才是导航的设计原则，设计者在建设网站时需要衡量取舍。

8.4 综合案例——优名养生馆

本章综合前面学习的 CSS 和导航栏知识，可以利用链接、列表、DIV 和 CSS 设计并制作

网页。本章案例有一个封面页和一个主页，下面是页面的具体设计与实现步骤。

1. 封面页的设计与实现

封面页页面布局较简单，要求有正文和底部两块区域，用于放图片信息和底部导航信息，而底部导航又根据优先级分上层的圆球菜单和下层的水平菜单。

(1) 为正文区域添加滚动图片信息，并设置图片的 CSS 样式。代码如下：

```
<style type="text/css">
img {
    margin-top:70px;
    margin-right:40px;
}
</style>
<body>
<div>
    <marquee direction="right" scrollamount="20">
        
        
        
    </marquee>
</div>
</body>
```

完成后，页面显示如图 8-26 所示。

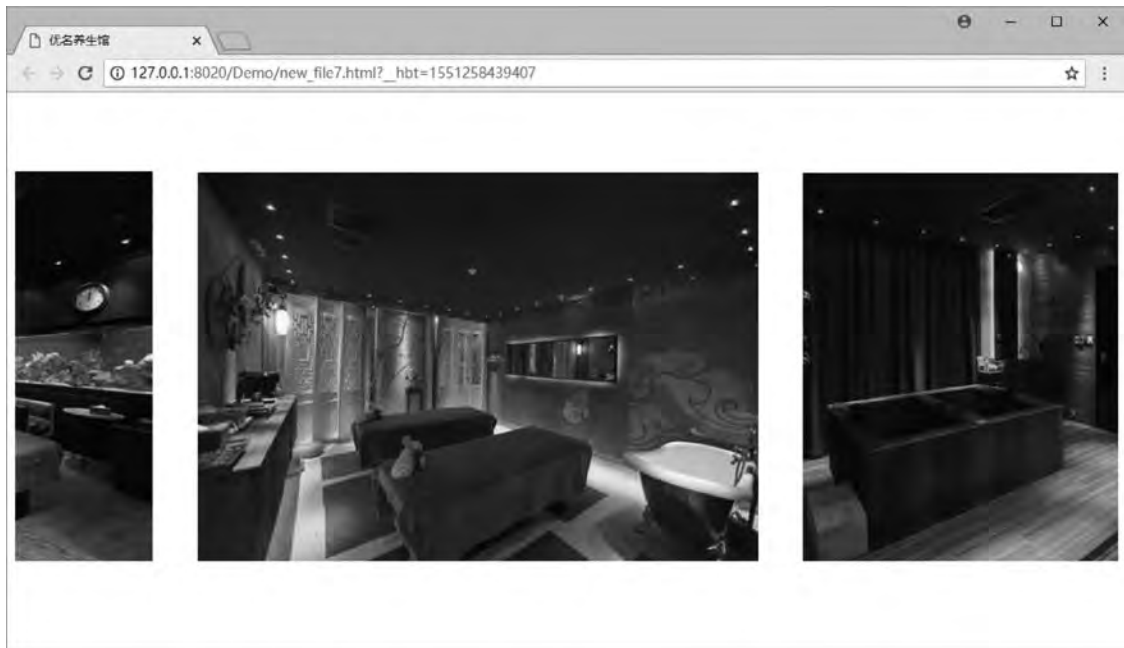


图 8-26 网页部分效果图



(2) 利用超链接为底部区域先添加圆球导航, 并设置相应的 CSS 样式。代码如下:

```
<style type="text/css">
.footer-top {
    height:90px;
    width:90px;
    text-align:center;
    line-height:90px;
    z-index:9999;
    position:fixed;
    bottom:0;
    left:45%;
}
.footer-top a {
    border-radius:90px;
}
.footer-top a:link,a:visited {
    display:block;
    font-size:20px;
    color:brown;
    text-decoration:none;
    font-weight:bold;
    border-top:3px solid gray;
    background-image:url(img/footer-top-bg1.jpg);
}
.footer-top a:hover {
    background-image:url(img/footer-top-bg2.jpg);
    color:white;
}
</style>
<body>
<div class="footer-top">
    <a href="index.html">进入官网</a>
</div>
</body>
```

设置定位和优先级后, 页面显示如图 8-27 所示。

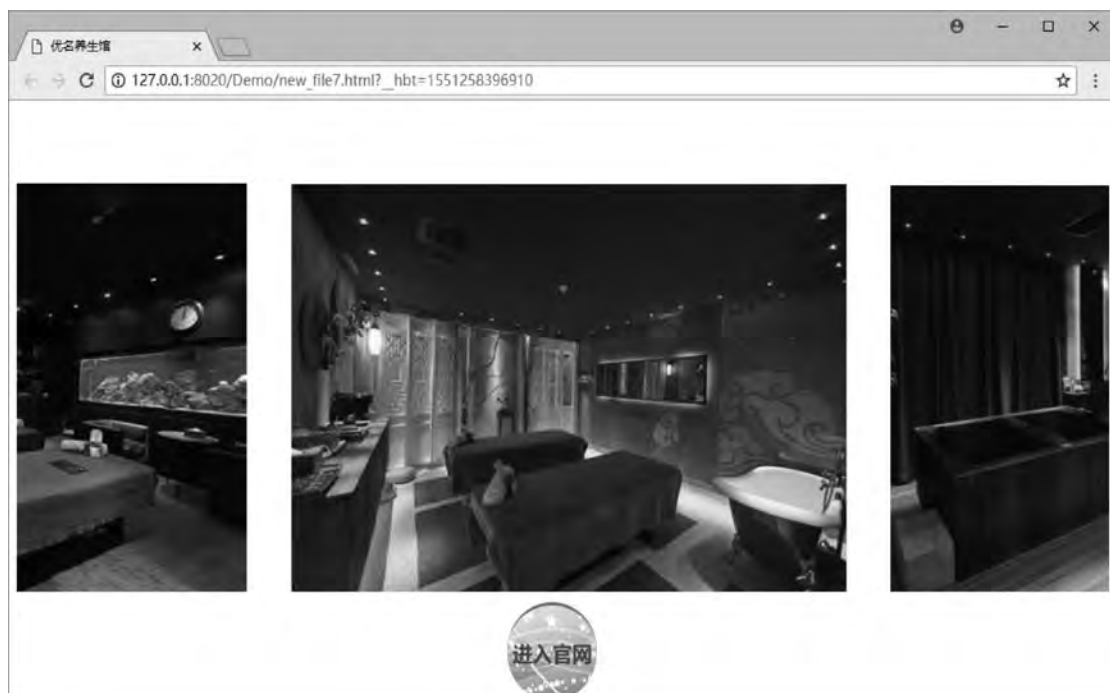


图 8-27 网页部分效果图

(3) 利用列表为底部区域后添加水平导航，并设置相应的 CSS 样式。代码如下：

```
<style type="text/css" >
ul {
    margin:0;
    padding:0;
    list-style-type:none;
}
li {
    width:25%;
    float:left;
    line-height:50px;
}
.footer {
    height:50px;
    width:100%;
    background-image:url(img/footer-bg.jpg);
    border-top:3px solid gray;
    text-align:center;
    z-index:9998;
    position:fixed;
    bottom:0;
    left:0;
}
```

```

.footer a:link,a:visited{
    display:block;
    font-size:20px;
    color:brown;
    text-decoration:none;
    font-weight:bold;
}
.footer a:hover{
    background-color:darkgray;
    border-left:2px solid white;
    border-right:2px solid white;
    font-size:16px;
    color:red;
}
</style>
<body>
<div class="footer">
    <ul>
        <li><a href="#">奇幻世界</a></li>
        <li><a href="#">主题风采</a></li>
        <li><a href="#">加盟相关</a></li>
        <li><a href="#">友情推荐</a></li>
    </ul>
</div>
</body>

```

设置定位和优先级后，水平菜单在底部显示，优先级低于圆球菜单。此时，页面显示如图 8-28 所示。



图 8-28 网页部分效果图

完成以上步骤，添加页面背景图片，综合案例封面页最终网页浏览效果如图 8-29 所示。



图 8-29 网页浏览效果图

以下为封面页的完整代码，以供参考。

```
<!--综合案例-封面页.html-->
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>优名养生馆</title>
<style type="text/css">
body{
    background-image:url(img/bg.jpg);
}
ul{
    margin:0;
    padding:0;
    list-style-type:none;
}
li{
    width:25%;
    float:left;
    line-height:50px;
```

```

}
img {
    margin-top:70px;
    margin-right:40px;
}
.footer {
    height:50px;
    width:100%;
    background-image:url(img/footer-bg.jpg);
    border-top:3px solid gray;
    text-align:center;
    z-index:9998;
    position:fixed;
    bottom:0;
    left:0;
}
.footer-top {
    height:90px;
    width:90px;
    text-align:center;
    line-height:90px;
    z-index:9999;
    position:fixed;
    bottom:0;
    left:45%;
}
.footer-top a {
    border-radius:90px;
}
.footer-top a:link,a:visited {
    display:block;
    font-size:20px;
    color:brown;
    text-decoration:none;
    font-weight:bold;
    border-top:3px solid gray;
    background-image:url(img/footer-top-bg1.jpg);
}
.footer-top a:hover {

```

```
background-image: url( img/footer-top-bg2. jpg );
color: white;
}
.footer a: link, a: visited {
display: block;
font-size: 20px;
color: brown;
text-decoration: none;
font-weight: bold;
}
.footer a: hover {
background-color: darkgray;
border-left: 2px solid white;
border-right: 2px solid white;
font-size: 16px;
color: red;
}
</style>
</head>
<body>
<div>
<marquee direction=" right" scrollamount=" 20" >



</marquee>
</div>
<nav>
<div class=" footer-top" >
<a href=" index. html" >进入官网</a>
</div>
<div class=" footer" >
<ul>
<li><a href=" #" >奇幻世界</a></li>
<li><a href=" #" >主题风采</a></li>
<li><a href=" #" >加盟相关</a></li>
<li><a href=" #" >友情推荐</a></li>
</ul>
</div>
```



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

2. 主页的设计与实现

主页页面布局要求有页头、正文和页脚三块区域。页头完成背景图片的添加和店名的输入；正文分左右两部分，分别完成侧边纵向导航和图文信息的添加；页脚完成版权信息。页头、正文和页脚三块区域均包含在 div-all 区域内。

(1) 为页头区域添加背景图片，并加入嵌套的 div，添加店名信息，最后设置相应的 CSS 样式。代码如下：

```
<style type="text/css">
.all{
    margin:0px auto;
    width:900px;
}
.top{
    width:900px;
    height:225px;
    background-image:url(img/top-bg.jpg);
}
.cent{
    width:320px;
    margin-left:auto;
    margin-right:auto;
    font-size:58px;
    font-family:"楷体";
    color:lemonchiffon;
    font-style:italic;
}
</style>
<body>
<div class="all">
    <div class="top">
        <br>
        <div class="cent">优名养生馆</div>
    </div>
</div>
</body>
```

完成后，页面显示如图 8-30 所示。



图 8-30 网页部分效果图

(2) 为正文区域左边添加纵向导航信息，并设置相应的 CSS 样式。代码如下：

```
<style type="text/css" >
.nav {
    width:200px;
    float:left;
    margin-top:30px;
    font-size:20px;
    font-weight:bold;
    color:saddlebrown;
}
ul {
    list-style-type:none;
}
li {
    height:22px;
}
a:link {
    font-size:16px;
    text-decoration:none;
```

```
        color: brown;
    }
    a: hover {
        background-color: burlywood;
        color: white;
    }
</style>
<body>
<div class = " all" >
    <nav>
        <div class = " nav" >
            <table>
                <tr>养生之道</tr>
                <tr>
                    <ul>
                        <li><a href = "#" >运动养生</a></li>
                        <li><a href = "#" >四季养生</a></li>
                        <li><a href = "#" >健康检测</a></li>
                    </ul>
                </tr>
                <tr>中医养生</tr>
                <tr>
                    <ul>
                        <li><a href = "#" >经络养生</a></li>
                        <li><a href = "#" >体质养生</a></li>
                        <li><a href = "#" >特色疗法</a></li>
                    </ul>
                </tr>
            </table>
        </div>
    </nav>
</div>
</body>
```

完成后，页面显示如图 8-31 所示。

体生命的行为,即根据人体生命过程的活动规律所进行的物质与精神的身心护养活动。既包括生理层面的养生,注重身体机能的维护和康复;也包括心理层面的养生,腔调内在精神的平衡和祥和。按照层次划分,生理层面的养生包括养颜、养体、养老;心理层面的养生包括养心、养性、养神。

```

</div>
</div>
</body>

```

完成后,页面显示如图 8-32 所示。



图 8-32 网页部分效果图

(4) 为页脚区域添加版权信息,并设置相应的 CSS 样式。代码如下:

```

<style type="text/css">
.footer {
    font-size: 14px;
    font-weight: bold;
    color: brown;
    text-align: center;
    clear: both;
    background-image: url(img/footer-bg.jpg);
}
</style>
<body>
<div class="all">
    <div class="footer">版权所有 &copy;优名养生馆</div>
</div>
</body>

```

完成后，页面显示如图 8-33 所示。



图 8-33 网页部分效果图

完成以上步骤，添加页面背景图片，综合案例主页最终网页浏览效果如图 8-34 所示。



图 8-34 网页浏览效果图



以下为主页的完整代码，以供参考。

```
<! --综合案例-主页.html-->
<! DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>优名养生馆</title>
<style>
body {
    background-image: url( img/bg. jpg );
}
.all {
    margin: 0px auto;
    width: 900px;
}
.top {
    width: 900px;
    height: 225px;
    background-image: url( img/top-bg. jpg );
}
.cent {
    width: 320px;
    margin-left: auto;
    margin-right: auto;
    font-size: 58px;
    font-family: " 楷体 ";
    color: lemonchiffon;
    font-style: italic;
}
.nav {
    width: 200px;
    float: left;
    margin-top: 30px;
    font-size: 20px;
    font-weight: bold;
    color: saddlebrown;
}
.main {
    width: 700px;
    height: 300px;
    float: left;
```

```
        margin-top:10px;
    }
    . footer {
        font-size:14px;
        font-weight:bold;
        color:brown;
        text-align:center;
        clear:both;
        background-image:url(img/footer-bg.jpg);
    }
    p {
        color:brown;
        font-size:16px;
    }
    img {
        margin:15px;
    }
    ul {
        list-style-type:none;
    }
    li {
        height:22px;
    }
    a:link {
        font-size:16px;
        text-decoration:none;
        color:brown;
    }
    a:hover {
        background-color:burlywood;
        color:white;
    }
</style>
</head>
<body>
<div class="all">
    <div class="top">
        <br>
        <div class="cent">优名养生馆</div>
    </div>
    <nav>
```




图 8-35 题 1 网页效果图

2. 利用 CSS 技术, 结合链接和列表, 设计并实现“茶韵”页面。图 8-36 供参考。



图 8-36 题 2 网页效果图

第 9 章 DIV+CSS 布局



学习目标

1. 理解 DIV+CSS 的定义；
2. 理解 DIV+CSS 的宽高设置；
3. 掌握 DIV+CSS 的布局方法。



知识要点

1. DIV 元素的布局技巧；
2. DIV 元素的宽高自适应；
3. DIV 元素的嵌套；
4. DIV+CSS 的常用布局类型。

如果依赖于图形化界面的网页编辑器，可以轻松地将一个网站布局建立起来，可是这有一个缺点：就是它的代码修改极其繁琐，设计者往往不能根据自己的思想进行修改或者设计。因此，网页制作就对网页源码提出了更高的要求。传统的网页布局是采用表格式的布局方式，遵循常规，模式单一。这样就催生出了更高级的布局，即 DIV+CSS。这也是当今网站常用的布局方式，灵活多样。

9.1 DIV+CSS 概述

DIV+CSS 是 Web 设计标准，它是一种网页的布局方法。与传统中通过表格（table）布局定位的方式不同，它可以实现网页页面内容与表现相分离。DIV 组成了网页的格局，CSS 则装饰了格局，比如建一栋房子，开始的架子是 DIV，架子搭建好后开始装饰，这个装饰就是 CSS 样式。使用了 DIV+CSS 布局的网页，它使 HTML 语言变得越来越复杂化、专用化，源码容易扩展，并且更加规整。

9.1.1 认识 DIV

div 标签在 Web 标准的网页中使用非常频繁，属于块状元素。div 标签是双标签，即以 `<div></div>` 的形式存在，中间可以放置任何内容，包括其他的 div 标签。

但是在没有 CSS 的影响下，每个 div 标签只占据一行，即一行只能容纳一个 div 标签。

【例 9.1】认识 div 标签。

```
<! --示例 9.1 程序-->
<! DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>示例 9.1</title>
</head>
<body>
<div>这是 div 标签形成的区域 1,单独占据一行。</div>
<div>这是 div 标签形成的区域 2,没有 CSS 样式影响,不具特性。</div>
</body>
</html>
```

本例浏览效果如图 9-1 所示。



图 9-1 标准流状态的 div 标签

以上示例中两个 div 标签都没有应用 CSS 样式，所以格局是一样的，透明的区域，独占一行。如果在 DIV 的格局上应用 CSS 样式，则 DIV 格局发生了变化。

【例 9.2】简单的 DIV+CSS。

```
<! --示例 9.2 程序-->
<! DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>示例 9.2</title>
<style>
.one {
    background-color:greenyellow;
}
.two {
    width:250px;
    border:dashed 2px red;
    font-size:large;
}
</style>
```

```
</head>
<body>
<div class="one">这是 div 标签形成的区域 1,单独占据一行。</div>
<div class="two">这是 div 标签形成的区域 2,没有 CSS 样式影响,不具特性。</div>
</body>
</html>
```

本例浏览效果如图 9-2 所示。



图 9-2 应用 css 的 div 标签

通过例 9.1 和例 9.2, 可以知道 div 标签没有什么特别之处, 也没有任何特性, 相当于一个简单的容器, 所以, 很容易被 CSS 代码控制样式。

9.1.2 DIV 的宽高设置

对 div 设置宽高样式, 即 div 宽度和高度同时设置。

1. 宽高属性

宽度: width

width: 250px——设置宽值为 250 像素。

width: 50%——设置宽值为父元素的百分之五十。

高度: height

height: 250px——设置高值为 250 像素。

height: 50%——设置高值为父元素的百分之五十。

2. div 标签内直接设置宽高

【例 9.3】 div 直接应用 CSS。

```
<!-- 示例 9.3 程序 -->
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>示例 9.3</title>
</head>
<body>
```

```
<div style=" width:250px;height:100px;border:2px solid red" >  
此 div 宽度 250px,高 100px  
</div>  
<div style=" width:150px;height:50px;border:2px dashed green" >  
此 div 宽度 150px,高度 50px  
</div>  
</body>  
</html>
```

本例浏览效果如图 9-3 所示。

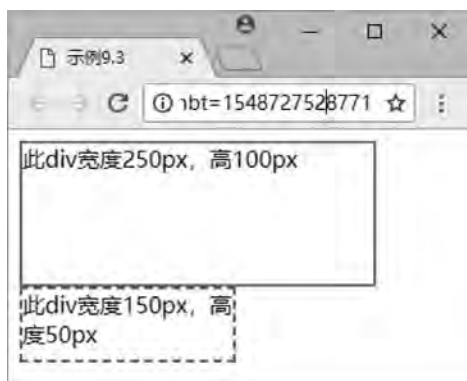


图 9-3 div 应用 CSS 样式设置宽高

3. div 使用选择器设置宽高

把 CSS 样式改为选择器形式，浏览效果也如图 9-3 所示。

【例 9.4】 div 应用选择器。

```
<! --示例 9.4 程序-->  
<! DOCTYPE html>  
<html>  
<head>  
<meta charset="utf-8" />  
<title>示例 9.4</title>  
<style>  
.na {  
width:250px;  
height:100px;  
border:2px solid red;  
}  
#na {  
width:150px;  
height:50px;  
border:2px dashed green;  
}  
</style>
```

```
</head>
<body>
<div class="na">
此div 宽度 250px,高 100px
</div>
<div id="na">
此div 宽度 150px,高度 50px
</div>
</body>
</html>
```

4. div 高度的百分比设置问题

如果 div 的宽度设置了百分比，则相对其父元素相应改变宽度。但是 div 的高度则无法根据百分比相应改变。

【例 9.5】 div 高度的百分比设置。

```
<! --示例 9.5 程序-->
<! DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>示例 9.5</title>
</head>
<body>
<div style="width:50%;height:25%;border:2px solid red">
此div 宽度 50%,高 25%
</div>
<div style="width:70%;height:50%;border:2px solid green">
此div 宽度 70%,高度 50%
</div>
</body>
</html>
```

本例浏览效果如图 9-4 所示。



图 9-4 div 百分比高度无法设置

以上示例中两个 div 标签都应用了宽高的百分比 CSS 样式，但是只有宽度发生了改变，高度并没有改变。

因为设置 div 的高度自适应是相对其父元素的高度而言，本例中 div 的父元素是浏览器本身，所以需要先定义父元素的宽高百分比。

在示例程序 9.5 的基础上，添加以下代码：

```
<style>
* {
    width:100%;
    height:100%;
}
</style>
```

得到效果图如图 9-5 所示。



图 9-5 div 百分比高度可以设置相应改变

9.2 DIV+CSS 的应用

标准流（normal flow）也叫常规流，文档流。在使用 div、span、p 标签进行布局时，默认就是使用标准流进行布局。

标准流是垂直布局，是由块元素及其行内元素构成的。从上到下、从左到右按顺序摆放好，默认情况下，互相之间不存在层叠现象。

9.2.1 DIV 元素的布局技巧

由于用户的计算机显示屏分辨率不同，因此在布局页面时，要充分考虑页面内容的布局宽度情况，并保证页面整体内容在页面居中。一旦内容宽度超过显示宽度，页面将出现水平滚动条。应尽量保证网页只有垂直滚动条，才符合用户的习惯，所以高度不需要考虑，一般都是由页面内容决定网页高度即可。

又因为浏览器的兼容情况，所以在布局页面前，设计者一定要把页面的默认边距清除。传统的表格布局时，可以使用属性“align: center;”设置表格居中问题，但是 DIV 的居中是没有属性可以设置的，只能通过 CSS 样式控制其位置。

使 div 元素水平居中的方法有多种，常用的方法是用 CSS 设置 div 的左右边距。

【例 9.6】 div 的居中布局。

```
<! --示例 9.6 程序-->
<! DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>示例 9.6</title>
<style>
* {
    margin:0;
    padding:0;                                /* 设置浏览器默认内外边距均为 0 */
}
#center{
    width:50%;
    height:100px;
    border:2px dashed red;
    margin-left:auto;
    margin-right:auto;
}
</style>
</head>
<body>
<div id="center">div 页面布局水平居中,div 内容不受影响。</div>
</body>
</html>
```

本例浏览效果如图 9-6 所示。



图 9-6 div 布局水平居中

以上示例中，设置 div 左外边距和右外边距的值为自动（auto）时，左外边距和右外边距将相等，即达到了 div 水平居中的效果。

如果把“margin: 0;”“margin-left: auto;”“margin-right: auto;”三个属性一并改为“margin: 0px auto;”，浏览效果也如图 9-6 所示。

或者在 div 标签外部直接添加 center 标签，也可以实现 div 水平居中的效果。

【例 9.7】 利用 center 标签实现 div 页面居中。

```
<! --示例 9.7 程序-->
<! DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>示例 9.7</title>
<style>
#center{
    width:50%;
    height:100px;
    border:2px dashed red;
}
</style>
</head>
<body>
<center>
    <div id="center">div 页面布局居中,div 内容受到影响,一并居中。</div>
</center>
</body>
</html>
```

本例浏览效果如图 9-7 所示。



图 9-7 div 及内容同时居中

以上示例中，使用 center 标签设置 div 页面布局居中时，div 文本内容也一并居中了。如果把代码直接改为“html, body { text-align: center; }”浏览效果也如图 9-7 所示。但是这两种方法将导致页面文本居中，所以不建议用。

9.2.2 DIV 元素的宽度自适应

有时会用到浮动效果，实现 DIV 元素的宽度自适应。宽度自适应是指两个并排的 div，其中左边的 div 为固定宽度，右边 div 则根据浏览器的宽度自动调整，这种用法常用于文章列表和文章内容的页面布局。

【例 9.8】 div 宽度自适应。

```
<! --示例 9.8 程序-->
<! DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>示例 9.8</title>
<style>
* {
    margin:0px;
    padding:0px;
}
.all {
    height:200px;
    border:2px dashed red;
}
.left {
    width:200px;
    height:100px;
    background-color:lightcyan;
    float:left;
}
.right {
    margin-left:200px;
    background-color:lightpink;
    height:100%;
}
</style>
</head>
<body>
<div class="all">
    <div class="left">左边 div</div>
    <div class="right">右边 div 宽度自适应</div>
</div>
</body>
</html>
```

本例浏览效果如图 9-8 所示。



图 9-8 右 div 宽度自适应

9.2.3 DIV 内容的居中

平时说的 div 内容居中，只是保持 div 内容的水平居中。但是很多网站需要用到的是水平和垂直居中。这里用行高（line-height）属性来实现，当行高的值等于容器的高度值，内容便垂直居中。

【例 9.9】 div 的内容居中。

```
<! --示例 9.9 程序-->
<! DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>示例 9.9</title>
<style>
.center{
    width:300px;
    height:80px;
    border:2px dotted darkorange;
    background-color:lemonchiffon;
    text-align:center;
    line-height:80px;
}
</style>
</head>
<body>
<div class="center">内容水平居中、垂直居中</div>
</body>
</html>
```

本例浏览效果如图 9-9 所示。



图 9-9 内容水平与垂直同时居中

9.2.4 DIV 元素的嵌套

传统的表格布局中，搜索引擎如果遇到多层表格嵌套时，可能抓取不到3层以上的内容，或者会跳过嵌套的内容直接放弃整个页面。

而DIV+CSS布局则不会存在这样的问题，为了实现复杂的布局结构，div元素也需要互相嵌套。但在布局页面时尽量少嵌套，否则将影响浏览器对代码的解析速度。

【例 9.10】 div 的嵌套布局。

```
<!-- 示例 9.10 程序 -->
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>示例 9.10</title>
<style>
.all{
    width:700px;
    height:700px;
    border:2px solid red;
    margin:0px auto;
}
.top{
    width:700px;
    height:100px;
    background-color:lightcyan;
}
.main{
    width:700px;
    height:520px;
}
.main.left{
    width:180px;
    height:500px;
```

```
background-color:yellow;
float:left;
margin:10px;
}
.main.right{
width:480px;
height:500px;
background-color:lightgreen;
float:left;
margin:10px;
}
.footer{
width:700px;
height:80px;
background-color:lightgrey;
}
</style>
</head>
<body>
<div class="all">
<div class="top"></div>
<div class="main">
<div class="left"></div>
<div class="right"></div>
</div>
<div class="footer"></div>
</div>
</body>
</html>
```

本例浏览效果如图 9-10 所示。



图 9-10 div 的嵌套布局

以上示例中为常见的 div 嵌套布局方法，div-all 嵌套 div-top、div-main、div-footer，而 div-main 又嵌套 div-left 和 div-right。

9.3 DIV+CSS 的典型布局

设计者为了让页面外观与结构分离，就要用 CSS 样式来规范布局。使用 CSS 样式可以让代码更加简洁和结构化，使站点的访问和维护更加容易。

网页设计的第一步是版面布局的设计。通过前面的学习，我们已经对页面布局的技巧有了基本理解。本节将结合前面知识，展示目前较为常用的 CSS 布局样式。

9.3.1 左中右布局

如图 9-11 所示，左中右布局在网页设计时最为常用，即 div-left 是导航菜单，div-main 是视觉集中点，放置主要内容，div-right 是表单或链接等。而左中右三个区域一并被包含在一个大的 div-all 中。



图 9-11 div 的左中右布局

【例 9.11】 div 的左中右布局。

```
<! --示例 9.11 程序-->  
<! DOCTYPE html>
```

```
<html>
<head>
<meta charset="utf-8" />
<title>示例 9.11</title>
<style>
* {
    margin:10px auto;
    padding:0px auto;
    font-size:large;
}
.all {
    background-color:red;
    width:700px;
    height:500px;
}
.left,.main,.right {
    text-align:center;
    height:480px;
    line-height:480px;
    float:left;
}
.left {
    background-color:antiquewhite;
    width:150px;
}
.main {
    background-color:lightcyan;
    width:400px;
}
.right {
    background-color:antiquewhite;
    width:150px;
}
</style>
</head>
<body>
<div class="all">
    <div class="left">导航菜单</div>
    <div class="main">视觉集中区域,主要内容</div>
    <div class="right">表单或链接</div>
</div>
</body>
</html>
```

9.3.2 上中下布局

如图 9-12 所示, 上中下布局符合许多网站共同的特点, 即 `div-top` 是导航或者横幅广告, `div-main` 是视觉集中点, 放置主要内容, `div-footer` 是版权信息等。而上中下三个区域一并被包含在一个大的 `div-all` 中。



图 9-12 div 的上中下布局

【例 9.12】 div 的上中下布局。

```
<!--示例 9.12 程序-->
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>示例 9.12</title>
<style>
* {
margin:0px auto;
padding:0px auto;
font-size:large;
}
```



```
. all {
    background-color: red;
    text-align: center;
    width: 700px;
    height: 500px;
}
. top {
    background-color: antiquewhite;
    width: 680px;
    height: 80px;
    line-height: 80px;
}
. main {
    background-color: lightcyan;
    width: 680px;
    height: 340px;
    line-height: 340px;
}
. footer {
    background-color: antiquewhite;
    width: 680px;
    height: 80px;
    line-height: 80px;
}
</style>
</head>
<body>
<div class = " all" >
    <div class = " top" >导航或者横幅广告</div>
    <div class = " main" >视觉集中区域,主要内容</div>
    <div class = " footer" >版权信息</div>
</div>
</body>
</html>
```

9.3.3 混合布局

在了解左中右和上中下布局之后，混合布局也就不难理解了，混合布局也可以叫综合型布局。混合布局可以在一列或一行布局的基础之上，分为多列或多行布局。网页布局的结构普遍都是三列布局，或者可以根据实际需求，对网页再进行划分。图9-13所示，就是常见的混合布局。

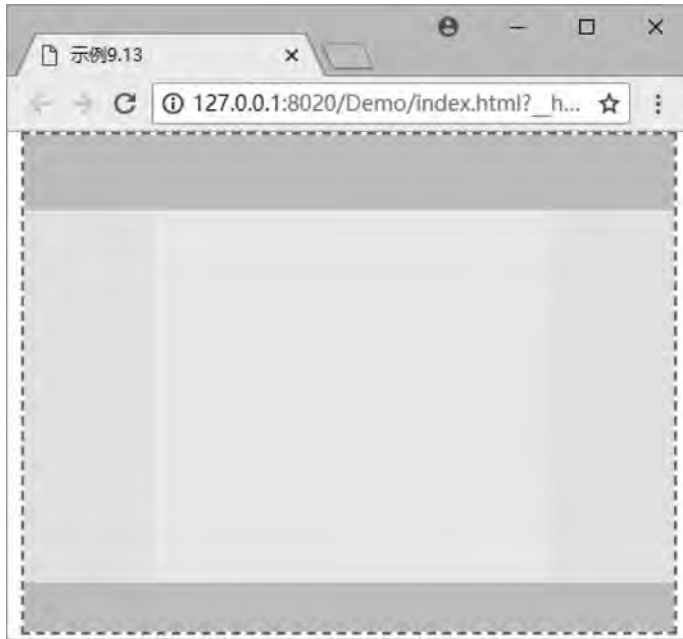


图 9-13 div 的混合布局

【例 9.13】div 的混合布局。

```
<!-- 示例 9.13 程序 -->
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>示例 9.13</title>
<style>
* {
    margin:0px auto;                /* 设置浏览器元素水平居中 */
    padding:0px auto;
    width:100%;
    height:100%;                    /* 先设置父元素的宽度和高度为 100%。 */
}
.all {
    border:2px dashed red;
    width:95%;
    height:100%;
}
.top {
    background-color:pink;
    width:100%;
    height:15%;
}
```

```
. main {
    width:100%;
    height:75%;
}
. left {
    background-color:yellow;
    width:20%;
    float:left;
}
. center {
    background-color:lightcyan;
    width:60%;
    float:left;
}
. right {
    background-color:yellow;                /* 中间 div 的右侧 div 宽度自适应 */
}
. footer {
    background-color:pink;
    width:100%;
    height:10%;
}
</style>
</head>
<body>
<div class="all">
    <div class="top"></div>
    <div class="main">
    <div class="left"></div>
    <div class="center"></div>
    <div class="right"></div>
    </div>
    <div class="footer"></div>
</div>
</body>
</html>
```

9.4 综合案例——众成远程教育

本章综合前面学习的 DIV+CSS 布局知识，可以利用各种 div 布局技巧制作综合型布局的网页。本案例以图 9-13 的布局为例，制作“众成远程教育”网页，本章项目页面布局要求如下：页面要求有最外层的 div-all，第二层嵌套上中下 3 行区域，分别为 div-top、div-main



和 div-footer。而 div-main 又第三层嵌套左中右三列区域，分别为 div-left、div-center 和 div-right。布局代码如下：

```
<! --综合案例——布局.html-->
<! DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>众成远程教育——布局</title>
<style>
* {
    margin:0px auto;
    font-family:" 宋体";
    font-size:30px;
    text-align:center;
    font-weight:700;
}
.all {
    width:1000px;
    height:840px;
    background-image:url(img/9-bg.jpg);
}
.top {
    width:1000px;
    height:150px;
}
.main {
    background-color:aliceblue;
    width:1000px;
    height:630px;
}
.left {
    width:200px;
    height:630px;
    float:left;
}
.center {
    border-left:2px dashed blue;
    border-right:2px dashed blue;
    width:500px;
    height:630px;
    float:left;
}
.right {
```

```
width:250px;
height:630px;
float:left;
}
.footer{
width:1000px;
height:60px;
}
</style>
</head>
<body>
<div class="all">
<div class="top">top</div>
<div class="main">
<div class="left">left</div>
<div class="center">center</div>
<div class="right">right</div>
</div>
<div class="footer">footer</div>
</div>
</body>
</html>
```

本例浏览效果如图 9-14 所示。

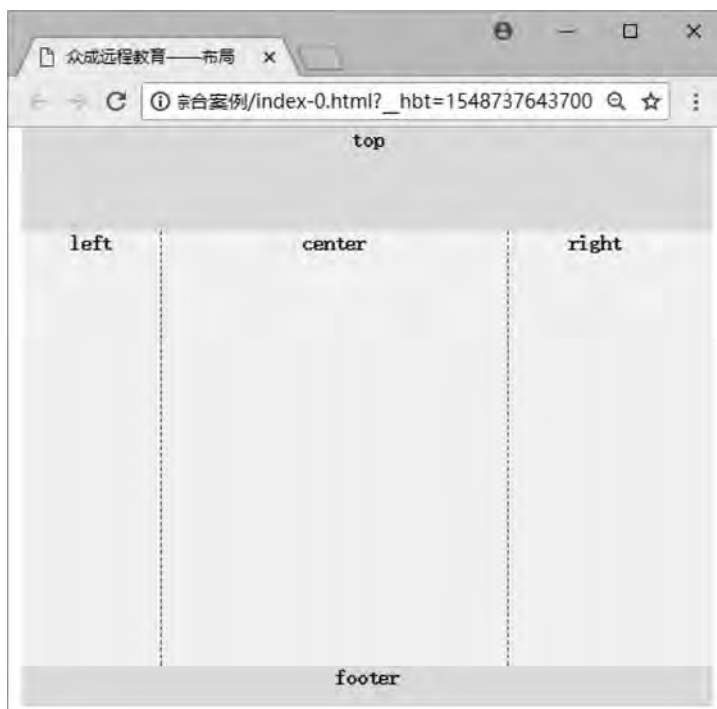


图 9-14 网页布局结构图



(1) 为 div-top 添加 logo 图片。代码如下:

```
<body>
<div class="top">
  
</div>
</body>
```

(2) 为 div-footer 添加文字信息, 并设置相应 CSS 样式。代码如下:

```
<style>
.footer {
  width:1000px;
  height:60px;
  font-family:"楷体";
  font-size:18px;
  text-align:center;
  line-height:30px;
}
span {
  color:red;
  font-weight:700;
  text-align:center;
}
</style>
<body>
<div class="footer">
  <span>免费电话:</span>400-XXX-XXX(18条线)||
  <span>(北京校区)</span>北京路 XX 大厦一楼 0000 号;||
  <span>(上海校区)</span>上海路 XX 科技园 7 栋 9999 号<br>
  此网站信息最终解释权 &copy;众成远程教育
</div>
</body>
```

(3) 为 div-main 的 div-left 添加图文信息, 并设置相应 CSS 样式。代码如下:

```
<style>
.left {
  padding-top:30px;
  padding-left:20px;
  width:200px;
  height:570px;
  float:left;
  line-height:60px;
}
a {
  color:red;
  font-weight:700;
```

```

        text-align:center;
    }
    p{
        font-family:"黑体";
        font-weight:700px;
        color:brown;
        font-size:28px;
        text-align:center;
    }
</style>
<body>
<div class="left">
    <p></p>
    <p></p>
    <p></p>
    <p></p>
    <p>相关信息</p>
    <a href="#">4 大学学历提升方案</a><br>
    <a href="#">新报考政策解读</a><br>
    <a href="#">6 大类专业报考指南</a><br>
    <a href="#">更多信息请点击</a>
</div>
</body>

```

(4) 为 div-main 的 div-center 添加表单和表格信息，并设置相应 CSS 样式。代码如下：

```

<style>
.center{
    border-left:2px dashed blue;
    border-right:2px dashed blue;
    padding-top:50px;
    width:500px;
    height:580px;
    float:left;
}
table{
    font-family:"黑体";
    font-weight:700px;
    color:blue;
    font-size:20px;
    line-height:55px;
}
</style>
<body>
<div class="center">

```

```
<p>入学报名表</p>
<form id="form2" name="form2" method="post" action="" >
  <table width="400" border="0" align="center" cellpadding="0" cellspacing="0">
    <tr>
      <td width="158" align="right">姓名 :<label for="textfield"></label>
      </td>
      <td width="242"><input type="text" name="textfield" id="textfield" />
      </td>
    </tr>
    <tr>
      <td align="right">联系电话:
      </td>
      <td><input type="text" name="textfield2" id="textfield2" />
      </td>
    </tr>
    <tr>
      <td align="right">邮箱:
      </td>
      <td><input type="text" name="textfield3" id="textfield3" />
      </td>
    </tr>
    <tr>
      <td align="right">资料邮寄地址:
      </td>
      <td><input type="text" name="textfield4" id="textfield4" />
      </td>
    </tr>
    <tr>
      <td align="right">最高学历:
      </td>
      <td>
        <select name="select2" id="select2">
          <option>大学本科</option>
          <option>大专</option>
          <option>高中</option>
          <option>初中</option>
          <option>小学</option>
        </select>
      </td>
    </tr>
    <tr>
      <td align="right">选择的课程:
      </td>
```



```
        <td><input type="text" name="textfield6" id="textfield6" />
      </td>
    </tr>
  <tr>
    <td align="right">意向学习方式:
      <label for="select2"></label>
    </td>
    <td>
      <select name="select" id="select">
        <option>网络授课</option>
        <option>周末班</option>
        <option>全日制</option>
      </select>
    </td>
  </tr>
  <tr>
    <td colspan="2" align="center">
      <input type="image" name="imageField" id="imageField" src="img/but1.jpg" />
    </td>
  </tr>
</table>
</form>
</div>
</body>
```

(5) 为 div-main 的 div-right 添加图片信息，并设置相应 CSS 样式。代码如下：

```
<style>
.right {
  padding-left:20px;
  width:250px;
  height:630px;
  float:left;
}
</style>
<body>
<div class="right">
  
  
  
  
</div>
</body>
```

完成以上步骤，综合案例最终网页浏览效果如图 9-15 所示。



图 9-15 众成远程教育网页浏览图

以下为完整代码，以供参考。

```
<! --综合案例.html-->
<! DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>众成远程教育</title>
<style>
* {
    margin:0px auto;
}
.all {
    width:1000px;
    height:840px;
    background-image:url(img/9-bg.jpg);
}

```

```
. top {
    width: 1000px;
    height: 150px;
}
. main {
    background-color: aliceblue;
    width: 1000px;
    height: 630px;
}
. left {
    padding-top: 30px;
    padding-left: 20px;
    width: 200px;
    height: 570px;
    float: left;
    line-height: 60px;
}
. center {
    border-left: 2px dashed blue;
    border-right: 2px dashed blue;
    padding-top: 50px;
    width: 500px;
    height: 580px;
    float: left;
}
. right {
    padding-left: 20px;
    width: 250px;
    height: 630px;
    float: left;
}
. footer {
    width: 1000px;
    height: 60px;
    font-family: "楷体";
    font-size: 18px;
    text-align: center;
    line-height: 30px;
}
a, span {
    color: red;
    font-weight: 700;
```

```
        text-align:center;
    }
    p{
        font-family:"黑体";
        font-weight:700px;
        color:brown;
        font-size:28px;
        text-align:center;
    }
    table{
        font-family:"黑体";
        font-weight:700px;
        color:blue;
        font-size:20px;
        line-height:55px;
    }
</style>
</head>
<body>
<div class="all">
    <div class="top">
        
    </div>
    <div class="main">
        <div class="left">
            <p></p>
            <p></p>
            <p></p>
            <p></p>
            <p>相关信息</p>
            <a href="#">4 大学学历提升方案</a><br>
            <a href="#">新报考政策解读击</a><br>
            <a href="#">6 大类专业报考指南</a><br>
            <a href="#">更多信息请点击</a>
        </div>
        <div class="center">
            <p>入学报名表</p>
            <form id="form2" name="form2" method="post" action="" >
                <table width="400" border="0" align="center" cellpadding="0" cellspacing="0" >
                    <tr>
                        <td width="158" align="right">姓名:<label for="textfield"></label>
                    </td>
```

```
<td width="242"><input type="text" name="textfield" id="textfield" />
</td>
</tr>
<tr>
<td align="right">联系电话:
</td>
<td><input type="text" name="textfield2" id="textfield2" />
</td>
</tr>
<tr>
<td align="right">邮箱:
</td>
<td><input type="text" name="textfield3" id="textfield3" />
</td>
</tr>
<tr>
<td align="right">资料邮寄地址:
</td>
<td><input type="text" name="textfield4" id="textfield4" />
</td>
</tr>
<tr>
<td align="right">最高学历:
</td>
<td>
<select name="select2" id="select2">
<option>大学本科</option>
<option>大专</option>
<option>高中</option>
<option>初中</option>
<option>小学</option>
</select>
</td>
</tr>
<tr>
<td align="right">选择的课程:
</td>
<td><input type="text" name="textfield6" id="textfield6" />
</td>
</tr>
<tr>
<td align="right">意向学习方式:
```

```

        <label for="select2"></label>
    </td>
    <td>
        <select name="select" id="select">
            <option>网络授课</option>
            <option>周末班</option>
            <option>全日制</option>
        </select>
    </td>
</tr>
<tr>
    <td colspan="2" align="center">
        <input type="image" name="imageField" id="imageField" src="img/
but1.jpg" />
    </td>
</tr>
</table>
</form>
</div>
<div class="right">
    
    
    
    
</div>
</div>
<div class="footer">
    <span>免费电话:</span>400-XXX-XXX(18条线)||
    <span>(北京校区)</span>北京路XX大厦一楼0000号;||
    <span>(上海校区)</span>上海路XX科技园7栋9999号<br>
    此网站信息最终解释权 &copy;众成远程教育
</div>
</div>
</body>
</html>

```

本章习题

1. 用 DIV+CSS 布局技术，制作一个“服务中心”页面。图 9-16 是布局结构图，图 9-17 是网页效果图。

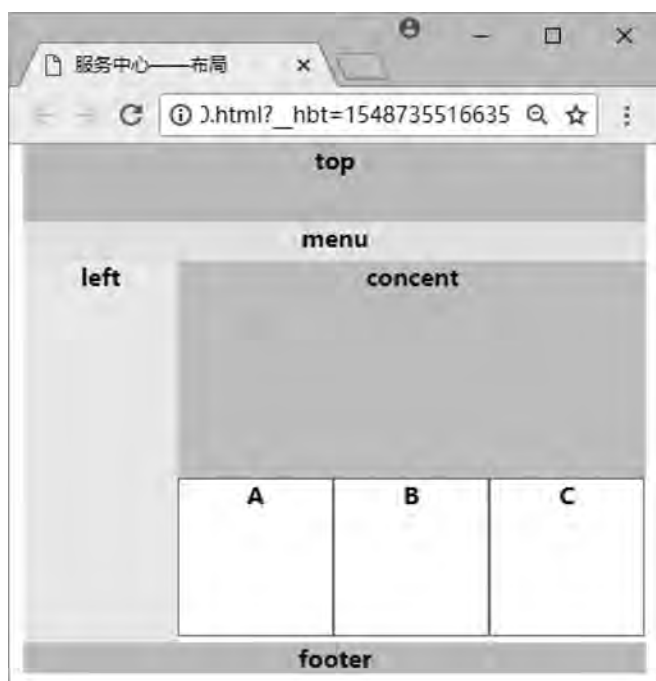


图 9-16 题 1 布局结构图



图 9-17 题 1 网页效果图

2. 用 DIV+CSS 布局技术, 制作一个“飞跃岩石”页面。图 9-18 是布局结构图, 图 9-19 是网页效果图。



图 9-18 题 2 布局结构图



图 9-19 题 2 网页效果图

第 10 章 JavaScript 的应用



学习目标

1. 掌握 JavaScript 的基本语法；
2. 学会编写和修改一些基本的 JavaScript 代码；
3. 了解 JavaScript 对象及事件的概念。



知识要点

1. JavaScript 的概念和基本语法；
2. JavaScript 对象及应用；
3. JavaScript 事件及应用。

采用 HTML+CSS 技术设计的网页具有信息丰富、样式美观等优势，但是对网页的交互行为却无能为力，此时脚本语言 JavaScript 提供了解决方案。本章讲述的就是实现网页交互与特效的行为语言——JavaScript。

10.1 JavaScript 概述

10.1.1 JavaScript 简介

JavaScript 是一种基于对象（Object）和事件驱动（Event Driven）并具有安全性能的脚本语言，能够与 HTML（超文本标记语言）、Java 语言一起在 Web 页面中与 Web 客户交互，它无须经过先将数据传给服务器端（Server）、再传回来的过程，而直接可以由客户端（Client）的应用程序处理，从而可以开发客户端的应用程序等。JavaScript 是通过嵌入或调入在标准的 HTML 语言中实现的，弥补了 HTML 语言的缺陷，是 Java 与 HTML 折中的选择，通常具有以下特点。

1. 简单性

JavaScript 是一种脚本语言，采用小程序段的方式实现编程。像其他脚本语言一样，JavaScript 同样是一种解释性语言，提供了一个简易的开发过程。

2. 动态性

JavaScript 是动态的，可以直接响应用户或客户的输入，无须经过 Web 服务程序，通过



事动来响应用户的请求。

3. 跨平台性

JavaScript 依赖浏览器，与操作环境无关，只要能运行浏览器的计算机、支持 JavaScript 的浏览器就可以正确执行。

4. 安全性

JavaScript 是一种安全性语言，不允许访问本地的硬盘，并不能将数据存入到服务器上，不允许对网络文档进行修改和删除，只能通过浏览器实现信息浏览或动态交互，可有效地防止数据的丢失。

5. 基于对象的语言

JavaScript 是一种基于对象的语言，同时可以看作是一种面向对象的语言。这意味着它能将特定功能封装在对象中。

10.1.2 JavaScript 入门案例

JavaScript 程序不能独立运行，必须依赖 HTML 文件，通常将 JavaScript 代码放置在 script 标记内，由浏览器 JavaScript 脚本引擎来解释执行。

语法：

```
<script type = "text/javascript" >
//(JavaScript 代码)
</script>
```

说明：

script 标记是成对标记，以<script>开始，以</script>结束。type 属性说明脚本的类型，属性值“text/javascript”意思是使用 Javascript 编写的程序是文本文件。script 标记既可以放在 HTML 的头部，也可以放在 HTML 的主体部分，只是装载的时间不同。

下面来看一个学习所有语言时都会提到的入门案例。

【例 10.1】 JavaScript 入门案例。

```
<! --示例 10.1 程序-->
<! DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>JavaScript 入门案例</title>
</head>
<body>
<script type="text/javascript" >
    document.write( 'Hello World! ');
</script>
</body>
</html>
```

本例浏览效果如图 10-1 所示。



图 10-1 JavaScript 入门案例

以上示例中 `document.write()` 是文档对象的输出函数，其功能是将括号中的字符或变量值输出到窗口。

10.1.3 JavaScript 放置的位置

JavaScript 代码一般放置在页面的 `head` 或 `body` 部分。当页面载入时，会自动执行位于 `body` 部分的 JavaScript；而位于 `head` 部分的 JavaScript 只有被显式调用时才会被执行。

1. head 标记中的脚本

`script` 标记放在头部 `head` 标记中，通 JavaScript 代码必须定义成函数形式，并在主体 `body` 标记内调用或通过事件触发。放在 `head` 标记内的脚本在页面装载时同时载入，这样在主体标记内调用时可以直接执行，提高了脚本执行速度。

语法：

```
function functionname( 参数 1, 参数 2, …… , 参数 n ) {  
    函数体语句;  
}
```

说明：

JavaScript 自定义函数必须以 `function` 关键字开始，然后给自定义函数命名，函数命名时一定遵守标识符命名规范。函数名称后面一定要有一对括号“`()`”，括号内可以有参数，也可以无参数，多个参数之间用逗号“`,`”分隔。函数体语句必须放在大括号“`{ }`”内。

【例 10.2】 在 `head` 标记内定义两个 JavaScript 函数。

```
<!-- 示例 10.2 程序 -->  
<! DOCTYPE html>  
<html>  
<head>  
<meta charset="UTF-8">  
<title>head 中定义的 JS 函数</title>  
<script type="text/javascript">  
function message() {  
    alert("调用 JS 函数! sum(100,200)=" + sum(100,200));  
}  
function sum(x,y) {  
    return x+y;  
} // 返回函数计算结果
```

```

</script>
</head>
<body>
  <h4>head 标记内定义两个 JS 函数</h4>
  <p>无返回值函数: message() </p>
  <p>有返回值函数: sum(x,y) </p>
  <form>
    <input name = " btncal" type = " button" onclick = " message ( );" value = " 计算并显示两个数的
和" >
  </form>
</body>
</html>

```

本例浏览效果如图 10-2 所示。



图 10-2 调用 head 标记中定义的 JavaScript 函数

以上示例中在 head 部分插入 script 标记，在 script 标记内定义 JavaScript 函数 message ()、sum (x, y)。alert () 函数是弹出告警消息框，并调用 sum (100, 200) 函数，计算出结果并输出相关信息；body 中定义了一个普通按钮 btncal，当单击该按钮时触发按钮的 onclick 事件，调用在 head 部分定义的 message 函数，弹出告警框。

2. body 标记中的脚本

script 标记放在主体 body 标记中，JavaScript 代码可以定义成函数形式，在主体 body 标记内调用或通过事件触发。也可以在 script 标记内直接编写脚本语句，在页面装载时同时执行相关代码，这些代码执行的结果直接构成网页的内容，在浏览器中可以查看。

【例 10.3】 在 body 标记中添加 JavaScript 脚本。

```

<! --示例 10.3 程序-->
<! DOCTYPE html>
<html>
<head>
<meta charset = " UTF-8" >
<title>body 中的 JavaScript 脚本</title>
</head>
<body>

```

```
<p id="clk">Click Here</p>
<script type="text/javascript">
var demo=document.getElementById("clk");
demo.onclick=msg;
function msg(){
    alert("我是 body 中的 JavaScript 脚本");
}
</script>
</body>
</html>
```

本例浏览效果如图 10-3 所示。



图 10-3 body 中的 JavaScript 脚本

以上示例中，为 p 元素添加了一个 id 属性，是为了使 JavaScript 脚本方便获取该元素。通过 document.getElementById("clk") 方法就可根据 id 访问这个元素，该方法返回的结果是一个 DOM 对象：demo。然后，通过“DOM 对象.事件名=函数名”就能设置该对象在事件发生时将执行的函数；当用户点击“Click Here”文本时，就会弹出如图 10-3 所示的告警框，

3. 外部 js 文件中的脚本

除了将 JavaScript 代码写在 head 和 body 部分以外，也可将 JavaScript 函数单独写成一个 js 文件，在 HTML 文档中引用该 js 文件。

【例 10.4】调用外部 js 文件中的 JavaScript 函数。

js/demo.js 文件的代码如下：

```
/* -- demo.js */
function message()
{
    alert("调用外部 js 文件中的函数!");
}

```

示例程序 10.4 文件的代码如下：

```
<!-- 示例 10.4 程序 -->
<!DOCTYPE html>
<html>
```

```

<head>
<meta charset="UTF-8">
<title>调用外部 js 文件的 JavaScript 函数</title>
<script type="text/javascript" src="js/demo.js">
    document.write("这条语句没有执行,被忽略掉了!");
</script>
</head>
<body>
    <form>
        <input name="btn1" type="button" onclick="message()" value="调用外部 js 文件的 JavaScript
函数">
    </form>
</body>
</html>

```

本例浏览效果如图 10-4 所示。



图 10-4 调用外部 js 文件的 JavaScript 函数

以上示例中将 JavaScript 函数写在一个文件 demo.js 中，代码中定义了一个函数 message，注意在“js”文件中不需要使用<script></script>标记来包围代码。当点击“示例程序 10.4”中的按钮时，触发 onclick 事件，执行 demo.js 中定义的 message（）函数实现在页面上弹出告警框的功能，如图 10-4 所示。

4. 事件处理代码中的脚本

JavaScript 代码除了上述三种放置位置外，还可直接写在事件处理代码中。

【例 10.5】调用直接写在事件处理代码中 JavaScript 程序。

```

<! --示例 10.5 程序-->
<! DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>直接在事件处理代码中加入 JavaScript 代码</title>
</head>

```

```
<body>
  <form>
    <input type="button" onclick="alert('直接在事件处理代码中加入 JavaScript 代码')" value="
直接调用 JavaScript 代码">
  </form>
</body>
</html>
```

本例浏览效果如图 10-5 所示。

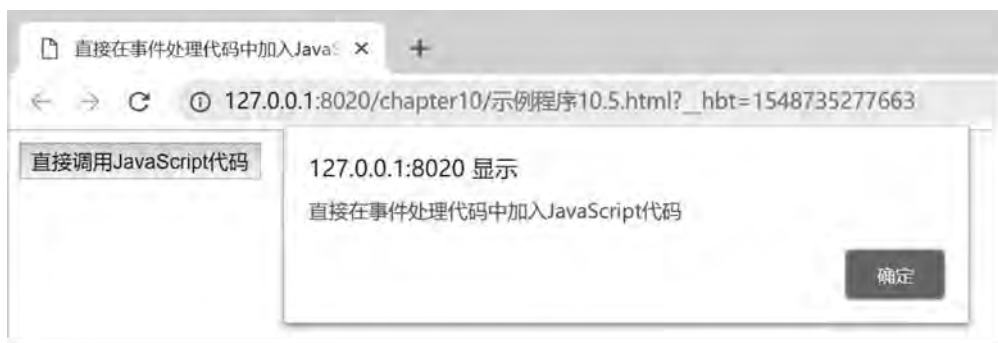


图 10-5 调用事件处理代码中的 JavaScript 脚本

以上示例中直接在普通按钮的 onclick 事件中插入了 JavaScript 代码，注意 JavaScript 代码需要用双引号 (" ") 括起来，单击该按钮时弹出告警框。

用浏览器打开 JavaScript 程序时，安全级别设置较高的浏览器会阻止程序的运行，应选择“允许阻止的内容”选项，才能运行。

10.2 JavaScript 语法

10.2.1 语法基础

任何语言的语法学习都是枯燥、单调的，JavaScript 语法也一样，但学好语法是更好地使用 JavaScript 编写程序的前提。

在运用 JavaScript 语言编写程序时应注意以下几点：

1. 区分大小写

在 JavaScript 语言中，对大小写是敏感的，这在现在的很多开发语言中都是如此，所以读者要养成一种良好的习惯，这样就可以避免调试程序时遇到这些低级错误。

2. 变量不区分类型

JavaScript 中变量声明，采用其弱类型，即变量在使用前无须进行变量数据类型声明，而由解释器在运行时检查其数据类型，例如：

```
x = 1234;
```

```
y="1234"
```

根据 JavaScript 变量的申明方式,可知前者说明 x 为数值型变量,而后者说明 y 为字符串型变量。

3. 每行代码结尾可以省略分号

在 JavaScript 里面每条语句结尾的分号“;”不是必要的,如果多个语句写在一行上,那么语句之间的分号“;”才是必需的,最后一条语句的分号“;”可以省略。为了养成良好的编程习惯,建议还是在每条语句后面加上分号。

4. 注释与 C、C++、Java 等语言相同

注释就是对代码的解释和说明,目的是为了别人和自己容易读懂程序。JavaScript 的注释形式与 C、C++、Java 等语言相同,格式如下:

```
// 单行注释
/*
多行注释
*/
```

10.2.2 标识符和常用变量

1. 标识符

标识符是计算机语言中用来表示变量名、函数名等的有效字符序列,简单来说,标识符就是一个名字,JavaScript 关于标识符的规定如下:

- (1) 必须使用英文字母或者下划线开头。
- (2) 必须使用英文字母、数字、下划线组成,不能出现空格或制表符。
- (3) 不能使用 JavaScript 关键字与 JavaScript 保留字。
- (4) 不能使用 JavaScript 语言内部的单词,比如 Infinity, NaN, undefined 等。
- (5) 大小写敏感,如 name 和 Name 是不同的两个标识符。

合法的标识符: Hello、_12th、sum、Dog23 等。

不合法的标识符: if、3com、case、switch 等。

2. 变量声明

在 JavaScript 语言环境中,变量是用于存储信息的容器,关键字 var 可应用于 JavaScript 中任何类型的变量声明。JavaScript 是采用的隐式变量声明,但要注意在 JavaScript 语言的任何位置进行隐式变量声明会将变量声明为全局变量,而函数内的局部变量则必须使用 var 来声明变量,因此,在声明变量时,需要根据变量的作用范围来声明变量。

变量声明格式如下:

```
var count; //单个声明
var count,amount,level; //用单个 var 关键字声明的多个变量
var count=0,amount=100; //一条语句中的变量声明和初始化
```

3. 变量类型

JavaScript 有 6 种数据类型,主要的类型有 Number 数值型、String 字符型、Object 对象

以及 Boolean 布尔型，其他两种类型为 Null 和 Undefined。

(1) Number 数值型。JavaScript 支持整数和浮点数。整数可以为正数、0 或负数；浮点数包含小数点，例如 3.53、-534.87 等都是浮点数。浮点数还可以采用科学计数法进行表示，如 3.5E15 表示 3.5×10^{15} 。

(2) String 字符型。字符型数据又称为字符串型，由若干个字符组成，并且用单引号或双引号封装起来，例如："你好!" "this is string" "学习语言"。

在使用字符串的过程中，有时会遇到一种情况：在一个字符串中需要使用单引号或双引号。正确的方法是在由双引号标记的字符串中引用字符时使用单引号，在由单引号标记的字符串中引用字符时使用双引号，即单引号、双引号必须成对使用相互包含，但不能交叉。例如：

```
"热烈欢迎参加JavaScript技术研讨的专家"; //正确的用法
"学习不是一件容易"的事件'; //错误的用法
```

(3) Boolean 布尔型。Boolean 值有 true 和 false，这是两个特殊值，可以将他们转换为其他类型的数据，例如可将值为 true 的布尔型数据转换为整数 1，而将值为 false 的布尔型数据转换为整数 0。但不能用 true 表示 1 或 false 表示 0。

(4) Undefined 数据类型。JavaScript 会对未赋值变量赋值为 undefined。

(5) Null 数据类型。这是一个对象，但是为空。因为是对象，所以 typeof null 返回“Object”，注意 null 是 Javascript 保留关键字。

(6) Object 类型。除了上面提到的各种常用类型外，对象类型也是 JavaScript 中的重要组成部分。

10.2.3 运算符与表达式

JavaScript 运算符主要有：算术运算符、关系运算符、逻辑运算符、赋值运算符、条件运算符、逗号运算符等。根据操作数的个数，可以将运算符分为一元运算符、二元运算符和三元运算符。

由操作数（变量、常量、函数调用等）和运算符结合在一起构成的式子称为“表达式”，最简单的表达式可以是常量名称。

1. 算术运算符和表达式

算术运算符负责算术运算，用算术运算符和运算对象（操作数）连接起来符合规则的式子，称为算术表达式。JavaScript 中常用算术运算符见表 10-1。

表 10-1 算术运算符

| 运算符 | 描述 | 示例（假设 a = 2） | 结果 |
|-----|----|--------------|-------|
| + | 加法 | b = a+2 | b = 4 |
| - | 减法 | b = a-1 | b = 1 |
| * | 乘法 | b = a * 2 | b = 4 |

续表

| 运算符 | 描述 | 示例 (假设 a = 2) | 结果 |
|-----|----|----------------------|-------|
| / | 除法 | b = a / 2 | b = 1 |
| % | 取模 | b = a%2 | b = 0 |
| ++ | 递增 | b = a++ (后置, 先使用再运算) | b = 2 |
| -- | 递减 | b = --a (前置, 先运算再使用) | b = 1 |

2. 关系运算符和表达式

关系运算符用于比较运算符两端的表达式的值, 确定二者的关系, 根据运算结果返回一个布尔值。用关系运算符和操作数连接起来符合规则的式子, 称为关系表达式。JavaScript 中常用关系运算符见表 10-2。

表 10-2 关系运算符

| 运算符 | 描述 | 示例 (假设 a = 5, b = 8) | 结果 |
|-----|------------|----------------------|-------|
| == | 等于 | a == 5 | true |
| === | 全等于 (值和类型) | a === " 5" | false |
| > | 大于 | a > b | false |
| >= | 大于等于 | a >= b | false |
| < | 小于 | a < b | true |
| <= | 小于等于 | a <= b | true |
| != | 不等于 | a != b | true |
| !== | 不全等于 | a !== b | true |

3. 逻辑运算符和表达式

逻辑运算符用来执行逻辑运算, 其操作数都应该是布尔型数值和表达式或者是可以转为布尔型的数值和表达式, 其运算结果返回 true 或 false。用逻辑运算符和操作数连接起来符合规则的式子, 称为逻辑表达式。JavaScript 中常用逻辑运算符见表 10-3。

表 10-3 逻辑运算符

| 运算符 | 描述 | 示例 (假设 a = 5, b = 8) | 结果 |
|-----|---------|----------------------|-------|
| && | 与 (and) | a < 10 && b > 10 | flase |
| | 或 (or) | a < 10 b > 10 | true |
| ! | 非 (not) | !(a == b) | true |

4. 赋值运算符和表达式

赋值运算符是 JavaScript 中使用频率最高的运算符之一。赋值运算符要求其左操作数是一个变量、数组元素或对象属性, 右操作数是一个任意类型的值, 可以为常量、变量、数组

元素或对象属性。赋值运算符的作用就是将右操作数的值赋给左操作数。用赋值运算符和操作数连接起来符合规则的式子，称为赋值表达式。JavaScript 中常用赋值运算符见表 10-4。

表 10-4 赋值运算符

| 运算符 | 描述 | 示例 (假设 a = 10, b = 5) | 结果 |
|-----|------|-----------------------|-------------------|
| = | 赋值 | a = 10, b = 5 | a 的值为 10, b 的值为 5 |
| += | 加法赋值 | a += b | a = 15 |
| -= | 减法赋值 | a -= b | a = 5 |
| *= | 乘法赋值 | a *= b | a = 50 |
| /= | 除法赋值 | a /= b | a = 2 |
| %= | 取余赋值 | a %= b | a = 0 |

5. 条件运算符和表达式

条件运算符是一个三元运算符，条件表达式由条件运算符和三个操作数构成。

语法：

变量 = 表达式 1 ? 表达式 2 : 表达式 3

说明：

该条件表达式表示，如果表达式 1 的结果为真 (true)，则将表达式 2 的值赋给变量，否则将表达式 3 的值赋给变量。例如，变量 number1、number2 比较大小，将较大的数赋值给变量 max，代码如下：

```
var max = (number1 > number2) ? number1 : number2;
```

6. 逗号运算符和表达式

逗号运算符是一个二元运算符，逗号运算符的优先级最低，由逗号运算符和操作数连接起来符合规则的式子，称为逗号表达式，其运算规则是先计算第一个表达式的值，再计算第二个表达式的值，依次类推，计算完所有表达式的值，运算结果为最后一个表达式的值。例如：

```
var rs = (3+5, 10*6); /* 先计算第一个表达式 3+5 的值为 8, 再计算第二个表达式 10*6 的值为 60, 最后将第二个表达式的值 60 赋给变量 rs */
```

10.2.4 程序设计

JavaScript 脚本语言的基本构成是由控制语句、函数、对象等来实现编程的，其中控制语句主要有条件分支语句和循环语句。

1. 条件分支语句

if...else 语句完成了程序流程块中分支功能：如果其中的条件成立，则程序执行紧接着条件的语句或语句块；否则；程序执行 else 中的语句或语句块。其语法如下：

```
if(条件)
```

```
{
    /* 条件为真执行的语句 */
}
else
{
    /* 条件为假执行的语句 */
}
```

if…else 语句还有一些变形语句。例如，只有 if 部分省略 else 分支的语句；if… else if … else…多分支语句等。

switch 分支语句可以根据一个变量或表达式的不同取值采取不同的处理方法，其语法如下：

```
switch(表达式)
{
    case case1:执行语句 1;break;
    case case2:执行语句 2;break;
    case case3:执行语句 3;break;
    default:默认执行语句;
}
```

如果表达式取的值同程序中提供的任何一条语句都不匹配，将执行 default 中的语句。关键字 break 会使代码跳出 switch 语句，如果没有关键字 break，代码就会继续进入下一个 case 分支。

2. 循环语句

JavaScript 中的循环用来控制同一段代码执行的次数。for 语句是最基本的循环语句，语法如下：

```
for(变量=开始值;变量<=结束值;变量=变量+步进值)
{
    /* 循环体语句 */
}
```

只要循环的条件成立，循环体就被反复的执行。

for…in 语句与 C#中 foreach () 语句很像，它循环的范围是一个对象所有的属性或是一个数组的所有元素。for…in 语句的语法如下：

```
for(变量 in 对象或数组)
{
    /* 循环体语句 */
}
```

while 语句所控制的循环，不断地测试条件，如果条件成立，则一直循环，直到条件不再成立。其语法如下：

```
while(条件)
```

```
{  
    /* 循环体语句 */  
}
```

do... while 语句表示先执行循环体语句一次，再判断条件是否成立，如果条件还成立，则再执行循环体语句，直到条件不再成立。其语法如下：

```
do  
{  
    /* 循环体语句 */  
}
```

while(条件);

break 语句结束当前的各种循环，并执行循环的下一条语句。例如：

```
var i=0;  
for(i=0;i<=5;i++){  
    if(i=3){  
        break;                /* 终止 for 循环 */  
    }  
    document.write("The number is" + i + "<br/>");  
}
```

输出结果为：

```
The number is 0  
The number is 1  
The number is 2
```

continue 语句结束当前的循环，并开始下一轮循环。例如：

```
var i=0;  
for(i=0;i<=5;i++){  
    if(i=3)continue;        /*跳过 i=3 的这次循环*/  
    document.write("The number is" + i + "<br/>");  
}
```

输出结果为

```
The number is 0  
The number is 1  
The number is 2  
The number is 4  
The number is 5
```

10.2.5 函数

由事件驱动或当它被调用时执行的可重复使用的代码块称为函数，将脚本编写为函数，这样就可以实现按需调用相应的代码函数。

函数在页面起始位置定义，即网页的<head>部分或定义在外部 JS 文件里面，然后进行



外部调用。

1. 定义函数

```
function 函数名 (var1, var2, var3)
{
    /* 函数代码 */
}
```

var1、var2、var3 等指的是传入函数的变量或值，为函数的参数；大括号定义了函数的开始和结束。

2. 函数返回值

需要返回某个值的函数必须使用 return 语句。例如：

```
function sum(a,b)
{
    x = a + b;
    return x;
}
```

JavaScript 中的函数无须对函数的返回值进行申明，直接使用 return，返回值的类型可以是本节提到的 6 种类型中的任意一种。

3. 函数调用

函数的调用很简单，可以通过其名字加上括号中的参数进行调用。如调用上面的 sum 函数，可以这样写：sum (5, 8)。

【例 10.6】调用 JavaScript 函数。

```
<! --示例 10.6 程序-->
<! DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>函数调用</title>
<script type="text/javascript">
    function sayHello() {
        alert("Hello World!");
    }
</script>
</head>
<body>
    <button onclick="sayHello()">单击这里</button>
</body>
</html>
```

本例浏览效果如图 10-6 所示。



图 10-6 函数调用

10.3 JavaScript 对象

10.3.1 对象基础

1. 概述

JavaScript 编程是使用“面向对象”的概念，也称为“对象化编程”。JavaScript “面向对象编程”的意思就是把 JavaScript 能涉及的范围划分成大大小小的对象，对象下面还继续划分对象，所有的编程都以对象为出发点，基于对象，小到一个变量，大到网页文档、窗口甚至屏幕，都是对象。

在 JavaScript 里面可以创建自己的对象，但在创建用户自定义对象前，需要先学习创建的 JavaScript 对象，以及如何使用它们。本节后续开始将具体地依次讲解 JavaScript 的内建对象。

在 JavaScript 里面创建一个新的空对象，例如：

```
var o = new object();
```

JavaScript 的对象可分为本地对象或内置对象、Browser 对象和 HTML DOM 对象。

(1) 本地对象。本地对象就是 ECMA-262 标准内定义的类，见表 10-5。

表 10-5 本地对象列表

| 对象名称 | 说明 |
|---------|----------------------------------|
| Array | Array 对象用于在单个的变量中存储多个值 |
| Boolean | Boolean 对象表示两个值" true" 或" false" |
| Date | Date 对象用于处理日期和时间 |
| Math | Math 对象用于执行数学任务 |
| Number | Number 对象是原始数值的封装对象 |
| String | String 对象用于处理文本（字符串） |

续表

| 对象名称 | 说明 |
|--------|-----------------------------------|
| RegExp | RegExp 对象表示正则表达式，是对字符串执行模式匹配的强大工具 |
| Global | 全局属性和函数可用于所有内建的 JavaScript 对象 |

全局对象是预定义的对象，作为 JavaScript 的全局函数和全局属性的占位符。通过使用全局对象，可以访问所有其他预定义的对象、函数和属性。全局对象不是任何对象的属性，所以它没有名称。

全局对象只是一个对象，而不是类，它既没有构造函数，也无法实例化一个新的全局对象。

(2) Browser 对象。Browser Objects 也可被称作 BOM，是用来与浏览器窗体网页产生互动的对象。Browser 对象列表见表 10-6。

表 10-6 Browser 对象列表

| 对象名称 | 说明 |
|-----------|-------------------------------|
| window | window 对象表示浏览器中打开的窗口 |
| navigator | navigator 对象包含有关浏览器的信息 |
| screen | screen 对象包含有关客户端显示屏幕的信息 |
| history | history 包含用户（在浏览器窗口中）访问过的 URL |
| location | location 对象包含有关当前 URL 的信息 |

(3) HTML DOM 对象。HTML DOM 定义了用于 HTML 的一系列标准对象，以及访问和处理 HTML 文档的标准方法。其中，最重要的一个对象就是 document 对象，document 代表整个 HTML 文档，用来访问页面中的所有元素。

2. 属性

属性是对象的特性值的表述，例如：

```
<script type="text/javascript">
var str = 'Hello World!';
document.writeln(str.length);
</script>
```

以上示例定义了一个字符串变量，其实字符串也是一个对象，length 作为 str 对象的属性用来取得字符串的长度。

JavaScript 内建对象的属性可以参阅相关的技术文档，当然在 Visual Studio 平台中由于有智能感应功能，就可以很方便地访问对象的属性。

访问用户自定义对象属性的方式和访问 JavaScript 内建对象的属性方式是一样的，例如：

```
var person = new object();
person.birthday = '1982-09-09';
```

属性是通过对一个已有对象的新属性进行赋值的方式来创建的，前面提到声明变量时通常用关键字 var，但是创建对象属性和赋值时不使用 var。

3. 方法

方法指对象可以执行的行为（或可以完成的功能），与属性相同，JavaScript 内建对象本身就自带了很多功能强大的方法，可以直接调用来完成某些特定的功能，例如：

```
var str =Hello World!
document. writeln(str. toUpperCase());
```

运行结果为：

```
HELLO WORLD!
```

以上示例使用了字符串对象的 `toUpperCase()` 方法，其功能是将 `str` 字符串全部转变为大写字母。

10.3.2 常用对象

1. window 对象

`window` 对象表示一个浏览器窗口或一个框架。在客户端 JavaScript 中，`window` 对象是全局对象，所有的表达式都在当前的环境中计算。

(1) 窗口操作。`window` 对象是 JavaScript 中最大的对象，主要用于操作浏览器窗口。通常要引用它的属性和方法时不用 `window.xx` 形式，而直接使用方法名称即可。

通过 `window` 对象移动或调整浏览器窗口的大小，有 4 种方法：

`moveBy(x, y)`：从当前位置水平移动窗体 x 个像素，垂直移动窗体 y 个像素。 x 为负数，将向左移动窗体； y 为负数，将向下移动窗体。

`moveTo(x, y)`：移动窗体左上角到相对于屏幕左上角的 (x, y) 点。当使用负数作为参数时，窗体会移出屏幕的可视区域。

`resizeBy(w, h)`：相对窗体当前的大小，宽度调整 w 个像素，高度调整 h 个像素。如果参数为负值，将缩小窗体，反之扩大窗体。

`resizeTo(w, h)`：把窗体宽度调整为 w 个像素，高度调整为 h 个像素。

【例 10.7】 使用 `window` 对象移动浏览器窗口。

```
<! --示例 10.7 程序-->
<! DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>移动窗口</title>
<script type="text/javascript">
    function moveWin() {
        myWindow. moveTo(50,50);
    }
</script>
</head>
<body>
    <script type="text/javascript">
        myWindow = window. open(" ", " ", `width=200,height=100`);
        myWindow. document. write(" This is myWindow");
```

```
</script>

```

本例浏览效果如图 10-7 和图 10-8 所示。



图 10-7 窗口移动前



图 10-8 窗口移动后

以上示例开始时运行效果如图 10-7 所示，点击“Move myWindow”按钮后，小窗口的左上角移到屏幕左上角的 (50, 50) 点，如图 10-8 所示。

(2) 打开窗口。用 `open()` 打开新窗口。

语法：

```
window.open(url, name, features, replace);
```

说明：

`url`：要载入窗体的 URL。

`name`：新建窗体的名称。

`features`：代表窗体特性的字符串，字符串中每个特性使用逗号分隔。

`replace`：一个布尔值，说明新载入的页面是否替换当前载入的页面，此参数通常不用指定。

(3) 关闭窗口。用 `close()` 关闭窗口。

语法：

```
window.close();
```

例如，关闭一个新建的窗口：

```
var NewWin = window.open(……);  
NewWin.close();
```

(4) 系统对话框。window 对象有三种常用的对话框：

alert ()：显示提示信息，不能对脚本产生任何改变。常见于提醒用户信息不完整、有误等情况。

confirm ()：弹出消息对话框（对话框中包含一个 OK 按钮与一个 Cancel 按钮），一般用于确认信息，返回 true 或 false，所以主要用于 if…else…判断。常见于用户对信息更改前的确认。

prompt ()：一个带输入框的对话框，可以返回用户填入的字符串，常见于某些留言板或论坛输入内容那里的插入 UBB 格式图片。

alert () 方法前面示例讲解过，这里再举例说明 confirm () 方法和 prompt () 方法的使用。

【例 10.8】 使用 confirm () 选择对话框。

```
<head>  
<meta charset="UTF-8">  
<title>confirm()的使用</title>  
</head>  
<body>  
  <script>  
    var userChoice = window.confirm("请选择“确定”或“取消”");  
    if(userChoice == true) {  
      document.write("OK!");  
    }  
    if(userChoice == false) {  
      document.write("Cancel!");  
    }  
  </script>  
</body>  
</html>
```

本例浏览效果如图 10-9、图 10-10、图 10-11 所示。



图 10-9 初始显示



图 10-10 单击“确认”后的运行效果



图 10-11 单击“取消”后的运行效果

以上示例中，如果用户单击“确定”按钮，则网页中显示“OK!”如图 10-10 所示；如果单击“取消”按钮，则网页中显示“Cancel!”如图 10-11 所示。

【例 10.9】 使用 prompt() 提示对话框。

```
<! --示例 10.9 程序-->
<! DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>prompt() 的使用</title>
</head>
<body>
  <script type="text/javascript">
    document.write("您的爱好是---"+prompt('请问您的爱好? ', '请输入'));
  </script>
</body>
</html>
```

本例浏览效果如图 10-12 和图 10-13 所示。



图 10-12 在文本框中输入内容



图 10-13 单击“确认”后的运行效果

说明：`prompt()` 方法需要定义两个参数，而第 2 个参数是可选的，是指文本框输入的默认文本。与 `confirm()` 方法不同的是，`prompt()` 方法只返回 1 个值，当浏览器单击“确定”按钮时，返回文本框中输入的文本，如图 10-13 所示；单击“取消”按钮时，返回值为 `null`。

(5) `history` 历史对象。`window` 对象中的 `history` 历史对象包含了用户已浏览的 URL 的信息，是指浏览器的浏览历史。鉴于安全性的需要，该对象有很多限制。`history` 历史对象有 `length` 这个属性来列出历史的项数。JavaScript 所能管到的历史被限制在用浏览器的“前进”“后退”键链接到的范围。

`history` 历史对象有以下方法：

`back()`：后退，与按下“后退”键是等效的。

`forward()`：前进，与按下“前进”键是等效的。

`go()`：用法为 `history.go(x)`；在历史的范围内指向指定的 URL 访问地址。如果 $x < 0$ ，则后退 x 个地址；如果 $x > 0$ ，则前进 x 个地址；如果 $x = 0$ ，则刷新现在打开的网页。

2. `document` 对象

`document` 本身是一个对象，但又是 JavaScript 中 `window` 对象和 `frames` 对象的一个属性，其描述当前窗口或指定窗口对象的文档。它包含文档从 `<head>` 标签到 `</body>` 标签的全部内容。

用法为 `document` 或 `<窗口对象>.document`，其中 `document` 表示当前窗口的文档，`<窗口对象>.document` 表示指定窗口的文档。

(1) `document` 对象属性。表 10-7 为 `document` 对象比较常用的属性，其他更多的属性请查阅 JavaScript 的相关技术文档。

表 10-7 `document` 对象属性

| 属性 | 描述 |
|----------------------------------|--|
| <code>document.title</code> | 设置文档标题等价于 HTML 的 <code><title></code> 标签 |
| <code>document.bgColor</code> | 设置页面背景色 |
| <code>document.fgColor</code> | 设置前景色（文本颜色） |
| <code>document.linkColor</code> | 未点击过的链接颜色 |
| <code>document.alinkColor</code> | 激活链接（焦点在此链接上）的颜色 |

续表

| 属性 | 描述 |
|---------------------------|-------------------------|
| document.vlinkColor | 已点击过的链接颜色 |
| document.URL | 设置 URL 属性从而在同一窗口打开另一网页 |
| document.fileCreatedDate | 文件建立日期, 只读属性 |
| document.fileModifiedDate | 文件修改日期, 只读属性 |
| document.fileSize | 文件大小, 只读属性 |
| cookie | 设置或返回与当前文档有关的所有 cookie。 |
| document.charset | 设置字符集简体中文为 gb2312 |

【例 10.10】 查看 document 对象属性。

```
<!-- 示例 10.10 程序 -->
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title> Document 属性</title>
</head>
<body>
    <br/>
    <script type="text/javascript">
        document.write(" 文件地址:" + document.location+" <br/>" )
        document.write(" 文件标题:" + document.title+" <br/>" );
        document.write(" 图片路径:" + document.images[0].src+" <br/>" );
    </script>
</body>
</html>
```

本例浏览效果如图 10-14 所示。



图 10-14 Document 属性

(2) document 对象方法。document 对象方法与描述见表 10-8。

表 10-8 document 对象方法

| 方法 | 描述 |
|--------------------------|--|
| open () | 打开已经载入的文档 |
| getElementById () | 返回对拥有指定 id 的第一个对象的引用 |
| getElementsByName () | 返回带有指定名称的对象集合 |
| getElementsByTagName () | 返回带有指定标签名的对象集合 |
| write () | 向文档写 HTML 表达式 或 JavaScript 代码 |
| writeln () | 等同于 write () 方法，不同的是在每个表达式之后写一个换行符 |
| close () | 关闭 document.open () 方法打开的输出流，并显示选定的数据 |
| clear () | 清空当前文档 |

document 对象的方法在 JavaScript 程序中使用的频率非常高。

【例 10.11】 应用 document 对象的方法。

```

<! --示例 10.11 程序-->
<! DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title> Document 方法</title>
</head>
<body>
  <br/>
  <div id="book">网页设计基础</div>
  <br/>
  <script type="text/javascript">
    var book = document.getElementById("book");
    book.onclick = msg;
    function msg() {
      alert(book.innerText);
    }
  </script>
</body>
</html>

```

本例浏览效果如图 10-15 所示。



图 10-15 Document 方法

以上示例中通过调用 `getElementById()` 方法可以得到 `<div>` 标签的内容，并通过消息窗口显示。

3. location 对象

location 对象用于获取或设置窗体的 URL，并且可以用于解析 URL，是 JavaScript 中最重要的对象之一，location 的中文意思是“位置”。location 既是 window 对象的属性又是 document 对象的属性，即“`window.location == document.location;`”。

(1) location 对象属性。location 对象包含 8 个属性，最重要的一个是 href 属性，代表当前窗体的 URL，其余 7 个都是当前窗体的 URL 的一部分。其具体描述见表 10-9。

表 10-9 location 对象属性

| 属性 | 描述 |
|----------|-----------------------------|
| hash | 设置或返回从井号 (#) 开始的 URL |
| host | 设置或返回主机名和当前 URL 的端口号 |
| hostname | 设置或返回当前 URL 的主机名 |
| href | 设置或返回完整的 URL |
| pathname | 设置或返回当前 URL 的路径部分 |
| port | 设置或返回当前 URL 的端口号 |
| protocol | 设置或返回当前 URL 的协议 |
| search | 设置或返回从问号 (?) 开始的 URL (查询部分) |

location 对象的 8 个属性都是可读写的，但通常只对 href 与 hash 属性进行写操作。图 10-16 是对 location 对象中属性的图解。

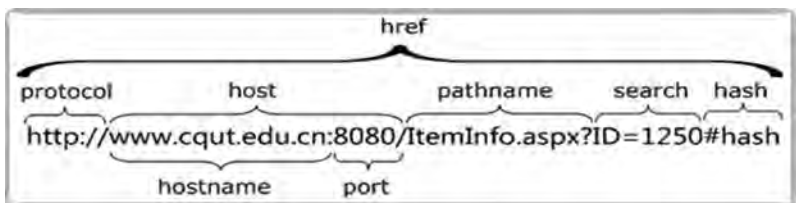


图 10-16 location 对象的属性图解

【例 10.12】 location 对象属性。

```
<!--示例 10.12 程序-->
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>location 对象属性</title>
<script type="text/javascript" >
    function currLocation() { alert(window.location) }
    function newLocation() { window.location=" 示例程序 10.11. html" }
</script>
<body>
    <input type="button" onclick="currLocation()" value=" 显示当前的 URL"/>
    <input type="button" onclick="newLocation()" value=" 改变 URL"/>
</body>
</html>
```

本例浏览效果如图 10-17 至图 10-19 所示。



图 10-17 location 对象属性



图 10-18 点击“显示当前的 URL”后的效果

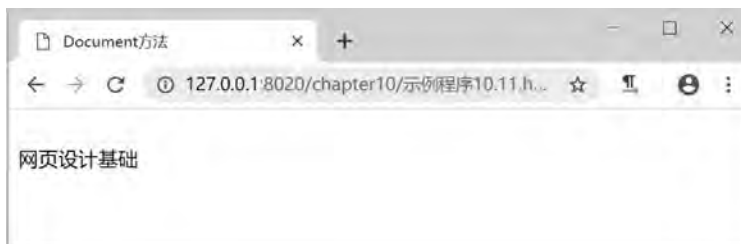


图 10-19 点击“改变 URL”后的效果

(2) location 对象方法。location 对象方法见表 10-10。

表 10-10 location 对象方法

| 方法 | 描述 |
|-------------|--|
| assign () | 设置当前文档的 URL。 |
| replace () | 设置当前文档的 URL，并在 history 对象的地址列表中删除这个 URL。 |
| reload () | 重新载入当前文档（从 server 服务器端）。 |
| toString () | 返回 location 对象 href 属性当前的值。 |

4. navigator 对象

navigator 对象的中文意思为“导航器”，通常用于检测浏览器与操作系统的版本，也是 window 对象的属性。

由于 navigator 没有统一的标准，因此各个浏览器都有自己不同的 navigator 版本，navigator 的属性和方法也会有一定的不同。这里介绍普遍支持且常用的部分属性和方法。

(1) navigator 对象属性。navigator 中最重要的是 userAgent 属性，它用来返回包含浏览器版本等信息的字符串；其次 cookieEnabled 也很重要，使用它可以判断用户浏览器是否开启 cookie。navigator 对象属性及其描述见表 10-11。

表 10-11 navigator 对象属性

| 属性 | 描述 |
|---------------|--------------------------------|
| appName | 浏览器代码名的字符串表示 |
| appVersion | 浏览器版本信息的字符串表示 |
| cookieEnabled | 如果启用 cookie 返回 true，否则返回 false |
| platform | 浏览器所在计算机平台的字符串表示 |
| plugins | 安装在浏览器中的插件数组 |
| taintEnabled | 如果启用了数据污点，返回 true，否则返回 false |
| userAgent | 用户代理头的字符串表示 |

(2) navigator 对象方法。navigator 对象方法见表 10-12。

表 10-12 location 对象方法

| 方法 | 描述 |
|-----------------|-------------------------------|
| javaEnabled () | 规定浏览器是否启用了 java |
| taintEnabled () | 规定浏览器是否启用数据污点 (data tainting) |

5. screen 对象

screen 对象用于获取用户的屏幕信息，是 window 对象的属性。

每个 window 对象的 screen 属性都引用一个 screen 对象。可以通过 screen 对象获取一些用户屏幕的信息，如宽度、高度、像素等，而这些信息对于设置图片和页面在浏览器中显示的大小都是非常有用的。screen 对象的属性应用和浏览器的关系比较密切，所以在应用这些属性时，一定要注意浏览器之间的兼容性。screen 对象属性的具体描述见表 10-13。

表 10-13 screen 对象属性

| 属性 | 描述 |
|----------------------|----------------------------|
| availHeight | 返回显示屏幕的高度（除 Windows 任务栏之外） |
| availWidth | 返回显示屏幕的宽度（除 Windows 任务栏之外） |
| bufferDepth | 设置或返回调色板的比特深度 |
| colorDepth | 返回目标设备或缓冲器上的调色板的比特深度 |
| deviceXDPI | 返回显示屏幕的每英寸水平点数 |
| deviceYDPI | 返回显示屏幕的每英寸垂直点数 |
| fontSmoothingEnabled | 返回用户是否在显示控制面板中启用了字体平滑 |
| height | 返回显示屏幕的高度 |
| logicalXDPI | 返回显示屏幕每英寸的水平方向的常规点数 |
| logicalYDPI | 返回显示屏幕每英寸的垂直方向的常规点数 |
| pixelDepth | 返回显示屏幕的颜色分辨率（比特每像素） |
| updateInterval | 设置或返回屏幕的刷新率 |
| width | 返回显示器屏幕的宽度 |

availWidth 与 availHeight 属性比较常用，它们可以使窗口填满用户显示器的屏幕。

10.4 JavaScript 事件

10.4.1 事件及事件处理

JavaScript 中的事件是可以被 JavaScript 侦测到的行为。网页中的每个元素都可以产生某些可以触发 JavaScript 函数的事件。例如，在用户单击页面中某按钮时产生一个 onClick 事件。事件可以是用户在某些内容上的单击、鼠标经过特定元素或按下了键盘上的按键；也可以是某个 Web 页面加载完成、用户改变了窗口的大小或是滚动了窗口等。

事件处理是对象化编程的一个重要环节，JavaScript 会为特定文档元素的特定类型事件注册一个事件处理程序，该处理程序可以是个函数，也可以是一段代码。事件处理的过程可以这样表示：发生事件→启动事件处理程序→事件处理程序做出反应。其中，要使事件处理程序能够启动，必须先告诉对象发生了什么事情，要启动什么处理程序。事件的处理程序可以是任意 JavaScript 语句，但是一般用特定的自定义函数来处理。

指定事件处理程序有三种方法：



(1) 直接在 HTML 标记中指定。这种方法用得最多。

一般形式:

```
<标记...事件=" 事件处理程序" [事件=" 事件处理程序" ...] >
```

例如:

```
<html>
<head>
<meta charset="utf-8" />
<title></title>
</head>
<body onload="alert('欢迎访问页面! ')">
</body>
</html>
```

以上示例<body>标记中,在文档加载完毕时弹出一个告警框,显示“欢迎访问页面”。

(2) 编写特定对象特定事件的 JavaScript。这种方法用得少,但是在某些场合还是很好用的。

一般形式:

```
<script language="JavaScript" for="对象" event="事件">
```

(事件处理程序代码)

```
</script>
```

例如:

```
<html>
<head>
<meta charset="utf-8" />
<title></title>
<script language="JavaScript" for="window" event="onload">
    alert('欢迎访问页面! ')
</script>
</head>
<body>
</body>
</html>
```

(3) 在 JavaScript 中说明。

一般形式:

```
<事件主角 - 对象>. <事件> = <事件处理程序>;
```

该方法需要注意的是,“事件处理程序”是真正的代码,而不是字符串形式的代码。如果事件处理程序是一个自定义函数,如无使用参数的需要,就不要加“()”。

```
<html>
<head>
<script type="text/javascript">
    window.onerror = ignoreError;
    function ignoreError() {
        return true;
    }
</script>
```

```

</script>
</head>
<body>
</body>
</html>

```

这个例子将 `ignoreError()` 函数定义为 `window` 对象的 `onerror` 事件的处理程序。它的功能是忽略该 `window` 对象下任何错误。

10.4.2 常用事件

JavaScript 中常用事件主要包括一般页面事件、鼠标事件、键盘事件等。表 10-14、表 10-15、表 10-16 为这三类事件中的常用事件描述。

表 10-14 页面事件描述

| 描述 | 方法 |
|----------|------------------------------|
| onBlur | 当前元素失去焦点时触发此事件 |
| onChange | 当前元素失去焦点并且元素的内容发生变化时触发此事件 |
| onError | 当前页面出现错误时触发此事件 |
| onFocus | 当某个元素获得焦点时触发此事件 |
| onLoad | 当页面内容加载完成时触发此事件 |
| onMove | 当浏览器窗口被移动时触发此事件 |
| onResize | 当浏览器窗口大小被改变时触发此事件 |
| onReset | 当表单中 RESET 属性被激发时触发此事件 |
| onScroll | 当浏览器滚动条位置发生变化时触发此事件 |
| onStop | 当浏览器的停止按钮被按下或正在下载的文件被中断时触发此事 |
| onSubmit | 当表单被提交时触发此事件 |
| onUnload | 当前页面将被改动时触发此事件 |

表 10-15 鼠标事件描述

| 方法 | 描述 |
|-------------|----------------------|
| onClick | 当鼠标单击时触发此事件 |
| onDbClick | 当鼠标双击时触发此事件 |
| onMouseDown | 当按下鼠标按钮时触发此事件 |
| onMouseUp | 当按下鼠标按钮后松开时触发此事件 |
| onMouseOver | 当鼠标移动到某对象范围的上方时触发此事件 |
| onMouseMove | 当鼠标移动时触发此事件 |
| onMouseOut | 当鼠标离开某对象范围时触发此事件 |

表 10-16 键盘事件描述

| 方法 | 描述 |
|------------|---------------------|
| onKeyPress | 当键盘上某个键被按下并释放时触发此事件 |
| onKeyDown | 当键盘上某个键被按下时触发此事件 |
| onKeyUp | 当键盘上某个键被释放时触发此事件 |

对于事件的类型，随着 HTML5 的出现和发展，又新增了 HTML5 事件、设备事件、触摸事件、手势事件等各种事件，感兴趣的读者可查阅相关资料，本书中不再一一列举。

10.4.3 事件应用举例

事件的应用常见于下面几种场景：鼠标单击某一元素、页面或图像载入、鼠标悬浮于页面的某个热点之上、在表单中选取输入框、确认表单、键盘按键等。下面通过几个示例来学习事件的具体应用。

【例 10.13】 onClick 事件。

```
<! --示例 10.13 程序-->
<! DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>onClick 事件</title>
<script type="text/javascript">
    function changetext(id) {
        id.innerHTML="谢谢";
    }
</script>
</head>
<body>
    <h1 onclick="changetext(this)">请点击我</h1>
</body>
</html>
```

本例浏览效果如图 10-20 和图 10-21 所示。



图 10-20 onClick 事件 (1)



图 10-21 onClick 事件 (2)

以上示例中，当用户在<h1>元素上单击时，会改变其内容如图 10-21 所示。

【例 10.14】 onLoad 事件。

```
<! --示例 10.14 程序-->
<! DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>onLoad 事件</title>
</head>
<body onload="checkCookies()">
  <script type="text/javascript">
    function checkCookies() {
      if(navigator.cookieEnabled == true)
        alert("已启用 Cookie");
      else
        alert("未启用 Cookie");
    }
  </script>
  <p>提示框会告诉你，浏览器是否已启用 Cookie。</p>
</body>
</html>
```

本例浏览效果如图 10-22 所示。



图 10-22 onLoad 事件

以上示例中，当用户打开页面时会提示客户端浏览器是否启用了 Cookie。

【例 10.15】 onchange 事件。

<! --示例 10.15 程序-->

<! DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>onchange 事件</title>

<script type="text/javascript">

function myFunction() {

var x = document.getElementById("fname");

x.value=x.value.toUpperCase();

}

</script>

</head>

<body>

请输入英文字符:<input type="text" id="fname" onchange="myFunction()" />

<p>当您离开输入字段时,会触发将输入文本转换为大写的函数.</p>

</body>

</html>

本例浏览效果如图 10-23 和图 10-24 所示。



图 10-23 onchange 事件 (1)



图 10-24 onchange 事件 (2)

以上示例中，当用户在输入框里输入英文字符后，离开输入框时，会使输入字符转换为

大写，如果 10-24 所示。

【例 10.16】 鼠标事件（onmouseover、onmouseout、onmousedown、onmouseup）。

```
<! --示例 10.16 程序-->
<! DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title></title>
<script type="text/javascript">
    function mouseOver(obj) {
        obj.style.color="blue";
        obj.innerHTML="把鼠标移开"
    }
    function mouseOut(obj) {
        obj.style.color="red";
        obj.innerHTML="把鼠标移到上面"
    }
    function mouseDown(obj) {
        obj.style.backgroundColor="blue";
        obj.innerHTML="请释放鼠标按钮";
    }
    function mouseUp(obj) {
        obj.style.backgroundColor="red";
        obj.innerHTML="请按下鼠标按钮";
    }
</script>
</head>
<body>
    <div onmouseover="mouseOver(this)" onmouseout="mouseOut(this)"
    style="background-color:green;width:120px;height:20px;padding:20px;color:
#ffffff;">
        把鼠标移到上面
    </div>
    <br />
    <div onmousedown="mouseDown(this)" onmouseup="mouseUp(this)"
    style="background-color:green;width:120px;height:20px;padding:20px;color:
#ffffff;">
        请按下鼠标按钮
    </div>
</body>
</html>
```

本例浏览效果如图 10-25、图 10-26、图 10-27 所示。



图 10-25 鼠标事件前效果图



图 10-26 鼠标经过



图 10-27 按下鼠标左键

以上示例中，第一个层中显示文本“把鼠标移到上面”，第二个层中显示文本“请按下鼠标按钮”，如图 10-25 所示；当用户移动鼠标经过第一个层上时，文本内容改变为“把鼠标移开”，且文本颜色改变为蓝色，如图 10-26 所示；当用户鼠标离开第一个层时，文本内容改变为“把鼠标移到上面”，且文本颜色还原为白色；当用户在第二个层面上按下鼠标左键时，文本内容改变为“请释放鼠标按钮”，同时背景颜色改变为蓝色，如图 10-27 所示；当用户释放鼠标左键时，文本内容改变为“请按下鼠标按钮”，同时背景颜色还原为绿色。

【例 10.17】 onkeypress 事件、onkeydown 事件。

```
<!-- 示例 10.17 程序 -->
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>键盘事件</title>
<script type="text/javascript">
    function keyDown() {
        alert("你按下了按键");
        if(event.ctrlKey) {
            alert("你按下了 Ctrl 键");
        }
    }
}
```

```
function keyPress() {  
    alert("你按下了键,并且释放了按键");  
}  
</script>  
</head>  
<body onkeydown="keyDown()" onkeypress="keyPress()">  
</body>  
</html>
```

本例浏览效果如图 10-28 和图 10-29 所示。



图 10-28 按下键盘任意键



图 10-29 释放按键

以上示例中,当用户按下任意一个按键,浏览器会弹出信息框“你按下了按键”,如果按下键为 Ctrl 键,还会弹出信息框“你按下了 Ctrl 键”;当用户释放按下的按键后,又会弹出信息框“你按下了键,并且释放了按键”,如图 10-29 所示。

10.5 综合案例——轮播广告

在网站的首页经常看到轮流播放的图片广告,既美化了页面又节省了版面,本例要求制作一个图片轮播页面,效果如图 10-30 所示。

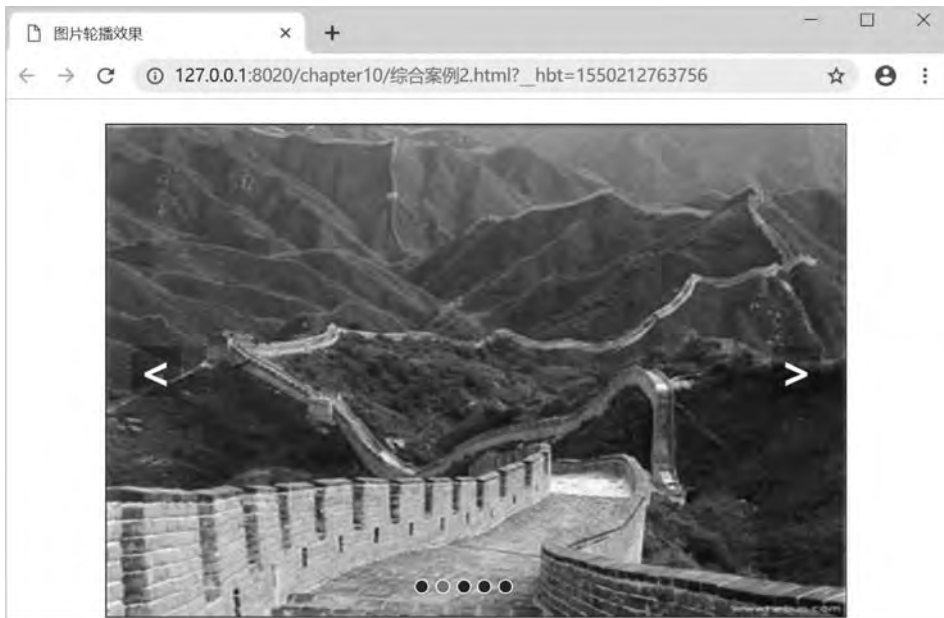


图 10-30 图片轮播效果

图片轮播原理：将一系列图片平铺，利用 CSS 布局只显示其中一张图片，其余隐藏。通过计算偏移量利用定时器实现自动播放，或通过手动点击事件切换图片。

HTML 布局：页面布局如图 10-31 所示。



图 10-31 页面布局

以下为完整代码，以供参考。

```

<! --综合案例 1. html-->
<! DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>图片轮播效果</title>
<style type="text/css">
* {
margin:0;
padding:0;
text-decoration:none;

```

```
}  
body {  
    padding:20px;  
}  
#container {  
    position: relative;  
    width: 600px;  
    height: 400px;  
    border: 1px solid #333;  
    overflow: hidden;  
    margin-left: auto;  
    margin-right: auto;  
}  
#list {  
    position: absolute;  
    z-index: 1;  
    width: 420px;  
    height: 400px;  
}  
#list img {  
    float: left;  
    width: 600px;  
    height: 400px;  
}  
#buttons {  
    position: absolute;  
    left: 250px;  
    bottom: 20px;  
    z-index: 2;  
    height: 10px;  
    width: 100px;  
}  
#buttons span {  
    float: left;  
    margin-right: 5px;  
    width: 10px;  
    height: 10px;  
    border: 1px solid #fff;  
    border-radius: 50%;  
    background: #333;  
    cursor: pointer;  
}  
#buttons. on {
```

```
        background:orangered;
    }
    .arrow{
        position:absolute;
        top:180px;
        z-index:2;
        display:none;
        width:40px;
        height:40px;
        font-size:36px;
        font-weight:bold;
        line-height:39px;
        text-align:center;
        color:#fff;
        background-color:RGBA(0,0,0,.3);
        cursor:pointer;
    }
    .arrow:hover{
        background-color:RGBA(0,0,0,.7);
    }
    #container:hover .arrow{
        display:block;
    }
    #prev{
        left:20px;
    }
    #next{
        right:20px;
    }
</style>
<script type="text/javascript">
    window.onload = function() {
        var container = document.getElementById( 'container' );
        var list = document.getElementById( 'list' );
        var buttons = document.getElementById( 'buttons' ).getElementsByTagName( 'span' );
        var prev = document.getElementById( 'prev' );
        var next = document.getElementById( 'next' );
        var index = 1;
        var timer;
        function animate( offset ) {
            var newLeft = parseInt( list.style.left ) + offset;
            list.style.left = newLeft + 'px';
            //无限滚动判断
```

```
        if( newLeft > -600) {
            list. style. left = -3000 + 'px';
        }
        if( newLeft < -3000) {
            list. style. left = -600 + 'px';
        }
    }
    function play() {
        //重复执行的定时器
        timer = setInterval(function() {
            next. onclick();
        },2000)
    }
    function stop() {
        clearInterval(timer);
    }
    function buttonsShow() {
        //将之前的小圆点的样式清除
        for( var i = 0; i < buttons. length; i++) {
            if( buttons[ i]. className == "on" ) {
                buttons[ i]. className = "";
            }
        }
        //数组从 0 开始, index 需要减 1
        buttons[ index - 1]. className = "on";
    }
    prev. onclick = function() {
        index -= 1;
        if( index < 1) {
            index = 5
        }
        buttonsShow();
        animate(600);
    };
    next. onclick = function() {
        index += 1;
        if( index > 5) {
            index = 1
        }
        animate(-600);
        buttonsShow();
    };
    for( var i = 0; i < buttons. length; i++) {
```

```
(function(i) {
    buttons[i]. onclick = function() {
        var clickIndex = parseInt(this.getAttribute('index'));
        var offset = 600 * (index - clickIndex);
        animate(offset);
        index = clickIndex;
        buttonsShow();
    }
})(i)
}
container.onmouseover = stop;
container.onmouseout = play;
play();
}
</script>
</head>
<body>
<div id="container">
    <div id="list" style="left:-600px;">
        
        
        
        
        
        
        
    </div>
    <div id="buttons">
        <span index="1" class="on"></span>
        <span index="2"></span>
        <span index="3"></span>
        <span index="4"></span>
        <span index="5"></span>
    </div>
    <a href="javascript:;" id="prev" class="arrow">&lt;</a>
    <a href="javascript:;" id="next" class="arrow">&gt;&gt;/a>
</div>
</body>
</html>
```

本章习题

1. 在网页中显示一个工作中的“数字时钟”，如图 10-32 所示。



图 10-32 数字时钟

2. 制作一个“禁止使用鼠标右键”操作的网页。当浏览者在网页上右击时，自动弹出一个警告对话框，禁止用户使用右键快捷菜单，实例效果如图 10-33 所示。



图 10-33 禁止使用右键

第 11 章 综合案例——“精品课程网站”开发

本章主要运用前面章节讲解的各种网页制作技术介绍网站的开发流程，从而进一步巩固网页设计与制作的基本知识。

11.1 网站的开发流程

建立一个网站就像盖一幢大楼，它是一个系统工程，有着自己特定的工作流程，只有遵循以下几个流程，才有可能设计出一个令人满意的网站。

网站的开发流程一般包括以下阶段：

- (1) 确定网站主题；
- (2) 搜集材料；
- (3) 规划网站；
- (4) 网站制作；
- (5) 测试、发布网站；
- (6) 维护更新。

11.1.1 确定网站主题

网站主题就是网站所要包含的主要内容，一个网站必须要有一个鲜明的主题。特别是对于个人网站，能力、精力、财力都有限，因此就不能选择制作像“网易”“搜狐”“新浪”等那样包罗万象、内容大而全的综合网站。应该找准一个自己最感兴趣的方向，做深、做透，办出自己的特色，这样才能给用户留下深刻的印象，才能有存在的价值。在确定网站主题时应该注意以下三个问题。

1. 确定建站的目的

建立网站的目的要么是增加利润，要么是传播信息或观点。显然，精品课程网站的目的是第二种：传播信息。

随着计算机技术、网络技术和远程教育事业的高速发展，在现代教学过程中，知识的传授方式也随之变革。学院教师在教学过程中积累了丰富的教学资源，为了更好地实现课程资源的共享，提高教师的教学效率和学生的自主学习能力，特进行“精品课程网站”的开发和建设。

2. 确定目标用户

不同身份、年龄、爱好的浏览者，对网站的要求是不同的。所以最初的规划阶段，确定

目标用户是一个至关重要的步骤。精品课程网站的用户主体是学生和教师。

3. 确定网站的内容

内容决定一切，内容价值决定了浏览者是否有兴趣继续关注网站。“精品课程”网站一般包含以下页面：

- (1) 首页；
- (2) 课程介绍；
- (3) 教学团队；
- (4) 教学资源；
- (5) 教学视频；
- (6) 作品展示；
- (7) 在线答疑。

11.1.2 搜集材料

明确网站的主题以后，就要围绕主题搜集相关材料了，包括图片、文字、声音、影像等材料，既可以从图书、报纸、光盘上得来，也可以从互联网上搜集，然后把搜集的材料去粗取精、去伪存真，作为自己制作网页的素材。

11.1.3 规划网站

网站设计得成功与否，很大程度上取决于设计者规划水平的好坏，规划网站就像设计师设计大楼一样，图纸设计好了，才能建成一座漂亮的楼房。根据网站的内容，规划各页面的结构、布局、风格和色彩搭配等，从而做出详细设计图。

11.1.4 网站制作

依照规划，利用网站制作软件一步步地把想法变成现实，这是一个复杂而细致的过程，需要按照先大后小、先简单后复杂来进行网页的制作。

所谓先大后小，就是在制作网页时，先把大的结构设计好，然后再逐步完善小的结构设计。

先简单后复杂，就是先设计出简单的内容，然后再设计复杂的内容，以便出现问题时好修改。

11.1.5 测试、发布网站

在本机将网站制作完毕后，还要测试网站在不同用户端的显示是否合适。实际上，在网站建设过程中，最好经常对网站进行测试并解决出现的问题，这样可以尽早发现问题并避免重犯错误。测试网站主要从以下三方面着手：

- (1) 页面的效果是否美观；
- (2) 页面中的链接是否正确；
- (3) 页面的浏览器兼容性是否良好。

当完成了网站的设计、制作、调试和测试等工作后，需要把设计好的网站上传到服务器

来完成整个网站的发布。可以使用网站发布工具将文件上传到远程 Web 服务器以发布该网站，这样才能够让全世界的人上网观看。

11.1.6 维护更新

网站要注意经常维护更新内容，保持内容的新鲜，只有不断地给它补充新的内容，才能够吸引住浏览者。同时，要对发现的错误和问题进行及时修正。

11.2 设计首页布局

熟悉了网站的开发流程后，就可以开始制作首页了。制作首页前，用户还需要创建站点目录，搭建整个网站的大致结构。

11.2.1 创建站点目录

在制作各个页面前，用户需要确定整个网站的目录结构，包括创建站点根目录和根目录下的通用目录。

1. 创建站点根目录

本章讲解的综合案例建立在 D:\HBulier Web\chapter11 目录中，该目录作为站点根目录。

2. 根目录下的通用目录

对于中小型网站，一般会创建如下通用的目录结构。

- (1) img 目录：存放网站的所有图片。
- (2) css 目录：存放 css 样式文件，实现内容和样式的分离。
- (3) js 目录：存放 JavaScript 脚本文件。
- (4) media 目录：存放音频和视频文件。

在 D:\HBulier Web\chapter11 目录中依次建立上述目录，整个网站的目录结构如图 11-1 所示。



图 11-1 网站目录结构

11.2.2 首页布局规划

网站首页布局为“国”字型，主要内容包括学校的 Logo、banner 广告条、导航菜单、获奖作品展示、其他各子页主内容、用户登录窗口、友情链接，下面滚动图片及版权信息，布局示意图如图 11-2 所示，效果图如图 11-3 所示。

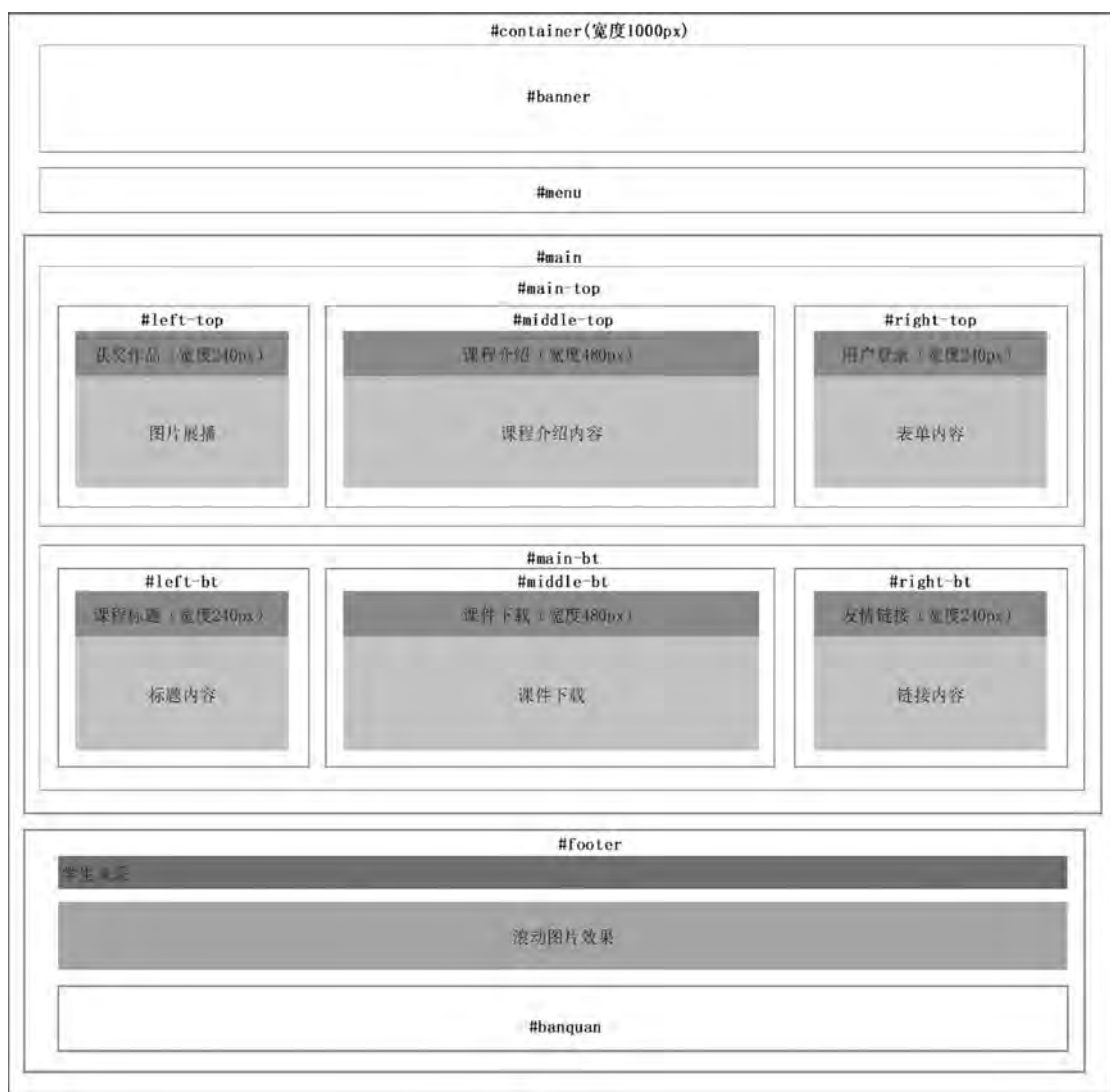


图 11-2 首页布局



图 11-3 首页效果图

11.3 首页的制作

在实现了首页的整体布局后，接下来就要完成首页的制作。制作过程如下。

1. 页面容器层的制作

页面全部内容都是放在一个 container 层中，其 CSS 代码和全局样式如下：

```
* {
padding:0;
margin:0;
}
```

```
#container {
    width: 1000px;
    margin-left: auto;
    margin-right: auto;
}
```

2. banner 广告条的制作

利用 Photoshop 软件制作 banner 广告条，将 logo、背景图片和实物等合成 banner 图片，由于操作比较简单，这里不做操作过程介绍，完成效果如图 11-4 所示。



图 11-4 banner.jpg

由于制作 Flash 比较难，而我们又希望首页有 Flash 效果，这里可以在网上下载一个透明 Flash，先在相关 banner 层添加一个背景图片，也就是上面用 Photoshop 制作出来的图片，层的尺寸与图片尺寸相同，设置图片不要重复；然后再在 banner 层添加 Flash，设置 Flash 的尺寸与层的尺寸相同，并设置 wmode 属性值为 transparent。

相关 CSS 代码如下：

```
#banner {
    width: 1000px;
    height: 240px;
    background: url(../img/banner1.jpg) no-repeat;
}
```

body 中的相关代码如下：

```
<div id="banner">
    <embed src="media/1.swf" width="1000" height="240" autostart="true" loop="true" wmode="transparent"></embed>
</div>
```

完成效果如图 11-5 所示。



图 11-5 banner 层效果

3. 制作二级下拉菜单

导航菜单是网站重要的组成部分。导航菜单的设计关系着网站的可用性和用户体验，有吸引力的导航能够吸引用户去浏览更多的网站内容。设计一个优秀的页面导航菜单会给网站增色不少。

网站菜单表现形式丰富多样。从菜单层次上看，可以分为一级、二级和多级菜单。从排列方式上看，可分为水平导航、垂直导航菜单。从技术实现角度上看，导航菜单通常采用无序列表、表格、超链接和样式表相结合的方法来实现，也可以使用如 JavaScript、CSS3 Menu、jQuery 等第三方插件技术来实现。不过要考虑到不同浏览器之间的兼容性，这里使用无序列表、超链接和 CSS 样式定义来制作简单的二级下拉菜单，完成效果如图 11-6 所示。



图 11-6 二级下拉菜单

CSS 代码如下：

```
#menu {
    width:1000px;
    height:30px;
    margin:8px auto;
    background-color:#1b99ee;
}
#menu ul {
    margin:0px;
    padding:0px;
}
#menu ul li {
    height:30px;
    width:120px;
    list-style:none;
    float:left;
    display:inline;
    font-family:"微软雅黑";
    font-size:14px;
    font-weight:bold;
    text-align:center;
}
#menu ul li a {
    color:#FFF;
```



```
width:120px;
margin:0px;
padding:0px 0px 0px 8px;
text-decoration:none;
display:block;
background:#1b99ee;
line-height:30px;
}
#menu ul li ul li {
height:28px;
text-decoration:none;
}
#menu ul li ul li a {
background:#1b99ee;
line-height:27px;
text-decoration:none;
border:0px solid #FFFFFF;
}
#menu ul li a:hover {
background:#0051A2;
}
#menu ul li ul {
position:relative;
visibility:hidden;/* 也可以用 display:none; */
z-index:999;
}
#menu ul li:hover ul {
visibility:visible;/* 也可以用 display:block; */
}
#men ul li ul li a:hover {
background:#0051A2;
}
}
```

HTML 代码如下:

```
<! DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>二级下拉菜单</title>
<link rel="stylesheet" type="text/css" href="menu.css"/>
</head>
<body>
<div id="menu">
<ul>
```

```

<li><a href="index.html">首页</a></li>
<li><a href="kechengjieshao.html">课程介绍</a></li>
<li><a href="jiaoxuetuandui.html">教学团队</a></li>
<li><a href="jiaoxueziyuan.html">教学资源</a>
<ul>
<li><a href="jiaoxuedagang.html">教学大纲</a></li>
<li><a href="jiaolan.html">教案</a></li>
<li><a href="jiaoxuekejian.html">教学课件</a></li>
<li><a href="kehouliaoxi.html">课后练习</a></li>
</ul>
</li>
<li><a href="jiaoxueshiping.html">教学视频</a></li>
<li><a href="zuopinzhanshi.html">作品展示</a></li>
<li><a href="zaixiandayi.html">在线答疑</a></li>
</ul>
</div>
</body>
</html>

```

4. 制作“获奖作品”轮播效果

这里可以参考第 10 章的综合案例——“图片轮播效果”，只需要修改图片尺寸和偏移尺寸就行了。完成效果如图 11-7 所示。



图 11-7 “获奖作品”效果图

相关 CSS 代码如下：

```

#left-top{
    width:240px;
    float:left;
    margin-right:10px;
}
.title1{
    color:#FFF;

```

```
font-size: 12px;
font-weight: bold;
background-color: #1b99ee;
padding: 4px 10px;
}
.nr1 {
padding-top: 5px;
position: relative;
width: 240px;
height: 180px;
overflow: hidden;
margin-left: auto;
margin-right: auto;
background-color: #F0F0F0;
}
#list {
position: absolute;
z-index: 1;
width: 150px;
height: 176px;
}
#list img {
float: left;
width: 240px;
height: 176px;
}
#buttons {
position: absolute;
left: 70px;
bottom: 10px;
z-index: 2;
height: 10px;
width: 100px;
}
#buttons span {
float: left;
margin-right: 5px;
width: 10px;
height: 10px;
border: 1px solid #fff;
border-radius: 50%;
background: #333;
cursor: pointer;
```

```
}
#buttons.on {
    background:orangered;
}
.arrow {
    position: absolute;
    top: 65px;
    z-index: 2;
    display: none;
    width: 10px;
    height: 10px;
    font-size: 20px;
    font-weight: bold;
    line-height: 20px;
    text-align: center;
    color: #fff;
    background-color: RGBA(0,0,0,.3);
    cursor: pointer;
}
.arrow: hover {
    background-color: RGBA(0,0,0,.7);
}
#nr1: hover .arrow {
    display: block;
}
#prev {
    left: 20px;
}
#next {
    right: 20px;
}
```

head 中的相关代码如下:

```
<script type="text/javascript">
    window.onload = function() {
        var container = document.getElementById( 'container' );
        var list = document.getElementById( 'list' );
        var buttons = document.getElementById( 'buttons' ). getElementsByTagName( 'span' );
        var prev = document.getElementById( 'prev' );
        var next = document.getElementById( 'next' );
        var index = 1;
        var timer;
        function animate( offset ) {
            var newLeft = parseInt( list. style. left ) + offset;
```

```
list.style.left = newLeft + 'px';
    //无限滚动判断
if( newLeft > -240) {
    list.style.left = -1200 + 'px';
}
if( newLeft < -1200) {
    list.style.left = -240 + 'px';
}
}
function play() {
    //重复执行的定时器
    timer = setInterval(function() {
        next.onclick();
    },2000)
}
function stop() {
    clearInterval(timer);
}
function buttonsShow() {
    //将之前的小圆点的样式清除
    for( var i = 0; i < buttons.length; i++) {
        if( buttons[i].className == "on" ) {
            buttons[i].className = "";
        }
    }
    //数组从 0 开始,故 index 需要-1
    buttons[ index - 1 ].className = "on";
}
prev.onclick = function() {
    index -= 1;
    if( index < 1) {
        index = 5
    }
    buttonsShow();
    animate(240);
};
next.onclick = function() {
    index += 1;
    if( index > 5) {
        index = 1
    }
    animate(-240);
    buttonsShow();
}
```

```

    };
    for(var i = 0;i < buttons.length;i++) {
        (function(i) {
            buttons[i]. onclick = function() {
                var clickIndex = parseInt(this.getAttribute( 'index' ));
                var offset = 240 * (index - clickIndex);
                animate(offset);
                index = clickIndex;
                buttonsShow();
            }
        })(i)
    }
    container.onmouseover = stop;
    container.onmouseout = play;
    play();
}
</script>

```

body 中的相关代码如下:

```

<div id="left-top" >
    <div class="title1" >获奖作品</div>
    <div class="nr1" >
        <div id="list" style="left:-240px;" >
            
            
            
            
            
            
            
        </div>
        <div id="buttons" >
            <span index="1" class="on" ></span>
            <span index="2" ></span>
            <span index="3" ></span>
            <span index="4" ></span>
            <span index="5" ></span>
        </div>
        <a href="javascript:;" id="prev" class="arrow" >&lt;</a>
        <a href="javascript:;" id="next" class="arrow" >&gt;></a>
    </div>
</div>

```

5. 中间的“课程介绍”“用户登录”“课程标题”和“课件下载”的制作

我们可以运用前面的 DIV+CSS 知识制作“课程介绍”“用户登录”“课程标题”和“课

件下载”等内容，由于这几部分的操作相对简单，这里不再做详细介绍。

6. 制作“友情链接”上下滚动效果

“友情链接”的相关内容是一个无序列表，列表样式是用自定义图标表示，然后使用 `margin` 标签设置列表上下循环滚动。另外，在 `margin` 标签内添加 `onmouseover` 和 `onmouseout` 事件调用当前对象的 `stop()` 和 `start()` 方法。完成效果如图 11-8 所示。



图 11-8 “友情链接”效果图

相关 CSS 代码如下：

```
#right-bt {
    width: 240px;
    float: left;
}

.title_nav {
    width: 240px;
    height: 200px;
    padding-top: 14px;
    background-color: #FFFFFF;
}

.title_nav ul li {
    list-style: none;
    list-style-position: inside;
    list-style-image: url(../img/pic_1.gif);
    margin-left: 20px;
    line-height: 30px;
    border-bottom: 1px dashed #000000;
}

.title_nav ul li a, .title_nav ul li a:link, .title_nav ul li a:visited {
    text-decoration: none;
```

```
color:#000000;
```

```
}
```

body 中的相关代码如下：

```
<div id="right-bt">
  <div class="title1">友情链接</div>
  <div class="title_nav">
    <marquee direction="up" scrollamount="5" scrolldelay="100" onmouseover="this.stop()" on-
mouseout="this.start()">
      <ul>
        <li><a href="http://www.gdnfu.com">广东南方职业学院</a></li>
        <li><a href="http://www.webjx.com">网页教学网</a></li>
        <li><a href="http://www.wangyeba.com">网页吧</a></li>
        <li><a href="http://www.dabaoku.com">网页制作大宝库</a></li>
        <li><a href="http://www.16sukai.com">16 素材网</a></li>
        <li><a href="http://www.dyunr.com">网页资源站</a></li>
      </ul>
    </marquee>
  </div>
</div>
</div>
```

7. 制作页脚

页脚内容包括“学生风采”滚动效果和版权信息，“学生风采”滚动效果与“友情链接”的滚动效果不同，这里是采用 JavaScript 脚本实现。完成效果如图 11-9 所示。



图 11-9 “页脚”效果图

相关 CSS 代码如下：

```
#footer {
  width:1000px;
  height:200px;
}
.scroll_div {
```



```
width:1000px;
height:135px;
padding-top:10px;
padding-bottom:10px;
margin:0 auto;
overflow:hidden;
white-space:nowrap;
background:#ffffff;
}
.scroll_div img {
height:130px;
border:1px solid #000;
margin:auto 5px;
}
#scroll_begin,#scroll_end,#scroll_begin ul,#scroll_end ul,#scroll_begin ul li,#scroll_end ul li {
display:inline;
}
.banquan {
width:1000px;
height:70px;
background-image:url(.. /img/bg.gif);
background-repeat:repeat;
text-align:center;
padding-top:10px;
line-height:16px;
}
```

body 中的相关代码如下:

```
<div id="footer">
<script language="javascript">
function ScrollImgLeft() {
var speed=20
var scroll_begin=document.getElementById("scroll_begin");
var scroll_end=document.getElementById("scroll_end");
var scroll_div=document.getElementById("scroll_div");
scroll_end.innerHTML=scroll_begin.innerHTML
function Marquee() {
if(scroll_end.offsetWidth-scroll_div.scrollLeft<=0)
scroll_div.scrollLeft-=scroll_begin.offsetWidth
else
```

```

        scroll_div.scrollLeft++
    }
    var MyMar=setInterval( Marquee , speed)
    scroll_div.onmouseover=function() { clearInterval( MyMar) }
    scroll_div.onmouseout=function() { MyMar=setInterval( Marquee , speed) }
}
</script>
<div class="title1">学生风采</div>
<div style="text-align:center">
    <div class="sqBorder">
        <div id="scroll_div" class="scroll_div">
            <div id="scroll_begin">
                <ul>
                    <li><a href="#"></a></li>
                    <li><a href="#"></a></li>
                    <li><a href="#"></a></li>
                    <li><a href="#"></a></li>
                    <li><a href="#"></a></li>
                    <li><a href="#"></a></li>
                    <li><a href="#"></a></li>
                    <li><a href="#"></a></li>
                    <li><a href="#"></a></li>
                    <li><a href="#"></a></li>
                    <li><a href="#"></a></li>
                    <li><a href="#"></a></li>
                    <li><a href="#"></a></li>
                    <li><a href="#"></a></li>
                    <li><a href="#"></a></li>
                    <li><a href="#"></a></li>
                    <li><a href="#"></a></li>
                    <li><a href="#"></a></li>
                    <li><a href="#"></a></li>
                    <li><a href="#"></a></li>
                </ul>
            </div>
            <div id="scroll_end"></div>
        </div>
    </div>
</div>
<script type="text/javascript">ScrollImgLeft() ;</script>

```


教学课件子页面用于展示课件下载列表，页面效果如图 11-11 所示。



图 11-11 教学课件

在首页的基础上做如下更改：

main 层：用内联样式表更改 main 层的高度和背景颜色（`style = " background: #FFFFFF; height: 920px;"`）；在 main 层里添加 m-left 和 m-right 层。

footer 层：去掉页脚 footer 层内的“学生风采”部分的内容，只保留版权信息。

相关 CSS 代码如下：

```
.m-left {
```

```
width:260px;
background-color:#FFFFFF;
padding-left:10px;
float:left;
margin:4px;
}
.m-left ul li {
list-style:none;
margin-left:20px;
line-height:30px;
border-bottom:1px dashed #000000;
}
.m-left ul li a,.m-left ul li a:link,.m-left ul li a:visited {
color:#000000;
text-decoration:none;
}
.m-right {
width:700px;
float:left;
margin:4px;
padding-left:10px;
border-left:1px dashed #000000;
}
.yso1 {
color:#000;
border-bottom:1px dashed #000000;
}
```

完整的 HTML 代码如下：

```
<! --教学课件-->
<! DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>教学课件</title>
<link type="text/css" rel="stylesheet" href="css/style.css" />
<script src="js/jquery-1.3.2.min.js" type="text/javascript"></script>
<script src="js/scroll.js" type="text/javascript"></script>
</head>
<body>
<div id="container">
<div id="banner">
<embed src="media/1.swf" width="1000" height="240" autostart="true" loop="true"
wmode="transparent"></embed>
```



```
</tr>
<tr>
  <td><table class="ys01" width="100%" border="0" cellspacing="1" cellpadding="
5">
  <tr>
    <td width="18%" rowspan="3" align="center" class="ppt"></td>
    <td colspan="2">课程标题:模块 9&nbsp;  制作可让用户提交信息的表单文档<a name
="kj9" id="kj9"></a></td>
    <td width="28%">创建时间:2016-11</td>
  </tr>
  <tr>
    <td width="29%">媒体类型:ppt</td>
    <td width="25%">资源类型:教学课件</td>
    <td>资源大小:</td>
  </tr>
  <tr>
    <td colspan="2"><a href="kejian/m9.pptx"></a></td>
    <td>&nbsp;  </td>
  </tr>
</table></td>
</tr>
<tr>
  <td><table class="ys01" width="100%" border="0" cellspacing="1" cellpadding="
5">
  <tr>
    <td width="18%" rowspan="3" align="center" class="ppt"></td>
    <td colspan="2">课程标题:模块 10&nbsp;  制作能提供会员注册、登录的动态页面<a
name="kj10" id="kj10"></a></td>
    <td width="28%">创建时间:2016-11</td>
  </tr>
  <tr>
    <td width="29%">媒体类型:ppt</td>
    <td width="25%">资源类型:教学课件</td>
    <td>资源大小:</td>
  </tr>
  <tr>
    <td colspan="2"><a href="kejian/m10.pptx"></a></td>
    <td>&nbsp;  </td>
```




图 11-12 作品展示

修改 main 层的高度 360px，m-left 层的内容修改如下：

```
<div id="m-left" style="width:100px;">
  <div class="title1">作品展示</div>
  <div>
    <ul>
      <li><a href="zuopinzhanshi.html">作品展示 1</a></li>
      <li><a href="zuopinzhanshi2.html">作品展示 2</a></li>
      <li><a href="zuopinzhanshi3.html">作品展示 3</a></li>
    </ul>
  </div>
</div>
```

m-right 层内用旋转的方式展示作品,旋转功能用 JavaScript 脚本实现,m-right 层的代码如下:

```
<div id="m-right" style="width:860px;">
  <script>
    window.onload=function() {
      var rt=new imgRound("imgContainer",300,180,430,50,400,0.01);
      setInterval(function(){rt.roundMove(),20}
    );
  }
  function imgRound(id,w,h,x,y,r,dv,rh,ah){
    if(ah==undefined)ah=1;
```

```

if( rh == undefined) rh = 10;
var dv = dv * ah; // 旋转速度
var pi = 3.1415926575;
var d = pi/2;
var pd = Math.asin( w/2/r );
var smove = true;
var imgArr = new Array();
var objectId = id;
var o = document.getElementById( objectId );
o.style.position = "relative";
var arrimg = o.getElementsByTagName( "img" );
var pn = arrimg.length; // 图片数量
var ed = pi * 2/pn;
for( n = 0; n < arrimg.length; n++ ) {
var lk = arrimg[ n ].getAttribute( "link" );
if( lk != null ) arrimg[ n ].setAttribute( "title", lk );
arrimg[ n ].onclick = function() {
if( this.getAttribute( "link" ) != null ) {
if( this.getAttribute( "target" ) != "_blank" ) window.location = ( this.getAttribute( "
link" ) )
else window.open( this.getAttribute( "link" ) )
}
}
arrimg[ n ].onmouseout = function() { smove = true; };
arrimg[ n ].onmouseover = function() { smove = false; };
arrimg[ n ].style.position = "absolute";
imgArr.push( arrimg[ n ] );
}
this.roundMove = function() {
for( n = 0; n <= pn-1; n++ ) {
var o = imgArr[ n ];
var ta = Math.sin( d+ed * n ), strFilter;
if( ta < 0 ) o.style.left = Math.cos( d+ed * n-pd ) * r+x+"px";
else o.style.left = Math.cos( d+ed * n+pd ) * r+x+"px";
o.style.top = ta * rh+rh+y+"px";
var zoom = Math.abs( Math.sin( ( d+ed * n )/2+pi/4 ) ) * 0.5+0.5;
o.style.width = Math.abs( Math.cos( d+ed * n+pd ) - Math.cos( d+ed * n-pd ) ) * zoom * r
+"px";

o.style.height = zoom * h+"px";
if( ta < 0 ) { ta = ( ta+1 ) * 80+20; o.style.zIndex = 0; }
else { ta = 100; o.style.zIndex = 1 }
if( o.style.zIndex <= 0 ) strFilter = "FlipH( enabled; true )"
else strFilter = "FlipH( enabled; false )" ;
}
}

```

```
        strFilter=strFilter+" alpha(opacity="+ta+")";
        o. style. opacity=ta/100;
        o. style. filter=strFilter;
    }
    if( smove)d=d+dv;
}
}
</script>
<div id="imgContainer" style="width:860px;height:350px;border:1px solid red">
    
    
    
    
    
    
    
    
</div>
</div>
```

imgRound (id, w, h, x, y, r, dv, rh, ah) 方法各参数说明如下:

id: 容器 ID。

w: 图片宽度。

h: 图片高度。

x: 圆心水平位置。

y: 圆心垂直位置。

r: 圆圈半径。

dv: 旋转速度。

rh: 垂直距离 (可选, 默认为 10)。

ah: 旋转方向 (可选, 1: 顺时针, 2: 逆时针, 默认为 1)。

至此, 精品课程网站的主要页面制作完毕, 还有其余 7 个页面没有完成, 分别是课程介绍 (kechengjieshao. html)、教学团队 (jiaoxuetuandui. html)、教学大纲 (jiaoxuedagang. html)、教案 (jiaoan. html)、课后练习 (kehoulianxi. html)、教学视频 (jiaoxueshipin. html)、在线答疑 (zaixiandayi. html) 等, 读者可以在此基础上根据自己的喜好修改相关的 CSS 样式, 进一步美化页面, 也可以结合本章所学内容, 完善这 7 个页面。

参 考 文 献

- [1] 张兵义, 朱立, 王蓓. 网页设计与制作案例教程 (HTML5+CSS3+JavaScript) [M]. 北京: 清华大学出版社, 2018.
- [2] 缪亮, 陶颖. Web 前端设计与开发 (HTML5+CSS3+JavaScript) [M]. 微课版. 北京: 清华大学出版社, 2018.
- [3] 郑娅峰, 张永强. 网页设计与开发 [M]. 北京: 清华大学出版社, 2018.
- [4] 储久良. Web 前端开发技术 (微课版) [M]. 北京: 清华大学出版社, 2018 .
- [5] 储久良. Web 前端设计与开发 (HTML5+CSS3+JavaScript) [M]. 3 版. 北京: 清华大学出版社, 2018.
- [6] 储久良. Web 前端设计基础 (HTML5+CSS3+JavaScript) [M]. 北京: 清华大学出版社, 2017.
- [7] 刘继山. 网页设计技术实用教程从基础到前沿 (HTML5+CSS3+JavaScript) [M]. 北京: 清华大学出版社, 2017.
- [8] 周文洁. HTML5 网页前端设计实战 [M]. 北京: 清华大学出版社, 2017 .
- [9] 王柯柯, 周宏, 刘亚辉, 等. 网页设计技术 (HTML5+CSS3+JavaScript) [M]. 北京: 清华大学出版社, 2017.
- [10] 明日科技. HTML5 从入门到精通 [M]. 2 版. 北京: 清华大学出版社, 2017.
- [11] 刘欢. HTML5 基础知识核心技术与前沿案例 [M]. 北京: 人民邮电出版社, 2016.
- [12] 冯涛, 王海波, 等. 网页设计与制作项目教程 [M]. 大连: 大连理工大学出版社, 2014.