第3章

HTML 技术与 CSS

学习要点

(1) 了解 HTML 文档的基本结构。

(2) 掌握 HTML 的各种常用标记。

(3) 熟练使用各种 HTML 控件标记。

(4) 了解 CSS 的特点。

(5) 掌握 CSS 中各种选择符的定义及其使用方法。

(6) 掌握 CSS 样式常用的属性,例如字体属性、文本属性、颜色和背景属性、边框属性等。

(7) 掌握在 HTML 中使用 CSS 的几种方式。

1989 年,在瑞士日内瓦 CERN 实验室主任 Tim Berners-Lee 的带领下开始实施的信息全球网,其目的在于设计和开发出一系列的概念、通信协议和系统,以支持各种类型信息之间的相互链接。其中发布了基于超文本和超媒体技术的 HTML,该语言属于 1986 年发布的一个信息管理方面的国际标准 SGML 的子集。

这种语言因在 NCSA 的 Mosaic 浏览器中使用而广为流传。在 20 世纪 90 年代,由于互联网的迅速兴起,使得 HTML 空前繁荣。在当时,HTML 被发展成为许多不同的版本。只有那些网页设计者和用户都共有的 HTML 部分才可以被正确浏览。出于这种混乱局面的考虑,合作制定一个公认的 HTML 规范成为当务之急。1995 年 11 月,当时的 Internet Engineering Task Force(IETF)在对 1994 年的常用实践进行编纂整理的基础上,倡导开发了 HTML 2.0 规范。同时,HTML+和 HTML 3.0 为 HTML 提供了更为丰富的内容。这些 HTML 规范的可贵之处在于它们并不在意这些标准的讨论从未取得过一致,而是致力于吸收接纳广泛的新特性。1996 年,W3C 组织的 HTML Working Group 开始编纂新的规范,1997 年 1 月推出了 HTML 3.2,并在其中做了许多重要的改动。

许多人同意的一种观点就是 HTML 文档应该在不同的浏览器和操作平台之间都表现良好。人们希望只设计一种文档的版本,就可以被所有的人看懂。如果不是这样,网络这个无国界的空间就会成为一个混乱不堪的地方,对于所有人来说都需要一个统一的标准。1998 年 4 月 24 日,W3C 组织发布了 HTML 4.0 版本,将原来 HTML 3.2 扩展到了一些全新的领域,例如样式表单、Script 语言、框架结构、内嵌对象、丰富的表格以及增强的表单功能、支持从右到左的文本等。

HTML 5 全称是 HyperText Markup Language 5,用于取代 1999 年所制定的 HTML 4.01 和 XHTML 1.0 标准的 HTML 标准版本。2014 年,W3C 发布了 HTML 5 的最终版,

从那时起，各个网站都开始从 Flash 转向 HTML 5，以期能在互联网应用迅速发展时使网络标准达到符合当代的网络需求。

学习 Web 开发技术，必须首先熟悉 HTML 的常用标记，熟悉其含义和作用。另外，每一个 HTML 标记或者称为 HTML 元素（有时也称 HTML 标签）都可看成一个在浏览器中显示的对象。层叠样式单（Cascading Style Sheets，CSS）的作用就是为网页上的 HTML 元素精确地定位，控制 HTML 元素的外观显示，可以把网页上的显示内容和显示外观相分离。

注意，HTML 标记（tag）和 HTML 元素（element）是两个不同的概念，在 HTML 文档对象编程模型中把 HTML 标记称为元素。实际使用中常常不加区分，本书有时会混用这两种术语。

3.1 HTML 文档基本结构

视频讲解

HTML 文件的结构包括头部（head）、主体（body）两大部分，头部描述浏览器所需的信息，主体包含所要说明的具体内容。HTML 文件通常使用“<标记名>”和“</标记名>”来表示标记的开始和结束，因此在 HTML 文件中标记一般是配对使用的。不配对使用的标记占少数。一个 HTML 页面文件的最基本结构为：

```
<html>
<head>   <title>  浏览器窗口显示的窗口标题 </title>  </head>
<body>  … Web 页面内容  </body>
</html>
```

首先文件中各个标记必须包含在< html >和</html >之间，< head >和</head >标记是网页头部标识，其中的浏览器窗口标题放在< title >和</title >之间。所有需要在浏览器中显示的标记内容放在< body >和</body >之间。

【例 3-1】 在浏览器中显示下面的 HTML 文档。

```
<! DOCTYPE HTML PUBLIC "- //W3C//DTD HTML 4.01 Transitional//EN" "http://www. w3.
org/TR/html4/loose.dtd">
<html>
<!--  这是一个 HTML 文档基本标记演示-->
<head>
<meta http-equiv= "Content-Type" content= "text/html; charset=utf-8">
<title>HTML 文档基本标记演示</title>
</head>
<body>
这是一个 HTML 文档基本标记演示效果！
</body>
</html>
```

该文档的显示效果如图 3-1 所示。

图 3-1 HTML 文档的基本标记显示效果

从例3-1可以看出,一个完整的HTML文档是由文档类型、HTML标记、头元素标记、网页标题标记、主体元素标记、标记的属性、注释标记组成。HTML文档基本结构如表3-1所示。

表3-1 HTML文档基本结构

序号	标记类型	说明

1	文档类型	HTML文档中的文档类型标记DOCTYPE指定了文档中使用了哪个版本的HTML,并可以和哪个验证工具一起使用,以保证此HTML文档与HTML推荐标准的一致。例如:
		`<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">`
		表明此文档应符合W3C制定的HTML 4.01规范。又如在VS 2019中新建的HTML文档中第一行为:
		`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`
		表明此文档应符合W3C制定的XHTML 1.0规范,也就是要求此文档应按照XML文档规范来配对所有标记。
		文档类型是每个HTML文档必需的,如果HTML文档中没有文档类型标记,浏览器会采用默认的方式即W3C推荐的HTML 4.0来处理此HTML文档

2	HTML标记	在编写HTML源代码时以<html>来标记一个HTML文档的开始,以</html>标记整个文档的结束
3	头元素标记	每个HTML文档都包含一个头元素。头元素中的内容一般不会在显示窗口中显示出来。HTML文档的头元素是以<head>开头,以</head>结束
4	网页标题标记	标题是头元素的一部分,因此,<title>…</title>必须包含在<head>…</head>之内。标题会出现在浏览器窗口标题栏上。用户将该网页添加到收藏夹或书签时,其名称默认为网页标题;另外,搜索引擎在进行分类搜索时也会按网页标题搜索

5	主体元素标记	HTML文档的主体部分是一个区域,用来放置文档的内容,例如文本、图像等。主体以<body>开始,以</body>结束。页面背景颜色与图像在该标记中设置方法如下。
		(1) 用图像填充背景。
		`<body background=url>`
		其中,background表示背景图像文件所在的URL地址。例如:
		`<body background="file:///C:/windows/Bubbles.bmp">`
		(2) 用某种颜色填充背景。
		`<body bgcolor=颜色名>`
		其中,bgcolor表示背景颜色。例如:
		`<body bgcolor=Red>`
		颜色可用颜色名称如"green"、十六进制#RRGGBB如"#00FFEE"、函数rgb(r,g,b)如"rgb(20,20,50)"、函数rgb(r%,g%,b%)如"rgb(20%,20%,50%)"来表示。红、绿、蓝颜色分量取值为0~255。颜色分量百分数是相对255而言的。例如:
		`<body bgcolor="rgb(20,20,50)">`

续表

序号	标记类型	说　明
6	标记的属性	HTML 标记中，可用属性来描述标记的外观和行为方式以及内在表现。上面主体元素中 bgcolor 就是 body 的属性。可根据实际情况给标记设置属性。例如： 我是超链接 上述超链接标记中，id 为超链接定义了一个标识，因为 HTML 页面文档中可能有很多个超链接，通过 ID 可以确定是哪个超链接，也可通过 name 属性来指定。title 属性实现了将鼠标指针放在该超链接标记上时，会显示一个动态文本提示框"It's me"。title 属性是为大多数标记所具有的属性。每个标记有很多属性，但有许多是共同属性
7	注释标记	HTML 中的注释始终以"<!--"开始，以"-->"结尾。注释可以帮助人们理解代码。浏览器会忽略注释内部的所有文本和标记

下面介绍 HTML 页面 head 元素中 meta 标签。

meta 标签是 HTML head 区的一个辅助性标签，它位于 HTML 文档头部的< head >标记和< title >标记之间，它提供用户不可见的信息。meta 标签通常用来为搜索引擎 Robots 定义页面主题，或者是定义用户浏览器上的 cookie；它可以用于鉴别作者，设定页面格式，标注内容提要和关键字；还可以设置页面使其可以根据定义的时间间隔刷新本页面的显示等。下面介绍一些实用的 meta 标签。meta 标签分两大部分：HTTP 标题信息（HTTP-EQUIV）和页面描述信息（NAME）。

1. HTTP-EQUIV

HTTP-EQUIV 类似于 HTTP 的头部协议，它回应给浏览器一些有用的信息，以帮助正确和精确地显示网页内容。常用的 HTTP-EQUIV 类型有以下几种。

1）Content-Type 和 Content-Language（显示字符集的设定）

设定页面使用的字符集，用于说明页面制作所使用的文字以及语言，浏览器会根据此来调用相应的字符集显示 page 内容。例如：

```
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<meta http-equiv="Content-Language" content="zh-CN">
```

注意，该 meta 标签定义了 HTML 页面所使用的字符集为 GB2132，就是国标汉字码。如果将其中的"charset＝GB2312"替换成"charset＝Big5"，则该页面所用的字符集就是繁体中文 Big5 码。当浏览一些国外的站点时，IE 浏览器会提示要正确显示该页面需要下载该种语言支持。这个功能就是通过读取 HTML 页面 meta 标签的 content-Type 属性而得知需要使用哪种字符集显示该页面的。如果系统中没有装相应的字符集，则 IE 就提示下载。不同的语言对应不同的 charset，如日文的字符集是 iso-2022-jp，韩文的是 ks_c_5601。目前 charset 一般取为 UTF-8（Unicode Transformation Format-8），它是一种通用字符集。

2）Refresh（刷新）

让网页多长时间（秒）刷新自己，或在多长时间后让网页自动链接到其他网页。例如：

```
<meta http-equiv="Refresh" content="30">
<meta http-equiv="Refresh" content="5; url=http://www.xia8.net">
```

注意,其中的 5 是指停留 5 秒后自动刷新到 URL 网址。

3) Expires(网页失效期限)

指定网页在缓存中的过期时间,一旦网页过期,必须到服务器上重新下载。例如:

```
<meta http-equiv="Expires" content="0">
<meta http-equiv="Expires" content="Wed, 26 Feb 1997 08:21:57 GMT">
```

注意,必须使用 GMT 的时间格式表示多少时间后过期或直接设为数字 0 表示本次调用后就过期。

4) Pragma(Cache 模式)

禁止浏览器从本地机的缓存中调阅页面内容。例如:

```
<meta http-equiv="Pragma" content="No-cache">
```

注意,网页不保存在缓存中,每次访问都刷新页面。这样设定,访问者将无法脱机浏览。

5) Set-Cookie(cookie 设定)

浏览器访问某个页面时会将它保存在缓存中,下次再次访问时就可从缓存中读取,以提高速度。当希望访问者每次都刷新广告的图标,或每次都刷新计数器时,就要禁用缓存了。通常 HTML 文件没有必要禁用缓存,对于 ASP 等页面,就可以使用禁用缓存,因为每次看到的页面都是在服务器动态生成的,缓存就失去意义。如果网页过期,那么存盘的 cookie 将被删除。注意,必须使用 GMT 的时间格式。例如:

```
<meta http-equiv="Set-Cookie" content="cookievalue=xxx;
            expires=Wednesday,21-Oct-98 16:14:21 GMT; path=/">
```

6) Content-Script-Type(设置默认脚本语言)

通过 content 属性为整个页面设置默认脚本语言。例如设为 JavaScript 语言:

```
<meta http-equiv="Content-Script-Type" content="text/javascript">
```

2. name 属性

使用 name 属性的一般格式是:< meta name="…" content="…">

name 属性用于指定网页参数类型,常用的有 keywords、description 等。content 用于指定该参数的实际内容,以便于搜索引擎机器人查找、分类该网页。

1) keywords(关键字)

为搜索引擎提供搜索关键字列表。用法:

```
<meta name="keywords" content="关键字 1,关键字 2,关键字 3,关键字 4,…">
```

注意,各关键字间用英文逗号隔开。

2) description(简介)

description 用来告诉搜索引擎网站的主要内容。例如:

```
<meta name="description" content="你网站的简介">
```

注意,根据现在流行搜索引擎(Google、Lycos、AltaVista 等)的工作原理,搜索引擎先派机器人自动在 Web 上搜索,当发现新的网站时,便检索页面中的 keywords 和 description,并将其加入自己的数据库,再根据关键字的密度将网站排序。

3) Robots(机器人向导)

Robots 用来告诉搜索机器人哪些页面需要索引,哪些页面不需要索引。content 的参

数有 all、none、index、noindex、follow、nofollow，默认是 all。用法：

```
<meta name="Robots" content="all|none|index|noindex|follow|nofollow">
```

注意，all 表示文件将被检索，且页面上的链接可以被查询；index 表示文件仅被检索；follow 表示页面上的链接可以被查询。

4）Author（作者）、Copyright（版权）、Generator（编辑器）

说明：分别标注网页的作者或制作组、标注版权、编辑器的说明。

用法：

```
<meta name="Author" content="张三,abc@sohu.com">
<meta name="Copyright" content="本页版权归 Tim 所有。">
<meta name="Generator" content="PCDATA|FrontPage|">
```

【例 3-2】 一个 meta 标签应用的例子。

```
<head>
<title>文件头，显示在浏览器标题区</title>
<meta http-equiv="Content-Language" content="zh- cn">
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<meta name="制作人" content="Simonzy">
<meta name="主题词" content="HTML 网页制作 C#.NET JavaScript JS">
<meta name="description" content="网站的简介">
<meta name="keywords" content="Web,DHTML,软件开发">
</head>
```

3.2 文本和格式标记

视频讲解

丰富多彩的 HTML 网页需要修饰和衬托其内容，文本和格式标记是标记语言中的最基本的标记。文本和格式标记如表 3-2 所示。

表 3-2 文本和格式标记

序号	标记	说明
1	字体加粗、斜体和下画线标记	…用来使文本以黑体字的形式输出，也可用…标记；…用来使文本以斜体字的形式输出；<u>…</u>用来使文本以加下画线的形式输出
2	标题字体大小标记	描绘网页的新的小节和子小节的 7 个标题，依次为 … 、…、<h7>…</h7>。这些标题的字体大小依次从大到小
3	段落标记	段落标记的作用是将 … 标记之间的文本内容自动组成一个完整的段落。一个段落是有格式的，因此段落标记往往与 align（对齐方式）属性共同使用。使用的语法是 > ，其中，属性 align 的值可以取 left（左对齐）、center（居中）和 right（右对齐）
4	文本换行标记	 用来创建一个软回车换行，没有结束标记。在 … 标记后输入 ，会在该段落后创建一个较大行距的回车换行；如果在段落标记之间输入 ，则会导致行距较小。<wbr>…</wbr>标记可将其中的文本段自动换行显示

续表

序号	标　　记	说　　明
5	文本缩进标记	在文本缩进标记(<blockquote>…</blockquote>)之间加入的文本将会在浏览器中按两边缩进的方式显示出来
6	文本居中	… 标记可将其中的文本居中显示
7	列表标记	(1) <dl>…</dl>、<dt>…</dt>和<dd>…</dd>。 <dl>…</dl>用来创建一个普通的列表,<dt>…</dt>用来创建列表中的上层项目,<dd>…</dd>用来创建列表中最下层项目,<dt>…</dt>和<dd>…</dd>都必须放在<dl>…</dl>标记对之间。 (2) … 、 … 和 … 。 … 标记对用来创建一个标有顺序的列表; … 标记对用来创建无顺序的列表; … (li 是 list item 的缩写,即列表项目)标记对只能在 … 或 … 标记对之间使用,此标记对用来创建一个列表项,若 … 放在 … 之间则每个列表项加上一个数字,若在 … 之间则每个列表项加上一个圆点。无序列表和有序列表分别与 Microsoft Word 中项目符号和编号相对应。它们的含义是一样的
8	文本块标记	… 标记对用来排版大块段落,此标记的用法与 … 标记非常相似,同样可以使用 align 属性。…此处也是文本块标记,用法和 div 标记类似,但该文本块起始和结束不换行(称为行内元素 inline)。div 和 span 标记除可用作文本编辑块功能外还可用作容器标记,也即按钮、文本框等各种标记放在 div 或 span 里面将作为它的子对象元素处理。<address>…</address>标记与 标记作用相同,就是显示一块文本,但里面的字体只能是斜体,主要用来在网页上放置署名信息
9	特殊符号的表示	<hr>标记在网页中可画一条指定粗细和长短的水平线。例如,<hr color=red noshade width=200 size=4 align=right> 表示画一条右对齐的粗红线。 " "、"©"、"®"、"&"、"<"、">"、"&123;"等显示在浏览器中分别表示为空格、"©"、"®"、"&"、"<"、">"、"}"符号

【例 3-3】 体验 HTML 文档格式标记的应用。

该文档的显示效果如图 3-2 所示。

```
<html><head>
<title>HTML 文档格式标记的演示效果</title></head>
<body>
<p align="left"><blockquote>组成一个计算机系统的各种设备称为硬件。可将一台计算机
划分为六个逻辑部件或部分。具体划分如下:</blockquote></p><br>
<ol type=A><li><p align="left">输入部件。</p></li>
<li>  <p align="left">输出部件。</p></li>
<li>  <p align="left">存储器。</p></li>
</ol>
<ul type=circle><li><p align="left">算术逻辑部件(ALU)。</p></li>
<li><p align="left">辅助存储器。     </p></li>
<li><p align="left">中央处理器(CPU)。</p></li>
<li><p align="left">辅助存储器。     </p></li>
</ul>
<div>此处是文本块标记,该文本块起始和结束自动加了软回车功能,起始和结束都换行</div>
<span>此处也是文本块标记,该文本块起始和结束不换行</span>
</body></html>
```

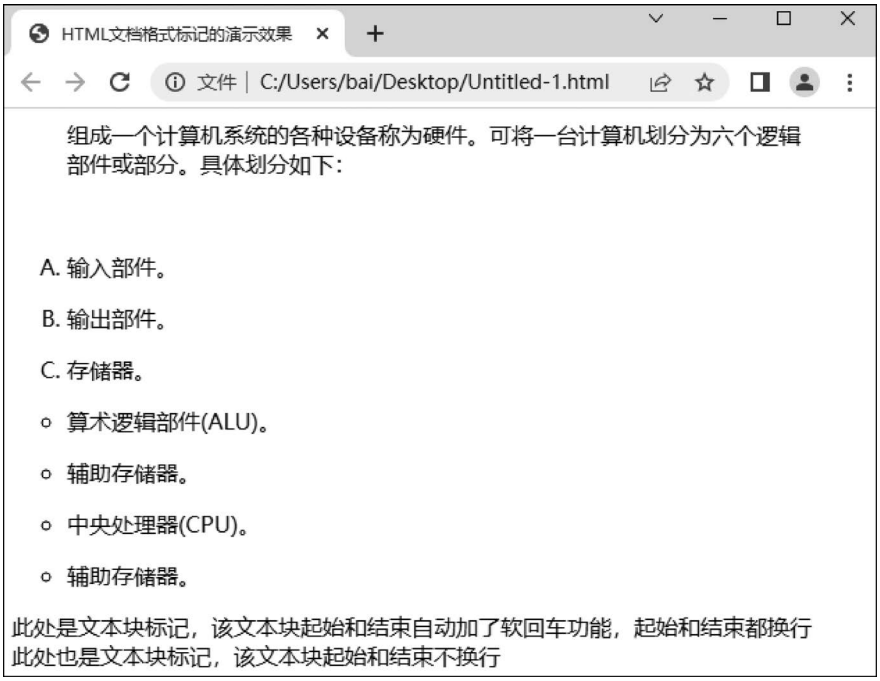

图 3-2　HTML 文档格式标记显示效果

【例 3-4】 HTML 文档常用文本标记。

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>HTML 文档格式标记的演示效果</title>
</head>
<body>
    <h2>这是一本专业的 Dreamweaver MX 2004 的图书</h2>
    <h3><i>What we are doing is just what you need. </i></h3>
    <center><h4><font color="blue"><u>www.cqu.edu.cn</u></font></h4>
</center>
</body>
</html>
```

该文档的显示效果如图 3-3 所示。

图 3-3　HTML 文档常用标记显示效果

3.3 超链接和表格标记

1. 超链接标记

HTML 最重要的功能之一是能创建到其他文档的超链接。在网页中,链接标记是用<a>…来表示的。利用<a>(anchor,锚点)就可以在文档之间建立链接。在 HTML 中,可以充当锚点的既可以是文本,也可以是图像、声音文件。因此锚点元素要求有特定的属性 href(hot reference)来指定超链接。href 属性放在锚点元素<a>内,例如:

```
<a href="http://www.cqu.edu.cn">重庆大学</a>
```

href 属性所指定的链接文件可以是另外一个网站的页面,也可以是本网站中的某一个页面或者是本页面中的某个部位。如果是本网站内的页面链接,href 所指定的文件名应该用相对路径,而不用绝对路径,避免因为域名或者 IP 地址变更等需要对超链接标记进行修改的问题。例如,当前的页面位于根目录下的 files 子目录下,需要超链接到根目录下的 asp 子目录中的 b.htm 页面,超链接标记写法如下:

视频讲解

```
<a href="../asp/b.htm">公司简介</a>
```

也可写成:

```
<a href="/asp/b.htm">公司简介</a>
```

a 标记有一个 target 属性,其取值可为_blank、_top、_self、_parent 等,其中,_blank 是链接文件在新的窗口中打开;_parent 是链接文件将在当前窗口的父级窗口中打开;_self 是链接文件在当前窗口(帧)中打开;_top 是链接文件全屏显示。下例表示在新打开的浏览器窗口中显示重庆大学的主页:

```
<a href="http://www.cqu.edu.cn" target="_blank">重庆大学</a>
```

另外,也可在同一个文件中建立链接,方法是先在 href 中声明链接名称(必须带有#号),再用 name 属性说明被链接的名称,这样单击超链接就会转向本页中其他位置。例如:

```
<a href="# myAnchor">跳转到本页尾部 </a>
…
<a name="myAnchor">此处为本页尾部信息 </a>
```

如果要在一个页面 A.htm 中直接跳转到另外一个页面 B.htm 的某个部位,在 A.htm 页面中建立超链接:

```
<a href="b.htm# myAnchor">查看软件工程专业介绍</a>
```

注意页面 A.htm 中 href 属性的书写格式。

2. 表格标记

与文本相比,表格的主要优点是简洁,给人一目了然的感觉。应用到表格的所有标记和文本都包含在<table>…</table>内。表格主要有三个属性,即 border、width 和 height。border 属性用于设置表格边框的宽度,以像素点为单位。如果想显示边框,可指定 border="1",反之则设置为 border="0"。而 width 和 height 属性分别设置表格的宽度和高度。可将表格的宽度/高度设置为一个像素值,或所占屏幕宽度/高度的百分比值。通过下面的方法定义表格:

```
<table border=1 width=80%>
<tr><th>Heading 1</th><TH>Heading 2</th></tr>
<tr><td>Row 1, Column 1 text.</td><td>Row 1, Column 2 text.</td></tr>
<tr><td>Row 2, Column 1 text.</td><td>Row 2, Column 2 text.</td></tr>
</table>
```

表格样式结构如图 3-4 所示。

Heading 1	Heading 2
Row 1, Column 1 text.	Row 1, Column 2 text.
Row 2, Column 1 text.	Row 2, Column 2 text.

图 3-4　表格显示页面

其中，< table >…</table >定义表格；< tr >…</tr >定义表的行；< th >…</th >定义列标题；< td >…</td >定义表格数据单元。图 3-4 所示表格的另外一种定义方法是：

```
<table border="1" width="500">
<thead><tr><th>Heading 1</th><th>Heading 2</th></tr></thead>
<tbody>
<tr><td>row 1, Column 1 text.</td><td>Row 1, Column 2 text.</td></tr>
</tbody>  <tbody>
<tr><td>Row 2, Column 1 text.</td><td>Row 2, Column 2 text.</td></tr>
</tbody>  </table>
```

其中，< thead >…</thead >定义表格的表头；< tbody >…</tbody >用于格式化和分组表格，可将表格按行进行分组，以便进行分组色彩指定或用于其他目的。

很多网站在页面布局时喜欢采用表格来布局。一个网页要尽量避免用整个一张大表格，所有的内容都嵌套在这个大表格之内。因为浏览器在解释页面的元素时，是以表格为单位逐一显示，如果一张网页是嵌套在一个大表格之内，那么其后果很可能是，当浏览者访问页面时，必须等待所有页面信息下载完毕后网页内容才出现。在这种情况下，应使用< tbody >标记，以便能够使这个大表格分块显示，减少用户等待时间。

【例 3-5】 体会用< tbody >将表格的行进行分组的过程。

```
<table border=" " bgcolor="lightslategray">
<thead bgcolor="lightskyblue">
  <tr><th>Stock symbol</th><th>High</th><th>Low</th><th>Close</th></tr>
</thead>
<tbody bgcolor="lemonchiffon">
  <tr><td>ABCD</td><td>88.625</td><td>85.50</td><td>85.81</td></tr>
  <tr><td>EFGH</td><td>102.75</td><td>97.50</td><td>100.063</td></tr>
</tbody>
<tbody bgcolor="goldenrod">
  <tr><td>IJKL</td><td>56.125</td><td>54.50</td><td>55.688</td></tr>
  <tr><td>MNOP</td><td>71.75</td><td>69.00</td><td>69.00</td></tr>
</tbody>
<tfoot bgcolor="lightskyblue">
  <tr><td colspan="4">Quotes are delayed by 20 minutes.</td></tr>
</tfoot>
<caption valign="BOTTOM" style="font-size=12;">
  Created using HTML
</caption>
</table>
```

该文档的显示效果如图 3-5 所示。< tfoot >…</tfoot >定义了表格的脚注。< caption >…</caption >提供表格的简要信息描述。

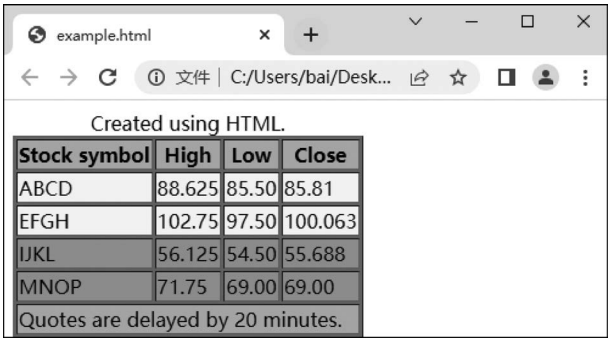

图 3-5　带 tbody 的表格显示效果

3.4　图像、视频、声音处理标记

视频讲解

图像、视频与动画、声音处理标记如表 3-3 所示。

表 3-3　图像、视频与动画、声音处理标记

序号	标　　记	说　　明

1	图像标记	标记格式为：
		`<img src=url alt=text_1 border=n_1 height=n_2 width=n_3 align=mode hspace=n_3 vspace=n_4>`
		其中，src 表示图像来源(source)文件所在的 URL 地址，alt 表示将鼠标移到该图像上出现的文字提示 text_1，border 表示图像对象的边界厚度为 n_1，height 和 width 表示图像的高度和宽度分别为 n_2 和 n_3，hspace 和 vspace 表示图像横向和纵向的空白边幅分别为 n_3 和 n_4。align 表示图像的放置方式，mode＝absbottom|absmiddle|baseline|bottom|left|middle|right|texttop|top。例如：
		`<img src="http://202.202.0.35/images/1.jpg" height=200 width=300>`

2	视频标记	标记格式为：
		`<video src=url_1 width=n_1 height=n_2 poster=url_2 preload=auto autoplay loop controls>`
		其中，src 用于指定视频的地址。width 和 height 分别用于设置播放器的宽度和高度。poster 用于指定一张图片，在当前视频数据无效时显示。preload 定义视频是否预加载，none 表示不进行预加载，metadata 表示部分预加载，auto 表示全部预加载。添加 autoplay 则视频就绪后会马上播放。添加 loop 则视频会循环播放。添加 controls 则播放器会向用户显示控件。例如：
		`<video src="avideo.mp4" height=200 width=300 Loop >`
		`<video src="animation.webm" height=20 width=30 controls >`

3	音频标记	标记格式为：
		`<audio src=url controls loop muted preload="auto">`
		其中，src 用于指定音频的地址。添加 controls 会显示默认的音乐面板。添加 loop 会开启循环播放。添加 muted 会静音。preload 用于控制音频的预加载方式，none 表示不进行预加载，metadata 表示部分预加载，auto 表示全部预加载。例如：
		`<video src=" ieeec.mp3" loop preload="auto">`

续表

序号	标　　记	说　　明
4	插件标记	用浏览器插件来播放声音的方法如下： < embed width = "128" height = "128" src = "file:///C:/windows/media/canyon.mid" loop = false autostart = true mastersound hidden=true> 其中，autostart 表示页面一装入即开始播放，hidden 表示无任何播放界面出现。要求必须事先装入 ActiveMovie 多媒体插件才可播放

【例 3-6】 在一个 3×3 的表格中间显示一张图片。

```
<! DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html><head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>HTML 文档图像、表格标记的演示效果</title></head>
<body>
   <table border=1 style="border-collapse: collapse" ><caption >表格显示图片
</caption>
      <tr><td>公司名称</td><td>图片</td><td>备注</td></tr>
      <tr><td>company1</td><td><img src="ad1.jpg"></td><td>半年</td></tr>
<tr><td>company2</td><td>无</td><td>未签约</td></tr></table>
</body></html>
```

该文档的显示效果如图 3-6 所示。

图 3-6　在表格中间显示图片

3.5　控件标记

视频讲解

控件标记包括表单标记、按钮标记、文本框、文本区域、复选框、选项按钮、选项菜单、文件域、分组框等标记。它们的介绍如下。

1. 表单

表单(form)用于从用户(站点访问者)收集信息，然后将这些信息提交给服务器进行处理。表单中可以包含允许用户进行交互的各种控件，例如文本框、列表框、复选框和单选按钮等。用户在表单中输入或选择数据之后将其提交，该数据就会送交给表单处理程序进行处理。表单的使用包括两部分：一是用户界面，提供用户输入数据的元件；二是处理程序，可以是客户端程序，在浏览器中执行，也可以是服务器处理程序，处理用户提交的数据，返回

结果。表单通过 form 标记来定义：

```
<form 属性="值"… 事件="代码">…</form>
```

1）属性

name：表单的名称。命名表单后，可以使用脚本语言来引用或控制该表单。

method：表单数据传输到服务器的方法。其取值如下：post，在 HTTP 请求中嵌入表单数据；get，将表单数据附加到请求该页的 URL 中。注意，若要使用 get 方法发送，URL 的长度应限制在 8192 个字符以内。如果发送的数据量太大，数据将被截断，从而导致意外的或失败的处理结果。此外，在发送用户名和密码、信用卡号或其他机密信息时，不要使用 get 方法，而应使用 post 方法。

action：接收表单数据的服务器端程序或动态网页的 URL 地址。

target：目标窗口。其取值如下：_blank，在未命名的新窗口中打开目标文档；_parent，在显示当前文档的窗口的父窗口中打开目标文档；_self，在提交表单所使用的窗口中打开目标文档；_top，在当前窗口内打开目标文档，确保目标文档占用整个窗口。

在一个网页中可以创建多个表单，每个表单都可以包含各种各样的控件，例如文本框、单选按钮、复选框、下拉菜单以及按钮等。表单不能嵌套使用。

2）事件

onsubmit：提交表单时调用的事件处理程序。

onrest：重置表单时调用的事件处理程序。

2. 普通按钮、提交按钮、复位按钮

使用 input 标记可以在表单中添加三种类型的按钮：提交按钮、重置按钮和自定义按钮。创建按钮的方法如下：

```
<input type="submit | reset | button" 属性="值"… onclick="代码">
```

1）属性

type=submit：创建一个提交按钮。在表单中添加提交按钮后，站点访问者就可以在提交表单时，将表单数据（包括提交按钮的名称和值）以 ASCII 文本形式传送到由表单的 action 属性指定的表单处理程序。一般来说，表单中必须有一个提交按钮。

type=reset：创建一个重置按钮。单击该按钮时，将删除任何已经输入到域中的文本并清除所做的任何选择。但是，如果框中含有默认文本或选项为默认，单击重置按钮将会恢复这些设置值。

type=button：创建一个自定义按钮。在表单中添加自定义按钮时，为了赋予按钮某种操作，必须为该按钮编写脚本。

name：按钮的名称。

value：显示在按钮上的标题文本。

2）事件

onclick：单击按钮执行的脚本代码。

3. 文本框

在表单中添加文本框可以获取站点访问者提供的一行信息。创建文本框的方法如下：

```
<input type="text" 属性="值"…事件="代码"…>
```

1）属性

name：单行文本框的名称，通过它可以在脚本中引用该文本框控件；value：文本框的值；

defaultvalue：文本框的初始值；size：文本框的宽度（字符数）；maxlength：允许在文本框内输入的最大字符数，用户输入的字符数可以超过文本框的宽度，这时系统会将其滚动显示，但输入的字符数不能超过输入的最大字符数；form：所属的表单（只读）。当 type="password"时，用户输入的文本以 * 呈现，可用于输入用户密码。

2）方法

click：单击该文本框；focus：得到焦点；blur：失去焦点；select：选择文本框的内容。

3）事件

onclick：单击该文本框执行的代码；onblur：失去焦点执行的代码；onchange：内容变化执行的代码；onfocus：得到焦点执行的代码；onselect：选择内容执行的代码。

例如：

```
<input type=text name="nm" value="">
```

4. 文本区域

在表单中添加文本区域可以接受站点访问者输入多于一行的文本。创建文本区域方法如下：

```
<textarea 属性="值"… 事件="代码"…>初始值</textarea >
```

属性介绍如下。

name：滚动文本框控件的名称；rows：控件的高度（以行为单位）；cols：控件的宽度（以字符为单位）；readonly：滚动文本框的内容不被用户修改。

创建多行文本框时，在< textarea >和</textarea >标记之间输入的文本将作为该控件的初始值。它的其他属性、方法和相关事件与单行文本框基本相同。当提交表单时，该域名称和内容都会包含在表单结果中。

5. 复选框

在表单中添加复选框可以让站点访问者去选择一个或多个选项或不选项。创建复选框的方法如下：

```
<input type="checkbox" 属性="值"… 事件="代码"…>选项文本
```

1）属性

name：复选框的名称。

value：选中时提交的值。

checked：设置当第一次打开表单时该复选框处于选中状态。该复选框被选中时，值为true，否则为 false。该属性是可选的。

defaultchecked：判断复选框是否定义了 checked 属性。若定义了 checked 属性，则defaultchecked 为 true，否则为 false。

2）方法

focus：得到焦点；blur：失去焦点；click：单击该复选框。

3）事件

onfocus：得到焦点执行的代码；onblur：失去焦点执行的代码；onclick：单击该复选框

执行的代码。

当提交表单时,假如复选框被选中,它的内部名称和值都会包含在表单结果中。否则,只有名称会被纳入表单结果中,值则为空白。

例如:

```
签字笔<input type=checkbox name="ch1" checked>
钢笔<input type=checkbox name="ch2">
圆珠笔<input type=checkbox name="ch3">
```

6. 选项按钮

在表单中添加选项按钮可以让站点访问者从一组选项中选择其中之一。在一组单选按钮中,一次只能选择一个。创建选项按钮方法如下:

```
<input type="radio" 属性="值"… 事件="代码"…>选项文本
```

属性介绍如下。

name:单选按钮的名称,若干名称相同的单选按钮构成一个控件组,在该组中只能选中一个选项;value:提交时的值;checked:设置当第一次打开表单时该单选按钮处于选中状态,该属性是可选的。

单选按钮的方法和事件与复选框基本相同。当提交表单时,该单选按钮组名称和所选取的单选按钮指定值都会包含在表单结果中,如果没有任何单选按钮被选取,组名称会被纳入表单结果中,值则为空白。

例如:

```
<input type=radio checked name=kd value="教师">教师
<input type=radio name=kd value="学生">学生
<input type=radio name=kd value="公务员">公务员
<input type=radio name=kd value="医生">医生
```

7. 选项菜单

表单中的选项菜单让站点访问者从列表或菜单中选择选项。菜单中可以选择一个选项,也可以设置为允许进行多重选择。创建选项菜单方法如下:

```
<select name="值" size="值" [multiple]>
<option [selected] value="值">选项 1</option>
<option [selected] value="值">选项 2</option>
…
</select>
```

属性介绍如下。

name:选项菜单控件的名称;size:在列表中一次可以看到的选项数目(默认为 1),若大于则相当于列表框;multiple:允许作多项选择;selected:该选项的初始状态为选中。

当提交表单时,菜单的名称会被包含在表单结果中,并且其后有一份所有选项值的列表。

例如:

```
<html><head><title>文件域示例</title></head>
<body>
<form action="Getcourse.asp" method="post">
<select name="课程">
```

```
<option value="计算机基础" selected>计算机基础</option>
<option value="C语言程序设计">c语言程序设计</option>
<option value="数据结构">数据结构</option>
<option value="数据库原理">数据库原理</option>
<option value="C++程序设计">C++程序设计</option>
</select>
</form>
</body></html>
```

8. 文件域

文件域由一个文本框和一个“浏览”按钮组成，用户既可以在文本框中输入文件的路径和文件名，也可以通过单击“浏览”按钮从磁盘上查找和选择所需文件。文件域一般用于选择文件上载到服务器。创建文件域方法如下：

```
<input type="file" 属性="值"…>
```

属性介绍如下。

name：文件域的名称；value：初始文件名；size：文件名输入框的宽度。

一个文件域如图 3-7 所示。

图 3-7　一个文件域

源代码如下：

```
<html><head><title>文件域示例</title></head>
<body>
<form action="GetCourse.asp" method="post" enctype="multipart/ form-data">
<table align=center bgcolor=#D6D3CE width=368>
<tr><th colspan=2 bgcolor=#0034FF><font color=#FFFFFF>文件域</font></th>
</tr>
<tr><td height=52 align=right>请选择文件:</td>
<td height=52><input type=file name=F1 size=16></td></tr>
<tr align=center>
<td height=52 align=right><input type=submit value=提交 name=btnSubmit></td>
<td height=52><input type=reset value=全部重写 name=btnReset></td></tr>
</table></form></body></html>
```

为了能使服务器接收到选择的文件，表单中应包含 enctype 属性值指定提交数据的格式。

9. 分组框

可以使用 fieldset 标记对表单控件进行分组，从而将表单细分为更小、更易于管理的部分。fieldset 标记必须以 legend 标记开头，以指定控件组的标题，在 legend 标记之后可以跟其他表单控件，也可以嵌套 fieldset。创建表单控件分组的方法如下：

```
<fieldset>
<legend>控件组标题</legend>
组内表单控件
</fieldset>
```

例如：

```
<html><head><title>文件域示例</title></head><body>
<center> 
<fieldset style="width:300px; z-index: 103; left: 115px; position: absolute;
top: 260px; background-color: #ff99cc;">
```

```
<legend>用户登录</legend>
<form name="login.asp" method="post">
账号:<input name="UserName"></input><br><br>
密码:<input type="password" name="UserPassword"></input><br><br>
<input type="submit" value="登录" name="Submit"></input>   
<input type="reset" value="重填" name="Reset"></input>
</form>
</fieldset>
</body></html>
```

显示效果如图 3-8 所示。

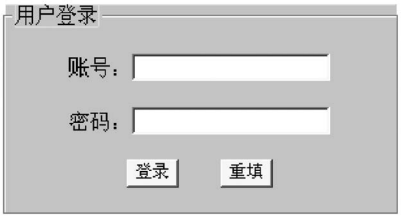

图 3-8　分组框显示效果

视频讲解

3.6　HTML 5 介绍

2014 年,W3C 发布了 HTML 5 的最终版。广义论及 HTML 5 时,实际指的是包括 HTML、CSS 和 JavaScript 在内的一套技术组合。它希望能够减少浏览器对于插件和富互联网应用(Plug-in-based Rich Internet Application,RIA),如 Adobe Flash、Microsoft Silverlight,与 Oracle JavaFX 等的需求,并且提供更多能有效增加网络应用的标准集。

具体来说,HTML 5 添加了许多新的语法特性,其中包括< video >、< audio >和< canvas >元素,同时整合了 SVG 内容。这些元素是为了更容易地在网页中添加和处理多媒体与图片内容而添加的。其他新的元素包括< section >、< article >、< header >和< nav >,是为了丰富文档的数据内容。新属性的添加也是为了同样的目的。同时也有一些属性和元素被移除掉了。一些元素,像< a >、< cite >和< menu >被修改,重新定义或标准化。同时应用程序接口(API)和文档对象模型(DOM)已成为 HTML 5 中的基础部分。HTML 5 还定义了处理非法文档的具体细节,使得所有浏览器和客户端程序能够一致地处理语法错误。

HTML 5 具有以下特性。

1. 语义特性

HTML 5 赋予网页更好的意义和结构。更加丰富的标签可构建对程序和用户都更有价值的数据驱动。

2. 本地存储特性

基于 HTML 5 开发的网页将拥有更短的启动时间和更快的联网速度,这些全得益于 HTML 5 APP Cache,以及本地存储功能。

3. 设备兼容特性

HTML 5 为网页应用开发者们提供了更多功能上的优化选择,带来了更多体验功能的优势。HTML 5 提供了前所未有的数据与应用接入开放接口,使外部应用可以直接与浏览

器内部的数据直接相连，例如视频影音可直接与麦克风及摄像头相连。

4. 连接特性

更有效的连接工作效率，使得基于页面的实时聊天、更快速的网页游戏体验、更优化的在线交流得以实现。HTML 5 拥有更有效的服务器推送技术，Server-Sent Event 和 WebSockets 就是其中的两个特性，这两个特性能够帮助实现服务器将数据“推送”到客户端的功能。

5. 网页多媒体特性

支持网页端的 Audio、Video 等多媒体功能。

6. 三维、图形及特效特性

基于 SVG、Canvas、WebGL 及 CSS3 的 3D 功能，用户会惊叹于在浏览器中所呈现的惊人视觉效果。

7. 性能与集成特性

没有用户会永远等待你的网页加载，HTML 5 会通过 XMLHttpRequest2 等技术，帮助 Web 应用和网站在多样化的环境中更快速地工作。

8. CSS3 特性

在不牺牲性能和语义结构的前提下，CSS3 中提供了更多的风格和更强的效果。此外，较之以前的 Web 排版，Web 的开放字体格式也提供了更高的灵活性和控制性。

目前，支持 HTML 5 的浏览器众多，包括 EDGE、IE 10、Mozilla Firefox 16、Opera 12、Safari 6、Google Chrome 等。

除了之前提到的标签外，HTML 5 还添加了新的标签，介绍如下。

(1) 视频与音频。

HTML 5 规定了一种通过 video 标记来播放视频。video 标记支持 Ogg、MP4、WebM、M4V、3GPP、QuickTime 等多种视频格式。Ogg：带有 Theora 视频编码和 vorbis 音频编码的文件。MP4：带有 H.264 视频编码和 AAC 音频编码的文件。WebM：带有 VP8 视频编码和 Vorbis 音频编码的文件。video 标记提供了播放、暂停和音量控件来控制视频。其基本语法为：

```
<video src="movie.ogg" width="320"height="240" controls="controls" >
您的浏览器不支持 video 标记。
</video>
```

可通过 audio 标记来播放音频，例如：

```
<audio src="movie.mp4" width="320" height="240" controls="controls">
</audio>
```

(2) 画布。

画布(canvas)用于位图的绘制，通过脚本完成，通常使用 JavaScript 语言。画布通过 canvas 标记来定义：

```
<canvas 属性="值">…</canvas>
```

属性介绍如下。

id：画布标记的编号，用来在脚本中获取标记。

width：画布的宽度。

height：画布的高度。

在脚本中，先使用 DOM 获取位图标记，再调用 CanvasRenderingContext2D 对象，使用 CanvasRenderingContext2D 对象的方法即可进行绘图，代码如下：

```
var canvas=document.getElementById("canvas 标记的 id");
var ctx=canvas.getContext('2d');
```

CanvasRenderingContext2D 对象的绘图方法有很多，如以下几种。

绘制填充的矩形：fillRect(x,y,width,height)。

绘制矩形的边框：strokeRect(x,y,width,height)。

绘制字符串：fillText(text,x,y[,maxWidth])，strokeText(text,x,y[,maxWidth])，[]中的内容为可选内容。

完整例子如下：

```
<!DOCTYPE html>
<html><head><title>canvas 例子</title></head>
    <body><canvas id="myCanvas" width="400" height="500"></canvas></body>
</html>
<script type="text/javascript">
    var canvas=document.getElementById("myCanvas");
    var ctx=canvas.getContext('2d');
    ctx.fillStyle='#f00';
    ctx.strokeRect(0,0,220,50);
    ctx.fillStyle='#f0f';
    ctx.font='bold 40px 宋体';
    ctx.fillText('canvas 例子',2,40);
</script>
```

绘图效果如 3-9 所示。

canvas例子

图 3-9　canvas 标记绘制效果

(3) 矢量图。

矢量图标记 svg 可以直接在 HTML 页面中绘制可伸缩矢量图形。svg 使用 XML 格式定义图形，改变尺寸的情况下其图形质量不会有损失。创建 svg 标记的方法如下：

```
<svg>  绘图标记  </svg>
```

绘图标记如下。

circle：用来绘制圆形。属性 cx、cy 定义圆的坐标，r 定义半径，stroke 定义边的颜色，fill 定义圆内填充的颜色。

polygon：用来画多边形。属性 points 中填写所有点的坐标，style 定义多边形样式。

例如：

```
<svg height="190">
  <polygon points="100,10 40,180 190,60 10,60 160,180" style="fill:red;stroke:
purple;stroke-width:5;fill-rule:evenodd;">
</svg>
```

效果如图 3-10 所示。

(4) 计算。

output 标记用来显示计算或用户操作的结果。创建 output 标记的方法如下：

```
<output 属性="值">…</output>
```

属性介绍如下。

name：output 标记的唯一名称，用于表单提交。

for：描述计算中使用的元素与计算结果之间的关系。

form：定义输入字段所属的一个或多个表单。

例如：

```
<form method="get" oninput="num.value=parseInt(num1.value)+parseInt(num2.
value)">
<input type="number" id="num1">+
<input type="number" id="num2">=
<output name="num" for="num1 num2"></output>
</form>
```

效果如图 3-11 所示。

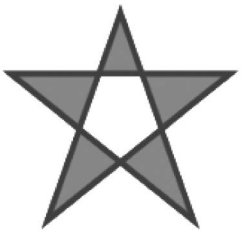

图 3-10　svg 绘图效果

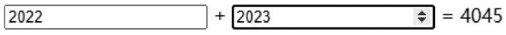

图 3-11　output 标记效果

(5) 预定义选项。

预定义选项标记 datalist 被用来为 input 标记提供“自动完成”的特性。用户能看到一个下拉列表，里边的选项是预定义好的，将作为用户的输入数据。需要使用 input 标记的 list 属性来绑定 datalist 标记。

例如：

```
<input list="browsers">
      <datalist id="browsers">
      <option value="Internet Explorer"></option>
      <option value="Firefox"></option>
      <option value="Chrome"></option>
      <option value="Opera"></option>
      <option value="Safari"></option>
      </datalist>
```

效果如图 3-12 所示。

图 3-12　datalist 标记效果

(6) 块级标签。

块级标签总是独占一行，表现为另起一行开始，其后元素也必须另起一行显示，并且宽度、高度、内边距和外边距都可以控制。为了使代码更具语义性，HTML 5 新加了许多块级标签。

① <header>：标签定义文档的页眉(介绍信息)。

② <section>：标签定义文档中的节，如章节、页眉、页脚或文档中的其他部分。

③ <footer>：标签定义文档或节的页脚，通常包含文档的作者、版权信息、使用条款链接、联系信息等。

④ <aside>：标签定义其所处内容之外的内容，其内容应该与附近的内容相关，可做侧栏。

⑤ <nav>：标签定义导航链接的部分。

⑥ <article>：标签规定独立的自包含内容，元素的潜在来源有论坛帖子、报纸文章、博客条目、用户评论等。

⑦ <figure>：标签规定独立的流内容，如图像、图表、照片、代码等。

⑧ <main>：标签规定文档的主要内容，其标签中的内容对于文档来说应当是唯一的，不应包含在文档中重复出现的内容。

(7) 新的输入类型。

新增 input 标记的输入类型是 color(选择颜色)、tel(电话)、email(邮箱)、number(数值)、range(数值范围内的滑动条)、search(站点搜索)、url(URL 地址)和可供选取日期和时间的日期选择器。即 color：颜色选择器；date：选取日、月、年；month：选取月、年；week：选取周和年；time：选取时间(小时和分钟)；datetime：选取时间、日、月、年(UTC 时间)；datetime-local：选取时间、日、月、年(本地时间)。例如，颜色选择如下：

```
<label for="bgColor">选择背景颜色,改变浏览器背景色:</label>
<input type="color" id="bgColor" name="bgColor" onchange="document.body.
style.backgroundColor=this.value;">
```

上述代码在浏览器中的显示效果如图 3-13 所示。其他输入类型如下：

```
开学日期:<input type="date"/><br/>
开学起始周:<input type="week" name="userdate"/><br/>
起始月:<input type="month" name="userdate"/><br/>
交费时间:<input type="time" name="userdate"/><br/>
日期与时间:<input type="datetime" name="userdate"/><br/>
本地日期时间:<input type="datetime- local" name="mydate"/><br/>
输入邮箱:<input type="email" name="myemail"/><br/>
站内搜索:<input type="search" name="insidesearch"/><br/>
电话号码:<input type="tel" name="usrtel"/><br/>
个人主页:<input type="url" name="homepage"/><br/>
年龄:<input type="range" name="age" min="1" max="120"/><br/>
期望薪酬:<input type="number" name="quantity" min="2500" max="10000" step=
"100" value="2500"/><br/>
```

上述代码在浏览器中显示效果如图 3-14 所示。

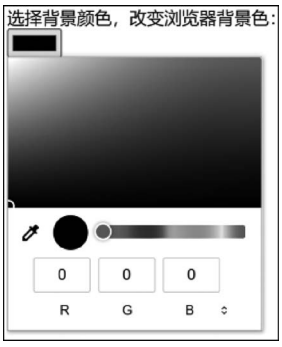

图 3-13　颜色选择

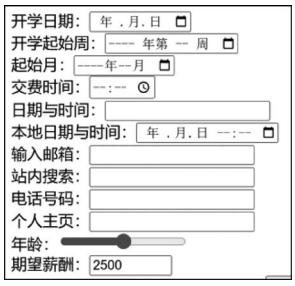

图 3-14　新的输入类型

(8) 本地存储。

Web Storage 是 HTML 5 引入的一个非常重要的功能,可以在客户端本地存储数据,类似 HTML 4.0 的 cookie,但它是为了更大容量存储而设计的,可实现的功能要比 cookie 强大得多。cookie 的大小是受限的,被限制在 4KB,并且每次请求新的页面时,cookie 都会被发送过去。Web Storage 中提供了 localStorage 和 sessionStorage 对象以"键-值对"实现本地存储,网站在页面加载完毕后可以通过 JavaScript 来获取这些本地数据。localStorage:用于长期存储数据,即使在浏览器关闭和重新打开后,数据仍然存在。存储限制通常为每个域名 5～10MB。sessionStorage:用于短期存储数据,只在单个浏览器会话期间有效。存储限制通常与 localStorage 相同。例如:

```
<script type="text/javascript">
    sessionStorage.setItem("姓名", "张三");          //保存姓名
    alert(sessionStorage.getItem("姓名"));          //获取姓名,输出张三
    sessionStorage.removeItem("姓名");              //删除姓名
    sessionStorage.clear();                         //删除所有数据
</script>
```

localStorage 和 sessionStorage 只能提供存储简单数据结构的数据,对于复杂的 Web 应用数据却无能为力。于是 HTML 5 还提供了 IndexedDB 对象,一个轻量级的 NoSQL 浏览器端的数据库,允许用户直接通过 JavaScript 的 API 在浏览器端创建一个本地的数据库,而且支持增、删、改、查操作,让离线的 Web 应用能更加方便地存储数据。原本必须要保存在服务器数据库中的内容,现在可以直接保存在客户端本地了,这将大幅减轻服务器端的负担,同时加快访问数据的速度。本地存储类似于比较老的技术——cookie 和客户端数据库的融合,但由于支持多个 Windows 存储,因而拥有更好的安全性和更强的性能,即使在浏览器关闭后也可以进行保存。

能够保存数据到用户浏览器中意味着可以简单创建一些应用特性,例如,保存用户信息、缓存数据、加载用户上一次的应用状态。

关于 HTML 5 其他标志,限于篇幅,可参看相关书籍。

3.7 CSS 基础

视频讲解

CSS 是 W3C 协会为弥补 HTML 在显示属性设定上的不足而制定的一套扩展样式标准。早在 1996 年 W3C 便提出了一个定义 CSS 的草案,很快这个草案就成为一个被广泛采纳的标准。1998 年,W3C 在原有草案的基础上进行了扩展,建立了 CSS2 规范。2001 年,W3C 继续进行扩展,完成了 CSS3 的草案规范,它不仅包括原有的表现形式标准,还有许多增强功能。

CSS 重新定义了 HTML 中原来的文字显示样式,并增加了一些新概念,如类、层等,还可以处理文字重叠、定位等,它提供了更丰富的样式。同时 CSS 可集中进行样式管理,允许将样式定义单独存储于样式文件中,把显示的内容和样式定义分离,便于多个 HTML 文件共享。一个 HTML 文件也可以应用多个 CSS 样式文件。

CSS 是一种制作网页的新技术,现在已经成为网页设计必不可少的工具之一。W3C 将 CSS3 分成了一系列模块,浏览器厂商按 CSS 节奏快速更新,因此通过采用模块方法,CSS3

规范中的元素以不同的速度向前发展。CSS4 仍在开发过程中。本书将介绍 CSS3 标准的部分内容。

3.7.1 CSS 的特点

前面学习了 HTML 以及编写网页的基本方法,已经注意到 HTML 注重的是内容本身,而不是显示方式。但实际上,网页作者不得不考虑很多关于网页布局、排版方面的问题,以提供给用户尽量美观、易读的网页。

HTML 中标记和文档内容是混写在一起的,标记只是以特有的“<”和“>”记号来区分的,这给用户编写、管理和维护网页带来了很大不便。假如有 100 个网页,每个网页都包含了一段特定格式的文字(如字体为宋体、绿色),一旦需要对显示格式进行修改(如将绿色改为红色),可能需要对这 100 个网页都进行一遍相同的修改,这是让人无法忍受的。

使用 CSS 可以很方便地管理显示格式方面的工作,首先,它能够为网页上的元素精确地定位,让网页设计者自由控制文字、图片在网页上按要求显示;其次,它能够实现把网页上的内容结构和格式控制相分离。浏览者想要看的是网页上的内容结构,而为了让浏览者更好地看到这些信息,就要通过格式控制来帮忙。内容结构和格式控制相分离,使得网页可以仅由内容构成,而将所有网页的格式控制指向某个 CSS 样式表文件。这样就带来以下两方面的好处。

(1) 简化了网页的格式代码,外部的样式表还会被浏览器保存在缓存里,加快了下载显示的速度。

(2) 只要修改保存着网站格式的 CSS 样式文件就可以改变整个站点的风格特色,在修改页面数量庞大的站点时,显得格外有用,避免了一个一个网页地修改,大大减少了重复劳动的工作量。

CSS 中层叠的意思是多种样式可应用于一个单一的 HTML 页或应用于在该页中一个单一的 HTML 标记,当多种样式应用于同一 HTML 标记时,浏览器便会根据 CSS 标准中定义的层叠规则来决定哪一种样式优先。不同类型的样式设置将共同应用,而相同类型的样式设置将根据优先顺序覆盖,这就是层叠样式表的由来。在网页制作时采用 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,就可以改变同一页面的不同部分或者页数不同的网页的外观和格式。

3.7.2 CSS 的定义

CSS 是一种格式化网页的标准方法,它就颜色、字体、间隔、定位以及边距等格式提供了几十种属性,这些属性可通过 style 应用于 HTML 标记中。

一个样式表是由许多样式规则组成的,用来控制网页元素的显示方式。其规则的形式为:

```
选择符{属性 1:值 1; 属性 2:值 2; …}
```

规则由选择符以及紧跟其后的一系列“属性:值”对组成,所有“属性:值”对用“{}”将其包括在内,各“属性:值”对之间用分号“;”分隔。如 p{color:red; font-size:20pt}就是一个规则,其中 p 是选择符,color:red、font-size:20pt 是“属性:值”对。本规则表示所有<p>标记

中的文字颜色为红色、大小为 20 磅。如果属性的值由多个单词组成，则必须在值上加引号，如字体的名称经常是几个单词的组合：p {font-family:"sans serif"}。

选择符是指要引用样式的对象，可以是一个或多个 HTML 标记(各标记之间用逗号分开)，也可以是类选择符(如. text)、id 选择符或上下文选择符。如果使用 HTML 中的标记名，称此规则为重置定义，也就是说重新定义了某些标记的显示样式(如标记 p 的默认文字颜色为黑色，在上面的规则中重新定义为红色，那么以后< p >中的文字就显示为红色)。可以把相同属性和值的选择符组合起来书写，用逗号将选择符分开，这样可以减少样式重复定义。如 p,table{font-size:9pt}效果完全等效于：

```
p{font-size:9pt}
table{font-size:9pt}
```

对字体而言，9pt 相当于 Word 中小 5 号字体，10pt 相当于 Word 中 5 号字体。CSS 中常用描述长度的单位有 cm(厘米)、in(英寸)、mm(毫米)、pc(picas，1pc＝12pt)、pt(点，1pt＝1/72in)、px(像素)、%(百分比，例如 150%表示 1.5 倍)、em(当前字体中字母 M 的宽度)、ex(当前字体中字母 X 的高度)。

样式定义中还可以加入注释来说明代码的意思，注释有利于自己或别人以后编辑和更改代码时理解代码的含义。在浏览器中，注释是不显示的。格式为：

```
/*字符串*/
```

【例 3-7】 CSS 定义。

```
<html>
<head>
<title>CSS 示例</title>
<meta http-equiv="Content-Type" content="text/html;charset=utf-8">
<!-- 样式定义开始-->
<style type="text/css">
h1{font-family:"隶书","宋体";color:# ff8800}
.text{font-family:"宋体";font- size:14pt;color:red}
</style>
<!-- 样式定义结束-->
</head>
<body topmargin=4>
<h1>一个 CSS 示例！ </h1>
<span class="text">这行文字应是红色的。</span>
</body>
</html>
```

在浏览器中显示结果如图 3-15 所示。

图 3-15　一个 CSS 示例

3.7.3 CSS 中的选择符

CSS 中有六种选择符,分别是 HTML 标记、具有上下文关系的 HTML 标记、用户定义的类选择符、用户定义的 ID 选择符、虚类、虚元素,分别说明如下。

1. HTML 标记类选择符

直接用 HTML 标记或 HTML 元素名称作为 CSS 选择符,例如:

```
td,input,select,body {font-family:Verdana;font-size:12px;}
form,body {margin:0;padding:0;}
select,body,textarea {background:#fff;font-size:12px;}
select {font-size:13px;}
textarea {width:540px;border:1px solid #718da6;}
img {border:none}
a {text-decoration:underline;cursor:pointer;}
h1 {color: #ff0000}
```

2. 具有上下文关系的 HTML 标记类选择符

如果定义了这样的样式规则 body {color: blue},则网页中所有的文字都以蓝色显示,除非另外指定样式或格式来更改这一设定,这是因为 body 标记包含了所有其他的标记符和内容。如果要为位于某个元素内的元素设置特定的样式规则,则应将选择符指定为具有上下文关系的 HTML 标记。例如,如果只想使位于 h1 标记符的 b 标记符具有特定的属性,应使用的格式为:

```
h1 b {color: blue} <!-- 注意:元素之间以空格分隔-->
```

这表示只有位于 h1 标记内的 b 元素具有蓝色属性,其他任何 b 元素保持原有颜色。

【例 3-8】 该例中,包含在 div 内有一个 input,div 和 input 之间就有上下文关系;div 内有 span,span 内有 B,它们之间也构成了上下文关系。实际上这种上下文关系可以嵌套任意层次。用 div span b {color:yellow}表示了 div 标记中的 span 标记中的 b 元素用黄颜色显示。

```
<html><head><title>CSS 选择符问题 </title>
<style type=text/css>
    input {color: white;}
    div input {color:red}
    div span b {color:yellow}
</style>
</head><body>
    <input type="text" value="change me"><br>
<div>
      <input type="text" value="change me"><br>
      <span>I'm a <b>good </b>  student </span>
</div>
</body></html>
```

3. 用户定义的类选择符

使用类选择符能够把相同的标记分类定义成不同的样式。定义类选择符时,在自定义类的名称前面加一个点号。假如想要两个不同的段落,一个段落向右对齐,一个段落居中,可以先定义两个类:

```
p.right {text-align: right}
p.center {text-align: center}
```

“.”符号后面的 right 和 center 为类名。类的名称可以是用户自定义的任意英文单词或以英文开头的与数字的组合，一般以其功能和效果简要命名。如果要用在不同的段落中，只要在 HTML 标记中加入前面定义的类：

```
<p class="right">这个段落向右对齐的</p>
<p class="center">这个段落是居中排列的</p>
```

用户定义的类选择符的一般格式为：

```
selector.classname { property: value; …}
```

类选择符还有一种用法，在选择符中省略 HTML 标记名，这样可以把几个不同的元素定义成相同的样式。例如，.center {text-align：center}，自定义 center 类选择符为文字居中排列。这样不限定某一个 HTML 标记，可以被应用到任何元素上。下面将 h1 元素（标题1）和 p 元素（段落）都设为 center 类，使这两个元素的文字居中显示：

```
<h1 class="center">这个标题是居中排列的   </h1>
<p class="center">   这个段落也是居中排列的 </p>
```

这种省略 HTML 标记的类选择符是最常用的 CSS 方法，使用这种方法，可以很方便地在任意元素上套用预先定义好的类样式，但是要注意，前面的“.”号不能省略。

【例 3-9】 CSS 类选择符示例。

```
<html><head><title>CSS 类选择符示例</title>
<style type="text/css">
<!--
p.English{font-family:"Times New Roman";color:blue}
p.Chinese{font-family:"幼圆";font-size:20pt}
.phone{color:blue}
-->
</style></head>
<body>
<p class=English>www.cqu.edu.cn</p>
<p class=Chinese>重庆大学</p>
<p class=phone>20824812 </p>
</body>
</html>
```

在浏览器中显示结果如图 3-16 所示。

4. 用户定义的 ID 选择符

用户定义的 ID 选择符的一般格式为：

```
#IDname { property: value; …}
```

其中，IDname 为某个标记 ID 属性值。ID 选择符的用途及概念和类选择符相似，不同之处在于同一个 ID 选择符样式只能在 HTML 文件内被应用一次，而类选择符样式则可以被多次应用。也就是说，如果有些较特别的标记需要应用较为特殊的样式，则建议使用 ID 选择符。在定义 ID 选择符时以“#”号开头而不是“.”号。

【例 3-10】 ID 选择符示例。

```
<html>
<head><title>CSS ID 使用示例</title>
<style type="text/css">
<!--
  #English {font-family:"Times New Roman";color:blue}
```

```
    #Chinese {font-family:"幼圆"}
-->
</style></head>
<body>
<h1 id="English">www.cqu.edu.cn</h1>
<h2 id="H2_1">www.cqu.edu.cn</h2>
<h1 id="Chinese">重庆大学</h1>
<h2 id="H2_2">重庆大学</h2>
</body></html>
```

在浏览器中显示结果如图 3-17 所示。

图 3-16　类选择符示例

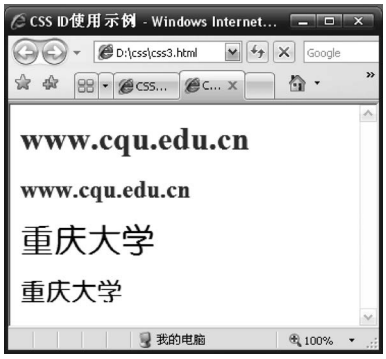

图 3-17　ID 选择符示例

例 3-10 中 HTML 标记若引用相同的 ID 属性，则会给用脚本语言操控该元素带来麻烦。同一个 #English 选择符同时应用到< h1 >及< h2 >标记，可以正常显示，这是因为浏览器在 ID 选择符这部分并没有完全遵循 W3C 的规格建议书，结果 ID 选择符和类选择符的用法几乎相同。

5. 虚类

虚类可以看作一种特殊的类选择符，是能被支持 CSS 的浏览器自动识别的特殊选择符。虚类主要针对超链接 a 标记符，可以为超链接定义不同状态下的不同样式效果。

虚类的形式如下：

```
选择符:虚类 { property: value; …}
```

定义虚类的方法和常规类很相似，但有两点不同：一是连接符是冒号而不是句点号；二是它们有预先定义好的名称，不能随便取名。这里仅以虚类中最常见的 a 虚类为例，a 虚类可以指定超链接标记 a 以不同的方式显示链接，表示链接处在以下四种不同的状态下：link、visited、active、hover(未访问的链接、已访问的链接、活动链接(即单击链接后)、鼠标指针停留在链接上)。可以把它们分别定义为不同的效果，如例 3-11 所示。

【例 3-11】 CSS 虚类示例。

```
<html><head><title>CSS 虚类示例</title>
<style type="text/css">
a:link {font-size: 18pt; font-family:隶书; text-decoration:none}
                                                        /* 未访问的链接 */
a:visited {font-size: 18pt;font-family:宋体;text-decoration:line-through}
                                                        /* 已访问的链接 */
```

```
a:hover {font-size: 18pt; font-family:黑体;text-decoration:overline}
                                                    /* 鼠标指针停留在链接上 */
a:active {font-size:18pt; font-family:幼圆; text-decoration:underline}
/* 活动链接 */
</style></head>
<body>
<a href="#">软件学院</a>
</body></html>
```

在浏览器中显示结果如图 3-18 所示。

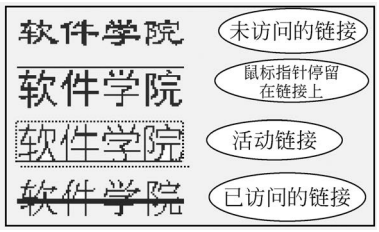

图 3-18　虚类示例

在例 3-11 中，这个链接未访问时的字体是隶书，无下画线，访问后是宋体，打上了删除线，单击链接后变成活动链接时字体为幼圆并有下画线，鼠标指针在链接上时为黑体，有上画线。

值得注意的是，有时这个链接访问前鼠标指针指向链接时有效果，而链接访问后鼠标指针再次指向链接时却无效果了。这是因为把 a:hover 放在了 a:visited 的前面，这样由于后面的优先级高，当访问链接后就忽略了 a:hover 的效果。所以根据层叠顺序，在定义这些链接样式时，一定要按照 a:link、a:visited、a:hover、a:active 的顺序书写。

还可以将虚类和类选择符及其他选择符组合起来使用，其形式如下：

```
选择符.类:虚类 { property: value; …}
```

这样就可以在同一个页面中做几组不同的链接效果了。

【例 3-12】 CSS 虚类和类选择符组合使用。可以定义一组链接为红色、访问后为蓝色，另一组为绿色、访问后为黄色。

```
<html><head>
<style type="text/css">
a.myred:link {color: #FF0000; text-decoration: underline:}
a.myred:visited {color: #0000FF; text-decoration: none:}
a.myblue:link {color: #00FF00; text-decoration: none; }
a.myblue:visited {color: #FF00FF; text-decoration: underline;}
</style></head>
<body>
<a class="myred" href="#">这是第一组链接</a>
<a class="myblue" href="#">这是第二组链接</a>
</body></html>
```

6. 虚元素

在 CSS 中有两个特殊的选择符，用于 p、div、span 等块级元素的首字母和首行效果，它们是 first-letter 和 first-line。有些浏览器不支持这两个虚元素。其格式为：

```
选择符: first-letter { property: value; …}
选择符: first-line { property: value; …}
选择符.类: first-letter { property: value; …}
选择符.类: first-line { property: value; …}
```

【例 3-13】 CSS 虚元素示例。显示效果如图 3-19 所示。

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

```
p:first-letter {font-size:200%;float:left;}
p:first-line {color:blue;}
div.odd:first-letter {font-family: 黑体;font-size:30pt;}
</style></head>
<body><p>
在 CSS 中有两个特殊的选择符,<br>用于文本段的首字母和首行效果<br>
有些浏览器不支持这两个虚元素</p>
<Div class="odd">
前面我们了解了 CSS 的语法,但要想在浏览器中显示出效果,<br>
就要让浏览器识别并调用样式表,当浏览器读取样式表时,</Div>
</body></html>
```

首字母效果也可在样式中定义 span {font-size：200%},然后在需要的位置应用该样式,例如“< p >< span >前面我们了解了 CSS 的语法</p >”可以达到图 3-19 中的效果。

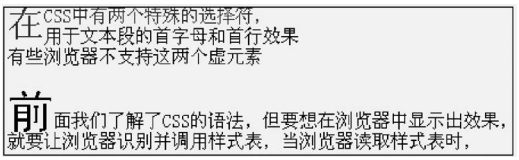

图 3-19　CSS 虚元素示例效果

3.7.4　CSS 的使用方法

前面介绍了 CSS 的语法,但要在浏览器中显示出效果,就要让浏览器识别并调用样式表。当浏览器读取样式表时,要依照文本格式来读。为网页添加样式表的方法有四种：链入外部样式表、导入外部样式表、联入样式表和内联样式。其中,联入样式表和内联样式是将 CSS 的功能组合于 HTML 文件之内,而链接及导入外部样式表则是将 CSS 功能以文件方式独立于 HTML 文件之外,然后通过链接或导入的方式将 HTML 文件和 CSS 文件链接在一起。

1. 链入外部样式表

链入外部样式表是把样式表保存为一个 CSS 文件,在 HTML 的头信息标识符< head >中添加< link >标记链接到这个 CSS 文件即可使用。

【例 3-14】 链入外部样式表示例。

先将样式定义存放在文件 mystyle.css 中,包含如下内容：

```
h1{font-family:"隶书","宋体";color:# ff8800}
p{background-color:yellow;color:# 000000}
.mytext{font-family:"宋体";font-size:14pt;color:red}
```

css5.htm 引用该样式表：

```
<html>
<head><title>链接样式表 CSS 示例</title>
<link rel="stylesheet" type="text/css" href=mystyle.css" media=screen></head>
<body topmargin=4>
<h1>这是一个链接样式表 CSS 示例!</h1>
<span class="mytext">这行文字应是红色的。</span>
<p>这一段底色应是黄色。</p>
</body></html>
```

在浏览器中显示结果如图 3-20 所示。

图 3-20　链接样式表示例

外部样式表不能含有任何像< head >或< style >这样的 HTML 标记，其仅仅由样式规则或声明组成，并且只能以 CSS 为扩展名。

< link >标记放置在 HTML 文档头部，其属性主要有 rel、href、type、media。

rel 属性表示样式表将以何种方式与 HTML 文档结合。一般取值为 stylesheet，指定一个外部的样式表。href 属性指出 CSS 文件的地址，如果样式文件和 HTML 文件不是放在同一路径下，则要在 href 里加上完整路径。type 属性指出样式类别，通常取值为 text/css。media 是可选的属性，表示使用样式表的网页将用什么媒体输出，取值范围：screen(默认)，输出到计算机屏幕；print，输出到打印机；projection，输出到投影仪等。

一个外部样式表文件可以应用于多个页面。当改变这个样式表文件时，所有页面的样式都随之而改变。在制作大量相同样式页面的网站时，非常有用，不仅减少了重复的工作量，而且有利于以后的修改、编辑。同时，大多数浏览器会保存外部样式表在缓冲区，从而浏览时也减少了重复下载代码，避免了在展示网页时的延迟。

2. 导入外部样式表

导入外部样式表是指在 HTML 文件头部的< style > …</ style >标记之间，利用 CSS 的@import 声明引入外部样式表。

【例 3-15】 导入外部样式表示例。

```
<html>
<head><title>导入外部样式表 CSS 示例</title>
<style type="text/css">
      <!-- @import  "mystyle.css";-->
h2 {font-family:"隶书","宋体";color:blue}
</style>
<head>
<body topmargin=4>
<h1>这是一个导入外部样式表 CSS 示例！ </h1>
<span class="text">这行文字应是红色的。</span>
<h2>这行文字应是蓝色的。</h2>
<p>这一段底色应是黄色。</p>
</body></html>
```

图 3-21　导入外部样式表示例

在浏览器中显示结果如图 3-21 所示。

@import "mystyle. css"表示导入 mystyle. css 样式表，注意使用时外部样式表的路径。方法和链入样式表的方法很相似，但导入外部样式表输入方式更有优势。因为除了导入外部样式外，还可添加本页面的其他样式。注意，@import 声明必须放在样式表的开始部分，其他的 CSS 规则应仍然包括在 style 元素中。

3. 联入样式表

利用< style >标记将样式表联入 HTML 文件的头部。前面的例子大都采用这种方法。

【例 3-16】 联入样式表示例。

```
<head>
<style type="text/css">
<!--
hr {color: sienna}
p {margin-left: 20px}
body {background-image: url("images/back40.gif")}
-->
</style>
</head>
```

style 元素放在文档的 head 部分。必需的 type 属性用于指出样式类别,通常取值为 text/css。有些低版本的浏览器不能识别 style 标记,这意味着低版本的浏览器会忽略 style 标记中的内容,直接以源代码的方式在网页上显示设置的样式表。为了避免这样的情况发生,用加 HTML 注释的方式(<!--注释 -->)隐藏内容而不让它显示,像上述例子那样。联入样式表的作用范围是本 HTML 文件。

4. 内联样式

内联样式是混合在 HTML 标记中使用的,用这种方法,可以很简单地对某个元素单独定义样式。这是连接样式和 HTML 标记的最简单的方法。内联样式的使用是直接在 HTML 标记中加入 style 参数。而 style 参数的内容就是 CSS 的属性和值。

【例 3-17】 内联样式表示例。

```
<h1 style="font-family: '隶书', '宋体';color:# ff8800>这是一个 CSS 示例!</h1>
<p style="color:red;background-color:yellow" >这一段底色应是黄色。</p>
<body style="font-family: '宋体';font-size:14pt" >
```

此时,样式定义的作用范围仅限于此标记范围内。style 属性是随 CSS 扩展出来的。它可以应用于任意 body 元素(包括 body 本身),除了 basefont、param 和 script。还应注意,若要在一个 HTML 文件中使用内联样式,可以在文件头部对整个文档进行单独的样式表语言声明,即

```
<META HTTP-EQUIV="CONTENT-TYPE" CONTENT="TEXT/CSS">
```

内联样式会向标记中添加更多属性及内容,因此对于网页设计者来说很难维护,更难阅读。而且由于它只对局部起作用,因此必须对所有需要的标签都进行设置,这样就失去了 CSS 在控制页面布局方面的优势。所以,内联样式主要用于样式仅适用于单个页面的情况,应尽量减少使用内联样式,而采用其他样式。

5. 多重样式表的叠加

如果在同一个选择器上使用几个不同的样式表时,这个属性值将会叠加几个样式表,遇到冲突的地方会以最后定义的为准。下面是一个多重样式表的叠加示例。

首先链入一个外部样式表,其中定义了 h3 选择符的 color、text-align 和 font-size 属性:

```
h3 {color: red;text-align: left;font-size: 8pt;}
```

然后在内部样式表中也定义了 h3 选择符的 text-align 和 font-size 属性:

```
h3 {text-align: right; font-size: 20pt;}
```

那么这个页面叠加后的样式为：

```
color: red; text-align: right; font-size: 20pt;
```

即标题3的文字颜色为红色，向右对齐，尺寸为20号字。字体颜色从外部样式表中保留下来，而对齐方式和字体尺寸都有定义时，按照后定义优先的原则。

因此，依照后定义优先的原则，优先级最高的是内联样式，联入样式表高于导入外部样式表，链入的外部样式表和联入的样式表之间，最后定义的样式优先级高。

3.8 用CSS控制Web元素的显示外观

CSS可以通过字体属性、颜色及背景属性、文本属性、方框属性、分类属性和定位属性等方式控制Web元素的显示外观。在Web开发中可以通过很多工具来自动生成样式表，不需要记忆复杂的样式名称，但必须对样式名称有所了解，只有这样才可以在自动生成的样式单上进行修改。

视频讲解

1. 字体属性

字体属性包括字体(font-family)、字号(font-size)、字体风格(font-style)、字体加粗(font-weight)、字体变化(font-variant)及字体综合设置(font)属性。

2. 文本属性

文本属性设置文字之间的显示特性，包括字符间隔(letter-spacing)、文本修饰(text-decoration)、大小写转换(text-transform)、文本横向排列(text-align)、文本纵向排列(vertical-align)、文本缩排(text-indent)、行高(line-height)。

1) 字符间隔

设定字符之间的间距。其基本格式为：

```
letter-spacing: 参数
```

2) 文本修饰

文字修饰的主要用途是改变浏览器显示文字链接时的下画线。

基本格式如下：

```
text-decoration: 参数
```

参数取值：underline，为文字加下画线；overline，为文字加上画线；line-through，为文字加删除线；blink，使文字闪烁；none，不显示上述任何效果。

【例3-18】 用CSS控制文本显示示例。

```
<html>
<head><title>文本修饰</title>
<style type="text/css">
<!--
.style1{text-decoration: underline overline line-through blink}
-->
</style></head>
<body><span class="style1">给文字加上画线、下画线、删除线和闪烁。</span>
</body></html>
```

在浏览器中显示结果如图3-22所示。

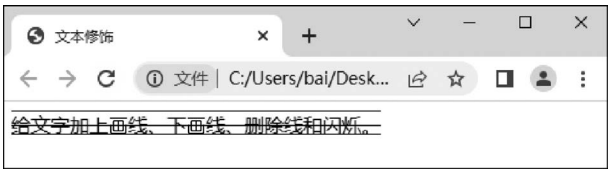

图 3-22　用 CSS 控制文本显示示例

3）大小写转换

文字大小写转换使网页的设计者不用在输入文字时就完成文字的大小写，而可以在 输入完毕后，再根据需要对局部的文字设置大小写。

基本格式如下：

```
text-transform: 参数
```

参数取值：uppercase，所有文字大写显示；lowercase，所有文字小写显示；capitalize，每个单词的头字母大写显示；none，不继承母体的文字变形参数。

注意，继承是指 HTML 的标识符对于包含自己的标识符的参数会继承下来。

4）文本横向排列

文本的水平对齐可以控制文本的水平对齐，而且并不仅仅指文字内容，也包括设置图片、影像资料的对齐方式。

基本格式如下：

```
text-align: 参数
```

参数的取值：left，左对齐；right，右对齐；center，居中对齐；justify，相对左右对齐。

但需要注意的是，text-align 是块级属性，只能用于< p >、< blockquote >、< ul >、< h1 >～< h7 >等标识符中。

5）文本纵向排列

文本的垂直对齐是相对于文本母体的位置而言的，不是指文本在网页中垂直对齐。例如，在表格的单元格中一段文本设置为垂直居中，文本将在单元格的正中显示，而不是整个网页的正中。

基本格式如下：

```
vertical-align: 参数
```

参数取值：top，顶对齐；bottom，底对齐；text-top，相对文本顶对齐；text-bottom，相对文本底对齐；baseline，基准线对齐；middle，中心对齐；sub，以下标的形式显示；super，以上标的形式显示。

6）文本缩排

文本缩进可以使文本在相对默认值较窄的区域中显示，主要用于中文版式的首行缩进，或是为大段的引用文本和备注做成缩进的格式。

基本格式如下：

```
text-indent: 缩进距离
```

缩进距离取值：带长度单位的数字；比例关系。

需要注意的是，text-indent 是块级属性，只能用于< p >、< blockquote >、< ul >、< h1 >～

<h7>等标识符中。

7）行高

行高是指上下两行基准线之间的垂直距离。

基本格式如下：

```
line-height: 行间距离
```

行间距离取值：①不带单位的数字，以1为基数，相当于比例关系的100%。②带长度单位的数字，以具体的单位为准。③比例关系。

注意，如果文字字体很大，而行距相对较小，可能会发生上下两行文字互相重叠的现象。

3. 控制颜色和背景属性

控制颜色和背景属性包括颜色（color）、背景颜色（background-color）、背景图片（background-image）、背景图片重复（background-repeat）、背景图片固定（background-attachment）、背景图片定位（background-position）六部分。

1）颜色

基本格式如下：

```
color: 参数
```

2）背景颜色

基本格式如下：

```
background-color: 参数
```

其参数取值和颜色属性一样。

3）背景图片

基本格式如下：

```
background-image: url(URL)
```

URL就是背景图片的存放路径。如果用none来代替背景图片的存放路径，将什么也不显示。

4）背景图片重复

背景图片重复控制的是背景图片平铺与否，也就是说，结合背景定位的控制可以在网页上的某处单独显示一幅背景图片。

基本格式如下：

```
background-repeat: 参数
```

参数取值：no-repeat，不重复平铺背景图片；repeat-x，使图片只在水平方向上平铺；repeat-y，使图片只在垂直方向上平铺。

如果不指定背景图片重复属性，则浏览器默认的是背景图片在水平、垂直两个方向上平铺。

5）背景图片固定

背景图片固定控制背景图片是否随网页的滚动而滚动。如果不设置背景图片固定属性，则浏览器默认背景图片随网页的滚动而滚动。为了避免过于花哨的背景图片在滚动时伤害浏览者的视力，可以解除背景图片和文字内容的捆绑，改为和浏览器窗口捆绑。

基本格式如下：

```
background-attachment: 参数
```

参数取值：fixed，网页滚动时，背景图片相对于浏览器的窗口而言固定不动；scroll，网页滚动时，背景图片相对于浏览器的窗口而言一起滚动。

6）背景图片定位

背景图片定位用于控制背景图片在网页中显示的位置。

基本格式如下：

```
background-position: 参数表
```

参数取值：①带长度单位的数字参数；top，相对前景对象顶对齐；bottom，相对前景对象底对齐；left，相对前景对象左对齐；right，相对前景对象右对齐；center，相对前景对象中心对齐。②比例关系。

参数中的 center 如果用于另外一个参数的前面，则表示水平居中；如果用于另外一个参数的后面，则表示垂直居中。

【例 3-19】 用 CSS 控制图片显示示例。

```
<! DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www. w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>无标题文档</title>
<style type="text/css">
<!--
.list{position:relative;}
.list span img{ /* CSS for enlarged image */
border-width: 0;
padding: 2px; width:300px;
}
.list span{
position: absolute;
padding: 3px;
border: 1px solid gray;
visibility: hidden;
background-color:# FFFFFF;
}
.list:hover{background-color:transparent;}
.list:hover span{visibility: visible;top:0; left:60px;}
-->
</style></head>
<body>
<p> </p>
<div><a class="list" href="# "><img src="img1.jpg" width="156" height="122"
border="0" /><span><img src="img1.jpg" alt="big" /></span></a></div>
</body></html>
```

在浏览器中显示结果如图 3-23 所示。

4. 方框属性

方框属性用于设置元素的边界(margin)、边界补白(padding)、边框(border)等属性值。使用方框属性的大多是块元素。

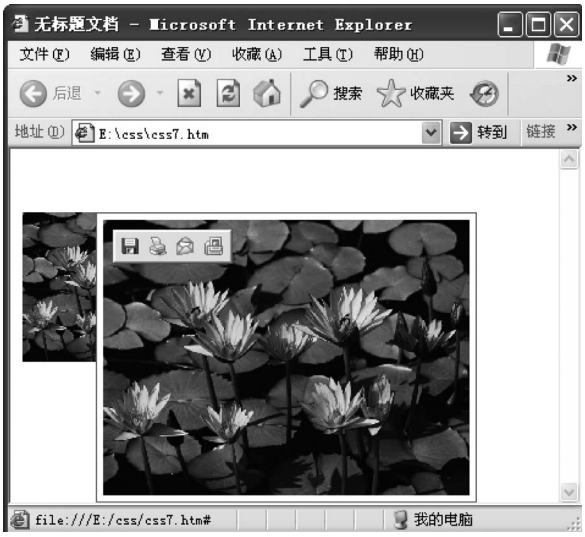

图 3-23　用 CSS 控制图片显示示例

5. 列表属性

列表属性用于设置列表标记(< ol >和< ul >)的显示特性,包括 list-style-type、list-style-image、list-style-position、list-style 等。

(1) list-style-type：表示项目符号。

具体格式如下：

```
list-style-type: 参数
```

参数取值：①无序列表值,disc——实心圆点；circle——空心圆；square——实心方形。②有序列表值,decimal——阿拉伯数字；lower-roman——小写罗马数字；upper-roman——大写罗马数字；lower-alpha——小写英文字母；upper-alpha——大写英文字母；none——不设定。

(2) list-style-image：使用图像作为项目符号。

具体格式如下：

```
list-style-image:url(URL)
```

(3) list-style-position：设定项目符号是否在文字里面与文字对齐。

具体格式如下：

```
list-style-position:outside/inside
```

(4) list-style：综合设置项目属性。

具体格式如下：

```
list-style:type,position
```

6. 定位属性

CSS 提供 position、top、left 和 z-index 属性用于在二维或三维空间定位某个元素相对于其他元素的相对位置或绝对位置。

(1) position 属性用于设置元素位置的模式,有以下三种取值。

① absolute：表示绝对定位,原点在所属窗口或框架页面文档显示区域的左上角。

② relative：表示相对位置，相对本元素的父元素的位置。

③ static(默认值)：表示静态位置，按照 HTML 文件中各元素的先后顺序显示。

(2) 当 position 为 absolute 时，top 和 left 属性分别用于设置元素与窗口或框架上端以及左端的距离；当 position 为 relative 时，top 和 left 属性分别用于设置元素与父元素上端以及左端的距离。例如，div 标记中有一个按钮，当 position 为 relative 时，按钮位置的 top 和 left 是以 div 标记左上角为坐标原点。

(3) 在三维空间中定位元素时，与之相关的属性是 z-index。它将页面中的元素分成多个“层”，形成“堆叠”的三维效果。z-index 的取值为整数，可为正，也可为负。值较高的元素覆盖较低的元素。

【例 3-20】 定位属性示例：立体效果的“三维空间定位”和阴影效果的“Web 开发技术”。

```
<html><head><title>三维空间定位</title>
<style type="text/css">
span{font-size:18pt}
span.level2{position:absolute;z-index:2;left:100;top:100;color:red;}
span.level1{position:absolute;z-index:1;left:101;top:101;color:green;}
span.level0{position:absolute;z-index:0;left:102;top:102;color:yellow;}
p.lev1{position:absolute;top:200;left:220;z-index:2;font-size:36pt;color:
blue;}
p.lev2{position:absolute;top:220;left:250;z-index:-2;font-size:30pt;color:
darked;}
</style></head>
<body>
<span class="level2">三维空间定位</span>
<span class="level1">三维空间定位</span>
<span class="level0">三维空间定位</span>
<p class="lev1">Web 开发技术</p>
<p class="lev2">Web 开发技术</p>
</body></html>
```

在浏览器中显示结果如图 3-24 所示。

图 3-24 定位属性示例

7. 控制鼠标指针形状

在网页上，鼠标指针平时呈箭头形，指向链接时成为手形，等待网页下载时成为圆圈转动形（程序正忙）……虽然这样的设计能使我们知道浏览器现在的状态或是可以做什么，但这些还不能完全满足我们的需要。以链接为列，可以是指向一个帮助文件，也可以是向前进一页或是向后退一页，针对如此多的功能，光靠千篇一律的手形鼠标是不能说明问题的。CSS 提供了十多种鼠标指针形状供选择，如表 3-4 所示。

表 3-4　鼠标指针形状的样式定义

基本格式（cursor）	鼠标指针形状参数	基本格式（cursor）	鼠标指针形状参数
style="cursor:pointer"	手形	style="cursor:crosshair"	十字形
style="cursor:wait"	圆圈转动形	style="cursor:move"	十字箭头形
style="cursor:e-resize"	右箭头形	style="cursor:n-resize"	上箭头形
style="cursor:text"	文本形	style="cursor:w-resize"	左箭头形
style="cursor:help"	问号形	style="cursor:s-resize"	下箭头形
style="cursor:nw-resize"	左上箭头形	style="cursor:ne-resize"	右上箭头形
style="cursor:sw-resize"	左下箭头形	style="cursor:se-resize"	右下箭头形

8. CSS 中的 float、clear、display 及 visibility 属性

CSS 中的 float（浮动）、clear（清除浮动）和 display 及 visibility（指定元素的显示方式）属性通常用于控制元素在页面中的布局和显示。综合使用这些属性，可以创建复杂的布局和设计，但要小心使用浮动，因为它可能引起一些问题，如清除浮动和产生不必要的间隙。在现代网页设计中，推荐使用 Flexbox 和 Grid 等新的布局技术。

1）float 属性

该样式属性用于指定元素在其包含块中的浮动方向，使其脱离顺序输出页面内容的文档流并沿着包含块的左侧或右侧浮动。常见值包括 none（默认值，不浮动）、left（左浮动）和 right（右浮动）等，其功能类似于在 Word 中插入图片后的布局选项设置，即浮于文字上方、嵌入型、四周型、紧密型等。例如：

```
<div style="border-style:solid; border-width:2px; float:right">此文本显示在右侧</div>
```

浮动的元素会影响其周围的布局，可能导致其他元素环绕在其周围。为了避免出现这种情况，可以使用 clear 属性。

2）clear 属性

该属性用于指定元素周围不允许有浮动元素的一侧。它防止浮动元素的影响，确保在指定方向上没有浮动元素。常见值包括 left（在左侧不允许浮动元素）、right（在右侧不允许浮动元素）、both（在左右两侧均不允许浮动元素）、none（默认值，允许浮动元素出现在两侧）、inherit（规定从父元素继承 clear 属性的值）。

以下代码实现三个手写签名在右侧竖向排列，若去掉样式，则三个签名会呈现在一行。

```
<html><head>
<style type="text/css">
img { float:right;  clear:both; }
</style>
</head>
```

```
<body>
<img src="手写签名1.jpg" width="238" height="95"/>
<img src="手写签名2.jpg" width="238" height="95"/>
<img src="手写签名3.jpg" width="238" height="95"/>
</body></html>
```

3) display 及 visibility 属性

两者都用于指定元素的显示方式。display 属性常见的值包括 none(不显示元素)、block(块级元素)、inline(行内元素)、inline-block(行内块级元素)。visibility 属性常见的值包括 visible(默认值,在页面上显示元素)、hidden(在页面上隐藏元素)、collapse(仅针对表格有效,隐藏表格的行和列,若用于其他元素,会呈现为 hidden)。

样式 display: none 和 visibility: hidden 都用于在页面中隐藏元素,但前者会将元素完全从文档流中移除,不占据任何空间,对布局没有影响,用户无法与其交互,也无法通过脚本访问,而后者元素仍然占据文档流中的空间,但它变得不可见,对布局有影响,即使元素不可见,但它仍会占据相应的空间,可以通过脚本访问,并与其交互。

display 属性常用于改变元素的默认显示方式,从而实现更灵活的页面布局,例如将行内元素变为块级元素或相反。

display: block 用于指定元素的显示方式为块级元素。块级元素会在页面上独占一行,并且会在其前后创建新的行,形成一个块。display: inline 用于指定元素的显示方式为行内元素。行内元素不会在页面上独占一行,而是在同一行上排列,不会强制换行。display: inline-block 结合了行内元素和块级元素的特性。它使元素在同一行内水平排列,同时保留块级元素的特性,可以设置宽度、高度以及垂直方向的一些属性。行内块级元素具有块级元素的特性,但在水平方向上仍保留行内元素的特性。

visibility: collapse 通常用于隐藏表格的行。以下代码就隐藏了该表格的行:

```
<tr style="visibility: collapse;"><td>John Doe</td><td>25</td><td>New York
</td></tr>
```

一个 3 行 3 列的表格通过合理设置该属性就会显示为 1 行 3 列(但通过脚本代码访问还会是 3 行)。例如,把属性改为 hidden,表格还是 3 行 3 列,但数据被隐藏,不显示在表格中。若要隐藏表格的某列,对该列的所有单元格建议用 display: none 样式属性。

下面代码中可改变 display 的不同属性值,在浏览器中可查看其实际效果。

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Display Inline-Block Example</title>
  <style>
    /* 将 div 元素设置为行内块级元素 */
    .inline-block-element {
      display: inline-block;
      width: 150px;
      height: 100px;
      background-color: #2ecc71;
      color: #fff;
      text-align: center;
      line-height: 100px;
      margin: 10px;
```

```
      }
    </style>
  </head>
  <body>
    <!-- 行内块级元素 -->
    <div class="inline-block-element">Inline Block1111</div>
    <div class="inline-block-element">Example1111</div>
    <!-- 其他内容 -->
    <p>This is some other content on the page.</p>
  </body>
  </html>
```

3.9 CSS3 介绍

视频讲解

CSS3 是 CSS 技术的升级版本，CSS3 语言开发是朝着模块化发展的。以前的规范作为一个模块实在是太庞大而且比较复杂，所以，把它分解为一些小的模块，更多的模块也被加入进来。这些模块包括盒子模型、列表模型、超链接方式、语言模块、背景和边框、文字特效、多栏布局等。

CSS3 具有以下特性。

(1) 圆角矩形。对应属性：border-radius。

(2) 以往对网页上的文字加特效职能用 filter 属性，在 CSS3 中专门制定了一个加文字特效的属性，而且不止加阴影一种特效效果。对应属性：font-effect。

(3) 丰富了对链接下画线的样式，以往的下画线都是直线，在 CSS3 中出现了波浪线、点线、虚线等，更可对下画线的颜色和位置进行任意改变。还有对应上画线和中横线的样式，效果与下画线类似。对应属性：text-underline-style、text-underline-color、text-underline-mode、text-underline-position。

(4) 在文字下加几个点或打个圈以示重点，CSS3 也开始加入了这项功能，在显示某些特定网页上很有用。对应属性：font-emphasize-style 和 font-emphasize-position。

还有一些对边框、背景、文字效果、颜色等进行修改的新属性出现，进一步丰富和发展了 CSS 的功能。

CSS3 的原理与 CSS 相同，是在网页中自定义样式表的选择符，然后在网页中大量引用这些选择符。CSS3 将完全向后兼容，所以没有必要修改现在的设计来让它们继续运作。网络浏览器也还将继续支持 CSS2。CSS3 主要的影响是将可以使用新的可用的选择器和属性，这些会允许实现新的设计效果，而且可以很简单地实现出所需要的设计效果。

CSS3 新增属性比较多，下面介绍一些常用的属性。

1. 颜色属性

CSS3 颜色模块的引入，让制作 Web 效果时不再局限于 RGB 和十六进制两种模式。CSS3 增加了 RGBA、HSL、HSLA 几种新的颜色模式。RGBA 是 CSS3 中定义颜色的函数，语法为 rgba(R,G,B,A)，代表由红(R)、绿(G)、蓝(B)和透明度(A)的变化以及相互叠加来得到各种各样的颜色。参数 R、G、B 的取值范围为 0～255，A 的取值范围为 0～1。

2. 圆角矩形

一个矩形有四个角，可分别定义四个角的圆角半径，其样式大都用于 div 标记。每个矩

形的角还可定义为椭圆形圆角,也即每个角可定义水平半径和垂直半径。

“style="border-radius:20px;""表示四个角的圆角半径都是20px。

“style="border-radius:5px 6px 7px 8px;""表示从矩形左上角开始,顺时针方向四个角的圆角半径分别是5px、6px、7px、8px。

“style="border-top-left-radius:5px; border-top-right-radius:6px; border-bottom-right-radius:7px; border-bottom-left-radius:8px;""含义和上面相同,可以单独指定四个角的半径。

“style="border-radius:10px 20px 30px 40px/20px 30px 40px 10px;""表示从矩形左上角开始,顺时针方向四个角的水平半径分别是10px、20px、30px、40px,垂直半径分别是20px 30px 40px 10px,形成椭圆形圆角。

3. 背景属性

CSS3允许使用多个属性在一个元素上添加多层背景图片,如background-image、background-clip、background-size、background-repeat、background-break等。该属性的应用大大改善以往面对多层次设计需要多层布局的问题,方便Web前端开发者对页面背景进行设计,简化背景图片的维护成本。其主要属性如下。

(1) background-image。

在CSS3中,可以在一个标签元素中应用多个背景图片。第一个图片是定位在最上面的背景,后面的图片依次在其下面显示。

格式:

```
background-image: url(图片地址)[,url(图片地址) …]
```

(2) background-size。

background-size属性可以用来设置背景图片的大小,属性如下。

① background-size:contain,缩小背景图片使其适应标签元素。

② background-size:cover,让背景图片放大延伸到整个标签元素。

③ background-size:宽度 高度,标注背景图片缩放的尺寸,可以是具体数值,也可以是百分比,例如background-size:100px 100px、background-size:50% 90%。

(3) background-repeat。

CSS3为了解决背景图片被标签元素截取而显示不全的问题,对background-repeat进行修改,增加了以下两个属性值。

① background-repeat:space,图片以相同的间距平铺且填充整个标签元素。

② background-repeat:round,图片自动缩放直到适应且填充整个标签元素。

(4) background-break。

CSS3中标签元素能够分在不同区域,background-break属性能够控制背景图片在不同区域的显示,属性如下。

① background-break:continuous,默认值,忽视区域之间的间隔空隙。

② background-break:bounding-box,重新考虑区域之间的间隔。

③ background-break:each-box,对每一个独立的标签区域进行背景的重新划分。

4. 多列布局属性

CSS3多列布局属性可以不使用多个div标签就能实现多列布局,可以通过对应属性将一个简单的区块拆成多列,主要属性如下。

① column-width：列的宽度。

② column-count：列的数量。

③ column-gap：列之间的间隔。

④ column-rule：定义每列之间边框的宽度、样式和颜色。

⑤ column-span：设置跨列显示。取值：none，只在本列显示；all，横跨所有列。

⑥ column-fill：定义列的高度是否统一。取值：auto，各列高度随其内容的变化而变化；balance，各列的高度根据最高列进行统一。例如，以下代码实现的效果如图 3-25 所示。

```
<h2 style="column-span:all; text-align:center;">CSS3 的使用</h2>
<p style="text-indent:2em; column-count:3; column-gap:50px; column-rule:
4px outset blue;text-align:justify;">CSS3 渐变(gradients)可以让你在两个或多个指
定的颜色之间显示平稳的过渡。以前，你必须使用图像来实现这些效果。但是，通过使用 CSS3 渐
变(gradient)，你可以减少下载的时间和宽带的使用。此外，渐变效果的元素在放大时看起来效
果更好，因为渐变(gradient)是由浏览器生成的。CSS3 定义了两种类型的渐变(gradient)：线
性渐变(linear gradient)- 向下/向上/向左/向右/对角方向；径向渐变(radial gradient)
- 由它们的中心定义。线性渐变相关属性：background-image。</p>
```

图 3-25　定位属性示例

5. 弹性盒模型布局属性

弹性盒模型布局属性可以根据复杂的前端分辨率进行弹性布局，让开发者轻松实现页面中某一区块在水平、垂直方向的对齐。属性如下。

① flex-direction：决定主轴方向，即子标签排列的方向。取值有 row(从左往右)、row-reverse(从右往左)、column(垂直向下)、column-reverse(垂直向上)。

② flex-wrap：是否允许子项目换行。取值有 nowrap(默认不换行)、wrap(允许换行)、wrap-reverse(反向换行)。

③ justify-content：定义子标签在主轴(x 轴)上的对齐方式。取值有 flex-start(默认居左对齐)、flex-end(居右对齐)、center(居中对齐)、space-around(环绕对齐)、space-between(两端对齐)、space-evenly(等距离对齐)。

④ align-items：定义子标签在 y 轴上的对齐方式。取值有 stretch(当子标签不设置高度时，默认拉伸为最大可承受高度)、flex-start(与当前行中的 x 轴的顶端对齐)、flex-end(与

当前行中的 x 轴的底端对齐)、center(在当前行中居中对齐)。

⑤ align-content：所有的行在容器中的对齐方式。取值有 flex-start(排在容器顶端)、flex-end(排在容器底端)、center(排在容器中间)、space-around(多个行环绕对齐)、space-between(多个行上下两端对齐)、space-evenly(多个行等距离对齐)。

6. 变形属性

在 CSS 2.1 中,想要让某个元素变形需要借助 JavaScript 写大量的代码实现。在 CSS3 中加入了变形属性,该属性在 2D 或 3D 空间里操作元素的位置和形状来实现旋转、扭曲、缩放、位移等操作。2D 变形属性如下。

① transform：设置元素的变形方式,可以将元素旋转、缩放、移动、倾斜等。属性值：translate(x,y),水平方向移动 x,垂直方向移动 y；rotate(angle),元素顺时针旋转 angle 角度；scale(x,y),元素水平方向缩放比为 x,垂直方向缩放比为 y；skew(angleX,angleY),元素沿着 x 轴倾斜 angleX 角度,沿着 y 轴倾斜 angleY 角度；matrix(a,b,c,d,e,f),将所有 2D 变形函数(旋转、缩放、移动及倾斜)组合在一起。

② transform-origin：用于表示元素旋转的中心点,默认值为 50% 50%。

3D 变形属性如下。

① transform：设置元素的变形方式。与 2D 相比,新增属性值：rotateX(angle),元素沿着 x 轴旋转的角度；rotateY(angle),元素沿着 y 轴旋转的角度；rotateZ(angle),元素沿着 z 轴旋转的角度。

② transform-style：指定嵌套元素如何在 3D 空间中呈现。属性值：flat,默认值；表示所有子元素在 2D 平面呈现；preserve-3d,表示所有子元素在 3D 空间中呈现。

7. 过渡与动画属性

CSS3 可以使用过渡属性 transition 来实现某种元素在某段时间内的一些简单动画效果,使其更具流线性和平滑性。其主要有以下属性。

① transition-property：属性的名字,表示对哪个属性进行变化,属性之间用逗号隔开,所有属性则用 all 表示即可。

② transition-duration：变化持续的时间。

③ transition-timing-function：实现过渡的方式。取值：linear,匀速；ease,先慢后快再慢；ease-in,先慢后快；ease-out,先快后慢；ease-in-out,开头慢结尾慢,中间快。

④ transition-delay：过渡开始前等待的时间。例如：

```
transition:all 1s linear 0s
```

CSS3 可以使用动画属性 animation 实现更复杂的样式变化以及一些交互效果,不需要使用任何 Flash 或 JavaScript 脚本代码。其主要有以下属性。

① @keyframes：规定动画。

② animation：所有动画属性的简写属性,除了 animation-play-state 属性。

③ animation-name：规定@keyframes 动画名称。

④ animation-timing-function：规定动画的速度曲线,取值与 transition-timing-function 相同。

⑤ animation-delay：动画开始前等待的时间,取值可为负,如－2s 指动画跳过 2s 进入动画周期。

⑥ animation-iteration-count：规定动画播放次数，默认值为 1。

⑦ animation-direction：规定动画是否在下一周期逆向地播放，默认值为 normal。

⑧ animation-play-state：规定动画是否正在运行或暂停，默认值为 running。

⑨ animation-fill-mocd：规定对象动画时间之外的状态。

8. 阴影属性

CSS3 中的阴影属性可以分为 text-shadow（文本阴影）和 box-shadow（盒子阴影），该属性提供了一种新的跨浏览器的解决方案，让文本看起来更加醒目，也可以轻易地为任何元素添加盒子阴影。文本阴影的属性如下。

① w-shadow：水平方向的距离，必须有，可为负值。

② h-shadow：垂直方向的距离，必须有，可为负值。

③ blur：阴影的模糊程度，可选，不支持负值。

④ color：阴影的颜色。

⑤ 特殊效果属性：stroke，空心效果；outset，阳文效果；inset，阴文效果。

盒子阴影的属性和文本阴影的大致相同，有差异的属性如下。

① spread：阴影的大小，可为负值。

② inset：内阴影，阴影在元素里面。

关于 CSS3 的详细使用可参考 https://www.runoob.com/css3/css3-tutorial.html 或 http://www.w3school.com.cn/css/index.asp。

思考练习题

1. 简要说明什么是 HTML。

2. 创建一个页面，该页面由两段不同的文字组成，第一段文字全部是黑体，颜色为红色，字体大小为 4；第二段文字的第一个字大小为 5，颜色为蓝色，字体为隶书，其他文字全部是宋体，大小为 3，颜色为黑色。

3. 简要说明框架在网页布局中的作用。

4. 什么是 CSS？与传统的 HTML 文档相比较，使用 CSS 的 HTML 文档有什么优点？

5. 为网页添加样式表的方法有哪几种？它们之间有什么区别？

6. 创建一个包含 10 个页面左右的网站。要求所有页面中的标题、正文、图像和链接具有相同的样式，所有的样式定义都位于单独的样式文件中，所有 HTML 文件都使用链接的方法将样式文件链接到当前文件中。要求合理使用文字、图像并用表格进行适当排版，导航结构清晰，具有一定的站点风格。

7. 上网浏览，查看源代码，分析优秀网站是如何使用 CSS 技术的。