第 8 章　网页设计基础

本章学习目标

本章主要讲解 HTML 语言、级联式样式表 CSS 和 JavaScript 语言的基础知识。通过本章学习，读者应该掌握以下内容。

- HTML 语言的基础知识。
- 级联式样式表（CSS）。
- JavaScript 语言的基础知识。

8.1　HTML 语言

8.1.1　HTML 的结构

HTML（超文本标记语言）是一种描述文档结构的标注语言，HTML 使用一些约定的标记对 WWW 上的各种信息进行标注。当用户浏览 WWW 上的信息时，浏览器会自动解释这些标记的含义，并按照一定的格式显示这些被标记的文件。HTML 的优点是其跨平台性，即任何可以运行浏览器的计算机都能显示 HTML 文件，而且不管其操作系统是什么，浏览器中的显示结果都相同。

HTML 文件是标准的 ASCII 文件，是后缀名为.htm 或.html 的文件。HTML 文件看起来像是加入了许多被称为标签（tag）的特殊字符串的普通文本文件。从结构上讲，HTML 文件由元素（element）组成，组成 HTML 文件的元素有许多种，用于组织文件的内容和标明文件的输出格式。绝大多数元素都是“容器”，即有起始标记和结束标记。元素的起始标记叫做起始标签（start tag），元素的结束标记叫做结束标签（end tag），在起始标签和结束标签中间的部分是元素体。每一个元素都有名称和许多可选择的属性，元素的名称和属性都在起始标签内标明。下面是一个标准的 HTML 文件，其在浏览器中显示的结果如图 8-1 所示。

代码清单 w8-1.htm

```
<html>
    <head>
        <title>武汉工业学院</title>
    </head>
    <body    bgcolor= yellow>
        <p>这是一 HTML 的测试文件</p>
    </body>
</html>
```

图 8-1 HTML 文件在浏览器的显示结果

从代码清单 w8-1.htm 中可以看出，HTML 文件仅由一个 HTML 元素组成，即文件以<html>开始，以</html>结束，文件其余部分都是 HTML 的元素体。而 HTML 元素的元素体可以由头元素<head>…</head>、体元素<body>…</body>和一些注释组成；头元素和体元素的元素体可以由其他的元素、文本及注释组成。

在代码清单 w8-1.htm 中第 5 行是体元素的起始标签，标明体元素从此开始。因为所有的标签都具有相同的结构，所以通过仔细分析这个标签的各个部分，读者对标签的写法应该有一定的了解。其格式为：

<起始标签 属性名=属性值> 内容 <结束标签>

在 HTML 中有三个字符具有特殊的意义，即：“<”表示一个标签的开始、“>”表示一个标签的结束和“&”表示转义序列的开始。每个转义字符都以“&”开始，以分号“;”结束。元素名也叫标签名。需要注意的是：

（1）“<”和起始标签之间不能有空格；

（2）元素名称不区分大小写；

（3）一个元素可以有多个属性，属性及其属性值不区分大小写，且各个属性用空格分开。

HTML 文件中，有些元素只能出现在头元素中，而绝大多数元素只能出现在体元素中。在头元素中的元素表示的是该 HTML 文件的一般信息，比如文件名称，是否可检索等，这些元素书写的次序是无关紧要的，只表明该 HTML 有还是没有该属性。与此相反，出现在体元素中的元素次序是敏感的，改变元素在 HTML 文件中的次序会改变该 HTML 文件的输出形式。

8.1.2 构成网页的基本元素

下面介绍一下常用的有关 Web 页文本格式的标记。

1. <title>标记

<title>标记用来给网页命名，网页的名称写在<title>与</title>标记之间，显示在浏览器的标题栏中。例如，在图 8-1 中所示的浏览器页面中，其标题栏所显示的“武汉工业学院”，是在 HTML 文件中由“<title>武汉工业学院</title>”所定义的。

2. <hn>标记

<h1>…</h1>到<h6>…</h6>标题元素有 6 种，用于表示文章中的各级标题的字体大小。<h1>到<h6>表示字号从大到小。下面这个例子中分别使用了<h1>到<h6>的标记。其 HTML 文件如代码清单 w8-2.htm 所示，在浏览器中的显示结果如图 8-2 所示。

代码清单 w8-2.htm

<html>

```
    <head>
        <title>这是一个测试网页</title>
    </head>
    <body>
        <h1>标题测试</h1>
        <h2>标题测试</h2>
        <h3>标题测试</h3>
        <h4>标题测试</h4>
        <h5>标题测试</h5>
        <h6>标题测试</h6>
    </body>
</html>
```

图 8-2　<hn>标记

3. 预格式化文本标记<pre>

HTML 的输出是基于窗口的，因而 HTML 文件在输出时都是要重新排版的，即把文本上任何额外的字符（如空格、制表符和回车符）都忽略，若确实不需要重新排版的内容，可以用<pre>…</pre>通知浏览器。从图 8-3 和图 8-4 中可以看出有无预格式化文本标记<pre>的显示区别。

下面是图 8-3 和图 8-4 的 HTML 源文件。

代码清单 w8-3.htm

```
<html>
    <head>
        <title>这是一个测试网页</title>
    </head>
    <body>
        <pre>       <!--（图 8-4 无此标记）-->
                    HTML 是一种描述文档结构的标注语言，它使用一些
                约定的标记对各种信息进行标注。
                </pre>        <!--（图 8-4 无此标记）-->
    </body>
</html>
```

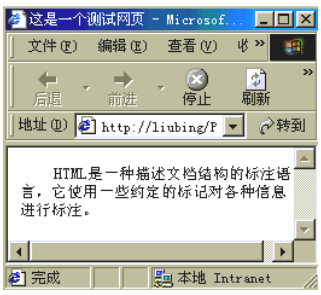

图 8-3　有<pre>标记

图 8-4　无<pre>标记

4.
和<p>标记

用于强制换行。<p>表示一个段落的开始。

5. <b>、<i>、<u>、和<s>标记

这几个标记都是用来修饰所包含文档的。<b>标记使文本加粗；<i>标记使文本倾斜；<u>标记给文本加下划线；<s>标记给文本加删除线；标记使文本字体加重。下面给出一个 html 源文件，其在浏览器中的显示结果如图 8-5 所示。

代码清单 w8-4.htm

```
<html>
    <head>
        <title>这是一个测试网页</title>
    </head>
    <body>
        <strong>html</strong>是一种
        <em>描述文档结构</em>的
        <u>标注语言</u>,
        <b>它使用</b>一些
        <i>约定的标记</i>对各种信息进行
        <s>标注<s>。
    </body>
</html>
```

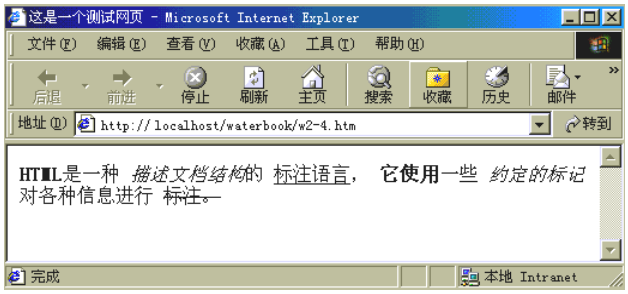

图 8-5　文档标记的修饰

6. 标记

…用来修改字体和颜色。其中 color 属性指定文字颜色，颜色的表示可以用 6 位十六进制代码，如<font　color=#00ff00>。另外，如果用户想要设置网页的背景色和文字

颜色，也可以将<body>标记扩充为：

<body bgcolor=#　text=#　link=#　alink=#　vlink=#　background="imageURL">

其中各个元素的说明如表 8-1 所示，表 8-2 列出了一些常用颜色的 RGB 值。

表 8-1　设置背景色和文字颜色

标记	说明
bgcolor	设置网页背景颜色
text	设置网页非可链接文字的颜色
link	设置网页中的可链接文字的颜色
alink	设置网页中的正被点击的可链接文字的颜色
vlink	设置网页中的已经点击的可链接文字的颜色
background	设置网页背景图案
imageURL	设置网页背景图案的 URL 地址
#	代表颜色 RGB 值（格式为 rrggbb）。它是用十六进制的红绿蓝（red-green-blue，RGB）值来表示

表 8-2　常见颜色 RGB 值

颜色	RGB 值	颜色	RGB 值
黑色（Black）	000000	橄榄色（Olive）	808000
红色（Red）	FF0000	深青色（Teal）	008080
绿色（Green）	00FF00	灰色（Gray）	808080
蓝色（Blue）	0000FF	深蓝色（Navy）	000080
白色（White）	FFFFFF	浅绿色（Lime）	00FF00
黄色（Yellow）	FFFF00	紫红色（Fuchsia）	FF00FF
银色（Silver）	C0C0C0	紫色（Purple）	800080
浅色（Aqua）	00FFFF	茶色（Maroon）	800000

例如，要将网页背景颜色设置为蓝色，可在 HTML 文件中把“body”标签的“bgcolor”属性设置为“#0000ff”，其写法如下所示：

<body　bgcolor=#0000ff>

8.1.3　超文本链接指针

超文本链接指针是 HTML 最吸引人的优点之一，可以这样说，如果没有超文本链接指针，就没有万维网。使用超文本链接指针可以使顺序存放的文件具有一定程度的随机访问能力，这更加符合人类的跳跃思维方式。超文本链接指针可把不连续的两段文字或两个文件联系起来。

1. 链接到其他站点

在 HTML 文件中，用链接指针指向一个目标。其语法格式为：

　zzz　

其中，zzz 可以是文字或图片，并显示在网页中，当用户单击这个指针时，浏览器就会显示由 href 属性中的统一资源定位器（URL）所指向的目标，实际上这个 zzz 在 HTML 文件中充当

指针的角色，一般显示为蓝色且带有下划线。href 中的 h 表示超文本，而 ref 表示访问或引用的意思。例如：

武汉工业学院

用户单击“武汉工业学院”，即可看到武汉工业学院的主页内容。在这个例子中，充当指针的是“武汉工业学院”。

在编写 HTML 文件时，需要知道目标的 URL。那么如何才能得到目标的 URL 呢？对于本地主机内的文件，其 URL 地址可以根据该文件的实际情况决定。对于 Internet 上的资源，在用浏览器查看时，其 URL 地址会在浏览器的状态栏中显示出来，把它抄下来写到新制作的 HTML 文件中即可。

在编写 HTML 文件时，对能确定关系的一组资源（比如在同一个目录中）应采用相对 URL，这不仅简化 HTML 文件的编写，而且便于维护。例如，当需要将某个目录整体移到另外一个地方，或把某一主机的资源移到另一台主机时，用相对 URL 写的 HTML 文件，对其中的 URL 不需要进行任何更改（只要它们的相对关系没有改变）。但如果用绝对 URL 编写 HTML 文件，就不得不修改每个链接指针中的 URL，这是一件很乏味也很容易出错的工作。对于各个资源之间没有固定的关系，所指向的目标分布在全球的不同主机中，这时就只能使用绝对 URL 了。

2. 同一个文件中的链接

上面提到的链接指针可以使读者在整个 Internet 上方便地进行文件之间的链接，这种链接方式称作远程链接。但如果编写了一个很长的 HTML 文件，从头到尾地浏览很浪费时间，可在同一文件的不同部分之间也建立链接，使用户方便地在上下文之间跳跃浏览，这种指向同一文件的链接方式叫做本地链接。前面曾提到过一个超文本链接指针包括两个部分，一个指向目标的链接指针，另一个是被指向的目标。对于一个完整的文件，可以用 URL 来唯一地标识；对于同一文件的不同部分，可用目标标识。标识一个目标的方法为：

…

name 属性将放置该标记或书签的地方标记为“KKK”，KKK 是全文唯一的标记串。在这种情况下，<a>和之间的内容可有可无。这样，就把放置标记的地方做了一个叫做“KKK”的标记。做好标记后，可以用下列方法来指向它：

转向下一处

这时单击“转向下一处”这段文字，浏览器就从当前位置转到标记名为 KKK 的部分。

8.1.4 在 HTML 文件中使用图像

1. 在 HTML 文件中显示图像

在浏览器上显示图像，必须有特定的格式，目前使用的浏览器通常支持 GIF 和 JPEG 格式的图像。在 HTML 网页中添加图像是通过<img>标记实现的，它有如下几个较为重要的属性。

（1）src 属性：指明图形的 url 地址。

（2）height 属性：决定图形的高度。

（3）width 属性：决定图形的宽度。

（4）border 属性：决定边框线的宽度，0 表示无边框。

（5）alt 属性：指明图像显示的备用文本。

下面通过一个示例来说明<img>标记的使用方法。要显示图像的文件名为“center1.gif”，

它是当前目录下的 images 子目录中的文件。其 HTML 源文件如下：

代码清单 w8-5.htm

```
1    <html>
2        <head>
3            <title>测试页</title>
4        </head>
5        <body>
6            <img alt="校庆" src="images/center1.gif" >
7            <img alt="校庆" src="images/center1.gif"    border=8>
8            <img alt="校庆" src="images/center1.gif"    height=150 width=150>
9        </body>
10   </html>
```

执行代码清单 w8-5 的 HTML 代码后，在浏览器中显示的结果如图 8-6 所示。图 8-6 中的第一个图是通过调用“center1.gif”图像文件显示出来的（代码清单 w8-5.htm 中的第 6 行）。

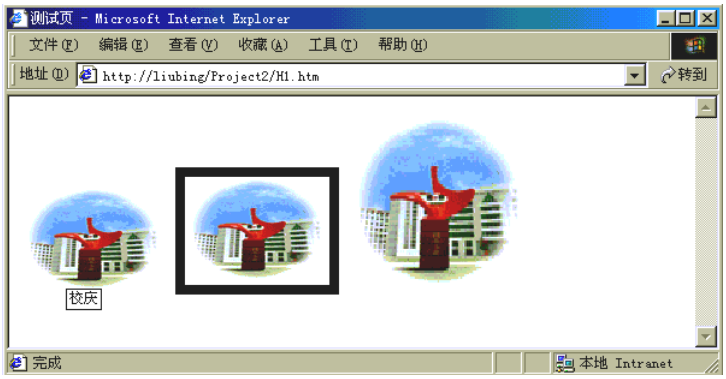

图 8-6　HTML 文件举例

如果在同一文件中需要反复使用同一个图像文件时，最好在<img>标记中使用相对路径名，而不要使用绝对路径名或 URL，因为使用相对路径名时，浏览器只需将图像文件下载一次，再次使用这个图像时，只要再重新显示一次即可。如果使用绝对路径名，每次显示图像时都要到服务器下载一次图像文件，这样会大大降低图像的显示速度。在代码清单 w8-5.htm 中，使用的是相对路径，表示所调用的图片是当前目录下的“images”子目录下的“center1.gif”文件。

在<img>标记中，还可以对显示的图像添加边框。如图 8-6 的中间所示，其边框的像素值为 8。其 HTML 语法为：

```
<img alt="校庆" src="images/center1.gif"    border=8>
```

<img>标记中还提供了两个属性：height 和 width，其属性值的单位是像素，用来确定一个图像的高度和宽度。如果对一个图像设置的 height 和 width 值与原来的取值不一致时，在浏览器上所看到的图像大小就会发生相应的变化。例如：

```
<img alt="校庆" src="images/center1.gif"    height=150 width=150>
```

这个脚本会把图像按高度为 150 像素，宽度为 150 像素的大小在浏览器中显示出来，显示结果如图 8-6 最右边的图所示。

在网页制作过程中，可以利用上述功能提高图像的传输速度。由于小的图像占用的磁盘空间比较少，在网上传输的时间比较短，所以可以创建一个比较小的图像，然后再在 Web 上按比例放大，达到所希望的尺寸。但需要注意的是，放大倍数太大可能会使图像显得有些模糊。在图像制作时既要考虑到传输速度，又要兼顾图像的显示效果。

虽然图像可以使网页变得绚丽多彩，富有吸引力，但也会带来传输速度降低的问题。有些浏览者为了提高网页下载速度，也许会关掉浏览器中载入图像的命令。这样在客户浏览器上就不能显示出图片，为了使浏览文本的用户能够了解页面上图片的内容，可以使用<img>标记中的 alt 属性对加入的图像进行文字说明。当浏览器不能显示该图像时，可以将 alt 引导的文字显示在屏幕上，从而替代看不到的图像。例如：

<img alt="校庆" src="images/center1.gif" >

<img…alt=…>语句被执行后，如果浏览器支持这种图像类型，则浏览器就会把图像显示在窗口中，alt 引导的内容被忽略；如果浏览器不支持该图像类型，alt 引导的内容会出现在屏幕上，以弥补无法显示的图像。

2. 在 HTML 文件中利用图像建立超级链接

如果在链接标记<a>和的中间放置一个<img>标记，这个图像将会成为一个链接点，产生一个链接。例如：

 <IMG src="images/center1.gif" align=left>

当用户单击这个图像后，浏览器就会显示“default.asp”这个文件的内容。

8.1.5 框架结构的使用

框架能够将页面分成多个独立变化的窗口，每个窗口可以显示不同的 Web 页面，并可以不断更换显示的对象。使用框架结构，可以使屏幕的信息量增大，使 Web 网页更加吸引读者。有关框架内容的 HTML 语法为：

```
<frameset>
    <noframes>…</noframes>
    <frame SRC="URL">
…
</frameset>
```

其中，<noframes>...</noframes>中的内容显示在不支持框架结构的浏览器窗口中，一般用来指向一个普通版本的 HTML 文件，以便让不支持框架结构的浏览器用户阅读。

框架结构由<frameset>指定，并且可以嵌套，分区中各部分显示的内容用<frame>指定。框架结构可以将窗口横向分成几个部分，也可以纵向分成几个部分，还可以混合分框。

框架结构标记可以嵌套，用以实现大框架中套小框架。它主要有两个属性：rows 和 cols，这两个属性可以将浏览器页面分为 n 行 m 列，当然也可以各自独立使用。

下面来看一个框架结构的例子，其 HTML 源文件如代码清单 w8-6.htm 所示，运行结果如图 8-7 所示。

代码清单 w8-6.htm

```
1       <html>
2           <head>
```

```
3                    <title>武汉工业学院</title>
4                    <frameset cols="*,140" >
5                    <frameset rows="*,80" >
6                              <frame src="a.htm" name="f1">
7                              <frame src="b.htm" name="f2" scrolling="no">
8                    </frameset>
9                     <frameset rows="*,80"    >
10                         <frame src="c.htm" name="f3">
11                         <frame src="d.htm" name="f4" >
12                    <frameset>
13              </frameset>
14              </head>
15         </html>
```

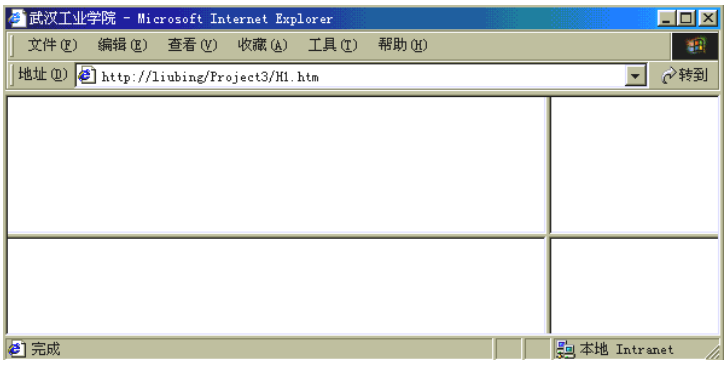

图 8-7　框架结构示意图

代码清单 w8-6.htm 中第 4 行表示把浏览器窗口分成两列，如果浏览器窗口的大小为 640 像素×480 像素，那么框架中右面一列的宽度为 140 像素，左面一列的宽度为浏览器窗口所占宽度减去 140 像素。其中“*”表示除了明确的值以外剩下的值。

代码清单 w8-6.htm 中的第 5 行表示把浏览器窗口的第一列分成两行，下面一行的高度为 80 像素，上面一行是浏览器窗口的高度减去 80 像素。第 9 行与前述相同。

代码清单 w8-6.htm 中第 6、7、10 和 11 行使用的是<frame>标记，该标记有以下主要属性。

（1）src 属性：指定框架单元的 URL 源，如第 6 行中指出的是当前主机当前目录下的“a.htm”文件。即在此框中显示“a.htm”的内容。

（2）name 属性：为该框架单元命名，主要为将来改变框架内容提供入口。

（3）scrolling 属性：设置框架是否使用滚动条。有 yes、no 和 auto 三个值，分别表示强制使用滚动条，禁止使用滚动条和自动判断使用滚动条。

8.1.6　表单的应用

到目前为止，所介绍的 HTML 文件的制作方法对于 Internet 网络用户来说都是单方向的，也就是说，读者只能通过浏览器查看网上的信息。但是在大多数网站上都能看到其网页上还有文本框、按钮、下拉框等。例如，如果想在网上查找某种信息，那么可在搜索引擎的文本框中输入该信息的关键字，然后单击“搜索”按钮，搜索引擎就会把与该关键字相关的信息罗列出

来，这就使得读者与 Web 服务器之间能够进行交流，通常把这种方式叫做交互的（或者双向的）。这种方式可以使 Internet 网络用户在很短的时间内查到所需要的信息，提高浏览效率。这一交互方式是由 HTML 和驻留在 Web 服务器上的程序共同完成的，驻留在 Web 服务器上的程序有许多种，编写这些程序，除了要熟悉 HTML 以外，还需要熟悉 Web 服务器所驻留主机的操作系统，以及操作系统所支持的某种语言。下面，主要介绍用 HTML 如何编写表单，为用户提供输入信息的界面。

1. 什么是表单

HTML 提供的表单是用来将用户数据从浏览器传递给 Web 服务器的。例如，可以利用表单建立一个录入界面，也可以利用表单对数据库进行查询。在这里需要说明的是，表单的操作是与服务器进行交互的操作，而服务器端的操作是通过服务器端的程序来实现的。实现在服务器端的操作有许多种方式，其中 ASP（动态服务网页）的方式就是一种，它可以通过 ADO 方式与多种数据库相连。

ASP（Active Server Page）程序在服务器端工作，并且通过服务器端的编译动态地送出 HTML 文件给客户端，它负责处理 HTML 文件与运行在服务器端的程序之间的数据交换。当用户输入信息（这个信息可以是查询条件，也可以是传送给服务器的某些内容）并提交给服务器后，便激活了一个 ASP 程序。该 ASP 程序又可以调用操作系统下的其他程序（如数据库管理系统）完成读者的查询任务，当操作系统下的程序完成查询之后，便把查询结果传给 ASP，通过 ASP 传给 Web 服务器。由此可以看出，ASP 程序在用户与服务器之间进行交互查询时所起的重要作用。

通过图 8-8 所显示的表单，来给读者介绍一组新的标记，分别是 FORM、INPUT、SELECT、TEXTAREA 等。在学习了这几个标记的使用之后，便可以使用 HTML 制作表单了。

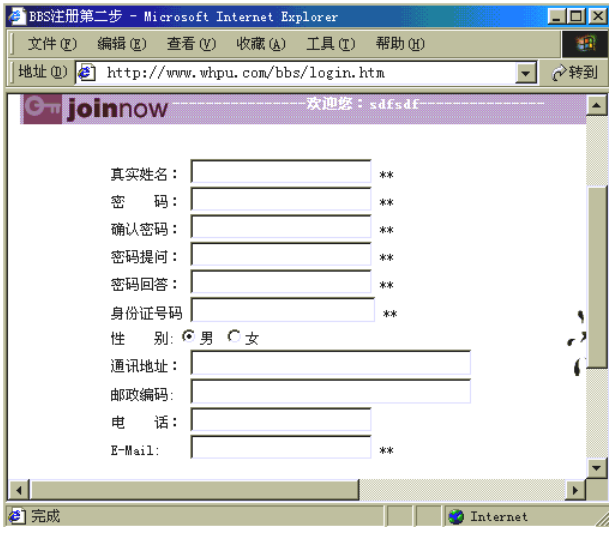

图 8-8 表单示例

2. 表单的标记

表单就是为 Internet 网络用户在浏览器上建立一个交互接口，使 Internet 网络用户可以在这个接口上输入信息，然后使用提交按钮，将 Internet 网络用户的输入信息传送给 Web 服务器。

在 HTML 中，有一个专门的标记 FORM 提供表单的功能，由表单开始标记<form>和表单结束标记</form>组成，表单中所设置的文本框、按钮或下拉菜单，也是通过标记完成。在表单的开始标记<from>中带有两个属性：action 和 method。创建表单的 HTML 语法格式如下：

< form　action = "…" method="…" >

…

</form>

FORM 标记有以下主要属性。

（1）action 属性。action 属性是用来指出，当这个 FORM 提交后需要执行的驻留在 Web 服务器上的程序名（包括路径）。一旦 Internet 网络用户提交输入信息后服务器便激活这个程序，完成某种任务。例如：

<form action = login.asp method = post > … </form>

当用户单击“提交”按钮以后，Web 服务器上的“login.asp”将接收用户输入的信息，以登记用户信息。

（2）method 属性。method 属性是用来说明从客户端浏览器将 Internet 网络用户输入的信息传送给 Web 服务器时所使用的方式，共有两种：post 和 get，默认的方式是 get。这两者的区别是：在使用 post 时，在表单中所有由用户输入的值都按一定的规律放入报文中，而不是附加在 action 所设定的 URL 之后；在使用 get 时，将 FORM 的输入信息作为字符串附加在 action 所设定的 URL 的后面，中间用“？”号隔开，即在客户端浏览器的地址栏中可以直接看见这些内容。

在<form>与</form>之间，可以使用除<form>以外的任何 HTML 标记，这将使 FORM 变得非常灵活。只使用<FORM>标记是很难完成 Internet 网络用户的输入信息的，在 FORM 的开始与结束标记之间，除了可以使用本书前面讲的那些标记外，还有三个特殊标记，分别是：INPUT（在浏览器的窗口上定义一个可以供用户输入的单行窗口、单选或多选按钮）、SELECT（在浏览器的窗口上定义一个可以滚动的菜单，用户在菜单内进行选择）、TEXTAREA（在浏览器的窗口上定义一个域，用户可以在这个域内输入多行文本）。

3．HTML 中的 INPUT 标记

HTML 中的 INPUT 标记是在表单中最常用的标记。在网页上所见到的文本框、按钮等都由这个标记引出。下面给出的是 INPUT 标记的标准格式：

<input　type="…" value ="…">

其中，type 属性是用来说明给用户提供的信息输入的方式，如文本框、单选按钮或复选框，其取值如下：

1）type = "TEXT"　表示在表单中使用单行文本框。

2）tpye = "PASSWORD"　表示在表单中为用户提供密码输入框。

3）tpye = "RADIO"　表示在表单中使用单选按钮。

4）tpye = "CHECKBOX"　表示在表单中使用复选框。

5）tpye = "SUBMIT"　表示在表单中使用提交按钮。

6）tpye = "RESET"　表示在表单中使用重置按钮。

（1）文字输入和密码输入。下面介绍文字输入和密码输入的制作。请看代码清单 w8-7.htm 其在浏览器中显示的结果如图 8-9 所示。

图 8-9　文字输入和密码输入

代码清单 w8-7.htm

```
1   <html>
2       <head>
3           <title>这是个测试页</title>
4       </head>
5       <body>
6               <form action="REG.ASP" method=post>
7               请输入您的真实姓名: <input type=text name=姓名><br>
8               您的主页的网址: <input type=text name=网址  value=HTTP://><br>
9               密码: <input type=password name=密码><br>
10              <input type=submit value="发送"><input type=reset value="重设">
11          </form>
12      </body>
13  </html>
```

从代码清单 w8-7.htm 可以看出，第 6 行至第 11 行使用了制作表单的标记<FORM>…</FORM>说明。第 7 行是单行文本框标记，并设置属性 name=“姓名”，这个属性定义了文本框在这个表单中的名字叫“姓名”，以便和其他的文本框区别开来。当用户在这个文本框中输入信息并送到 Web 服务器后就激活了“REG.ASP”程序，在该程序中要获得这个文本框输入的内容，就要用到“姓名”这个名字。在第 8 行同样定义了一个文本框，但其设置了属性 VALUE=“HTTP://”，表示该文本框的默认值为“HTTP：//”。第 9 行是密码输入框，其与文本框是有区别的。文本框是用户输入什么值，在文本框中就显示什么值，而密码输入框则是不管用户输入什么值都以“*”来显示。

另外要控制用户输入数据的长度，可在 INPUT 标记中用最大长度的属性。例如，一般中国人的名字最多为 4 个汉字即 8 个字节，所以在控制用户输入姓名时，限制其最大长度为 8，则可把代码清单 w8-7.htm 的第 7 行改成：

请输入您的真实姓名: <input type=text name=姓名 maxlength=8>

（2）复选框（Checkbox）和单选按钮（Radio Button）。在图 8-8 中要求 Internet 网络用户输入一些个人的基本信息，其中“性别”一项不是输入而是进行选择，因为性别只有两种：男和女，两者选一，这种形式的选择框叫单选按钮，即在几个选择中仅能选中一个。另外有一种选择框叫复选框。即允许用户可以选中多个。创建单选按钮和复选框的语法格式如下。

单选按钮：<input type=radio value="…" checked>。

复选框：<input type= checkbox value="…" checked>。

其中 checked 属性用来表示在初始情况下选项是否被选中。代码清单 w8-8.htm 是复选框与单选按钮标记的应用示例，在浏览器中的显示结果如图 8-10 所示。

代码清单 w8-8.htm

```
<html>
    <head>
        <title>这是个测试页</title>
    </head>
    <body>
            <form action="REG1.ASP" method=post>
              选择一种你喜爱的水果:
              <br><input type=radio name=水果  value="香蕉">香蕉
              <br><input type=radio name=水果  checked value="草莓">草莓
              <br><input type=radio name=水果  value="橘子">橘子
              <br>选择你所喜爱的运动:
              <br><input type="checkbox"   name=ra1 checked value="足球">足球
              <br><input type="checkbox"   name=ra2 checked value="篮球">篮球
              <br><input type="checkbox"   name=ra3 value="排球">排球
              <br><input type=submit value="发送"><input type=reset value="重设">
            </form>
        </body>
</html>
```

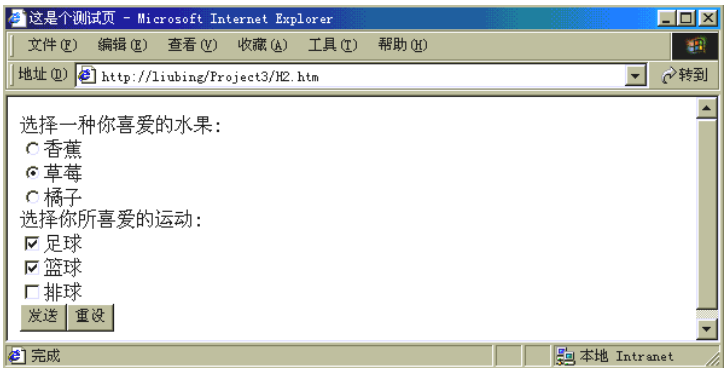

图 8-10　单选按钮和复选框

（3）按钮的制作。在上面几个例子中，都有两个按钮，一个是“发送”按钮，另一个“重设”按钮。其实“发送”按钮真正的含义是“提交”，即当 Internet 网络用户单击这个按钮后，用户所输入的信息便提交给一个驻留在 Web 服务器上的程序，让服务器进行处理，其典型的格式是：<INPUT TYPE="SUBMIT" VALUE="发送">。提交按钮在 FORM 中是必不可少的，前几个例子只是说明 INPUT 语句中类型的使用，作为 FORM 语句并不完整，每个 FORM 中都应有且仅有一个提交按钮。当设置“提交”按钮标记时，如果缺省 value 属性，则浏览器窗口中的按钮上出现“submit”的字样，这个属性也可以自行设定，以改变按钮上的提示，如 value=“提交”。

另一个在浏览器常用的按钮是“重设”按钮，当 Internet 网络用户单击这个按钮后，用户所输入的信息全部被清除，以便用户重新输入信息。其常用的格式：<input type = "reset" value="

重新输入">，而且在这个标记设置中如果缺省 value 属性，则浏览器窗口中的按钮上出现“reset”的字样，这个属性也可以自行设定，来改变按钮上的提示，如 value=“重新输入”。

4. HTML 中的 SELECT 标记

在制作 HTML 文件时，使用<form>…</form>标记可以在浏览器窗口中设置下拉式菜单或带有滚动条的菜单，Internet 网络用户可以在菜单中选中一个或多个选项。图 8-11 显示了一个下拉菜单，其 HTML 源文件如代码清单 w8-9.htm 所示。

代码清单 w8-9.htm

```
<html>
    <head>
        <title>武汉工业学院</title>
    </head>
    <body>
        请从下面课程中选择几门选择课:
        <form action="h1.asp" method=post id=form1 name=form1>
        <select  name=x1  multiple>
            <option value=network>网络技术
            <option value=write>书法
            <option value=music>音乐欣赏
            <option value=literature>现代文学
            <option value=media>多媒体技术
        </select>
        </form>
    </body>
</html>
```

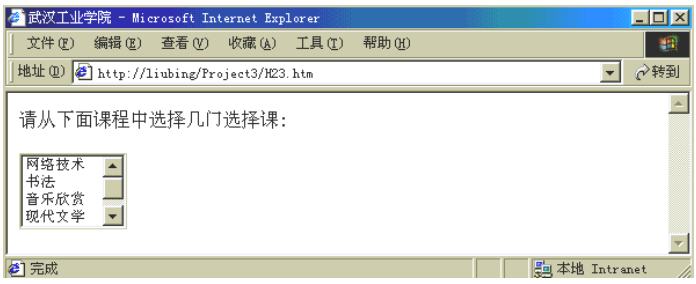

图 8-11 设置下拉菜单

从代码清单 w8-9.htm 可以看出，创建下拉菜单的标准格式如下所示：

```
<select…>
    <option value=值 1>选项一
    <option value=值 2>选项二
    <option value=值 3>选项三
</select>
```

SELECT 标记有几个可以设置的属性值，分别是 name、size 和 multiple。其中 name 属性是当 Internet 网络用户将表单提交时作为输入信息的名字；size 属性控制在浏览器窗口中这个菜单选项的显示条数；multiple 属性允许读者一次可选多个选项，如果缺省 multiple，用户一

次只能选一项，类似于单选，当有 multiple 属性时就是多选。

在 SELECT 的开始和结束标记之间，有几个 OPTION 标记就有几个选项，选项的具体内容写在每个 OPTION 之后。若在 SELECT 标记中设定 multiple 属性的话，可以在多个 OPTION 标记中带有 selected 属性，表示这些选项已经预选。另外，OPTION 标记中的 value 属性值是用户选择了某个选项后，提交给服务器的返回值。

8.1.7　HTML 中的表格

在浏览器中浏览制作的网页时，刚开始总会感觉到控制不好网页内容的排版，总是不能按照设计的意愿去把一些文字或图片放到指定的位置，这时就可以考虑采用表格来控制。表格是组织数据的有效手段。利用所见即所得的网页编辑器进行表格的自动生成可以避免手工制表的烦琐，但如果要更好地控制表格的表现形式，熟悉与掌握表格的 HTML 标记是十分必要的。图 8-12 是用 HTML 制作的一个简单的表格。

图 8-12　　表格示例

从图 8-12 可以看出一个表格有一个标题（Caption），用来表明表格的主要内容，并且一般位于表的上方；表格中由行和列分割成的单元叫做表元（Cell），又被分为表头（用 TH 标记来表示）和表数据（用 TD 标记来表示）；表格中分割行列的线称为框线（Border）。

1. 表格的标记

创建表格基本框架的 HTML 语法格式如下所示：

```
<table width=75% border=1 cellspacing=1 cellpadding=1>
    <caption></caption>
    <tr>
        <td></td>
        <td></td>
        <td></td>
    </tr>
    <tr>
        <td></td>
        <td></td>
        <td></td>
    </tr>
```

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

（1）TABLE 标记。一个表格至少有一个 table 标记，用来决定一个表格的开始和结束，而且 TABLE 标记可以嵌套。TABLE 标记有以下 5 种属性。

1）border 属性：指定围绕表格的框的宽度（单位用像素）。

2）cellspacing 属性：指定框线的宽度。

3）cellpadding 属性：用于设置表元内容与边框线之间的间距。

4）align 属性：用来控制表格本身在页面上的对齐方式。其取值可是 left（左对齐）、center（居中对齐）、right（右对齐）。

5）width 属性，用来设置表格的宽度，可以以像素为单位，也可用占浏览器窗口的百分比来定义。

（2）CAPTION 标记。CAPTION 标记用来标注表格标题。CAPTION 标记必须紧接在 TABLE 开始标记之后，放在第一个 TR 标记之前。通过该标记所定义的表格标题一般显示在表格的上方，而且在水平方向居中对齐。另外，如需要突出显示表格的标题，可以在 CAPTION 标记之间加入对字体进行加重显示的标记。如：

```
<table width=75% border=1 cellspacing=1 cellpadding=1>
    <caption>
        <h2>表格标题强调</h2>
    </caption>
    <tr>
    …
    </tr>
</table>
```

（3）TR 标记。定义表格的行。TR 标记有两个属性，一个是 align 属性，用来设置表行中的每个表元在水平方向的对齐方式，其取值可以是 left（左对齐）、center（居中对齐）、right（右对齐）；另一个是 vlign 属性，用来设置表行中的每个表元在垂直方向的对齐方式，其取值可以是 top（向上对齐）、center（居中对齐）、bottom（向下对齐）。例如，要使表行中各单元的内容水平方向右对齐，垂直方向向上对齐，可使用如下源代码：

```
<tr   align=right valigh=top>
```

（4）TD 标记。TD 标记用来表示一个表行中的各个单元。TD 标记内几乎可以包含所有的 HTML 标记，甚至还可以嵌套表格。该标记与 TR 标记一样，有 align 和 valign 两个属性，如果在 TD 标记和 TR 标记中都设置了 align 和 valign 属性，而且所设置的属性值不相同，这时以 td 标记所设置的属性值为准。另外，td 标记还有两个属性，一个是 width 属性，用来设置表元的宽度，另一个是 height 属性，用来设置表元的高度。这两个属性的取值单位都是像素。在同一行中将多个表元设置为不同高度，或者在同一列中将多个表元设置为不同宽度，都有可

能导致不可预料的结果。

2. 表格使用实例

在这个实例中，通过制作一个登记表格来向读者说明如何制作一个比较复杂的表格。在表格中经常会出现跨多行、多列的表元，这就要利用到 TD 标记的另外两个属性，即 colspan 和 rowspan 属性来完成。例如：

<th　colspan=3 > 登记照<th >　　<!表示这个表项标题将横跨三个表项的位置>

<th　rowspan=3 > 登记照<th >　　<!表示这个表项标题将纵跨三个表项的位置>

另外每个表元还可以设置其背景颜色。例如

<th　colspan=3　bgcolor=yellow> 登记照<th >

还可以在表格中插入超链接或图片，如果能对这个例子举一反三的话，那么仅需制作一个无框线的表格，就可以把各种数据按照设计者所希望达到的效果在页面上表现出来。

下面就给出一个具体的实例，相关程序代码如代码清单 w8-10.htm 所示，其页面效果如图 8-13 所示。

图 8-13　综合表格实例

代码清单 w8-10.htm

```
<html>
    <head>
        <title>表格综合实例</title>
    </head>
    <body>
        <p>
        <table border=1 cellpadding=1 cellspacing=1 width="75%">
        <caption>
        <h3>大奖赛登记表</h3>
        </caption>
            <tr>
                <td bgcolor=LightGoldenrodYellow>报名号</td>
                <td>757</td>
                <td bgcolor=LightYellow>性别</td>
                <td>女</td>
                <td rowspan=2>
```

```
                    <img src="image\center.gif">
                </td>
            </tr>
            <tr>
                <td bgcolor=FloralWhite>姓名</td>
                <td  colspan=3>
                <a href="http://www.jljiangli.com.cn">江小丽</a></td>
            </tr>
            <tr>
                <td bgcolor=Cornsilk>推荐单位</td>
                <td colspan=4>宇宙公司</td>
                </tr>
        </table>
        </p>
    </body>
</html>
```

8.2 级联式样式表（CSS）

级联式样式表（CSS）的作用是为 HTML 文档提供美观而一致的外观。通过将许多 Web 页链接到同一个外部样式表，就可以为整个 Web 站点制定统一的风格，而且改变这些外观也非常方便。

任何支持 HTML 4.0 或更高版本的 Web 浏览器都会支持大多数的 CSS 样式属性。使用 CSS 样式定义 HTML 页和 Web 窗体中元素和文本的外观和位置，可将 style 属性以内联方式添加到许多 HTML 元素上，还可将 CSS 样式嵌入到<style>块中或存储在外部级联式样式表（.css）文件中。

8.2.1 定义 CSS

级联式样式表（CSS）包含应用于 HTML 文档中元素的样式定义，CSS 样式定义元素的显示方式，在页面中放置元素的位置等。可以创建一个通用规则，只要 Web 浏览器遇到一个元素实例，或是一个分配给某个样式 CLASS 的元素，该规则就立刻应用属性，而不是将属性逐个分配给页面中的每个元素。

CSS 样式可以通过内联方式放置在单个 HTML 元素内，也可以在 Web 页的<head>部分的<style>块内加以分组，或从单独的外部 CSS 样式表文件中导入。同一个外部样式表文件可链接到很多 Web 页，从而使整个 Web 站点具有统一的外观。

1. 在 STYLE 块内定义 CSS 样式规则

每个 CSS 样式规则都有两个主要部分：选择器（如 H1）和声明（如 color:red）。声明包括属性（如 color）和属性的值（如 red）。例如，一条说明“将<h1></h1>标记内包含的所有文本设置为居中，并采用红色字体颜色”的简单 CSS 样式规则可以写成：

```
H1 {text-align:center; color:red;}
```

CSS 样式规则可以在 HTML 文档的<HEAD>部分的<STYLE>块内定义。代码清单

w8-11.htm 中，定义了一条 CSS 样式规则，并将该规则应用到 Web 页上的所有<H1>元素，运行结果如图 8-14 所示。

代码清单 w8-11.htm

```
<html>
    <head>
        <title>HTML CSS 示例</title>
            <style type="text/css">
                H1 {text-align:center; color:red;}
            </style>
    </head>
    <body>
            <h1>这些文字居中且显示成红色。</h1>
    </body>
</html>
```

在此 Web 页上，任何出现在<h1></h1>标记内的文本都将居中并显示为红色。因此，每当文档中出现 <h1> 标记时，不再需要重复地重新分配这些样式属性。另外，如果想更改<h1></h1> 标记内所有文本的颜色（或任何其他属性），则只需简单地编辑一条样式规则即可。

图 8-14　CSS 示例

2. CSS 样式规则的优先级

CSS 样式规则从一定意义上讲是级联的，即全局样式规则会一直应用于 HTML 元素，直到有局部样式规则将其取代为止。一般而言，局部样式规则的优先级高于通用样式规则。举例来说，在某 Web 页的<style>块内定义的样式，可为该页修改外部 CSS 样式表中定义的 Web 站点样式。同样，在该页的单个 HTML 标记内定义的行内样式，可替代所有在其他地方为元素定义的任何样式。

在局部样式应用于 HTML 元素之后，全局样式规则中与局部 CSS 样式规则不冲突的部分继续应用于这些元素。在代码清单 w8-11.htm 中，控制<h1>标记中文本的局部 CSS 样式替换 Web 浏览器针对<h1>文本的某些全局样式规则（使<h1></h1>标记内的文本居中并显示为红色），但局部 CSS 样式并不更改其他全局样式规则（所有<h1></h1>标记内的文本继续以较大字体和粗体样式显示）。全局样式规则和局部样式规则都按照该顺序应用，结果是该页中的所有<H1></h1>标记内的文本都以较大字体、加粗、居中且为红色的样式显示。

3. 将 HTML 元素分配给 CSS 样式 CLASS

每当文本出现在分配给特定 CLASS 的 HTML 标记中时，就会自动应用 CSS 样式。在<style>块中，CLASS 样式规则的选择器以一个句点开始。例如：

.head2 {font-size:14pt; text-align:center; color:red; font-weight:bold; font-style:italic;}

若要以内联方式应用这种类型的样式，请向支持内联样式的标记或标签添加 CLASS 属性：

作为 CLASS 属性值输入的字符串，应与该页所应用样式规则的选择器相匹配。在代码清单 w8-12.htm 中，定义和应用了名为 head2 的 CSS 样式 CLASS。

代码清单 w8-12.htm

```
<html>
    <head>
    <title> HTML CSS 示例</title>
    <style type="text/css">
        body {background:#FBFBFB; font-family:Verdana, Arial, Helvetica, sans-serif; font-size:9pt;}
        A:link {color:blue; text-decoration:none}
        A:active {color:red; text-decoration:none}
        A:visited {color:green; text-decoration:none}
        .head2 {font-size:14pt; text-align:center; color:red; font-weight:bold; font-style:italic;}
    </style>
    </head>
    <body>
        <div class="head2">This text is centered, large, red
        <SPAN style="color:green; font-style:normal; text-decoration:underline;">
            and green
        </span>
        , bold, and italic
        </div>
    </body>
</html>
```

CLASS head2 的所有样式规则都应用于<DIV>中所包含的文本。由于在<BODY>中的一个标记内定义的内联样式的优先级高于在<HEAD>中定义的<STYLE>块样式，因此文本“and green”将以绿色、不加粗且带下划线的样式显示。

4. 外部 CSS 样式表

外部 CSS 样式表文档是指包含样式规则，并且以 .css 为扩展名的纯文本文件。语法格式如下：

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

表示将外部 CSS 样式表 mystyles.css 中的样式规则应用到该页。在代码清单 w8-13.htm 中，Mystyles.css 是一个外部 CSS 样式表文档，包含<h1>标记的样式规则。

代码清单 w8-13.htm

```
<html>
    <head>
        <title> CSS 示例</title>
        <link rel="stylesheet" href="Mystyles.css" type=" text/css">
    </head>
    <body>
```

```
        <h1>这里的文字显示红色</h1>
    </body>
</html>
```

外部 CSS 样式表中列出的样式规则的写法，与在内部 CSS 样式表的<style>块中的写法一样，只是前后没有<style> </style>标记：

```
H1 { text-align:center; color:red; }
.head2 { font-size:14pt; text-align:center; color:red; font-weight:bold; font-style:italic; }
```

一个外部 CSS 样式表可链接到许多 HTML 页，从而在整个 Web 站点内应用一致的样式。CSS 样式表将格式设置规则与内容分开，从而大大方便了样式规则的定位和编辑。

5. CSS 的注释

对于任何代码，添加注释都是非常有用的。注释用来说明所写代码的含义，因此对于其他用户读懂这些代码是很有用的。

CSS 用 C/C++语言的标记进行注释，“/*”放在注释的开始处，“*/”在结束处。代码清单 w8-14.htm 说明了如何进行注释。

代码清单 w8-14.htm

```
<head>
    <title>CSS 例子</title>
    <style type="text/css">
        h1 { font-size: x-large; color: red }          /*这是一个 CSS 的定义*/
        h2 { font-size: large; color: blue }
    </style>
</head>
```

养成注释的习惯，不仅有利于其他人读懂编码，而且对编程者也是很有好处的。例如，当把一个格式页提交用户使用，经过很长时间，用户又需要重新修改格式页时，可能作者已经忘记代码的准确含义。注释可以为用户标识特殊格式页的重点及技巧的细节。

8.2.2　CSS 属性分类

CSS 包括字体、颜色和背景、文本、边框、用户界面、表和视觉效果属性。

1. 字体

字体格式属性确定格式化文本在 Web 浏览器中将如何显示。可设置首选字体系列，字体大小和颜色，以及文本是否显示为粗体、斜体或带下划线。可将字体格式设置属性添加到在外部 CSS 样式表中定义的 CSS 样式规则，或添加到 Web 页<head>部分的<style>块中定义的 CSS 样式规则。

若要将样式规则应用于 Web 页的<body>块中的某个特定 HTML 元素，需要将 CLASS 或 ID 属性添加到所需 CSS 样式规则选择器的元素的开始标记中，也可将 CSS 字体属性直接添加到支持 STYLE 属性的 HTML 元素中。

注意：如果定义了 body{}样式，则 Web 页中所有没有内联样式格式化的文本都将以该定义中的指定样式显示。

表 8-3 列出了 CSS 样式中的可设置的字体属性。

表 8-3 字体属性

属性	属性含义	属性值
font-family	选择字体	所有字体
font-size	选择字体大小	绝对大小、相对大小、长度、百分比
font-weight	字体加粗	normal、bold、bolder、lighter、100、200、300、400、500、600、700、800、900
font-variant	字体变形	normal（普通）、small-caps（小型大写字母）
font-style	字体风格	normal（普通）、italic（斜体）、oblique（倾斜）

其中，字体大小（font-size）属性取值说明如下：

（1）绝对大小的属性值允许是：xx-small、x-small、small、medium、large、x-large、xx-large。长度分绝对长度和相对长度两种，将字体大小的属性值设为绝对长度（使用的单位为像素和英寸）时需要谨慎地考虑到其适应不同浏览环境时的弱点。对于一个用户来说，绝对长度的字体很有可能会很大或很小。

（2）相对大小的属性值允许是：larger 或 smaller。

（3）百分比是指与父元素的百分比关系。

在代码代码清单 w8-15.htm 中，设置了相关字体属性，其在浏览器中的显示结果如图 8-15 所示。

代码清单 w8-15.htm

```
<html>
    <head>
        <title>字体样式测试</title>
    </head>
    <body>
1       <p style="font-family:lucida console">www.whpu.edu.cn</p>
2       <p style="font-style:italic">www.whpu.edu.cn</p>
3       <p style="font-style:oblique;font-weight:bold;font-size:24pt">www.whpu.edu.cn
</p>
4       <pstyle="font-style:oblique;font-weight:bold;font-size:24pt;
                font-variant:  small-caps">www.whpu.edu.cn</p>
5       <p style="font:italic small-caps bold 36pt,GlitzyCurl">www.whpu.edu.cn</p>
    </body>
</html>
```

在代码清单 w8-15.htm 中，第 1 行只是采用了 font-family 属性，这行代码定义了“www.whpu.edu.cn”将以“lucida console”的字体显示；第 2 行 font-family 属性值为默认值，定义了 font-style 属性属性值为“italic”斜体；第 3 行 font-family 属性值为默认值，font-style 属性值为“oblique”，另外还定义了 font-weight 属性值为“bold”，font-size 属性值为 24pt（默认情况下为 absolute size 绝对大小）；第 4 行只是在第 3 行定义的基础上又增添了 font-variant 属性值“small-caps”（小体大写）。

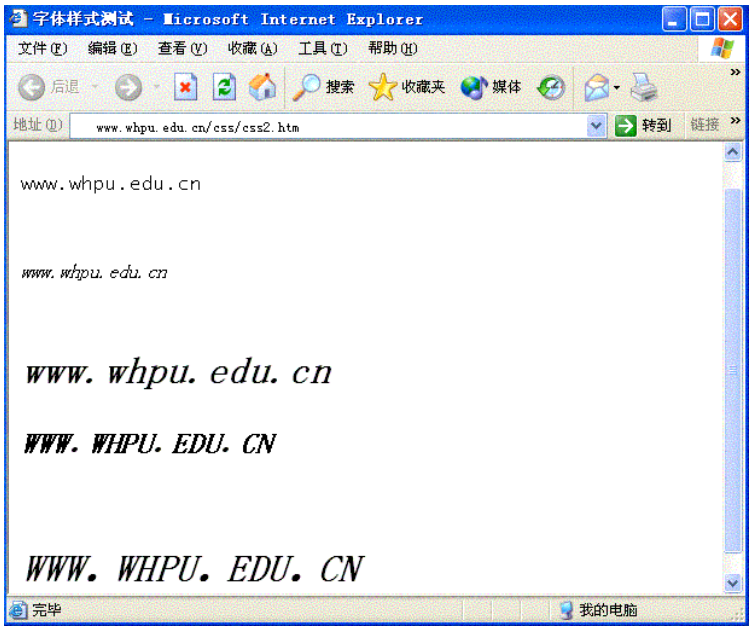

图 8-15　CSS 中的字体设定

2. 颜色和背景

背景属性的作用是通过更改颜色或包含图像来控制背景。如果用图像作为 Web 页的背景，也可指定其位置和平铺属性。通过组合使用这两种属性，可确定背景图像在页面上的位置及图像显示的频率。表 8-4 中列出了 CSS 的颜色和背景属性。

表 8-4　CSS 的颜色和背景属性

属性	属性含义	属性值	属性书写格式
color	定义前景色	颜色	如：H2 { color: #000080 }
background-color	定义背景色	颜色、transparent （透明）	如：BODY { background-color: white }
background-image	背景图像	图片路径	如：body { background-image: url(http://www.whpu. edu.cn/images/bg.gif) }
font-variant	字体变形		
font-style	字体风格		

3. 文本

CSS 文本属性用以改变页面文本的显示方式，这些属性可以细化页面的排版，生成特殊的文本效果。表 8-5 中列出了 CSS 的文本属性。

垂直对齐的属性值的说明如下。

1）baseline：使元素与上级元素的基线对齐。

2）middle：纵向对齐元素基线加上上级元素的高度。

3）sub：下标。

4）super：上标。

5）text-top：使元素与上级元素的字体向上对齐。

6）text-bottom：使元素与上级元素的字体向下对齐。

影响相对于元素行位置的关键字如下：

1）top：使元素与行中最高的元素向上对齐。

2）bottom：使元素与行中最低的元素向下对齐。

表 8-5 CSS 的文本属性

属性	属性含义	属性值
Text-align	水平对齐	左（left）、居中（center）、右（right）或两端对齐（justify）
Vertical-align	垂直对齐	baseline、sub、super、top、text-top、middle、bottom、text-bottom、<百分比>
word-spacing	单词间距	“正常”（normal）或“自定义”
letter-spacing	字母间距	“正常”（normal）或“自定义”
line-height	行间距	normal、<数字>、<长度>、<百分比>
text-indent	文本缩进	<长度>、<百分比>

在代码代码清单 w8-16.htm 中，用 CSS 来控制显示方式，其在浏览器中的显示结果如图 8-16 所示，图 8-17 是不用 CSS 控制显示方式的显示结果。

代码清单 w8-16.htm

```
<html>
    <head>
        <title>文本样式测试</title>
    </head>
    <body>
        <p style="letter-spacing:1em;text-align:justify;text-indent:4em;
            line-height:17pt">
        武汉工业学院 WEB 服务器的域名为：www.whpu.edu.cn；
        武汉工业学院 FTP 服务器的域名为：FTP.whpu.edu.cn；</p>
    </body>
</html>
```

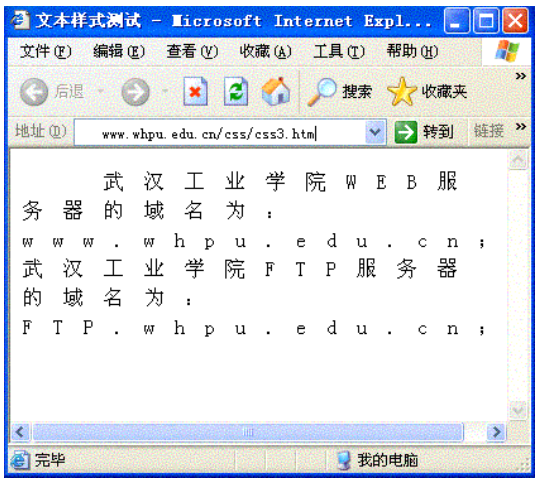

图 8-16 加 CSS 控制显示

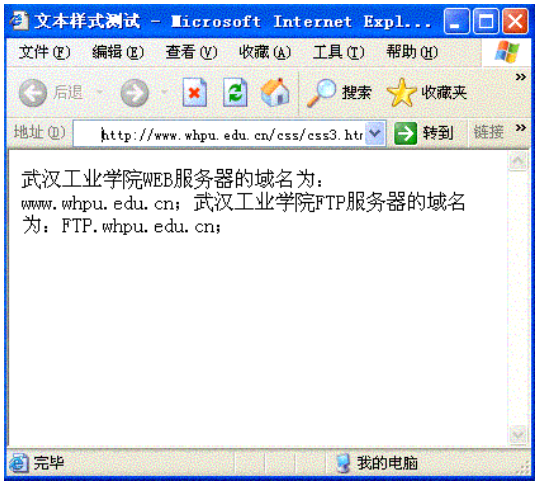

图 8-17 不加 CSS 控制显示

4. 鼠标属性

当把鼠标指针移动到不同的地方时，当鼠标指针需要执行不同的功能时，当系统处于不同的状态时，都会使鼠标指针的形状发生改变。用 CSS 来改变鼠标指针的属性，就是当鼠标指针移动到不同的元素对象上面时，让鼠标指针以不同的形状、图案显示。

在 CSS 当中，这种样式是通过 cursor 属性来实现的。cursor 属性具有很多的属性值，详细内容如表 8-6 所示。

下面来看代码清单 w8-17.htm，了解 cursor 属性在编码过程中的应用。

代码清单 w8-17.htm

```
<html>
    <head>
        <title>changemouse</title>
    </head>
    <body>
        <h1 style=”font-family:文鼎新艺体简”>鼠标效果</h1>
        <p style=”font-family:行书体;font-size:16pt;color:red”>
            请把鼠标移到相应的位置观看效果。</p>
        <div style=”font-family:行书体；font-size:24pt；color:green；”>
            <p><span style=”cursor:hand”>手的形状</span><br>
                <span style=”cursor:move”>移动</span><br>
                <span style=”cursor:ne-resize”>反方向</span><br>
                <span style=”cursor:wait”>等待</span><br>
                <span style=”cursor:help”>求助</span>
            </p>
        </div>
    </body>
</html>
```

表 8-6　cursor 属性值

属性值	说明
auto	自动，根据默认状态自行改变
crossshair	十字线光标
default	默认光标，通常为箭头光标
hand	手型光标
help	“帮助”光标，它是箭头和问号的组合
move	移动
e-resize	箭头朝右方
ne-resize	箭头朝右上方
nw-resize	箭头朝左上方
n-resize	箭头朝上方
se-resize	箭头朝右下方
sw-resize	箭头朝左下方

续表

属性值	说明
s-resize	箭头朝下方
text	文本“I”型
wait	等待

8.2.3 CSS 的单位

1. 长度单位

一个长度单位的值由可选的正号“＋”或负号“－”，一个数字，还有标明单位的两个字母组成。在一个长度值中是没有空格的，例如，1.3 em 就不是一个有效的长度的值，而 1.3em 就是有效的。

无论是相对值还是绝对值长度，CSS 都支持。相对值单位确定一个相对于另一长度属性的长度，因为它能更好地适应不同的媒体，所以是首选的。以下是有效的相对长度单位：

（1）em（em，元素的字体的高度）；

（2）ex（x-height，字母“x”的高度）；

（3）px（像素，相当于屏幕的分辨率）。

绝对长度单位视输出介质而定，所以逊色于相对单位。以下是有效的绝对长度单位：

（1）in（英寸，1 英寸=2.54 厘米）；

（2）cm（厘米，1 厘米=10 毫米）。

（3）pt（点，1 点=1/72 英寸）。

（4）pc（帕，1 帕=12 点）。

2. 百分比单位

一个百分比单位值由可选的正号“＋”或负号“－”、一个数字，还有百分号“%”组成。在一个百分比值之中是没有空格的。

百分比值是相对于其他数值而言的。一般在设置元素字体大小时用到百分比值。

3. 颜色单位

颜色值是一个关键字或一个 RGB 格式的数字。RGB 颜色可以有 4 种形式：

（1）#rrggbb（如#00cc00）；

（2）#rgb（如#0c0）；

（3）rgb（x,x,x），其中 x 是一个介于 0～255 之间的整数，如 rgb（0,204,0）。

（4）rgb（y%,y%,y%），y 是一个介于 0.0～100.0 之间的整数，如 rgb（0%,80%,0%）。

8.3 网页脚本语言——JavaScript

8.3.1 JavaScript 的基础知识

1. 什么是脚本语言

脚本语言是一种简单的描述性语言，并针对 HTML 语言不能很好地解决动态交互这个缺

点而引入的，是对 HTML 语言最重要的一个补充，且能对 Web 页面中的元素进行控制。脚本语言的语法与一般的编程语言并没有什么不同，只是去掉了可能会引起对 Web 浏览用户造成伤害的那部分内容。一般来说，脚本语言是通过一个<script>的标记嵌入到 HTML 文档中，并可以被浏览器解释执行，插入的脚本语言就如同子程序一样被 HTML 元素所调用，成为 HTML 的一部分。目前比较流行的脚本语言有网景公司（Netscope）的 JavaScript 和微软公司（Microsoft）的 VBScript。

JavaScript 是基于 Netscape 浏览器的，类似于 Java 编程语言的脚本语言，并且是一种基于对象的，面向 Internet 或 Intranet 的编程语言，使用它可以开发关于 Internet 或 Intranet 客户端和服务器的应用程序，也可以方便地嵌入到计算机文件中。由于 JavaScript 是第一个在 WWW 上使用的脚本语言，因而一度是最流行的 Web 站点脚本语言，用它可以方便地编排 HTML 网页，同时还可以控制动态 HTML。

VBScript 是 Microsoft 公司在 Visual Basic 编程语言的基础上设计的，由于其在企业界广为流行，且与 Microsoft 公司的其他产品有着密切的联系，VBScript 的使用范围越来越广，逐渐成为一种主要的脚本语言。

下面将着重介绍 JavaScript 脚本语言的语法基础，掌握了这些语法基础，再加上对动态网页对象模型的了解，就可制作出更加精彩的网页。

2. JavaScript 的产生与发展

JavaScript 语言起初并不叫此名称，它的早期是 Netscape 的开发者们称之为“Mocha”的语言，开始在网上进行β测试（由软件的多个用户在其实际的使用环境下进行的测试叫β测试）时，名字改为 LiveScript，Sun 公司推出 Java 之后，Netscape 引进了 Sun 的有关概念，在其发行 Netscape 2.0β测试版时才称其为 JavaScript。它不仅支持 Java 的 Applet 小程序，同时向 Web 页的制作者提供一种嵌入 HTML 文档进行编程的，基于对象的 Script（脚本）程序设计语言，采用的许多结构与 Java 相似。

支持 JavaScript 的 Navigator 2.0 的网络浏览器能够解释并执行嵌在 HTML 中的用 JavaScript 语言书写的“程序”。JavaScript 具有采用 CGI/PERL 编写的 Script 的能力，其优点是可以引用主机资源，响应位于服务器 Web 页中相应语法元素要完成的功能，而又不与主机服务器进行交互会话。

嵌入 HTML 文档中的 JavaScript 源代码，实际上是作为 HTML 文档 Web 页的一部分存在的。在用户使用 Netscape 2.x 浏览器浏览有 JavaScript 源代码的 HTML 文档页时，由浏览器本身对该 HTML 文档进行分析、识别、解释并执行用 JavaScript 编写的源代码（用户可以使用查看 HTML 源代码的功能看到 JavaScript 源代码的存在）。

3. JavaScript 编程特点

JavaScript 的编程工作复杂与否，和 HTML 文档所提供的功能大小密切相关，用下面的简单的程序代码来介绍它的编程特点。

```
<html>
    <head>
        <title>JavaScript 测试</title>
    </head>
    <body>
        你好
```

```
        <script language=javascript >
            document.write("hello, JavaScript!")
    </script>
    </body>
</html>
```

显示结果如图 8-18 所示。

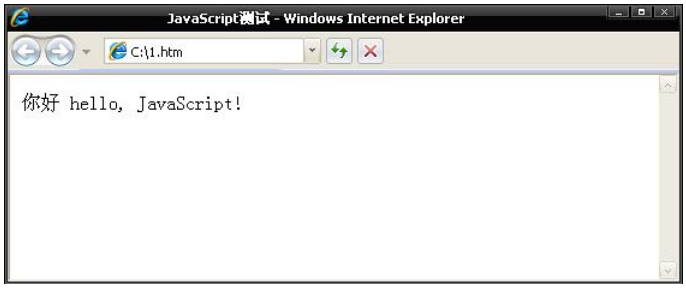

图 8-18　一个 JavaScript 的例子

从上面的程序代码中可以发现，JavaScript 源代码被嵌在一个 HTML 文档中，而且可以出现在文档头部（HEAD 块）和文档体部（BODY 块）。SCRIPT 标记的一般格式为：

```
<script language=javascript >
<!--
    JavaScript 语句串……
-->
</script>
```

为了使老版本的浏览器（即 Navigator 2.0 版以前的浏览器）避开不识别的 JavaScript 语句串，用 JavaScript 编写的源代码可以用注释括起来，即使用 HTML 的注释标记<!--…-->，而 Navigator 2.x 可以识别放在注释行中的 JavaScript 源代码。

说明：SCRIPT 标记可声明一个脚本程序，language 属性声明该脚本是一个用 JavaScript 语言编写的脚本。在<script>和</script>之间的任何内容都视为脚本语句，会被浏览器解释执行。在 JavaScript 脚本中，用“//”作为行的注释标注。

8.3.2　JavaScript 语言

1. JavaScript 数据类型

在 JavaScript 中，数据类型的使用是十分宽松的，程序员在声明变量时可以不指定该变量的数据类型，JavaScript 会自动按照用户给该变量赋的初值来确定适当的数据类型。这一点和 Java 或 C++是截然不同的。JavaScript 有以下几种基本的数据类型。

（1）数字类型。如 34，3.14 表示为十进制数；

034 表示为八进制数，用十进制表示其值为 28；

0x34 表示为十六进制数，用十进制表示其值为 52。

（2）字符串类型。如："Hello!"。

（3）逻辑值类型。取值仅可能是“真”或“假”，用 True 或 False 来表示。

（4）空值。当定义一个变量并且没有赋初值时，则该变量为空值。如 var ch1，此时 ch1

就为空值，并且不属于任何一种数据类型。

2. JavaScript 变量

JavaScript 变量的定义要求与 C 语言相仿，例如，以字母或下划线开始 ，变量不能是保留字（如 int，var 等），不能使用数字作为变量名的第一个字母等。但它的定义方法与 C 语言有很大的差别。

C 语言的变量定义语法格式为：

int a=1;　　float f1=3.14;

而 JavaScript 的变量定义的语法格式为：

var 变量名；　　或者　var 变量名=初始值；

JavaScript 并不是在定义变量时来说明变量的数据类型，而是在给变量赋初值时来确定该变量的数据类型；JavaScript 对字母的大小写是敏感的。如 var my；和 var My；JavaScript 认为这是两个不同的变量。

说明：在使用变量之前，最好对每个变量使用关键字 var 进行变量声明，防止发生变量的有效区域冲突的问题。

3. JavaScript 常量

JavaScript 常量分为 4 类：整数、浮点数、布尔值和字符串。下面分别加以说明。

（1）整数常量。在 JavaScript 中，整数可以表示为以下几种类型。

1）十进制数：即一般的十进制整数，它前面不可有前导 0，如 75。

2）八进制数：以 0 为前导，表示八进制数，如 075。

3）十六进制数：以 0x 为前导，表示十六进制数，0x0F。

（2）浮点数常量。浮点数可以用一般的小数格式来表示，也可以使用科学计数法来表示。如 7.54343、3.0e9。

（3）布尔型常量。布尔型常量只有两个值：True 和 False。

（4）字符串常量。字符串常量是由单引号或双引号括起来的 0 个或多个字符组成。如"Test String"\"12345"。

4. JavaScript 语句的结构

在 JavaScript 的语法规则中，每一条语句的最后必须使用一个分号，这一点与 C 语言、C++语言的语法规则是相同的。例如：

document.write("kkk");　　//此语句的功能是在浏览器中输出“kkk”字符串

另外，在编写 JavaScript 程序时，一定要养成一个良好的习惯：一行写一条语句，如果使用复合语句块时，注意把复合语句块用大括号括起来，并且根据每一句作用范围的不同，应有一定的缩进。一个好的编写程序的风格，对于程序的调试和阅读都是非常有好处的。另外一个好的编程风格也要适当加一些注释。例如：

```
<script language=javascript>
<!--
    document.write("switch 语句测试------");
    SUM=0;                    //初始化累加和
    P=1;
    for (i=1; i<100; i++)
        {
```

```
            SUM+=i;                    //求累加和
            P*=i;
        }
    document.write(SUM,P);
//-->
```

5. JavaScript 运算符和表达式

JavaScript 拥有一般编程语言（如 C 语言）的运算符，包括算术运算符、比较运算符等，下面将逐一介绍。

（1）算术运算符。用于连接运算表达式的各种算术运算符如表 8-7 所示。

表 8-7 算术运算符

运算符	运算符定义	举例	说明
+	加法符号	X=A+B	
–	减法符号	X=A–B	
*	乘法符号	X=A*B	
/	除法符号	X=A+B	
%	取模符号	X=A%B	X 等于 A 除以 B 所得的余数
++	加 1	A++	A 的内容加 1
––	减 1	A––	A 的内容减 1

（2）位操作运算符。位操作运算符对两个表达式相同位置上的位进行位对位运算。JavaScript 支持的位操作运算符如表 8-8 所示。

表 8-8 位运算符

运算符	运算符定义	举例	说明
~	按位求反	X=~A	
<<	左移	X=B<<A	A 为移动次数，左边移入 0
>>	右移	X=B>>A	A 为移动次数，右边移入 0
>>>	无符号右移	X=B>>>A	A 为移动次数，右边移入符号位
&	位“与”	X=B & A	
∧	位“异或”	X=B∧A	
|	位“或”	X=B | A	

（3）复合赋值运算符。复合赋值运算符执行的是一个表达式的运算。在 JavaScript 中，合法的复合赋值运算符如表 8-9 所示。

表 8-9 复合赋值运算符

运算符	运算符定义	举例	说明
+=	加	X+=A	X=X+A
–=	减	X–=A	X=X–A
*=	乘	X*=A	X=X*A
/=	除	X/=A	X=X/A

续表

运算符	运算符定义	举例	说明
%=	模运算	X%=A	X=X%A
<<=	左移	X<<=A	X=X<<A
>>=	右移	X>>=A	X=X>>A
>>>=	无符号右移	X>>>=A	X=X>>>A
&=	位“与”	X&=A	X=X&A
∧=	位“异或”	X∧= A	X=X∧A
|=	位“或”	X|=A	X=X|A

（4）比较运算符。比较运算符用于比较两个对象之间的相互关系，返回值为 True 和 False。各种比较运算符如表 8-10 所示。

表 8-10　比较运算符

运算符	运算符定义	举例	说明
==	等于	A==B	A 等于 B 时为 True
>	大于	A>B	A 大于 B 时为 True
<	小于	A<B	A 小于 B 时为 True
!=	不等于	A!=B	A 不等于 B 时为 True
>=	大于等于	A>=B	A 大于等于 B 时为 True
<=	小于等于	A<=B	A 小于等于 B 时为 True
?:	条件选择 E？A:B	E 为 True 时选 A，否则选 B	

（5）逻辑运算符。逻辑运算符返回 True 和 False，其主要作用是连接条件表达式，表示各条件间的逻辑关系。各种逻辑运算符如表 8-11 所示。

表 8-11　逻辑运算符

运算符	运算符定义	举例	说明
&&	逻辑“与”	A && B	A 与 B 同时为 True 时，结果为 True
!	逻辑“非”	!A	如 A 原值为 True，结果则为 False
||	逻辑“或”	A || B	A 与 B 有一个取值为 True 时，结果为 True

（6）运算符的优先级（如表 8-12 所示）。

表 8-12　运算符的优先级（由高到低）

运算符	说明
. 、[]、()	字段访问、数组下标及函数调用
++、--、~、!、typeof、new、void、delete	一元运算符、返回数据类型、对象创建、未定义值
*、/、%	乘法、除法、取模
+、-、+	加法、减法、字符串连接
<<、>>、>>>	移位

续表

运算符	说明
<<=、>>=	小于、小于等于、大于、大于等于
==、!==	等于、不等于
&	按位与
^	按位异或
|	按位或
&&	逻辑“与”
||	逻辑“或”
?:	条件
=	赋值

（7）表达式。JavaScript 表达式可以用来计算数值，也可以用来连接字符串和进行逻辑比较。JavaScript 表达式可以分为以下三类。

（1）算术表达式。算术表达式用来计算一个数值，如 2*4.5/3。

（2）字符串表达式。字符串表达式可以连接两个字符串，例如："hello" + "world! "，该表达式的计算结果为"helloworld! "。

（3）逻辑表达式。逻辑表达式计算结果为一个布尔型常量（True 或 False）。例如：12>24，其返回值为 False。

6. 脚本语言的注释

JavaScript 允许加一些注释。有两种注释方法：单行注释和多行注释。其中单行注释：以“//”开始，以同一行的最后一个字符作为结束符；多行注释：以“/*”开始，以“*/”结束，符号“*/”可放在同一行或不同的行中。

下面的程序代码中使用这两种注释方法：

```
<Script language = "JavaScript">
    /*这是多行注释的第一行
      这是多行注释的第二行*/
    k=24*7;        //这是一个单行注释的例子
</Script>
```

从上面的程序代码可以清楚地看出两种注释的使用方法，并且也可以看出 JavaScript 的语法大部分是与 C++语言相通的。

7. JavaScript 程序流程控制

JavaScript 的脚本语言同 C++语言是类似的，提供了相同的程序流程控制语句。这些语句分别是 if、switch、for、do 和 while 语句。

（1）条件语句。

（1）if 语句。If 语句是一个条件判断语句，它根据一定的条件执行相应的语句块，定义的语法格式如下：

```
if (expr)
    {
```

```
        code_block1
    }
else
    {
        code_block2
    }
```

这里，expr 是一个布尔型的值或表达式（特别强调：expr 一定要用小括号将其括起来），code_block1 和 code_block2 是由多个语句组成的语句块。当 expr 值为 Ture 时，执行 code_block1，当 expr 值为 False 时，执行 code_block2。另外有一点要说明的是，if 语句是可以嵌套的，即在 if 语句的模块中，还可以包含其他的 if 语句。例如：

```
if (expr)
    {
        code_block1
        if   (expr1) { code_block3 }
    }
else
    {
        code_block2
    }
```

（2）switch 语句。switch 语句测试一个表达式并有条件地执行一段语句，其语法格式如下：

```
switch (表达式) {
        case  值 1：code_block1
                    break;
        case  值 2：code_block2
                    break;
        case  值 3：code_block3
                    break;
        …
        default：    code_blockn
    }
```

switch 语句首先计算表达式的值，然后根据表达式所计算出的值来选择与之匹配的 case 后面的值，并执行该 case 后面的语句，直到遇到了一个 break 语句为止，如果所计算出的值与任何一个 case 后面的值都不相符的话，则执行 default 后的语句。

在代码清单 w8-18.htm 中用到了 switch 语句，其在浏览器中的显示结果如图 8-19 所示。

代码清单 w8-18.htm

```
<html>
<head>
<title>例 2-3 显示</title>
<script language=javascript>
<!--
    document.write("switch 语句测试------");
    switch (14%3) {
        case 0: sth="您好";
                    break;
```

```
            case 1: sth="大家好";
                        break;
            default: sth="世界好";
                        break;
            }
        document.write(sth);
//-->
</script>
</head>
<body>
</body>
</html>
```

图 8-19 switch 语句测试

从图 8-19 可以看出，执行的是 default 后的语句，因为表达式（14%3）的运行结果是 2。如果表达式改为 15%3，则浏览器中显示结果为“switch 语句测试------您好”。另外需要强调说明，在每一个 case 语句的值后都要加冒号。

（2）循环语句。有许多时候，需要把一个语句块重复执行多次，每次执行仅改变部分参数的值，这时可以使用循环语句，直到某一个条件不成立为止。

1）for 语句。for 语句用来产生一段程序循环，其语法格式如下：

```
for ( init;   test;   incre)
            {
            code_block
            }
```

这里 init 和 incre 是两个语句，test 是一个条件表达式。init 语句只执行一次，用来初始化循环变量。test 表达式在每次循环后都要被计算一次，如果其运算值为 False，则循环中止并立即继续执行 for 语句之后的语句，否则执行 code_block 语句块，循环完成后执行一次 incre 语句块。使用 break 语句可用来从循环中退出。for 语句一般用在已知循环次数的场合，而且 init、test、incre 三个语句之间要用分号隔开。

在代码清单 w8-19.htm 使用 for 语句，计算从数字 1 到 10 的累加和，其在浏览器中的运行结果如图 8-20 所示。

代码清单 w8-19.htm

```
<script language=javascript>
    var   sum=0;
```

```
    for(n=1;n<11;n++)
    {
        sum=sum+n
        document.write (n,"        SUM=",sum,"<br>");
    }
</script>
```

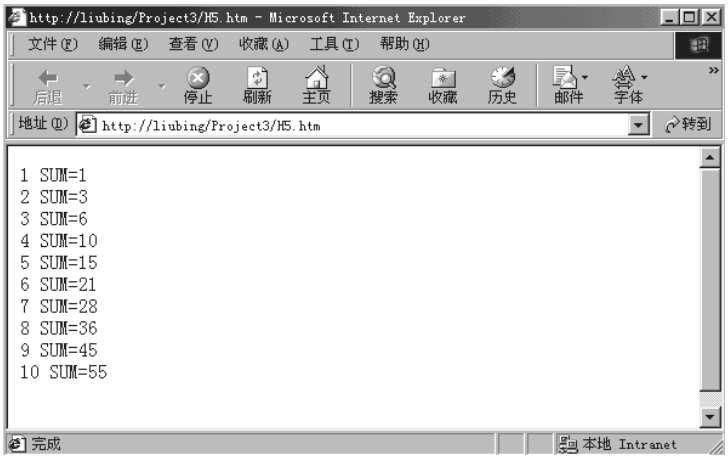

图 8-20　for 语句循环的例子显示结果

2）while 语句。对于有些程序，如果不知道其循环体要执行多少次时，就不能使用 for 循环语句了。这时可以考虑使用 while 语句，while 语句也是产生一段程序的循环，其语法格式如下：

```
while (expr) {
        code_block;
    }
```

这里，当表达式 expr 为 Ture 时，code_block 循环体被执行，执行完该循环体后，会再次判断表达式 expr 的运算结果是否为 True，以决定是否再次执行该循环体；如果 expr 开始时便为 Fasle，则语句块 code_block 将一次也不会被执行。使用 break 语句可从这个循环中退出。其实 while 语句非常好理解，只要知道“表达式为真则执行循环体”即可。现举例说明 while 语句的用法。

下面是通过代码清单 w8-20.htm 来说明 while 语句的用法，此程序仍然是计算从数字 1 到 10 之间的累加和，其在浏览器中的运行结果如图 8-21 所示。

代码清单 w8-20.htm

```
<script language=javascript>
<!--
    var i,sum;
    i=1;
    sum=0;
    while(i<=10){
        sum+=i;
        document.write(i,"     ",sum,"<br>") ;
```

```
            i++;
        }
//-->
</script>
```

图 8-21 while 语句的实例

3）do…while 语句。do…while 语句与 while 语句所执行的功能完全一样，唯一的不同之处就是 do…while 语句不管条件是否成立，其循环体至少执行一次，然后再去判断表达式的取值是否为真。do…while 语句的语法格式如下：

```
do{
    code_block
} while (expr) ;
```

这里，无论表达式 expr 的值是否为 True，code_block 循环体都被执行，即语句块 code_block 至少执行一次。另外，使用 break 语句可从循环中退出。下面用代码清单 w8-21.htm 来说明当条件并不成立时，循环体仍被执行一次的情况，其在浏览器中的显示结果如图 8-22 所示。

代码清单 w8-21.htm

```
<script language=javascript>
<!--
    var i,sum;
    i=1;
    sum=0;
    do{
        sum += i;
        document.write (i,"    ",sum*100,"<br>") ;
        document.write ("i 小于 10 条件不成立,但本循环体却执行一次!");
        i++;
    } while (i>10)
//-->
</script>
```

图 8-22　do…while 语句实例

（3）转移语句。

1）break 语句。break 语句的作用就是使程序跳出各种流程。它常常是用在异常情况下终止流程。在循环体中，可以使用多个 break 语句，一个 break 语句只会影响离它最近的循环。但是最好不要过多使用 break 语句，否则程序运行结果将难以预料。

2）continue 语句。有时，在循环语句中，在某个特定的情况下，希望不再执行下面的循环体，但是又不想退出循环，这时就要使用 continue 语句。在 for 循环中，执行到 continue 语句后，程序立即跳转到迭代部分，然后到达循环条件表达式，而对 while 循环，程序立即跳转到循环条件表达式。

8.3.3　JavaScript 的函数

1．JavaScript 函数概述

把相关的语句组织在一起，并给它们标注相应的名称，利用这种方法可以把程序分块，这种形式的组合就称为函数，用参数往函数中传递信息，有些函数不需要任何参数，有些函数则可以带多个参数。函数的定义方法如下：

```
function 函数名( [ 参数 ] [，参数] ){
        函数语句块
    }
```

下面通过代码清单 w8-22.htm 来了解 JavaScript 中函数的定义和调用方法，在浏览器中的显示结果如图 8-23 所示。

代码清单 w8-22.htm

```
<html>
    <head>
        <title>一个 JavaScripte 程序测试</title>
        <script language=javascript>
        <!--
            function total (i,j) {   //声明函数 total，参数为 i,j
                                     var sum;          //定义变量 sum
                                     sum=i+j;          //i+j 的值赋给 sum
```

```
                return(sum);        //返回 sum 的值
            }
        document.write("调用这个函数 total(100,20) ,结果为:", total(100,20) )
        //-->
        </script>
    </head>
    <body>
    </body>
</html>
```

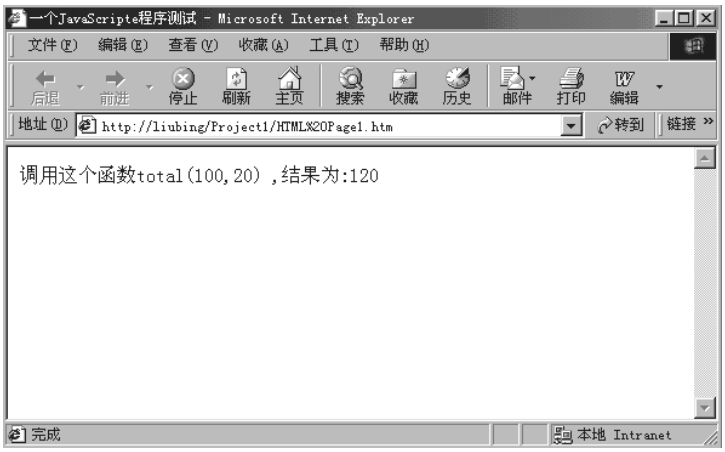

图 8-23 函数的定义与调用实例

在代码清单 w8-22.htm 中，定义了一个函数 total(i,j)，它有两个形式参数 i 和 j，当调用这个函数时，可以给函数中的形参 i 和 j 一个具体的值，例如：total(100,20)，变量 i 的值为 100，变量 j 的值为 20。

由此可以看出，函数通过名称调用。函数可以有返回值，但并不是必需的。如果需要函数返回一个值时，就使用语句 return(表达式)。

2. 内部函数

在面向对象编程语言中，函数一般是作为对象的方法来定义的。而有些函数由于其应用的广泛性，可以作为独立的函数定义，还有一些函数根本无法归属于任何一个对象，这些函数是 JavaScript 脚本语言所固有的，并且没有任何对象的相关性，称为内部函数，由于篇幅限制不能一一讲述，在此仅通过一个例子来说明。例如，内部函数 IsNaN，用于测试某个变量是否是数值类型。如果变量的值不是数值类型，则返回 True，否则返回 False。

下面通过一个例子来说明，当用户在浏览器的输入对话框中输入一个值，如果输入的值不是数值类型时，则给用户一个提示，当用户输入的值是数值类型时，也同样给出一个提示。这个例子的源代码如下：

```
<script language=javascript>
<!--
    var str;
    str = prompt ("请你输入一个值,如 3.14" , "");
    if ( isNaN ( str ) )
```

```
        {
        document.write("唉？受不了您,有例子都输不对!!!");
        }
    else
        {
        document.write("您真棒,输入正确(数值类型)!!!");
        }
//-->
</script>
```

在上例程序的执行过程中，首先要求用户输入一个数值，如图 8-24 所示，然后对用户的输入进行判断，如果输入的是数值类型，则在浏览器中显示结果如图 8-25 所示，如果输入的是其他类型数据，则在浏览器中显示的结果如图 8-26 所示。

图 8-24　请求用户输入一个数值的对话框

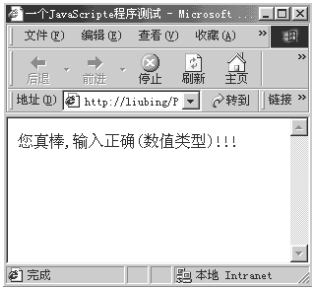

图 8-25　函数调用实例（1）

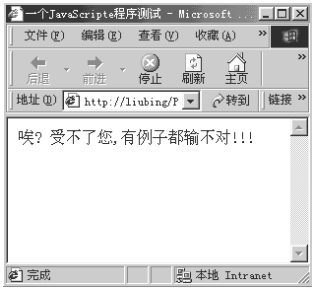

图 8-26　函数调用实例（2）

3. 用户自定义函数

在 JavaScript 中，用户可以定义自己的函数，如代码清单 w8-23.htm 所示，其在浏览器中的显示结果如图 8-27 所示。

代码清单 w8-23.htm

```
<html>
    <head>
        <title>This is a function's test</title>
        <script language=”JavaScript”>
            function square ( i ){
                document.write ("The call passed",i,"to the square function.","<br>")
                return i*i
            }
          document.write ("The function re-turned","<br>")
          document.write(square(8))
    </script>
    </head>
```

```
    <body>
        <br>
        All done.
    </body>
</html>
```

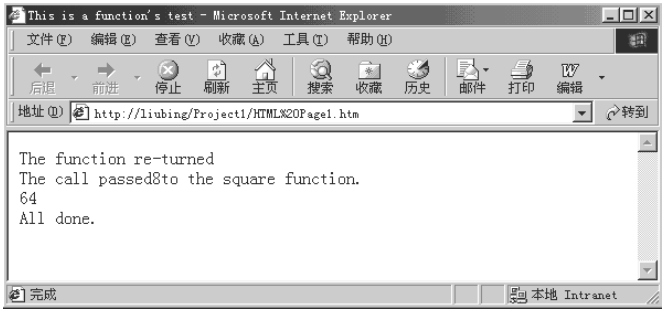

图 8-27 用户自定义函数

从执行结果可以看出，一个函数定义时并不发生作用，只有在引用时（函数定义后的 document.write 语句）才被激活。

8.3.4 JavaScript 的事件

JavaScript 事件

JavaScript 语言是一个事件驱动的编程语言。事件是脚本处理响应用户动作的唯一途径，它利用了浏览器对用户输入的判断能力，通过建立事件与脚本的一一对应关系，把用户输入状态的改变准确地传给脚本，并予以处理，然后把结果反馈给用户，这样就实现了一个周期的交互过程。

JavaScript 对事件的处理分为定义事件和编写事件脚本两个阶段，可以定义的事件类型几乎影响到 HTML 的每一个元素，如浏览器窗口、窗体文档、图形、链接等。表 8-13 列出了事件类型和相关说明。

表 8-13 JavaScript 的事件列表

事件名称	事件说明
Abort	用户中断图形装载
Blur	元素失去焦点
Change	元素内容发生改变，如文本域中的文本和选择框的状态
Click	点击鼠标按钮或按键盘按键
Dragdrop	浏览器外的物体被拖到浏览器中
Error	元素装载发生错误
Focus	元素得到焦点
Keydown	用户按下一个键
Keypress	用户按住一个键不放
Keyup	用户将按下的键抬起

续表

事件名称	事件说明
Load	元素装载
Mousemove	鼠标移动
Mouseover	鼠标移过元素上方
Mouseout	鼠标从元素上方移开
Mousedown	鼠标按键按下
Mouseup	鼠标按键抬起
Move	帧或者窗体移动
Reset	表单内容复位
Resize	元素大小属性发生改变
Submit	表单提交
Select	元素内容发生改变，如文本域中的文本和下拉菜单中的选项
Unload	元素卸载

要使 JavaScript 的事件生效，必须在对应的元素标记中，指明将要发生在这个元素上的事件。例如，<Input type=text onBlur="kk()"onKeyup="mm()">，它在<Input>标记中定义了两个事件，一个是该文本框失去焦点时执行的 onBlur 事件，另一个是当键按下抬起后所发生的 Keyup 事件，即每当用户按下键盘并松开按键后，就触发 mm()脚本函数，每当该文本框失去焦点（用户把输入光标移动到其他位置）时，就触发 kk()脚本函数。

2. 为事件编写脚本

接下来要为这些事件编写处理的函数，这些函数就是脚本函数。脚本函数包含在<script>和</script>标记之间。下面通过代码清单 w8-24.htm，看看它是如何工作的。代码清单 w8-24.htm 的功能是建立一个按钮，当单击按钮后弹出一个对话框，对话框中显示“XX，久仰大名,请多多关照”。

代码清单 w8-24.htm

```
<html>
    <head>
        <title>一个 JavaScripte 程序测试</title>
        <script language=javascript>
        <!--
        function kkk(){
                {
                    username=prompt("请问您是何方神圣,报上名来","");
                }while (username=="")
                document.write(username,",久仰大名,请多多关照.");
        }
        //-->
        </script>
    </head>
    <body>
```

```
        <input type="button" value=你敢碰我吗? name=button1 onclick="kkk()">
    </body>
</html>
```

这个 HTML 页的起始界面如图 8-28 所示，上面仅有一个元素，即一个按钮。如果不设置任何事件，当单击该按钮后不会产生任何响应。但现在，定义了单击按钮的 onClick 事件，并把事件的处理权交给了脚本程序 kkk()。这是实现交互的第一步。

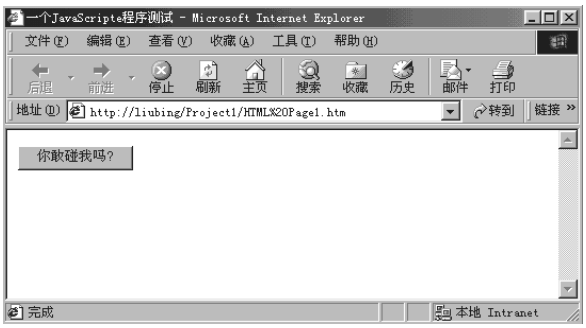

图 8-28　脚本实例的初始界面

当用户单击按钮后，浏览器中将出现一个如图 8-29 所示的 JavaScript 对话框，框中提示用户输入姓名。这时，只要输入名称并单击“确定”按钮，就可以看到浏览器的显示结果，如图 8-30 所示。

图 8-29　JavaScript 对话框

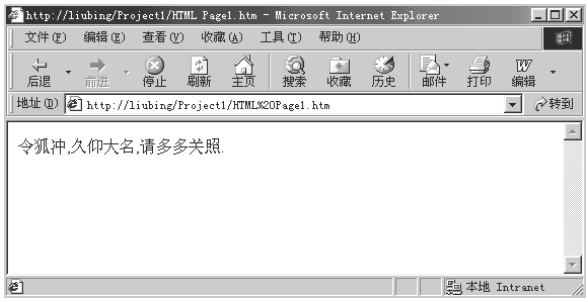

图 8-30　确认对话框后的输出

8.3.5　JavaScript 的对象

1. 基本概念

面向对象的系统包含三个要素：对象、类、继承。

JavaScript 语言是一种基于对象的语言，它不能算是一个面向对象的语言，因为它不支持

类和继承。下面来介绍一下对象的概念。

对象，从概念级上说，表示客观世界的实体，任何实物都可以被称为对象；从物理实现说，一个对象是一种状态和一系列可被外部调用的操作方法的一个封装体，即指的是状态和操作的组合，状态通过一组属性来确定，而操作通过一组方法来确定。

以一台饮料机为例来说明封装对象的概念。一台饮料机可以被认为是封装的，因为它的功能被密封在一个金属盒子内。它有两个方法：①制一杯水；②制一杯咖啡。

这个对象的状态由剩余的茶叶量、咖啡量、牛奶量、糖量和水量等属性给出。与这个对象的接口是由盒子前分别标以“咖啡”和“茶”的两个按钮提供，这两个按钮提供用户能够执行这个对象的方法。

一个对象的操作接口被限制在用户的需求上，而方法的实现，外部是不可见的，也就是说，具有了信息隐藏，这是封装的主要目的，它阻止非法的访问，因为金属盒阻止这台机器的用户（当然这个用户从未研究过饮料机这种复杂难懂的机器）改动这台机器的内部结构。操作接口的另一个很重要的方面是，它提供了一个对象的行为视图，即所知道的仅是这个对象提供了某种功能，除此之外，对其他细节一无所知。在处理某个复杂的问题时，这一点是很重要的，因为一旦实现了一个对象，了解这个对象的算法和数据结构的内部细节不再是重要的，重要的是要知道这个对象所提供的操作接口。

在 JavaScript 中，浏览器本身就是一个对象，浏览器的文本也是对象，文本中的表单也是对象，表单中的按钮仍然是对象，不同的按钮也可以是不同的对象。另外，在 JavaScript 中，一种对象类型是一个用于创建对象的模板，在这个模板中定义了对象的属性和方法。在 JavaScript 中一个新对象的定义方法如下：

对象的变量名 = new 对象类型（可选择的参数）

访问对象属性的语法如下：

对象的变量名.属性名

访问对象方法的语法如下：

对象的变量名.方法名（方法可选参数）

定义一个字符串对象（即 String 对象）：

```
var gamma;
gamma = new String("This is a string");   //定义一个字符串对象，对象名为 gamma
document.write (gamma.substr(5,2));     //使用对象的方法，本例是取子串的方法
document.write (gamma.length);          //使用对象的属性，本例是获得字符串的长度
```

2. String 对象

String 对象是 JavaScript 的内置对象，被封装了一个字符串。它提供了许多字符串的操作方法。

String 对象的唯一属性是 length。String 对象的方法如表 8-14 所示。

表 8-14　String 对象的方法及其功能

名称	功能
CharAt(n)	返回字符串的第 n 个字符
IndexOf(srchStr[,index])	返回第一次出现子字符串 srchStr 的位置，index 从某一指定处开始，而不从头开始。如果没有该子串，返回-1

续表

名称	功能
LastIndexOf(srchStr[,index])	返回最后一次出现子字符串 srchStr 的位置，index 从某一指定处开始，而不从头开始
Link(href)	显示 href 参数指定的 URL 的超链接
SubString(n1,n2)	返回第 n1 和第 n2 个字符之间的子字符串
ToLowerCase()	将字符转换成小写格式显示
ToUpperCase()	将字符转换成大写格式显示

下面通过代码清单 w8-25.htm 来说明对象的属性及方法的应用，其在浏览器中的显示结果如图 8-31 所示。

代码清单 w8-25.htm

```
<html>
    <head>
        <title>一个 JavaScript 对象的属性和方法的使用</title>
            <script language=javascript>
            <!--
                sth=new String("这是一个字符串对象");
                document.write ("sth='这是一个字符串对象'","<br>");
                document.writeln ( "sth 字符串的长度为:",sth.length, "<br>");
                document.writeln ( "sth 字符串的第 4 个字符为:'",sth.charAt (4),"'<br>");
                document.writeln ( "sth 字符串的第 2 到第 5 个字符为:'",sth.substring (2,5),"'<br>");
                document.writeln ( sth.link("http://www.lllbbb.com"),"<br>");
            //-->
            </script>
    </head>
    <body>
    </body>
</html>
```

图 8-31 String 对象的属性和方法的应用实例

3. Array 对象

数组是一个有相同类型的有序数据项的数据集合。在 JavaScript 中的 Array 对象允许用户创建和操作一个数组，它支持多种构造函数。数组下标从零开始，所建的元素拥有从 0 到 size−1

的索引。在数组创建之后，数组的各个元素都可以使用[ ]标识符进行访问。Array 对象的部分方法如表 8-15 所示。

表 8-15　Array 对象的部分方法

方法	说明
Concat(array1,array2)	方法返回一个包含 array1 和 array2 级联的 Array 对象
Reverse( )	把一个 Array 对象中的元素在适当位置进行倒转
Pop()	从一个数组中删除最后一个元素并返回这个元素
Push()	添加一个或多个元素到某个数组的后面并返回添加的最后一个元素
Shift()	从一个数组中删除第一个元素并返回这个元素
Slice(start,end)	返回数组的一部分。从 start 到最后一个元素来创建一个新数组
Sort()	排序数组元素，将没有定义的元素排在最后
Unshift()	添加一个或多个元素到某个数组的前面并返回数组的新长度

4. Math 对象

Math 对象所提供的属性和方法在进行数学运算时非常有用。它有很多的方法和属性，如 sin()，cos()，abs()，PI，max()，min()等是用于计算的数学函数。用法如下：

```
<script language=javascript>
    <!--
        document.write (Math.PI); //取得 3.1415926
        document.write (Math.random());//产生一个 0 到 1 之间随机数
    //-->
    </script>
```

5. Date 对象

Date 对象提供了几种获取日期和时间的方法。定义 Date 对象的方法如下：

```
var  d1= new Date();
```

一旦定义了该对象，则提供了很多种方法。利用这些方法可以在网页上制作出很多漂亮的效果，而且这些效果都很新奇。例如，在网页上显示今天的年月日，计算用户在本网页上的逗留时间，在网页上显示一个电子表，网上考试的计时器等。在表 8-16 中列出了 Date 对象的方法。

表 8-16　Date 对象的方法

方法	说明
GetDate()	返回在一个月中的哪一天（1~ 31）
GetDay()	返回在一个星期中的哪一天（0~ 6），其中星期天为 0
GetHours()	返回在一天中的哪一个小时（0~ 23）
GetMinutes()	返回在一小时中的哪一分钟（0~ 59）
GetSeconds()	返回在一分钟中的哪一秒（0~ 59）
GetYear()	返回年号
SetDate(day)	设置日期
SetHours(hours)	设置小时数
SetMinutes(mins)	设置分钟数

续表

方法	说明
SetSeconds(secs)	设置秒
SetYear(year)	设置年

下面给出一个在浏览器显示当前日期和时间的示例，如代码清单 w8-26.htm 所示，在浏览器中的显示结果如图 8-32 所示。

代码清单 w8-26.htm

```
<html>
    <head>
        <title>一个 Date 对象的属性和方法的使用</title>
        <script language=javascript>
        <!--
            Stamp = new Date();
            document.write('<font  size="2" ><B>' + Stamp.getYear()+"年"+(Stamp.getMonth() + 1) +"月
"+Stamp.getDate()+ "日"+ '</B></font><BR>');
            var Hours;
            var Mins;
            var Time;
            Hours = Stamp.getHours();
            if (Hours >= 12)
                    { Time = "  下午"; }
            else
                    { Time = "  上午"; }
            if (Hours > 12) { Hours -= 12;}
            if (Hours == 0) { Hours = 12;}
            Mins = Stamp.getMinutes();
            if (Mins < 10) { Mins = "0" + Mins; }
            document.write('<font size="2"><B>' + Time + Hours + ":"+ Mins    + '</B></font>');
        //-->
        </script>
    </head>
    <body>
    </body>
</html>
```

图 8-32　Date 对象的方法使用示例

6. 浏览器和 HTML 对象

使用脚本语言离不开 HTML 对象模型，否则脚本语言只能作为一种退化的编程语言，并不能在 Web 应用中发挥它的强大功能。脚本语言和 HTML 对象模型结合在一起，才有可能使 Web 更具吸引力。

（1）什么是 HTML 对象模型。HTML 对象模型定义了表达网页及其元素的对象。这种技术形成了支持动态 HTML 的基础。对象模型以事件、属性和方法定义了一组对象，用户可以用来创建应用或为应用编写脚本。这些对象都按一定的层次组织。如图 8-33 所示的对象模型是一个由对象组成的层次结构。

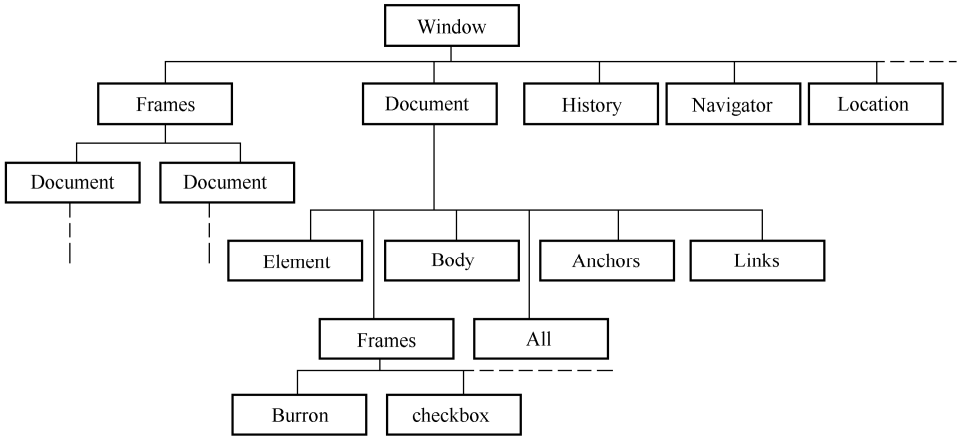

图 8-33　HTML 对象模型

顶层是 Window 对象，代表浏览器的窗口。对于其他的对象，如实际存在的当前文档（Document）、超链接（Links）、文档的锚点（Anchors），以及其他被显示文档的地址都可以作为窗体对象的属性被访问。Window 对象包括了对其他 6 个对象的引用：Document、History、Location、Navigator、Screen 和 Event。例如，windows.document.write("hello, world!")。

第二层对象可以是框架结构（Frames）或者文档对象（Document）。其中框架结构的每一个 Frame 对象中都包含一个文档对象。而文档对象可包括以下对象。

1）History 对象：文档的历史记录（曾经访问过该文档的 URL 地址记录清单）。

2）Location 对象：当前文档所在的位置（URL 地址、文件名，以及与当前文档位置有关的其他属性）。

3）Navigator 对象：返回浏览器被使用的信息。

动态 HTML 对象模型中的对象具有一般对象化编程语言的特性，有属性、方法和事件。例如，Document 对象有它自己的属性，并且有些属性的本身也是对象。除了 HTML 元素（文本和图像）以外，每个文档都包含一些可编程的对象，如锚点或超链接和窗体。借助这些对象可以访问当前文档的超链接和锚点。

（2）Window 对象。Window 对象封装了当前浏览器的环境信息。一个 Window 对象中可能包含几个 Frame（框架）对象。每个 Frame 对象在它所在的框架区域内作为一个根基，相当于整个窗口的 Window 对象。

Document 对象封装了当前文档：

1）History 对象封装浏览器历史记录清单；

2）Location 对象封装浏览器当前位置；

3）Navigator 对象提供客户环境的信息；

4）Screen 对象访问显示器屏幕参数；

5）Event 对象提供最新事件信息及控制事件处理。

下面详细介绍 Window 对象的属性、方法和事件。

1）Window 对象的属性。广义的 Window 对象包括浏览器的每一个窗口，每一个框架（Frame）或者活动框架（IFrame）。每个 Window 对象都有以下一些属性。

2）Name：这是 Window 对象的一个可读写属性，它返回当前窗口的名称。用 Window 对象的 Open 方法可以赋予窗口一个属性，从超链接中也可以取得窗口名字。下面的超链接将打开一个 Name 属性为“IE-Window”的 Window 对象：

```
<a href ="http://Master/MySite/example.htm"    target="IE-Window">
        example
 </a>
```

当单击这个超链接时，会打开一个新的浏览器窗口且名字叫“IE-Window”。

2）Parent：这是 Window 对象的一个只读属性，如果当前窗口有父窗口，它返回当前窗口的父窗口的对象，可以使用返回对象的属性和方法。

3）Opener：这是 Window 对象的一个只读属性，属性返回产生当前窗口对象，可以使用返回对象的属性和方法。

4）Self：这是 Window 对象的一个只读属性，属性返回当前窗口的一个对象，可以通过这个对象访问当前窗口的属性和方法。

5）Top：这是 Window 对象的一个只读属性，属性返回的是代表最上层窗口的一个对象，可以通过这个对象访问当前窗口的属性和方法。

6）DefaulItStatus：这是 Window 对象的一个可读写属性，使用它可以返回或者设置将在浏览器状态栏中显示的默认内容。

7）Status：这是 Window 对象的一个可读写属性，使用它可以返回或者设置将在浏览器状态栏中显示的内容。例如，在浏览器状态栏中显示当天的日期：

```
Status=DataFormat(Date)
```

2）Window 对象的方法。

① Alert：使用 Alert 方法可以弹出一个警告框，警告框显示一条信息，并且有一个“确定”按钮。例如，window.alert("这次你可真走运!")，其在浏览器中的显示结果如图 8-34 所示。

图 8-34 Alert 警告框

② Confirm：使用 Confirm 方法可以弹出一个对话框，显示一条信息，并且显示“确定”和“取消”两个按钮。它能返回一个逻辑布尔量的值，可以被脚本程序使用。例如

```
<script language= JavaScript>
    <!--
```

```
        Res = window.confirm("您有勇气确认吗?");
        if (Res) {document.write("您真勇敢!")}
            else {document.write("您太年轻,还需要锻炼!")}
    //-->
</script>
```

首先请用户进行选择，如图 8-35 所示，如果用户单击“确定”按钮，则显示如图 8-36 所示的浏览器窗口，如果用户单击“取消”按钮，则显示如图 8-37 所示的浏览器窗口。

图 8-35　Confirm 对话框

图 8-36　Confirm 实例

图 8-37　Confirm 实例

3）Prompt：用 Prompt 方法可以弹出一个信息框，显示一条信息，并且有一个文本输入框，一个“确定”按钮和一个“取消”按钮。如果选择“确定”按钮，则文本框中输入的内容将被返回，可以被脚本程序使用。这个方法有两个参数：第一个是要在对话框中显示的信息；第二个是文本输入框内默认显示的内容。例如，Str=window.prompt("有胆子报上名来!",""),其在浏览器中的显示结果如图 8-34 所示。在 Prompt 对话框中，如果单击“确定”按钮，将向变量 Str 返回当前文本输入框内的字符串；如果单击“取消”按钮，将不执行任何操作。

图 8-34　Prompt 对话框

④ Open：这种方法可以建立一个新的窗口，它可以使用许多参数。第一个参数是要在新窗口中打开文件的 URL 地址，这个参数是必需的；第二个参数是 Target，即打开文件窗口的名字；随后的参数都是对新窗口属性的描述。例如，要打开一个没有工具条、定位框和目录框的窗口，这个窗口中显示“Search.htm ”，可以使用语句：window.open("Search.htm ","kkk" , "toolbar=no location=no")。

⑤ Close：这种方法用来关闭一个窗口。例如，window.close()，这行代码将关闭当前窗口。

⑥ SetTimeout：这是 Window 对象的一个方法。这种方法用来设置一个计时器，该计时器以毫秒为单位，当所设置的时间到时，会自动调用一个函数。SetTimeout 方法可以使用三个参数：第一个参数用来指定设定时间到后调用函数的名称；第二个参数用来设定计时器的时间间隔；第三个参数用来指定函数使用的脚本语言类型（JavaScript 或 VBScript）。在代码清单 w8-27.htm 中使用 SetTimeout 方法，在文本框中显示一个电子表，而其在浏览器中的显示结果如图 8-35 所示。

代码清单 w8-27.htm

```
<html>
    <head>
        <title>一个 JavaScript 计时器的应用</title>
        <script language = JavaScript>
        <!--
            var flag;
            interval=1000;
            function change()   {
            var today = new Date();
            text1.value = today.getHours() + ":" + today.getMinutes() + ":" + today.getSeconds();
            timerID=window.setTimeout("change()",interval);
            }
        //-->
        </script>
    </head>
    <body onload="change()">
        <INPUT id=text1 name=text1>
    </body>
</html>
```

图 8-35　脚本计时器的应用

timerID=window.setTimeout("change()",interval)这行代码可以创建一个计时器，每一秒钟调用“change()”子函数一次，这个函数使用的脚本是 JavaScript 脚本。在设置计时器时，创建了一个计时器对象，它的句柄是“timerID”，以后可以对这个对象进行操作。

ClearTimeout：这种方法用来清除一个计时器，例如，Window.clearTimeout timerID。清除名字为“timerID”的计时器对象。

3）Window 对象的事件。

在脚本模型中，对象都有自己的事件。大多数的对象的事件都是相同的，它们都是浏览器中的一些事件，这些事件有 onBlur、onDblclick 、onFocus、onKeydown、onKeyup、onMousemove、onMouseover、onSelectstart、onClick、onDragstart、onHelp、onkeypress、onMousedown、onMouseout、onMouseup 等。可以为这些对象事件编写事件处理程序，当事件被激活时，事件处理程序被执行。

Window 对象包含上面讲到的大多数对象的事件，这里就不一一详细介绍，只介绍两个 Window 对象特有的事件：OnLoad 事件和 OnUnload 事件。

① OnLoad：Window 对象的 OnLoad 事件在分析完 HTML 文件的所有代码内容后被激活。可以使用这个对象事件在网页加载时执行一定的任务。例如，可以在网页被加载时同时加载一个广告页，脚本程序代码如下：

```
<html>
    <head>
        <title>一个 JavaScripte 程序测试</title>
        <script language=javascript>
        <!--
            function kkk(){
            window.open("H1.htm", "", " toolbar=no,menubao=no")
            }
        //-->
        </script>
    </head>
    <body onLoad = "kkk()">
        <A href="http://www.whpu.com/e1.htm">test</a>
    </body>
</html>
```

② OnUnload：在窗口被卸载时，也就是离开当前浏览窗口时，事件内容被激活。也可以在网页被卸载时同时加载一个广告页，脚本程序代码如下：

```
<script language=javascript>
<!--
        function kkk(){
        window.open("H1.htm", "", " toolbar=no,menubao=no")
        }
//-->
</script>
<body onUnload="kkk()">
```

在<body>标记中，加上事件的触发。

（3）Document 对象。Document 对象指的是在浏览器窗口中显示的 HTML 文档。Document 对象的属性，简单的，如文档的背景、文档字体的颜色等；复杂的，如各种链接和锚的结合体，Form 以及 ActiveX 控件等。

Document 对象提供了一些强有力的方法，使得可以在文档中直接传送 HTML 语句。Document 对象作为 Window 对象下的一个对象，可以利用“Window.document”访问当前文档的属性和方法，如果当前窗体中包含框架对象，可以使用表达式“Window.frames(n).document”来访问框架对象中显示的 Document 对象，式中的“n”表示框架对象在当前窗口的索引号。

1）Document 对象的属性。

① Linkcolor：用来设置当前文档中超链接显示的颜色，例如，window.document.linkcolor="red"。

② Bgcolor 和 Fgcolor：这两个属性分别用来读取或者设置 Document 对象所代表的文档的背景色和前景色。使用方法与 Linkcolor 属性使用方法相同。它们可以被任意地设置和更改。

③ Title：是 Document 对象的一个只读属性，它返回当前网页的标题。

④ LastModified：是 Document 对象的一个只读属性，它返回当前网页最近一次被修改的时间。

⑤ All 属性：是一个对象的序列，它是当前文档中的所有 HTML 标记组成的对象序列。当前窗口中的文档对象的第一个 HTML 标记是“Document.all(0)”。可以使用 All 属性对象的属性和方法，如 Document.all.length，将返回文档中 HTML 标记的个数。

2）Document 对象的方法。Document 对象提供了一些在脚本模式中强有力的方法。这些方法使得用户可以在脚本中建立显示在用户浏览器中的 HTML 文档。

① Write：Write 方法用于将一个字符串放在当前文档中，放入的内容将被浏览器所识别。如果是一般文本，将在页面显示；如果是 HTML 标记，将被浏览器解释。

② Open：Open 方法用于打开要输入的文档。当前文档的内容将被清除掉，而新的字符串可以通过 Write 方法放入当前文档。

③ Clear：Clear 方法用于清除当前文档中的内容，更新屏幕。

（4）Location 对象。Location 对象封装了窗口中显示的 URL 的信息。

1）Location 对象的属性。

① Href：Location 对象的 Href 属性可以返回或者设置页面完整的 URL 地址。例如，如下语句将把浏览器连接到武汉工业学院的主页：Document.Location.href =http://www.whpu.edu.cn/。这和使用 Window 对象的 Navigate 方法的效果是相同的。

② Host：Location 对象的 Host 属性可以返回网页主机名，以及所连接的 URL 的端口。

③ Protocal：这个属性用来返回当前使用的协议。例如，现在正在浏览器中访问 FTP 站点，那这个属性将返回字符串“ftp”。

2）Location 对象的方法。Location 对象支持以下三种方法。

① Assign：将当前 URL 地址设置为其参数所给出的 URL。

② Reload：重载当前网址。

③ Replace：用参数中给出的网址替换当前网址。

（5）History 对象。History 对象有一个唯一的只读属性：length。它可以返回历史记录表中的 URL 地址数目。利用这个属性可以帮助我们在历史记录表中进行搜索。History 对象的方法如下。

① Back：Back 方法指引浏览器在历史记录清单中向前移动。例如：window.history.back l，将指引浏览器跳向历史记录中的前一条记录。

② Forward：Forward 方法指引浏览器在历史记录清单中向后移动。例如，window.history.forward 2，指引浏览器跳向历史记录中后面的第二条记录。

③ Go：Go 方法指引浏览器跳向历史记录中的一条记录。例如，window.history.go 10，将指引浏览器跳向历史记录中的第 10 条记录。

本章小结

本章主要介绍标准的 HTML 语言，而且通过一些示例使读者对 HTML 语言有更进一步的认识。另外对 JavaScript 也做了比较详细介绍，这一部分对于一个从未学习过任何编程语言的读者来说可能有些困难，但要想使自己制作的网页更加引人入胜，JavaScript 是是最好的选择。

学好 HTML 和 JavaScript 语言，可了解网上非常精彩的网页是如何制作的，从而吸取别人的经验并把它应用到自己的网页制作中。

其实本章所介绍的只是一种客户端的网页语言，如果想要实现网上的交互，还必须要使用服务器端的脚本语言，目前常用的服务器端的脚本语言有 ASP.NET、PHP 和 JSP，如果想了解这些知识，请查阅相关的书籍。

习　题　8

一、填空题

1．HTML 文件是标准的_____文件，且其后缀名为_____或_____的文件。

2．HTML 文件由_____组成。

3．元素的起始标记叫做_____，元素的结束标记叫做_____，在这两个标记中间的部分是_____。

4．HTML 文件仅由一个_____元素组成，即文件以_____开始，以_____结尾，文件其余部分都是 HTML 的元素体。

5．<Hn>标题元素有 6 种，用于表示文章中的各种题目。字体大小从<H1>到<H6>顺序_____。

6．
用于_____。<p>表示一个_____开始。

7．用来修改字体和颜色的元素。其中 color 属性指定文字颜色，颜色的表示可以用 6 位十六进制代码，如<font color=#00FF00>，而其中的 00FF00 表示的含义是_____。

8．统一资源定位器（URL）的构成：_____。

9．在 HTML 文件中用链接指针指向一个目标。其基本的语法格式为：_____。

10．在 HTML 网页中，加图像是通过<img>标记实现的，它有几个较为重要的属性。其中：src 属性是用于_____；border 属性是用于_____。

11．使用框架结构可以使 Web 页的信息量_____。

12．HTML 提供的_____是用来将用户数据从浏览器传递给 Web 服务器的。

13．在 HTML 中提供表单的功能标记是_____。

14．HTML 中的_____标记是用来把表单中的文本框、按钮等引出的。

15．一个表格有一个_____，它表明表格的主要内容，并且一般位于表的上方；表格中由行和列分割成的单元叫做_____。

二、判断题

（　）1．脚本语言是一种简单的面向对象的语言。

（　）2．JavaScript 是第一个在 WWW 上使用的脚本语言。

（　）3．嵌入 HTML 文档中的 JavaScript 源代码实际上不是作为 HTML 文档 Web 页的一部分存在的。

（　）4．JavaScript 源代码被嵌在一个 HTML 文档中，而且只能出现在文档头部（HEAD 块）。

（　）5．在 JavaScript 中程序员在声明变量时可以不指定该变量的数据类型。

（　）6．JavaScript 语言是一种面向对象的语言。

三、简答题

1．HTML 的文件是由什么组成的？请给出一个标准的 HTML 文档的结构。

2．统一资源定位器（URL）的主要作用是什么？另外请写出其标准的结构形式。

3．请说明 JavaScript 脚本语言注释语句的方法。

4．说明下面程序的运行结果。

```
<html>
<head>
<title>程序测试结果</title>
<script language=javascript>
<!--
      document.write("switch 语句测试------");
      switch (8%3) {
            case 0: sth="您好";
                        break;
            case 1: sth="大家好";
                        break;
            default: sth="世界好";
                        break;
            }
      document.write(sth);
//-->
</script>
</head>
<body>
</body>
</html>
```

5．说明下面程序的运行结果。

```
<html>
    <head>
        <title>一个 JavaScripte 程序测试</title>
        <script language=javascript>
        <!--
          function mul (j) {
                    var sum,i;
                    sum=1;
                    for(i=1;i<j;i++){
                sum=sum*i;
                    }
                 return(sum);
                    }
            document.write("调用这个函数 mul(5) ,结果为:", mul(5) )
        //-->
```

```
        </script>
    </head>
    <body>
    </body>
</HTML>
```

四、程序题

1．请制作一个个人网站并发布到互联网上。

2．试用 JavaScript 制作一个乘法口诀表的 Web 页。

3．客户认证。请制作一个表单，其中包括内容：姓名、年龄、密码、确认密码、提交按钮和重新填写按钮。要求如下：

（1）各项内容在提交之前不能为空；

（2）姓名仅能输入 8 个字符；

（3）年龄必须输入数字型数据；

（4）密码和确认密码不能为空。

4．请编写一个在浏览器上显示当前日期和时间的程序。

如果用户的输入不符合以上要求，要能分别给出提示；如果达到以上要求，显示提示信息“恭喜您，您已经通过了我们的客户验证”。请上机练习，并给出相应的 HTML 源代码。