第 5 章　标记语言应用速成

HTML 是 Hyper Text Markup Language 的缩写。HTML 是一种超文本标记语言，是用来描述 WWW 上超文本文件的。

它主要是在原来文本文件的基础上，加一系列的标记符号描述其格式，形成网络文件，当用户使用浏览器下载文件时，就把这些标记解释成它应有的含义，按照一定的格式，将这些被标记的文件，显示在屏幕上。而 HTML 的标记符号并不显示在屏幕上。

利用 HTML 标记语言，可以将 INTERNET 上不同地区的服务器上的信息文件连接起来；有的标记是去连接另一个文件，有的是形成表格，有的是接收用户的信息等等。有了这些标记，用户在浏览器中看到的不再是呆板的纯文本，而是五彩缤纷的画面。

利用 HTML 语言也可以将声音文件、图像文件，甚至视频文件连接起来。此外，HTML 还可以与数据库中管理的数据相连接，满足读者的查询要求以及与用户交互的功能等等。

本章通过简明讲述 HTML 标记语法，目的是使 CGI 的程序员具备直接读写 HTML 源代码的能力。在脚本的开发中具备借鉴他人资源的阅读和理解能力。

5.1　HTML 的优点和局限性

5.1.1　优点

尽管使用 HTML 制作文件的时候，会觉得十分笨拙，但对于网络所提供的环境来说，HTML 有着非常明显的优势。

每一个 HTML 文件都不太大，它能够尽可能快地通过网络传输和显示，不需要加入字体或格式等其他控制信息。当然，如果加入那些修饰内容，文本显示会更美观，那将会降低文件的下载和显示速度，可以选择合适的方案进行设计。

HTML 文档是独立于平台的，它对多平台兼容。只要你有一个可以阅读和解释 HTML 的浏览器就能够在任何平台上阅读网络文件。

虽然 HTML 是一个标记性的语言，但是它学习起来非常容易，比任何一种计算机语言都简单易学。做一个 HTML 文件并不需要特殊的软件，只要一个字符编辑器就可以完成。

5.1.2　局限性

HTML 通过标记符号对文本的成分进行控制，它的基本特征就是各种标记符号，这些标记符号，为 HTML 提供了一些排版功能。当浏览器通过网络获取用 HTML 标记的文件时，

需要全本请访问 www.ertongbook.com

使用不同的浏览器阅读同一个文件，它们显示的形式是不同的。

注意：在设计网络文件时，既要考虑不同形式的浏览器，又要使文件结构清晰，内容易读易懂。

综上所述，HTML 提供的功能是有限的，它远不如一般的文字编辑软件那样直观而简便。用它建立的网页所包含的成分也是非常有限的，仅有标题、段落、少量的列表和一些图形可供选择使用。

CGI 脚本的执行与传送要依赖 HTML 标记语言的框架载体。CGI 脚本在 HTML 中的使用增强了客户端与服务器之间信息交流的互动性。在本章随后章节 CSS 语言基础速成和 DHTML 的实现两章讲解的内容中，我们将扼要讲述使用的交互式和动态 HTML 技术，这么做的目的是为了设计出丰富，极具个性的客户端和服务器页面。

5.2　HTML 文件基础

本节讲述：

- HTML 文件的基本构成
- 四个基本标记
- 文件的子标题
- 几个常用的 HTML 标记

先让我们来看如下代码：

```
<HTML>
   <HEAD>
      <TITLE> CGI 中的标记语言基础速成 </TITLE>
   </HEAD>
   <BODY>
    CGI 中的标记语言基础速成简介... ...
   </BODY>
</HTML>
```

这是一个最简单的 HTML 文件。包含 4 个最基本的标记。它的执行结果如图 5-1 所示。

5.2.1　文件的基本构成

HTML 文件是用 ASCII 码写成的，它包括两部分：

- 文件的内容
- HTML 的标记部分

大多数 HTML 标记的书写格式如下：

<标记名> 文件内容 </标记名>

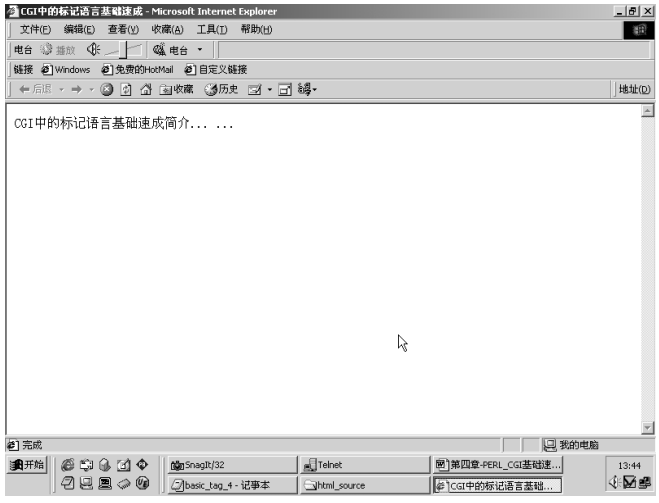

图 5-1 HTML 文件的基本构成范例

标记名通常写在“<>”内。一般的 HTML 标记都具有起始和结束标记，并且成对出现，结束标记前要加“/”。

常识：

1. **并不是所有的标记都有起始和结束标记。**
2. **对于 HTML 标记，大小写或混写是无关的。**

5.2.2 四个基本标记

5.2.2.1 <HTML>和</HTML>标记

在任何一个 HTML 文件里，最先出现的 HTML 标记是<HTML>，它表明这个文件的内容是用 HTML 来实现的。

<HTML>和</HTML>成对出现。文件中的所有文本和 HTML 标记都包含在 HTML 的这个起始和结束标记中。例如：

```
<HTML> 文件的全部内容 </HTML>
```

5.2.2.2 <HEAD>和</HEAD>标记

这是一个网页头部的标记.

5.2.2.3 <TITLE>和</TITLE>标记

这是一个网页名称。每个 HTML 文件都需要有一个文件名称，在浏览器中作为窗口名称显示在该窗口的最上方。网页的名称要写在<TITLE>与</TITLE>标记之间，并且<TITLE>标记应包含在<HEAD>与</HEAD>标记之中。

5.2.2.4　<BODY>和</BODY>标记

网页中的主体内容应该写在<BODY>和</BODY>之间，而<BODY>标记包含在<HTML>标记之内。同样请参见开始的例子。

注意：

保存文件注意 4 点：

1. 如果在 DOS 系统中，扩展名要用“htm”，如果在 Unix 系统中，扩展名要用“html”，因为绝大多数的网络软件要求文件具备这样的扩展名。

2. 不要在文件名中使用空格和特殊字符。

3. 在 UNIX 系统中，文件名中字母的大小写是敏感的。

4. 将起好文件名的文件以纯文本的形式建立。

HTML 对格式的要求并不严格，当文件被浏览器扫描时，所有包含在文件中的空格、回车等均被忽略了。也就是说，文件中只有文本和 HTML 标记。

注意：<PRE>...</PRE>标记中的内容除外。

5.2.3　文件的子标题

子标题是用来区分内容的不同章节的，浏览时会显示不同的大小字形。

HTML 定义了 6 个级别的子标题，它的书写格式为：

```
<Hn> CGI 中的标记语言基础速成 <Hn>
```

其中“H”后面的 n 为数字，用来代表不同子标题级别的标号，n 可在 1~6 中进行选择。

子标题在浏览器中的显示结果与文件正文是不一样的，通常它的字体比较大也较黑，或居中、或带有下划线等等。如：

```
<HTML>
   <HEAD>
      <TITLE> CGI 中的标记语言基础速成 </TITLE>
   </HEAD>
   <BODY> CGI 中的标记语言基础速成
       <H1> CGI 中的标记语言基础速成简介 <H1>
       <H2> CGI 中的标记语言基础速成简介 <H2>
       <H3> CGI 中的标记语言基础速成简介 <H3>
       <H4> CGI 中的标记语言基础速成简介 <H4>
       <H5> CGI 中的标记语言基础速成简介 <H5>
       <H6> CGI 中的标记语言基础速成简介 <H6>
   </BODY>
</HTML>
```

效果如图 5-2 所示。

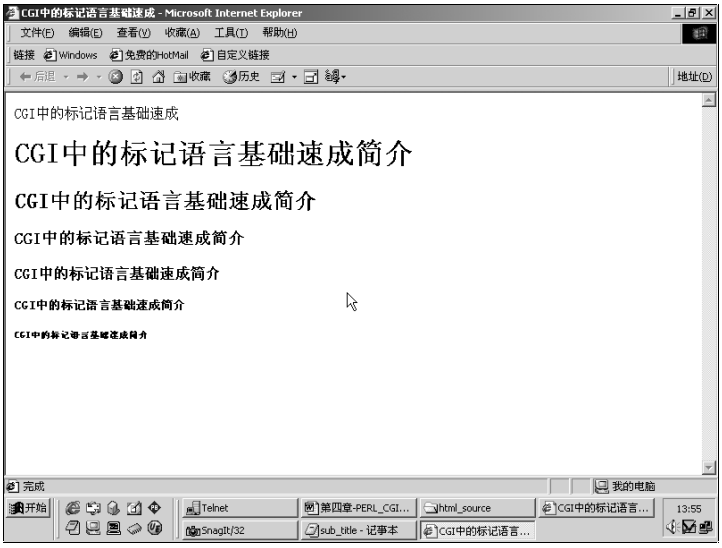

图 5-2　文件的子标题

5.2.4　几个常用标记

5.2.4.1　预格式化文本标记：<PRE>

把制作好的那段文本前后分别加上<PRE>**和**</PRE>**标记，放到** HTML **文件中，原来的文本内容和格式就会原封不动地保留并显示出来。**

5.2.4.2　段落标记：<P>

HTML **允许在文本中间分几个段落，可以使用**<P>**标记。**<P>**标记会自动地为文本进行分段，且段与段之间有一个空行。如：**

```
<HTML>
    <HEAD>
       <TITLE> CGI 中的标记语言基础速成 </TITLE>
    </HEAD>
    <BODY> CGI 中的标记语言基础速成
        <H1> CGI 中的标记语言基础速成简介 <H1>
        <PRE> 这行是      PRE 标记   的 效果       呵呵</PRE>
            <P>HTML 是 Hyper Text Markup Language 的缩写。<P>
            <PRE>
HTML 是一种超文本标记语言，是用来描述 WWW 上超文本文件的。
它主要是在原来文本文件的基础上，加一系列的标记符号描述
其格式，形成网络文件，当用户使用浏
```

需要全本请访问 www.ertongbook.com

```
        </PRE>
    </BODY>
</HTML>
```

效果如图 5-3 **所示**。

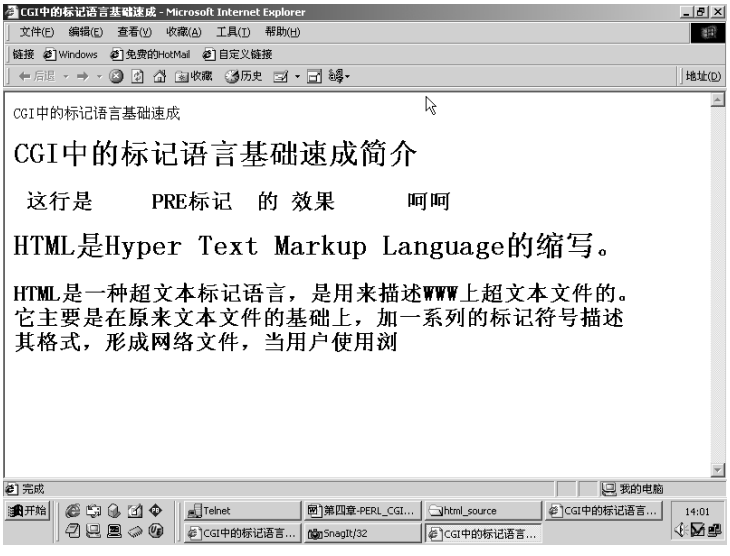

图 5-3　**常用的** HTML **标记效果**

5.2.4.3　**行中断标记**:

要想中断文本中的一行，可以用行中断标记
。**若文本使用了行中断标记，文本将从**
之后开始新的一行。例如:

```
<HTML>
    <HEAD>
       <TITLE> CGI 中的标记语言基础速成 </TITLE>
    </HEAD>
    <BODY>
    <H3>CGI 中的标记语言基础速成<H3>
HTML 是一种超文本标记语言，是用来描述 WWW 上超文本文件的。<BR>
它主要是在原来文本文件的基础上，<BR>加一系列的标记符号描述
其格式，形成网络文件，当用户使用浏
    </BODY>
</HTML>
```

它的执行结果如图 5-4 **所示**。

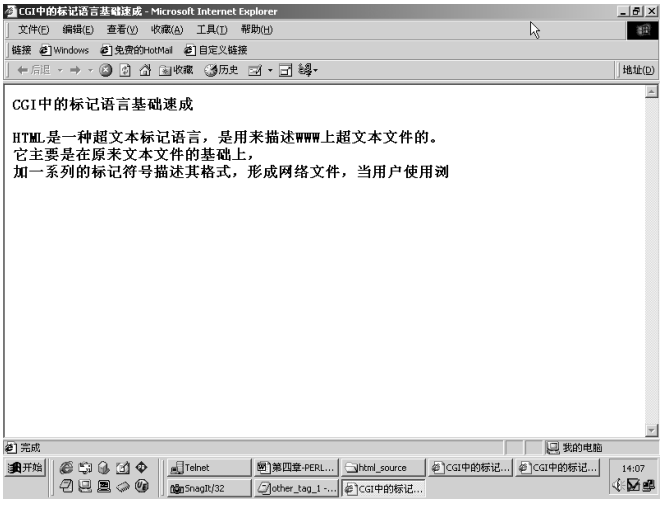

图 5-4　行中断标记效果

5.3　连接技巧

本节讲述建立连接的方法，内容包括：

- ❖ HTML **的连接标记**
- ❖ **在同一个文件中建立连接**
- ❖ **不同文件的连接**

5.3.1　连接标记

HTML **中的连接就是在并不连续的两段文字之间或文件之间用标记符号架起一座桥梁，使读者可根据需要，将它们连贯起来阅读。**

在做文件连接的时候，首先要给连接的段落或文件起个名字，然后做上相应的标记符号。当读者用浏览器浏览你的网页时，文件中设置了连接标记的位置，显示为高亮度。一旦读者点击了这个高亮度显示，就实现了连接。

在 HTML **文件中建立一个连接时，必须做两件事：**

- ❖ **为被连接的文件起个名字；**
- ❖ **在文件中确定连接点，使用户可以在浏览窗口中看到一个高亮度的文字，在高亮度处选择这个连接。**

HTML **文件的连接是通过连接标记**

```
<A>...</A>
```

来实现的。在<A>标记中还包括多个属性，其中最常用的是 HREF 和 NAME 属性。

HREF 是 Hypertext REFerence 的缩写，该属性用来指明所要连接文件的路径和名称或 URL。其简单的结构形式为：

```
<A HREF="#被连接文件名" >
```

NAME 属性用来指出文件中被连接点的定位名称，也称锚名。通过锚名建立连接时，要在锚名前加一个“#”符号，其简单的结构形式为：

```
<A NAME="#锚名">
```

连接标记也是由写在起始和结束标记之间的文字构成一个实际的连接，当网络文件在屏幕上显示时，这些文字将以高亮度或带有下划线或以蓝色或以红色的形式出现。由于高亮度显示部分是给读者的阅读提示，因此，编写时语名应该尽量短小、精悍。格式为：

```
<A HREF="被连接文件名及其路径"> 提示字符串 </A>
```

例如：

```
<A HREF="http://freemail.263.net/"> 263 电子邮局 </A>
```

HTML 还提供了另一个标记<BASE>，与<A>标记联合使用。

<BASE>是个单向标记，写出在<HEAD>...</HEAD>标记之间，并带有一个 HREF 属性。<BASE>的使用格式如下：

```
<BASE HREF="URL">
```

注意：其中“URL”只能写绝对地址。 当 HTML 文件的<HEAD>...</HEAD>中带有<BASE>标记时，该文件内所有与<BASE>标记指出的 URL 相同的文件连接可以只写文件名，而不必写出全部路径的 URL。

举例：

```
<HTML>
    <HEAD>
    <TITLE> HOMEPAGE </TITLE>
    <BASE HREF="http://Perl.linux.org/index.html">
    </HEAD>
    <BODY>
    ...
    <A HREF="friendlinks.html">友情链接</A>
    ...
    </BODY>
</HTML>
```

其中，<A HREF="friendlinks.html">友情链接</A>语句中“friendlinks.html”的 URL，就是<BASE>语句指出的“http://Perl.linux.org/friendlinks.html”。 如图 5-5 所示。

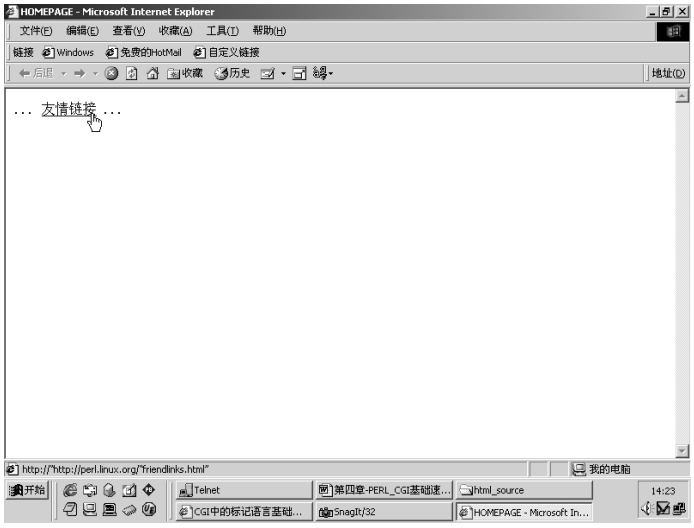

图 5-5 <BASE HREF="URL">**举例**

5.3.2 在同一个文件中建立连接

一个很长的文件，读者在浏览器上可以从头读到尾，但是如果读者不想全部阅读，而只是对其中的一部分感兴趣，这时我们可以运用连接解决这个问题。

方法是在文件的开头处列几个小标题，这相当于本篇文章的目录，在每个标题的后面建立一个连接，使它高亮度显示，当读者对某个问题感兴趣时，他可以直接点击某个标题，而无须从头到尾阅读。

步骤如下：

- **整理出文件的小标题，列在文件的最前面。每个标题单独起一行。接着是文件正文。正文分段书写，与每个标题相对应的内容做为一段；**
- **在文件连接处加连接标记，书写时，要在锚名前面加一个“#”符号。其格式如下：**

```
<A HREF="#prot1"> 标题一 </A>
<A HREF="#prot2"> 标题二 </A>
<A HREF="#prot3"> 标题三 </A>
```

- **为被连接的部分起名字并加标记，这个名字就是定位名称或锚名。锚名被放在文件正文中该标题对应那一段的开头。 文件正文中锚名的位置如下所示：**

```
<A NAME="#prot1">（标题一对应的内容）</A>
<A NAME="#prot2">（标题二对应的内容）</A>
<A NAME="#prot3">（标题三对应的内容）</A>
    ... ...
```

举例：

```
<HTML>
    <HEAD>
    <TITLE> 关于 WWW </TITLE>
    </HEAD>
    <BODY>
    <H3> 关于 WWW </H3>
    <A HREF=“#prot1”>WWW 的基本概念 </A><BR>
    <A HREF=“#prot2”>WWW 的工作环境 </A><BR>
    <A HREF=“#prot3”>Web 服务器 </A><BR>
    <P><A NAME=“#prot1”>WWW 的基本概念 </A><BR>
    WWW 是 World Wide Wed（全球信息网）的缩写。... ...</P>
    <P><A NAME=“#prot2”>WWW 的工作环境 </A><BR>
    既然 WWW 的信息量之大，以致于任何组织和机构 ... ...</P>
    <P><A NAME=“#prot3”>Web 服务器 </A><BR>
    WWW 的大量信息存在 Web 服务器上， Web 服务器的作用 ... ...</P>
    </BODY>
</HTML>
```

执行结果如图 5-6 **所示**。

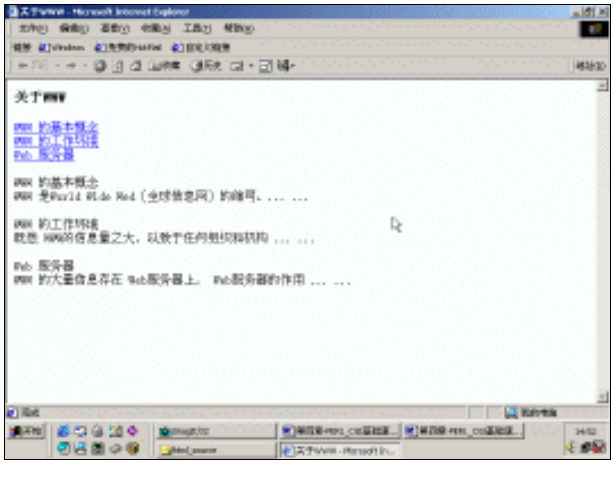

图 5-6　**在同一个文件中建立连接**

5.3.3　不同文件间建立连接

当对不同文件进行连接时，需要有两个 HTML **文件，为它们起好名字，明确主连接文件及被连接文件**。

被连接的文件是全部内容被连接，只要在主连接文件中的连接处加上：

<A HREF=“被连接文件名及其路径”> 提示字符串 </A>

的标记就可以实现两个文件之间的连接。

例如：

```
<HTML>
    <HEAD>
    <TITLE> HOMEPAGE </TITLE>
    </HEAD>
    <BODY>
    ...
    <A HREF=“http://Perl.linux.org /index.html”>SPACEWALK</A>
    ...
    </BODY>
</HTML>
```

5.4 列表技巧

本节的内容包括：

- 列表标记的书写形式
- 带编号的有序列表
- 无序列表
- 有序列表和无序列表属性的应用
- 菜单和目录列表
- 定义列表
- 嵌套列表
- 建立连接列表

建立一个完整的网页，合理、灵活地使用列表标记是十分重要的。在浏览器上阅读文件时，希望看到的是按一定规则和格式排序的文件。若在 HTML 文件中使用列表标记，就可以达到这个目的。

5.4.1 列表标记的书写形式

建立列表，有以下共同点：

- 一个列表，均应有起始标记和结束标记，其内容均包含在相应的起始和结束标记之中。
- 列表中的每一项都有相应的标记，说明列表中的具体内容。

在 HTML 文件中，列表标记和列表项可以任意形式书写。但在建立列表时，为清楚起见，

需要全本请访问 www.ertongbook.com

书写时最好是让列表中的每一项均从新一行开始。

下面列出的两种书写形式都是允许的。它们的书写形式不同，但输出结果相同，显然，下述的第一种书写形式要比第二种书写形式清晰、易读。

第一种形式：

```
<P>...</P>
<OL>
   <LI>...
   <LI>...
   <LI>...
</OL>
```

第二种形式：

```
<P>...</P>
<UL>
<LI>...<LI>...<LI>...
</UL>
```

5.4.2　有序列表

有序列表用于说明文件中的某些部分，需按特定顺序排列和显示。有序列表的输出结果是在列表的每项前都有编号，因此，读者可以清晰地了解它每行的顺序。

有序列表是用如下标记来实现的：

```
<OL>...</OL>
```

其中的每一个列表项都应以标记<LI>开始。标记<LI>是单向的，没有结束标记，在同一个有序列表中，下一个<LI>的出现，就表明上一个列表项的结束。

无须用手工添加编号，当在浏览器上显示时，在每一个<LI>列表项指明的项前，便具有了连续的编号。如果插入或删除一个列表项，待下次浏览时，编号会自动调整。举例建立一个有序列表：

```
<html><body>
<P>建立列表</P>
<OL>
   <LI>列表标记
   <LI>列表的书写形式
   <LI>带编号的有序列表
   <LI>无序列表
</OL>
</body>
```

```
</html>
```

执行结果如图 5-7 **所示**。

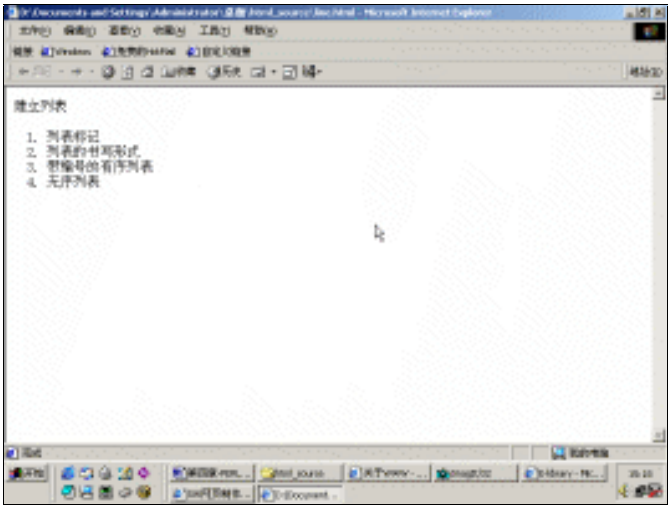

图 5-7　**带编号的有序列表**

5.4.3　无序列表

无序列表是用于说明文件中需要列表的某些成分，可以任意顺序显示出来。无序列表的输出结果是在列表每项前有一个特定符号或是空白。

无序与有序列表写法相似，在无序列表中，除了用标记<UL>**来代替有序列表中的**<OL>**外，其他写法与有序列表完全相同。例子如下：**

```
<html>
<body>
<P>建立列表</P>
<UL>
    <LI>列表标记
    <LI>列表的书写形式
    <LI>带编号的有序列表
    <LI>无序列表
</UL>
</body>
</html>
```

执行结果如图 5-8 **所示**。

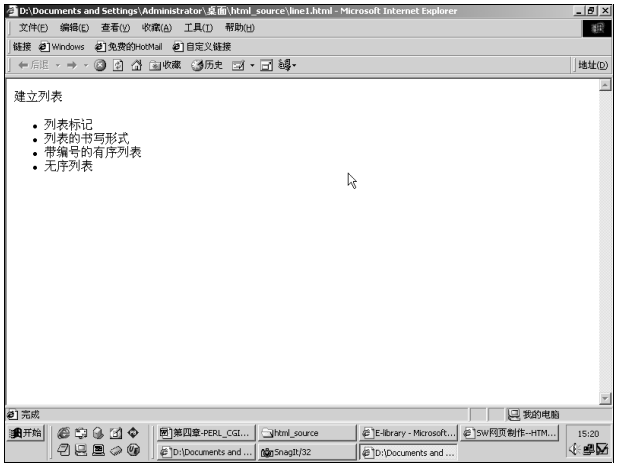

图 5-8　无序列表

5.4.4　列表中属性的应用

一般情况下，如果在 HTML **中创建了一份列表，浏览器会按已有标记自动设置无序列表的行标志符号类型，或者有序列表的顺序编号（如** 1，2，3，...）。

在 HTML **标记中，有些标记还具有属性，由这些属性的取值来设置类型。这些属性和标记相配合，可以达到上述目的。现在介绍**<OL>**中的** TYPE **和** START **属性；**<UL>**中的** TYPE **属性及其应用。**

5.4.4.1　TYPE 属性

在有序列表中，TYPE **属性对列表项给定的几种行标志符号类型是：**

- TYPE=1，**表示列表项用数字标号（**1，2，3，...），**表示缺省项。**
- TYPE=A，**表示用大写字母标号（**A，B，C，...）
- TYPE=a，**表示用小写字母标号（**a，b，c，...）
- TYPE=I，**表示用大写罗马字母标号（**Ⅰ，Ⅱ，Ⅲ，...）
- TYPE=i，**表示用小写罗马字母标号（**ⅰ，ⅱ，ⅲ，...）

使用 TYPE **属性的有序列表举例：**

```
<html>
<body>
<P>建立列表</P>
<OL TYPE=a>
    <LI>列表标记
    <LI>列表的书写形式
    <LI>带编号的有序列表
```

```
    <LI>无序列表
</OL>
<OL TYPE=A>
    <LI>列表标记
    <LI>列表的书写形式
    <LI>带编号的有序列表
    <LI>无序列表
</OL>
</OL>
<OL TYPE=i>
    <LI>列表标记
    <LI>列表的书写形式
    <LI>带编号的有序列表
    <LI>无序列表
</OL>
</body>
</html>
```

执行结果如图 5-9 **所示**。

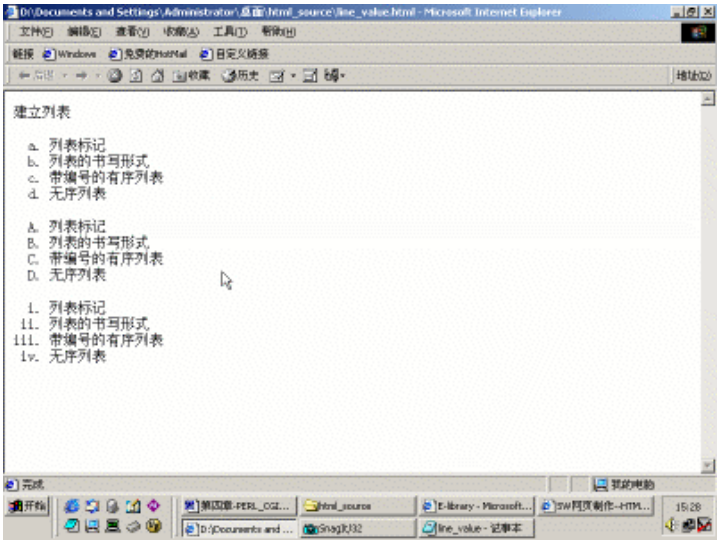

图 5-9　**有序列表** TYPE **属性**

在无序列表中，TYPE **属性对列表项给定的几种行标志符号类型是**：

❖　TYPE=DISC，**设定一个实心圆点的行标号，表示缺省项**。

❖　TYPE=CIRCLE，设定一个方形中空（有的浏览器显示圆形中空）的行标号。

❖　TYPE=SQUARE，设定一个方形实心的行标号。

看下面使用 TYPE 属性的无序列表。举例：

```
<html>
<body>
<P>建立列表</P>
<UL TYPE=DISC>
    <LI>列表标记
    <LI>列表的书写形式
    <LI>带编号的有序列表
    <LI>无序列表
</UL>
</body>
</html>
```

执行结果如图 5-10 所示。

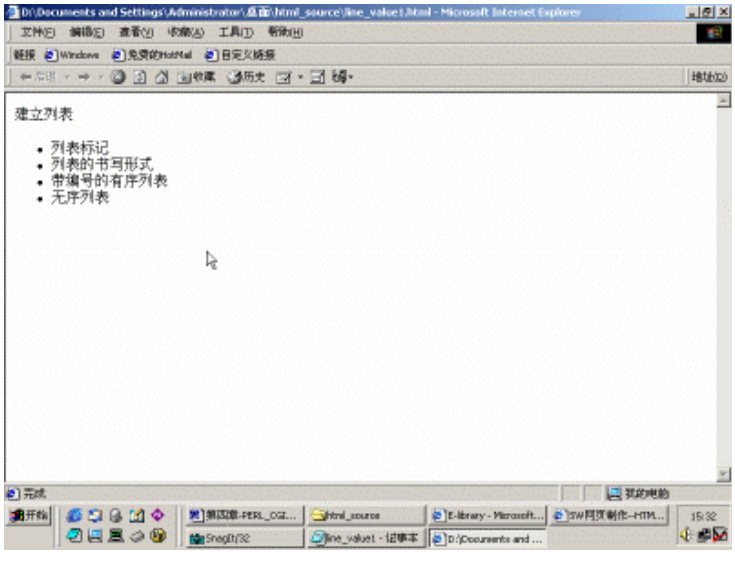

图 5-10　无序列表的 TYPE 属性

5.4.4.2　START 属性

在<OL>中，该属性可以设定列表清单的标号从第几项开始。

在有序列表中，如果<OL>标记的属性用的是：TYPE=A，START=3，那么列表清单标号将从大写字母 C 开始，并接着用 D，E，...作下面的标号。

如果<OL>的属性 TYPE=1，START=3，那么列表的标号将从 3 开始，接着用 4，5，...

作下面的标号。也就是说，在有序列表中，用 START 值给<LI>标记项一个特定标号，这个标号指定<LI>对应的某项，在此项之后的各项将依次改变标号。

注意：TYPE 属性和 START 属性一样，在一些浏览器中有可能会被忽略，所以，欲产生特定的效果，完全依赖这些属性并非是最好的方法。

5.4.5 菜单和目录列表

菜单是一些项或短小段落的列表。在某些浏览器中，菜单列表用以区分文件的段落。其标记为：

```
<MENU>...</MENU>
```

并且在每一项前都要加上<LI>标记。

目录列表是比菜单列表更短的列表，在浏览器里被垂直对齐地列出，目录列表与 Unix 系统中的目录列表很相似，它的标记为：

```
<DIR>...</DIR>
```

并且列表中的第一项前也均加<LI>标记。 例如建立一个菜单列表和一个目录列表。

举例：

```
<html>
<body>
<p>建立一个菜单列表<p>
<MENU>
    <LI>列表标记
    <LI>列表的书写形式
    <LI>带编号的有序列表
    <LI>无序列表
</MENU>
<p>建立一个目录列表<p>
<DIR>
    <LI>列表标记
    <LI>列表的书写形式
    <LI>带编号的有序列表
    <LI>无序列表
</DIR>
</body>
</html>
```

执行结果如图 5-11 所示。

需要全本请访问 www.ertongbook.com