

第 3 章 XHTML 基础

本章学习目标

- (1) 掌握 XHTML 的基本结构。
- (2) 熟练掌握与文字处理、列表、多媒体、超链接、表格等网页元素相关的 XHTML 标签及标签属性。

本章主要介绍 XHTML 的基本结构及描述网页元素所用的标签和属性。通过学习标签和属性,为将来页面结构的设计奠定基础。

3.1 XHTML 的基本结构与语法

XHTML 的页面结构要比 HTML 的页面结构要求严谨,更容易让浏览器知道该怎么去解释网页。下面先来了解一下 XHTML 的基本结构。

3.1.1 XHTML 的基本结构

编写 XHTML 需要更严格的语法规则,XHTML 是更加严谨的 HTML。在实际中应遵循规则,形成一份规范的 XHTML 文件。

1. 基本结构

XHTML 的基本结构主要包括如下 3 部分。

- (1) DOCTYPE——文档类型声明。
- (2) head——文件头部。
- (3) body——文件主体。

基本语法结构如下:

```
<!DOCTYPE...>
<html>
<head>
    <title>...</title>
</head>
<body>...</body>
</html>
```

2. 文档类型声明

XHTML 文档是从文档类型声明开始的,在 XHTML 1.0 中有如下 3 种文档类型定义声明可以选择。

- (1) Transitional(过渡类型)。
- (2) Strict(严格类型)。
- (3) Frameset(框架类型)。

1) 过渡类型

过渡类型是一种要求不很严格的 DTD,允许在页面中使用 HTML 4.01 的标识(符合 XHTML 语法标准)。过渡的 DTD 的写法如下:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

2) 严格类型

严格类型是一种要求严格的 DTD,不允许使用任何表现层的标识和属性,如
等。严格的 DTD 的写法如下:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

3) 框架类型

框架类型是一种专门针对框架页面所使用的 DTD,当页面中含有框架元素时,就要采用这种 DTD。框架 DTD 的写法如下:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

3. 语法规则

所有 XHTML 文档必须进行文件类型声明。在 XHTML 文档结构中必须存在<html>、<head>、<body>标记元素,而<title>标记元素必须位于<head>标记元素中。所以这 4 个标记元素是在 DTD 中定义了强制使用的 HTML 元素。

XHTML 文档中标记元素属性的编写规则如下。

- (1) XHTML 文档必须拥有根元素。
- (2) 标签名和属性名必须用小写字母。
- (3) 属性名不能简写,属性值必须加引号。
- (4) 用 id 属性代替 name 属性。
- (5) XHTML DTD 定义了强制使用的 HTML 元素。
- (6) XHTML 元素必须被正确地嵌套。
- (7) XHTML 元素必须被关闭。

【例 3.1】 XHTML 的基本结构示例。

```
<!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=utf-8" />
<title>XHTML 基本结构</title>
</head>
<body>
<center>
<h3>学习 XHTML 基本语法</h3>
```

```
<br />
<hr />
<font>
这是一个演示 XHTML 文档结构的案例。
</font>
</center>
</body>
</html>
```

程序运行效果如图 3.1 所示。

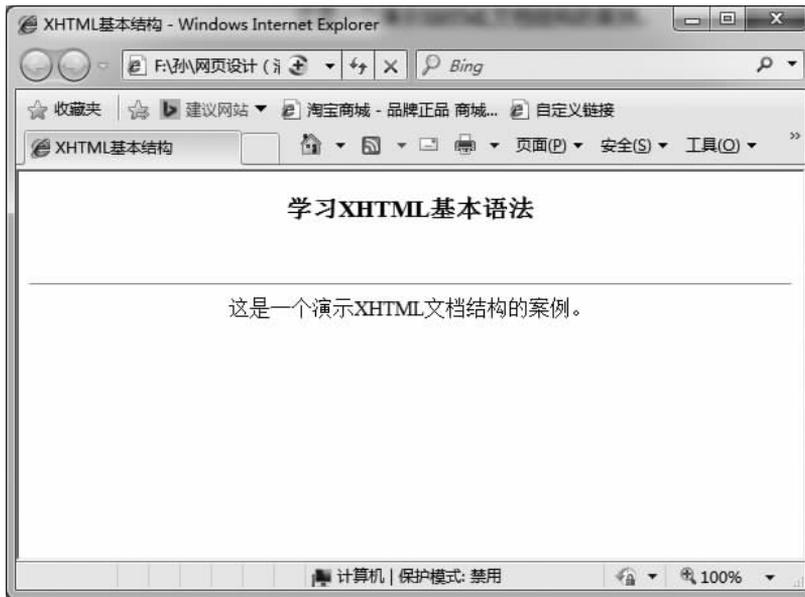


图 3.1 XHTML 的基本结构示例图

下面对网页结构中的标签做如下说明。

(1) 文档的声明部分。

`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`声明了文档的类型为过渡类型,允许有 HTML 4.0 标签语法。文件类型声明并非 XHTML 文档自身的组成部分。它并不是 XHTML 元素,也没有关闭标签。

(2) 文档标签`<html></html>`。

`<html></html>`表示标签之间的内容是 HTML 文档,保存的页面文件类型为 .html 或 .htm,该类型文件会被 Web 浏览器打开。

在 XHTML 中,`<html>`标签内的 `xmlns` 属性是必需的。然而,即使当 XHTML 文档中没有这个属性时,w3.org 的验证工具也不会提示错误。这是因为,"`xmlns = http://www.w3.org/1999/xhtml`"是一个固定的值,即使你没有把它包含在代码中,这个值也会被添加到`<html>`标签中。

(3) 页面的头标签`<head></head>`。

头标签<head>和</head>构成 XHTML 的开头部分,在浏览器窗口中,头部信息是不被显示在正文中的,在此标签中,可以插入其他标签,用以说明文件的标题和整个文件的一些公用属性。在此标签对之间,可以使用<title></title>、<script></script>、<link />等标签。

(4) 页面的标题标签<title></title>。

标题标签<title></title>决定了显示在页面标题栏中的文字,一般是网页的“主题”。如在例 3.1 中,出现在标题栏的文字“XHTML 基本结构”。

(5) 主体标签<body></body>。

<body></body>标签是 XHTML 文档的主体,其中放置的是页面中所有的内容,如图片、文字、表格、表单、超链接等,在此标签之间可以包含<p></p>、<h1></h1>、
、<div></div>等众多标签。它们所定义的内容将会在浏览器的窗口内显示出来。

<body>标签有自己的属性,设置<body>属性,可以控制整个页面的显示方式。可以通过 Dreamweaver 来进行页面属性的设置,如图 3.2 所示,在 Dreamweaver 中设置<body>的属性。在后面章节将会介绍通过 CSS 样式设置<body>的显示属性。



图 3.2 “页面属性”对话框

3.1.2 HTML 标签

在 HTML 页面中,带有<>符号的元素称为 HTML 标签,也称为 HTML 标记。如上面提到的<html>、<body>、<head>等都是 HTML 标签。所谓标签,就是放在<>标记符中表示某个功能的编码指令,也称为 HTML 标记或 HTML 元素。

标签分为单标签和双标签。

单标签是指用一个标记符号即可完整地描述某个功能的标签,基本语法如下:

<标签名 />

如例 3.1 中的水平线标签<hr/>,就是定义了一条水平线。

网页中大部分的标签是成对出现的,也就是双标签,如例 3.1 中的<body>、<center>、

<h3>等。双标签是指由开始和结束两个标记符组成的标签,其基本语法如下:

```
<标签名>内容</标签名>
```

<标签名>表示该标签的作用开始,也称为“开始标记”;</标签名>表示该标签作用的结束,也称为“结束标记”。在 XHTML 的语法要求中,双标签必须有开始标记和结束标记。

3.1.3 HTML 属性

HTML 标签都有自己相应的属性,通过设置标签的属性,可以修改内容的样式。其基本语法如下:

```
<标签名 属性 1="属性值" 属性 2="属性值"……>内容</标签名>或<标签名 属性 1="属性值" 属性 2="属性值"…/>
```

标签可以有多个属性,但必须写在开始标记中,位于标记名后面。属性不分先后顺序,标记名与属性、属性与属性之间均以空格分开。任何标签的属性都有默认值,省略该属性,则会取默认值。

如例 3.1 中,<h3>标签中的内容是加粗显示的,那就是<h3>标签的默认属性。<hr/>水平线的宽度默认是 100%,就是占满整行。

下面介绍网页中常用的标签和其相应的属性。

3.1.4 元数据标签<meta>

元数据标签<meta>位于文档的头部,不包含任何内容。<meta> 标签的属性定义了与文档相关联的名称-值对,提供有关页面的元信息(meta-information),例如页面编码方式、作者信息、针对搜索引擎和更新频度的描述和关键词。

在 HTML 中,<meta> 标签没有结束标签,在 XHTML 中,<meta> 标签必须被正确地关闭。<meta> 标签永远位于页面的头标签<head></head>内部,在一个 HTML 头页面中可以有多个<meta>标签。元数据总是以名称-值的形式被成对传递的,属性主要有页面描述信息 name 和标题信息 http 两种,格式分别如下:

```
<meta name="属性值" content="名称"/>
```

其中 name 属性的取值有如下几种。

1) <meta name="generator" content="网站编辑工具名称"/>

在该语法中,name 为属性名称,这里设置为 generator,即设置为编辑软件的名称,在 content 中定义具体的编辑软件的名称。现在有很多软件可以制作网页,可以在源代码的头部页面的元数据标签中设置网页编辑工具的名称。

2) <meta name="keywords" content="关键字"/>

在该语法中,name 为属性名称,这里设置为 keywords,即设置网页关键字属性,在 content 中定义具体的关键字的内容。

3) <meta name="description" content="描述网页的主要内容及主题等"/>

在该语法中,name 为属性名称,这里设置为 description,也就是将元数据属性设置为页

面描述,在 content 中定义具体的描述语言。

4) `<meta name="author" content="作者名称"/>`

在该语法中,name 为属性名称,这里设置为 author,即设置作者信息,在 content 中定义具体的作者信息。

5) `<meta name="robots" content="指令组合"/>`

在该语法中,name 为属性名称,这里设置为 robots,是泛指所有的搜索引擎,也可以特指某个搜索引擎,例如 Googlebot、Baiduspider 等,在 content 中定义搜索引擎对页面的搜索方式。

content 取值及其对应的含义如表 3.1 所示。

表 3.1 content 取值及其对应的含义

content 取值	含 义
index, follow	当前页面允许被检索,页面中的链接也允许被检索
index, nofollow	当前页面允许被检索,页面中的链接不允许被检索
noindex, follow	当前页面不允许被检索,页面中的链接允许被检索
noindex, nofollow	当前页面不允许被检索,页面中的链接不允许被检索

表 3.1 中的第一种方式可以写成 all,第四种方式可以写成 none。

需要注意的是,不可把两个对立的反义词写到一起,例如:

```
<meta name="robots" content="index,noindex"/>
```

或者直接同时写上两句也是不可以的。

```
<meta name="robots" content="index, follow"/>
<meta name="robots" content="noindex, follow"/>
```

常见的 http 标题信息有: http-equiv="content-type" 或 "expire" 或 "refresh" 或 "cache-control" 或 "set-cookie" 等,具体格式如下。

1) `<meta http-equiv="content-type" content="text/html; charset=utf-8" />`

在该语法中,http-equiv 用于传送 HTTP 通信协议的标头,也就是设置标头属性的名称,在 content 中才设置具体的属性值。在 charset 中设置网页的内码语系,即设置字符集的类型,charset 往往设置为 gb2312,即简体中文。英文是 ISO-8859-1 字符集,此外,还有 BIG5、utf-8 等字符集。

2) `<meta http-equiv="refresh" content="10;URL=http://www.bzmc.edu.cn" />`

在该语法中,refresh 表示网页的刷新,在 content 中设置刷新的时间和刷新后的地址,时间和链接地址之间用分号分隔,默认情况下跳转时间是以秒(s)为单位的。该代码中页面将在 10s 后自动跳转到滨州医学院首页。

3) `<meta http-equiv="expires" content="Tue, 22 Aug 2017 14: 25: 27 GMT" />`

在该语法中,expires 表示网页到期,在 content 中设置网页的到期时间,当时间超过 content 属性值的日期时,缓存里关于此网页的内容将过期而无法使用,必须到服务器上重

新调用,需要注意的是时间必须使用 GMT 的时间格式。

```
4) <meta http-equiv="cache-control" content="no-cache" />
```

使用网页缓存可以加快网页的浏览速度,省去读取同一网页的时间。但如果网页的内容经常频繁更新,网页制作者希望用户能看到最新的网页内容,可通过该语句来禁用页面缓存。在该语法中,cache-control 用来设定缓存的属性,在 content 中设置禁止调用缓存的语句,每次打开此页面都需要重新读取最新页面。

```
5) <meta http-equiv="set-cookie" content="Wed, 15 Mar 2017 12: 06: 08 GMT"/>
```

在该语法中,如果网页过期,那么存盘的 cookie 将被删除,cookie 是由站点创建的将信息存储在计算机上的文件,content 值设定到期时间。该代码设定此网页的 cookie 到达 content 值代表的时间时,将被删除,需要注意的是时间也必须使用 GMT 格式。

3.2 文字处理

在 Word 文档中,通过软件中提供的工具实现文字、段落等的排版,在 HTML 页面中,需要运用标签来描述文字、段落等,并运用相关属性对文字、段落进行相应的修饰,以使页面更加整齐美观。本节就来介绍文字相关的标签及属性。

3.2.1 标题标签 <hn>

一般文章都有标题、副标题、章、节等结构,XHTML 也提供了相应的标题标签 <hn> </hn>,用于设置网页中的标题文字,被设置的文字将以黑体或粗体的方式显示在网页中。其中 n 为标题的等级,总共提供 6 个等级的标题,n 越小,标题字号越大。

标题标签的格式:

```
<hn 属性="属性值">标题内容</hn>
```

说明: <hn> 标签成对出现。

常用的属性: align 对齐属性,属性值有 left、center、right。

在网页中设置文本的对齐方式主要通过 align 属性进行设置,该属性主要在 <p>、<h1>~<h6> 标记元素中。

注意: 在 XHTML 1.0 中,如果文档类型声明为严格类型,则 <p>、<h1>~<h6> 标记元素的 align 属性不被支持。align 的取值为 left(左对齐,默认值)、center(居中对齐)、right(右对齐)。

XHTML 的基本属性如表 3.2 所示。

表 3.2 XHTML 的基本属性

属 性	值	描 述
class	class_rule 或 style_rule	元素的类(class)
id	id_name	元素的某个特定 id
style	样式定义	内联样式定义
title	提示文本	显示于提示工具中的文本

【例 3.2】 标题标签的应用。

```
<!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=utf-8" />
<title>标题标签 hn</title>
</head>
<body>
<h1 align="center">一级标题文字居中</h1>
<h2 align="left">二级标题文字,左对齐</h2>
<h3 align="right">三级标题文字,右对齐</h3>
<h4 align="justify">四级标题文字,两端对齐</h4>
<h5>五级标题文字,默认对齐属性</h5>
<h6>六级标题文字,默认对齐属性</h6>
</body>
</html>
```

程序运行效果如图 3.3 所示。

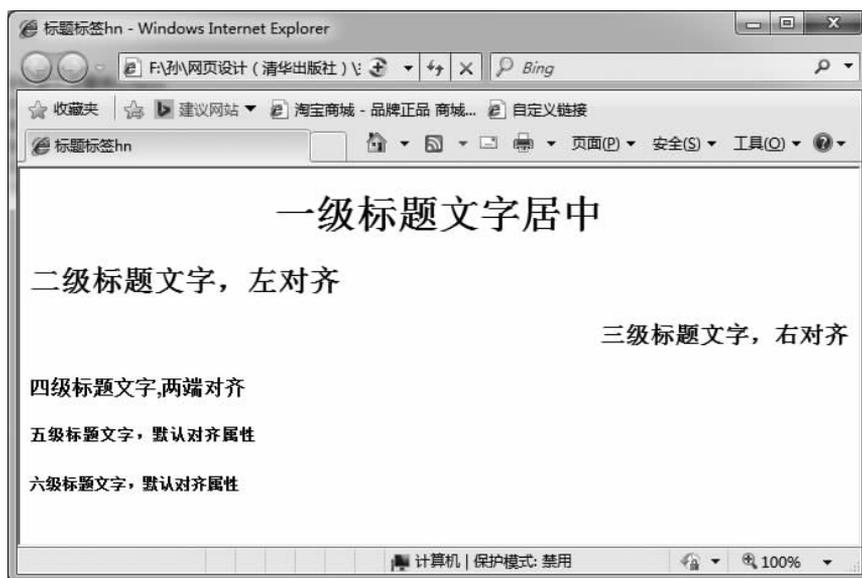


图 3.3 <h> 标题标记

3.2.2 段落标签<p>

段落标签<p></p>是用来创建一个段落,在此标签对之间加入的文本将按照段落的格式显示在浏览器上。使用该标签后,每块文本段落之间都会空出一行。

段落标签的格式:

<p 属性="属性值">段落内容</p>

说明：<p>标签成对出现。

常用的属性：align 对齐方式，一般值有 left、center、right 3 种，分别设置段落文字的左、中、右对齐方式。

3.2.3 换行标签

是一个单标签，在 XHTML 中，该标签必须被关闭。换行标签没有属性，它主要的作用是分行。当希望内容重起一行，并且在行与上一行之间不空出一行间距，那就需要换行标签，而不是段落标签。

注意：

(1) <p>标签和
标签都能另起一行。

(2) <p>换行后与上一行产生一个距离，
换行后没有这一距离。

(3) 多个<p>标签连续使用时不能产生更多的空行，而多个
标签连续使用时可以产生多个空行。

【例 3.3】 换行标签与段落标签区别演示案例。

```
<!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=utf-8" />
<title> 段落标签和换行标签的区别</title>
</head>
<body>
<!--
```

用段落标签换行，会出现段前段后间距，默认首行不会空出 2 个汉字。当内容较多时，段内文字会自动换行。

```
-->
<h3 align="center">念奴娇·赤壁怀古</h3>
<p align="center">
作者：苏轼
</p>
<p>
大江东去，浪淘尽，千古风流人物。
故垒西边，人道是，三国周郎赤壁。
乱石穿空，惊涛拍岸，卷起千堆雪。
江山如画，一时多少豪杰。
遥想公瑾当年，小乔初嫁了，雄姿英发。
羽扇纶巾，谈笑间，檣櫓灰飞烟灭。
故国神游，多情应笑我，早生华发。
人生如梦，一尊还酹江月。
```

```

</p>
<hr />
<!--用换行标签进行换行,没有段前段后间距。当文字较多时,也会自动换行。-->
念奴娇·赤壁怀古<br />
作者: 苏轼<br />
大江东去,浪淘尽,千古风流人物。
故垒西边,人道是,三国周郎赤壁。
乱石穿空,惊涛拍岸,卷起千堆雪。
江山如画,一时多少豪杰。<br />
遥想公瑾当年,小乔初嫁了,雄姿英发。
羽扇纶巾,谈笑间,檣櫓灰飞烟灭。
故国神游,多情应笑我,早生华发。
人生如梦,一樽还酹江月。<br />
</body>
</html>

```

程序运行效果如图 3.4 所示。

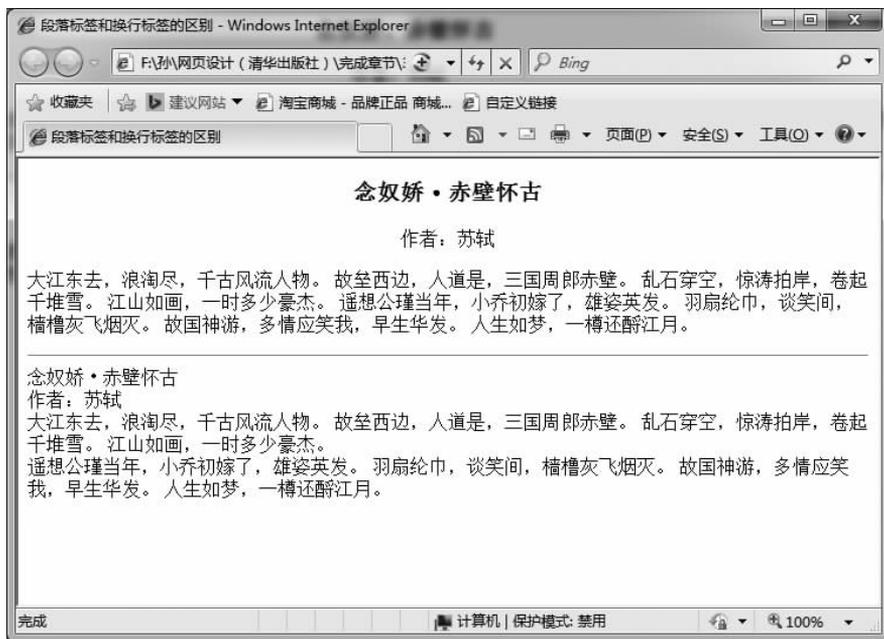


图 3.4 换行标签与段落标签的区别演示图

3.2.4 水平线标签<hr/>

<hr/>标签是单标签,该标签可以在屏幕上显示一条水平线,用于分割页面的不同部分,会使文档结构清晰,文字的编排更整齐。

通过设置<hr/>标签的属性,可以控制水平分割线的样式。<hr/>标签的属性如表 3.3 所示。

表 3.3 <hr/> 标签的属性

属 性	参 数	功 能	默 认 值
size		设置水平线的粗细	2
width		设置水平线的宽度	100%
align	left, center, right	设置水平线的对齐方式	center
color		设置水平线的颜色	black
noshade		取消水平线的 3D 阴影	

【例 3.4】 水平线的应用。

```
<!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=utf-8" />
<title>水平线的用法</title>
</head>
<body>
<p> 以下是第一条线,使用默认属性显示< /p>
<hr />
<p> 以下是第二条线,设置了粗细 size=5,颜色 color 为红色</p>
<hr size="5" color="# FF0000"/>
<p>以下是第三条线,设置显示宽度为 50%,粗细为 10,颜色为蓝色,没有阴影</p>
<hr size="10" width="50%" color="# 0000FF" noshade="noshade"/>
</body>
</html>
```

程序运行效果如图 3.5 所示。

3.2.5 原样显示文字标签<pre>

要保留原始文字排版的格式,可以通过<pre>标签来实现,方法是把制作好的文字排版内容前后分别加上<pre>和</pre>。

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

可以导致段落断开的标签(例如标题、<p> 和<address> 标签)绝不能包含在<pre>所定义的块里。尽管有些浏览器会把段落结束标签解释为简单地换行,但是这种行为在所有浏览器上并不都是一样的。

<pre>标签中允许的文本可以包括物理样式和基于内容的样式变化,还有链接、图像和水平分隔线。当把其他标签(例如<a> 标签)放到<pre>块中时,就像放在 HTML/XHTML 文档的其他部分中一样即可。

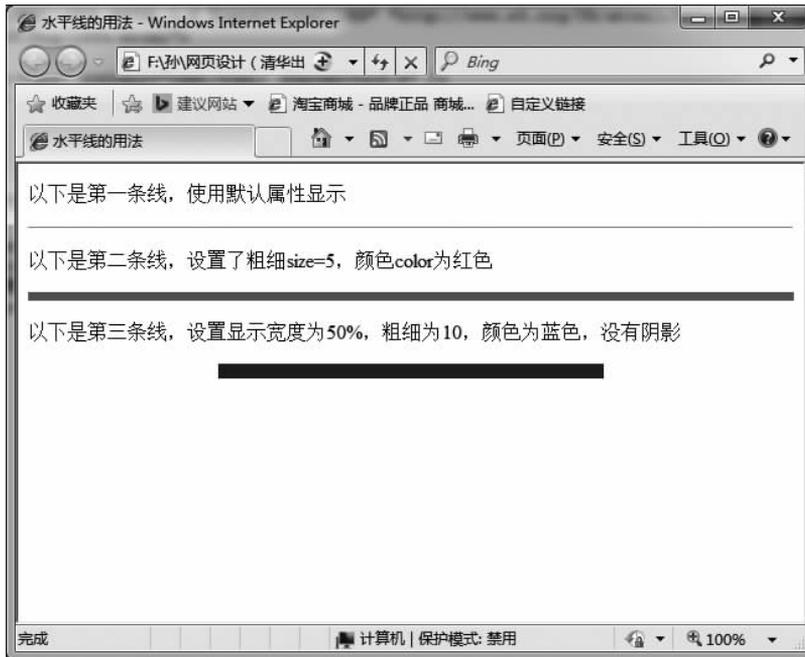


图 3.5 水平线应用效果图

3.2.6 居中对齐标签<center>

文本在页面中使用<center>标签进行居中显示,<center>标签是双标签,可以增加标准属性:id、class、title、style。下面通过案例来了解原样显示标签和居中标签的用法。

注意:在 XHTML 1.0 Strict DTD 中,center 标签不被支持。

【例 3.5】 标签<center>的应用。

```
<!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=utf-8" />
<title>原样显示标签和居中标签</title>
</head>
<body>
<center>登鹳雀楼</center>
<pre>
    白日依山尽,
      黄河入海流。
        欲穷千里目,
          更上一层楼。
</pre>
```

```
</body>
</html>
```

程序运行效果如图 3.6 所示。

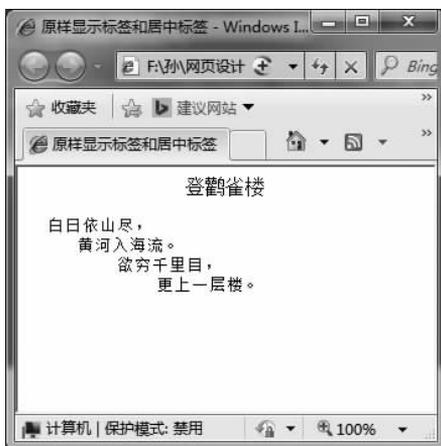


图 3.6 原样显示标签和居中标签的运行效果图

3.2.7 文字格式标签

在 XHTML 中设置网页中的文字字体使用 标签。

注意：在 XHTML 1.0 Strict DTD 中，font 标签不被支持，在网页中尽量使用 CSS 来定义文本的字体、字号和颜色。

 标签的主要属性及实例如表 3.4 所示。

表 3.4 标签的主要属性及实例

属 性	作 用	例 子
size	设置字体大小	size="5"
face	设置字体名称	face="Times"
color	设置字体颜色	color="red"

3.2.8 字体格式化

在有些文字的显示中，常常会用一些特殊的字形或字体来强调、突出、区别以达到提示的效果。为了让文字富有变化，或者为了着意强调某一部分，HTML 提供了一些标签产生这些效果。这些标签可以分为物理类型和逻辑类型，下面分别来介绍这两种类型的标签。

1. 物理类型标签

这类标签用于设置文本在网页中显示时的格式，告诉浏览器应该以何种格式显示文字，常用的物理类型标签及其作用如表 3.5 所示。

表 3.5 常用的物理类型标签及其作用

标 签	作 用
...	标记中的文本字体加粗
<i>...</i>	标记中的文本倾斜
<tt>...</tt>	标记中的文本显示打字机字体,字体较宽
<u>...</u>	标记中的文本加下画线
<strike>...</strike>	标记中的文本加横贯文字的删除线
_{...}	标记中的文本以下标显示
^{...}	标记中的文本以上标显示
<big>...</big>	标记中的文本以较大字体显示
<small>...</small>	标记中的文本以较小字体显示

【例 3.6】 字体中物理类型标签显示字体格式的运用。

```

<body>
  a<sub>2</sub>
  <br/>
  <b><u><i>format text</i></u></b>
  <br/>
  a<sup>2</sup>
</body>

```

程序运行效果如图 3.7 所示。

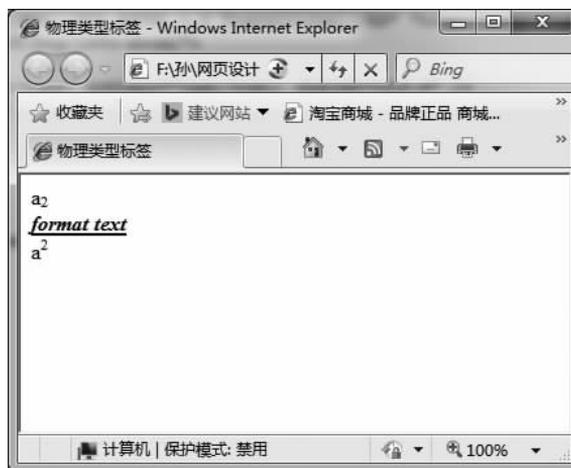


图 3.7 物理类型标签示例图

2. 逻辑类型标签

逻辑类型标签是定义了包含文本内容的属性,告诉浏览器该文本所具有的特性,由浏览器决定以何种格式显示这些文本。常见逻辑类型标签及其作用如表 3.6 所示。

表 3.6 逻辑类型标签及其作用

标 签	作 用
...	强调标记中的文本,一般显示斜体字
...	强调标记中的文本,一般显示粗体字
<code>...</code>	用于指出这是一组代码,一般是一种定宽的字体
<kbd>...</kbd>	定义键盘文本,一般显示较粗的定宽字体
<var>...</var>	用来表示变量,一般显示斜体字
<samp>...</samp>	定义样本文本,显示一段常用的文字,一般是一种定宽的字体
<dfn>...</dfn>	表示一个定义或说明,一般为斜体字
<cite>...</cite>	用于引证、举例,一般显示斜体字

【例 3.7】 逻辑类型标签使用案例。

```
<body>
  <dfn>cccc</dfn><br/>
  <cite>dddd</cite><br/>
  <code>Computer code</code><br/>
  <kbd>Keyboard input</kbd><br/>
  <samp>Sample text</samp><br/>
  <var>Computer variable</var><br/>
  <em>aaaa</em><br/>
  <strong>bbbb</strong>
</body>
```

程序运行效果如图 3.8 所示。

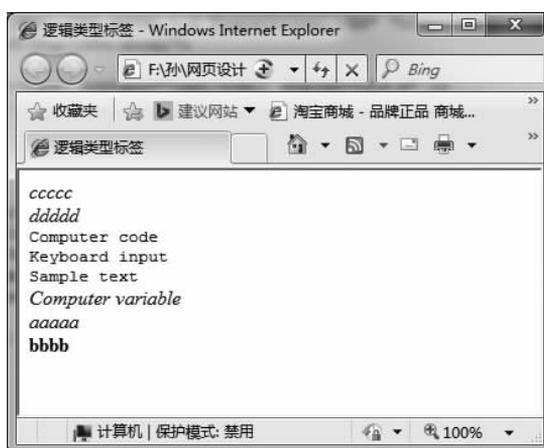


图 3.8 逻辑类型标签示例图

3.2.9 特殊符号的使用

在 XHTML 中定义了一些程序开发者不能直接从键盘输入的特殊字符,如©。或者有些特殊字符在键盘上可以找到,但在 XHTML 中有特殊的含义,如>、& 等。这样可以用特殊符号来表示,实现输入的特殊字符,避免信息的混淆。常见特殊字符代码如表 3.7 所示。

表 3.7 常见特殊字符代码

特殊字符	字符代码	数字代码	特殊字符	字符代码	数字代码
<	<	&# 60;	©	&.copy;	&# 169;
>	&.gt;	&# 62;	®	&.reg;	&# 174;
&	&.amp;	&# 38;	空格	&.nbsp;	&# 160;
“	&.quot;	&# 34;			

在建立网页文件时,利用键盘上的空格键输入多个空格时,无论输入空格有多少个,显示时只有一个空格。如果需要多个空格,利用空格的特殊字符代码“&.nbsp;”,实现要完成的效果。

【例 3.8】 特殊字符的表现方法。

```
<body>
  <p>    How are you ?</p>
  <p>&.nbsp; &.nbsp; How &.nbsp; &.nbsp; are &.nbsp; &.nbsp; you &.nbsp;
&.nbsp;?</p>
  <hr width="90%" size="4" align="center"/>
  <address>
    &.copy;2015 版权所有 XHTML 学习
  </address>
</body>
```

程序运行效果如图 3.9 所示。

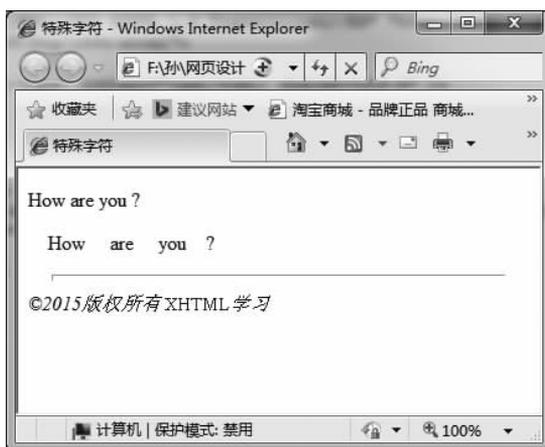


图 3.9 特殊字符的显示效果

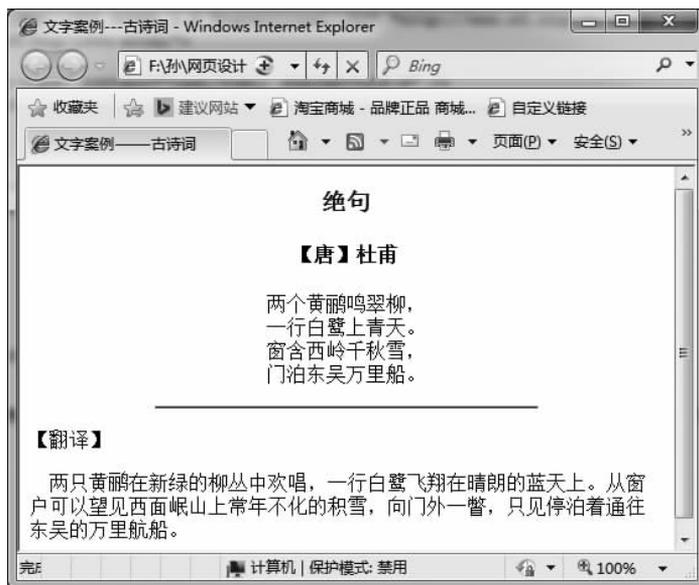


图 3.10 古诗排版的效果图

3.3 列表

在创建网页时,要清晰展现一些要点,使得网页内容整洁而有条理,就要能够合理且灵活地使用列表元素,让文件按照一定的规则和格式排序,方便浏览者阅读。

3.3.1 无序列表

无序列表是指列表内容可以按任意顺序排列。每一列表项前不是用连续编号,而是用一个特定符号来标记。通常是在每一列表项前加上一个小圆点儿(•)。

注意:在 XHTML 1.0 中,如果文档类型声明为严格类型,则标签的 type 属性不被支持。

1. 基本用法

无序列表的基本用法如下:

```
<ul type="值">
  <li>…</li>
  <li>…</li>
  <li>…</li>
  :
</ul>
```

其中,标签的 type 属性决定了每一列表项前面的标记符号。常见的标签的 type 属性取值如表 3.8 所示。

表 3.8 标签的 type 属性取值

type	作用
type="disc"	设置列表标记为 • (默认值)
type="circle "	设置列表标记为 ◦
type="square"	设置列表标记为 ◼

2. 实例代码

【例 3.10】 无序列表的应用。

```
<body>
  <p><strong>宋词鉴赏</strong></p>
  <ul type="circle">
    <li>宴山亭</li>
    <li>浣溪沙</li>
    <li>踏莎行</li>
    <li>天仙子</li>
    <li>御街行</li>
  </ul>
</body>
```

程序运行效果如图 3.11 所示。

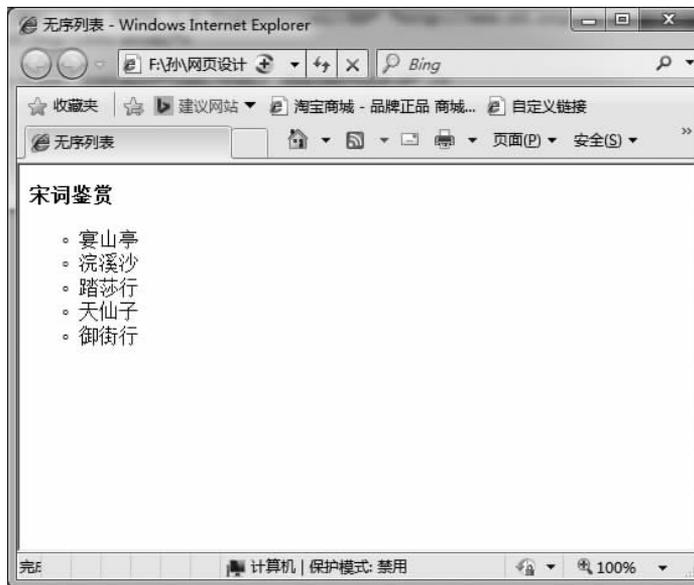


图 3.11 无序列表效果图

3.3.2 有序列表

有序列表用于对网页中的一些内容进行编号排列,以便读者可以清晰地了解每行的顺序。有序列表的实现方法与无序列表相似,只是用标签来代替无序列表中的。

注意: 在 XHTML 1.0 中,如果文档类型声明为严格类型,则标签的 start 和 type 属性不被支持。

1. 基本用法

有序列表的基本用法如下:

```
<ol type="值" start="值">
  <li>...</li>
```

```

    <li>...</li>
    <li>...</li>
    ⋮
</ol>

```

其中, type 属性的默认取值为阿拉伯数字, start 属性的取值为数字, 默认从 1 开始编号。元素的 type 属性的取值及作用如表 3.9 所示。

表 3.9 元素的 type 属性取值及作用

type	作用	type	作用
type="1"	设置列表编号为阿拉伯数字	type="a"	设置列表编号为小写英文字母
type="I"	设置列表编号为大写罗马数字	type="A"	设置列表编号为大写英文字母
type="i"	设置列表编号为小写罗马数字		

2. 实例代码

【例 3.11】 实现介绍图书馆信息服务的页面, 服务项目被排列成有序列表。

```

<body>
  <h2 align="center">唐诗鉴赏</h2>
  <center>
    <ol type="a" start="2">
      <li>送杜少府之任蜀州</li>
      <li>回乡偶书</li>
      <li>登鹳雀楼</li>
      <li>过故人庄</li>
      <li>望庐山瀑布</li>
    </ol>
  </center>
</body>

```

程序运行效果如图 3.12 所示。



图 3.12 有序列表运行效果图