

# 单元一 HTML 的基本标签

在当今社会中,Web 页成为网络信息共享和发布的主要形式,而 HTML(Hyper Text Markup Language, 超文本标记语言或叫超文本标签语言)是用来描述网页的一种语言,是创建 Web 页的基础。HTML 不是一种编程语言,而是一种标记语言(Markup Language),标记语言是一套标记标签,HTML 使用标记标签来描述网页。HTML 标记标签通常被称为 HTML 标签,本单元将介绍 HTML 的基本结构、组成 HTML 文档的各类常用标签以及相关标准。本单元的重点是各类标签的基本语法,学习 HTML 最好的方式就是边学边做实验,多练习是记住这些标签及其语法的最好方法。

## 单元要点

- 静态网页的开发环境
- 行级标签和块级标签的使用

## 技能目标

- 能使用 Dreamweaver 编写 HTML 代码
- 能使用各种基本标签建立简单网页

## 1.1 工作场景导入

### 【工作场景 1】

实现李清照的如梦令和作者简介的网页效果,如工作场景图 1 所示。

### 【工作场景 2】

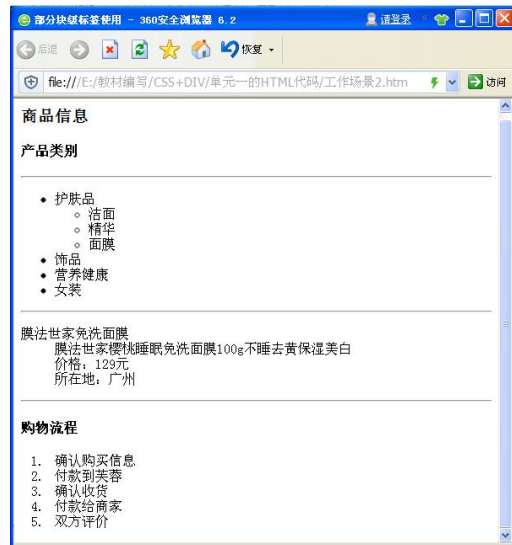
实现“产品信息”的网页效果,如工作场景图 2 所示。

### 【工作场景 3】

实现“促销信息”的网页效果,如工作场景图 3 所示。



工作场景图 1 李清照宋词赏析



工作场景图 2 商品信息



工作场景图 3 相宜本草促销信息

## 1.2 技术与知识准备

### 1.2.1 HTML 文件的基本结构

要深入学习 HTML，首先我们来了解一下什么是 HTML 以及 HTML 和浏览器的关系。

### 1. HTML 及其特点

HTML 被称为超文本标签语言，它包括很多标签，例如<p>段落标签、<h1>一级标题标签，告诉浏览器如何显示页面，它是网页制作的“核心语言”，是目前网络上应用最为广泛的语言，也是构成网页文档的主要语言。HTML 文档是由 HTML 标签和纯文本组成，HTML 标签可以说明段落、图形、表格、链接等，HTML 文档也被称为网页。它具备以下特点：

(1) 简易性：HTML 版本升级采用超集方式，从而更加灵活方便，并且各类 HTML 标签简单易学，方便网站制作者学习开发。

(2) 可扩展性：HTML 语言的广泛应用带来了加强功能、增加标识符等要求，HTML 采取子类元素的方式，为系统扩展提供保证。

(3) 平台无关性：这是 HTML 语言的最大优点，也是当今 Internet 盛行的原因之一。虽然 PC 机大行其道，但使用 MAC 等其他机器的大有人在，HTML 可以使用在广泛的平台上，它包括“硬件”平台无关性和“软件”平台无关性，不管你的计算机是普通的个人电脑，还是用于专业的苹果机，不管你的操作系统是常见的 Windows 还是高端 UNIX 或 Linux（一般用于服务器），HTML 文档都可以得到广泛的应用和传输。

### 2. HTML 和浏览器的关系

有了 HTML 源代码，还需要一个“解释和执行”的工具，而浏览器就是用来解释并执行显示 HTML“源码”的工具，随着市场的竞争，目前市场上的浏览器很多，主要有微软公司的 IE（Internet Explorer），网民大多数人都在使用 IE，这要感谢它对 Web 站点强大的兼容性。谷歌浏览器 Chrome 是由 Google 公司开发的网页浏览器，浏览速度在众多浏览器中走在前列，属于高端浏览器，谷歌浏览器在 2012 年 8 月份市场份额正式超过 IE 浏览器，跃居第一。Mozilla Firefox（火狐浏览器）现在是市场占有率第三的浏览器，仅次于微软的 Internet Explorer 和 Google 的 Chrome，搜狗浏览器是首款给网络加速的浏览器，本书使用 IE 8.0 和 360 安全浏览器进行介绍，关于浏览器的其他介绍学员可以上网获取更多资料。



图 1.1 常用浏览器图标

浏览器兼容性问题又被称为网页兼容性或网站兼容性问题，指网页在各种浏览器上的显示效果可能不一致而产生浏览器和网页间的兼容问题。在网站的设计和制作中，做好浏览器兼容，才能够让网站在不同的浏览器下都正常显示。而对于浏览器软件的开发和设计，浏览器对标准的更好兼容能够给用户更好的使用体验。规范化书写 HTML 可以解决浏览器兼容的问题，也可以避免给后台制作带来不必要的麻烦，随着课程的深入我们再予以介绍。

如果希望查看某个页面的“源代码”，我们可以通过点击浏览器（例如 IE）的菜单“查看”→“源文件”选项或右击，在弹出的菜单中选择“查看源文件”命令。下面将介绍 HTML 文档的基本结构。

### 3. HTML 的基本结构

HTML 文件的结构包括头部（head）、主体（body）两大部分，其中头部描述浏览器所需

的信息，而主体则包含所要说明的具体内容，如图 1.2 所示，整个 HTML 包括头部<head>和主体<body>两部分。头部包括网页标题（title）等基本信息，主体包括网页的内容信息（如所要呈现的图片、文字等）。注意大部分标签都以“<>”开始，以“</>”结束，要求成对出现，并且标签之间要有缩进，体现层次感，以便阅读和修改。

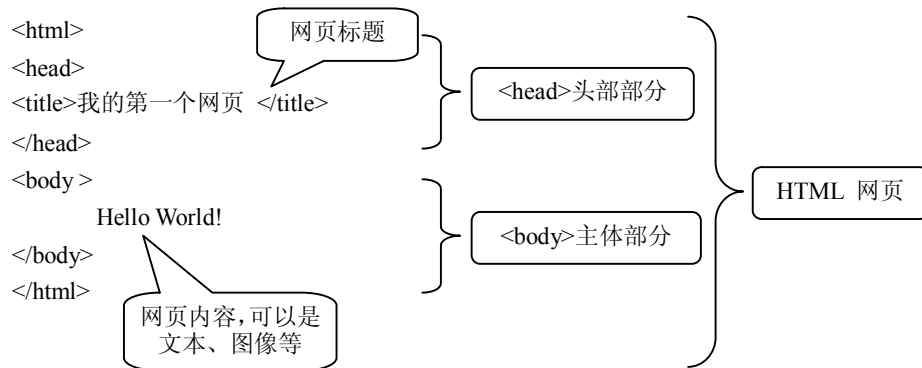


图 1.2 HTML 的基本结构

### 1.2.2 编辑工具

了解了 HTML 文档的基本结构以后，下面介绍常用的 HTML 代码编辑工具。

#### 1. 记事本

记事本是 Windows 自带安装的编辑附件，使用简单方便，实际项目开发中常用于代码较少的编辑或维护。使用记事本编辑 HTML 文档的步骤如下：

- (1) 在 Windows 中打开记事本程序：开始→所有程序→附件→记事本。
- (2) 在记事本中输入 HTML 代码，如图 1.3 所示。



图 1.3 在记事本里编辑 HTML 网页

(3) 单击菜单“文件”→“保存”命令，弹出“另存为”对话框，如图 1.4 所示，将上述文档保存为\*.html 的 html 文档，如 my-firstpage.html。需要注意，因记事本默认保存文档后缀名为“\*.txt”，所以需要英文的双引号（"）将文件名括起来。既可以使用.htm 也可以使用.html 扩展名。两者没有区别，完全根据自己的喜好。

(4) 双击保存的 html 文档，Windows 将自动调动浏览器软件（如 IE）打开 html 文档，如图 1.5 所示，也可以先启动浏览器，然后选择“文件”→“打开文件”命令。



图 1.4 “另存为”对话框



图 1.5 我的第一个网页

## 2. Dreamweaver

可以使用专业的 HTML 编辑器来编辑,如 Adobe Dreamweaver、Microsoft Expression Web、CoffeeCup HTML Editor。Dreamweaver 是由 Macromedia 公司开发的一款所见即所得的网页编辑器,与二维动画设计软件 Flash,专业网页图像设计软件 Fireworks,并称为“网页三剑客”。它有“所见即所得”的网页编辑器的优点,即直观、使用方便、容易上手,本书所有单元都将采用 Dreamweaver 作为 HTML 文档的编辑工具,启动方法为:单击桌面左下角“开始”→“程序”→“Macromedia”→“Macromedia Dreamweaver CS5”选中后单击就可以打开 Dreamweaver 了,或者下载绿色版,绿化后直接双击 Dreamweaver.exe 运行。

对于上述第一个网页在 Dreamweaver 中的编辑如下:选择“文件”→“新建”命令,单击“常规”选项卡,如图 1.6 所示,单击“创建”按钮,弹出如图 1.7 所示的窗口,一般情况选用拆分视图的编辑环境,编辑好且保存之后如图 1.8 所示,然后根据图 1.9 选择一种浏览器进行浏览,浏览效果如图 1.10 所示。

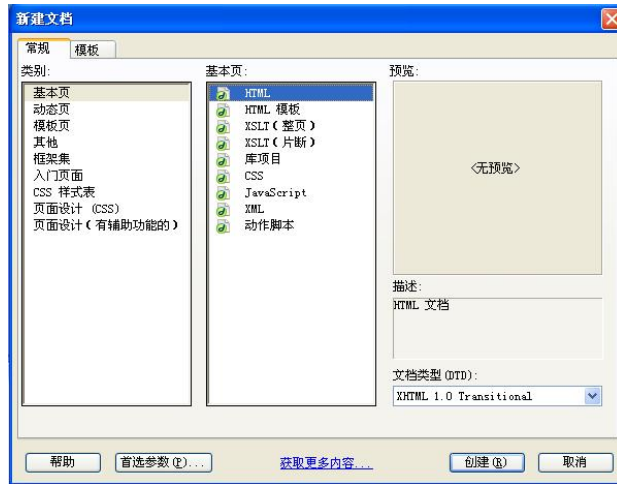


图 1.6 “常规”选项卡

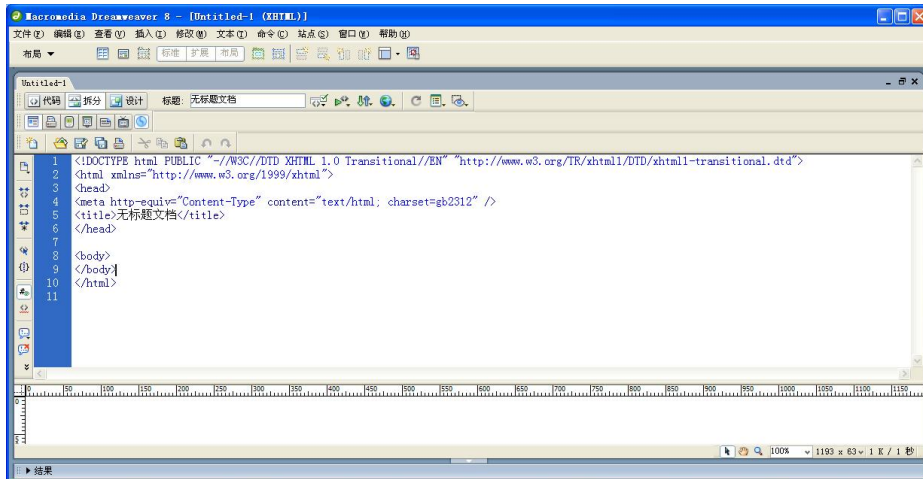


图 1.7 未编辑、保存的默认网页

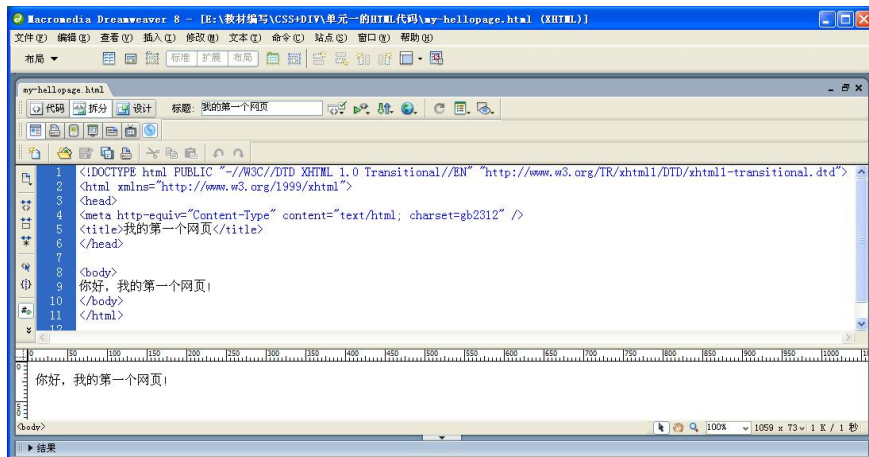


图 1.8 编辑、保存好的网页

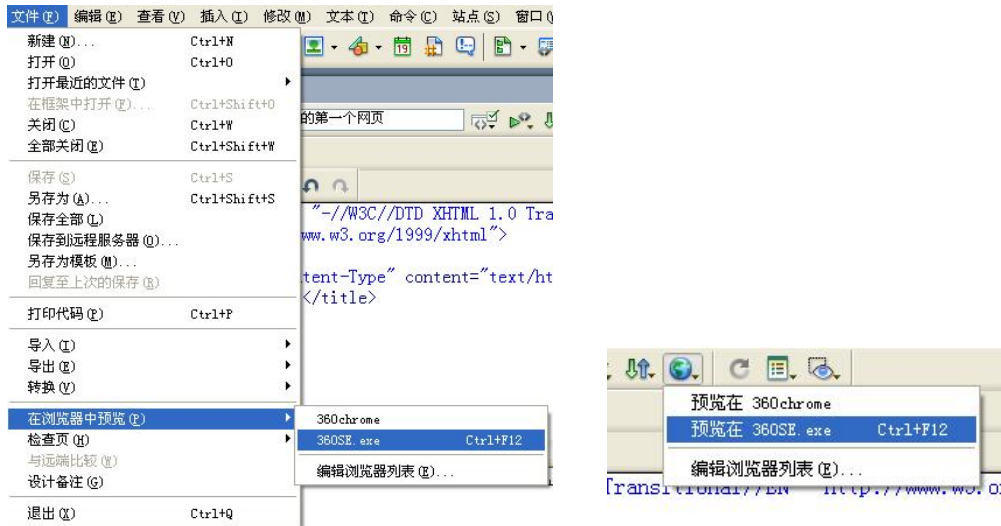


图 1.9 浏览器的选择

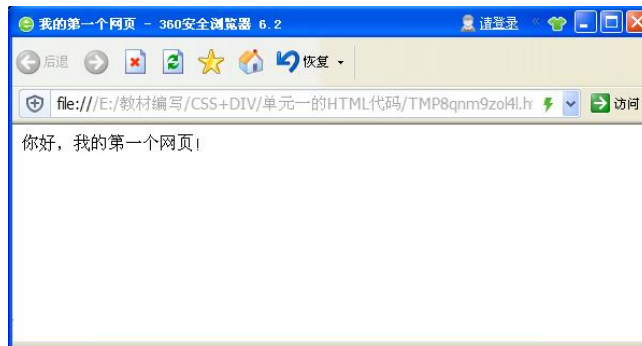


图 1.10 浏览器中的显示效果

### 3. 其他编辑器

除了记事本, Notepad (PC) 或 TextEdit (Mac) 也是文本编辑器。Notepad是在微软视窗下的一个纯文本编辑器。TextEdit (Mac) 是苹果电脑 Mac OS X 操作系统自带的 TextEdit (文本编辑器)。

#### 1.2.3 网页的摘要信息

网页一般包含大量文字及图片等信息内容, 和报纸一样, 它需要一个简短的摘要信息, 方便用户浏览和查找。如果希望自己的网页能被百度、Google 等搜索引擎搜索, 或提高在搜索结果中的排名, 那么在制作网页时更需要注意编写网页的摘要信息。网页的摘要信息一般放在 HTML 文档的头部 (head) 区域内容, 主要通过如下两个标签进行描述。

##### 1. <title>标签

使用该标签描述网页的标题, 类似一篇文章的标题, 一般为一个简洁的主题, 并能吸引读者有兴趣读下去。例如, 搜狐网站的主页, 对应的网页标题为: <title>搜狐-中国最大的门户网站</title>, 打开网页后, 将在浏览器窗口的标题栏显示网页标题。

## 2. <meta/>标签

<meta/>标签是 head 区的一个辅助性标签，是一个空标签，没有相应的结束标签，规范写法用<meta/>，表示开始和结束，其中“/”代表结束。该标签不包含任何内容，也就是指<meta/>标签描述的内容并不显示，其目的是方便浏览器解析或利于搜索引擎搜索，使用该标签描述网页的具体摘要信息，包括文档内容类型、字符编码信息、搜索关键字、网站提供的功能和服务的详细描述等。<meta/>标签的属性定义了与文档相关联的名称/值对。

(1) 文档内容类型，字符编码信息。

对应的 HTML 代码如下：

```
<meta http-equiv="Content-type" content="text/html; charset=gb2312"/>
```

其中属性“http-equiv”提供“名称/值”中的名称，“content”提供“名称/值”中的值，所以上述 HTML 代码的含义如下：名称 Content-Type（文档内容类型），值 text/html；charset=gb2312（文本类型的 HTML 类型，字符编码为简体中文）中 charset 表示字符编码，常用字符编码有如下 4 种：

- gb2312：简体中文，一般用于包含中文和英文的页面。
- ISO-885901：纯英文一般用于只包含英文的页面。
- big5：繁体，一般用于带有繁体字的页面。
- utf-8：国际通用字符编码，同样适用于中文和英文的页面。与 gb2312 编码相比，utf-8 的国际通用性更好，但字符编码的压缩比较低，对网页性能有一定的影响。需要注意，不正确的编码设置将带来网页乱码。

(2) 搜索关键字和描述信息。

对应的 HTML 代码如下：

```
<meta name="keywords" content="IT 培训"/>
```

```
<meta name="description" content="国内最大的 IT 教育集团，致力于为中国培养优秀的 IT 技术人才"/>
```

实现的方式仍然为“名称/值”对的形式，其中 keywords 表示搜索关键字，description 表示网站内容的具体描述。通过提供搜索关键字和内容描述信息，方便搜索引擎的搜索。

meta 标签共有两个属性，它们分别是 name 属性和 http-equiv 属性，不同的属性有不同的参数值，这些不同的参数值就实现了不同的网页功能。

(1) name 属性。

name 属性主要用于描述网页，与之对应的属性值为 content，content 中的内容主要是便于搜索引擎机器人查找信息和分类信息用的。

meta 标签的 name 属性语法格式如下：

```
<meta name="参数" content="具体的参数值">
```

其中 name 属性主要有以下几种参数：

1) keywords（关键字）。

说明：keywords 用来告诉搜索引擎你网页的关键字是什么。

举例：<meta name="keywords" content="政治,经济,科技,文化,卫生,情感,心灵,娱乐,生活,社会,企业,交通">，按照搜索引擎的工作原理，搜索引擎首先派出机器人自动检索页面中的 keywords，设置好这些关键字后，搜索引擎将会自动把这些关键字添加到数据库中，并根据这些关键字的密度来进行合适的排序。因此，我们必须设置好关键字，来提高页面的搜索点击率。



## 2) description (网站内容描述)。

说明: description 用来告诉搜索引擎网站的主要内容。网站内容描述 (description) 的设计要点:

①网页描述为自然语言而不是罗列关键词 (与 keywords 设计正好相反)。

②尽可能准确地描述网页的核心内容。

## 3) robots (机器人向导)。

说明: robots 用来告诉搜索机器人哪些页面需要索引, 哪些页面不需要索引。content 的参数有 all, none, index, noindex, follow, nofollow, 默认是 all。

举例: `<meta name="robots" content="none">`。

## 4) author (作者)。

说明: 标注网页的作者。

举例: `<meta name="author" content="zys666,zys666@21cn.com">`

## (2) http-equiv 属性。

http-equiv 顾名思义, 相当于 http 的文件头作用, 它可以向浏览器传回一些有用的信息, 以帮助正确和精确地显示网页内容, 与之对应的属性值为 content, content 中的内容其实就是各个参数的变量值。

meta 标签的 http-equiv 属性语法格式如下:

`<meta http-equiv="参数" content="参数变量值">`

其中 http-equiv 属性主要有以下几种参数:

### 1) Expires (期限)。

说明: 可以用于设定网页的到期时间。一旦网页过期, 必须到服务器上重新传输。

用法: `<meta http-equiv="expires" content="Fri,12 Jan 2001 18:18:18 GMT">`。

注意: 必须使用 GMT 的时间格式。

### 2) Pragma (cache 模式)。

说明: 禁止浏览器从本地计算机的缓存中访问页面内容。

用法: `<meta http-equiv="Pragma" content="no-cache">`。

注意: 这样设定, 访问者将无法脱机浏览。

### 3) Refresh (刷新)。

说明: 自动刷新并指向新页面。

用法: `<meta http-equiv="Refresh" content="2;URL=http://www.root.net">` (注意后面的引号, 分别在秒数的前面和网址的后面)。

注意: 其中的 2 是指停留 2 秒钟后自动刷新到 URL 网址。

### 4) Set-Cookie (cookie 设定)。

说明: 如果网页过期, 那么存盘的 cookie 将被删除。

用法: `<meta http-equiv="Set-Cookie" content="cookievalue=xxx; expires=Friday,12-Jan-2001 18:18:18 GMT; path="/">`。

注意: 必须使用 GMT 的时间格式。

### 5) Window-target (显示窗口的设定)。

说明: 强制页面在当前窗口以独立页面显示。

用法: `<meta http-equiv="Window-target" content="_top">`。

注意: 用来防止别人在框架里调用自己的页面。

6) Content-Type (显示字符集的设置)。

说明: 设定页面使用的字符集。

用法: `<meta http-equiv="Content-Type" content="text/html; charset=gb2312">`

7) Content-Language (显示语言的设置)。

用法: `<meta http-equiv="Content-Language" content="zh-cn" />`

在团队开发中, 需要注意养成良好的习惯。书写顺序应该成对书写, 从外层写到内层, 结构清晰, 利于阅读和调试排错。例如编写:

```
<html>
  <head>
    <title>标题</title>
  </head>
</html>
```

推荐书写顺序如下:

①`<html></html>`

②写入 `head` 再排版缩进(默认缩进为 2 个空格或 4 个空格, 各个团队的具体规定不一样)。

```
<html>
  <head></head>
</html>
```

③`title` 先写标签, 再写里面的内容:

```
<html>
  <head>
    <title></title>
  </head>
</html>
```

④同理再编写`<body>`中的其他标签。

#### 1.2.4 块级标签

介绍了头部 (`head`) 的常用标签后, 下面再介绍主体 (`body`) 内常用的各类标签, 如图 1.11 所示, 从页面布局和显示外观的角度看, 一个页面的布局就类似一篇报纸的排版, 需要分为多个区块, 大的区块再细分为小区块。块内为多行逐一排列的文字、图片、超链接等内容, 这些区块一般称为块级元素, 而区块内的文字、图片或超链接等一般称为行级元素, 页面这种布局结构, 其本质上是由各种 HTML 标签组织完成的, 因此本文将 HTML 标签分为相应的块级标签和行级标签 (有的教材也称为块级元素和行级元素), 以方便理解后续单元讲解的页面布局。

顾名思义, 块级标签显示的外观按“块”显示, 具有一定的高度和宽度, 例如`<div>`块标签、`<p>`段标签等, 行级元素显示的外观按“行”显示, 类似文本的显示, 例如`<img>`图片标签、`<a>`超链接标签等。和行级标签相比, 块级标签具有如下特点:

- (1) 前后断行显示, 如图 1.11 所示。块级标签比较“霸道”, 默认状态占据一整行。
- (2) 具有一定的宽度和高度, 可以通过设置块级标签 `width`、`height` 属性来控制。

块级标签常用于作容器，即可再容纳其他块级标签和行级标签，而行级标签一般用于组织内容，即只能用于“容纳”文字、图片或其他行级标签。



图 1.11 页面中的块级元素和行级元素

从页面布局的角度，块级标签又可细分为基本的块级标签和常用于页面布局的块级标签。

#### 1.2.4.1 基本的块级标签

基本的块级标签包括标题标签、段落标签及水平线标签。

##### 1. 标题标签<h1>~<h6>

在 HTML 文档中，标题很重要，是通过 <h1>~<h6> 等标签进行定义的。<h1> 定义最大的标题，<h6>定义最小的标题，浏览器会自动地在标题的前后添加空行，默认情况下，HTML 会自动地在块级元素前后添加一个额外的空行，比如段落、标题元素前后。因为用户可以通过标题来快速浏览您的网页，所以用标题来呈现文档结构是很重要的。应该将 h1 用作主标题（最重要的），其后是 h2（次重要的），再其次是 h3，以此类推。例如，一级标题采用<h1>，如还有二级标题采用<h2>，HTML 共提供了六级标题，并赋予了标题一定的外观：所有标题字体加粗。<h1>字号最大，<h6>字号最小。这 6 个标题标签都有对应的结束标签，通过结束标签来关闭。HTML 清楚地标记某个元素在何处开始，并在何处结束，不论对您还是对浏览器来说，都会使代码更容易理解。例如，如图 1.12 所示，其所对应的 HTML 代码如示例 1.1 所示，代码在浏览器中的效果就是图 1.12 的效果。

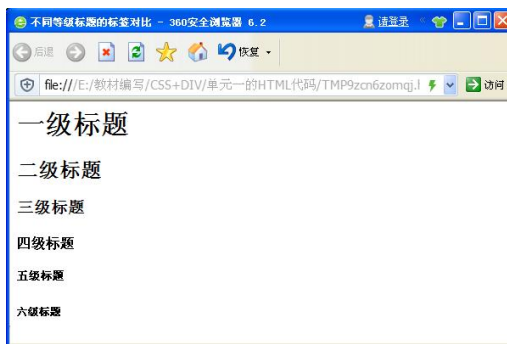


图 1.12 不同级别的标题输出的结果

### 示例 1.1

```
<html>
<head>
<title>不同等级标题的标签对比</title>
</head>
<body>
  <h1>一级标题</h1>
  <h2>二级标题</h2>
  <h3>三级标题</h3>
  <h4>四级标题</h4>
  <h5>五级标题</h5>
  <h6>六级标题</h6>
</body>
</html>
```

### 2. 段落标签<p>

段落是通过<p>标签定义的。顾名思义，段落标签表示一段文字的内容，如图 1.13 所示，其所对应的 HTML 代码如下例 1.2 所示，应用了标题和段落标签。实际上，一个段落中可以包含多行文字，文字内容将随浏览器窗口的大小自动换行，从图 1.13 可以看出，黑色加粗的字是标题，标题下面是段落，它们都属于块级标签，隔行显示。



图 1.13 段落标签的应用

### 示例 1.2

```
<html>
<head>
<title>优美段落</title>
</head>
<body>
  <h1>阳光，不只来自太阳，也来自我们的心</h1>
  <p> 阳光，不只来自太阳，也来自我们的心。心里有阳光，能看到世界美好的一面；心里有阳光，能与有缘的人心心相映；心里有阳光，即使在有遗憾的日子，也会保留温暖与热情；心里有阳光，才能提升生命品质。自信、宽容、给予、爱、感恩吧，让心里的阳光，照亮生活中的
```

点点滴滴，阳光的心，造就阳光的命运。

```
</p>
<h1>静静的心里，都有一道最美丽的风景</h1>
<p> 静静的心里，都有一道最美丽的风景。尽管世事繁杂，心依然，情怀依然；尽管颠簸流离，脚步依然，追求依然；尽管岁月沧桑，世界依然，生命依然。守住最美风景，成为一种风度，宁静而致远；守住最美风景，成为一种境界，悠然而豁达；守住最美风景，成为一种睿智，淡定而从容。
</p>
</body>
</html>
```

### 3. 水平线标签<hr/>

<hr/>标签在 HTML 页面中创建水平线。顾名思义，水平线标签表示一条水平线，注意该标签比较特殊，没有结束标签，直接使用“<hr/>”表示标签的开始和结束。<hr/>可用于分隔内容，分隔文章中的小节。例如，为了让版面更加清晰直观，可以如图 1.14 那样加一条水平分隔线，对应的 HTML 代码如示例 1.3 所示。



图 1.14 水平线的应用

### 示例 1.3

```
<html>
<head>
<title>优美段落</title>
</head>
<body>
  <h1>阳光，不只来自太阳，也来自我们的心</h1>
  <p>阳光，不只来自太阳，也来自我们的心。……</p>
  <hr/>
  <h1>静静的心里，都有一道最美丽的风景</h1>
  <p>静静的心里，都有一道最美丽的风景。……</p>
</body>
</html>
```

### 1.2.4.2 常用于布局的块级标签

这类标签包括有序列表、无序列表、定义列表、表格、表单、分区标签 (<div>), 它们常用于布局网页, 组织 HTML 的内容结构。

#### 1. 有序列表标签<ol>

是一个项目的列表, 列表项目使用数字进行标记。有序列表标签表示多个并列的列表项, 它们之间有明显的先后顺序, 有序列表始于 <ol> 标签, 每个列表项始于 <li> 标签, 也就是使用<ol>、</ol>表示有序列表, <li>、</li>表示各列表项。例如描述网页后台的几种语言, 如图 1.15 所示, 对应的 HTML 代码如示例 1.4 所示。



图 1.15 有序列表

#### 示例 1.4

```
<html>
<head>
<title>有序列表</title>
</head>
<body>
<h3>网页后台的学习: </h3>
<ol>
  <li>ASP</li>
  <li>ASP.net</li>
  <li>PHP</li>
  <li>JSP</li>
</ol>
</body>
</html>
```

#### 2. 无序列表标签<ul>

无序列表和有序列表类似, 但多个并列的列表项之间没有先后顺序, 项目使用粗体圆点 (典型的小黑圆圈) 进行标记。无序列表始于<ul>标签, 每个列表项始于<li>。也就是使用<ul>、</ul>表示无序列表, <li>、</li>表示各列表项。例如描述某个网站的使用帮助, 如图 1.16 所示, 对应的 HTML 代码如示例 1.5 所示。



图 1.16 无序列表

### 示例 1.5

```
<html>
<head>
<title>无序列表</title>
</head>
<body>
<h4>一个无序列表: </h4>
<ul>
  <li>咖啡</li>
  <li>茶</li>
  <li>牛奶</li>
</ul>
</body>
</html>
```

有序列表和无序列表也可以互相嵌套，如示例 1.6 所示，注意嵌套后无序列表的符号选择可以使用 `type` 属性，效果如图 1.17 所示。

### 示例 1.6

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>无序列表的嵌套</title>
</head>
<body>
<h4>一个无序列表: </h4>
<ul>
  <li>
    咖啡
    <ul>
      <li>卡布其诺</li>
      <li>浓缩咖啡</li>
      <li>拿铁</li>
      <li>蓝山</li>
      <li>摩卡</li>
    </ul>
  </li>
</ul>
```

```
</li>
<li>
  茶
  <ul type="square">
    <li>绿茶</li>
    <li>红茶</li>
    <li>乌龙茶</li>
  </ul>
</li>
<li>牛奶</li>
</ul>
</body>
</html>
```

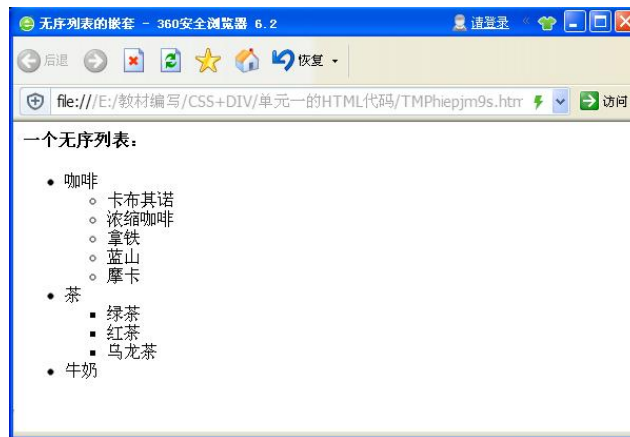


图 1.17 无序列表的嵌套

### 3. 定义列表标签<dl>

顾名思义，自定义列表不仅仅是一列项目，而是项目及其注释的组合，用于描述某个术语或产品的定义或解释，例如计算机、Java 语言、MP4 电子产品的定义等。它使用<dl>、</dl>表示定义列表，<dt>、</dt>表示术语，<dd>、</dd>表示术语的具体描述。例如，对春节的描述如图 1.18 所示，对应的 HTML 代码如示例 1.7 所示。

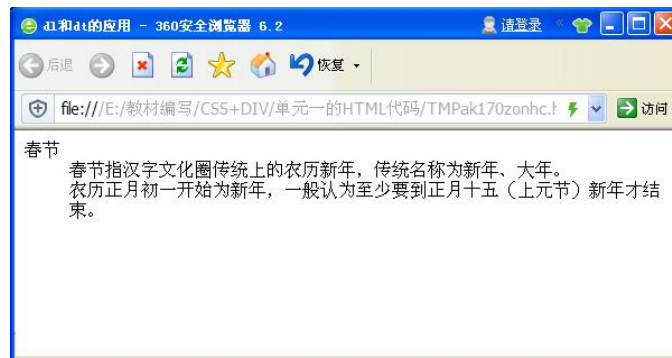


图 1.18 dl-dt-dd 的应用



### 示例 1.7

```
<html>
<head>
<title>dl 和 dt 的应用</title>
</head>
<body>
<dl>
<dt>春节</dt>
<dd>春节指汉字文化圈传统上的农历新年，传统名称为新年、大年。</dd>
<dd>农历正月初一开始为新年，一般认为至少要到正月十五（上元节）新年才结束。</dd>
</dl>
</body>
</html>
```

在实际应用中，定义列表还被扩展到图文混编的场合，如图 1.19 所示。将图片作为术语标题<dt>，文字内容作为术语描述<dd>。这种局部布局结构将在后续章节进行详细讲解。

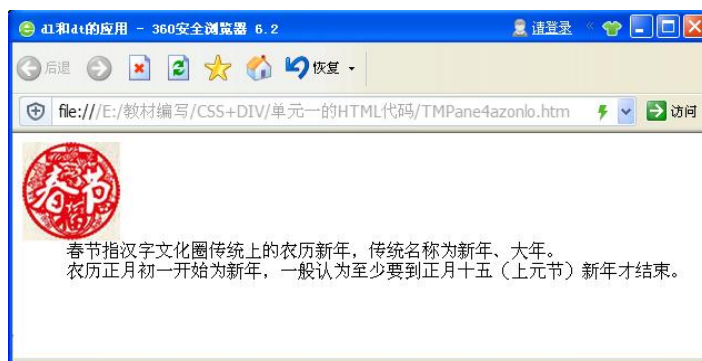


图 1.19 定义列表标签用于图文混编的场合

### 4. 表格标签<table>

顾名思义，表格标签用于描述一个表格，它使用规整的数据显示，如图 1.20 所示。它使用<table>、</table>表示表格，<tr>、</tr>表示行，<td>、</td>表示列。表格的用法将在单元三进行详细介绍。

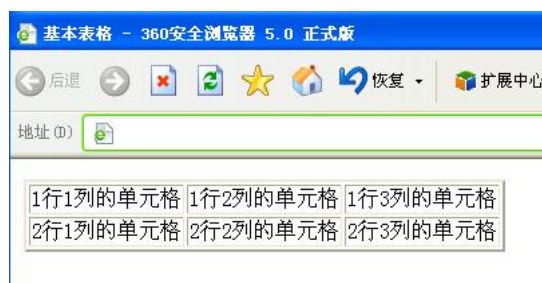


图 1.20 表格标签的应用

### 5. 表单标签<form>

表单标签用于描述需要用户输入的页面内容，例如图 1.21 注册页面，它使用<form>，

</form>表示表单，<input/>表示输入内容项。表单的具体用法将在单元 2 进行详细介绍。

图 1.21 表单标签的应用

## 6. 分区标签<div>

前几类块级标签一般用于组织小区块的内容，为了方便管理，数目众多的小区块还需要放到一个大区块中进行布局。分区标签<div>常用于页面布局时对区块的划分，它相当于一个大的容器，可以容纳无序列表、有序列表、表格等块级标签，同时也可容纳普通的段落、标题、文字、图片等内容，如图 1.22 所示，对应的 HTML 代码如示例 1.8 所示。由于<div>标签不像<h1>等标签，没有明显的外观效果，所以特意添加“style”样式属性，设置<div>标签的宽、高尺寸以及背景色。样式方面的用法将在后续章节详细介绍。

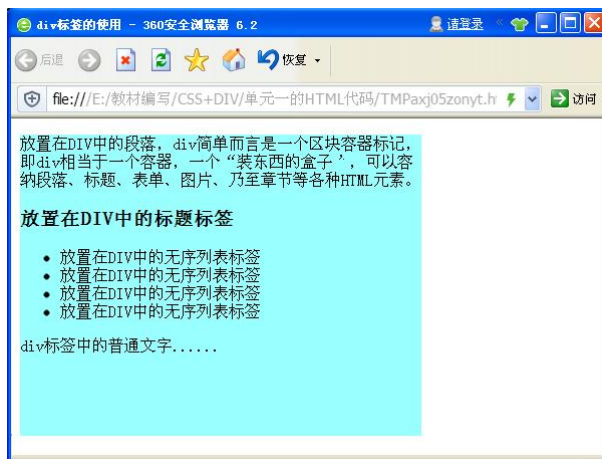


图 1.22 div 标签的应用

### 示例 1.8

```
<html>
<head>
<title>div 标签的使用</title>
</head>
<body>
```

```
<div style="width:400px; height:300px; background:#9FF">
<p>放置在 DIV 中的段落, div 简单而言是一个区块容器标记, 即 div 相当于一个容器, 一个“装东西的
盒子”, 可以容纳段落、标题、表单、图片、乃至章节等各种 HTML 元素。
</p>
<h3>放置在 DIV 中的标题标签</h3>
<ul>
<li>放置在 DIV 中的无序列表标签</li>
<li>放置在 DIV 中的无序列表标签</li>
<li>放置在 DIV 中的无序列表标签</li>
<li>放置在 DIV 中的无序列表标签</li>
</ul>
div 标签中的普通文字.....
</div>
</body>
</html>
```

到目前为止, 我们学习了常用的各种块级标签。在实际开发中, 常使用<div>进行分区, ul-li 或 ol-li 实现无序或有序的项目列表, dl-dt-dd 实现图文混编, table-tr-td 实现规整数据的显示。由此, 在页面局部布局中, 形成了如下四种常用的块状结构:

- div-ul(ol)-li: 常用于分类导航或菜单等场合。
- div-dl-dt-dd: 常用于图文混编等场合。
- table-tr-td: 常用于规整数据的显示场合。
- form-table-tr-td: 常用于表单布局的场合。

这四种块状结构非常实用, 它们的具体应用还将在后续章节进行深入讲解和训练。

### 1.2.5 行级标签

行级标签也称行内标签。使用块级标签为页面“搭建完成组织结构”后, 往每个小区块添加内容时, 就需要用到行级标签。行级标签类似文本的显示, 按“行”逐一显示。常用的行级标签包括图像标签<img/>, 超链接标签<a>, 范围标签<span>, 换行标签<br/>以及和表单相关的输入框标签<input/>, 文本域标签<textarea>等, 表单涉及的行级标签将在单元 2 详细介绍。

#### 1.2.5.1 图像标签<img/>

在日常生活中, 使用比较多的图像格式有四种, 即 JPG、GIF、BMP、PNG。在网页中使用比较多的是 JPG、GIF 和 PNG, 大多数浏览器都可以显示这些格式的图像。

图像标签是<img/>, 要在页面上显示图像, 需要使用源属性 (src), src 指"source"。源属性的值是图像的 URL 地址。

定义图像的语法格式如下:

```

```

其中图片地址指存储图像的位置, 如果名为"boat.gif"的图像位于 www.w3school.com.cn 的 images 目录中, 那么地址为 http://www.w3school.com.cn/images/boat.gif。

alt 属性指定的替代文本, 表示图像无法显示 (例如, 图片路径错误或网速太慢等) 时的替代显示文本。这样, 即使图像无法显示, 用户还是可以看到网页丢失的信息内容, 如图 1.23 所示, 所以在制作网页时一般推荐和“src”配合使用。如图片叫春节.jpg, 如果写成春节 1.jpg,

``就找不到图片，浏览时就会看到图 1.23 的情景。

其次，使用“title”属性，鼠标滑过时显示的文字提示，还可以提供额外的提示或帮助信息，方便用户使用，如图 1.24 所示。设计和制作网页时，需从方便客户的角度考虑问题，用户体验已越来越成为 Web 设计和开发需要考虑的重要因素，用户体验的原则之一就是以用户为中心，并体现在细微之处。例如，使用

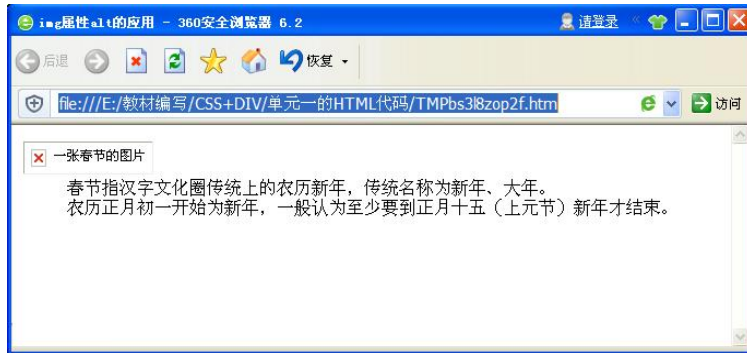


图 1.23 img 属性 alt 的应用

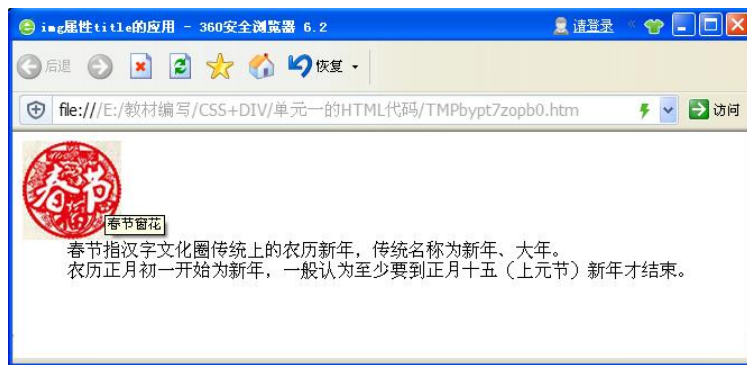


图 1.24 img 属性 title 的应用

### 示例 1.9

```
<html>
<head>
<title>img 属性 alt 和 title 的应用</title>
<body>
<dl>
<dt></dt>
<dd>春节指汉字文化圈传统上的农历新年，传统名称为新年、大年。</dd>
<dd>农历正月初一开始为新年，一般认为至少要正月十五（上元节）新年才结束。</dd>
</dl>
</body>
</html>
```

### 1.2.5.2 范围标签<span>

范围标签<span>用于标识行内的某个范围，是被用来修饰文档中的行内元素。span 没有固定的格式表现。当对它应用样式时，它才会产生视觉上的变化。示例 1.10 实现行内每个部分的特殊设置以区分其他内容，如图 1.25 所示。span 标签加入到 HTML 中的主要目的是用于样式表，所以当样式表失效时它就没有任何作用了。

#### 示例 1.10

```
<html>
<head>
<title>span 标签的应用</title>
<body>
  <dl>
    <dt></dt>
    <dd>春节指汉字文化圈传统上的农历新年，传统名称为
    <span style="color:#FF0000;font-size:30px;" >新年、大年</span>。
  </dd>
    <dd>农历正月初一开始为新年，一般认为至少要到正月十五（上元节）新年才结束。</dd>
  </dl>
</body>
</html>
```

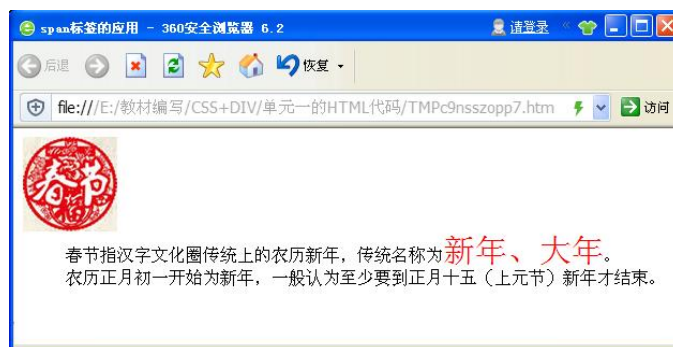


图 1.25 span 标签应用

### 1.2.5.3 换行标签<br/>

使用<br/>换行，但是不间隔行，<br/>标签只是简单地开始新的一行。当使用<br/>标签时，其后面的所有内容都将在下一行出现，br 是 line break 的意思，如图 1.26 所示，对应的代码如示例 1.11 所示。

#### 示例 1.11

```
<html>
<head>
<title>换行标签 br 的应用</title>
</head>
<body>
  <h1>阳光，不只来自太阳，也来自我们的心</h1>
  <p>阳光，不只来自太阳，也来自我们的心。<br/>心里有阳光，能看到到世界美好的一面；<br/>心里有阳光，能与有缘的人心心相映；<br/>心里有阳光，即使在有遗憾的日子，也会保留温暖与热情；<br/>心里
```

有阳光，才能提升生命品质。<br/>自信、宽容、给予、爱、感恩吧，让心里的阳光，照亮生活中的点点滴滴，阳光的心，造就阳光的命运。

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



图 1.26 br 换行标签应用

#### 1.2.5.4 超链接<a>标签

HTML 使用超级链接与网络上的另一个文档相连。几乎可以在所有的网页中找到链接。点击链接可以从一张页面跳转到另一张页面。超链接可以是一个字、一个词或一组词，也可以是一幅图像，可以点击这些内容来跳转到新的文档或者当前文档中的某个部分。当把鼠标指针移动到网页中的某个链接上时，箭头会变为一只小手。通过使用<a>标签在 HTML 中创建链接，a 是 anchor（锚）的第一个字母，有两种使用<a>标签的方式：

- (1) 通过使用 href 属性创建指向另一个文档的链接。
- (2) 通过使用 name 属性创建文档内的书签。

超链接的基本语法如下：

```
<a href="链接地址" >链接文本或图像</a>
```

属性说明如下：

**href:** 表示链接地址的路径，它指定链接的目标，可以是某个网址或文件的路径，也可以是某个锚点名称。

**链接文本或图像:** 单击该文本或图像，将跳转到 href 属性指定的链接地址，对应为<a>标签中的文字或图片。如<a href="http://www.baidu.com/">百度首页</a>，上面代码显示为：百度首页，点击这个超链接会把用户带到搜索引擎百度的首页。

对于链接路径，当单击某个链接时，将指向万维网上的文档，万维网使用称为 URL（Uniform Resource Location，统一资源定位器）的方式来定义一个链接地址。例如，一个完整的链接地址的常见形式为 http://www.baidu.com。

URL 地址的统一格式为 scheme://host.domain: port/path/filename。对上面的格式解释如下：

**scheme:** 表示各类通讯协议，例如，常用的是 http（超文本传输协议）、ftp（文件传输协议）。

**domain:** 定义因特网域名，以方便访问，例如 baidu.com 等。

**host:** 定义域中的主机名，如果被省略，http 协议默认的主机是 WWW。

**port:** 定义主机的端口，一般默认为 80 端口，可以省略。

**path:** 定义服务器上的路径, 例如 reg 目录。

**filename:** 定义文档的名称, 例如 register.html。

HTML 初学者经常会遇到这样一个问题, 如何正确引用一个文件。比如, 怎样在一个 HTML 网页中引用另外一个 HTML 网页作为超链接 (hyperlink)? 我们有相对路径和绝对路径两种书写方式。

**绝对路径:** 是以硬盘根目录或者站点根目录为参考点建立的路径, 指向目标地址的完整描述, 一般指向本站点外的文件, 如 `<a href="http://www.baidu.com">百度</a>`, 这种写法就是指链接指向站点外。

**相对路径:** 是以当前文件所在位置为参考点而建立的路径, 相对于当前页面的路径, 一般指向本站点内的文件, 所以一般不需要一个完整的 URL 地址的形式, 如 `<a href="login/login.html" >登录<a >链接`, 表示当前页面所在路径的“login”目录下的“login.html”页面, 假定当前页面所在的目录为“c:\site”, 则链接地址对应的页面为“c:/site/login/login.html”。相对路径经常用到下面两个特殊符号: “../”和“../../”。“../”表示当前目录的上级目录, “../../”表示当前目录的上上级目录。

因此简单理解就是, 绝对路径是从盘符开始的路径, 形如 C:\windows\system32\cmd.exe  
**相对路径:** 是从当前路径开始的路径, 假如当前路径为 C:\windows, 要描述上述路径, 只需输入 system32\cmd.exe。实际上, 严格的相对路径写法应为.\system32\cmd.exe。其中, “.”表示当前路径, 在通道情况下可以省略。假如当前路径为 c:\program files, 要调用上述命令, 则需要输入..\windows\system32\cmd.exe。其中, “..”为父目录。当前路径如果是 c:\program files\common files 则需要输入..\..\windows\system32\cmd.exe。实验证明: 绝对路径不利于搜索引擎表现, 相对路径在搜索引擎中表现良好。

超链接经常使用的语法如下:

```
<a href="链接地址" target="目标窗口位置">链接文本或图像</a>
```

属性 **target** 指定链接在哪个窗口打开, 常用的取值有 `_self` (自身窗口)、`_blank` (新建窗口) 等, 见表 1.1, 自己可以尝试不同取值的效果, 这个属性在框架知识中有详细解释和使用。

表 1.1 target 属性的有效值

target 属性的值	属性含义描述
<code>_blank</code>	在新窗口中打开被链接文档
<code>_self</code>	默认。在原浏览器窗口或者相同的框架中打开被链接文档
<code>_parent</code>	在父框架集中打开被链接文档
<code>_top</code>	在顶级窗口中打开被链接文档
<code>framename</code>	在指定的框架中打开被链接文档

超链接的 3 种应用场合: 页面间链接、锚链接和功能性链接。

1. 页面间链接: A 页到 B 页

#### 示例 1.12

```
<html >
<head>
<title>链接到其他页面</title>
```

```

</head>
<body>
<a href="register/register.htm">[免费注册]</a>
<a href="login/login.htm">[登录]</a>
</body>
</html>

```

示例 1.12 的效果如图 1.27 所示。



图 1.27 页面间超链接

## 2. 锚链接

A 页的甲位置到 A 页的乙位置，或者 A 页的甲位置到 B 页的乙位置。如果一个页面内容过多，导致页面过长，用户需要滚动滚动条来阅读相应内容时，可以使用锚点连接。实现 A 页的甲位置到 A 页（本页）的乙位置，首先在页面的乙位置标记（锚点），`<a name="锚点名">目标位置乙</a>`，它的功能类似古代固定船的锚，所以叫锚点，就像使用 `name` 属性在页面中创建一个书签，书签不会以任何特殊方式显示，它对读者是不可见的，当使用命名锚时，我们可以创建直接跳至该命名锚（比如页面中某个小节）的链接，这样使用者就无需不停地滚动页面来寻找它们需要的信息了，使用 `<a href="#锚名">提示文字</a>`；如果实现 A 页的甲位置到 B 页的乙位置，还是要先在 B 页利用 `<a name="锚点名">目标位置乙</a>` 标记锚点，然后使用 `<a href="B 页的 URL#锚名">提示文字</a>`，点击“提示文字”到 B 页相应位置。

## 3. 功能性链接

链接到电子邮箱、QQ、MSN 等，接下来以最常用的电子邮件链接为例，当单击“联系我们”时，打开用户的电子邮件程序，并自动填写“收件人”文本框中的电子邮件地址，如图 1.28 所示。完整的 HTML 代码如下示例 1.13 所示。

### 示例 1.13

```

<html >
<head>
<title>功能性链接</title>
</head>
<body>
<a href="mailto:example@example.com">[免费注册]</a>

```



```

<a href="login/login.htm">[登录]</a>
<a href="mailto:lygzcm@126.com">联系我们</a>
</body>
</html>

```

在所有浏览器中，链接的默认外观是：

- 未被访问的链接带有下划线而且是蓝色的。
- 已被访问的链接带有下划线而且是紫色的。
- 活动链接带有下划线而且是红色的。具体介绍在单元 4。



图 1.28 功能性连接

## 1.2.6 注释和特殊符号

### 1. HTML 注释

我们经常要在一些代码旁做一些 HTML 注释，这样做的好处很多，比如方便查找，方便比对，方便项目组里的其他程序员了解你的代码，而且可以方便以后你对自己代码的理解与修改等，当服务器遇到注释时会自动忽略注释内容。HTML 注释的开始使用<!--，结束使用-->，如<!--到搜狗搜索引擎了解它的搜索功能-->

```
<a href="http://www.sogou.com" />
```

搜狗首页

```
<!--链接结束-->
```

### 2. 特殊符号

在 HTML 中不能使用小于号 (<) 和大于号 (>)，这是因为浏览器会误认为它们是标签，如需显示小于号，我们必须这样写：&lt;，如果您在文本中写 10 个空格，在显示该页面之前，浏览器会删除它们中的 9 个，如需在页面中增加空格的数量，您需要使用 &nbsp;。所以，如果要页面中显示这些特殊符号，就必须使用相应的 HTML 代码表示，这些特殊符号对应 HTML 代码被称为字符实体。一些常用的特殊符号及对应的字符实体如表 1-2 所示，这些实体字符都以“&”开头，以“;”结束。

表 1-2 常用的特殊符号及对应的字符实体

特殊符号	字符实体	特殊符号	字符实体
空格	&nbsp;	引号	&quot;
大于号	&gt;	版权符号	&copy;
小于号	&lt;		

### 1.2.7 W3C 标准

万维网（World Wide Web）是作为欧洲核子研究组织的一个项目发展起来的，在那里 Tim Berners-Lee 开发出万维网的雏形。Tim Berners-Lee 是万维网的发明人和万维网联盟的主任。W3C 在 1994 年被创建的目的是为了完成麻省理工学院（MIT）与欧洲粒子物理研究所（CERN）之间的协同工作，并得到了美国国防部高级研究计划局（DARPA）和欧洲委员会（European Commission）的支持。

万维网联盟（World Wide Web Consortium, W3C），又称 W3C 理事会。1994 年 10 月在麻省理工学院计算机科学实验室成立。万维网联盟是国际最著名的标准化组织，1994 年成立后，至今已发布近百项相关万维网的标准，对万维网发展做出了杰出的贡献。万维网联盟拥有来自全世界 40 个国家的 400 多个会员组织，已在全世界 16 个地区设立了办事处。2006 年 4 月 28 日，万维网联盟在中国内地设立首个办事处。

正如前面所述，发明 HTML 的初衷是为了信息资料的网络传播和共享，希望 HTML 文档具有平台无关性，即同一 HTML 文档在不同平台上（包括使用不同的浏览器）将看到同样的页面内容和效果。但遗憾的是，随着浏览器市场的激烈竞争，各大浏览器厂商为了吸引用户，都在早期 HTML 版本的基础上进行各类标签的扩展。但因浏览器之间互不兼容，导致 HTML 编码规则混乱，违背了 HTML 发明的初衷，因此需要一个组织来制定和维护统一的 Web 开发标准，W3C 正是这样一个组织。下面介绍为什么是 W3C 标准以及 Web 开发方面的基础规范。

#### 1.2.7.1 什么是 W3C 标准

W3C 标准不是某一个标准，而是一系列标准的集合。网页主要由三部分组成：结构（Structure）、表现（Presentation）和行为（Behavior）。结构化标准语言主要包括 XHTML 和 XML，表现标准语言主要包括 CSS，行为标准主要包括对象模型（如 W3C DOM）、ECMAScript 等。W3C 万维网联盟主要职责是负责 Web 标准的制定和维护。Web 开发方面常设计的 W3C 标准如下：

- （1）HTML 内容方面——XHTML
- （2）样式美化方面——CSS
- （3）行为标准方面——DOM
- （4）页面交互方面——ECMAScript

其中 DOM 和 ECMAScript 将在第二学期学习。本课程主要涉及 XHTML 和 CSS 两类标准。HTML 方面目前比较常用的版本是 XHTML，表示可扩展超文本标签语言（EXTENSIBLE HYPER TEXT LANGUAGE）。它是更严格更纯净的 HTML 版本，且是一个 W3C 标准，规定了 HTML 编写的具体规范，所有主流浏览器都支持。

#### 1.2.7.2 W3C 提倡的 Web 页结构

##### 1. 内容和样式分离

内容 XHTML 只负责页面的内容结构，CSS（Cascading Style Sheet，简称 CSS，通常又称为“样式表”）负责表现样式（例如字体颜色、大小、背景图、显示位置等），方便网站的修改和维护。HTML 结构是页面的骨架，一个页面就好像一幢房子，HTML 结构就是钢精钢筋混凝土的墙，一幢房子如果没有钢精钢筋混凝土的墙那就是一堆费砖头，不能住人，不能办公。CSS 是装饰材料，是原木地板，是大理石，是油漆，是用来装饰房子的，如需要重新装饰，

则只需要更换装饰材料即可。

## 2. HTML 内容结构要语义化

HTML 是一种对文本内容进行结构和意义（或者说“语义”）进行补充的方法。它会告诉我们说：“这行是一个标题，这几行组成了一个段落。这些文字是项目列表，这些文字是链接到互联网上另一个文件的超链接。”值得注意的是，不应该让 HTML 来告诉我们：“这些文字是蓝色的，这些文字又是红色的，这部分内容是最靠右的一栏，这行内容是斜体字。”这些和表现相关的信息是 CSS 的工作。在做前端开发的时候要记住：HTML 告诉我们一块内容是什么（或其意义），而不是它长什么样子。当我们提到“语义标记”的时候，我们所说的 HTML 应该是完全脱离表现信息的，其中的标签应该都是语义化地定义了文档的结构。写语义化的 HTML 结构其实很简单，首先掌握 HTML 中各个标签的语义，<div>是一个容器，<strong>是表示强调，<ul><li>是一个无序列表等。看到内容的时候想想用什么标签能更好地描述它，是什么就用什么标签。即要求能根据 HTML 代码就能够看出这部分内容是什么，代表什么含义，例如，是标题、段落还是项目列表等。这样的好处：一是前面提及的方面搜索引擎搜索，二是方便在各种平台上传递，除普通的计算机外，还包括手机、pda、mp4 等，这些轻量的级显示器显示终端可能不具有普通计算机上的能力，它将按照 HTML 结构的语义，使用自身设备的渲染能力显示页面内容，因此，HTML 结构语义化越来越成为一种主流趋势。不防看一个糟糕的 HTML 文档，如示例 1.14 所示。

### 示例 1.14

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>不规范的示例</title>
</head>
<body>
  <font size="7">一级主题</font><br/>
  一级主题阐述文字 <br /><br />
  <font size="5">二级主题</font><br />
  二级主题阐述文字
  <p>项目列表 1
  <p>项目列表 2
  <p>项目列表 3
</body>
</html>
```

这是使用了 HTML 早期标签<font>表示字体大小，标签大小写不统一，段落<p>标签没有配对，但在浏览器中还能正常显示（这是因为浏览器为了“讨好”而“纵容”用户的原因）。

这样编写有什么问题？仔细查看 HTML 结构和表述的页面内容，不难发现存在如下弊端：

（1）内容和表现没分离，后期很难维护和修改。编写的 HTML 代码既表示字体大小等样式，又包含内容。如网站升级改版时需要改变字体大小等样式，则需要逐行修改 HTML 代码，非常繁琐。

（2）HTML 代码不能表示页面的内容语义，不利于搜索引擎搜索。即从 HTML 代码不能看出页面内容的关系，很难判断哪些内容是主体，哪些内容是相关的阐述文字，很难看出各列表的内容之间的关系。而搜索引擎的爬虫在搜索页面时只识别含有语义化的标签（例如<h1>

标题、<p>段落等)。而不识别表示样式的标签(例如<font>字体、<b>加粗等),因此上述示例 1.14 规范化的写法是示例 1.15。

#### 示例 1.15

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>规范的示例</title>
</head>
<body>
  <h1>一级主题</h1>
  <p>一级主题阐述文字</p>
  <h2>二级主题</h2>
  <p>二级主题阐述文字</p>
  <ul>
    <li>项目列表 1</li>
    <li>项目列表 2</li>
    <li>项目列表 3</li>
  </ul>
</body>
</html>
```

#### 1.2.7.3 XHTML 1.0 的基本规范

了解了 W3C 提倡的 Web 结构后,下面介绍 XHTML 的基本规范。

(1) 页面顶部必须添加文档类型。即必须使用<!DOCTYPE>标签添加文档类型声明,声明 HTML 文档遵守 W3C XHTML 哪个级别的规范。需要注意:该声明必须位于 HTML 文档的第一行。XHTML 有三个版本:XHTML 1.0 Transitional、XHTML 1.0 Strict、XHTML 1.0 Frameset。XHTML 1.0 Transitional(过度类型):也称松散声明,相比 Strict 而言,要求相对宽松些,也是我们经常选择使用的,包含 HTML 4.01 的所有标记以及属性,是一种不是那么严格的 XHTML,目的是使现有的 HTML 开发者更容易接受并使用 XHTML。XHTML 1.0 Strict(严格类型),这种声明完全符合 W3C 的标准,但要求比较严格,就是严格版本的 XHTML 了,开发者必须要严格遵守文档的结构与表现分开的规则,也就是说需要用 CSS 控制页面的显示而不是使用<font>、bgcolor 之类的标记或属性。Frameset(框架类型),Strict 严格标准中不允许使用框架,当页面中需要使用框架时,则使用该声明,框架页将在后续单元讲解。

(2) 所有的 XHTML 元素必须被嵌套于<html>根元素中。

(3) 所有标签、属性必须小写。例如,规范的写法为<html>,不规范的写法为<HTML>等。

(4) 属性不允许缩写。

(5) XHTML 规定所有属性都必须有一个值,属性值必须用引号""括起来:即必须使用单引号或双引号将属性值括起来,如规范的写法:<meta name="keyword"content="IT 培训"/>,不规范的写法:<meta name=keyword content=IT 培训/>。没有值的就重复本身,例如:

```
<td nowrap> <input type="checkbox" name="shirt" value="medium" checked>
```

必须修改为:

```
<td nowrap="nowrap"> <input type="checkbox" name="shirt" value="medium" checked="checked">
```

(6) 用 id 属性来替代 name 属性。注意:为了版本比较低的浏览器,应该同时使用 name

和 id 属性，并使它们两个的值相同的。

(7) 所有标签必须被关闭，空标签也必须关闭：如换行和水平线标签等，使用<br/>和<hr/>表示关闭。

(8) 不要在注释内容中再出现双横线“--”。

(9) 图片必须有说明文字。

(10) 代码必须正确缩进。

(11) 把所有<、>和&等特殊符号用编码表示。

## 1.3 工作场景训练

有了前面的技术和知识准备，下面去完成场景中的任务。

### 1.3.1 实现工作场景 1 的任务

使用 Dreamweaver 编辑工具编写 HTML 代码，实现如图 1.29 所示的页面效果，注意内容间的层次结构。



图 1.29 基本块级标签练习

#### 场景任务 1 参考代码

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
  <title>李清照宋词赏析</title>
</head>
```

```
<body>
  <h1>李清照宋词赏析</h1>
  <hr />
  <h2>目录</h2>
  <p>第一首：如梦令</p>
  <p>第二首：一剪梅</p>
  <hr />
  <h3>如梦令</h3>
  <p>作者：李清照</p>
  <p>昨夜雨疏风骤， </p>
  <p>浓睡不消残酒， </p>
  <p>试问卷帘人， </p>
  <p>却道海棠依旧。 </p>
  <p>知否， </p>
  <p>知否， </p>
  <p>应是绿肥红瘦。 </p>
  <hr />
  <h4>【李清照简介】 </h4>
  <p>山东省济南章丘人，号易安居士。宋代女词人，婉约词派代表，有“千古第一才女”之称。.....</p>
</body>
</html>
```

### 1.3.2 实现工作场景 2 的任务

使用 Dreamweaver 编辑工具编写 HTML 代码，实现如图 1.30 所示的页面效果。使用<div>分区标签作为整个页面内容的容器，然后放置标题、无序列表、有序列表等。块级标签都支持嵌套。例如，无序列表可以再嵌套无序列表。

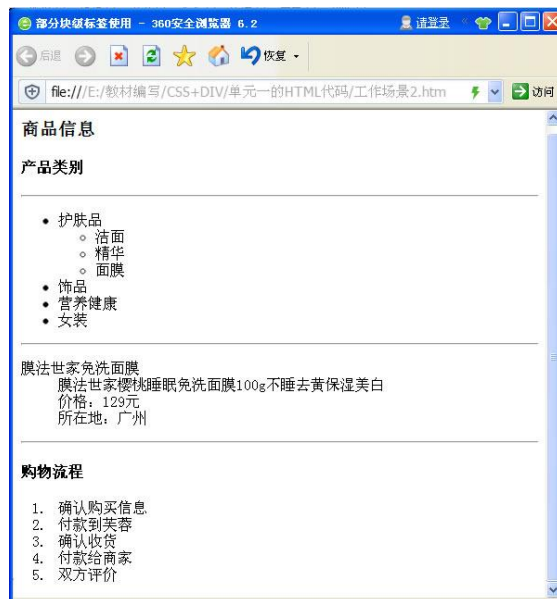


图 1.30 常用于页面布局的块级标签练习

## 场景任务 2 参考代码

```
<html>
<head>
<title>部分块级标签使用</title>
</head>
<body>
<div>
  <h3>商品信息</h3>
  <h4>产品类别</h4>
  <hr/>
  <ul>
    <li>护肤品
      <ul><li>洁面</li><li>精华</li><li>面膜</li></ul>
    </li>
    <li>饰品</li>
    <li>营养健康</li>
    <li>女装</li>
  </ul>
  <hr/>
  <dl>
    <dt>膜法世家免洗面膜</dt>
    <dd>膜法世家樱桃睡眠免洗面膜 100g 不睡去黄保湿美白</dd>
    <dd>价格：129 元</dd>
    <dd>所在地：广州</dd>
  </dl>
  <hr/>
  <h4>购物流程</h4>
  <ol>
    <li>确认购买信息</li>
    <li>付款到芙蓉</li>
    <li>确认收货</li>
    <li>付款给商家</li>
    <li>双方评价</li>
  </ol>
</div>
</body>
</html>
```

### 1.3.3 实现工作场景 3 的任务

对于场景的任务我们通过分析，应该整体使用自定义列表，将图片作为术语标题<dt>，文字内容作为术语描述<dd>，范围标签<span>用于标识行内的 10 这个数字，“促销信息”用标题标签修饰。



图 1.31 行级元素

### 场景任务 3 参考代码

```

</html>
<head>
<title>上机练习-行级元素</title>
</head>
<body>
<h1>促销信息</h1>
<dl>
  <dt></dt>
  <dd>相宜本草</dd>
  <dd>滋润型的洁面产品，适合中至干性，混合性肌肤，秋冬等干燥季节</dd>
  <dd>跳楼疯抢价<span style="color:red;font-size:60px;">10</span>元起 </dd>
</dl>
</body>
</html>

```

## 1.4 重点问题分析

我们刚开始接触标记语言，重点注意以下几个问题：

(1) XHTML 文件与普通的纯文本文件的最大不同在于一些用“<>”包含的东西，例如<body>，我们把它们叫做标签。通常情况下 XHTML 标签都是成对出现的，例如<html></html>。它们只相差一个“/”，我们把类似<html>的没有“/”的标签叫做起始标签，而它对应的有“/”的</html>则叫终止标签，终止标签与起始标签只相差一个“/”符号。当然，XHTML 也有一些标签并不成对出现，它们没有终止标签，我们把这样的标签叫做“空标签”，如分割线标签写成写成<hr/>，而不是对称的<hr></hr>，这样的书写格式是为了满足 XHTML 中任何标签都需要“关闭”的规则。

(2) 关于大小写，以前各个版本 HTML 标签并不区分大小写，例如标签<HTML>和标签<html>是等效的。而在 XHTML 中，所有标签均使用小写。

(3) 理解 XHTML 标签的作用和标签的属性。



## 1.5 小结

HTML 标签分为块级和行级标签，块级标签按“块”显示，行级标签按“行”逐一显示。基本块级标签包括段落标签<p>、标题标签<h1>~<h6>、水平线标签<br/>等。常用于布局的块级标签包括无序列表标签<ul>、有序列表标签<ol>、定义列表标签<dl>、分区标签<div>等。行级标签包括图片标签<img/>、范围标签<span>、换行标签<br/>、超链接标签<a>等。插入图片时，要求“src”和“alt”属性必选，“title”和“alt”属性推荐同时使用。编写 HTML 文档注意遵循 W3C 相关标准，W3C 提倡内容和结构分离，HTML 结构具有语义化。