第1章 1

HTML5使用基础

高楼万丈平地起。学习任何一门新知识，首先要打好基础。本章将介绍使用 HTML5 所需要的一些基础知识，主要包括 HTML5 简介、使用 HTML 元素及其属性、创建 HTML 文档结构、使用元数据元素以及脚本元素等。

1.1 HTML5 简介

HTML5 是新一代互联网标准的开发语言。在使用 HTML5 设计和制作网页之前，首先需要了解什么是 HTML 以及 HTML5 有哪些新特性。

1.1.1 什么是 HTML

HTML（HyperText Markup Language）即超文本标记语言。使用该语言创建的 HTML 文档是 WWW（万维网）上的一个超媒体文档，通常称为一个页面，也称为 Web 页或网页。“超文本”是指页面内不仅可以包含文本，还可以包含图片、链接，甚至音频、视频、脚本程序等非文本元素。

HTML 是编写网页的主要语言，它通过各种标签（亦称标记）来告诉浏览器如何呈现网页的内容。使用 HTML 语言可以编写包含文本、表格、图像、链接、音频、视频以及各种其他媒体元素的网页，这些网页和相关资源存储在 Web 服务器上，在建立网络连接的情况下可通过 Web 浏览器来请求和访问。

HTML 语言在其发展历程中经历了一系列的版本。

- HTML 1.0：在 1993 年 6 月，作为互联网工程工作小组（IETF）工作草案（并非标准）发布。
- HTML 2.0：1995 年 11 月作为 RFC 1866 发布，在 RFC 2854 于 2000 年 6 月发布之后被宣布已经过时。
- HTML 3.2：1997 年 1 月 14 日发布，W3C（万维网联盟）推荐标准。
- HTML 4.0：1997 年 12 月 18 日发布，W3C 推荐标准。
- HTML 4.01（微小改进）：1999 年 12 月 24 日发布，W3C 推荐标准。
- HTML5：2014 年 10 月 29 日发布，W3C 推荐标准。

1.1.2 HTML5 的新功能

HTML5 是 HTML 的第五次重大修改。W3C 于 2014 年 10 月 29 日宣布，经过接近 8 年的艰苦努力，HTML5 标准规范终于制定完成。HTML5 实际上是一系列创新的代表。HTML5 提

供了一些新标签和新方法，并通过与 CSS3 和 JavaScript 的相互作用形成了一个通用的开发框架，这是以客户端为中心的应用程序处理的核心。除了将 HTML5 技术的技巧和方法部署到桌面外，还可以在特性丰富的 Web 移动手机浏览器中实现 HTML5，随着 Apple iPhone、Google Android 和运行 Palm webOS 的手机的流行和普及，这注定是一个不断增长的市场。

HTML5 是一种专门用于组织 Web 内容的语言，它通过创建一种标准化的、直观的 UI 标记语言来简化 Web 设计和开发。HTML5 提供了解析和划分页面的方法，它允许创建各种独立的组件来按照逻辑组织站点，同时还为站点提供联合功能。HTML5 可以称作“面向站点设计的信息映射方法”，因为它融入了信息映射、信息划分和消息标签等基本内容，使信息变得易于使用和理解，这构成了 HTML5 的生动语义和审美工具的基础。HTML5 使具备不同能力的设计师和开发人员能够发布从简单文本到丰富的交互式多媒体等各种内容。

HTML5 提供了有效的数据管理、绘图、视频和音频工具，它简化了面向 Web 和便携式设备的跨浏览器应用程序的开发。HTML5 是推动移动云计算服务的技术之一，因为它可以实现更高程度的灵活性，可以开发出激动人心的交互式网站。它还引入了新的标记和增强，包括一个优雅的结构、表单控件、API、多媒体、数据库支持，并极大地加快了处理速度。

HTML5 实现了以下新功能：提供了可以准确描述所包含的内容的标记；增强的网络通信；显著改善了一般存储；用于运行后台流程的 Web Worker；在应用程序和服务器之间建立持久连接的 WebSocket 接口；更好地检索存储的数据；改善了网页保存和载入速度；支持 CSS3 管理 GUI，意味着 HTML5 具备面向内容的特性；改善了浏览器的表单处理；一个基于 SQL 的数据库 API，允许客户端本地存储；画布和视频，无须安装第三方插件即可添加图形和视频；Geolocation API 规范，使用智能手机位置功能来合并移动云服务和应用程序；智能表单减少了下载 JavaScript 代码的需求，在移动设备和云服务器之间实现了更有效的通信。

HTML5 创建了更加吸引人的用户体验：使用 HTML5 设计的页面可以提供与桌面应用程序类似的体验。HTML5 还将 API 功能和浏览器结合在一起，提供了增强的多平台开发。通过使用 HTML5，开发人员可以提供在不同平台之间切换的现代应用程序体验。

1.2 定义 HTML 元素

一个网页由各种各样的 HTML 元素组成。每个元素都是用特定的 HTML 标签定义的。要使用 HTML 标签定义所需要的元素，就应该了解 HTML 的语法结构和语法规则。

1.2.1 HTML 语法结构

HTML 元素可用标签来定义。大多数标签的语法格式如下：

```
<标签 属性="值" 属性="值"...>内容</标签>
```

其中，标签是用一对尖括号即“<”（小于号）和“>”（大于号）括起来的单词或单词缩写。HTML 标签通常由开始标签“<标签>”和结束标签“</标签>”组成。开始标签和结束标签连同两者之间的内容构成了 HTML 元素。开始标签和结束标签之间的内容是元素的内容。

例如，要在网页中显示一个三级标题，可用 h3 元素来实现，代码如下：

```
<h3>基于 HTML5 的网页设计</h3>
```

大多数 HTML 元素可拥有属性。属性在开始标签中指定，用于设置 HTML 元素的相关选项，通常以“属性名="值"”的形式表示。多个属性用空格分隔，并且不分先后顺序。

例如，用 a 元素可在网页中创建一个超链接，通过设置 href 属性可指定该链接指向的目标网址，代码如下：

```
<a href="http://www.baidu.com">百度一下</a>
```

元素的开始标签与结束标签之间不一定要有内容。没有内容的元素称为空元素。对于空元素，可以只用一个标签表示，即把开始标签与结束标签合二为一，将斜线符号（/）放在开始标签的末尾。

举个例子。code 元素用于表示计算机代码文本，例如：

```
<code>document.writeln("Hello World");</code>
```

如果在 code 元素的开始标签与结束标签之间没有内容，则可以写成以下形式：

```
<code />
```

在这里，<code />称为自闭合标签，它与<code></code>是等价的。

有些元素只能使用一个标签表示，在其中放置任何内容都不符合 HTML 规范。这种没有内容的元素也称为虚元素。例如，hr 就是虚元素，<hr>标签表示主题内容的变化，在网页中显示一条水平分隔线。虚元素也可以用空元素结构表示，例如，<hr>也可以写成<hr />。

要在 HTML 文档中插入注释内容，可使用注释标签。语法格式如下：

```
<!-- 在此输入注释文字 -->
```

注释文字内容会被浏览器忽略，在浏览器查看网页时是看不到这些内容的。使用注释可以对源代码进行解释，这样做有助于在后期对代码进行编辑和维护。

1.2.2 HTML 语法规则

使用 HTML 语言编写网页时必须符合以下语法规则。

（1）尖括号、元素名、属性名等都必须使用半角的西文字符，而不能使用全角字符。

（2）元素名、属性名和属性值不区分大小写，习惯上使用小写字母表示。

（3）多数标签可以嵌套使用，但不允许交叉。

（4）一行可以写多个标签，一个标签也可以分成多行写，但标签中的一个单词不能分成两行写。写成多行时浏览器一般忽略回车符；空格通常也不按源代码中的效果显示。

（5）建议使用双引号将属性值括起来，但也可以使用单引号，或者不用引号。

1.3 设置元素属性

使用 HTML 标签定义一个元素时，可用属性对该元素的相关选项进行设置。属性只能用在开始标签或单个标签上，而不能用于结束标签。大多数属性都具有名称和值两个部分。元素的属性分为全局属性和局部属性（专有属性）。全局属性可用于所有 HTML 元素，局部属性则为个别元素提供其特有的配置信息。例如，使用 a 元素时可通过其专有属性 href 来设置超链接的目标 URL。

1.3.1 对元素应用多个属性

对一个 HTML 元素可以同时应用多个属性，这些属性之间用一个或多个空格分隔。例如，使用 a 元素创建超链接时可对该元素设置多个属性，代码如下：

```
欢迎您访问<a class="link" href="http://www.phei.com.cn" title="请单击此链接">电子工业出版社</a>!
```

其中，class 和 title 属性都是全局属性，本章稍后会讲到这些属性。设置多个属性时对这些属性的顺序没有什么要求，全局属性和专有属性可以交错出现。

1.3.2 使用布尔属性

在元素的属性中有些属性属于布尔属性，对于这种属性不需要设定一个值，只需要将属性名添加到元素的开始标签中即可。例如，使用 button 元素创建提交按钮时，通过添加布尔属性 disable 可以禁用该按钮，代码如下：

```
<button type="submit" disabled>提交</button>
```

这个例子中的 type 属性指定按钮的类型，submit 表示要创建的是一个提交按钮。button 元素为用户提交表单数据提供了一种手段。disabled 属性是一个布尔属性，在 button 元素中仅仅添加了属性名 disabled，并没有设置 disabled="true"。设置 disabled 属性将禁用按钮，从而阻止用户提供表单数据。

对于布尔属性，也可以指定一个空字符串（""）或属性名称字符串作为其值，这样也会收到同样的效果，代码如下：

```
<button type="submit" disabled="">提交</button>
<button type="submit" disabled="disabled">提交</button>
```

1.3.3 使用自定义属性

定义 HTML 元素时用户还可以创建自定义属性，这种属性也称为作者定义属性或扩展属性。这类属性的名称必须以 data-作为前缀。

例如，下面的例子使用 input 元素创建文本框并添加了两个自定义属性：

```
用户名：<input type="text" data-creator="admin" data-purpose="collection">
```

之所以在这类属性名称之前添加前缀 data-，是为了避免与 HTML 的未来版本中可能增加的属性名称发生冲突。自定义属性与 CSS 和 JavaScript 结合起来很有用。

1.3.4 全局属性概述

全局属性用来配置所有 HTML 元素共有的行为。全局属性可以用在任何元素上，不过不一定会带来有意义或有用的行为改变。

表 1.1 列出了 HTML 中的全局属性。

在表 1.1 中所列出的全局属性中有 3 个属性比较重要，它们分别是 id、class 和 style 属性。通过这些全局属性可以将 HTML 元素与 CSS 样式以及 JavaScript 程序联系起来。下面对这些全局属性加以说明。

表 1.1　全局属性

属　　性	描　　述
accesskey	规定激活元素的快捷键
class	规定元素的一个或多个类名（引用 CSS 样式表中的类）
contenteditable	规定元素内容是否可编辑
contextmenu	规定元素的上下文菜单（上下文菜单在用户单击元素时显示）
data-*	用于存储页面或应用程序的私有定制数据
dir	规定元素中内容的文本方向
draggable	规定元素是否可拖动
dropzone	规定在拖动元素时是否进行复制、移动或链接
hidden	规定将元素隐藏起来
id	规定元素的唯一标识符
lang	规定元素内容的语言
spellcheck	规定是否对元素进行拼写和语法检查
style	规定元素的行内 CSS 样式
tabindex	规定元素的 tab 键次序
title	规定有关元素的额外信息
translate	规定是否应该翻译元素内容

1．id **属性**

id 属性用来给元素分配一个唯一的标识符。使用该标识符可以将 CSS 样式应用到元素上，或者在 JavaScript 程序中用来选择元素。

2．class **属性**

class 属性用来将元素归类。使用该属性可以对文档中某一类元素应用 CSS 样式，或者在 JavaScript 程序中选择某一类元素。

3．style **属性**

style 属性用来直接在元素上定义 CSS 样式，由此定义的样式称为元素内嵌样式。

1.4　创建 HTML 文档

网页是一种 HTML 文档，实际上就是一种纯文本格式的文件，其文件扩展名是.html。网页可以使用各种文本编辑软件来创建和修改，可以直接从本地磁盘加载到浏览器中来查看其效果。为了创建、编辑和测试 HTML5 网页，建议使用 Adobe Dreamweaver CC 软件，因为该软件支持 HTML5 新增加的标签和属性，不仅提供了代码提示功能，还可以直接在实时视图中查看网页的布局效果。在学习本课程的过程中，除了网页编辑软件之外，建议在所使用的计算机上同时安装当今的几种主流浏览器，包括 Internet Explorer、Google Chrome、Mozilla Firefox、Opera 及 Apple Safari 等。

1.4.1 HTML 文档结构

HTML 文档具有特定的结构，通常要包含一些关键性的元素。其基本结构如下：

```
<!doctype html>
<html>
<head>
<meta charset="gb2312">
<title>网页标题</title>
</head>

<body>
网页内容
</body>
</html>
```

下面对 HTML 文档基本结构包含的各个部分加以说明。

1．doctype 指令

文档类型声明<!doctype html>位于文档的第一行，是文档中的第一个成分，但它并不是 HTML 标签而是一条指令，它告诉浏览器编写网页所使用的 HTML 规范是什么版本。这条指令告诉浏览器编写网页所使用的HTML 规范是HTML5。如果希望在HTML 网页中使用HTML5 文档类型，就必须在网页的第一行添加这条指令，只有这样浏览器才能了解所预期的文档类型。

2．html 元素

html 元素是网页的根元素，它告诉浏览器这是一个 HTML 文档。html 元素的开始标签位于<!doctype>标签之后，它告诉浏览器：由此开始直到 html 元素的结束标签，所有元素都应当作为 HTML 文档来处理。

html 元素有一个局部属性，即 manifest。该属性是 HTML5 中新增的，其值是一个 URL，用于为脱机使用定义缓存信息。html 元素包含的内容由两个元素组成，即 head 元素和 body 元素。

3．head 元素

head 元素用于向浏览器提供有关 HTML 文档的信息，通常称为头部信息。浏览器不会向用户显示这些头部信息。每个 HTML 文档都应该有一个 head 元素。

head 元素通过开始标签<head>和结束标签</head>来创建，在这两个标签之间可以包含 base、link、meta、script、style 及 title 元素。head 元素中通常包含两个元素，即 meta 元素和 title 元素，前者通过 charset 属性指定文档的默认编码（gb2312 表示简体中文），后者则用于指定网页的标题。应记住始终为文档指定标题。当在浏览器中加载 HTML 文档时，文档标题就显示在当前文档所在选项卡的标题栏中。一个文档中不能有一个以上的 title 元素。

head 元素没有任何局部属性。

4．body 元素

body 元素用于定义 HTML 文档的主体，该元素包含文档的所有内容，例如文本、超链接、图形、图像、列表、表格及视频等。在 HTML5 中删除了 body 元素的所有局部属性，这些属性即使在 HTML 4.01 中也是不赞成使用的。

例 1.1　本例中创建了一个简单的网页，用于演示 HTML 文档的基本结构。源代码如下：

```
<!doctype html>
<html>
<head>
<meta charset="gb2312">
```

```
<title>一个简单的网页</title>
</head>

<body>
<h3>开课啦</h3>
<hr>
<p>大家好！从今天开始我们一起来学习 HTML5 网页设计，共同进入 Web 开发新时代。</p>
</body>
</html>
```

在这个例子中，html 元素是整个文档的根元素。html 元素包含两个元素，即 head 元素和 body 元素，两者分别用于定义文档头部和文档主体。文档头部包含两个元素，即 meta 元素和 title 元素，前者指定网页的默认编码为简体中文，后者则设置了网页标题。文档主体包含 3 个元素：h3 元素用于定义一个三级标题；hr 元素标签用于定义内容中的主题变化并显示为一条水平线；p 元素在文档中定义一个段落。标题与段落之间通过一条水平线分隔开。在 Google Chrome 浏览器中打开这个网页，其显示效果如图 1.1 所示。

图 1.1　一个简单的网页

1.4.2 元素之间的关系

HTML 文档包含各种元素，这些元素之间具有明确的关系。有一些元素可以包含另一些元素，包含其他元素的元素是被包含元素的父元素，被包含元素则是父元素的子元素。一个元素可以拥有多个子元素，但只能拥有一个父元素。

在例 1.1 中，html 元素包含 head 元素和 body 元素，html 元素是 head 元素和 body 元素的父元素，head 元素和 body 元素都是 html 元素的子元素。head 元素包含 meta 元素和 title 元素，head 元素是 meta 元素和 title 元素的父元素，meta 元素和 title 元素都是 head 元素的子元素。body 元素包含 h3 元素、hr 元素和 p 元素，body 元素是 h3 元素、hr 元素和 p 元素的父元素，h3 元素、hr 元素和 p 元素都是 body 元素的子元素。

html 元素包含 body 元素，而后者又包含着 h3 元素、hr 元素和 p 元素。body 元素、h3 元素、hr 元素和 p 元素都是 html 元素的后代元素，但是这些后代元素中只有 body 元素才是 html 元素的子元素。子元素是关系最近的后代元素。

具有同一个父元素的几个元素互为兄弟元素。在例 1.1 中，head 元素和 body 元素就是兄弟元素，因为它们同是 html 元素的子元素。h3 元素、hr 元素和 p 元素也是兄弟元素，因为它们同是 body 元素的子元素。

1.4.3 元素分类

在 HTML5 中，所有元素主要分为 7 类：元数据元素、流式元素、章节元素、标题元素、短语元素、嵌入元素及交互元素。这些分类集合互相之间也存在一定的交集，即一个元素可以

同时属于多个分类，其交集关系如图 1.2 所示。

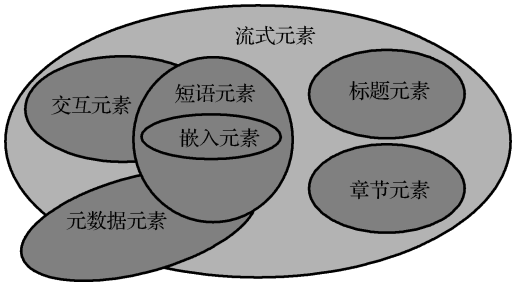

图 1.2　元素交集关系

1．元数据元素

元数据元素是指那些定义文档元数据信息的元素，其作用是影响文档中其他节点的展现与行为、定义文档与其他外部资源之间的关系等。以下元素属于元数据元素：base、link、meta、noscript、script、style、template、title。

2．流式元素

所有可以放在 body 元素内构成文档内容的元素均属于流式元素。因此，除了 base、link、meta、style、title 等只能放在 head 元素内的元素外，其余所有元素均属于流式元素。

3．章节元素

章节元素是指用于定义页面结构的元素，包括 article、aside、nav 及 section。

4．标题元素

标题元素包括 h1、h2、h3、h4、h5 和 h6。

5．短语元素

所有可以放在 p 元素内构成段落内容的元素均属于短语元素。因此，所有短语元素均属于流式元素，但并非所有流式元素都是短语元素。一个短语元素内部一般只能包含其他短语元素。

6．嵌入元素

所有用于在网页中嵌入外部资源的元素均属于嵌入元素，包括 audio、video、img、canvas、svg、iframe、embed、object 及 math。

7．交互元素

所有与用户交互有关的元素均属于交互元素，包括 a、input、textarea 及 select 等。

根据上述元素分类，HTML5 标准定义了任何元素的内容模型，即对于一个元素而言使用哪种子元素才合法。例如，p 元素的内容模型为短语元素，该元素只接受短语元素为子元素，而不接受非短语元素（如 div）。

1.5　使用元数据元素

如前所述，HTML 文档分为头部和主体两部分。文档头部用 head 元素来定义，通过在 head 元素中放置元数据元素可以提供关于 HTML 文档的信息。虽然元数据元素本身并不是 HTML 文档的内容，但提供了关于该文档的信息。

1.5.1 设置文档标题

使用 title 元素可以设置 HTML 文档的标题。这个元素的内容通常显示在浏览器窗口顶部的标题栏或标签页的标签上。

title 元素为元数据元素，它允许具有的父元素是 head。title 元素没有任何局部属性。该元素的内容应是文档标题或对文档内容的简要说明。title 标签用法是在开始标签和结束标签之间包含文字内容。

每个 HTML 文档都应该有且只有一个 title 元素，在其开始标签和结束标签之间包含的文字需要有一定的实际意义，以便让用户能根据标题文字来区分各个浏览器窗口或浏览器的各个标签页，并且知道哪个显示的才是所浏览的网页或所使用的 Web 应用系统。

由于标题文字显示在浏览器标题栏或标签页的标签上，因此不能对标题文字设置格式（例如字体、字号和颜色等），但可以通过 JavaScript 脚本改变标题文字的内容，例如通过动态修改标题文字在标题栏显示一个数字时钟。

1.5.2 用元数据说明文档

在 HTML 文档中，可以用 meta 元素来定义文档的各种元数据。meta 元素有多种不同用法，而且一个 HTML 文档可以包含多个 meta 元素。

meta 元素属于元数据元素，它允许具有的父元素是 head。meta 元素有以下局部属性：name、content、charset 和 http-equiv，其中 charset 属性是 HTML5 中新增的。

meta 元素为虚元素形式，因此它不能包含任何内容。

meta 元素有多种用途，但每个 meta 元素只能用于一种用途。根据需要，可以在 head 元素中添加多个 meta 元素。

1．声明字符编码

meta 元素的另一种用途是声明 HTML 文档内容所用的字符编码。

在例 1.1 中通过 meta 元素将网页的字符编码指定为简体中文，源代码如下：

```
<meta charset="gb2312">
```

如果要在同一网页上显示简体中文、繁体中文以及其他语言（如英文、俄文、日文及韩文等），则应将字符编码设置为 utf-8，源代码如下：

```
<meta charset="utf-8">
```

注意：当使用 Windows 自带的记事本程序编写网页时，如果将网页的字符编码设置为 utf-8，在保存文件时就必须将编码方式设置为 utf-8，而不是使用默认的 ANSI 编码方式，否则在浏览器中打开网页时就可能会出现乱码现象。

2．指定名/值元数据对

meta 元素的一个用途是以“名/值”形式定义元数据，为此需要用到 name 和 content 属性。name 属性提供了名称/值对中的名称，表示元数据的类型，用于将 content 属性关联到一个名称；content 属性提供了名称/值对中的值，该值可以是任何有效的字符串。content 属性始终要和 name 属性或 http-equiv 属性一起使用。

meta 元素可以使用的几种预定义元数据类型（即 name 属性的值）如下。

（1）author：表示当前页面的作者。

（2）description：给出当前页面的说明。

（3）keywords：一组以逗号分开的关键字，用来描述页面的内容。

（4）generator：指定用来生成 HTML 的编辑器软件的名称。

（5）robots：用来告诉搜索引擎哪些页面需要索引，哪些页面不需要索引；相应的 content 属性值有 all、none、index、follow、noindex、noarchive、nofollow。

下面给出指定名/值元数据对的一些例子。

指定页面的作者：

```
<meta name="author" content="张三, zhangsan@msn.com">
```

定义针对搜索引擎的关键词：

```
<meta name="keywords" content="HTML5, CSS 3, JavaScript">
```

定义对页面的描述：

```
<meta name="description" content="网页设计与制作教程。" />
```

3．模拟 HTTP 标头字段

服务器与浏览器之间传输 HTML 数据时一般使用 HTTP 协议。服务器的每个响应都包含一组向浏览器说明其内容的 HTTP 标头字段。使用 meta 元素的属性可以改写部分 HTTP 标头字段的值，指示浏览器按照需要显示网页内容。此时应使用 meta 元素的 http-equiv 属性指定标头字段的名称，同时使用 content 属性设置标头字段的值。

下面介绍使用 meta 元素改写 HTTP 标头字段值的几个例子。

（1）expires：设定网页的到期时间，此时必须使用 GMT 时间格式。一旦网页过期，必须到服务器上重新传输。例如：

```
＜meta http-equiv="expires" content="Sat, 17 Sep 2016 22:33:00 GMT"＞
```

（2）pragma：设置网页的 cache 模式，可禁止浏览器从本地缓存中调阅页面内容，设定后一旦离开网页就无法从 cache 中再调出，此时访问者将无法脱机浏览。代码如下：

```
＜meta http-equiv="pragma" content="no-cache"＞
```

（3）refresh：设置自动刷新或转向新的页面。例如：

```
＜meta http-equiv="refresh" content="6; URL=http://www.phei.com.cn"＞
```

其中的数字 6 指定当前页面停留 6 秒钟，然后自动转到新的 URL 网址。如果未指定 URL 参数，则当前页面每隔 6 秒钟就会自动刷新一次。

例 1.2　本例用于演示如何使用 meta 元素定义元数据。源代码如下：

```
<!doctype html>
<html>
<head>
<meta charset="gb2312">
<meta name="author" content="ZHAO Zengmin">
<meta name="keywords" content="HTML5, CSS 3, JavaScript">
<meta http-equiv="refresh" content="6; url=http://www.phei.com.cn">
<title>使用 meta 元素定义元数据</title>
</head>

<body>
```

```
<h3>meta 元素应用示例</h3>
<hr>
<p>欢迎您访问本网站！6 秒钟后将进入电子工业出版社官方网站。</p>
</body>
</html>
```

在本例中，head 元素包含 4 个 meta 元素，它们的作用分别是：将网页的字符编码设置为简体中文；设置网页的作者；设置网页的关键字；改写 HTTP 标头字段 refresh 的值，使得当前页面显示 6 秒钟之后自动转到电子工业出版社官方网站，如图 1.3 和图 1.4 所示。

图 1.3　在浏览器中打开页面

图 1.4　6 秒钟后自动跳转到指定页面

1.5.3 定义内嵌 CSS 样式表

为了规定在浏览器中如何呈现 HTML 文档，使用 style 元素可以为文档定义内嵌的 CSS 样式表信息。style 元素属于元数据元素，可将其放置在任何可包含元数据元素的元素中，包括 head、div、noscript、section、article、aside、type、media 及 scoped，不过通常是将其放在网页的 head 部分。

style 元素使用开始标签和结束标签来定义，在其开始标签中可用 type 属性规定样式表的 MIME 类型，该属性唯一可能的值是“text/css”。开始标签与结束标签之间的内容是 CSS 样式表，这个样式表由一些 CSS 样式规则组成。每个 CSS 样式规则均以 CSS 选择器开头，后面跟的是 CSS 声明，CSS 声明包含在一对花括号之间。

例如，下面的源代码使用 style 元素定义了一个 CSS 样式表，其中仅包含一个 CSS 样式规则，这个规则中的 CSS 选择器是 p 标签，即指示浏览器将样式规则应用到文档中的所有 p 元素上；在 CSS 声明中将段落中的文本字体设置为微软雅黑。

```
<style type="text/css">
p {
   font-family: "微软雅黑"
}
</style>
```

CSS 声明对一些 CSS 属性的值进行设置，属性名与属性值之间用冒号分开。如果要设置多个属性的值，不同的属性设置之间可用分号隔开。

除了 type 属性之外，style 元素还有以下两个局部属性。

media：为 CSS 样式表规定不同类型的目标媒介，其取值可以是 screen、tty、tv、projection、handheld、print、braille、aural、all。

scoped：这是 HTML5 中新增的属性，是一个布尔属性。如果使用该属性，则所定义的样式仅仅应用到 style 元素的父元素及其子元素。

例 1.3　本例用于演示如何使用 style 元素在网页中定义内嵌的 CSS 样式表。源代码如下：

```
<!doctype html>
<html>
<head>
<meta charset="gb2312">
<title>定义内嵌 CSS 样式表</title>
<style type="text/css">
h3, p {
   font-family:  微软雅黑
}
p {
   background-color: gray;
   color: white
}
</style>
</head>

<body>
<h3>什么是 CSS</h3>
<hr>
<p>CSS 是英文 Cascading Style Sheets 的缩写，中文含义是层叠样式表。CSS 是用于控制或增加网页外观样式的一种标记语言。</p>
</body>
</html>
```

在上述源代码中，使用 style 标签在文档的 head 部分定义了一个 CSS 样式表。这个样式表包含以下两个 CSS 样式规则：第一个规则的 CSS 选择器是 h3 和 p 标签（两者之间用逗号分隔开），用于选择文档中的所有 h3 和 p 元素，并通过花括号的 CSS 声明将三级标题和段落中的文本字体设置为微软雅黑；第二个规则的 CSS 选择器是 p 标签，用于选择文档中的所有 p 元素，并通过 CSS 声明将段落的背景颜色设置为灰色，将段落中的文本设置为白色。在浏览器中打开这个网页，显示效果如图 1.5 所示。

图 1.5　定义内嵌的 CSS 样式表

1.5.4 链接外部 CSS 样式表

使用 style 标签在网页中定义的内嵌 CSS 样式表只能用在当前网页中。如果希望将同一个 CSS 样式表应用于多个网页中，则需要将所定义的 CSS 样式规则保存到独立于网页的文件中，这样的文件称为外部样式表文件，其文件扩展名为.css。

在外部样式表文件中定义 CSS 规则时，不再需要使用<style>标签。

为了在 HTML 文档中引用外部样式表中的 CSS 样式规则，可以使用<link>标签定义当前文档与外部样式表文件之间的关系。link 元素是一个空元素，它仅包含一些属性。这个元素只能存在于文档中的 head 部分，不过它可以出现任意次。

link 元素具有以下局部属性。

href：指定目标文档或资源的 URL 地址。

hreflang：定义目标 URL 的基准语言。

media：规定文档将显示在什么设备上。

rel：定义当前文档与目标文档之间的关系。当链接外部 CSS 样式表时该属性的值应设置为“stylesheet”。

type：规定目标 URL 的 MIME 类型，链接外部 CSS 样式表时该属性的值应为“text/css”。

例 1.4　本例说明如何创建和引用外部 CSS 样式表文件。

外部 CSS 样式表文件 mystyle.css 的内容如下：

```
h3, p {
    font-family: 微软雅黑;
}

p {
    border-width: 1px;
    border-style: solid;
    border-color: red;
}
```

HTML 网页的源代码如下：

```
<!doctype html>
<html>
<head>
<meta charset="gb2312">
<title>链接外部 CSS 样式表</title>
<link rel="stylesheet" href="mystyle.css" type="text/css">
</head>

<body>
<h3>什么是 CSS</h3>
<hr>
<p>CSS 是英文 Cascading Style Sheets 的缩写，中文含义是层叠样式表。CSS 是用于控制或增加网页外观样式的一种标记语言。</p>
</body>
</html>
```

在这个例子中，创建了一个 CSS 样式表文件和一个 HTML 网页，这两个文件位于相同的文件夹中。

在 CSS 样式表文件中定义了两条 CSS 样式规则。第一条 CSS 规则将 h3 和 p 元素的字体设置为微软雅黑；第二个 CSS 规则是对 p 元素设置了以下边框属性：border-width 属性指定边框宽度为 1 像素，border-style 属性指定边框样式为实线，border-color 属性指定边框颜色为红色。

在浏览器中打开上述网页，显示效果如图 1.6 所示。

图 1.6　链接外部 CSS 样式表文件

1.6 使用脚本元素

除了 HTML 元素和 CSS 样式之外，在 HTML 网页中还可以使用 JavaScript 语言编写脚本代码，以实现仅用 HTML 和 CSS 所不能完成的功能。在 HTML5 中，与脚本相关的元素有两个：一个是 script 元素，用于定义脚本并控制其执行过程；另一个是 noscript 元素，用于规定在浏览器不支持脚本或禁用了脚本的情况下的处理办法。

1.6.1 在文档中添加脚本

在 HTML 文档中定义脚本有两种方法可供选择：一种方法是定义内嵌脚本，此时脚本是 HTML 文档的一部分；另一种方法是定义外部脚本，此时脚本包含在另一文件（文件扩展名为.js）中，通过一个 URL 引用。

在 HTML 文档中，可以使用 script 元素来定义一段脚本，脚本内容可以使用 JavaScript 或 VBScript 语言来编写。

script 元素具有下列局部属性。

async：定义脚本是否异步执行，它是 HTML5 中新增的一个属性。

type：指示脚本的 MIME 类型，如果使用 JavaScript 语言编写脚本，可将该属性的值设置为“text/javascript”；如果使用 VBScript 语言编写脚本，则应将该属性设置为“text/vbscript”。

defer：指示脚本不会生成任何的文档内容，浏览器可以继续解析并绘制页面。

src：指定包含脚本的文件的 URL，这样就可以引用某个包含脚本的文件，而不是直接把脚本插入 HTML 文档中。

例 1.5　本例说明如何在 HTML 网页中编写一段内嵌的 JavaScript 脚本。源代码如下：

```
<!doctype html>
<html>
<head>
<meta charset="gb2312">
<title>编写内嵌脚本</title>
</head>

<body>
<script type="text/javascript">
var now=new Date();
var hour=now.getHours();

if (hour<12) {
   document.writeln("上午好！ ");
} else {
   document.writeln("下午好！ ");
}
</script>
</body>
</html>
```

在本例中，使用 script 元素在网页的 body 部分编写了一段 JavaScript 脚本，其功能是根据当前系统时间的不同显示相应的问候语。

在脚本代码中，首先使用 var 关键字声明变量 now，并用它指向当前计算机的日期和时间；然后使用 var 关键字声明变量 hour，并从日期对象中取出本地时间的小时值。接下来，使用 if

语句对当前时间进行判断，如果当前在 12 点之前，则通过调用 document 对象的 writeln 方法显示问候语“上午好!”，否则显示“下午好!”。脚本运行结果如图 1.7 所示。

图 1.7　使用 script 元素编写内嵌脚本

在上述例子中，在 HTML 网页中编写了一段 JavaScript 脚本代码，当打开该网页时就会执行这段脚本代码。如果希望在多个网页中执行这段脚本代码，则应将这个脚本代码保存到一个独立的脚本文件中，这种文件称为外部脚本文件，其文件扩展名为.js。在脚本文件中编写代码时，不再需要使用<script>标签。

创建脚本文件之后，就可以在 HTML 网页中使用<script>标签来导入脚本代码，为此只需要将 src 属性设置为这个脚本文件的 URL 即可。

提示：一旦设置了 script 元素的 src 属性，则不允许在开始标签<script>与结束标签</script>之间放置任何脚本代码。

例 1.6　本例说明如何在 HTML 网页中导入外部脚本文件。

编写 JavaScript 脚本文件 myscript.js，源代码如下：

```
var now=new Date();
var hour=now.getHours();

if (hour<12) {
   document.writeln("上午好！ ");
} else {
   document.writeln("下午好！ ");
}
```

编写 HTML 网页，源代码如下：

```
<!doctype html>
<html>
<head>
<meta charset="gb2312">
<title>导入外部脚本</title>
</head>

<body>
<script type="text/javascript" src="myscript.js"></script>
</body>
</html>
```

这个例子的运行结果与例 1.5 相同，这里不再赘述。

1.6.2　定义脚本未执行时的内容

目前 JavaScript 语言已经得到了多数主流浏览器的广泛支持，但是仍然有一些专门用途的浏览器不支持它。在某些情况下，就算浏览器支持 JavaScript，用户也可能根据需要禁用它，

许多公司都有禁止员工启用 JavaScript 的规定。使用 noscript 元素可以用来应对这种情况，其办法是显示不需要 JavaScript 功能的内容，或者告诉用户需要启用 JavaScript 才能使用此网站或页面。

例 1.7　本例说明如何使用 noscript 元素定义脚本未执行时的内容。源代码如下：

```
<!doctype html>
<html>
<head>
<meta charset="gb2312">
<title>定义脚本未执行时的内容</title>
</head>

<body>
<script defer src="myscript.js"></script>
<noscript>
<h3>需要启用 JavaScript</h3>
<hr>
<p>您的浏览器不支持或禁用了 JavaScript。</p>
</noscript>
</body>
</html>
```

在 IE 浏览器中打开该页面，显示结果如图 1.8 所示。

图 1.8　使用 noscript 元素显示脚本未执行时的内容

习题 1

一、选择题

1．对一个 HTML 元素可以同时应用多个属性，这些属性之间用（　　）分隔。

A．制表符　　B．逗号　　C．空格　　D．分号

2．对于布尔属性，以下用法中错误的是（　　）。

A．将属性名添加到开始标签中　　B．将其值设置为空字符串

C．将其值设置为属性名字符串　　D．将属性名添加到结束标签中

3．要应用 CSS 样式表中的类，需要设置元素的（　　）属性。

A．accesskey　　B．class　　C．contenteditable　　D．contextmenu

4．要规定元素内嵌样式，需要设置元素的（　　）属性。

A．id　　B．title　　C．style　　D．lang

5．位于 HTML 文档第一行的是（　　）元素。

A．html　　B．doctype　　C．head　　D．body

二、判断题

1.（　　）HTML5 于 2014 年 10 月 29 日作为 W3C 推荐标准发布。

2.（　　）h3 和 a 元素都是虚元素。

3.（　　）HTML 元素的属性可以用在开始标签、结束标签或单个标签上。

4.（　　）虚元素也可以用空元素结构表示。

5.（　　）在 HTML 文档中，head 元素和 body 元素就是兄弟元素，它们同是 html 元素的子元素。

三、简答题

1. HTML5 包含哪 3 项重要技术？它们的作用分别是什么？

2. HTML5 具有哪些新特性？

3. 在 HTML5 中元素分为哪些类型？

4. 如何为 HTML 文档设置标题？

5. 如何在 HTML 文档中定义内嵌 CSS 样式表？

6. 如何在 HTML 文档中链接外部 CSS 样式表？

7. 如何在 HTML 文档中添加 JavaScript 脚本？

8. 如何在 HTML 文档中导入外部脚本文件？

上机操作 1

1. 编写一个网页，用于显示一条欢迎信息。

2. 编写一个网页，用 h1 元素显示一个标题，用 hr 元素显示一条水平分隔线，用 p 元素定义一个段落，要求创建内嵌 CSS 样式表，用于设置 h1 和 p 元素的字体为微软雅黑，p 元素的背景为灰色，文字为白色。

3. 编写一个网页，用 h1 元素显示一个标题，用 hr 元素显示一条水平分隔线，用 p 元素定义一个段落，要求链接外部 CSS 样式表，通过此样式表设置 h1 和 p 元素的字体为微软雅黑，p 元素的背景为灰色，文字为白色。