

CHAPTER 01

HTML5 轻松上手

本章概述 SUMMARY

HTML5 在废除很多标签的同时，也增加了很多标签，比如新增的结构标签：section 元素 / video 元素等。本章主要通过讲解与 HTML4 的不同逐渐对这些新增和废除的元素加深了解，希望通过本章的讲解大家能够掌握这些知识。

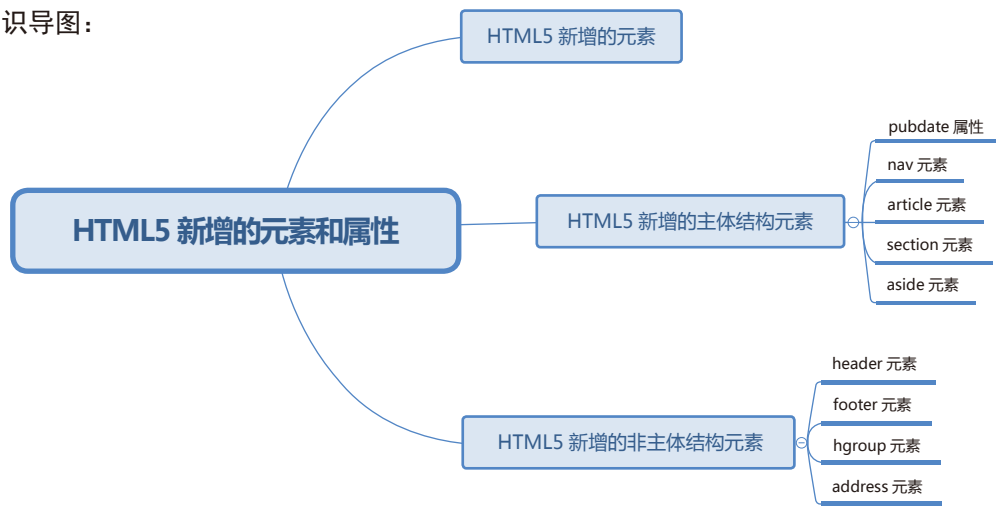
■ 学习目标

了解 HTML5 的语法和元素分类。
掌握 HTML5 中新增主体结构元素的定义。
掌握 HTML5 中新增的非主体结构元素的定义。
掌握 HTML5 中新增主体和非主体结构元素的使用方法。
掌握 HTML5 中废除的元素和属性。

■ 课时安排

理论知识 1 课时。
上机练习 1 课时。

知识导图：



1.1 HTML5 新增知识

HTML5 将成为 HTML、XHTML 以及 HTML DOM 的新标准。而 HTML5 本身并非技术，而是标准。它所使用的技术早已很成熟，国内通常所说的 HTML5 实际上是 HTML 与 CSS3，及 JavaScript 和 API 等的一个组合，大概可以用下式说明：

HTML5 $\approx$ HTML+CSS3+JavaScript+API

1.1.1 HTML5 的兼容性

HTML5 的一个核心理念就是保持一切新特性的平滑过渡。一旦浏览器不支持 HTML5 的某项功能，针对该项功能的备用方案就会被启用。另外，互联网上有些 HTML 文档已经存在很长时间了，因此，支持所有的现存 HTML 文档是非常重要的。HTML5 的研究者们还花费了大量精力来实现 HTML5 的通用性。很多开发人员使用 `<div id="header">` 来标记页眉区域。而在 HTML5 当中添加一个 `<header>` 就可以解决这个问题。

在浏览器方面，支持 HTML5 的浏览器包括 Firefox（火狐浏览器），IE9 及其更高版本，Chrome（谷歌浏览器），Safari，Opera 等；国内的各种基于 IE 或 Chromium（Chrome 的工程版或称实验版）所推出的 360 浏览器、搜狗浏览器、QQ 浏览器、猎豹浏览器等国产浏览器同样具备支持 HTML5 的能力。

HTML5 将会取代 1999 年制定的 HTML 4.01、XHTML 1.0 标准，以期能在互联网应用迅速发展的时候，使网络标准符合当代的网络需求，为桌面和移动平台带来无缝衔接的丰富内容。HTML5 在功能上做了以下几个方面的改进。

- 重新简化了 DOCTYPE。
- 重新简化了字符集声明。
- 简单而强大的 HTML5 API。
- 以浏览器的原生能力替代复杂的 JavaScript 代码。

1.1.2 HTML5 的通用访问

通用访问的原则可以分为以下三个方面。

（1）可访问性

出于对残障用户的考虑，HTML 与 WAI(Web Accessibility Initiative, Web 可访问性倡议) 和 ARIA(Accessible Rich Internet Applications, 可访问的 Internet 应用) 做到了紧密的结合，WAI-ARIA 中以屏幕阅读器为基础的元素已经被添加到 HTML 中。

（2）媒体中立

在不久的将来，将实现 HTML5 的所有功能都能够所有不同的设备和平台上正常运行。

（3）支持所有语种

能够支持所有的语种，例如，新的 `<ruby>` 标签支持在页面排版中会用到 Ruby 注释。

1.1.3 HTML5 标准改进

HTML5 提供了一些新的元素和属性，如 `<nav>`（网站导航栏）和 `<footer>`。这种标

签将有利于搜索引擎的索引整理，同时也能更好地帮助小屏幕装置和视障人士使用。除此之外，还为其他浏览要素提供了新的功能，如 `<audio>` 和 `<video>` 标签。

HTML5 吸取了 XHTML2 的一些建议，包括一些用来改善文档结构的功能，例如一些旧的 HTML 标签 `header`、`footer`、`section`、`dialog` 和 `aside` 的使用，使得内容创作者能够更加轻松地创建文档，之前开发人员在这些场合一律使用 `<div>` 标签。

HTML5 还包含了一些将内容和样式分离的功能，`<b>` 和 `<i>` 标签仍然存在，但是它们的意义已经和之前有了很大的不同，这些标签的意义只是将一段文字标识出来，而不是单纯为了设置粗体和斜体文字样式。`<u>`、`<font>`、`<center>` 和 `<strike>` 这些标签则完全被废弃了。

新标准使用了一些全新的表单输入对象，包括日期、URL 和 Email 地址，其他的对象则增加了对拉丁字符的支持。HTML 还引入了微数据，一种使用机器可以识别的标签标注内容的方法，使语义 Web 的处理更为简单。总的来说，这些与结构有关的改进使开发人员可以创建更干净、更容易管理的网页。

HTML5 具有全新的、更合理的 tag，多媒体对象不再全部绑定到 `object` 中，而是视频有视频的 tag，音频有音频的 tag。canvas 对象将给浏览器带来直接在上面绘制矢量图的能力，这意味着用户可以脱离 flash 和 silverlight，直接在浏览器中显示图形和动画。很多最新版的浏览器，除了 IE，都已经支持了 canvas。浏览器中的真正程序将提供 API 浏览器内的编辑、拖放，以及各种图形用户界面的能力。内容修饰 tag 将被移除，而使用 CSS。

1.2 HTML5 语法

HTML5 中的语法和之前版本相比有些变化，因为 HTML5 设计以化繁为简的准则对文档类型和字符说明等都进行了简化。下面将分别进行说明。

1.2.1 文档类型声明

DOCTYPE 声明是 HTML 文件中必不可少的，位于文件第一行，在 HTML4 中，它的声明方法如下：

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

在 HTML5 中，刻意不使用版本声明，一份文档将会使用于所有版本的 HTML。HTML5 中的 DOCTYPE 声明方法（不区分大小写）如下：

```
<!DOCTYPE html>
```

另外，当使用工具时，也可以在 DOCTYPE 声明方式中加入 SYSTEM 识别符，声明方法如下：

```
<!DOCTYPE HTML SYSTEM "about:legacy-compat">
```

在 HTML5 中，像这样的 DOCTYPE 声明方式是允许的，不区分大小写，引号不区分单引号和双引号。

知识拓展

使用 HTML5 的 DOCTYPE 会触发浏览器以标准兼容模式显示页面。众所周知，网页都有多种显示模式，浏览器会根据 DOCTYPE 来识别该使用哪种模式，以及使用什么规则来验证页面。

1.2.2 字符编码

在 HTML4 中，使用 `<meta>` 元素的形式指定文件中的字符编码，如下：

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" >
```

在 HTML5 中，可以使用对 `<meta>` 元素直接追加 `charset` 属性的方式来指定字符编码，如下：

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

两种方法都有效，可以继续使用前面一种方式，即通过 `content` 元素的属性来指定，但是不能同时混合使用两种方式。在以前的网站代码中可能会存在下面代码的标记方式，但在 HTML5 中，下面这种字符编码方式将被认为是错误的：

```
<meta charset="utf-8" http-equiv="Content-Type" content="text/html; charset=utf-8" >
```

从 HTML5 开始，对于文件的字符编码推荐使用 UTF-8。

1.2.3 省略引号

属性两边既可以用双引号，也可以用单引号。HTML5 在此基础上做了一些改进。当属性值不包括空字符串、`<`、`>`、`=`、单引号、双引号等字符时，属性值两边的引号可以省略。下面的写法都是合法的：

```
<input type="text">
<input type='text'>
<input type=text>
```

1.3 HTML5 元素分类

HTML5 新增了很多个元素，也废除了不少元素，根据现有的标准规范，把 HTML5 的元素按等级定义为结构性元素、级块性元素、行内语义性元素和交互性元素四大类。

1.3.1 结构性元素

结构性元素主要负责 Web 的上下文结构的定义，确保 HTML 文档的完整性，这类元素包括以下几个：

- **Section:** 在 Web 页面应用中, 该元素也可以用于区域的章节表述。
- **Header:** 页面主体上的头部, 注意区别于 head 元素。这里可以给初学者提供一个判断的小技巧: head 元素中的内容往往是不可见的; header 元素往往在一对 body 元素之中。
- **Footer:** 页面底部, 通常会在这里标出网站的一些相关信息, 例如, 关于我们、法律声明、邮件信息、管理入口等。
- **Nav:** 是专门用于菜单导航、链接导航的元素, 是 navigator 的缩写。
- **Article:** 用于表示一篇文章的主体内容, 一般文字集中显示的区域。

■ 1.3.2 级块性元素

级块性元素主要完成 Web 页面区域的划分, 确保内容的有效分隔, 这类元素包括以下几个:

- **Aside:** 用以表示注记、贴士、侧栏、摘要、插入的引用等作为补充主体的内容。从一个简单页面显示上看, 就是侧边栏, 可以在左边, 也可以在右边。从一个页面的局部看, 就是摘要。
- **Figure:** 是对多个元素组合并展示的元素, 通常与 figcaption 联合使用。
- **Code:** 表示一段代码块。
- **Dialog:** 用于表达人与人之间的对话, 该元素还包括 dt 和 dd 这两个组合元素, 它们常常同时使用, dt 用于表示说话者, 而 dd 则用来表示说话者说的内容。

■ 1.3.3 行内语义性元素

行内语义性元素主要完成 Web 页面具体内容的引用和表示, 是丰富内容展示的基础, 这类元素包括以下几个:

- **Meter:** 表示特定范围内的数值, 可用于工资、数量、百分比等。
- **Time:** 表示时间值。
- **Progress:** 用来表示进度条, 可通过对其 max、min、step 等属性进行控制, 完成进度的表示和监视。
- **Video:** 视频元素, 用于支持和实现视频文件的直接播放, 支持缓冲预载和多种视频媒体格式, 如 MPEG-4、OGGV、WEBM 等。
- **Audio:** 音频元素, 用于支持和实现音频文件的直接播放, 支持缓冲预载和多种音频媒体格式。

■ 1.3.4 交互性元素

交互性元素主要用于功能性的内容表达, 会有一定的内容和数据的关联, 是各种事件的基础, 这类元素包括以下几个:

- **Details:** 用来表示一段具体的内容, 但是内容默认可能不显示, 通过某种手段 (如单击) legend 交互才会显示。
- **Datagrid:** 用来控制客户端数据与显示, 可以由动态脚本及时更新。
- **Menu:** 主要用于交互表单。
- **Command:** 用来处理命令按钮。

1.4 HTML5 新增主体结构元素

HTML5 引用更多灵活的段落标签和功能标签。与 HTML4 相比，HTML5 的结构元素更加成熟。本节将带领大家了解这些新增的结构元素，包括它们的定义、表示意义和使用示例。

1.4.1 article 元素

article 元素一般用于文章区块，定义外部内容。比如某篇新闻的文章，或者来自微博的文本，或者来自论坛的文本。通常用来表示来自其他外部源内容，它可以独立被外部引用。

语法描述：

```
<article> </article>
```

小试身手——文章区块和外部内容的定义方式

下面的代码就是 article 的用法：

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>article 元素 </title>
<style>
h1, h2, p{text-align: center;}
</style>
</head>
<body>
<article>
<header>
<hgroup>
<h1>article 元素 </h1>
<h2>article 元素 HTML5 中的新增结构元素 </h2>
</hgroup>
</header>
<p>Article 元素一般用于文章区块，定义外部内容。</p>
<p>比如某篇新闻的文章，或者来自微博的文本，或者来自论坛的文本。</p>
<p>通常用来表示来自其他外部源内容，它可以独立被外部引用。</p>
</article>
</body>
</html>
```

代码的运行效果如图 1-1 所示。



图 1-1

1.4.2 section 元素

section 元素主要用来定义文档中的节（section）。比如章节、页面、页脚或文档中的其他部分。通常它用于成节的内容，或在文档流中开始一个新的节。

语法描述：

```
<section> </section>
```

小试身手——定义文档中的节

Section 元素的具体使用代码如下：

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>section 元素 </title>
<style>
h1, p{text-align: center;}
</style>
</head>
<body>
<section>
<h1>section 元素 </h1>
<p>section 元素是 HTML5 中新增的结构元素 </p>
<p>section 元素是 HTML5 中新增的结构元素 </p>
<p>section 元素是 HTML5 中新增的结构元素 </p>
<p>section 元素是 HTML5 中新增的结构元素 </p>
<p>section 元素是 HTML5 中新增的结构元素 </p>
</section>
</body>
</html>
```

代码的运行结果如图 1-2 所示。

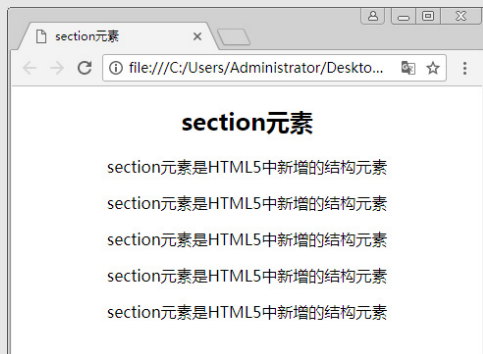


图 1-2

知识拓展

对于那些没有标题的内容，不推荐使用 section 元素。section 元素强调的是一个专题性的内容，一般会带有标题。当元素内容聚合起来表示一个整体时，应该使用 article 元素替代 section 元素。section 元素应用的典型情况有文章的章节标签、对话框中的标签页，或者网页中有编号的部分。section 元素不仅仅是一个普通的容器元素。当 section 元素只是为了样式或者方便脚本使用，这时应该使用 div。一般来说，当元素内容明确地出现在文档大纲中时，section 就是适用的。

1.4.3 nav 元素

nav 元素用来定义导航栏链接的部分。链接用来链接到本页的某部分或其他页面。

我们需要注意的是，并不是所有成组的超链接都需要放在 nav 元素里。nav 元素里应该放入一些当前页面的主要导航链接。

语法描述：

```
<nav> </nav>
```

小试身手——网页中的各种导航栏的定义

nav 的元素使用代码如下：

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>nav 元素 </title>
</head>
<body>
```



```
<h1>HTML5 结构元素 </h1>
<nav>
<ul>
<li><a href="#">items01</a></li>
<li><a href="#">items02</a></li>
</ul>
</nav>
<header>
<h2>nav 元素 </h2>
<nav>
<ul>
<li><a href="#">nav 元素的应用场景 01</a></li>
<li><a href="#">nav 元素的应用场景 02</a></li>
<li><a href="#">nav 元素的应用场景 03</a></li>
<li><a href="#">nav 元素的应用场景 04</a></li>
</ul>
</nav>
</header>
</body>
</html>
```

代码的运行效果如图 1-3 所示。



图 1-3

这里需要注意的是，上面的示例就是 `nav` 元素应用的场景，我们通常会把主要的链接放入 `nav` 当中。

■ 1.4.4 aside 元素

`aside` 元素用来定义 `article` 以外的内容，用于成节的内容，也可以用于表达记注、侧栏、摘要及插入的引用等诸如补充主体的内容。它会在文档流中开始一个新的节，一般用于与文章内容相关的侧栏。

语法:

```
<aside> </aside>
```

小试身手——定义文章的侧栏

aside 元素的使用方法代码如下:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<title>aside 元素 </title>
<link rel="stylesheet" href="">
</head>
<body>
<article>
<h1>HTML5aside 元素 </h1>
<p>正文部分 </p>
<aside>正文部分的附属信息部分,其中的内容可以是与当前文章有关的相关资料、名词解释,等等。
</aside>
</article>
</body>
</html>
```

代码的运行效果如图 1-4 所示。



图 1-4

1.4.5 pubdate 属性

pubdate 属性是一个可选的、boolean值的属性,它可以用到 article 元素中的 time 元素上,意思是 time 代表了文章或整个网页的发布日期。

小试身手——定义时间

定义时间的具体应用方法代码如下：

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>pubdate 属性 </title>
</head>
<body>
<article>
<header>
<h1> 澳门 </h1>
<p> 我国澳门特别行政区是于 <time datetime="1999-12-10">1999 年 12 月 20 日 </time> 回归的
</p>
<p>notice date <time datetime="2017-08-15" pubdate>2017 年 08 月 15 日 </time></p>
</header>
<p> 正文部分 ...</p>
</article>
</body>
</html>
```

代码的运行效果如图 1-5 所示。



图 1-5

在这个示例中有两个 time 元素，分别定义了两个日期，一个是回归日期，另一个是发布日期。由于都是用了 time 元素，所以需要使用 pubdate 属性表明哪个 time 元素代表了发布日期。

1.5 HTML5 新的非主体结构元素

HTML5 中不仅新增了主体结构元素，还增加了一些非主体元素，比如 header 元素、hgroup 元素、footer 元素和 address 元素等，这些元素的增加使我们的工作又轻松很多。本节分别讲解非主体结构元素的使用。

1.5.1 header 元素

header 元素是一种具有引导和导航作用的辅助元素，它通常代表一组简介或者导航性质的内容。其位置表现在页面或节点的头部。

通常 header 元素用于包含页面标题，当然这不是绝对的，header 元素也可以用于包含节点的内容列表导航，如数据表格、搜索表单或相关的 logo 图片等。

在整个页面中，标题一般放在页面的开头，一个网页中没有限制 header 元素的个数，可以拥有多个，可以为每个内容区块加一个 header 元素。

语法：

```
<header> </header>
```

小试身手——页面标题的制作

页面标题的制作代码如下：

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>header 元素 </title>
</head>
<body>
<header>
<h1> 这是页面的标题 </h1>
</header>
<article>
<h2> 这是第一章 </h2>
<p> 第一章的正文部分 ...</p>
</article>
<header>
<h2> 第二个 header 标签 </h2>
<p> 因为 html 文档不会对 header 标签进行限制，所以我们可以创建多个 header 标签 </p>
</header>
</body>
</html>
```

代码的运行效果如图 1-6 所示。

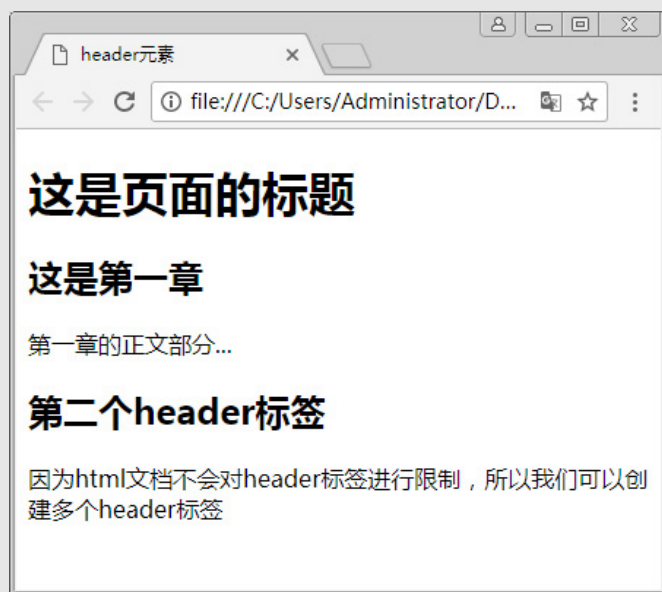


图 1-6

1.5.2 hgroup 元素

在上节中介绍 header 元素时，使用了 hgroup 元素，hgroup 元素的目的是将不同层级的标题封装成一组，通常会将 h1~h6 标题进行组合，比如一个内容区块的标题及其子标题为一组。如果要定义一个页面的大纲，使用 hgroup 元素非常合适，如定义文章的大纲层级。

小试身手——组合标题

组合标题的使用代码如下：

```
<hgroup>
<h1> 第三节 </h1>
<h2>2.5hgroup 元素 </h2>
</hgroup>
```

在以下两种情况下，header 元素和 hgroup 元素不能一起使用。

第一，当只有一个标题的时候。

示例代码如下：

```
<header>
<hgroup>
<h1> 第三节 </h1>
<p> 正文部分 ...</p>
</hgroup>
</header>
```

在这种情况下，只能将 hgroup 元素移除，仅仅保留其标题元素即可。

```
<header>
<h1> 第三节 </h1>
<p> 正文部分 ...</p>
</header>
```

第二，当 header 元素的子元素只有 hgroup 元素的时候。

示例代码如下：

```
<header>
<hgroup>
<h1>HTML5 hgroup 元素 </h1>
<h2>hgroup 元素使用方法 </h2>
</hgroup>
</header>
```

在上面的代码中，header 元素的子元素只有 hgroup 元素，这时并没有其他元素放到 header 中，就可以直接将 header 元素去掉，如下：

```
<hgroup>
<h1>HTML5 hgroup 元素 </h1>
<h2>hgroup 元素使用方法 </h2>
</hgroup>
```

知识拓展

如果只有一个标题元素，这时并不需要 hgroup 元素。当出现两个或者两个以上的标题元素时，适合用 hgroup 元素来包围它们。当一个标题有副标题或者其他的与 section 或者 article 有关的元数据时，适合将 hgroup 元素和元数据放到一个单独的 header 元素中。

1.5.3 footer 元素

长久以来，人们习惯于使用 <div id="footer"> 这样的代码来定义页面的页脚部分。但是在 HTML5 中我们不需要如此了。HTML5 为我们提供了用途更广、扩展性更强的 footer 元素。<footer> 标签定义文档或节的页脚。<footer> 元素应当含有其包含元素的信息。页脚通常包含文档的作者、版权信息、使用条款链接、联系信息等。您可以在一个文档中使用多个 <footer> 元素。

小试身手——网页尾部的设计

过去，程序员在标脚注的时候通常使用这样的代码：

```
<div id="footer">
<ul>
```

```
<li> 关于我们 </li>
<li> 网站地图 </li>
<li> 联系我们 </li>
<li> 回到顶部 </li>
<li> 版权信息 </li>
</ul>
</div>
```

而我们现在不需要再这样写了，而是使用 footer：

```
<footer>
<ul>
<li> 关于我们 </li>
<li> 网站地图 </li>
<li> 联系我们 </li>
<li> 回到顶部 </li>
<li> 版权信息 </li>
</ul>
</footer>
```

代码的运行效果如图 1-7 所示。



图 1-7

相比较而言，使用 footer 元素更加语义化了。

知识拓展

同样，在一个页面中也可以使用多个 footer 元素，既可以用做页面整体的页脚，也可以作为一个内容区块的结尾。

比如在 `article` 元素中添加脚注，代码如下：

```
<article>
<h1> 文章标题 </h1>
<p> 正文部分 ...</p>
<footer> 文章脚注 </footer>
</article>
```

在 `section` 元素中添加脚注，代码如下：

```
<section>
<h1> 段落标题 </h1>
<p> 正文部分 </p>
<footer> 本段脚注 </footer>
</section>
```

以上部分就是在 `article` 元素和 `section` 元素中添加脚注的方式。

1.5.4 address 元素

`<address>` 标签定义文档或文章的作者 / 拥有者的联系信息。

如果 `<address>` 元素位于 `<body>` 元素内，则它表示文档的联系信息。

如果 `<address>` 元素位于 `<article>` 元素内，则它表示文章的联系信息。

`<address>` 元素中的文本通常呈现为斜体。大多数浏览器会在 `address` 元素前后添加折行。

小试身手——定义文章作者信息

`address` 元素的具体使用方法：

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>address 元素 </title>
</head>
<body>
<header>
<address>
写信给我们 <br/>
<a href="xxxitanyxxx.com"> 进入官网 </a><br/>
地址：江苏徐州云龙区矿大软件园 458 号 8 栋 <br/>
tel: 221333
</address>
</header>
</body>
</html>
```

代码的运行效果如图 1-8 所示。

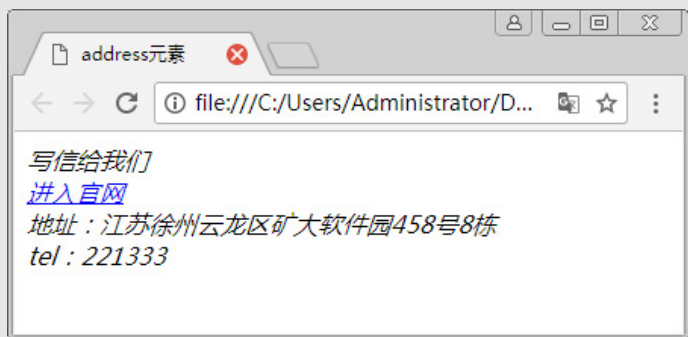


图 1-8

1.6 新增的属性

在 HTML5 中不仅新增了许多元素，还新增了一些属性，在新增的这些属性中表单的属性最为重要，下面将一一讲解。

1.6.1 表单相关属性

在 HTML5 中，表单新增的属性如下：

- autofocus 属性：该属性可以用于 input (type=text, select, textarea, button) 元素当中。autofocus 属性可以让元素在打开页面时自动获得焦点。
- placeholder 属性：该属性可以用于 input(type=text, password, textarea) 元素当中，使用该属性会对用户的输入进行提示，通常用于提示用户可以输入的内容。
- form 属性：该属性用于 input、output、select、textarea、button 和 fieldset 元素当中。
- Required 属性：该属性用于 input(type=text) 元素和 textarea 元素当中，表示用户提交时进行检查，检查该元素内一定要有输入内容。
- 在 input 元素与 button 元素中增加了新属性 formaction、formenctype、formmethod、formnovalidate 与 formtarget，这些属性可以重载 form 元素的 action、enctype、method、novalidate 与 target 属性。
- 在 input 元素、button 元素和 form 元素中增加了 novalidate 属性，该属性可以取消提交时进行的有关检查，表单可以被无条件地提交。

1.6.2 其他相关属性

在 HTML5 中，新增的与链接相关的属性分别如下：

- 在 a 与 area 元素中增加了 media 属性，该属性规定目标 URL 是用什么类型的媒介进行优化的。
- 在 area 元素中增加了 hreflang 属性与 rel 属性，以保持与 a 元素和 link 元素的一致。

- 在 link 元素中增加了 sizes 属性。该属性用于指定关联图标 (icon 元素) 的大小, 通常可以与 icon 元素结合使用。
- 在 base 元素中增加了 target 属性, 主要目的是保持与 a 元素的一致性。
- 在 meta 元素中增加了 charset 属性, 该属性为文档的字符编码的指定提供了一种良好的方式。
- 在 meta 元素中增加了 type 和 label 两个属性。label 属性为菜单定义一个可见的标注; type 属性让菜单可以以上下文菜单、工具条与列表菜单 3 种形式出现。
- 在 style 元素中增加了 scoped 属性, 用来规定样式的作用范围。
- 在 script 元素中增加了 async 属性, 该属性用于定义脚本是否异步执行。

1.7 课堂练习

以上是 HTML5 中新增的知识点。温故而知新, 本小节为大家准备了一个课堂练习, 运行效果如图 1-9 所示。



图 1-9

相信大家看到这张图片一定在想到底该用什么元素来设计呢? 下面给大家展示完整的代码。

完整的代码如下:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title> 课堂练习 </title>
</head>
<body>
<!-- 通常不推荐没有标题内容使用 section 元素 -->
  <!-- 不要与 article 元素混淆 -->
  <section>
    <h1> 香蕉 </h1>
    <p> 这是一种水果 </p>
  </section>
  <article>
    <h1> 苹果 </h1>
    <p> 这是一种水果 </p>
    <section>
      <h2> 红富士 </h2>
      <p> 这个苹果很好吃！！ </p>
    </section>
    <section>
      <h2> 洛川苹果 </h2>
      <p> 这个是陕西盛产的苹果！！ </p>
    </section>
  </article>
<!-- 注意 section 和 article 的区别 -->
  <section>
    <h1> 有很多水果 </h1>
    <article>
      <h2> 香蕉 </h2>
      <p> 食用香蕉的好处 </p>
    </article>
    <article>
      <h2> 苹果 </h2>
      <p> 苹果含有维生素 </p>
    </article>
    <article>
      <h2> 西瓜 </h2>
      <p> 西瓜很好吃，其中含有 ...</p>
    </article>
  </section>
</body>
</html>
```

强化训练

学习完了本章的知识后, 相信大家对 HTML5 中新增属性和元素有了一个全新的认识。下面的练习就是根据本节所涉及的知识设计的, 请根据图 1-10 ~ 图 1-12 所示制作出一样的效果。

图 1-10 是在对话框中输入数字 2。

图 1-11 是在第二个对话框中输入数字 3。

图 1-12 是单击“确定”按钮, 可以得到两个数的乘积为 6。

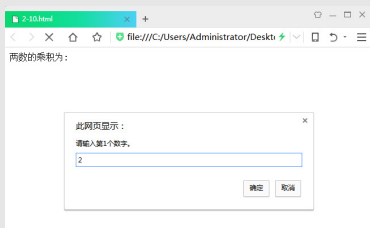


图 1-10

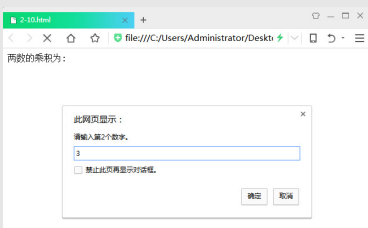


图 1-11

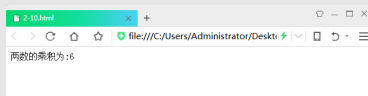


图 1-12

操作技巧

利用新增的元素属性 function multi() 函数可以做出一个简单的计算器的效果。

提示代码如下:

```
<script type="text/javascript">
function multi(){
    a=parseInt(prompt(" 请输入第 1 个数字。", 0));
    b=parseInt(prompt(" 请输入第 2 个数字。", 0));
    document.forms["form"]["result"].value=a*b;
}
</script>
```

本章结束语

本章详细讲述了 HTML5 新增的元素和属性、HTML5 和 HTML4 在语法、元素和属性上的差异。

通过本章的学习, 相信大家已对 HTML5 主体结构元素和非主体结构元素有了一定的了解。这些元素明显地比以前的 div 标签更加具有语义化; 但是如何使用和熟悉这些标签还是需要大家不断地去使用它们。