



第3章 HTML 5的结构

- 3.1 新增的主体结构元素
- 3.2 新增的非主体结构元素
- 3.3 HTML 5结构

在HTML 5对HTML 4所做的各种修改中，一个比较重大的修改就是为了使文档结构更加清晰明确，容易阅读，增加了很多新的结构元素。本章将详细介绍这些新增的结构元素，包括它们的定义、使用方法以及使用示例，最后再介绍一下在HTML 5中，究竟怎样将这些新增的结构元素结合起来综合使用。

学习内容：

- 掌握HTML 5中新增的主体结构元素的定义、使用方法，以及使用场合。新增的主体结构元素包括article元素、section元素、nav元素及aside元素。
- 掌握HTML 5中新增的非主体结构元素的定义、使用方法，以及使用场合。新增的非主体结构元素包括header元素、hgroup元素、footer元素及address元素。
- 掌握HTML 5中应该怎样结合运用这些新增结构元素来合理编排页面总体布局，掌握什么是显式编排，什么是隐式编排，HTML 5分析器是按什么原则来分析页面结构的，以及怎样对这些新增元素使用CSS样式。

3.1 新增的主体结构元素

在HTML 5中，为了使文档的结构更加清晰明确，追加了几个与页眉、页脚、内容区块等文档结构相关联的结构元素。需要说明的是，本章所讲的内容区块是指将HTML页面按逻辑进行分割后的单位。例如对于书籍来说，章、节都可以称为内容区块；对于博客网站来说，导航菜单、文章正文、文章的评论等每一个部分都可称为内容区块。

接下来将详细讲解HTML 5中在页面的主体结构方面新增加的结构元素。

3.1.1 article元素

article元素代表文档、页面或应用程序中独立的、完整的、可以独自被外部引用的内容。它可以是一篇博客或报刊中的文章、一篇论坛帖子、一段用户评论或独立的插件，或其他任何独立的内容。

除了内容部分，一个article元素通常有它自己的标题（一般放在一个header元素里面），有时还有自己的脚注。

现在，让我们以博客为例来看一段关于article元素的代码示例，如代码清单3-1所示。

代码清单3-1 article元素示例

```
<article>
  <header>
    <h1>苹果</h1>
    <p>发表日期: <time pubdate="pubdate">2010/10/09</time></p>
  </header>
  <p><b>苹果</b>，植物类水果，多次花果... ("苹果"文章正文)</p>
  <footer>
    <p><small>著作权归***公司所有。</small></p>
  </footer>
</article>
```

这个示例是一篇讲述苹果的博客文章，在header元素中嵌入了文章的标题部分，在这部分中，文章的标题“苹果”被镶嵌在h1元素中，文章的发表日期镶嵌在p元素中。在标题下部的p元素中，嵌入了一大段该博客文章的正文，在结尾处的footer元素中，嵌入了文章的著作权，作为脚注。整个示例的内容相对比较独立、完整，因此，对这部分内容使用了article元素。

article元素是可以嵌套使用的，内层的内容在原则上需要与外层的内容相关联。例如，一篇博客文章中，针对该文章的评论就可以使用嵌套article元素的方式；用来呈现评论的article元素被包含在表示整体内容的article元素里面。

接着，让我们来看一个关于article元素嵌套的代码示例，如代码清单3-2所示。

代码清单3-2 article元素嵌套示例

```

<article>
  <header>
    <h1>苹果</h1>
    <p>发表日期:
      <time pubdate datetime="2010/10/09">2010/10/09</time>
    </p>
  </header>
  <p><b>苹果</b>，植物类水果，多次花果... ("苹果"文章正文) </p>
  <section>
    <h2>评论</h2>
    <article>
      <header>
        <h3>发表者：陆凌牛</h3>
        <p>
          <time pubdate datetime="2010-10-10T19:10-08:00">
            1小时前
          </time>
        </p>
      </header>
      <p>我喜欢苹果，我最喜爱的品种是红富士。</p>
    </article>
    <article>
      <header>
        <h3>发表者：张玉</h3>
        <p>
          <time pubdate datetime="2010-10-10T19:15-08:00">
            1小时前
          </time>
        </p>
      </header>
      <p>苹果？我不喜欢，我喜欢吃橘子。</p>
    </article>
  </section>
</article>

```

这个示例中的内容比代码清单3-1中的内容更加完整了，它添加了文章读者的评论内容，示例的整体内容还是比较独立、完整的，因此对其使用article元素。具体来说，示例内容又分为几部分，文章标题放在了header元素中，文章正文放在了header元素后面的p元素中，然

后section元素把正文与评论部分进行了区分（section元素马上就要讲到，是一个分块元素，用来把页面中的内容进行分块），在section元素中嵌入了评论的内容，评论中每一个人的评论相对来说又是比较独立、完整的，因此对它们都使用一个article元素，在评论的article元素中，又可以分为标题与评论内容部分，分别放在header元素与p元素中。

关于示例中提到的time元素与pubdate属性，请查看本节结尾处有关time元素与pubdate属性的说明。

另外，article元素也可以用来表示插件，它的作用是使插件看起来好像内嵌在页面中一样。代码清单3-3是它的一个示例。

代码清单3-3 用article元素表示插件

```
<article>
  <h1>My Fruit Spinner</h1>
  <object>
    <param name="allowFullScreen" value="true">
    <embed src="#" width="600" height="395"></embed>
  </object>
</article>
```

3.1.2 section元素

section元素用于对网站或应用程序中页面上的内容进行分块。一个section元素通常由内容及其标题组成。但section元素并非一个普通的容器元素；当一个容器需要被直接定义样式或通过脚本定义行为时，推荐使用div而非section元素。

我们可以这样理解：section元素中的内容可以单独存储到数据库中或输出到Word文档中。代码清单3-4是它的一个示例（注意：标题部分位于它的内部，而不是它的前面）。

代码清单3-4 section元素示例

```
<section>
  <h1>苹果</h1>
  <p><b>苹果</b>，植物类水果，多次花果...（"苹果"文章正文）</p>
</section>
```

通常不推荐为那些没有标题的内容使用section元素，可以使用HTML 5轮廓工具来检查页面中是否有没标题的section，HTML 5轮廓工具的网址为“<http://gsnedders.html5.org/outliner/>”，如果使用该工具进行检查后，发现某个section的说明中有“untitled section”（没有标题的section）文字，这个section就有可能使用不当（但是nav元素或aside元素没有标题是合理的）。

section元素的作用是对页面上的内容进行分块，或者说对文章进行分段，请不要与“有着自己的完整的、独立的内容”的article元素混淆。

下面，我们来看article元素与section元素结合使用的两个示例，希望能够帮助你更好地理解article元素与section元素的区别。

首先来看一个带有section元素的article元素示例，如代码清单3-5所示。

代码清单3-5 带有section元素的article元素示例

```

<article>
  <h1>苹果</h1>
  <p><b>苹果</b>，植物类水果，多次花果...</p>
  <section>
    <h2>红富士</h2>
    <p>红富士是从普通富士的芽(枝)变中选育出的着色系富的统称...</p>
  </section>
  <section>
    <h2>国光</h2>
    <p>国光苹果品，又名小国光、万寿。原产美国，1600年发现的偶然实生苗...</p>
  </section>
</article>

```

代码清单3-5中的内容首先是一段独立的、完整的内容，因此使用article元素。该内容是一篇关于苹果的文章，该文章分为3段，每一段都有一个独立的标题，因此使用了两个section元素。请记住，对文章分段的工作也是使用section元素完成的。可能有人会问，为什么没有对第一段使用section元素，这里其实是可以使用section元素的，但是由于其结构比较清晰，分析器可以识别第一段内容在一个section元素里，所以也可以将第一个section元素略，但是如果第一个section元素里还要包含子section元素或子article元素，那么就必须写明第一个section元素了。

接着，我们来看一个包含article元素的section元素示例，如代码清单3-6所示。

代码清单3-6 包含article元素的section元素示例

```

<section>
  <h1>水果</h1>
  <article>
    <h2>苹果</h2>
    <p>苹果，植物类水果，多次花果...</p>
  </article>
  <article>
    <h2>橘子</h2>
    <p>橘子，是芸香科柑橘属的一种水果...</p>
  </article>
  <article>
    <h2>香蕉</h2>
    <p>香蕉，属于芭蕉科芭蕉属植物，又指其果实，热带地区广泛栽培食用...</p>
  </article>
</section>

```

这个示例比前面的示例复杂了一些，首先，它是一篇文章中的一段，因此最初没有使用article元素。但是，在这一段中有几块独立的内容，因此，嵌入了几个独立的article元素。

看到这里，你可能又会糊涂了，这两个元素可以互换使用吗？它们的区别到底是什么呢？事实上，在HTML 5中，article元素可以看成是一种特殊种类的section元素，它比section元素更强调独立性。即section元素强调分段或分块，而article强调独立性。具体来说，如果一块内容相对来说比较独立、完整的时候，应该使用article元素，但是如果你想将一块内容分成几段的时候，应该使用section元素。另外，在HTML 5中，div元素变成了一种容器，当

使用CSS样式的时候，可以对这个容器进行一个总体的CSS样式的套用。

另外再补充一点，在HTML 5中，你可以将所有页面的从属部分，譬如导航条、菜单、版权说明等包含在一个统一的页面中，以便统一使用CSS样式来进行装饰。

最后，关于section元素的使用禁忌总结如下：

- 1) 不要将section元素用作设置样式的页面容器，那是div元素的工作。
- 2) 如果article元素、aside元素或nav元素更符合使用条件，不要使用section元素。
- 3) 不要为没有标题的内容区块使用section元素。

3.1.3 nav元素

nav元素是一个可以用作页面导航的链接组，其中的导航元素链接到其他页面或当前页面的其他部分。并不是所有的链接组都要被放进nav元素，只需要将主要的、基本的链接组放进nav元素即可。例如，在页脚中通常会有一组链接，包括服务条款、首页、版权声明等，这时使用footer元素是最恰当。一个页面中可以拥有多个nav元素，作为页面整体或不同部分的导航。

接着让我们来看一个nav元素的使用示例，在这个示例中，一个页面由几部分组成，每个部分都带有链接，但只将最主要的链接放入了nav元素中，如代码清单3-7所示。

代码清单3-7 nav元素示例

```
<body>
<h1>技术资料</h1>
<nav>
  <ul>
    <li><a href="/">主页</a></li>
    <li><a href="/events">开发文档</a></li>
    ...more...
  </ul>
</nav>
<article>
  <header>
    <h1>HTML 5与CSS 3的历史</h1>
    <nav>
      <ul>
        <li><a href="#HTML 5">HTML 5的历史</a></li>
        <li><a href="#CSS 3">CSS 3的历史</a></li>
        ...more...
      </ul>
    </nav>
  </header>
  <section id="HTML 5">
    <h1>HTML 5的历史</h1>
    <p>讲述HTML 5的历史的正文</p>
  </section>
  <section id="CSS 3">
    <h1>CSS 3的历史</h1>
    <p>讲述CSS 3的历史的正文</p>
  </section>
```

```

...more...
<footer>
  <p>
    <a href="?edit">编辑</a> |
    <a href="?delete">删除</a> |
    <a href="?rename">重命名</a>
  </p>
</footer>
</article>
<footer>
  <p><small>版权所有：陆凌牛</small></p>
</footer>
</body>

```

在这个例子中，第一个nav元素用于页面导航，将页面跳转到其他页面上去（跳转到网站主页或开发文档目录页面）；第二个nav元素放置在article元素中，用作这篇文章中两个组成部分的页内导航。

具体来说，nav元素可以用于以下这些场合：

- ❑ 传统导航条。现在主流网站上都有不同层级的导航条，其作用是将当前画面跳转到网站的其他主要页面上去。
 - ❑ 侧边栏导航。现在主流博客网站及商品网站上都有侧边栏导航，其作用是将页面从当前文章或当前商品跳转到其他文章或其他商品页面上去。
 - ❑ 页内导航。页内导航的作用是在本页面几个主要的组成部分之间进行跳转。
 - ❑ 翻页操作。翻页操作是指在多个页面的前后页或博客网站的前后篇文章滚动。
- 除此之外，nav元素也可以用于其他所有你觉得是重要的、基本的导航链接组中。

请注意：在HTML 5中不要用menu元素代替nav元素。过去有很多Web应用程序的开发员喜欢用menu元素进行导航，我想有必要再次强调，menu元素是用在一系列发出命令的菜单上的，是一种交互性的元素，或者更确切地说是使用在Web应用程序中的。

3.1.4 aside元素

aside元素用来表示当前页面或文章的附属信息部分，它可以包含与当前页面或主要内容相关的引用、侧边栏、广告、导航条，以及其他类似的有别于主要内容的部分。

aside元素主要有以下两种使用方法。

1) 被包含在article元素中作为主要内容的附属信息部分，其中的内容可以是与当前文章有关的参考资料、名词解释，等等。这部分代码请看代码清单3-8。

代码清单3-8 文章内部的aside元素示例

```

<!DOCTYPE html>
<head>
<meta charset="utf-8">
<title>aside元素示例</title>
</head>
<body>

```

```

<header>
  <h1>F#入门</h1>
</header>
<article>
  <h1>第四节 词法闭包</h1>
  <p>lambda表达式可以创建词法闭包...(文章正文)</p>
  <aside>
    <!-- 因为这个aside元素被放置在一个article元素内部，
    所以分析器将这个aside元素的内容理解成是和article元素的内容相关联的。 -->
    <h1>名词解释</h1>
    <dl>
      <dt>F#</dt>
      <dd>F#为.Net2010中引入的新型函数型编程语言</dd>
    </dl>
    <dl>
      <dt>词法闭包</dt>
      <dd>词法闭包是指，将创建lambda表达式时的环境保存起来...(详细解释)</dd>
    </dl>
  </aside>
</article>
</body>

```

程序运行结果如图3-1所示。



图3-1 aside元素示例

这是笔者博客网页中的一篇文章，网页的标题放在了header元素中，在header元素的后面将所有关于文章的部分放在了一个article元素中，将文章的正文部分放在了一个p元素中，但是该文章还有一个名词解释的附属部分，用来解释该文章中的一些名词，因此，在p元素的下部又放置了一个aside元素，用来存放名词解释部分的内容。

2) 在article元素之外使用，作为页面或站点全局的附属信息部分。最典型的形式是侧边栏，其中的内容可以是友情链接，博客中其他文章列表、广告单元等。下面这个示例为标准博客网页中一个侧边栏的示例，示例中的“IT新技术”为博客的名称，如代码清单3-9所示。

代码清单3-9 侧边栏示例

```

<aside>
  <nav>
    <h2>评论</h2>
    <ul>
      <li>
        <a href="http://blog.sina.com.cn/1683">erway</a>      10-24 14:25
      </li>
      <li>
        <a href="http://blog.sina.com.cn/u/1345">太阳雨</a>10-22 23:48<br/>
        <a href="http://blog.sina.com.cn/s/blog_6a9kv8f.html#comment">
          顶，拜读一下老牛的文章
        </a>
      </li>
      <li>
        <a href="http://blog.sina.com.cn/u/1259295385">新浪官博</a>
        08-12 08:50<br/>
        <a href="#">恭喜！您已经成功开通了博客</a>
      </li>
    </ul>
  </nav>
</aside>

```

如果对这部分再加上CSS样式，在浏览器中的显示效果如图3-2所示。

该示例为一个典型的博客网站中的侧边栏部分，因此放在了aside元素中，但是该侧边栏又是具有导航作用的，因此放置在nav元素中，该侧边栏的标题是“评论”，放在了h2元素中，在标题之后使用了一个ul列表，用来存放具体的导航链接中。



图3-2 用aside元素实现的侧边栏示例

3.1.5 time元素与微格式

首先来说一下微格式，它是一种利用HTML的class属性来对网页添加附加信息的方法，附加信息例如新闻事件发生的日期和时间、个人电话号码、企业邮箱等。

微格式并不是在HTML 5之后才有的，在HTML 5之前它就和HTML结合使用了，但是在使用过程中发现在日期和时间的机器编码上出现了一些问题，编码过程中会产生一些歧义。HTML 5增加了一种新的元素来无歧义地、明确地对机器的日期和时间进行编码，并且以让人易读的方式来展现它。这个元素就是time元素。

time元素代表24小时中的某个时刻或某个日期，表示时刻时允许带时差。它可以定义很多格式的日期和时间，如下所示：

```

<time datetime="2010-11-13">2010年11月13日</time>
<time datetime="2010-11-13">11月13日</time>
<time datetime="2010-11-13">我的生日</time>
<time datetime="2010-11-13T20:00">我生日的晚上8点</time>
<time datetime="2010-11-13T20:00Z">我生日的晚上8点</time>
<time datetime="2010-11-13T20:00+09:00">我生日的晚上8点的美国时间</time>

```

编码时机器读到的部分在datetime属性里，而元素的开始标记与结束标记中间的部分是显示在网页上的。datetime属性中日期与时间之间要用“T”文字分隔，“T”表示时间。请注意倒数第二行，时间加上Z文字表示给机器编码时使用UTC标准时间，倒数第一行则加上了时差，表示向机器编码另一地区时间，如果是编码本地时间，则不需要添加时差。

3.1.6 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元素代表了文章（article元素的内容）或整个网页的发布日期，pubdate属性的具体使用方法如代码清单3-10所示。

代码清单3-10 pubdate与time结合使用

```
<article>
  <header>
    <h1>苹果</h1>
    <p>发布日期
      <time datetime="2010-10-29" pubdate>2010年10月29日</time>
    </p>
  </header>
  <p>苹果，植物类水果，多次花果... ("苹果"文章正文)</p>
  ...
</article>
```

你也许会疑惑为什么需要用到pubdate属性，为什么不能认为time元素就直接表示了文章或网页的发布日期呢？请看代码清单3-11。

代码清单3-11 pubdate与time结合使用

```
<article>
  <header>
    <h1>关于<time datetime=2010-10-29>10月29日</time>的舞会通知</h1>
    <p>发布日期:
      <time datetime=2010-10-11 pubdate>2010年10月11日</time>
    </p>
  </header>
  <p>大家好:我是法律系3年级学生代表,.....(关于舞会的通知)</p>
</article>
```

在这个例子中，有两个time元素，分别定义了两个日期——一个是舞会日期，另一个是通知发布日期。由于都使用了time元素，所以需要使用pubdate属性表明哪个time元素代表了通知的发布日期。

3.2 新增的非主体结构元素

除了以上几个主要的结构元素之外，HTML 5内还增加了一些表示逻辑结构或附加信息的非主体结构元素。下面分别来介绍。

3.2.1 header元素

header元素是一种具有引导和导航作用的结构元素，通常用来放置整个页面或页面内的一个内容区块的标题，但也可以包含其他内容，例如数据表格、搜索表单或相关的logo图片。

很明显，整个页面的标题应该放在页面的开头，我们可以用如下所示的形式书写页面的标题：

```
<header><h1>页面标题</h1></header>
```

需要强调的一点是：一个网页内并未限制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，如代码清单3-12所示。

代码清单3-12 多个header元素示例

```
<header>
  <h1>网页标题</h1>
</header>
<article>
  <header>
    <h1>文章标题</h1>
  </header>
  <p>文章正文</p>
</article>
```

在HTML 5中，一个header元素通常包括至少一个heading元素（h1-h6），也可以包括我们后面将要讨论的hgroup元素，还可以包括其他元素（譬如table或form），根据最新的W3C HTML 5标准，还可以包括nav元素。最后，让我们看一下博客网页中header元素的一个应用示例。示例中header元素处于页面顶部。详见代码清单3-13。

代码清单3-13 博客网页中header元素的示例

```
<header>
<hgroup>
<h1>IT新技术</h1>
<a href="http://blog.sina.com.cn/itnewtech">
http://blog.sina.com.cn/itnewtech
</a>
<a href="#">[订阅]</a>
<a href="#">[手机订阅]</a>
</hgroup>
<nav>
<ul>
<li>首页</li>
<li><a href="http://blog.sina.com.cn/articlelist1.html">博文目录</a></li>
<li><a href="http://photo.blog.sina.com.cn/itnewtech1">图片</a></li>
<li><a href="http://photo.blog.sina.com.cn/itnewtech">关于我</a></li>
</ul>
</nav>
</header>
```

如果对这段代码使用CSS样式，显示界面如图3-3所示。



图3-3 博客网页中header元素示例

3.2.2 hgroup元素

hgroup元素是将标题及其子标题进行分组的元素。hgroup元素通常会将h1~h6元素进行分组，譬如一个内容区块的标题及其子标题算一组。

通常，如果文章只有一个主标题，是不需要hgroup元素的，如代码清单3-14所示。

代码清单3-14 只有header元素的示例

```
<article>
  <header>
    <h1>文章标题</h1>
    <p><time datetime="2010-03-20">2010年10月29日</time></p>
  </header>
  <p>文章正文</p>
</article>
```

但是，如果文章有主标题，主标题下有子标题，就需要使用hgroup元素了，如代码清单3-15所示。

代码清单3-15 hgroup元素示例

```
<article>
  <header>
    <hgroup>
      <h1>文章主标题</h1>
      <h2>文章子标题</h2>
    </hgroup>
    <p><time datetime="2010-03-20">2010年10月29日</time></p>
  </header>
  <p>文章正文</p>
</article>
```

3.2.3 footer元素

footer元素可以作为其上层父级内容区块或是一个根区块的脚注。footer通常包括其相关区块的脚注信息，如作者、相关阅读链接及版权信息等。

在HTML 5出现之前，我们使用下面的方式编写页脚，如代码清单3-16所示。

代码清单3-16 HTML 5之前的页脚示例

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

```

    <ul>
      <li>版权信息</li>
      <li>站点地图</li>
      <li>联系方式</li>
    </ul>
  </div>

```

但是到了HTML 5之后,这种方式将不再使用,而是使用更加语义化的footer元素来替代,如代码清单3-17所示。

代码清单3-17 footer元素示例

```

<footer>
  <ul>
    <li>版权信息</li>
    <li>站点地图</li>
    <li>联系方式</li>
  </ul>
</footer>

```

与header元素一样,一个页面中也未限制footer元素的个数。同时,可以为article元素或section元素添加footer元素,请看下面两个示例。

代码清单3-18为一个在article元素中添加footer元素的示例。

代码清单3-18 在article元素中添加footer元素

```

<article>
  文章内容
  <footer>
    文章脚注
  </footer>
</article>

```

代码清单3-19为一个在section元素中添加footer元素的示例。

代码清单3-19 在section元素中添加footer元素

```

<section>
  分段内容
  <footer>
    分段内容的脚注
  </footer>
</section>

```

3.2.4 address元素

address元素用来在文档中呈现联系信息,包括文档作者或文档维护者的名字、他们的网站链接、电子邮箱、真实地址、电话号码等。address应该不只是用来呈现电子邮箱或真实地址,还应用来展示跟文档相关的联系人的所有联系信息。譬如,在代码清单3-20中,展示了一些博客中某篇文章评论者的名字及其在博客中的网址链接。

代码清单3-20 address元素示例

```
<address>
  <a href=http://blog.sina.com.cn/itnewtech>陆凌牛</a>
  <a href=http://blog.sina.com.cn/zhangyu>张玉</a>
  <a href=http://blog.sina.com.cn/baiquanli>白权立</a>
</address>
```

下面我们通过代码清单3-21来看如何把footer元素、time元素与address元素结合起来使用。

代码清单3-21 footer、time与address结合使用的例子

```
<footer>
  <div>
    <address>
      <a title="文章作者：陆凌牛" href="http://blog.sina.com.cn/itnewtech">
        陆凌牛</a>
      </address>
      发表于<time datetime="2010-10-04">2010年10月4日</time>
    </div>
  </footer>
```

在这个示例中，把博客文章的作者、博客的主页链接作为作者信息放在了address元素中，把文章发表日期放在了time元素中，把这个address元素与time元素中的总体内容作为脚注信息放在了footer元素中。

3.3 HTML 5结构

前面两节中详细介绍了在HTML 5中具体新增了哪些结构元素，以及这些元素的定义和使用方法。接下来，让我们看一下在HTML 5中进行总体页面布局的时候，具体应该怎样来综合运用这些结构元素。

3.3.1 大纲

通过使用新的结构元素，HTML 5的文档结构比大量使用div元素的HTML 4的文档结构清晰、明确了很多。如果再规划好文档结构的大纲，就可以创建出对于阅读者或屏幕阅读程序来说，都很清晰易读的文档结构。

所谓大纲，简单来说就是文档中各内容区块的结构编排。内容区块可以使用标题元素（h1~h6）来展示各级内容区块的标题。综合运用各级内容区块的标题创建好文档的目录后，该目录就是一个大纲了。

关于内容区块的编排，可以分为“显示编排”与“隐式编排”两种方式。

1. 显式编排内容区块

显式编排是指明确使用section等元素创建文档结构，在每个内容区块内使用标题（h1~h6、hgroup等），如代码清单3-22所示。

代码清单3-22 显式编排内容区块示例

```
<body>
  <h1>网页级内容区块标题</h1>
  <p>网页级内容区块的正文</p>
  <section>
    <h2>section级内容区块的标题</h2>
    <p>section级内容区块的正文</p>
  </section>
</body>
```

2. 隐式编排内容区块

所谓隐式编排，是指不明确使用section等元素，而是根据页面中所书写的各级标题（h1 ~ h6、hgroup）等把内容区块自动创建出来。因为HTML 5分析器只要看到书写了某个级别的标题，就会判断存在相对应的内容区块。代码清单3-23为一个隐式编排内容区块的示例。

代码清单3-23 隐式编排内容区块示例

```
<body>
  <h1>网页级内容区块标题</h1>
  <p>网页级内容区块的正文</p>
  <!-- 分析器根据h2等元素判断生成内容区块-->
  <h2>section级内容区块的标题</h2>
  <p>section级内容区块的正文</p>
</body>
```

将这两种编排方式进行对比，很明显，显式编排更加清晰、易读。

3. 标题分级

不同的标题有不同的级别，h1的级别最高，h6的级别最低。隐式编排时按如下规则自动生成内容区块：

- ☐ 如果新出现的标题比上一个标题级别低，生成下级内容区块。
- ☐ 如果新出现的标题比上一个标题级别高或级别相等，生成新的内容区块。

第一条规则的示例与前面一样，现在我们来看关于第二条规则的示例，如代码清单3-24所示。

代码清单3-24 第二条规则示例

```
<body>
<section>
  <h2>section级别的内容区块的标题</h2>
  <p>section级别的内容区块的正文</p>
  <!-- 因为下面的标题级别比上一个标题级别高，所以自动创建新的内容区块 -->
  <h1>新的section级别的内容区块的标题</h1>
  <p>新的section级别的内容区块的正文</p>
</section>
</body>
```

如果把上一个示例改成显式编排，如代码清单3-25所示。

代码清单3-25 第二条规则的显式编排示例

```
<body>
<section>
```

```

    <h2>section级别的内容区块的标题</h2>
    <p>section级别的内容区块的正文</p>
  </section>
  <section>
    <h1>新的section级别的内容区块的标题</h1>
    <p>新的section级别的内容区块的正文</p>
  </section>
</body>

```

因为隐式编排容易让自动生成的整个文档结构与想要的文档结构不一样，而且也容易引起文档结构的混乱，所以请尽量使用显式编排。

4. 不同的内容区块可以使用相同级别的标题

另外，不同的内容区块可以使用相同级别的标题。例如，父内容区块与子内容区块可以使用相同级别的标题h1。这样做的好处是：每个级别的标题都可以单独设计，如果既需要“整个网页的标题”，又需要“文章的标题”（譬如书写文档时），这样做将会带来很大的便利性，如代码清单3-26所示。

代码清单3-26 不同的内容区块可以使用相同级别的标题

```

<body>
<h1>网页的标题</h1>
<article>
  <header>
    <hgroup>
      <h1>文章标题</h1>
      <h2>文章子标题</h2>
    </hgroup>
    <p>文章正文</p>
  </header>
</article>
</body>

```

5. 网页编排示例

基于以上讲解过的知识点，让我们来看应该怎样编排网页的内容。代码清单3-27为一个标准博客网页的示例，在这个示例中，具备了一个标准博客网页所需具备的基本要素，只缺少了为了使用样式而补充添加的div元素。

代码清单3-27 网页编排示例

```

<!DOCTYPE html>
<head>
  <title>网页编排示例</title>
  <meta charset="UTF-8">
</head>
<body>
<!-- 网页标题 -->
<header>
  <h1>网页标题</h1>
  <!-- 网站导航链接 -->
  <nav>

```



```

        <ul>
            <li><a href="index.html">首页</a></li>
            <li><a href="help.html">帮助</a></li>
        </ul>
    </nav>
</header>
<!-- 文章正文 -->
<article>
    <hgroup>
        <h1>文章主标题</h1>
        <h2>文章子标题</h2>
    </hgroup>
    <p>文章正文</p>
    <!-- 文章评论 -->
    <section class="comments">
        <article>
            <h1>评论标题</h1>
            <p>评论正文</p>
        </article>
    </section>
</article>
<!-- 版权信息 -->
<footer>
    <small>版权所有：陆凌牛</small>
</footer>
</body>

```

在这个示例中，使用了嵌套article元素的方式，将关于评论的article元素嵌套在了主article元素中，在HTML 5中，推荐使用这种方式。

3.3.2 对新的结构元素使用样式

因为很多浏览器尚未对HTML 5中新增的结构元素提供支持，我们无法知道客户端使用的浏览器是否支持这些元素，所以需要使用CSS追加如下声明，目的是通知浏览器页面中使用的HTML 5中新增元素都是以块方式显示的，如下所示。

```

//追加block声明
article, aside, dialog, figure, footer, header, legend, nav, section { display: block; }
//正常使用样式
nav{float:left;width:20%;}
article{float:right;width:79%;}

```

另外，IE 8及之前的浏览器是不支持用CSS的方法来使用这些尚未支持的结构元素的，为了在Internet Explorer浏览器中也能正常使用这些结构元素，需要使用JavaScript脚本，如下所示：

```

//在脚本中创建元素
<script>
document.createElement("header");
document.createElement("nav");
document.createElement("article");
document.createElement("footer");
</script>

```

```
<style>
//正常使用样式
nav{float:left;width:20%;}
article{float:right;width:79%;}
</style>
```

尽管这段JavaScript脚本在其他浏览器中是不需要的，但它不会对这些浏览器造成什么不良影响。另外，到了IE 9之后，这段脚本就不需要了。

3.3.3 article元素的样式

一个网页中可能有多个独立的article元素，每一个article元素都允许有自己的标题与脚注等从属元素，并允许对自己的从属元素单独使用样式。譬如一个网页中的样式可能如下所示：

```
header{display:block;color:red;text-align:right;}
article header{color:blue;text-align:center;}
```

