



## 第2章 HTML 5与HTML 4的区别

- 2.1 语法的改变
- 2.2 新增的元素和废除的元素
- 2.3 新增的属性和废除的属性
- 2.4 全局属性

HTML 5以HTML 4为基础，对HTML 4进行了大量的修改。本章从总体上概要介绍到底HTML 5对HTML 4进行了哪些修改，HTML 5与HTML 4之间比较大的区别是什么。

学习内容：

- ❑ 掌握HTML 5与HTML 4在基本语法上有什么区别，这个基本语法区别包括DOCTYPE声明、内容类型（Content Type）字符编码的指定方法、元素标记的省略、具有布尔值的属性、引号的省略等几个方面。
- ❑ 了解在HTML 5中新增了哪些元素，删除了哪些HTML 4中的元素，为什么要删除这些元素，用什么元素或方法来取代这些被删除的元素。
- ❑ 了解在HTML 5中新增了哪些属性，删除了哪些HTML 4中的属性，在HTML 5中用什么方法来取代这些被删除的属性。
- ❑ 掌握什么是全局属性，掌握本章中介绍的几个常用全局属性，它们是contentEditable属性、designMode属性、hidden属性、spellcheck属性，以及tabindex属性。

## 2.1 语法的改变

### 2.1.1 HTML 5的语法变化

与HTML 4相比，HTML 5在语法上发生了很大的变化。可能有很多人会有疑问，“之前的HTML已经相当普及了！”，“如果改变基础语法，会产生什么影响？”等。

但是，HTML 5中的语法变化，与其他开发语言中的语法变化在根本意义上有所不同。它的变化，正是因为，在HTML 5之前几乎没有符合标准规范的Web浏览器！

HTML的语法是在SGML（Standard Generalized Markup Language）语言的基础上建立起来的。但是SGML语法非常复杂，要开发能够解析SGML语法的程序也很不容易，所以很多浏览器都不包含SGML的分析器。因此，虽然HTML基本上遵从SGML的语法，但是对于HTML的执行在各浏览器之间并没有一个统一的标准。

在这种情况下，各浏览器之间的互兼容性和互操作性在很大程度上取决于网站或网络应用程序的开发者在开发上所做的共同努力，而浏览器本身始终是存在缺陷的。

如上所述，在HTML 5中提高Web浏览器之间的兼容性是它的一个很大的目标，为了确保兼容性，就要有一个统一的标准。因此，在HTML 5中，就围绕着这个Web标准，重新定义了一套在现有的HTML的基础上修改而来的语法，使它运行在各浏览器时各浏览器都能够符合这个通用标准。

因为关于HTML 5语法解析的算法也都提供了详细的记载，所以各Web浏览器的供应商们可以把HTML 5分析器集中封装在自己的浏览器中。最新的Firefox（默认为4.0以后的版本）与WebKit浏览器引擎中都迅速地封装了供HTML 5使用的分析器，IE（Internet Explorer）与Opera也在努力加快对于HTML 5的支持——浏览器兼容性的提高指日可待。

接下来，让我们具体看一下在HTML 5中，到底对语法进行了哪些改变。

### 2.1.2 HTML 5中的标记方法

首先，让我们来看一下在HTML 5中的标记方法。

#### 1. 内容类型 (ContentType)

首先，HTML 5的文件扩展符与内容类型保持不变。也就是说，扩展符仍然为“.html”或“.htm”，内容类型 (ContentType) 仍然为“text/html”。

#### 2. DOCTYPE声明

DOCTYPE声明是HTML文件中必不可少的，它位于文件第一行。在HTML 4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HTML 5中，刻意不使用版本声明，一份文档将会适用于所有版本的HTML。HTML 5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，声明方法如下面的代码所示：

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

在HTML 5中像这样的DOCTYPE声明方式是允许的（不区分大小写,引号不区分是单引号还是双引号）。

#### 3. 指定字符编码

在HTML 4中，使用meta元素的形式指定文件中的字符编码，如下所示：

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

在HTML 5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，如下所示：

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

两种方法都有效，可以继续使用前面一种方式（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所示的标记方式，但在HTML 5中，这种字符编码方式将被认为是错误的，这一点请注意：

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

从HTML 5开始，对于文件的字符编码推荐使用UTF-8。

### 2.1.3 HTML 5确保了与之前HTML版本的兼容性

HTML 5的语法是为了保证与之前的HTML语法达到最大程度的兼容而设计的。例如，符合“没有<p>的结束标记”的HTML代码随处可见，HTML 5中并没有把这种情况作为错误来处理，而是允许存在这种情况，但明确地规定了这种情况应该怎么处理。

那么，针对这个问题，让我们从元素标记的省略、具有boolean值的属性、引号的省略这几方面来详细看一下在HTML 5中是如何确保与之前版本的HTML达到兼容的。

### 1. 可以省略标记的元素

在HTML 5中，元素的标记可以省略。具体来说，元素的标记分为“不允许写结束标记”、“可以省略结束标记”和“开始标记和结束标记全部可以省略”三种类型。让我们来针对这三类情况列举一个元素清单，其中包括HTML 5中的新元素（关于这些新元素，2.2节将介绍）。

- ❑ 不允许写结束标记的元素有：area、base、br、col、command、embed、hr、img、input、keygen、link、meta、param、source、track、wbr。
- ❑ 可以省略结束标记的元素有：li、dt、dd、p、rt、rp、optgroup、option、colgroup、thead、tbody、tfoot、tr、td、th。
- ❑ 可以省略全部标记的元素有：html、head、body、colgroup、tbody。

说明：“不允许写结束标记的元素”是指，不允许使用开始标记与结束标记将元素括起来的形式，只允许使用“<元素/>”的形式进行书写。例如“<br>...</br>”的书写方式是错误的，正确的书写方式为“<br/>”。当然，HTML 5之前的版本中<br>这种写法可以被沿用。

“可以省略全部标记的元素”是指，该元素可以完全被省略。请注意，即使标记被省略了，该元素还是以隐式的方式存在的。例如将body元素省略不写时，但它在文档结构中还是存在的，可以使用document.body进行访问。

### 2. 具有boolean值的属性

对于具有boolean值的属性，例如disabled与readonly等，当只写属性而不指定属性值时，表示属性值为true；如果想要将属性值设为false，可以不使用该属性。另外，要想将属性值设定为true时，也可以将属性名设定为属性值，或将空字符串设定为属性值。

属性值的设定方法可以参考下面的代码示例：

```
<!--只写属性不写属性值代表属性为true-->
<input type="checkbox" checked>
<!--不写属性代表属性为false-->
<input type="checkbox">
<!--属性值=属性名，代表属性为true-->
<input type="checkbox" checked="checked">
<!--属性值=空字符串，代表属性为true-->
<input type="checkbox" checked="">
```

### 3. 省略引号

大家已经知道，指定属性值的时候，属性值两边既可以用双引号，也可以用单引号。

HTML 5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。如下面的代码所示：

```
<!-- 请注意type的属性值两边的引号 -->
```

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

### 2.1.4 标记示例

现在，让我们通过前面学到的HTML 5的语法知识来看一个关于HTML 5标记的示例。

代码清单2-1完全是用HTML 5写成的，省略了<html>、<head>、<body>等元素。可以通过这个示例复习一下HTML 5的DOCTYPE声明、用<meta>元素的charset属性指定字符编码、<p>元素的结束标记的省略、使用<元素/>的方式来结束<meta>元素，以及<br>元素等本节中所介绍到的知识要点。

代码清单2-1 HTML 5标记示例

```
<!DOCTYPE html>
<meta charset="UTF-8">
<title>HTML 5标记示例</title>
<p>这段代码是根据HTML 5语法
<br />编写出来的。
```

这段代码在Firefox 4浏览器中的运行结果如图2-1所示，另外，本书中如果没有特别说明使用什么浏览器的时候，默认使用的都是Firefox 4浏览器。

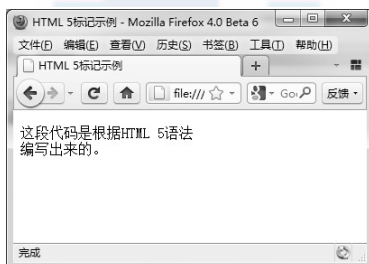


图2-1 HTML 5标记示例

## 2.2 新增的元素和废除的元素

本节将详细介绍HTML 5中新增和废除了哪些元素<sup>①</sup>。

### 2.2.1 新增的结构元素

在HTML 5中，新增了以下与结构相关的元素：

□ section元素

<sup>①</sup> 其他资料介绍的新增元素可能会比下面要介绍的多，这是因为HTML 5在最新发布的版本中，又把这些本来想新增的元素给删除了。

section元素表示页面中的一个内容区块，比如章节、页眉、页脚或页面中的其他部分。它可以与h1、h2、h3、h4、h5、h6等元素结合起来使用，标示文档结构。

HTML 5中代码示例：

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

HTML 4中代码示例：

```
<div>...</div>
```

#### ❑ article元素

article元素表示页面中的一块与上下文不相关的独立内容，譬如博客中的一篇文章或报纸中的一篇文章。

HTML 5中代码示例：

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

HTML 4中代码示例：

```
<div>...</div>
```

#### ❑ aside元素

aside元素表示article元素的内容之外的、与article元素的内容相关的辅助信息。

HTML 5中代码示例：

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

HTML 4中代码示例：

```
<div>...</div>
```

#### ❑ header元素

header元素表示页面中一个内容区块或整个页面的标题。

HTML 5中代码示例：

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

HTML 4中代码示例：

```
<div>...</div>
```

#### ❑ hgroup元素

hgroup元素用于对整个页面或页面中一个内容区块的标题进行组合。

HTML 5中代码示例：

```
<hgroup>...</hgroup>
```

HTML 4中代码示例：

```
<div>...</div>
```

#### ❑ footer元素

footer元素表示整个页面或页面中一个内容区块的脚注。一般来说，它会包含创作者的姓名、创作日期以及创作者联系信息。

HTML 5中代码示例：

```
</footer></footer>
```

HTML 4中代码示例：

```
<div>...</div>
```

#### ❑ nav元素

nav元素表示页面中导航链接的部分。

HTML 5中代码示例：

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

HTML 4中代码示例：

```
<ul></ul>
```

#### ❑ figure元素

figure元素表示一段独立的流内容，一般表示文档主体流内容中的一个独立单元。使用figcaption元素为figure元素组添加标题。

HTML 5中代码示例：

```
<figure>
<figcaption>PRC</figcaption>
<p>The People's Republic of China was born in 1949...</p>
</figure>
```

HTML 4中代码示例：

```
<dl>
<h1>PRC</h1>
<p>The People's Republic of China was born in 1949...</p>
</dl>
```

## 2.2.2 新增的其他元素

除了结构元素外，在HTML 5中，还新增了以下元素：

#### ❑ video元素

video元素定义视频，比如电影片段或其他视频流。

HTML 5中代码示例：

```
<video src="movie.ogg" controls="controls">video元素</video>
```

HTML 4中代码示例：

```
<object type="video/ogg" data="movie.ogv">
  <param name="src" value="movie.ogv">
</object>
```

#### ❑ audio元素

audio元素定义音频，比如音乐或其他音频流。

HTML 5中代码示例：

```
<audio src="someaudio.wav">audio元素</audio>
```

HTML 4中代码示例：

```
<object type="application/ogg" data="someaudio.wav">
<param name="src" value="someaudio.wav">
</object>
```

#### ❑ embed元素

embed元素用来插入各种多媒体，格式可以是 Midi、Wav、AIFF、AU、MP3等。

HTML 5中代码示例：

```
<embed src="horse.wav" />
```

HTML 4中代码示例：

```
<object data="flash.swf" type="application/x-shockwave-flash"></object>
```

#### ❑ mark元素

mark元素主要用来在视觉上向用户呈现那些需要突出显示或高亮显示的文字。mark元素的一个比较典型的应用就是在搜索结果中向用户高亮显示搜索关键词。

HTML 5中代码示例：

```
<mark></mark>
```

HTML 4中代码示例：

```
<span></span>
```

#### ❑ progress元素

progress元素表示运行中的进程，可以使用 progress元素来显示 JavaScript 中耗费时间的函数的进程。

HTML 5中代码示例：

```
<meter></meter>
```

这是HTML 5中新增功能，故无法用HTML 4代码来实现。

#### ❑ time元素

time元素表示日期或时间，也可以同时表示两者。

HTML 5中代码示例：

```
<time></time>
```

HTML 4中代码示例：

```
<span></span>
```

#### ❑ ruby元素

ruby元素表示ruby注释（中文注音或字符）。

HTML 5中代码示例：

```
<ruby>汉 <rt><rp>(</rp>ㄏㄢˋ<rp>)</rt></ruby>
```

这也是HTML 5中新增的功能。

#### ❑ rt元素

rt元素表示字符（中文注音或字符）的解释或发音。



HTML 5中代码示例：

```
<ruby>汉 <rt> ㄣ'</rt></ruby>
```

这是HTML 5中新增的功能。

#### □ rp元素

rp元素在ruby注释中使用，以定义不支持ruby元素的浏览器所显示的内容。

HTML 5中代码示例：

```
<ruby>汉 <rt><rp>(</rp>ㄣ'<rp>)</rp></rt></ruby>
```

这是HTML 5中新增的功能。

#### □ wbr元素

wbr元素表示软换行。wbr元素与br元素的区别是：br元素表示此处必须换行；而wbr元素的意思是浏览器窗口或父级元素的宽度足够宽时（没必要换行时），不进行换行，而当宽度不够时，主动在此处进行换行。wbr元素好像对字符型的语言作用挺大，但是对于中文，貌似没多大用处。

HTML 5中代码示例：

```
<p> To learn AJAX, you must be fami<wbr>liar with the XMLHttpRequest Object. </p>
```

这是HTML 5中新增的功能。

#### □ canvas元素

canvas元素表示图形，比如图表和其他图像。这个元素本身没有行为，仅提供一块画布，但它把一个绘图API展现给客户端JavaScript，以使脚本能够把想绘制的东西绘制到这块画布上。

HTML 5中代码示例：

```
<canvas id="myCanvas" width="200" height="200"></canvas>
```

HTML 4中代码示例：

```
<object data="inc/hdr.svg" type="image/svg+xml" width="200" height="200">
</object>
```

#### □ command元素

command元素表示命令按钮，比如单选按钮、复选框或按钮。

HTML 5中代码示例：

```
<command onclick=cut()" label="cut">
```

这是HTML 5中新增的功能。

#### □ details元素

details元素表示用户要求得到并且可以得到的细节信息。它可以与summary元素配合使用。summary元素提供标题或图例。标题是可见的，用户点击标题时，会显示出细节信息。summary元素应该是details元素的第一个子元素。

HTML 5中代码示例：

```
<details>
  <summary>HTML 5</summary>
  This document teaches you everything you have to learn about HTML 5.
</details>
```

这是HTML 5中新增的功能。

#### ❑ datalist元素

datalist元素表示可选数据的列表，与input元素配合使用，可以制作出输入值的下拉列表。

HTML 5中代码示例：

```
<datalist></datalist>
```

这是HTML 5中新增的功能。

#### ❑ datagrid元素

datagrid元素表示可选数据的列表，它以树形列表的形式来显示。

HTML 5中代码示例：

```
<datagrid></datagrid>
```

这是HTML 5中新增的功能。

#### ❑ keygen元素

keygen元素表示生成密钥。

HTML 5中代码示例：

```
<keygen>
```

这是HTML 5中新增的功能。

#### ❑ output元素

output元素表示不同类型的输出，比如脚本的输出。

HTML 5中代码示例：

```
<output></output>
```

HTML 4中代码示例：

```
<span></span>
```

#### ❑ source元素

source元素为媒介元素（比如<video>和<audio>）定义媒介资源。

HTML 5中代码示例：

```
<source>
```

HTML 4中代码示例：

```
<param>
```

#### ❑ menu元素

menu元素表示菜单列表。当希望列出表单控件时使用该标签。

HTML 5中代码示例：

```
<menu>
  <li><input type="checkbox" />Red</li>
  <li><input type="checkbox" />blue</li>
</menu>
```

在HTML 4中，menu元素不被推荐使用。

### 2.2.3 新增的input元素的类型

HTML 5中新增了很多input元素的类型，现列举如下：

☐ email

email类型表示必须输入E-mail地址的文本输入框。

☐ url

url类型表示必须输入URL地址的文本输入框。

☐ number

number类型表示必须输入数值的文本输入框。

☐ range

range类型表示必须输入一定范围内数字值的文本输入框。

☐ Date Pickers

HTML 5拥有多个可供选取日期和时间的新型输入文本框：

date——选取日、月、年

month——选取月、年

week——选取周和年

time——选取时间（小时和分钟）

datetime——选取时间、日、月、年（UTC时间）

datetime-local——选取时间、日、月、年（本地时间）

### 2.2.4 废除的元素

由于各种原因，在HTML 5中废除了很多元素，简单介绍如下。

#### 1. 能使用CSS替代的元素

对于basefont、big、center、font、s、strike、tt、u这些元素，由于它们的功能都是纯粹为画面展示服务的，而HTML 5中提倡把画面展示性功能放在CSS样式表中统一编辑，所以将这些元素废除了，并使用编辑CSS、添加CSS样式表的方式进行替代。其中font元素允许由“所见即所得”的编辑器来插入，s元素、strike元素可以由del元素替代，tt元素可以由CSS的font-family属性替代。

#### 2. 不再使用frame框架

对于frameset元素、frame元素与noframes元素，由于frame框架对网页可用性存在负面影响，在HTML 5中已不支持frame框架，只支持iframe框架，或者用服务器方创建的由多个页

面组成的复合页面的形式，同时将以上这三个元素废除。

### 3. 只有部分浏览器支持的元素

对于applet、bgsound、blink、marquee等元素，由于只有部分浏览器支持这些元素，特别是bgsound元素以及marquee元素，只被Internet Explorer所支持，所以在HTML 5中被废除。其中applet元素可由embed元素或object元素替代，bgsound元素可由audio元素替代，marquee可以由JavaScript编程的方式所替代。

### 4. 其他被废除的元素

其他被废除元素还有：

- ☐ 废除rb元素，使用ruby元素替代
- ☐ 废除acronym元素，使用abbr元素替代
- ☐ 废除dir元素，使用ul元素替代
- ☐ 废除isindex元素，使用form元素与input元素相结合的方式替代
- ☐ 废除listing元素，使用pre元素替代
- ☐ 废除xmp元素，使用code元素替代
- ☐ 废除nextid元素，使用GUIDS替代
- ☐ 废除plaintext元素，使用“text/plain” MIME类型替代

## 2.3 新增的属性和废除的属性

在HTML 5中，在增加和废除了很多元素的同时，也增加和废除了很多属性，本节对于这些增加和废除的属性进行简单介绍<sup>⊖</sup>。

### 2.3.1 新增的属性

#### 1. 表单相关的属性

新增的与表单相关的元素如下：

- ☐ 可以对input ( type=text )、select、textarea与button元素指定autofocus属性。它以指定属性的方式让元素在画面打开时自动获得焦点。
- ☐ 可以对input元素 ( type=text ) 与textarea元素指定placeholder属性，它会对用户的输入进行提示，提示用户可以输入的内容。
- ☐ 可以对input、output、select、textarea、button与fieldset指定form属性，声明它属于哪个表单，然后将其放置在页面上任何位置，而不是表单之内。
- ☐ 可以对input元素 ( type=text ) 与textarea元素指定required属性。该属性表示在用户提交的时候进行检查，检查该元素内一定要有输入内容。
- ☐ 为input元素增加了几个新的属性：autocomplete、min、max、multiple、pattern与step。

⊖ 其他资料介绍的新增属性可能会比下面要介绍的多，这是因为HTML 5在最新发布的版本中，又把这些本来想新增的属性给删除了。

同时还有一个新的list元素与datalist元素配合使用。datalist元素与autocomlete属性配合使用。multiple属性允许在上传文件时一次上传多个文件。

- ❑ 为input元素与button元素增加了新属性formaction、formenctype、formmethod、formnovalidate与formtarget，他们可以重载form元素的action、enctype、method、novalidate与target属性。为fieldset元素增加了disabled属性，可以把它的子元素设为disabled（无效）状态。
- ❑ 为input元素、button元素、form元素增加了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。

## 2. 链接相关属性

新增的与链接相关的属性如下：

- ❑ 为a与area元素增加了media属性，该属性规定目标URL是为什么类型的媒介/设备进行优化的，只能在href属性存在时使用。
- ❑ 为area元素增加了hreflang属性与rel属性，以保持与a元素、link元素的一致。
- ❑ 为link元素增加了新属性sizes。该属性可以与icon元素结合使用（通过rel属性），该属性指定关联图标（icon元素）的大小。
- ❑ 为base元素增加了target属性，主要目的是保持与a元素的一致性。

## 3. 其他属性

除了上面介绍的与表单和链接相关的属性外，HTML 5还增加了下面的属性：

- ❑ 为ol元素增加属性reversed，它指定列表倒序显示。
- ❑ 为meta元素增加charset属性，因为这个属性已经被广泛支持了，而且为文档的字符编码的指定提供了一种比较好的方式。
- ❑ 为menu元素增加了两个新的属性——type与label。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、工具条与列表菜单的三种形式出现。
- ❑ 为style元素增加scoped属性，用来规定样式的作用范围，譬如只对页面上某个树起作用。
- ❑ 为script元素增加async属性，它定义脚本是否异步执行。
- ❑ 为html元素增加属性manifest，开发离线Web应用程序时它与API结合使用，定义一个URL，在这个URL上描述文档的缓存信息。
- ❑ 为iframe元素增加三个属性sandbox、seamless与srcdoc，用来提高页面安全性，防止不信任的Web页面执行某些操作。

### 2.3.2 废除的属性

HTML 4中的一些属性在HTML 5中不再被使用，而是采用其他属性或其他方案进行替代，具体如表2-1所示。

表2-1 在HTML 5中被废除了的属性

在HTML 4中使用的属性	使用该属性的元素	在HTML 5中的替代方案
rev	link、a	rel
charset	link、a	在被链接的资源的中使用HTTP Content-type头元素
shape、coords	a	使用area元素代替a元素
longdesc	img、iframe	使用a元素链接到较长描述
target	link	多余属性，被省略
nohref	area	多余属性，被省略
profile	head	多余属性，被省略
version	html	多余属性，被省略
name	img	id
scheme	meta	只为某个表单域使用scheme
archive、classid、codebase、codetype、declare、standby	object	使用data与type属性类调用插件。需要使用这些属性来设置参数时，使用param属性
valuetype、type	param	使用name与value属性，不声明值的MIME类型
axis、abbr	td、th	使用以明确简洁的文字开头、后跟详述文字的形式。可以对更详细内容使用title属性，来使单元格的内容变得简短
scope	td	在被链接的资源的中使用HTTP Content-type头元素
align	caption、input legend、div、h1、h2、h3、h4、h5、h6、p	使用CSS样式表替代
alink、link、text、vlink、background、bgcolor	body	使用CSS样式表替代
align、bgcolor、border、cellpadding、cellspacing、frame、rules、width	table	使用CSS样式表替代
align、char、charoff、height、nowrap、valign	tbody、thead、tfoot	使用CSS样式表替代
align、bgcolor、char、charoff、height、nowrap、valign、width	td、th	使用CSS样式表替代
align、bgcolor、char、charoff、valign	tr	使用CSS样式表替代
align、char、charoff、valign、width	col、colgroup	使用CSS样式表替代
align、border、hspace、vspace	object	使用CSS样式表替代
clear	br	使用CSS样式表替代

(续)

在HTML 4中使用的属性	使用该属性的元素	在HTML 5中的替代方案
compact、type	ol、ul、li	使用CSS样式表替代
compact	dl	使用CSS样式表替代
compact	menu	使用CSS样式表替代
width	pre	使用CSS样式表替代
align、hspace、vspace	img	使用CSS样式表替代
align、noshade、size、width	hr	使用CSS样式表替代
align、frameborder、scrolling、marginheight、marginwidth	iframe	使用CSS样式表替代
autosubmit	menu	

## 2.4 全局属性

在HTML 5中，新增了一个“全局属性”的概念。所谓全局属性，是指可以对任何元素都使用的属性，本节将详细介绍几种常用的全局属性。

### 2.4.1 contentEditable属性

contentEditable是由微软开发、被其他浏览器反编译并投入应用的一个全局属性。该属性的主要功能是允许用户编辑元素中的内容，所以该元素必须是可以获得鼠标焦点的元素，而且在点击鼠标后要向用户提供一个插入符号，提示用户该元素中的内容允许编辑。contentEditable属性是一个布尔值属性，可以被指定为true或false。

除此之外，该属性还有个隐藏的inherit（继承）状态，属性为true时，元素被指定为允许编辑；属性为false时，元素被指定为不允许编辑；未指定true或false时，则由inherit状态来决定，如果元素的父元素是可编辑的，则该元素就是可编辑的。

另外，除了contentEditable属性外，元素还具有一个isContentEditable属性，当元素可编辑时，该属性为true；当元素不可编辑时，该属性为false。

代码清单2-2中给出了一个使用contentEditable属性的示例，当列表元素被加上contentEditable属性后，该元素就变成可编辑的了，读者可自行在浏览器中对该示例进行试验。

代码清单2-2 contentEditable属性示例

```
<!DOCTYPE html>
<head>
<meta charset="UTF-8">
<title>contentEditable属性示例</title>
</head>
<h2>可编辑列表</h2>
<ul contentEditable="true">
```

```
<li>列表元素1</li>
<li>列表元素2</li>
<li>列表元素3</li>
</ul>
```

这段代码运行后的结果如图2-2所示。



图2-2 可编辑列表示例

在编辑完元素中的内容后，如果想要保存其中内容，只能把该元素的innerHTML发送到服务器端进行保存，因为改变元素内容后该元素的innerHTML内容也会随之改变，目前还没有特别的API来保存编辑后元素中的内容。

最后，在这里列举一下支持contentEditable属性的元素：defaults、A、ABBR、ACRONYM、ADDRESS、B、BDO、BIG、BLOCKQUOTE、BODY、BUTTON、CENTER、CITE、CODE、CUSTOM、DD、DEL、DFN、DIR、DIV、DL、DT、EM、FIELDSET、FONT、FORM、hn、I、INPUT type=button、INPUT type=password、INPUT type=radio、INPUT type=reset、INPUT type=submit、INPUT type=text、INS、ISINDEX、KBD、LABEL。

## 2.4.2 designMode属性

designMode属性用来指定整个页面是否可编辑，当页面可编辑时，页面中任何支持上文所述的contentEditable属性的元素都变成了可编辑状态。designMode属性只能在JavaScript脚本里被编辑修改。该属性有两个值——“on”与“off”。属性被指定为“on”时，页面可编辑；被指定为“off”时，页面不可编辑。使用JavaScript脚本来指定designMode属性的方法如下所示：

```
document.designMode="on"
```

针对designMode属性，各浏览器的支持情况也各不相同：

- ❑ IE8：出于安全考虑，不允许使用designMode属性让页面进入编辑状态。
- ❑ IE9：允许使用designMode属性让页面进入编辑状态。



- ❑ Chrome 3和Safari：使用内嵌frame的方式,该内嵌frame是可编辑的。
- ❑ Firefox和Opera：允许使用designMode属性让页面进入编辑状态。

### 2.4.3 hidden属性

在HTML 5中，所有的元素都允许使用一个hidden属性。该属性类似于input元素中的hidden元素，功能是通知浏览器不渲染该元素，使该元素处于不可见状态。但是元素中的内容还是浏览器创建的，也就是说页面装载后允许使用JavaScript脚本将该属性取消，取消后该元素变为可见状态，同时元素中的内容也即时显示出来。Hidden属性是一个布尔值的属性，当设为true时，元素处于不可见状态；当设为false时，元素处于可见状态。

### 2.4.4 spellcheck属性

spellcheck属性是HTML 5针对input元素（type=text）与textarea这两个文本输入框提供的一个新属性，它的功能为对用户输入的文本内容进行拼写和语法检查。spellcheck属性是一个布尔值的属性，具有true或false两种值。但是它在书写时有一个特殊的地方，就是必须明确声明属性值为true或false，书写方法如下所示：

```
<!--以下两种书写方法正确-->
<textarea spellcheck="true" >
<input type=text spellcheck=false>
<!--以下书写方法为错误-->
<textarea spellcheck >
```

需要注意的是，如果元素的readOnly属性或disabled属性设为true，则不执行拼写检查。

目前除了IE之外，Firefox、Chrome、Safari、Opera等浏览器都对该属性提供了支持。

图2-3为Opera浏览器中spellcheck属性的表现形式。



图2-3 Opera浏览器中spellcheck的属性示例

### 2.4.5 tabindex属性

tabindex是开发中的一个基本概念，当不断敲击Tab键让窗口或页面中的控件获得焦点，

对窗口或页面中的所有控件进行遍历的时候，每一个控件的`tabindex`表示该控件是第几个被访问到的。

过去这个属性在编辑网页时是非常有用的，但如今控件的遍历顺序是由元素在页面上所处的位置决定的，所以就不再需要了。

但是`tabindex`还有另外一个作用，在默认情况下，只有链接元素与表单元素可以通过按键获得焦点。如果对其他元素使用`tabindex`属性后，也能让该元素获得焦点，那么当脚本中执行`focus()`语句的时候，就可以让该元素获得焦点了。但这样做会产生一个副作用：该元素也可以通过按Tab键获得焦点，而这有时可能不是开发者想要的结果。

把元素的`tabindex`值设为负数（通常为-1）后就解决这个问题了。`tabindex`的值为负数后，仍然可以通过编程的方式让元素获得焦点，但按下Tab键时该元素就不能获得焦点了，这在复杂的页面中或复杂的Web应用程序中是十分有用的。在HTML 4中，-1是一个无用的属性值，但到了HTML 5中，通过巧妙地运用让该属性值得到了极大的应用。

