

目 录

HTML5 介绍 5

教程前言 14

HTML 5 简介 15

HTML 5 视频 17

HTML 5 音频 21

HTML 5 Canvas 25

HTML 5 Web 存储 36

HTML 5 Input 类型 41

HTML 5 表单元素 51

HTML 5 表单属性 56

HTML 5 参考手册 70

HTML 5 标准属性 79

HTML 5 事件属性 79

HTML 5 标签 85

 <!--...--> 标签 85

 <!DOCTYPE> 标签 86

 <abbr> 标签 91

 <acronym> 标签 92

 <address> 标签 93

 <applet> 标签 95

 <area> 标签 95

 <article> 标签 98

 <aside> 标签 100

 <audio> 标签 100

 标签 101

 <base> 标签 102

 <basefont> 标签 103

 <bdo> 标签 104

 <blockquote> 标签 105

 <body> 标签 105

 标签 106

 <button> 标签 107

 <canvas> 标签 109

 <caption> 标签 111

 <center> 标签 113

 <dfn> <code> <samp> <kbd> <var> <cite> 标签 113

 <col> 标签 114

 <colgroup> 标签 117

 <command> 标签 119

 <datalist> 标签 120

 <dd> 标签 121

 标签 123

<details> 标签.....	124
<dir> 标签.....	125
<div> 标签.....	125
<dl> 标签.....	126
<dt> 标签.....	128
<embed> 标签.....	130
<fieldset> 标签.....	131
<figcaption> 标签.....	133
<figure> 标签.....	134
 标签.....	135
<footer> 标签.....	135
<form> 标签.....	137
<frame> 标签.....	139
<frameset> 标签.....	139
<h1> 至 <h6> 标签.....	140
<head> 标签.....	141
<header> 标签.....	143
<hgroup> 标签.....	144
<hr> 标签.....	146
<html> 标签.....	147
<i> 标签.....	149
<iframe> 标签.....	150
 标签.....	151
<input> 标签.....	155
<ins> 标签.....	162
<keygen> 标签.....	164
 <dfn> <code> <samp> <kbd> <var> <cite> 标签.....	165
<label> 标签.....	168
<legend> 标签.....	170
标签.....	172
<link> 标签.....	177
<map> 标签.....	179
<mark> 标签.....	181
<menu> 标签.....	182
<meta> 标签.....	183
<meter> 标签.....	188
<nav> 标签.....	190
<noframes> 标签.....	191
<object> 标签.....	196
标签.....	197
<optgroup> 标签.....	200
<option> 标签.....	201
<output> 标签.....	204
<p> 标签.....	205

<param> 标签	209
<pre> 标签	210
<progress> 标签	212
<q> 标签	213
<rp> 标签	214
<rt> 标签	216
<ruby> 标签	217
<s> 标签	218
<script> 标签	219
<section> 标签	221
<select> 标签	222
<small> 标签	225
<source> 标签	225
 标签	226
<strike> 标签	227
<style> 标签	227
<sub> 和 <sup> 标签	230
<summary> 标签	231
<table> 标签	232
<tbody> 标签	242
<td> 标签	244
<textarea> 标签	246
<tfoot> 标签	248
<th> 标签	250
<thead> 标签	253
<time> 标签	254
<title> 标签	256
<tr> 标签	257
<tt> 标签	258
<u> 标签	258
标签	259
<video> 标签	260

声明：相关测试代码仅用于学习参考。笔者不保证内容的正确性。通过使用本站内容随之而来的风险与笔者无关。当使用资料时，代表您已接受了本站的使用条款和隐私条款。本资料的所有内容仅供测试，对任何法律问题及风险不承担任何责任。

HTML5 介绍

HTML5 草案的前身名为 Web Applications 1.0。於 2004 年被 WHATWG 提出，於 2007 年被 W3C 接纳，并成立了新的 HTML 工作团队。在 2008 年 1 月 22 日，第一份正式草案已公布,预计将在 2010 年 9 月正式向公众推荐。WHATWG 表示该规范是目前正在进行的工作，仍须多年的努力。

新标记

HTML5 提供了一些新的元素和属性，例如<nav>（网站导航块）和<footer>。这种标签将有利于搜索引擎的索引整理，同时更好的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>标记。

一些过时的 HTML4 标记将被取消。其中包括纯粹显示效果的标记，如和<center>，它们已经被 CSS 取代。

<video>标记

定义和用法：

<video> 标签定义视频，比如电影片段或其他视频流。

实例：

一段简单的 HTML5 视频

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

您的浏览器不支持 video 标签。

```
</video>
```

各系统与浏览器支持的视频格式：

```
<th>HTML5 </th>
```

操作系统	浏览器	支持度	
Windows	Internet Explorer 6	不支持	如果安装了 Google Chrome Frame，支持 HTML5
	Internet Explorer 7	不支持	如果安装了 Google Chrome Frame，支持 HTML5
	Internet Explorer 8	不支持	如果安装了 Google Chrome Frame，支持 HTML5
	Internet Explorer 9	支持(mp4,webm*)	*如果安装了 VP8 解码器
	Firefox < 3.5	不支持	

	Firefox 3.5, 3.6	支持 (ogg)	
	Firefox 4	支持 (webm,ogg)	
	Chrome < 3	不支持	
	Chrome 3, 4, 5	支持 (mp4,ogg)	
	Chrome 6	支持 (mp4,webm,ogg)	
	Opera < 10.5	不支持	
	Opera 10.5	支持 (ogg)	
	Opera 10.6	支持 (webm,ogg)	
	Opera 11	支持 (webm,ogg)	
	Safari 3.1, 4, 5	支持 (mp4)	
Mac	Safari < 3.1	不支持	
	Safari 3.1, 4, 5	支持 (mp4)	
	Firefox < 3.5	不支持	
	Firefox 3.5, 3.6	支持 (ogg)	
	Firefox 4	支持 (webm,ogg)	
	Chrome < 3	不支持	
	Chrome 3, 4, 5	支持 (mp4,ogg)	
	Chrome 6	支持 (mp4,webm,ogg)	
	Opera < 10.5	不支持	
	Opera 10.5	支持 (ogg)	
	Opera 10.6	支持 (webm,ogg)	
	Opera 11	支持 (webm,ogg)	
Linux	Firefox < 3.5	不支持	
	Firefox 3.5, 3.6	支持 (ogg)	
	Firefox 4	支持 (webm,ogg)	

	Chrome < 3	不支持	
	Chrome 3, 4, 5	支持 (mp4,ogg)	
	Chrom	支持 (mp4,webm,ogg)	
	Opera < 10.5	不支持	
	Opera 10.5	支持 (ogg)	
	Opera 10.6	支持 (webm,ogg)	
	Opera 11	支持 (webm,ogg)	
	Konqueror < 4.4	不支持	
	Konqueror 4.4+	支持 (ogg)	
iOS	(iPhone,iOS 3,4) Safari	支持 (mp4)	早期版本 (iOS 1,2) 不支持 HTML5 视频
	(iPad, iOS 3.2) Safari	支持 (mp4)	
Android	Android 2.1, 2.2	支持 (mp4)	
	Android 2.3	支持 (mp4)	

ogg = 带有 Thedora 视频编码和 Vorbis 音频编码的 Ogg 文件。
mp4 = 带有 H.264 视频编码和 AAC 音频编码的 MPEG 4 文件。

<audio> 标记

定义和用法
<audio> 标签定义声音，比如音乐或其他音频流。
实例：
一段简单的 HTML 5 音频
<audio src="someaudio.wav">
您的浏览器不支持 audio 标签。
</audio>
音频格式：
当前，audio 元素支持三种音频格式：

	IE 8	Firefox 3.5	Opera 10.5	Chrome 3.0	Safari 3.0
Ogg Vorbis		√	√	√	

MP4				√	√
Wav		√	√		√

<canvas> 标记

定义和用法：

<canvas> 标签定义图形，比如图表和其他图像。

实例：

如何通过 canvas 元素来显示一个红色的矩形：

```
<canvas id="myCanvas"></canvas>
```

```
<script type="text/javascript">
```

```
var canvas=document.getElementById('myCanvas');
```

```
var ctx=canvas.getContext('2d');
```

```
ctx.fillStyle='#FF0000';
```

```
ctx.fillRect(0,0,80,100);
```

```
</script>
```

什么是 Canvas？

HTML5 的 canvas 元素使用 JavaScript 在网页上绘制图像。

画布是一个矩形区域，您可以控制其每一像素。

canvas 拥有多种绘制路径、矩形、圆形、字符以及添加图像的方法。

编辑本段 HTML5: 标签的改变

HTML5 吸取了 XHTML 2 一些建议，包括一些用来改善文档结构的功能，比如，新的 HTML 标签 header, footer, dialog, aside, figure 等的使用，将使内容创作者更加语义地创建文档，之前的开发者在这些场合是一律使用 div 的。

HTML5 还包含了一些将内容和展示分离的努力，开发者们也许会惊讶，b 和 i 标签依然存在，但它们的意义已经和之前有所不同，这些标签的意义只是为了将一段文字标识出来，而不是为了为它们设置粗体或斜体式样。u, font, center, strike 这些标签则被完全去掉了。

新标准适用了一些全新的表单输入对象，包括日期，URL，Email 地址，其它的对象则增加了对非拉丁字符的支持。HTML5 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内容创建者可以创建更干净，更容易管理的网页，这样的网页对搜索引擎，对读屏软件等更为友好。

编辑本段新应用程序接口（API）

除了原先的 DOM 接口，HTML5 增加了更多 API,如：

1. 用于即时 2D 绘图的 Canvas 标签
2. 定时媒体回放
3. 离线数据库存储
4. 文档编辑

5. 拖拽控制

6. 浏览历史管理

编辑本段与 HTML 4 的不同之处

新的解析顺序新的元素：section, video, progress, nav, meter, time, aside, canvasinput 元素的新属性：日期和时间，email, url。新的通用属性：ping, charset, async 全域属性：id, tabindex, repeat。移除元素：center, font, strike。

HTML 5 有两大特点：

首先，强化了 Web 网页的表现性能。除了可描绘二维图形外，还准备了用于播放视频和音频的标签。

其次，追加了本地数据库等 Web 应用的功能。

编辑本段异常处理

HTML5(text/html)浏览器将在错误语法的处理上更加灵活。HTML5 在设计时保证旧的浏览器能够安全的忽略掉新的 HTML5 代码。与 HTML4.01 相比，HTML5 给出了解析的详细规则，力图让不同的浏览器即使在发生语法错误时也能返回相同的结果。

编辑本段 html5 标签

按字母顺序排列的标签列表

?4: 指示在 HTML 4.01 中定义了该元素

?5: 指示在 HTML 5 中定义了该元素

标签 描述 4 5

<!--...--> 定义注释。 4 5

<!DOCTYPE> 定义文档类型。 4 5

<a> 定义超链接。 4 5

<abbr> 定义缩写。 4 5

<acronym> HTML 5 中不支持。定义首字母缩写。 4

<address> 定义地址元素。 4 5

<applet> HTML 5 中不支持。定义 applet。 4

<area> 定义图像映射中的区域。 4 5

<article> 定义 article。 5

<aside> 定义页面内容之外的内容。 5

<audio> 定义声音内容。 5

 定义粗体文本。 4 5

<base> 定义页面中所有链接的基准 URL。 4 5

<basefont> HTML 5 中不支持。请使用 CSS 代替。 4

<bdo> 定义文本显示的方向。 4 5

<big> HTML 5 中不支持。定义大号文本。 4

<blockquote> 定义长的引用。 4 5

<body> 定义 body 元素。 4 5

 插入换行符。 4 5

<button> 定义按钮。 4 5

<canvas> 定义图形。 5

<caption> 定义表格标题。 4 5

<center> HTML 5 中不支持。定义居中的文本。 4

<cite> 定义引用。 4 5

<code> 定义计算机代码文本。 4 5

<col> 定义表格列的属性。 4 5

<colgroup> 定义表格列的分组。 4 5

<command> 定义命令按钮。 5

<datalist> 定义下拉列表。 5

<dd> 定义定义的描述。 4 5

 定义删除文本。 4 5

<details> 定义元素的细节。 5

<dfn> 定义定义项目。 4 5

<dir> HTML 5 中不支持。定义目录列表。 4

<div> 定义文档中的一个部分。 4 5

<dl> 定义定义列表。 4 5

<dt> 定义定义的项目。 4 5

 定义强调文本。 4 5

<embed> 定义外部交互内容或插件。 5

<fieldset> 定义 fieldset。 4 5

<figcaption> 定义 figure 元素的标题。 5

<figure> 定义媒介内容的分组，以及它们的标题。 5

 HTML 5 中不支持。 4

<footer> 定义 section 或 page 的页脚。 5

<form> 定义表单。 4 5

<frame> HTML 5 中不支持。定义子窗口（框架）。 4

<frameset> HTML 5 中不支持。定义框架的集。 4

<h1> to <h6> 定义标题 1 到标题 6。 4 5

<head> 定义关于文档的信息。 4 5

<header> 定义 section 或 page 的页眉。 5

<hgroup> 定义有关文档中的 section 的信息。 5

<hr> 定义水平线。 4 5

<html> 定义 html 文档。 4 5

<i> 定义斜体文本。 4 5

<iframe> 定义行内的子窗口（框架）。 4 5

 定义图像。 4 5

<input> 定义输入域。 4 5

<ins> 定义插入文本。 4 5

<keygen> 定义生成密钥。 5

<isindex> HTML 5 中不支持。定义单行的输入域。 4

<kbd> 定义键盘文本。 4 5

<label> 定义表单控件的标注。 4 5

<legend> 定义 fieldset 中的标题。 4 5

 定义列表的项目。 4 5

<link> 定义资源引用。 4 5

<map> 定义图像映射。 4 5

<mark> 定义有记号的文本。 5

<menu> 定义菜单列表。 4 5

<meta> 定义元信息。 4 5

<meter> 定义预定义范围内的度量。 5

<nav> 定义导航链接。 5

<noframes> HTML 5 中不支持。定义 noframe 部分。 4

<noscript> 定义 noscript 部分。 4 5

<object> 定义嵌入对象。 4 5

 定义有序列表。 4 5

<optgroup> 定义选项组。 4 5

<option> 定义下拉列表中的选项。 4 5

<output> 定义输出的一些类型。 5

<p> 定义段落。 4 5

<param> 为对象定义参数。 4 5

<pre> 定义预格式化文本。 4 5

<progress> 定义任何类型的任务的进度。 5

<q> 定义短的引用。 4 5

<rp> 定义若浏览器不支持 ruby 元素显示的内容。 5

<rt> 定义 ruby 注释的解释。 5

<ruby> 定义 ruby 注释。 5

<s> HTML 5 中不支持。定义加删除线的文本。 4

<samp> 定义样本计算机代码。 4 5

<script> 定义脚本。 4 5

<section> 定义 section。 5

<select> 定义可选列表。 4 5

<small> 定义小号文本。 4 5

<source> 定义媒介源。 5

`<span>` 定义文档中的 section。 4 5
`<strike>` HTML 5 中不支持。定义加删除线的文本。 4
`<strong>` 定义强调文本。 4 5
`<style>` 定义样式定义。 4 5
`<sub>` 定义下标文本。 4 5
`<summary>` 定义 details 元素的标题。 5
`<sup>` 定义上标文本。 4 5
`<table>` 定义表格。 4 5
`<tbody>` 定义表格的主体。 4 5
`<td>` 定义表格单元。 4 5
`<textarea>` 定义 textarea。 4 5
`<tfoot>` 定义表格的脚注。 4 5
`<th>` 定义表头。 4 5
`<thead>` 定义表头。 4 5
`<time>` 定义日期/时间。 5
`<title>` 定义文档的标题。 4 5
`<tr>` 定义表格行。 4 5
`<tt>` HTML 5 中不支持。定义打字机文本。 4
`<u>` HTML 5 中不支持。定义下划线文本。 4
`<ul>` 定义无序列表。 4 5
`<var>` 定义变量。 4 5
`<video>` 定义视频。 5
`<xmp>` HTML 5 中不支持。定义预格式文本。 4

HTML 5 正在改变 Web

HTML5 是近十年来 Web 标准最巨大的飞跃。和以前的版本不同, HTML 5 并非仅仅用来表示 Web 内容, 它的使命是将 Web 带入一个成熟的应用平台, 在这个平台上, 视频, 音频, 图象, 动画, 以及同电脑的交互都被标准化。尽管 HTML 5 的实现还有很长的路要走, 但 HTML 5 正在改变 Web。

HTML 最近的一次升级是 1999 年 12 月发布的 HTML 4.01。自那以后, 发生了很多事。最初的浏览器战争已经结束, Netscape 灰飞烟灭, IE5 作为赢家后来又发展到 IE6, IE7, IE8 到 IE9。Mozilla Firefox 从 Netscape 的死灰中诞生, 并跃居第二位。苹果和 Google 各自推出自己的浏览器, 而小家碧玉的 Opera 仍然嘤嘤嗡嗡地活着, 并以推动 Web 标准为已命。我们甚至在手机和游戏机上有了真正的 Web 体验, 感谢 Opera, iPhone 以及 Google 已经推出的 Android。

然而这一切, 仅仅让 Web 标准运动变得更加混乱, HTML 5 和其它标准被束之高阁, 结果, HTML 5 一直以来都是以草案的面目示人。

于是, 一些公司联合起来, 成立了一个叫做 Web Hypertext Application Technology Working Group (Web 超文本应用技术工作组 - WHATWG) 的组织, 他们将重新拣起 HTML 5。这个组织独立于 W3C, 成员来自 Mozilla, KHTML/Webkit 项目组, Google, Apple, Opera 以及微软。尽管 HTML 5 草案不会在短期内获得认可, 但 HTML 5 总算得以延续。

HTML 5 将带来什么？以下是 HTML 5 草案中最激动人心的部分：

全新的，更合理的 Tag，多媒体对象将不再全部绑定在 object 或 embed Tag 中，而是视频有视频的 Tag，音频有音频的 Tag。本地数据库。这个功能将内嵌一个本地的 SQL 数据库，以加速交互式搜索，缓存以及索引功能。同时，那些离线 Web 程序也将因此获益匪浅。不需要插件的富动画。Canvas 对象将给浏览器带来直接在上面绘制矢量图的能力，这意味着我们可以脱离 Flash 和 Silverlight，直接在浏览器中显示图形或动画。一些最新的浏览器，除了 IE，已经开始支持 Canvas。浏览器中的真正程序。将提供 API 实现浏览器内的编辑，拖放，以及各种图形用户界面的能力。内容修饰 Tag 将被剔除，而使用 CSS。理论上讲，HTML 5 是培育新 Web 标准的土壤，让各种设想在他的组织者之间分享，但 HTML 5 目前仍处于试验阶段。

Mozilla 的技术副总裁 Mike Shaver 说，HTML 5 是一个被寄予厚望的概念，它既是 WHATWG 组织的实验田，又是 W3C 的标准之路。

Shaver 认为，Mozilla 的兴趣和 WHATWG 实验相吻合，Mozilla 在 HTML 5 工作组中非常活跃，我们对一些早期的细则进行实验并将成熟的结果提交 W3C。

在过去的几年，Mozilla 随着各种出现的新标准，推出多个富有前瞻性的项目，包括 Prism，一个用于离线运行 Web 程序的系统，以及 Weave，一个数据存储框架。

Shaver 说，HTML 5 运动肇始于对 W3C 的不耐烦，Web 标准中的很多进展都因 W3C 将重点从 HTML 转移到 XML 而停滞不前。

很多基于 XML 架构的新技术被设计出来替代 HTML，Shaver 说，这不是一条正确的道路，人们不应象黑瞎子掰玉米把样一边掰一边丢。

HTML 5 的新实验在 Firefox 以及基于 Webkit 的 Safari 和 Chrome 浏览器中逐渐得到强化，但仍有不少问题。

Chrome 的开发者 Darin Fisher 说，Chrome 仍在襁褓中时，就不得不面临几个问题，尽管使用的是最新的 Webkit，HTML 5 的本地数据库功能在 Chrome 的初期版本中并没有实现。因为 Chrome 的沙箱机制和 Webkit 的数据库功能有冲突。

而由于 Chrome 属于秘密开发，Chrome 的开发人员也不便参与 Webkit 的开发。

我们要想保守 Chrome 的秘密，就无法参与 Webkit 社区。Fisher 说，我们很希望可以在某些方面给 Webkit 以帮助，我们拥有众多经验丰富的开发者，我们很想知道人们目前遇到的挑战并乐意提供帮助。

随着 Chrome 的发布，Fisher 说他的团队成员有时会 and Webkit 的人一起吃饭，有些人私下里还成了好朋友。Fisher 称，他们迫切地想同其他 Webkit 开发组一起工作解决离线数据库的问题。

Chrome 里面还包含 Google 的开源 Gears 技术，用来实现与 HTML 5 类似的离线功能。

Gears 可以看作已有 API 的替代品，Fisher 说，HTML 5 对新浏览器来说是非常好的东西，但绝大多数用户还使用旧浏览器。Gears 可以让那些旧浏览器也获得这样的 API，我们正在为 HTML 5 版 API 提供兼容。

Gears 兼容性非常好，它正成为将 HTML 5 带向人们桌面的另外一条途径。

目前，绝大多数工作由 Apple, Mozilla, Opera, Google 以及 Trolltech 展开。微软在干什么？IE 因其对 Web 标准的迟钝而闻名，更不要说 HTML 5。但 IE8 可能会做出改变。

微软 IE 平台与 WHAT 工作组主席 Chris Wilson 在邮件中称，我们希望我们现在开始的工作可

以在 HTML 工作组创建一套测试系统。Wilson 说，IE 开发组仍然对 HTML 5 的一些提议感到担忧。我觉得工作组的所有成员都会承认我们还有很多事要做。

目前处于 Beta 版的 IE9，已经包含 HTML 5 的诸多新功能。它拥有一个跨文档消息系统，本地存储，以及一些离线事件来检测网络的中断。但还有些功能还未提上议程，如 Canvas。

HTML 5 非常庞大，仍处在开发阶段，我认为浏览器厂商应当尽快达成一致，而每个浏览器的具体实现时间可以自己选择。Web 开发者和浏览器厂商会同意 Wilson 的下面这句话，这确切无疑是一个激动人心的时刻，我们希望看到 Web 成为新的应用平台。

教程前言

HTML 5 是下一代的 HTML。

本教程向您讲解 HTML 5 中的新特性，以及每种标签的使用。

每一章中的实例

通过我们的 HTML 编辑器，您能够编辑 HTML，然后点击按钮来查看结果。

实例

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<video width="320" height="240" controls="controls">
```

```
  <source src="movie.ogg" type="video/ogg">
```

```
  <source src="movie.mp4" type="video/mp4">
```

```
Your browser does not support the video tag.
```

```
</video>
```

```
</body>
```

```
</html>
```

TIY

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<video width="320" height="240" controls="controls">
```

```
  <source src="/i/movie.ogg" type="video/ogg">
```

```
  <source src="/i/movie.mp4" type="video/mp4">
```

Your browser does not support the video tag.

```
</video>
```

```
</body>
```

```
</html>
```

实例 **TIY** 中的代码用来查看它是如何工作的。

现在就开始学习 HTML 5 ！

HTML 5 简介

HTML 5 是下一代的 **HTML**。

什么是 **HTML 5**？

HTML 5 将成为 HTML、XHTML 以及 HTML DOM 的新标准。

HTML 的上一个版本诞生于 1999 年。自从那以后，Web 世界已经经历了巨变。

HTML 5 仍处于完善之中。然而，大部分现代浏览器已经具备了某些 HTML 5 支持。

HTML 5 是如何起步的？

HTML 5 是 W3C 与 WHATWG 合作的结果。

编者注：W3C 指 World Wide Web Consortium，万维网联盟。

编者注：WHATWG 指 Web Hypertext Application Technology Working Group。

WHATWG 致力于 web 表单和应用程序，而 W3C 专注于 XHTML 2.0。在 2006 年，双方决定进行合作，来创建一个新版本的 HTML。

为 HTML 5 建立的一些规则：

新特性应该基于 HTML、CSS、DOM 以及 JavaScript。

减少对外部插件的需求（比如 Flash）

更优秀的错误处理

更多取代脚本的标记

HTML 5 应该独立于设备

开发进程应对公众透明

新特性

HTML 5 中的一些有趣的新特性：

用于绘画的 canvas 元素

用于媒介回放的 video 和 audio 元素

对本地离线存储的更好的支持

新的特殊内容元素，比如 article、footer、header、nav、section

新的表单控件，比如 calendar、date、time、email、url、search

浏览器支持

最新版本的 Safari、Chrome、Firefox 以及 Opera 支持某些 HTML 5 特性。Internet Explorer 9 将支持某些 HTML 5 特性。

HTML 5 视频

许多时髦的网站都提供视频。HTML 5 提供了展示视频的标准。

Web 上的视频

直到现在，仍然不存在一项旨在网页上显示视频的标准。

今天，大多数视频是通过插件（比如 Flash）来显示的。然而，并非所有浏览器都拥有同样的插件。

HTML 5 规定了一种通过 video 元素来包含视频的标准方法。

视频格式

当前，video 元素支持三种视频格式：

| 格式 | IE | Firefox | Opera | Chrome | Safari |
|--------|------|---------|-------|--------|--------|
| Ogg | No | 3.5+ | 10.5+ | 5.0+ | No |
| MPEG 4 | 9.0+ | No | No | 5.0+ | 3.0+ |
| WebM | No | 4.0+ | 10.6+ | 6.0+ | No |

Ogg = 带有 Theora 视频编码和 Vorbis 音频编码的 Ogg 文件

MPEG4 = 带有 H.264 视频编码和 AAC 音频编码的 MPEG 4 文件

WebM = 带有 VP8 视频编码和 Vorbis 音频编码的 WebM 文件

如何工作

如需在 HTML 5 中显示视频，您所有需要的是：

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

```
</video>
```

TIY

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<video src="/i/movie.ogg" width="320" height="240" controls="controls">
```

Your browser does not support the video tag.

```
</video>
```

```
</body>
```

```
</html>
```

control 属性供添加播放、暂停和音量控件。

包含宽度和高度属性也是不错的主意。

<video> 与 </video> 之间插入的内容是供不支持 video 元素的浏览器显示的：

实例

```
<video src="movie.ogg" width="320" height="240" controls="controls">
```

Your browser does not support the video tag.

```
</video>
```

TIY

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<video width="320" height="240" controls="controls">
```

```
  <source src="/i/movie.ogg" type="video/ogg">
```

```
  <source src="/i/movie.mp4" type="video/mp4">
```

Your browser does not support the video tag.

```
</video>
```

```
</body>
```

```
</html>
```

上面的例子使用一个 Ogg 文件，适用于 Firefox、Opera 以及 Chrome 浏览器。

要确保适用于 Safari 浏览器，视频文件必须是 MPEG4 类型。

video 元素允许多个 source 元素。source 元素可以链接不同的视频文件。浏览器将使用第一个可识别的格式：

实例

```
<video width="320" height="240" controls="controls">
```

```
  <source src="movie.ogg" type="video/ogg">
```

```
<source src="movie.mp4" type="video/mp4">
```

Your browser does not support the video tag.

```
</video>
```

TIY

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<video width="320" height="240" controls="controls">
```

```
<source src="/i/movie.ogg" type="video/ogg">
```

```
<source src="/i/movie.mp4" type="video/mp4">
```

Your browser does not support the video tag.

```
</video>
```

```
</body>
```

```
</html>
```

Internet Explorer

Internet Explorer 8 不支持 video 元素。在 IE 9 中，将提供对使用 MPEG4 的 video 元素的支持。

<video> 标签的属性

| 属性 | 值 | 描述 |
|----|---|----|
|----|---|----|

| | | |
|----------|----------|---|
| autoplay | autoplay | 如果出现该属性，则视频在就绪后马上播放。 |
| controls | controls | 如果出现该属性，则向用户显示控件，比如播放按钮。 |
| height | pixels | 设置视频播放器的高度。 |
| loop | loop | 如果出现该属性，则当媒介文件完成播放后再次开始播放。 |
| preload | preload | 如果出现该属性，则视频在页面加载时进行加载，并预备播放。

如果使用 "autoplay"，则忽略该属性。 |
| src | url | 要播放的视频的 URL。 |
| width | pixels | 设置视频播放器的宽度。 |

相关页面

参考手册: [HTML 5 <video> 标签](http://www.w3school.com.cn/HTML5/HTML5_video.asp)--[http://www.w3school.com.cn/HTML 5/HTML 5_video.asp](http://www.w3school.com.cn/HTML5/HTML5_video.asp)

HTML 5 音频

HTML 5 提供了播放音频的标准。

Web 上的音频

直到现在，仍然不存在一项旨在网页上播放音频的标准。

今天，大多数音频是通过插件（比如 **Flash**）来播放的。然而，并非所有浏览器都拥有同样的插件。

HTML 5 规定了一种通过 **audio** 元素来包含音频的标准方法。

audio 元素能够播放声音文件或者音频流。

音频格式

当前，**audio** 元素支持三种音频格式：

| | IE 9 | Firefox 3.5 | Opera 10.5 | Chrome 3.0 | Safari 3.0 |
|------------|------|-------------|------------|------------|------------|
| Ogg Vorbis | | √ | √ | √ | |
| MP3 | √ | | | √ | √ |
| Wav | | √ | √ | | √ |

如何工作

如需在 HTML 5 中播放音频，您所有需要的是：

```
<audio src="song.ogg" controls="controls">
```

```
</audio>
```

control 属性供添加播放、暂停和音量控件。

<audio> 与 </audio> 之间插入的内容是供不支持 audio 元素的浏览器显示的：

实例

```
<audio src="song.ogg" controls="controls">
```

Your browser does not support the audio tag.

```
</audio>
```

TIY

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<audio src="/i/song.ogg" controls="controls">
```

Your browser does not support the audio element.

```
</audio>
```

```
</body>
```

```
</html>
```

上面的例子使用一个 Ogg 文件，适用于 Firefox、Opera 以及 Chrome 浏览器。

要确保适用于 Safari 浏览器，音频文件必须是 MP3 或 Wav 类型。

audio 元素允许多个 source 元素。source 元素可以链接不同的音频文件。浏览器将使用第一个可识别的格式：

实例

```
<audio controls="controls">
```

```
  <source src="song.ogg" type="audio/ogg">
```

```
  <source src="song.mp3" type="audio/mpeg">
```

Your browser does not support the audio tag.

```
</audio>
```

TIY

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<audio controls="controls">
```

```
  <source src="/i/song.ogg" type="audio/ogg">
```

```
  <source src="/i/song.mp3" type="audio/mpeg">
```

Your browser does not support the audio element.

```
</audio>
```

```
</body>
```

```
</html>
```

Internet Explorer

Internet Explorer 8 不支持 **audio** 元素。在 IE 9 中，将提供对 **audio** 元素的支持。

<audio> 标签的属性

| 属性 | 值 | 描述 |
|-----------------|-----------------|---|
| autoplay | autoplay | 如果出现该属性，则音频在就绪后马上播放。 |
| controls | controls | 如果出现该属性，则向用户显示控件，比如播放按钮。 |
| loop | loop | 如果出现该属性，则每当音频结束时重新开始播放。 |
| preload | preload | 如果出现该属性，则音频在页面加载时进行加载，并预备播放。

如果使用 "autoplay"，则忽略该属性。 |
| src | url | 要播放的音频的 URL。 |

相关页面

参考手册: HTML 5 **<audio>** 标签--http://www.w3school.com.cn/HTML 5/HTML 5_audio.asp

HTML 5 Canvas

canvas 元素用于在网页上绘制图形。

什么是 Canvas?

HTML 5 的 **canvas** 元素使用 **JavaScript** 在网页上绘制图像。

画布是一个矩形区域，您可以控制其每一像素。

canvas 拥有多种绘制路径、矩形、圆形、字符以及添加图像的方法。

创建 Canvas 元素

向 HTML 5 页面添加 **canvas** 元素。

规定元素的 **id**、宽度和高度：

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

通过 **JavaScript** 来绘制

canvas 元素本身是没有绘图能力的。所有的绘制工作必须在 **JavaScript** 内部完成：

```
<script type="text/javascript">
```

```
var c=document.getElementById("myCanvas");
```

```
var cxt=c.getContext("2d");
```

```
cxt.fillStyle="#FF0000";
```

```
cxt.fillRect(0,0,150,75);
```

```
</script>
```

JavaScript 使用 **id** 来寻找 **canvas** 元素：

```
var c=document.getElementById("myCanvas");
```

然后，创建 **context** 对象：

```
var cxt=c.getContext("2d");
```

`getContext("2d")` 对象是内建的 HTML 5 对象，拥有多种绘制路径、矩形、圆形、字符以及添加图像的方法。

下面的两行代码绘制一个红色的矩形：

```
cxt.fillStyle="#FF0000";
```

```
cxt.fillRect(0,0,150,75);
```

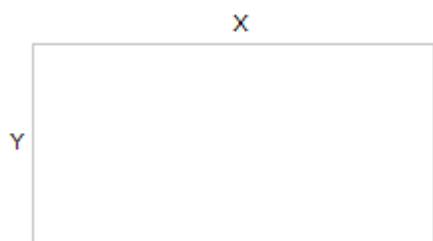
`fillStyle` 方法将其染成红色，`fillRect` 方法规定了形状、位置和尺寸。

理解坐标

上面的 `fillRect` 方法拥有参数 `(0,0,150,75)`。

意思是：在画布上绘制 **150x75** 的矩形，从左上角开始 `(0,0)`。

如下图所示，画布的 **X** 和 **Y** 坐标用于在画布上对绘画进行定位。



实例：把鼠标悬停在矩形上可以看到坐标，亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
<style type="text/css">
```

```
body
```

```
{
```

```
font-size:70%;
```

```
font-family:verdana,helvetica,arial,sans-serif;

}

</style>

<script type="text/javascript">

function cnvs_getCoordinates(e)

{

x=e.clientX;

y=e.clientY;

document.getElementById("xycoordinates").innerHTML="Coordinates: (" + x + "," + y + ")";

}

function cnvs_clearCoordinates()

{

document.getElementById("xycoordinates").innerHTML="";

}

</script>

</head>

<body style="margin:0px;">

<p>把鼠标悬停在下面的矩形上可以看到坐标: </p>

<div id="coordiv" style="float:left;width:199px;height:99px;border:1px solid #c3c3c3"

onmousemove="cnvs_getCoordinates(event)"

onmouseout="cnvs_clearCoordinates()"></div>

<br />
```

```
<br />
```

```
<br />
```

```
<div id="xycoordinates"></div>
```

```
</body>
```

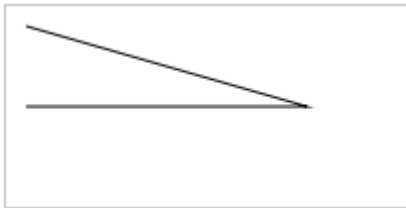
```
</html>
```

更多 Canvas 实例

下面的在 **canvas** 元素上进行绘画的更多实例：

实例 - 线条

通过指定从何处开始，在何处结束，来绘制一条线：



JavaScript 代码：

```
<script type="text/javascript">
```

```
var c=document.getElementById("myCanvas");
```

```
var cxt=c.getContext("2d");
```

```
cxt.moveTo(10,10);
```

```
cxt.lineTo(150,50);
```

```
cxt.lineTo(10,50);
```

```
cxt.stroke();
```

```
</script>
```

canvas 元素:

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">
```

Your browser does not support the canvas element.

```
</canvas>
```

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">
```

Your browser does not support the canvas element.

```
</canvas>
```

```
<script type="text/javascript">
```

```
var c=document.getElementById("myCanvas");
```

```
var cxt=c.getContext("2d");
```

```
cxt.moveTo(10,10);
```

```
cxt.lineTo(150,50);
```

```
cxt.lineTo(10,50);
```

```
cxt.stroke();
```

```
</script>
```

```
</body>
```

</html>

实例 - 圆形

通过规定尺寸、颜色和位置，来绘制一个圆：



JavaScript 代码：

```
<script type="text/javascript">
```

```
var c=document.getElementById("myCanvas");
```

```
var cxt=c.getContext("2d");
```

```
cxt.fillStyle="#FF0000";
```

```
cxt.beginPath();
```

```
cxt.arc(70,18,15,0,Math.PI*2,true);
```

```
cxt.closePath();
```

```
cxt.fill();
```

```
</script>
```

canvas 元素：

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">
```

Your browser does not support the canvas element.

```
</canvas>
```

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">
```

Your browser does not support the canvas element.

```
</canvas>
```

```
<script type="text/javascript">
```

```
var c=document.getElementById("myCanvas");
```

```
var cxt=c.getContext("2d");
```

```
cxt.fillStyle="#FF0000";
```

```
cxt.beginPath();
```

```
cxt.arc(70,18,15,0,Math.PI*2,true);
```

```
cxt.closePath();
```

```
cxt.fill();
```

```
</script>
```

```
</body>
```

```
</html>
```

实例 - 渐变

使用您指定的颜色来绘制渐变背景:



JavaScript 代码:

```
<script type="text/javascript">
```

```
var c=document.getElementById("myCanvas");
```

```
var cxt=c.getContext("2d");
```

```
var grd=cxt.createLinearGradient(0,0,175,50);
```

```
grd.addColorStop(0,"#FF0000");
```

```
grd.addColorStop(1,"#00FF00");
```

```
cxt.fillStyle=grd;
```

```
cxt.fillRect(0,0,175,50);
```

```
</script>
```

canvas 元素:

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">
```

Your browser does not support the canvas element.

</canvas>

亲自试一试吧，代码如下：

<!DOCTYPE HTML>

<html>

<body>

<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">

Your browser does not support the canvas element.

</canvas>

<script type="text/javascript">

var c=document.getElementById("myCanvas");

var cxt=c.getContext("2d");

var grd=cxt.createLinearGradient(0,0,175,50);

grd.addColorStop(0,"#FF0000");

grd.addColorStop(1,"#00FF00");

cxt.fillStyle=grd;

```
cxt.fillRect(0,0,175,50);
```

```
</script>
```

```
</body>
```

```
</html>
```

实例 - 图像

把一幅图像放置到画布上：



JavaScript 代码：

```
<script type="text/javascript">
```

```
var c=document.getElementById("myCanvas");
```

```
var cxt=c.getContext("2d");
```

```
var img=new Image()
```

```
img.src="flower.png"
```

```
cxt.drawImage(img,0,0);
```

```
</script>
```

canvas 元素：

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">
```

Your browser does not support the canvas element.

```
</canvas>
```

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #c3c3c3;">
```

Your browser does not support the canvas element.

```
</canvas>
```

```
<script type="text/javascript">
```

```
var c=document.getElementById("myCanvas");
```

```
var cxt=c.getContext("2d");
```

```
var img=new Image()
```

```
img.src="/i/eg_flower.png"
```

```
cxt.drawImage(img,0,0);
```

```
</script>
```

</body>

相关页面

参考手册: HTML 5 <canvas> 标签--http://www.w3school.com.cn/HTML 5/HTML 5_canvas.asp

参考手册: HTML DOM Canvas 对象

--http://www.w3school.com.cn/html5/dom_obj_canvas.asp

HTML 5 Web 存储

在客户端存储数据

HTML 5 提供了两种在客户端存储数据的新方法:

localStorage - 没有时间限制的数据存储

sessionStorage - 针对一个 session 的数据存储

之前, 这些都是由 **cookie** 完成的。但是 **cookie** 不适合大量数据的存储, 因为它们由每个对服务器的请求来传递, 这使得 **cookie** 速度很慢而且效率也不高。

在 **HTML 5** 中, 数据不是由每个服务器请求传递的, 而是只有在请求时使用数据。它使在不影响网站性能的情况下存储大量数据成为可能。

对于不同的网站, 数据存储于不同的区域, 并且一个网站只能访问其自身的数据。

HTML 5 使用 **JavaScript** 来存储和访问数据。

localStorage 方法

localStorage 方法存储的数据没有时间限制。第二天、第二周或下一年之后, 数据依然可用。

如何创建和访问 **localStorage**:

实例

```
<script type="text/javascript">
```

```
localStorage.lastname="Smith";  
  
document.write(localStorage.lastname);  
  
</script>
```

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>  
  
<html>  
  
<body>  
  
  
  
  
  
  
  
  
  
<script type="text/javascript">  
  
localStorage.lastname="Smith";  
  
document.write("Last name: " + localStorage.lastname);  
  
  
  
  
  
  
  
  
  
</script>  
  
  
  
  
  
  
  
  
  
</body>  
  
</html>
```

下面的例子对用户访问页面的次数进行计数：

实例

```
<script type="text/javascript">  
  
if (localStorage.pagecount)
```

```
{  
  
    localStorage.pagecount=Number(localStorage.pagecount) +1;  
  
}  
  
else  
  
    {  
  
        localStorage.pagecount=1;  
  
    }  
  
document.write("Visits " + localStorage.pagecount + " time(s).");  
  
</script>
```

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>  
  
<html>  
  
<body>  
  
    <script type="text/javascript">  
  
        localStorage.lastname="Smith";  
  
        document.write("Last name: " + localStorage.lastname);  
  
    </script>  
  
</body>  
  
</html>
```

sessionStorage 方法

sessionStorage 方法针对一个 session 进行数据存储。当用户关闭浏览器窗口后，数据会被删除。

如何创建并访问一个 sessionStorage:

实例

```
<script type="text/javascript">

sessionStorage.lastname="Smith";

document.write(sessionStorage.lastname);

</script>
```

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>

<html>

<body>

<script type="text/javascript">

if (localStorage.pagecount)

    {

        localStorage.pagecount=Number(localStorage.pagecount) +1;

    }

else

    {

        localStorage.pagecount=1;

    }

document.write("Visits: " + localStorage.pagecount + " time(s).");

</script>
```

<p>刷新页面会看到计数器在增长。</p>

<p>请关闭浏览器窗口，然后再试一次，计数器会继续计数。</p>

</body>

</html>

下面的例子对用户在当前 **session** 中访问页面的次数进行计数：

实例

```
<script type="text/javascript">
```

```
if (sessionStorage.pagecount)
```

```
{
```

```
    sessionStorage.pagecount=Number(sessionStorage.pagecount) +1;
```

```
}
```

```
else
```

```
{
```

```
    sessionStorage.pagecount=1;
```

```
}
```

```
document.write("Visits "+sessionStorage.pagecount+" time(s) this session.");
```

```
</script>
```

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<script type="text/javascript">

if (sessionStorage.pagecount)

    {

        sessionStorage.pagecount=Number(sessionStorage.pagecount) + 1;

    }

else

    {

        sessionStorage.pagecount=1;

    }

document.write("Visits " + sessionStorage.pagecount + " time(s) this session.");

</script>

<p>刷新页面会看到计数器在增长。</p>

<p>请关闭浏览器窗口，然后再试一次，计数器已经重置了。</p>

</body>

</html>
```

HTML 5 Input 类型

HTML 5 新的 Input 类型

HTML 5 拥有多个新的表单输入类型。这些新特性提供了更好的输入控制和验证。

本章全面介绍这些新的输入类型：

email

url

number

range

Date pickers (date, month, week, time, datetime, datetime-local)

search

color

浏览器支持

Input type	IE	Firefox	Opera	Chrome	Safari
email	No	4.0	9.0	10.0	No
url	No	4.0	9.0	10.0	No
number	No	No	9.0	7.0	No
range	No	No	9.0	4.0	4.0
Date pickers	No	No	9.0	10.0	No
search	No	4.0	11.0	10.0	No
color	No	No	11.0	No	No

注释: **Opera** 对新的输入类型的支持最好。不过您已经可以在所有主流的浏览器中使用它们了。即使不被支持, 仍然可以显示为常规的文本域。

Input 类型 - email

email 类型用于应该包含 **e-mail** 地址的输入域。

在提交表单时，会自动验证 **email** 域的值。

实例

E-mail:

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<form action="/example/HTML 5/demo_form.asp" method="get">
```

```
E-mail: <input type="email" name="user_email" /><br />
```

```
<input type="submit" />
```

```
</form>
```

```
</body>
```

```
</html>
```

提示: iPhone 中的 Safari 浏览器支持 **email** 输入类型, 并通过改变触摸屏键盘来配合它(添加 @ 和 .com 选项)。

Input 类型 - url

url 类型用于应该包含 **URL** 地址的输入域。

在提交表单时，会自动验证 **url** 域的值。

实例

Homepage: <input type="url" name="user_url" />

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<form action="/example/HTML 5/demo_form.asp" method="get">
```

Homepage: <input type="url" name="user_url" />


```
<input type="submit" />
```

```
</form>
```

```
</body>
```

```
</html>
```

提示：iPhone 中的 Safari 浏览器支持 url 输入类型，并通过改变触摸屏键盘来配合它（添加 .com 选项）。

Input 类型 - number

number 类型用于应该包含数值的输入域。

您还能够设定对所接受的数字的限定：

实例

Points: <input type="number" name="points" min="1" max="10" />

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<form action="/example/HTML 5/demo_form.asp" method="get">
```

```
Points: <input type="number" name="points" min="1" max="10" />
```

```
<input type="submit" />
```

```
</form>
```

```
</body>
```

```
</html>
```

请使用下面的属性来规定对数字类型的限定：

属性	值	描述
max	number	规定允许的最大值
min	number	规定允许的最小值
step	number	规定合法的数字间隔（如果 step="3" ，则合法的数是 -3,0,3,6 等）
value	number	规定默认值

请试一下带有所有限定属性的例子：亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<form action="/example/HTML 5/demo_form.asp" method="get">

<input type="number" name="points" min="0" max="10" step="3" value="6" />

<input type="submit" />

</form>


</body>

</html>
```

提示：iPhone 中的 Safari 浏览器支持 **number** 输入类型，并通过改变触摸屏键盘来配合它（显示数字）。

Input 类型 - range

range 类型用于应该包含一定范围内数字值的输入域。

range 类型显示为滑动条。

您还能够设定对所接受的数字的限定：

实例

```
<input type="range" name="points" min="1" max="10" />
```

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>

<html>

<body>
```

```
<form action="/example/HTML 5/demo_form.asp" method="get">
```

```
Date: <input type="date" name="user_date" />
```

```
<input type="submit" />
```

```
</form>
```

```
</body>
```

```
</html>
```

请使用下面的属性来规定对数字类型的限定：

属性	值	描述
max	number	规定允许的最大值
min	number	规定允许的最小值
step	number	规定合法的数字间隔（如果 step="3" ，则合法的数是 -3,0,3,6 等）
value	number	规定默认值

Input 类型 - Date Pickers（数据检出器）

HTML 5 拥有多个可供选取日期和时间的新输入类型：

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）

下面的例子允许您从日历中选取一个日期：

实例

Date:

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<form action="/example/HTML 5/demo_form.asp" method="get">
```

Date:

```
<input type="submit" />
```

```
</form>
```

```
</body>
```

```
</html>
```

输入类型 "month":亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<form action="/example/HTML 5/demo_form.asp" method="get">
```

```
Month: <input type="month" name="user_date" />
```

```
<input type="submit" />
```

```
</form>
```

```
</body>
```

```
</html>
```

输入类型 **"week"**:亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<form action="/example/HTML 5/demo_form.asp" method="get">
```

```
Week: <input type="week" name="user_date" />
```

```
<input type="submit" />
```

```
</form>
```

```
</body>
```

```
</html>
```

输入类型 **"time"**:亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<form action="/example/HTML 5/demo_form.asp" method="get">
```

```
Time: <input type="time" name="user_date" />
```

```
<input type="submit" />
```

```
</form>
```

```
</body>
```

```
</html>
```

输入类型 "datetime": 亲自试一试吧, 代码如下:

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<form action="/example/HTML 5/demo_form.asp" method="get">
```

```
Date and time: <input type="datetime" name="user_date" />
```

```
<input type="submit" />
```

```
</form>
```

```
</body>
```

```
</html>
```

输入类型 "datetime-local":亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<form action="/example/HTML 5/demo_form.asp" method="get">
```

```
Date and time: <input type="datetime-local" name="user_date" />
```

```
<input type="submit" />
```

```
</form>
```

```
</body>
```

```
</html>
```

Input 类型 - search

search 类型用于搜索域，比如站点搜索或 Google 搜索。

search 域显示为常规的文本域。

HTML 5 表单元素

HTML 5 的新的表单元素：

HTML 5 拥有若干涉及表单的元素和属性。

本章介绍以下新的表单元素：

datalist

keygen

output

浏览器支持

Input type	IE	Firefox	Opera	Chrome	Safari
datalist	No	No	9.5	No	No
keygen	No	No	10.5	3.0	No
output	No	No	9.5	No	No

datalist 元素

datalist 元素规定输入域的选项列表。

列表是通过 **datalist** 内的 **option** 元素创建的。

如需把 **datalist** 绑定到输入域，请用输入域的 **list** 属性引用 **datalist** 的 **id**：

实例

Webpage: `<input type="url" list="url_list" name="link" />`

```
<datalist id="url_list">
```

```
<option label="W3School" value="http://www.W3School.com.cn" />
```

```
<option label="Google" value="http://www.google.com" />
```

```
<option label="Microsoft" value="http://www.microsoft.com" />
```

```
</datalist>
```

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<form action="/example/HTML 5/demo_form.asp" method="get">
```

```
Webpage: <input type="url" list="url_list" name="link" />
```

```
<datalist id="url_list">
```

```
    <option label="W3School" value="http://www.w3school.com.cn" />
```

```
    <option label="Google" value="http://www.google.com" />
```

```
    <option label="Microsoft" value="http://www.microsoft.com" />
```

```
</datalist>
```

```
<input type="submit" />
```

```
</form>
```

```
</body>
```

```
</html>
```

提示: option 元素永远都要设置 value 属性。

keygen 元素

keygen 元素的作用是提供一种验证用户的可靠方法。

keygen 元素是密钥对生成器（key-pair generator）。当提交表单时，会生成两个键，一个是私钥，一个公钥。

私钥（private key）存储于客户端，公钥（public key）则被发送到服务器。公钥可用于之后验证用户的客户端证书（client certificate）。

目前，浏览器对此元素的糟糕的支持度不足以使其成为一种有用的安全标准。

实例

```
<form action="demo_form.asp" method="get">
```

```
Username: <input type="text" name="usr_name" />
```

Encryption: <keygen name="security" />

<input type="submit" />

</form>

亲自试一试吧，代码如下：

<!DOCTYPE HTML>

<html>

<body>

<form action="/example/HTML 5/demo_form.asp" method="get">

Username: <input type="text" name="usr_name" />

Encryption: <keygen name="security" />

<input type="submit" />

</form>

</body>

</html>

output 元素

output 元素用于不同类型的输出，比如计算或脚本输出：

实例

<output id="result" onforminput="resCalc()"></output>

亲自试一试吧，代码如下：

<!DOCTYPE HTML>

```
<html>

<head>

<script type="text/javascript">

function resCalc()

{

numA=document.getElementById("num_a").value;

numB=document.getElementById("num_b").value;

document.getElementById("result").value=Number(numA)+Number(numB);

}

</script>

</head>

<body>

<p>使用 output 元素的简易计算器: </p>

<form onsubmit="return false">

  <input id="num_a" /> +

  <input id="num_b" /> =

  <output id="result" onforminput="resCalc()"></output>

</form>

</body>

</html>
```

HTML 5 表单属性

HTML 5 的新的表单属性

本章讲解涉及 `<form>` 和 `<input>` 元素的新属性。

新的 `form` 属性:

`autocomplete`

`novalidate`

新的 `input` 属性:

`autocomplete`

`autofocus`

`form`

`form overrides` (`formaction`, `formenctype`, `formmethod`, `formnovalidate`, `formtarget`)

`height` 和 `width`

`list`

`min`, `max` 和 `step`

`multiple`

`pattern` (regexp)

`placeholder`

`required`

浏览器支持

Input type	IE	Firefox	Opera	Chrome	Safari
<code>autocomplete</code>	8.0	3.5	9.5	3.0	4.0

autofocus	No	No	10.0	3.0	4.0
form	No	No	9.5	No	No
form overrides	No	No	10.5	No	No
height and width	8.0	3.5	9.5	3.0	4.0
list	No	No	9.5	No	No
min, max and step	No	No	9.5	3.0	No
multiple	No	3.5	No	3.0	4.0
novalidate	No	No	No	No	No
pattern	No	No	9.5	3.0	No
placeholder	No	No	No	3.0	3.0
required	No	No	9.5	3.0	No

autocomplete 属性

autocomplete 属性规定 form 或 input 域应该拥有自动完成功能。

注释：autocomplete 适用于 <form> 标签，以及以下类型的 <input> 标签：text, search, url, telephone, email, password, datepickers, range 以及 color。

当用户在自动完成域中开始输入时，浏览器应该在该域中显示填写的选项：

实例

```
<form action="demo_form.asp" method="get" autocomplete="on">
```

```
First name: <input type="text" name="fname" /><br />
```

```
Last name: <input type="text" name="lname" /><br />
```

E-mail: <input type="email" name="email" autocomplete="off" />

<input type="submit" />

</form>

亲自试一试吧，代码如下：

<!DOCTYPE HTML>

<html>

<body>

<form action="/example/HTML 5/demo_form.asp" method="get" autocomplete="on">

First name: <input type="text" name="fname" />

Last name: <input type="text" name="lname" />

E-mail: <input type="email" name="email" autocomplete="off" />

<input type="submit" />

</form>

<p>请填写并提交此表单，然后重载页面，来查看自动完成功能是如何工作的。</p>

<p>请注意，表单的自动完成功能是打开的，而 e-mail 域是关闭的。</p>

</body>

</html>

注释：在某些浏览器中，您可能需要启用自动完成功能，以使该属性生效。

autofocus 属性

autofocus 属性规定在页面加载时，域自动地获得焦点。

注释: **autofocus** 属性适用于所有 `<input>` 标签的类型。

实例

User name: `<input type="text" name="user_name" autofocus="autofocus" />`

亲自试一试吧, 代码如下:

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<form action="/example/HTML 5/demo_form.asp" method="get">
```

User name: `<input type="text" name="user_name" autofocus="autofocus" />`

```
<input type="submit" />
```

```
</form>
```

```
</body>
```

```
</html>
```

form 属性

form 属性规定输入域所属的一个或多个表单。

注释: **form** 属性适用于所有 `<input>` 标签的类型。

form 属性必须引用所属表单的 **id**:

实例

```
<form action="demo_form.asp" method="get" id="user_form">
```

First name: `<input type="text" name="fname" />`

```
<input type="submit" />
```

```
</form>
```

```
Last name: <input type="text" name="lname" form="user_form" />
```

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<form action="/example/HTML 5/demo_form.asp" method="get" id="user_form">
```

```
First name: <input type="text" name="fname" />
```

```
<input type="submit" />
```

```
</form>
```

```
<p>下面的输入域在 form 元素之外，但仍然是表单的一部分。</p>
```

```
Last name: <input type="text" name="lname" form="user_form" />
```

```
</body>
```

```
</html>
```

注释：如需引用一个以上的表单，请使用空格分隔的列表。

表单重写属性

表单重写属性（**form override attributes**）允许您重写 **form** 元素的某些属性设定。

表单重写属性有：

formaction - 重写表单的 action 属性

formenctype - 重写表单的 enctype 属性

formmethod - 重写表单的 method 属性

formnovalidate - 重写表单的 novalidate 属性

formtarget - 重写表单的 target 属性

注释：表单重写属性适用于以下类型的 `<input>` 标签：submit 和 image。

实例

```
<form action="demo_form.asp" method="get" id="user_form">
```

```
E-mail: <input type="email" name="userid" /><br />
```

```
<input type="submit" value="Submit" />
```

```
<br />
```

```
<input type="submit" formaction="demo_admin.asp" value="Submit as admin" />
```

```
<br />
```

```
<input type="submit" formnovalidate="true" value="Submit without validation" />
```

```
<br />
```

```
</form>
```

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<form action="/example/HTML 5/demo_form.asp" method="get" id="user_form">
```

```
E-mail: <input type="email" name="userid" /><br />
```

```
<input type="submit" value="Submit" /><br />
```

```
<input type="submit" formaction="/example/HTML 5/demo_admin.asp" value="Submit as  
admin" /><br />
```

```
<input type="submit" formnovalidate="true" value="Submit without validation" /><br />
```

```
</form>
```

```
</body>
```

```
</html>
```

注释：这些属性对于创建不同的提交按钮很有帮助。

height 和 width 属性

height 和 width 属性规定用于 **image** 类型的 **input** 标签的图像高度和宽度。

注释：**height** 和 **width** 属性只适用于 **image** 类型的 **<input>** 标签。

实例

```
<input type="image" src="img_submit.gif" width="99" height="99" />
```

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<form action="/example/HTML 5/demo_form.asp" method="get">
```

```
User name: <input type="text" name="user_name" /><br />
```

```
<input type="image" src="/i/eg_submit.jpg" width="99" height="99" />
```

```
</form>
```

```
</body>
```

```
</html>
```

list 属性

list 属性规定输入域的 **datalist**。**datalist** 是输入域的选项列表。

注释: list 属性适用于以下类型的 **<input>** 标签: text, search, url, telephone, email, date pickers, number, range 以及 color。

实例

Webpage: `<input type="url" list="url_list" name="link" />`

```
<datalist id="url_list">
```

```
<option label="W3Schools" value="http://www.w3school.com.cn" />
```

```
<option label="Google" value="http://www.google.com" />
```

```
<option label="Microsoft" value="http://www.microsoft.com" />
```

```
</datalist>
```

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<form action="/example/HTML 5/demo_form.asp" method="get">
```

Webpage: `<input type="url" list="url_list" name="link" />`

```
<datalist id="url_list">

    <option label="W3School" value="http://www.w3school.com.cn" />

    <option label="Google" value="http://www.google.com" />

    <option label="Microsoft" value="http://www.microsoft.com" />

</datalist>

<input type="submit" />

</form>

</body>

</html>
```

min、max 和 step 属性

min、max 和 step 属性用于为包含数字或日期的 input 类型规定限定（约束）。

max 属性规定输入域所允许的最大值。

min 属性规定输入域所允许的最小值。

step 属性为输入域规定合法的数字间隔（如果 step="3"，则合法的数是 -3,0,3,6 等）。

注释：min、max 和 step 属性适用于以下类型的 <input> 标签：date pickers、number 以及 range。

下面的例子显示一个数字域，该域接受介于 0 到 10 之间的值，且步进为 3（即合法的值为 0、3、6 和 9）：

实例

Points: <input type="number" name="points" min="0" max="10" step="3" />

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<form action="/example/HTML 5/demo_form.asp" method="get">
```

```
Points: <input type="number" name="points" min="0" max="10" step="3"/>
```

```
<input type="submit" />
```

```
</form>
```

```
</body>
```

```
</html>
```

multiple 属性

multiple 属性规定输入域中可选择多个值。

注释: **multiple** 属性适用于以下类型的 **<input>** 标签: **email** 和 **file**。

实例

```
Select images: <input type="file" name="img" multiple="multiple" />
```

亲自试一试吧, 代码如下:

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<form action="/example/HTML 5/demo_form.asp" method="get">
```

Select images: <input type="file" name="img" multiple="multiple" />

<input type="submit" />

</form>

<p>当您浏览文件时，请试着选择多个文件。</p>

</body>

</html>

novalidate 属性

novalidate 属性规定在提交表单时不应该验证 **form** 或 **input** 域。

注释: **novalidate** 属性适用于 **<form>** 以及以下类型的 **<input>** 标签: **text**, **search**, **url**, **telephone**, **email**, **password**, **date pickers**, **range** 以及 **color**.

实例

<form action="demo_form.asp" method="get" novalidate="true">

E-mail: <input type="email" name="user_email" />

<input type="submit" />

</form>

亲自试一试吧，代码如下：

<!DOCTYPE HTML>

<html>

```
<body>
```

```
<form action="/example/HTML 5/demo_form.asp" method="get" novalidate="true">
```

```
E-mail: <input type="email" name="user_email" />
```

```
<input type="submit" />
```

```
</form>
```

```
</body>
```

```
</html>
```

pattern 属性

pattern 属性规定用于验证 **input** 域的模式（**pattern**）。

模式（**pattern**） 是正则表达式。您可以在我们的 **JavaScript** 教程中学习到有关正则表达式的内容。

注释：**pattern** 属性适用于以下类型的 **<input>** 标签：**text**, **search**, **url**, **telephone**, **email** 以及 **password**。

下面的例子显示了一个只能包含三个字母的文本域（不含数字及特殊字符）：

实例

Country code: <input type="text" name="country_code"

pattern="[A-z]{3}" title="Three letter country code" />

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<form action="/example/HTML 5/demo_form.asp" method="get">  
Country code: <input type="text" name="country_code" pattern="[A-z]{3}"  
title="Three letter country code" />  
<input type="submit" />  
</form>  
  
</body>  
</html>
```

placeholder 属性

placeholder 属性提供一种提示（hint），描述输入域所期待的值。

注释：placeholder 属性适用于以下类型的 <input> 标签：text, search, url, telephone, email 以及 password。

提示（hint）会在输入域为空时显示出现，会在输入域获得焦点时消失：

实例

```
<input type="search" name="user_search" placeholder="Search W3School" />
```

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>  
  
<html>  
  
<body>
```

```
<form action="/example/HTML 5/demo_form.asp" method="get">

<input type="search" name="user_search" placeholder="Search W3School" />

<input type="submit" />

</form>

</body>

</html>
```

required 属性

required 属性规定必须在提交之前填写输入域（不能为空）。

注释: **required** 属性适用于以下类型的 **<input>** 标签: **text**, **search**, **url**, **telephone**, **email**, **password**, **date pickers**, **number**, **checkbox**, **radio** 以及 **file**。

实例

Name: **<input type="text" name="usr_name" required="required" />**

亲自试一试吧，代码如下：

```
<!DOCTYPE HTML>

<html>

<body>

<form action="/example/HTML 5/demo_form.asp" method="get">

Name: <input type="text" name="usr_name" required="required" />

<input type="submit" />
```

```
</form>
```

```
</body>
```

```
</html>
```

HTML 5 参考手册

通过制定如何处理所有 HTML 元素以及如何从错误中恢复的精确规则，HTML 5 改进了互操作性，并减少了开发成本。

HTML 5 中的新特性包括了嵌入音频、视频和图形的功能，客户端数据存储，以及交互式文档。

HTML 5 还包含了新的元素，比如：<nav>, <header>, <footer> 以及 <figure> 等等。

HTML 5 工作组包括：AOL, Apple, Google, IBM, Microsoft, Mozilla, Nokia, Opera, 以及数百个其他的供应商。

注释：HTML 5 还没有成为 W3C 正式的推荐标准。

如需阅读更多有关 W3C HTML 5 活动的内容，请阅读我们的 W3C 教程。

按字母顺序排列的标签列表

4: 指示在 HTML 4.01 中定义了该元素

5: 指示在 HTML 5 中定义了该元素

标签	描述	4	5
<!--...-->	定义注释。	4	5
<!DOCTYPE>	定义文档类型。	4	5
<a>	定义超链接。	4	5
<abbr>	定义缩写。	4	5

<acronym>	HTML 5 中不支持。定义首字母缩写。	4	
<address>	定义地址元素。	4	5
<applet>	HTML 5 中不支持。定义 applet。	4	
<area>	定义图像映射中的区域。	4	5
<article>	定义 article。		5
<aside>	定义页面内容之外的内容。		5
<audio>	定义声音内容。		5
	定义粗体文本。	4	5
<base>	定义页面中所有链接的基准 URL。	4	5
<basefont>	HTML 5 中不支持。请使用 CSS 代替。	4	
<bdo>	定义文本显示的方向。	4	5
<big>	HTML 5 中不支持。定义大号文本。	4	
<blockquote>	定义长的引用。	4	5
<body>	定义 body 元素。	4	5
 	插入换行符。	4	5
<button>	定义按钮。	4	5
<canvas>	定义图形。		5

<caption>	定义表格标题。	4	5
<center>	HTML 5 中不支持。定义居中的文本。	4	
<cite>	定义引用。	4	5
<code>	定义计算机代码文本。	4	5
<col>	定义表格列的属性。	4	5
<colgroup>	定义表格列的分组。	4	5
<command>	定义命令按钮。		5
<datalist>	定义下拉列表。		5
<dd>	定义定义的描述。	4	5
	定义删除文本。	4	5
<details>	定义元素的细节。		5
<dfn>	定义定义项目。	4	5
<dir>	HTML 5 中不支持。定义目录列表。	4	
<div>	定义文档中的一个部分。	4	5
<dl>	定义定义列表。	4	5
<dt>	定义定义的项目。	4	5
	定义强调文本。	4	5

<embed>	定义外部交互内容或插件。		5
<fieldset>	定义 fieldset 。	4	5
<figcaption>	定义 figure 元素的标题。		5
<figure>	定义媒介内容的分组，以及它们的标题。		5
	HTML 5 中不支持。	4	
<footer>	定义 section 或 page 的页脚。		5
<form>	定义表单。	4	5
<frame>	HTML 5 中不支持。定义子窗口（框架）。	4	
<frameset>	HTML 5 中不支持。定义框架的集。	4	
<h1> to <h6>	定义标题 1 到标题 6。	4	5
<head>	定义关于文档的信息。	4	5
<header>	定义 section 或 page 的页眉。		5
<hgroup>	定义有关文档中的 section 的信息。		5
<hr>	定义水平线。	4	5
<html>	定义 html 文档。	4	5
<i>	定义斜体文本。	4	5
<iframe>	定义行内的子窗口（框架）。	4	5

	定义图像。	4	5
<input>	定义输入域。	4	5
<ins>	定义插入文本。	4	5
<keygen>	定义生成密钥。		5
<isindex>	HTML 5 中不支持。定义单行的输入域。	4	
<kbd>	定义键盘文本。	4	5
<label>	定义表单控件的标注。	4	5
<legend>	定义 fieldset 中的标题。	4	5
	定义列表的项目。	4	5
<link>	定义资源引用。	4	5
<map>	定义图像映射。	4	5
<mark>	定义有记号的文本。		5
<menu>	定义菜单列表。	4	5
<meta>	定义元信息。	4	5
<meter>	定义预定义范围内的度量。		5
<nav>	定义导航链接。		5
<noframes>	HTML 5 中不支持。定义 noframe 部分。	4	

<noscript>	定义 noscript 部分。	4	5
<object>	定义嵌入对象。	4	5
	定义有序列表。	4	5
<optgroup>	定义选项组。	4	5
<option>	定义下拉列表中的选项。	4	5
<output>	定义输出的一些类型。		5
<p>	定义段落。	4	5
<param>	为对象定义参数。	4	5
<pre>	定义预格式化文本。	4	5
<progress>	定义任何类型的任务的进度。		5
<q>	定义短的引用。	4	5
<rp>	定义若浏览器不支持 ruby 元素显示的内容。		5
<rt>	定义 ruby 注释的解释。		5
<ruby>	定义 ruby 注释。		5
<s>	HTML 5 中不支持。定义加删除线的文本。	4	
<samp>	定义样本计算机代码。	4	5
<script>	定义脚本。	4	5

<section>	定义 section。		5
<select>	定义可选列表。	4	5
<small>	定义小号文本。	4	5
<source>	定义媒介源。		5
	定义文档中的 section。	4	5
<strike>	HTML 5 中不支持。定义加删除线的文本。	4	
	定义强调文本。	4	5
<style>	定义样式定义。	4	5
<sub>	定义下标文本。	4	5
<summary>	定义 details 元素的标题。		5
<sup>	定义上标文本。	4	5
<table>	定义表格。	4	5
<tbody>	定义表格的主体。	4	5
<td>	定义表格单元。	4	5
<textarea>	定义 textarea。	4	5
<tfoot>	定义表格的脚注。	4	5
<th>	定义表头。	4	5

<thead>	定义表头。	4	5
<time>	定义日期/时间。		5
<title>	定义文档的标题。	4	5
<tr>	定义表格行。	4	5
<tt>	HTML 5 中不支持。定义打字机文本。	4	
<u>	HTML 5 中不支持。定义下划线文本。	4	
	定义无序列表。	4	5
<var>	定义变量。	4	5
<video>	定义视频。		5
<xmp>	HTML 5 中不支持。定义预格式文本。	4	

属性	值	描述
accesskey	character	规定访问元素的键盘快捷键
class	classname	规定元素的类名（用于规定样式表中的类）。
contenteditable	true false	规定是否允许用户编辑内容。
contextmenu	menu_id	规定元素的上下文菜单。
data-yourvalue	value	创作者定义的属性。 HTML 文档的创作者可以定义他们自己的属性。 必须以 "data-" 开头。
dir	ltr rtl	规定元素中内容的文本方向。
draggable	true false auto	规定是否允许用户拖动元素。
hidden	hidden	规定该元素是无关的。被隐藏的元素不会显示。
id	id	规定元素的唯一 ID。
item	empty url	用于组合元素。
itemprop	url group value	用于组合项目。
lang	language_code	规定元素中内容的语言代码。语言代码参考手册。 http://www.w3school.com.cn/tags/html_ref_language_codes.asp

spellcheck	true false	规定是否必须对元素进行拼写或语法检查。
style	style_definition	规定元素的行内样式。
subject	id	规定元素对应的项目。
tabindex	number	规定元素的 tab 键控制次序。
title	text	规定有关元素的额外信息。

HTML 5 标准属性

所有 HTML 5 标签均支持下面列出的属性，仅有少数例外。

HTML 5 标准属性

NEW: HTML 5 中新的标准属性。

注释: HTML 4.01 不再支持 **accesskey** 属性:

HTML 5 事件属性

标准事件属性

HTML 4 增加了通过事件触发浏览器中行为的能力，比如当用户点击某个元素时启动一段 **JavaScript**。

如果需要学习更多有关使用这些事件进行编程的内容，请学习我们的 **JavaScript** 教程 和 **DHTML** 教程。

下面的表格列出了可插入 **HTML 5** 元素中以定义事件行为的标准事件属性。

Window 事件属性 - Window Event Attributes

表单事件 - Form Events

键盘事件 - Keybord Events

鼠标事件 - Mouse Events

媒介事件 - Media Events

Window 事件属性

window 对象触发的事件。

适用于 <body> 标签:

属性	值	描述
onafterprint	script	在打印文档之后运行脚本
onbeforeprint	script	在文档打印之前运行脚本
onbeforeunload	script	在文档加载之前运行脚本
onblur	script	当窗口失去焦点时运行脚本
onerror	script	当错误发生时运行脚本
onfocus	script	当窗口获得焦点时运行脚本
onhaschange	script	当文档改变时运行脚本
onload	script	当文档加载时运行脚本
onmessage	script	当触发消息时运行脚本
onoffline	script	当文档离线时运行脚本
ononline	script	当文档上线时运行脚本
onpagehide	script	当窗口隐藏时运行脚本

onpageshow	script	当窗口可见时运行脚本
onpopstate	script	当窗口历史记录改变时运行脚本
onredo	script	当文档执行再执行操作（redo）时运行脚本
onresize	script	当调整窗口大小时运行脚本
onstorage	script	当文档加载加载时运行脚本
onundo	script	当文档执行撤销操作时运行脚本
onunload	script	当用户离开文档时运行脚本

表单事件

由 HTML 表单内部的动作触发的事件。

适用于所有 HTML 5 元素，不过最常用于表单元素中：

属性	值	描述
onblur	script	当元素失去焦点时运行脚本
onchange	script	当元素改变时运行脚本
oncontextmenu	script	当触发上下文菜单时运行脚本
onfocus	script	当元素获得焦点时运行脚本
onformchange	script	当表单改变时运行脚本
onforminput	script	当表单获得用户输入时运行脚本
oninput	script	当元素获得用户输入时运行脚本

oninvalid	script	当元素无效时运行脚本
onreset	script	当表单重置时运行脚本。HTML 5 不支持。
onselect	script	当选取元素时运行脚本
onsubmit	script	当提交表单时运行脚本

键盘事件

由键盘触发的事件。

适用于所有 HTML 5 元素：

属性	值	描述
onkeydown	script	当按下按键时运行脚本
onkeypress	script	当按下并松开按键时运行脚本
onkeyup	script	当松开按键时运行脚本

鼠标事件

由鼠标货相似的用户动作触发的事件。

适用于所有 HTML 5 元素：

属性	值	描述
onclick	script	当单击鼠标时运行脚本
ondblclick	script	当双击鼠标时运行脚本
ondrag	script	当拖动元素时运行脚本
ondragend	script	当拖动操作结束时运行脚本

ondragenter	script	当元素被拖动至有效的拖放目标时运行脚本
ondragleave	script	当元素离开有效拖放目标时运行脚本
ondragover	script	当元素被拖动至有效拖放目标上方时运行脚本
ondragstart	script	当拖动操作开始时运行脚本
ondrop	script	当被拖动元素正在被拖放时运行脚本
onmousedown	script	当按下鼠标按钮时运行脚本
onmousemove	script	当鼠标指针移动时运行脚本
onmouseout	script	当鼠标指针移出元素时运行脚本
onmouseover	script	当鼠标指针移至元素之上时运行脚本
onmouseup	script	当松开鼠标按钮时运行脚本
onmousewheel	script	当转动鼠标滚轮时运行脚本
onscroll	script	当滚动元素滚动元素的滚动条时运行脚本

媒介事件

由视频、图像以及音频等媒介触发的的事件。

适用于所有 HTML 5 元素，不过在媒介元素（诸如 **audio**、**embed**、**img**、**object** 以及 **video**）中最常用：

属性	值	描述
onabort	script	当发生中止事件时运行脚本
oncanplay	script	当媒介能够开始播放但可能因缓冲而需要停止时运行脚本

oncanplaythrough	script	当媒介能够无需因缓冲而停止即可播放至结尾时运行脚本
ondurationchange	script	当媒介长度改变时运行脚本
onemptied	script	当媒介资源元素突然为空时（网络错误、加载错误等）运行脚本
onended	script	当媒介已抵达结尾时运行脚本
onerror	script	当在元素加载期间发生错误时运行脚本
onloadeddata	script	当加载媒介数据时运行脚本
onloadedmetadata	script	当媒介元素的持续时间以及其他媒介数据已加载时运行脚本
onloadstart	script	当浏览器开始加载媒介数据时运行脚本
onpause	script	当媒介数据暂停时运行脚本
onplay	script	当媒介数据将要开始播放时运行脚本
onplaying	script	当媒介数据已开始播放时运行脚本
onprogress	script	当浏览器正在取媒介数据时运行脚本
onratechange	script	当媒介数据的播放速率改变时运行脚本
onreadystatechange	script	当就绪状态（ ready-state ）改变时运行脚本
onseeked	script	当媒介元素的定位属性 [1] 不再为真且定位已结束时运行脚本
onseeking	script	当媒介元素的定位属性为真且定位已开始时运行脚本
onstalled	script	当取回媒介数据过程中（延迟）存在错误时运行脚本

onsuspend	script	当浏览器已在取媒介数据但在取回整个媒介文件之前停止时运行脚本
ontimeupdate	script	当媒介改变其播放位置时运行脚本
onvolumechange	script	当媒介改变音量亦或当音量被设置为静音时运行脚本
onwaiting	script	当媒介已停止播放但打算继续播放时运行脚本

[1]: 定位属性的英文译文是：seeking attribute。

HTML 5 标签

<!--...--> 标签

定义和用法

注释标签用于在源文档中插入注释。注释会被浏览器忽略。您可使用注释对您的代码进行解释，这样做有助于您在以后的时间对代码的编辑。

您也可以在注释内容存储针对程序所定制的信息。在这种情况下，这些信息对用户是不可见的，但是对程序来说是可用的。一个好的习惯是把注释或样式元素放入注释文本中，这样就可避免不支持脚本或样式的老浏览器把它们显示为纯文本。

除了在源文档中有非常明显的作用外，许多 Web 服务器也利用注释来实现文档服务端软件特有的特性。这些服务器可以扫描文档，从传统的 HTML/XHTML 注释中找到特定的字符序列，然后再根据嵌在注释中的命令采取相应的动作。这些动作可能是简单的包括其他文件中的文本（即所谓的服务器端包含，server-inside include），也可能是复杂地执行其他命令去动态生成文档的内容。

HTML 4.01 与 HTML 5 之间的差异

NONE

例子：

<!--This text is a comment-->

<p>This is a regular paragraph</p>

属性：无

TIY 实例

隐藏的注释

本例演示如何在 HTML 源代码中插入隐藏的注释。

代码如下：

```
<html>

<body>

<!--这是一段注释。注释不会在浏览器中显示。-->

<p>这是一段普通的段落。</p>

</body>

</html>
```

<!DOCTYPE> 标签

定义和用法

<!DOCTYPE> 声明必须位于 HTML 5 文档中的第一行，也就是位于 <html> 标签之前。该标签告知浏览器文档所使用的 HTML 规范。

doctype 声明不属于 HTML 标签；tag；它是一条指令，告诉浏览器编写页面所用的标记的版本。

在所有 HTML 文档中规定 doctype 是非常重要的，这样浏览器就能了解预期的文档类型。

HTML 4.01 中的 doctype 需要对 DTD 进行引用，因为 HTML 4.01 基于 SGML。而 HTML 5 不基于 SGML，因此不需要对 DTD 进行引用，但是需要 doctype 来规范浏览器的行为（让浏览器按照它们应该的方式来运行。）。

实例

具有正确文档类型（doctype）的 HTML 5 文档：

```
<!DOCTYPE HTML>

<html>
```

```
<head>
```

```
<title>Title of the document</title>
```

```
</head>
```

```
<body>
```

The content of the document.....

```
</body>
```

```
</html>
```

亲自试一试下面代码吧。

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
<title>Title of the document</title>
```

```
</head>
```

```
<body>
```

The content of the document.....

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

在 HTML 4.01 中有 3 个不同的文档类型，在 HTML 5 中只有一个：

`<!DOCTYPE HTML>`

在 HTML 5 中，文档类型声明是不是很简单。

提示和注释：

注释： `<!DOCTYPE>` 标签没有结束标签！

提示： `<!DOCTYPE>` 对大小写不敏感。

属性

NONE

定义和用法

`<a>` 标签定义超链接，用于从一个页面链接到另一个页面。

`<a>` 元素最重要的属性是 `href` 属性，它指定链接的目标。

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

实例

链接到 Sina:

```
<a href="http://www.sina.com.cn">Sina</a>
```

亲自试一试

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<a href="http://www.sina.com.cn">Visit sina.com.cn!</a>
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

在 HTML 4.01 中，<a> 标签既可以是超链接，也可以是锚。这取决于是否描述了 href 属性。

在 HTML 5 中，<a> 是超链接，但是假如没有 href 属性，它仅仅是超链接的一个占位符。

HTML 5 有一些新的属性，同时不再支持一些 HTML 4.01 的属性。

提示和注释

提示：如果没有使用 href 属性，则不能使用 hreflang, media, ping, rel, target 以及 type 属性。

提示：通常在当前浏览器窗口中显示被链接页面，除非规定了其他 target。

提示：请使用 CSS 来改变链接的样式。

属性

属性	值	描述
charset	char_encoding	HTML 5 中不支持。
coords	coordinates	HTML 5 中不支持。
href	URL	链接的目标 URL。
hreflang	language_code	规定目标 URL 的基准语言。仅在 href 属性存在时使用。
media	media query	规定目标 URL 的媒介类型。 默认值：all。仅在 href 属性存在时使用。
name	section_name	HTML 5 中不支持。
ping	URL	由空格分隔的 URL 列表。 当用户点击该链接时，这些 URL 会获得通知。 仅在 href 属性存在时使用。

rel	alternate	规定当前文档与目标 URL 之间的关系。
	archives	仅在 href 属性存在时使用。
	author	
	bookmark	
	contact	
	external	
	first	
	help	
	icon	
	index	
	last	
	license	
	next	
	nofollow	
	noreferrer	
	pingback	
	prefetch	
	prev	
	search	
	stylesheet	
	sidebar	
	tag	

	up	
rev	text	HTML 5 中不支持。
shape	default rect circle poly	HTML 5 中不支持。
target	_blank _parent _self _top	在何处打开目标 URL。仅在 href 属性存在时使用。
type	mime_type	规定目标 URL 的 MIME 类型。仅在 href 属性存在时使用。 注：MIME = Multipurpose Internet Mail Extensions。

标准属性

<a> 标签支持 HTML 5 中的标准属性。

事件属性

<a> 标签支持 HTML 5 中的事件属性。

<abbr> 标签

定义和用法

表示一个缩写形式，比如 "Inc."、"etc."。通过对缩写词语进行标记，您就能够为浏览器、拼写检查程序、翻译系统以及搜索引擎分度器提供有用的信息。

<abbr> 标签最初是在 HTML 4.0 中引入的，表示它所包含的文本是一个更长的单词或短语的缩写形式。

浏览器可能会根据这个信息改变对这些文本的显示方式，或者用其他文本代替。也许当前流行的浏览器对 `<abbr>` 标签中包含的文本还没有任何处理方式，但是我们无法预测将来的版本将会如何实现该标签。

实例

The `<abbr title="People's Republic of China">PRC</abbr>` was founded in 1948.

输出：

etc.

亲自试一试下面代码吧！

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
The <abbr title="People's Republic of China">PRC</abbr> was founded in 1949.
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

NONE

提示和注释：

注释：在某些浏览器中，当您把鼠标移至缩略词语上时，**title** 可用于展示表达的完整版本。

标准属性

`<abbr>` 标签支持 HTML 5 中的标准属性。

事件属性

`<abbr>` 标签支持 HTML 5 中的事件属性。

`<acronym>` 标签

定义和用法

<acronym> 标签定义首字母缩写词，比如 "NASA"。通过对只取首字母缩写进行标记，您就能够为浏览器、拼写检查程序、翻译系统以及搜索引擎分度器提供有用的信息。

HTML 4.01 与 HTML 5 之间的差异

HTML 5 不支持 **<acronym>**。请使用 **<abbr>** 代替。

例子：

```
<acronym title="World Wide Web">WWW</acronym>
```

输出：

WWW

提示和注释：

注释：在某些浏览器中，当您把鼠标移至缩略词语上时，**title** 可被用来展示表达的完整版本。

<address> 标签

定义和用法

<address> 标签定义文档作者或拥有者的联系信息。

如果 **<address>** 元素位于 **<article>** 元素内部，则它表示该文章作者或拥有者的联系信息。

实例

```
<address>
```

```
Written by W3School.com.cn<br />
```

```
<a href="mailto:us@example.org">Email us</a><br />
```

```
Address: Box 564, Disneyland<br />
```

```
Phone: +12 34 56 78
```

```
</address>
```

亲自试一试,代码如下:

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<address>
```

```
Written by W3School.com.cn<br />
```

```
<a href="mailto:us@example.org">Email us</a><br />
```

```
Address: Box 564, Disneyland<br />
```

```
Phone: +12 34 56 78
```

```
</address>
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

HTML 4.01 不支持 `<article>` 标签,因此在 HTML 4.01 中 `<address>` 标签永远定义文档作者或拥有者的联系信息。

提示和注释:

注释: 不应该使用 `<address>` 标签来描述邮政地址,除非这些信息是联系信息的组成部分。

注释: `<address>` 元素通常呈现为斜体。大多数浏览器会在该元素的前后添加换行。

标准属性

`<address>` 标签支持 HTML 5 中的标准属性。

事件属性

`<address>` 标签支持 HTML 5 中的事件属性。

<applet> 标签

定义和用法

<applet> 定义嵌入的 applet。

HTML 4.01 与 HTML 5 之间的差异

HTML 4.01 中不赞成使用 applet 元素。

HTML 5 中不支持 applet 元素。

提示和注释：

提示：HTML 5 中不支持 applet 元素。请使用 <object> 代替。

<area> 标签

定义和用法

<area> 标签定义图像映射中的区域。

HTML 4.01 与 HTML 5 之间的差异

HTML 5 有一些新的属性，且不再支持 HTML 4.01 的一些属性。

例子：

```
<img src ="planets.gif" alt="Planets" usemap ="#planetmap" />
```

```
<map id ="planetmap">
```

```
  <area shape ="rect" coords ="0,0,82,126" href ="sun.htm" alt="Sun" />
```

```
  <area shape ="circle" coords ="90,58,3" href ="mercur.htm" alt="Mercury" />
```

```
  <area shape ="circle" coords ="124,58,8" href ="venus.htm" alt="Venus" />
```

```
</map>
```

属性

属性	描述	4	5
alt	定义此区域的替换文本。如果 href 存在，则该属性是必需的。 仅在 href 属性存在时使用。	4	5
coords	定义可点击区域的坐标。	4	5
href	定义此区域的目标 URL。	4	5
hreflang	规定目标 URL 的基准语言。仅在 href 属性存在时使用。	4	5
nohref	不赞成。从图像映射排除一个区域。	4	
media	规定目标 URL 的媒介类型。默认值：all。仅在 href 属性存在时使用。		5
ping	由空格分隔的 URL 列表，当用户点击该链接时，这些 URL 会获得通知。 仅在 href 属性存在时使用。		5
rel	规定当前文档与目标 URL 之间的关系。仅在 href 属性存在时使用。		5
shape	规定区域的形状。	4	5
target	在何处打开目标 URL。	4	5
type	规定目标 URL 的 MIME 类型。仅在 href 属性存在时使用。		5

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant,

lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu,

ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover,

ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup,

onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout,

onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访 [HTML 5 中事件属性](#)。

TIY

创建图像映射，代码如下：

```
<html>

<body>

<p>请点击图像上的星球，把它们放大。</p>



<map name="planetmap" id="planetmap">

<area

shape="circle"

coords="180,139,14"

href = "/example/html/venus.html"

target = "_blank"

alt="Venus" />

<area

shape="circle"

coords="129,161,10"

href = "/example/html/mercur.html"

target = "_blank"
```

```
alt="Mercury" />
```

```
<area
```

```
shape="rect"
```

```
coords="0,0,110,260"
```

```
href = "/example/html/sun.html"
```

```
target = "_blank"
```

```
alt="Sun" />
```

```
</map>
```

```
<p><b>注释: </b>img 元素中的 "usemap" 属性引用 map 元素中的 "id" 或 "name" 属性（根据浏览器），所以我们同时向 map 元素添加了 "id" 和 "name" 属性。</p>
```

```
</body>
```

```
</html>
```

本例显示如何创建带有可供点击区域的图像映射。其中的每个区域都是一个超级链接。

<article> 标签

定义和用法

<article> 标签定义外部的内容。

外部内容可以是来自一个外部的新闻提供者的一篇新的文章，或者来自 **blog** 的文本，或者是来自论坛的文本。亦或是来自其他外部源内容。

实例

```
<article>
```

```
<a href="http://www.apple.com">Safari 5 released</a><br />
```

```
7 Jun 2010. Just after the announcement of the new iPhone 4 at WWDC,
```

```
Apple announced the release of Safari 5 for Windows and Mac.....
```

```
</article>
```

亲自试一试，代码如下：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<article>
```

```
<a href="http://www.apple.com">Safari 5 released</a><br />
```

```
7 Jun 2010. Just after the announcement of the new iPhone 4 at WWDC,
```

```
Apple announced the release of Safari 5 for Windows and Mac.....
```

```
</article>
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

`<article>` 标签是 HTML 5 的新标签。

提示和注释：

注释： `<article>` 标签的内容独立于文档的其余部分。

标准属性

`<article>` 标签支持 HTML 5 中的标准属性。

事件属性

`<article>` 标签支持 HTML 5 中的事件属性。

<aside> 标签

定义和用法

<aside> 标签定义 **article** 以外的内容。**aside** 的内容应该与 **article** 的内容相关。

实例

```
<p>Me and my family visited The Epcot center this summer.</p>
```

```
<aside>
```

```
<h4>Epcot Center</h4>
```

The Epcot Center is a theme park in Disney World, Florida.

```
</aside>
```

HTML 4.01 与 HTML 5 之间的差异

<aside> 标签是 HTML 5 的新标签。

提示和注释：

提示：<aside> 的内容可用作文章的侧栏。

标准属性

<article> 标签支持 HTML 5 中的标准属性。

事件属性

<article> 标签支持 HTML 5 中的事件属性。

<audio> 标签

定义和用法

<audio> 标签定义声音，比如音乐或其他音频流。

实例

一段简单的 HTML 5 音频：

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

您的浏览器不支持 `audio` 标签。

`</audio>`

HTML 4.01 与 HTML 5 之间的差异

`<audio>` 标签是 HTML 5 的新标签。

提示和注释

提示：可以在开始标签和结束标签之间放置文本内容，这样老的浏览器就可以显示出不支持该标签的信息。

属性

属性	值	描述
<code>autoplay</code>	<code>autoplay</code>	如果出现该属性，则音频在就绪后马上播放。
<code>controls</code>	<code>controls</code>	如果出现该属性，则向用户显示控件，比如播放按钮。
<code>loop</code>	<code>loop</code>	如果出现该属性，则每当音频结束时重新开始播放。
<code>preload</code>	<code>preload</code>	如果出现该属性，则音频在页面加载时进行加载，并预备播放。 如果使用 <code>"autoplay"</code> ，则忽略该属性。
<code>src</code>	<code>url</code>	要播放的音频的 URL。

`<b>` 标签

定义和用法

`<b>` 标签定义粗体的文本。

`<b>` 标签定义了文本中的部分比其余的部分更重要，并呈现为粗体。

实例

The house with the `<b>red</b>` door

HTML 4.01 与 HTML 5 之间的差异

没有区别。

但是在 HTML 4.01 中，我们的理解是使用 CSS 来制作粗体的文本。在 HTML 5 中，我们仍然可以这么理解，不过我们应该使用 `<b>` 标签定义文本中更重要的部分。

`<base>` 标签

定义和用法

`<base>` 标签为页面上的所有链接规定默认地址或默认目标。

通常情况下，浏览器会从当前文档的 URL 中提取相应的元素来填写相对 URL 中的空白。

使用 `<base>` 标签可以改变这一点。浏览器随后将不再使用当前文档的 URL，而使用指定的基本 URL 来解析所有的相对 URL。这其中包括 `<a>`、`<img>`、`<link>`、`<form>` 标签中的 URL。

`<base>` 标签必须位于 `head` 元素内部。

实例

例子 1

假设图像的绝对地址是：

```

```

现在我们在页面中的 `head` 部分插入 `<base>` 标签，规定页面中所有链接的基准 url：

```
<head>
```

```
<base href="http://www.w3school.com.cn/i/" />
```

```
</head>
```

在上例中的页面上插入图像时，我们必须规定相对的地址，浏览器会寻找文件所使用的完整 URL：

```

```

例子 2

规定页面上所有链接的默认 URL 和默认目标：

```
<head>
```

```
<base href="http://www.w3school.com.cn/css/" target="_blank" />
```

```
</head>
```

```
<body>

<a href="default.asp">W3School's CSS Tutorial</a>

</body>
```

亲自试一试

HTML 4.01 与 HTML 5 之间的差异

None

提示和注释

提示：请把 **<base>** 标签排在 **head** 元素中第一个元素的位置，这样 **head** 中其他元素就可以利用 **<base>** 元素中的信息了。

注释：在一个文档中，最多能使用一个 **<base>** 元素。

属性

属性	值	描述
href	URL	规定作为基准 URL 在页面中使用的 URL。
target	_blank _parent _self _top	在何处打开页面上的链接。该属性会被每个链接中的 target 属性覆盖。

标准属性和事件

没有。

<basefont> 标签

定义和用法

<basefont> 标签定义基准字体。

HTML 4.01 与 HTML 5 之间的差异

HTML 5 不支持 `<basefont>` 标签。请用 CSS 代替。

`<bdo>` 标签

定义和用法

`<bdo>` 标签覆盖默认的文本方向。

实例

```
<bdo dir="rtl">Here is some text</bdo>
```

亲自试一试

HTML 4.01 与 HTML 5 之间的差异

NONE

属性

属性	值	描述
dir	ltr	必需。定义文本方向。
	rtl	

标准属性

`<bdo>` 标签支持 HTML 5 中的标准属性。

事件属性

`<bdo>` 标签支持 HTML 5 中的事件属性。

`<big>` 标签

定义和用法

`<big>` 标签制作更大的文本。

HTML 4.01 与 HTML 5 之间的差异

HTML 5 不支持 `<big>` 标签。请用 CSS 代替。

<blockquote> 标签

定义和用法

<blockquote> 标签定义摘自另一个源的块引用。

<blockquote> 与 </blockquote> 之间的所有文本都会从常规文本中分离出来，经常会在左、右两边进行缩进，而且有时会使用斜体。也就是说，块引用拥有它们自己的空间。

实例

毛泽东说过：

<blockquote>帝国主义都是纸老虎 ... </blockquote>

亲自试一试

HTML 4.01 与 HTML 5 之间的差异

无。

提示和注释

提示：如果标记是不需要段落分隔的短引用，请使用 <q> 元素。

注释：blockquote 只应该应用于来自另一个源的引用。

属性

属性	值	描述
cite	URL	引用的 URL，如果引用来自网络的话。

标准属性

<blockquote> 标签支持 HTML 5 中的标准属性。

事件属性

<blockquote> 标签支持 HTML 5 中的事件属性。

<body> 标签

定义和用法

<body> 元素定义文档的主体。

<body> 元素包含文档的所有内容，比如文本、超链接、图像、表格、列表等等。

实例

一个简单的 HTML 文档，拥有尽可能少的必需的标签：

```
<html>
```

```
<head>
```

```
<title>文档的标题</title>
```

```
</head>
```

```
<body>
```

文档的内容

```
</body>
```

```
</html>
```

亲自试一试

HTML 4.01 与 HTML 5 之间的差异

在 HTML 5 中，删除了所有 **<body>** 的特殊属性。而那些属性在 HTML 4.01 中是不被赞成使用的。

标准属性

<body> 标签支持 HTML 5 中的标准属性。

事件属性

<body> 标签支持 HTML 5 中的事件属性。

**
** 标签

定义和用法

`<br />` 标签插入简单的换行符。

`<br />` 标签是一个空标签，意味着它没有结束标签。所以这样是错误的：`<br></br>`

实例

This text contains`<br />`a line break

输出：

This text contains

a line break

亲自试一试

HTML 4.01 与 HTML 5 之间的差异

无。

提示和注释

提示：请使用 `<br />` 标签来输入空行，而不是分隔段落。

标准属性

`<br />` 标签支持 HTML 5 中的标准属性。

事件属性

`<br />` 标签支持 HTML 5 中的事件属性。

`<button>` 标签

定义和用法

`<button>` 标签定义按钮。

您可以在 `button` 元素放置内容，比如文本或图像。这是该元素与通过 `input` 元素创建的按钮的不同之处。

请始终为按钮规定 `type` 属性。不同的浏览器根据 `type` 属性使用不同的默认值。

实例

标记一个按钮：

```
<button type="button">Click Me!</button>
```

亲自试一试

HTML 4.01 与 HTML 5 之间的差异

HTML 5 中的新属性: autofocus, form, formaction, formenctype, formmethod, formnovalidate 以及 formtarget。

提示和注释

注释: 如果在 HTML 表单中使用 button 元素, 不同的浏览器会提交不同的按钮值。请使用 input 元素在 HTML 表单中创建按钮。

属性

属性	值	描述
autofocusNew	autofocus	如果设置, 则当页面加载后使按钮获得焦点。
disabled	disabled	禁用按钮。
formNew	form_name	规定按钮属于哪个表单。
formactionNew	url	规定当提交表单时向何处提交表单数据。 覆盖表单的 action 属性。
formenctypeNew	见注释	规定如何在表单数据发送到服务器之前如何进行编码。 覆盖表单的 enctype 属性。
formmethodNew	delete get post put	规定如何发送表单数据。 覆盖表单的 method 属性。
formnovalidateNew	formnovalidate	如果设置, 指示是否在提交时验证表单。 覆盖表单的 novalidate 属性。
formtargetNew	_blank _self	规定在何处打开 action 中的 URL。 覆盖表单的 target 属性。

	_parent _top frameName	
name	button_name	规定按钮的名称。
type	button reset submit	定义按钮的类型。
value	some_value	规定按钮的初始值。可由脚本进行修改。

注释：formenctypeNew 属性可能的值：

application/x-www-form-urlencoded

multipart/form-data

text/plain

标准属性

<button> 标签支持 HTML 5 中的标准属性。

事件属性

<button> 标签支持 HTML 5 中的事件属性。

<canvas> 标签

定义和用法

<canvas> 标签定义图形，比如图表和其他图像。

实例

如何通过 canvas 元素来显示一个红色的矩形：

<canvas id="myCanvas"></canvas>

```
<script type="text/javascript">
```

```
var canvas=document.getElementById('myCanvas');
```

```
var ctx=canvas.getContext('2d');
```

```
ctx.fillStyle='#FF0000';
```

```
ctx.fillRect(0,0,80,100);
```

```
</script>
```

TIY

HTML 4.01 与 HTML 5 之间的差异

<canvas> 标签是 HTML 5 中的新标签。

<canvas> 的历史

这个 HTML 元素是为了客户端矢量图形而设计的。它自己没有行为，但却把一个绘图 API 展现给客户端 JavaScript 以使脚本能够把想绘制的东西都绘制到一块画布上。

<canvas> 标记由 Apple 在 Safari 1.3 Web 浏览器中引入。对 HTML 的这一根本扩展的原因在于，HTML 在 Safari 中的绘图能力也为 Mac OS X 桌面的 Dashboard 组件所使用，并且 Apple 希望有一种方式在 Dashboard 中支持脚本化的图形。

Firefox 1.5 和 Opera 9 都跟随了 Safari 的引领。这两个浏览器都支持 **<canvas>** 标记。

我们甚至可以在 IE 中使用 **<canvas>** 标记，并在 IE 的 VML 支持的基础上用开源的 JavaScript 代码（由 Google 发起）来构建兼容性的画布。参见：<http://excanvas.sourceforge.net/>。

<canvas> 的标准化的努力由一个 Web 浏览器厂商的非正式协会在推进，目前 **<canvas>** 已经成为 HTML 5 草案中一个正式的标签。参见：<http://www.whatwg.org/specs/web-apps/current-work/>

<canvas> 标记和 SVG 以及 VML 之间的差异

<canvas> 标记和 SVG 以及 VML 之间的一个重要的不同是，**<canvas>** 有一个基于 JavaScript 的绘图 API，而 SVG 和 VML 使用一个 XML 文档来描述绘图。

这两种方式在功能上是等同的，任何一种都可以用另一种来模拟。从表面上看，它们很不相同，可是，每一种都有强项和弱点。例如，SVG 绘图很容易编辑，只要从其描述中移除元素就行。

要从同一图形的一个 `<canvas>` 标记中移除元素，往往需要擦掉绘图重新绘制它。

如何使用 `<canvas>` 标记绘图

大多数 Canvas 绘图 API 都没有定义在 `<canvas>` 元素本身上，而是定义在通过画布的 `getContext()` 方法获得的一个“绘图环境”对象上。

Canvas API 也使用了路径的表示法。但是，路径由一系列的方法调用来定义，而不是描述为字母和数字的字符串，比如调用 `beginPath()` 和 `arc()` 方法。

一旦定义了路径，其他的方法，如 `fill()`，都是对此路径操作。绘图环境的各种属性，比如 `fillStyle`，说明了这些操作如何使用。

注释: Canvas API 非常紧凑的一个原因上它没有对绘制文本提供任何支持。要把文本加入到一个 `<canvas>` 图形，必须要么自己绘制它再用位图图像合并它，或者在 `<canvas>` 上方使用 CSS 定位来覆盖 HTML 文本。

属性

属性	值	描述
height	pixels	设置 canvas 的高度。
width	pixels	设置 canvas 的宽度。

标准属性

`<canvas>` 标签支持 HTML 5 中的标准属性。

事件属性

`<canvas>` 标签支持 HTML 5 中的事件属性。

参阅

HTML DOM Canvas 对象

`<caption>` 标签

定义和用法

`<caption>` 标签定义表格标题。`<caption>` 标签必须紧随 `<table>` 标签之后。

HTML 4.01 与 HTML 5 之间的差异

HTML 4.01 不赞成使用 align 属性。在 HTML 5 中，不支持该标签。请使用 CSS 代替。

例子：

```
<table>

  <caption>This is a caption</caption>

  <tr>

    <td>Cell 1</td>

    <td>Cell 2</td>

  </tr>

</table>
```

属性

属性	值	描述	4	5
align	left right top bottom	如何对齐标题。不支持。请使用样式代替。	4	

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant,
lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访 HTML 5 中标准属性。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu,
ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover,
ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup,

onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout,
onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访 HTML 5 中事件属性。

<center> 标签

定义和用法

<center> 标签对其包围的文本进行水平居中处理。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 5 中，不支持该标签。请使用 CSS 代替。

提示和注释

提示：请使用 CSS 来居中文本。

 <dfn> <code> <samp> <kbd> <var> <cite> 标签

定义和用法

以下元素都是短语元素。并不返回使用它们，但是通过使用样式表，可能取得更丰富的效果。

	呈现为被强调的文本。
	定义重要的文本。
<dfn>	定义一个定义项目。
<code>	定义计算机代码文本。
<samp>	定义样本文本。
<kbd>	定义键盘文本。它表示文本是从键盘上键入的。它经常用在与计算机相关的文档或手册中。
<var>	定义变量。您可以将此标签与 <pre> 及 <code> 标签配合使用。
<cite>	定义引用。可使用该标签对参考文献的引用进行定义，比如书籍或杂志的标题。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 4.01 中， 定义语气更重的强调文本，但是在 HTML 5 中， 定义重要的文本。

属性

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant,

lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访 HTML 5 中标准属性。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu,

ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover,

ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup,

onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout,

onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访 HTML 5 中事件属性。

HTML 文本格式化实例

文本格式化

此例演示如何在一个 HTML 文件中对文本进行格式化

“计算机输出” 标签

此例演示不同的“计算机输出”标签的显示效果。

<col> 标签

定义和用法

<col> 标签为表格中的一个或多个列定义属性值。您只能在表格或列组中使用该元素。

HTML 4.01 与 HTML 5 之间的差异

4.01 中的大多数属性都不再得到支持。

提示和注释

注释：col 元素是空元素。如需创建列，必须在 tr 元素内规定 td 元素。

提示：如果需要为一个或多个列规定属性值，请使用该元素。

提示：如果需要向一个列组规定相同的属性值，请使用 <colgroup> 元素。

例子：

```
<table>

<col style="color:blue" />

<col span="2" style="color:yellow" />

<tr>

  <td>1</td>

  <td>2</td>

  <td>3</td>

  <td>4</td>

</tr>

<tr>

  <td>1</td>

  <td>2</td>

  <td>3</td>

  <td>4</td>

</tr>

</table>
```

属性

属性	值	描述	4	5
align	right	定义列中的单元格内容的水平对齐方式。	4	
	left			

	center justify char			
char	character	定义一个字符，按照该字符对齐文本。 与 align="char" 使用。	4	
charoff	pixels %	规定第一个对齐字符的偏移量。	4	
span	number	定义 <col> 应当横跨的列数。	4	5
valign	top middle bottom baseline	定义列中的单元格内容的垂直对齐方式。	4	
width	% pixels relative_length	定义列的宽度。	4	

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant,
lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访 HTML 5 中标准属性。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu,
ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover,
ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup,
onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout,

onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访 [HTML 5 中事件属性](#)。

<colgroup> 标签

定义和用法

<colgroup> 标签定义表格列的组。通过此标签，您可以对列进行组合，以便格式化。该元素只有在 <table> 中才是合法的。

HTML 4.01 与 HTML 5 之间的差异

4.01 中的大多数属性都不再得到支持。

提示和注释

注释：colgroup 元素只能包含 col 元素。

注释：colgroup 元素无法创建表格列。如需创建列，必须在 tr 元素内规定 td 元素。

提示：如果需要向一个列组规定相同的属性值，请使用该元素。

提示：如果需要为一个或多个列规定属性值，请使用 col 元素。

例子：

<table>

```
<colgroup span="3" style="color:blue"></colgroup>
```

```
<tr>
```

```
<td>1</td>
```

```
<td>2</td>
```

```
<td>3</td>
```

```
<td>4</td>
```

```
</tr>
```

```
<tr>
```

```
<td>1</td>
```

```
<td>2</td>

<td>3</td>

<td>4</td>

</tr>

</table>
```

属性

属性	值	描述	4	5
align	right left center justify char	定义列中的单元格内容的水平对齐方式。	4	
char	character	定义一个字符，按照该字符对齐文本。	4	
charoff	pixels %	规定第一个对齐字符的偏移量。	4	
span	number	定义 <colgroup> 应当横跨的列数。	4	5
valign	top middle bottom baseline	定义列中的单元格内容的垂直对齐方式。	4	
width	% pixels relative_length	定义列的宽度。	4	

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant,
lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访 HTML 5 中标准属性。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu,
ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover,
ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup,
onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout,
onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访 HTML 5 中事件属性。

<command> 标签

定义和用法

<command> 标签定义命令按钮，比如单选按钮、复选框或按钮。

实例

标记一个按钮：

```
<menu>  
  
<command type="command">Click Me!</command>  
  
</menu>
```

亲自试一试

HTML 4.01 与 HTML 5 之间的差异

<command> 标签是 HTML 5 中的新标签。

属性

属性	值	描述
----	---	----

checked	checked	定义是否被选中。仅用于 radio 或 checkbox 类型。
disabled	disabled	定义 command 是否可用。
icon	url	定义作为 command 来显示的图像的 url。
label	command name	为 command 定义名称。label 是可见的。
radiogroup	name of radiogroup	定义该 command 所属的 radiogroup 的名称。 仅在类型为 radio 时使用。
type	checkbox command radio	定义该 command 的类型。默认是 command。

标准属性

<command> 标签支持 HTML 5 中的标准属性。

事件属性

<command> 标签支持 HTML 5 中的事件属性。

<datalist> 标签

定义和用法

<datalist> 标签定义选项列表。请与 input 元素配合使用该元素，来定义 input 可能的值。

datalist 及其选项不会被想显示出来，它仅仅是合法的输入值列表。

请使用 input 元素的 list 属性来绑定 datalist。

实例

下面是一个 input 元素，datalist 中描述了其可能的值：

```
<input id="myCar" list="cars" />

<datalist id="cars">

  <option value="BMW">

  <option value="Ford">
```

```
<option value="Volvo">
```

```
</datalist>
```

亲自试一试

HTML 4.01 与 HTML 5 之间的差异

<datalist> 标签是 HTML 5 中的新标签。

标准属性

<datalist> 标签支持 HTML 5 中的标准属性。

事件属性

<datalist> 标签支持 HTML 5 中的事件属性。

<dd> 标签

定义和用法

<dd> 标签定义一个定义列表中对项目的描述。

HTML 4.01 与 HTML 5 之间的差异

无。

```
<dl>
```

```
<dt>Coffee</dt>
```

```
<dd>Black hot drink</dd>
```

```
<dt>Milk</dt>
```

```
<dd>White cold drink</dd>
```

```
</dl>
```

属性

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant,

lang, ref, registrationmark, tabindex, template, title

如需完整的描述, 请访 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu,

ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover,

ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup,

onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout,

onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述, 请访 [HTML 5 中事件属性](#)。

TIY 实例

定义列表

```
<html>
```

```
<body>
```

```
<h2>一个定义列表: </h2>
```

```
<dl>
```

```
<dt>计算机</dt>
```

```
<dd>用来计算的仪器 ... ..</dd>
```

```
<dt>显示器</dt>
```

```
<dd>以视觉方式显示信息的装置 ... ..</dd>
```

```
</dl>
```

```
</body>
```

```
</html>
```

本例演示一个定义列表。

 标签

定义和用法

 标签定义文档中已删除的文本。

实例

a dozen is 21 12 pieces

亲自试一试

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<p>a dozen is <del>21</del> 12 pieces</p>
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

无。

提示和注释

提示：与 <ins> 一起使用，描述文档中的更新和修正。

属性

属性	值	描述
cite	URL	定义指向另一文档的 URL，该文档解释了文本被删除的原因。
datetime	YYYY-MM-DDThh:mm:ssTZD	定义文本被删除的日期和时间。

标准属性

 标签支持 HTML 5 中的标准属性。

事件属性

 标签支持 HTML 5 中的事件属性。

<details> 标签

定义和用法

<details> 标签用于描述文档或文档某个部分的细节。

实例

关于文档的细节：

```
<details>This document was written in 2010.</details>
```

亲自试一试

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<details>This document was written in 2010.</details>
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

<details> 标签是 HTML 5 中的新标签。

提示和注释

提示：与 <summary> 标签配合使用可以为 details 定义标题。标题是可见的，用户点击标题时，会显示出 details。

注释：details 元素可用于描述文档的任何部分。

属性

属性	值	描述
open	open	定义 details 是否可见。

标准属性

<details> 标签支持 HTML 5 中的标准属性。

事件属性

<details> 标签支持 HTML 5 中的事件属性。

<dir> 标签

定义和用法

<dir> 标签定义目录列表。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 4.01 中，不赞成使用 <dir> 标签。

HTML 5 不支持 <dir> 标签。请用 CSS 代替。

<div> 标签

定义和用法

<div> 标签定义文档中的 division/section。

HTML 4.01 与 HTML 5 之间的差异

HTML 4.01 中的 “align” 是不赞成使用的，在 HTML 5 中不支持该属性。

提示和注释

注释：浏览器通常会在 div 元素之前和之后插入换行符。

提示：请使用 <div> 标签对块级元素进行组合，这样就可以使用样式对它们进行格式化。

属性

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant,

lang, ref, registrationmark, tabindex, template, title

如需完整的描述, 请访 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu,

ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover,

ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup,

onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout,

onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述, 请访 [HTML 5 中事件属性](#)。

<dl> 标签

定义和用法

<dl> 标签定义一个定义列表。

HTML 4.01 与 HTML 5 之间的差异

无。

<dl>

 <dt>Coffee</dt>

 <dd>Black hot drink</dd>

 <dt>Milk</dt>

 <dd>White cold drink</dd>

</dl>

属性

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant,

lang, ref, registrationmark, tabindex, template, title

如需完整的描述, 请访 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu,

ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover,

ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup,

onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout,

onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述, 请访 [HTML 5 中事件属性](#)。

TIY 实例, 定义列表

如何创建定义列表。

```
<html>
```

```
<body>
```

```
<h2>一个定义列表: </h2>
```

```
<dl>
```

```
<dt>计算机</dt>
```

```
<dd>用来计算的仪器 ... ..</dd>
```

```
<dt>显示器</dt>
```

```
<dd>以视觉方式显示信息的装置 ... ..</dd>
```

```
</dl>
```

```
</body>
```

```
</html>
```

<dt> 标签

定义和用法

<dt> 标签定义一个定义列表中的一个项目，以及 dialog 中的角色。

HTML 4.01 与 HTML 5 之间的差异

HTML 5 引入了 <dialog> 标签。在 dialog 中，<dt> 标签定义句子所属的角色。参见例子 2。

例子

例子 1

```
<dl>

  <dt>Coffee</dt>

  <dd>Black hot drink</dd>

  <dt>Milk</dt>

  <dd>White cold drink</dd>

</dl>
```

例子 2

```
<dialog>

  <dt>老师</dt>

  <dd>2+2 等于? </dd>

  <dt>学生</dt>

  <dd>4</dd>

  <dt>老师</dt>

  <dd>答对了! </dd>

</dialog>
```

属性

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant,

lang, ref, registrationmark, tabindex, template, title

如需完整的描述, 请访 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu,

ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover,

ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup,

onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout,

onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述, 请访 [HTML 5 中事件属性](#)。

TIY 实例, 定义列表

如何创建定义列表。

```
<html>
```

```
<body>
```

```
<h2>一个定义列表: </h2>
```

```
<dl>
```

```
<dt>计算机</dt>
```

```
<dd>用来计算的仪器 ... ..</dd>
```

```
<dt>显示器</dt>
```

```
<dd>以视觉方式显示信息的装置 ... ..</dd>
```

```
</dl>
```

```
</body>
```

</html>

<embed> 标签

定义和用法

<embed> 标签定义嵌入的内容，比如插件。

实例

```
<embed src="helloworld.swf" />
```

亲自试一试

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<embed src="/i/helloworld.swf" />
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

<embed> 标签是 HTML 5 中的新标签。

提示和注释

注释：<embed> 标签必须有 src 属性。

提示：您无法在开始标签和结束标签之间写文本，来说明旧式的浏览器不支持该标签，就像 <audio> 和 <video>。

属性

属性	值	描述
----	---	----

height	pixels	设置嵌入内容的高度。
src	url	嵌入内容的 URL。
type	type	定义嵌入内容的类型。
width	pixels	设置嵌入内容的宽度。

标准属性

<embed> 标签支持 HTML 5 中的标准属性。

事件属性

<embed> 标签支持 HTML 5 中的事件属性。

<fieldset> 标签

定义和用法

fieldset 元素可将表单内的相关元素分组。

<fieldset> 标签将表单内容的一部分打包，生成一组相关表单的字段。<fieldset> 标签没有必需的或唯一的属性。

当一组表单元素放到 <fieldset> 标签内时，浏览器会以特殊方式来显示它们，它们可能有特殊的边界、3D 效果，或者甚至可创建一个子表单来处理这些元素。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 5 中有一些新属性，同时不再支持 HTML 4.01 中的一些属性。

属性

属性	值	描述	4	5
disabled	true false	定义 fieldset 是否可见。		5
form	true false	定义该 fieldset 所属的一个或多个表单。		5

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant,

lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访 HTML 5 中标准属性。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu,
ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover,
ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup,
onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout,
onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访 HTML 5 中事件属性。

TIY 实例, 围绕数据的 Fieldset

本例演示如何在数据周围绘制一个带标题的框。

```
<html>

<body>

<fieldset>

<legend>健康信息: </legend>

<form>

<label>身高: <input type="text" /></label>

<label>体重: <input type="text" /></label>

</form>

</fieldset>

<p>如果表单周围没有边框，说明您的浏览器太老了。</p>

</body>

</html>
```

<figcaption> 标签

定义和用法

<figcaption> 标签定义 figure 元素的标题。

“figcaption” 元素应该被置于 “figure” 元素的第一个或最后一个子元素的位置。

实例

文档中的一个小节，描述 PRC：

```
<figure>

  <figcaption>PRC</figcaption>

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

</figure>
```

TIY 实例

```
<!DOCTYPE HTML>

<html>

<body>

  <figure>

    <figcaption>PRC</figcaption>

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

  </figure>

</body>

</html>
```

HTML 4.01 与 HTML 5 之间的差异

<figcaption> 标签是 HTML 5 中的新标签。

提示和注释

提示：“figcaption”元素应该被置于“figure”元素的第一个或最后一个子元素的位置。

标准属性

<figcaption> 标签支持 HTML 5 中的标准属性。

事件属性

<figcaption> 标签支持 HTML 5 中的事件属性。

<figure> 标签

定义和用法

<figure> 标签用于对元素进行组合。

实例

文档中的一个小节，描述 PRC：

```
<figure>
```

```
  <h1>PRC</h1>
```

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

```
</figure>
```

TIY 实例

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
  <figure>
```

```
    <h1>PRC</h1>
```

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

```
</figure>
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

<figure> 标签是 HTML 5 中的新标签。

提示和注释

提示：请使用 <figcaption> 元素为元素组添加标题。

标准属性

<figure> 标签支持 HTML 5 中的标准属性。

事件属性

<figure> 标签支持 HTML 5 中的事件属性。

 标签

定义和用法

 标签规定文本的字体外观、字体尺寸和字体颜色。不要使用该元素，请使用 CSS 向元素添加样式。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 4.01 中，不赞成使用该元素。

在 HTML 5 中，不支持该元素，但是允许由所见即所得的编辑器来插入该元素。

在 HTML 5 中，仅支持 style 属性。

<footer> 标签

定义和用法

<footer> 标签定义 section 或 document 的页脚。

在典型情况下，该元素会包含创作者的姓名、文档的创作日期以及/或者联系信息。

实例

文档的页脚：

```
<footer>This document was written in 2010</footer>
```

亲自试一试代码：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<footer>This document was written in 2010.</footer>
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

<footer> 标签是 HTML 5 中的新标签。

提示和注释

提示：假如您使用 footer 来插入联系信息，应该在 footer 元素内使用 <address> 元素。

标准属性

<footer> 标签支持 HTML 5 中的标准属性。

事件属性

<footer> 标签支持 HTML 5 中的事件属性。

<form> 标签

定义和用法

<form> 标签创建供用户输入的表单。

form 元素包含一个或多个表单元素：

button

input

keygen

object

output

select

textarea

表单用于向服务器（指定的 URL）传递用户数据。

实例

文档的页脚：

```
<form action="form_action.asp">
```

```
First name: <input type="text" name="fname" value="Mickey" /><br />
```

```
Last name:<input type="text" name="lname" value="Mouse" /><br />
```

```
<input type="submit" value="Submit" />
```

```
</form>
```

亲自试一试代码：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<form action="/html5/form_action.asp">

First name: <input type="text" name="fname" value="Mickey" /><br />

Last name:<input type="text" name="lname" value="Mouse" /><br />

<input type="submit" value="Submit" />

</form>

<p>点击“提交”按钮，输入的数据会发送到服务器上名为 “form_action.asp” 的页面。</p>

</body>

</html>
```

HTML 4.01 与 HTML 5 之间的差异

在 HTML 5 中有一些新属性，同时不再支持 HTML 4.01 中的一些属性。

属性

属性	值	描述
accept	MIME_type	HTML 5 中不支持。
accept-charset	charset_list	表单数据的可能的字符集列表（逗号分隔）。
action	URL	定义当点击提交按钮时向何处发送数据。
autocomplete	on off	规定是否自动填写表单。
enctype	见说明	用于对表单内容进行编码的 MIME 类型。
method	get post put delete	用于向 action URL 发送数据的 HTTP 方法。默认是 get。
name	form_name	定义表单唯一的名称。
target	_blank	在何处打开目标 URL。

	<code>_self</code>	
	<code>_parent</code>	
	<code>_top</code>	

说明

enctype 属性可能的值：

`application/x-www-form-urlencoded`

`multipart/form-data`

`text/plain`

autocomplete 属性如果设置为 on，则浏览器可以存储表单的输入值，然后在用户返回该页面时自动填写表单。
如果设置为 off，则浏览器不应存储这些信息。

标准属性

<form> 标签支持 HTML 5 中的标准属性。

事件属性

<form> 标签支持 HTML 5 中的事件属性。

<frame> 标签

定义和用法

<frame> 标签定义子窗口（框架）。

由于该标签对网页可用性的负面影响，在 HTML 5 中 <frame> 标签没有得到支持。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 5 中不支持 <frame> 标签。

<frameset> 标签

定义和用法

`<frameset>` 标签定义框架集。它用于组织多个窗口（框架）。每个框架存在一个独立的文档。在最简单的应用中，`frameset` 元素仅仅声明框架集中存在的列或行的数目。

由于该标签对网页可用性的负面影响，在 HTML 5 中 `<frameset>` 标签没有得到支持。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 5 中不支持 `<frameset>` 标签。

`<h1>` 至 `<h6>` 标签

定义和用法

`<h1>` - `<h6>` 标签可定义标题。`<h1>` 定义最大的标题。`<h6>` 定义最小的标题。

由于 `h` 元素拥有确切的语义，因此请您慎重地选择恰当的标签层级来构建文档的结构。因此，请不要利用标题标签来改变同一行中的字体大小。相反，我们应当使用层叠样式表定义来达到漂亮的显示效果。

如果您希望了解更多有关 HTML 标签选择和使用的信息，请阅读《Web 品质》。

实例

六个不同的 HTML 标题：

```
<h1>这是标题 1</h1>
```

```
<h1>这是标题 2</h1>
```

```
<h2>这是标题 3</h2>
```

```
<h4>这是标题 4</h4>
```

```
<h5>这是标题 5</h5>
```

```
<h6>这是标题 6</h6>
```

亲自试一试代码：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<h1>This is header 1</h1>
```

```
<h2>This is header 2</h2>
```

```
<h3>This is header 3</h3>
```

```
<h4>This is header 4</h4>
```

```
<h5>This is header 5</h5>
```

```
<h6>This is header 6</h6>
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

在 HTML 4.01 中, h1 - h6 元素的 “align” 属性不被赞成使用。

在 HTML 5 中, h1 - h6 元素的 “align” 属性不被支持。

标准属性

<h1> - <h6> 标签支持 HTML 5 中的标准属性。

事件属性

<h1> - <h6> 标签支持 HTML 5 中的事件属性。

<head> 标签

定义和用法

head 元素是所有头部元素的容器。位于 <head> 内部的元素可以包含脚本、指引浏览器找到样式表、提供元信息, 等等。

下面是可用在 head 部分的标签:

```
<base>
```

```
<link>
```

```
<meta>
```

```
<script>
```

```
<style>
```

```
<title>
```

实例

一份简单的 HTML 文档，带有标题：

```
<html>
```

```
<head>
```

```
  <title>文档的标题</title>
```

```
</head>
```

```
<body>
```

```
  文档的内容 ... ..
```

```
  ...
```

```
  ...
```

```
  ...
```

```
</body>
```

```
</html>
```

亲自试一试代码：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
<title>文档的标题</title>
```

```
</head>
```

```
<body>
```

```
文档的内容 .....
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

在 HTML 5 中不再支持 profile 属性。

提示和注释：

提示：请记得始终为文档规定标题！

标准属性

<head> 标签支持 HTML 5 中的标准属性。

<header> 标签

定义和用法

<header> 标签定义文档的页眉（介绍信息）。

实例

对主页的介绍：

```
<header>
```

```
<h1>Welcome to my homepage</h1>
```

```
<p>My name is Donald Duck</p>
```

```
</header>
```

```
<p>The rest of my home page...</p>
```

亲自试一试代码：

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<header>
```

```
<h1>Welcome to my homepage</h1>
```

```
<p>My name is Donald Duck</p>
```

```
</header>
```

```
<p>The rest of my home page...</p>
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

<header> 标签是 HTML 5 中的新标签。

标准属性

<header> 标签支持 HTML 5 中的标准属性。

事件属性

<header> 标签支持 HTML 5 中的事件属性。

<hgroup> 标签

定义和用法

<hgroup> 标签用于对网页或区段（section）的标题进行组合。

实例

使用 <hgroup> 标签对网页或区段（section）的标题进行组合：

```
<hgroup>
```

```
  <h1>Welcome to my WWF</h1>
```

```
  <h2>For a living planet</h2>
```

```
</hgroup>
```

```
<p>The rest of the content...</p>
```

TIY 代码:

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<hgroup>
```

```
  <h1>Welcome to my WWF</h1>
```

```
  <h2>For a living planet</h2>
```

```
</hgroup>
```

```
<p>The rest of the content...</p>
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

<hgroup> 标签是 HTML 5 中的新标签。

提示和注释

提示: 请使用 <figcaption> 元素为元素组添加标题。

标准属性

<hgroup> 标签支持 HTML 5 中的标准属性。

事件属性

<hgroup> 标签支持 HTML 5 中的事件属性。

<hr> 标签

定义和用法

<hr> 标签水平线，它应该定义内容中的主题变化。

HTML 4.01 与 HTML 5 之间的差异

属性：在 HTML 4.01 中，不赞成使用 align, noshade, size 以及 width 属性。在 HTML 5 中不再支持这些属性。

在 HTML 5 中，<hr> 标签定义内容中的主题变化，并显示为一条水平线。

在 HTML 4.01 中，它仅仅显示为一条水平线。

属性

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant,

lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访 HTML 5 中标准属性。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu,

ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover,

ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup,

onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout,

onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访 HTML 5 中事件属性。

TIY 实例，水平线，如何插入水平线。

```
<html>

<body>

<p>hr 标签定义水平线： </p>

<hr />

<p>这是段落。 </p>

<hr />

<p>这是段落。 </p>

<hr />

<p>这是段落。 </p>

</body>

</html>
```

<html> 标签

定义和用法

<html> 标签告知浏览器这是一个 HTML 文档。

html 元素是 HTML 文档中最外层的元素。

html 元素也可称为根元素。

实例

简单的 HTML 5 文档：

```
<!DOCTYPE HTML>

<html>
```

```
<head>

<title>Title of the document</title>

</head>


<body>

The content of the document.....

</body>


</html>
```

亲自试一试

HTML 4.01 与 HTML 5 之间的差异

在 HTML 4.01 中的 xmlns 属性，在 XHTML 中是必须的。实际上它没有任何效果，但是出于验证的原因，在把 HTML 转换为 XHTML 的过程中，是很有帮助的。在 HTML 5 中，没有理由这么做，但是你仍然可以把 xmlns 属性定义为 “http://www.w3.org/1999/xhtml”。

在 HTML 5 中，有一个新属性：manifest。

提示和注释

注释：如果您出于某种原因，希望定义 xmlns 属性，唯一合法的值是 “http://www.w3.org/1999/xhtml”。

属性

属性	值	描述
manifest	url	定义一个 URL，在这个 URL 上描述了文档的缓存信息。
xmlns	http://www.w3.org/1999/xhtml	定义 XML namespace 属性。

标准属性

<HTML> 标签支持 HTML 5 中的标准属性。

事件属性

<HTML> 标签支持 HTML 5 中的事件属性。

<i> 标签

定义和用法

<i> 标签呈现斜体的文本。

<i> 标签定义与文本中其余部分不同的部分，并把这部分文本呈现为斜体文本。在没有其他元素可以使用时，比如 , <cite>, <dfn>, , <q>, <small>, , 请使用 <i> 标签。

HTML 4.01 与 HTML 5 之间的差异

没有区别。

但是在 HTML 4.01 中，存在着一种理解，即应该使用 CSS 来制作斜体文本。在 HTML 5 中，情况没有改变，但是你应该使用 <i> 标签把部分文本定义为某种类型，而不只是利用它在布局中所呈现的样式。

例子

```
<p>The <i class="taxonomy">Felis silvestris catus</i> is cute.</p>
```

```
<p>The term <i>prose content</i> is defined above.</p>
```

```
<p>There is a certain <i lang="fr">je ne sais quoi</i> in the air.</p>
```

属性

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant,

lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu,

ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover,

ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup,

onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout,

onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 [HTML 5 中事件属性](#)。

<iframe> 标签

定义和用法

<iframe> 标签创建包含另一个文档的行内框架。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 5 中，仅仅支持 src 属性

例子

```
<iframe src="/index.html"></iframe>
```

属性

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant,

lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu,

ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover,

ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup,

onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout,

onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 [HTML 5 中事件属性](#)。

TIY 实例，

内联框架，本例演示如何创建内联框架（HTML 页中的框架）。

```
<html>
```

```
<body>

<iframe src="/i/eg_landscape.jpg"></iframe>

<p>一些老的浏览器不支持 iframe。</p>

<p>如果得不到支持，iframe 是不可见的。</p>

</body>

</html>
```

 标签

定义和用法

 标签定义图像。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 4.01 中，图像的 "align", "border", "hspace" 以及 "vspace" 不赞成使用。在 HTML 5 中，不再支持这些属性。

例子

```

```

属性

属性	值	描述	4	5
alt	text	定义有关图形的短的描述。	4	5
src	URL	要显示的图像的 URL。	4	5
align	top bottom middle left	规定如何根据周围的文本来对齐图像。不支持。请使用 CSS 代替。	4	

	right			
border	pixels	定义图像周围的边框。不支持。请使用 CSS 代替。	4	
height	pixels %	定义图像的高度。	4	5
hspace	pixels	定义图像左侧和右侧的空白。不支持。请使用 CSS 代替。	4	
ismap	URL	把图像定义为服务器端的图像映射。	4	5
longdesc	URL	一个 URL，指向了描述该图像的文档。不支持。	4	
usemap	URL	定义作为客户端图像映射的一幅图像。请参阅 <map> 和 <area> 标签，了解其工作原理。	4	5
vspace	pixels	定义图像顶部和底部的空白。不支持。请使用 CSS 代替。	4	5
width	pixels %	设置图像的宽度。	4	5

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant,
lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 HTML 5 中标准属性。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu,
ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover,
ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup,
onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout,
onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 HTML 5 中事件属性。

TIY 实例

插入图像，本例演示如何在网页中显示图像。（略）

从不同的位置插入图片（略）

本例演示如何将其他文件夹或服务器的图片显示到网页中。

浮动图像, 本例演示如何使图片浮动至段落的左边或右边。

```
<html>
```

```
<body>
```

```
<p>
```

```
<img src ="/i/eg_cute.gif" align ="left">
```

带有图像的一个段落。图像的 align 属性设置为 "left"。图像将浮动到文本的左侧。

```
</p>
```

```
<p>
```

```
<img src ="/i/eg_cute.gif" align ="right">
```

带有图像的一个段落。图像的 align 属性设置为 "right"。图像将浮动到文本的右侧。

```
</p>
```

```
</body>
```

```
</html>
```

调整图像尺寸, 本例演示如何将图片调整到不同的尺寸。

```
<html>
```

```
<body>
```

```

```

```
<br />
```

```

```

```
<br />
```

```

```

<p>通过改变 img 标签的 "height" 和 "width" 属性的值, 您可以放大或缩小图像。</p>

```
</body>
```

```
</html>
```

制作图像链接, 本例演示如何将图像作为一个链接使用。

```
<html>
```

```
<body>
```

```
<p>
```

您也可以把图像作为链接来使用:

```
<a href="/example/html/lastpage.html">
```

```

```

```
</a>
```

```
</p>
```

```
</body>
```

```
</html>
```

创建图像地图, 本例显示如何创建带有可供点击区域的图像地图。其中的每个区域都是一个超级链接。

```
<html>
```

```
<body>
```

```
<p>请点击图像上的星球, 把它们放大。</p>
```

```

```

```
<map name="planetmap" id="planetmap">
```

```
<area
```

```
shape="circle"
```

```
coords="180,139,14"
```

```
href="/example/html/venus.html"
```

```
target = "_blank"
```

```
alt="Venus" />
```

```
<area
```

```
shape="circle"
```

```
coords="129, 161, 10"
```

```
href = "/example/html/mercur.html"
```

```
target = "_blank"
```

```
alt="Mercury" />
```

```
<area
```

```
shape="rect"
```

```
coords="0, 0, 110, 260"
```

```
href = "/example/html/sun.html"
```

```
target = "_blank"
```

```
alt="Sun" />
```

```
</map>
```

```
<p><b>注释: </b>img 元素中的 "usemap" 属性引用 map 元素中的 "id" 或 "name" 属性（根据浏览器），所以  
我们同时向 map 元素添加了 "id" 和 "name" 属性。</p>
```

```
</body>
```

```
</html>
```

<input> 标签

定义和用法

<input> 标签定义输入字段，用户可在其中输入数据。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 5 中，该元素有很多新属性，同时不再支持 HTML 4.01 中的一些属性。

在 HTML 5 中，type 属性有很多新的值。

例子

注释：input 元素是空的，仅包含属性。

提示：请使用 label 元素为表单控件定义标注。

属性

属性	值	描述	4	5
accept	<i>list_of_mime_types</i>	一个逗号分隔的 MIME 类型列表，指示文件传输的 MIME 类型。注释：仅可与 type="file" 配合使用。	4	5
align	left right top texttop middle absmiddle baseline bottom absbottom	定义图像之后的文本对齐方式。 不支持，请使用 CSS 代替。	4	
alt	text	定义图像的替代文本。 注释：仅可与 type="image" 配合使用。	4	5
autocomplete				5
autofocus	true false	当页面加载时，使输入字段获得焦点。 注释：type="hidden" 时，无法使用。		5
checked	true false	指示此 input 元素首次加载时应当被选中。 注释：请与 type="checkbox" 及 type="radio" 配合使用。	4	5
disabled	true false	当 input 元素首次加载时禁用此元素，这样用户就无法在其中写 文本，或选定它。 注释：不能与 type="hidden" 一同使用。	4	5

form	true false	定义输入字段属于一个或多个表单。		5
inputmode	inputmode	定义预期的输入类型。		5
list	id of a datalist	引用 datalist 元素。如果定义，则一个下拉列表可用于向输入字段插入值。		5
max	<i>number</i>	输入字段的最大值。		5
maxlength	<i>number</i>	定义文本域中所允许的字符最大数目。	4	5
min	<i>number</i>	输入字段的最小值。		5
name	<i>field_name</i>	为 input 元素定义唯一的名称。	4	5
pattern				5
readonly	readonly	指示是否可修改该字段的值。	4	5
replace	<i>text</i>	定义当表单提交时如何处理该输入字段。		5
required	true false	定义输入字段的值是否是必需的。 当使用下列类型时无法使用： hidden , image , button , submit , reset		5
size	number_of_char	定义 input 元素的大小。不再支持。	4	
src	URL	定义要显示的图像的 URL。 仅用于 type="image" 时。	4	5
step				5
template	<i>template</i>	定义一个或多个模板。		5
type	button checkbox date datetime datetime-local email file hidden image month number password radio	指示 input 元素的类型。 默认值是 "text" 注释：该属性不是必需的。但是我们认为应该使用它。	4	5

	range reset submit text time url week			
value	value	对于按钮、重置按钮和确认按钮：定义按钮上的文本。 对于图像按钮：定义传递向某个脚本的此域的符号结果。 对于复选框和单选按钮：定义 <code>input</code> 元素被点击时的结果。 对于隐藏域、密码域以及文本域：定义元素的默认值。 注释：不能与 <code>type="file"</code> 一同使用。 注释：与 <code>type="checkbox"</code> 和 <code>type="radio"</code> 一同使用时，此元素是必需的。	4	5

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant,
lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 HTML 5 中标准属性。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu,
ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover,
ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup,
onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout,
onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 HTML 5 中事件属性。

TIY 实例

文本域 (Text fields)

本例演示如何在 HTML 页面创建文本域。用户可以在文本域写入文本。

```
<html>

<body>

<form>
名:
<input type="text" name="firstname">
<br />
姓:
<input type="text" name="lastname">
</form>

</body>
</html>
```

密码域

本例演示如何创建 HTML 的密码域。

```
<html>

<body>

<form>
用户:
<input type="text" name="user">
<br />
密码:
<input type="password" name="password">
</form>
<p>
请注意，当您在密码域中键入字符时，浏览器将使用项目符号来代替这些字符。
</p>
</body>
</html>
```

复选框

本例演示如何在 HTML 页中创建文本框。用户可以选中或取消选取复选框。

```
<html>

<body>

<form>
我喜欢自行车:
```

```
<input type="checkbox" name="Bike">
<br />
我喜欢汽车:
<input type="checkbox" name="Car">
</form>

</body>
</html>
```

单选按钮

本例演示如何在 HTML 中创建单选按钮。

```
<html>

<body>

<form>
男性:
<input type="radio" checked="checked" name="Sex" value="male" />
<br />
女性:
<input type="radio" name="Sex" value="female" />
</form>

<p>当用户点击一个单选按钮时，该按钮会变为选中状态，其他所有按钮会变为非选中状态。</p>

</body>
</html>
```

创建按钮

本例演示如何创建按钮。你可以对按钮上的文字进行自定义。

```
<html>

<body>

<form>
<input type="button" value="Hello world!">
</form>

</body>
</html>
```

带有输入框和确认按钮的表单

本例演示如何向页面添加表单。此表单包含两个输入框和一个确认按钮。

```
<html>
<body>

<form action="/example/html/form_action.asp" method="get">
  <p>First name: <input type="text" name="fname" /></p>
  <p>Last name: <input type="text" name="lname" /></p>
  <input type="submit" value="Submit" />
</form>

<p>请单击确认按钮，输入会发送到服务器上名为 "form_action.asp" 的页面。</p>

</body>
</html>
```

带有复选框的表单

此表单包含两个复选框和一个确认按钮。

```
<html>

<body>

<form name="input" action="/html/html_form_action.asp" method="get">
I have a bike:
<input type="checkbox" name="vehicle" value="Bike" checked="checked" />
<br />
I have a car:
<input type="checkbox" name="vehicle" value="Car" />
<br />
I have an airplane:
<input type="checkbox" name="vehicle" value="Airplane" />
<br /><br />
<input type="submit" value="Submit" />
</form>

<p>如果您点击 "Submit" 按钮，您将把输入传送到名为 html_form_action.asp 的新页面。</p>

</body>
</html>
```

带有单选按钮的表单

此表单包含两个单选框和一个确认按钮。

```
<html>
```

```
<body>
```

```
<form name="input" action="/html/html_form_action.asp" method="get">
```

Male:

```
<input type="radio" name="Sex" value="Male" checked="checked">
```

```
<br />
```

Female:

```
<input type="radio" name="Sex" value="Female">
```

```
<br />
```

```
<input type="submit" value="Submit">
```

```
</form>
```

```
<p>如果您点击 "Submit" 按钮, 您将把输入传送到名为 html_form_action.asp 的新页面。</p>
```

```
</body>
```

```
</html>
```

<ins> 标签

定义和用法

<ins> 标签定义文档的其余部分之外的插入文本。

HTML 4.01 与 HTML 5 之间的差异

没有。

例子

提示：请与 `<del>` 标签一起使用，来描述对文档的更新和修正。

属性

属性	值	描述	4	5
<code>cite</code>	URL	指向另一个文档的 URL，该文档解释文本插入的原因。	4	5
<code>datetime</code>	YYYYMMDD	定义文本插入的日期和时间。	4	5

标准属性

`class`, `contenteditable`, `contextmenu`, `dir`, `draggable`, `id`, `irrelevant`,
`lang`, `ref`, `registrationmark`, `tabindex`, `template`, `title`

如需完整的描述，请访问 [HTML 5 中标准属性](#)。

事件属性

`onabort`, `onbeforeunload`, `onblur`, `onchange`, `onclick`, `oncontextmenu`,
`ondblclick`, `ondrag`, `ondragend`, `ondragenter`, `ondragleave`, `ondragover`,
`ondragstart`, `ondrop`, `onerror`, `onfocus`, `onkeydown`, `onkeypress`, `onkeyup`,
`onload`, `onmessage`, `onmousedown`, `onmousemove`, `onmouseover`, `onmouseout`,
`onmouseup`, `onmousewheel`, `onresize`, `onscroll`, `onselect`, `onsubmit`, `onunload`

如需完整的描述，请访问 [HTML 5 中事件属性](#)。

TIY 实例

删除字效果和插入字效果

`<html>`

`<body>`

`<p>`一打有 `<del>`二十`</del>` `<ins>`十二`</ins>` 件。`</p>`

<p>大多数浏览器会改写为删除文本和下划线文本。</p>

<p>一些老式的浏览器会把删除文本和下划线文本显示为普通文本。</p>

</body>

</html>

此例演示如何标示删除文本和插入文本。

<keygen> 标签

定义和用法

<keygen> 标签定义生成密钥。
HTML 4.01 与 HTML 5 之间的差异
keygen 是 HTML 中的新元素。

属性

属性	值	描述
autofocus	disabled	在页面加载时使输入字段获得焦点。 注释：不能与 type="hidden" 一同使用。
challenge	challenge	如果使用，则将 keygen 的值设置为在提交时询问。
disabled	disabled	在 input 元素首次载入时禁用该元素，这样用户就无法在其中输入或选取文本。 注释：不能与 type="hidden" 一同使用。
form	formname	定义一个或多个输入字段所属的表单。
keytype	rsa	定义 keytype。rsa 生成 RSA 密钥。
name	fieldname	定义 input 元素的唯一的名称。name 属性用于在提交时搜集字段的值。

标准属性

<keygen> 标签支持 HTML 5 中的标准属性。

事件属性

<keygen> 标签支持 HTML 5 中的事件属性。

 <dfn> <code> <samp> <kbd> <var> <cite> 标签

定义和用法

以下元素都是短语元素。并不返回使用它们，但是通过使用样式表，可能取得更丰富的效果。

	呈现为被强调的文本。
	定义重要的文本。
<dfn>	定义一个定义项目。
<code>	定义计算机代码文本。
<samp>	定义样本文本。
<kbd>	定义键盘文本。它表示文本是从键盘上键入的。它经常用在与计算机相关的文档或手册中。
<var>	定义变量。您可以将此标签与 <pre> 及 <code> 标签配合使用。
<cite>	定义引用。可使用该标签对参考文献的引用进行定义，比如书籍或杂志的标题。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 4.01 中， 定义语气更重的强调文本，但是在 HTML 5 中， 定义重要的文本。

属性

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访 HTML 5 中标准属性。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访 HTML 5 中事件属性。

HTML 文本格式化实例

文本格式化

<html>

<body>

This text is bold

This text is strong

<big>This text is big</big>

This text is emphasized

<i>This text is italic</i>

<small>This text is small</small>

This text contains

_{subscript}

This text contains

^{superscript}

</body>

</html>

此例演示如何在一个 HTML 文件中对文本进行格式化

“计算机输出” 标签

此例演示不同的“计算机输出”标签的显示效果。

<html>

<body>

<code>Computer code</code>

<kbd>Keyboard input</kbd>


```
<tt>Teletype text</tt>

<br />

<samp>Sample text</samp>

<br />

<var>Computer variable</var>

<br />

<p>

<b>注释: </b>这些标签常用于显示计算机/编程代码。

</p>

</body>

</html>
```

<label> 标签

定义和用法

<label> 标签定义控件的标注。如果您在 label 元素内点击文本，就会触发此控件。就是说，当用户选择该标签时，浏览器就会自动将焦点转到和标签相关的表单控件上。

HTML 4.01 与 HTML 5 之间的差异

没有。

提示和注释

注释：“for” 属性可把 label 绑定到另外一个元素。请把 “for” 属性的值设置为相关元素的 id 属性的值。

属性

属性	值	描述	4	5
----	---	----	---	---

for	id_of_another_field	定义 label 针对哪个表单元素。设置为表单元素的 id。 注释：如果此属性未被规定，那么 label 会关联其内容。	4	5
-----	---------------------	---	---	---

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 HTML 5 中标准属性。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 HTML 5 中事件属性。

TIY 实例 label

如何为控件定义标记。

```
<html>

<body>

<p>请点击文本标记之一，就可以触发相关控件：</p>

<form>

<label for="male">Male</label>

<input type="radio" name="sex" id="male" />

<br />

<label for="female">Female</label>

<input type="radio" name="sex" id="female" />
```

```
</form>
```

```
</body>
```

```
</html>
```

<legend> 标签

定义和用法

<legend> 标签为 <fieldset>、<figure> 以及 <details> 元素定义标题。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 4.01 中，<legend> 仅用于 <fieldset> 元素。

在 HTML 5 中，<legend> 标签应用于 <fieldset>、<figure> 以及 <details> 元素。

在 HTML 5 中，不支持 align 属性。请使用 CSS 代替。

例子

```
<fieldset>
```

```
<legend>健康信息:</legend>
```

```
<form>
```

```
<label>身高:<input type="text" /></label>
```

```
<label>体重:<input type="text" /></label>
```

```
</form>
```

```
</fieldset>
```

属性

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 [HTML 5 中事件属性](#)。

TIY 实例围绕数据的 Fieldset

本例演示如何在数据周围绘制一个带标题的框。

```
<html>

<body>

<fieldset>

<legend>健康信息: </legend>

<form>

<label>身高: <input type="text" /></label>

<label>体重: <input type="text" /></label>

</form>

</fieldset>

<p>如果表单周围没有边框，说明您的浏览器太老了。</p>

</body>

</html>
```

 标签

定义和用法

 标签定义列表项，有序列表和无序列表中都使用 标签。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 4.01 中， 不赞成使用 “type” 和 “value” 属性。

在 HTML 5 中，不再支持 “type” 属性。

在 HTML 5 中， 并没有不赞成使用 “value” 属性，但是仅能够与 元素一起使用。

提示和注释

提示：请使用 CSS 来定义列表的类型。

例子

```
<ol>
```

```
    <li>Coffee</li>
```

```
    <li>Tea</li>
```

```
</ol>
```

```
<ol>
```

```
    <li value="8">Coffee</li>
```

```
    <li>Tea</li>
```

```
</ol>
```

```
<ul>
```

```
    <li>Coffee</li>
```

```
    <li>Tea</li>
```

```
</ul>
```

属性

属性	值	描述	DTD
type	A a I i l disc square circle	规定列表的类型。不赞成使用。请使用样式替代。	TF
value	number_of_list_item	不赞成使用。请使用样式替代。	TF

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 HTML 5 中标准属性。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 HTML 5 中事件属性。

TIY 实例.

无序列表

```
<html>
```

```
<body>
```

```
<h4>一个无序列表: </h4>
```

```
<ul>
```

```
  <li>咖啡</li>
```

```
  <li>茶</li>
```

```
  <li>牛奶</li>
```

```
</ul>
```

```
</body>
```

```
</html>
```

上例演示一个无序列表。

有序列表

```
<html>
```

```
<body>
```

```
<h4>一个有序列表: </h4>
```

```
<ol>
```

```
  <li>咖啡</li>
```

```
  <li>茶</li>
```

```
  <li>牛奶</li>
```

```
</ol>
```

```
</body>
```

```
</html>
```

上例演示一个有序列表。

嵌套列表

```
<html>
```

```
<body>
```

```
<h4>一个嵌套列表: </h4>
```

```
<ul>
```

```
  <li>咖啡</li>
```

```
  <li>茶
```

```
    <ul>
```

```
      <li>红茶</li>
```

```
      <li>绿茶</li>
```

```
    </ul>
```

```
  </li>
```

```
  <li>牛奶</li>
```

```
</ul>
```

```
</body>
```

```
</html>
```

上例演示如何嵌套列表。

嵌套列表 2,

```
<html>
```

```
<body>
```

```
<h4>一个嵌套列表: </h4>
```

```
<ul>
```

```
  <li>咖啡</li>
```

```
  <li>茶
```

```
    <ul>
```

```
      <li>红茶</li>
```

```
      <li>绿茶
```

```
        <ul>
```

```
          <li>中国茶</li>
```

```
          <li>非洲茶</li>
```

```
        </ul>
```

```
      </li>
```

```
    </ul>
```

```
  </li>
```

```
  <li>牛奶</li>
```

```
</ul>
```

```
</body>
```

```
</html>
```

上例演示更复杂的嵌套列表。

<link> 标签

定义和用法

<link> 标签定义两个连接文档之间的关系。

HTML 4.01 与 HTML 5 之间的差异

不再支持 HTML 4.01 中的一些属性。

提示和注释：

注释：link 元素是空元素，它仅包含属性。

注释：此元素只能存在于 head 部分，不过它可出现任意次数。

例子

如何使用 <link> 来链接外部样式表：

```
<head>

<link rel="stylesheet" type="text/css" href="c3.css" />

</head>
```

属性

属性	描述	4	5
charset	定义目标 URL 的字符编码方式。默认值是 "ISO-8859-1"。不支持。	4	
href	目标文档或资源的 URL。	4	5
hreflang	定义目标 URL 的基准语言。	4	5
media	规定文档将显示在什么设备上。	4	5
rel	定义当前文档与目标文档之间的关系。	4	5
rev	定义目标文档与当前文档之间的关系。	4	
target	在何处打开目标 URL。	4	
type	规定目标 URL 的 MIME 类型。	4	5

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant,

lang, ref, registrationmark, tabindex, template, title

如需完整的描述, 请访问 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述, 请访问 [HTML 5 中事件属性](#)。

TIY 实例

链接到一个外部样式表

本例演示如何 `<link>` 标签链接到一个外部样式表。

```
<html>
```

```
<head>
```

```
<link rel="stylesheet" type="text/css" href="/html/csstest1.css" >
```

```
</head>
```

```
<body>
```

```
<h1>我通过外部样式表进行格式化。</h1>
```

```
<p>我也一样! </p>
```

```
</body>
```

```
</html>
```

<map> 标签

定义和用法

<map> 标签定义客户端的图像映射。图像映射是带有可点击区域的图像。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 5 中，不再支持 name 属性。

提示和注释：

注释：area 元素永远嵌套在 map 元素内部。area 元素可定义图像映射中的区域。

属性

属性	值	描述	4	5
id	unique_name	为 map 标签定义唯一的名称。	4	5
name	unique_name	为 map 标签定义唯一的名称。不支持。	4	

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 HTML 5 中标准属性。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 HTML 5 中事件属性。

TIY 实例

创建图像地图

本例显示如何创建带有可点击区域的图像地图。其中的每个区域都是一个超级链接。

<html>

```
<body>
```

```
<p>请点击图像上的星球，把它们放大。</p>
```

```

```

```
<map name="planetmap" id="planetmap">
```

```
<area
```

```
shape="circle"
```

```
coords="180,139,14"
```

```
href ="/example/html/venus.html"
```

```
target = "_blank"
```

```
alt="Venus" />
```

```
<area
```

```
shape="circle"
```

```
coords="129,161,10"
```

```
href ="/example/html/mercur.html"
```

```
target = "_blank"
```

```
alt="Mercury" />
```

```
<area
shape="rect"
coords="0, 0, 110, 260"
href ="/example/html/sun.html"
target = "_blank"
alt="Sun" />
</map>
```

注释: `img` 元素中的 `usemap` 属性引用 `map` 元素中的 `id` 或 `name` 属性（根据浏览器），所以我们同时向 `map` 元素添加了 `id` 和 `name` 属性。

</body>

</html>

<mark> 标签

定义和用法

<mark> 标签定义带有记号的文本。请在需要突出显示文本时使用 <m> 标签。

实例

突出显示部分文本:

<p>Do not forget to buy <mark>milk</mark> today.</p>

TIY

```
<!DOCTYPE HTML>

<html>
```

```
<body>
```

```
<p>Do not forget to buy <mark>milk</mark> today.</p>
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

<mark> 标签是 HTML 5 中的新标签。

标准属性

<mark> 标签支持 HTML 5 中的标准属性。

事件属性

<mark> 标签支持 HTML 5 中的事件属性。

<menu> 标签

定义和用法

<menu> 标签定义菜单列表。当希望列出表单控件时使用该标签。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 4.01 中，不赞成使用 menu 元素。

在 HTML 5 中，重新定义了 menu 元素，且使用用于排列表单控件。

提示和注释

提示：请使用 CSS 来定义列表的类型。

例子

```
<menu>
```

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

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

</menu>
```

属性

属性	值	描述	4	5
autosubmit	true false	如果为 true，那么当表单控件改变时会自动提交。		5
compact	compact_rendering	不支持。请使用 CSS 代替。	4	
label	menulabel	为菜单定义一个可见的标注。		5
type	context	定义显示那种类型的菜单。默认值是 "list"。		5
	toolbar			
	list			

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 [HTML 5 中事件属性](#)。

<meta> 标签

定义和用法

<meta> 元素可提供有关页面的元信息（meta-information），比如针对搜索引擎和更新频度的描述和关键词。

<meta> 标签位于文档的头部，不包含任何内容。<meta> 标签的属性定义了与文档相关联的名称/值对。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 5 中，不再支持 `scheme` 属性。

在 HTML 5 中，有一个新的 `charset` 属性，它使字符集的定义更加容易。

在 HTML 4.01 中，不得不这么写：

```
<meta http-equiv="content-type" content="text/html; charset=ISO-8859-1">
```

在 HTML 5 中，这样就够了：

```
<meta charset="ISO-8859-1">
```

提示和注释

提示：请使用 CSS 来定义列表的类型。

例子

定义针对搜索引擎的关键词：

```
<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript" />
```

定义对页面的描述：

```
<meta name="description" content="免费的 web 技术教程。" />
```

定义页面的最新版本：

```
<meta name="revised" content="David, 2008/8/8/" />
```

每 5 秒刷新一次页面：

```
<meta http-equiv="refresh" content="5" />
```

属性

属性	值	描述	4	5
<code>charset</code>	<i>character encoding</i>	定义文档的字符编码。		5
<code>content</code>	<code>some_text</code>	定义与 <code>http-equiv</code> 或 <code>name</code> 属性相关的元信息。	4	5
<code>http-equiv</code>	<code>content-type</code> <code>expires</code> <code>refresh</code>	把 <code>content</code> 属性关联到 HTTP 头部。	4	5

	set-cookie			
name	- author - description - keywords - generator - revised - others	把 content 属性关联到一个名称。	4	5
scheme	some_text	定义用于翻译 content 属性值的格式。不支持。	4	

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 HTML 5 中标准属性。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 HTML 5 中事件属性。

TIY 实例

文档描述

```
<html>

<head>

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

<meta name="author"

content="w3school.com.cn">
```

```
<meta name="revised"
```

```
content="David Yang, 8/1/07">
```

```
<meta name="generator"
```

```
content="Dreamweaver 8.0en">
```

```
</head>
```

```
<body>
```

```
<p>本文档的 meta 属性标识了创作者和编辑软件。</p>
```

```
</body>
```

```
</html>
```

Meta 元素中的信息可以描述 HTML 文档。

文档关键字

```
<html>
```

```
<head>
```

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

```
<meta name="description"
```

```
content="HTML examples">
```

```
<meta name="keywords"
```

```
content="HTML, DHTML, CSS, XML, XHTML, JavaScript, VBScript">
```

```
</head>
```

```
<body>
```

```
<p>本文档的 meta 属性描述了该文档和它的关键词。</p>
```

```
</body>
```

```
</html>
```

Meta 元素中的信息可以描述文档的关键词。

重定向

```
<html>
```

```
<head>
```

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

```
<meta http-equiv="Refresh"
```

```
content="5;url=http://www.w3school.com.cn">
```

```
</head>
```

```
<body>
```

```
<p>
```

对不起。我们已经搬家了。您的 URL 是 http://www.w3school.com.cn

```
</p>
```

<p>您将在 5 秒内被重定向到新的地址。</p>

<p>如果超过 5 秒后您仍然看到本消息，请点击上面的链接。</p>

</body>

</html>

这个例子演示：在网址已经变更的情况下，将用户重定向到另外一个地址。

<meter> 标签

定义和用法

<meter> 标签定义度量衡。仅用于已知最大和最小值的度量。

实例

```
<meter min="0" max="20">5</meter>
```

```
<meter>2 out of 10</meter>
```

```
<meter>30%</meter>
```

TIY

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<meter min="0" max="10">2</meter><br />
```

```
<meter>2 out of 10</meter><br />
```

```
<meter>20%</meter>
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

<meter> 标签是 HTML 5 中的新标签。

提示和注释

注释：必须定义度量的范围，既可以在元素的文本中，也可以在 min/max 属性中定义。

属性

属性	值	描述
high	number	定义度量的值位于哪个点，被界定为高的值。
low	number	定义度量的值位于哪个点，被界定为低的值。
max	number	定义最大值。默认值是 1。
min	number	定义最小值。默认值是 0。
optimum	number	定义什么样的度量值是最佳的值。 如果该值高于 “high” 属性，则意味着值越高越好。 如果该值低于 “low” 属性的值，则意味着值越低越好。
value	number	定义度量的值。

标准属性

<meter> 标签支持 HTML 5 中的标准属性。

事件属性

<meter> 标签支持 HTML 5 中的事件属性。

<nav> 标签

定义和用法

<nav> 标签定义导航链接的部分。

实例

```
<nav>

<a href="index.asp">Home</a>

<a href="html5_meter.asp">Previous</a>

<a href="html5_noscript.asp">Next</a>

</nav>
```

TIY

```
<!DOCTYPE HTML>

<html>

<body>


<nav>

<a href="/html5/index.asp">Home</a>

<a href="/html5/html5_meter.asp">Previous</a>

<a href="/html5/html5_noscript.asp">Next</a>

</nav>


</body>

</html>
```

HTML 4.01 与 HTML 5 之间的差异

<nav> 标签是 HTML 5 中的新标签。

提示和注释

提示：如果文档中有“前后”按钮，则应该把它放到 <nav> 元素中。

标准属性

<nav> 标签支持 HTML 5 中的标准属性。

事件属性

<nav> 标签支持 HTML 5 中的事件属性。

<noframes> 标签

定义和用法

<noframes> 标签向浏览器显示无法处理框架的提示文本。noframes 元素位于 frameset 元素中。

由于该标签对网页可用性的负面影响，在 HTML 5 中 <noframes> 标签没有得到支持。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 5 中<noscript> 标签

定义和用法

noscript 元素用来定义在脚本未被执行时的替代内容（文本）。

此标签用于可识别 <script> 标签但无法支持其中脚本的浏览器。

HTML 4.01 与 HTML 5 之间的差异

没有。

提示和注释

注释：如果浏览器支持脚本，那么它不会显示出 noscript 元素中的文本。

注释：无法识别 <script> 标签的浏览器会把标签的内容显示到页面上。为了避免浏览器这样做，您应当在注释标签中隐藏脚本。老式的（无法识别 <script> 标签的）浏览器会忽略注释，这样就不会把标签的内容写到页面上，而新式的浏览器则懂得执行这些脚本，即使它们被包围在注释标签中！

例子

JavaScript

```
<body>

...

...

<script type="text/javascript">

  <!--

  document.write("Hello World!")

  //-->

</script>

<noscript>Your browser does not support JavaScript!</noscript>

...

...

</body>
```

VBScript

```
<body>

...

...

<script type="text/vbscript">

  <!--

  document.write("Hello World!")

  ' -->

</script>
```

```
<noscript>Your browser does not support VBScript!</noscript>
```

```
...
```

```
...
```

```
</body>
```

TIY 实例

运行于不支持脚本的浏览器

本例演示如何对付不支持脚本的浏览器。

不支持 `<noframes>` 标签。

`<noscript>` 标签

定义和用法

`noscript` 元素用来定义在脚本未被执行时的替代内容（文本）。

此标签用于可识别 `<script>` 标签但无法支持其中脚本的浏览器。

HTML 4.01 与 HTML 5 之间的差异

没有。

提示和注释

注释：如果浏览器支持脚本，那么它不会显示出 `noscript` 元素中的文本。

注释：无法识别 `<script>` 标签的浏览器会把标签的内容显示到页面上。为了避免浏览器这样做，您应当在注释标签中隐藏脚本。老式的（无法识别 `<script>` 标签的）浏览器会忽略注释，这样就不会把标签的内容写到页面上，而新式的浏览器则懂得执行这些脚本，即使它们被包围在注释标签中！

例子

JavaScript

```
<body>
```

...

...

```
<script type="text/javascript">
```

```
<!--
```

```
document.write("Hello World!")
```

```
//-->
```

```
</script>
```

```
<noscript>Your browser does not support JavaScript!</noscript>
```

...

...

```
</body>
```

VBScript

```
<body>
```

...

...

```
<script type="text/vbscript">
```

```
<!--
```

```
document.write("Hello World!")
```

```
'-->
```

```
</script>
```

```
<noscript>Your browser does not support VBScript!</noscript>
```

```
...
```

```
...
```

```
</body>
```

TIY 实例

运行于不支持脚本的浏览器

```
<html>
```

```
<head>
```

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

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

```
</head>
```

```
<body>
```

```
<script type="text/javascript">
```

```
<!--
```

```
document.write("如果显示了这段文字，那么您的浏览器支持 JavaScript！")
```

```
//-->
```

```
</script>
```

```
<noscript>No JavaScript support!</noscript>
```

```
<p>
```

不支持 JavaScript 的浏览器将显示 noscript 元素中的文本。

```
</p>
```

```
</body>
```

```
</html>
```

本例演示如何对付不支持脚本的浏览器。

<object> 标签

定义和用法

定义一个嵌入的对象。请使用此元素向您的 XHTML 页面添加多媒体。此元素运行您规定插入 HTML 文档中的对象的数据和参数，以及可用来显示和操作数据的代码。

HTML 4.01 与 HTML 5 之间的差异

不再支持 HTML 4.01 中的一些属性。

提示和注释

注释：object 元素可位于在 head 元素或 body 元素内部。<object> 与 </object> 之间的文本是替换文本，针对不支持此标签的浏览器。<param> 标签可定义用于对象的 run-time 设置。

提示：至于图像，请使用 标签代替 <object> 标签。

提示：至少必须定义 “data” 和 “type” 属性之一。

属性

属性	值	描述	4	5
align	left	定义围绕该对象的文本对齐方式。不支持。使用 CSS 代替。	4	
	right			
	top			
	bottom			
archive	URL	一个空格分隔的指向档案文件的 URL 列表。这些档案文件包含了与对象相关的资源。不支持。	4	
border	pixels	定义对象周围的边框。不支持。使用 CSS 代替。	4	

classid	class ID	用于指定浏览器中包含的对象的位置。不支持。	4	
codebase	URL	定义在何处可找到对象所需的代码。不支持。	4	
codetype	MIME type	通过 classid 属性所引用的代码的 MIME 类型。不支持。	4	
data	URL	定义引用对象数据的 URL。	4	5
declare	true false	可定义此对象仅可被声明，但不能被创建或例示，直到此对象得到应用为止。不支持。	4	
height	pixels	定义对象的高度。	4	5
hspace	pixels	定义对象周围水平方向的空白。不支持。使用 CSS 代替。	4	
name	unique_name	为对象定义唯一的名称（以便在脚本中使用）。不支持。使用 id 代替。	4	
standby	text	定义当对象正在加载时所显示的文本。不支持。	4	
type	MIME_type	定义被规定在 data 属性中指定的文件中出现的数据的 MIME 类型。	4	5
usemap	URL	规定与对象一同使用的客户端图像映射的 URL。	4	5
vspace	pixels	定义对象的垂直方向的空白。不支持。使用 CSS 代替。	4	
width	pixels	定义对象的宽度。	4	5

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 HTML 5 中标准属性。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 HTML 5 中事件属性。

 标签

定义和用法

 标签定义有序列表。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 4.01 中，不赞成使用 “start” 属性，在 HTML 5 中是允许的。

在 HTML 4.01 中，不赞成使用 “compact” 和 “type” 属性，在 HTML 5 中，不再支持这两个属性。

提示和注释

提示：请使用 CSS 来定义列表的类型。

例子

```
<ol>

  <li>Coffee</li>

  <li>Tea</li>

</ol>
```

```
<ol>

  <li start="60">Coffee</li>

  <li>Tea</li>

</ol>
```

属性

属性	值	描述	4	5
compact	compact_rendering	不赞成。使用 CSS 代替。	4	
start	start_on_number	规定起始数字。	4	5
type	A a I	规定列表的类型。不赞成。使用 CSS 代替。	4	

	i			
	1			

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 [HTML 5 中事件属性](#)。

TIY 实例

有序列表

```
<html>

<body>

<h4>一个有序列表: </h4>

<ol>

  <li>咖啡</li>

  <li>茶</li>

  <li>牛奶</li>

</ol>

</body>
```

```
</html>
```

本例演示一个有序列表。

<optgroup> 标签

定义和用法

<optgroup> 标签定义选项组。此元素允许您组合选项。当您使用一个长的选项列表时，对相关的选项进行组合会使处理更加容易。

HTML 4.01 与 HTML 5 之间的差异

没有。

例子

```
<select>

  <optgroup label="Swedish Cars">

    <option value ="volvo">Volvo</option>

    <option value ="saab">Saab</option>

  </optgroup>

  <optgroup label="German Cars">

    <option value ="mercedes">Mercedes</option>

    <option value ="audi">Audi</option>

  </optgroup>

</select>
```

属性

属性	值	描述	4	5
label	text_label	定义选项组的标注。	4	5
disabled	disabled	在其首次加载时，禁用该选项组。	4	5

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述, 请访问 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述, 请访问 [HTML 5 中事件属性](#)。

<option> 标签

定义和用法

<option> 标签定义下拉列表中的一个选项。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 5 中, <option> 标签也用于新元素 <datalist> 中。

提示和注释

注释: <option> 标签能够在不带任何属性的情况下使用, 但是通常需要 value 属性, 该属性定义了发送到服务器的数据。

注释: 请与 <select> 或 <datalist> 标签结合使用。在其他地方, <option> 标签是无意义的。

例子

```
<select>
```

```
  <option value="volvo">Volvo</option>
```

```
  <option value="saab">Saab</option>
```

```
  <option value="opel" selected="selected">Opel</option>
```

```
  <option value="audi">Audi</option>
```

```
</select>
```

属性

属性	值	描述	4	5
disabled	disabled	规定此选项应在首次加载时被禁用。	4	5
label	text	定义当使用 <optgroup> 时所使用的标注。	4	5
selected	selected	规定选项（在首次显示在列表中时）表现为选中状态。	4	5
value	text	定义送往服务器的选项值。	4	5

标准属性

```
class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title
```

如需完整的描述，请访问 [HTML 5 中标准属性](#)。

事件属性

```
onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload
```

如需完整的描述，请访问 [HTML 5 中事件属性](#)。

TIY 实例

简单的下拉列表

```
<html>

<body>

<form>

<select name="cars">

<option value="volvo">Volvo</option>
```

```
<option value="saab">Saab</option>
```

```
<option value="fiat">Fiat</option>
```

```
<option value="audi">Audi</option>
```

```
</select>
```

```
</form>
```

```
</body>
```

```
</html>
```

本例演示如何在 HTML 页面中创建简单的下拉列表框。下拉列表框是一个可选列表。

另一个下拉列表

```
<html>
```

```
<body>
```

```
<form>
```

```
<select name="cars">
```

```
<option value="volvo">Volvo</option>
```

```
<option value="saab">Saab</option>
```

```
<option value="fiat" selected="selected">Fiat</option>
```

```
<option value="audi">Audi</option>
```

```
</select>
```

```
</form>
```

```
</body>
```

```
</html>
```

本例演示如何创建一个简单的带有预选值的下拉列表。（注：预选值指预先指定的首选项。）

<output> 标签

定义和用法

<output> 标签定义不同类型的输出，比如脚本的输出。

实例

如何在表单中使用 output 元素：

```
<form action="form_action.asp" method="get" name="sumform">
```

```
<output name="sum"></output>
```

```
</form>
```

TIY

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
<script type="text/javascript">
```

```
function write_sum()
```

```
{
```

```
x=5
```

```
y=3
```

```
document.forms["sumform"]["sum"].value=x+y
```

```
}
```

```
</script>
```

```
</head>
```

```
<body onload="write_sum()">

<form action="form_action.asp" method="get" name="sumform">

<output name="sum"></output>

</form>

</body>

</html>
```

HTML 4.01 与 HTML 5 之间的差异

<output> 标签是 HTML 5 中的新标签。

属性

属性	值	描述
for	id of another element	定义输出域相关的一个或多个元素。
form	formname	定义输入字段所属的一个或多个表单。
name	unique name	定义对象的唯一名称。（表单提交时使用）

标准属性

<output> 标签支持 HTML 5 中的标准属性。

事件属性

<output> 标签支持 HTML 5 中的事件属性。

<p> 标签

定义和用法

<p> 标签定义段落。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 4.01 中，不赞成使用 align 属性，在 HTML 5 中不再支持该属性。

属性

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 [HTML 5 中事件属性](#)。

TIY 实例

简单的段落

```
<html>
```

```
<body>
```

```
<p>这是段落。</p>
```

```
<p>这是段落。</p>
```

```
<p>这是段落。</p>
```

```
<p>段落元素由 p 标签定义。</p>
```

```
</body>
```

</html>

此例演示：段落元素中的文字如何被浏览器显示。

更多的段落

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
```

```
"http://www.w3.org/TR/html4/loose.dtd">
```

<html>

<body>

<p>

这个段落

在源代码中

包含许多行

但是浏览器

忽略了它们。

</p>

<p>

这个段落

在源代码 中

包含 许多行

但是 浏览器

忽略了 它们。

</p>

<p>

段落的行数依赖于浏览器窗口的大小。如果调节浏览器窗口的大小，将改变段落中的行数。

</p>

</body>

</html>

本例演示段落元素的某些缺省的行为。

“诗歌”问题

<html>

<body>

<h1>春晓</h1>

<p>

春眠不觉晓，

处处闻啼鸟。

夜来风雨声，

花落知多少。

</p>

<p>注意，浏览器忽略了源代码中的排版（省略了多余的空格和换行）。</p>

```
</body>
```

```
</html>
```

本例演示 HTML 格式化的某些问题。

<param> 标签

定义和用法

param 元素允许您为插入 XHTML 文档的对象规定 run-time 设置，也就是说，此标签可为包含它的 <object> 标签提供参数。

HTML 4.01 与 HTML 5 之间的差异

不再支持 HTML 4.01 中的一些属性。

例子

```
<object id="Slider1" width="100" height="50">

  <param name="BorderStyle" value="1" />

  <param name="MousePointer" value="0" />

  <param name="Enabled" value="1" />

  <param name="Min" value="0" />

  <param name="Max" value="10" />

</object>
```

属性

属性	值	描述	4	5
name	unique_name	定义参数的唯一的名称（唯一标识符）。	4	5
type	MIME type	规定参数的 MIME 类型（internet media type）。不支持。	4	
value	value	规定参数的值。	4	5

valuetype	data	规定值的 MIME 类型。不支持。	4	
	ref			
	object			

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 HTML 5 中标准属性。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 HTML 5 中事件属性。

<pre> 标签

定义和用法

pre 元素可定义预格式化的文本。被包围在 pre 元素中的文本通常会保留空格和换行符。而文本也会呈现为等宽字体。

<pre> 标签的一个常见应用就是用来表示计算机的源代码。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 4.01 中，不赞成使用 width 属性，在 HTML 5 中不再支持该属性。

提示和注释

提示：请使用 <pre> 标签来显示非常规的格式化内容，或者某类计算机代码。

属性

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 [HTML 5 中事件属性](#)。

TIY 实例

预格式文本

```
<html>
```

```
<body>
```

```
<pre>
```

这是

预格式文本。

它保留了 空格

和换行。

```
</pre>
```

```
<p>pre 标签很适合显示计算机代码:</p>
```

```
<pre>
```

```
for i = 1 to 10
```

```
    print i
```

```
next i
```

```
</pre>
```

```
</body>
```

```
</html>
```

此例演示如何使用 `pre` 标签对折行和空格进行控制。

<progress> 标签

定义和用法

`<progress>` 标签定义运行中的进度（进程）。

可以使用 `<progress>` 标签来显示 JavaScript 中耗费时间的函数的进度。

实例

标记“下载进度”：

对象的下载进度：

```
<progress>
```

```
<span id="objprogress">85</span>%
```

```
</progress>
```

TIY

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

The object's downloading progress:

```
<progress>
```

```
<span id="objprogress">76</span>%
```

```
</progress>
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

<progress> 标签是 HTML 5 中的新标签。

提示和注释

提示：请使用 <progress> 标签来显示下载的进度。

属性

属性	值	描述
max	number	定义完成的值。
value	number	定义进程的当前值。

标准属性

<progress> 标签支持 HTML 5 中的标准属性。

事件属性

<progress> 标签支持 HTML 5 中的事件属性。

<q> 标签

定义和用法

<q> 标签定义一个短的引用。

浏览器经常会在这种引用的周围插入引号。

实例

标记一段短的引用：

```
<p>WWF's goal is to:  
  
<q cite="http://www.wwf.org">  
  
build a future where people live in harmony with nature  
  
</q> we hope they succeed.</p>  
  
亲自试一试
```

HTML 4.01 与 HTML 5 之间的差异

没有。

提示和注释

提示：请使用 cite 属性来定义该引用的来源。

提示：请使用 blockquote 元素来标记长的引用。

属性

属性	值	描述
cite	url	定义引用的引文。

标准属性

<q> 标签支持 HTML 5 中的标准属性。

事件属性

<q> 标签支持 HTML 5 中的事件属性。

<rp> 标签

定义和用法

<rp> 标签在 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。

ruby 注释是中文注音或字符。

在东亚使用，显示的是东亚字符的发音。

与 <ruby> 以及 <rt> 标签一同使用：

ruby 元素由一个或多个字符（需要一个解释/发音）和一个提供该信息的 rt 元素组成，还包括可选的 rp 元素，定义当浏览器不支持“ruby”元素时显示的内容。

实例

一个 ruby 注释：

```
<ruby>
```

```
  漢 <rt><rp>(</rp>厂马`<rp></rp></rt>
```

```
</ruby>
```

TIY

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<ruby>
```

```
  漢 <rt><rp>(</rp>厂马`<rp></rp></rt>
```

```
</ruby>
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

<rp> 标签是 HTML 5 的新标签。

提示和注释

提示：支持“ruby”元素的浏览器不会显示“rp”元素的内容。

标准属性

<rp> 标签支持 HTML 5 中的标准属性。

事件属性

<rp> 标签支持 HTML 5 中的事件属性。

<rt> 标签

定义和用法

<rt> 标签定义字符（中文注音或字符）的解释或发音。

ruby 注释是中文注音或字符。

在东亚使用，显示的是东亚字符的发音。

与 <ruby> 以及 <rt> 标签一同使用：

ruby 元素由一个或多个字符（需要一个解释/发音）和一个提供该信息的 rt 元素组成，还包括可选的 rp 元素，定义当浏览器不支持“ruby”元素时显示的内容。

实例

一个 ruby 注释：

```
<ruby>
```

```
  漢 <rt> 尸马 ` </rt>
```

```
</ruby>
```

TIY

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<ruby>
```

```
  漢 <rt> 尸马 ` </rt>
```

```
</ruby>
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

<rt> 标签是 HTML 5 的新标签。

提示和注释

提示：支持 “ruby” 元素的浏览器不会显示 “rp” 元素的内容。

标准属性

<rt> 标签支持 HTML 5 中的标准属性。

事件属性

<rt> 标签支持 HTML 5 中的事件属性。

<ruby> 标签

定义和用法

<ruby> 标签定义 ruby 注释（中文注音或字符）。

在东亚使用，显示的是东亚字符的发音。

与 <ruby> 以及 <rt> 标签一同使用：

ruby 元素由一个或多个字符（需要一个解释/发音）和一个提供该信息的 rt 元素组成，还包括可选的 rp 元素，定义当浏览器不支持 “ruby” 元素时显示的内容。

实例

一个 ruby 注释：

```
<ruby>
```

```
  漢 <rt><rp>(</rp>厂马`<rp></rp></rt>
```

```
</ruby>
```

TIY

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<ruby>
```

```
  漢 <rt> 尸马 ` </rt>
```

```
</ruby>
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

<ruby> 标签是 HTML 5 的新标签。

提示和注释

提示：支持 “ruby” 元素的浏览器不会显示 “rp” 元素的内容。

标准属性

<ruby> 标签支持 HTML 5 中的标准属性。

事件属性

<ruby> 标签支持 HTML 5 中的事件属性。

<s> 标签

定义和用法

<s> 标签定义加删除线的文本。HTML 5 中不再支持这个标签。请使用 CSS 代替。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 4.01 中不赞成使用 <s> 标签。

在 HTML 5 中不支持 <s> 标签。

提示和注释

提示：请使用 `<del>` 标签代替。

`<script>` 标签

定义和用法

`<script>` 标签用于定义客户端脚本，比如 JavaScript。

`script` 元素即可包含脚本语句，又可通过 `src` 属性指向外部脚本文件。

必需的 `type` 属性规定脚本的 MIME 类型。

JavaScript 通常的用途是图像操作、表单验证以及内容动态更改。

实例

```
<script type="text/javascript">
```

```
document.write("Hello World!")
```

```
</script>
```

亲自试一试

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

```
<script type="text/javascript">
```

```
document.write("Hello World!")
```

```
</script>
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

async 属性是 HTML 5 中的新属性。

在 HTML 5 中，不再支持 HTML 4.01 中的一些属性。

提示和注释

注释：脚本按照如下方式执行：

如果 async 属性为 true，则脚本会相对于文档的其余部分异步执行，这样脚本会在页面继续解析的过程中执行。

如果 async 属性为 false，而 defer 属性为 true，则脚本会在页面完成解析后执行。

如果 async 和 defer 属性均为 false，那么脚本会立即执行，页面会在脚本执行完毕后继续解析。

提示：如果存在 src 属性，则 <script> 标签必须是空的。

属性

属性	值	描述
async	async	定义脚本是否异步执行。
type	text/ecmascript text/javascript application/ecmascript application/javascript text/vbscript	指示脚本的 MIME 类型。
charset	charset	定义脚本中使用的字符编码。不支持。
defer	defer	指示脚本不会生成任何文档内容。浏览器可继续解析并绘制页面。
language	javascript livescript vbscript other	HTML 5 中不支持。
src	URL	定义指向包含脚本的文件的 URL（这样您就可以引用某个包含脚

		本的文件，而不是直接把脚本插入您的 HTML 文档中）。
xml:space	preserve	HTML 5 中不支持。

标准属性

<script> 标签支持 HTML 5 中的标准属性。

事件属性

<script> 标签支持 HTML 5 中的事件属性。

<section> 标签

定义和用法

<section> 标签定义文档中的节（section、区段）。比如章节、页眉、页脚或文档中的其他部分。

实例

文档中的区段，解释了 PRC:

```
<section>

  <h1>PRC</h1>

  <p>The People’s Republic of China was born in 1949...</p>

</section>
```

TIY

```
<!DOCTYPE HTML>

<html>

<body>

<section>

  <h1>PRC</h1>

  <p>The People’s Republic of China was born in 1949...</p>
```

```
</section>
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

`<section>` 标签是 HTML 5 中的新标签。

属性

| 属性 | 值 | 描述 |
|-------------------|-----|-------------------------------------|
| <code>cite</code> | URL | section 的 URL，假如 section 摘自 web 的话。 |

标准属性

`<section>` 标签支持 HTML 5 中的标准属性。

事件属性

`<section>` 标签支持 HTML 5 中的事件属性。

`<select>` 标签

定义和用法

`<select>` 标签创建下拉列表。

HTML 4.01 与 HTML 5 之间的差异

HTML 5 有一些新的属性，同时不再支持一些 HTML 4.01 的属性。

提示和注释：

提示：请在 `form` 元素中使用此标签来接受用户的输入。

例子

```
<select>
```

```
<option value="volvo">Volvo</option>

<option value="saab">Saab</option>

<option value="opel">Opel</option>

<option value="audi">Audi</option>

</select>
```

属性

| 属性 | 值 | 描述 | 4 | 5 |
|-----------|-------------|---------------------------|---|---|
| autofocus | true | 在页面加载时使这个 select 字段获得焦点。 | | 5 |
| | false | | | |
| data | url | 供自动插入数据。 | | 5 |
| disabled | true | 当该属性为 true 时，会禁用该菜单。 | 4 | 5 |
| | false | | | |
| form | true | 定义 select 字段所属的一个或多个表单。 | | 5 |
| | false | | | |
| multiple | true | 当该属性为 true 时，规定可一次选定多个项目。 | 4 | 5 |
| | false | | | |
| name | unique_name | 定义下拉列表的唯一标识符。 | 4 | 5 |
| size | number | 定义菜单中可见项目的数目。不支持。 | 4 | |

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 [HTML 5 中事件属性](#)。

TIY 实例

简单的下拉列表

```
<html>
```

```
<body>
```

```
<form>
```

```
<select name="cars">
```

```
<option value="volvo">Volvo</option>
```

```
<option value="saab">Saab</option>
```

```
<option value="fiat">Fiat</option>
```

```
<option value="audi">Audi</option>
```

```
</select>
```

```
</form>
```

```
</body>
```

```
</html>
```

本例演示如何在 HTML 页面中创建简单的下拉列表框。下拉列表框是一个可选列表。

另一个下拉列表

```
<html>
```

```
<body>
```

```
<form>

<select name="cars">

<option value="volvo">Volvo</option>

<option value="saab">Saab</option>

<option value="fiat" selected="selected">Fiat</option>

<option value="audi">Audi</option>

</select>

</form>


</body>

</html>
```

本例演示如何创建一个简单的带有预选值的下拉列表。（注：预选值指预先指定的首选项。）

<small> 标签

定义和用法

<small> 标签制作更小字号的文本。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 5 中不支持 <small> 标签。

<source> 标签

定义和用法

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

HTML 4.01 与 HTML 5 之间的差异

<source> 标签是 HTML 5 中的新标签。

属性

| 属性 | 值 | 描述 |
|-------|---------------|---------------------------------|
| media | media query | 定义媒介资源的类型，供浏览器决定是否下载。 |
| src | url | 媒介的 URL。 |
| type | numeric value | 定义播放器在音频流中的什么位置开始播放。默认，音频从开头播放。 |

标准属性

<source> 标签支持 HTML 5 中的标准属性。

事件属性

<source> 标签支持 HTML 5 中的事件属性。

** 标签**

定义和用法

 标签用于对文档中的行内元素进行组合。

HTML 4.01 与 HTML 5 之间的差异

没有。

提示和注释：

提示：请使用 span 元素对行内元素进行分组，以便通过样式对它们进行格式化。

例子

<p>This is a paragraph This is a paragraph

This is a paragraph</p>

属性

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 HTML 5 中标准属性。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 [HTML 5 中事件属性](#)。

<strike> 标签

定义和用法

<strike> 标签定义加删除线的文本。HTML 5 中不再支持这个标签。请使用 CSS 代替。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 4.01 中不赞成使用 <strike> 标签。

在 HTML 5 中不支持 <strike> 标签。

提示和注释

提示：请使用 标签代替。

<style> 标签

定义和用法

<style> 标签定义 HTML 文档的样式信息。

在 style 元素中，您规定的是 HTML 元素如何在浏览器中呈现。

实例

在 HTML 文档中使用 style 元素：

```
<html>

<head>

<style type="text/css">

h1 {color:red}

p {color:blue}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Header-1</h1>
```

```
<p>This is a paragraph.</p>
```

```
</body>
```

```
</html>
```

亲自试一试

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
<style type="text/css">
```

```
h1 {color:red}
```

```
p {color:blue}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Header-1</h1>
```

```
<p>This is a paragraph.</p>
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

scoped 属性是 HTML 5 中的新属性，它允许我们为文档的指定部分定义样式，而不是整个文档。

如果使用 “scoped” 属性，那么所规定的样式只能应用到 style 元素的父元素及其子元素。

提示和注释：

提示：如需链接外部样式表，请使用 <link> 标签。

提示：如需学习更多有关样式表的知识，请访问相关的 CSS 教程。

注释：如果未定义 scoped 属性，那么 <style> 元素必须位于 <head> 部分中。

属性

属性	值	描述
type	text/css	定义内容类型。
media	screen tty tv projection handheld print braille aural all	样式信息的目标媒介。
scoped	scoped	如果使用该属性，则样式仅仅应用到 style 元素的父元素及其子元素。

标准属性

<style> 标签支持 HTML 5 中的标准属性。

事件属性

<style> 标签支持 HTML 5 中的事件属性。

<sub> 和 <sup> 标签

定义和用法

<sub> 标签可定义下标文本。<sup> 可定义上标文本。

HTML 4.01 与 HTML 5 之间的差异

没有。

例子

这段文本包含 _{下标}

这段文本包含 ^{上标}

属性

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 [HTML 5 中事件属性](#)。

TIY 实例

上标和下标

```
<html>
```

```
<body>
```

<p>

This text contains _{subscript}

</p>

<p>

This text contains ^{superscript}

</p>

</body>

</html>

本例演示如何使用 sup 和 sub 元素。

<summary> 标签

定义和用法

<summary> 标签包含 details 元素的标题，“details”元素用于描述有关文档或文档片段的详细信息。

实例

有关文档的详细信息：

<details>

<summary>HTML 5</summary>

This document teaches you everything you have to learn about HTML 5.

</details>

亲自试一试

<!DOCTYPE HTML>

<html>

```
<body>
```

```
<details>
```

```
<summary>HTML 5</summary>
```

This document teaches you everything you have to learn about HTML 5.

```
</details>
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

`<summary>` 标签是 HTML 5 中的新标签。

提示和注释

提示：请与 `<details>` 标签一起使用。标题是可见的，当用户点击标题时会显示出详细信息。

注释：“summary”元素应该是“details”元素的第一个子元素。

标准属性

`<summary>` 标签支持 HTML 5 中的标准属性。

事件属性

`<summary>` 标签支持 HTML 5 中的事件属性。

<table> 标签

定义和用法

`<table>` 标签可定义表格。在 `<table>` 标签内部，你可以放置表格的标题、表格行、表格列、表格单元以及其他的表格。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 4.01 中，不赞成使用 “align” 和 “bgcolor” 属性。

在 HTML 5 中，不支持 <table> 标签的任何属性。

例子

```
<table>

<tr>

<td>Cell A</td>

<td>Cell B</td>

</tr>

</table>
```

属性

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 [HTML 5 中事件属性](#)。

TIY 实例

表格

```
<html>

<body>
```

<p>每个表格由 table 标签开始。</p>

<p>每个表格行由 tr 标签开始。</p>

<p>每个表格数据由 td 标签开始。</p>

<h4>一列: </h4>

```
<table border="1">
```

```
<tr>
```

```
<td>100</td>
```

```
</tr>
```

```
</table>
```

<h4>一行三列: </h4>

```
<table border="1">
```

```
<tr>
```

```
<td>100</td>
```

```
<td>200</td>
```

```
<td>300</td>
```

```
</tr>
```

```
</table>
```

<h4>两行三列: </h4>

```
<table border="1">
```

```
<tr>
```

```
<td>100</td>

<td>200</td>

<td>300</td>

</tr>

<tr>

<td>400</td>

<td>500</td>

<td>600</td>

</tr>

</table>
```

```
</body>
```

```
</html>
```

这个例子演示如何在 HTML 文档中创建表格。

表格中的标题(Headings)

```
<html>
```

```
<body>
```

```
<h4>表头: </h4>
```

```
<table border="1">
```

```
<tr>
```

```
<th>姓名</th>
```

```
<th>电话</th>
```

```
<th>电话</th>

</tr>

<tr>

<td>Bill Gates</td>

<td>555 77 854</td>

<td>555 77 855</td>

</tr>

</table>
```

<h4>垂直的表头: </h4>

```
<table border="1">

<tr>

<th>姓名</th>

<td>Bill Gates</td>

</tr>

<tr>

<th>电话</th>

<td>555 77 854</td>

</tr>

<tr>

<th>电话</th>

<td>555 77 855</td>

</tr>

</table>
```

```
</body>
```

```
</html>
```

本例演示如何显示表格标题。

空单元格

```
<html>
```

```
<body>
```

```
<table border="1">
```

```
<tr>
```

```
  <td>Some text</td>
```

```
  <td>Some text</td>
```

```
</tr>
```

```
<tr>
```

```
  <td></td>
```

```
  <td>Some text</td>
```

```
</tr>
```

```
</table>
```

<p>正如您看到的，其中一个单元没有边框。这是因为它是空的。在该单元中插入一个空格后，仍然没有边框。
</p>

<p>我们的技巧是在单元中插入一个 no-breaking 空格。</p>

<p>no-breaking 空格是一个字符实体。如果您不清楚什么是字符实体，请阅读关于字符实体的章节。</p>

<p>no-breaking 空格由和号开始（"&"），然后是字符"nbsp"，并以分号结尾(";")。</p>

</body>

</html>

本例展示如何使用 " " 处理没有内容的单元格。

带有标题的表格

<html>

<body>

<h4>这个表格有一个标题，以及粗边框：</h4>

<table border="6">

<caption>我的标题</caption>

<tr>

<td>100</td>

<td>200</td>

<td>300</td>

</tr>

<tr>

<td>400</td>

```
<td>500</td>
```

```
<td>600</td>
```

```
</tr>
```

```
</table>
```

```
</body>
```

本例演示一个带标题（caption）的表格

跨行或跨列的表格单元格

```
<html>
```

```
<body>
```

```
<h4>横跨两列的单元格：</h4>
```

```
<table border="1">
```

```
<tr>
```

```
<th>姓名</th>
```

```
<th colspan="2">电话</th>
```

```
</tr>
```

```
<tr>
```

```
<td>Bill Gates</td>
```

```
<td>555 77 854</td>
```

```
<td>555 77 855</td>
```

```
</tr>
```

```
</table>
```

<h4>横跨两行的单元格：</h4>

```
<table border="1">
```

```
<tr>
```

```
  <th>姓名</th>
```

```
  <td>Bill Gates</td>
```

```
</tr>
```

```
<tr>
```

```
  <th rowspan="2">电话</th>
```

```
  <td>555 77 854</td>
```

```
</tr>
```

```
<tr>
```

```
  <td>555 77 855</td>
```

```
</tr>
```

```
</table>
```

```
</body>
```

```
</html>
```

本例演示如何定义跨行或跨列的表格单元格。

表格内的标签

本例演示如何在不同的元素内显示元素。

```
<html>
```

```
<body>
```

```
<table border="1">

<tr>

  <td>

    <p>这是一个段落。</p>

    <p>这是另一个段落。</p>

  </td>

  <td>这个单元包含一个表格:

    <table border="1">

      <tr>

        <td>A</td>

        <td>B</td>

      </tr>

      <tr>

        <td>C</td>

        <td>D</td>

      </tr>

    </table>

  </td>

</tr>

<tr>

  <td>这个单元包含一个列表:

    <ul>

      <li>苹果</li>
```

```
<li>香蕉</li>

<li>菠萝</li>

</ul>

</td>

<td>HELLO</td>

</tr>

</table>

</body>

</html>
```

<tbody> 标签

定义和用法

定义一段表格主体（正文）。

使用 <tbody> 标签，可以将表格分为一个单独的部分。<tbody> 标签可将表格中的一行或几行合成一组。

虽然您可能想包括一个，甚至会在表格中包括两个或更多个 <tbody> 标签，但是我们建议最好在表格中没有 <tbody> 标签。

在 <tbody> 标签中，只有 <tr> 标签可以定义表格行。并且一旦定义，一个 <tbody> 标签就是表格中的一个独立的部分。例如不能从一个 <tbody> 跨越到另一个 <tbody> 中。

thead、tfoot 以及 tbody 元素使您有能力对表格中的行进行分组。当您创建某个表格时，您也许希望拥有一个标题行，一些带有数据的行，以及位于底部的一个总计行。这种划分使浏览器有能力支持独立于表格标题和页脚的表格正文滚动。当长的表格被打印时，表格的表头和页脚可被打印在包含表格数据的每张页面上。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 5 中，不支持 <tbody> 标签的任何属性。

提示和注释：

注释: <thead> 内部必须拥有 <tr> 标签!

注释: 如果您使用 `thead`、`tfoot` 以及 `tbody` 元素, 您就必须使用全部的元素。它们的出现次序是: `thead`、`tfoot`、`tbody`, 这样浏览器就可以在收到所有数据前呈现页脚了。您必须在 `table` 元素内部使用这些标签。

例子

```
<table>
```

```
  <thead>
```

```
    <tr>
```

```
      <td>THEAD 中的文本</td>
```

```
    </tr>
```

```
  </thead>
```

```
  <tfoot>
```

```
    <tr>
```

```
      <td>TFOOT 中的文本</td>
```

```
    </tr>
```

```
  </tfoot>
```

```
  <tbody>
```

```
    <tr>
```

```
      <td>TBODY 中的文本</td>
```

```
    </tr>
```

```
  </tbody>
```

```
</table>
```

属性

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述, 请访问 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述, 请访问 [HTML 5 中事件属性](#)。

<td> 标签

定义和用法

<td> 标签定义表格中的一个单元格。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 4.01 中, td 元素的 "bgcolor"、"height"、"width" 以及 "nowrap" 是不被赞成使用的。

在 HTML 5 中, 仅仅支持 "colspan" 和 "rowspan" 属性。

例子

```
<table>
```

```
  <tr>
```

```
    <td>Cell A</td>
```

```
    <td>Cell B</td>
```

```
  </tr>
```

</table>

属性

| 属性 | 值 | 描述 | 4 | 5 |
|---------|--|--|---|---|
| abbr | abbr_text | 规定单元格中内容的缩写版本。不支持。 | 4 | |
| align | left

right

center

justify

char | 规定单元格内容的水平排列方式。不支持。请使用 CSS 代替。 | 4 | |
| axis | category_names | 为单元格定义一个名字。不支持。 | 4 | |
| bgcolor | rgb(x, x, x)

#xxxxxx

colorname | 规定单元格的背景颜色。不支持。请使用 CSS 代替。 | 4 | |
| char | character | 规定根据哪个字符来进行文本对齐。不支持。请使用 CSS 代替。 | 4 | |
| charoff | pixels

% | 规定第一个对齐字符的偏移量。不支持。请使用 CSS 代替。 | 4 | |
| colspan | number | 规定此单元格可横跨的列数。 | 4 | 5 |
| headers | header_cells'_id | 空格分隔的单元格 ID 列表。不支持。 | 4 | |
| height | pixels | 规定表格单元格的高度。不支持。请使用 CSS 代替。 | 4 | |
| nowrap | nowrap | 是否禁用或启用单元格中的文本环绕。不支持。请使用 CSS 代替。 | 4 | |
| rowspan | number | 规定此单元格可横跨的行数。 | 4 | 5 |
| scope | col

colgroup

row

rowgroup | 规定此单元格是否为以下部分提供表头信息：包含此单元格的行（row）的其余部分，包含此单元格的列（col）的其余部分，包含此单元格的行组（rowgroup）的其余部分，包含此单元格的列组（colgroup）的其余部分。不支持。 | 4 | |

| | | | | |
|--------|----------|--------------------------------|---|--|
| valign | top | 规定单元格内容的垂直排列方式。不支持。请使用 CSS 代替。 | 4 | |
| | middle | | | |
| | bottom | | | |
| | baseline | | | |
| width | pixels | 规定表格单元格的宽度。不支持。请使用 CSS 代替。 | 4 | |
| | % | | | |

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 [HTML 5 中事件属性](#)。

TIY 实例(见 table 标签)

<textarea> 标签

定义和用法

定义一个文本区域（text-area）（一个多行的文本输入区域）。用户可在此文本区域中写文本。在一个文本区中，您可输入无限数量的文本。文本区中的默认字体是等宽字体（fixed pitch）。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 5 中有一些新属性，同时不再支持 HTML 4.01 中的一些属性。

例子

```
<textarea rows="3" cols="30">
```

这里是文本域中的文本

```
</textarea>
```

属性

| 属性 | 值 | 描述 | 4 | 5 |
|-----------|------------------|--------------------------------|---|---|
| autofocus | true | 在页面加载时，使这个 textarea 获得焦点。 | 4 | 5 |
| | false | | | |
| cols | number | 规定文本区内可见的列数。 | 4 | 5 |
| disabled | true | 当此文本区首次加载时禁用此文本区。 | 4 | 5 |
| | false | | | |
| form | true | 定义该 textarea 所属的一个或多个表单。 | | 5 |
| | false | | | |
| inputmode | inputmode | 定义该 textarea 所期望的输入类型。 | | 5 |
| name | name_of_textarea | 为此文本区规定的一个名称。 | 4 | 5 |
| readonly | true | 指示用户无法修改文本区内的内容。 | 4 | 5 |
| | false | | | |
| required | true | 定义为了提交该表单，该 textarea 的值是否是必需的。 | | 5 |
| | false | | | |
| rows | number | 规定文本区内可见的行数。 | 4 | 5 |

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 HTML 5 中标准属性。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 [HTML 5 中事件属性](#)。

TIY 实例

Textarea, 如何创建 textarea。

```
<html>
```

```
<body>
```

```
<p>
```

This example cannot be edited

because our editor uses a textarea

for input,

and your browser does not allow

a textarea inside a textarea.

```
</p>
```

```
<textarea rows="10" cols="30">
```

The cat was playing in the garden.

<tfoot> 标签

定义和用法

定义表格的页脚（脚注）。

thead、tfoot 以及 tbody 元素使您有能力对表格中的行进行分组。当您创建某个表格时，您也许希望拥有一个标题行，一些带有数据的行，以及位于底部的一个总计行。这种划分使浏览器有能力支持独立于表格标题和页脚的表格正文滚动。当长的表格被打印时，表格的表头和页脚可被打印在包含表格数据的每张页面上。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 5 中，不支持 <tfoot> 标签的任何属性。

提示和注释：

注释：<tfoot> 内部必须拥有 <tr> 标签！

注释：如果您使用 `thead`、`tfoot` 以及 `tbody` 元素，您就必须使用全部的元素。它们的出现次序是：`thead`、`tfoot`、`tbody`，这样浏览器就可以在收到所有数据前呈现页脚了。您必须在 `table` 元素内部使用这些标签。

例子

```
<table>
```

```
    <thead>
```

```
        <tr>
```

```
            <td>THEAD 中的文本</td>
```

```
        </tr>
```

```
    </thead>
```

```
    <tfoot>
```

```
        <tr>
```

```
            <td>TF00T 中的文本</td>
```

```
        </tr>
```

```
    </tfoot>
```

```
    <tbody>
```

```
        <tr>
```

```
            <td>TBODY 中的文本</td>
```

```
        </tr>
```

```
</tbody>
```

```
</table>
```

属性

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 [HTML 5 中事件属性](#)。

<th> 标签

定义和用法

<th> 标签定义表格内的表头单元格。此 th 元素内部的文本通常会呈现为粗体。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 4.01 中，th 元素的 "bgcolor"、"height"、"width" 以及 "nowrap" 属性是不被赞成使用的。

在 HTML 5 中，仅仅支持 "colspan"、"rowspan" 以及 "scope" 属性。

例子

```
<table>
```

```
<tr>
```

```
<th>Header 1</th>
```

```
<th>Header 2</th>
```

```
</tr>

<tr>

  <td>Cell A</td>

  <td>Cell B</td>

</tr>

</table>
```

属性

属性	值	描述	4	5
abbr	abbr_text	规定单元格中内容的缩写版本。不支持。	4	
align	left right center justify char	规定单元格内容的水平排列方式。不支持。请使用 CSS 代替。	4	
axis	category_names	为单元格定义一个名字。不支持。	4	
bgcolor	rgb(x, x, x) #xxxxxx colorname	规定单元格的背景颜色。不支持。请使用 CSS 代替。	4	
char	character	规定根据哪个字符来进行文本对齐。不支持。请使用 CSS 代替。	4	
charoff	pixels %	规定第一个对齐字符的偏移量。不支持。请使用 CSS 代替。	4	
colspan	number	规定此单元格可横跨的列数。	4	5
headers	header_cells' _id	空格分隔的单元格 ID 列表。不支持。	4	

height	pixels	规定表格单元格的高度。不支持。请使用 CSS 代替。	4	
nowrap	nowrap	是否禁用或启用单元格中的文本环绕。不支持。请使用 CSS 代替。	4	
rowspan	number	规定此单元格可横跨的行数。	4	5
scope	col colgroup row rowgroup	规定此单元格是否为以下部分提供表头信息：包含此单元格的行（row）的其余部分，包含此单元格的列（col）的其余部分，包含此单元格的行组（rowgroup）的其余部分，包含此单元格的列组（colgroup）的其余部分。不支持。	4	5
valign	top middle bottom baseline	规定单元格内容的垂直排列方式。不支持。请使用 CSS 代替。	4	
width	pixels %	规定表格单元格的宽度。不支持。请使用 CSS 代替。	4	

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 HTML 5 中标准属性。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 HTML 5 中事件属性。

TIY 实例(见标签 table)

<thead> 标签

定义和用法

定义表格的表头。

thead、tfoot 以及 tbody 元素使您有能力对表格中的行进行分组。当您创建某个表格时，您也许希望拥有一个标题行，一些带有数据的行，以及位于底部的一个总计行。这种划分使浏览器有能力支持独立于表格标题和页脚的表格正文滚动。当长的表格被打印时，表格的表头和页脚可被打印在包含表格数据的每张页面上。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 5 中，不支持 <thead> 标签的任何属性。

提示和注释：

注释：<thead> 内部必须拥有 <tr> 标签！

注释：如果您使用 thead、tfoot 以及 tbody 元素，您就必须使用全部的元素。它们的出现次序是：thead、tfoot、tbody，这样浏览器就可以在收到所有数据前呈现页脚了。您必须在 table 元素内部使用这些标签。

例子

```
<table>
```

```
  <thead>
```

```
    <tr>
```

```
      <td>THEAD 中的文本</td>
```

```
    </tr>
```

```
  </thead>
```

```
  <tfoot>
```

```
    <tr>
```

```
      <td>TF00T 中的文本</td>
```

```
</tr>
```

```
</tfoot>
```

```
<tbody>
```

```
<tr>
```

```
<td>TBODY 中的文本</td>
```

```
</tr>
```

```
</tbody>
```

```
</table>
```

属性

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述, 请访问 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述, 请访问 [HTML 5 中事件属性](#)。

<time> 标签

定义和用法

<time> 标签定义日期或时间, 或者两者。

实例

<p>

我们在每天早上 <time>9:00</time> 开始营业。

</p>

<p>

我在 <time datetime="2008-02-14">情人节</time> 有个约会。

</p>

TIY

<!DOCTYPE HTML>

<html>

<body>

<p>

我们在每天早上 <time>9:00</time> 开始营业。

</p>

<p>

我在 <time datetime="2010-02-14">情人节</time> 有个约会。

</p>

</body>

</html>

HTML 4.01 与 HTML 5 之间的差异

<time> 标签是 HTML 5 中的新标签。

属性

属性	值	描述
datetime	datetime	定义元素的日期和时间。 如果未定义该属性，则必须在元素的内容中规定日期或时间。

标准属性

<time> 标签支持 HTML 5 中的标准属性。

事件属性

<time> 标签支持 HTML 5 中的事件属性。

<title> 标签

定义和用法

<title> 标签定义文档的标题。

HTML 4.01 与 HTML 5 之间的差异

没有。

提示和注释

注释：一个文档中不能有一个以上的 <title> 元素。

例子

```
<html>

<head>

<title>HTML 5 Tag Reference</title>

</head>

<body>
```

The content of the document.....

</body>

</html>

属性

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述, 请访 [HTML 5 中标准属性](#)。

<tr> 标签

定义和用法

<tr> 标签定义表格中的行。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 5 中, 不支持 <tr> 标签的任何属性。

例子

```
<table>
```

```
  <tr>
```

```
    <td>Cell A</td>
```

```
    <td>Cell B</td>
```

```
  </tr>
```

```
</table>
```

属性

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 [HTML 5 中事件属性](#)。

TIY 实例(见 [table](#))

<tt> 标签

定义和用法

<tt> 标签定义打字机文本。HTML 5 中不再支持该标签。请使用 CSS 代替。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 5 中不支持 <tt> 标签。

提示和注释

提示：请使用 CSS 向元素添加样式。

<u> 标签

定义和用法

<u> 标签定义下划线文本。HTML 5 中不再支持该标签。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 4.01 中不赞成使用 <u> 标签。

在 HTML 5 中不支持 <u> 标签。

 标签

定义和用法

 标签定义无序列表。

HTML 4.01 与 HTML 5 之间的差异

在 HTML 4.01 中，ul 元素的 “compact” 和 “type” 属性是不被赞成使用的。在 HTML 5 中，不再支持这两个属性。

提示和注释

提示：请使用 CSS 来定义列表的类型。

例子

```
<ul>

  <li>Coffee</li>

  <li>Tea</li>

</ul>
```

属性

标准属性

class, contenteditable, contextmenu, dir, draggable, id, irrelevant, lang, ref, registrationmark, tabindex, template, title

如需完整的描述，请访问 [HTML 5 中标准属性](#)。

事件属性

onabort, onbeforeunload, onblur, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onerror, onfocus, onkeydown, onkeypress, onkeyup, onload, onmessage, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup, onmousewheel, onresize, onscroll, onselect, onsubmit, onunload

如需完整的描述，请访问 [HTML 5 中事件属性](#)。

TIY 实例(见标签 li)

<video> 标签

定义和用法

<video> 标签定义视频，比如电影片段或其他视频流。

实例

一段简单的 HTML5 视频：

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

您的浏览器不支持 video 标签。

```
</video>
```

TIY

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<body>
```

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

```
your browser does not support the video tag
```

```
</video>
```

```
</body>
```

```
</html>
```

HTML 4.01 与 HTML 5 之间的差异

<video> 标签是 HTML 5 的新标签。

提示和注释

提示：可以在开始标签和结束标签之间放置文本内容，这样老的浏览器就可以显示出不支持该标签的信息。

属性

属性	值	描述
autoplay	autoplay	如果出现该属性，则视频在就绪后马上播放。
controls	controls	如果出现该属性，则向用户显示控件，比如播放按钮。
height	pixels	设置视频播放器的高度。
loop	loop	如果出现该属性，则当媒介文件完成播放后再次开始播放。
preload	preload	如果出现该属性，则视频在页面加载时进行加载，并预备播放。 如果使用 “autoplay”，则忽略该属性。
src	url	要播放的视频的 URL。
width	pixels	设置视频播放器的宽度。

标准属性

<video> 标签支持 HTML 5 中的标准属性。

事件属性

<video> 标签支持 HTML 5 中的事件属性。

<完>

声明：相关测试代码仅用于学习参考。笔者不保证内容的正确性。通过使用资料内容随之而来的风险与笔者无关。当使用资料时，代表您已接受了资料的使用条款和隐私条款。本资料的所有内容仅供测试，对任何法律问题及风险不承担任何责任。