

# HTML5 与 CSS3 的应用

高洛峰 ( 草稿 – PHPCMS 在线课专用版 )

2014/9

HTML5 与 CSS3 的应用 .....	1
1 现在是用 HTML5 的时候 .....	2
1.1 HTML4+CSS2 与 HTML5+CSS3 区别.....	3
2 HTML5 语法的改变 .....	3
2.1 HTML5 中的标记方法 .....	3
2.2 HTML5 做到了与之前版本的兼容.....	4
2.3 HTML5 语法改变的简单示例 .....	5
3 HTML5 新增的元素和废除的元素 .....	5
3.1 新增的结构元素.....	5
3.2 HTML5 表单增加的新特性 .....	8
3.3 新增多媒体播放元素.....	11
3.4 新增的其他元素.....	12
3.5 废除的元素.....	13
4 HTML5 新增的属性和废除的属性 .....	13
5 HTML5 的高级功能介绍 .....	15
6 CSS3 概述 .....	17
6.1 CSS3 在选择器上面的支持.....	17
6.2 CSS3 在样式上的支持.....	17
6.3 CSS3 对于动画的支持.....	17
6.4 在开发中该如何去用 CSS3.....	18
7 CSS3 中的样式选择器.....	18
7.1 基本选择器.....	18
7.2 多元素的组合选择器.....	19
7.3 属性选择器.....	19
7.4 结构性伪类选择器.....	20
8 CSS3 的样式属性.....	22
8.1 使用 CSS3 属性前的准备.....	22
8.2 边框属性.....	23
8.3 背景属性.....	24
8.4 文本属性.....	25
8.5 用户界面属性.....	25
8.6 动画属性.....	26
8.7 多列布局属性.....	26
8.8 渐变属性.....	27
8.9 透明属性.....	27
8.10 旋转属性.....	28

8.11 服务器端字体属性 .....	28
9 使用 HTML5 中新增元素来构建网页 .....	28
10 LESS CSS 框架简介 .....	33

虽然我们现在已经普遍使用了 HTML4 和 CSS2.1，但是我们还可以做得更好！我们可以重组我们代码的结构并能让我们的页面代码更富有语义化特性。我们可以缩减带给页面美丽外观样式代码量并让他们有更高的可扩展性。如今，HTML5 和 CSS3 的技术和应用环境已经日趋成熟，它们将书写 Web 前端领域的新历史，HTML5 将成为新一代的 Web 技术标准，必定会改变整个 Web 应用领域的游戏规则，它在给新的 Web 应用带来无限可能性的同时，还能带来更快、更好、更炫的用户体验。CSS3 也将为 Web 开发带来革命性的影响，很多以前需要 JavaScript 和 Ajax 才能实现的复杂效果（如背景、圆角、3D 动画，等等），现在使用 CSS3 就能简单地实现，极大地提高了程序的开发效率。HTML5 和 CSS3 已经成为全球各大互联网巨头必争之地，Microsoft、Google、Apple、Mozilla、Opera 等浏览器厂商对它们的支持犹如一场竞赛，似乎已经成为衡量它们的浏览器性能优劣的一个重要标准。

## 1 现在是用 HTML5 的时候

近几年来，HTML5 的大潮似乎是势不可挡。在桌面应用环境中，各大浏览器纷纷地朝着支持 HTML5、结合 HTML5 的方向迈进着；在移动平台上，HTML5 已经成为了 iOS、Windows Phone 和 Android 移动互联网的唯一选择。放眼望去，仿佛四野皆充满着 HTML5 千秋万代一统江湖的高呼。

HTML5 是 W3C 与 WHATWG 合作的结果，WHATWG（Web Hypertext Application Technology Working Group）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。HTML5 刚发布时由于各浏览器之间的标准不统一，开发者的时间都浪费在解决 Web 浏览器之间的兼容性上。但由于 W3C 和 WHATWG 对 HTML5 新版本的制定，以及近年来对 HTML5 的使用，再加上各大浏览器鼎力支持，已经有非常丰富的兼容性解决方案，HTML5 在老版本的浏览器上也可以正常运行，正是因为保障了兼容性才能让人毫不犹豫的用 HTML5 开发网站。HTML5 的目标是将 Web 带入一个成熟的应用平台，在 HTML5 平台上，视频、音频、图像、动画，以及同电脑的交互都被标准化。另外，HTML5 内部并没有封装什么很复杂的、不切实际的功能，而是封装了简单实用的功能，HTML5 内部功能不是革命性的而是发展性的，并不代表 HTML4 创建出来的网站必须全部要重建，只会要求各 Web 浏览器今后能正常运行用 HTML5 开发出来的功能。虽然 HTML5 还有一些不足，但 HTML5 应用已经是大势所趋，HTML5 已经开始提供各种各样 Web 应用上的解决方案。

最新版本的 Safari、Chrome、Firefox 以及 Opera 支持某些 HTML5 特性。IE 9 将支持某些 HTML5 特性，IE10 将全面支持 HTML5。IE8 及以下 IE 版本对 HTML5 标签的支持是有限的，我们可以通过在网页中添加脚本的方式来解决目前 IE 浏览器对 HTML5 支持的问题。让 IE（包括 IE6）支持 HTML5 元素，我们需要在 HTML 头部添加以下 JavaScript，Opera，FireFox 等其他非 IE 浏览器就会忽视这段代码，也不会存在 http 请求。

```
<!--[if IE]>
    <script src="http://yun.itxdl.cn/js/html5.js"></script>
<![endif]>
```

上面这段代码仅会在 IE 浏览器下运行，还有一点需要注意，在页面中调用 html5.js 文件必须添加在页面的 head 元素内，因为 IE 浏览器必须在元素解析前知道这个元素，所以这个 js 文件不能在页面底部调用。这个 html5 的 js 文件如果觉得这样会影响你的网页打开速度，你可以把 html5 的 js 文件直接下载下来让后上传到自己的服务器单独调用。

## 1.1 HTML4+CSS2 与 HTML5+CSS3 区别

互联网发展越来越快，有些东西总会要被取代，HTML4+CSS2 将渐渐退出舞台，而 HTML5+CSS3 将取而代之，HTML5 规范是 2004 年新成立的 WHATWG 工作组创立的；2006 年 W3C 加入 HTML，并于 2008 年发布了 HTML5 工作草案，2009 年，XHTML2 工作组停运；2010 年 HTML5 开始解决实际问题，渐渐的各大浏览器厂家开始升自己的产品以支持 HTML5+CSS3，HTML5 得到了持续的完善，2012 年 HTML5 完成规范编写工作。HTML5 以 HTML4 为基础，对 HTML4 进行了大量的修改。学习 HTML5 则需要掌握 HTML5 与 HTML4 在基本语法上有什么区别；了解在 HTML5 中新增了哪些元素和属性，删除了哪些 HTML4 中的元素和属性，为什么要删除这些元素，用什么元素或方法来取代这些被删除的元素和属性；还需要掌握什么是全局属性；

## 2 HTML5 语法的改变

HTML5 的一个很大的目标就是提高浏览器之间的兼容性，需要有一个统一的标准，因此 HTML5 重新定义了一套在现有 HTML4 基础上修改而来的语法，使它运行在知浏览器都能够符合这个通用标准。

### 2.1 HTML5 中的标记方法

HTML5 的文件扩展符号内容类型(Content-Type)保持不变的，扩展符仍然为“.html”或“.htm”，内容类型仍然为“text/html”。

#### 1. HTML 5 的 DTD 的声明

在编写 HTML5 文档时，要求指定文档类型，以确保浏览器能在 HTML5 的标准模式下进行渲染。在 HTML5 中刻意不使用版本声明，一份文档将会适用于所有版本的 HTML，非常简便，HTML5 中的 DTD 声明方法如下：

```
<!DOCTYPE html>
```

在 HTML5 中不区分大小写， 引号不区分是单引号还是双引号。

2. 设置页面字符编码

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

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

从 HTML5 开始， 对于文件的字符编码推荐使用 UTF-8。

2.2 HTML5 做到了与之前版本的兼容

为了保证 HTML5 能与之前的 HTML 版本达到最大的兼容， HTML5 对一些元素标记的省略、boolean 值的属性，以及引号的省略这几方面进行了兼顾， 确保与之前版本的 HTML 达到兼容。

1. 可以省略标记的元素

元素的标记分为三种情况：不允许写结束标记的元素，可以省略结束标记的元素和开始标记结结束标记都可省略三种类型。不允许写结束标记的元素是指不允许使用开始标记和结束标记将元素括起来的形式，例如，换行标记正确的书写方式为“<br/>”，而“<br>...</br>”的书写就是错误的。可以省略全部标记的元素是指该元素可以完全被省略，当然被省略的标记还是以隐式的方式存在的，例如“<html>”元素省略不写时还是存在的。针对这三种情况的列举清单如表 1-1 所示：

表 1-1 三种情况列举清单

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

2. 具有 boolean 值的属性

在 html 中有一些元素的属性，当只写属性名称而不指定属性值时，表求属性值为 true，如果设置该属性值为 false，则不使用该属性即可。另外，要想将属性值设定 true 时，也可以将属性名设定为属性值， 或将空字符串设定为属性值。例如 input 元素中的 disabled 与 readonly 就是这样的属性。

3. 引号使用

在 html 中使用属性时，属性值可以使用双引号，也可以使用单引号括起来，在 HTML5 中做了一些改进，当属性值不包括空字符串、“<”、“>”、“=”、单引号、双引号等字符时，属性两边的引号是可以省略的。

## 2.3 HTML5 语法改变的简单示例

在这个示例中，将前面介绍的几个 HTML5 应用方法集成到本例中，另外，没有特别说明使用什么浏览器，本章默认使用的都是 Firefox 4 浏览器。代码演示结果如图 3-1 所示：

```
1 <!DOCTYPE html>                                <!-- 在HTML5中声明DTD的方法 -->
2                                                    <!-- 可以全部省略标记html和head及body -->
3 <meta charset="UTF-8">                          <!-- 使用HTML5的方法指定字符编码 -->
4 <title>HTML语法演示</title>
5
6 <br/>这段代码是HTML5语法编写的                <!-- 不允许使用结束标记元素br -->
7 <p>第一段                                       <!-- 可以省略结束标记的元素p -->
8 <p>第二段
9
10 <input type="checkbox" checked>                   <!-- 只写属性不写属性值代表属性为true -->
11 <input type="checkbox">                           <!-- 不写属性代表属性值为false -->
12 <input type="checkbox" checked="checked">         <!-- 属性值=属性名，代表属性为true -->
13 <input type="checkbox" checked>                   <!-- 属性值=空字符串，代表属性为true -->
14
15 <input type="text">                             <!-- 属性值使用双引号 -->
16 <input type='text'>                             <!-- 属性值使用单引号 -->
17 <input type=text>                               <!-- 属性值不使用引号 -->
```

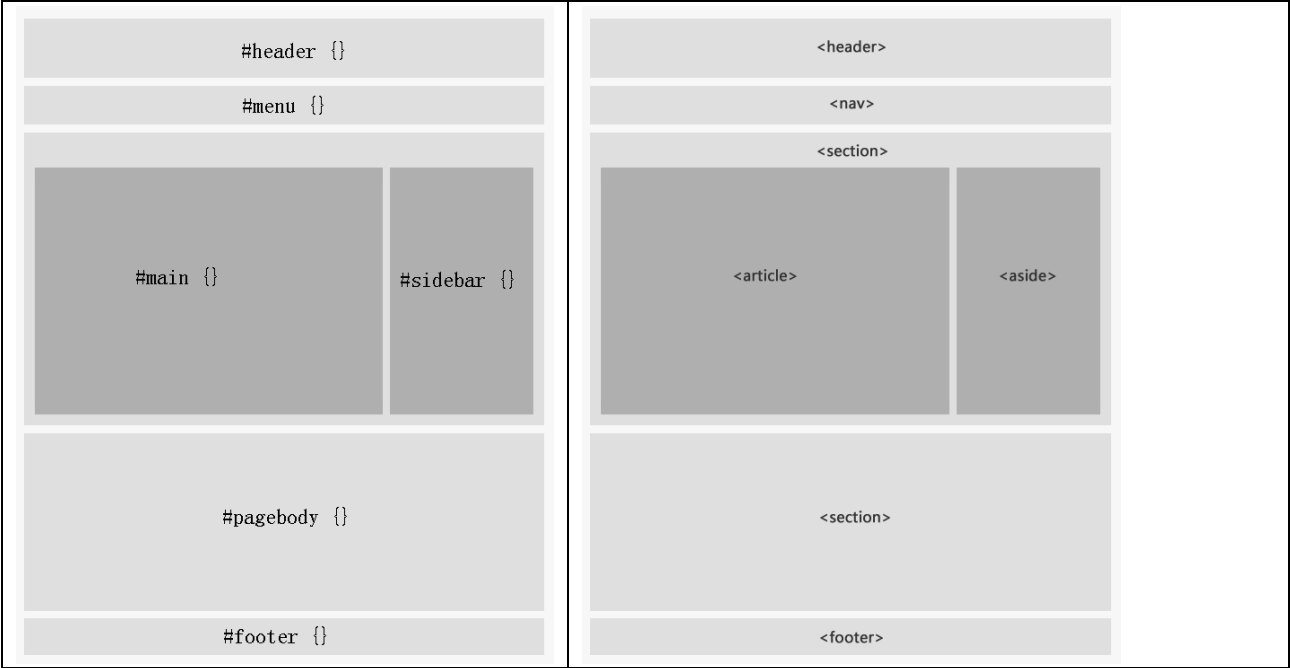
## 3 HTML5 新增的元素和废除的元素

HTML 5 提供了一些新的元素和属性，反映典型的现代用法网站。其中有些是技术上类似 <div> 和 <span> 标签，但有一定含义，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、小屏幕装置和视障人士使用。同时为其他浏览要素提供了新的功能，通过一个标准接口，如 <audio> 和 <video> 标记。还有一些过时的 HTML 4 标记将取消，其中包括纯粹用作显示效果的标记，如 <font> 和 <center>，因为它们已经被 CSS 取代，还有一些是透过 DOM 的网络行为。

### 3.1 新增的结构元素

曾经，网页编写经常会跟频繁使用基于 table 的没有任何语义的布局。不过最终还是慢慢的接受了相对更语义化的<div>布局替代了 table 布局，并且开始调用外部样式表。但不幸

的是，复杂的网页设计需要大量不同的标签结构代码，我们把它叫做“<div>-soup”综合症。我们可以重组我们代码的结构并能让我们的页面代码更富有语义化特性。我们可以缩减带给页面美丽外观样式代码量并让他们有更高的可扩展性。现在，HTML5 和 CSS3 正跃跃欲试的等待大家。下图是 HTML4 的 DIV 和 HTML5 新增标记布局的对比，如图 1-1 所示：



```

1 <style>
2   div#news    {width:660px;margin:0 auto}
3   div.section {width:100%;}
4   div.article {width:500px;float:left;}
5   div.header  {width:100%;text-align:center}
6   div.content {width:100%;background:#ddd}
7   div.footer  {width:100%;background:#ddd}
8   div.aside   {width:150px;float:right}
9 </style>
10 <div id="news">
11   <div class="section">
12     <div class="article">
13       <div class="header">
14         <h1>Div 布局示范</h1>
15         <p>发表于2014年09月10日</p>
16       </div>
17       <div class="content">
18         <p>内容文本等等.</p>
19         <p>内容文本等等.</p>
20         <p>内容文本等等.</p>
21       </div>
22       <div class="footer">
23         <p>Tags: HTML, 代码, 演示</p>
24       </div>
25     </div>
26     <div class="aside">
27       <div class="header">
28         <h1>附属信息</h1>
29       </div>
30       <div class="content">
31         <p>内容文本等等.</p>
32         <p>内容文本等等.</p>
33         <p>内容文本等等.</p>
34         <p>内容文本等等.</p>
35       </div>
36       <div class="footer">
37         <p>Tags: HTML, 演示</p>
38       </div>
39     </div>
40   </div>
41 </div>

```

```

1 <style>
2   section {width:660px;margin:0 auto}
3   article {width:500px;float:left;}
4   header  {width:100%;background:#ddd;text-align:center}
5   footer  {width:100%;background:#ddd}
6   aside   {width:150px;float:right}
7 </style>
8 <section>
9   <article>
10     <header>
11       <h1>Div 布局示范</h1>
12       <p>发表于2014年09月10日</p>
13     </header>
14     <section>
15       <p>内容文本等等.</p>
16       <p>内容文本等等.</p>
17       <p>内容文本等等.</p>
18     </section>
19     <footer>
20       <p>Tags: HTML, 代码, 演示</p>
21     </footer>
22   </article>
23   <aside>
24     <header>
25       <h1>附属信息</h1>
26     </header>
27     <section>
28       <p>内容文本等等.</p>
29       <p>内容文本等等.</p>
30       <p>内容文本等等.</p>
31       <p>内容文本等等.</p>
32     </section>
33     <footer>
34       <p>Tags: HTML, 演示</p>
35     </footer>
36   </aside>
37 </section>
38 </section>
39

```



图 1-1 DIV+CSS 布局 VS HTML5 结构元素布局

尽管这有些勉强，但上面这个实例还是可以说明使用 HTML4 对一个复杂的设计进行代码化后依然过于臃肿。不过值得激动的是，HTML5 解决“<div>-soup”综合症并带给我们一套新的结构化元素。这些新的 HTML5 元素富有更细致的语义从而代替了那些毫无语义的 <div> 标签，并同时为 CSS 的调用提供了“自然”的 CSS 钩子。这种语义化的特性不仅提升了我们网页的质量和语义，并且大量减少了曾经代码中用于 CSS 必须调用的 class 和 id 属性。

事实上，后面课程中介绍的 CSS3 也是可以忽略掉所有 class 和 id 的。结合了富有新的语义化标记的 HTML5，CSS3 为 web 设计师们的网页提供了神一般的力量。新增的结构元素及说明如下表 1-2 所示：

表 1-2 新增的结构元素及说明

元素名	说明
section	表示页面中的一个内容区块，比如章节、页眉、页脚或页面中的其他部分。它可以与 h1、h2、h3、h4、h5、h6 等元素结合起来使用，标示文档结构。
article	表示页面中的一块与上下文不相关的独立内容，例如博客中的一篇文章。
aside	表示 article 元素内容之外的，与 article 元素内容相关的辅助信息。
header	表示页面中一个内容区块或整个页面的标题。
hgroup	用于整个页面或页面中一个内容区块的标题进行组合。
footer	表示整个页面或页面中一个内容区块的脚注。一般来说，它会包含创作者的姓名，创建日期及创作者联系信息。
nav	表示页面中导航链接的部分
figure	表示一段独立的流内容，一般表示文档主体流内容中的一个独立单元，使用 figcaption 元素为 figure 元素组添加标题。

### 3.2 HTML5 表单增加的新特性

在创建 Web 应用程序时，会用到大量的表单元素，在 HTML5 中关于表单给我们增加了很多诱人的属性，强化了表单元素的功能，使得关于表单的编写更快、更方便。先了解一下 form 的新属性，如表 1-3 所示：

表 1-3 HTML5 中 form 表单的新增属性

属性名	说明及示例	支持浏览器
from	在 HTML4 中， 表单内的从属元素必须书写在表单内部，但是在 HTML5 中，可以书写在页面上任何位置，只要给该元素指定一个 form 属性，属性值为该表单的 id 就可以了，这样做的好处是给页面中的元素添加样式时更方便。 <pre>1 &lt;form id="formtest"&gt; 2   &lt;input type="text"&gt; 3 &lt;/form&gt; 4 &lt;textarea form="testform"&gt;&lt;/textarea&gt;</pre>	Opera 10
formation	在 HTML4 中，一个表单内的所有元素都只能通过表单的 action 属性统一提交到另一个页面，而在 HTML5 中可以给所有的提交按钮都增加不同的 formation 属性，使得点击不同的按钮，将表单提交到不同的页面。 <pre>1 &lt;form id="formtest" action="server.php"&gt; 2   &lt;input type="submit" name="k1" value="v1" formation="k1.php" formmethod="post"&gt;提交到k1 3   &lt;input type="submit" name="k2" value="v2" formation="k2.php" formmethod="get"&gt;提交到k2 4   &lt;input type="submit" name="k3" value="v3" formation="k3.php" formmethod="post"&gt;提交到k3 5 &lt;/form&gt;</pre>	没有浏览器支持
formmethod	在 HTML5 中可以使用 formation 属性来对每个表单元素分别指定不同的提交页面，同时也可以使用 formmethod 属性来对每个表单元素分别指定不同的提交方法。	没有浏览器支持
placeholder	当文本框“text”或“textarea”处于未输入状态时文本框中显示的输入提示，只要在表单中加上 placeholder 属性，再指定提示文字就可以实现。 <pre>&lt;input type="text" placeholder="please input value!"&gt;</pre>	Safaru 4 Chrome 3 Firefox 4



<b>autofocus</b>	<p>当页面一打开，文本框、选择框或按钮可以自动获得光标焦点。不过一个页面只能有一个控件使用该属性。例如登录页面一打开焦点在用户名上。</p> <pre>&lt;input type="text" name="username" autofocus&gt;</pre>	<p>Safari 4 Chrome 3 Firefox 4</p>
<b>list</b>	<p>该属性需要和 <code>datalist</code> 元素(HTML5 新增)一起使用，<code>list</code> 属性的值就是 <code>datalist</code> 的 <code>id</code>。使用 <code>datalist</code> 元素设置一个选择列表（类似 <code>select</code>），当用户输入的值在列中可以出现选择提示，当用户相要设定的值不在选择列表之内时，允许其自行输入。</p> <pre> 1 text: &lt;input type="text" name="bookname" list="bookname"&gt; 2 &lt;datalist id="bookname"&gt; 3   &lt;option value="php book"&gt;php book&lt;/option&gt; 4   &lt;option value="good"&gt;good&lt;/option&gt; 5   &lt;option value="php good book"&gt;php good book&lt;/option&gt; 6 &lt;/datalist&gt; </pre>	Opera 10
<b>autocomplete</b>	<p>辅助输入所有的自动完成功能，是一个节省输入时间，同时也十分方便的功能，在 HTML5 之前，因为谁都可以看见输入的值，所以存在安全隐患。对于 <code>autocomplete</code> 属性，可以指定“on”、“off”和“”（不指定）三种值，不指定时，使用浏览器的默认值。把该属性设为 on 时，可以显式指定候补输入的数据列表。使用 <code>datalist</code> 元素与 <code>list</code> 属性提供候补输入的数据列表，自动完成时，可以将该 <code>datalist</code> 元素中的数据作为候补输入的数据在输入框中自动显示。</p> <pre>&lt;input type="text" name="bookname" autocomplete="on" list="bookname"&gt;</pre>	Opera 10

在 HTML5 中，除了 form 表单元素增加了一些好用的属性外，大幅度增加与改良 input 元素的种类，可以简单地使用这些元素来实现 HTML5 之前需要使用 JavaScript 才能实现的许多功能。如表 1-4 所示：

表 1-4 HTML5 中增加与改良的 input 元素

种类	说明及示例	支持浏览器
<b>search</b>	<p><code>search</code> 类型的 input 元素是一种专门用来输入搜索关键词的文本框。<code>Search</code> 类型与 <code>text</code> 类型仅仅在外观上有区别，在 Safari 4 浏览器中，它的外观为操作系统默认的圆角矩形文本框，但这个外观可以用 CSS 样式进行改写。在其他浏览中，它的外观暂与 <code>text</code> 类型的文本框外观相同，但可以用 css 样式进行改写。</p> <pre> &lt;style&gt;   input[type="search"] {-webkit-appearance: textfield;} &lt;/style&gt; &lt;input type="search" name="keywords"&gt; </pre>	所有最新版本浏览器
<b>tel</b>	<p>被设计为用来输入电话号码的专用文本框，它没有特殊的校验规则，不强制输入数字，因为许多电话号码通常都带有其它文字，但开发者可以通过 <code>pattern</code> 属性来指定对于输入的电话号码格式的验证。</p>	所有最新版本浏览器
<b>url</b>	<p>是一种专门用来输入 url 地址的文本框，提交时如果该文本框中内容不是 url 地址格式的文字，则不允许提交。</p>	Opera 10
<b>email</b>	<p>是一种专门用来输入 email 地址的文本框，提交时如果该文本框中内容不是 email 地址格式的文字则不允许提交，但是它并不检查该 email 地址是否存在，提交时该文本框可以为空，除非加了 <code>required</code> 属性。Email 类型的文本框还具有一个 <code>multiple</code> 属性，它允许在该文本框中输入一个串以逗号分隔的 email 地址。当然并不强制要求用户输入该 email 地址列表。</p>	Opera 10
<b>Datetime, date, month, week, time, datetime-local</b>	<p><code>datetime</code> 类型的 input 元素是一种专门用来输入 UTC 日期和时间的文本框，并且在提交时对输入的日期格式进行有效检查。</p> <p><code>date</code> 类型的 input 元素是深受开发者喜爱的一种元素，以日历的形式方便用户输入。</p> <p><code>month</code> 类型的 input 元素是一种专门用来输入月份的文本框，并且在提交时对输入</p>	Opera 10

	<p>的月份格式进行有效检查。</p> <p><b>week</b> 类型的 <b>input</b> 元素是一种专门用来输入周号的文本框，并且在提交时对输入的周号格式进行有效检查。它可能是一个简单的输入文本框，允许用户输入一个数字，也可能更复杂更精确。它的格式类型类似于“2014-W10”，代表 2014 年第 10 周。</p> <p><b>time</b> 类型的 <b>input</b> 元素是一种专门用来输入时间的文本框，并且在提交时对输入的时间格式进行有效检查。它的外观取决于浏览器，可能是简单的文本框，只是提交时检查是否在其中输入了有效的时间，也可能以时钟形式出现，还可以携带时区。</p> <p><b>datetime-local</b> 类型的 <b>input</b> 元素是一种专门用来输入本地日期和时间的文本框，并且在提交时对输入的日期和时间格式进行有效检查。</p> <pre>&lt;input name="test1" type="datetime"&gt; &lt;input name="test2" type="date" value="2014-10-20"&gt; &lt;input name="test3" type="month" value="2014-10"&gt; &lt;input name="test4" type="week" value="2014-W10"&gt; &lt;input name="test5" type="time" value="11:11"&gt; &lt;input name="test6" type="datetime-local"&gt;</pre>	
<b>number</b>	<p><b>number</b> 类型的 <b>input</b> 元素是一种专门用来输入数字的文本框，并且在提交时检查输入的内容是否为数字。它具有 <b>min</b>、<b>max</b> 与 <b>step</b> 属性。</p> <pre>&lt;input name="test7" type="number" value="50" min="10" max="100" step="5"&gt;</pre>	Opera 10
<b>range</b>	<p>是一种只允许输入一段范围内数值的文本框，它具有 <b>min</b>、<b>max</b> 与 <b>step</b> 属性，默认范围是 0 与 100。</p> <pre>&lt;input name="test8" type="range" value="50" min="10" max="100" step="5"&gt;</pre>	Opera 10
<b>color</b>	用来选择颜色，它提供了一个颜色选择器。	BlackBerry (黑莓)
<b>file</b>	文件选择文本框，与 HTML4 最大的不同是，可以通过指定 <b>multiple</b> 属性，一次选择多个文件， <b>value</b> 属性的值为用逗号分隔的一个或多个文件名。同时，通过把 MIME 类型指定给 <b>accept</b> 属性，可以限制选择文件的种类。	所有最新版 本浏览器

到目前为止，对于这些 **input** 的新增种类来说，支持最多、最全面的是 Opera 10 浏览器。对于不支持新增 **input** 元素的浏览器来说，统一将这些 **input** 元素视为 **text** 类型，另外，HTML5 中也没有规定这些元素在各浏览中的外观形式，所以同样的 **input** 元素在不同的浏览器中可能会有不同的外观。

在 HTML5 追加了新的 **output** 元素，也增加了大量在提交时对表单与表单内新增元素进行内容有效性验证的功能。其中 **output** 元素定义不同类型的输出，比如计算结果或脚本的输出。**output** 元素必须从属于某个表单，写在表单的内部，或者对它添加 **form** 属性，目前，只被 Opera 10 浏览器支持。**output** 结合 **range** 元素使用代码如下：

```
1 <!DOCTYPE html>
2 <meta charset="UTF-8">
3 <form id="formtest">
4   <input name="test" onchange="num.value=value" type="range" value="50" min="10" max="100" step="5" />
5   <output name="num">50</output>
6 </form>
```

在 HTML5 中，通过对表单元素使用一些新增属性，能实现在表单提交时执行自动验证的功能。新增的 **required** 属性可以应用在大多数输入元素上，在提交表单时，如果元素中内容为空白，则不允许提交，同时在浏览器中显示信息提示文字，提示用户这个元素中必须输入内容。如果要求输入内容必须符合一定的格式，可以对 **input** 元素使用 **pattern** 属性，并且将属性值设为某个格式的正则表达式，在提交表单时会检查其内容是否符合给定格式，

不符合就不允许提交，同时在浏览器中显示信息提示文字，提示输入的内容必须符合给定格式。

```
1 <!DOCTYPE html>
2 <meta charset="UTF-8">
3 <form action="demo_form.php" method="get">
4   Name1: <input type="text" name="usr_name1" required="required" /><br>
5   Name2: <input type="text" name="usr_name2" required="required" pattern="\d.*$"/><br>
6   Name3: <input type="text" name="usr_name3" required="required" pattern="\d.*$" placeholder="请输入数字!"/>
7   <input type="submit" />
8 </form>
```

例如，在开发阶段我们可能想把表单临时提交一下，又不想让它进行表单中所有元素内容的有效性检查，有两种方法可以取消表单验证。一种是在 form 元素中使用 novalidate 属性，让整个表单验证全都失效，另一种是利用 input 元素中使用 formnovalidate 属性，让指定的单个 input 元素验证失效。

### 3.3 新增多媒体播放元素

在 HTML5 之前，要在网站上展示视频、音频、动画等多媒体信息，除了使用第三方自主开发的播放器之外，使用的最多的工具应该算是 FLASH 了，但是它们都需要在浏览器中安装各种插件才能使用，有时速度很慢。HTML5 的出现使这一局面得到了改观。在 HTML5 中，提供了音频的标准接口，多媒体播放再也不需要安装插件了，只需要一个支持 HTML5 的浏览器就可以了。本节介绍 video 元素和 audio 两个 HTML5 新增加的元素，它们分别用来处理视频与音频数据。目前 Safari 3 以上、Firefox 4 以上、Opera 10 以上，以及 Google chrome 3.0 以上的浏览器都实了对这两个媒体元素的支持。以 audio 元素为例，只要把播放视频的 url 地址指定给该元素的 src 属性就可以了。代码如下所示：

```
1 <!doctype html>
2 <meta charset="utf-8">
3 <audio src="test.mp3" controls>您的浏览器不支持音频audio功能</audio>
4
5 <audio controls>
6   <source src="test.mp3" type="audio/mp3" >
7   <source src="test.ogg" type="audio/ogg" >
8   <source src="test.wav" type="audio/wav" >
9   您的浏览器不支持音频audio功能
10 </audio>
```

视频 video 元素的使用方法和 audio 相似，只要设定好元素的长、宽等属性。

```
1 <!doctype html>
2 <meta charset="utf-8">
3 <!-- 视频标签 -->
4 <video src="test.webm" controls loop width="1600" height="1000" >
5   您的浏览器不支持视频video标签
6 </video>
7 <!-- 完整视频格式 -->
8 <video controls>
9   <source src="test.webm" type="video/webm">
10  <source src="test.ogg" type="video/ogg">
11  您的浏览器不支持视频video标签
12 </video>
```

各浏览器对于各种媒体的类型及编码格式的支持情况都各不相同，你可以通过 source 元素来为同一个媒体数据指定多个播放格式与编码方式，以确保浏览器可以从中选择一种自

已支持的播放格式进行播放，浏览器的选择顺序为代码中的书写顺序，它会从上往下判断自己对该播放格式是否支持，选到为止。各浏览器对于编码格式的支持情况，如表 1-5 所求。

表 1-5 各浏览器对于编码格式的支持情况

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

两个媒体元素 `audio` 和 `video` 通过一些常用属性的使用，可以定义和扩展一些功能，它们所具有的属性大致相同，两个元素常用的属性如下表 1-6 所示：

表 1-6 `audio` 和 `video` 常见属性说明

属性名	值	说明
<code>src</code>	<code>url</code>	在该属性中指定媒体数据的 URL 地址。
<code>autoplay</code>	<code>autoplay</code>	在该属性中指定媒体是否在页面加载后自动播放。
<code>preload</code>	<code>none</code> <code>metadata</code> <code>auto(默认)</code>	在该属性中指定视频或音频数据是否预加载。 如果使用预加载，浏览器会预先将数据进行缓冲，这样可以加快播放的速度。有三个可选择值： <code>none</code> 表示不进行预加载 <code>metadata</code> 表示只预加载媒体的元数据（媒体字节数、第一帧、播放列表、持续时间等） <code>auto</code> 表示预加载全部视频或音频
<code>poster</code>	图片 url	<code>video</code> 元素独有属性，当视频不可用时，向用户展示一幅替代的图片。
<code>loop</code>	<code>loop</code>	如果出现该属性，则当媒介文件完成播放后再次开始播放。
<code>controls</code>	<code>controls</code>	指定是否为视频或音频添加浏览器自带的播放用的控制条，控制条中具有播放、暂停等按钮。
<code>width/height</code>	<code>pixels</code>	<code>video</code> 元素独有属性，设置视频的宽度与高度，以像素为单位。

除了上表中的提供的属性外， 还有一些属性、一些方法和多个事件可用，但需要结合 JavaScript 来实现一些特定的效果。例如，通过 `error` 属性可以处理出现的错误等，使用 `networkState` 属性读取当前网络状态。使用 `play` 方法来播放媒体，使用 `load` 方法重新载入媒体进行播放等，通过 `timeupdate` 事件来通知当前播放位置的改变，再结合 JavaScript 来显示当前的播放进度等等。

### 3.4 新增的其他元素

在 HTML5 中新增加了很多元素，除了结构元素、表单元素和媒体元素外，还新增加了以下一些比较常用的元素。如表 1-7 所示，列出部分 HTML5 新增元素：

表 1-7 HTML5 中增加的其它元素

新元素	说明及示例
<code>mark</code>	表示页面中需要突出显示或高亮显示的，对于当前用户具有参考作用的一段文字。它通常使用于引用原文的时候，目的是吸引当前用户的注意。例如，对网页全文检索某个关键词时显示的检索结果。

	<code>&lt;mark&gt;HTML5 内容&lt;/mark&gt;</code>
<b>wbr</b>	该元素表示软换行，br 元素表示此处必须换行，而 wbr 元素的意思是浏览器窗口或父级元素的宽度足够宽时（没有必要换行时），不进行换行，而当宽度不够时，主动在此处进行换行，但是对于中文没有多大用。
<b>progress</b>	定义任何类型的任务的进度。可以使用<progress>标签来显示 javascript 中耗费时间的函数的进度
<b>meter</b>	定义预定义范围内的度量。仅用于已知最大和最小值的度量
<b>canvas</b>	定义图形,绘制路径，矩形，圆形，字符以及添加图像的方法 Canvas 元素本身没有绘图能力，所有的绘制工作必须在 javascript 内部完成渐变
<b>command</b>	表示命令按钮，例如单选按钮、复选框或按钮。 <code>&lt;command onclick="cut()" label="cut"&gt;</code>
<b>details</b>	表示用户要求得到并且可以得到的细节信息，即定义元素的细节。
<b>keygen</b>	标签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）。
<b>ruby</b>	标签定义 ruby 注释（中文注音或字符）在东亚使用

## 3.5 废除的元素

HTML 在进步，有一些落后的元素就要被淘汰掉，在 HTML5 中废除了一些元素，像有一些功能标签纯粹是为页面展示服务的，完全可以用 CSS 来替代统一进行编辑，所以 basfont、big、center、font、s、strike、tt、u 这些元素在 HTML5 中废除了。在 HTML5 中不再使用 frame 框架了，由于 frame 框架对网页可用性存在负面影响，所以像 frameset、frame 和 noframes 元素在 HTML5 中已不支持了，只支持 iframe 框架。另外，有一些元素只有部分浏览器支持，也在 HTML5 中被废除，像 bgsound 和 marquee 元素只有 IE 支持，applet 元素可以使用 embed 或 object 元素替代，所以也被废除。还有废除 rb 元素使用 ruby 元素替代，废除 acronym 元素由 abbr 替代，以及我们不常使用的 dir、isindex、listing、xmp、nextid、plaintext 都有替代的元素，所以都在 HTML5 中废除了。

## 4 HTML5 新增的属性和废除的属性

在 HTML5 中，在增加和废除了很多元素的同时，也增加和废除了很多属性。像我们前面介绍过的与表单相关的属性就有很多，例如，autofocus、placeholder、form、required、autocomplete、min、max、step、multiple、pattern、formaction、formmethod、novalidate 属性等与特定的表单元素结合使用。在 HTML5 也为链接元素增加了几个比较有用的属性，包括为 a 与 area 元素增加了 media 属性，规定目标 URL 是为什么类型的媒介或设备进行优化的，需要在 href 属性存在时使用。也为 area 元素增加了 hreflang 属性，以保持与 a 和 link 元素的一致。为 link 元素增加了 sizes 属性，在和 icon 元素结合使用时，该属性指定关联图标(icon 元素)的大小。另外，也为 base 元素增加了 target 属性，主要目的是保持与 a 元

素的一致性。其它为一些特定元素增加的常用属性如下所示：

- 为 ol 元素增加了 **reversed** 属性，指定列表倒序显示。
- 为 menu 元素增加了两个新的属性，一个是 **label** 属性为菜单定义一个可见的标注，另一个 **type** 属性让菜单可以以上下文菜单、工具条与列表菜单三种形式出现。
- 为 style 元素增加了 **scoped** 属性，用来规定样式的作用范围，可以只对页面上某个树起作用。
- 为 script 元素增加 **async** 属性，可以让脚本异步执行。
- 为 iframe 元素增加三个属性 **sandbox**、**seamless** 和 **srcdoc**，用来提高页面安全性，防止不信任的 web 页面执行某些操作。

还有，在 HTML5 中，新增了一些“全局属性”，即可以对任何元素都使用的属性，几种常用的全局属性如下所示：

- **contentEditable**：是否允许用户编辑元素中的内容。contentEditable 有两个值（true 和 false）。
- **designMode**：用来指定整个页面是否可编辑，有两个值(on 和 off)。该属性只可以通过 JavaScript 来修改值。如果 design 设置为 on 的话，所以允许设置 contentEditable 的元素都可以编辑。
- **hidden**：在 HTML5 中所有的元素都可以使用 hidden 属性，表示元素的不可见状态，有两个值(true 和 false)。
- **spellcheck**：针对 input 元素（text,textarea）用户输入的内容进行拼写和语法检查，有两个值(true 和 false)，必须先声明属性值是 true 或者 false。

在 HTML5 中还废除了一些 HTML4 中用的元素属性，而是采用其它属性或 CSS 来替代。就像在 HTML4 中好多元素的属性是设置页面显示效果的，而我们都是通过 CSS 来统一实现，所以在 HTML5 将不再被使用这样的属性。如下表 1-8 所示：

表 1-8 在 HTML 中被废除的属性

在 HTML 4 中使用的属性	使用该属性的元素	在 HTML 5 中的替代方案
rev	link、a	rel
charset	link、a	在被链接的资源的中使用 HTTP Content-type 头元素
shape、coords	a	使用 area 元素代替 a 元素
longdesc	img、iframe	使用 a 元素链接到校长描述
target	link	多余属性，被省略
nohref	area	多余属性，被省略
profile	head	多余属性，被省略
version	html	多余属性，被省略
name	img	id
scheme	meta	只为某个表单域使用 scheme
archive、chlassid、codebase、codetype、declare、standby	object	使用 data 与 type 属性类调用插件。需要使用这些属性来设置参数时，使用 param 属性
valuetype、type	param	使用 name 与 value 属性，不声明之的 MIME 类型
axis、abbr	td、th	使用以明确简洁的文字开头、后跟详述文字的形式。可以对更详细内容使用 title 属性，来使单元格的内容变得简短

scope	td	在被链接的资源的中使用 HTTP Content-type 头元素
align	caption、input、legend、div、h1、h2、h3、h4、h5、h6、p	使用 CSS 样式表替代
alink、link、text、vlink、background、bgcolor	body	使用 CSS 样式表替代
align、bgcolor、border、cellpadding、cellspacing、frame、rules、width	table	使用 CSS 样式表替代
align、char、charoff、height、nowrap、valign	tbody、thead、tfoot	使用 CSS 样式表替代
align、bgcolor、char、charoff、height、nowrap、valign、width	td、th	使用 CSS 样式表替代
align、bgcolor、char、charoff、valign	tr	使用 CSS 样式表替代
align、char、charoff、valign、width	col、colgroup	使用 CSS 样式表替代
align、border、hspace、vspace	object	使用 CSS 样式表替代
clear	br	使用 CSS 样式表替代
compace、type	ol、ul、li	使用 CSS 样式表替代
compace	dl	使用 CSS 样式表替代
compace	menu	使用 CSS 样式表替代
width	pre	使用 CSS 样式表替代
align、hspace、vspace	img	使用 CSS 样式表替代
align、noshade、size、width	hr	使用 CSS 样式表替代
align、frameborder、scrolling、marginheight、marginwidth	iframe	使用 CSS 样式表替代
autosubmit	menu	

## 5 HTML5 的高级功能介绍

作为一名 Web 开发者能掌握前面介绍过的 HTML5 技术，就几乎可以满足我们的需求，当然，HTML5 的内容远不指这些，如果你从事一些 HTML5 游戏开发，或基于移动设备的 APP 应用开发，你还需要了解更多 HTML5 的内容。像绘制图形、本地存储、离线应用程序、通信 API、处理线程、获取地理位置信息等。HTML5 的这些功能的介绍如下：

### 1. 绘制图形

如果人见过基于 HTML5 的游戏，多数都是通过 HTML5 新增元素 canvas 元素实现的，这个元素有一套编程接口(Canvas API)，专门用来绘制图形，可以实现非常漂亮的图形与图像，创建出更多丰富多彩、赏心悦目的下一代 Web 页面。在 canvas 元素里进行绘画，并



不是指拿鼠标来作画，`canvas` 元素只是一个无色透明区域，是 JavaScript 语言的画布，需要利用 JavaScript 编写脚本进行绘画。

## 2. 本地存储与离线应用

在 HTML5 中除了 `canvas` 元素之外，另一个新增的非常重要的功能是在客户端本地保存数据。在 HTML5 中与本地存储相关的有两个重要内容，分别是 `Web Storage` 与本地数据库。其中 `Web Storage` 存储机制是对 HTML4 中 `cookies` 存储机制的一个改善。本地数据库是 HTML5 新增的功能，可能使用它在客户端本地建立一个数据库，实现原本必须保存在服务器端数据库中的内容，可以直接保存在客户端本地了，这就大大减轻了服务器端的负担，提高了访问数据库的速度。现在 Web 应用程序已经变得越来越复杂，如果没有和 Internet 建立连接，就不能使用这个 Web 应用程序，在 HTML5 中新增加的 API，利用本地缓存机制很好地解决了这个问题，让 Web 应用程序在离线状态时也可能正常工作。就是把所有构成 Web 应用程序的资源文件，像 HTML 文件、CSS 文件和 JavaScript 脚本等放在本地缓存中，当服务器没有和 Internet 建立连接时，也可以利用本地缓存中的资源文件来正常运行 Web 应用程序。

## 3. 通信 API

在 HTML5 中提供了网页文档之间互相接收与发送信息的功能，使用这个功能，只要获取到网页所在窗口对象的实例，不仅同源的 Web 网页之间可以互相通信，甚至可以实现跨域通信。HTML5 新增两个和通信相关的功能，跨文档消息传输功能与使用 `Web Sockets` API 来通过 `socket` 端口传递数据的功能。使用跨文档消息传输功能，可以在不同网页文档、不同端口、不同域之间进行消息的传递。使用 `Web Sockets` API 可以让客户端与服务器通过 `Socket` 端口来传递数据，这样做就可以实现数据推送技术，服务器不再是被动地等待客户端发出的请求，只要客户端与服务器建立了一次连接之后，服务器端就可以在需要的时候，主动地将数据推送到客户端，直到客户端显示关闭这个连接。

## 4. 处理线程

在使用 HTML4 与 JavaScript 编写的 Web 程序中，因为所有的处理都是在单线程内执行的，如果花费时间比较长的话，程序界面会处于长时间没有响应的状态，而当时间长到一定程度，浏览器还会跳出一个脚本运行时间过长的提示框，用户就不得不中断正在执行的处理。为了解决这个问题，HTML5 新增了一个 `Web Workers` API，用户可以很容易地创建在后台运行的线程（在 HTML5 中被称为 `worker`），创建一个不会影响前台处理的后台程序，并且在这个后台线程中还能创建多个子线程，你可以将耗时较长的处理交给后台线程去运行，对用户在前台页面中执行的操作就完全没有影响了。

## 5. 获取地理位置信息

HTML5 中提供了地理位置信息的 API，通过浏览器来获取用户当前位置。如果浏览器支持，且设备具有定位功能，就能够直接使用这组 API 来获取当前位置信息。在获取地理位置信息前，首先浏览器都会向用户询问是否愿意共享其位置信息，待用户同意后才能使用。基于此特性可以开发基于位置的服务应用。



## 6 CSS3 概述

CSS3 对于我们 Web 开发者来说不只是新奇的技术,更重要的是这些全新概念的 Web 应用给我们带来更多无限的可能性,也极大地提高了我们的开发效率。我们将不必再依赖图片或者 JavaScript 去完成圆角、多背景、用户自定义字体、3D 动画、渐变、盒阴影、文字阴影、透明度等提高 Web 设计质量的特色应用。

### 6.1 CSS3 在选择器上面的支持

CSS3 在选择器上的丰富支持让我们可以灵活的控制样式,通过选择器的使用,你不再需要在编辑样式时使用多余的以及没有任何语义的 class 属性,而可以直接将样式与元素绑定起来,利用属性选择器我们可以根据属性值的开头或结尾很容易选择某个元素,利用兄弟选择器可以选择同级兄弟结点或紧邻下一个结点的元素,利用伪类选择器可以选择某一类元素,从而节省在网站或 Web 应用程序完成之后又要修改样式所需花费的大量时间。

### 6.2 CSS3 在样式上的支持

只要提起 CSS3 的特性是什么? 回答最多的就是“圆角”,不错,圆角这个功能可以让前端布局省去很多时间和精力去切图拼凑一个圆角。CSS3 还支持阴影(盒阴影和文本阴影),以及渐变,之前你可能是通过 IE 的滤镜看到过,其实 CSS3 实现起来更加方便。还可以自定义字体,传统的方式是使用特殊文字的图片实现,而通过 CSS3 用@font-face 就可以了。CSS3 对于连续文本换行也新增了一个属性 word-wrap,你可以设置其为 normal(不换行)或 break-word(换行),这解决了连续英文字符出现页面错位的问题,也不需要后端去截取这个连续字符。使用 CSS3 你还可以给边框加背景,这在 iphone 上也有应用的例子。CSS3 在背景上支持背景调整大小,如当你用一张大图要做一个小的元素背景时可以通过 background-size 来调整背景图的大小来适应这个元素。CSS3 支持 opacity, rgba 和 hsl/a, opacity 和 rgba 的区别是 opacity 设置的透明对其内容也会产生影响,而 rgba 只对你应用的元素产生影响。CSS3 在布局上对于盒模型提供了支持,可以设置 box-sizing 为 content-box 或 border-box,应用为 content-box 就是正常的模式,而应用为 border-box 和 IE5.5 的盒模型很相似,即元素的宽度包括 border 和 padding,这个在布局上可能会比较方便,不用去管到底这个元素会占用多大的宽度,而用 content-box 还需要手动计算一下这个元素实际占用的宽度。

### 6.3 CSS3 对于动画的支持

CSS3 支持的动画类型有: transform(变换)、transition(过渡)和 animation(动画)。你可以对特定的属性设置 transition, transition 和

animation 的区别不大，animation 的动画是自己定义的，面向的更多的是脚本开发者，往往更加复杂。为了使用大部分 CSS3 特性，我们不得不与原来的属性一起使用生产商专有扩展。原因是直到现在，大部分浏览器只支持部分 CSS3 属性。最常见的私有属性是用于 Webkit 核心浏览器的(比如, Safari)， 它们以 -webkit- 开始，以及 Gecko 核心的浏览器(比如, Firefox)，以 -moz- 开始，还有 Konqueror (-khtml-)、Opera (-o-) 以及 Internet Explorer (-ms-) 都有它们自己的属性扩展（目前只有 IE8 支持 -ms- 前缀）

## 6.4 在开发中该如何去用 CSS3

首先是遵循一个优雅降级的原则，比如前面谈到的圆角，我们可以针对 Firefox 和 Safari 等这些支持圆角的浏览器中应用 CSS 圆角，而那些不支持 CSS 圆角的浏览器则显示为直角。其次就是对于不支持 CSS3 的浏览器可以使用 Javascript 来实现。在向用户或企业推广新技术的同时也要关注他们的目标与可行性，不能为了技术而技术。

## 7 CSS3 中的样式选择器

要使用 CSS 对 HTML 页面中的元素实现一对一，一对多或者多对一的控制，这就需要用到 CSS 选择器。选择器是 CSS3 中一个重要的内容，使用它可以大幅度提高开发人员书写或修改样式表的工作效率。在大型网站中，样式表中的代码可能会达到几千行，而当我们需要对样式进行修改时，在大篇 CSS 代码中，并没有说明什么样式服务于什么元素，只是在元素中使用了 class 属性。而 class 属性本身没有语义，它纯粹是用来为 CSS 服务的，属于多余属性，并没把样式和元素绑定起来，一个 CSS 类选择器，文本框可以使用，下拉框也可以使用，甚至按钮也可以使用，这样其实非常混乱，修改时就不方便，所以 CSS3 中提倡使用选择器来将样式与元素直接绑定起来，这样的话，在样式中什么样式与什么元素相匹配变得一目了然，修改起来也很方便。不仅如此，通过选择器我们还可以实现各种复杂的指定，同时也能大量减少样式表的代码书写量，最终书写出来的样式表也会变得简洁明了。那么从现在开始我们先丢开他们版本的区别，从头一起来看看 CSS 选择器的运用。

### 7.1 基本选择器

CSS 是一种用于屏幕上渲染 HTML 一种语言，CSS 主要是在相应的元素中应用样式，来渲染相对应用的元素，那么这样我们选择相应的元素就很重要了，如何选择对应的元素，此时就需要我们所说的选择器。选择器主要是用来确定 HTML 的树形结构中的 DOM 元素节点。可以把 CSS 选择器分开成几个部分，有常用的部分，通常叫做基本选择器；属性选择器；伪类选择器。基本选择器如表 1-9 所示：

表 1-9 CSS3 基本选择器及实例

选择器	说明
-----	----

*	通用元素选择器，匹配任何元素
E	标签选择器，匹配所有使用 E 标签(所有 HTML 元素)的元素
.info	class 选择器，匹配所有 class 属性中包含 info 的元素
#footer	id 选择器，匹配所有 id 属性等于 footer 的元素
<pre> *      { margin:0; padding:0; }      /* 通用元素选择器，匹配任何元素      */ .info *{ border:1px solid blue; }    /* 选择某个元素下的所有元素      */ p      { font-size:2em; }           /* 标签选择器，匹配所有使用p标签的元素 */ .info { background:#ff0; }          /* class选择器，匹配所有class属性中包含info的元素 */ p.info { background:#ff0; }          /* class选择器，特定元素下的类属性      */ #info  { background:#ff0; }          /* id选择器，匹配所有id属性等于info的元素 */ </pre>	

## 7.2 多元素的组合选择器

多个选择器可以组合在一起使用，有多种形式，包括多元素选择器、后代元素选择器、子元素选择器，及毗邻选择器四种类型。如表 1-10 所示：

表 1-10 CSS3 多元素的组合选择器及实例

选择器	说明
E,F	多元素选择器，同时匹配所有 E 元素或 F 元素，E 和 F 之间用逗号分隔
E F	后代元素选择器，匹配所有属于 E 元素后代的 F 元素，E 和 F 之间用空格分隔
E > F	子元素选择器，匹配所有 E 元素的子元素 F（第一层后代）
E + F	毗邻元素选择器，匹配所有紧随 E 元素之后的同级元素 F
E ~ F	同级元素通用选择器，匹配任何在 E 元素之后的同级 F 元素
<pre> #nav,li { display:inline; }      /* 多元素选择器,同时匹配所有id为nav元素或li元素      */ div p   { color:#f00; }          /* 后代元素选择器，匹配所有属于div元素后代的p元素      */ div &gt; p { color:#f00; }          /* 直接子选择器 应用与div元素的直接子元素p上          */ h1 + p { color:#f00; }          /* 毗邻元素选择器，匹配所有紧随h1元素之后的同级元素p */ p ~ ul { background:#ff0; }      /* 匹配任何在p元素之后的同级ul元素                      */ </pre>	

## 7.3 属性选择器

属性选择器可以根据元素的属性及属性值来选择元素。如果希望选择有某个属性的元素，而不论属性值是什么，可以使用简单属性选择器。CSS2.1 就引入了一些属性选择器，如表 1-11 所示：

表 1-11 CSS2.1 属性选择器及实例

选择器	说明
E[att]	匹配所有具有 att 属性的 E 元素，不考虑它的值。（注意：E 在此处可以省略，比如“[checked]”，以下同。）
E[att=val]	匹配所有 att 属性等于“val”的 E 元素
E[att~=val]	匹配所有 att 属性具有多个空格分隔的值、其中一个值等于“val”的 E 元素
E[att =val]	匹配所有 att 属性具有多个连字号分隔（hyphen-separated）的值、其中一个值以“val”开头的 E 元素，主要用于 lang 属性，比如“en”、“en-us”、“en-gb”等等

```
p[title] { color:#f00; } /* 匹配所有具有title属性的p元素，不考虑title的值 */
div[class=error] { color:#f00; } /* 匹配所有class属性等于"error"的div元素 */
td[headers~=coll] { color:#f00; } /* 匹配所有headers属性具有多个空格分隔的值、其中一个值等于"coll"的td元素 */
p[lang=en] { color:#f00; } /* 匹配所有lang属性具有多个连字号分隔的值、其中一个值以"en"开头的元素 */
blockquote[class=quote][cite] { color:#f00; } /* 可以多个属性组给使用 */
```

在 CSS3 中又添加了一些属性选择器，使用 CSS3 属性选择器，你可以只指定元素的某个属性，或者你还可以同时指定元素的某个属性和其对应的属性值。CSS3 的属性选择器，如表 1-12 所示：

表 1-12 CSS3 属性选择器及实例

选择器	说明
E[att^="val"]	属性 att 的值以"val"开头的元素
E[att\$="val"]	属性 att 的值以"val"结尾的元素
E[att*="val"]	属性 att 的值包含"val"字符串的元素
<pre>.demo a[href^="http://"]{ background:orange;color:green;} /* 属性href的值以"http://"开头的元素 */ .demo a[href\$=".png"] { background:orange;color:green;} /* 属性href的值以".png"结尾的元素 */ .demo a[title*="site"] { background:black;color:white;} /* 属性title的值包含"site"字符串的元素 */</pre>	

7.4 结构性伪类选择器

在学习结构性伪类选择器之前，先了解两个概念(CSS 中的伪类选择器和伪元素)。伪类选择器是 CSS 中已经定义好的选择器，不能随便取名 常用的伪类选择器是使用在 a 元素上的几种，如 a:link|a:visited|a:hover|a:active。而伪元素选择器并不是针对真正的元素使用的选择器，而是针对 CSS 中已经定义好的伪元素使用的选择器，CSS 中有如下四种伪元素选择器(first-line、first-letter、before、after)，例如 p:first-line{ color:#ff0000;}， li:after{ content:url(dot.png)}。而在 CSS3 中引入的结构性伪类选择器的共同特征是允许开发者根据文档树中的结构来指定元素的样式，以下进行分类介绍，如表 1-13 到 3-20 所示：

表 1-13 四个最基本的结构性伪类选择器 root、not、empty 和 target

选择器	说明
E:root	将样式绑定到页面的根元素中。所谓根元素，是指位于文档树中最顶层结构的元素，在 HTML 页面中就是指包含着整个页面的<html>部分
E:not	想对某个结构元素使用样式，但想排除这个结构元素下的子结构元素，就是用 not 样式
E:empty	指定当元素内容为空白时使用的样式
E:target	对页面中某个 target 元素指定样式，该样式只在用户点击了页面中的链接，并且跳转到 target 元素后生效
<pre>:root { background-color:yellow } /* 指定整个网页背景色为黄色 */ body *:not(h1) { background-color:yellow } /* 指定body元素背景为黄色，但排除h1 */ :empty { background-color:yellow } /* 指定某个元素内容为空白时，该元素背景为黄色（例如表格） */ target { background-color:yellow } /* 链接的目标背景为黄色 */ /* &lt;a href="#text3"&gt;示例&lt;/a&gt; &lt;div id="text3"&gt;内容 &lt;/div&gt; 点击时目标div背景为黄色 */</pre>	

表 1-14 选择器 first-child、last-child、nth-child、nth-last-child

选择器	说明
-----	----

<b>E:first-child</b>	对一个父元素中的第一个子元素指定样式
<b>E:last-child</b>	对一个父元素中的最后一个子元素指定样式
<b>E:nth-child</b>	对指定序号的子元素设置样式(正数) :nth-child(odd): 所有正数下来第偶数个子元素 :nth-child(even):所有正数下来第奇数个子元素
<b>E:nth-last-child</b>	对指定序号的子元素设置样式(倒数) :nth-last-child(odd): 所有倒数上去第偶数个子元素 :nth-last-child(even):所有倒数上去第奇数个子元素
<pre> p:first-child      { color:yellow }      /* 第一个p元素的样式 */ p:last-child       { color:yellow }      /* 倒数第一个p元素的样式 */ p:nth-child(2)     { color:yellow }      /* 第2个p元素的样式 */ p:nth-last-child(2){ color:yellow }      /* 倒数第2个p元素的样式 */ </pre>	

表 1-15 选择器 nth-of-type、nth-last-of-type

选择器	说明
<b>E:nth-of-type(n)</b>	与:nth-child()作用类似，但是仅匹配使用同种标签的元素
<b>E:nth-last-of-type(n)</b>	与:nth-last-child() 作用类似，但是仅匹配使用同种标签的元素
<pre> h2:nth-of-type(even) { color:yellow }      /* h2元素的偶数行为黄色 */ h1:nth-of-type(odd)  { color:green }       /* h1元素的奇数行为绿色 */ </pre>	

表 1-16 循环使用样式

选择器	说明
<b>E:nth-child(an+b)</b>	a 表示每次循环中共包括几张样式； b 表示指定的样式在循环中所在的位置；
<pre> li:nth-child(4n+1){background-color:yellow;} /*第一个li背景色为 黄色，这样依次循环下去 */ li:nth-child(4n+2){background-color:bule; } /*第二个li背景色为 蓝色..... */ li:nth-child(4n+3){background-color:red; } /*第三个li背景色为 红色..... */ li:nth-child(4n+4){background-color:green; } /*第四个li背景色为 绿色..... (4n+4可缩写为4n) */  /* 前面所讲的nth-child(odd) &amp;nth-child(even) 可以用如下代码替代: */ li:nth-child(2n+1){} /* 所有正数下来的第奇数个子元素 */ li:nth-child(2n+2){} /* .....第偶数..... */ li:nth-last-child(2n+1){} /* 所有倒数上去的第奇数个子元素 */ li:nth-last-child(2n+2){} /* .....第偶数..... */ </pre>	

表 1-17 选择器 only-child、only-of-type

选择器	说明
<b>E:only-child</b>	匹配父元素下仅有的一个子元素 等同于: first-child:last-child 或 :nth-child(1):nth-last-child(1)
<b>E:only-of-type</b>	匹配父元素下使用同种标签的唯一一个子元素 等同于: first-of-type:last-of-type 或 :nth-of-type(1):nth-last-of-type(1)
<pre> li :nth-child(1):nth-last-child(1){ background-color:yellow} /* 可用only-child代替 */ li :only-child { background-color:yellow} /* ..... 同上..... */ li :only-of-type{ background-color:yellow} /* 也可用only-of-type代替 */ </pre>	

表 1-18 CSS 3 中与用户界面有关的伪类

选择器	说明
<b>E:enabled</b>	匹配表单中激活的元素

<b>E:disabled</b>	匹配表单中禁用的元素
<b>E:checked</b>	匹配表单中被选中的 radio（单选框）或 checkbox（复选框）元素
<b>E::selection</b>	匹配用户当前选中的元素
<pre>input[type="text"]:disabled { background:#ddd; } /* 将表单中禁用的文本text元素背景设置成灰色 */</pre>	

## 8 CSS3 的样式属性

通过 CSS 选择器找到元素，就要使用 CSS 属性给找到元素设置样式。尽管现代浏览器已经支持了众多的 CSS3 属性，但作为初学者最应该关注的就是一些“主流”的属性，如 border-radius、box-shadow 或者 transform 等。它们有良好的文档、很好的测试并且最常用到，所以如果你在设计网站，你很难能脱离它们。学习 CSS3 的属性就要先从主流的属性开始，其它的属性则可通过 CSS3 手册获得帮助。

### 8.1 使用 CSS3 属性前的准备

虽然大多数的 CSS3 属性在很多流行的浏览器中不支持，但是还是鼓励在前端开发中要学会并且运行这些 CSS3 属性，因为这是未来的趋势。关键是首先确定你是否对各个浏览器之间的细微的差别有所了解，你能肯定的说 IE 显示的 90 度的角就不圆滑嘛？这取决于你的决定。但是要永远记住，网站设计不必看到所有浏览器的不同。在使用 CSS3 的一些高级特性时，需指定所有浏览器的前缀，CSS3 的前缀是一个浏览器生产商经常使用的一种方式，它暗示该 CSS 属性或规则尚未成为 W3C 标准的一部分。CSS3 前缀的浏览器属性规则如表 3-10 所示：

表 1-19 CSS 3 前缀的浏览器规则

CSS3 前缀	代表浏览器
<b>-webkit-</b>	代表 Chrome、Safari
<b>-moz-</b>	代表 Firefox
<b>-ms-</b>	代表 IE
<b>-o-</b>	代表 Opera
<pre>-webkit-transform: rotate(-3deg); /* 为Chrome Safari */ -moz-transform: rotate(-3deg); /* 为Firefox */ -ms-transform: rotate(-3deg); /* 为IE */ -o-transform: rotate(-3deg); /* 为Opera */ transform: rotate(-3deg); /* 为nothing */</pre>	

既然 CSS3 代码中（暂时）需要写上这么多前缀，就需要注意顺序问题，要先写私有的 CSS3 属性，再写标准的 CSS3 属性。如果以后当一个属性成为标准，并且被 Firefox、Chrome 等浏览器的最新版普遍兼容的时候，就可以去除这些前缀了。

## 8.2 边框属性

通过 CSS3，您能够创建圆角边框，向矩形添加阴影，使用图片来绘制边框。圆角实现是一大堆 CSS3 属性中最受欢迎的一种，通过 `border-radius` 属性来实现，是 CSS3 中级别最高的一个属性。`radius` 就是半径的意思。用这个属性可以很容易做出圆角效果，当然，也可以做出圆形效果。原理很简单，“正方形的内切圆的半径等于正方形边长的一半”。使用格式如下：

```
-webkit-border-radius: 4px;      /* 适用基于webkit的浏览器 */
-moz-border-radius: 4px;        /* 适用于firefox浏览器 */
border-radius: 4px;             /* 请在Safari 5和IE9浏览器中执行这个语法 */

border-radius: 2em;              /* 四角都是圆角 */
/*等价于: */
border-radius: 2em 2em 2em 2em; /* 四角都是圆角,左上角顺时针 */
/*等价于: */
border-top-left-radius: 2em;     /* 左上角 */
border-top-right-radius: 2em;    /* 右上角 */
border-bottom-right-radius: 2em; /* 右下角 */
border-bottom-left-radius: 2em;  /* 左下角 */

/* 根据需要可设置每个角的半径例如设置左上角的边框半径的css代码如下: */
-webkit-border-top-left-radius: 5px;
-moz-border-radius-topleft: 5px;
```

可以通过 `box-shadow` 属性向边框添加一个或多个阴影，也是开发时常用的效果。使用示例如下所示：

```
<!DOCTYPE html>
<meta charset="utf-8" />
<style>
div {
    width: 300px; height: 100px; border: 1px solid #2cb7fe; text-align: center;
    -webkit-box-shadow: 3px 3px 10px #9edeff;
    box-shadow: 3px 3px 10px #9edeff;
}
</style>

<div>box-shadow:3px 3px 10px #9edeff;</div>
```

`box-shadow` 属性是由逗号分隔的阴影列表，每个阴影由 2-4 个长度值、可选的颜色值以及可选的 `inset` 关键词来规定，省略长度的值是 0。该属性可用值如下所示：

- `none`：无阴影
- `<length>①`：第 1 个长度值用来设置对象的阴影水平偏移值。可以为负值
- `<length>②`：第 2 个长度值用来设置对象的阴影垂直偏移值。可以为负值
- `<length>③`：如果提供了第 3 个长度值则用来设置对象的阴影模糊值。不允许负值
- `<length>④`：如果提供了第 4 个长度值则用来设置对象的阴影外延值。可以为负值
- `<color>`：设置对象的阴影的颜色。
- `inset`：设置对象的阴影类型为内阴影。该值为空时，则对象的阴影类型为外阴影

此属性还可以设置多组值，中间用","隔开。另外，通过 CSS3 的 `border-image` 属



性，还可以使用图片来创建边框，详见手册。

## 8.3 背景属性

在 CSS3 中提供了多个背景属性，这里只介绍两个比较常用的属性，其它可以从手册中获取帮助。CSS3 的诞生为我们解决了这一问题，在 CSS3 里，通过 `background-image` 或者 `background` 可以为一个容器设置多张背景图像，也就是说可以把不同背景图像只放到一个块元素里。多个背景图片的 `url` 之间使用逗号隔开即可，如果有多个背景图片，而其他属性只有一个，那么所有背景图片都应用该属性值。代码如下所示：

```
.div1{
    background-image:url(images/1.jpg),url(images/2.jpg),url(images/3.jpg),url(images/4.jpg);
    background-repeat:no-repeat,no-repeat,no-repeat,no-repeat;
    background-position:top left,top right,bottom left,bottom right;
}

/*在上面的代码中有这一句 */
background-repeat:no-repeat,no-repeat,no-repeat,no-repeat;
/*可以简化为: */
background-repeat:no-repeat;

/*上面设置背景图片的各个属性时是分开写的，那么我们也可以把背景图片的各个属性写在一块 */

```

背景大小调整也是 CSS3 提供的一个新特性，它可以让你随心所欲的控制背景图的尺寸大小。我们知道在 CSS2 中，背景图的大小在样式中是不可控的，比如要想使得背景图填满某个区域，你要么重新做张大点的图，要么就只能让它以平铺的方式来填充。在 CSS3 中提供了 `background-size` 属性，让你既可以直接缩放背景图来填满这个区域，也可以让你先给背景图设置大小，然后以设置好的尺寸去平铺满这个区域。`background-size` 需要 1 个或 2 个值（1 个为必填，1 个为可选），这些值既可以是像素 `px`，也可以是百分比`%`或 `auto`，还可以是特定值 `cover`, `contain`。示例代码如下所示：

```
body {
    background: url(path/to/image.jpg) no-repeat;
    -moz-background-size: 100% 100%;
    -o-background-size: 100% 100%;
    -webkit-background-size: 100% 100%;
    background-size: 100% 100%;
}
```

其中 `background-size` 第 1 个值用于指定背景图的 `width`，第 2 个值用于指定背景图的 `height`，如果只给 `background-size` 设置 1 个值，则第 2 个值默认为 `auto`（`cover` 和 `contain` 特定值除外）。`background-size` 的特定值：

- `cover`: 保持图像本身的宽高比例，将图片缩放到正好完全覆盖定义背景的区域；
- `contain`: 保持图像本身的宽高比例，将图片缩放到宽度或高度正好适应定义背景的



## 8.4 文本属性

关于 css3 的文本新属性还是满多的，在这些属性中比较出彩的属性有两个：一个是可以为文字添加阴影的 `text-shadow`，另外一个则是可以强制对单词进行换行处理的 `word-wrap` 属性。`text-shadow` 属性要求的浏览器版本较高，对于 ie 来说，至少需要 ie10 以上版本的支持，至于 Firefox、Chrome、Safari 以及 Opera 等浏览器则完全支持这一新属性。大家或许常常会借鉴一些国外的主题，常常会发现有些标题会带有很好看的阴影，但在 ie 下无法正常的显示出来，这正是 `text-shadow` 属性的功劳。`word-wrap` 属性算是一个能够被广泛支持的新属性，几乎所有的主流浏览器都支持这一属性，即使是 ie 也不例外。

### 1. 给文字加上阴影的 `text-shadow` 属性：

属性 `text-shadow` 是我们可以省略前缀的几个之一，这个属性与边框的阴影属性（`box-shadow`）差不多，总共包含了四个参数：水平阴影、垂直阴影、模糊距离，以及阴影的颜色。与我们所讲的边框阴影属性同样类似的是，这些值除了最后的颜色之外，均可以用负值来表示。下面给个简单的代码示例：

```
h1 {  
    text-shadow: 0 1px 0 white;           /* 文本阴影样式 */  
    color: #292929;  
}
```

### 2. 强制单词换行的 `word-wrap` 属性：

当我们的段落中出现特别长的单词的话，因为没有强制换行，可能导致某个单词大量溢出或者直接自动换行而出现行尾留出很大的空白，这些都使得我们的文本便的很不整齐，`word-wrap` 这个属性也正是因此而生。请看下面的简单用法：

```
word-wrap: break-word;
```

## 8.5 用户界面属性

在 CSS3 中，新的用户界面特性包括重设元素尺寸、盒尺寸以及轮廓等。本节着重介绍一下 `resize` 属性，只有 Firefox 4 和 Safari 3 支持此属性，`resize` 是可用于重定义 `textarea` 的大小。可能的值：

- **none:** UserAgent 没提供尺寸调整机制，用户不能操纵机制调节元素的尺寸。
- **both:** UserAgent 提供双向尺寸调整机制，让用户可以调节元素的宽度和高度。
- **horizontal:** UserAgent 提供单向水平尺寸调整机制，让用户可以调节元素的宽度。
- **vertical:** UserAgent 提供单向垂直尺寸调整机制，让用户可以调节元素的高度。

```

<!-- 默认情况下webkit浏览器和Firefox 4支持水平和垂直方向上的重定义 -->
<style>
textarea {
    -moz-resize: vertical;
    -webkit-resize: vertical;
    resize: vertical;
}
</style>
<textarea name="elem" id="elem" rows="5" cols="50"></textarea>

```

## 8.6 动画属性

也许 CSS3 最令人兴奋的增补，就是在没有 JavaScript 的元素的情况下产生动画。CSS3 的动画有三个常用属性：transform、transition、animation。transform 属性虽然看起来可以实现动画的效果，但本质其实是静态的，其实就是一个图形的变形工具；而 transition 属性是一个简单的动画属性，操作起来非常的简单；Animations 属性是个名副其实的动画属性，是 transition 属性的扩展，但功能可谓十分的强大，Animations 可以定义多个关键帧以及定义每个关键帧中元素的属性值来实现复杂的动画效果。下面的示例使用 transition 属性模拟一个效果，当鼠标滑过右侧链接时，文本向右滑动。代码如下：

```

<style>
ul a {
    -webkit-transition: padding .4s;
    -moz-transition: padding .4s;
    -o-transition: padding .4s;
    transition: padding .4s;
}

a:hover { padding-left: 20px; }
</style>
<ul>
<li> <a href="#"> Hover Over Me </a> </li>
<li> <a href="#"> Hover Over Me </a> </li>
<li> <a href="#"> Hover Over Me </a> </li>
</ul>

```

## 8.7 多列布局属性

通过 CSS3，您能够创建多个列来对文本进行布局，就像报纸那样！在 CSS2 时代，对于多列布局的设计，大多采用浮动布局和绝对定位布局两种方式。浮动布局比较灵活，但是需要编写大量的附加样式代码，而且在网页缩放等操作下容易发生错位，影响网页整体效果。绝对定位方式要精确到标签的位置，但固定标签位置的方式无法满足标签的自适应能力，也影响文档流的联动。CSS3 新增 columns 属性，即多列自动布局功能，利用多列布局属性可以自动将内容按指定的列数排列（例：columns: 250px 3; 250px 表示的是列宽，3 表示的是多列的数目）。可以结合 column-gap 属性定义列之间的间距、使用 column-rule 属性可以定义每列之间边框的宽度、样式和颜色。应用示例代码如下所示：

```

body {
    -webkit-column-count: 3;          /* 定义列数为3列 */
    -webkit-column-gap: 3em;          /* 定义列之间的间距为3em */
    columns: 250px 3;                 /* 定义每列宽度为250px, 共定义3列 */
    column-gap: 3em;                  /* 定义列之间的间距为3em */
    -webkit-column-rule:dashed 2px gray; /* 定义列的宽度2px、样式dashed和颜色gray */
    column-rule:dashed 2px gray;      /* 定义列的宽度2px、样式dashed和颜色gray */
}

```

CSS3 多列布局还有很多新属性和特性，如栏目高度 `column-fill` 属性以及分列打印等，本节只做简单介绍，全做抛砖引玉。

## 8.8 渐变属性

对于正常的渐变背景，一般都是切丝然后平铺，至于横向平铺还是纵向平铺还是得看实际情况而定，复杂点的背景就只能切块图，然而 CSS3 可以让 Firefox 和 Safari/Chrome 实现颜色渐变，IE 可以用滤镜，这也是一种值得考虑的很酷的方法。先看语法：

```

.box_gradient {
    background-image: -moz-linear-gradient(top, #444444, #999999);
    background-image: -webkit-gradient(linear, left top, left bottom, color-stop(0, #444444), color-stop(1, #999999));
    /* IE6, IE7 */
    filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#444444', endColorstr='#999999', GradientType='0');
    /* IE8 */
    -ms-filter: "progid:DXImageTransform.Microsoft.gradient(startColorstr='#444444', endColorstr='#999999', GradientType='0')";
}

```

先看第一行 Firefox，属性“-moz-linear-gradient”有三个参数。第一个参数表示线性渐变的方向，top 是从上到下、left 是从左到右，如果定义成 left top，那就是从左上角到右下角。第二个和第三个参数分别是起点颜色和终点颜色。你还可以在它们之间插入更多的参数，表示多种颜色的渐变。

再看第二行属性“-webkit-gradient”是 webkit 引擎对渐变的实现，一共有五个参数。第一个参数表示渐变类型（type），可以是 linear（线性渐变）或者 radial（辐射渐变）。第二个参数和第三个参数，都是一对值，分别表示渐变起点和终点。这对值可以用坐标形式表示，也可以用关键值表示，比如 left top（左上角）和 left bottom（左下角）。第四个和第五个参数，分别是两个 color-stop 函数。color-stop 函数接受两个参数，第一个表示渐变的位置，0 为起点，0.5 为中点，1 为结束点；第二个表示该点的颜色。

IE 依靠滤镜实现渐变。startColorstr 表示起点的颜色，endColorstr 表示终点颜色。GradientType 表示渐变类型，0 为缺省值，表示垂直渐变，1 表示水平渐变。

## 8.9 透明属性

元素透明也是我们常用的样式，在 CSS2 中使用滤镜属性 opacity 实现透明效果。现在有了 CSS3 的 rgba 属性，就不用这么麻烦了，当然这也得要浏览器支持才行。通常我们定义颜色都是用十六进制表示。如：background:#000000，表示背景色为黑色。当然也可以用 RGB 三原色值表示，如：background:rgb(0,0,0)，也表示背景色为黑色。rgba 只是在 rgb 的基础上增加了一个 a，这个 a 表示透明度。而且这个属性不会被子元素继承下去，可以在定义颜色的属性使用，应用代码如下：

```

.test{ background:rgba(0,0,0,0.5) } /* 表示背景色黑色且半透明 */
.test{ color:rgba(0,0,0,0.5) } /* 表示字体颜色为黑色且半透明 */
.test{ background:#000000; filter:alpha(opacity=40) } /* IE 滤镜 */

```

除了 IE，其他浏览器几乎都支持 `rgba` 函数。它有四个参数，前三个为一种颜色的 RGB 值，第四个为透明度。

## 8.10 旋转属性

在 CSS3 中，可以使用 `transform` 允许我们对元素进行旋转，放缩，倾斜，平移的功能，以旋转为例，代码如下所示：

```
.box_rotate {
    transform: rotate(7.5deg);
    -ms-transform: rotate(7.5deg);          /* IE 9 */
    -moz-transform: rotate(7.5deg);         /* Firefox */
    -webkit-transform: rotate(7.5deg);      /* Safari 和 Chrome */
    -o-transform: rotate(7.5deg);          /* Opera */
}
```

除了早版本的 IE 以外，其他浏览器都是用 `rotate` 函数，实现某个对象的旋转。比如 `rotate(7.5deg)` 就表示顺时针旋转 7.5 度 (degree)。利用 CSS3，还可以完成 `skew` (扭曲) 和 `scale` (缩放)，以及 `css transitions` (动态变换) 等。

## 8.11 服务器端字体属性

设计网页的时候，可能会用到某种特殊的字体。如果用户的机器中没有安装，文字只能以普通字体显示。CSS 可以让用户的浏览器自行下载服务器端字体，然后就能呈现出设计者想要的效果。应用示例代码如下所示：

```
@font-face {
    /* 表示为这种字体起一个名称，可以随意设置，我这里用的是MyFont */
    font-family: 'MyFont';
    /* IE6+ 这一行表示字体位置，由于ie只支持服务器端的eot字体，所以这一行是ie专用的 */
    src: url('myfont.eot');
    /* local()表示在本机(客户端)查找该字体，如果本机已经安装了，就不用下载了。
    url()表示字体在服务器上的位置，format()用来说明字体格式。
    Firefox 3.5支持TrueType和OpenType字体，Firefox 3.6又增加了WOFF字体。
    其他基于Webkit引擎的浏览器(sarif, opera, chrome)，目前好像只支持truetype。 */
    src: local('myfont.ttf'),
        url('myfont.woff') format('woff'), /* FF3.6 */
        url('myfont.ttf') format('truetype'); /* FF3.5+, Saf3+,Chrome,Opera10+ */
}
/* 使用的时候这样写就可以了 */
h2{ font-family: "MyFont"; }
```

需要注意的是，字体文件必须与网页文件来自同一个域名，符合浏览器的“同源政策”。另外，由于中文字体文件太大，服务器端字体显然只适用于英文字体。

## 9 使用 HTML5 中新增元素来构建网页

HTML5 中新增了几个结构元素：`section` 元素、`article` 元素、`nav` 元素、`aside` 元素、`header` 元素和 `footer` 元素，通过运用这些结构元素，可以让网页的整体结构更加直观和明确、更加

富有语义化和更具有现代风格。本例以一个博客网站为例介绍如何综合运用 HTML5 中的这些结构元素,并与 CSS3 样式结合,让你了解在 HTML5 实现的网页中应该如何使用 CSS3 样式来对页面中的元素进行页面布局及视觉美化。在一个用 HTML5 实现的网页中,每个网页都将由一些主体结构元素构成,在大型的网站中,一个网页通常由以下几个结构元素组成。

- **header 结构元素:** 通常用来展示网站的标题、企业或公司的 LOGO 图片、BANNER、网站导航条,等等。
- **section 结构元素:** 一个网页中要显示的主体内容通常被放置在 section 结构元素中,每个 section 结构元素都应该有一个标题来显示当前要展示的主要内容的标题信息。每个 section 结构元素中通常还应该包括一个或多个 section 元素或 article 元素,用来显示网页主体内容中每一个相对独立的部分。
- **aside 结构元素:** 通常用来展示与当前网页或整个网页相关的一些辅助信息。例如,在博客网站中,可以用来显示博主的文章列表和浏览者的评论信息等。在购物网站中,可以用来显示商品清单、用户信息、用户购买历史等。在企业网站中,可以用来显示产品信息、企业联系方式、友情链接等。aside 结构元素可以有多种形式,其中最常见的形式是侧边栏。
- **footer 结构元素:** 通常,每个网页中都具有 footer 结构元素,用来放置网站的版权声明和备案信息等与法律相关的信息,也可以放置企业的联系信息。

在本例中,都是使用 HTML5 的标记方法,也通过 html5.js 文件的导入,实现让早期 IE 版本可以兼容 HTML5,一个简单的博客系统首页面的 HTML5 布局代码如下所示:

```
1 <!DOCTYPE html>                                <!-- 文档HTML5格式的定义 -->
2 <html>
3   <head>
4     <meta charset="utf-8">                        <!-- HTML5的页面字符设置 -->
5     <title>HTML5+CSS3布局示例(博客首页)</title>
6     <!--[if IE]> <script src="html5.js"></script> <![endif]--> <!-- 让HTML5兼容IE早期版本 -->
7     <link rel="stylesheet" href="layout.css" type="text/css">
8   </head>
9
10  <body>
11    <header id="page_header">                     <!-- 使用header定义页面头部 -->
12      <h1>Header</h1>
13      <nav>                                         <!-- 使用nav定义页面导航条 -->
14        <ul>
15          <li><a href="#">首页</a></li>
16          <li><a href="#">日志</a></li>
17          <li><a href="#">相册</a></li>
18          <li><a href="#">收藏</a></li>
19        </ul>
20      </nav>
21    </header>
22
23    <section id="posts">                           <!-- 使用section定义页面主体区 -->
24      <h2>Section</h2>
25      <article class="post">                         <!-- 使用article定义文章区 -->
26        <h2>文章标题</h2>
27
28        <header>                                    <!-- 使用header定义文章头 -->
29          <h3>文章 Header</h3>
30        </header>
```

```

31
32 |
33 |         <aside>                                <!-- 使用aside定义文章侧边栏 -->
34 |             <h3>文章 Aside</h3>
35 |         </aside>
36
37 |         <p>这里是文章的内容...</p>
38 |         <p>这里是文章的内容...</p>
39 |
40 |         <footer>                                <!-- 使用footer定义文章底部信息 -->
41 |             <h3>文章 Footer</h3>
42 |         </footer>
43 |     </article>
44
45 |     <article class="post">                        <!-- 使用article定义文章区 -->
46 |         <h2>文章标题</h2>
47 |
48 |         <header>                                <!-- 使用header定义文章头 -->
49 |             <h3>文章 Header</h3>
50 |         </header>
51 |
52 |         <aside>                                <!-- 使用aside定义文章侧边栏 -->
53 |             <h3>文章 Aside</h3>
54 |         </aside>
55
56 |         <p>这里是文章的内容...</p>
57 |         <p>这里是文章的内容...</p>
58 |
59 |         <footer>                                <!-- 使用footer定义文章底部信息 -->
60 |             <h3>文章 Footer</h3>
61 |         </footer>
62 |     </article>
63 | </section>
64
65 | <aside id="sidebar">                            <!-- 使用aside定义页面的侧边栏 -->
66 |     <h2>Aside</h2>
67 |     <header>                                    <!-- 使用header侧边列表头 -->
68 |         <h2>Sidebar Header</h2>
69 |     </header>
70 |     <nav>                                       <!-- 使用nav定义侧边导航 -->
71 |         <h3>栏目标题</h3>
72 |         <ul>
73 |             <li><a href="#">文章标题一...</a></li>
74 |             <li><a href="#">文章标题二...</a></li>
75 |             <li><a href="#">文章标题三...</a></li>
76 |             <li><a href="#">文章标题四...</a></li>
77 |             <li><a href="#">文章标题五...</a></li>
78 |         </ul>
79 |     </nav>
80 | </aside>
81
82 | <footer id="page_footer">                        <!-- 使用footer定义页面页脚 -->
83 |     <h2>Footer</h2>
84 | </footer>
85 </body>
</html>

```

接下来，我们来看一下这部分 HTML5 页面代码所使用到的 CSS 样式代码，除可用到了 CSS2 中可以让 HTML5 中新增的结构元素而使用的样式，还使用了一些 CSS3 设置一下简单的页面特效。例如，一些圆角和阴影效果，为同一个元素使用多张背景图片等。全部 CSS3 代码如下所示：

```

1  charset 'utf-8';                                /* 设置文档字符集 */
2  body {                                           /* 整个页面的属性设定 */
3      background-color: #CCCCCC;                  /* 背景色 */
4      font-family: Geneva, sans-serif;            /* 可用字体 */
5      margin: 10px auto;                          /* 页边空白 */
6      max-width: 800px;
7      border: solid;                               /* 边缘立体 */
8      border-color: #FFFFFF;                      /* 边缘颜色 */
9  }
10
11
12  a { text-decoration: none; color: #FF0; }        /* 整个body页面的p和a适用 */
13  p { color: #F36; }
14  ul { margin: 0px; padding: 0px; list-style: none; }
15
16  h2, h3 {                                         /* 设定整个body内的h2和h3的共同属性 */
17      text-align: center;                         /* 文本居中 */
18  }
19
20  header {                                         /* 整个body页面的header适用 */
21      background-color: #F47D31;
22      color: #FFFFFF;
23      text-align: center;
24      margin-bottom: 15px;
25  }
26
27  article {                                        /* 整个body页面的article适用 */
28      background-color: #eee;
29  }
30
31  section, aside, header, footer {                /* 几个大区域使用圆角和阴影 */
32      -webkit-box-shadow: 0px 0px 6px #757575;    /* must use for IE */
33      -moz-box-shadow: 0px 0px 6px #757575;
34      box-shadow: 0px 0px 6px #757575;
35      -moz-border-radius: 2px;                    /* Firefox */
36      -webkit-border-radius: 2px;                 /* Safari, Chrome */
37      border-radius: 2px;
38  }

```



```

39
40 nav,article{ /* 共同属性 */
41     margin: 10px;
42     padding: 10px;
43     display: block;
44 }
45
46 header#page_header nav { /* header#page_header nav的属性 */
47     padding: 0 0 10px 0;
48 }
49
50 header#page_header nav ul li { /* header#page_header nav ul li属性 */
51     padding: 0;
52     margin: 0 20px 0 0;
53     display: inline;
54 }
55
56 section#posts { /* #posts 的section属性 */
57     display: block;
58     float: left;
59     width: 70%;
60     height: auto;
61     background-color: #F69;
62     margin-bottom: 15px;
63 }
64
65 section#posts article footer { /* section#posts article footer属性 */
66     background-color: #039;
67     clear: both;
68     height: 50px;
69     display: block;
70     color: #FFFFFF;
71     text-align: center;
72     padding: 15px;
73 }
74
75 section#posts aside { /* section#posts aside属性 */
76     background-color: #069;
77     display: block;
78     float: right;
79     width: 35%;
80     margin-left: 5%;
81     font-size: 20px;
82     line-height: 40px;
83     margin-bottom: 20px;
84 }
85
86 aside#sidebar { /* aside#sidebar属性 */
87     display: block;
88     float: right;
89     width: 28%;
90     height: auto;
91     background-color: #699;
92 }
93
94 aside#sidebar li { /* li 一起使用两张背景图片 */
95     padding: 5px 15px;
96     background: url(li_icon.gif) no-repeat center left, url(dot.gif) repeat-x bottom;
97 }
98
99 footer#page_footer { /* footer#page_footer属性 */
100     display: block;
101     clear: both;
102     width: 100%;
103     display: block;
104     color: #FFFFFF;
105     text-align: center;
106     background-color: #06C;
107 }

```



本例是一个博客系统页面的半成品，目的就是让你了解 HTML5 结构元素和 CSS3 样式的使用。其中在每个区块的盒子中，都标记出来使用的 HTML5 结构元素，让你对 HTML5 的结构元素有一个更直观的认识。本例你可以使用多种浏览器查看效果，包括 IE 的几个常用版本，运行结构如图 1-2 所示：



图 1-2 HTML5 和 CSS3 的页面布局

## 10 LESS CSS 框架简介

CSS 需要书写大量看似没有逻辑的代码，不方便维护及扩展，不利于复用，尤其对于非前端开发工程师来讲，往往会因为缺少 CSS 编写经验而很难写出组织良好且易于维护的 CSS 代码，造成这些困难的很大原因源于 CSS 是一门非程序式语言，没有变量、函数等概念。LESS 为 Web 开发者带来了福音，它在 CSS 的语法基础之上，引入了变量，Mixin（混入），运算以及函数等功能，大大简化了 CSS 的编写，并且降低了 CSS 的维护成本，就像它的名称所说的那样，LESS 可以让我们用更少的代码做更多的事情，它对于大网站海量的页面编写非常有帮助。

本质上，LESS 包含一套自定义的语法及一个解析器，用户根据这些语法定义自己的样式规则，这些规则最终会通过解析器，编译生成对应的 CSS 文件。LESS 并没有裁剪 CSS 原有的特性，更不是用来取代 CSS 的，而是在现有 CSS 语法的基础上，为 CSS 加入程

序式语言的特性。示例如下所示：

```
/* LESS 文件 */
@color: #4D926F;

#header { color: @color; }
h2      { color: @color; }
```

```
/* 经过编译生成的 CSS 文件 */
#header { color: #4D926F; }
h2      { color: #4D926F; }
```

LESS 可以直接在客户端使用，也可以在服务器端使用。在实际项目开发中，我们更推荐使用第三种方式，将 LESS 文件编译生成静态 CSS 文件，并在 HTML 文档中应用。

Mixins（混入）功能对用开发者来说并不陌生，很多动态语言都支持 Mixins（混入）特性，它是多重继承的一种实现，在 LESS 中，混入是指在一个 CLASS 中引入另外一个已经定义的 CLASS，就像在当前 CLASS 中增加一个属性一样。这种方式是我们在开发大型网站多页面时常用的方式，我们先简单看一下 Mixins 在 LESS 中的使用：

```
/* 定义一个样式选择器 */
.roundedCorners(@radius:5px) {
  -moz-border-radius: @radius;
  -webkit-border-radius: @radius;
  border-radius: @radius;
}

/* 在另外的样式选择器中使用 */
#header {
  .roundedCorners;
}
#footer {
  .roundedCorners(10px);
}
```

```
/* 经过编译生成的 CSS 文件 */
#header {
  -moz-border-radius:5px;
  -webkit-border-radius:5px;
  border-radius:5px;
}
#footer {
  -moz-border-radius:10px;
  -webkit-border-radius:10px;
  border-radius:10px;
}
```

上例是我们使用 LESS 最多的情况，如果有海量的样式要编写时，就不需要大量的重复写解决各浏览器兼容性的前缀了。本节只是让你对 LESS 有所了解，在开发中我们可以根据我们的项目特性选择使用 LESS 的部分特性，我们只需用很少的成本就可以换了很大的回报，借助 LESS 可以更便捷的进行 Web 开发。