第3章

CSS 样式设计

3.1 CSS 简介

层叠样式单(Cascading Style Sheets,CSS)是 W3C 协会为弥补 HTML 在显示属性设定上的不足而制定的一套扩展样式标准。CSS 重新定义了 HTML 中原来文字的显示式样,并增加了一些新概念,如类、层等。CSS 重新定义了 HTML 中原来的文字显示样式,并还可以处理文字重叠、定位等,它提供了更丰富的样式,同时,CSS 可集中进行样式管理,允许将样式定义单独存储于样式文件中,把显示的内容和样式定义分离,便于多个文件共享。

1996 年,W3C 提出了一个定义 CSS 的草案,很快这个草案成为一个被广泛采纳的标准。CSS 1 定义了许多简单文本格式化属性,还定义了颜色、字体和边框、级联的原理、CSS 与 HTML 之间的链接机制等属性。

1998 年,W3C 又在原有草案的基础上进行了扩展,建立了 CSS 2 规范功能。CSS 2 使网络开发者能够使用 CSS 布置页面、替换 HTML 表;为特定输出设备创建样式表;对于页面的接收样式部分,具有精细的控制。CSS 2 包含且扩展了 CSS 1 中的所有属性和已经定义的值。

目前,CSS 的最新版本是 CSS 3。CSS 3 语言开发是朝着模块化发展的。以前的规范作模块比较大而且复杂。所以,把它分解为一些小的模块,这些模块包括:盒子模型、列表模型、超链接方式、语言模块、背景和边框、文字效、多栏布局等。

下一版的 CSS 4 仍在开发过程中,直至现在,CSS 4 也只有极少数功能被部分浏览器支持。

使用 CSS 可以很方便地管理显示格式方面的工作。首先,它能够对网页上的元素精确地定位,让网页设计者在网页上自由控制文字图片,使它们按要求显示;其次,它能够实现把网页上的内容结构和格式控制相分离。浏览者想要看的是网页上的内容结构,而为了让浏览者更好地看到这些信息,就要通过格式控制来帮忙。内容结构和格式控制相分离使得网页可以仅由内容构成,而将所有网页的格式控制指向某个 CSS 样式表文件。这样就带来两方面的好处:

(1) 简化了网页的格式代码,外部的样式表还可被浏览器保存在缓存里,从而加快下载的速度。

(2) 只要修改保存着网站格式的 CSS 样式表,文件就可以改变整个站点的风格特色,在改页面数量庞大的站点时,显得格外有用,避免了网页一个一个地修改,大大减少了重复的工作量。

3.2　CSS 语法与使用

3.2.1　CSS 定义语法

CSS 是格式化网页的标准方法，它就颜色、字体、间隔、定位以及边距等几十种属性，这些属性可通过 Style 应用于 HTML 标记中，但篇幅限制，本书只讨论其中常见的几个属性，包括字体、颜色、背景等。CSS 样式表是由许多样式规则组成的，用来控制网页元素的显示方式。

【语法】

```
选择符{属性 1: 值 1; 属性 2: 值 2…}
```

规则由选择符以及紧跟其后的一系列“属性：值”对组成，所有“属性：值”对用“{}”包括，各“属性：值”对之间用分号“；”分隔。如

```
p{color: red ; font - size:20pt}
```

其中，p 是选择符；color：red，font-size：20pt 是“属性：值”对。本规则表示所有< p >标的文字颜色为红色、大小为 20 磅。

样式属性值有以下几条规则：

(1) 如果属性的值由多个单词组成，则必须在值上加引号，如字体的名称经常是几个单词的组合。

```
p{font - family: 'sans serif'}
```

(2) 属性值不区分大小写，如 small，Small，SMALL，smALL 都是一样的。

(3) 当属性值没有具体的单位表示物理意义时，就用数值表示，数值可以是整数或小数，可以是正数或负数。

(4) 用数值指定长度时，后面用 2 个字符组成单位的缩写，数字和单位之间不能用空格隔开。常见的长度单位见表 3-1。这些长度是近似值，具体长度取决于屏幕分辨率。

表 3-1　常见的长度单位

名　称	像　素	点	厘　米	毫　米	12 点活字
全称	pixel	point	centiment	millimeter	pica
缩写	px	pt	cm	mm	pc

(1) 相对长度 em，ex 分别表示当前字体的大小和字母 x 的高度。

(2) 百分比类型的属性值表示相对于前面使用过的尺寸的百分比值。

(3) 许多属性值可以被后代元素继承，如 fon-size，如果定义< body >标签的字体大小为< body font-size:5px >，则文档中的所有元素的字体大小默认都会继承这个值。也有部分属性是不能被继承的，如 backgroundcolor、边距等。

3.2.2　CSS 的使用

CSS 规则总是由 Web 浏览器进行解释，HTML 和 CSS 标准对浏览器应该如何显示这

些规则作了明确规定——但是，它们并非总是遵守这些规定。用 CSS 设计网页，用户不但需要知道这些标准（如 CSS 规范所述），而且还要理解浏览器的特性和缺点会如何影响 Web 设计的结果。

如果浏览器在编写时，就让它理解所遇到的情况，那么它就会根据规范尝试显示相应的内容。如果浏览器不知道遇到的情况，就会忽略它。这两种选择可以认为是“正确执行”。否则，浏览器可能错误执行。浏览器可能对遇到的情况迷惑不解，以一些非标准的方式显示，甚至会崩溃，虽然这很少发生。当然，上述错误是用户不希望看到的，也是问题的根源。

级联样式表从开始就设计成能合理地退化处理。意思就是，如果因为某些原因不能识别 CSS 规则，页面仍然可用，仍然可以访问其内容。因为显示与内容是分开的，虽然在删除显示效果后不是很漂亮，但内容应该能够独立显示。

前面介绍了 CSS 的语法，但要在浏览器中显示出效果，就要让浏览器识别并调用样式表，当浏览器读取样式表时，要依照文本格式来读。为网页添加样式表的方法有 4 种：链入外部样式表、导入外部样式表、联入样式表和内联样式。其中，联入样式表和内联样式是将 CSS 的功能组合于 HTML 文件之内，而链入及导入外部样式表则是将 CSS 功能以文件方式独立于 HTML 文件之外，然后再通过链入或导入的方式将 HTML 文件和 CSS 文件链接在一起。

1. 链入外部样式表

链入外部样式表是把样式表保存为一个 CSS 文件，标记放置在 HTML 文档头部，在 HTML 的头信息标识符< head >里添加< link >标记链接到这个 CSS 文件即可使用。外部样式表不能含有任何像< head >或< style >这样的 HTML 标记，仅仅由样式规则或声明组成，并且只能以.CSS 为扩展名。

【语法】

```
<link rel = "样式与文档",href = "CSS 文件",type = "",media = "输出媒体" >
```

属性主要有：

rel 属性表明样式表将以何种方式与 HTML 文档结合。一般取值为 Stylesheet，指定一个外部的样式表。

href 属性指出 CSS 文件的地址，如果样式文件和 HTML 文件不是放在同路径下，则要在 href 里加上完整路径。

type 属性指出样式类别，通常取值为 text/CSS。

media 是可选的属性，表示使用样式表的网页将用什么媒体输出，取值范围包括：（默认）输出到计算机屏幕；print，输出到打印机；projection，输出到投影仪等。

一个外部样式表文件可以应用于多个页面。当改变样式表文件后，所有页面的样式都随之而改变。在制作大量相同样式页面的网站时非常有用，不仅减少了重复的工作量，而且有利于以后的修改、编辑。同时，大多数浏览器会保存外部样式表在缓冲区，从而浏览时可减少重复下载代码，避免展示网页时的延迟。

【例 3-1】 链入外部样式表实例。

第 1 步，用 VS 2013 新建空网站 CSSWebsites 项目。

第 2 步，添加一个 Web 窗体，命名为 Ex3-1.HTML，添加代码：

```
<HTML>
<head>
<title>链接样式表 CSS 示例</title>
<link rel = "stylesheet" type = "text/CSS" href = "3 - 1.CSS" media = "screen">
</head>
<body topmargin = 4 >
<h1 >这是一个链接样式 CSS 示例!</h1 >
<span class = "mytext">这行文字应是红色的.</span>
<p>这一段底色应是黄色.</p>
</body>
</HTML>
```

第 3 步,添加一个样式表文件,命名为 3-1.CSS,代码如下:

```
h1{font - family:"隶书","宋体";color: # ff8800}
p{background - color:yellow;color: # 000000}
.mytext{font - family:"宋体";font - size:14pt;color:red}
```

第 4 步,运行程序,结果如图 3-1 所示。

图 3-1　运行结果

2. 导入外部样式表

导入外部样式表是指在 HTML 文件头部的< style >…</style >标记之间,利用 CSS 的@import 声明导入外部样式表。@Import "3-1.CSS"表示导入 3-1.CSS 样式表,注意使用时外部样式表的路径,这和链入外部样式表的方法很相似,但导入外部样式表输入方式更有优势,因为除外部样式外,还可添加本页面的其他样式。注意:@import 声明必须放在样式表的开头部分,其他 CSS 规则应仍然包括在 style 元素中。

【例 3-2】 import 导入样式表。

第 1 步,为 CSSWebsites 项目添加一个 Web 窗体,命名为 Ex3-2.HTML,添加如下代码:

```
<HTML>
<head>
<title>导入外部样式表 CSS 示例</title>
<style type = "text/CSS">
<! -- @import "3 - 1.CSS"; -- >
h2 {font - family:"隶书","宋体";color:blue}
</style>
</head>
<body topmargin = 4 >
<h1 >这是一个链接样式 CSS 示例!</h1 >
<span class = "text">这行文字应是红色的.</span>
<h2 >这行文字应是蓝色的.</h2 >
<p>这一段底色应是黄色.</p>
</body>
```

```
</HTML>
```

第 2 步，运行程序，结果如图 3-1 所示。

3. 联入样式表

利用< style >标记将样式表联入 HTML 文件的头部。style 元素放在文档的 head 部分，必需的 type 属性用于指出样式类别，通常取值为 text/CSS、有些低版本的浏览器不能识别 style 标记，这意味着低版本的浏览器会忽略 style 标记里的内容，直接以源代码的方式在网页上显示设置的样式表。为了避免这种情况发生，用加 HTML 注释的方式(<! —注释-->)隐藏内容而不让它显示。联入样式表的作用范围是本 HTML 文件。

【例 3-3】 联入样式表实例。

第 1 步，为 CSSWebsites 项目添加一个 Web 窗体，命名为 Ex3-3. HTML，添加如下代码：

```
< HTML >
< head >
< style type = "text/CSS">
    body {background - image:url(022.gif);}
</style >
</head >
< body >
</body >
</HTML >
```

第 2 步，运行程序，结果如图 3-2 所示。

图 3-2 程序结果

4. 内联样式

内联样式是混合在 HTML 标记里使用的，用这种方法，可以很简单地对某个元素单独定义样式，它是连接样式和 HTML 标记的最简单的方法。内联样式的使用即直接在 HTML 标记里加入 style 参数，而 style 参数的内容就是 CSS 的属性和值。

此时样式定义的作用范围仅限于此标记范围之内。style 属性是随 CSS 扩展出来的，它可以应用于任意 body 元素(包括 body 本身)，除了 basefont、param 和 script。还应注意，若要在一个 HTML 文件中使用内联样式，可以在文件头部对整个文档进行单独的样式表语言声明，即< Meta http－equiv＝"Content-Type" content ＝"text/CSS">。

如果使用内联样式向标记中添加太多属性及内容，对于网页设计者来说很难维护，更难阅读，而且由于它只对局部起作用，因此必须对所有需要的标签都作设置，这样就失去了CSS 在控制页面布局方面的优势。所以，内联样式主要用于样式仅适用于单个页面的情况，应尽量减少使用内联样式，而采用其他样式。

【例 3-4】 内联样式。

第 1 步，为 CSSWebsites 项目添加一个 Web 窗体，命名为 Ex3-4. HTML，添加代码：

```
<! DOCTYPE HTML >
< HTML xmlns = "http://www.w3.org/1999/xHTML">
< head >
< Meta http - equiv = "Content - Type" content = "text/HTML; charset = utf - 8"/>
    < title ></title >
    < h1 style = " font - family: '隶书','宋体';color: # ff8800">这是一个链接样式 CSS 示例!
</h1 >
</head >
< span style = "color:red;">这行文字应是红色的.</span >
  < p style = "color:red;background - color:yellow">这一段底色应是黄色.</p >
  < body style = "font - family: '宋体';font - size:14pt" >
  </body >
</HTML >
```

第 2 步，运行程序，结果如图 3-1 所示。

5. 多重样式表的叠加

有时会遇到这样一种情况：几个不同的样式使用了同一个选择器，此时 CSS 会对各属性值进行叠加处理，遇到冲突的地方，会以最后定义的为准。

按照后定义优先的原则，优先级最高的是内联样式。联入样式、导入外部样式、链入外部样式之间则是最后定义的优先级最高。

首先链入一个外部样式表，其中定义了 h3 选择符的 color、text-align 和 font-size 属性。

```
h3 {color: red;text - align: left;font - size: 8pt;}
```

然后在内部样式表里也定义了 h3 选择符的 text-align 和 font-size 属性。

```
h3 {text - align: right; font - size: 20pt;}
```

那么，这个页面叠加后的样式就是：

```
color: red; text - align: right; font - size: 20pt;
```

即标题 3 的文字颜色为红色；向右对齐；尺寸为 20 号字。字体颜色从外部样式表里保留下来，而对齐方式和字体尺寸都有定义时，按照后定义优先的原则。

3.2.3　选择符

CSS 中有 HTML 标记类选择符、具有上下文关系的 HTML 标记类选择符、用户自定义类选择符、用户定义的 ID 选择符、虚元素和虚类。

1. HTML 标记类选择符

直接用 HTML 标记或 HTML 元素名称作为选择符，例如：

```
td,input,select,body {font - family:Verdana;font - size;e;12px;}
form,body {margin: 0; padding: 0 }
select,body,textarea {background?and: #fff; font - size:12px;}
select (font - size: 13px; }
Img{border: none}
a {text - decoration: under1ine; cursor: pointer; }
h1{color:#ff0000}
```

2. 具有上下文关系的 HTML 标记类选择符

如果定义了这样的样式规则 Body{color：blue}，则网页中所有的文字都以蓝色显示。因为 body 标记包含了所有的其他标记符，除非另外指定样式或格式来更改这一设定。可以为某个容器元素的内部元素设置特定的样式规则，使其具有上下文关系的 HTML 标记。例如，如果只想使位于 h1 标记符的 b 标记符具有特定的属性，使用的格式应为

```
h1 b{color: blue}
```

注意：元素之间以空格分开，表示只有位于 h1 标记内的 b 元素具有蓝色属性，其他任何 b 元素都保持原有颜色。

【例 3-5】 上下文关系的 HTML 标记类选择。

第 1 步，为 CSSWebsites 项目添加一个 Web 窗体，命名为 Ex3-5. HTML，添加如下代码：

```
<HTML>
<head>
<title>CSS 选择符问题 </title>
<style type = text/CSS >
input {color: white;}
div input {color:red}
div span b {color:yellow}
</style>
</head>
<body>
<input type = "text" value = "change me">
<br>
<div>
    <input type = "text" value = "change me"><br>
    <span> I'm a <b> good </b> student </span>
</div>
</body>
</HTML>
```

第 2 步，运行程序，结果如图 3-3 所示。

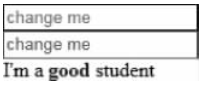

图 3-3 程序结果

3. 用户自定义类选择符

使用类选择符能对相同的标记分类，并定义成不同的样式。定义类选择符时，在定义类的名称前面加一个点号；假如想要两个不同的段落，一个段落右对齐，一个段落居中，可以先定义两个类：

```
p.right(text - align:right}
center(text - align:center }
```

“.”符号后面的 right 和 center 为类名。类的名称可以是用户自定义的任意英文单词

或以英文开头的与数字的组合，一般以其功能和效果简要命名。如果要用在不同的段落里，在 HTML 标记里加入前面定义的类即可。

```
<p class = "right">这个段落右对齐</p>
<p class = "center">这个段落是居中排列的</p>
```

用户定义的类选择符的一般格式是，

```
selector.classname{属性:值; …}
```

类选择符还有一种用法，在选择符中，省略 HTML 标记名，这样可以把几个不同的元素定义成相同的样式。例如：.center(text-align：center)。自定义 Center 类选择符为文字居中排列。这样可以不限定某个 HTML 标记，而将其应用到任何元素上。例如，将 hl 元素标题 D 和 p 元素都设为 center 类，使这两个元素的文字居中显示。

```
<h1 class = "center">这个标题是居中排列的</h1>
<p class = "center">这个段落也是居中排列的</p>
```

这种省略 HTML 标记的类选择符是最常用的 CSS 方法，使用这种方法可以很方便地在任意元素上套用预先定义好的类样式，但是前面的“.”号不能省略。

4. 用户定义的 ID 选择符

用户定义的 ID 选择符的语法如下：

```
#IDname{property:value; …}
```

其中，IDname 为某个标记 ID 属性的值。ID 选择符的用途及概念和类选择符相似，不同之处在于，同一个 ID 选择符样式只能在 HTML 文件内被应用一次，而类选择符样式则可以多次被应用。也就是说，如果有些较特别的标记需要应用较特殊的样式，则建议使用 ID 选择符。定义 ID 选择符时用“#”号开头，而不是“.”号。

5. 虚元素

在 CSS 中有两个特殊的虚元素选择符，用于 p、div、span 等块级元素的首字母和首行效果，它们是：first-letter 和 first-line。不过，有些浏览器不支持这两个虚元素。它们的使用语法如下：

```
选择符:first - letter{property:value; …}
选择符:first - line{property: value; …}
选择符.类:first - letter{property:value; …}
选择符.类:first - line{property: value; …}
```

也可在样式中定义 $ span{font-size:200%)，然后在需要的位置应用该样式来实现首字母效果。例如，“< p >< span >前面了解了 CSS 的语法</p >”可以达到同样的效果。

6. 虚类

对于超链接 a 标记符，可以使用虚类方式设置不同类型访问链接的显示方式。虚类是一种特殊的类选择符，能被支持的浏览器自动识别的特殊选择符。

【语法】

```
选择符:虚类{property:value; …}
```

定义虚类的方法和常规类很相似，但有两点不同：一是连接符是冒号，而不是句点号；二是虚类有预先定义好的名称，也就是链接可处在 4 种不同的状态下，即 link、visited、active、hover，分别代表未访问的链接、已访问的链接、活动链接和鼠标停留在链接上。

【例 3-6】 上下文关系的 HTML 标记类选择。

第 1 步，为 CSSWebsites 项目添加一个 Web 窗体，命名为 Ex3-6.HTML，添加如下代码：

```
<HTML>
<head>
<title>CSS 伪类示例</title>
<style type = "text/CSS">
a:link {font - size: 18pt; font - family:隶书; text - decoration:none}      /* 未访问的链接 */
a:visited {font - size: 18pt;font - family:宋体;text - decoration:line - through}
                                                                          /* 已访问链接 */
a:hover {font - size: 18pt; font - family:黑体;text - decoration:overline}      /* 鼠标在链接上 */
a:active {font - size:18pt; font - family:幼圆; text - decoration:underline}   /* 活动链接/
</style>
</head>
<body>
<a href = "http://Web.cse.cslg.cn">计算机学院</a>
</body>
</HTML>
```

图 3-4 程序结果

第 2 步，运行程序，得到图 3-4(a)，鼠标进入区域后得到图 3-4(b)，选中区域后得到图 3-4(c)，访问之后呈现如图 3-4(d)所示的结果。

在例 3-6 中，链接未访问时的字体是隶书且无下画线；访问后是宋体，并打上了删除线；单击变成活动链接时字体为幼圆并有下画线；鼠标在链接上时为黑体和上画线。根据层叠顺序，定义这些链接样式时，一定要按照 a:link，a:visited，a:hover，a:active 的顺序书写。

还可以将伪类和类选择符及其他选择符组合起来使用，其形式如下：

```
选择符.类: 伪类{property: value; …}
```

可以在同一个页面中做出几组不同的链接效果。

3.3 CSS 样式设计简介

3.3.1 字体样式

几乎所有的 HTML 文档都包含了文本，文本的字体属性是最常用的样式。字体样式包括字体族(font-family)、字体大小(font-size)、字体风格(font-style)、字体变体(font-variant)，字体粗细(font-weight)、字体综合设置(font)属性。

1. 字体族(font-family)

字体族是指字体名称,浏览器利用字体列表中能够支持的第一个字体显示文本。

【语法】

```
font-family:第 1 个字体名称,第 2 个字体名称,…,第 n 个字体名称
```

如果浏览器支持第 1 个字体,则用这种字体显示;否则,判断第 2 个字体是否支持,如果支持这种字体,则用第 2 个字体,否则判断第 3 个字体,以此类推。

font-family 的属性值可以指定一个通用的字体。常见的通用字体见表 3-2。每个浏览器都有自己的字体,可以把通用字体放到属性指定的最后字体,如果浏览器不支持指定的字体,可以从相同类别字体中选择一种可用的字体,如 font-family: Arial,Helvetica,Futura,sans-serif。

如果一个字体的名称不止一个单词,那么整个字体名称需要用单引号,如 font-family: 'Times New Roman'。

表 3-2　常见的通用字体

通用字体名称	对 应 字 体
serif	Times New Roman,Garamond
sans-serif	Arial,Helvetica
cursive	Caflisch Script,Zapf-Chancery
fantasy	Critter,Cottonwood
monospace	Courier,Prestige

2. 字体大小(font-size)

字体大小分两种:绝对大小和相对大小。

字体大小的绝对值单位有点、12 点活字、像素,或用关键字 xx-small,x-small,small,medium,large,x-large,xx-large。使用关键字的大小时,相邻两个关键字之间的大小不同,浏览器不一样大。

字体大小的相对值有 smaller 和 larger。它们是根据父元素的字体大小调整子元素的字体大小,具体调整的程度取决于浏览器。也可以用百分比值调整子元素的大小,不同浏览器用百分比的相对大小是一样的。最后,可以以 em 为单位表示相对大小。font-size: 120%和 font-size: 1.2em 是等价的。

3. 字体风格(font-style)

font-style 属性用于设置使用斜体、倾斜或正常字体。斜体字体通常定义为字体系列中的一个单独的字体。可能的取值有 3 种。

normal:默认值取值,表示浏览器显示一个标准的字体样式。

italic:浏览器会显示一个斜体的字体样式。

oblique:浏览器会显示一个倾斜的字体样式。

后两者的显示效果差不多,都是向右倾斜,italic 的字体衬线稍微长一点。

4. 字体变体(font-variant)

font-variant 属性用于设置小型大写字母的字体显示文本,这意味着所有的小写字母均会转换为大写,但是所有使用小型大写字体的字母与其余文本相比,其字体尺寸更小。可能的取值有:

normal：默认的取值，表示浏览器会显示一个标准的字体。

small-caps：浏览器会显示小型大写字母的字体。

5. 字体粗细(font-weight)

font-weight 属性用于设置显示元素的文本中所用的字体加粗。数字值 400 相当于关键字 normal，700 等价于 bold，可能的取值有 100，200，300，400，500，600，700，800，900。每个数字值对应的字体加粗必须至少与下一个最小数字一样细，而且至少与下一个最大数字一样粗。可能的取值有：

normal：默认值，表示定义标准的字符。

bold：表示定义粗体字符。

Bolder：表示定义更粗的字符。

Lighter：表示定义更细的字符。

或 100，200，300，400，500，600，700，800，900 中的一个值。

6. 字体综合设置(font)

font 简写属性在一个声明中设置所有字体属性。这个简写属性用于一次设置元素字体的两个或更多方面，至少要指定字体大小和字体系列。font 各个属性的顺序很重要，font-family 必须放到最后，font-size 为倒数第 2，其他属性可有可无，如果有，则位于 font-size 之前。

【例 3-7】 字体样式的综合运用。

第 1 步，为 CSSWebsites 项目添加一个 Web 窗体，命名为 Ex3-7.HTML，添加如下代码：

```
<!DOCTYPE HTML>
<HTML lang = "en">
<head>
    <title>字体样式的综合运用</title>
    <Meta charset = "utf-8" />
<style type = "text/CSS">
p.myp1 { font-family: Arial,Helvetica,sans-serif; font-size:xx-small }
p.myp2 { font-family: Arial,Helvetica,sans-serif; font-size:xx-large }
p.myp3{font-size:1.2em;font-variant:small-caps}
p.myp4{font-size:1.2em; font-style:italic}
p.myp5{font-weight:bolder; font-size:1.2em; font-style:oblique}
p.myp6{ font: bold 1.1em 'Times New Roman'}
</style>
</head>
<body>
<p class = "myp1"> 绝对值的大小单位有：点、12 点活字、像素，或用关键字 xx-small,x-small,
small,medium,large,x-large,xx-large </p>
<p class = "myp2"> 绝对值的大小单位有：点、12 点活字、像素，或用关键字 xx-small,x-small,
small,medium,large,x-large,xx-large </p>
<p class = "myp3">all possible values: xx-small,x-small,small,medium,large,x-large,xx-
large </p>
<p class = "myp4"> It's italic style. </p>
<p class = "myp5"> It's oblique style. </p>
<p class = "myp6"> font 简写属性在一个声明中设置所有字体属性 </p>
```

```
</body>
</HTML>
```

第 2 步，运行程序，结果如图 3-5 所示。

图 3-5　运行结果

3.3.2　文本样式

设置文字之间的显示特性包括字符间隔、单词间距、文本行高、文本修饰、大小写转换。

1. 字符间隔(letter-spacing)

letter-spacing 用于控制单词的字符之间的距离。

【语法】

```
letter - spacing:参数
```

参数的取值：normal，恢复到与父元素相同。任意长度属性的值，长度值为正值会增加字母之间的距离，为负值会减少间距。

2. 单词间距(word-sapcing)

word-sapcing 用于控制单词之间的距离。

【语法】

```
word - sapcing:参数
```

参数的取值：normal，恢复到与父元素相同。任意长度属性的值，长度值为正值会增加单词之间的距离，为负值会减少间距。

3. 文本行高(line-height)

行高是指上下两行基准线之间的垂直距离。

【语法】

```
line - height:参数
```

参数的取值：不带单位的数字，以 1 为基数，相当于比例关系的 100%。带长度单位的数字，以具体的单位为准。

normal，恢复到与父元素相同。

【例 3-8】 文本间距的实例。

第 1 步，为 CSSWebsites 项目添加一个 Web 窗体，命名为 Ex3-8.HTML，添加如下代码：

```
<HTML lang = "en">
  <head>
    <title> Text spacing properties </title>
    <Meta charset = "utf-8" />
    <style type = "text/CSS">
        p.bigtracking {letter-spacing: 0.4em;}
        p.smalltracking {letter-spacing: -0.08em;}
        p.bigbetweenwords {word-spacing: 0.4em;}
        p.smallbetweenwords {word-spacing: -0.1em;}
        p.bigleading {line-height: 2.5;}
        p.smallleading {line-height: 1.0;}
    </style>
  </head>
  <body>
    <p class = "bigtracking">
      字符间距 On the plains of hesitation [letter-spacing: 0.4em]
    </p><p />
    <p class = "smalltracking">
     字符间距 Bleach the bones of countless millions [letter-spacing: -0.08em]
    </p><br />
    <p class = "bigbetweenwords">
     单词间距 Who at the dawn of victory [word-spacing: 0.4em]
    </p><p />
    <p class = "smallbetweenwords">
     单词间距 Sat down to wait and waiting died [word-spacing: -0.1em]
    </p><br />
    <p class = "bigleading">
     行高 If you think CSS is simple,[line-height: 2.5]<br />
      You are quite mistaken
    </p><br />
    <p class = "smallleading">
     行高 If you think HTML 5 is all old stuff,[line-height: 1.0]<br />
      You are quite mistaken
    </p>
  </body>
</HTML>
```

第 2 步，运行程序，结果如图 3-6 所示。

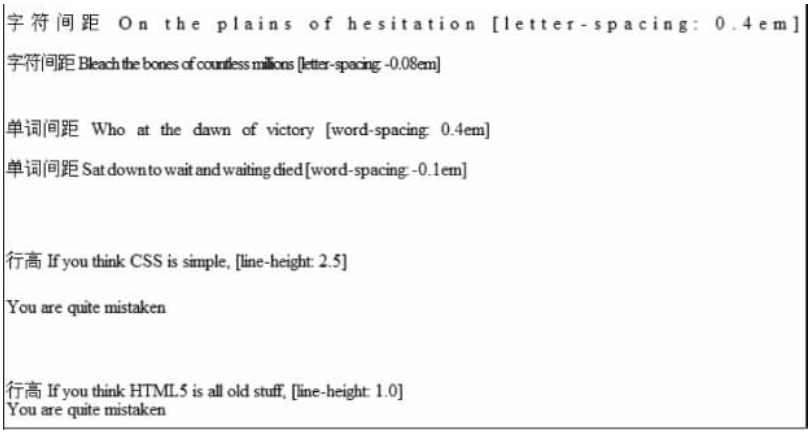

图 3-6 运行结果

4. 文本修饰(text-decoration)

文本修饰的主要用途是改变浏览器显示文字链接时的下画线。

【语法】

```
text-decoration: 参数
```

参数的可能取值：

underline,为文字加下画线。

overline,为文字加上画线。

line-through,为文字加删除线。

blink,使文字闪烁。

none,删除下画线。

【例 3-9】 文本修饰。

第 1 步,为 CSSWebsites 项目添加一个 Web 窗体,命名为 Ex3-9.HTML,添加如下代码：

```
<HTML lang = "en">
  <head>
    <title> Text decoration </title>
    <Meta = charset = "utf-8" />
    <style type = "text/CSS">
      p.through {text-decoration: line-through;}
      p.over {text-decoration: overline;}
      p.under {text-decoration: underline;}
    </style>
  </head>
  <body>
    <p class = "through">
      This illustrates line-through
    </p>
    <p class = "over">
      This illustrates overline
    </p>
    <p class = "under">
      This illustrates underline
    </p>
  </body>
</HTML>
```

第 2 步,运行程序,结果如图 3-7 所示。

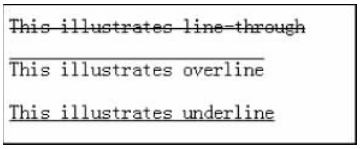

图 3-7　运行结果

5. 大小写转换(text-transform)

文字大小写转换使网页的设计者不用在输入文字时就完成文字大写,而是在输入完毕后,根据需要再对局部的文字设置大小写。

【语法】

```
text-transform: 参数
```

参数的可能取值：uppercase,所有文字大写显示；lowercase,所有文字小写显示；captalize,每个单词的头字母大写显示；none,不继承母体的文字变形参数。

3.3.3 颜色样式

颜色有 3 种定义方式。

第 1 种是 Web 页面的原始组，有 17 种颜色，直接使用的名字的颜色值称为命名颜色，但是颜色数太少。CSS 支持 17 种合法命名颜色（标准颜色）：aqua，fuchsia，lime，olive，red，white，black，gray，maroon，orange，silver，yellow，blue，green，navy，purple，teal。W3C 的 HTML 4.0 标准仅支持 16 种颜色名，它们是 aqua，black，blue，fuchsia，gray，green，lime，maroon，navy，olive，purple，red，silver，teal，white、yellow。

第 2 种包含了 147 种命名颜色，被浏览器广泛支持，具体名称和颜色图参看 http://www.w3school.com.cn/tags/HTML_ref_colornames.asp。

第 3 种是调试板，有 216 种颜色，又称为 216 种 Web 安全颜色，之所以不是 256 种 Web 安全颜色，是因为 Microsoft 和 Mac 操作系统有 40 种不同的系统保留颜色。

1. 前景色属性（color）

【语法】

```
color:参数
```

参数的取值可以参考以上 3 种预定义的颜色和 2.2.2 节关于字体颜色的内容。

2. 背景色属性（background-color）

【语法】

```
background-color:参数
```

参数的取值可以参考以上 3 种预定义的颜色和 2.2.2 节关于字体颜色的内容。

目前，专业的 Web 设计师喜欢自己定义颜色。有关 Web 配色基础的色彩设计方法，可以参阅 http://www.shejidaren.com/se-cai-she-ji-fang-fa.HTML。很多网站提供了一些经典的配色方案，如 http://www.xin126.cn/show.asp?id=3141 上提供了 13 套 Web 页面标准配色方案，部分方案如图 3-8 所示。

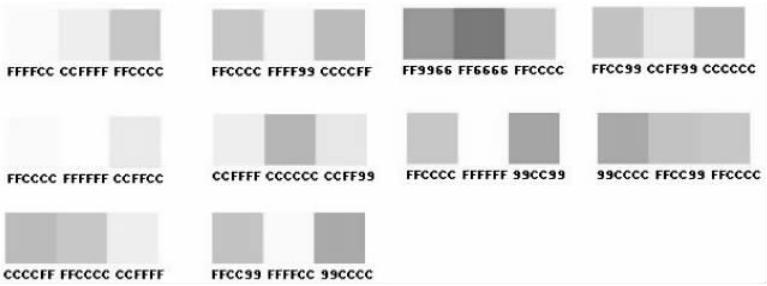

图 3-8 柔和、明亮和温和的 Web 配色方案

3.3.4 列表样式

用于设置列表标记（< ol >和< ul >）的显示特性包括 list-style-type、list-style-image、list-style-position、list-style 等。

(1) list-style-type:表示项目符号。

【语法】

```
list - style - type: 参数
```

参数取值如下:

无序列表的项目符号形状值:disc-实心圆点;circle-空心圆;square-实心方形。

有序列表值:decimal-阿拉伯数字,如 1、2、3、4 等;lower-roman-小写罗马数字,如 i、ii、iii、iv 等;Upper-roman-大写罗马字母,如Ⅰ、Ⅱ、Ⅲ、Ⅳ等;lower-alpha-小写英文字母,如 a、b、c、d 等;Upper-alpha-大写英文字母,如 A、B、C、D 等;none-不设定;lower-greek:小写希腊字母,如 α(alpha)、β(beta)、γ(gamma)等。

(2) list-style-image:使用图像作为项目符号。

【语法】

```
list - style - image: url(URL)
```

(3) list-style-position:设定项目符号是否在字幕里面也文字对齐。格式:list-style-position;outside/inside。

(4) list-style:综合设置项目属性。格式:list-style:type,position。

【例 3-10】 无序列表样式。

第 1 步,为 CSSWebsites 项目添加一个 Web 窗体,命名为 Ex3-10.HTML,添加如下代码:

```
<HTML>
<head>
<style type = "text/CSS">
ul.disc {list - style - type: disc}
ul.circle {list - style - type: circle}
ul.square {list - style - type: square}
ul.none {list - style - type: none}
ul.myimage{list - style - image: url('images\\down.gif')}
</style>
</head>
<body>
<ul class = "disc">
<li>咖啡</li>
<li>茶</li>
<li>可口可乐</li>
</ul>
<ul class = "circle">
<li>咖啡</li>
<li>茶</li>
<li>可口可乐</li>
</ul>
<ul class = "square">
<li>咖啡</li>
<li>茶</li>
<li>可口可乐</li>
</ul>
<ul class = "none">
<li>咖啡</li>
```

```
<li>茶</li>
<li>可口可乐</li>
</ul>
<ul class = "myimage">
<li>咖啡</li>
<li>茶</li>
<li>可口可乐</li>
</ul>
</body>
</HTML>
```

第 2 步,运行程序,结果如图 3-9 所示。

【例 3-11】 有序列表样式。

第 1 步,为 CSSWebsites 项目添加一个 Web 窗体,命名为 Ex3-11. HTML,添加如下代码:

```
<HTML>
<head>
<style type = "text/CSS">
ol.decimal {list - style - type: decimal}
ol.lroman {list - style - type: lower - roman}
ol.uroman {list - style - type: upper - roman}
ol.lalpha {list - style - type: lower - alpha}
ol.ualpha {list - style - type: upper - alpha}
</style>
</head>
<body>
<ol class = "decimal"><li>咖啡</li><li>茶</li><li>可口可乐</li></ol>
<ol class = "lroman"><li>咖啡</li><li>茶</li><li>可口可乐</li></ol>
<ol class = "uroman"><li>咖啡</li><li>茶</li><li>可口可乐</li></ol>
<ol class = "lalpha"><li>咖啡</li><li>茶</li><li>可口可乐</li></ol>
<ol class = "ualpha"><li>咖啡</li><li>茶</li><li>可口可乐</li></ol>
</body>
</HTML>
```

第 2 步,运行程序,结果如图 3-10 所示。

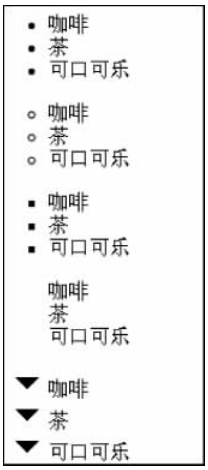

图 3-9 无序列表样式结果

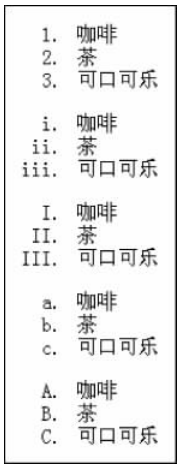

图 3-10 有序列表样式结果

3.3.5　表格样式

1. 设置表格宽度(width)、高度(height)

表格的宽度和高度可以使用< table >标签的属性 width 和 height 设置,单位都可以用像素,width 还可以用%表示。如果要控制行高,可以用< tr >或< td >标签的属性 height 设置。

【语法】

```
< table width = "宽度" height = "高度">
< tr height = "高度" > < td > … </td ></tr >
… .
</table >
```

2. 设置边框样式的 border、frame 和 rules 属性

表格的边框粗细和外边框样式可以使用< table >标签的 border 和 frame 属性设置,其中 frame 常见的属性值表 3-3;表格内边框使用< table >标签的 rules 属性设置。rules 常见的属性值见表 3-4。

【语法】

```
< table border = "边框粗细" frame = "外边框" rules = "内边框">
< tr > < td > … </td ></tr >
… .
</table >
```

其中,边框粗细的单位是像素。

表 3-3　frame 常见的属性值

属　性　值	说　　明
above	显示上边框
border	显示上、下、左、右边框
below	显示下边框
hsides	显示上、下边框
lhs	显示左边框
rhs	显示右边框
void	不显示边框
vsides	显示左、右边框

表 3-4　rules 常见的属性值

属　性　值	说　　明
all	显示所有内部边框
groups	显示介于行列边框
none	不显示内部边框
cols	仅显示列边框
rows	仅显示行边框

【例 3-12】 表格内外边框属性的使用。

第 1 步，为 CSSWebsites 项目添加一个 Web 窗体，命名为 Ex3-12. HTML，添加如下代码：

```
<HTML>
<body>
<p><b>注释: frame 外边框属性设置</b>rules 内边框属性</p>
<p>Table with rules = "rows":</p>
<table rules = "rows" frame = "border">
<tr>
<th>Month</th>
<th>Savings</th>
</tr>
<tr>
<td>January</td>
<td>$ 100</td>
</tr>
</table>
<p>Table with rules = "cols":</p>
<table rules = "cols" >
<tr>
<th>Month</th>
<th>Savings</th>
</tr>
<tr>
<td>January</td>
<td>$ 100</td>
</tr>
</table>
<p>Table with rules = "all":</p>
<table rules = "all">
    <tr>    <th>Month</th>    <th>Savings</th>  </tr>
    <tr>    <td>January</td>  <td>$ 100</td>    </tr>
</table>
</body>
</HTML>
```

第 2 步，运行程序，结果如图 3-11 所示。

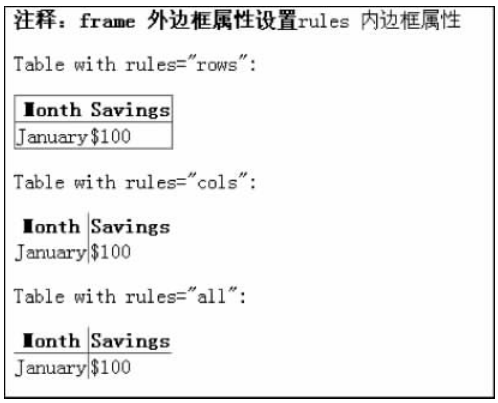

图 3-11 表格样式结果

3. 水平和垂直对齐属性

设置行内水平对齐方式，需要设置< tr >标记的 align 属性值。常用的 align 属性值有 left、right 和 center，分别表示左对齐、右对齐和居中对齐。

设置行内垂直对齐方式，需要设置< tr >标记的 valign 属性值。常用的 valign 属性值有顶端对齐(top)、居中对齐(middle)、底线对齐(bottom)、文本的底端对齐(text-bottom)、文本的顶端对齐(text-top)和基线(baseline)。

【语法】

```
< table >
< tr align = "水平对齐方式"> < td > … </td ></tr >
< tr valign = "垂直对齐方式"> < td > … </td ></tr >
…
</table >
```

4. 设置单元格间距与边距属性 cellspacing 和 cellpadding

在 HTML 文件中，表格中单元格的间距由< table >标签的 cellspacing 属性值设置；表格单元格的内容与边框之间的间距由< table >标签的 cellpadding 属性值设置。

【语法】

```
< table cellspacing = "单元格的间距或 %" cellpadding = "内容与边框的间距或 %">
    < tr > < td > … </td > </tr >
    …
</table >
```

【例 3-13】　表格属性的使用。

第 1 步，为 CSSWebsites 项目添加一个 Web 窗体，命名为 Ex3-13. HTML，添加如下代码：

```
< HTML >
< head >
< style type = "text/CSS">
table. hovertable {
font - family: verdana,arial,sans - serif;
font - size:11px;
color: # 333333;
border - width: 1px;
border - color: # 999999;
border - collapse: collapse;
}
table. hovertable th {
background - color: # c3dde0;
border - width: 1px;
padding: 8px;
border - style: solid;
border - color: # a9c6c9;
}
table. hovertable tr {
background - color: # d4e3e5;
}
```

```
table.hovertable td {
border-width: 1px;
padding: 8px;
border-style: solid;
border-color: #a9c6c9;
}
</style>
</head>
<body>
<table class="hovertable">
<tr>
<th>Info Header 1</th><th>Info Header 2</th><th>Info Header 3</th>
</tr>
<tr onmouseover="this.style.backgroundColor='#ffff66';"
    onmouseout="this.style.backgroundColor='#d4e3e5';">
<td>Item 1A</td><td>Item 1B</td><td>Item 1C</td>
</tr>
<tr onmouseover="this.style.backgroundColor='#ffff66';"
    onmouseout="this.style.backgroundColor='#d4e3e5';">
<td>Item 2A</td><td>Item 2B</td><td>Item 2C</td>
</tr>
<tr onmouseover="this.style.backgroundColor='#ffff66';"
    onmouseout="this.style.backgroundColor='#d4e3e5';">
<td>Item 3A</td><td>Item 3B</td><td>Item 3C</td>
</tr>
<tr onmouseover="this.style.backgroundColor='#ffff66';"
    onmouseout="this.style.backgroundColor='#d4e3e5';">
<td>Item 4A</td><td>Item 4B</td><td>Item 4C</td>
</tr>
<tr onmouseover="this.style.backgroundColor='#ffff66';"
    onmouseout="this.style.backgroundColor='#d4e3e5';">
<td>Item 5A</td><td>Item 5B</td><td>Item 5C</td>
</tr>
</table>
</body>
</HTML>
```

第 2 步,运行程序,结果如图 3-12 所示。

Info Header 1	Info Header 2	Info Header 3
Item 1A	Item 1B	Item 1C
Item 2A	Item 2B	Item 2C
Item 3A	Item 3B	Item 3C
Item 4A	Item 4B	Item 4C
Item 5A	Item 5B	Item 5C

图 3-12　表格样式结果

3.3.6　鼠标样式

在网页上，鼠标的形状可以表示浏览器的当前状态：平时呈箭头形、指向链接时成为手形、等待网页下载时成为沙漏型、链接指向一个帮助文件、链接也可以是向前进一页或是向后退一页等。CSS 提供了多种鼠标形状供选择，见表 3-5。

表 3-5　不同的鼠标形状

参　数	鼠 标 形 状	参　数	鼠 标 形 状
cursor:hand	手形	cursor:crosshair	十字形
cursor:wait	沙漏形	cursor:move	十字箭头形
cursor:e-resize	右箭头形	cursor:n-resize	上箭头形
cursor:text	文本形	cursor:w-resize	左箭头形
cursor:help	问号形	cursor:s-resize	下箭头形
cursor:nw-resize	左上箭头形	cursor:se-resize	右下箭头形
cursor:sw-resize	左下箭头形	cursor: no-drop	无法释放
cursor: auto;	自动	cursor:not-allowed	禁止
cursor:progress	处理中	cursor: url('#');	用户自定义(可用动画)

其中，# ＝光标文件地址，且文件格式必须为.cur 或.ani。

【语法】

```
style = "cursor:参数"
```

其中，参数见表 3-5。

【例 3-14】　光标样式。

第 1 步，为 CSSWebsites 项目添加一个 Web 窗体，命名为 Ex3-14.HTML，添加如下代码：

```
<HTML>
<head>
   <Meta http-equiv = "Content-Type" content = "text/HTML; charset = utf-8 " />
   <title>CSS cursor 属性示例</title>
   <style type = "text/CSS" media = "all">
  p#auto          { cursor: auto;}
  p#crosshair     { cursor: crosshair;}
  p#default       { cursor: default; }
  p#pointer       { cursor: pointer; }
  p#move          { cursor: move; }
  p#e-resize      { cursor: e-resize;}
  p#ne-resize     { cursor: ne-resize; }
  p#nw-resize     { cursor: nw-resize; }
  p#n-resize      { cursor: n-resize; }
  p#se-resize     { cursor: se-resize; }
  p#sw-resize     { cursor: sw-resize; }
  p#s-resize      { cursor: s-resize; }
  p#w-resize      { cursor: w-resize; }
  p#text          { cursor: text; }
```

```
    p#wait          { cursor: wait; }
    p#help          { cursor: help;}
    p#progress      { cursor: progress; }
    p               { border: 1px solid black;
    background: lightblue; }
    </style>
    </head>
  <body>
  <p id="auto">梦之都 XHTML 教程,auto 正常鼠标</p>
  <p id="crosshair">梦之都 CSS 教程,crosshair 十字鼠标</p>
  <p id="default">梦之都 XHTML 教程,default 默认鼠标</p>
  <p id="pointer">梦之都 CSS 教程,pointer 鼠标</p>
  <p id="move">梦之都 XHTML 教程,move 移动鼠标</p>
  <p id="e-resize">梦之都 CSS 教程,e-resize 鼠标</p>
  <p id="ne-resize">梦之都 XHTML 教程,ne-resize 鼠标</p>
  <p id="nw-resize">梦之都 CSS 教程,nw-resize 鼠标</p>
  <p id="n-resize">梦之都 XHTML 教程,n-resize 鼠标</p>
  <p id="se-resize">梦之都 CSS 教程,se-resize 鼠标</p>
  <p id="sw-resize">梦之都 XHTML 教程,sw-resize 鼠标</p>
  <p id="s-resize">梦之都 CSS 教程,s-resize 鼠标</p>
  <p id="w-resize">梦之都 XHTML 教程,w-resize 鼠标</p>
  <p id="text">梦之都 CSS 教程,text 文字鼠标</p>
  <p id="wait">梦之都 XHTML 教程,wait 等待鼠标</p>
  <p id="help">梦之都 CSS 教程,help 求助鼠标</p>
  <p id="progress">梦之都 XHTML 教程,progress 过程鼠标</p>
  </body>
  </HTML>
```

第 2 步,运行程序,结果如图 3-13 所示,光标放到对应区域会显示不同的形状。

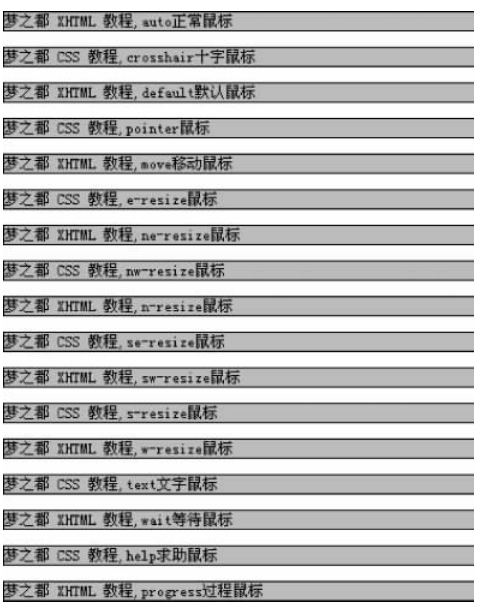

图 3-13　光标样式结果

3.3.7　滤镜样式

CSS 提供了一些内置的多媒体滤镜特效，使用这种技术可以把可视化的滤镜和转换效果添加到一个标准 HTML 元素上，如图片、文本容器，以及其他一些对象。CSS 的滤镜有 IE 浏览器支持，而 Firefox、Chrome 和 Opera 没有特别的滤镜，它们几乎完全支持 CSS 3。

1. alpha 滤镜

【语法】

```
{filter: alpha(opacity = 属性值 1,finishopacity = 属性值 2,style = 属性值 3,startx = 属性值 4,
starty = 属性值 5,finishx = 属性值 6,finishy = 属性值 7); }
```

作用：该滤镜能够使对象呈现渐变透明效果，效果由小括号中的各属性名及其对应的属性值决定。

参数：

opacity 属性用于设置不透明的程度，用百分比表示其属性值，大小从 0～100，0 表示完全透明，100 表示完全不透明。

finishopacity 属性是同 opacity 一起使用的一个选择性的参数，当设定了 opacity 和 finishopacity 时，可以制作出透明渐进的效果；其属性值也是 0～100，0 表示完全透明，100 表示完全不透明。

style 属性用于设置渐变风格，当同时设定了 opacity 和 finishopacity 产生透明渐进时，它主要用来制定渐进的显示形状：0 代表均匀渐进；1 代表线性渐进；2 代表放射渐进；3 代表长方形的直角渐进。

startx 属性用来设置水平方向渐进的起始位置。

finishx 属性用来设置水平方向渐进的结束位置。

finishy 属性用来设置竖直方向渐进的结束位置。

【例 3-15】　alpha 样式。

第 1 步，为 CSSWebsites 项目添加一个 Web 窗体，命名为 Ex3-15. HTML，添加如下代码：

```
<HTML>
   <head>
    <Meta http-equiv = "Content-Type" content = "text/HTML; charset = utf-8" />
    <title>alpha</title>
<style>
div{position: absolute; left: 50;top: 70; width:150; }
img{position: absolute;top:20;left:40;
filter:alpha(opacity = 0,finishopacity = 100,style = 1,startx = 0,starty = 85,finishx = 650,
finishy = 685)}
</style>
   </head>
<body>
<div>
<p style = "font-size:48;font-weight:bold;color:red;">
Beautiful </p>
</div>
```

```
<p><img src="images//keyan.jpg"></p>
</body>
 </HTML>
```

第 2 步，运行程序，结果如图 3-14 所示。

图 3-14　alpha 样式结果

2. blur 滤镜

【语法】

```
{filter: blur(add=属性值 1,direction=属性值 2,strength=属性值 3);}.
```

作用：该滤镜能够使对象表现为一种模糊的效果。

参数：

add 属性用来确定是否在运动模糊中使用原有目标，其属性值有 0 和 1 两种，0 表示在模糊运动中不使用原有目标，大多数情况下适用于图像；1 代表在模糊运动中使用原有目标，大多数情况下适用于文本。

direction 属性用来表示模糊移动时的角度，其属性值的范围为 0°～360°。

strength 属性用来表示模糊移动时的距离，该属性值可以任意设置。

【例 3-16】 blur 样式。

第 1 步，为 CSSWebsites 项目添加一个 Web 窗体，命名为 Ex3-16.HTML，添加如下代码：

```
<HTML>
   <head>
     <Meta http-equiv="Content-Type" content="text/HTML; charset=utf-8" />
     <title>blur</title>
<style>
.blur {
      filter: url(blur.svg#blur); <!--FireFox,Chrome,Opera-->
      -Webkit-filter: blur(10px); <!--Chrome,Opera-->
      -moz-filter: blur(10px);
      -ms-filter: blur(10px);
```

```
        filter: blur(10px);
        filter: progid:DXImageTransform .Microsoft.Blur(PixelRadius = 10,MakeShadow = false);
    <! -- IE6～IE9 -->}
     </style>
       </head>
    <body>
    <img src = "images//keyan.jpg" />
    <img src = "images//keyan.jpg" class = "blur" />
    </body>
     </HTML>
```

第 2 步，运行程序，结果如图 3-15 所示。

图 3-15　blur 样式结果

3. dropshadow 滤镜

【语法】

```
{filter: dropshadow(color = 属性值 1,offx = 属性值 2,offy = 属性值 3,positive = 属性值 4);}.
```

作用：滤镜用来产生图像的重叠效果，添加对象的阴影效果。dropshadow 属性对图像的支持不好，因为这种图像的颜色很丰富，很难找到一个投射阴影的位置。

参数：

color 属性用来设置投射阴影的颜色。

offx 属性值代表投影文字与原文字之间水平方向的偏移量。

offy 属性代表投影文字与原文字之间垂直方向上的偏移量。

positive 属性是一个布尔值(0 或者 1)，如果为 true(非 0)，那么就为任何非透明像素建立可见的投影；如果为 false(0)，那么就为透明的像素部分建立透明效果。

【例 3-17】 dropshadow 样式。

第 1 步，为 CSSWebsites 项目添加一个 Web 窗体，命名为 Ex3-17. HTML，添加如下代码：

```
<HTML>
 <head>
<title>dropshadow</title>
<style>
   div {position:absolute; top:20;width:300; font-family:matisse itc; font-size:64; font
-weight:bold; color: #CC00CC; filter: dropshadow (color = #ffccff, offx = 15, offy = 10,
positive = 1);}
   <!-- 定义 DIV 范围内的样式，绝对定位，投影的颜色为#FFCCFF,
   投影坐标为向右偏移 15 个像素，向下偏移 10 个像素 -->
   </style>
</head>
<body>
<div>Love Leaf</div>
</body>
 </HTML>
```

图 3-16 dropshadow 样式结果

第 2 步，运行程序，结果如图 3-16 所示。

4. glow 滤镜

【语法】

```
{filter: glow(color = 属性值 1,strength = 属性值 2);}.
```

作用：该滤镜能够在原对象周围产生一种类似发光的效果。

参数：

color 属性指定发光的颜色。

strength 则是发光强度的表现，也指光晕的厚度，其大小为 1～255。

【例 3-18】 glow 样式。

第 1 步，为 CSSWebsites 项目添加一个 Web 窗体，命名为 Ex3-18. HTML，添加如下代码：

```
<HTML>
<head>
<title>filter glow</title>
<Meta http-equiv = "Content-Type" content = "text/HTML; charset = utf-8" />
<style>
.leaf{position:absolute; top:20; width:400; filter: glow(color = #FF3399,strength = 15);}
.weny{position:absolute; top:70; left:50; width:00;filter:glow(color = #9966CC,strength =
10);}
p{font-family:bailey; font-size:48pt;font-weight:bold; color:#99CC66;}
</style>
  </head>
<body>
<div class = "leaf">
<p style = "ont-family:lucida handwriting; font-size:54pt;font-weight: bold; color:
#003366;">
Leaf Mylove</p>
</div>
```

```
<div class = "weny">
<p > Weny Good!</p >
</div >
</body >
</HTML >
```

第 2 步，运行程序，结果如图 3-17 所示。

图 3-17　glow 样式结果

5. chroma 滤镜

【语法】

```
{filter: chroma(color = 属性值 1);}
```

作用：该滤镜能够使图像中的某一颜色变为透明色。chroma 属性对于图片文件不是很适合，因为很多图片经过了减色和压缩处理（如 JPG、GIF 等格式），所以它们很少有固定的位置可以设置为透明。

参数：color 属性用来指定要变为透明色的颜色。通过该属性值的设定，可以过滤某图像中的指定颜色。

【例 3-19】 chroma 样式。

第 1 步，为 CSSWebsites 项目添加一个 Web 窗体，命名为 Ex3-19. HTML，添加如下代码：

```
<HTML >
<head >
  <Meta http - equiv = "Content - Type" content = "text/HTML; charset = utf - 8" />
<title > chroma filter </title >
<style > <! --
div{position:absolute;top:70;left:50;filter:chroma(color = green)}
p{font - family:bailey;font - size:68;font - weight:bold;color:green}
em{font - family:lucida handwriting italic;font - size:68;
   font - weight:bold;color:rgb(255,51,153)}
 --> </style >
</head >
<body > <div > <p > JUST <em > DO IT </em ></p >
</div >
</body >
</HTML >
```

第 2 步，运行程序，结果如图 3-18 所示。绿色的 JUST 字体不见了，实际上它是透明了，在 IE 浏览器下单击它所在的区域，还会显示出来，如图 3-19 所示。

图 3-18　chroma 样式结果 1

图 3-19　chroma 样式结果 2

6. wave 滤镜

【语法】

```
{filter: wave(add = 属性值 1, freq = 属性值 2, lightstrength = 属性值 3, phase = 属性值 4,
strength = 属性值 5);}.
```

作用：该滤镜能够使被过滤对象生成正弦波形，从而能造成一种变形幻觉。

参数：

add 属性是一个布尔值，用来决定是否将原始图像加入最后的效果中。

freq 属性是指波纹的频率，也就是指定在对象上一共需要产生多少个完整的波纹。

phase 属性用来设置正弦波的偏移量，它决定波形的形状，其属性值的取值范围为 0°～360°。

lightstrength 属性用来对波纹增强光影的效果，其取值范围为 0～100。

strength 属性用来决定波纹振幅的大小。

【例 3-20】 wave 样式。

第 1 步，为 CSSWebsites 项目添加一个 Web 窗体，命名为 Ex3-20. HTML，添加如下代码：

```
<HTML>
<head>
<Meta http-equiv = "Content-Type" content = "text/HTML; charset = utf-8" />
<title> wave CSS </title>
<style>
    .leaf{position:absolute;top:10;width:300; filter:wave(add = true, freq = 3, lightstrength
 = 100,
    phase = 45, strength = 20);}
    <! -- // * 设置 leaf 类的样式，绝对定位，wave 属性，产生 3 个波纹，光强为 100，波纹从 162°
(360 * 45 % )开始，振幅为 20 * // -->
</style>
</head>
<body>
<div class = "leaf">
<p style = "font-family:lucida handwriting;
           font-size:72pt; font-weight:bold;
           color:rgb(189,1,64);"> Leaf </p>
</div>
</body>
</HTML>
```

第 2 步，运行程序，结果如图 3-20 所示。

图 3-20　wave 样式结果

7. shadow 滤镜

【语法】

```
{filter: shadow(color = 属性值 1,direction = 属性值 2);}.
```

作用：该滤镜能够使对象产生一种阴影效果。

参数：

color 属性用来设置阴影的颜色。

direction 属性用来设置投影的方向，取值范围为 0°～360°，其中 0°代表垂直向上，然后每 45°为一个单位，该属性的默认值是向左的 270°。

【例 3-21】 shadow 样式。

第 1 步，为 CSSWebsites 项目添加一个 Web 窗体，命名为 Ex3-21. HTML，添加如下代码：

```
<HTML>
<head>
<Meta http - equiv = "Content - Type" content = "text/HTML; charset = utf - 8" />
<title> shadow CSS </title>
<style>
<! --
.shadow{position:absolute;top:100;left:80;filter:shadow(color = red,direction = 60);}
 -- >
</style>
</head>
<body>
<div class = "shadow">
<p style = "font - family:隶书,宋体;font - size:60pt;font - weight:bold;color:blue;">欢迎光临</P></div>
</body>
</HTML>
```

第 2 步，运行程序，结果如图 3-21 所示。

图 3-21　shadow 样式结果

8. mask 滤镜

【语法】

```
{filter: mask(color = 属性值);}
```

作用：该滤镜能够利用一个 HTML 对象在另一个对象上产生图像的遮罩，可以为对象建立一个覆盖于表面的膜，其效果就像戴着有色眼镜看物体一样。

参数：color 属性用来制定要被遮罩的颜色。

【例 3-22】 mask 样式。

第 1 步，为 CSSWebsites 项目添加一个 Web 窗体，命名为 Ex3-22. HTML，添加如下代码：

```
<HTML>
  <head>
 <Meta http - equiv = "Content - Type" content = "text/HTML; charset = utf - 8" />
 <title> mask filter </title>
 <style>
 <! -- //* 定义 DIV 区域的样式，绝对定位，mask 属性的 color 参数值指定用什么颜色遮住对象
```

```
* // -->
 div{position:absolute;top: 20;left: 40; filter:mask(color:#666699);}
 p{font-family:bailey;font-size:72pt; font-weight:bold; color:#FF9900:}
 </style>
 </head>
 <body>
 <div><p> wenyleaf </p></div>
</body>
</HTML>
```

第 2 步，运行程序，图 3-22(a)、(b)分别是添加 mask 样式前后的结果。

(a) 添加mask样式前　　(b) 添加mask样式后

图 3-22　mask 样式结果

9. light 滤镜

【语法】

```
{filter: light;}.
```

作用：该滤镜能够使 HTML 对象产生一种模拟光源的投射效果。light 可用的方法有：

AddAmbient，加入包围的光源。

AddCone，加入锥形光源。

AddPoint，加入点光源。

Changcolor，改变光的颜色。

Changstrength，改变光源的强度。

Clear，清除所有光源。

MoveLight，移动光源。

可以定光源的虚拟位置，以及通过调整 X 轴和 Y 轴的数值来控制光源焦点的位置，还可以调整光源的形式(点光源或者锥形光源)指定光源是否模糊边界、光源的颜色、亮度等属性。如果动态地设置光源，可能会产生一些意想不到的效果。

(1) addAmbient (iRed,iGreen,iBlue,iStrength)：为滤镜添加环境光。环境光是无方向的，并且均匀地洒在页面的表面。环境光有颜色和强度值，可以为对象添加更多的颜色。它通常和其他光一起使用，无返回值。参数如下。

iRed：必选项。整数值(Integer)，指定红色值，取值范围为 0～255。

iGreen：必选项。整数值(Integer)。指定绿色值，取值范围为 0～255。

iBlue：必选项。整数值(Integer)。指定蓝色值，取值范围为 0～255。

iStrength：必选项。整数值(Integer)。指定光强度，取值范围为 0～100。

(2) addCone(iX1,iY1,iZ1,iX2,iY2,iRed,iGreen,iBlue,iStrength,iSpread)：为滤镜

添加锥形光，以向对象的表面投射有方向的光束。光束会随延伸的距离而逐渐减弱，无返回值。参数如下。

iX1：必选项。整数值(Integer)。指定光源的左坐标值。

iY1：必选项。整数值(Integer)。指定光源的上坐标值。

iZ1：必选项。整数值(Integer)。指定光源的 Z 坐标值。

iX2：必选项。整数值(Integer)。指定光焦点的左坐标值。

iY2：必选项。整数值(Integer)。指定光焦点的上坐标值。

iRed：必选项。整数值(Integer)。指定红色值，取值范围为0～255。

iGreen：必选项。整数值(Integer)。指定绿色值，取值范围为0～255。

iBlue：必选项。整数值(Integer)。指定蓝色值，取值范围为0～255。

iStrength：必选项。整数值(Integer)。指定光强度，取值范围为0～100。

iSpread：必选项。整数值(Integer)。指定光源的虚拟位置与对象的表面之间的角度或张度，取值范围为0～90。

(3) addPoint (iX,iY,iZ,iRed,iGreen,iBlue,iStrength)，为滤镜添加点光源，无返回值，参数说明如下。

iX：必选项。整数值(Integer)。指定光源的左坐标值。

iY：必选项。整数值(Integer)。指定光源的上坐标值。

iZ：必选项。整数值(Integer)。指定光源的 Z 坐标值。

iRed：必选项。整数值(Integer)。指定红色值，取值范围为0～255。

iGreen：必选项。整数值(Integer)。指定绿色值，取值范围为0～255。

iBlue：必选项。整数值(Integer)。指定蓝色值，取值范围为0～255。

iStrength：必选项。整数值(Integer)。指定光强度，取值范围为0～100。

(4) changeStrength (iLightNumber,iStrength,fAbsolute)，改变光的强度。无返回值，参数说明如下。

iLightNumber：必选项。整数值(Integer)。指定光的标识符。

iStrength：必选项。整数值(Integer)。指定光强度，取值范围为0～100。

fAbsolute：必选项。布尔值(Boolean)。指定改变是替换当前设置的绝对值，还是加到当前设置的相对值。此参数不等于零表示采用绝对值，否则表示采用相对值。

(5) changeColor (iLightNumber,iRed,iGreen,iBlue,fAbsolute)：改变光的颜色。无返回值，参数说明如下。

iLightNumber：必选项。整数值(Integer)。指定光的标识符。

iRed：必选项。整数值(Integer)。指定红色值，取值范围为0～255。

iGreen：必选项。整数值(Integer)。指定绿色值，取值范围为0～255。

iBlue：必选项。整数值(Integer)。指定蓝色值，取值范围为0～255。

fAbsolute：必选项。布尔值(Boolean)。指定改变是替换当前设置的绝对值，还是加到当前设置的相对值。此参数不等于零表示采用绝对值，否则表示采用相对值。

(6) clear ()，清除所有与当前滤镜关联的光。无返回值。

移动锥形光的焦点或点光的原点。对于锥形光来说，此方法改变 x，y 目标坐标值；对于点光来说，此方法改变 x，y，z 源坐标值。此方法不作用于环境光。

(7) moveLight (iLightNumber,iX,iY,iZ,fAbsolute),移动光源。无返回值,参数说明如下。

iLightNumber:必选项。整数值(Integer)。指定光的标识符。

iX:必选项。整数值(Integer)。指定光源的左坐标值。

iY:必选项。整数值(Integer)。指定光源的上坐标值。

iZ:必选项。整数值(Integer)。指定光源的 Z 坐标值。

fAbsolute:必选项。布尔值(Boolean)。指定改变是替换当前设置的绝对值,还是加到当前设置的相对值。此参数等于 true 表示采用绝对值,等于 false 表示采用相对值。

【例 3-23】 light 滤镜样式。

第 1 步,为 CSSWebsites 项目添加一个 Web 窗体,命名为 Ex3-23.HTML,添加如下代码:

```
<HTML>
<head>
 <Meta http-equiv="Content-Type" content="text/HTML; charset=utf-8" />
<title>light CSS</title>
</head>
<body>
<img id="lightsy" src="images//c.jpg" width="400" height="260" style="filter: light
(enabled=1)" >
<script type="text/javascript">
 window.onload=setlights1 ;
 lightsy.onmousemove=mousehandler ;
 function setlights1(){
 lightsy.filters[0].addcone(380,-20,5,100,100,255,255,0,40,25); }
 function mousehandler(){
 x=(window.event.x-40);
 y=(window.event.y-40);
 lightsy.filters[0].movelight(0,x,y,5,1); }
</script>
</body>
</HTML>
```

第 2 步,运行程序,结果如图 3-23 所示。图 3-23(a)、(b)分别是初始页面和鼠标移动之后的效果。

(a) 初始页面　　(b) 鼠标移动后的页面

图 3-23　light 样式结果

10. gray 滤镜、invert 滤镜、xray 滤镜

【语法】

```
{filter: gray; }
{filter: invert;}
{filter: xray; }
```

作用：gray 滤镜能使一张彩色图片转变为灰色调图像。Invert 滤镜能使图像产生照片底片的效果。xray 滤镜能让对象反映出它的轮廓，并把这些轮廓加亮显示。这 3 个滤镜都没有附带参数。

【例 3-24】 gray、invert 和 xray 滤镜样式。

第 1 步，为 CSSWebsites 项目添加一个 Web 窗体，命名为 Ex3-24. HTML，添加如下代码：

```
<HTML>
<head><title>gray invert xray CSS</title>
</head>
<body>
<img src = "images//c.jpg" alt = gray width = 210 height = 130 style = "filter:gray">
<img src = "images//c.jpg" alt = invert width = 210 height = 130 style = "filter:invert">
<img src = "images//c.jpg" alt = xray width = 210 height = 130 style = "filter:xray">
</body>
</HTML>
```

第 2 步，运行程序，结果如图 3-24 所示，从左到右分别为 gray、invert 和 xray 滤镜的效果图。

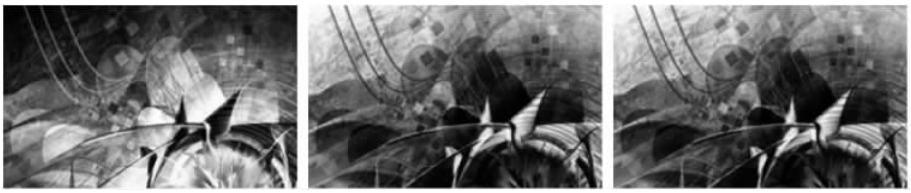

图 3-24　gray、invert 和 xray 滤镜的效果

3.4　CSS 页面布局

3.4.1　文本对齐

1. 文本横向排列(text-align)

文本水平对齐可以控制文本的水平对齐方式，而且不仅限于文字内容，也包括设置图片、影像资料的对齐方式。

【语法】

```
text - align: 参数.
```

参数的取值：

left，左对齐。

right，右对齐。

center，居中对齐。

justify，相对左右对齐。

需要注意的是，text-align 是块级属性，只能用在< p >，< blockquote >，< ul >，< h1 >～< h7 >等标识符里。

2. 文本纵向排列(vertical-align)

文本的垂直对齐应当是相对于文本母体(或父元素)的位置而言的，不是指文本在网页里垂直对齐。例如，表格的单元格里一段文本设置为垂直居中，文本将在单元格的正中显示，而不是整个网页的正中。垂直对齐属性只对行内元素有效。

【语法】

```
vertical - align:参数.
```

参数取值：

Top，把元素的顶端与行中最高元素的顶端对齐。

bottom，底对齐。

text-top，把元素的顶端与父元素字体的顶端对齐。

text-bottom，相对文本底对齐。

middle，中心对齐。

sub，以下标的形式显示。

super，以上标的形式显示。垂直对齐属性只对行内元素有效。

baseline，是 vertical-align 的默认值。元素放置在父元素的基线上。vertical-align：＋/－ n px元素相对于基线向下偏移 *n* 个像素。也可以使用百分比 vertical-align：＋/－n%，通过距离升高(正值)或降低(负值)元素，'0cm'等同于'baseline'。

例如，.test{vertical-align：-10%；}，假设这里的.test 的标签继承的行高是 20 像素，这里的 vertical-align：－10%代表的实际值是：－10%×20＝2(像素)。IE6/IE7 浏览器下的 vertical-align 的百分比值不支持小数 line-height，而 Firefox 3.6 等可以支持。

“行高”顾名思义指一行文字的高度，具体是指两行文字间基线之间的距离。基线是在英文字母中用到的一个概念，使用的四线格英语本子每行有四条线，如图 3-25 所示，其中底部第二条线就是基线，是 a，c，z，x 等字母的底边线。四线格从上到下对应的 vertical-align 的 4 个位置：顶线、中线、基线和底线，如图 3-26 所示。

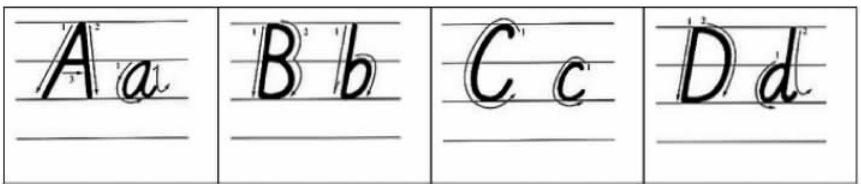

图 3-25　英文字母的四线格

不同浏览器由于兼容性问题，可能效果会不一样。知道了 vertical-align 垂直对齐的含

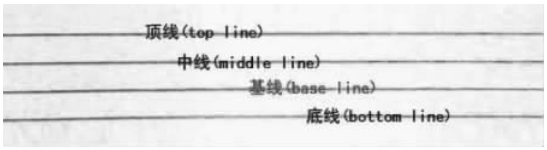

图 3-26　vertical-align 的 4 个位置

义，不少经验尚浅的同行会试着使用这个属性实现一些垂直方向上的对齐效果，会发现有时候可以，有时候又不起作用。因为 display 有很多属性值，其中 inline/inline-block/block 3 个最常见。而 vertical-align 是 inline-block 依赖型元素，只有一个元素属于 inline 或是 inline-block(table-cell 也可以理解为 inline-block 水平)水平，其身上的 vertical-align 属性才会起作用。所以，类似下面的代码就不会起作用。

```
span{vertical-align:middle;}
div{vertical-align:middle;}
```

图片、按钮、单复选框、单行/多行文本框等 HTML 控件，在默认情况下对 vertical-align 属性起作用。

【例 3-25】 vertical-align 垂直对齐样式。

第 1 步，为 CSSWebsites 项目添加一个 Web 窗体，命名为 Ex3-25.HTML，添加如下代码：

```
<HTML>
<head>
<style>
.box{background:black; color:white; padding-left:20px;}
.dot1{display:inline-block; width:4px; height:4px; background:white; vertical-align:
baseline;}
.dot2{display:inline-block; width:4px; height:4px; background:white; vertical-align:
middle;}
.dot3{display:inline-block; width:4px; height:4px; background:white; vertical-align:top;}
.dot4{display:inline-block; width:4px; height:4px; background:white; vertical-align:text
-top;}
.dot5{display:inline-block; width:4px; height:4px; background:white; vertical-align:text
-buttom;}
.dot6{display:inline-block; width:4px; height:4px; background:white; vertical-align:
buttom;}
</style>
</head>
<body>
<span class="box">
   <span class="dot1"></span>我是 baseline.
 </span>
<span class="box">
   <span class="dot2"></span>我是 middle.
 </span>
<span class="box">
   <span class="dot3"></span>我是 top.
```

```
 </span>
<span class = "box">
   <span class = "dot4"></span> 我是 text - top.
 </span>
<span class = "box">
   <span class = "dot5"></span> 我是 text - buttom.
 </span>
<span class = "box">
   <span class = "dot6"></span> 我是 buttom.
 </span>
</body>
</HTML>
```

第 2 步,运行程序,结果如图 3-27 所示。

图 3-27　vertical-align 垂直对齐结果

3. 文本缩进(text-indent)

文本缩进可以使文本在相对段默认值较窄的区域里显示,主要用于中文版式的首行缩进,或者将大段的引用文本和备注做成缩进的格式。

【语法】

```
text - indent 缩进距离
```

缩进距离取值:带长度单位的数字;比例关系。

需注意的是,text-indent 也是块级属性,只能用在< p >,< blockquote >,< ul >,< h1 >～< h7 >等标识符里。

【例 3-26】 text-indent 样式。

第 1 步,为 CSSWebsites 项目添加一个 Web 窗体,命名为 Ex3-26. HTML,添加如下代码:

```
<HTML>
<head>
<style type = "text/CSS">
p {text - indent: 1cm}
</style>
</head>
<body>
<p>
```

文本缩进可以使文本在相对段默认值较窄的区域类显示,主要用于中文版式的首行缩进,或者将大段的引用文本和备注做成缩进的格式。

```
</p>
</body>
</HTML>
```

第 2 步,运行程序,结果如图 3-28 所示。

文本缩进可以使文本在相对段默认值较窄的区域类显示，主要用于中文版式的首行缩进，或者将大段的引用文本和备注做成缩进的格式。

图 3-28　缩进结果

3.4.2　盒子模型

CSS 中所有页面元素都包含在一个矩形框内，这个矩形框称为盒子。盒子描述了元素及属性在页面布局中所占空间的大小，因此盒子可以影响其他元素的位置及大小。

盒子模型用于设置元素的边界、边界补白、边框等属性值，使用这一属性的大多是块元素。W3C 组织建议把所有网页上的对象都放在一个盒子(box)中，设计师可以通过创建定义来控制这个盒子的属性，这些对象包括段落、列表、标题、图片以及层。盒子模型主要定义四个区域：内容(content)、页边距或补白(padding)、边界(border)和外边距(margin)。content 是盒子模型中必需的部分，可以是文字、图片等元素。padding 也称页边距或补白，用来设置内容和边框之间的距离。border 可以设置内容边框线的粗细、颜色和样式等。margin 是边距，用来设置内容与内容之间的距离。margin，padding，content，border 之间的层次、关系和相互影响的盒子模型如图 3-29 所示。

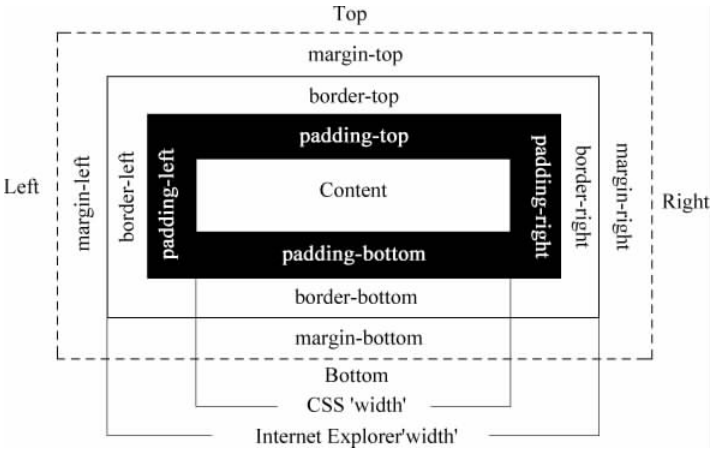

图 3-29　方框盒子模型

(1) margin：包括 margin-top，margin-right，margin-bottom，margin-left，控制块级元素之间的距离，它们是透明不可见的。对于上、右、下、左，margin 值均为 40 像素，因此代码为：

```
margin - top: 40px;margin - right: 40px;margin - bottom: 40px;
margin - left: 40px;
```

根据上、右、下、左的顺时针规则，简写为 margin：40px 40px 40px 40px；为便于记忆，请参考图 3-30 所示的顺序书写。

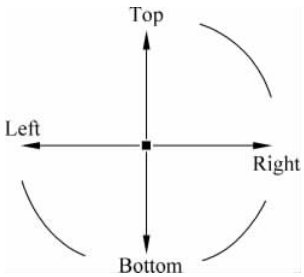

图 3-30　margin 四个位置的书写顺序

当上、下，左、右 margin 值分别一致时，可简写为：

```
margin: 40px 50px;
```

40px 代表上、下 margin 值，50px 代表左、右 margin 值。

当上、下、左、右 margin 值均一致时，可简写为：

```
margin: 40px;
```

当上、下 margin 值不一致，而左、右 margin 值均一致时，可简写为：

```
margin: 40px 50px 20px;
```

40px，20px 代表上、下 margin 值，50px 代表左、右 margin 值。

由此可见，如果只提供一个数，将用于全部的四条边；如果提供两个数，第一个用于上、下，第二个用于左、右；如果提供三个数，第一个用于上，第二个用于左、右，第三个用于下；如果提供四个参数值，将按上、右、下、左的顺序作用于四边。

margin 不会在绝对元素上折叠。假设有一个 margin-bottom 值为 20px 的段落，段落后面是一个具有 30px 的 margin-top 的图片，那么，段落和图片之间的空间不会是 50px (20px＋30px)，而是 30px(30px ＞ 20px)。这就是所谓的 margin-collapse，两个 margin 会合并(折叠)成一个 margin。

绝对定位元素不会像那样进行 margin 的折叠。这会使它们与预期的不一样。

【例 3-27】 margin 的样式示例。

第 1 步，为 CSSWebsites 项目添加一个 Web 窗体，命名为 Ex3-27. HTML，添加如下代码：

```
<HTML>
<head>
<style type = "text/CSS">
#ID1 {
background - color: #333;
color:#FFF;
margin:10px;
}
#ID2 {
font: normal 14px/1.5 Verdana,sans - serif;
margin:30px;
border: 1px solid #F00;
}
</style>
</head>
<body>
<div id = "ID1">
 Hello,world
<h1 id = "ID2"> Margins of ID1 and ID2 collapse vertically.<br/>
元素 ID1 与 ID2 的 margins 在垂直方向折叠.</h1>
</div>
</body>
</HTML>
```

第 2 步，运行程序，结果如图 3-31 所示。

图 3-31　margin 样式结果

(2) padding:包括 padding-top、padding-right、padding-bottom、padding-left,控制块级元素内部 content 与 border 之间的距离。简写请参考 margin 属性的写法,如

```
body { padding: 36px;}                    //对象四边的补丁边距均为 36px
body { padding: 36px 24px; }              //上、下两边的补丁边距为 36px,左、右两边的补丁边距
                                          //为 24px
body { padding: 36px 24px 18px; }         //上、下两边的补丁边距分别为 36px、18px,左、右两边的补
                                          //丁边距为 24px
body { padding: 36px 24px 18px 12px; }    //上、右、下、左补丁边距分别为 36px、24px、18px、12px
```

(3) border:简写属性在一个声明中设置所有的边框属性。可以按顺序设置如下属性:border-width,border-style,border-color。如果不设置其中的某个值,也不会出问题,例如 border:solid #ff0000; 也是允许的。

border-width 简写属性为元素的所有边框设置宽度,或者单独地为各边边框设置宽度。只有当边框样式不是 none 时才起作用。如果边框样式是 none,边框宽度实际上会重置为 0。不允许指定负长度值。可能的 border-width 取值及其对应的描述如下。

thin,定义细的边框。

medium,默认值,定义中等的边框。

thick,定义粗的边框。

length,允许自定义边框的宽度。

border-style 属性用于设置元素所有边框的样式,或者单独为各边设置边框样式。只有当这个值不是 none 时,边框才可能出现。可能的 border- style 取值及其对应的描述如下。

none,定义无边框。

hidden,与 none 相同。不过,应用于表时除外,对于表,hidden 用于解决边框冲突。

dotted,定义点状边框。在大多数浏览器中呈现为实线。

dashed,定义虚线。在大多数浏览器中呈现为实线。

solid,定义实线。

double,定义双线。双线的宽度等于 border-width 的值。

groove,定义 3D 凹槽边框。其效果取决于 border-color 的值。

ridge,定义 3D 垄状边框,其效果取决于 border-color 的值。

inset,定义 3D inset 边框,其效果取决于 border-color 的值。

outset,定义 3D outset 边框,其效果取决于 border-color 的值。

最不可预测的边框样式是 double。它定义为两条线的宽度再加上这两条线之间的空间,等于 border-width 值。

border-color 属性设置四条边框的颜色。此属性可设置 1～4 种颜色。border-color 属性是一个简写属性,可设置一个元素的所有边框中可见部分的颜色,或者为 4 个边分别设置不同的颜色。简写请参考 margin 属性的写法。

```
border-color:red green blue pink;
```

表示上边框是红色，右边框是绿色，下边框是蓝色，左边框是粉色。

```
border-color:red green blue;
```

表示上边框是红色，右边框和左边框是绿色，下边框是蓝色。

```
border-color:dotted red green;
```

表示上边框和下边框是红色，右边框和左边框是绿色。

```
border-color:red;
```

表示所有 4 个边框都是红色。

可能的 border- color 取值及其对应的描述如下。

color_name，规定颜色值为颜色名称的边框颜色，如 red。

hex_number，规定颜色值为十六进制值的边框颜色，如 #ff0000。

rgb_number，规定颜色值为 rgb 代码的边框颜色，如 rgb(255,0,0)。

transparent，默认值。边框颜色为透明。

【例 3-28】 padding 和 border 样式。

第 1 步，为 CSSWebsites 项目添加一个 Web 窗体，命名为 Ex3-28. HTML，添加如下代码：

```
<HTML>
<head>
<style type = "text/CSS">
#ID1 {
background - color: #333;
color:#FFF;
margin:10px;
padding:15px;
}
#ID2 {
font: normal 14px/1.5 Verdana,sans - serif;
margin:30px;
padding:15px;
border: 1px groove red blue green;
}
</style>
</head>
<body>
<div id = "ID1">
 Hello,world
<h1 id = "ID2"> Margins of ID1 and ID2 collapse vertically.<br/>
元素 ID1 与 ID2 的 margins 在垂直方向折叠.</h1>
</div>
</body>
</HTML>
```

第 2 步，运行程序，结果如图 3-32 所示。

图 3-32　padding 和 border 样式结果

3.4.3　文字环绕 float 样式

在传统的印刷布局中,文本可以按照需要围绕图片。一般把这种方式称为“文本环绕”。在网页设计中,应用了 CSS 的 float 属性的页面元素就像在印刷布局里面的被文字包围的图片一样。

【语法】

```
{float: none | left |right}
```

参数值:

none:对象不浮动。

left:对象浮在左边。

right:对象浮在右边。

【例 3-29】　float 样式示例。

第 1 步,为 CSSWebsites 项目添加一个 Web 窗体,命名为 Ex3-29.HTML,添加如下代码:

```
<HTML>
<head>
<Meta http-equiv="Content-type" content="text/HTML; charset=utf-8" />
<link rel="stylesheet" type="text/CSS" href="main.CSS" />
<title>CSS FLOAT</title>
<style type="text/CSS">
.top {
     width:500px;            /*div 框的宽度*/
     background:#f1f1f1;     /*div 框的背景色*/
}
.img {
     float:left;             /*图片向左浮动*/
     margin-right:10px;      /*图片右侧与文字的边距*/
     margin-bottom:5px;      /*图片下部与文字的边距*/
border:thin dotted red;
}
</style>
</head>
<body>
<!--环绕的图片及文字,图片的 CSS 类为 img-->
<div class="top">
<img src="images/c.jpg" alt="文字环绕" class="img" />
```

盒子模型主要定义四个区域：内容(content)、页边距(padding)、边界(border)和边距(margin)。初学者经常搞不清楚 margin，background-color，background- image，padding，content，border 之间的层次、关系和相互影响。这里提供一张盒子模型的 3D 示意图，便于理解和记忆。

```
</div>
</body>
</HTML>
```

第 2 步，运行程序，结果如图 3-33 所示。

图 3-33　float 样式结果

注意：不能在同一个属性中应用定位属性和浮动。因为对使用什么样的定位方案来说，两者的指令是相冲突的。如果把两个属性都添加到一个相同的元素上，那么 CSS 取最后设置的那个属性。

3.4.4　元素定位

CSS 提供 position、top、left 和 z-index 属性，用于在二维或三维空间定位某个元素相对于其他元素的相对位置或绝对位置。

1. position 元素位置模式

position 属性用于设置元素位置的模式。当 position 为 absolute 时，top 和 left 属性分别用于设置元素与窗口或框架上端以及左端的距离；当 position 为 relative 时，top 和 left 属性分别用于设置元素与父元素上端以及左端的距离。

定位模式规定了一个盒子在总体的布局上应该处于什么位置，以及对周围的盒子会有什么影响。定位模式包括了常规文档流、浮动和 5 种类型的 position 定位的元素。

【语法】

```
HTML 标签 { position: absolute | relative | fixed | static| inherit }
```

static 是 position 默认的属性值。任何应用了 position：static 的元素都处于常规文档流中。它处于什么位置，以及它如何影响周边的元素都是由盒子模型决定的。一个 static 定位的元素会忽略所有 top，right，bottom，left 以及 z-index 属性声明的值。为了元素能使用这些属性，需要先为它的 position 属性应用这 3 个值的其中之一：absolute、relative、fixed。

absolute，绝对定位的元素会从常规文档流中脱离。对于包围它的元素而言，它会将该绝对定位元素视为不存在。如果需要保持它所占有的位置而不被其他元素所填充，那么需要使用其他的定位方式。可以通过 top，right，bottom 和 left 四个属性来设置绝对定位元素的位置。但通常只会设置它们其中的两个：top 或者 bottom，以及 left 或者 right。默认它们的值都为 auto。

绝对定位的关键是起点在哪里。如果 top 被设置为 20px，那么这 20px 是从哪里开始计算的。一个绝对定位的元素的起点位置是相对于它的第一个 position 值不为 static 的父元素而言的。如果在它的父元素链上没有满足条件的父元素，那么绝对定位元素则会相对于文档窗口进行定位。在一个元素的样式上设置 position：absolute 意味着需要考虑父元素，并且如果父元素的 position 值不为 static，那么绝对定位元素的起点为父元素的左上角位置。如果父元素没有应用除了 static 以外的 position 定位，那么它会检查父元素的父元素是否应用非 static 定位。如果该元素应用了定位，那么它的左上角便会成为绝对元素的起点位置。如果没有，则会继续向上遍历 DOM，直到找到一个定位元素或者寻找失败，以到达最外层的浏览器窗口。

relative，相对定位的元素也是根据 top，right，bottom 和 left 四个属性来决定自己的位置的。但只是相对于它们原来所处的位置进行移动。从某种意义上来说，为元素设置相对定位和为元素添加 margin 有点相似，但也有一个重要的区别。区别就是围绕在相对定位元素附近的元素会忽略相对定位元素的移动。相对定位元素离开了正常文档流，但仍然影响着围绕它的元素。这些元素觉得相对定位元素仍然在正常文档流中。

fixed，固定定位的行为类似于绝对定位，但也有一些不同的地方。第一个不同点，固定定位总是相对于浏览器窗口进行定位，并且通过 top，right，bottom 和 left 属性决定其位置，它抛弃了它的父元素。第二个不同点是继承性，固定定位的元素是固定的。它们并不随页面的滚动而移动。

2. z-index 空间中定位元素

三维空间中定位元素的属性是 z-index，打破了二维平面的约束，具有宽度和高度。z-index 属性设置元素的堆叠顺序。拥有更高堆叠顺序的元素总会处于堆叠顺序较低的元素的前面，如图 3-34 所示。该属性设置一个定位元素沿 Z 轴的位置，Z 轴定义为垂直延伸到显示区的轴。如果为正数，离用户更近，为负数则表示离用户更远。

【例 3-30】 z-index 样式示例。

第 1 步，为 CSSWebsites 项目添加一个 Web 窗体，命名为 Ex3-30.HTML，添加如下代码：

```
<HTML>
<head>
<style type="text/CSS">
img
{
position:absolute;
left:0px;
top:0px;
z-index:-1;
}
```

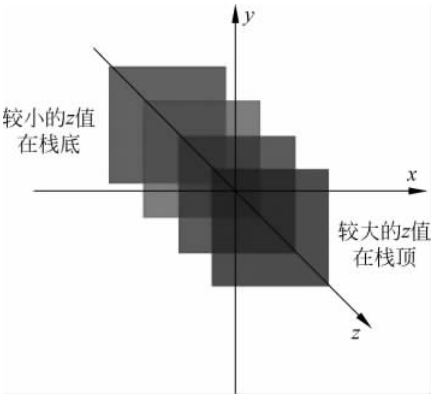

图 3-34　z-index 的堆叠顺序

```
</style>
</head>
<body>
<h1 style="color:red">This is a heading</h1>
<img src="images/c.jpg" />
<p style="color:red">由于图像的 z-index 是 -1,因此它在文本的后面出现.</p>
</body>
</HTML>
```

第 2 步,运行程序,结果如图 3-35 所示。

图 3-35　z-index 样式结果

由图 3-35 可知,z-index 高的位于 z-index 低的上面,并朝页面的上方运动。相反,一个低的 z-index 在高的 z-index 的下面,并朝页面下方运动。所有元素默认的 z-index 值都为 0,并且可以对 z-index 使用负值。

假如只是开发简单的弹窗效果,懂得通过 z-index 来调整元素间的层叠关系就够了。但要将多个弹窗间层叠关系处理好,那么充分理解 z-index 背后的原理及兼容性问题就是必要的知识储备了。常接触到的 z-index 只是分层显示中的一个属性,而理解 z-index 背后的原理实质上就是要理解分层显示原理。

3.5　习　　题

1. 简要说明什么是 CSS。
2. 比较几种网页添加样式的方法。
3. 比较字体样式和文本样式的区别。
4. 设计一个表格样式，其中，整体表格的样式：

```
font-family:"Trebuchet MS",Arial,Helvetica,sans-serif;width:100%;
border-collapse:collapse;
```

表格标题的样式：

```
font-size:1.1em;  text-align:left;  padding-top:5px;  padding-bottom:4px;  background
-color:#A7C942;  color: #ffffff;
```

表格奇数行数据的样式：

```
color:#000000;  background-color:#EAF2D3;
```

表格偶数行数据的样式：

```
font-size:1em;  border:1px solid #98bf21;  padding:3px 7px 2px 7px;
```

效果如图 3-36 所示。

Company	Contact	Country
Apple	Steven Jobs	USA
Baidu	Li YanHong	China
Google	Larry Page	USA
Lenovo	Liu Chuanzhi	China
Microsoft	Bill Gates	USA
Nokia	Stephen Elop	Finland

图 3-36　表格 CSS 样式结果

5. 补全下面的代码，实现一个< div >元素页面布局样式，效果如图 3-37 所示。

```
  <!DOCTYPE HTML>
<HTML lang="en" xmlns="http://www.w3.org/1999/xHTML">
<head>
    <Meta charset="utf-8" />
    <title></title>
    <style>
        #header {
            background-color: black;
            color: white;
        (1)
        ____________________________
            padding: 5px;
        }
```

```
        #nav {
            line-height: 30px;
            background-color: #eeeeee;
            height: 300px;
            width: 100px;
        (2)
    ____________________________________
            padding: 5px;
        }

        #section {
            width: 350px;
        (3)
    ____________________________________
            padding: 10px;
        }

        #footer {
            background-color: black;
            color: white;
            clear: both;
        (4)
    ____________________________________
            padding: 5px;
        }
    </style>
</head>
<body>
        <div id="header">
            <h1>City Gallery</h1>
        </div>

        <div id="nav">
            London<br>
            Paris<br>
            Tokyo<br>
        </div>

        <div id="section">
            <h1>London</h1>
            <p>
    London is the capital city of England. It is the most populous city in the United Kingdom,
                with a metropolitan area of over 13 million inhabitants.
            </p>
            <p>
    Standing on the River Thames, London has been a major settlement for two millennia, its
history going back to its founding by the Romans, who named it Londinium.
            </p>
        </div>
        <div id="footer">
            Copyright W3School.com.cn
```

```
    </div>
</body>
</HTML>
```

图 3-37　页面布局效果

6．分析一个经典网站的 CSS。