

CSS 快速入门

CSS 是 Cascading Style Sheets（层叠样式表单）的简称。更多的人把它称作样式表。顾名思义，它是一种设计网页样式的工具。借助 CSS 的强大功能，网页将在您丰富的想象力下千变万化。

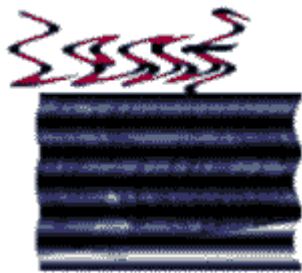


图 1



图 2

看到上面的两幅图片，您可能会认为这是用 photoshop 或者是其他图形处理软件制作的吧。可是上面的例子却完全是用 CSS 编写的。我们来看一下它的源代码（不要以为讲代码您会搞不懂，很简单的^_^）：

```
<html>
  <head>
    <title>css word</title>
    <style>/**CSS 样式定义开始*/
    <!--
    div
    {width: 200; filter
blur(add=true,direction=135,strenght=20);
    } /**定义 DIV 范围内的样式，包括宽度(width)、模糊滤镜属性(blur)*/
-->
    </style>/**样式定义结束*/
  </head>
  <body>
    <div>/**以下的区域内采用<head>中<style>里面定义的格式*/
    <p style="font-size: 48;font-style: bold;
color:red;">hongen </p>
    /**定义字体样式(font-style)，包括字体大小(size)、粗细(bold)、颜色
```

```
(color)*//  
</div>  
</body>  
</html>
```

上面的代码中以红色显示的是注释部分，用于帮助您更好的理解代码的含义。上面的这一段代码产生的是如图 2 的效果。而其中字体的模糊效果就是运用了 CSS 的滤镜功能。滤镜功能是 CSS 非常有趣也非常吸引人的一个功能，我们将在后面的章节中介绍。

实际上 CSS 的代码都是由一些最基本的语句构成的。它的基本语句的结构是这样的：

选择符{属性：属性值}

看，是不是很简单，只有三部分。

我们结合上面的代码来讲解，比如上面代码的<head>...</head>中，使用<style>来指定样式。

一般说来，<style>下面的 CSS 语句是以注释语句的形式书写的，也就是上面代码中的<!-- ... -->符号包含的部分。所以，上面的例子中定义页面样式的语句是：

```
div{width: 200;filter: blur(add=true,direction=35,strenght=20);}
```

在上面的语句中，DIV 是选择符，选择符可以是 HTML 中任何的标识符，比如 P、DIV、IMG 甚至 BODY 都可以作为选择符。

这里用 DIV 做选择符，就是说在 HTML 中，编辑在<DIV>中的页面格式将以上面语句中大括号内定义的格式显示。

括号内的 WIDTH 和 FILTER 就是属性。

WIDTH 定义了 DIV 区域内的页面的宽度，200 是属性值。

FILTER 定义了滤镜属性，BLUR 是它的属性值，该属性值产生的是一种模糊效果，其小括号内定义的是 BLUR 属性值的一些参数。

ADD 参数有两个值：True 和 False。分别指定图片是否被设置成模糊效果。

Direction 参数是用来设置模糊的方向。0 度代表垂直向上，然后每 45 度一个单位，例子中的 135 代表底部向右 135 度，每一个度数单位都代表一个模糊方向，如果您感兴趣的话，可以参照后面的讲解中详细的参数设定。

Strenght 代表有多少像素的宽度将受到模糊影响，参数值是用整数来设置的。

我们看到除了在<HEAD>中有 CSS 的定义，在<BODY>中也有一段 CSS 定义：

```
<p style="font-size: 48;font-style: bold;color: red;">hongen</p>
```

这里 Style 是内嵌到<P>中来定义该段落内的格式的。我们发现在<BODY>中的 CSS 语句与定义在<HEAD>中还有些不同，它是用<STYLE =>直接定义的。这种定义方法非常适用于编写的代码比较多的情况。

而上面的代码的 CSS 定义格式则非常适用于代码较少、结构较简单的情况。

按照 CSS 语句的基本格式，我们可以看出上面定义 P 段落内的 CSS 代码中 font-size、

font-style 和 color 是属性，分别定义<P>中“hongen”字体的大小(size)、样式(style)和颜色(color)；

而 48、bold、red 是属性值。意思是“hongen”将以 48pt、粗体、红色的样式显示。由于整个段落是定义在<DIV>中的，所以 hongen 又将显示出<head>中定义的滤镜属性来。最终的结果就如同图 2 一样。

图 1 中的效果也是用很简单的 CSS 语句实现的，只是运用了 CSS 中的滤镜属性而已。

通过上面的讲解，我们可以看到用很简单的 CSS 语句就可以实现许多需要专业软件才可以达到的效果。利用属性可以设置字体、颜色、背景等页面格式；利用定位可以使页面布局更加规范、好看；利用滤镜可以使页面产生多媒体效果。怎么样，CSS 的功能是不是很强大，在下面的教程中我们将对它进行详细的讲解。

怎样编写 CSS?

从上面的例子中，我们可以看到 CSS 的语句是内嵌在 HTML 文档内的。所以，编写 CSS 的方法和编写 HTML 文档的方法是一样的。

您可以用任何一种文本编辑工具来编写。比如 Windows 下的记事本和写字板、专门的 HTML 文本编辑工具（Frontpage、Ultraedit 等），都可以用来编辑 CSS 文档。

那么您可能会问，独立编辑好的 CSS 文档怎样加入到 HTML 文档中呢？其实在第一章中的例子里已经介绍了两种方法。

一种是把 CSS 文档放到<head>文档中：

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

其中<style>中的“type=‘text/css’”的意思是<style>中的代码是定义样式表单的。

另一种方法是把 CSS 样式表写在 HTML 的行内，比如下面的代码：

```
<p style="font-size: 14pt; color: blue">蓝色 14 号文字</p>
```

这是采用<Style=“ ”>的格式把样式写在 html 中的任意行内，这样比较方便灵活。

还有一种方法是：把您编辑好的 CSS 文档保存成“.CSS”文件，然后在<head>中定义。定义的格式是这样的：

```
<head> <link rel=stylesheet href="style.css"> ..... </head>
```

我们看到这里应用了一个<Link>，“rel=stylesheet”指连接的元素是一个样式表（stylesheet）文档。一般这里是不需要您改动的。

而后面的“href=‘style.css’”指的是需要连接的文件地址。您只需把编辑好的“.CSS”文件的详细路径名写进去就可以了。这种方法非常适宜同时定义多个文档。它能使多个文档同时使用相同的样式，从而减少了大量的冗余代码。

本章主要介绍了编写 CSS 的方法以及如何把作好的 CSS 文档与 HTML 结合起来运用。您可以根据自己编写的 HTML 代码的结构、长度来选择用哪一种方法将 CSS 与之结合。

下一章我们将进入 CSS 的具体知识的学习，我们将先从最基本的属性开始讲起。

从 CSS 的基本语句就可以看出，属性是 CSS 非常重要的部分。熟练掌握了 CSS 的各种属性将会是您编辑页面更加得心应手。下面我们就借助一些实例来讲解。

三、CSS 属性

1、 字体属性

这是最基本的属性，您经常都会用到。它主要包括以下这些属性：

属 性	属 性 含 义	属 性 值
font-family	使用什么字体	所有的字体
font-style	字体是否斜体	Normal、 italic、 oblique
font-variant	字体是否用小体大写	Normal、 small-caps
font-weight	定义字体的粗细	Normal、 bold、 bolder、 lighter等
font-size	定义字体的大小	Absolute-size、 relative-size、 length、 percentage等

光看表格是不是有些难懂，不要紧，我们来看一个用表格中的属性定义的字体样式（如下图）：

hongen

hongen

hongen

HONGEN

第一行只是采用了“font-family”属性，代码是这样的：
`<p style="font-family: lucida console">hongen</p>`
这行代码定义了“hongen”将以“lucida console”的字体显示；
第二行 font-family 为默认值，而是定义了“font-style”属性为“italic”斜体，代码如下：

```
<p style="font-style: italic">hongen</p>
```

第三行 font-family 为默认值，font-style 为“oblique”，另外还定义了“font-weight”属性为“bold”，“font-size”属性为 24pt（缺省情况下为 absolute size 绝对大小）代码如下：

```
<p style="font-style: oblique; font-weight: bold; font-size: 24pt">hongen</p>
```

最后一行只是在第三行定义的基础上又增添了“font-variant”属性为“small-caps”（小体大写），代码如下：

```
<p style="font-style: oblique; font-weight: bold; font-size: 24pt; font-variant: small-caps">hongen</p>
```

值得注意的是：“font-variant”属性值“small-caps”在指定的小体大写字体不存在的时候，CSS 将用大写字母代替所有的小写字母。

可能您会问：定义一个字体怎么有那么长的代码呢？其实字体属性中还有一个“font”属性，可以实现简便定义。我们来看下面这段代码：

```
<p style="font: italic small-caps bold 36pt,GlitzyCurl">Leaf</p>
```

此代码产生的效果是这样的（如下图）：



这段代码定义了“leaf”的字体为 GlitzyCurl，并且是斜体、小体大写、粗体、36 号字。这段代码等同于：

```
<p style="font-style: italic; font-variant: small-caps; font-weight: bold; font-size: 36pt; font-family: GlitzyCurl">Leaf</p>
```

怎么样，是不是精简了许多。

但需要注意的是：如果您用属性直接定义，一定要注意属性值的排放顺序。它的排放规则是按照“font-style”、“font-variant”、“font-weight”、“font-size”、“font-family”的顺序，其中没有定义的以默认值显示。

其实我们习惯上还是不用 FONT 属性来统一定义，因为分开来定义的代码虽然很长，但是看上去很直观，让人一看就知道定义的内容是什么。

说一点遗憾之处吧，就是 CSS 对中文字体的支持可能不是很好。

2、颜色和背景属性

您总不希望自己的页面处于黑白世界吧。如果给用属性定义的文本添上颜色是不是会更好呢？

下面我们就讲一下 CSS 的颜色和背景属性。先来看一下 CSS 下的颜色和背景都有哪些属性吧（如下表）：

属性	属性含义	属性书写格式	属性值
Color	定义前景色	例:p {color: red}	颜色
Background-color	定义背景色	例:body {background-color:yellow}	颜色
Background-image	定义背景图案	例:body {background-image:url (.jpg) }	图片路径
Background-repeat	背景图案重复方式	例:body {background-repeat:repeat-y}	Repeat-x、repeat-y、no-repeat
Background-attachment	设置滚动	例:body {background-attachment:scroll}	Scroll Fixed
Background-position	背景图案的初始位置	例:body {background:url(.jpg)top center}	Percentage、length、top、left、right、bottom等

我们还是举个例子吧。比如上一节讲到的“LEAF”字体，我们在其中加入颜色属性，在<style>再加入以下代码：

```
color: rgb (204, 82, 28) ;
background-img: url(ss01038.jpg);
background-repeat: no-repeat; margin-right: 15em
```

/*设置前景色的 rgb 值为(204, 82, 28)，您也可以用颜色的英文名称直接定义；
设置背景图案的地址；设置背景图案不重复显示；另外还设置了页边距 margin 属性
，这个我们以后再讲*/

加入代码后“Leaf”的字体显示效果是这样的（如下图）：



怎么样，是不是比光是定义属性显得更加漂亮了。当然您也可以简便的定义 background 属性（象定义 font 属性那样），下面我告诉一下您 background 属性的书写格式：

属性：background 属性值：<background-color>||<background-image>||
<background-repeat>||<background-attacement>||<background-position>

也就是说，使用 `background` 属性可以一次定义前面讲到的一切有关背景的属性，包括背景色、背景图案等等。因此，上面定义“leaf”颜色属性中有关背景的代码可以写成这样：“`background: url(ss01038.jpg) no-repeat`”

好了，至于颜色和背景属性的功能，您自己也可以去试一试。下一节我们将介绍 CSS 的文本属性。

3、文本属性

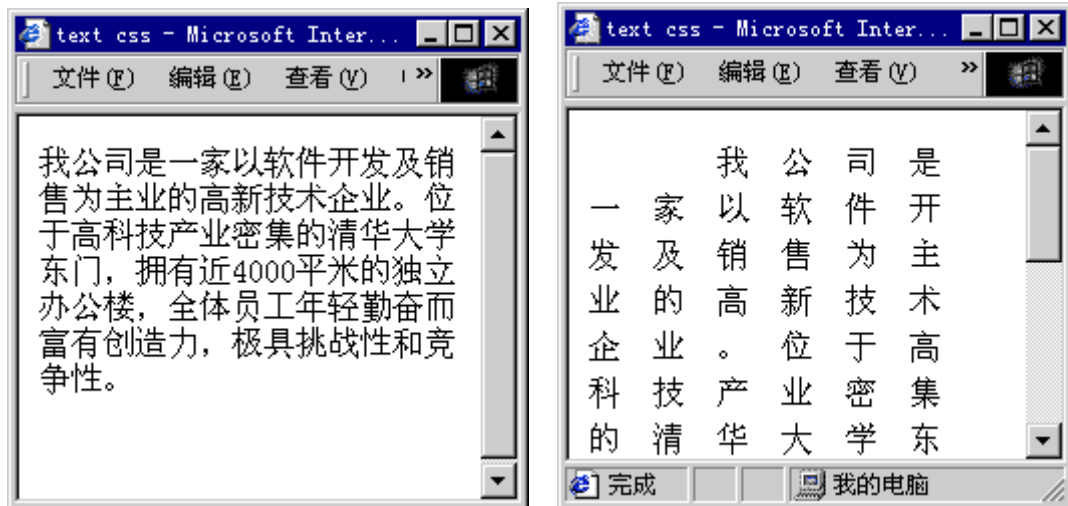
※定义间距

前面说了如何用 CSS 定义字体、颜色和背景属性，那么定义好的文本我们怎么对它进行排版呢？

下面我们讲述一下 CSS 的文本属性，还是先来看一下文本属性的详细列表：

属性	属性含义	属性值
Word-spacing	定义了各个单词之间的间距	Normal <length>（必须以长度为单位）
Letter-spacing	定义了每个字母之间的间距	同上
Text-decoration	定义文字的“装饰”样式	None underline overline line-through blink
Vertical-align	定义了元素在垂直方向上的位置	Baseline sub super top text-top middle bottom text-bottom <percentage>
Text-transform	使文本转换为其他形式	Capitalize uppercase lowercase none
Text-align	定义了文字的对齐方式	Left right center justify
Text-indent	定义文本的首行的缩进方式	<length> <percentage>
Line-height	定义了文本的行高	Normal <number> <length> <percentage>

我们可以从表中看到在这里可以定义文本的文字间距、字母间距、装饰、对齐方式、缩进方式和行高等属性。举个例子来看看吧。比如下面这段文字（如左下图）：



我们对这段文字加上文本属性重新排版，排版后的效果如右上图所示。

我们看到经过文本属性处理的文本字与字之间多了间距，行与行之间多了行高，对齐方式变成了两端对齐，并且段首又多缩进了两格。这是怎样实现的呢？很简单，只是在 html 中加了如下的代码：

```
<p style="letter-spacing: 1em; text-align: justify; text-indent: 4em; line-height: 17pt">.....</p>
```

```
/*letter-spacing 设置了字间距为 1em，其中 1em 为一个长度单位；text-align 设置了对齐方式为两端对齐；indent 设置了缩进格为 4em；line-height 设置了行高为 17pt*/
```

从上面的例子，我们可以看出利用 CSS 的文本属性可以方便的对页面中的文本进行排版。下一节我将向您介绍文本的 Text-decoration 属性的一个小应用。

※装饰超链接

网页默认的链接方式是这样的：未访问过的链接是蓝色文字并带蓝色的下划线，访问过的超级链接是深紫色的文字并带深紫色的下划线。如果您所有的网页都是这种样式，是不是很单调呢？

其实，利用文本属性中的 Text-decoration 属性就可以实现对超链接的修饰。我们先看下面的这段代码：

```
<html>
<title>link css</title>
<head>
<style>
<!--
```



```

/*定义伪类元素（a：），大括号内定义了前景色属性和文本装饰属性，
hover 加上‘font-size’属性目的是让鼠标激活链接时改变字体*/
a: link{color: green;text-decoration: none}
/*未访问时的状态，颜色为绿色(green)，文本装饰属性(text-decoration)值
为没有(none)*/
a: visited{color: red;text-decoration: none}
/*访问过的状态，颜色为红色(red)，文本装饰属性值为没有*/
a: hover{color: blue;text-decoration: underline;font-size: 20pt}
/*鼠标激活的状态，颜色为蓝色(blue)，文本装饰属性值为上划(underline)，
字体大小为 20pt*/
-->
</style>
</head>
<body>
<p style="font-family:行书体; font-size: 18pt">
<a href="http: //www.hongen.com">未访问的链接</a></p>
/*加链接，显示三种不同状态，并且定义了链接文本的字体和大小*/
<p> <a href="http: //www.hongen.com">访问过的链接</a></p>
<p> <a href="http: //www.hongen.com">鼠标激活的链接</a></p>
</body>
</html>

```

您如果想看上述代码的效果，请[点击这里](#)。

我们从例子中看到没有访问过的链接以绿色显示，并且去掉了下划线；而访问过的链接以红色且没有下划线显示；另外，当鼠标激活链接时，链接以蓝色显示，并且加上了上划线。这种效果是怎么实现的呢？它除了运用了文本属性中的 `text-decoration` 属性，而且采用了伪类元素。

通过上面的代码注释，相信您应该对伪类元素有一个大概认识。实际上我们用到的这种伪类应当称之为“锚伪类”，它规定了链接不同状态下的效果。

怎么样，是不是很简单就可以实现动态链接的效果，赶紧自己动手试一试吧！下一节我将向您介绍“容器”属性。

4、“容器”属性

※边距属性

听起来是不是很有趣，什么叫“容器”属性呢？CSS 的容器属性包括边距、填充距、边框和宽度、高度、浮动、清除等属性。

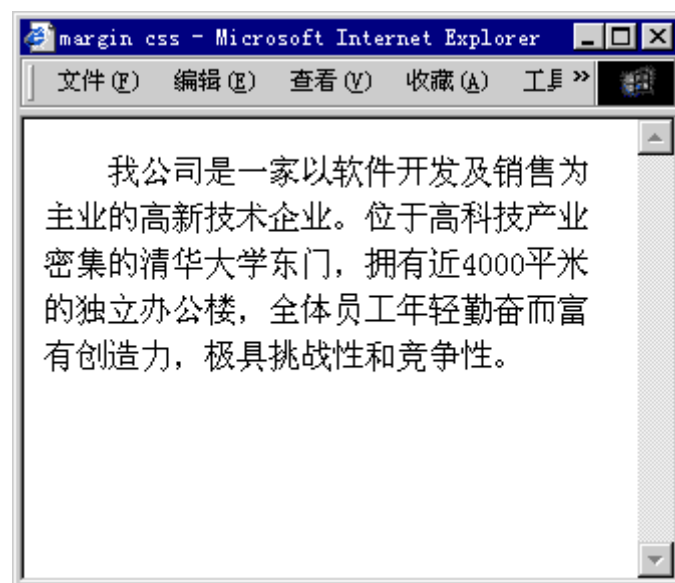
您想，网页中的内容肯定是都需要“装”进这个“容器”里。“容器”属性是 CSS 中非常

重要的一种属性，我们将分类进行学习。

先来看一下边距属性的详细列表：

属性	属性含义	属性值
Margin-top	设置顶端边距	Length percentage auto
Margin-right	设置右侧边距	同上
Margin-bottom	设置底端边距	同上
Margin-left	设置左侧边距	同上

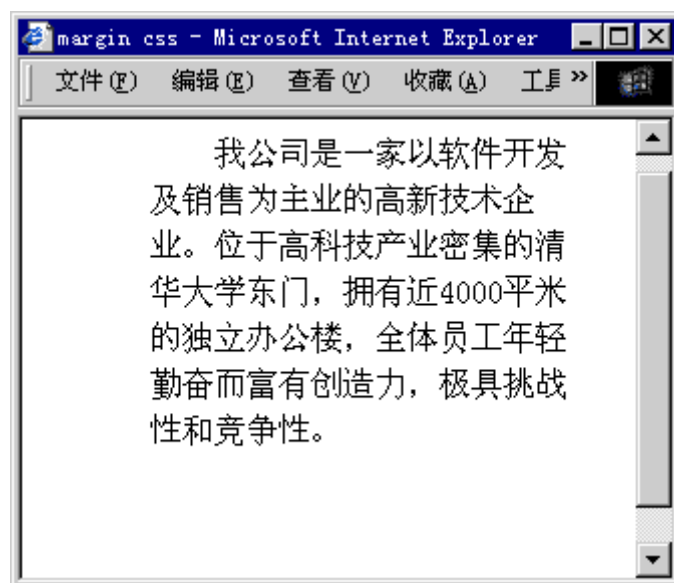
和 font 属性一样，表中的四个属性可以用一个属性一次。边距顺序是上、右、下、左。我们还是用定义边距属性前后的两个例子来看一下吧。请看下面的这幅图：



我们看到上图是没有加上边距属性的效果，而下图则在<head>中加入了如下代码：

```
body{margin: 1em 2em 3em 4em}  
/*定义文本的上、右、下、左的边距分别为 1、2、3、4em*/
```

产生的效果如下图所示：



定义了边距后的页面，无论您怎么改变窗口的大小，它都会按照您定义好的边距样式显示。

如果在 `margin` 里有缺失的属性，则按照顺序依次排列。比如：

`body{margin: 1em 3em}` 上面这段代码的含义是：`body` 的上、右边距为 1、3em。下一节我将向您介绍填充距属性。

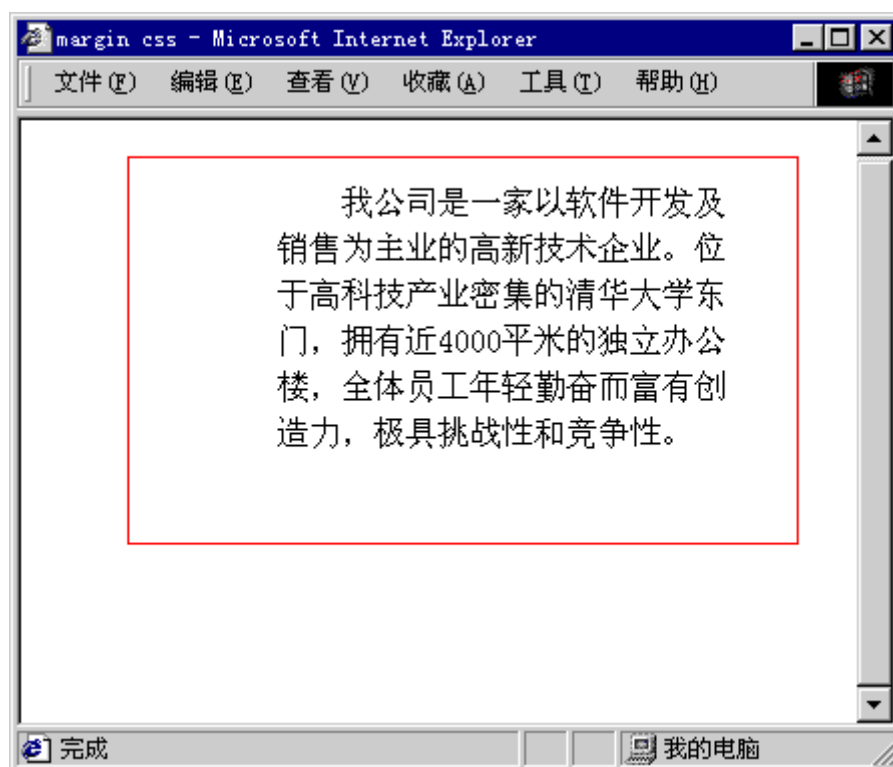
※填充距属性

本节我将向您介绍填充距属性。我们先来看一下填充距的属性列表：

属性	属性含义	属性值
Padding-top	设置顶端填充距	Length percentage
Padding-right	设置右侧填充距	同上
Padding-bottom	设置底端填充距	同上
Padding-left	设置左侧填充距	同上

首先我们得先了解什么叫填充距，它与边距有什么差别呢？填充距指的是文本边框与文本之间的距离。

这样讲是不是有些难懂，由于填充距属性与 `margin` 一样，都有一个一次定义的属性：`padding`，所以我们举个例子，看一下到底什么叫填充距，请看下图：



我们看到上图中红框圈住的地方就是填充距起作用的部分。实际上上图的效果只是在上一节例子的<head>中把下面这段代码加入到 Body 定义的 margin 后面：

```
padding: 1em 2em 3em 4em
```

从上可以看出，红框标记外的部分属于 margin 属性在起作用。
下一节将向您介绍边框属性。

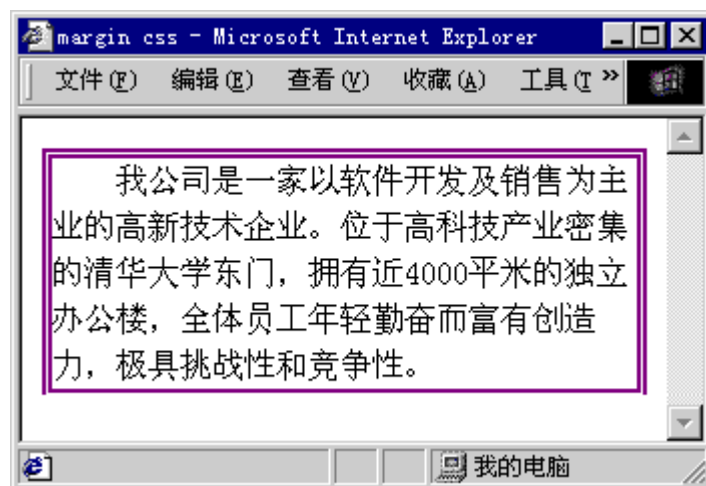
※边框属性

本节我们来看一下种类繁多的边框属性，先来看一下属性列表：

属性	属性含义	属性值
Border-top -width	设置顶端边框宽度	Thin medium thick length
Border-right -width	设置右侧边框宽度	同上
Border-bottom -width	设置底端边框宽度	同上
Border-left -width	设置左侧边框宽度	同上
Border-width	一次定义边框宽度	同上
Border-color	设置边框颜色	Color
Border-style	设置边框样式	None dotted dash solid double groove ridge inset outset
Border-top	一次定义顶端的各种属性	Border-top-width border-style color
Border-right	一次定义右侧的各种属性	同上
Border-bottom	一次定义底端的各种属性	同上
Border-left	一次定义左侧的各种属性	同上

是不是看上去很多，其实应用起来很方便。

只要灵活的运用一次定义的属性就会方便很多。其实还有一个一次设置所有边框的宽度、样式和颜色属性的 **border** 属性，但是 **border** 只能使四条边框的属性值是一样的。我们现在来给前面的例子设置一个 5 像素宽、双边框并且是紫色的边框（如下图）：



这就是在<head>中定义<p>的属性，代码如下：

```
P{border: 5px double purple}/*定义了四条边框为一样的值*/
```

您还可以找一段文本自己多尝试一下其他的边框属性，看看还有什么效果。下一节将向您介绍一下“容器”属性的其他属性，您将能欣赏到图文混排的效果。

※图文混排

我们在来看看“容器”属性还有什么其他的属性，请看属性列表：

属性	属性含义	属性值
Width	定义宽度属性	Length percentage auto
Height	定义高度属性	Length auto
Float	使文字环绕在一个元素的四周	Left right none
Clear	定义某一边是否有环绕文字	Left right none both

上表中的 float 属性很有意思，我们看看用它作的一个页面效果（如下图）：



您是不是以为在图片那里插入了一个表格才实现这种图文混排的效果的呢？

其实不是的，在这里只是在<head>中定义了一下的 float 和 margin 属性，代码如下：

```
<html>
<title>margin css</title>
<head>
<style>
<!--
img{margin-right: 2em; float: left}
/*定义图片的右边距为 2em，图片浮动在文字的左边*/
-->
</style>
</head>
```

```

<body>
<p class="p2" style="line-height: 17pt">

    我公司是一家以软件开发及销售为主业的高新技术企业。位于高科技产业
    密集的清华大学东门，拥有近 4000 平米的独立办公楼，全体员工年轻勤奋而
富
    有创造力，极具挑战性和竞争性。</p>
</body>
</html>

```

怎么样，不用表格也能实现图文混排的效果，并且减少了大量的代码。您还可以试试 float 的其他效果。

好了，“容器”属性就讲这么多，我们可以看出，充分利用 CSS 的“容器”属性，将会使您的页面布局更加合理，样式更加漂亮。下一节我将向您介绍分级属性。

5、分级属性

如果您使用过 Word，那您一定知道 Word 中有一个“项目符号和编号”的功能。

在 html 中，您无需使用前面提到的一些字体、颜色、容器属性来对字体、颜色和边距、填充距等进行初始化。因为在 CSS 中，已经提供了进行分级的专用分级属性。

分级属性包括了“list-style”（列表样式）、“list-item”（列表项）等多种属性，我们先来看一下分级属性的详细列表：

属性	属性含义	属性值
Display	定义是否显示	Block、inline、list-item、none
White-space	决定怎样处理空白部分	Normal、pre、nowrap
List-style-type	在列表项前加项目编号	Disc、circle、square、decimal、lower-roman、upper-roman、lower-alpha、upper-alpha、none
List-style-image	在列表项前加图案	<url> none
List-style-position	决定列表项中第二行的起始位置	Inside、outside
List-style	一次定义前面的列表属性	<keyword> <positon> <uRl>

下面我们来看一个分级属性的例子，代码如下所示：

```

<html>
<head>
<title> fenji css </title>
<style type="text/css">/**定义 CSS*/
<!--
p{display: block; white-space: normal}

```

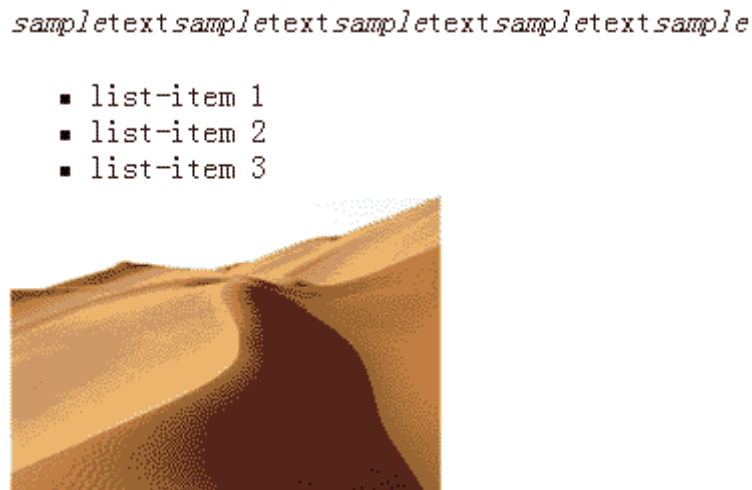
```

/**定义显示属性为“block”，意思是在新的区域里显示； white-space 的属性
    normal 使多重空白合成为一个*/
em{display: inline}
/**定义 EM 的显示属性为“inline” 目的是为了其控制下的文本显示中不折行*/
li{display: list-item; list-style: square}
/**定义 li 的显示属性为“list-item”，显示在列表项中加入一个符号标记， 而
list-style 属性值定义为 square 使列表项前的标记为方块*/
img{display: block} /**定义图片属性为 block 使图片在新的位置打开*/
-->
</style>
</head>
<body>
    <p><em>sample</em>text<em>sample</em>text<em>sample</em>
text<em>sample</em> text<em>sample</em></p>/**定义一段文本*/
    <ul>/**定义列表项*/ <li>list-item 1</li>
    <li>list-item 2</li> <li>list-item 3</li> </ul>
    <p><img src=“ss01068.jpg” width=“280”height=“185”
        alt=“invisible”></p>/**定义一幅图片*/

</body>
</html>

```

上段代码的显示效果如下图：



我们看到由于定义了<p>的属性为 Block，所以文本、列表、图片都在不同的位置上打开，Inline 属性使文本不折行，list-style-type 的属性值为 square 使列表项前的符号为方块；

如果我们在上面的代码中做一些改动，则将以另一种效果显示，我们在<head>中把“EM”的 display 属性值改为 block，使其都在新的位置打开；li 的“list-style”属性值改为

“Upper-roman”（大写罗马符号），img 的“display”属性值改为“none”（让图片不显示）。修改后的显示效果如下图：

```
sample
text
sample
text
sample
text
sample
text
sample
text
sample

I. list-item 1
II. list-item 2
III. list-item 3
```

我们看到分级属性也能达到排版的部分功能，几处代码的改动就可以使页面发生很大的变化。自己来试一试其他的效果吧！

下一节我将向您介绍鼠标属性。

6、鼠标属性

我们知道，当把鼠标移动到不同的地方时，当鼠标需要执行不同的功能时，当系统处于不同的状态时，都会使鼠标的形状发生改变。

用 CSS 来改变鼠标的属性，就是当鼠标移动到不同的元素对象上面时，让鼠标以不同的形状、图案显示。

在 CSS 当中，这种样式是通过“cursor”属性来实现的。Cursor 属性有很多的属性值，我们来看一下它的详细列表：

属性值	解释
Auto	自动，按照默认状态自行改变
Crosshair	精确定位“十”字
Default	默认指针
Hand	手形
Move	移动
e-resize	箭头朝右方
Ne-resize	箭头朝右上方
Nw-resize	箭头朝左上方
n-resize	箭头朝上方
Se-resize	箭头朝右下方
Sw-resize	箭头朝左下方
s-resize	箭头朝下方
w-resize	箭头朝左方
Text	文本“I”形
Wait	等待
Help	帮助

我们来看一个例子吧，请看下面这段代码：

```

<html>
  <head>
    <title>changemouse</title>
  </head>
  <body>
    <h1 style="font-family:文鼎新艺体简">鼠标效果</h1>/**设置字体*/
    <p style="font-family:行书体; font-size:16pt; color:red">
      请把鼠标移到相应的位置观看效果。</p>/**设置字体的名称、大小、颜色*/
    <div style="font-family:行书体; font-size:24pt; color:green; ">
      /**设置该区域内的字体名称、大小、绿色*/
      <p><span style="cursor:hand">手的形状</span><br>
        /**设置鼠标属性为手的形状*/
        <span style="cursor:move">移动</span><br>/**设置鼠标属性为移动*/
        <span style="cursor:ne-resize">反方向</span><br>
        /**设置鼠标属性为反方向*/
        <span style="cursor:wait">等待</span><br>/**设置鼠标属性为等待*/
        <span style="cursor:help">求助</span>/**设置鼠标属性为求助*/
      </p>
    </div>
  </body>
</html>

```

上述代码产生的效果请您[点击这里](#)

CSS 的属性就讲到这里，我们看到它的属性还是比较多的，但是您只要搞清楚它们都是什么意思，应用起来就会很方便。在您应用的时候，还可以参照我为您提供的[属性详细列表](#)。

下一章将向您介绍 CSS 定位。

四、CSS 定位

1.动态转换

通过在 CSS 中设置属性，我们可以准确的定义一个页面的样式，如颜色、字体、边框等。现在我们要讲的 CSS 定位主要是在页面的布局和控制上进行定义，使您的页面从这两个方面都展现的非常完美，更加富有动感。

另外，在讲解之前，我们首先介绍两个定义：相对定位和绝对定位。相对定位就是允许在文档的原始位置上进行偏移。而绝对定位则允许任意定位。

实现 CSS 的定位最终还是要靠属性。我们来看一下定位属性的详细列表（见下图）：

属性	属性含义	属性值
Position	定义位置	Absolute relative static
Left、top	指定横向、纵向坐标位置	Length percentage auto
Width、height	指定占用空间的大小	同上
Clip	剪切	Shape auto
Overflow	内容超出时的处理方法	Visible hidden scroll auto
z-index	产生立体效果	Auto integer
Visibility	定义可见性	Inherit visible hidden

我们在下面举一个例子，其中带了一点 Script 的内容，我们来看看它是如何实现动态显示的。讲解之前我们还是先看一下这个例子的效果吧，请[点击这里](#)

看，通过简单的 CSS 定位实现了颇具动感的效果。这种效果的代码如下：

```
<html>
  <head>
    <title>dingwei css</title>
```

```

<style type="text/css">
<!--
#container1{position: absolute; top: 100}
/*定义 container1 为绝对位置*/
#container2{position: absolute; top: 100; visibility: hidden; }
/*定义 container2 为绝对定位，初始可见度为 hidden（隐藏）*/
p{font-size: 12pt; }/*定义 p 的字体*/
-->
</style>
</head>
<body>
<p style="font-family:行书体; font-size:15pt; color:#cc33cc">
请选择一幅图片： </p>/*设置字体大小、名称、颜色*/
<div id="container1">
<dd>
/*导入一张图片，标识符为 container1*/
<p style="font-family:行书体; color:#cc9933; font-size:12pt">
名称：大漠</p>/*设置字体名称、颜色、大小*/
</dd>
</div>
<div id="container2">
<dd>
/*导入另一种图片，标识符为 container2*/
<p style="font-family:行书体; color:#3366cc; font-size:12pt">
名称：大海</p>
</dd>
</div>
<form name="myform">/*定义两个按钮*/
<p><input type="button" value="大漠"
onclick="container1.style.visibility='visible';
container2.style.visibility='hidden'">
/*定义鼠标点击事件图片 1 为可见，图片 2 为不可见*/
<input type="button" value="大海";
container1.style.visibility='hidden';
onclick="container2.style.visibility='visible';
container1.style.visibility='hidden'"></p>
/*定义鼠标点击事件图片 1 为不可见，图片 2 为可见*/
</form>
</body>
</html>

```

代码虽然长了一些，但结构很简单，您只要用心看注释，很容易理解的。下一节我将向您继续介绍一个 CSS 定位的例子。

2.空间定位

在这一节里，我们来看一个利用 z-index 定位的例子，这个例子的效果 [在这里](#)。

我们看到例子中的两幅图片和一段文字分别处于不同的空间位置，文字覆盖在那朵花的图片上，而挥动小旗的小老鼠却又覆盖在文字的上面。那么这种效果是怎样实现的呢？这里利用了 CSS 定位的 z-index 属性，代码如下：

```
<html>
<head>
<title>zindex</title>
<style type="text/css">
<!--
.pile {position:absolute; left:2in; top:2in;
width:3in; height:3in; } /*定义了类 pile，以及它的位置*/
.pile1 {position:absolute; left:3in; top:2in;
width:1in; height:1in; } /*定义了类 pile1，以及它的位置*/
-->
</style>
</head>
<body>

/*导入一张图片，使它为 pile 类，z-index 属性定义为 1，位置处于最下方
*/

<div class="pile" id="text1"
style="color:#ffff33;z-index:2"> 这段文字将覆盖在图片上。
</div> /*定义一段文字的颜色和 z-index 属性为 2，处于中间位置*/

/*导入第二张图片，使它为"pile1"类，z-index 属性为 3，位置处于最上方*/
</body>
</html>
```

通过这两节的例子，我们可以看到 CSS 定位具有强大的功能，至于其他的一些定位属性，您可以自己尝试一下，很简单的。

利用好了 CSS 的定位功能，会使您的页面更加精致，更加富有动感。
下一章我将带您进入 CSS 滤镜的精彩世界。

五、CSS 滤镜

1、概述

好了，下面我们将进入 CSS 的最精彩的部分--滤镜，它将把我们带入绚丽多姿的多媒体世界。正是有了滤镜属性，页面才变得更加漂亮。

CSS 的滤镜属性的标识符是 **filter**。为了使您对它有个整体的印象，我们先来看一下它的书写格式：

`filter: filtername (parameters)`

怎么样？是不是很简单，看上去与前面讲的属性定义没什么太大的差别。**Filter** 是滤镜属性选择符。

也就是说，只要您进行滤镜操作，就必须先定义 **filter**；**filtername** 是滤镜属性名，这里包括 **alpha**、**blur**、**chroma** 等等多种属性，详细内容请看下表：

属性名称	属性解释
Alpha	设置透明度
Blur	设置模糊效果
Chroma	设置指定颜色透明
Dropshadow	设置投射阴影
Fliph	水平翻转
Flipv	垂直翻转
Glow	为对象的外边界增加光效
Grayscale	设置灰度（降低图片的彩色度）
Invert	设置底片效果
Light	设置灯光投影
Mask	设置透明膜
Shadow	设置阴影效果
Wave	利用正弦波纹打乱图片
Xray	只显示轮廓

上面 **filter** 表达式中括号内的 **parameters** 是表示各个滤镜属性的参数，也正是这些参数决定了滤镜将以怎样的效果显示。

看了上面长长的列表，是不是觉得很困难呀？不要紧，我们接下来一个一个的介绍

这些属性在 CSS 中是怎样实现的（很简单哟^_^）。

下一节我们将首先学习 Alpha 透明属性的应用。

2、alpha 属性

alpha 是用来设置透明度的。先来看一下它的表达格式：

```
filter: alpha (opacity=opacity, finishopacity=finishopacity,  
style=style, startX=startX, startY=startY, finishX=finishX,  
finishY=finishY)
```

哇，怎么这么长。是啊，不过这些参数都各有其用。

Opacity 代表透明度等级，可选值从 0 到 100，0 代表完全透明，100 代表完全不透明。Style 参数指定了透明区域的形状特征。其中 0 代表统一形状；1 代表线形；2 代表放射状；3 代表长方形。

Finishopacity 是一个可选项，用来设置结束时的透明度，从而达到一种渐变效果，它的值也是从 0 到 100。StartX 和 StartY 代表渐变透明效果的开始坐标，finishX 和 finishY 代表渐变透明效果的结束坐标。

从上面讲的我们可以看出，如果不设置透明渐变效果，那么只需设置 opacity 这一个参数就可以了。说了这么多，我们来看一个实例吧（见下图）：



实现上面这种效果的代码如下：

```
<html>  
<head>  
<title>alpha</title>  
<style>/**定义 CSS 样式*/  
<!--  
div{position: absolute; left: 50; top: 70; width: 150; }  
/**定义 DIV 区域内的样式（位置为绝对定位，left、top、width 的坐标）*/  
img{position: absolute; top: 20; left: 40;  
filter: alpha(opacity=80)}  
/**定义图片的样式，绝对定位，滤镜属性是透明度为 80*/
```

```

-->
</style>
</head>
<body>
<div>
<p style="font-size: 48; font-weight: bold; color: red; ">
Beautiful </p>/**定义字体属性，前景色为红色*/
</div>
<p> </p>
/**导入一张图片*/
</body>
</html>

```

如果在上面的代码中稍做改动，则将产生另外多种效果。我们只修改 `img` 的样式属性，把 `head` 中的 `Img` 样式属性代码改为如下所示：

```

img{position: absolute; top: 20; left: 40;
filter: alpha(opacity=0,finishopacity=100,
style=1,startx=0,starty=85,finishx=150,finishy=85);}
/**设置透明渐变效果，起始坐标，终止渐变坐标，并设置透明样式值（style=1）为
线形*/

```

这段代码产生的效果如左下图所示，右面的两幅图分别是把 `Alpha` 中的 `Style` 参数值为 2 和 3 后的效果，点击缩略图可放大。



Style=1



Style=2



Style=3

以上是 CSS 的 `Alpha` 滤镜属性的应用，具体应用还需要您自己找个例子练一练。下一节我们将介绍 `blur`（模糊）属性。

3、blur 属性

假如您用手在一幅还没干透的油画上迅速划过，画面就会变得模糊。CSS 下的 `blur` 属性就会达到这种模糊的效果。

先来看一下 `blur` 属性的表达式：

```
filter: blur (add=add, direction, strength=strength)
```


我们看到 blur 属性有三个参数：add、direction、strength。

Add 参数有两个参数值：true 和 false。意思是指定图片是否被改变成模糊效果。Direction 参数用来设置模糊的方向。模糊效果是按照顺时针方向进行的。其中 0 度代表垂直向上，每 45 度一个单位，默认值是向左的 270 度。角度方向的对应关系见下表：

角度	方向
0	Top（垂直向上）
45	Top right（垂直向右）
90	Right（向右）
135	Bottom right（向下偏右）
180	Bottom（垂直向下）
225	Bottom left（向下偏左）
270	Left（向左）
315	Top left（向上偏左）

Strength 参数值只能使用整数来指定，它代表有多少像素的宽度将受到模糊影响。默认值是 5 像素。

还是看一个例子吧。[点击这里看效果](#)

看起来是不是有些像万花筒，在这个例子中加入了一些 JavaScript 的语句，代码如下：

```
<html>
  <head>
    <title>blur css</title>
    <script>
      function handlechange(obj)
      /*设置一个循环函数 handlechange，对象是 obj*/
      { with(obj.filters(0))/*Obj 的 filter 属性*/
        { if (strength<255)/*设置循环条件*/
          { strength +=1; direction +=45; }
          /*每循环一次 strength 就加 1，direction 加 45 度*/
        }
      }
    </script>
  </head>
  <body>
    <p>
```

```

    /*导入一幅图片，初始 blur 属性 strength 等于 1，同时调用 onfilterchange 函数*/
</p>
</body>
</html>

```

注：在 javascript 中 blur 属性是这样定义的：

```
[oBlurfilter=] object.Filters.blur
```

这个例子是 Blur 属性的一个比较复杂的例子，下一节我将向您介绍两个较简单的 blur 属性效果。

通过 blur 属性还可以设置页面中的字体。如果把字体的 blur 属性 add 参数值定义为 1，得出来的字体效果是这样的（如下图）：



怎么样，是不是有些印象派的意思，这种效果的实现代码如下：

```

<html>
  <head>
    <title>filter blur</title>
    <style>/*CSS 样式定义开始*/
    <!--
    div{ width: 200;
    filter: blur(add=true,direction=90,strength=25); }
    /*设置 DIV 样式，滤镜 blur 属性*/
    -->
  </style>
</head>
<body>
  <div style="width:702; height: 288">
    <p style="font-family:lucida handwirting italic;
    font-size:72; font-style:bold; color:rgb(55,72,145); ">
    LEAF</p>
    /*定义字体名称、大小、样式、前景色*/
  </div>
</body>
</html>

```

我们看到 strength 设置为 25，如果把其值再改大一些，就会达到非常夸张的效果，同时把 Direction 参数值为 180，显示效果如下图：



用 blur 属性设置字体可以达到很多效果，把 direction 和 strength 再做修改，还能达到多种效果，您可以自己修改试一试。

这一节我们介绍了 Blur 属性，下一节将向您介绍 Chroma 属性。

4、Chroma 属性

Chroma 属性可以设置一个对象中指定的颜色为透明色，它的表达式如下：

Filter: Chroma (color=color)

这个属性的表达式是不是很简单，它只有一个参数。只需把您想要指定透明的颜色用 Color 参数设置出来就可以了。比如下面这幅图：



图中显示两种字体，两种颜色，我们现在对“leaves”字体添加 chroma 属性，使其透明。代码如下：

```
<html>
  <head>
    <title>chroma filter</title>
    <style>
      <!--
      div{position: absolute; top: 70; width: 200;
        filter: chroma(color=green)}
      /**定义 DIV 范围内绿色为透明色，另外设置 DIV 的位置*/
      p{font-family: bailey; font-size: 48; font-weight: bold;
```

```

color: green} /*设置 P 的字体名称、大小、粗细、颜色*/
em{font-family:lucida handwriting italic; font-size:48;
font-weight: bold; color: rgb(255,51,153)}
/*设置 EM 的字体名称、大小、粗细、颜色*/
-->
</style>
</head>
<body>
<div>
<p>LEAVES <em>LOVE</em></p>
</div>
</body>
</html>

```

通过上面代码中对 `chroma` 的属性设置，使绿色透明。显示效果如下图：



我们看到绿色的 `leaves` 字体不见了，实际上它是透明了，在 IE 下点击它所在的区域，它还是会显示出来（见下图）：



另外，需要注意的是，`chroma` 属性对于图片文件不是很适合。因为很多图片是经过减色和压缩处理（比如 JPG、GIF 等格式），所以它们很少有固定的位置可以设置为透明。

本节我们讲述了 `chroma` 属性的应用，下一节将向您介绍 `Dropshadow` 属性。

5、DropShadow 属性

DropShadow 属性是为了添加对象的阴影效果的。它实现的效果看上去就像使原来的对象离开页面，然后在页面上显示出该对象的投影。看一看它的表达式：

Filter: DropShadow (Color=color, Offx=Offx, Offy=offy,
Positive=positive)

该属性一共有四个参数： Color 代表投射阴影的颜色。 Offx 和 offy 分别 X 方向和 Y 方向阴影的偏移量。偏移量必须用整数值来设置。如果设置为正整数，代表 X 轴的右方向和 Y 轴的向下方向。设置为负整数则相反。

Positive 参数有两个值： True 为任何非透明像素建立可见的投影， False 为透明的像素部分建立可见的投影。

同样，我们先来看一个例子（见下图）：



看，图中的文字就像是从页面上飞出来一样，并且留下了一层淡淡的影子。实际上在这里应用的就是 CSS 的 DropShadow 属性，我们来看一下它的代码：

```
<html>
  <head>
    <title>dropshadow </title>
    <style>/**定义 CSS 样式*/
    <!--
    div {position:absolute; top:20; width:300;
    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>
    <p style="font-family: matisse itc; font-size: 64;
      font-weight: bold; color: #CC00CC; ">
      Love Leaf </p>
  </div>
</body>
</html>
```

和 **chroma** 属性一样，**Dropshadow** 属性对图象的支持不好，我指的是 **JPEG**、**GIF** 格式的图象文件。

不能支持的原因与 **Chroma** 一样，因为这种图象的颜色很丰富，很难找到一个投射阴影的位置。

本节讲述了 **Dropshadow** 属性的应用，下一节将向您介绍 **FlipH**、**FlipV** 的属性应用。

6、**FlipH**、**FlipV** 属性

Flip 是 **CSS** 滤镜的翻转属性，**FlipH** 代表水平翻转，**FlipV** 代表垂直翻转。它们的表达式很简单，分别是：

Filter: **FlipH**

Filter: **FlipV**

我们先来看一幅图：点击可放大



下面我们分别对它实现水平翻转和垂直翻转，并且在图片上方的一段文字，也发生翻转。代码如下：

```
<html>
  <head>
    <title>flip css</title>
    <style>/**设置 CSS 样式开始*/
    <!--
    div{position: absolute; top: 20; width: 300;
      filter: flipH (flipV) ; }
    /**定义 DIV 范围内的样式，绝对定位，翻转为水平翻转或垂直翻转。
      注意：在这里 flipH 和 flipV 只取其中的一个*/
    img{position: absolute; top: 70; left: 40;
      filter: flipH (flipV) ; }
    /**定义图片的样式，绝对定位，翻转属性和 DIV 一样。*/
    -->
  </style>
</head>
<body>
```

```

<div>
<p style="font-family: bailey; font-size: 36pt;
        font-weight: bold; color: rgb(10,128,156); ">
    Leaf Village </p>
    /**定义字体名称、大小、粗细、颜色*/
</div>
<p></p>
    /**导入一张图片*/
</body>
</html>

```

代码产生的两个效果分别如下图：（点击可以放大）



水平翻转



垂直翻转

翻转的属性应用是不是很简单。下一节我将向您介绍 Glow 属性。

8、Glow 属性

当对一个对象使用“Glow”属性后，这个对象的边缘就会产生类似发光的效果。它的表达式如下：

Filter: Glow (Color=color, Strength=strength)

Glow 属性的参数只有两个：Color 是指定发光的颜色，Strength 指定发光的强度，参数值从 1 到 255。让我们先来看一下加上 Glow 属性的效果图：



怎么样，是不是有一种燃烧的火焰的感觉。实现这种效果的代码如下：

```
<html>
  <head>
    <title>filter glow</title>
    <style>/**开始设置 CSS 样式*/
    <!--
      .leaf{position: absolute; top: 20; width: 400;
        filter: glow(color=#FF3399,strength=15); }
    /**设置类 leaf，绝对定位，Glow 滤镜属性，发光颜色值为#FF3399，强度为
    15*/
      .weny{position: absolute; top: 70; left: 50; width: 300;
        filter: glow(color=#9966CC,strength=10); }
    /**设置类 weny，绝对定位，glow 滤镜属性，发光颜色值为#9966CC，强度为
    15*/
    -->
  </style>
</head>
<body>
  <div class="leaf">/**leaf 类样式*/
  <p style="font-family: lucida handwriting;
    font-size: 54pt; font-weight: bold; color: #003366; ">
    Leaf Mylove</p>/**设置字体名称、大小、粗细、颜色*/
  </div>
  <div class="weny">/**weny 类样式*/
  <p style="font-family: bailey; font-size: 48pt;
    font-weight: bold; color: #99CC66; ">
    /**设置字体名称、大小、粗细、颜色*/
    Weny Good!</p>
  </div>
</body>
</html>
```

您还可以随意修改颜色值，看看其他的发光效果是怎样的。

本节主要讲了 Glow 属性的应用，下一节将向您介绍 Gray 属性。

7、Gray 属性

Gray 属性把一张图片变成灰度图。它的表达式很简单：

Filter: Gray

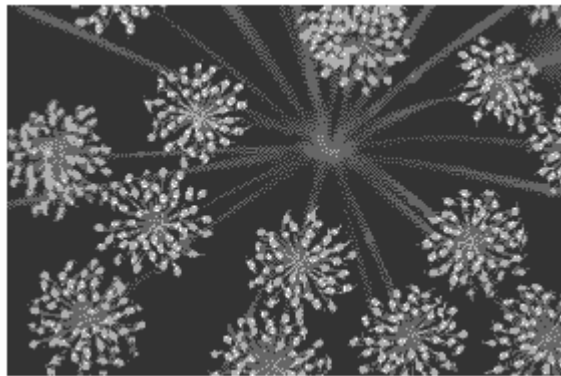
其实这个属性没什么好讲的，只需在您定义的 **IMG** 样式中加入一句代码：

{Filter: Gray}就一切 OK 了。

下面两幅图分别代表未加 **Gray** 属性和添加了 **Gray** 属性的效果：



这是原图，点击可放大



转成 gray 属性后的效果图

本节讲述的内容比较简单，下一节将向您介绍 **Invert** 转化属性。

9、Invert 属性

Invert 属性可以把对象的可视化属性全部翻转，包括色彩、饱和度和亮度值。它的表达式也很简单：

Filter: Invert

我们再来看一下加上 **Invert** 属性前后的图片效果变化（如下图）：



原图
(点击可放大)



Invert 属性效果图
(点击可放大)

我们看到 **Invert** 属性实际上达到的是一种“底片”的效果。
自己拿别的图来试试吧。本节介绍的是 **Invert** 属性，下一节将向您介绍 **Mask** 属性。

10、Mask 属性

Mask 属性为对象建立一个覆盖于表面的膜。它的表达式也很简单：

Filter: Mask (Color=颜色)

只有一个 **Color** 参数，用来指定使用什么颜色作为掩膜。
同样，我们来看一下一幅图片在加上 **mask** 属性前后的效果（见下图）：

wenyleaf

原 图



Mask 属性效果图

加上 **MASK** 属性的效果就好象是在用有色眼镜看物体一样。上面的效果的代码如下：

```
<html>
  <head>
    <title> mask filter </title>
    <style>/**设置 CSS 样式开始**/
    <!--
      div{position: absolute; top: 20; left: 40;
        filter: mask(color: #666699); }
    /**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**/
```

```

    p{font-family: bailey; font-size: 72pt;
      font-weight: bold; color: #FF9900; }
  /*定义 P 区域内的样式，字体名称、大小、粗细、前景色*/
-->
</style>
</head>
<body>
<div>
<p> wenyleaf </p>
</div>
</body>
</html>

```

其实，您就算在代码中去掉对字体前景色的定义，得到的效果还是一样的。因为有了 **Mask** 属性的定义，它遮罩下的字体颜色的设置就已经失去了意义。

还有一点需要您注意的地方，**mask** 属性对图片文件的支持还是不够，不能达到应该有的效果。

本节讲解了 **Mask** 属性的应用，下一节将向您介绍 **Shadow**（阴影）属性。

11、Shadow 属性

Shadow 属性可以在指定的方向建立物体的投影。它的表达式是这样的：

Filter: Shadow (Color=color, Direction=direction)

在这里，**Shadow** 有两个参数值：**Color** 参数用来指定投影的颜色；**Direction** 参数用来指定投影的方向。

这里说的方向与我们在第二节 **Blur** 属性中提到的“方向与角度的关系”是一样的。也许您会问，前面讲到的 **Dropshadow** 属性和 **Shadow** 属性有什么不同吗？

光说的话，您恐怕还难以理解，让我们看一看分别利用这两个属性做出来的效果有什么不同（见下图）：



Shadow 效果

Dropshadow 效果

这样一对比，就可以很明显的看出两者的不同。

Shadow 属性可以在任意角度进行投射阴影，Dropshadow 属性实际上是用偏移来定义阴影的。所以，看上去左图的文字和阴影就像是一体的，而右图的文字就像脱离了阴影一样。

本例的代码如下：

```
<html>
  <head>
    <title> shadow</title>
    <style>/**开始设置 CSS 样式*/
    <!--
    .shadow{position:absolute; top:20; width:300;
              filter:shadow(color=#cc66ff,direction=225); }
    /**定义 Shadow 类的样式，绝对定位，Shadow 属性，阴影颜色、投影方向*/
    .dropshadow{position:absolute; top:180; width:300;
              filter:dropshadow(color=#cc66ff,offx=10,offy=10,positive=1); }
    /**设置 Dropshadow 类的样式，样式与 Shadow 类相似，不同的是滤镜用了
    Dropshadow 属性，设置 X 轴和 Y 轴的偏移量*/
    -->
  </style>
</head>
<body>
  <div class="shadow">/**区域内为 Shadow 类*/
  <p style="font-family: bailey; font-size: 48pt;
          font-weight: bold; color: #FF9900; ">
    Hongen Online</p>/**定义字体名称、大小、粗细、前景色*/
  </div>
  <div class="dropshadow">/**区域内为 Dropshadow 类*/
  <p style="font-family: bailey; font-size: 48pt;
          font-weight: bold; color: #FF9900; ">
    Hongen Online</p>/**定义字体样式与 Shadow 类的一样*/
  </div>
</body>
</html>
```

本节讲述了 Shadow 属性的应用，下一节将向您介绍 Wave（波纹）属性。

12、Wave 属性

Wave 属性用来把对象按照垂直的波纹样式打乱。它的表达式如下：

Filter: Wave (Add=True (False), Freq=频率, LightStrength=增强光效,
Phase=偏移量, Strength=强度)

我们看到 Wave 属性的表达式还是比较复杂的, 它一共有五个参数。Add 参数有两个参数值: True 代表把对象按照波纹样式打乱; False 代表不打乱;

Freq 参数指生成波纹的频率, 也就是指定在对象上共需要产生多少个完整的波纹。LightStrength 参数是为了使生成的波纹增强光的效果。参数值可以从 0 到 100。Phase 参数用来设置正弦波开始的偏移量。这个值的通用值为 0, 它的可变范围为从 0 到 100。这个值代表开始时的偏移量占波长的百分比。比如该值为 25, 代表正弦波从 90 度 (360*25%) 的方向开始。

说了一大堆, 我们还是先看一个实例吧。比如下面这幅图片 (点击可放大):



下面我们对上面这个页面加上 Wave 效果, 代码如下:

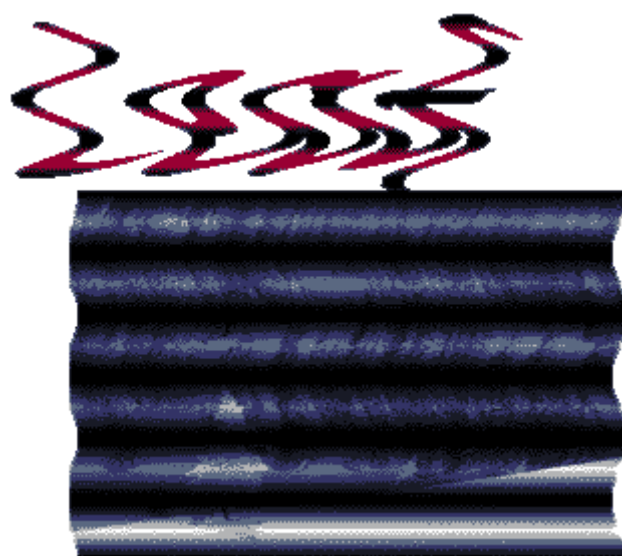
```
<html>
<head>
<title> wave css</title>
<style>/**定义 CSS 样式开始*/
<!--
.leaf{position:absolute; top:10; width:300;
      filter:wave(add=true,freq=3,lightstrength=100,
                  phase=45,strength=20); }
/**设置 leaf 类的样式, 绝对定位, wave 属性, 产生 3 个波纹, 光强为 100, 波
纹
从 162 度 (360*45%) 开始, 振幅为 20*/
img{position:absolute;top:110;left:40;
    filter:wave(add=true,freq=3,lightstrength=100,
                phase=25,strength=5); }
/**设置 IMG 的样式, 绝对定位, wave 属性, 产生 3 个波纹, 光强为 100, 波纹从
90 度开始, 振幅为 5*/
-->
</style>
</head>
<body>
<div class="wave">/**定义 DIV 区域内为 Wave 类*/
<p style="font-family: lucida handwriting;
```

```

        font-size=72pt;  font-weight: bold;
        color: rgb(189,1,64); ">Leaf</p>
    /**设置字体名称、大小、粗细、颜色**/
</div>
    <p></p> /**导入图片**/
</body>
</html>

```

这段代码实现的效果如下图：



如果把 Wave 的参数随便做一下改动，就会达到多种效果，请看另外一种效果：（如下图）：



其实这种效果只是增大了 `freq` 参数的值，减小了 `Strength`、`LightStrength` 的值就可以了。您也可以多试试，改变其他的参数值，还可以达到许多不同的效果来。

本节主要讲述了 `Wave` 属性的应用，下一节将向您介绍 `Xray` 属性。

13、Xray 属性

`Xray` 就是 X 射线的意思。

`Xray` 属性，顾名思义，这种属性产生的效果就是使对象看上去有一种 X 光片的感觉。它的表达式很简单：

Filter: `Xray`

我们还是先来看一个页面（如下图）：



点击可放大

如果在上面的页面中加入 `Xray` 属性，也就是在 `<head>` 的 `<Style>` 中，增添下面这一句代码：

Filter: `Xray`

您再看这个页面就会是另一种效果了：（如下图）



看，是不是就像给它拍了一张 X 光片一样。

