

第 6 章 CSS 层叠样式表

CSS 层叠样式表用于对页面进行更加细微的格式调整。在这里首先介绍一下 DHTML (Dynamic HTML) 的概念，它的基础是 HTML 语言。它使用 JavaScript 控制网页元素，利用 CSS 层叠样式表进行布局。JavaScript 在前一章已经介绍，本章重点就是要向读者介绍 CSS 层叠样式表。它是对页面布局、字体、颜色、背景以及其他图文效果实现更加精确的控制。在第 21 章将要介绍的 Ajax 技术也会使用 CSS 来动态修改页面中某个节点的显示样式。

本章要点包括以下内容：

- ❑ CSS 的概念与特点
- ❑ CSS 的类型与基本写法
- ❑ CSS 中各类属性和值的设置

6.1 CSS 的概念和特点

传统 Web 站点中的 HTML 页面文档结构和显示格式一直是混合在一起的，这给页面开发和调整带来了很多的不便。另外，仅仅通过 HTML 进行字体显示格式的调整是很粗糙的，并不能得到某些特定效果的要求。为了使页面显示格式彻底地独立于文档的结构，以及为了能够更加精确地设置文档显示效果，W3C 为 HTML 制定样式单机制，这就是 CSS。总结起来，CSS 的特点如下：

- ❑ 可以将显示格式和文档结构分离：HTML 标签语言定义了页面的文档结构（例如数据怎样分布和组织，对显示格式的调整能力比较弱）。而层叠样式表可以单独定义文档的显示格式，能够对页面的布局进行更多的控制，可以使得显示格式的控制和文档结构的控制相分离。从而，HTML 可以专门用于组织文档结构，CSS 独立出来控制页面的显示格式。
- ❑ 扩展了以前控制页面布局的能力：HTML 标签语言对文档格式和布局的控制是非常有限和粗糙的，CSS 可以对显示格式进行更准确的控制。例如文字在窗口中的精确定位、之间的行距或者字间距等，都可以通过 CSS 轻松来完成。
- ❑ 制作体积更小、下载更快的网页：和 HTML 标签一样，CSS 样式表也是简单的文本，不需要执行程序。另外，使用 CSS 可以只定义一段代码就对同一页面中的多处文档显示格式进行统一管理，这也大大减少了页面代码的数量。
- ❑ 可以统一定义多个页面显示格式：样式表的主旨就是让文档格式和结构分离。利用样式表，可以将站点上所有的网页都指向单一的一个 CSS 文件。这样只需要修改 CSS 文件中的某一行，就可以让整个站点的网页格式随之改变。
- ❑ 浏览器将成为更友好的界面：样式表的代码兼容很好，哪怕用户丢失了某一个插件也不会发生中断，或者使用老版本浏览器时页面也不会出现杂乱无章。只要可以识别串接样式表的浏览器就可以应用它。

6.2 CSS的类型和基本写法

接下来重点介绍 CSS 的基本定义类型以及将 CSS 样式表应用到 HTML 页面上的方法。

6.2.1 CSS的定义类型

定义 CSS 可以有三种类型，如下所示。

6.2.1.1 自定义 CSS

开发者可以自定义 CSS，并且赋予该 CSS 一个名称，然后可以在整个页面中使用 Class 属性来引用这个 CSS。定义和引用代码如下：

```
.bg {background-image: url(bg.gif);}
...
<body class="bg">
```

代码说明：这里定义了一个名为 bg 的 CSS（请注意定义的 bg 名称前面有一个点号），该 CSS 定义了背景图片为 bg.gif。然后在 HTML 标签<body>中使用 class 属性来引用名称为 bg 的 CSS。

6.2.1.2 重定义 HTML 标签的 CSS

也可以对 HTML 中的某个标签重新定义 CSS，当浏览该页面时，将使用重新定义的 CSS 来显示所有该标签中的文档信息。如下面一段代码：

```
td {color: #000099; font-size: 9pt }
```

代码说明：该代码重新定义了 HTML<td>标签的 CSS，页面中所有包含在<td></td>标签中的文字将安装该 CSS 定义的格式显示。注意和自定义 CSS 方式的不同，另外这里的 td 名称前没有点号。

6.2.1.3 CSS 选择符

CSS 选择符是用来为特殊的 HTML 组合标签定义 CSS 的。常见组合标签有如下 4 种：

- ☐ a:link：设置正常状态下的链接的显示样式；
- ☐ a:active：设置当鼠标单击该链接时的样式；
- ☐ a:visited：设置已经访问过的链接显示样式；
- ☐ a:hover：设置当鼠标放置（没有单击）在链接上时的显示样式。

下面通过一段代码来展示该标签组的 CSS 定义方法：

```
a:link { color: #FF3366;font-family: "宋体";text-decoration:none;}
a:hover { color: #FF6600;font-family: "宋体";text-decoration:underline;}
a:visited { font-family: "宋体";color: #339900; text-decoration:none;}
```

代码说明：该段 CSS 代码定义了当链接在正常状态下、鼠标放置以及访问过的状态下显示不同的颜色和外观。关于这里属性的设置在后面将一一介绍。

6.2.2 CSS的基本写法

CSS 有三种定义方法，也有三种基本写法，具体如下所示。

6.2.2.1 在 HTML 的 head 头部实现

可以在 HTML 的<head></head>标签内，使用<style>开始，</style>标签结束来定义 CSS。

```
<html>
<head>
```

```
<title>CSS 在 head 中重定义 HTML 标签</title>
<style type="text/css">
H1 { font-size: x-large; color:red}
H2 { font-size: large; color:blue}
</style>
</head>
<body>
<h1>这里介绍的是 CSS 在 head 中的定义方法</h1>
<h2>这里介绍的是 CSS 在 head 中的定义方法</h2>
</body>
</html>
```

代码说明：在<style>和</style>标签内部重定义了 HTML 种的<H1>和<H2>标签的 CSS 代码。该页面程序的运行效果如图 6.1 所示。

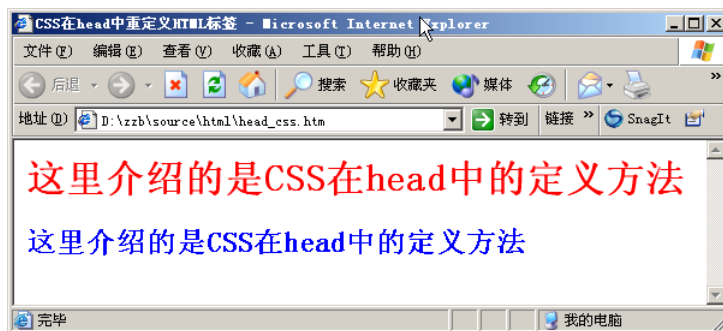


图 6.1 在 head 中定义 CSS

6.2.2.2 在标签体内直接定义

在 HTML 标签体中直接使用 style 属性来指定 CSS，例如指定<td>标签中字体大小，可以这样写：

```
<td style="font-size:15pt">
```

这样直接书写方法虽然方便，但是并不能很好体现 CSS 层叠样式表的优势，所以并不推荐使用。

6.2.2.3 通过 CSS 文件调用实现

以上介绍的 CSS 实现方法都是直接页面文件中书写。在实际详细开发过程中，常常将 CSS 书写在一个单独的文件中，然后在页面中通过下面的语句来引用该 CSS 文件：

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

代码说明：其中 href 属性值指定 CSS 文件所在的位置，一般使用相对路径。另外 rel 和 type 属性来表明引用的这个文件是一个样式表文件。

接下来通过一个完整的实例来演示 CSS 文件外部调用的实现方法，页面程序如下：

```
<html>
<head>
<title>CSS 文件外部调用方法的实现</title>
<link href="out_css.css" rel="stylesheet" type="text/css">
</head>

<body>
<h1>JSP 大全</h1>
<p class="tt">这是一本详细介绍 JSP 使用的书籍！</p>
</body>
</html>
```

代码说明：其中<link href="out_css.css" rel="stylesheet" type="text/css">语句指定引用外部同目录下的 out_css.css 样式表文件。

out_css.css 文件中的 CSS 定义如下：

```
H1 { font-size: 24pt; font-family: "黑体"; font-style:italic;}
.tt { font-family: "宋体"; font-size:9pt; text-decoration:underline; letter-spacing: 3px; line-height: 12pt;}
```

运行以上的页面程序，出现如图 6.2 所示的效果。

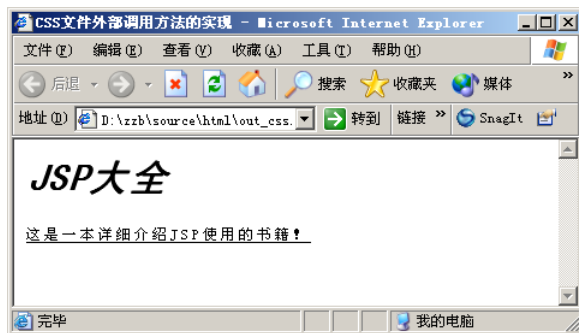


图 6.2 外部调用 CSS 样式表

6.2.3 CSS引用所产生的冲突

在页面程序中使用 CSS，有时会产生冲突。例如，对同一个文本应用了多个 CSS 样式定义，这时可能显示无法预期的效果。浏览器为了正确显示 CSS 样式，一般遵循如下的规则：

- ❑ 当对一个文本引用了多个 CSS 定义，浏览器将尽量显示出这个文本所引用的所有属性。除非多个 CSS 样式定义出现冲突，即属性定义产生重叠。例如在一个 CSS 定义中设置了文本字体大小为 9pt，而在另一个 CSS 样式定义中设置了字体大小为 11pt。
- ❑ 当文本所引用的 CSS 产生冲突时，浏览器将按照文本引用 CSS 的远近来决定显示哪一个属性。
- ❑ 另外，HTML 定义的样式和 CSS 所定义的样式相比较，CSS 样式具有较高的优先级，即浏览器会按照 CSS 所示的样式来显示。

下面还是通过一个实例来演示浏览器对这种冲突的处理，代码如下：

```
<html>
<head><title>浏览器处理 CSS 样式冲突</title>
<style type="text/css">
.red {color: red;}
.blue { color: blue;}
</style>
</head>
<body>
<h2><div class="red">这里显示的文字为红色<div class="blue">这里显示的文字却是蓝色</div>这里显示的文字为
红色</div></h2>
</body>
</html>
```

代码说明：第二个<div>标签中的文字根据远近原则，将调用名为 blue 的 CSS，所以显示蓝色。该页面程序运行的效果如图 6.3 所示。

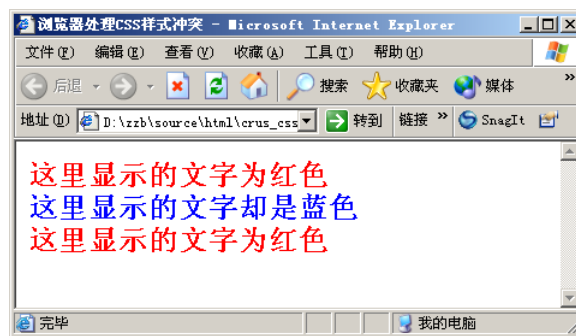


图 6.3 CSS 产生的冲突

至此，已经将 CSS 定义以及引用方法介绍完了，下面将重点讲解一下 CSS 中的各类属性的设置。

6.3 CSS中各类属性和值

本小节开始介绍 CSS 的一些基本属性。

6.3.1 字体属性

经过以上小节的实例介绍，读者应该对 CSS 属性的设置方法有所了解。其中字体属性包括：字体显示类型、字体风格、字体加粗和字体大小，详细见表 6.1 所示。

表 6.1 字体属性

字体属性	描述
Font-family	该属性指定文字所采用字体类型，例如“宋体”还是“楷体”等。
Font-size	设置文本显示的字体大小
Font-style	设置文本显示的字体风格，例如“斜体”。
Font-weight	设置文本显示的字体粗细，例如使用bold值使字体显示加粗。

下面对这些属性的使用逐一介绍：

6.3.1.1 字体显示类型

字体显示类型可以指定一个文本使用“宋体”还是“楷体”或者其他类型来显示，其中可以通过字体名(例如“宋体”)或者字体族科(例如 Arial, Helvetica, sans-serif)来指定。另外,使用 font-family 属性来设置字体显示类型时，也可以指定多个字体显示类型（其中以逗号隔开），这时浏览器会以从前到后的顺序来选择适当的字体。

下面通过实例来具体演示 font-family 属性设置的使用，页面代码如下：

```
<html>
<head>
<title>font-family 属性设置</title>
<style type="text/css">
<!--
  H1 { font-family: "隶书";}
  .text { font-family: "宋体,仿宋_gb3212";}
  .text1 { font-family:"方正舒体";}
-->
</style>
```

```

</head>
<body>
<h1>JSP 大全</h1>
<p class="text">整本书都以实例贯穿其中，以<div class="text1">全面、循序渐进的方式</div>向读者介绍 JSP 各
类相关技术，并对一些重要知识进行重点和概括性总结，这样易于读者的阅读和理解。</p>
</body>
</html>

```

代码说明：该页面中的 CSS 定义了<h1>标签使用隶书字体类型，另外还自定义了 text 和 text1 两个 CSS 样式。该页面程序运行的效果如图 6.4 所示。

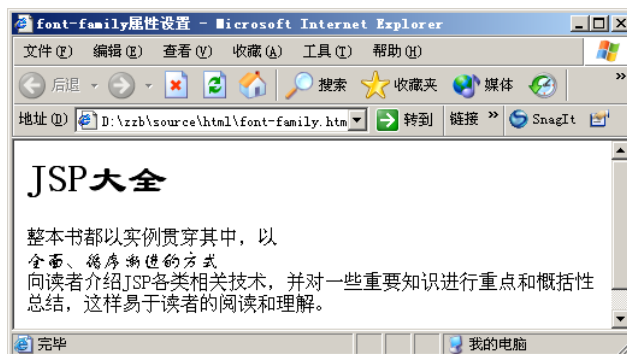


图 6.4 CSS 字体类型设置

6.3.1.2 字体大小

文本的字体大小是通过 font-size 属性来设置的，设定的字体大小既可以是绝对的也可以是相对的。Font-size 属性值的单位可以是厘米、像素或者磅等，也可以是一些其他值，例如：xx-small、x-small、smaller、x-large、xx-large 等。但是最为常用的还是 pt 单位。下面同样通过一个实例来演示，页面详细代码如下：

```

<html>
<head>
<title>CSS 的字体大小设置</title>
<style type="text/css">
H1 { font-size:15pt;}
.text { font-size:9pt;}
.text1 { font-size: 12pt;}
</style>
</head>
<body>
<h1>JSP 大全</h1>
<p class="text">整本书都以实例贯穿其中，以<div class="text1">全面、循序渐进的方式</div>向读者介绍 JSP 各
类相关技术，并对一些重要知识进行重点和概括性总结，这样易于读者的阅读和理解。</p>
</body>
</html>

```

代码说明：该段代码中使用 CSS 设置了<h1>标签中显示的文本字体大小为 15pt，自定义的名为 text 的 CSS 字体大小为 9pt，text1 为 12pt。该页面的显示效果如图 6.5 所示。

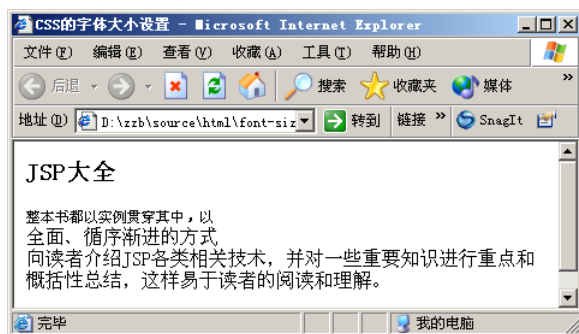


图 6.5 CSS 设置的字体大小

6.3.1.3 字体显示风格

通过 CSS 中的 `font-style` 属性可以来设置的文本字体显示风格，属性值可以如下三种：`normal`（正常显示）、`italic`（斜体显示）或者 `oblique`（倾斜显示，与 `italic` 很相似）。

下面通过实例来演示这三种属性值的不同，详细的页面代码如下：

```
<html>
<head>
<title>CSS 设置的字体风格</title>
<style type="text/css">
  H1 { font-style:italic;}
  .text { font-style:oblique;}
  .text1 { font-style:normal;}
</style>
</head>
<body>
<h1>JSP 大全</h1>
<p class="text">整本书都以实例贯穿其中，以全面、循序渐进的方式
向读者介绍 JSP 各类相关技术，并对一些重要知识进行重点和概括性总结，<div class="text1">这样易于读者的阅
读和理解。</div></p>
</body>
</html>
```

代码说明：设置了 `<h1>` 标签中的文本以斜体风格显示，定义的 `text` 为倾斜风格，另外 `text1` 以正常的风格显示。该页面的运行效果如图 6.6 所示。



图 6.6 CSS 定义的字体风格

6.3.1.4 设置字体的粗细

要对文本的字体进行加粗是通过 CSS 中的 `font-weight` 属性进行设置，该属性一般有如下四个属性

值：normal（正常显示文字）、bold（加粗显示的字体）、bolder（特粗显示的字体）和 lighter（加细显示的字体）。另外还可以通过数字大小来设置字体的粗细（例如 font-weight: 200）。实例代码如下：

```
<html>
<head>
<title>CSS 设置的字体风格</title>
<style type="text/css">
  H1 { font-size: 20pt;font-weight: bolder;}
  .text { font-weight: 200;}
  .text1 { font-weight:bold;}
</style>
</head>
<body>
<h1>JSP 大全</h1>
<p class="text">整本书都以实例贯穿其中，以全面、循序渐进的方式
向读者介绍 JSP 各类相关技术，并对一些重要知识进行重点和概括性总结，<div class="text1">这样易于读者的阅
读和理解。</div></p>
</body>
</html>
```

代码说明：该页面程序将<h1>标签中的文本字体设置为特粗，text 中是通过数值来设置字体粗细，text1 设置了字体显示的是粗体。该页面运行的效果如图 6.7 所示。

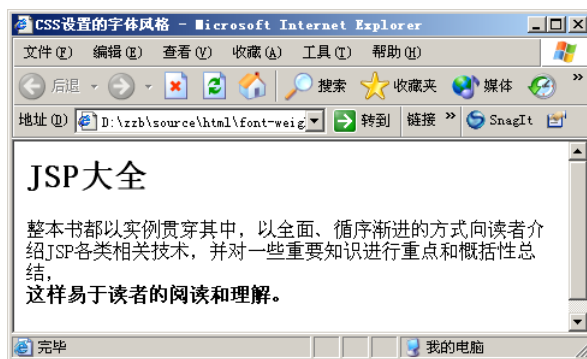


图 6.7 CSS 设置的字体粗细

6.3.2 文本属性

这里文本属性设置包括文字之间的间隔大小、文字修饰、文本排列、文本缩进以及行高等。详细见表 6.2 对各属性的描述。

表 6.2 文本属性

文本属性	描述
Letter-spacing	定义文本字符之间的间隔距离
Text-decoration	该属性设置文本是否有下划线、上划线或者删除线等
Text-align	设置文本的水平对齐方式，例如左对齐、右对齐、居中或者两端对齐
Text-indent	设置文本的首行缩进
Line-height	该属性允许设置文本基线之间的间隔距离

6.3.2.1 文字之间的间隔

Letter-spacing 属性用来设置文字之间的水平间距，该属性值并允许使用负值。

下面仍然通过一个实例来展示该属性的设置和效果，其详细的页面代码如下：

```
<html>
<head>
<title>CSS 设置的字体风格</title>
<style type="text/css">
  H1 { letter-spacing: -8px;}
  .text { letter-spacing: 5px;}
  .text1 { letter-spacing:15px;}
</style>
</head>
<body>
<h1>JSP 大全</h1>
<p class="text">整本书都以实例贯穿其中，以全面、循序渐进的方式
向读者介绍 JSP 各类相关技术，并对一些重要知识进行重点和概括性总结，<div class="text1">这样易于读者的阅
读和理解。</div></p>
</body>
</html>
```

代码说明：当 letter-spacing 属性值设置为负值时，这时文字之间会出现重叠，如图 6.8 所示的效果。

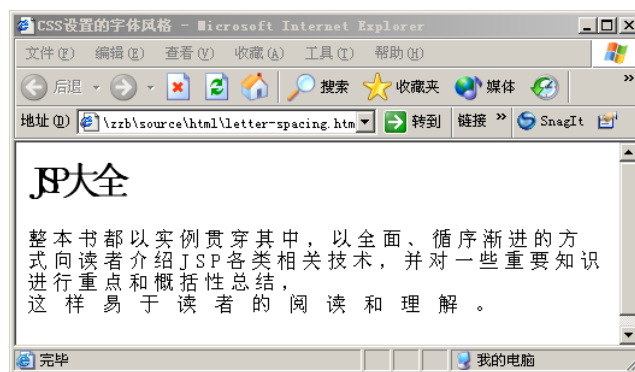


图 6.8 CSS 的 letter-spacing 属性设置

6.3.2.2 文字修饰

文字修饰是通过属性 text-decoration 来进行设置的，可以在文字下方设置下划线、上划线、中划线以及闪动效果（只对 NetScape 浏览器有效）。当需要同时设置两个属性时，只用在两个属性值之间用空格隔开就行。该属性的各个值的描述如表 6.3 所示。

表 6.3 文字修饰属性值

文字修饰属性值	描述
Underline	下划线
Overline	上划线
Line-through	中间线（删除线）
blink	闪烁，但是只在NetScape浏览器中才有效
none	无任何修饰

下面通过一个实例具体演示该属性的使用，详细代码如下：

```
<html>
<head>
<title>CSS 设置的字体风格</title>
<style type="text/css">
  H1 { text-decoration:line-through;}
</style>
```

```
.text { text-decoration:underline;}
.text1 { text-decoration:overline;}
</style>
</head>
<body>
<h1>JSP 大全</h1>
<p class="text">整本书都以实例贯穿其中，以全面、循序渐进的方式
向读者介绍 JSP 各类相关技术，并对一些重要知识进行重点和概括性总结，<br><div class="text1">这样易于读者的
的阅读和理解。</div></p>
</body>
</html>
```

代码说明：该页面程序的运行效果如图 6.9 所示。

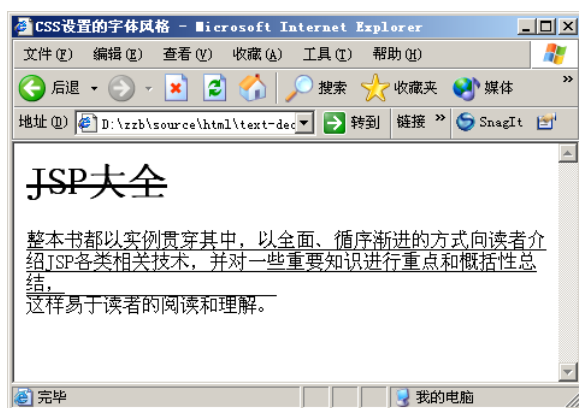


图 6.9 CSS 的 text-decoration 属性

6.3.2.3 文本的排列

对文本排列属性 text-align 的设置 CSS 中最为常见，其中包括：left（文本的左对齐）、right（右对齐）、（center）居中以及 justify（两端对齐）这些属性值的设置。实例程序代码如下：

```
<html>
<head>
<title>CSS 设置的字体风格</title>
<style type="text/css">
  H1 { text-align:center;}
  .text { text-align:justify;}
</style>
</head>
<body>
<h1>JSP 大全</h1>
<p class="text">整本书都以实例贯穿其中，以全面、循序渐进的方式
向读者介绍 JSP 各类相关技术，并对一些重要知识进行重点和概括性总结，这样易于读者的阅读和理解。</p>
</body>
</html>
```

代码说明：text-align 属性设置<h1>标签中的文本居中，下面的文本为两端对齐格式。该页面程序的运行效果如图 6.10 所示。

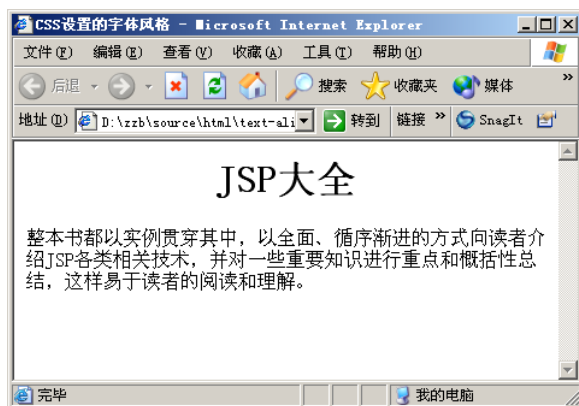


图 6.10 CSS 的 text-align 属性

6.3.2.4 文本缩进

文本缩进是通过 text-indent 属性来设置的，用来定义文本第一行缩进的数量。该属性值可以是一个百分比或者是一个长度，如果是百分比则视上一级元素的宽度而定。实例代码如下：

```
<html>
<head>
<title>CSS 设置的字体风格</title>
<style type="text/css">
  H1 { text-indent: 100pt;}
  .text { text-indent: 24pt;}
</style>
</head>
<body>
<h1>JSP 大全</h1>
<p class="text">整本书都以实例贯穿其中，以全面、循序渐进的方式
向读者介绍 JSP 各类相关技术，并对一些重要知识进行重点和概括性总结，这样易于读者的阅读和理解。</p>
</body>
</html>
```

代码说明：<h1>标签中的文本缩进为 100pt，text 设置了文本缩进为 24pt。该页面程序的运行效果如图 6.11 所示。



图 6.11 CSS 的文本缩进

6.3.2.5 行间距

行间距属性 Line-height 可以控制文本基线之间的距离。其属性值可以设置为百分比或者数字。实例代码如下：

```
<html>
<head>
<title>CSS 设置的字体风格</title>
<style type="text/css">
.text { text-indent: 24pt;}
</style>
</head>
<body>
<h1>JSP 大全</h1>
<p class="text">整本书都以实例贯穿其中，以全面、循序渐进的方式
向读者介绍 JSP 各类相关技术，并对一些重要知识进行重点和概括性总结，这样易于读者的阅读和理解。</p>
</body>
</html>
```

代码说明：使用 line-height 属性来设置文本之间行间距，如图 6.12 显示了运行效果。

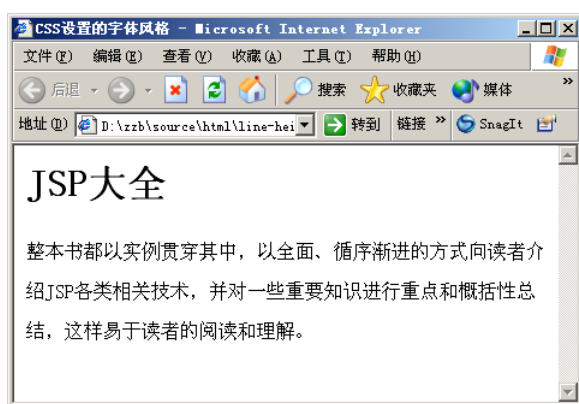


图 6.12 CSS 的行间距

6.3.3 颜色和背景属性

使用 CSS 可以对文本进行颜色设置，也可以设置页面的背景颜色或者图片。其中使用到的属性如表 6.4 所列举。

表 6.4 设置颜色和背景的属性

颜色和背景设置的属性	描述
Color	定义文字的显示颜色
Backgroup-color	设置某元素的背景颜色，例如页面背景
Backgroup-image	设置背景图片
Backgroup-repeat	设置背景图片是否重复显示
Backgroup-position	设置背景图片在页面中显示的位置

6.3.3.1 字体颜色的设置

CSS 中的 color 属性可以用来设置文本的显示颜色。具体使用方法如下：

```
<style type="text/css">
H1 {color: #336699}
.text {color: #ffcc00}
</style>
```

6.3.3.2 背景颜色的设置

用来设置背景颜色的属性为 `background-color`。具体实例代码如下：

```
<html>
<head>
<title>CSS 设置的字体风格</title>
<style type="text/css">
body { background-color:#336699;}
H1 { background-color: #ffffff;}
.text { background-color: #ffcc00;}
</style>
</head>
<body>
<h1>JSP 大全</h1>
<p class="text">整本书都以实例贯穿其中，以全面、循序渐进的方式
向读者介绍 JSP 各类相关技术，并对一些重要知识进行重点和概括性总结，这样易于读者的阅读和理解。</p>
</body>
</html>
```

代码说明：图 6.13 展示了以上代码的显示效果。

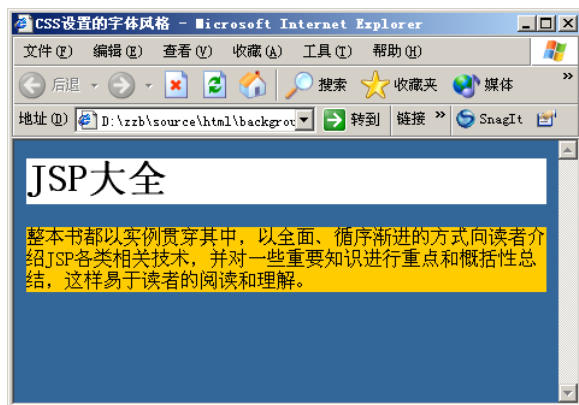


图 6.13 CSS 背景颜色

6.3.3.3 背景图片的设置

使用 `background-image` 属性来设置一个元素的背景图片。列举的实例代码如下：

```
<html>
<head>
<title>CSS 设置背景图片</title>
<style type="text/css">
body { background-image:url(images/bg.gif);}
</style>
</head>
<body>
<h1>JSP 大全</h1>
<p class="text">整本书都以实例贯穿其中，以全面、循序渐进的方式
向读者介绍 JSP 各类相关技术，并对一些重要知识进行重点和概括性总结，这样易于读者的阅读和理解。</p>
</body>
</html>
```

代码说明：背景图片设置的效果如图 6.14 所示。



图 6.14 CSS 的背景图片设置

6.3.3.4 背景图片重复

Background-repeat 属性一般和 background-image 属性配套使用, 它用来设置当图片大小不能占据整个页面时, 是否可重复。该属性各值的含义如表 6.5 所示。

表 6.5 背景图片重复属性值

背景图片重复显示的属性值	描述
Repeat-x	图片可以在横向方向上重复
Repeat-y	图片可以在纵向方向上重复显示
Repeat	图片可以在横向纵向上重复
No-repeat	设置图片不可以重复。

6.3.3.5 背景图片的位置

当背景图片不能完全占据整个页面时, 就存在图片显示位置的问题。CSS 就是通过 background-position 属性来指定背景图片的位置。属性值包括: left (左端)、right (居右端)、center (水平居中)、top (上对齐)、bottom (下对齐) 和 middle (垂直居中)。

6.3.4 边框属性的设置

下面介绍的各个属性是用来设置一个元素的边框, 例如边框的宽度、样式和颜色等。其中涉及到的属性见表 6.6 所示。

表 6.6 边框属性

边框属性	描述
Border	边框 (包括四条边) 的属性设置
Border-top	上边框属性设置
Border-left	左边框属性设置
Border-right	右边框属性设置
Border-bottom	下边框属性设置

Border 属性是用来设置一个元素的所有边框样式, 而其他四个属性用来设置其中的某一条边框。这些属性的值表示的含义如表 6.7 所示。

表 6.7 边框各属性值

边框各属性值	描述
None	无边框，即边框不可见
Dotted	边框由点组成
Dash	边框由短线组成
Solid	边框由实线组成
Double	边框由两条双实线组成
Groove	边框带有立体感的凹槽组成
Ridge	边框由脊形线条组成
Inset	边框是由一个内嵌的立体边框组成
Outset	边框外嵌一个立体边框

接下来仍然通过一个实例来演示这些属性的使用，详细代码如下：

```
<html>
<head>
<title>CSS 设置背景图片</title>
<style type="text/css">
p { border-top: 4px double #990000;
    border-right: 4px dotted #3399FF;
    border-bottom: 4px solid #00FF33;
    border-left: 4px inset #CC33FF;}
</style>
</head>
<body>
<h1>JSP 大全</h1>
<p class="text">整本书都以实例贯穿其中，以全面、循序渐进的方式
向读者介绍 JSP 各类相关技术，并对一些重要知识进行重点和概括性总结，这样易于读者的阅读和理解。</p>
</body>
</html>
```

代码说明：使用 CSS 设置该文本元素的四个边框，效果如图 6.15 所示。

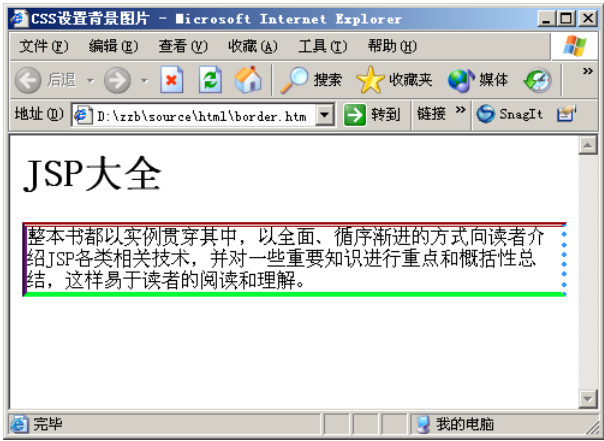


图 6.15 CSS 边框属性设置

6.3.5 鼠标光标属性的设置

网页仅仅默认两种鼠标形式，但是使用了 CSS 后，就可以通过属性来设置多种鼠标形式。这是通过 Cursor 属性来实现的，当对某个元素设置该属性后，鼠标放置在该元素区域之上就会显示相应的形状。Cursor 属性值所包括的含义如表 6.8 所示。

表 6.8 鼠标光标属性值

鼠标光标的属性值	描述
Hand	手型鼠标状态
Crosshair	交叉十字型
Text	文本选择符号
Wait	Windows的漏沙形状
Default	默认的鼠标形状
Help	带问号的鼠标
e-resize	向东的箭头
n-resize	向北的箭头
Ne-resize	指向东北方向的箭头
Nw-resize	指向西北方向的箭头
w-resize	向西的箭头
Sw-resize	指向西南方向的箭头
s-resize	向南的箭头
Se-resize	指向东南方向的箭头

下面实例来演示该属性的使用，详细代码如下：

```
<html>
<head>
<title>CSS 设置背景图片</title>
<style type="text/css">
body { cursor:crosshair;}
img { cursor:help;}
</style>
</head>
<body>
<h1>JSP 大全</h1>

<p>整本书都以实例贯穿其中，以全面、循序渐进的方式
向读者介绍 JSP 各类相关技术，并对一些重要知识进行重点和概括性总结，这样易于读者的阅读和理解。</p>
</body>
</html>
```

代码说明：当鼠标放置在图片区域时，鼠标会显示带问号；当鼠标放置在其他区域，则显示十字形状，如图 6.16 和 6.17 所示。



图 6.16 鼠标成十字形状



图 6.17 鼠标成带问号形状

6.3.6 定位属性

这一小节将要介绍的属性是用来控制页面元素的位置和大小等。其中提供了两种定位方式，一种是绝对位置，另一个是相对位置。所谓绝对位置即为参照原始文档进行的偏移量；而相对位置即为参照于其他元素而进行的偏移。有关定位方面的属性见表 6.9 所示。

图 6.9 定位属性

定位属性	描述
Position	指定元素使用绝对定位还是相对定位，属性值： absolute （绝对）、 relative （相对）
Top	元素距离页面顶部的距离
Left	元素距离页面左侧的距离
Width	元素的宽度
Height	元素的高度
Z-index	指定元素（或者叫层）之间先后顺序或者覆盖关系，值高就会覆盖值低的元素（或层）
Clip	限定只显示裁切出来的区域，该区域为矩形，可以使用两个点来决定其位置
overflow	当元素中的内容超出元素所设定的范围时： Visible: 元素的大小和内容都可以显示出来 Hidden: 超出元素范围的内容会隐藏 Scroll: 不管内容是否超出范围，都设定一个滚动条 Auto: 当内容超出元素范围时，才显示滚动条
visibility	该属性是针对元素嵌套使用的，各属性值包含的意思如下： Inherit: 嵌套元素（子元素）继承被嵌套元素（父元素），父元素可见，子元素也可见 Visible: 无论父元素是否可见，子元素都可见 Hidden: 无论父元素是否可见，子元素都不可见

下面还是通过一个实例来介绍该属性的使用，代码如下：

```
<html>
<head>
<title>CSS 设置背景图片</title>
<style type="text/css">
.text {
position:absolute;
top:90px;
left:50px;
width:300px;
height:100px;
overflow:scroll;
background-color:#336699;
color:#FFFFFF;
z-index:1;
visibility:visible;
}
</style>
</head>
<body>
<h1>JSP 大全</h1>
<p class="text">整本书都以实例贯穿其中，以全面、循序渐进的方式
向读者介绍 JSP 各类相关技术，并对一些重要知识进行重点和概括性总结，这样易于读者的阅读和理解。</p>
</body>
</html>
```

代码说明：通过上面介绍的多个属性来设置本页面中的文本元素（由<p>标签来标识），效果如图

6.18 所示。

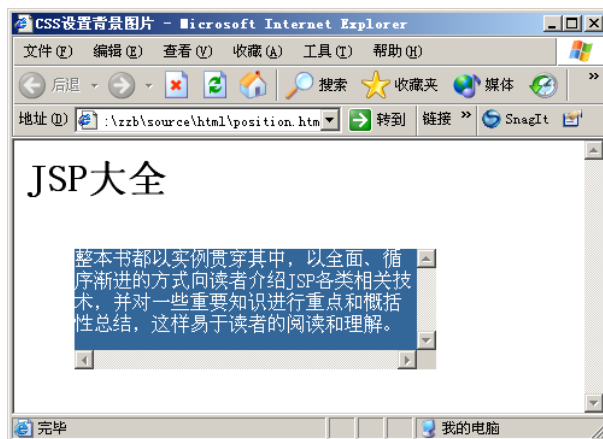


图 6.18 CSS 定位属性

6.3.6 区块属性

这里首先介绍一下什么叫着“区块”，当 CSS 将所有对象都放置一个容器里，这个容器就称为区块。其可以设置的属性见表 6.10 所示。

表 6.10 区块属性

区块属性	描述
Width	设置某一区块对象的宽度
Height	设置一区块对象的高度
Float	让区块中的文字环绕在四周
Clear	指定在某一个元素的某一边是否可以允许有文字或者对象环绕
Padding	设置区块的边框和文字内容之间可以插入多少空间距离，它共有四个属性：Padding-top、padding-bottom、padding-left、padding-right，分别用于设置上下左右的填充距离
Margin	设置该区域对象距离浏览器窗口的四边距离，它同样有四个属性：Margin-top、margin-bottom、margin-left、margin-right。

同样，下面列举一个实例来介绍，代码如下：

```
<html>
<head>
<title>CSS 设置背景图片</title>
<style type="text/css">
.text {
width:300px;
height:100px;
margin-top:20px;margin-right:30px;margin-bottom:20px;margin-left:30px;
}
</style>
</head>
<body>
<h1>JSP 大全</h1>
<p class="text">整本书都以实例贯穿其中，以全面、循序渐进的方式
向读者介绍 JSP 各类相关技术，并对一些重要知识进行重点和概括性总结，这样易于读者的阅读和理解。</p>
</body>
</html>
```

代码说明：该页面程序的运行效果如图 6.19 所示。

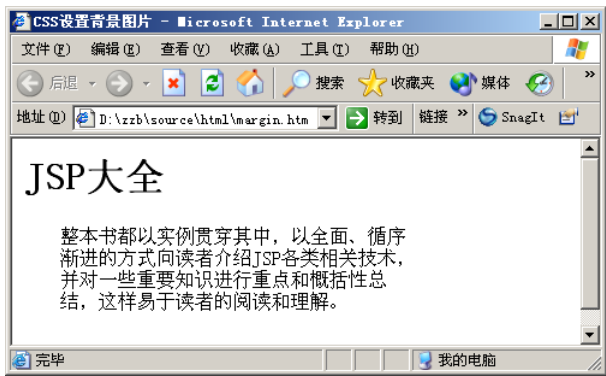


图 6.19 CSS 区块属性

6.3.7 列表属性

在 HTML 章节中，已经介绍有关列表的标签，这里通过设置 CSS 的列表属性可以丰富列表的外观。其详细属性如表 6.11 所示。

表 6.11 列表属性

列表属性	描述
List-style-type	设置引导列表项的符号类型（可以英文字母或者罗马数字等）
List-style-image	可以指定一个图片来引导列表项
List-style-position	指定各列表项的缩进程度：Outside：列表贴近左侧边框；Inside：列表缩进

针对 list-style-type 属性，可以设置多个属性，也就是可以指定多种符号类型来引导列表项。各属性值的含义如表 6.12 所示。

表 6.12 列表符号类型属性值

属性值所指定的符号类型	描述
Disc	以“●”符号作为列表项的引导放置在前面
Circle	以“○”符号作为列表项的引导
Square	以“■”符号作为列表项的引导
Decimal	以普通的阿拉伯数字作为列表项的引导
Lower-roman	以小写罗马数字作为引导
Upper-roman	以大写罗马数字作为引导
Lower-alpha	以小写英文字母作为引导
Upper-alpha	以大写英文字母作为引导
none	不显示任何项目符号

实例代码如下：

```
<html>
<head>
<title>CSS 设置背景图片</title>
<style type="text/css">
ol {
list-style-type:upper-roman;
list-style-position:outside;
}
</style>
</head>
```

```
<body>
<h1>JSP 大全</h1>
<ol>
<li>JSP</li>
<li>Tomcat</li>
<li>Eclipse</li>
<li>Lomboz</li>
</ol>
</body>
</html>
```

代码说明：使用 list-style-type 属性指定了以大写罗马数字作为列表项的引导，另外 list-style-position 属性指定列表贴近左侧边框。效果如图 6.20 所示。

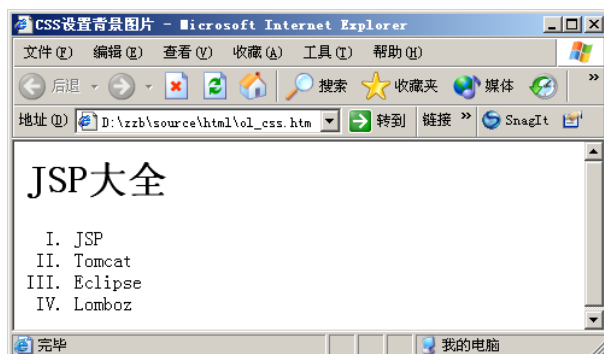


图 6.20 列表属性

至此，基本的 CSS 使用和设置方法已经介绍了。至于其他一些属性（例如滤镜属性等）的设置，由于使用比较少，本章就不一一具体介绍。

6.4 本章小结

本章重点介绍了层叠样式表（CSS）的定义和引用方法。针对各个 CSS 属性，作了详细讲解，并且穿插了实例进行演示。JavaScript 和 CSS 带来了 DHTML 技术的发展，下面将要介绍的 Ajax 也将使用到这两个技术。