

第 2 章 CSS 与 CSS3 基础

本章要点

- CSS 的基本概念
- CSS 与 HTML 结合
- CSS 选择器
- CSS3 简介

2.1 CSS 的基本概念和特点

Internet 是当今世界上最大的计算机网络，它将全球成千上万的计算机网络和数量众多的计算机主机有机地结合在一起，形成了一个全球信息网。目前 Internet 上可以提供的服务种类非常多，如远程登录(Telnet)、电子邮件(E-mail)、文件传输(FTP)、万维网(World Wide Web)等，其中，Web 和 E-mail 是最常用的服务。

ASP 与 Internet 上的 Web 服务有着密切的关系。为了真正理解 ASP 的工作机制，首先需要了解 Web 的一些基本知识。

2.1.1 CSS 的基本概念

CSS 是 Cascading Style Sheets(层叠样式表)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。借助于 CSS 的强大功能，网页将在设计者丰富的想象力下千变万化。CSS 可以更精确地控制页面的版式风格和布局，它将弥补 HTML 对网页格式化的不足。利用 CSS 可以设置字体变化和大小、页面格式的动态更新和排版定位等。

自从 1998 年 5 月 12 日 W3C 组织推出 CSS2 以来，这项技术在世界范围内得到广泛的支持。CSS2 成为 W3C 的新标准。样式可以定义在 HTML 文件的标记里，也可以定义在外加的文件中。当样式表定义在外部文件中时，一个样式表可以用于多个页面甚至整个网站，因此具有更好的易用性和扩展性。

总体来说，CSS 可以完成以下工作。

- 弥补 HTML 对网页格式化功能的不足，如段落间距、行距等。
- 设置字体变化和大小。
- 设置页面格式的动态更新。
- 进行排版定位。

2.1.2 CSS 的特点

CSS 具有以下特点。

(1) 将格式和结构分离。

HTML 定义了网页的结构和各要素的功能，而 CSS 通过将定义结构的部分和定义格式的部分分离，使用户能够对页面的布局施加更多的控制。HTML 仍可以保持简单明了的初衷，CSS 代码独立出来，从另一个角度控制页面外观。

(2) 控制页面布局。

HTML 对页面总体上的控制很有限。像精确定位、行间距或字间距等任务都可以通过 CSS 来完成。

(3) 制作体积更小且下载更快的网页。

样式表只是简单的文本，就像 HTML 那样。它不需要图像、执行程序及插件。使用 CSS 可以减少表格标签及其他加大 HTML 体积的代码，还可以减少图像数量从而减小文件的大小。

(4) 更新速度更快。

没有 CSS 时，如果想更新整个站点中所有主体文本的字体，必须一页一页地修改每个网页文件。CSS 的主旨就是将格式和结构分离。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，因此只要修改 CSS 文件中的某一行，那么整个站点都会随之发生改变。

(5) 更有利于搜索引擎的搜索。

CSS 减少了代码量，使得正文更加突出，有利于搜索引擎更有效地搜索到 Web 页面。

2.2 CSS 选择器

CSS 由一系列的样式规则构成，样式规则具体定义和控制网页文档的显示方式。每个规则由一个“选择器”(Selector)和一个定义部分组成。每个定义部分包含一组由半角分号(;)分离的定义。这组定义放在一对大括号{}之间。每个定义由一个特性、一个半角冒号(:)和一个值组成。

2.2.1 CSS 样式规则的定义

CSS 样式规则的定义格式为：

```
选择器 { 属性 1: 属性值 1; 属性 2: 属性值 2; ... }
```

说明:

选择器用于指定样式所作用的对象。选择器可以是 HTML 标记,也可以是一个类名。大括号中的部分用于定义具体样式的规则,它由若干组属性名和相应的属性值构成,各组间用分号分隔,属性名与对应的属性值间用冒号分隔。

例如,若要定义 h1 的字体为黑体,字体大小为 20pt,颜色为红色,则该种样式的定义方法为:

```
h1 { font-family:黑体; font-size:20pt; color:red }
```

2.2.2 标记选择器

一个 HTML 页面由很多的标记组成,而 CSS 标记选择器就是声明哪些标记采用哪种 CSS 样式。例如 p 选择器就是用于声明页面中所有<p>标记的样式风格。同样可以通过 h1 选择器来声明页面中所有的<h1>标记的 CSS 风格。例如:

```
<style>
p {
color:red;
font-size:12px;
}
</style>
```

以上这段代码声明了 HTML 页面所有的<p>标记,文字的颜色都采用红色,大小都为 12px。每一个 CSS 选择器都包含了选择器本身、属性和值,其中属性和值可以设置多个,从而实现对同一个标记声明多种样式风格。

在网站的后期维护中,如果希望所有<p>标记不再采用红色,而是采用蓝色,这时仅仅需要将属性 color 的值修改为 blue 即可。

2.2.3 CSS 选择器分组

可以对选择器进行分组,这样,被分组的选择器就可以分享相同的声明。可用逗号将需要分组的选择器分开。

例如,在下面的例子中,对所有的标题元素进行了分组,所有的标题元素都是绿色的:

```
h1,h2,h3,h4,h5,h6 {
color: green;
}
```

2.2.4 派生选择器

通过依据元素在其位置的上下文关系来定义样式,可以使标记更加简洁。通过这种方

式来应用规则的选择器称为派生选择器。

派生选择器允许根据文档的上下文关系来确定某个标签的样式。通过合理地使用派生选择器，可以使 HTML 代码变得更加整洁。

例如，我们希望列表中的 **strong** 元素变为斜体字，而不是通常的粗体字，可以这样定义一个派生选择器：

```
li strong {
font-style: italic;
font-weight: normal;
}
<p><strong>我是粗体字，不是斜体字，因为我不在列表当中，所以这个规则对我不起作用
</strong></p>
<ol>
<li><strong>我是斜体字。这是因为 strong 元素位于 li 元素内。</strong></li>
<li>我是正常的字体。</li>
</ol>
```

在上面的例子中，只有 li 元素中的 strong 元素的样式为斜体字，无需为 strong 元素定义特别的 class 或 id，代码更加简洁。

2.2.5 id 选择器

id 选择器可以为标有特定 id 的 HTML 元素指定特定的样式。id 选择器以“#”来定义。

下面的两个 id 选择器，第一个可以定义元素的颜色为红色，第二个定义元素的颜色为绿色：

```
#red {color:red;}
#green {color:green;}
```

下面的 HTML 代码中，id 属性为 red 的 p 元素显示为红色，而 id 属性为 green 的 p 元素显示为绿色：

```
<p id="red">这个段落是红色的。</p>
<p id="green">这个段落是绿色的。</p>
```

注意：id 属性只能在每个 HTML 文档中出现一次。

在现代布局中，id 选择器常常用于建立派生选择器：

```
#sidebar p {
font-style: italic;
text-align: right;
margin-top: 0.5em;
}
```

上面的样式只会应用于 id 是 sidebar 的元素内的段落。这个元素很可能是 div 或者是表格单元，尽管它也可能是一个表格或者其他块级元素。

2.2.6 类选择器

在 CSS 中，类选择器以一个点号显示。例如：

```
.center {text-align: center}
```

在上面的例子中，所有拥有 center 类的 HTML 元素均为居中。

在下面的 HTML 代码中，h1 和 p 元素都有 center 类，这意味着两者都将遵守“.center”选择器中的规则：

```
<h1 class="center">
This heading will be center-aligned.
</h1>
<p class="center">
This paragraph will also be center-aligned.
</p>
```

注意：类名的第一个字符不能使用数字！它无法在 Mozilla Firefox 浏览器(也称火狐浏览器)中起作用。

与 id 一样，class 也可被用作派生选择器：

```
.fancy td {
color: #f60;
background: #666;
}
```

在上面的这个例子中，类名为 fancy 的元素内部的表格单元都会以灰色背景来显示橙色的文字。

2.2.7 伪类选择器

CSS 伪类用于向某些选择器添加特殊的效果。

语法如下：

```
selector: pseudo-class {property: value}
```

CSS 类也可与伪类搭配使用：

```
selector.class : pseudo-class {property: value}
```

最常用的伪类是锚伪类。在支持 CSS 的浏览器中，链接的不同状态都可以不同的方式显示，这些状态包括活动状态、已被访问状态、未被访问状态和鼠标悬停状态：

```
a:link {color: #FF0000} /* 未访问的链接 */
a:visited {color: #00FF00} /* 已访问的链接 */
a:hover {color: #FF00FF} /* 鼠标移动到链接上 */
a:active {color: #0000FF} /* 选定的链接 */
```

注意:在 CSS 定义中, a:hover 必须置于 a:link 和 a:visited 之后, a:active 必须置于 a:hover 之后, 才是有效的。

2.3 CSS 的引用

当读到一个样式表时, 浏览器会根据它来格式化 HTML 文档。插入样式表的方法有如下几种。

2.3.1 内部样式表

当单个文档需要特殊的样式时, 就应该使用内部样式表。可以使用<style>标签在文档头部定义内部样式表。

【例 2-1】使用内部样式表。源文件(char2\2-1.html)的代码如下:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>第一个使用了 CSS 的 HTML 文件</title>
<style type="text/css">
<!--
h2 { color: green; font-size: 37px; font-family: 黑体 }
P { text-indent: 1cm; background: yellow; font-family: 宋体 }
-->
</style>
</head>
<body>
<h2 align="center">第一个使用了 CSS 的 HTML 文件</h2>
<hr>
<p>这是第一个使用了 CSS 的 HTML 网页文件。</p>
</body>
</html>
```

此文件在浏览器中的预览效果如图 2-1 所示。

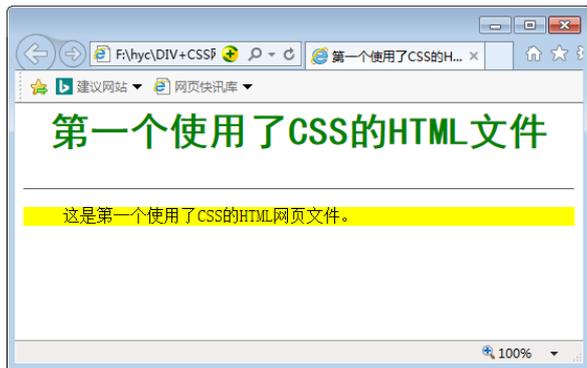


图 2-1 引用 CSS 样式的效果

2.3.2 外部样式表

当样式需要应用于很多页面时，外部样式表将是理想的选择。在使用外部样式表的情况下，可以通过改变一个文件来改变整个站点的外观。每个页面使用<link>标签链接到样式表。<link>标签写在文档的头部。例如：

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

浏览器会从文件 mystyle.css 中读到样式声明，并根据它来格式文档。

外部样式表可以在任何文本编辑器中进行编辑。文件不能包含任何 HTML 标签。样式表应该以.css 扩展名保存。下面是一个样式表文件的例子：

```
hr {color: sienna;}
p {margin-left: 20px;}
body {background-image: url("images/back40.gif");}
```

注意：不要在属性值与单位之间留有空格。假如你使用“margin-left: 20 px”而不是“margin-left: 20px”，它仅在 IE6 浏览器中有效，但是在 Mozilla Firefox 或 Netscape 浏览器中却无法正常工作。

【例 2-2】使用外部样式表。源文件(char2\2-2.html 和 mystyle.css)的代码如下：

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"><title>外部样式表的使用</title>
<link rel="stylesheet" type="text/css" href="mystyle.css" />
</head>
<body>
<h2 align="center">使用了 CSS 的 HTML 文件</h2>
```

```

<hr />
<p>这是使用了 CSS 外部样式表的 HTML 网页文件。</p>
</body>
</html>

```

此文件在浏览器中的预览效果如图 2-2 所示。



图 2-2 引用 CSS 外部样式的效果

2.3.3 内联样式表

要使用内联样式，需要在相关的标签内使用样式(Style)属性。Style 属性可以包含任何 CSS 属性。例如，使用内联样式改变段落的颜色和左外边距：

```

<p style="color: sienna; margin-left: 20px">
This is a paragraph.
</p>

```

【例 2-3】使用内联样式表。源文件(char2\2-3.html)的代码如下：

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>内联样式表的使用</title>
</head>
<body>
<h2 align="center">使用了 CSS 的 HTML 文件</h2>
<p>这是使用了 CSS 内联样式表的 HTML 网页文件。</p>
<p style="color: sienna; margin-left: 20px">
这是使用了 CSS 内联样式表的 HTML 网页文件。</p>

```

```
<P>这是使用了 CSS 内联样式表的 HTML 网页文件。</p>
</body>
</html>
```

此文件在浏览器中的预览效果如图 2-3 所示。



图 2-3 内联样式的使用效果

由于要将表现和内容混杂在一起，内联样式会损失掉样式表的许多优势。要慎用这种方法，一般当样式仅需要在一个元素上应用一次时使用内联样式。

2.3.4 输入样式表

输入样式表的方法同外部样式表的使用方法类似。不同之处在于外部样式表不能同其他方法结合使用，但输入样式表则可以，例如：

```
<style type="text/css">
<!--
@import url(style1.css);
h2 { color: green; font-size: 37px; font-family: 黑体 }
p { text-indent: 1cm; background: yellow; font-family: 宋体 }
-->
</style>
```

在本例中，浏览器首先输入 style1.css 样式(@import 必须打头)，然后加入移植的样式，从而为这个网页产生样式集合。

【例 2-4】使用输入样式表。源文件(char2\2-3.html 和 mystyle.css)的代码如下：

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
```

```

<title>输入样式表的使用</title>
<style type="text/css">
<!--
@import url(mystyle.css);
h2 { color: green; font-size: 37px; font-family: 黑体 }
p { text-indent: 1cm; background: yellow; font-family: 宋体 }
-->
</style>
</head>
<body>
<h2 align="center">使用了 CSS 的 HTML 文件</h2>
<p>这是使用了 CSS 输入样式表的 HTML 网页文件。</p>
<p>这是使用了 CSS 输入样式表的 HTML 网页文件。</p>
<p>这是使用了 CSS 输入样式表的 HTML 网页文件。</p>
</body>
</html>

```

此文件在浏览器中的预览效果如图 2-4 所示。

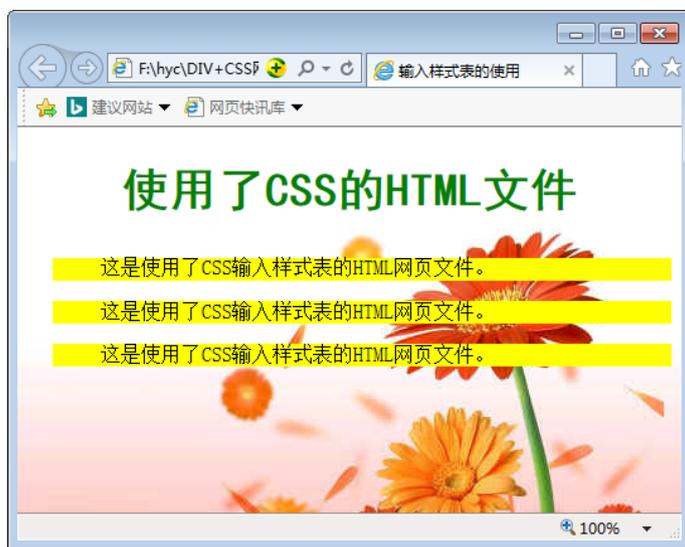


图 2-4 输入样式表的使用效果

在这个例子中，网页背景的效果和段落的左边距分别来自于 `mystyle.css` 文件中的 `p` 选择器和 `body` 选择器中设置的样式。

2.3.5 多重样式表

如果某些属性在不同的样式表中被同样的选择器定义，那么属性值将从更具体的样式表中被继承过来。

例如，外部样式表拥有针对 `h3` 选择器的 3 个属性：

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

而内部样式表拥有针对 h3 选择器的两个属性:

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

假如拥有内部样式表的这个页面同时与外部样式表链接,那么 h3 得到的样式是:

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

即颜色属性将继承外部样式表,而文字排列(text-alignment)和字体尺寸(font-size)会被内部样式表中的规则取代。

【例 2-5】使用多重样式表。源文件(char2\2-5.html 和 style1.css)的代码如下:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>多重样式表的使用</title>
<link rel="stylesheet" type="text/css" href="style1.css" />
<style type="text/css">
<!--
h3 {
text-align: center;
font-size: 20pt;
}
-->
</style>
</head>
<body>
<h3>多重样式</h3>
<h3>多重样式</h3>
<h3>多重样式</h3>
</body>
</html>
```

此文件在浏览器中的预览效果如图 2-5 所示。

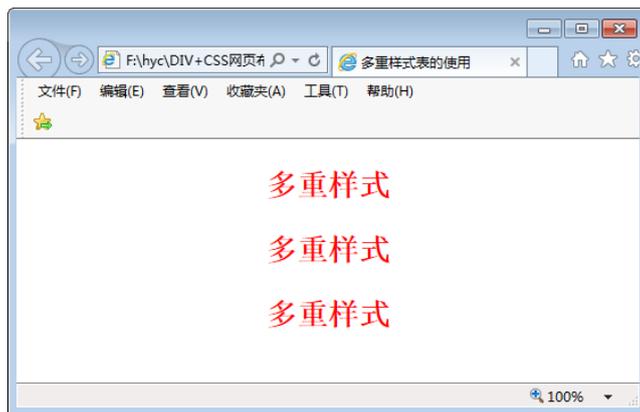


图 2-5 多重样式的使用效果

如果在网页的<head>标记中同时使用了 style 标记(指定嵌入式样式)和 link 标记(指定链接式样式), 并且这两个样式指定中同时应用了具有同一优先级别的样式, 则 style 标记和 link 标记的先后顺序将决定样式的优先级。

2.4 CSS 的继承

CSS 的继承是指被包在内部的标签将拥有外部标签的样式性质。继承特性最典型的应用通常发挥在整个网页的样式预设, 即整体布局声明。而需要指定为其他样式的部分设定在个别元素里即可达到效果。这项特性可以给网页设计者提供更理想的发挥空间。但同时继承也有很多规则, 应用的时候容易让人迷惑。

2.4.1 CSS 的继承关系

CSS 的一个主要特征就是继承, 它是依赖于“祖先-后代”的关系的。继承是一种机制, 它允许样式不仅可以应用于某个特定的元素, 还可以应用于它的后代。HTML 文件的组织结构如图 2-6 所示。

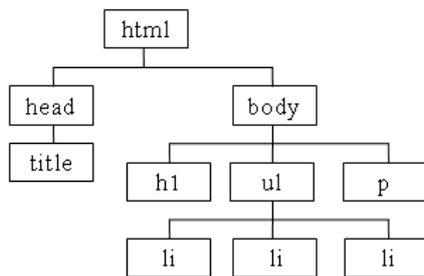


图 2-6 “祖先-后代”的关系

【例 2-6】CSS 的继承。源文件(char2\2-6.html)的代码如下:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>CSS 继承</title>
<style type="text/css">
<!--
body {color:blue; font-size:18pt;}
-->
</style>
</head>

<body>
<p>CSS<strong>继承性</strong>, 段落继承了 body 的样式属性(颜色和字号)</p>
</body>
</html>
```

例 2-6 是一个很简单的 HTML 文档, 在这个文档的 body 中定义的颜色值和字体大小也会应用到段落的文本中, 如图 2-7 所示。

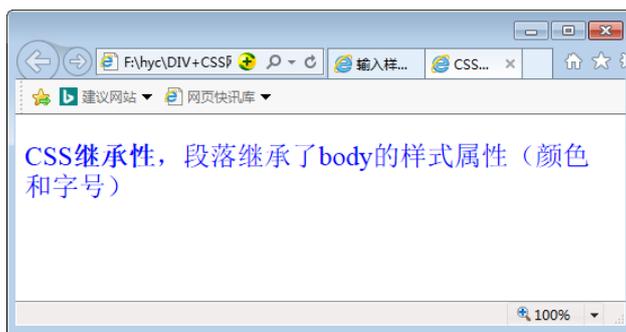


图 2-7 CSS 的继承效果

2.4.2 CSS 继承的应用

在实际工作中, 我们编写代码, 往往在 CSS 文档的最前部, 首先定义:

```
*{margin: 0; padding: 0; border: 0;}
```

这些代码的真正用意在于, 在缺省定义的情况下, 所有元素的 margin、padding、border 的值都为零。当需要应用不同样式的时候, 再单独地对某元素进行定义即可。它就是整个网页的样式预设、整体布局声明。而需要指定为其他样式的部分设定在个别元素里即可达到效果。

在 CSS 中, 继承是一种非常自然的行为, 我们甚至不需要考虑是否能够这样做, 但

是继承也有其局限性。

有些属性是不能继承的。这没有任何原因，只是因为它就是这么设置的，标准就是如此。举个例子来说：`border`属性的作用是设置元素的边框，它没有继承性。如果继承了边框属性，那么文档看起来就会很奇怪。例如我们定义容器 `div` 的边框为 `1px`，而在此容器内的 `ul li` 在正常情况下我们都不希望它有边框，如果 `border` 有继承性，我们就要再去掉它们的边框。这样显然是不合理的。

多数边框类的属性，如 `border`(边框)、`padding`(补白)、`margin`(边界)、背景等，都是没有继承性的。

在某些时候，继承也会带来一些错误，比如下面这条 CSS 定义：

```
body {color:blue}
```

这是定义了 `body` 中的文本颜色为蓝色。如果 `body` 中含有表格，在有些浏览器中这句定义会使除表格之外的文本变成蓝色，而表格内部的文本颜色并不是蓝色。从技术上来说，这是不正确的，但是它确实存在。

所以我们经常需要借助于某些技巧，比如将 CSS 定义成这样：

```
body,table,th,td {color:blue}
```

这样表格内的文字也会变成蓝色了。

2.5 CSS3 简介

CSS3 是 CSS 技术的升级版，CSS3 语言开发是朝着模块化发展的。以前的规范作为一个模块实在是太庞大而且比较复杂，所以，把它分解为一些小的模块，更多新的模块也被加入进来。这些模块包括：盒子模型、列表模块、超链接方式、语言模块、背景和边框、文字特效、多栏布局等。

2.5.1 CSS3 的边框

CSS3 的边框有更大的灵活性，可以创建圆角边框、阴影边框、图像边框，还可以控制边框颜色，产生渐变效果。新增的边框属性有：

(1) `border-radius`：圆角属性。可以给任何元素制作“圆角”。`border-radius` 属性中只指定一个值，那么将生成 4 个圆角。

如果要在四个角上一一指定，可以使用以下规则。

四个值：第一个值为左上角，第二个值为右上角，第三个值为右下角，第四个值为左下角。

三个值：第一个值为左上角，第二个值为右上角和左下角，第三个值为右下角。

两个值：第一个值为左上角与右下角，第二个值为右上角与左下角。

一个值：四个圆角值相同。

(2) `box-shadow`：用来添加阴影的属性。

(3) `border-image`：使用图像创建一个边框的属性。

【例 2-7】 创建圆角边框和阴影边框。源文件(char2\2-7.html)的代码如下：

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>CSS3 边框</title>
<style>
.div1{
    border:2px solid #999;
    padding:10px 40px;
    margin:10px;
    background:#ddd;
    width:300px;
    height:50px;
    border-radius:25px;
}
.div2{
    border:2px solid #999;
    padding:10px 40px;
    margin:10px;
    background:#ddd;
    width:300px;
    height:50px;
    box-shadow: 10px 10px 5px #999;
}
</style>
</head>
<body>
<div class="div1">border-radius 属性设置圆角边框! </div>
<div class="div2">box-shadow 属性设置带阴影边框! </div>
</body>
</html>
```

该例题设置了一个圆角边框和阴影边框，如图 2-8 所示。



图 2-8 CSS3 边框效果

2.5.2 CSS3 的渐变

CSS3 的渐变(gradients)可以在两个或多个指定的颜色之间显示平稳的过渡。以前, 必须使用图像来实现这些效果。现在, 通过使用 CSS3 实现渐变(gradients)。这样可以减少下载的事件和宽带的使用。此外, 渐变效果的元素在放大时看起来效果更好, 因为渐变(gradient)是由浏览器生成的。

CSS3 定义了两种类型的渐变(gradients):

线性渐变(Linear Gradients): 向下/向上/向左/向右/对角方向。

径向渐变(Radial Gradients): 由它们的中心定义。

【例 2-8】创建 CSS3 线性渐变效果。源文件(char2\2-8.html)的代码如下:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>CSS3 线性渐变效果</title>
<style>
p{ font-size: 12px;
}
#grad1 {
    height: 100px;
    background: linear-gradient(white, gray);
}
#grad2 {
    height: 100px;
    background: linear-gradient(to right, white, gray);
}
#grad3 {
```

```
height: 100px;
background: linear-gradient(to bottom right, white, gray);
}
</style>
</head>
<body>
<h3>线性渐变 - 从上到下</h3>
<p>从顶部开始的线性渐变。起点是白色，慢慢过渡到灰色： </p>
<div id="grad1"></div>
<h3>线性渐变 - 从左到右</h3>
<p>从左边开始的线性渐变。起点是白色，慢慢过渡到灰色： </p>
<div id="grad2"></div>
<h3>线性渐变 - 对角</h3>
<p>从左上角开始(到右下角)的线性渐变。起点是白色，慢慢过渡到灰色： </p>
<div id="grad3"></div>
</body>
</html>
```

该例题设置了三个线性渐变，如图 2-9 所示。



图 2-9 CSS3 线性渐变效果

【例 2-9】创建 CSS3 径向渐变效果。源文件(char2\2-9.html)的代码如下：

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>径向渐变</title>
<style>
p{ font-size: 8pt;
}
#grad {
    height: 100px;
    background: radial-gradient(red, green, blue);
}
</style>
</head>
<body>
<h5>径向渐变 - 颜色结点均匀分布</h5>
<div id="grad"></div>
<p><strong>注意: </strong> Internet Explorer 9 及之前的版本不支持渐变。</p>
</body>
</html>
```

该例题设置了一个径向渐变，如图 2-10 所示。

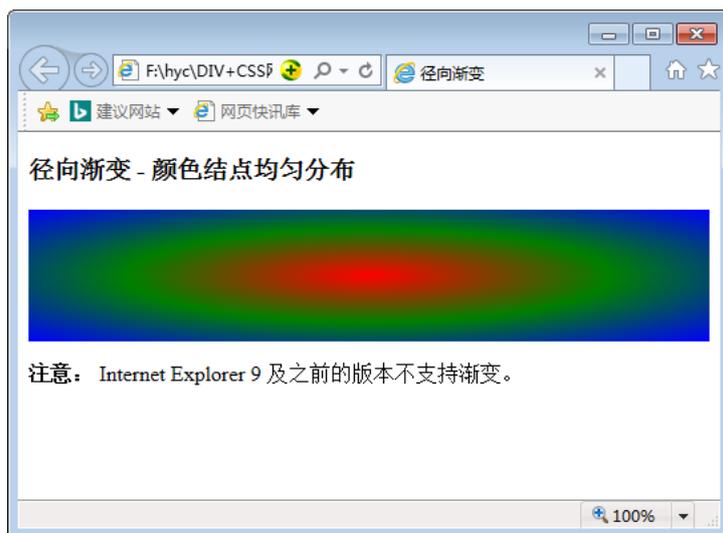


图 2-10 CSS3 径向渐变的效果

2.5.3 CSS3 的其他常用属性

1. CSS3 的文本效果

CSS3 中包含几个新的文本特征。主要文本属性有：

text-shadow: 设置文本阴影。

box-shadow: 设置框阴影。

text-overflow: 设置溢出内容的显示。值为 ellipsis 裁剪加“...”，值为 clip 直接裁剪。

word-wrap: 设置换行方式。

word-break: 设置单词拆分换行。

2. CSS3 的 2D 转换

CSS3 可以设置移动、比例化、翻转、旋转和拉伸元素。2D 变换常用的方法有:

translate(): 根据左(X 轴)和顶部(Y 轴)位置给定的参数, 从当前元素位置移动。

rotate(): 在一个给定度数顺时针旋转的元素。负值是元素逆时针旋转。

scale(): 用于增加或减少元素的大小, 取决于宽度(X 轴)和高度(Y 轴)的参数。

3. CSS3 的过渡

CSS3 中, 可以添加某种效果从一种样式转变到另一个样式, 无需使用 Flash 动画或 JavaScript。过渡效果的常用属性为 transition。例如, 下面的代码设置了鼠标停留在元素上时, 元素宽度在 2 秒内从 100px 变为 300px。

```
div
{
    width:100px;
    height:100px;
    background:red;
    transition:width 2s;
}
div:hover
{
    width:300px;
}
```

4. CSS3 的多列

CSS3 可以将文本内容设计成像报纸一样的多列布局。多列的属性有:

column-count: 设置分割的列数。

column-gap: 设置列与列之间的间隙。

column-rule-style: 设置列与列之间的边框样式。

column-rule-color: 设置两列的边框颜色。

习 题 2

1. 选择题

- (1) 层叠样式表文件的扩展名为()。
- A. htm B. lib C. css D. dwt
- (2) CSS 的全称是()。
- A. Cascading Sheet Style B. Cascading System Sheet
C. Cascading Style System D. Cascading Style Sheet
- (3) 使用 CSS 设置格式时, `h1 b{color:blue}`表示()。
- A. h1 标记内的 b 元素为蓝色 B. h1 标记内的元素为蓝色
C. b 标记内的 h1 元素为蓝色 D. b 标记内的元素为蓝色
- (4) 如果某样式名称前有一个“.”, 则这个“.”表示()。
- A. 此样式是一个类样式
B. 此样式是一个序列样式
C. 在一个 HTML 文件中, 只能被调用一次
D. 在一个 HTML 元素中只能被调用两次
- (5) 用 CSS 样式表设置鼠标悬停在超链接上的状态, 应该定义选择器为()。
- A. a:link B. a:hover C. a:active D. a:visited
- (6) 下列哪一项是 CSS 正确的语法构成? ()
- A. `body:color=black` B. `{body;color:black}`
C. `body {color: black;}` D. `{body:color=black(body)}`
- (7) 在 CSS 中不属于添加在当前页面的形式是()。
- A. 内联式样式表 B. 嵌入式样式表
C. 层叠式样式表 D. 链接式样式表
- (8) 为 div 设置类 a 与 b, 应编写 HTML 代码()。
- A. `div.a,b {}` B. `div.a,div.b {}`
C. `div a,b {}` D. `div:a,div:b {}`

2. 思考与回答

- (1) CSS 的主要特点有哪些?
- (2) CSS 选择器有哪几种? 各有什么特点?
- (3) HTML 文档中引用 CSS 样式的方法有哪些?

上机实验 2

1. 实验目的

熟悉并掌握 CSS 的基本概念；掌握 CSS 选择器的类别和使用；掌握 CSS 在 HTML 文档中的应用方法。其中应重点掌握 CSS 选择器的使用。

2. 实验内容

(1) 在记事本中调试书上的各个实例。

(2) 将 h1 ~ h6 标题文字“CSS 样式的使用方法”设置成蓝色，CSS 选择器分别用标记选择器、id 选择器和类选择器，CSS 引用分别用内部样式表、外部样式表和内联样式表来实现。效果如图 2-11 所示。

(3) 要求编写的代码符合 HTML5 格式要求。

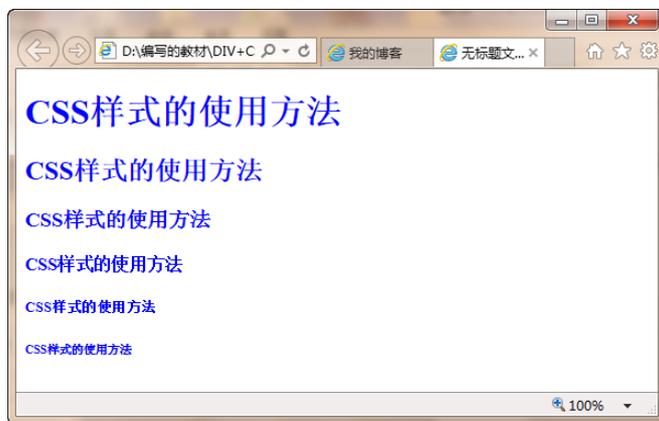


图 2-11 CSS 样式效果