

第六章 样式设计

所样式就是层叠样式表单的简称。其英语翻译为 Cascading Styl Sheets ，所以在网页设计领域一般把层叠样式表单简写为 CSS。

CSS 是一种能够直接可读可写的通俗易懂的语言。它允许设计在 HTML 文档中加入样式(如字体、颜色和空格)。在 CSS 灵活多变的定义方法背后，它所能够指定的样式类型相对于 HTML 来说有过之而无不及。除了通用的颜色、字体、背景等样式，CSS 还可以控制字符间距、填充距、大小写等多达 50 个左右的样式，显示出了强大的定义能力。样式可以定义在 HTML 文档的标记(tag)里，也可以在外部附加文档作为外加文件。一个样式表单可以作用于多个页面——甚至整个站点，具有易用性和扩展性。

6.1 网页构图与样式设计

网页构图的核心就是样式设计，页面的图形布局、文字排版、色彩应用等不能脱离样式而独立存在。页面的可视化元素的要真正为网页的美观实用服务，就必须受后端样式控制和约束。所以说，网页的端构图是由后端样式设计来实现的。

CSS 技术是样式设计中的常用的一种后端语言。我们在进行网页构图时，不能一味地在表面上做文章，要考虑到浏览条件的变化时所带来的可视元素混乱问题。这种情况下就需要用 CSS 定位技术。由平面设计出身的网页设计人员尤其应注意这方面知识的提高。

我们可以通过一个例子来具体说明 CSS 在网页布局中的作用。如：把文字和图片定位在我们想要的位置上，一个页使用了 CSS 的定位技术，而另一个使用表格来控制。在没有改变浏览条件的时候，并看不出有什么差别。但是，当改变浏览器窗口大小，或改变系统的其它设置时，这两种定位手段之间的差别就会明显地显现出来。

如图 6-1 所示，在使用表格时，图片由于窗口面积减小而向左边挤，使得文字由 23 列变为 11 列，从而整个表格的高度增加，页面布局也随着变形；而使用 CSS 定位技术的页面，就不会出现这种情况，无论窗口的形状怎样改变，都不能改变其中内容的位置，这

就是 CSS 技术起到了关键的作用。



图6-1 样式作用的对比

CSS 对于设计者来说是一种非常灵活的工具，不必再把繁杂的样式定义编写在文档结构中，可以将所有有关于文档的样式指定内容全部脱离出来，在行内定义、在标题中定义，可作为外部样式文件供 HTML 调用。这对整个站点或同一栏目的网页风格的统一与变化的实现提供了便利的条件。

多种样式表可以被用于一个网页的文档中，依照重要性不同而分级。如果在同一个 HTML 标记中定义了有冲突的样式，那么最靠近标记的定义将生效。例如的优先级胜过<HEAD>，<HEAD>的优先级胜过外部调用的*.CSS 文件。如果冲突在同一级中发生，较迟执行的将胜出。我们在设计网页的具体过程中应注意多样式冲突问题。

6.2 CSS样式定义的便捷性

定义一个样式表单被多个文档调用，这是一个简便的样式设计手法，是 CSS 的优点所在。这要比在每一个档中去定义它的样式方便的多。我们可以对以下两个程序作一下比较：

```
<HTML>
<HEAD>
<TITLE>没有用 CSS 作一次性定义</TITLE>
</HEAD>
<BODY>
<EM><FONT face="宋体" color="red">
<H1>这是一个红色的宋体 H1 标题
</H1>
</FONT></EM>
<EM><FONT face="宋体" color="red">
<H1>这是第二个红色的宋体 H1 标题
</H1>
</FONT></EM>
</BODY>
</HTML>
```

显然，每一个 H1 标题都要去定义一次的话，效率实在是太低了。CSS 提供了一种叫选择符(Selector)的功能一次定义，就像文件的批处理一样，请分析下面的程序。

```
<html>
<head>
<title>用 CSS 作一次性定义</title>
```

```
<style>
<!--
h1{
font-family: 宋体;
font-style:italic;
color: red;
}
-->
</style>
</head>
<body>
<h1>这是一个红色的宋体 H1 标题</h1>
<h1>这是第 2 个红色的宋体 H1 标题</h1>
</body>
</html>
```

花括号()中的内容定义了选择符的属性和属性值，其中，

font-family：宋体 等效于；

font-style：italic 等同于；

color：red 等同于

在 HTML 文档中定义一个样式，那么每个<H1>标题都显示红色、斜体和宋体字体，如图 6-2 所示。

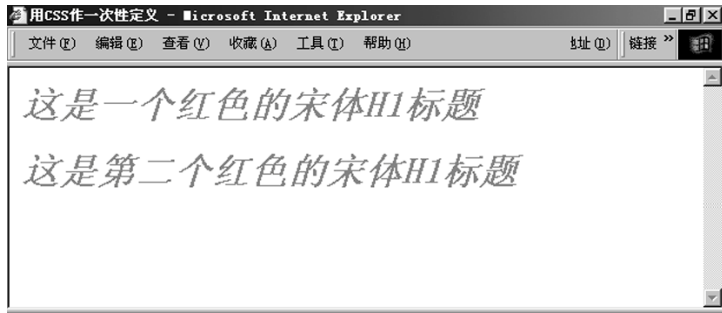


图6-2 使用CSS作一次性定义

6.3 样式与HTML结合方式

共有 4 种方法可将 CSS 定义的样式与 HTML 文档结合。最简单的两种方法，是把样式定义直接写在<HEAD>或<BODY>里。同时也可以独立建立外部样式表文件并附于 HTML 文档。

6.3.1 在文档<HEAD>中定义

最好的定义单一 HTML 文档样式的方法是把样式定义语句放在<HEAD>里，而如果多个文档使用同一样式，那最好使用外部独立 CSS 定义。把样式定义写在注释标记中然后放进<STYLE>标记，写<HEAD>中即可。例如：

```
<HTML>
<TITLE>在 HEAD 中定义样式</TITLE>
<HEAD>
<STYLE type="text/css">
<!--
h1 {font-family: Times; color:blue }
-->
</STYLE>
</HEAD>
```

```

<BODY>

<H1> This is a blue, Times, H1 header.</H1>

<p>

<H1> The second heading</H1>

</BODY>

</HTML>
    
```

将上面的代码输入在 HTML 编辑器中，就生成如图 6-3 所示效果。



图6-3 在HEAD中加入样式定义

6.3.2 在行内定义CSS

除了上面提到的把样式定义放在<HEAD>中，也可以把属性加入 HTML 文档到<BODY> 内来定义。如下面的例子仅使首个<H1>变形，而对第二个<H1>不起作用，如图 6-4 所示。

```

<HTML>

<HEAD>

<TITLE>在 BODY 内定义样式</TITLE>

</HEAD>

<BODY>

<H1 style="font-family: Arial; font-style: italic; color : red ">
    
```

This is a Heading, italic, Arial H1 header.

</H1>

<P>

<H1>This is an H1 header, but it's not red, italic, or Arial.</H1>

</BODY>

</HTML>

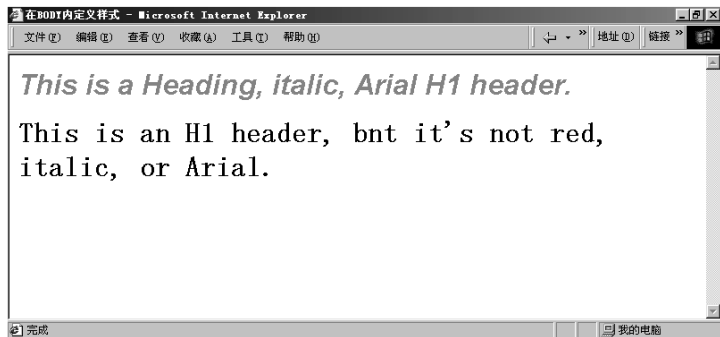


图6-4 在BODY中加入样式定义

6.3.3 CLASS和ID

有时想要定义更为确切的样式,这时用 CLASS 和 ID 就较为方便。一旦定义了 CLASS 或 ID,就可以把它附加于任何 HTML 标记中来定义样式,而不用限制标识只定义一部分风格。

给 CLASS 一个名称(以一个点号“.”开头),再把标准的属性和值定义写在大括号中:

```
.zxq1 {color : #006633}
```

一旦包含有 CLASS 的样式表作用于一个 HTML 文档,就可以把 CLASS 加入文档中的任何一个 HTML 标记:

```
<STRONG class = "zxq">这是一个用 CLASS 作为选择符的样式标题</STRONG>
```

这比把标识作为选择符要简单得多,它能使每一个文本变为深绿色。

ID 的用法和 CLASS 差不多,只不过它以“#”号开头而不是点号:“.”

```
#zxq2{  
font-weight: bolder;  
color: #FF9900  
}
```

把它这样加入 HTML 标记中：

```
<P id = "zxq2"> 北京逆向英语学校 </P>
```

把上面两个例子放在同一个文档中，如图 6-5 所示。

```
<html>  
<head>  
<title>用 ID 、 CLASS 作为选择符定义样式</title>  
<style type="text/css">  
<!--  
.zxq1 {color : #006633;font-size=12pt}  
#zxq2{  
font-weight: bolder;  
font-size:12pt;  
color: #FF9900  
}  
-->  
</style>  
</head>  
<body>  
<strong class="zxq1">  
<P>北京逆向英语教研中心</strong>  
<p id="zxq2">云梦寒多媒体工作室
```

</body>

</html>



图6-5 用ID、CLASS作为选择符定义样式

6.3.4 用CSS文件进行广域定义

前面我们知道了样式定义嵌入 HTML 文档的三种方法，其实最具易用性和可塑性是外挂 CSS 文件调用法。这是控制大量文档的常用方法。

用文档编辑一个样式表单文件，然后使所有的 HTML 页面都调用这个 CSS 文件，从而使它们具有相同的属性，设计出风格统一的网页。例如，把下面的代码用任何文本编辑器编辑后另存为 menghan.CSS 文件：

```
h1, h2, h3, h4, h5 { color: #006633 }  
  
body {  
background-color: #CCCCFF;  
font-family: 宋体, 楷体, 黑体  
}  
  
p { line-height: 200% }  
  
ul li { font-size: 70% }
```

```
.red { color: red }  
.green { color: green. }  
.blue { color: blue }  
#big { font-size: 150% }
```

第 1 行样式定义使<H1>到<H5>标题为暗绿色。第 2 行使页面的背景色为淡蓝色，让 Arial 成为默认字体。以下，使有<P>标签的文字产生双倍行高(200%行高)。第四行让<URL>中的扎<L1>字缩小 70%。然后 3 行建立类： red class 让元素变红，其它也一样。最后两行建立两个 ID：bigID 把文本字体扩展为 150%。

现在把下面程序命名为 menghan1 . HTML，效果如图 6-6 所示。

```
<HTML>  
<HEAD>  
<LINK rel="stylesheet" href="menghan.css">  
<TITLE>外挂 CLASS 样式文件来控制文档</TITLE>  
</HEAD>  
<BODY>  
<H1>你好! </H1>  
<H3>利用外挂 CLASS 样式文件来控制文档中文字颜色:</H3>  
<UL>  
<SPAN class="red"><LI>红色</SPAN>  
<SPAN class="green"><LI>绿色</SPAN>  
<SPAN class="blue"><LI>蓝色</SPAN>  
</UL>
```

用文档编辑一个样式表单文件，

然后使所有的 HTML 页面都调用这个 CSS 文件，

从而使它们具有相同的属性，设计出风格统一的网页。

</BODY>

</HTML>



图6-6 外部调用CSS文件

把 menghan.css 修改如下，并保存重新命名为 menghan2.css。

```
h1, h2, h3, h4, h5 { color: #006633 }
```

```
body {
```

```
background-color: #CCCCCC;
```

```
font-family: Arial, Helvetica, sans-serif
```

```
}
```

```
p { line-height: 200% }
```

```
ul li { font-size: 70% }
```

```
.red { color: red }
```

```
.green { color: green. }
```

```
.blue { color: blue }
```

```
#big { font-size: 150% }
```

```
#upper {text-transform: uppercase}
```

现在用下面代码建立第 2 个 HTML 文档，命名为 menghan2.htm，效果如图 6-7 所示。

```

<HTML>
<HEAD>
<LINK rel="stylesheet" href="menghan2.css">
<TITLE>外挂 CLASS 样式文件来控制文档</TITLE>
</HEAD>
<BODY>
<H2>Theory should be combined with practice.</H2>
<STRONG id="big">The problem has been laid aside for a long time.
</STRONG>
Hard work results in success.
<SPAN id="upper" class="red">This text is uppercase and red !</SPAN>
<P>

```

Much of today's science of physics is based on Newton's discovery of the three laws of motion and his theory of gravity. Newton also developed one of the most powerful tools of mathematics. It is the technique we call calculus. Late in his life, Newton said of his work: "If I saw further than other men, it was because I stood on the shoulders of giants."

```

</P>
</BODY>
</HTML>

```

编辑 STYLE1.CSS 文件就可以转换 HTML 文档形式——不用打开编辑 HTML 文档。比如：可以把 STYLE1.CSS 的<BODY>行转换为：

```

body {
background-color: #000000;
font-family: Courier, monospace;
color: #eeeeee
}

```

把修改后的文本保存，刷新浏览器，将会得到如图 6-8 所示效果。

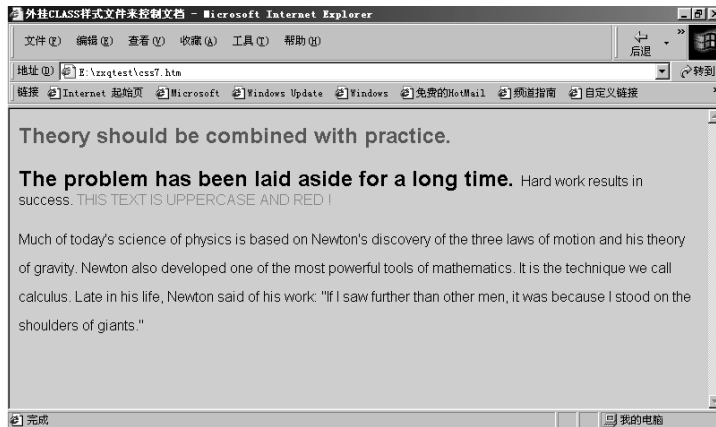


图6-7 menghan2.htm的浏览效果

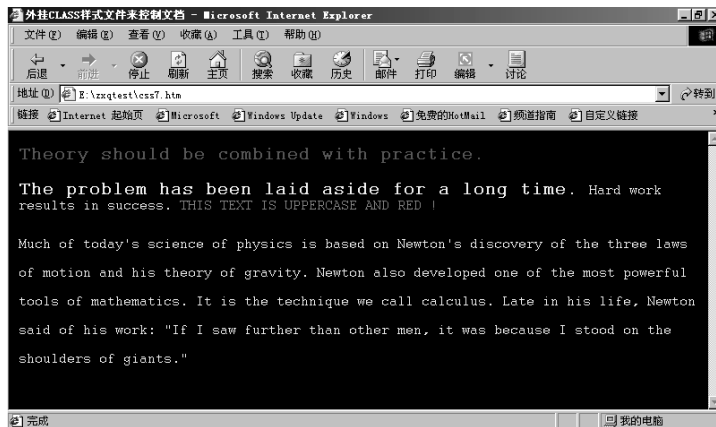


图6-8 编辑CSS后的menghan2.htm浏览效果

6.4 样式的组合与继承

6.4.1 组合

如果一个文档的样式较为复杂，在定义时，需要为各个部分分别定义，这时 CSS 文件需要编写得很大。为了使样式表文件不至于过分冗长，可以把具有相同声明定义的选择符或声明进行组合。

(1)相同声明定义的选择符的组合：用逗号隔开。如：

```
H1, H2, H3 { font-family: helvetica }
```

(2)声明的组合：声明之间用分号隔开。如：

```
H1 {  
font-weight: bold;  
font-size: 12pt;  
line-height: 14pt;  
font-family: helvetica;  
font-variant: normal;  
font-style: normal;  
}
```

可以组合成为：

```
H1 { font: bold 12pt/14pt helvetica }
```

以下两个 HTML 的文档代码是等效的，在浏览器中效果是相同的，如图 6-9 所示。

[代码示例一]

```
<HTML>  
<HEAD>  
<TITLE>样式的组合</TITLE>  
<STYLE>  
<!--  
H1 {  
font-weight: bold;  
font-size: 12pt;  
line-height: 14pt;  
font-family: helvetica;
```

```
font-variant: normal;
font-style: normal;
}
-->
</STYLE>
<HEAD>
<BODY>
<H1>This is a compounding CSS example.</H1>
</BODY>
</HTML>
```

[代码示例二]

```
<HTML>
<HEAD>
<TITLE>样式的组合</TITLE>
<STYLE>
<!--
H1 { font: bold 12pt/14pt helvetica }
-->
</STYLE>
<HEAD>
<BODY>
<H1>This is a compounding CSS example.</H1>
</BODY>
</HTML>
```

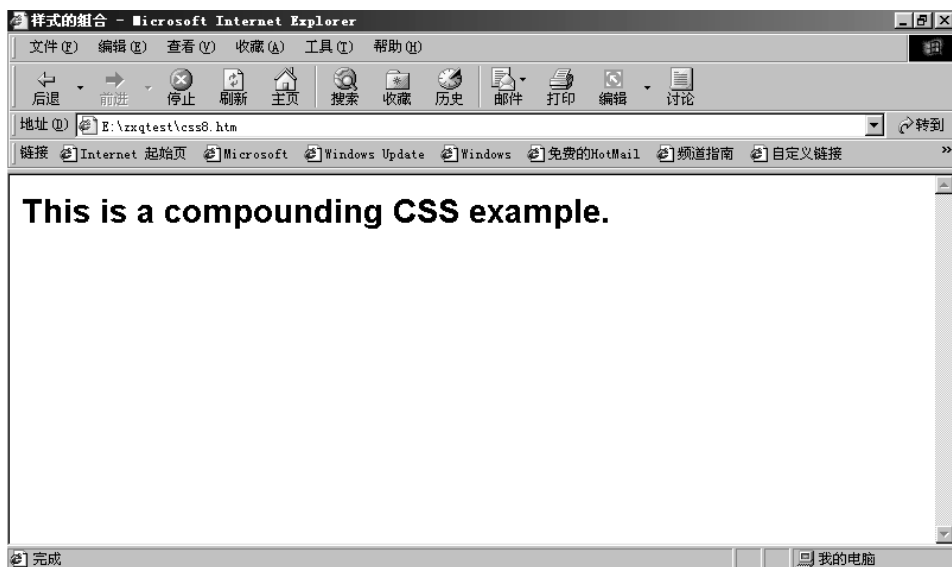


图6-9 CSS的语法组合

6.4.2 继承

我们分析下例：

```
<H1> The headline <EM>is</EM>important! </H1>
```

在这里可以看到，<H1>标记中包含了标记。现在假设把<H1>的颜色设置为红色，而不去设置的颜色，那么，中的“is”将“继承”上级元素(parent element)<H1>的样式设置，仍为红色。

所谓“继承”，就是指在子元素没有被定义样式的情况下，它将继承上级元素的样式定义。“继承”这个概念是非常重要的，在今后的内容里，有很多地方要涉及到“继承”。能够被“继承”的属性有许多，比如“font-family”和“font-size”等。

现在如果要使<BODY>标记中的文本变为黑色，只要写上：

```
BODY {  
  
color: black;  
  
background: url(textpic.gif) white;
```