第5章

div+CSS网页布局

通过CSS能够定制和控制文本、图像等内容在网页中的表现，尤其能够使文本在网页中“表现”为不同的字体、大小和颜色。此外，使用div元素和CSS还能够对网页进行布局——控制元素及其内容在网页中的位置和所占平面空间。

5.1 盒子模型

盒子模型(**Box Model**)在网页布局中至为关键。在盒子模型中，HTML文档主体中的每个元素都被Web浏览器看成一个矩形盒子。盒子模型指定在网页上如何显示元素及其内容。如图5-1所示，元素盒子的最里面是元素中的文字、图像等**内容**(**Content**)。直接包围内容的是**填充**(**Padding**)，在填充中可以呈现元素及其内容的背景。填充以外是**边框**(**Border**)，边框包围着填充。边框以外是最外层的**边界**(**Margin**)，边界默认是透明的，不会遮挡其后的任何内容。

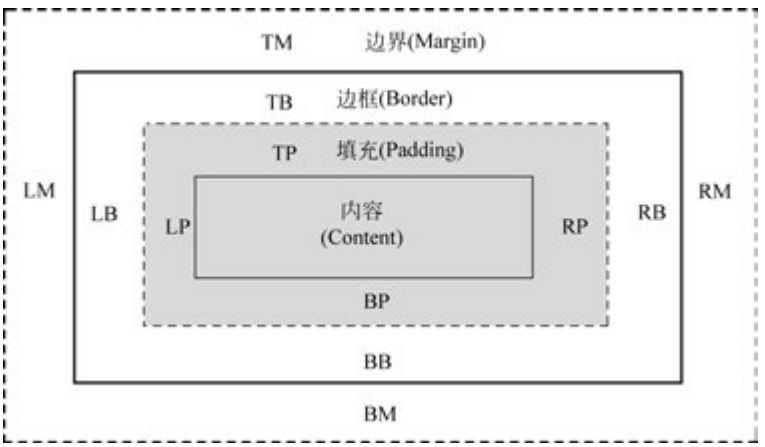


图5-1 盒子模型

注意：

1. 在如图5-1所示的盒子模型中，从内向外的四个矩形框依次表示内容与填充、填充

与边框、边框与边界、盒子与外界的分隔线。

2. 在盒子模型中，填充、边框和边界还可以分解为上、右、下和左四部分。如，填充又可以分解为上填充(padding-top)、右填充(padding-right)、下填充(padding-bottom)和左填充(padding-left)。同理，边框也可以分解为上边框(border-top)、右边框(border-right)、下边框(border-bottom)和左边框(border-left)。



5.2 与盒子模型有关的 CSS 样式特性

与盒子模型有关的 CSS 样式特性主要分为**布局**和**边框**两大类别。

5.2.1 与布局有关的 CSS 特性

1. width(宽度)。

该特性用于设置元素内容所占的宽度。width 特性值通常是以 px(像素)或 em(字体高度)为单位的相对值。

2. height(高度)。

该特性用于设置元素内容所占的高度。height 特性值通常是以 px(像素)或 em(字体高度)为单位的相对值。

3. min-height(最小高度)。

该特性用于设置元素内容所占的最小高度。如果不希望 HTML 元素盒子“太矮”，则可以设置该样式特性。min-height 特性值通常是以 px(像素)或 em(字体高度)为单位的相对值。

4. padding(填充宽度)。

该特性用于设置元素盒子中的填充宽度。padding 特性值通常是以 px(像素)或 em(字体高度)为单位的相对值，也可以使用 padding-top、padding-right、padding-bottom 和 padding-left 特性分别设置上、右、下和左填充的宽度。

5. margin(边界宽度)。

该特性用于设置元素盒子中的边界宽度。与 padding 特性值类似，margin 特性值通常是以 px(像素)或 em(字体高度)为单位的相对值，也可以使用 margin-top、margin-right、margin-bottom 和 margin-left 特性分别设置上、右、下和左边界的宽度。

6. float(浮动方式)。

该特性用于设置元素及其盒子的浮动方式。float 特性值可以是 left、right 或 none，分别表示元素及其盒子向左浮动、向右浮动或不浮动，其中 none 是默认值。

7. display(显示)。

该特性用于设置是否显示以及如何显示元素及其盒子。display 特性值可以是 inline、block 或 none。其中，inline 表示将元素显示为行内元素，行内元素的前后没有换行；block 表示将元素显示为块级元素，块级元素的前后带有换行；none 表示不显示元素。

5.2.2 与边框有关的 CSS 特性

1. border-width(边框宽度)。

该特性用于设置元素盒子中的边框宽度。border-width 特性值通常是以 px(像素)或

em(字体高度)为单位的相对值,也可以使用 border-top-width、border-right-width、border-bottom-width 和 border-left-width 特性分别设置上、右、下和左边框的宽度。

2. border-style(边框样式)。

该特性用于设置元素盒子中的边框样式。border-style 特性值可以是 none(无边框)、dotted(点划线)、dashed(虚线)、solid(实线)等,也可以使用 border-top-style、border-right-style、border-bottom-style 和 border-left-style 特性分别设置上、右、下和左边框的样式。

3. border-color(边框颜色)。

该特性用于设置元素盒子中的边框颜色。与 color 和 background-color 特性值类似,border-color 特性值可以是颜色名称,也可以是 rgb 代码,还可以是一个三位或六位的十六进制值数。另外,也可以使用 border-top-color、border-right-color、border-bottom-color 和 border-left-color 特性分别设置上、右、下和左边框的颜色。

5.2.3 盒子模型演示

以下将通过在 p 元素上应用有关的 CSS 样式特性来演示盒子模型,并说明元素盒子中的内容、填充、边框和边界。

【例 5-1】 演示盒子模型。

具体步骤如下:

1. 将以下 XHTML 和 CSS 代码保存在“5-1.htm”文件中。

```
<!DOCTYPE html PUBLIC " - //W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1 - strict.dtd">
< html xmlns = "http://www.w3.org/1999/xhtml">
  < head >
    < meta http - equiv = "content - type" content = "text/html; charset = gb2312"/>
    < title >盒子模型</title >
    < style type = "text/css">
      body {  font - size:16px;  }
      .box {  color:red;  background - color:yellow;
        padding:40px;  border:40px solid blue;  margin:40px;  }
    </style >
  </head >
  < body >
    < div style = "border:1px solid olive;  width:352px;">
      < p class = "box">元素的文字内容</p >
    </div >
  </body >
</html >
```

在以上 XHTML 文档头部,使用 style 元素定义了内部样式表。在第 1 条 CSS 规则中定义了类型选择器(body)。按照 CSS 的继承性,body 元素的后代元素 div 和 p 都会继承样式特性(font-size:16px;)。在第 2 条 CSS 规则中定义了类选择器(.box),在该类选择器后面进一步定义了颜色(color)、背景颜色(background-color)、填充(padding)、边框(border)和边界(margin)等与盒子模型有关的样式特性。其中,CSS 特性声明(border:40px solid blue;)定义了边框的宽度、样式和颜色,等价于特性声明(border-width:40px;border-style:

solid；　border-color:blue;)。

2. 用 Firefox 浏览器打开 XHTML 文档“5-1.htm”,网页“表现”如图 5-2 所示。

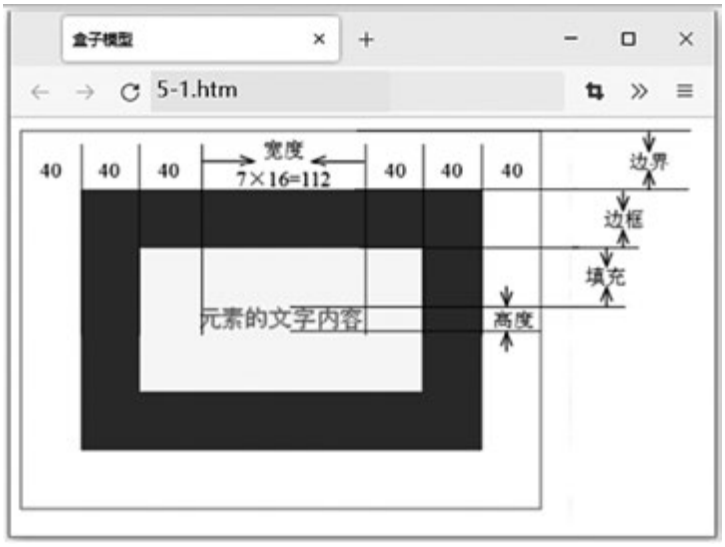


图 5-2　在 Firefox 浏览器中演示盒子模型(p 元素盒子)

借助 Firefox 浏览器自带的“Web 开发者工具”,既能轻松地观察 HTML 元素的盒子模型,又能方便地查看或调整作用于 HTML 元素的 CSS 样式。具体操作如下：使用 Firefox 浏览器打开网页后,按 F12 键(或 Ctrl＋Shift＋I 组合键),可以在浏览器窗口下方显示“Web 开发者工具”窗口。如图 5-3 所示,在“Web 开发者工具”窗口中选择“查看器”选项卡,在左边窗格中选中 p 元素,会在中间窗格中显示内部样式表中的第 2 条 CSS 规则,在右边窗格的“布局”选项卡中显示上、右、下、左填充(padding)、边框(border)和边界(margin)的宽度。

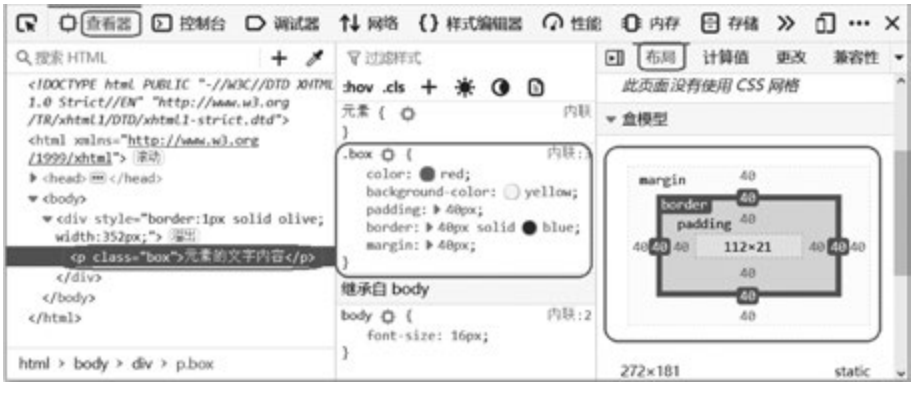


图 5-3　在“Web 开发者工具”中观察盒子模型

注意：

1. 为了说明和演示 p 元素盒子的边界,在 XHTML 文档主体中将 p 元素嵌套在 div 元素内。因此,div 元素和 p 元素是父元素和子元素的关系。同时,在 div 元素的开始标签中

使用 style 属性声明行内特性(border:1px solid olive;)。这样,在 p 元素盒子的边界外可以增加一个橄榄色的实线矩形框。

2. 在类选择器(.box)后面的特性声明中,填充(padding)、边框(border)和边界(margin)的宽度都是 40px。因此,在 p 元素盒子中,左/右 padding、左/右 border 和左/右 margin 共计占用 240px 的宽度。此外,p 元素继承祖先元素 body 的样式特性(font-size:16px;),其中的七个汉字"元素的文字内容"总共需要 16px×7=112px 的宽度。

另外,div 元素和 p 元素是父子关系。父元素 div 的 width(352px)是供子元素 p 盒子的整个宽度使用的。在这 352px 的宽度中,首先减去子元素 p 的左/右 padding、左/右 border 和左/右 margin 共计 240px 的宽度,之后剩下的 112px 才是子元素 p 的 width。因此子元素 p 的 width(112px)恰好满足其中七个汉字"元素的文字内容"的宽度需求,所以在网页中七个汉字"元素的文字内容"在同一行上。

如果父元素 div 的 width 小于 352px,减去子元素 p 的左/右 padding、左/右 border 和左/右 margin 共计 240px 的宽度,剩下的 111px 将无法满足七个汉字"元素的文字内容"的宽度需求,此时在网页中七个汉字"元素的文字内容"就不能在同一行上。

3. 一般情况下,与盒子模型相关的样式特性只应用于 body、div、p、ul 等**块级元素**上,而在 em、strong、a、img 和 span 等**行内元素**上不直接应用这些样式特性。

5.3 元素及其盒子在网页中的排列

根据盒子模型及其原理,HTML 文档主体中的每个元素都被 Web 浏览器看成一个矩形盒子。在网页中,元素及其盒子的排列方式主要有**正常流**(**Normal Flow**)和**浮动**(**Float**)两种。

5.3.1 正常流中的元素盒子

默认情况下,所有元素盒子都是在正常流中排列的。在正常流中,元素盒子在网页中的位置是由元素在 HTML 文档中的位置决定的。具体而言,具有兄弟关系的行内元素及其盒子在同一行中从左到右依次排列;具有兄弟关系的块级元素及其盒子从上到下依次排列,其中每个块级元素及其盒子独占一行;块级父元素盒子包含子元素盒子,块级父元素盒子在外面,子元素盒子在里面。

【例 5-2】 正常流中的元素盒子。

XHTML 和 CSS 代码如下:

```
<!DOCTYPE html PUBLIC " - //W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1 - strict.dtd">
< html xmlns = "http://www.w3.org/1999/xhtml">
  < head >
    < meta http - equiv = "content - type" content = "text/html; charset = gb2312"/>
    < title >正常流中的元素盒子</title >
    < style type = "text/css">
      div {  padding:10px;  border:2px solid red;  margin:10px;  }
      a {  padding:5px;  border:2px dashed green;  margin:5px;  }
      ul {  padding:10px;  border:2px solid red;  margin:10px;  }
```

```
        li {  padding:5px;  border:2px dotted green;  margin:5px;  }
    </style>
  </head>
  <body>
    <div>
      这是外面的块级 div 元素(父元素)盒子
      <a href="#">第 1 个行内 a 元素盒子</a>
      <a href="#">第 2 个行内 a 元素盒子</a>
    </div>
    <ul>
      这是外面的块级 ul 元素(父元素)盒子
      <li>第 1 个 li 子元素盒子</li>
      <li>第 2 个 li 子元素盒子</li>
      <li>第 3 个 li 子元素盒子</li>
    </ul>
  </body>
</html>
```

在 XHTML 文档的 body 元素内,有一个块级 div 子元素和一个块级 ul 子元素,div 元素与 ul 元素具有兄弟关系。在块级 div 元素中包含两个相邻的行内 a 元素。在块级 ul 元素中包含三个 li 子元素。此外,块级 ul 元素和 li 子元素还定义了一个无序列表。

用 Firefox 浏览器打开 XHTML 文档“5-2.htm”,网页的“表现”如图 5-4 所示。



图 5-4 正常流中的元素及其盒子

其中,两个红色的实线框分别为块级 div 元素盒子和块级 ul 元素盒子的边框。因为 div 元素与 ul 元素具有兄弟关系,并且均是块级元素,所以 div 元素盒子和 ul 元素盒子从上到下依次排列。

在第 1 个红色实线框表示的块级 div 元素盒子中,包含两个绿色虚线框表示的行内 a 元素盒子,并且这两个行内 a 元素盒子在同一行中从左到右依次排列。

在第 2 个红色实线框表示的块级 ul 元素盒子中,包含三个绿色点画线框表示的 li 子元素盒子,其中的每个 li 子元素盒子独占一行,并且这三个 li 子元素盒子从上到下依次排列。

注:本书是黑色印刷,具体颜色可参见操作界面。

注意：

1. 在 Strict 类型的 XHTML 文档中，body 元素既不能直接出现文本内容，也不能直接包含 a、img 等行内元素。

2. 如果使用 W3C 标记验证服务验证本例的 XHTML 文档，则会得到如下错误提示：

```
Line 21, Column 7: character data is not allowed here
这是外面的块级 ul 元素(父元素)盒子
```

这是因为在 ul 元素的开始标签与结束标签之间直接出现了文本内容。因为本例只是说明和演示元素盒子在正常流中的排列情况，所以可以忽视这个错误提示。

5.3.2 元素盒子的浮动

在正常流中，具有兄弟关系的行内元素及其盒子在同一行中从左到右依次排列，具有兄弟关系的块级元素及其盒子从上到下依次排列。元素及其盒子的这些排列方式十分有限，大大限制了网页布局的灵活性。为此，CSS 还允许使用浮动方法在网页中排列元素盒子。

默认情况下，每个元素盒子的 float 特性值都是 none，也就是元素盒子在正常流中且不浮动。如果将 float 特性值设置为 left 或 right，则元素盒子就会在其块级父元素盒子内向左或向右浮动。

【例 5-3】 元素盒子的浮动。

XHTML 和 CSS 代码如下：

```
<!DOCTYPE html PUBLIC " - //W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1 - strict.dtd">
< html xmlns = "http://www.w3.org/1999/xhtml">
  < head >
    < meta http - equiv = "content - type" content = "text/html; charset = gb2312"/>
    < title >元素盒子的浮动</title >
    < style type = "text/css">
      ul {   list - style - type:none;   }
      li {   float:left;   width:260px;   border:1px solid;   }
    </style >
  </head >
  < body >
    < ul >
      < li >第 1 个 li 元素盒子</li >
      < li >第 2 个 li 元素盒子</li >
      < li >第 3 个 li 元素盒子</li >
    </ul >
  </body >
</html >
```

在内部样式表的第 1 条 CSS 规则中，使用类型选择器(ul)将 list-style-type 特性值设置为 none，能够消除无序列表中每个列表项前面的小黑点。

在第 2 条 CSS 规则中，使用类型选择器(li)将 float 特性值设置为 left，且该类型选择器与主体中的每个 li 元素相匹配。因此，主体中三个 li 元素的 float 特性值均为 left。

用 Firefox 浏览器打开 XHTML 文档“5-3.htm”，网页的“表现”如图 5-5 所示。



图 5-5 元素盒子的浮动

因为三个 li 元素的 float 特性值均为 left，所以第 1 个 li 元素盒子会在其块级父元素(ul)盒子内向左浮动。因为第 1 个 li 元素盒子的右侧还有足够的空间，所以第 2 个 li 元素盒子也会上浮到第 1 个 li 元素盒子所在行，并紧靠第 1 个 li 元素盒子的右侧。同理，因为第 2 个 li 元素盒子的右侧也还有足够的空间，所以第 3 个 li 元素盒子也会上浮到第 2 个 li 元素盒子所在行，并紧靠第 2 个 li 元素盒子的右侧。这样，三个 li 元素盒子会在同一行中从左到右依次排列。

注意：

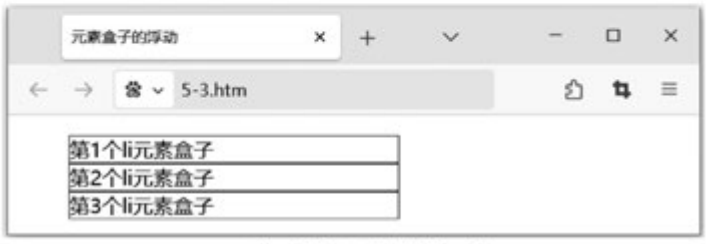
1. 在 XHTML 文档头部的内部样式表中，CSS 特性声明(border:1px solid;)定义元素盒子的上、右、下和左四个边框的宽度均为 1px，且样式都为实线。

2. 如图 5-6(a)所示，当缩窄 Firefox 浏览器窗口时，第 2 个 li 元素盒子的右侧没有足够的空间，所以第 3 个 li 元素盒子无法上浮，只能在下方单独占一行。

如图 5-6(b)所示，当进一步缩窄 Firefox 浏览器窗口时，第 1 个 li 元素盒子的右侧也没有足够的空间，所以第 2 个 li 元素盒子也无法上浮，只能在下方单独占一行。最后，每个 li 元素盒子都单独占一行。



(a) 第3个li元素盒子无法上浮



(b) 三个li元素盒子都单独占一行

图 5-6 元素盒子的浮动(浏览器窗口变窄)

5.3.3 CSS 样式的来源

在设计和制作网页时，经常涉及 CSS 样式的两种来源。

1. 用户代理来源(User-Agent Origin)。每个 Web 浏览器都会针对许多 HTML 元素的某些 CSS 特性预先设置默认样式。表 5-1 列出了常见的 Web 浏览器默认样式。

表 5-1 常见的 Web 浏览器默认样式(以 Firefox 为例)

HTML 元素	Web 浏览器默认样式及特性声明
html	font-size:16px;
body	display:block; color:rgb(0,0,0); font-size:16px;
p	display:block; margin-top:16px; margin-bottom:16px;
ul	display:block; padding-left:40px; margin-top:16px; margin-bottom:16px; list-style-type:disc;
li	display:list-item; float:none;
a	display:inline; text-decoration-line:underline;
em	display:inline; font-style:italic;
strong	display:inline; font-weight:bolder;

2. 作者来源(Author Origin)。网页设计和制作者通过编写 CSS 代码设置样式,包括行内样式以及在内部样式表、外部样式表中设置的样式。

【例 5-4】 浏览器默认样式。

XHTML 和 CSS 代码如下:

```
<!DOCTYPE html PUBLIC " - //W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1 - strict.dtd">
< html xmlns = "http://www.w3.org/1999/xhtml">
< head >
  < meta http - equiv = "content - type" content = "text/html; charset = gb2312"/>
  < title >浏览器默认样式</title >
    < style type = "text/css">
      body {  color:red;  }
    </style ></head >
< body >
  < p >
    p 元素
    < a href = " # "> a 元素</a >
    < em > em 元素</em >
    < strong > strong 元素</strong >
  </p >
  < ul >
    < li >列表项 1 </li >
    < li >列表项 1 </li >
  </ul >
</body >
</html >
```

在本例的内部样式表中,只有 1 条 CSS 规则,而且仅使用类型选择器(body)将 color 特性值设置为 red。

使用 Firefox 浏览器打开网页后,按 F12 键(或 Ctrl+Shift+I 组合键),可以在浏览器窗口下方显示“Web 开发者工具”窗口。如图 5-7 所示,在“Web 开发者工具”窗口中选择

“查看器”选项卡，在左边窗格中选中 body 元素，会在中间窗格中显示内部样式表中的唯一一条 CSS 规则(body{ color:red; })。在中间窗格中取消勾选 color 复选框，可以禁用该条 CSS 规则，网页中文本的颜色会从红色变为黑色。在右边窗格中首先选择“计算值”选项卡，然后勾选“浏览器样式”复选框，再在左侧的文本框中输入 CSS 特性名称“color”，即可在下方查看 Firefox 浏览器针对 body 元素的 color 特性预先设置的默认样式(rgb(0,0,0)，即黑色)。

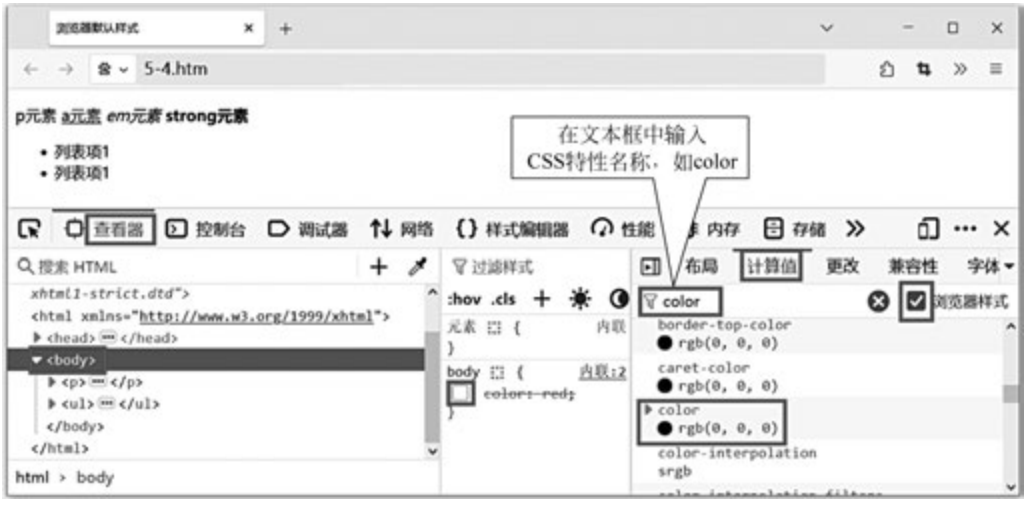


图 5-7 查看 Firefox 浏览器默认样式

在本例中，刚打开网页时，文本的颜色为红色，是因为 CSS 规则(body { color:red; })中的样式取代了浏览器默认样式(黑色)。禁用 CSS 规则(body { color:red; })后，会使用浏览器默认样式(黑色)作为文本颜色。

使用上述类似的操作方法，可以查看表 5-1 中常见的 Web 浏览器默认样式。

注意：在设计和制作网页时，经常需要针对表 5-1 中带下画线的 HTML 元素及 CSS 特性声明，通过编写 CSS 代码取代浏览器默认样式。

5.3.4 应用 CSS 和盒子浮动制作水平导航条

在 Web 标准网页设计和制作中，使用 ul、li 和 a 元素、并应用 CSS 和盒子浮动，可以制作水平导航条。

【例 5-5】 应用 CSS 和盒子浮动制作水平导航条。

XHTML 和 CSS 代码如下：

```
<!DOCTYPE html PUBLIC " - //W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1 - strict.dtd">
< html xmlns = "http://www.w3.org/1999/xhtml">
< head >
  < meta http - equiv = "content - type" content = "text/html; charset = gb2312"/>
  < title >水平导航条</title >
  < style type = "text/css">
    ul {  width:780px;  height:3em;  line - height:3em;  padding:0;  margin:0;
```

```
        list-style-type:none;  background-color:#fa1;  }
      li {  float:left;  width:260px;  text-align:center;  }
      a {  display:block;  text-decoration-line:none;  color:red;  }
      a:hover {  color:purple;  background-color:#dc8;  }
    </style>
  </head>
  <body>
    <ul>
      <li><a href="#">链接 1</a></li>
      <li><a href="#">链接 2</a></li>
      <li><a href="#">链接 3</a></li>
    </ul>
  </body>
  </html>
```

在主体的 li 元素中将 a 元素作为其子元素,则可以在无序列表的列表项中创建文本超链接。

内部样式表的第 1 条 CSS 规则能够取代浏览器对 ul 元素的 padding-left、margin-top、margin-bottom 和 list-style-tyle 特性设置的默认样式。其中,特性声明(list-style-type:none;)能够消除无序列表中每个列表项前面的小黑点。此外,height 和 ine-height 特性值相同,能够使 ul 元素盒子里的 li 子元素盒子及其中的文本超链接垂直居中。

在第 2 条 CSS 规则中,特性声明(float:left;)能够取代浏览器对 li 元素设置的默认样式(float:none;)。这样,主体中的每个 li 元素会在其父元素 ul 盒子内向上再向左浮动。特性声明(text-align:center)能够使 li 元素盒子里的 a 子元素盒子及其中的文本水平居中。

在第 3 条 CSS 规则中,特性声明(display:block;)能够取代浏览器对 a 元素设置的默认样式(display:inline;),这样可以扩大链接对鼠标的响应面积。特性声明(text-decoration-line:none;)能够取代浏览器对 a 元素设置的默认样式(text-decoration-line:underline;),这样文本超链接没有任何修饰线,当然也没有下画线。

在第 4 条 CSS 规则中,使用类型选择器及伪类(a:hover)可以定义文本超链接在 hover 状态下的样式——当光标悬停在文本超链接上时,文本及背景颜色都会发生变化。

用 Firefox 浏览器打开 XHTML 文档“5-5.htm”,网页的“表现”如图 5-8 所示。



图 5-8　使用 ul、li 和 a 元素以及盒子浮动制作水平导航条

在第 1 条 CSS 规则中,特性声明(width:780px;)设定 ul 元素盒子的内部宽度固定为 780px。在第 2 条 CSS 规则中,特性声明(width:260px;)设定 li 元素盒子的内部宽度固定为 260px,padding、border 和 margin 的宽度使用浏览器默认样式中的 0,所以每个 li 子元素盒子的宽度均为 260px。这样,ul 元素盒子恰好可以容纳三个 li 子元素盒子。因此,即使缩

窄 Firefox 浏览器窗口，三个 li 子元素盒子始终在同一行上。

5.3.5 CSS 简写特性

在 CSS 规则的特性声明中使用简写特性(Shorthand Properties)，可以同时设置多个特性值。

1. font 简写特性。

font 简写特性及其声明的语法如下：

```
font:['font-style'||'font-weight']?'font-size'[/'line-height']?'font-family';
```

其中，'font-style'表示 font-style 特性值，例如 italic。'font-weight'表示 font-weight 特性值，例如 bolder。'font-size'表示 font-size 特性值，例如 26px。||表示其前后的特性值属于不同类型，因此可以交换出现的顺序。此外，||前后的特性值可以出现，也可以不出现。如果不出现，则使用浏览器默认样式中的特性值。?表示[]中的特性值可以不出现，也可以出现一次，但不可以出现多次。但是，特性值'font-size'和'font-family'必须依次出现。

例如，“font：italic bolder 26px/100px 楷体；”等价于“font-style：italic； font-weight：bolder； font-size：26px； line-height：100px； font-family：楷体；”。

2. background 简写特性。

background 简写特性及其声明的语法如下：

```
background:'background-color'||'background-image'||'background-repeat'
```

例如，“background：pink url(../ch01/1-2.gif) no-repeat；”等价于“background-color：pink； background-image：url(../ch01/1-2.gif)； background-repeat：no-repeat；”。

又如，“background：pink；”等价于“background-color：pink；”。

再如，“background：url(../ch01/1-2.gif)；”等价于“background-image：url(../ch01/1-2.gif)；”。

3. border 简写特性。

border 简写特性及其声明的语法如下：

```
border:'border-width'||'border-style'||'border-color'
```

例如，“border：1px solid；”等价于“border-width：1px； border-style：solid；”。

又如，“border：solid wheat；”等价于“border-style：solid； border-color：wheat；”。

4. text-decoration 简写特性。

text-decoration 简写特性主要应用于 a 元素。

简写特性及其声明的语法如下：

```
text-decoration:'text-decoration-line'||'text-decoration-style'||'text-decoration-color'
```

例如，“text-decoration：none；”等价于“text-decoration-line：none；”。

通常在如下 CSS 规则中出现和使用 text-decoration 简写特性：

```
a {  text-decoration:none;  }
```

这样文本超链接没有任何修饰线，当然也没有下画线。

5. list-style 简写特性。

list-style 简写特性及其声明的语法如下：

```
list-style:'list-style-position'||'list-style-image'||'list-style-type'
```

例如，“list-style:none;”等价于“list-style-type:none;”。

通常在如下 CSS 规则中出现和使用 list-style 简写特性：

```
ul {  list-style:none;  }
```

这样能够消除无序列表中每个列表项前面的小黑点。

6. padding、margin、border-width、border-style、border-color 简写特性。

例如，“padding:0;”等价于“padding-top:0; padding-right:0; padding-bottom:0; padding-left:0;”。

又如，“margin:8px auto;”等价于“margin-top:8px; margin-right:auto; margin-bottom:8px; margin-left:auto;”。

再如，“border-width:1px 2px 3px;”等价于“border-top-width:1px; border-right-width:2px; border-bottom-width:3px; border-left-width:2px;”。

【例 5-6】 CSS 简写特性。

XHTML 和 CSS 代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type" content="text/html; charset=gb2312"/>
  <title>简写特性</title>
    <style type="text/css">
      #rule1 {  font:italic bolder 26px/100px 楷体;  }
      #rule2 {  height:200px;  background:pink url(../ch01/1-2.gif) no-repeat;  }
      #rule3 {  border:1px solid;  }
      a {  text-decoration:none;  }
      ul {  list-style:none;  }
    </style></head>
<body>
  <p id="rule1">
    font:italic bolder 26px/100px 楷体; =
    font-style:italic; font-weight:bolder; font-size:26px; line-heihgt:100px; font-family:楷体;
  </p>
  <div id="rule2"></div>
  <p id="rule3">border:1px solid; = border-width:1px; border-style:solid;</p>
  <p><a href="#">文本超链接</a></p>
  <ul>
    <li>list item 1</li>
    <li>list item 2</li>
  </ul>
</body>
</html>
```

5.4 应用 div 元素和 CSS 进行网页布局

网页布局是网页设计与制作的一项重要工作。通过合理的网页布局，可以将网页中的文字、图像等内容完美、整洁地展示在 Web 浏览器中，优化网页“表现”效果，甚至提高网页的下载速度。在传统的网页设计与制作中，曾经采用表格布局方法，即使用 table、tr、td 等元素将网页的平面空间划分为不同的矩形区域。

随着 CSS 技术的应用、发展和规范，div＋CSS 网页布局取代了传统的表格布局，成为 Web 网页设计与制作的标准方法。在 div＋CSS 网页布局中，主要应用 div 元素和 CSS 样式表对网页的平面空间进行分区。

在如图 5-9 所示的网页布局示意图中，每个实线矩形框代表一个 div 元素及其盒子的边框——id 属性值为 wrapper、header、nav、main 和 footer 的五个 div 元素分别定义了五个盒子，每个盒子占据网页平面空间的一个矩形区域。其中，wrapper 盒子定义整个网页在 Web 浏览器窗口中所占的平面空间，并且包含 HTML 文档主体中的所有元素及网页的全部内容。在 wrapper 盒子内，从上到下依次排列 header、nav、main 和 footer 四个 div 元素盒子。在 nav 盒子内，又包含使用 ul、li 和 a 元素以及盒子浮动制作的水平导航条。



图 5-9 div＋CSS 网页布局之一

【例 5-7】 根据如图 5-9 所示的网页布局示意图，使用 div＋CSS 进行网页布局。XHTML 和 CSS 代码如下：

```
<!DOCTYPE html PUBLIC " - //W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1 - strict.dtd">
< html xmlns = "http://www.w3.org/1999/xhtml">
< head >
  < meta http - equiv = "content - type" content = "text/html; charset = gb2312"/>
  < title >使用 div + CSS 对网页进行布局</title >
  < style type = "text/css">
    # wrapper {  width:802px;  border:1px solid;  margin:8px auto;  }

      # header {  border:1px solid;  margin:10px;  }

      # nav {  border:1px solid;  margin:10px;  background: # fa1;  }
      # nav > ul {  width:780px;  height:3em;  line - height:3em;  padding:0;  margin:0;
        list - style:none;  }
```

```
        #nav > ul > li {  float:left;  width:260px;  text - align:center;  }
        #nav a {  display:block;  text - decoration:none;  color:red;  }
        #nav a:hover {  color:purple;  background: #dc8;  }

        #main {  height:10em;  border:1px solid;  margin:10px;  }

        #footer {  border:1px solid;  margin:10px;  }
    </style>
</head>
<body>
    <div id = "wrapper"> id 属性值为 wrapper 的 div 元素盒子定义整个网页所占的平面空间
        <div id = "header"> id 属性值为 header 的 div 元素盒子定义网页的页眉</div>

        <!-- id 属性值为 nav 的 div 元素盒子定义网页的水平导航条 -->
        <div id = "nav">
            <ul>
                <li><a href = "#">链接 1 </a></li>
                <li><a href = "#">链接 2 </a></li>
                <li><a href = "#">链接 3 </a></li>
            </ul>
        </div>

        <div id = "main"> id 属性值为 main 的 div 元素盒子定义网页的主体部分</div>

        <div id = "footer"> id 属性值为 footer 的 div 元素盒子定义网页的页脚</div>
    </div>
</body>
</html>
```

注意：在内部样式表的第 1 条 CSS 规则中，ID 选择器 #wrapper 后面的特性声明(margin:8px auto;)代表上、下边界的宽度是 8px，而左、右边界的宽度可以随 Web 浏览器窗口宽度的变化而自动调整。这样，wrapper 盒子及其所包含的内容始终在 Web 浏览器内水平居中。

图 5-10 为一种常见的两栏网页布局。在 wrapper 盒子内，从上到下依次排列 header、nav、main 和 footer 四个 div 元素盒子。在 header 盒子内，从上到下依次排列 logo 和 banner 两个 div 元素盒子。在 main 盒子内，从左到右并行排列 aside 和 article 两个 div 元素盒子，即两栏布局。



图 5-10　div+CSS 网页布局之二

每个盒子在网页布局中有着特定的用途。其中，logo 盒子用于设置网站标志，可以将网站标志图像嵌入 logo 盒子。banner 盒子用于放置广告或宣传图片，一般使用 GIF 格式的图像文件，可以是静态图像，也可以是多帧图像拼接的动画图像。nav 盒子包含使用 ul、li 和 a 元素以及盒子浮动制作的水平导航条。aside 盒子可以包含二级垂直导航条。网站的主要文本内容则组织在 article 盒子内。

【例 5-8】（由学生自己完成）根据如图 5-10 所示的网页布局示意图，使用 div＋CSS 进行网页布局，并写出 XHTML 和 CSS 代码。其他要求如下：

1. 使用内部样式表。

2. 在 Web 浏览器中显示示意图中的文本和边框。

5.5 Web 标准网页设计举例

为了使网页能够在不同 Web 浏览器中正常显示，应该遵循“内容和结构与表现分离”的原则，这也是 Web 标准网页设计的基本原则。

在 Web 标准网页设计与制作中，“内容和结构”保存在 XHTML 文档中，“表现”的定义则保存在 CSS 文档的外部样式表中。在 XHTML 文档头部，使用 link 元素及其 src 属性可以链接 CSS 文档中的外部样式表。在 XHTML 文档主体，如果元素与外部样式表中的选择器及其 CSS 规则匹配，则元素中的内容将按照 CSS 规则中的特性声明“表现”在网页中。

图 5-11 为使用 XHTML＋div＋CSS 技术制作的 Web 标准网页。该网页从上到下依次划分为页眉(header)、水平导航条(nav)、主要内容(main)和页脚(footer)四部分。



图 5-11 使用 XHTML＋div＋CSS 技术制作的 Web 标准网页

网页中的每个部分使用CSS外部样式表的ID选择器和XHTML文档中的div元素来实现。通常情况下,XHTML文档的主体通常具有以下结构:

```
<body>
  <div id="wrapper">
    <div id="header">…</div>
    <div id="nav">…</div>
    <div id="main">…</div>
    <div id="footer">…</div>
  </div>
 </body>
```

其中,页眉(header)、水平导航条(nav)、主要内容(main)和页脚(footer)等对应的div元素包含在id属性值为wrapper的div元素内。在id属性值为nav的div元素内,又可以使用ul、li和a元素以及盒子浮动制作水平导航条。

【例5-9】 根据如图5-11所示的网页示意图,使用XHTML+div+CSS技术设计Web标准网页。要求使用XHTML文档和CSS文档实现"内容和结构与表现的分离"。

1. 在XHTML文档(5-9.htm)中使用如下XHTML代码:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type" content="text/html; charset=gb2312"/>
  <title>使用XHTML+div+CSS设计Web标准网页</title>
  <link rel="stylesheet" type="text/css" href="5-9.css"/>
</head>
<body>
  <div id="wrapper">
    <div id="header"></div>

    <div id="nav">
      <ul>
        <li><a href="#">课程介绍</a></li>
        <li><a href="#">课堂教学</a></li>
        <li><a href="#">学习参考</a></li>
      </ul>
    </div>

    <div id="main">
      <p class="para">本课程以HTML 4.01、XHTML 1.0、HTML5、CSS 2.1、CSS3、ECMAScript和DOM等技术规范为基础,重点讲解基于HTML、div+CSS、Flexible Box和DOM的Web标准网页设计原理与前段开发技术。</p>
      <p class="para">与HTML相比,XHTML的语法更加规范和严谨。CSS为网页的样式设计提供了灵活的手段,能够"通过样式实现网页的表现",并使网页"表现"出丰富的视觉效果。使用XHTML和CSS,能够实现"内容和结构与表现的分离"。在Web标准网页设计中,"内容和结构"保存在XHTML文档中,"表现"的定义则保存在CSS文档中。在XHTML文档的头部,使用link元素可以链接CSS文档中的外部样式表。在XHTML文档的主体中,如果元素与外部样式表中的选择器及其规则匹配,则元素中的内容将按照CSS规则中的特性声明"表现"在网页中。利用div+CSS以及Flexible Box,不仅能够对网页进行灵活的布局,而且能够统一网站内网页的设计风格,同时大大提高CSS代码的可重用性,
```

```
并降低创建、调整和修改网页的成本。</p>
        <p class = "para">本课程能够帮助学习者了解 Web 标准网页设计的基本原理,掌握 XHTML、div + CSS 以及 Flexible Box 等最新的网页制作技术,为今后的动态网页设计与制作以及网站开发与管理打下坚实的基础。</p>
    </div>

    <div id = "footer">
      <address > Cpoyright @ 2010—2025 "网页设计与制作"课程建设小组</address >
    </div>
  </div>
</body>
</html>
```

2. 在 CSS 文档(5-9.css)中创建外部样式表。CSS 代码如下:

```
#wrapper {  width:780px;  margin:8px auto;  }
  #header {  height:100px;  background:url(../ch01/1 - 2.gif);  }

  #nav { background: #fa1; }
  #nav > ul {  height:3em;  line - height:3em;  padding:0;  margin:0;
    list - style:none;  }
  #nav > ul > li {  float:left;  width:260px;  text - align:center;  }
  #nav a {  display:block;  text - decoration:none;  color:red;  }
  #nav a:hover {  color:purple;  background: #dc8;  }

  #main {  min - height:380px;  padding:10px;  background: #cff;  }
    .para {  text - indent:2em;  }

  #footer {  padding:10px;  background: #ff9;  }
    address {  text - align:center;  }
```

在 CSS 文档的样式表中,ID 选择器 #wrapper 对应 XHTML 文档主体中 id 属性值为 wrapper 的 div 元素。

ID 选择器 #header、#nav、#main 和 #footer 分别对应网页中的页眉(header)、水平导航条(nav)、主要内容(main)和页脚(footer)。

类选择器.para 用于定义网页主要内容中文本的段落样式,其后的特性声明 text-indent:2em 将段落的首行缩进设置为 2em。

通过类型选择器 address 定义的样式将作用于网页底部中的文本,并使文本水平居中。

注意:

1. 在 CSS 及其应用中,ID 选择器和类选择器似乎相似,但两者的用法有较大区别。首先,在一个 XHTML 文档中,同一 ID 选择器只能使用一次,而同一类选择器可以使用多次。例如,在 CSS 文档的外部样式表中,既有 ID 选择器 #header 和 #main,又有类选择器.para。而在 XHTML 文档中,只能有一个 div 元素的 id 属性值是 header,也只能有一个 div 元素的 id 属性值是 main,但可以有三个 p 元素的 class 属性值是 para。其次,ID 选择器主要用于对网页进行分区和搭建网页的框架,而类选择器通常用来定义细节性的样式。例如,在 XHTML 文档中,id 属性值为 header 的 div 元素定义了网页的页眉,id 属性值为 main 的 div 元素定义了网页的主要内容区,而类选择器.para 则用于定义段落样式。综上所述,相

对于类选择器,ID选择器的针对性更强。

2. 在XHTML文档主体,通常将块级元素div用作容器——在一个div元素中,既可以包含行内元素,又可以包含其他div元素或块级元素。例如,在XHTML文档主体,元素div#wrapper是一个最外层的容器,在其中又包含id属性值为header、nav、main和footer的四个div元素。

3. 样式特性width在同是块级的父子元素之间具有一定的"传导性"。

(1) 如果一个块级元素的padding、border和margin的左右宽度均是0,则该块级元素的width是其块级父元素width的100%。例如,元素div#header和div#nav的父元素都是元素div#wrapper,两者的padding、border和margin的左右宽度均是0(浏览器默认样式)。因此,元素div#header和div#nav的width就是父元素div#wrapper的width,即780px。

(2) 如果一个块级元素的padding、border和margin的左右宽度不都是0,则该块级元素的width是其块级父元素width减去其padding、border和margin中的非零左右宽度。例如,元素div#main和div#footer的父元素都是元素div#wrapper,元素div#main和div#footer的padding-right和padding-left均是10px。因此,元素div#main和div#footer的width就是760px。

4. 在本例中,元素div#wrapper的width是780px。通过width在同是块级的父子元素之间的"传导性",元素div#nav的以及ul元素的width也都是780px。另外,依据选择器(#nav>ul>li)后面的特性声明,每个li元素的padding、border和margin特性值都是浏览器默认样式设置的0,width都是260px。这样,三个li元素盒子的宽度之和就是780px,恰好占满ul元素的宽度。这样,就实现了div#nav、ul、li等相关元素盒子之间的无缝连接。

5. ID选择器(#main)后面的特性声明(min-height:380px)设置main盒子的最小高度为380px。这样,当main盒子中的(文字)内容很多时,特性声明(min-height:380px)并不会阻止main盒子"长高"。而当main盒子中的(文字)内容非常少时,特性声明(min-height:380px)又能保证main盒子不至于"太矮"。



5.6 习题

1. 在例5-1的XHTML文档中,如果将div元素的开始标签及其代码做如下修改:

```
<div style = "…width:350px">
```

然后用Firefox浏览器打开该XHTML文档,观察并分析p元素盒子及其中内容布局发生的改变。

2. 元素盒子的精确嵌套。XHTML文档的部分代码如下:

```
<!DOCTYPE html PUBLIC " - //W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1 - strict.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
  <head>
    <meta http - equiv = "content - type" content = "text/html; charset = gb2312"/>
    <title>元素盒子的精确嵌套</title>
    <style type = "text/css">
```

```
        #outer { width:300px;  height:200px;  padding:10px;
          border:2px solid red;  margin:0 auto; }
        #inner { width:(    );  height: (    );  padding:10px;
          border:10px dashed green;  margin:30px 10px; }
      </style>
    </head>
    <body>
      <div id = "outer">
        <div id = "inner"></div>
      </div>
    </body>
</html>
```

在上述两个空格中填入相应的CSS特性值，使id属性值为inner的div元素盒子在id属性值为outer的div元素盒子内水平居中和垂直居中，如图5-12所示。

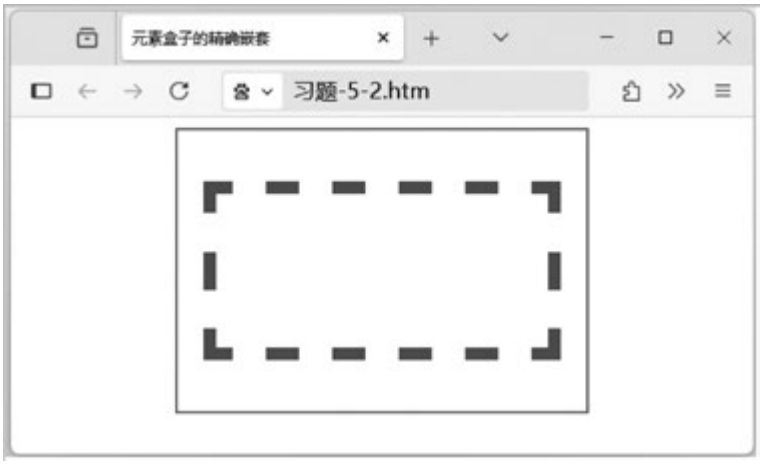


图5-12　元素盒子的精确嵌套

3. 阅读并理解以下CSS和HTML代码。

(1) CSS代码如下：

```
/* 将以下CSS规则保存在“习题-5-3.css”中 */
.c22  {  color:red;  }
#i3  {  color:red;  }
```

(2) HTML代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv = "content-type" content = "text/html; charset = gb2312"/>
  <title>Author 样式的优先级</title>
  <link rel = "stylesheet" type = "text/css" href = "习题-5-3.css"/>
  <style type = "text/css">
    .c21  {  color:green;  }
    .c3  {  color:green;  }
```

```
    </style>
  </head>
  <body>
    <p>Author Origin 的样式又分为行内样式、内部样式和外部样式等三种。</p>
    <p class = "c21 c22" style = "color:blue">
      优先级为:行内样式 > 通过类选择器定义的内部样式 > 通过类选择器定义的外部样式
    </p>
    <p id = "i3" class = "c3" style = "color:blue">
      优先级:行内样式 > 通过 ID 选择器定义的外部样式 > 通过类选择器定义的内部样式
    </p>
  </body>
</html>
```

按照例 5-4 中介绍的操作 Firefox 浏览器的方法,并结合以上代码,回答以下问题。

(1) 当发生冲突时,行内样式是否优先于内部样式或外部样式?

(2) 当通过类选择器定义的内部样式与通过类选择器定义的外部样式发生冲突时,哪种样式的优先级高?

(3) 当通过 ID 选择器定义的外部样式与通过类选择器定义的内部样式发生冲突时,哪种样式的优先级高?

4. 将例 5-9 样式表中的后代结合符改写为子结合符。

5. 从网上复制一篇比较长的文章,然后根据 CSS 样式表的原理,使用 Word 软件对这篇文章进行排版。