第3章 CSS基础知识

CSS是Cascading Style Sheets的缩写，中文译名为层叠样式表。它是用于控制网页样式并允许将样式信息与网页内容分离的一种标记性语言。其中的样式指的就是格式，对网页来说，像文字的大小、颜色以及图片位置等，都是网页显示信息的样式；层叠是指在HTML文件中同时引用多个定义好的样式文件时，若多个样式文件间所定义的样式发生了冲突，则依据优先的层次去处理。

3.1 CSS概述

3.1.1 CSS与HTML的关系

CSS技术诞生于1996年，由W3C负责组织和制定。由于HTML的主要功能是描述网页结构，所以控制网页外观和表现的能力比较差，如无法精确调整文字的大小、行间距等；而且不能对多个网页元素进行统一的样式设置，只能一个元素、一个元素地设置。CSS可以对网页的外观和排版进行更灵活、精确的控制，使网页更美观。简单讲，HTML和CSS的关系就是“内容”和“形式”，即由HTML来组织网页的结构和内容，由CSS来决定页面的表现形式。

3.1.2 CSS的优点

和传统的HTML相比，CSS除了具有强大的控制能力和排版能力之外，最主要的是实现了内容与样式的分离。这种做法带来了许多好处。

(1) 简化了网页的代码，提高了访问速度。外部的CSS文件会被浏览器保存在缓存里，加快下载显示的速度，也减少了需要上传的代码数量。

(2) 可以构建公共样式库，便于重用样式。把一些好的样式写成CSS文件，构建优秀的公共样式库，便于一个网站重复调用或不同的网站共享资源。

(3) 便于修改网站的样式。可以将站点上所有的网页风格都使用一个或几个CSS文件进行控制，只要修改相应的CSS文件，就可改变整个网站的风格特色，避免一个个网页

进行修改，大大减少了重复劳动的工作量。例如：

```
<style type="text/css">
h1 {
    color:red;
    font-size: 3em;
    font-family: Arial;
}
</style>
```

上例中，一条 CSS 指令就可以设置文档中的所有 h1 标签，非常省事。如果已写好一个页面，那么需要把 h1 的颜色全改为黄色时，只需将上述 CSS 代码中 h1 的 color 值改为“color:yellow;”，而不需要逐个去修改 h1 的 color 属性。这样可减少代码的数量，从而加快网页加载的速度。

(4) 方便团队的开发。开发一个网站往往需要美工和程序设计人员的配合。CSS 把内容结构和格式控制相分离，美工做样式，程序员写内容，从而方便美工和程序员分工协作、各司其职，为开发出优秀的网站提供了有力的保障。

3.1.3 一个 CSS 的应用实例

和学习 HTML 一样，在学习 CSS 的过程中只需要使用 Windows 自带的记事本就可以了。当然，如果使用 Dreamweaver 等专业软件为网页添加 CSS 将会更简便。通过例 3-1 可以很容易看出使用 CSS 前后两个网页的区别。

【例 3-1】 运用 CSS 前后的对比实例。

(1) 在记事本中输入下面没有加 CSS 的代码，在浏览器中的效果如图 3-1 所示。

```
<html >
  <head>
    <title>未加 CSS 前的效果!</title>
  </head>
  <body>
    <h1>我喜欢的名句：</h1>
    <h2>走自己的路，让别人去说吧!</h2>
    <h3>痛并快乐着!</h3>
    <h4>黑夜给了我黑色的眼睛，我却用它寻找光明!</h4>
  </body>
</html>
```

(2) 在记事本中输入下面加入 CSS 后的代码，在浏览器中的效果如图 3-2 所示。

```
<html >
  <head>
    <title>加了 CSS 后的效果!</title>
    <style type="text/css">/*设置 CSS*/
```

```
    h1,h2,h4 {
              font-size: 15px; text-align: center;
    }              /*将 h1、h2 和 h3 字体大小都设为 15 像素并居中排列*/
    </style>
  </head>
  <body>
    <h1>我喜欢的名句:</h1>
    <h2>走自己的路,让别人去说吧!</h2>
    <h3 style="display:none">痛并快乐着!</h3><!--将 h3 设为隐藏效果-->
    <h4>黑夜给了我黑色的眼睛,我却用它寻找光明!</h4>
  </body>
</html>
```

(3) 设置前、后的变化。设置 CSS 后,统一了文字大小和排列方式,隐藏了部分文字。

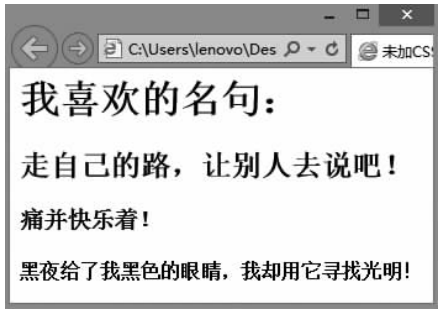

图 3-1　没有加 CSS 的网页效果

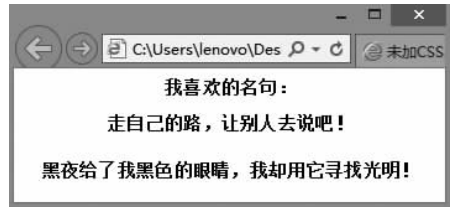

图 3-2　加入 CSS 后的网页效果

3.2　CSS 的基本语法

3.2.1　CSS 的语法

CSS 由一系列样式规则组成,浏览器将这些规则应用到相应的元素上。一条 CSS 规则由两部分构成:选择器(selector)以及一条或多条声明(declaration),多条声明之间用分号分开。选择器其实就是 CSS 样式的名字。常用的选择器有:标记、类、ID、伪类等;声明用于定义元素样式,使用花括号将其包围起来,每条声明由属性(property)和值(value)组成,其中属性是希望设置的样式属性,属性和值之间用冒号分开。CSS 规则的构成如图 3-3 所示。

下面来看一条 CSS 规则。这条规则的作用是将 h1 元素内的文字颜色定义为红色,同时将字体大小设置为 14 像素。

```
h1 { color: red; font-size: 14px; }
```

如图 3-4 所示的示意图展示了这条 CSS 规则的结构。

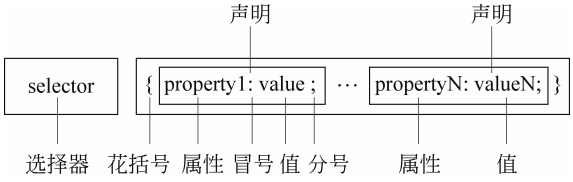

图 3-3　CSS 规则的构成

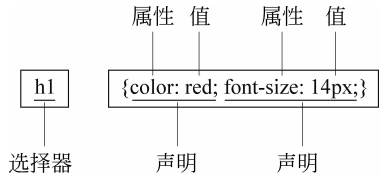

图 3-4　CSS 规则实例

3.2.2　CSS 的语法规则

CSS 的写法和 HTML 有很多不同之处，它有自己的语法要求和技巧，下面列举一些。

(1) 和 HTML 一样，可以在 CSS 中插入注释来说明代码的意思。CSS 注释以/ * 开头，以 * /结尾。例如：

```
/*定义段落样式表*/
P {
    text-align: center;          /*文本居中排列*/
    color: black;                /*文字为黑色*/
    font-family: arial           /*字体为 arial*/
}
```

(2) 属性和值可以设置多个，从而实现对同一标记声明多条样式风格。如果要设置多个属性和值，则每条声明之间要用分号隔开。要养成对最后一条声明也加上分号的习惯，这样在增删声明时不易出错。例如：

```
p { text-align: center; color: red; }
```

(3) 为了方便阅读，可以采用分行的方式书写样式表，即每行只描述一个属性。例如，可以将 p {text-align：center；color：black；font-family：arial；} 写成：

```
P {
    text-align: center;
    color: black;
    font-family: arial;
}
```

(4) 如果属性的值是多个单词组成，必须在值上加引号，比如字体的名称经常是几个单词的组合。例如：

```
P {font-family: "sans serif";}      /*注意代码里面的标点符号都是英文符号*/
```

(5) 如果一个属性有多个值，则每个值之间要用空格隔开。例如：

```
a {padding: 6px 4px 3px}            /*padding 的详解请看第 6 章的 6.1.1*/
```

(6) 如果要为某个属性设置多个候选值,则每个值之间用逗号隔开。例如:

```
P {font-family: "Times New Roman",Times,serif;}
```

(7) 可以把具有相同属性和值的选择器组合起来书写,用逗号将选择器分开,这样可以减少样式的重复定义,这也叫作选择器的集体声明,详见 3.4 节。例如:

```
p, table {font-size: 9pt;}
```

效果完全等效于:

```
p {font-size: 9pt;}
table {font-size: 9pt;}
```

3.3 CSS 的使用方法

HTML 和 CSS 是两种作用不同的语言,它们同时对一个网页产生作用,因此必须通过一些方法将 CSS 与 HTML 挂接在一起才能正常工作。在 HTML 中引入 CSS 的方法有行内式、嵌入式、链接式和导入式 4 种,每种方法都有自己适用的场合以及各自的优缺点。

3.3.1 行内式

所有的 HTML 标记都有一个通用的属性 style,行内式就是在这个 style 属性中为相应的标记添加要应用的样式,即将 CSS 代码直接写在 style 属性中。它在 BODY 中实现,主要在标记中引用,只对所在的标记有效。行内式的格式为:

```
<tag style="property1:value1; property2:value2; …">网页内容</tag>
```

【例 3-2】 行内式样式表的应用。

```
<html>
  <head>
    <title>行内式引入 CSS 的方法示例</title>
  </head>
  <body>
    <p style="font-size:20pt; font-weight: bold; color:red">这个内嵌样式定义段落里面的文字是 20pt 的粗体,字体颜色为红色。</p>
    <p>这段文字没有使用内嵌样式。</p>
  </body>
</html>
```

在浏览器中显示的效果如图 3-5 所示。

行内式是最为简单、直接的 CSS 使用方法,但如果有多个标记都需要设置同一个样

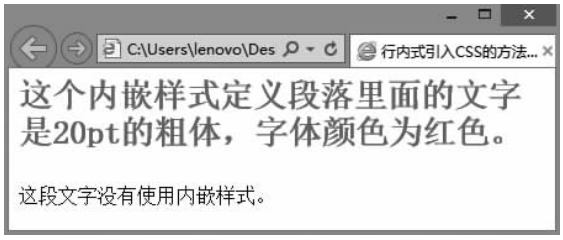

图 3-5　行内式引入 CSS 的效果

式，必须为每一个标记设置同样的 style 属性。由于样式不能共享，会增大代码量，不易维护，也会增大浏览时的流量，影响加载速度，因此不推荐使用，一般应用在某个特定标记需要特殊指定的时候。

3.3.2　嵌入式

嵌入式就是将页面中各种标记的 CSS 样式设置集中写在＜style＞和＜/style＞之间，＜style＞标记是专用于引入嵌入式 CSS 的一个 HTML 标记，它只能放置在 HTML 文档的头部＜head＞和＜/head＞标记之间。

嵌入式的格式为：

```
<style type="text/css">样式表的具体内容</style>
```

说明：type＝"text/css"属性定义了文件的类型为样式表文件。

【例 3-3】　嵌入式样式表的应用。

```
<html>
  <head>
    <title>嵌入式引入 CSS 的方法示例</title>
    <style type="text/css">
      h1.mylayout{
          border-width:1;border:solid;text-align:center;color:red;
      }
      /*将 h1 设置为红色、居中，并具有宽度为 1 像素的实心边框的样式*/
    </style>
  </head>
  <body>
    <h1 class="mylayout">这个标题使用了 style。</h1>
    <h1>这个标题没有使用 style。</h1>
  </body>
</html>
```

在浏览器中显示的效果如图 3-6 所示。

嵌入式对于单一的网页比较方便，它将应用到整个网页中的 CSS 代码统一放置在一起，但是对于一个包含很多页面的网站，如果每个页面都以嵌入式方式设置各自样式，不

图 3-6　嵌入式引入 CSS 的效果

仅麻烦，冗余代码多，维护成本也不低，而且网站每个页面的风格也不好统一，因此嵌入式仅适用于对特殊页面设置单独样式的风格时使用。

3.3.3　链接式

链接式将 CSS 样式代码写在一个以.css 为后缀的 CSS 文件里，然后在每个需要用到这些样式的网页里用＜link＞标记链接到这个样式表文件，这个＜link＞标记必须放到页面的头部＜head＞区域内。链接式的格式为：

```
<link href="外部样式表文件名.css rel="stylesheet" type="text/css">
```

说明：＜link＞标记表示浏览器从“外部样式表文件.css”文件中以文档格式读出定义的样式表；href 属性用于定义.css 文件的 URL；rel＝"stylesheet"属性定义在网页中使用外部的样式表。

【例 3-4】 链接式样式表的应用。

(1) 先用文本编辑器建立一个名为 home.css 的文件，它将 h1 设置了有 1 像素实线边框、内容居中且颜色为红色的样式。home.css 的代码为：

```
h1{border-width:1; border:solid; text-align:center; color:red;}
```

(2) 另建一个 HTML 文件 main.html，在该页面中引入 home.css 文件(假设两个文件放在同一个目录中)，main.html 的代码为：

```
<html>
  <head>
    <title>链接式引入 CSS 的方法示例</title>
    <link href="home.css" rel="stylesheet" type="text/css">
  </head>
  <body>
    <h1>我是使用了 style 的。</h1>
    <h2>我没有使用 style。</h2>
  </body>
</html>
```

注：调试本例时必须将两个文档放在同一个目录下，否则请注意路径。

打开 main.html，在浏览器中显示的效果如图 3-7 所示。

图 3-7　链接式引入 CSS 的效果

目前链接式是使用频率最高、最为实用的方法。它将 HTML 页面本身与 CSS 样式风格分离为两个或者多个文件，实现了页面框架 HTML 代码与美工 CSS 代码的完全分离，使得前期制作和后期维护都十分方便，网站后台的技术人员与美工设计也可以很好地分工合作。因为同一个 CSS 文件可以链接到多个 HTML 文件中，甚至可以链接到整个网站的所有页面中，所以使得网站整体风格统一协调。如果整个网站需要进行样式上的修改，则只需要修改相关的 CSS 文件即可。

3.3.4　导入式

导入式与链接式的功能基本相同，只是在语法上略有区别。链接式使用 HTML 的＜link＞标记引入外部 CSS 文件，而导入式则是用 CSS 的规则引入外部 CSS 文件。

导入式的格式为：

```
<style type="text/css">
    @import url("外部样式表文件名.css");                    /*行末的分号不能省略*/
</style>
```

除了语法不同，链接式和导入式在显示效果方面也有些区别：使用链接式时，会在装载页面主体部分之前装载 CSS 文件，这样显示出来的网页从一开始就是带有样式效果的；而使用导入式时，要在整个页面装载完之后再装载 CSS 文件，如果页面文件比较大，则开始装载时会显示无样式的页面，这样就会给浏览者不好的感觉。这也是现在大部分网站的 CSS 都采用链接式的最主要原因。当然一个网站的页面数达到一定程度时（比如新浪等门户），如果采用链接式就有可能因为多个页面调用同一个 CSS 文件而使速度下降。

3.3.5　引入方式的优先级

如果在各种引入 CSS 的方法中设置的属性不一样，那么在没有冲突时则同时有效。比如嵌入式设置字体为宋体，链接式设置颜色为红色，那么显示结果为宋体红色字。

如果各种引入 CSS 的方法中设置的属性发生了冲突，则 CSS 按引入方法的优先级执行优先级高的方式定义的样式。CSS 中四种引入方式优先级由高到低依次为：行内样式优先级最高；其次是采用＜link＞标记的链接式；再次是位于＜style＞＜/style＞之间的

嵌入式；最后是@import 导入式。

3.4 CSS 选择器

选择器是 CSS 中很重要的概念，所有 HTML 中的标记样式都是通过不同的 CSS 选择器进行控制的。用户只需要通过选择器对不同的 HTML 标记进行选择，并赋予各种样式声明，即可实现各种效果。CSS 常用的选择器包括标记选择器、类选择器、ID 选择器、伪类选择器、后代选择器和通用选择器等。

3.4.1 标记选择器

一个 HTML 页面由许多不同的标记组成，CSS 标记选择器用来声明哪些标记采用哪种 CSS 样式，因此，每一种 HTML 标记的名称都可以作为相应的标记选择器的名称。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。CSS 标记选择器的格式如图 3-8 所示。

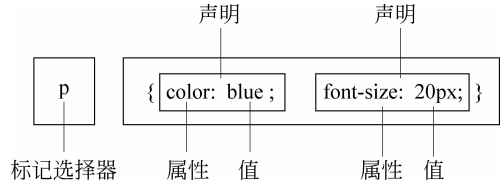

图 3-8 CSS 标记选择器的格式

【例 3-5】 标记选择器的应用实例。

```
<html >
  <head>
    <title>标记选择器的运用</title>
    <style type="text/css">
      P {                                    /* 标记选择器 */
           font-size:18px;                   /* 字体大小为 18 像素 */
           color:green;                      /* 字体颜色为绿色 */
           background:red;                   /* 背景颜色为红色 */
        }
    </style>
  </head>
  <body>
      <p>标记选择器 1</p>
      <p>标记选择器 2</p>
      <h1>h1 则不适用</h1>
      <h2>h2 则不适用</h2>
  </body>
</html >
```

在浏览器中显示的效果如图 3-9 所示。

说明：以上两个 p 元素都会应用 p 标记选择器定义的样式，而 h1 和 h2 元素则不会受到影响。在后期维护中，如果想改变整个网站中 p 标记背景的颜色，只需要修改 background 属性值就可以了。

图 3-9　标记选择器的应用

3.4.2　类选择器

类选择器一般用于以下两种情况。

(1) 通过类选择器把相同的标记分类定义为不同的样式,即实现同一种标记在不同的地方使用不同的样式。例如＜p＞标记的使用,有的段落需要向左对齐,有的段落需要居中对齐,那就可以先定义两个类,在应用时只要在标记中指定它属于哪一个类,就可以使用相应的样式了。这种情况的类选择器格式如图 3-10 所示,“类名称”为定义类的选择器名称,类名称可以是任意英文单词或以英文开头与数字的组合,一般以其功能和效果简要命名,“标记”名称可以用 HTML 的标记。

(2) 通过类选择器可以实现不同标记的元素应用相同的样式。先将这些公共样式定义为同一类,使用时再加上需要调用的标记名即可。例如＜p＞标记、＜h2＞标记和＜h3＞标记都要使用红色、20 像素的样式,那就可以先定义一个红色、20 像素的公共样式,再分别调用就可以了。这种情况的类选择器格式如图 3-11 所示,在“标记.类名称”中省略了“标记”名。

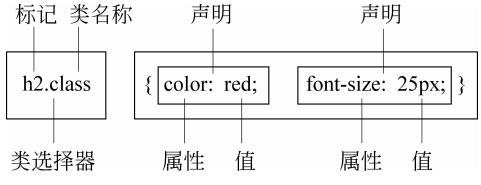

图 3-10　有“标记”名的类选择器的格式

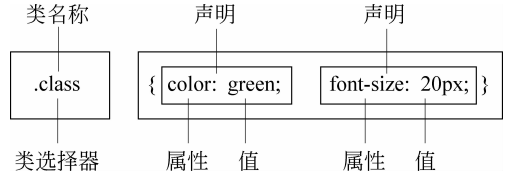

图 3-11　无“标记”名的类选择器的格式

有无“标记”名的类选择器的区别在于:有“标记”名的类选择器其适用范围将只限于该标记所包含的内容;而无“标记”名的类选择器是最常用的定义方法,它可以很方便地在任意标记上套用预先定义好的类样式。

【例 3-6】 类选择器的应用实例。

```
<html >
  <head>
    <title>类选择器的运用</title>
```

```
    <style type="text/css">
      p {                                        /* 标记选择器 */
          color:blue;
          font-size:18px;
      }
      .one {                                     /* 类选择器 1 */
          color: red;
      }
      .two {                                     /* 类选择器 2 */
          font-size:20px;
      }
    </style>
  </head>
  <body>
      <p>应用了标记选择器样式 1</p>
      <p class="one">应用第一种类选择器样式</p>
      <p class="two">应用第二种类选择器样式</p>
      <h2 class="two">h2 同样适用</h2>
      <p class="one two">同时应用两种类选择器样式</p>
  </body>
</html >
```

在浏览器中显示的效果如图 3-12 所示。

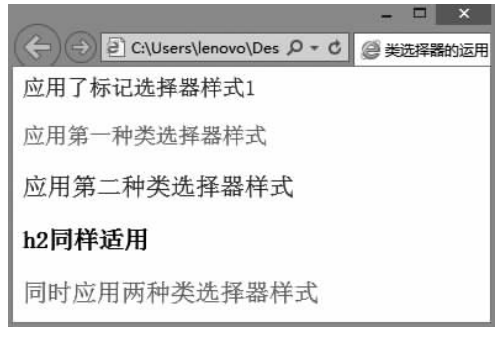

图 3-12　类选择器的应用

说明：首先通过标记选择器定义＜p＞标记的全局显示方案，这样页面中所有＜p＞标记的元素都会产生相应的变化；然后再通过两个类选择器对需要特殊修饰的＜p＞标记进行单独设置。例如希望某一些＜p＞元素的样式不是蓝色，而是红色，则使用.one 这个类选择器；任何一个类选择器都适用于所有 HTML 标记，例如＜h2＞标记也可以使用.two 这个类选择器；有时还可以同时给一个标记运用多个类选择器，从而将两个类别的样式风格同时运用到一个标记中，这在实际制作网站时往往会很有用，可以适当减少代码的长度。例如通过 class＝"one two"将两种样式同时加入，可得到红色 20 像素的效果。

3.4.3　ID 选择器

ID 选择器的使用方法与类选择器基本相同，不同之处在于 ID 选择器只能在 HTML 页面中使用一次，因此其针对性更强，而类选择器可以重复多次应用于多个元素。ID 选择器以半角“＃”开头，且 ID 名称的第一个字母不能为数字，ID 选择器的格式如图 3-13 所示。

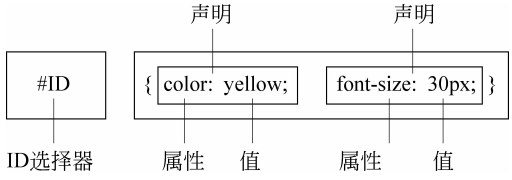

图 3-13　ID 选择器的格式

与类选择器类似，ID 选择器还有一种用法，在“＃ID 名”前加上 HTML“标记”名，这时其适用范围将只限于该标记所包含的内容。ID 选择器局限性很大，只能单独定义某个元素的样式，一般只在特殊情况下使用。

【例 3-7】 ID 选择器的应用实例。

```
<html>
  <head>
    <title>ID 选择器的运用</title>
    <style type="text/css">
      #one {
          font-weight:bold;                    /* 粗体 */
        }
        #two {
          font-size:30px;                      /* 字体大小 30 像素 */
          color:#008000;                       /* 颜色为绿色 */
        }
    </style>
  </head>
  <body>
      <p id="one">ID 选择器 1</p>
      <p id="two">ID 选择器 2</p>
      <p id="two">ID 选择器 3</p>
      <p id="one two">ID 选择器 4</p>
  </body>
</html >
```

在浏览器中显示的效果如图 3-14 所示。

图 3-14　ID 选择器的应用

说明：HTML 文件体的第一行应用了＃one 样式，而第二行与第三行都应用了＃two 样式，显然违反了一个 ID 选择器在一个页面只能使用一次的规定，但浏览器却也能正常显示定义的样式并不报错。虽然如此，但在编写 CSS 代码时，还是应该养成良好的编写习惯，一个 ID 最多只能赋予一个 HTML 元素，因为每个元素定义的 ID 不只是 CSS 可以调用，JavaScript 等

其他脚本语言同样也可以调用。如果一个 HTML 中有两个相同 ID 属性的元素，那么将会导致 JavaScript 在查找 ID 时出错。第四行在浏览器中将没有任何 CSS 样式风格显示，因为 ID 选择器不支持像类选择器那样的多风格同时使用，因为元素和 ID 是一一对应的关系，不能为一个元素指定多个 ID，也不能将多个元素定义为一个 ID。

3.4.4 伪类选择器

伪类是用来表示动态时间、状态改变或者是在文档中以其他方法不能轻易实现的情况，伪类允许设计者自由指定元素在一种状态下的外观。这种状态可以是鼠标停留在某个元素上，或者是访问一个超链接。伪类选择器必须指定标记名，且标记和伪类之间用“:”隔开。伪类选择器的格式如图 3-15 所示。

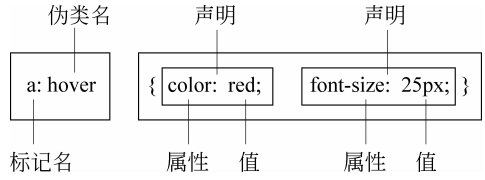

图 3-15 伪类选择器的格式

在 CSS 选择器中，伪类选择器种类非常多，常用的伪类有表示超链接状态的四个伪类选择器：a:link（链接原始存在的状态，但无鼠标动作）、a:visited（被点击或访问过的状态）、a:hover（鼠标悬停于链接上时的状态）、a:active（鼠标点击与释放之间的状态）。在默认的浏览器浏览方式下，超链接为统一的蓝色并且有下划线，被单击过的超链接则为紫色并也有下划线。因为伪类可以描述超链接在不同状态下的样式，所以通过定义 a 标记的各种伪类具有不同的属性风格，就能制作出千变万化的动态超链接。

【例 3-8】 伪类选择器的应用实例。

```
<html>
  <head>
    <title>伪类选择器的运用</title>
    <style type="text/css">
        a:link {color: #000000; font-size:20px;}       /*黑色、20像素*/
        a:visited {color: #0000FF; font-size:25px;}    /*蓝色、25像素*/
        a:hover {color: #FF0000; font-size:30px;}      /*红色、30像素*/
        a:active {color: #FFFF00; font-size:35px;}     /*黄色、35像素*/
    </style>
  </head>
  <body>
    <p><b><a href="http://www.sohu.com" target="_blank">This is a link</a>
    </b></p>
    <p><b>注意:</b>一定要按照 link－visited－hover－active 的先后顺序去定义控制
    超链接的伪类选择器，否则会失效。</p>
  </body>
</html>
```

在浏览器中显示的效果如图 3-16 所示，移动鼠标至超链接时效果如图 3-17 所示。

图 3-16　未单击超链接前的效果

图 3-17　鼠标移到超链接时的效果

说明：在超链接的四个状态会出现颜色和字体大小的变化。

链接伪类选择器的书写应遵循 LVHA 的顺序，即 CSS 代码中四个选择器出现的顺序应为 a:link→ a:visited→ a:hover→ a:active，若违反这种顺序，鼠标停留和激活样式就不起作用了。

伪类选择器可以应用到任意标签，不仅限于＜a＞＜/a＞标签。例如：

```
p: hover {color: red;}
h2: hover {color: red;}
```

3.4.5　后代选择器

后代选择器可将样式应用于包含在其他元素中的元素上。例如：p b {color：red;}这条规则将＜b＞＜/b＞标记中所有的文本都设置为红色，但只有当它们位于＜p＞…＜/p＞标记中才有效(例如，＜p＞＜b＞Hello＜/b＞＜/p＞)。

后代选择器可无限嵌套下去，因此，像 ul li b {color：blue;}这条规则是完全有效的，它表示使一个无序列表的列表元素中的粗体文本以蓝色显示。

3.4.6　通用选择器

通配或者通用选择器可匹配任何元素。例如：＊ {border：1px solid green;}这条规则将应用于整个文档，使其所有元素都有一个绿色边框。因此，虽然不太可能单用＊，但作为复合规则的一部分，它是非常强大的。例如：

```
#boxout ＊p {border: 1px solid green;}
```

这里，＃boxout 后面的第一个选择器是＊符号，表示选择 boxout 对象中的所有元素。后面的 p 选择器缩小了选择范围，变为样式只应用于＃boxout 中的所有 p 元素。

3.4.7　选择器的集体声明

在声明各种 CSS 选择器时，如果某些选择器的风格是完全相同的，或者部分相同，这时便可以利用集体声明的方法，将风格相同的 CSS 选择器同时声明，这样可以减少样式

的重复定义，减少代码长度，这时各个选择器之间使用逗号“,”隔开。

【例 3-9】 集体声明的应用实例。

```
<html>
  <head>
    <title>选择器的集体声明</title>
    <style type="text/css">
      h1,h2,h3,p{                                    /*集体声明*/
          color:purple;                              /*紫色*/
          font-size:15px;
      }
      h2.special,.special,#one{                      /*集体声明*/
          text-decoration:underline;                 /*下划线*/
      }
    </style>
  </head>
  <body>
      <h1>集体声明 h1</h1>
      <h2 class="special">集体声明 h2</h2>
      <h3>集体声明 h3</h3>
      <p>集体声明 p1</p>
      <p class="special">集体声明 p2</p>
      <p id="one">集体声明 p3</p>
  </body>
</html>
```

在浏览器中显示的效果如图 3-18 所示。

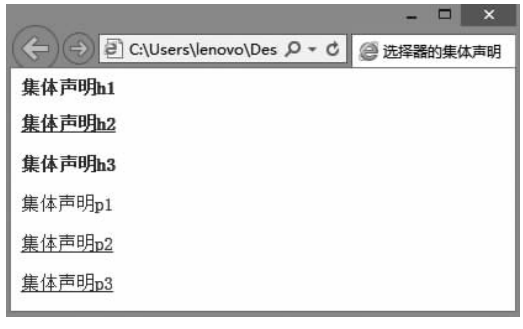

图 3-18 集体声明的应用

另外，对于实际网站中的一些小型页面，例如弹出的小对话框和上传附件的小窗口等，若希望这些页面中所有的标记都使用同一种 CSS 样式，但又不希望逐个加入到集体声明列表中来，这时就可以利用全局声明符号 * 。

【例 3-10】 全局声明的应用实例。

```
<html>
```

```
<head>
  <title>选择器的全局声明</title>
  <style type="text/css">
    * {                                                    /*全局声明*/
        color: purple;
        font-size:15px;
    }
    h2.special,.special,#one{                              /*集体声明*/
        text-decoration:underline;
  </style>
</head>
<body>
    <h1>全局声明 h1</h1>
    <h2 class="special">全局声明 h2</h2>
    <h3>全局声明 h3</h3>
    <p>全局声明 p1</p>
    <p class="special">全局声明 p2</p>
    <p id="one">全局声明 p3</p>
</body>
</html>
```

在浏览器中显示的效果如图 3-19 所示。

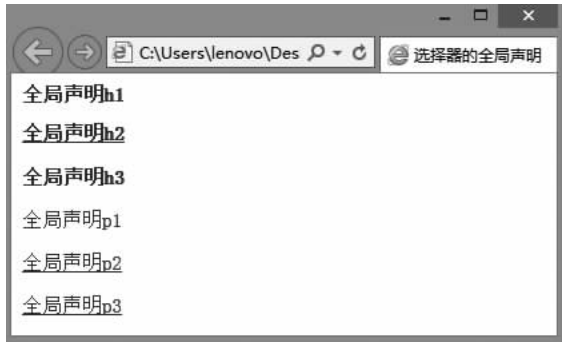

图 3-19　全局声明的应用

3.5　CSS 的层叠性

CSS 层叠性要解决的问题就是当有多个选择器作用于同一元素时，即多个选择器的作用范围发生重叠时，CSS 应该如何处理？它可以简单地理解为“冲突”的解决方案。遇到层叠情况，CSS 的处理原则如下。

（1）如果多个选择器定义的规则不发生冲突，则元素将应用所有选择器定义的样式。

【例 3-11】 CSS 层叠性(无冲突)的应用实例。

```
<html>
  <head>
    <title>CSS 的层叠性 (无冲突)</title>
    <style type="text/css">
      p{                                                    /* 标记选择器 */
          color:blue;
          font-size:18px;
      }
      .special{                                             /* 类别选择器 */
          font-weight: bold;                                /* 粗体 */
      }
      #underline{                                           /* ID 选择器 */
          text-decoration: underline;                       /* 有下划线 */
      }
    </style>
  </head>
  <body>
      <p>标记选择器 1</p>
      <p>标记选择器 2</p>
      <p class="special">受到标记、类两种选择器作用</p>
      <p id="underline" class="special">受到标记、类和 id 三种选择器作用</p>
  </body>
</html>
```

在浏览器中显示的效果如图 3-20 所示。

图 3-20 CSS 无冲突的层叠性的应用

(2) 如果多个选择器定义的规则发生了冲突,则 CSS 按选择器的优先级让元素应用优先级高的选择器定义的样式。CSS 规定的选择器的优先级从高到低的次序为:行内样式>ID 样式>类别样式>标记样式。总的原则是:越特殊的样式,优先级越高。

【例 3-12】 CSS 层叠性(有冲突)的应用实例。

```
<html>
  <head>
    <title>CSS 的层叠性 (有冲突)</title>
```

```
    <style type="text/css">
      p{                                        /*标记选择器*/
          color:blue;
          font-style: italic;                   /*斜体*/
      }
      .green{                                   /*类选择器*/
          color:green;
      }
      .purple{
          color:purple;
      }
      #red{                                     /*ID选择器*/
          color:red;
      }
    </style>
  </head>
  <body>
    <p>这是第1行文本</p>                                          <!--蓝色斜体-->
    <p class="green">这是第2行文本</p>                            <!--绿色斜体-->
    <p class="green" id="red">这是第3行文本</p>                   <!--红色斜体-->
    <p id="red" style="color:orange; ">这是第4行文本</p>          <!--黄色斜体-->
    <p class="purple green">这是第5行文本</p>                     <!--紫色斜体-->
  </body>
</html>
```

在浏览器中显示的效果如图 3-21 所示。

图 3-21　CSS 有冲突的层叠性的应用

说明：由于类选择器的优先级比标记选择器的优先级高，当两者发生冲突时将应用类选择器的样式，因此被两个选择器都选中的第二行 p 元素将应用.green 类选择器定义的颜色样式显示为绿色，但 p 标记选择器定义的其他规则如斜体还是有效的，因此第二行显示效果为绿色斜体的文字；同理，第三行将按优先级高低应用 ID 选择器的样式，显示为红色斜体；第四行将优先应用行内样式，显示为黄色斜体；第五行同时应用了两个类选择器 class＝"purple green"，两个选择器的优先级相同，这时会以前者为准，显示为紫色斜体。

(3) 可以通过!important 关键字来提升某个选择器的重要性。当不同选择器定义的规则发生冲突时,可以通过!important 强制改变选择器的优先级,则优先级次序变为:!important>行内样式>ID 样式>类别样式>标记样式。例如对于上例,如果给.green 选择器中的规则后添加一条!important,代码如下,则第三行和第五行文本将会变为绿色。

```
.green{                          /*类选择器*/
    color:green!important;
                  /*通过!important 提升该选择器中样式的优先级*/
}
```

3.6 CSS 的继承性

除了层叠性,CSS 还具有另外一个特性:继承性。CSS 的继承性是指如果子元素定义的样式没有和父元素定义的样式发生冲突,那么子元素将继承父元素的样式风格,并可以在父元素样式的基础上再加以修改或自己定义新的样式,而子元素的样式风格不会影响父元素。

【例 3-13】 CSS 继承性的应用实例。

```
<html>
  <head>
  <title>CSS 的继承性</title>
    <style type="text/css">
      body{
          text-align:center;
          font-size: 14px;
      }
      p{
          text-decoration:underline;
      }
      em{
          color:red;
      }
      .right{
          text-align:right;
      }
    </style>
  </head>
  <body>
      <h2>电子商务教研室</h2>
      <p><em>电子商务</em>教研室</p>
```

```
    <p class="right"><em>电子商务</em>教研室</p>
  </body>
</html>
```

在浏览器中显示的效果如图 3-22 所示。

图 3-22　CSS 继承性的应用

说明：＜Body＞标记选择器定义的文本居中的属性被所有子元素 h2、p 所继承，因此前两行应用了 body 定义的样式，而且 p 元素还把它继承的样式传递给了子元素 em，但第三行的 p 元素由于通过“.right”类选择器重新定义了右对齐的样式，所以将覆盖父元素 body 的居中对齐，显示为右对齐。另外第一行 h2 元素虽然没为它定义样式，但浏览器对标题元素预订了默认样式，因此它也将覆盖 body 元素定义的 14 像素大小的样式，显示为 h2 的字体大小、粗体。可见，继承来的样式的优先级要比元素具有的默认样式的优先级低。如果要使 h2 元素显示为 14 像素大小，需要对它直接定义字体大小。

CSS 的继承性贯穿整个 CSS 设计的始终，每个标记都遵循着 CSS 继承的概念，可以利用继承关系缩减代码的编写量和提高可读性，尤其在页面内容很多且关系复杂的情况下。例如，如果网页中大部分文字的字体大小都是 12 像素，可以对 body 或 td(若网页用表格布局)标记定义样式为 12 像素。这样由于其他标记都是 body 的子标记，会继承这一样式，就不需要对那么多的子标记去一一定义样式了。有些特殊的地方如果字体大小要求是 14 像素，则可以再利用类选择器或 ID 选择器单独定义。

需要注意的是：不是所有的 CSS 属性都具有继承性，一般是 CSS 的文本属性具有继承性，而其他属性(如背景属性、盒子属性等)不具有继承性。

3.7　CSS 属性的值和单位

样式表是由属性和属性值组成的，有些属性值会用到单位。如果没有单位，浏览器将不知道一个边框是 10 厘米还是 10 像素。在 CSS 中，属性值的单位与在 HTML 中有所不同。HTML 属性的值一般不要写单位，这是因为 HTML 属性的取值可用的单位只有像素或百分比，而 CSS 较复杂，涉及颜色单位和长度单位。