

第3章 CSS 技术

CSS(Cascading Style Sheets,层叠样式表)是对 Web 页面显示效果进行控制的一套标准。CSS 扩充了 HTML 标记的属性设定,使得页面显示效果更加丰富,表现效果更加灵活,更具有动态性。同时使用 CSS 可以将页面样式定义和 HTML 文件分离,使得页面开发及维护工作更易进行。

3.1 预备知识

3.1.1 CSS 基本语法

(1) CSS 定义的基本语法格式如下:

选择符 {规则列表}

其中选择符是指要使用该样式的对象,它可以是一个或多个 HTML 标记、CLASS 选择符或 ID 选择符,如果为多个则使用逗号“,”进行分隔。规则列表是由一个或多个属性定义语句组成的样式规则,各语句间使用分号“;”进行分隔。属性定义语句的语法格式为:

“属性名: 属性值”

(2) 在页面文件中对 CSS 的定义有以下几种方式:

① 直接在页面文件中使用 HTML 标记的 style 属性。其语法格式如下:

```
<标记名 style="样式属性名 1: 属性值 1;  
          样式属性名 2: 属性值 2;  
          ...">
```

② 在页面文件中定义内部样式表。其语法格式如下:

```
<style type="text/css" >  
<!--  
    选择符 1, 选择符 2, ... {样式属性名 1: 属性值 1;  
                           样式属性名 2: 属性值 2;  
                           ...}  
    :  
-->  
</style>
```

③ 在页面文件中嵌入外部样式表。其语法格式如下:

```
<style type="text/css">  
<!--  
    @import url("外部 CSS 样式表文件名");
```

```
-->
</style>
```

④ 链接外部样式表。其语法格式如下：

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

(3) CSS 的注释语句。CSS 的注释语句是位于“/*”和“*/”标记之间的语句内容。

3.1.2 CSS 选择符

CSS 选择符主要有 HTML 标记、CLASS 选择符和 ID 选择符 3 种。它们的定义和使用方法见表 3-1。

表 3-1 CSS 选择符的定义和使用

选 择 符	语 法 格 式	样式使用范围说明
HTML 标记	定义语法：标记{...} 使用语法：<标记>	在 HTML 文件中，所有该标记包含的文本都具有定义的 CSS 样式
CLASS 选择符	定义语法：*.类名{...}或 .类名{...} 使用语法：<标记 class=类名>	在 HTML 文件中的所有使用该类名的标记都具有定义的 CSS 样式
	定义语法：标记.类名{...} 使用语法：<标记 class=类名>	在 HTML 文件中的所有指定该类名的该标记都具有定义的 CSS 样式
ID 选择符	定义语法：# ID 名{...} 使用语法：<标记 id=ID 名>	在 HTML 文件中的所有使用该 ID 名的标记都具有定义的 CSS 样式
	定义语法：标记# ID 名{...} 使用语法：<标记 id=ID 名>	在 HTML 文件中的所有指定该 ID 名的该标记都具有定义的 CSS 样式

注意，这 3 种 CSS 选择符可以混合使用。除了这 3 种基本选择符外，CSS 中还提供了伪类。伪类是一类特殊的选择符，它和类选择符不同，不能由用户自己命名，而是由 CSS 定义，具有特定含义。伪类的基本语法如下：

选择符名：伪类名{属性名：属性值}

如定义 a:visited{color:#3300FF;text-decoration:none;}，则表示在页面中使用超链接标记<a>时，当超链接被访问过后则将使用该样式。

也可以将类选择符和伪类混用，其基本语法如下：

选择符名.类名：伪类名{属性名：属性值}

如：

```
a.red:visited{color:#FF0000;text-decoration:none;}
```

要使用该样式可用下面语句

```
<a class=red href="#">...</a>
```

常用的伪类见表 3-2。

表 3-2 CSS 常见的伪类

伪 类	说 明	例
锚 a 元素的伪类	link 表示动态链接的未访问的链接状态 visited 表示动态链接的已访问链接状态 hover 表示动态链接的鼠标放在链接上的状态 active 表示动态链接的激活链接的状态	a:link{color:green;} a:visited{color:red;} a:hover{color:blue;} a:active{color:black;}
first-letter	CSS 2.0 定义的首字母伪类,定义首字母的样式	p:first-letter{ font-size:30px; }
first-line	CSS 2.0 定义的首行伪类,定义首行的样式	div:first-line{font-size:12px;}

注意：

(1) 和 a 标记相关的伪类分别表示超链接在 4 种不同的状态下的显示效果：link(未访问过的超链接)、visited(已访问过的超链接)、active(单击时的超链接)和 hover(鼠标停留在超链接上)。对于这 4 个伪类在定义时要注意它们的顺序，要按照 a: link, a: visited, a:hover, a:active 的顺序定义。

(2) 和文字段落相关的两个伪类分别为 first-letter 和 first-line。first-letter 定义了第一个字符的状态，而 first-line 定义了段落第一行的状态。

3.1.3 样式表的层叠顺序

可以使用 CSS 语句对 HTML 标记设置不同的显示样式,但是由于 HTML 标记在使用中常常有嵌套情况出现,那么对于控制同一页面内容的嵌套标记,究竟哪一个样式起作用,可以按照以下规则进行判断。

- (1) 直接在页面文件中使用 HTML 标记的 style 属性定义的内联样式优先级最高。
- (2) 其他的样式定义按照在页面文件中出现的顺序,越后出现的优先级越高。
- (3) 由于 id 选择符一般最后定义,所以 id 选择符的优先级高于 class 选择符。
- (4) 没有被定义样式控制的内容将使用浏览器的默认样式。

3.1.4 CSS 基本属性

CSS 的基本属性主要包括背景属性、文本属性、字体属性、边界属性、边框属性、边距属性、列表属性和定位属性等。

(1) CSS 背景属性主要包括 background-color、background-image、background-repeat、background-attachment 和 background-position 等。

(2) CSS 文本属性主要包括 text-indent、text-align、vertical-align、line-height 和 letter-spacing 等。

(3) CSS 字体属性主要包括 font-family、font-style、font-size、font-weight、font-variant、text-decoration 和 text-transform 等。

(4) CSS 边界属性主要使用 margin 属性来控制元素边界与网页其他内容的水平和垂

直间距,除此以外也可以使用 margin-top、margin-right、margin-bottom 和 margin-left 给 4 个边界单独设置具体属性值。

(5) CSS 边框属性主要有 border、border-style 和 border-color。

(6) CSS 边距属性主要使用 padding 来设置元素的内容与元素边框之间的距离,也可以使用 padding-top、padding-right、padding-bottom 和 padding-left 分别设置上、右、下、左 4 个方向的属性值。

(7) CSS 列表属性主要使用 list-style 来设置文字列表属性,控制列表的符号和位置,也可以通过 list-style-type、list-style-position 和 list-style-image 单独进行设定。

(8) CSS 定位属性主要有 top、left、position 和 z-index。

3.2 实验 3.1 CSS 选择符的使用

实验目的:

- (1) 掌握 CSS 的基本语法。
- (2) 掌握 CSS 选择符的使用方法。

实验内容:

编写程序使用 CSS 的 3 种选择符(HTML 标记、CLASS 选择符和 ID 选择符)对 HTML 标记进行样式定义,并观察程序运行结果。

实验步骤:

新建一个 HTML 文件 Css_1. html,在其中输入程序清单 3-1 的代码并保存。为使程序运行结果符合实验任务规定的要求,将程序中的 代码 1 ~ 代码 3 补充完整,并在浏览器中查看程序运行结果是否和图 3-1 一致。

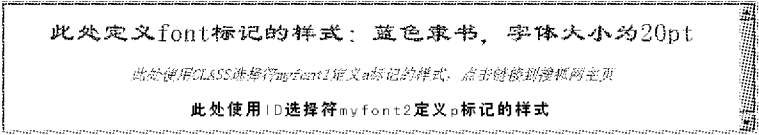


图 3-1 实验 3.1 运行结果

程序清单 3-1:

```
<!--Css_1.html 源代码-->
<html>
<head>
  <title>实验 1 CSS 选字符的使用</title>
</head>

<style type="text/css">
```

```

<!--
    .myfont1{
        font-style:italic;
        color:red;
        text-decoration:none
    }
    #myfont2{
        font-family:黑体;
        font-size:16px;
        letter-spacing:3px
    }

-->
</style>

<body>
<center>
    <font 代码1><!--定义蓝色隶书字体为 20pt-->
        此处定义 font 标记的样式：蓝色隶书，字体大小为 20pt
    </font>
    <br>
<br>
    <a href="www.sohu.com" 代码2><!--定义使用 myfont1-->
        此处使用 CLASS 选择符 myfont1 定义 a 标记的样式，单击链接到搜狐网首页
    </a>
    <br>
    <p 代码3><!--定义使用 myfont2-->
        此处使用 ID 选择符 myfont2 定义 p 标记的样式
    </p>
</center>
</body>
</html>

```

3.3 实验 3.2 制作菜单

实验目的：

- (1) 进一步熟悉 CSS 的基本语法。
- (2) 掌握 CSS 的样式定义方法。

实验内容：

编写 CSS 样式表制作出如图所示的菜单效果。当打开浏览器后可见如图 3-2 所示的菜单样式，当鼠标移至不同菜单项上时，该菜单项的显示样式将发生变化，如当鼠标移至“实验内容”上时，出现如图 3-3 所示效果。

图 3-2 实验 3.2 运行结果(一)



图 3-3 实验 3.2 运行结果(二)

实验步骤：

新建一个 HTML 文件 `Css_2.html`，在其中输入程序清单 3-2 的代码，并将 `代码段 1` 和 `代码段 2` 处样式定义补充完整，使得该 HTML 文件在浏览器中显示如图 3-2 所示效果。

程序清单 3-2：

```

<!--Css_2.html 源代码-->
<html>
<head>
<title>菜单</title>
<style type="text/css">
<!--
* { margin:0;
padding:0;
border:0;
}

body{
font-family: arial, 宋体, serif;
font-size:12px;
}

#menu{
line-height: 26px;
list-style-type: none;
}

#menu a{
display: block;
width: 80px;
text-align: center;
}

#menu a:link{

```

```

        color:#000000;
        text-decoration: none;
    }

#menu a:visited{
    color:#3300FF;
    text-decoration:none;
}

#menu a:hover{
    color: #FFFFFF;
    text-decoration: none;
    font-weight: bold;
}

#menu li{
    代码段 1 <!--设置菜单项-->
}

#menu li a:hover{
    代码段 2 <!--设置鼠标放在链接上的背景-->
}
-->
</style>
</head>

<body>
<ul id="menu">
<li>
    <a href="#">首页</a>
</li>
<li>
    <a href="#">课程简介</a>
</li>
<li>
    <a href="#">实验内容</a>
</li>
<li>
    <a href="#">教学资源</a>
</li>
<li>
    <a href="#">在线交流</a>

```

```
</li>
<li>
    <a href="#">联系我们</a>
</li>
</ul>
</body>
</html>
```

说明：在这个实验中，注意到在定义 CSS 样式时使用到了上下文选择符(contextual selector)，如本程序中的 #menu a{display: block; width: 80px; text-align:center;}。所谓上下文选择符是由两个或更多的选择符组成,这些选择符之间以空格隔开,它的含义是只有当最后一个选择符(如上例中的“a”)是第一个选择符(如上例中的 ID 选择符 menu)的直接后代时该样式才起作用。

3.4 实验 3.3 使用 CSS 样式设置页面布局

实验目的：

- (1) 熟悉运用 CSS 统一站点风格的技巧。
- (2) 了解如何使用 CSS 美化页面布局。

实验内容：

设计 CSS 样式表将已有的 HTML 页面(未使用 CSS 样式的 HTML 文件,运行效果如图 3-4 所示)按照图 3-5 的效果布局并显示。

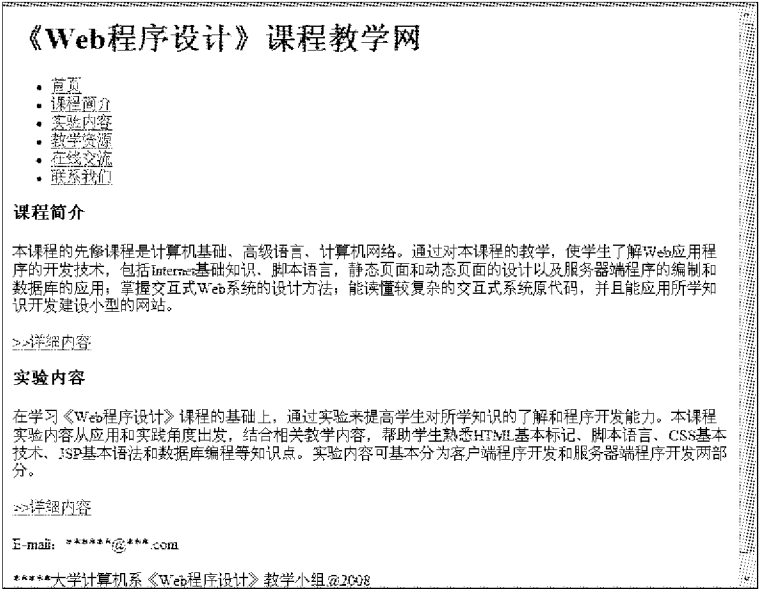


图 3-4 未使用样式表文件的 Css_3. html 运行结果

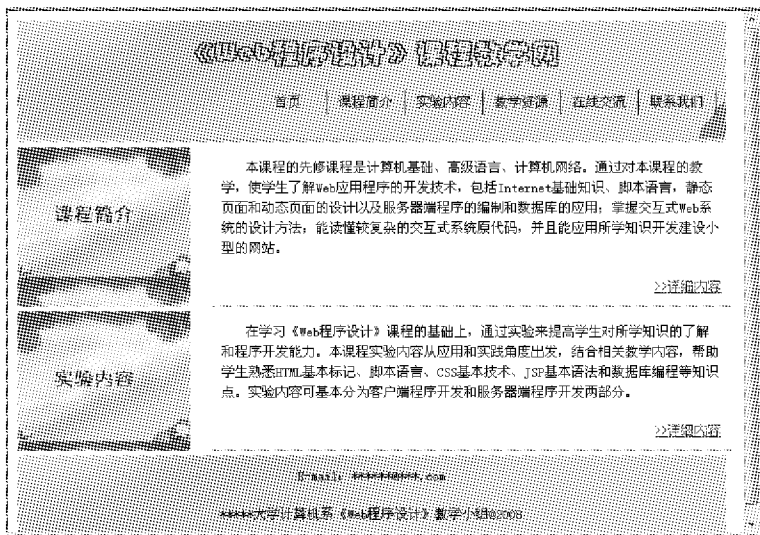


图 3-5 使用样式表文件 Css_3.css 后的 Css_3.html 应达到的效果

实验步骤：

(1) 新建 Css_3.html 文件,在其中输入程序清单 3-3 所示的代码。

程序清单 3-3:

```
<!--Css_3.html 源代码-->
<html>
  <head>
    <title>CSS 实验 3</title>
  </head>

  <body>
    <div>
      <h1>《Web 程序设计》课程教学网</h1>
      <ul>
        <li>
          <a href="#">首页</a>
        </li>
        <li>
          <a href="#">课程简介</a>
        </li>
        <li>
          <a href="#">实验内容</a>
        </li>
        <li>
          <a href="#">教学资源</a>
        </li>
        <li>
```

```
        <a href="#">在线交流</a>
    </li>
    <li>
        <a href="#">联系我们</a>
    </li>
</ul>
</div>
```

```
<div>
<div>
    <h3>课程简介</h3>
</div>
```

```
<div>
    <p>
        本课程的先修课程是计算机基础、高级语言、计算机网络。通过对本课程的教学,使学生了解 Web 应用程序的开发技术,包括 Internet 基础知识、脚本语言,静态页面和动态页面的设计以及服务器端程序的编制和数据库的应用;掌握交互式 Web 系统的设计方法;能读懂较复杂的交互式系统源代码,并且能应用所学知识开发建设小型的网站。
    </p>
    <p><a href="#">>详细内容</a></p>
</div>
</div>
```

```
<div>
<div>
    <h3>实验内容</h3>
</div>
<div>
    <p>
```

在学习《Web 程序设计》课程的基础上,通过实验来提高学生对所学知识的了解和程序开发能力。本课程实验内容从应用和实践角度出发,结合相关教学内容,帮助学生熟悉 HTML 基本标记、脚本语言、CSS 基本技术、JSP 基本语法和数据库编程等知识点。实验内容可基本分为客户端程序开发和服务器端程序开发两部分。

```
</p>
    <p><a href="#">>详细内容</a></p>
</div>
</div>
```

```
<div>
    <p>E-mail: *****@ ***.com</p>
    <p>*****大学计算机系《Web 程序设计》教学小组@ 2008</p>
</div>
</body>
</html>
```