

第 5 章 CSS 基本语法

本章学习目标

- (1) 了解 CSS 的发展现状。
- (2) 掌握 CSS 的语法规则。
- (3) 熟悉不同选择器的区别。
- (4) 掌握 CSS 样式表的使用法。
- (5) 熟悉 CSS 样式优先权。

前面使用 HTML 制作网页时,使用标签的属性对网页元素进行修饰,但是这种方式存在很大的局限性和不足,例如维护困难、不利于代码阅读等。如果希望网页美观大方,并且升级轻松,维护方便,就需要使用 CSS 实现结构和表现的分离。

本章从 CSS 的发展现状出发,介绍 CSS 的基本语法规则、CSS 选择器、CSS 样式表和 CSS 优先级,以及如何在网页中引入 CSS,并对 CSS 进行初步体验。通过本章的学习,读者可以全面了解 CSS 的基本语法,为后面章节的学习打下基础。

5.1 CSS 概述

CSS 是 Web 标准中表现层的技术。但它究竟能做些什么呢?通过本节介绍,我们将深入了解 CSS 的功能、优势等相关问题。

5.1.1 CSS 简介

CSS 是 Cascading Style Sheets(层叠样式表)的简称,可以翻译为“层叠样式表”或“级联样式表”,即样式表。CSS 语言是一种标记语言,它不需要编译,可以直接由浏览器解释执行(属于浏览器解释型语言)。在 Web 标准中,CSS 属于表现层技术,负责网页内容(XHTML)的表现。

CSS 的属性在 HTML 元素中是依次出现的,并不显示在浏览器中。它可以定义在 HTML 文档的标记里,也可以在外部附加文档中作为附加文件。此时,一个样式表可以作用于多个页面,乃至整个站点。

利用 CSS 不仅可以控制一篇文档中的文本格式,而且可以控制多篇文档的文本格式,因此使用 CSS 样式表定义页面文字,将会使工作量大大减小。一些好的 CSS 样式表的建立,可以更进一步地对页面进行美化,对文本格式进行精确定制。

由于允许同时控制多重页面的样式和布局,CSS 可以称得上 Web 设计领域的一个突破。作为网站开发者,可以为每个 HTML 元素定义样式,并将之应用于所希望的任意多的页面中。如需进行全局的更新,只需简单地改变样式,然后网站中的所有元素均会自动地更新。

CSS 样式表的功能一般可以归纳为以下几点。

- (1) 灵活控制页面中文字的字体、颜色、大小、间距、风格及位置。
- (2) 随意设置一个文本块的行高、缩进,并可以为其添加三维效果和边框。
- (3) 便于定位网页中的任何元素,设置不同的背景颜色和背景图片。
- (4) 精确控制网页中各元素的位置。
- (5) 给网页中的元素设置各种过滤器,从而产生诸如阴影、模糊、透明等效果,而这些效果只有在一些图像处理软件中才能实现。
- (6) 可以与脚本语言相结合,使网页中的元素产生各种动态效果。

5.1.2 CSS 的优势

在 CSS 还没有被引入页面设计之前,传统的 HTML 语言要实现页面美工上的设计是很麻烦的。比如要修改某些标签内容的颜色或大小,需要找到所要相关的标签,依次修改相应属性,如果仅修改一个页面内的相应文字,还能容忍,但要修改整个网站的其他页面的相关内容,那工作就相当繁重了。

【例 5.1】 在 HTML 代码中修改标题属性的实现案例。

源代码如下:

```
<body>
<h1 align="center">一级标题文字居中</h1>
<h2 align="left">二级标题文字,左对齐</h2>
<h3 align="right">三级标题文字,右对齐</h3>
<h4 align="justify">四级标题文字,两端对齐</h4>
<h5>五级标题文字,默认对齐属性</h5>
<h6>六级标题文字,默认对齐属性</h6>
</body>
```

如果要修改所有标题的颜色为蓝色,字体为楷体,实现方法如下:

```
<body>
<h1 align="center"><font color="#0066FF" face="楷体">一级标题文字居中</font>
</h1>
<h2 align="left"><font color="#0066FF" face="楷体">二级标题文字,左对齐</font>
</h2>
<h3 align="right"><font color="#0066FF" face="楷体">三级标题文字,右对齐</font>
</h3>
<h4 align="justify"><font color="#0066FF" face="楷体">四级标题文字,两端对齐
</font></h4>
<h5><font color="#0066FF" face="楷体">五级标题文字,默认对齐属性</font></h5>
<h6><font color="#0066FF" face="楷体">六级标题文字,默认对齐属性</font></h6>
</body>
```

其实传统的 HTML 的缺陷远不止上面所说的那些,相比运用 CSS 进行页面设计,其劣势总结如下。

- (1) 维护困难。为了修改某个特殊标记的格式,需要花费很多时间,尤其对于整个网站而言,后期修改和维护的成本很高。

(2) 标记不足。HTML 本身的标记十分少,很多标记都是为网页内容服务的,而关于美工样式的标记,如文字间距、段落缩进等标记在 HTML 中很难找到。

(3) 网页过“胖”。由于没有统一对各种风格样式进行控制,HTML 的页面往往体积过大,占用了很多宝贵的带宽。

(4) 定位困难。在整体布局页面时,HTML 对于各个模块的位置调整显得捉襟见肘,过多的<table>标记同样也导致页面的复杂和后期维护的困难。

当引入 CSS 以后,CSS 提供各种丰富的格式控制方法,使得网页设计者可以轻松地应对各种页面效果,并且 CSS 对网页整体控制较单纯的 HTML 语言有了突破性进展。

【例 5.2】 如例 5.1,通过 CSS 样式实现对所有标题内容颜色和字体的修改。

修改代码如下:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>运用 CSS 修改标题颜色和字体</title>
<style type="text/css">
h1,h2,h3,h4,h5,h6{
    color: "#0066FF";
    font-family: "楷体";
}
</style>
</head>
<body>
<h1 align="center">一级标题文字居中</h1>
<h2 align="left">二级标题文字,左对齐</h2>
<h3 align="right">三级标题文字,右对齐</h3>
<h4 align="justify">四级标题文字,两端对齐</h4>
<h5>五级标题文字,默认对齐属性</h5>
<h6>六级标题文字,默认对齐属性</h6>
</body>
</body>
</html>
```

不需要增加标签,只需在<head>区域增加控制样式的代码即可,如果需要再修改颜色或字体,只要简单地修改一个值就可以了。

CSS 在布局方面也有很大的优势。采用 CSS+DIV 进行网页重构相对与传统的 TABLE 网页布局而言具有以下 3 个显著优势。

(1) 表现和内容相分离。将设计部分剥离出来放在一个独立样式文件中,HTML 文件中只存放文本信息,这样的页面对搜索引擎更加友好。

(2) 提高页面浏览速度。对于同一个页面视觉效果,采用 CSS+DIV 重构的页面容量要比 TABLE 编码的页面文件容量小得多,前者一般只有后者的 1/2 大小,浏览器就不用去

编译大量冗长的标签。

(3) 易于维护和改版。只要简单修改几个 CSS 文件就可以重新设计整个网站的页面。在网页设计中有关布局的优势,将在后面的章节去体会。

5.1.3 CSS 与浏览器的兼容性

网上的浏览器各式各样,绝大多数的浏览器对 CSS 都有很好的支持,因此设计者往往不用担心其设计的 CSS 文件不被用户所支持。但目前主要的问题在于,各个浏览器之间对 CSS 很多细节的处理上存在差异,设计者在某一种浏览器上设计的 CSS 效果,和在其他浏览器上显示的效果可能不一样。在后面章节,会讨论如何解决浏览器兼容性问题。后面写到的 CSS 样式代码,在不特别说明的情况下,各个浏览器的解析是一样的。

5.2 CSS 语法

使用 HTML 时,需要遵从一定的规范。CSS 也是如此,想要熟练使用 CSS 对网页元素进行修饰,首先需要了解 CSS 语法。

5.2.1 CSS 语法规则

CSS 有自己独特的书写方法,如果掌握了它的语法特征,再去了解它的各种属性,那么学习 CSS 将会变得非常容易。下面介绍 CSS 的语法规则。

CSS 的语法规则由两部分构成:选择器(selector)和声明(declaration)。其中,选择器用于决定哪些因素要受到影响;声明由属性(property)和值(value)组成,每条声明由一个或多个属性值对组成。

基本语法如下:

```
Selector{property: value; property: value;...}
```

语法说明如下:

(1) Selector——选择器,又称为选择符,是指这组样式编码所要针对的对象,可以是一个 XHTML 标签(如<p>、<h1>等),也可以是定义了特定 id 或 class 的标签(后面会详细说明)。

(2) property——属性,它是 CSS 样式控制的核心,对于每个 XHTML 标签,CSS 都提供了丰富的样式属性,如颜色、大小、定位、浮动方式等。

(3) value——值,指属性的值,形式有两种,一种是指定范围的值(如 float 属性,只可能有 left、right、none 3 个值);另一种为数值(如 width 能够使用 0~9999 px 范围的值)。

注意:声明中的多个属性值对之间必须用分号隔开。

【例 5.3】 设置一个段落的格式,要求首行缩进 2 个汉字,字体大小为 14px,字体为楷体,文字颜色为蓝色。

分析:要设置的对象是段落,那么格式化的元素就是<p>,即选择器部分就是 p。设定段落的格式为:首行缩进 2 个汉字,字体大小为 14 像素,颜色为蓝色。声明部分包括 3 种属性,分别是 text-indent、font-size 和 color,属性值分别为 28px、14px 和 blue。实现代码

如下：

```
<style type="text/css">
p{
    text-indent: 28px;
    font-size: 14px;
    color: blue;
}
</style>
```

注意：若要定义多个声明，那么必须使用分号将每个声明分开。

5.2.2 选择器的类型

上面介绍的语法规则中提到，Selector 选择器可以是 XHTML 中的标签，也可以是特定的 id 或 class，下面来详细介绍这 3 种基本选择器的用法。

1. 标签选择器

一个完整的 HTML 页面由很多不同的标签组成，而标签选择器，则是决定哪些标签采用相应的 CSS 样式。如例 5.3，就是运用了标签选择器来定义段落的样式。如果多个标签具有相同的样式属性，就可以同时给多个标签进行样式的定义。

【例 5.4】 对 7 个标签同时进行相同样式的定义。

```
<style type="text/css">
p,h1,h2,h3,h4,h5,h6{
    color: red;
    font-family: "楷体";
}
</style>
```

注意：标签之间使用逗号进行分隔。可以使页面中所有的 p、h1~h6 标签内都使用相同的字体楷体，文字颜色为红色。这样做的好处是，对页面中需要使用相同样式的地方，只需要书写一次样式表即可，从而减少代码量，改善 CSS 代码结构。

2. 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。定义 id 选择器时，需要在名称前加 #，在运用 id 选择器时，通过 id="名称"来调用。

【例 5.5】 运用 id 选择器为网页元素进行样式定义。

```
<style type="text/css">
#title{
    text-align: center;
    font-size: 16px;
    color: blue;
    font-weight: bold;
    font-family: "楷体";
}
</style>
```

```
</head>
<body>
<p id="title">
将段落的样式设为标题,实现文字居中,字体大小为 16px,加粗,文字颜色为蓝色。
</p>
</body>
</html>
```

注意：上述案例定义 id 类型的选择器,在定义名称 title 前加 #,在标签 <p> 中,通过 id="title" 来运用 title 中所定义的样式。

根据元素 id 来选择元素,具有唯一性,这意味着同一 id 在同一文档页面中只能出现一次。例如,将一个元素的 id 取值为 nav,那么在同一页面就不能再将其他元素 id 取名为 nav 了。尽管把几个元素都命名成相同的 id 名字,css 选择器还是会把这些元素都选中应用样式(如 class 选择器那样),对于 css 选择器,id 属性的唯一性似乎不存在。然而,对于 JavaScript 而言,它只会选择具有相同 id 名字元素中的第一个。出于一个好的编程习惯,同一个 id 不要在页面中出现第二次。

3. class 选择器

class 直译为类或类别。类选择器能够把相同的元素分类定义成不同的样式。定义类选择器时,在自定义类的前面需要加一个“.”号,调用时,通过 class="名称"调用。如例 5.5,可以运用 class 选择器实现样式的定义。

【例 5.6】 运用 class 选择器为网页元素进行样式定义。

```
<style type="text/css">
.title{
    text-align: center;
    font-size: 16px;
    color: blue;
    font-weight: bold;
    font-family: "楷体";
}
</style>
</head>
<body>
<p class="title">
将段落的样式设为标题,实现文字居中,字体大小为 16px,加粗,文字颜色为蓝色。
</p>
</body>
</html>
```

使用 class 的好处是,对于不同的 XHTML 标签,CSS 可以直接根据 class 名称来进行样式指派。而且 class 选择器是对 CSS 代码重用性的良好体现,多个标签可以使用同一个 class 来进行样式指派,而不需要对每个标签编写样式代码。

5.2.3 选择器的综合运用

对于 CSS 选择器而言,无论什么样的选择器类型,都可以组合使用。

1. 包含选择器

当用户只打算对某个对象的子对象进行样式指定时,包含选择器就派上了用场。包含选择器是指选择器组合中前一个对象包含了后一个对象,对象之间使用空格作为分隔符。

【例 5.7】 定义 h1 标签中的 span 标签的样式。

```
<html>
  <head>
    <style>
      <!--h1 span{
        font-style: italic;
      }
    --></style>
  </head>
  <body>
    <h1>这是包含有 span 标签的一段文本<span>span 标签内的文本,将会斜体显示</span>
  </h1>
    <h1>这是没有包含 span 的一段文本</h1>
    <span>这是单独的一个 span 标签内的文本</span>
    <h2>这是 h2 标签里的文本,包含 span 标签<span>,这是 span 标签内的文本,看是否会被斜体显示</span></h2>
  </body>
</html>
```

程序运行结果如图 5.1 所示。

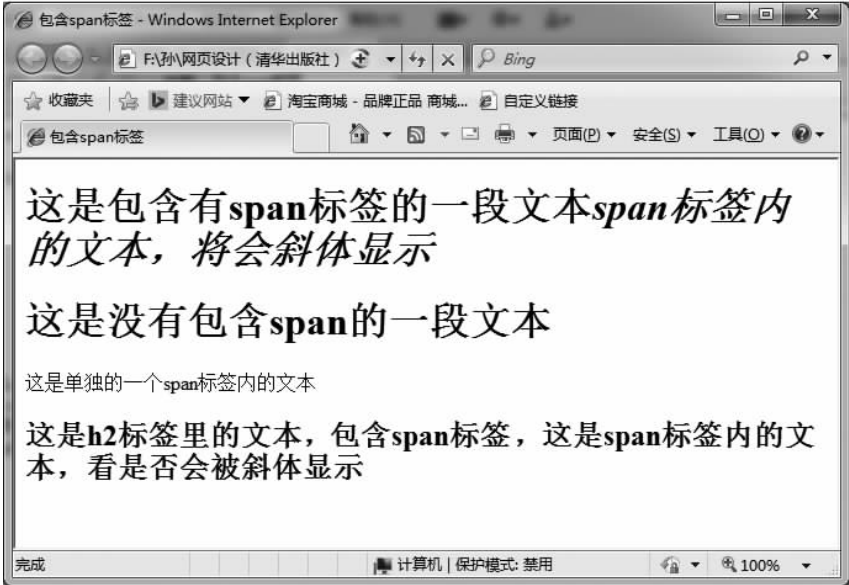


图 5.1 包含选择器的运用

注意: h1 和 span 之间是空格,表示只有在 h1 标签里面的 span 标签才会使用的样式。而“h1,span{…}”表示同时定义两个标签的样式。

2. 组合选择器

对于标签选择器、id 选择器和 class 选择器,它们之间均可以结合使用。

(1) 标签选择器和 class 选择器的结合。例如:

```
table .aa{定义样式}
```

表示 table 标签内的所有 class="aa" 的标签将会应用的样式。

(2) id 选择器和标签选择器的结合。例如:

```
#content p{定义样式}
```

表示 id="content" 的标签内的所有 p 标签将会应用的样式。

(3) id 选择器和 class 选择器的结合。例如:

```
#content .aa{定义样式}
```

表示 id="content" 且 class="aa" 的标签内将会应用的样式。

(4) class 选择器和标签选择器的结合。例如:

```
.con td{定义样式}
```

表示 class="con" 且 XHTML 标签为 td 的标签下将会应用的样式。

总体来说,CSS 选择器的使用是自由的,用户可以根据页面的需求,灵活地使用各种选择器来选择和指派。例 5.8 进行了 3 种选择器的组合运用。

【例 5.8】 选择器的综合运用。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>选择器的综合运用</title>
<style type="text/css">
#a{
    border: 1px #993300 solid;
    width: 400px;
    height: 200px;
}
#a .m{
    font-size: 14px;
    color: #930;
}
.b{
    border: 2px #0033FF dashed;
    width: 400px;
    height: 200px;
}
```



```

.b p{
    font-size: 18px;
    font-weight: bold;
    color: RGB(0,0,255);
}
.c{
    font-size: 24px;
    color: #F00;
    font-family: "楷体", sans-serif;
}
</style>
</head>
<body>
<div id="a">
<p>标识为 a 的区域中, 包含段落标签 p, 该段落没有定义其他样式, 内容将按照浏览器默认样式
显示。该 div 区域定义边框为暗红色, 宽度为 400 像素, 高度为 200 像素。</p>
<p class="m">标识为 a 的区域中, 第二个段落标签中, 使用样式为 m, 段落定义字体大小为 14
像素, 颜色为暗红色</p>
</div>
<div class="b">
<p>用 class 选择器定义的样式 b, 设定区域宽度为 400 像素, 高度为 200 像素</p>
<p>在该区域中, 定义所有的标签 p 的显示样式为文字大小 18 像素, 文字加粗显示, 颜色为蓝色</p>
</div>
<p class="c">该段落中, 调用了 class 类型的选择器定义的样式 c, 文字设定 24 像素, 红色, 字体
为楷体</p>
<div id="a" class="c">
该区域中, 同时调用 id 类型的选择器和 class 类型的选择器, 在样式不冲突的情况下, 它们定义的
样式将同时作用在该区域。
</div>
</body>
</html>

```

运行效果如图 5.2 所示。

5.2.4 CSS 注释

为了方便自己或者他人更好地理解 CSS 代码, 便于维护和应用, 通常会在样式表规则中添加注释。下面是一个添加了注释的样例。

```

/* -----
以下为导航条的样式定义
----- */
#nav
{
    width: 600px; !important           /* IE6 以下宽度为 600 像素 */
    width: 500px;                       /* IE5 以下宽度为 500 像素 */
}

```

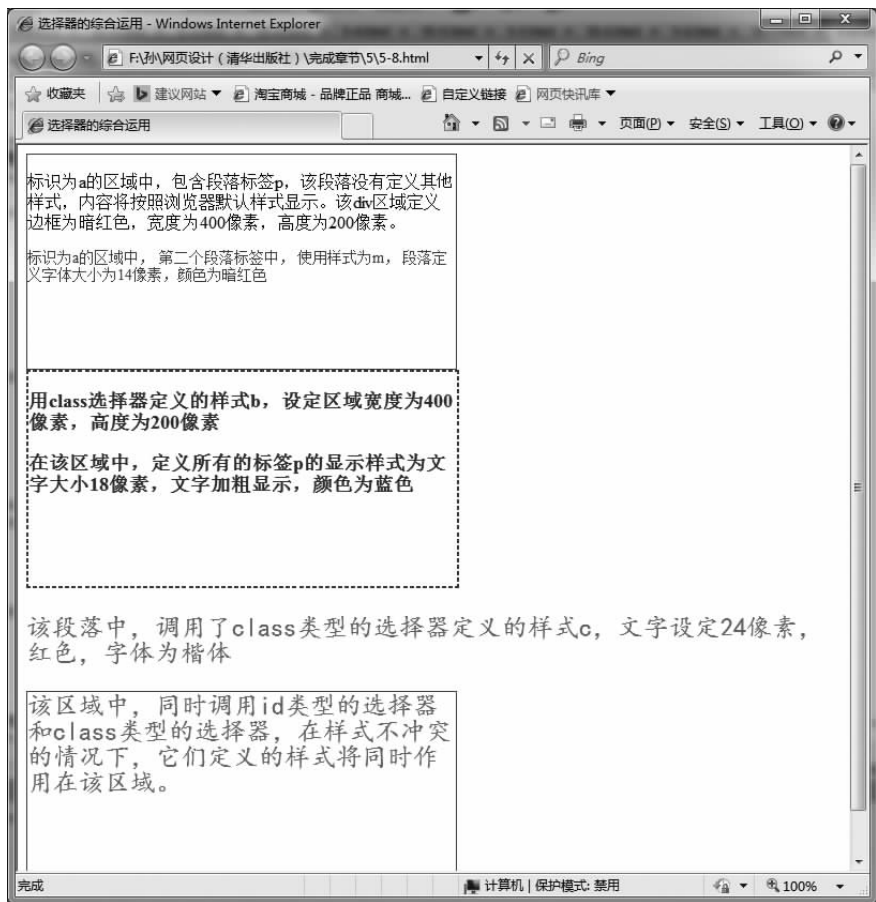


图 5.2 选择器综合运用案例的运行效果

注意：在 CSS 中只有 `/ * ... */` 注释方式，它可以单行也可以跨行注释。

5.3 CSS 样式表

对 CSS 有了大致了解以后，便可以使用 CSS 对页面进行全方位的控制。本节主要介绍使用 CSS 样式表的几种方法，包括行间样式表、内部样式表、外部样式表和导入外部样式表等。

5.3.1 行间样式表

行间样式表是指将 CSS 样式编码写在 XHTML 标签中，格式如下：

`<标记 style="样式属性: 属性值; 样式属性: 属性值; ...">`

语法说明如下。

- (1) 标记指 XHTML 标签，如 `body`、`table`、`p` 等。
- (2) 行间样式表是由 XHTML 元素的 `style` 属性支持的，它的定义只能影响标记本身。